

PYTHON DATA SCIENCE

ADRIENNE HAWKES MATTHES
AND JOHN TACKE



CODING WITH PYTHON

ADRIENNE HAWKES MATTHES
AND JOHN TACKE



PYTHON CRASH COURSE

ADRIENNE HAWKES MATTHES
AND JOHN TACKE



ADRIENNE HAWKES MATTHES AND JOHN TACKE

Python

PROGRAMMING

3 Manuscripts

THIS BOOK INCLUDES
CRASH COURSE
CODING WITH PYTHON
DATA SCIENCE

THE STEP BY STEP GUIDE FOR BEGINNERS TO MASTER
SOFTWARE PROJECTS, ALGORITHMS, TRICKS AND TIPS



PYTHON DATA SCIENCE

ADRIENNE HAWKES MATTHES
AND JOHN TACKE

3

CODING WITH PYTHON

ADRIENNE HAWKES MATTHES
AND JOHN TACKE

2

PYTHON CRASH COURSE

ADRIENNE HAWKES MATTHES
AND JOHN TACKE

1

ADRIENNE HAWKES MATTHES AND JOHN TACKE

Python

PROGRAMMING

3 Manuscripts

THIS BOOK INCLUDES
CRASH COURSE
CODING WITH PYTHON
DATA SCIENCE

THE STEP BY STEP GUIDE FOR BEGINNERS TO MASTER
SOFTWARE PROJECTS, ALGORITHMS, TRICKS AND TIPS



PYTHON PROGRAMMING

3 MANUSCRIPTS

This book includes

CRASH COURSE

CODING WITH PYTHON

DATA SCIENCE

THE STEP BY STEP GUIDE FOR

BEGINNERS TO MASTER

SOFTWARE PROJECTS,

ALGORITHMS, TRICKS AND TIPS

Adrienne Hawkes Matthes and John Tacke

PYTHON CRASH COURSE

Introduction

Chapter 1 - What Is Python and Why Should You Learn It

Chapter 2 - Variables and Operators

Chapter 3 - Learn About Simple Data Types

Chapter 4 - Conditional Statements in Python and Control Flow Statements

Chapter 5 - Working With Functions

Chapter 6 - Object-Oriented Programming

Chapter 7 - Working with Files

Chapter 8 - Python From Scratch

Chapter 9 – How Coding Works

Chapter 10 - Python Libraries

Chapter 11 – Getting Program Functioning to Work

Conclusion

CODING WITH PYTHON

Introduction

Chapter 1 Python Programming Basics

Chapter 2 - What Is Scikit-Learn, and Why Should I Learn About It?

Chapter 3 - Essential Libraries And Tools In Python

Chapter 4 – Learn Modules

Chapter 5 - Machine Learning Datasets

Chapter 6 - Training Simple Machine Learning Algorithms for Classification

Chapter 7 - Unsupervised Machine Learning

Chapter 8 - Learning in Artificial Neural Networks

Chapter 9 - The Perceptron

Chapter 10 - Execution and Repetitive Tasks

Chapter 11 - Variables and Strings

[Conclusion](#)

[**PYTHON DATA SCIENCE**](#)

[Introduction](#)

[Chapter 1 - What Is Data Science And Why Should You Learn It Using Python](#)

[Chapter 2 – Statistics and Probability.](#)

[Chapter 3 - Work With Numpy.](#)

[Chapter 4 – Work With Pandas](#)

[Chapter 5 - Learning About Functions](#)

[Chapter 6 - Developing a Machine Learning Model with Python](#)

[Chapter 7 - Identifying the Nearest Neighbors](#)

[Chapter 8 - Deep Learning vs Machine Learning](#)

[Chapter 9 - Other Basics of the Python Code](#)

[Chapter 10 - Modules](#)

[Chapter 11 - Applications of Big Data Analysis](#)

[Conclusion](#)

PYTHON CRASH COURSE

**AN INTRODUCTION GUIDE WITH
FUNDAMENTALS OF COMPUTER
SCIENCE FOR TOTAL BEGINNERS
WITH HANDS-ON PROJECTS,
TRICKS AND TIPS TO LEARN
FAST CODING CONCEPTS,
TECHNIQUES AND TOOLS**

Adrienne Hawkes Matthes and John Tacke

Introduction

Since computers were first invented, many people have had the job to make them more useful. To make this happen, there are a ton of programming tools that have been developed over the years to turn the instructions the computer is given into code that the computer can execute. When this first started, assembly language was the main type. While it did allow you to create programming that was tailored and then optimized for the low memory systems of that time, it was hard to learn, read, and maintain and only a few people could use it.

As computer technology grew and many systems became more complex, the programming languages needed to change as well. Some higher-level languages including Lisp, Cobol, and Fortran were produced to produce punch cards which the computers were able to read and then execute the instructions that were found on the card. Programming with this kind of tool was a lot easier than the languages in the past, allowing more people to write the complex programs they needed.

As time started to go on, there were a lot of improvements that were made to some of the technology that we are working with now, and it has resulted in a lot of better storage methods when it comes to our programs and more. For example, some of those punch cards have been replaced with a type of magnetic tape and then it was all moved over to work with a disc drive. As these new inventions came out, the tools to work on these programs improved as well.

Basic was one of the first coding languages that was designed to be easy to read through and this one was developed in the 1960s. then in the 1970s, we saw the introduction of Pascal and C, and these became tools that helped us to get a really simple and structured kind of programming language that would help a programmer to get the efficient code that they want. You will find that the structured languages like these would not have to rely on the goto statement found in some of the other languages. Instead, these programs would rely on a flow that has features including conditional statements, functions, and loops. These features were nice because they would allow a new programmer to make their own applications with the help of standalone reusable routines, which would then be able to lower the

amount of time that it would take to develop the program, and even debug it so that the program would work the way that you would like.

It was in the 1980's when the next big advancement in programming happened. This was where C++, and a few other languages, turned into OOP languages, or object-oriented programming languages. There are a number of these that are now available and usable, but they were useful because they helped us to hold onto our information better. While these tools of OOP languages were powerful and had some new capabilities when it came to coding, many times they would be seen as special because they could cut out some of the challenges that came with coding. As the use and the popularity of computers started to grow, and it becomes common for more people to use them, it was also an important factor to have a language for programming that was easy to use, one that we can use on a lot of different platforms. Because of this need, the language that we know as Python today was developed.

Python is going to be a general-purpose programming language. Whether you are a beginner or someone who has been in the coding world for some time, you will find that this language is easy to learn while still making sure that there is a lot of power behind it to get those codes done. You are able to use this for a lot of different purposes, from some general housekeeping tasks for your system to just having fun and making your programs and games. Of course, when you compare this language to some of the others out there, you will also notice that Python is going to have a lot of rules and structures that you must follow to get this to work. No matter what codes you would like to write in Python, you will need to follow the rules and ensure that the compiler is ready to handle it all. The good news is that the rules are pretty simple with this language and you will find that this language is simple, powerful, and compact all in one. As a beginner, it is easier to catch onto this kind of coding language than you think, and before long, and with the help of this guidebook, you will be able to get everything to work the way you would like and will be writing your codes in no time.

In the past, a lot of people were worried about learning a coding language. They worried that these languages were too tough to learn, that they would just get frustrated, and that only those who had spent their whole lives around computers could even attempt to write their codes. And maybe with some of the older codes, this was true. Thanks to a lot of the newer codes

that have been introduced recently, the idea that only those gifted in computer programming could code has faded away. With many of the codes that are coming out now, including Python, anyone can learn a few of the syntaxes for what they want to do, or even find some premade codes online and make some changes. And since many of these codes are open-sourced, it is easier than ever to learn how to use them and develop the codes to meet your needs.

You will find that many of the modern languages that are used for coding are going to be a lot better and easier to use than what we were able to find in the past with most coding languages. Gone are the days that even professionals would run into troubles regularly when it was time to find the bugs in the system. Now it is possible for anyone and everyone to learn how to use this coding language for these needs. And this is mainly because we have a lot of great OOP languages to work within Python.

Chapter 1 - What Is Python and Why Should You Learn It

You will learn everything you need to learn about a list in Python programming. Lists give you the opportunity to store a particular set of information in a place irrespective of the number of items in such a list. They are important features in Python that are accessible to programmers.

Definition of List

A list refers to a data type that contains a collection of items in an orderly manner. For instance, a list can comprise the alphabets, digits, or names of the street in a country. Below is the syntax structure for a list:

```
list = [item1, item2, item3, itemN]
```

Consider the list of items in a kitchen in this code.

```
kitchen_item = ["pot," "kettle," "fridge," "dishwasher," "microwave,"  
"knife," "bowls"]  
print(kitchen_item)
```

If you decide to print this code, the output will be as follows:

```
["pot," "kettle," "fridge," "dishwasher," "microwave," "knife," "bowls"]
```

Accessing a List Item

Perhaps, you don't want the users to see certain items in a list; you can decide to access only the items you want. To have access to a particular element in a list, you have to write the list name then followed by the index of the item, which is enclosed in squared brackets. For instance, we can pull the third item in our list of items in the kitchen.

```
kitchen_item = ["pot," "kettle," "fridge," "dishwasher," "microwave,"  
"knife," "bowls"]  
print(kitchen_item [3])
```

The output will be as follow:

```
fridge
```

```
dishwasher
```

Index Positions

The first index position in Python is not 1 but 0. If you observe in the example above, you will see that when indexed the third item, we started counting from 0. You can see that pot (0), kettle (1), fridge (2), while dishwasher (3). Peradventure, you run a Python program and get a different result; you have to check if you are making a number error.

Always remember that the second element in any list starts with an index of 1. A simple way of knowing what the particular index number will be is to substrate one from the item you want to index. Does that sound simple?

```
kitchen_item = ["pot," "kettle," "fridge," "dishwasher," "microwave,"  
"knife," "bowls"]  
print(kitchen_item)
```

For instance, we want to access only the item "microwave. To do that, you have to count the number of items to get to "microwave" and subtract one. from the code, if you count first, you will discover that "microwave" is the 5th element. However, to index it to print only "microwave," you have to substrate one from the element. Your code will be like this:

```
kitchen_item = ["pot," "kettle," "fridge," "dishwasher," "microwave,"  
"knife," "bowls"]  
print(kitchen_item [4])
```

Let us use another example to index the third and sixth items.

```
kitchen_item = ["pot," "kettle," "fridge," "dishwasher," "microwave,"  
"knife," "bowls"]  
print(kitchen_item [3])  
print(kitchen_item [6])
```

The output will be as follows:

```
dishwasher  
bowls
```

Assuming you only want to access the last element, Python has a unique way of doing that. You can do this by using an index of -1. With this, Python returns the last item.

```
kitchen_item = ["pot," "kettle," "fridge," "dishwasher," "microwave,"  
"knife," "bowls"]  
print(kitchen_item [-1])
```

In this example, the code will return "bowls." It is easy to remember this syntax. You do not have to know the particular length of the list. This also applies to other negative index values. It is like going backward. You can use the negative index to access the second or third items in a list. For instance,

```
kitchen_item = ["pot," "kettle," "fridge," "dishwasher," "microwave,"  
"knife," "bowls"]  
print(kitchen_item [-2])  
print(kitchen_item [-3])
```

The output will be as follows

```
knife  
microwave
```

Exercise to Try

- Write a Python program, which contains a list of all your special friends. Then print their names by accessing each element in your list.
- From the list, you created above, print a greeting message to each name you access in the list.
- Think of any means of transportation you can remember and write a list of items used within them. Then make a statement on some of these elements in the list and print it.

Adding, Changing, and Removing Elements in a List

The lists you will mostly create in Python are dynamic in nature. This means you have to build the list, add elements to it, and remove it in the course of running your program. For instance, you may decide to create a game where the player must shoot birds flying. You may record the first set of birds in a particular list before removing each bird that is shot down. Once a new bird appears, it is added to the screen. Your bird lists will increase, whereas the length will decrease throughout the game.

Modifying Elements

You can modify elements in an element similar to how you can access it. Furthermore, the same syntax applies to when you want to access an element. To modify an element, you will use the list name, followed by the index element that you want to modify, and then provide the new value to want to change. Well, this may sound foreign; however, let me use a simple example to illustrate how you can do this. For instance, we want to change dishwasher to spoon in the example below, how can you do that.

```
kitchen_item = ["pot," "kettle," "fridge," "dishwasher," "microwave,"  
"knife," "bowls"]  
print(kitchen_item)  
kitchen_item[3] = "spoon"  
print(kitchen_item)
```

The first two lines of code define our original list, which contains the word "dishwasher." The third line changes the element from dishwasher to spoon. If you do it correctly, you will get the output as:

```
["pot," "kettle," "fridge," "dishwasher," "microwave," "knife," "bowls"]  
["pot," "kettle," "fridge," "spoon," "microwave," "knife," "bowls"]
```

You can change any element irrespective of where it is positioned.

Adding Elements

At times, you may want to add an element in a list to expand your program. Remember the bird scenario, where you have additional new birds each time you shoot. It doesn't apply to that alone; you may want to expand your database to include new users or add new items to a list. Python provides

various means of adding elements in a list. We will focus on two ways you can do this as a beginner.

Append an Element to the List End

This method is the simplest to use when adding an element to a list. By appending the element, you add the item to the list end. Remember, we use the word “append” to add new items. However, for consistency purposes, I will use our previous `kitchen_item` example to explain. In this example, we will add cooking gas to the list.

```
kitchen_item = ["pot," "kettle," "fridge," "dishwasher," "microwave,"  
"knife," "bowls"]  
print(kitchen_item)  
kitchen_item.append("spoon")  
print(kitchen_item)
```

The `append()` method allows us to add an element at the end of the list. The output will be:

```
["pot," "kettle," "fridge," "dishwasher," "microwave," "knife," "bowls,"  
"spoon"]
```

If you have an empty list, you can utilize the `append()` method to add a new list. Assuming we want to add a new list element such as pot, kettle, and microwave, we can do that as follows:

```
kitchen_item []  
kitchen_item.append("pot")  
kitchen_item.append("kettle")  
kitchen_item.append("microwave")  
print(kitchen_item)
```

This code will produce:

```
["pot," "kettle," "microwave,"]
```

It is very common to build the list this way because you may not know the particular data type the users want to keep track of in the program. However, to give the users full control, first define an empty list, which will

hold values of the users. With this, you can use the append method to add new elements to the list created.

Inserting Elements

The second way of adding an element in a list is by inserting the element to the position you want through the insert() method. You have to specify the index of both the new element and the value, respectively. For instance, we want to insert a “spoon” after the microwave item on the list.

```
kitchen_item = ["pot," "kettle," "fridge," "dishwasher," "microwave,"  
"knife," "bowls"]  
print(kitchen_item)  
kitchen_item.insert(3, "spoon")  
print(kitchen_item)
```

Your output will be:

```
["pot," "kettle," "fridge," "dishwasher," "spoon" "microwave," "knife,"  
"bowls"]
```

What happened in the example above is that I inserted the value “spoon” at the third index. The insert method enables us to create a space between the “dishwasher” and the “microwave” while storing the value “spoon” to the location created.

Removing Elements

There are situations that may warrant you to remove an item in a list. For instance, a player may decide to remove a bird; he shoots down from the sky from the active list. Perhaps, you want to delete a customer’s account or cancel their account. You can remove such detail from the list. You can use the following two ways to remove elements in a list.

Remove Items Using del Statement

Python provides a unique way of removing or deleting an item if you know the position of such an item from the list. Look at the code below:

```
kitchen_item = ["pot," "kettle," "fridge," "dishwasher," "microwave,"  
"knife," "bowls"]
```

```
print(kitchen_item)
del kitchen_item [4]
print(kitchen_item)
```

What did you think will be deleted from the snippet above? If your answer is a dishwasher, then you are wrong. Remember that Python index begins at 0. This means we start counting 0 from the first element. Therefore, the final output will be:

```
["pot," "kettle," "dishwasher," "microwave," "knife," "bowls"]
```

You can remove any item from the position of your choice.

Using the pop() Method to Remove an Item

There are situations where you want to use an item value after removing it from the list. For instance, you want to know the coordinates of the bird you shot, to enable you to draw an explosion in such a position. The function of the pop() method is to remove the last item in the list. For clarity, let us pop an example.

```
kitchen_item = ["pot," "kettle," "fridge," "dishwasher," "microwave,"
"knife," "bowls"]
print(kitchen_item)
popped.kitchen_item = kitchen_item.pop()
print(kitchen_item)
print(popped.kitchen_item)
```

I won't explain the first two lines since you are already conversant with it. However, if you forget, I will refresh your memory. The first line begins by declaring "kitchen_item" as a variable with values including pot, kettle, fridge, dishwasher, microwave, knife, and bowls. The second line prints out the items in the kitchen_item list.

In the third line, we popped a value from our list and store the value to the new variable "popped.kitchen_item" additionally, we print out the remaining items in the kitchen_item list. From the code above, our output will be:

```
[“pot,” “kettle,” “fridge,” “dishwasher,” “microwave,” “knife,” “bowls”]  
[“pot,” “kettle,” “fridge,” “dishwasher,” “microwave,” “knife”]  
bowls
```

The first line shows the complete `kitchen_items`, whereas the second shows the remaining items after the `popped` method was used. Finally, line three shows the item removed from our list.

Chapter 2 - Variables and Operators

Understanding Python variables, classes, and how to operate is essential for both beginners and programmers who intend to expand their programming skills.

What Is A Variable in Python?

When writing complex codes, your program will demand data essential to conduct changes when you proceed with your executions. Variables are, therefore, sections used to store code values created after you assign a value during program development. Python, unlike other related language programming software, lacks the command to declare a variable as they change after being set. Besides, Python values are undefined like in most cases of programming in other computer languages.

Variation in Python is therefore described as memory reserves used for storing data values. As such, Python variables act as storage units, which feed the computer with the necessary data for processing. Each value comprises of its database in Python programming, and data are categorized as Numbers, Tuple, Dictionary and List, among others. As a programmer, you understand how variables work and how helpful they are in creating an effective program using Python. As such, the tutorial will enable learners to understand declare, re-declare, and concatenate, local and global variables as well as how to delete a variable.

Variable Vs. Constants

Variables and constants are two components used in Python programming but perform separate functions. Variables, as well as constants, utilize values used to create codes to execute during program creation. Variables act as essential storage locations for data in the memory, while constants are variables whose value remains unchanged. In comparison, variables store reserves for data while constants are a type of variable files with consistent values written in capital letters and separated by underscores.

Variables Vs. Literals

Variables also are part of literals which are raw data fed on either variable or constant with several literals used in Python programming. Some of the common types of literals used include Numeric, String, and Boolean, Special and Literal collections such as Tuple, Dict, List, and Set. The difference between variables and literals arises where both deal with unprocessed data but variables store the while laterals feeds the data to both constants and variables.

Variables Vs. Arrays

Python variables have a unique feature where they only name the values and store them in the memory for quick retrieval and supplying the values when needed. On the other hand, Python arrays or collections are data types used in programming language and categorized into list, tuple, set, and dictionary. When compared to variables, the array tends to provide a platform to include collectives functions when written while variables store all kinds of data intended. When choosing your charming collection, ensure you select the one that fits your requirements henceforth meaning retention of meaning, enhancing data security and efficiency.

Classifications of Python Arrays Essential for Variables

Lists

Python lists offer changeable and ordered data and written while accompanying square brackets, for example, "an apple," "cherry." Accessing an already existing list by referring to the index number while with the ability to write negative indexes such as '-1' or '-2'. You can also maneuver within your list and select a specific category of indexes by first determining your starting and endpoints. The return value with therefore be the range of specified items. You can also specify a scale of negative indexes, alter the value of the current item, loop between items on the list, add or remove items, and confirming if items are available.

Dictionaries

Python dictionaries comprise of indexed, changeable but unordered items typically written while with curly brackets with keys and values. Some of the activities involved include item access by use of a keyword inside the parentheses; conduct value changes, loop, check critical availability, length

of the dictionary, and both adding and removing unwanted items. Besides, Python allows you to copy the dictionary by writing 'dict2 = dict1'. 'dict2' will become a representation to 'dict1' therefore makes any necessary changes automatically. Another way of creating a copy is also by using a built-in Dictionary technique, that is, 'copy.'

In other instances, Python dictionaries can also have other dictionaries within it a process referred to as nested dictionaries. You can readily determine the number of dictionaries present in the nest through creating of three already available. You can also generate your dictionary through the 'dict()' contractor function. The function enables the copying of the dictionary or the creation of a completely new one. Within the Python dictionary, there exist several built-in techniques to implement and enjoy the efficiency of the dictionaries present.

Naming Variables

The naming of variables remains straightforward, and both beginners and experienced programmers can readily perform the process. However, providing titles to these variables accompany specific rules to ensure the provision of the right name. Consistency, style, and adhering to variable naming rules ensure that you create an excellent and reliable name to use both today and the future. The rules are:

- Names must have a single word, that is, with no spaces
- Names must only comprise of letters and numbers as well as underscores such as (`_`)
- The first letter must never be a number
- Reserved words must never be used as variable names

When naming variables, you should bear in mind that the system is case-sensitive, hence avoid creating the same names within a single program to prevent confusion. Another important component when naming is considering the style. It entails beginning the title with a lowercase letter while using underscores as spaces between your words or phrases used. Besides, the program customarily prevents starting the name with a capital

letter. Begin with a lowercase letter and either mix or use them consistently.

When creating variable names, it may seem so straightforward and easy, but sometimes it may become verbose henceforth becoming a disaster to beginners. However, the challenge of creating sophisticated names is quite beneficial for learned as it prepares you for the following tutorials. Similarly, Python enables you to write your desired name of any length consisting of lower- and upper-case letters, numbers as well as underscores. Python also offers the addition of complete Unicode support essential for Unicode features in variables.

Specific rules are governing the procedure for naming variables; hence adhere to them to create an exceptional name to your variables. Create more readable names that have meaning to prevent instances of confusion to your members, especially programmers. A more descriptive name is much preferred compares to others. However, the technique of naming variables remains illegible as different programmers decide on how they are going to create their kind of names.

Methods of Creating a Multi-Name for Python Variables

- Pascal case: this method entails the first, second, and subsequent words in the name as capitalized to enhance readability. For example, ConcentrationOfWhiteSmoke.
- Camel case: the second and subsequent words of the name created remains capitalized. For example, the ConcentrationofWhiteSmoke.
- Snake case: snake method of creating variable names entails separator of words using an underscore as mentioned earlier. For example, concentration_of_white_smoke.

Learning Python Strings, Numbers and Tuple

Python strings are part of Python variables and comprise of objects created from enclosing characters or values in double-quotes. For example, 'var = Hello World'. With Python not supporting character types in its functions, they are however treated as strings of one more characters as well as substrings. Within the Python program, there exist several string operators

making it essential for variables to be named and stored in different formats. Some of the string operators commonly used in Python are [], [:], 'in', r/R, %, + and *.

There exist several methods of strings today. Some include replacing Python string () to return a copy of the previous value in a variable, changing the string format, that is, upper and lower cases and using the 'join' function, especially for concatenating variables. Other methods include the reverse function and split strings using the command 'word.split'. What to note is that strings play an important role, especially in naming and storage of values despite Python strings being immutable.

On the other hand, Python numbers are categorized into three main types; that is, int, float, and complex. Variable of numbers are usually created when assigning value for them. For instance, int values are generally whole numbers with unlimited length and are either positive or negative such as 1, 2, and 3. Float numbers also either positive or negative and may have one or more decimals like 2.1, 4.3 and 1.1 while complex numbers comprise both of a letter 'j' as the imaginary portion and numbers, for example, 1j, -7j or 6j+5. As to verify the variable number is a string, you can readily use the function 'type().'

A collection of ordered values, which remain unchangeable especially in Python variables, is referred to as a tuple. Python tuples are indicated with round brackets and available in different ways. Some useful in Python variables are access tuple items by index numbers and inside square brackets. Another is tuple remaining unchanged, especially after being created but provides a loop by using the function 'for.' And it readily encompasses both count and index methods of tuple operations.

Types of Data Variables

String

A text string is a type of data variable represented in either String data types or creating a string from a range of type char. The syntax for string data comprises multiple declarations including 'char Str1[15]', 'char Str5[8] = "ardiono"; among others. As to declare a string effectively, add null character 'Str3', declare arrays of chars without utilizing in the form of 'Str1' and initialize a given array and leave space for a larger string such as Str6.

Strings are usually displayed with double quotes despite the several versions of available to construct strings for varying data types.

Char

Char are data types primarily used in variables to store character values with literal values written in single quotes, unlike strings. The values are stored in numbers form, but the specific encoding remains visibly suitable for performing arithmetic. For instance, you can see that it is saved as 'A' +, but it has a value of 66 as the ASCII 'A' value represents 66. Char data types are usually 8 bits, essential for character storage. Characters with larger volumes are stored in bytes. The syntax for this type of variable is 'char var = val'; where 'var' indicates variable name while 'val' represents the value assigned to the variable.

Byte

A byte is a data type necessary for storing 8-bit unsigned numbers that are between 0 to 255 and with a syntax of 'byte var = val;'. Like Char data type, 'var' represents variable name while 'val' stands for the value to be assigned that variable. The difference between char and byte is that char stores smaller characters and with a low space volume while byte stores values which are larger.

int

Another type of data type variable is the int, which stores 16-bit value yielding an array of between -32,768 and 32,767, which varies depending on the different programming platforms. Besides, int stores 2's complement math, which is negative numbers, henceforth providing the capability for the variable to store a wide range of values in one reserve. With Python, this type of data variable storage enables transparency in arithmetic operations in an intended manner.

Chapter 3 - Learn About Simple Data Types



Numbers

As indicated, Python accommodates floating, integer and complex numbers. The presence of a decimal point separates integers and floating points. For instance, 4 is integer while 4.0 is a floating point number.

On the other hand, complex numbers in Python are denoted as $r+tj$ where j represents the real part and t is the virtual part. In this context the function `type()` is used to determine the variable class. The Python function `instance()` is invoked to make a determination of which specific class function originates from.

Example:

Start IDLE.

Navigate to the File menu and then click New Window.

Type the following:

```
number=6
```

```
    print(type(number)) #should output class int
```

```
print(type(6.0)) #should output class float
```

```
complex_num=7+5j
```

```
print(complex_num+5)
```

```
print(isinstance(complex_num, complex)) #should output True
```

Important: Integers in Python can be of infinite length. Floating numbers in Python are assumed precise up to fifteen decimal places.

Number Conversion

This segment assumes you have prior basic knowledge of how to manually or using a calculator to convert decimal into binary, octal and hexadecimal. Check out the Windows Calculator in Windows 10, Calculator version Version 10.1804.911.1000 and choose programmer mode to convert automatically.

Programmers often need to convert decimal numbers into octal, hexadecimal and binary forms. A prefix in Python allows denotation of these numbers to their corresponding type.

Number System Prefix

Octal '0O' or '0o'

Binary '0B' or '0b'

Hexadecimal '0X' or '0x'

Example

```
print(0b1010101) #Output:85
```

```
print(0x7B+0b0101) #Output: 128 (123+5)
```

```
print(0o710) #Output:710
```

Practice Exercise

Write a Python program to display the following:

- 0011 11112
- 747
- 9316

Type Conversion

Sometimes referred to as coercion, type conversion allows us to change one type of number into another. The preloaded functions such as `float()`, `int()` and `complex()` enable implicit and explicit type conversions. The same functions can be used to change from strings.

Example:

Start IDLE.

Navigate to the File menu and then click New Window.

Type the following:

```
int(5.3) #Gives 5
```

```
int(5.9) #Gives 5
```

The `int()` will produce a truncation effect when applied to floating numbers. It will simply drop the decimal point part without rounding off. For the `float()` let us take a look:

Start IDLE.

Navigate to the File menu and then click New Window.

Type the following:

```
float(6) #Gives 6.0
```

```
ccomplex('4+2j') #Gives (4+2j)
```

Practice Exercise

Apply the `int()` conversion to the following:

✓ 4.1

✓ 4.7

✓ 13.3

✓ 13.9

Apply the float() conversion to the following:

✓ 7

✓ 16

✓ 19

Decimal in Python

Example

Start IDLE.

Navigate to the File menu and then click New Window.

Type the following:

```
(1.2+2.1)==3.3 #Will return False, why?
```

Discussion

The computer works with finite numbers and fractions cannot be stored in their raw form as they will create infinite long binary sequence.

Fractions in Python

The fractions module in Python allows operations on fractional numbers.

Example

Start IDLE.

Navigate to the File menu and then click New Window.

Type the following:

```
import fractions
```

```
print(fractions.Fraction(2.5)) #Output 5/2
```

```
print(fractions.Fraction(4)) #Output 4
```

```
print(fractions.Fraction(2,5)) #output 2/5
```

NOTE

Creating my_fraction from float can lead to unusual results due to the misleading representation of binary floating point.

Mathematics in Python

To carry out mathematical functions, Python offers modules like random and math.

Start IDLE.

Navigate to the File menu and click New Window.

Type the following:

```
import math
print(math.pi) #output:3.14159....
print(math.cos(math.pi)) #the output will be -1.0
print(math.exp(10)) #the output will be 22026.4....
print(math.log10(100)) #the output will be 2
print(math.factorial(5)) #the output will be 120
```

Practice Exercise

Write a python program that uses math functions from the math module to perform the following:

- ✓ Square of 34
- ✓ Log1010000
- ✓ Cos 45 x sin 90
- ✓ Exponent of 20

Before tackling flow control, it is important we explore logical operators.

Comparison operators are special operators in Python programming language that evaluate to either True or False state of the condition.

Program flow control refers to a way in which a programmer explicitly specifies the order of execution of program code lines. Normally, flow control involves placing some condition (s) on the program code lines. In this phase, we will explore and test various flow controls in Python.

Chapter 4 - Conditional Statements in Python and Control Flow Statements

The next type of coding that we need to spend some time looking at is the conditional statements. These are going to be an important part of the code that we work with because they are going to ensure that your system is able to respond to the kind of input that the user provides to you. It is hard to predict how a user is going to work with the system, but you can set up some of the conditions that you would like it to look at, and work from there to come up with the way your program is going to work.

As we can imagine here, it is pretty much impossible for a programmer to create something and guess ahead of time what answers or input the user is going to provide to the program. And the programmer can't be there watching each use work with the program either, which means that they need to work with the conditional statements. When these are set up in the right manner, it will ensure that the program is going to run in the proper manner, and will respond to any information that the user is providing to you.

There are a lot of different types of programs that are going to respond well to the conditional statements that we will talk about in this guidebook. These are actually pretty simple to work with, and we will take a look at some of the examples of how you can code with these conditional statements as well.

As we go through this section, we are going to take a look at the three main types of conditional statements including the if statement, the if else statement, and the elif statement. Let's take a look on how each of these three is going to work and when we would be able to use these conditional statements.

The If Statement

As we mentioned, there are three types of conditional statements that we are able to take a look at. And the first one that we need to explore a bit is the if statement. Out of the three that we will spend some time on, the if statement is going to be one of the most basic out of them. These are not going to be

used as much as some of the other options because they often leave a bit to be desired. But they are a good springboard for helping us to learn what these conditional statements are about and how we are able to work with them.

With the if statement, the program is set up so that it can only proceed forward if the user provides us with an input that works with the conditions that we set ahead of time. If the input that we get from the user isn't able to match up with our conditions, then the program will just stop and nothing is going to happen.

As we can see already, there are going to be some issues with this because we don't want the program to just stop with the answer, but it still provides us with some of the basis that we need.

With this in mind, we can work with a simple code that we are able to work with that shows us how the, if the statement is going to work, will be found below:

```
age = int(input("Enter your age:"))
if (age <=18):
print("You are not eligible for voting, try next election!")
print("Program ends")
```

There are a few things that will show up with this code. If you have a user go to the program and state that their age is under 18, then the program will work and display the message that is listed there. The user can read this message and then end the program right there.

But, things can go wrong if the user puts in that their age is above 18. This is true for the user, but because it doesn't meet the conditions that you wrote out, the program will see it as false. As the code is written right now, nothing is going to happen because it isn't set up to handle this. The user will just see a blank screen any time they enter an age that is over 18.

The If Else Statement

Now that we have had some time to take a look at the simple if statement, it is time for us to move on to the if else statement. The if the statement is going to be a good way to get a bit of practice with coding, but there are not

going to be all that many times when we are programming where we are going to need to work with this kind of statement.

You want to make sure that when your user works with the program, no matter what input they use, that something shows up on the screen.

If you use the if statement, like what we see in the example above, and the user puts in an answer that is above 18, using the code that we had from before, then the screen is just going to come back blank. This is definitely not something that we want to see, so we need to move on to the if else statement to see what we are able to do regardless of what information the user puts into the program.

The if else statement is going to provide us with an output, and will ensure that we provide these outputs to the user, regardless of what age, or other information, that we provide to the program. So, with the example above, if the user comes in and says that they are 40, then the code will still be able to respond to it.

There are a few options that you are able to use with this one, but with the idea of the voting option that we talked about with the if statement, we are able to use the following syntax to help us to turn it into an if else statement to help us out:

```
age = int(input("Enter your age:"))
if (age <=18):
print("You are not eligible for voting, try next election!")
else
print("Congratulations! You are eligible to vote. Check out your local
polling station to find out more information!")
print("Program ends")
```

With this option, you are adding in the else statement, which will cover every age that doesn't fall under 18. This way, if the user does list that as their age, something will still pull up on the screen for them. This can provide you with more freedom when working on your code and you can even add in a few more layers to this. If you want to divide it up so that you get four or five age groups and each one gets a different response, you

simply need to add in more if statements to make it happen. The else statement is at the end to catch everything else.

For example, you can take the code above and ask the user what their favorite color is. You could then have if statements to cover some of the basic colors, such as red, blue, green, yellow, orange, purple, and black.

If the user puts in one of those colors than the corresponding statement will show up on the screen. The else statement will be added in to help catch any other colors that the person may try to use, such as pink or white.

The Elif Statements

The third type of conditional statement that we are able to work within this process is known as the elif statement.

These are going to help us to add on another level to what we did with the other step, but they are still going to make sure that the codes we write are as easy as possible.

We are able to create as many of these elif statements as possible in the code, as long as we make sure to add in that else statement at the end. This else statement ensures that we are able to handle any of the other answers that the user puts in, the ones that we may not have thought about ahead of time.

When working with the elif statement, it is going to be similar to giving the user a menu to pick from. You can choose how many of these elif statements you would like to have present in the menu, similar to what is found in a lot of games, and then the user can pick and choose which one they would like to work with. You can then have a certain action happen, or a certain statement shows up on the program to meet your needs.

Another thing to notice with the elif statement is that you can add in as many different options as your code needs. It is possible to make a small menu that just has two or three items in it, or it is possible to expand this out to as many of these as you need to make the code work properly.

The fewer options that you work within this one though, the easier your code writing will so keep that in mind when determining how many options are actually needed.

Now that we know a little bit about these elif statements and how they work, let's dive in and take a look at a good example of one that you are able to write out. Open up your compiler and type in the following code:

```
Print("Let's enjoy a Pizza! Ok, let's go inside Pizzahut!")
print("Waiter, Please select Pizza of your choice from the menu")
pizzachoice = int(input("Please enter your choice of Pizza:"))
if pizzachoice == 1:
    print('I want to enjoy a pizza napoletana')
elif pizzachoice == 2:
    print('I want to enjoy a pizza rustica')
elif pizzachoice == 3:
    print('I want to enjoy a pizza capricciosa')
else:
    print("Sorry, I do not want any of the listed pizza's, please bring a Coca Cola for me.")
```

With this option, the user is able to choose the type of pizza they want to enjoy, but you can use the same syntax for anything you need in your code. If the user pushes the number 2 in the code, they are going to get a pizza rustica. If they don't like any of the options, then they are telling the program that they just want to have something to drink, in this case, a Coca Cola.

Control Flow

The control flow in a program highlights the order of program execution. In a Python program, control flow is carried out by function calls, conditional statements, and loops. This will deal with the If Statement, While, and for loops.

Chapter 5 - Working With Functions

Functions in any programming language are important because they help divide a large program into small sections that are easy to control. Functions are still important if you experience a situation where you write the same code at different points in your program. Python function blocks starts with the keyword `def` followed by the name of the function and parentheses. Any input arguments or parameters should be placed within these parentheses. Still, you can define parameters inside these parentheses.

Why You Need Functions

Functions control inputs and outputs in computer programs. Programming languages have been created to work on data and the functions are the best way to convert this type of data.

In particular, the main code of a program is a function. Every function is logically connected to work from the main code. However, if a function has not been previously defined, you will need to define one before you use it. So, functions can be said to be tasks that a user wants to conduct. However, defining a function once with a name provides the chance to reuse the functionality without making the main program to appear scary. This will rapidly eliminate the lines of code and make the debugging easier.

For now, you need to note that the reason why you use a function is because of the reusability. The fact that complicated operations can be compiled as singular tasks that can run with only a call is the reason why computer codes are clearer. Each language of programming allows you to create and use the functions to conduct different tasks with a single call. And you can make various calls without getting worried about logically structuring the code into the main code every single moment.

Python Functions

The Python functions are a great example of reusability. This helps serve a wide range of applications right from web development to testing. The Python interpreter has different functions that are available to use. And you can always bring in other libraries to your program that has pre-defined functions already available to use. All that you need to do is to download

the necessary packages and freely provide the relevant functionalities by importing them to your code. function, you only need the name, its arguments, and purpose if it takes any and the type of results if returns any.

It is the same as using an automobile or telephone where you don't need to know the working of its components to use them. Instead, it is already built to provide common purposes that you can directly use to attain your goals and devote your time to implement all innovative features of your application program. And no one really wants to understand the inner operation of a function as long as it does its work.

Therefore, with Python, unless you want to create a new function or change existing works, you don't need to know anything that happens on the inside until it works the way you need it to. It is like a vehicle, where you will be required to understand how it works before you build or fix one. Alternatively, once you create a working function, you can use it repeatedly without the need to look at the contents inside it ever again.

A function in Python may have a different number of parameters or none at all. Therefore, for the moments when you need your function to work on variables from other blocks of code or your main program, it may involve a different number of parameters and generate results. A python function can still return a value or not. The value can be generated from the function's execution or even an expression or value that you emphasize after the keyword "return". And once a return statement is executed, the program flow returns to the state next to the function call and gets executed from there.

Therefore, to call a Python function at any point in your code, you will only need to use its name and pass arguments in the parentheses, if any.

The rules for defining a function are similar to naming a variable. It starts with either letter from A-Z, a-z in both lower cases, and upper cases, or an underscore (_). The remaining name has underscores (_), digits (0-9), and letters in lowercase or upper case.

1. A reserved keyword might not be selected as an identifier.
2. The right use of grammar to improve the readability of the code.

It is the right practice to name a Python function based on what it achieves. Apply a docstring below the first line of a function declaration. It is a documentation string, and it describes what the function implements.

Basics of Functions

To define functions in Python, you use the `def` statement. This statement ends with a colon, and the code inside the function is indented just below the `def` statement. For example:

```
def print_hello():
    print('Hello!')

print_hello()
print('1234567')
print_hello()
```

Let's look at this program in detail.

From the following program, the last part of the code has a function call.

Functions are vital because they help you avoid the need to write the same code everywhere.

If you want to draw a box of stars in your program, a Python function can be helpful. Whenever you want a box, you simply insert the function to draw the box instead of typing different lines of

```
def draw_square():
    print('*' * 15)
    print('*', ' *11', '*')
    print('*', ' *11', '*')
    print('*' * 15)
```

redundant code. Below is

the function:

The advantage this brings is if you want to alter the size of the box, you only need to alter the code inside the function. On the other hand, if you try to copy and paste the box code everywhere, then you will require to change every place you paste it.

Arguments

In most programming languages, one can pass values to functions. For

```
def print_hello(n):  
    print('Hello ' * n)  
    print()  
  
print_hello(3)  
print_hello(5)  
times = 2  
print_hello(times)
```

example:

By calling the print_hello function using the value 3, the value is kept in the variable n. You can then point to this variable within the function's code.

In Python programming, it is possible to pass a different value to the same function.

```
def multiple_print(string, n)  
    print(string * n)  
    print()  
  
multiple_print('Hello', 5)  
multiple_print('A', 10)
```

Returning values

Programming is an interesting thing to learn, especially when writing functions. Besides just passing values to functions, you can also create functions that return a result after computation is over.

```
def convert(t):  
    return t*9/5+32  
print(convert(20))
```

The following example changes the temperature from Celsius to Fahrenheit.

This program has a return statement that sends the output of a function computation back to the caller. Keep in mind that the function doesn't print anything. The printing happens outside the function.

```
print(convert(20)+5)
```

In this case, you can compute the math with the following result:

The math module in Python has trig functions but the functions only work

```
from math import pi, sin  
def deg_sin(x):  
    return sin(pi*x/180)
```

in radians. Here is a sine function that can work in degrees.

A function can still return more than one value as a list.

Types of Python Functions

There are many categories of Python Functions. And each of the functions is unique. Here are different categories of Python functions:

- Python Recursion Functions
- Python Built-in Functions
- Python user-defined functions

- Python lambda functions

Let us dive deep and learn more about these functions.

In-built Python Functions

The python interpreter has different functions available to use. These functions are referred to as built-in functions. For instance, the print () function will display the object to the standard output device or the text stream file.

Python 3.6 has 68 built-in functions. For purposes of simplicity, let us look at the main functions used, and we can start on from there.

Python abs () Function

This function will return the absolute value of any number entered. If the number is complex, abs () displays a magnitude.

Syntax

The general syntax of the abs () function is:

Abs (num)

Parameters

The abs () function accepts a single argument:

- Num-This is a number whose absolute value should be returned. This number can be:
 - a. Integer
 - b. Complex number
 - c. Floating number

Example

```
# random integer
integer = -20
print('Absolute value of -20 is:', abs(integer))

#random floating number
floating = -30.33
print('Absolute value of -30.33 is:', abs(floating))
```

Python all () function

The all () function will output True if all elements in a specific iterable are true. If not, it will display False.

Syntax

The syntax of all () method is:

all (iterable)

Parameters

The all () method accepts a single parameter:

- Iterable- Any iterable (tuple, list, dictionary, etc.) which has the elements.

```
# all values true
l = [1, 3, 4, 5]
print(all(l))

# all values false
l = [0, False]
print(all(l))

# one false value
l = [1, 3, 4, 0]
print(all(l))

# one true value
l = [0, False, 5]
print(all(l))

# empty iterable
l = []
print(all(l))
```

Output

```
True
False
False
False
True
```

For example:

Python `ascii ()` Function

This method will return a string with a printable representation of an object. It skips the non-ASCII characters inside the string using `you`, `x`, or `U` escapes.

Syntax

The syntax of the `ascii ()` method is:

`ascii (object)`

Parameters

The `ascii` method accepts objects like list, string, etc.

Example

```
normalText = 'Python is interesting'
print(ascii(normalText))

otherText = 'Pythön is interesting'
print(ascii(otherText))

print('Pythn is interesting')
```

Output

```
'Python is interesting'
'Pythn is interesting'
Pythön is interesting
```

Python `bin ()` function

The `bin` function changes and returns the binary string of a specific integer. In case the parameter is not an integer, it has to execute the `_index_ ()` method to run an integer.

The syntax of a `bin ()` method is: `bin (num)`

The `bin ()` method accepts a single parameter:

- Num-This is an integer number whose binary equivalent has to be computed. If not an integer, it has to run the `_index_ ()` method to output an integer.

```
number = 5
print('The binary equivalent of 5 is:', bin(number))
```

Output

```
The binary equivalent of 5 is: 0b101
```

Python bool () function

The bool () method will change and return the equivalent binary string of a particular integer. If the parameter is not an integer, it has to execute the `__index__ ()` to display an integer.

The syntax of bool () method is:

Bool([value])

Parameters

It is not a must to pass a value to bool (). If you don't pass a value, bool () returns a False value.

Overall, bool () accepts a single parameter value.

Python user-defined functions

As the name suggests, user-defined functions refer to functions defined by the user.

Pros of user-defined functions

1. User-defined functions assist in the sub-dividing a huge program into small parts.
2. If there is a repeated code in a program, the function can include codes and execute when required by the calling function.
3. Programmers that work on a big project can divide the workload by computing various functions.

Syntax

```
def function_name(argument1, argument2, ...) :  
    statement_1  
    statement_2  
    ....
```

In summary, the concepts explored in this phase should help you develop your own Python functions by adding operability and functionality to the same.

This will be useful when you try to build an application by simplifying the process and make it suitable for your personal needs. Now, you need to be able to use Python functions to create applications easily.

```
test = []
print(test, 'is', bool(test))

test = [0]
print(test, 'is', bool(test))

test = 0.0
print(test, 'is', bool(test))

test = None
print(test, 'is', bool(test))

test = True
print(test, 'is', bool(test))

test = 'Easy string'
print(test, 'is', bool(test))
```

Output

```
[] is False
[0] is True
0.0 is False
None is False
True is True
Easy string is True
```

Python Recursive functions

In Python programming, a function can reference another function. Also, functions don't have to be called by other functions, but instead it can call itself.

An example of a recursive function is one to find the factorial of numbers. 6 factorial means $1*2*3*4*5*6= 720$

Example:

```
# An example of a recursive function to
# find the factorial of a number

def calc_factorial(x):
    """This is a recursive function
    to find the factorial of an integer"""

    if x == 1:
        return 1
    else:
        return (x * calc_factorial(x-1))

num = 4
print("The factorial of", num, "is", calc_factorial(num))
```

In this example, `calc_factorial ()` is a recursive function.

Pros recursion

1. A recursive function improves the appearance of code. A code looks elegant and clean.
2. A difficult problem is divided into small parts using recursion

Cons of recursion

1. Recursive calls consume a lot of time and memory.
2. Sometimes the logic of recursion is difficult to follow.

Python Lambda Functions

A function without a name is referred to as an anonymous function. The `lambda` keyword denotes anonymous.

Using Lambda functions in Python

The syntax of python lambda function includes:

Lambda arguments: expression

Chapter 6 - Object-Oriented Programming

What are Classes?

Object-oriented programming refers to a programming paradigm (a philosophy of programming) that aims to break down complex problems into simple objects that interact with one another. In Python, the blueprint that objects are built from is known as a “class,” while the actual item that is instantiated with these blueprints is the “object.”

Classes are the blueprints for objects that we can use when writing programs. Classes, and the actual objects they will become, have two important components: “attributes” and “methods.” You can think of attributes as the characteristics of the class, analogous to variables in a function. Meanwhile, methods are like functions themselves, actions that the object can take (ways we can interact with that object).

To make this more concrete, let’s think about a real-world example of an instance we can create a class in. Let’s say we oversee employee documentation for a company. We want to create a class that represents employees, tracks their attributes, and allows us to update their attributes. To accomplish this, we would create an Employee class, a blueprint that will allow us to create individual objects representing individual employees. The employee class will contain attributes like name, role, email, etc., and it will also have a variety of methods that we can use to update their role should they shift jobs within the company.

The advantage of creating a class that we can use as a blueprint is that we don’t have to create a dictionary for each employee from scratch.

Creating Classes

Let’s look at how we can create classes. To create a class in Python, we simply use the class keyword to begin writing a class, followed by the name of the class we want to create. (Class names are encouraged to be in CamelCase, with the first letter of each word capitalized.)

```
class Employee:
```

```
    pass
```

Above is how we would create an Employee class, and if you're not sure of what attributes or methods you want the class to include, you can just write the keyword pass, which will have Python recognize it as a valid class that currently does nothing.

The next thing we need to do to get our class up and running is initialize the class. We do this with a constructor or init method. The init method will pass in the attributes we want the class to have, but before we do that, we need to pass in self. Why do we need to pass in a self argument to the initialization method? You can think of it as the initialization method asking what arguments it should create an instance of, and naturally, the class must construct an instance of itself. We can then go on to specify other attributes that we would like the class to have. When we create the attributes, we need to specify self for the individual attributes in the init method and then assign the values that will be passed to the object when it's created.

```
class Employee:
```

```
    def __init__(self, name, email, role):
        self.name = name
        self.email = email
        self.role = role
```

Now that the class has been created, we can go about instantiating an actual object using the class blueprint. We'll create an object by calling the class, much like how we call a function, and assigning that to a variable. We pass in arguments consisting of the values we want the class to use, and we need to be sure we pass the arguments in the order they appear above in the init method. We do not need to specify "self" as that value is used in the constructor call by default.

```
employee_1 = Employee("E. Davis", "edavis@business.com", "Hiring  
Manager")
```

```
employee_2 = Employee("D. Wong", "dwong@business.com",  
"Developer")
```

After we've created an instance of the class, we can access the individual elements of the class and update them. We access the desired attribute using dot notation and update the value of an attribute simply by assigning it a

new value. After creating the class and objects from the class using the code above, add the code below to the script and run it to see how things change.

```
print(employee_1.role)
employee_1.role = "Lead developer"
print(employee_1.role)
```

Class Variables and Instance Variables

The difference between these was called scope. Classes and the objects that are created with the class blueprints can also have different variable scopes. In this case, the difference can be described as class variables vs. instance variables.

A class variable is a variable that belongs to the entire class and each instance created with that class will share that class variable. For example, let's say we update the class blueprint we created to include another variable, but place the variable outside a method under the main scope of the class.

```
class Employee:
    location = "Seattle, WA"
    def __init__(self, name, email, role):
        self.name = name
        self.email = email
        self.role = role
```

We have created a variable outside the constructor method, and this means that the variable applies to all instances of the class that are created. If we had placed the variable inside the init method and used self, it would mean that every instance created with the class would have its unique location. Instead, as a class variable, it applies to all instances, reflecting that all the employees are in one physical location.

We can try creating instances of the class and print out their location. Notice how they are the same.

```
employee_1 = Employee("E. Davis", "edavis@business.com", "Hiring Manager")
```

```
employee_2 = Employee("D. Wong", "dwong@business.com",  
"Developer")  
print(employee_1.location)  
print(employee_2.location)
```

As for the instance variables, we've already seen how these are used. The instance variables are unique to each instance of the class, and they are passed in as arguments when the class is created.

In summary, here's a quick breakdown of the differences between class and instance variables:

- Class variables are created outside of any method, and they can be accessed outside of the class by using the name of the class. Altering the value of a class variable will change the value in all instances of the class.
- Instance variables are created within a method of the class, and they must be created with self. They must be accessed using the name of the instance, not the class, and altering the value of an instance variable will only impact that specific instance.

Class Methods and Static Methods

Since we now know how to create classes and attributes, we can discuss how to create "methods." Methods are basically functions that belong to a class. They are used to carry out specific tasks, but they can only be used directly on an object that has been created using the class blueprint. Let's try adding a method to the class we created above that can be used to print out the name, email, and role of an instance, all with one method. We define a method in a class much like how we define functions, with the def keyword and then the arguments the method takes. Similar to the init method, every method within a class will take the instance as the first argument, so we need to pass self first.

```
class Employee:  
    location = "Seattle, WA"  
    def __init__(self, name, email, role):
```

```
self.name = name
self.email = email
self.role = role
def get_info(self):
return '{} {} {}'.format(self.name, self.email, self.role)
```

Creating a method like this will enable us to get all the relevant info just by calling the method on our object instance of the class in a print statement. Note that we're returning the info in the code above, so when we choose to print out the info, we need to wrap the method call in a print statement.

```
employee_1 = Employee("E. Davis", "edavis@business.com", "Hiring
Manager")
employee_2 = Employee("D. Wong", "dwong@business.com",
"Developer")
print(employee_1.get_info())
```

We can also use an alternate way of calling our method, calling it directly on the class, and then passing in the desired instance.

```
print(Employee.get_info(employee_1))
```

Much like how there is more than one kind of variables when it comes to classes (class and instance variables), there are also more than one kind of class methods. There are regular (instance) methods, class methods, and static methods.

Python also makes use of class and static methods. In all likelihood, you won't use either class or static methods anywhere near as often as you use the regular, instance methods. In order to create the class and static methods, we use what is referred to as a decorator. The decorator starts with an @ symbol, like this - @staticmethod or @classmethod. It is placed directly above the line containing the def keyword that starts the method.

Class methods take in classes as the first parameter instead of the self keyword, and this is usually represented by cls. The keyword cls is used to specify an entire class rather than an instance. If an instance (regular) method accesses instance variables, you might be able to guess that the primary use of class methods is to access and manipulate the class variables

you have created within a class. For example, using a class method, we can change the location class variable we created above.

Below, we'll create a class method that operates on the class. We use the `@classmethod` decorator and pass in `cls` as the first argument. We then update the class location to a new location we pass in when we call the method. We'll create a new employee to test the new class method.

```
class Employee:
    location = "Seattle, WA"
    def __init__(self, name, email, role):
        self.name = name
        self.email = email
        self.role = role
    def get_info(self):
        return '{} {} {}'.format(self.name, self.email, self.role)
    @classmethod
    def change_locale(cls, new_location):
        cls.location = new_location
employee_3 = Employee("R. Acevedo", "racevedo@business.com",
"Developer")
print(employee_3.location)
Employee.change_locale('Los Angeles, CA')
print(employee_3.location)
```

Python also has another form of method: the static method. Static methods are methods that receive neither an instance nor a class when they are called. Static methods are bound to a class instead of an object, and they deal with parameters of the class. Since static methods can't access the properties of the class itself, they have limited use cases. Static methods are typically only used when a utility function is required that doesn't operate on any properties of a class, but still makes sense for the method to belong to the class.

Chapter 7 - Working with Files

The next thing that we need to focus on when it comes to working with Python is making sure we know how to work and handle files. It may happen that you are working with some data and you want to store them while ensuring that they are accessible for you to pull up and use when they are needed. You do have some choices in the way that you save the data, how they are going to be found later on, and how they are going to react in your code.

When you work with the files, you will find that the data are going to be saved on a disk, or you can re-use in the code over and over again as much as you like. This phase is going to help us learn a bit more about how to handle some of the work that we need to do to ensure the files behave the way that they should, and so much more.

Now, we are going to enter into file mode on the Python language, and this allows you to do a few different options along the way. A good way to think about this is that you can think about it like working on a file in Word. At some point, you may try to save one of the documents that you are working with so that it doesn't get lost and you can find them later on. These kinds of files in Python are going to be similar. But you won't be saving pages as you did on Word, you are going to save parts of your code.

You will find with this one that there are a few operations or methods that you are able to choose when it comes to working with files. And some of these options will include:

- Closing up a file you are working on.
- Creating a brand new file to work on.
- Seeking out or moving a file that you have over to a new location to make it easier to find.
- Writing out a new part of the code on a file that was created earlier.

Creating New Files

The first task that we are going to look at doing here is working on creating a file. It is hard to do much of the other tasks if we don't first have a file in place to help us out. If you would like to be able to make a new file and then add in some code into it, you first need to make sure the file is opened up inside of your IDLE. Then you can choose the mode that you would like to use when you write out your code.

When it comes to creating files on Python, you will find there are three modes that you are able to work with. The three main modes that we are going to focus on here include append (a), mode(x) and write(w).

Any time that you would like to open up a file and make some changes in it, then you would want to use the write mode. This is the easiest out of the three to work with. The write method is going to make it easier for you to get the right parts of the code set up and working for you in the end.

The write function is going to be easy to use and will ensure that you can make any additions and changes that you would like to the file. You can add in the new information that you would like to the file, change what is there, and so much more. If you would like to see what you can do with this part of the code with the write method, then you will want to open up your compiler and do the following code:

```
#file handling operations
#writing to a new file hello.txt
f = open('hello.txt', 'w', encoding = 'utf-8')
f.write("Hello Python Developers!")
f.write("Welcome to Python World")
f.flush()
f.close()
```

From here, we need to discuss what you can do with the directories that we are working with. The default directory is always going to be the current directory. You are able to go through and switch up the directory where the code information is stored, but you have to take the time, in the beginning, to change that information up, or it isn't going to end up in the directory that you would like.

Whatever directory you spent your time in when working on the code is the one you need to make your way back to when you want to find the file later on. If you would like it to show up in a different directory, make sure that you move over to that one before you save it and the code. With the option that we wrote above, when you go to the current directory (or the directory that you chose for this endeavor, then you will be able to open up the file and see the message that you wrote out there.

For this one, we wrote a simple part of the code. You, of course, will be writing out codes that are much more complicated as we go along. And with those codes, there are going to be times when you would like to edit or overwrite some of what is in that file. This is possible to do with Python, and it just needs a small change to the syntax that you are writing out. A good example of what you can do with this one includes:

```
#file handling operations
#writing to a new file hello.txt
f = open('hello.txt', 'w', encoding = 'utf-8')
f.write("Hello Python Developers!")
f.write("Welcome to Python World")
mylist = ["Apple", "Orange", "Banana"]
#writelines() is used to write multiple lines into the file
f.write(mylist)
f.flush()
f.close()
```

The example above is a good one to use when you want to make a few changes to a file that you worked on before because you just need to add in one new line. This example wouldn't need to use that third line because it just has some simple words, but you can add in anything that you want to the program, just use the syntax above and change it up for what you need.

What Are The Binary Files?

One other thing that we need to focus on for a moment before moving on is the idea of writing out some of your files and your data in the code as a

binary file. This may sound a bit confusing, but it is a simple thing that Python will allow you to do. All that you need to do to make this happen is to take the data that you have and change it over to a sound or image file, rather than having it as a text file.

With Python, you are able to change any of the code that you want into a binary file. It doesn't matter what kind of file it was in the past. But you do need to make sure that you work on the data in the right way to ensure that it is easier to expose in the way that you want later on. The syntax that is going to be needed to ensure that this will work well for you will be below:

```
# put binary data to a file
# writing the file hello.dat put binary mode
F = open ('hello.dat', 'wb')
# writing as byte strings
f.write("I am writing data in binary file!/n")
f.write("Let's write another list/n")
f.close()
```

If you take the time to use this code in your files, it is going to help you to make the binary file that you would like. Some programmers find that they like using this method because it helps them to get things in order and will make it easier to pull the information up when you need it.

Opening Your File Up

So far, we have worked with writing a new file and getting it saved, and working with a binary file as well. In these examples, we got some of the basics of working with files down so that you can make them work for you and you can pull them up any time that you would like.

Now that this part is done, it is time to learn how to open up the file and use it, and later even make changes to it, any time that you would like.

```
# read binary data to a file
#writing the file hello.dat write append binary mode
with open("hello.dat", 'rb') as f:
data = f.read()
```

```
text = data.decode('utf-8')
print(text)
```

The output that you would get from putting this into the system would be like the following:

Hello, world!

This is a demo using with

This file contains three lines

Hello world

This is a demo using with

This file contains three lines.

Seeking out a file you need

And finally, we must need to take a look at how you can seek out some of the files that you need on this kind of coding language. We already looked at how to make the files, how to store them in different manners, how to open them and rewrite on them, and then how to seek the file. But there are times where you are able to move one of the files that you have over to a new location.

For example, if you are working on a file and as you do that, you find that things are not showing up the way that you would like it to, then it is time to fix this up. Maybe you didn't spell the time of the identifier the right way, or the directory is not where you want it to be, then the seek option may be the best way to actually find this lost file and then make the changes, so it is easier to find later on.

With this method, you are able to change up where you place the file, to ensure that it is going to be in the right spot all of the time or even to make it a bit easier for you to find it when you need. You just need to use a syntax like what is above to help you make these changes.

Working through all of the different methods that we have talked about in this guide book are going to help you to do a lot of different things inside of your code. Whether you would like to make a new file, you want to change up the code, move the file around, and more.

Chapter 8 - Python From Scratch

This phase will allow you to discover the Python language if you do not know already or if you need some reminders. We will get to some basic themes and some topics useful to the data scientist.

Basic Principles

An Interpreted, High-Level, Object-Oriented Language

As we saw earlier, Python is above all a language of programming and it must be addressed taking into account this specificity. This language is simple and allows you to progress very quickly (hence its success with data scientists developers).

Python is a high-level interpreted language. Moreover, it is also an object-oriented language which means that it is based on classes that simplify its use. What does that mean?

An interpreted language: it is a programming language that is processed in a direct way (on this point it is close to R). An interpreted language as opposed to a compiled language, which requires a compilation of the code to make an executable program. The main advantage of an interpreted language is its debugging simplicity. Conversely, its main defect is its slowness. Unlike a compiled language, code blocks are not optimized to be extremely fast by the machine. In the case of Python, we treat the code line by line with the Python interpreter.

A high-level language: a high-level language is a language that is as close as possible to natural language, that is to say, which reads "as applies ". It is therefore extremely simple to implement but it's not very optimized. Conversely, a low-level language will come as close as possible to the language of the machine to which it applies. The best-known low-level language is assembly language.

As a language, Python is based on a few principles:

It is based on indentation.

It is extremely flexible.

Its rules are set by the Python community.

We will dwell on all these points later. We will start by studying some important tool choices to code in Python.

Python 2 or Python 3

In 2008, the developers of Python within the Python Software Foundation decided to go from version 2 to version 3. But they also decided that big housework is needed in the language to clarify the code. That makes Python 3 not retro-compatible with Python 2. The millions of lines in Python 2 must be rewritten in order to be submitted in Python 3. After more than ten years of coexistence, Python 2 is disappearing. Indeed, more and more packages are announced that new features will not be developed in Python 2, which will inevitably speed up the process of switching to Python 3 (that's the case of NumPy for 2019). In addition, the completion of Python 2.7 maintenance is officially announced for 2020.

The advice I give you is this: if you are working on a new project in Python, so start with Python 3 without hesitation! Python 2 can win if you get an important legacy and that by necessity you have to continue to code in Python. Even in this case, the option to port to Python 3 may impose itself.

To recognize a code in Python 2, just look at the displays using the command `print` in the code. Indeed, the `print` was a Python 2 command that became a function in Python 3.

PYTHON 2

```
>>> print 55 55
```

PYTHON 3

```
>>> print (55) 55
```

The differences between these two versions are widely documented by Python Software Foundation. Whatever happens, Python 3 remains the future and a return to Python 2 is highly unlikely.

The Interpreters: Python And Ipython

The interpreter is the tool that translates your source code into action. IPython can be seen as an evolved version of the classic Python interpreter.

The Python Interpreter - An Advanced Calculator

The Python interpreter is a rich tool but also allows you to make simple calculations. So, once Python is launched from the command prompt, we can have :

```
4 + 6
```

```
10
```

You can use this line to submit code directly or to submit files in Python (in .py format)

```
x = 3
```

```
print (x)
```

```
3
```

To submit a file, we will use:

```
exec (open ( "./ test.py"). read ())
```

The Python interpreter can be used as a simple calculator with classical mathematical operations:

```
2 + 5
```

```
7
```

```
2 * 5
```

```
10
```

```
2/5
```

```
0.4
```

```
2-5
```

```
-3
```

```
5 ** 2
```

```
25
```

```
5% 2
```

```
1
```

We see here that the power is noted ** (not to be confused with ^ which is a logical operator). The modulo% allows extracting the rest of the whole division and %2 lets us know if a number is even or odd.

As you can see in this example when dividing two integers, we get a decimal number ($2/5 = 0.4$). This was not the case with Python 2 but has been added to Python 3. You can of course code in the Python interpreter directly but we tend to prefer an improved interpreter: IPython.

The IPython Interpreter - An Opening Towards More Possibilities

When you code in Python. Here, we will develop some of the advantages of IPython.

Auto-completion and help with IPython

This is a very important point and one of the strengths of IPython: it has very advanced completion tools. This completion is extremely powerful and will bring you help and your productivity is also constantly increased.

Access to help and code

You can very easily access the help functions and methods by using the combination (Shift + Tab).

In addition, using a question mark (?), You will have access to help (the docstring) and with two question marks (??), you will have access to the source code of the function studied.

The magic keys of IPython

IPython has many magic keys. These are specific commands, starting with % and that will simplify your coding. There are magic keys per line (cell magics) that will apply to a single line of code. We use simple keys with a single sign % that applies to a line of code. There are also magic keys using %% which apply to the entire cell of a Jupyter notebook. If you want to get a list of all magic keys, just type %lsmagic.

A number of them are described:

%CD: Change of working directory

debug %: Enabling the interactive debugger

% macro: Ability to combine execution lines to repeat several actions (use macro macro_name input_number)

% notebook: Exporting IPython history in a Jupyter notebook whose name we provide

prun%: Profiling of a function

`% PWD`: Displaying the working directory

`% run`: Launching a Python file directly in a notebook

`% save`: Save as .py blocks of code

`% time`: Display of the execution time

`timeit%`: Display the execution time of order by repeating it several times

`whos%`: Display all objects and functions loaded in memory

All keys are available here:

<http://iPython.readthedocs.io/en/stable/interactive/magics.html>

In addition, IPython provides access to system commands using the exclamation point (!). For example, we can use `! ls` that will display all files in the current directory as in the terminal.

Zoom: profiling and optimizing your code

In the list of magic keys, we can see that the functions `% time` and `% timeit` seem interesting. Their operation is simple: you have a code in Python using a function that seems inefficient. You will be able to use `% timeit` to test the launch of this function. `% timeit`, unlike `% time`, will launch this function many times to give you a more precise idea of his behavior:

```
In []:% timeit x = 5
```

```
Out []: 14.2 ns ± 0.0485 ns per loop (mean ± std dev of 7 runs, 100000000 loops each)
```

If you find that your function is a little slow, you still have to improve it. For this, other tools are solicited. For example, `% prun` breaks down all the steps followed by your code with an estimate of the time needed for each step. This key searches for bottlenecks in your code.

Zoom: Variables loaded in memory

Python is a language that loads all objects into memory. It may be interesting to know the state and number of objects loaded into memory. To obtain the list of these objects, we use `% who`. If we want more details on each object, we use `% whos`. We will have :

```
In []:% whos
```

```
Out []: Variable Type Data / Info
```

x int 5000

This will become more interesting for the arrays of the NumPy package. For other data structures, information is sometimes missing. For a DataFrame of Pandas, we can use the `.memory_usage ()` method which gives us the size of each column in bytes. For a NumPy array, the property `.nbytes` is useful.

If you want to apply this to all the elements loaded into memory, you can use the `getsizeof ()` function of the `sys` module.

Access to previous results

Another interesting feature of IPython is access to what you did previously. For this, we use different approaches:

To display the last output, use `_`

To recover the output `Out [33]` for example, we use `_33`

To get the last entry, we use `_i`

To recover the input `In [33]`, we will use `_i33`

This simplification makes it possible to store input and output data in objects. An entry is automatically stored in a string.

Zoom: storing objects

Sometimes you have to process objects from one notebook to another or important calculations are needed to obtain a data structure. In this case, it is obviously not very efficient to restart the calculations at next launch or let your kernel run until the next scan. You can use the magic key `% store` which will allow you to store an object in a persistent and simple way. This is close to the pickle approach that we use in machine learning models but in a much simpler way.

In your notebook, you will do:

```
In []: list1 = [3,5,7]
```

```
% store list1
```

And in any other notebook or when your kernel will be reset, you can do:

```
In []:% store -r
```

```
print (list1)
```

```
[3,5,7]
```

If you want to see what is stored, just run `% store all` by yourself. If you want to erase everything that is stored, just enter `% store -z`. Many other options are available to you when using IPython. We will now return to the Python language to understand its use.

Chapter 9 – How Coding Works

Since you have learned about accounting, swapping, summing, flag variable, maxis and mins, and simple debugging used in Python, it is now the time to start running some codes. I believe you have already installed Python on your computer at this time. If you don't have, kindly follow the installation procedures and install it on your computer now.

There are many ways a programmer can use to tell Python to execute the code he/she has typed. In this Phase, we will discuss all the techniques used for launching a program today. In addition to learning how to type code interactively, you will be able to know different ways of saving the code in the file to run in your desired way with the help of system command lines, exec calls, icon clicks, and module imports among others as you will see in this Phase.

However, if you have experience in other programming languages and you would like to start digging into Python, it is important to go through this phase thoroughly. It is an overview of debugging techniques and will help you understand different ways of importing and saving codes in the file.

Interactive Prompt

You will learn some interactive coding basics and then look at running the code before covering some preliminaries like setting up system path and directory. You will apply what you had learned about directory and system path here.

Starting an Interactive Session

The simplest way to start running a Python program is by typing it at the interactive prompt, commonly known by many programmers as Python's interactive command line. In fact, there are many ways of starting this command line and these include a system console among others. If you have installed the interpreter as an executable program on the system, you can just start by typing it to Python at the operating system's prompt as this is the most effective way of starting an interactive interpreter session. When we type the word "python" at our system shell, for example, we will be initiating the system to start an interactive Python session. Note that the

character “%” at the beginning of this listing represents a generic system prompt; it is not an input a programmer can type by herself/himself. A Ctrl-Z will get out of this session if you are using Windows: therefore, try to use Ctrl-D on the Unix instead.

Although a system shell has a generic notion, its accessibility varies from one platform to another as seen below:

- On Windows

If you are using Windows, launch Command Prompt (cmd.exe) and type the word “python” in the console window.

- On macOS

Python interactive interpreter can be started by launching Terminal from Spotlight or double-clicking the Terminal icon found in Applications > Utilities folder. In the Terminal window, type the word “python”.

- On the Linux and other UNIX operating systems

As a programmer, you can type this command in a terminal window or even in the shell to run your program.

Alternatively

Some platforms allow you to start the interactive prompt in different or additional ways that will not require you to type a command. Examples of such platforms include Windows 7 and Windows 8.

- Windows 7

Apart from typing python in a shell window, a programmer can start the same interactive session by selecting the python menu option. This is found in the Start button menu for Python.

- Windows 8

When using Windows 8, there is no need to have a Start button as there are many ways you can use to get to the tools such as search, and file explorer.

- Other Platforms

If you are using other platforms other than Windows 7 and Windows 8, do not worry. The appearance of the prompt on your computer shows you that you are already in the interactive Python interpreter session, and therefore, you are free to type any Python expression or statement there and run it immediately.

Where to find Command Prompt on the Windows

Stating the command-line interface is not easy for most of us, especially if you are a beginner. Although some Windows readers know it, Unix developers and beginners do not understand it because it is not prominent like console or terminal windows. The following procedures will help you to find your Command Prompt easily. If you are using Windows 7, you can find this in the Accessories section under the Start and then all Programs menu. Alternatively, you can type cmd in the Start → Run box and allow the program to run. If you are using Windows 10, you can launch Command Prompt by directly typing cmd on the Windows Search bar found on the taskbar.

The System Path

Usually, the system will locate your python program on its system's path after typing python. Depending on the version of the Python and platforms you are using, it is important to replace the word 'python' with your desired full one if you have not set the PATH environment variable of your system to include installed directory. You must make sure that the PATH environment is simple to allow the program to run properly.

Running Code Directories

Since we have started discussing how to run the code on your computer, it is essential to know where to run the code to ensure the program run without making some errors. You will be running code from a directory called folder, created on your Windows called C:/code found at the top of your main drive. Most of your interactive sessions will start from there. Also, ensure you save and run all your script files from there. However, if you have been using other programming languages like Java and C++ and you would like to use Python doing your program, follow the following

instructions. They will help you understand how to get started with a working directory using Python programming on your computer.

- **Unix-based Systems**

These include Linux and MacOS. For these systems, a working directory can be found in /usr/home or sometimes created by a mkdir command. With a working directory, you will be able to determine or see how the code runs.

- **Windows**

Windows systems allow programmers to make their working code directory in the Command Prompt window or File Explore easily. Search for the New Folder in the File Explorer, you will see File menu. Alternatively, type mkdir command in the Command Prompt and run it. you can locate and recall your working directory any time. Also, running out of one directory helps a programmer to track his/her work easily.

How to Run Code Interactively

Because you have learned all these preliminaries, it is now the time to start typing some actual code and run them interactively on your python program. Note that you had already started typing in the python interactive session by typing two lines of information text that not only gave the Python version number but also a few hints as illustrated in our early discussion. Usually, the result of our code will be displayed below the input lines when we work interactively and this is after pressing the Enter key. When you type the print statement at the prompt, for example, a Python string also called output will echo back right away. Therefore, there is no need of creating a source code file or run the code a compiler if you are not using Python language. You will learn how to run multiline statements and such statements run as soon as they are entered in their lines and press Enter button twice.

Reasons for the Interactive Prompt

Even though the interactive prompt will echo the results when you run it, it will not save the code in the file. This shows that you cannot handle the bulk of coding in the interactive sessions as you may think. The interactive

prompt has turned out to be a good place to test program files or experiment the language on the fly.

Experimenting

Due to its ability to execute the code immediately, the interactive prompt has become the best place to experiment with language. If you are not sure about the working of python code, you can see what takes place when you fire up the interactive command line. If you are reading code in the Python program, for example, you may see an expression that you do not understand its meaning. Example of such expressions could be 'Spam!'*8. You will spend a lot of your time reading the manuals, or books, or even search over the internet to see its meaning.

With immediate response you will receive at the interactive prompt, you can use it to determine the working of the code quickly. From here, for instance, it is clear that code does string repetition. The sign '*' is used in Python to mean the multiplication of numbers as well as repetition for the strings. It is just like concatenating the strings to themselves repeatedly. You will not break anything by this experiment. Generally, Python code is the most appropriate to run as it does not result in the deletion of the files.

Moreover, it is an error to use a variable that has not been assigned value in Python programming. Some errors can go undetected if you fill the names in with defaults. Therefore, to do away with such errors, it is important to start initial counters from zero before adding anything to them and also make sure you have initial lists to help you extend them properly. With initial lists and counting from zero, you will be able to run your program without producing any error.

Testing

In addition to serving as a tool or experimenting, the interactive interpreter is used to test the code you will be writing in the files while learning the Python language. In fact, we will show you how to import the module files interactively. Also, we will show you how to run the tests on the tools defined by typing calls at the interactive prompt.

Additionally, many programmers test programming components at the interactive prompt. As a programmer you can import, test, and run functions and classes in the Python files regardless of their sources. This is

achieved by typing calls to linked-in-C functions as well as exercising Java classes in Python. Finally, with the interactive nature of the Python, it can support an experimental programming style thus making it convenient for you to get started. This is making the Python programming to be simple, easy, and best for beginners to use to run the code on their programs.

Guidelines For Using The Interactive Prompt Effectively

Even though it is easy to use interactive prompt, as a beginner, there are many things you should consider when using it to ensure your code runs without producing errors. The following guidelines will help you to avoid making common mistakes seen by other beginners. Just take your time to read them:

Ensure You Only Type Python Command

In many cases, beginners make a big mistake by typing system commands in the interactive prompt. This makes their computer to display errors when they try to run their programs. Even though there are many different ways of running system commands from the python code, these methods do not involve typing the commands themselves.

Chapter 10 - Python Libraries

We have talked about Data Analysis, and now it is time to take some of that information and put it to good use. You are probably interested in deep learning, and maybe even in making some of your Convolutional Neural Networks, but are wondering where you should start. The best step is to pick out the library that you want to use. But this brings up another challenge because there are just so many coding libraries out there that you can choose from, and all of them have some amazing power and features behind them.

To start with, we are going to take a look at some of the best Python libraries that can help with deep learning. Other languages can help with things like machine learning and deep learning. But for most of the tasks that you want to do, especially if you are a beginner in Data Analysis and all of the processes that we have been talking about, then Python is going to be the choice for you. Even within Python, there are several libraries that you can choose from to get your deep learning work done. So, with that in mind, let's dive right in and see some of the best Python deep learning libraries that you can use for your Data Analysis.

Caffe

It is pretty hard to get started with a look at deep learning libraries through Python without spending some time talking about the Caffe library. It is likely that if you have done any research on deep learning at all, then you have heard about Caffe and what it can do for some of the projects and models that you want to create.

While Caffe is technically not going to be a Python library, it is going to provide us with some bindings into the Python language. We are going to use these bindings when it is time to deploy the network in the wild, rather than just when we try to train the model. The reason that we are going to include it in this Phase is that it is used pretty much everywhere and on all of the parts of a deep learning model that you need to create.

Theano

The next kind of library that we can work with is known as Theano. This one has helped to develop and work with a lot of the other deep learning libraries that we have that work with Python. In the same way that a programmer would not be able to have some options like scikit-image, scikit-learn, and SciPy without NumPy, the same thing can be said when we talk about Theano and some of the other higher-level abstractions and libraries that come with deep learning.

When we take a look at the core of this, Theano is going to be one of the Python libraries that not only helps out with deep learning, but can be used to define, optimize, and evaluate a lot of mathematical expressions that will involve multi-dimensional arrays. Theano is going to accomplish this because it is tightly integrated with the NumPy library, and it keeps its use of GPU pretty transparent overall. While you can use the Theano library to help build up some deep learning networks, this one is often seen as the building blocks of these neural networks, just like how the NumPy library is going to serve as the building blocks when we work on scientific computing. Most of the other libraries that we will talk about as we progress through all of this are going to wrap around the Theano library, which makes it more accessible and convenient than some of the other options.

TensorFlow

Similar to what we can find with the Theano library, TensorFlow is going to be an option that is open-sourced and can work with numerical computation with the help of a data flow graph. This one was originally developed to be used with research on the Google Brain Team within Google's Machine Intelligence organization. And this library, since that time, has turned into an open-sourced option so that the general public can use it for their deep learning and data science needs.

One of the biggest benefits that we are going to see with the TensorFlow library, compared to what we see with Theano, is that it can work with distributed computing. This is particularly true when we look at multiple-GPUs for our project, though Theano is working on improving this one as well.

Keras

Many programmers find that they love working with the Keras library when it comes to performing models and other tasks with deep learning. Keras is seen as a modular neural network library that is more minimalistic than some of the others that we talk about. This one can use either TensorFlow or Theano as the backend so you can choose the one that works the best for any needs you have. The primary goal that comes with this library is that you should be able to experiment on your models quickly and get from the idea that you have over to the result as fast as possible.

Many programmers like this library because the networks that you architect are going to feel almost natural and easy, even as a beginner. It is going to include some of the best algorithms out there for optimizers, normalization, and even activation layers so this is a great one to use if your process includes these.

If you want to get your network trained as fast as possible, working with a library like MXNet may be a better choice. But if you are looking to tune your hyperparameters, then you may want to work with the capability of Keras to set up four independent experiments and then evaluate how the results are similar or different between each of these.

Sklearn-Theano

There are going to be times when working with deep learning when you will want to train a CNN end-to-end. And then there are times when this is not needed. Instead, when this is not needed, you can treat your CNN as the feature extractor. This is going to be the most useful with some situations you may encounter where there is just not enough data to train the CNN from scratch. So, with this one, just pass your input images through a popular pre-trained architecture that can include some options like VGGNet, AlexNet, and OverFeat. You can then use these pre-trained options and extract features from the layer that you want, usually the FC layers.

Nolearn

A good library for you to work with is the Nolearn library. This is a good one to help out with some initial GPU experiments, especially with a MacBook Pro. It is also a good library to help out with performing some deep learning on an Amazon EC2 GPU instance.

While Keras wraps TensorFlow and Theano into a more user-friendly API, you will find that the Nolearn library will be able to do the same, but it will do this with the Lasagna library. Also, all of the code that we find with Nolearn is going to be compatible with Scikit-Learn, which is a big bonus for a lot of the projects that you want to work with.

Digits

The first thing to notice with this library is that it isn't considered a true deep learning library. Although it is written out in Python and it stands for Deep Learning GPU Training System. The reason for this is because this library is more of a web application that can be used for training some of the models of deep learning that you create with the help of Caffe. You could work with the source code a bit to work with a backend other than Caffe, but this is a lot of extra work in the process. And since the Caffe library is pretty good at what it does, and can help with a lot of the deep learning tasks that you want to accomplish, it is not worth your time.

If you have ever spent some time working with the Caffe library in the past, you can already attest to the fact that it is tedious to define your .prototxt files, generate the set of data for the image, run the network, and babysit the network training with the terminal that you are provided. The good news here is that the DIGITS library aims to fix all of this by allowing you to complete a lot of these tasks, if not all of these tasks, just from your browser. So, it may not be a deep learning library per se, but it does come into use when you struggle with the Caffe library.

In addition to all of the benefits above, the interface that the user gets to interact with is seen as excellent. This is because it can provide us with some valuable statistics and graphs to help you train your model more effectively. You can also easily visualize some of the activation layers of the network to help with various inputs as needed. And finally, another benefit that is possible with this library is that if you come in with a specific image that you want to test, you have a few options on how to get this done. The first choice is to upload the image over to the DIGITS server. Alternatively, you can enter the URL that comes with the image, and then the model you make with Caffe will automatically be able to classify the image and display the results that you want in the browser.

Python is one of the best coding languages available for helping with tasks like deep learning, machine learning, and even with the topic of artificial intelligence, which encompasses both of the other two ideas. Other languages can handle the deep learning that we have been talking about, but none are going to be as effective, as powerful, have as many options, or be designed for a beginner in the way that Python can.

This is why we have focused our attention on the Python language and some of the best libraries that we can choose to help with a variety of deep learning tasks. Each of these libraries can come on board with your project and will provide a unique set of functions and skills to get the job done. Take a look through some of these libraries and see which one is going to be just right for your Data Analysis and for providing you with great insights while completing deep learning.

Chapter 11 – Getting Program Functioning to Work

The objective of this section is to reinforce the concepts related to functions. To make this a little more interesting than others, I will be providing you with scenarios to carry out on your own and convert into fully functioning programs. You will be provided with a solution at the end. All you need is to ensure you read the scenario and visualize it to sort out what needs to go where, and then form a fully functional program.

From this point forward, I will be providing you with various scenarios, each consisting of programs that you will need to create. Some of these may require you to do a bit of research as well. However, if any special method, library, module, or package is used, I will provide those within the instructions.

Q-1: You are to create a function that calculates a taxi fare. The taxi fare is comprised of a base fare of \$3.00 and then \$0.10 for every 100 meters traveled. Create a function that takes distance as its only parameter (in km) and returns the value of the total applicable fare. Follow up with a program to show the functioning nature of the function.

The situation is rather easy. All you need is to work on the function and work out the details accordingly. The rest will just fall into place automatically. Take your time and process the information.

I know it is easy to be overwhelmed. A quick look at the internet will only leave you confused. Take small steps and start somewhere. With time and a bit of understanding, you will soon grasp the concept and be able to figure out such situations easily.

The next one to come is even tougher. We have entered the phase where we can say goodbye to the five to 10 line programs. It is best to let them go and practice on bigger, more complex programs to polish your skills and make yourself step out of your comfort zone and properly start exploring.

Q-2: A client has asked you to create a reusable program containing reusable functions. The first situation is to create a function that creates a virtual deck of playing cards.

Since there are 52 cards, the cards with numeric values from two to nine will be represented by their respective numbers. For 10, Jack, Queen, King, and Ace, you are required to use T, J, Q, K, and A.

Following the numeric/alphabetic value will be another character to represent the suit. Use h for hearts, c for clubs, d for diamonds, and s for spades.

You should create a function that does not take any parameter and uses loops to iterate through all the cards and store them with a two-character abbreviation in a list. The function should only return this list as a result.

Hint: You will need to use the following as your beginning line.

```
from random import randrange
```

For this, expect quite a few lines of code. Your complete knowledge about Python will be tested and tried in this one and the ones to come ahead. There is no need to rush into things. If you are not able to do it the first time around, you can do a bit of research and get some suggestions. I would recommend not to jump to the solutions right away. Try and push your brain to think a little outside the box. There is much to be learned, and the only way it can be learned is to try.

Q-3: You have been asked by a colleague to help out with a Python project. The assignment is to create a function that can generate a random password for a user, which would contain between 6 to 8 characters at most.

For this, you will need to use the following:

```
from random import randint
```

```
shortest_pass = 6
```

```
max_pass = 8
```

```
min_ASCII = 33
```

```
max_ASCII = 126
```

The function should generate a random value from positions 33 to 126 of the ASCII table. This function will not take any parameters.

The above exercise will test you further and might even require a little research on the ASCII table if you haven't seen one before. It is always nice to carry out a bit of research as it greatly helps us as a programmer to further excel at what we do.

One of the toughest calls to make when creating a function is knowing the right parameter you need to use to make the function work. You can use almost any argument you pass through the parentheses to make it into a parameter. Sometimes, you may not need a parameter at all, as we saw in some of the exercises above. In such cases, it is best to leave the parameter blank.

When in doubt, always consult the Python documentation to learn more about the various parameters you can make use of when defining functions. For recalling functions, hover your mouse over the parentheses, and a little prompt should display the kind of parameters you can use with said function.

Is This Correct? - Part 4

Q-1: Below is a user-made function that is designed to iterate through a given range and look for the highest number. Will the function work when it is called?

```
def high_number(numbers):  
    max = numbers[0]  
    for number in numbers:  
        if number < max:  
            max = number  
    return max  
list = [21, 200, 31, 1, 39]  
high_number(list)
```

Q-2: What seems to be the issue with the following?

```
def this_function():  
    print("Hello From This Function!")  
this_function_with_args(name, greeting) :  
    print(f"Hello {name}, From This Function!, I wish you {greeting}")  
this_function()  
this_function_with_args()
```

Q-3: What would this function do?

```
def plus(a,b):  
    sum = a + b  
(sum, a)  
sum, a = plus(3,4)  
print(sum)
```

Q-4: Can you place a loop within a function, as shown below?

```
def plus(*args):  
    total = 0  
    for i in args:  
total += i  
    return total  
print(plus(20,30,40,50))
```

Now that we have revisited quite a few concepts and methods, and functions as well, let us head over to our final project and see what it is all about.

Final Project – Hangman

Remember the old game called Hangman? The one that involved blanks and a limited number of guesses to guess a movie, a name, a person, a city, or something else? For your final project, I decided to come up with a tough one. A project that will use almost everything you have learned.

To make it even better, if you execute the program correctly, you can store it as a callable function or create a separate package so that you can use this over and over again. While there are hundreds of variations for this game online, use your unique approach. Feel free to browse the internet to get some inspiration.

Requirements:

You will need to use the following as your first line:

```
import time
```

Let this be a project that you are proud of, once it is finished. By the end of this project, rest assured you are ready to take on the challenges and offer

some exquisite programming skills to those who require programmers such as yourself.

The journey of a programmer does not end here. There are far too many things that lay ahead, which you will need to keep pace with. Learn about various libraries, modules, and packages to see how they can bring refinement to your projects.

For those interested in Machine Learning, Automation, Artificial intelligence, and Deep Learning, you will come across some names like Scikit-Learn, Turtle, and a few more. While it is still too early to jump into these, it is a good idea to have a look at them and see how they perform in action.

Conclusion

Python runs programs on the operating system, and generally, if you want to run some JavaScript on the operating system you have to use NodeJS, you can take it as a Python interpreter, because thanks to this program you will be able to run JavaScript programs just like you would run Python ones.

There is no better language than another, what differentiates them are the tools used by each one, and the applications that each one can generate. The main objective is to be able to solve a problem in the best and most optimal way.

In the case of Python and JavaScript, both are interpreted languages so that you will see in practice a file, and you will write your code in this file, you save it so that the computer can execute it, but for this to happen you will need an extra program that interprets these codes and allows the computer to read it, and this is what an interpreter does, if we install it in Linux you will be able to interpret the code in Linux, if we install it in Windows you will be able to interpret the code there and in the same way in Mac. Both languages would then be multiplatform.

Both languages have Open Source syntax; this means that you can see how the codes of the standard libraries are written, so you will be able to see more easily how each of these programs is developed.

Now let's talk about the differences between both languages. JavaScript (JS) was a language that was born to add interactivity to the browser. In JS, the functionality that brings the language itself is very minimal, since it lacks a large amount of codes it is complemented with another tool called npm, this way you will be able to download codes from the internet to complement this functionality because JS does not include all of the tools. It is at a disadvantage compared to Python because you are going to find a wide variety of codes that you will have to choose which one you are going to use. In Python, you will find certain types of applications in which it is unique. In JS, the most popular default type of application are web applications, web-based applications, both in the server and browser environment.

Python, unlike JS, was born as a general-purpose programming language, that is to say, it was not born for the purpose of the browser as JS, but was

born to create any type of application that runs on the computer, whether desktop applications, web, server applications, and that is already a great advantage in itself over JS.

It has a huge amount of libraries already included by default, while JS is very different because it has quite minimal information. You can also extend the functionality of Python by installing pip for Python, which would be more or less the equivalent of npm for JS if you like.

In Python, you will find packages with specific functionality while in JS, it is easier to find several packages that do basically the same thing.

As for Python and the programming language C, in the case of Python, it differs mainly because this is an interpreted language, and C is a compiled language, although it is true that Python is a slower language than C, it is also easier to learn and use.

C is a language used for real-time systems, operating system kernels, for low-level hardware programming, such as microwaves, digital clocks, sensors. While python has many more commercial applications and a variety of environments, including websites, desktops, programming small computers like Raspberry Pi, and many others.

CODING WITH PYTHON

**THE ESSENTIAL COURSE TO
MASTER IN A SMART WAY
SOFTWARE CONCEPTS, TOOLS, AND
ALGORITHMS FOR PRACTICAL
PROGRAMMING AND DEEP
MACHINE LEARNING TO BUILD
ARTIFICIAL
INTELLIGENT SYSTEM**

Adrienne Hawkes Matthes and John Tacke

Introduction

Python is among the most popular computer language programming tool initially created and designed by Guido Van Rossum in the late 1980s. Since its introduction into the computing world, Python has undergone multiple modifications and improvements, therefore, becoming among leading programming languages used by developers. The tool is dynamically typed, object-oriented, multi-paradigm, and imperative. It is used across different operating systems including Windows, Linux, Android, macOS, and iOS devices. Besides, it is compatible with both bit 32 and bit 64 gadgets of phones, laptops, and desktops.

Despite comprising of several areas essential for programmers, Python is easy to learn, especially when it comes to beginners with minimal knowledge in computer programming. Unlike most programming languages, Python accompanies an easy to use syntaxes where first time users can readily practice and become a pro within a few weeks. However, the programming processes may vary depending on the motive of the learner in programming. Despite accompanying multiple vocabularies and sometimes sophisticated tutorials for learning different programming techniques, engaging with Python is worth it to develop excellent programs.

Features of Python Programming

Simple Language

Most programming languages have complicated and lengthy coding languages which may become cumbersome to beginners. More so, long and challenging languages may become hard to learn and remember, therefore hindering amateurs learning abilities. Python accompanies very simple and fantastic syntax, making beginners read and write programs without complications readily. When compared to Java and C++, Python henceforth enables you to work with ease while focusing on the outcomes.

Portability

With its ability to run in any operating system, Python allows for portability where you can readily transfer your codes and the general program from one device to the other without affecting your progress. This programming

tool is quite useful for developers who change devices or transfer data from one platform to the other. You can, therefore, run your program in the new machine seamlessly with limited alliteration. Besides, Python allows the continuation of your application to your primary system and run effectively as intended.

Standard Libraries

Today, all programming languages consists of libraries where you can quickly select a program, make modifications are necessary, and execute your codes. Some of these libraries may have limited coding lines which will, therefore, require you to write your program. Python, on the other hand, comes with an extensive standard library which comprises of all your programming needs. For example, it consists of the MySQLdb library, which allows you to connect to the MySQL database without creating a pathway. As such, Python becomes among the leading programming tools to be used commercially when dealing with thousands of data as you can quickly retrieve and run with ease.

Free Open-Sources

Python also offers a fantastic free and open-source where you can use the tool in different areas, for example, in commercial use. Unlike other programming tools, a developer can choose to make changes on the program or instead select the desirable dataset to suit the field at hand mainly in the Python source code. While being used across several areas in the computing community, Python has experienced a constant increase in the usage, therefore, becoming more simple benefitting beginners significantly.

Downloading and Installing Python

Python, like most computer software, can be downloaded, run, and installed in a system for it to function with ease. However, this tool may become a bit challenging during download or updating, depending on the operating system. Some systems such as macOS and Linux typically accompany a preinstalled Python version which mostly is outdated. These versions of Python will hence require an update which usually uses unique techniques. On the contrary, other operating systems such as Windows and Android devices require a user to visit Python homepage or other relevant websites, download and install the software.

Python Development and Application

Python development is usually undertaken by the Python Enhancement Proposal (PEP), which has led to the creation of the most advanced and latest version. PEP has enhanced the features, Python documentation as well as the creation of bug fixes essential for eliminating problems arising during programming. Besides, it has managed to design modern coding processes as well as extending standard libraries to suit all developer needs during the creation of programs. In most cases, PEP collects information from developers utilizing Python and develops solutions on major problems raised.

When Python was first released in the 1980s, it accompanied multiple benefits but with numerous faults within the tool. Over the years, the Python Software Foundation has made significant modifications indicating differences between Python 1.0 and Python 3.7 used today. Python has henceforth gained popularity over time and applied in various areas in the computing community. For instances, the programming software has been used in the creation of Web apps in different websites such as Instagram, Mozilla, and Reddit. Other applications include the computation of both scientific and numerical values and the development of software prototype. Due to it's easy to use language, Python is widely used in educating children and beginners interested in learning computer language skills.

Python Variables

Python variables are named phases used to store codes in the system memory used mainly for the development of programs. Variables are critical in Python, especially for programmers who create complex programs in need of multiple code values. Unlike other programming software like Java and C++, Python doesn't demand variable declaration as they instantly change after being named. Python variables, therefore, are memory reserves used to store values fed to the program when needed. The data saved usually vary depending on the data type; for instance, they may be stored in the form of Numbers, Lists, Tuple, or as Dictionary.

Lists include ordered and changeable data written in the form of “my computer” with double-quotes. You can access values within the list using index numbers which are written up to negative integers. Dictionaries entail indexed and changeable variables but remain unordered and written with

single curly quotes. Accessing values in dictionaries consist of inputting a keyword in parentheses which also helps in other functions such as looping, making changes, and more. Numbers are of three forms int, float and complex representing different number value stored while tuple is data values which are ordered but remain unchangeable.

Chapter 1 Python Programming Basics



How to Install Python

In this time and age, being techy is a demand of the times, and the lack of knowledge, classifies one as an outback. This can result to being left out from the career world, especially in the field of programming.

Numerous big shot companies have employed their own programmers for purposes of branding, and to cut back on IT expenses.

In the world of programming, using Python language is found to be easier and programmer-friendly, thus, the universal use.

In this particular demo, we have chosen windows because it's the most common worldwide – even in not so progressive countries. We want to cater to the programming needs of everyone all over the globe.

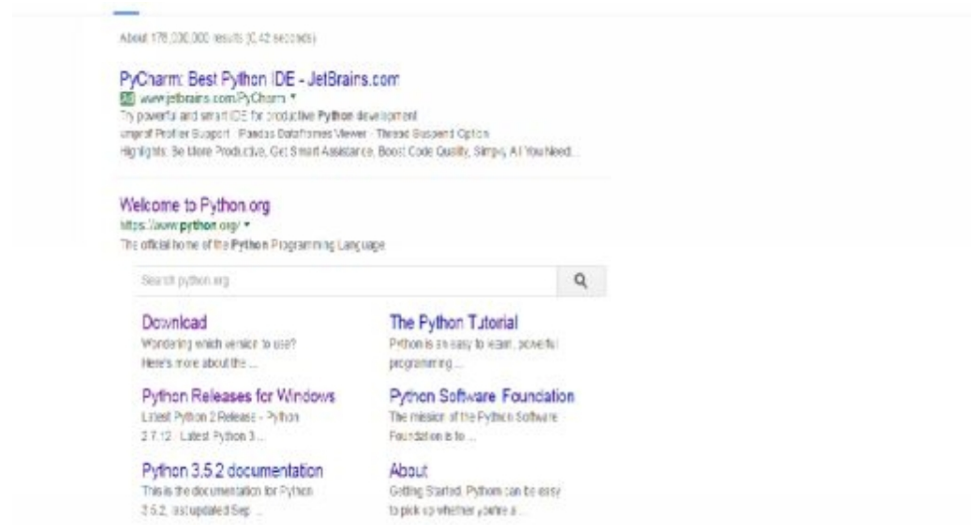
Python 2.7.12 version was selected because this version bridges the gap between the old version 2 and the new version 3.

Some of the updated functions/applications of version 3 are still not compatible with some devices, so 2.7.12 is a smart choice.

Steps in downloading Python 2.7.12, and installing it on Windows

Type python on your browser and press the Search button to display the search results.

Scroll down to find the item you are interested in. In this instance, you are looking for python. click “python releases for windows”, and a new page opens. See image below:



Select the Python version, python 2.7.12, and click, or you can select the version that is compatible to your device or OS.

Python Releases for Windows

- [Latest Python 2 Release - Python 2.7.12](#)
- [Latest Python 3 Release - Python 3.5.2](#)

- [Python 3.6.0b1 - 2016-09-12](#)
 - [Download Windows x86 web-based installer](#)
 - [Download Windows x86 executable installer](#)
 - [Download Windows x86 embeddable zip file](#)
 - [Download Windows x86-64 web-based installer](#)
 - [Download Windows x86-64 executable installer](#)
 - [Download Windows x86-64 embeddable zip file](#)
 - [Download Windows help file](#)

- [Python 3.6.0a4 - 2016-08-15](#)
 - [Download Windows x86 web-based installer](#)
 - [Download Windows x86 executable installer](#)
 - [Download Windows x86 embeddable zip file](#)
 - [Download Windows x86-64 web-based installer](#)
 - [Download Windows x86-64 executable installer](#)
 - [Download Windows x86-64 embeddable zip file](#)

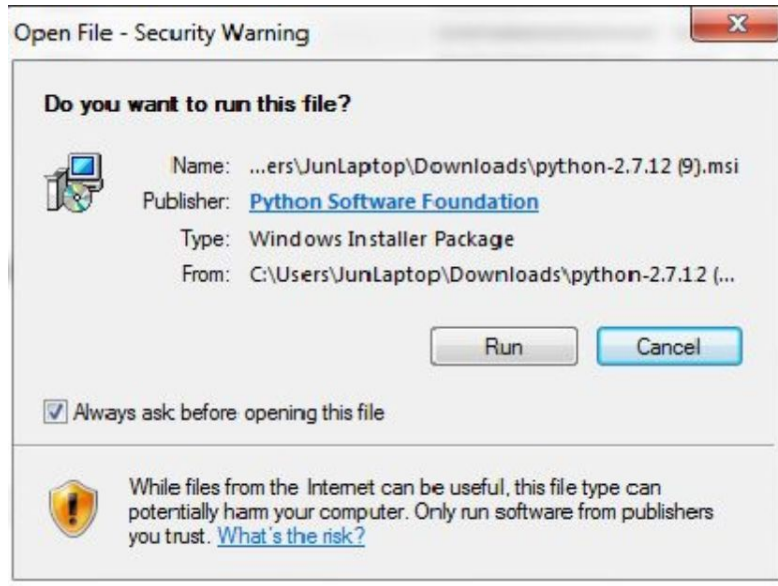
The new page contains the various python types. Scroll down and select an option: in this instance, select Windows x86 MSI installer and click.

Files

Version	Operating System	Description	MD5 Sum	File Size	GPG
Gzipped source tarball	Source release		88d61f02e9c16a+be952828b3694109d	16935960	SIG
XZ compressed source tarball	Source release		57dffce9cee6bb2ab5f82af1d8e9a69	12390820	SIG
Mac OS X 32-bit (i386)/PPC installer	Mac OS X	for Mac OS X 10.5 and later	3adbedc935a0db1ab08aa41f2ec4e33	24214628	SIG
Mac OS X 64-bit (x86_64) installer	Mac OS X	for Mac OS X 10.6 and later	06beddc2bced37335d27aa9d8f4952e1	22355024	SIG
Windows debug information files	Windows		1751598e16431be04e1f4f24ca52b53a	24678566	SIG
Windows debug information files for 64-bit binaries	Windows		c5433a7fca9edefe52835b440e40aa9d	25481382	SIG
Windows help file	Windows		7bc4e13eca8ede7c85e122f0a6d5f27	6224175	SIG
Windows x86-64 MSI installer	Windows	for AMD64/(EM64T)/x64, not Itanium processors	8fa13925db17638aa472a3e794ca4ee3	19820544	SIG
Windows x86 MSI installer	Windows		fe0ef5b6fd0272232f72843249349bd	18907136	SIG

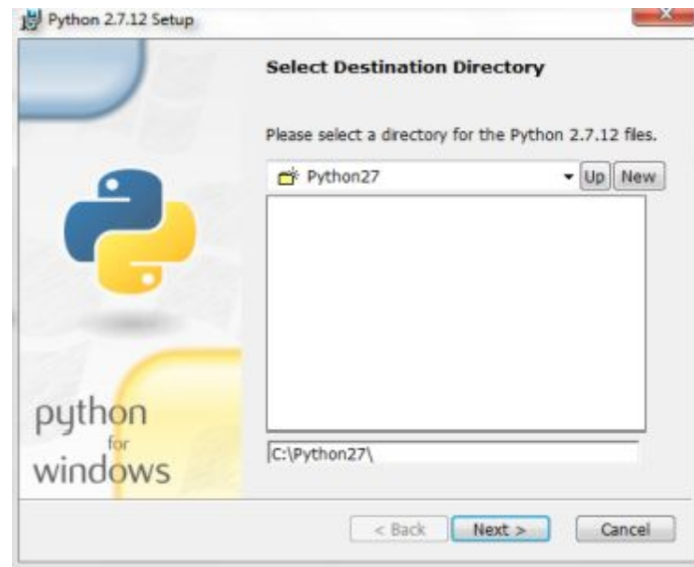
Press the Python box at the bottom of your screen.

Click the “Run” button and wait for the new window to appear.



Select the user options that you require and press “NEXT”.

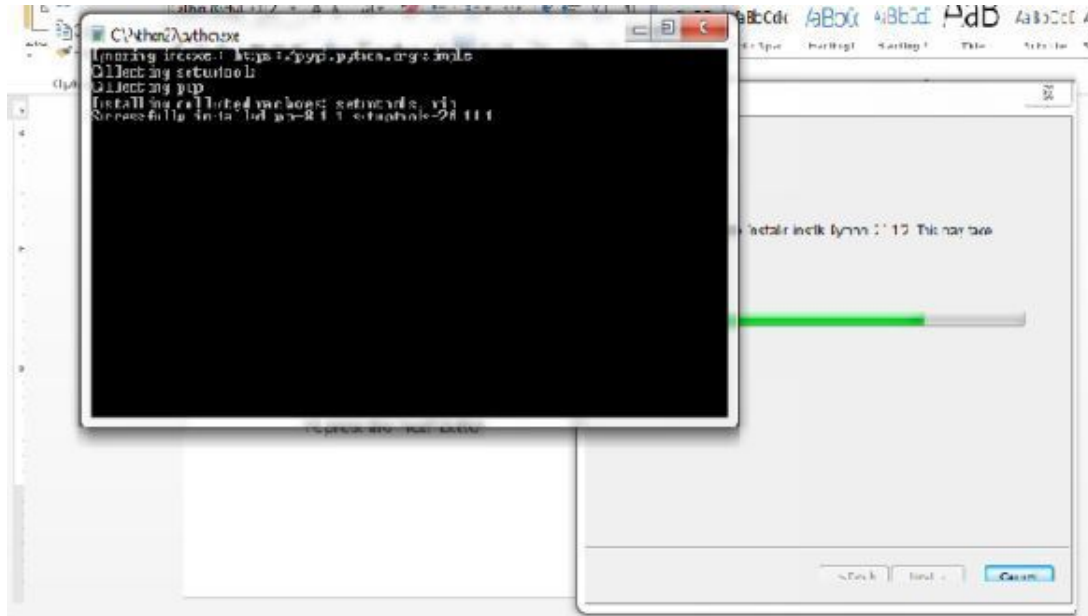
Your screen will display the hard drive where your python will be located.



Press the “NEXT” button.

Press yes, and wait for a few minutes. Sometimes it can take longer for the application to download, depending on the speed of your internet.

After that, click the FINISHED button to signify that the installation has been completed



Your python has been installed in your computer and is now ready to use. Find it in drive C, or wherever you have saved it.

There can be glitches along the way, but there are options which are presented in this article. If you follow it well, there is no reason that you cannot perform this task.

It's important to note that there's no need to compile programs. Python is an interpretive language and can execute quickly your commands.



Follow the step by step instructions prompted by the program itself. Save and run the program in your computer.

For Mac

To download Python on Mac, you can follow a similar procedure, but this time, you will have to access the “Python.mpkg” file, to run the installer.

For Linux

For Linux, Python 2 and 3 may have been installed by default. Hence, check first your operating system. You can check if your device has already a Python program, by accessing your command prompt and entering this: `python—version`, or `python3—version`.

If Python is not installed in your Linux, the result “command not found” will be displayed. You may want to download both Python 2.7.12 and any of the versions of Python 3 for your Linux. This is due to the fact that Linux can have more compatibility with Python 3.

For windows users, now that you have downloaded the program, you’re ready to start.

And yes, congratulations! You can now begin working and having fun with your Python programming system.

Basic Python Terms You Must Learn

Python is a language used in computer programming. As such, you must be familiar with the most commonly used lingo to facilitate your understanding of the language. It’s like learning the ABCs before you can read or write your first letters.

It’s important to remember that there may be slight variations with the different Python versions. The example here is from version 2. So, here goes:

Strings— are the values enclosed inside double, single quotes, or triple quotes. They can be a word/text, or a group of words, or a Unicode, or other items.

Example:

```
mystring = 'welcome'
```

```
mystring = "welcome"
```

```
mystring = 'My little corner.'
```

```
mystring = "My little corner."
```

The advantage of the double quotes is that you can include values within the double quotes.

The triple quotes signify long or lengthy strings. They are useful to avoid getting an EOL (End of the Line) error.

Variables –are containers for the strings. In the Python language, these are usually objects. These can be numbers or strings. Remember that you have to declare the variables, prior to using them.

The numbers can be floating point numbers or integers.

Use this syntax to define integers and floating point numbers. Integers are whole numbers, while floating point numbers are usually numbers with decimal points.

Example:

```
myint = 9
```

```
myfloat = 9.0
```

Statements –are stated sentences or syntax used to call a function to compute, to write a value, or other procedures needed in executing or performing Python commands.

Lists –are just like your ordinary lists for items you want to create. They can contain any variable/s that you want to include in your list. They can be comparable to arrays. The variables are usually enclosed in brackets, and the items or values are separated by commas. The semi-colon can be used between lists. Lists are immutable files–meaning they cannot be changed.

The word-values are enclosed in single or double quotes, while numbers are not.

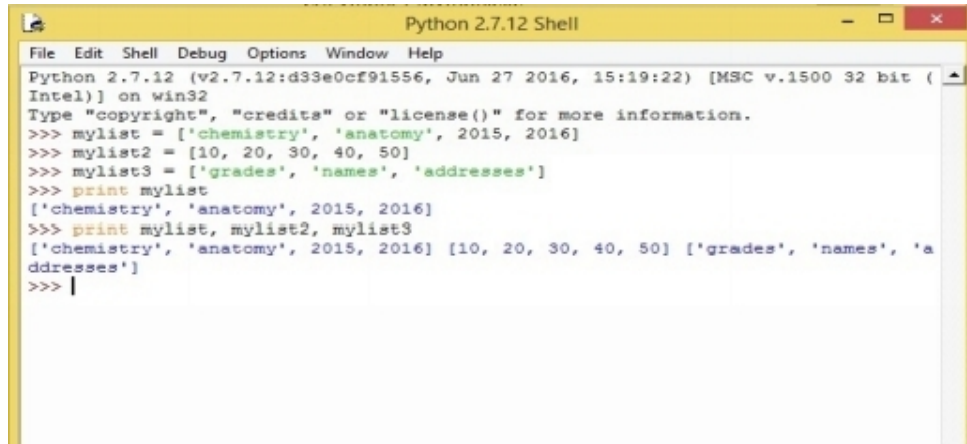
Example:

```
mylist1= ['chemistry', 'anatomy', 2015, 2016];
```

```
mylist2 = [10, 20, 30, 40, 50];
```

```
mylist3 = ['grades', 'names', 'addresses]
```

When you add the function 'print' and press 'enter' or execute, this will appear:

A screenshot of a Python 2.7.12 Shell window. The window title is "Python 2.7.12 Shell". The menu bar includes "File", "Edit", "Shell", "Debug", "Options", "Window", and "Help". The main text area shows the following code and output:

```
Python 2.7.12 (v2.7.12:d33e0cf91556, Jun 27 2016, 15:19:22) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> mylist = ['chemistry', 'anatomy', 2015, 2016]
>>> mylist2 = [10, 20, 30, 40, 50]
>>> mylist3 = ['grades', 'names', 'addresses']
>>> print mylist
['chemistry', 'anatomy', 2015, 2016]
>>> print mylist, mylist2, mylist3
['chemistry', 'anatomy', 2015, 2016] [10, 20, 30, 40, 50] ['grades', 'names', 'addresses']
>>> |
```

Take note of the variations of colors that can identify the command or function (red colored word), from the variables (green), and from the results (blue colored words).

Examples of double quotes:

```
mystring4=["Vanessa Redgrave", "Tom Cruise", "Mel Gibson", "Matt Damon"]
```

Loops – are statements that can be performed or executed one after the other – repeatedly, or once. There are two general types of loops, the ‘for’ and the ‘while’ loops..

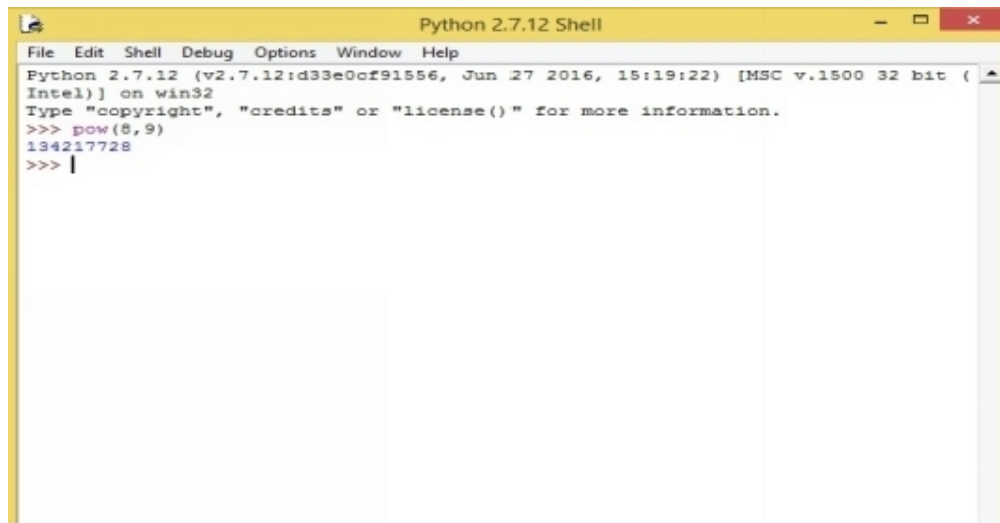
Function – a piece of code that executes some functions or logic. Examples are ‘print’, which prints your entry or variables; pow (power), which gets the answer for your numbers raised to a certain power. A specific example is this:

To know the value of 89, you can use the Python function (pow). On your Python shell enter this statement:

```
pow(8,9)
```

When you press the ‘enter’ or the execute key, the answer will appear:

In the specific example above, the answer is 134217728.



```
Python 2.7.12 Shell
File Edit Shell Debug Options Window Help
Python 2.7.12 (v2.7.12:d33e0cf91556, Jun 27 2016, 15:19:22) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> pow(8,9)
134217728
>>> |
```

Modules –are files that contain various items, such as variables, definitions of functions, and executable statements, among others. Modules are used in cases when you want to save functions that you have created for easier accessibility.

This is because after using the Python interpreter, all the definitions, variables and functions you have created will be lost. Hence, you need to compile them in modules, so you can use them again, whenever necessary.

Python will automatically ask you to save your files, so you will never forget this function. Your modules should be saved with the name you assigned to them, and the suffix - .py.

It's best to assign names based on the object, or the purpose of your module. This way, you can recall the names of the modules easily.

Shells –are the blank boxes where you write your Python commands or statements.

Tuples – are similar to lists; they are immutable. You cannot change them. However, you can create new Tuples to modify the old ones. They can also be used as keys to dictionaries.

Classes – are groups of related data, similar to strings, integers, and lists that use related functions. To introduce or identify a class, you can use the function word 'class'.

Objects – are used in Python language a great deal, because Python is object oriented. This means that the user can name his files according to what they are as a group, or as an individual value. An example is when a

user names his data about Geography as‘Geo”, or his research data on climate change, as‘climchange’.

Concatenation –a series of connected strings or variables use in Python programs. You can combine small strings to become larger strings by utilizing the‘join’ () procedure or the plus (+) sign.

There will be more Python terms that you will encounter as we proceed. Not all of them is included here.

Machine Learning is made up of many different segments that can retrieve, store and process data in a human fashion. By being able to do this, it brings its self from an uncertain position to a self-learn mode. So, when giving new information in a series of digits between 0 and 1, these electronic device programs are authorized to enlarge, learn, and develop to a high state of intelligence by themselves.

Machine Learning History

Deep Blue was an intelligent electronic device that defeated the titleholder of chess at his game in 1997. Did this happen by chance? No, this electronic device was programmed to do this and this alone. If given a different challenge, this electronic device probably would have malfunctioned. So, it was time to go back to the drawing board with a new program, new sense, and with new regulations. With that in mind, you can use algorithms that master new prototypes with proficiency, rather than depending on harsh coded regulations. You can say that this is truly the definition.



Machine Learning Types

The thought of using this type of learning came about years ago, but the qualification to independently give difficult mathematical thoughts (such as measurements or statistics) to an electronic device is a fresh idea that came to completion in this profession a few years ago. Likewise, there are different kinds of machine learning that work with algorithms with a fixed type of machine learning. Throughout this guide phase, we will discuss these types.

Supervised Machine Learning

Supervised machine learning can only handle simple tasks and algorithms that it recognizes. These tasks can only be carried out if the electronic device has access to preprogrammed data of input-output duos.

```
PYTHON
iris = datasets.load_iris()
X = iris.data[:, :2]
Y = iris.target
```

Summary

- Projecting Prototype
- Categorized data
- Classification Problems and Regression

Algorithms

- SVM's
- DT's
- K-NN
- Naïve Bayes
- Linear Reversion
- Neural Networks

Chapter 2 - What Is Scikit-Learn, and Why Should I Learn About It?

There are a lot of things that you are going to enjoy when it comes to the Scikit-Learn environment and library. This is one of the best options that you can work with through Python and will make your machine learning projects so much more successful than before. If you are a programmer who is learning how to work with the process of machine learning, or you want to do more with your Python codes, then you need to make sure you have a good understanding of this library and how it works.

The Scikit-Learn library was developed in 2007. Later, the company started growing and made a lot of changes over time. Currently, it gets to enjoy more than 30 active contributors, and there are even some paid sponsorships from the INRIA, Google, and the Python Software Foundation to ensure that this library is going to be developed. And it is all done in a way that ensures that the user is not going to have to pay to use it!

But this starts to bring up some questions about this library, and what it is all about. This library is going to ensure that the computer programmer has a lot of algorithms for both the unsupervised learning and the supervised learning that they want to do. And these algorithms are adjusted so that they can stay consistent with the Python environment. This means that you can use these algorithms and work on some machine learning projects all on Python.

This particular library is going to be licensed under what is known as a permissive simplified BSD license, and many of the Linux distributions can use it as well. It will be built using the SciPy library that will help make things even easier. The stack that is found inside of this, and which you will find helpful when you are working with machine learning includes:

1. NumPy: This is a good one to use because it allows you to work on the n-dimensional array package
2. SciPy: This one is going to be a fundamental library to use if you wish to do computations in the scientific field

3. Matplotlib: This is a good library to use because it helps you do some plotting, whether in 2D or 3D.
4. IPython: This is a good library to use because it is going to allow you a console that is more enhanced and interactive than others.
5. Sympy: This is a library that works well if you want to do some things in symbolic mathematics.
6. Pandas: This is the number one part that you need to use because it is going to include all of the analysis and the data structure needed to make machine learning successful.

The different extensions and modules that you can use with SciPy are known collectively as SciKits. This is why the module that provides us with the learning algorithms needed are going to be called the Scikit-Learn library.

The vision that is going to come in with this library will include a lot of support and robustness than you can find with some of the other topics that you explore. This is a good thing because both of these are going to require some higher levels to make sure that the production system works the way we expect and want. When going through this process, there has to be a deeper focus on the concerns, including ease of use, the collaboration, documentation, code quality, and performance, or it isn't going to work the way we want.

Knowing Some of the Features of This Library

At this point, we have talked a bit about this library, but we haven't gone into any of the details of the features, or the reasons that you would choose to work with this system over one of the others. When you decide to work with this particular library, you are probably going to be curious as to what it is all about, and even why some people want to work with this while learning and working with machine learning.

The Scikit-Learn library is going to be the most focused on modeling data. It isn't going to take that much time to look at how to summarize the data, load the data, or manipulate the data. If you want to work through these three topics, then you would want to spend some time in the libraries of

NumPy and Pandas. However, some of the features you can get through this library, and some of the different models that are available here include:

1. Supervised models: This Scikit-Learn library is going to provide you with many linear models (mostly generalized) that can work well in machine learning. This could include some of the algorithms like a discriminate analysis, decision trees, lazy methods, the Naïve Bayes, support vector machines, and neural networks, to name a few.
2. Manifold learning: These are important because they are often going to be used to help depict and summarize some of the multi-dimensional data that may seem hard to get through for a beginner.
3. Parameter tuning: This is a tool that you may find useful when you want to learn more and get more out of your supervised models.
4. Feature selection: This is going to be a part of the library that is going to help you see, and then identify meaningful attributes from creating a new supervised model.
5. Feature extraction: This one is helpful because it is going to help you learn how to define the attributes in text data and images presented to the machine.
6. Ensemble methods: You will enjoy this feature because it is going to be helpful when you combine the predictions that you have from several models of supervised learning, and then have these all come together to form a new prediction using all that information.
7. A reduction in dimensionality: This is a method found in this library that is helpful when you would like to find a way to reduce the number of attributes needed in data to help with feature selection, visualization, and summarization. A good example of how this works is going to be the principal component analysis.
8. Data sets: This is helpful because it is the place where you can test some of the sets of data that you have, the ones that

are going to generate the right sets of data with the right properties so you can do a bit of investigating.

9. Cross-validation: There are times when you will want to figure out whether a model is giving you accurate results or not. The cross-validation will help you get an estimation on how well that model is going to perform on the data that you have without it even seeing that data.

10.

Clustering this is where you can group any data that doesn't have a label, such as with the K-means method.

These are just a few of the benefits that you are going to be able to get when it comes to working with this library. It is a strong library to use, one that is going to provide you with the results and answers that you want to many supervised machine learning algorithms. Without this in place, it is going to be hard to figure out how you would like to do these algorithms at all and determine if the information you get is accurate when you work with machine learning.

Chapter 3 - Essential Libraries And Tools In Python

The library that we need to take a look at is known as PyTorch. This is going to be a Python-based package that works for scientific computing that is going to rely on the power that it can receive from graphics processing units. This library is also going to be one of the most common, and the preferred, deep learning platforms for research to provide us with maximum flexibility and a lot of speed in the process. There are a lot of benefits that come with this kind of library. And it is known for providing two of the most high-level features out of all the other deep learning libraries. These will include tensor computation with the support of a strong GPU acceleration, along with being able to build up the deep neural networks on an autograd-system that is tape-based.

There are a lot of different libraries through Python that can help us work with a lot of artificial intelligence and deep learning projects that we want to work with. And the PyTorch library is one of these. One of the key reasons that this library is so successful is because it is completely Pythonic and one that can take some of the models that you want to build with a neural network almost effortlessly. This is a newer deep learning library to work with, but there is a lot of momentum in this field as well.

The Beginnings of PyTorch

As we mentioned above, PyTorch is one of the newest libraries out there that works with Python and can help with deep learning. Even though it was just released in January 2016, it has become one of the go-to libraries that data scientists like to work with, mainly because it can make it easy to build up complex neural networks. This is perfect for a lot of beginners who haven't been able to work with these neural networks at all in the past. They can work with PyTorch and make their network in no time at all, even with a limited amount of coding experience. The creators of this Python library envisioned that this library would be imperative when they wanted to run a lot of numerical computations as quickly as possible. This is one of the ideal methodologies that also fits in perfectly with the programming style that we see with Python. This library, along with the Python library, as

allowed debuggers of neural networks, machine learning developers, and deep learning scientists to not only run but also to test, parts of their code in real-time. This is great news because it means that these professionals no longer have to wait for the entire code to complete and execute before they can check out whether this code works or if they need to fix certain parts.

In addition to some of the functionality that comes with the PyTorch library, remember that you can extend out some of the functionalities of this library by adding in other Python packages. Python packages like Cython, SciPy, and NumPy all work well with PyTorch as well.

Even with these benefits, we still may have some questions about why the PyTorch library is so special, and why we may want to use this when it is time to build up the needed models for deep learning. The answer with this is simple, mainly that PyTorch is going to be seen as a dynamic library. This means that the library is flexible and you can use it with any requirements and changes that you would like. It is so good at doing this job that it is being used by developers in artificial intelligence, students, and researchers in many industries. In fact, in a Kaggle competition, this library was used by almost all of the individuals who finished in the top ten.

While there are a lot of benefits that can come with the PyTorch library, we need to start with some of the highlights of why professionals of all sorts love this language so much. Some of these include:

1. The interface is simple to use. The PyTorch interface is going to offer us an API that is easy to use. This means that we will find it simple to operate and run as we do with Python.
2. It is Pythonic in nature. This library, since it is considered Pythonic, will smoothly integrate to work with the Python data science stack. Those who do not want to work with other coding languages along the way, and want to just stick with the basics, and some of the power, of Python, will be able to do so with this library. You will get the leverage of using all of the functionalities and services that are offered through the environment of Python.
3. Computational graphs: Another highlight that comes with the PyTorch library is that it is going to provide us with the

platform with some dynamic computational graphs. This means that you can change these graphs up even during runtime. This is going to be useful any time that you need to work on some graphs and you are not sure how much memory you need to use while creating this model for a neural network.

The Community for PyTorch

The next thing that we need to take a look at here is some of the community that comes with the PyTorch library. Because of all the benefits that come with PyTorch, we can see that the community of developers and other professionals is growing daily. In just a few years, this library has shown a lot of developments and has even led this library to be cited in many research papers and groups. And when it comes to artificial intelligence and models of deep learning, PyTorch is starting to become one of the main libraries to work with.

One of the interesting things that come with PyTorch is that it is still in the early-release beta. But because so many programmers are adopting the framework for deep learning already, and at such a brisk pace, it already shows the power and the potential that comes with it, and how the community is likely to continue growing. For example, even though we are still on the beta release with PyTorch, there are currently 741 contributors just on the GitHub repository right now. This means that more than 700 people are working to enhance and add some improvements to the functionalities of PyTorch that are already there.

Think of how amazing this is! PyTorch is not technically released yet and is still in the early stages. But there has been so much buzz around this deep learning library, and so many programmers have been using it for deep learning and artificial intelligence, that there are already a ton of contributors who are working to add some more functionality and improvements to this library for others to work with.

PyTorch is not going to limit the specific applications that we are working with because of the modular design and the flexibility that comes with it. It has seen a heavy amount of use by some of the leading tech giants, and you may even recognize some of the names. Those who have already started to work with PyTorch to improve their deep learning models will include

Uber, NVIDIA, Twitter, and Facebook. This library has also been used in a lot of domains for research including neural networks, image recognition, translation, and NLP among other key areas.

Why Use PyTorch with the Data Analysis

Anyone who is working with the field of data science, Data Analysis, artificial intelligence, or deep learning has likely spent some time working with the TensorFlow library that we also talked about in this guidebook. TensorFlow may be the most popular library from Google, but because of the PyTorch framework for deep learning, we can find that this library is able to solve a few new problems when it comes to research work that these professionals want to fix.

It is often believed that PyTorch is now the biggest competitor out there to TensorFlow when it comes to handling data, and it's going to be the one of the best and most favorite artificial intelligence and deep learning library when it comes to the community of research. There are many reasons for this happening, and we will talk about some of these below:

First, we will notice that the dynamic computational graphs are going to be popular among researchers. This library is going to avoid some of the static graphs that can be used in other frameworks from TensorFlow. This allows researchers and developers to change up how the network is going to behave at the last minute. Some of those who are adopting this library will like it because these graphs are more intuitive to learn when we compare it to what TensorFlow can do.

The second benefit is that this one comes with a different kind of back-end support. PyTorch is going to use a different backend based on what you are doing. The GPU, CPU, and other functional features will all come with a different backend rather than focusing on just one back-end to handle all of these. For example, we are going to see the THC for our GPU, and TH for CPU. Being able to use separate backends can make it easier for us to deploy this library through a variety of constrained systems.

The imperative style is another benefit of working with this kind of library. This means that when we work with this library, it is easy to use and very intuitive. When you execute a line of code, it is going to get executed just as you want, and you are able to work with some tracking that is in real-time.

this allows the programmer to keep track of how the models for neural networks are doing. Because of the excellent architecture that comes with this, and the lean and fast approach, it has been able to increase some of the overall adoptions that we are seeing with this library throughout the communities of programmers.

Another benefit that we are going to enjoy when it comes to working with PyTorch is that it is easy to extend. This library, in particular, is integrated to work well with the code for C++ and it is going to share a bit of the backend with this language when we work on our framework for deep learning. This means that a programmer is going to be able to not just use Python for the CPU and GPU, but it can also add in the extension of the API using the C or the C++ languages. This means that we can extend out the usage of PyTorch for some of the new and experimental cases, which can make it even better when we want to do some research with it.

And finally, the last benefit that were going to focus on here is that PyTorch is going to be seen as a Python approach library. This is because the library is a native Python package just by the way that it is designed. The functionalities that come with this are built as classes in Python, which means that all of the code that you write here can integrate seamlessly with the modules and packages of Python.

Chapter 4 – Learn Modules

Modules are basically files with some Python code in them, and we use the module to organize the code into multiple files. This is a way to keep your code clean and well-classified. Just like when we walk into a pharmacy and find that they have all of their medicine placed in different drawers; we do the same in Python. No one wants to have one mega file ‘app.py’ with all of our functions and classes. What we do instead is simply break down pertinent parts into individual files. Think how much more productive would that be while working with a team. Also, how much easier it is to maintain the program as a whole; if you need to upgrade your payment features, you would just need to pull out that particular module (file) and update it. Moreover, given that Python is an open-source language, you can reuse and adapt modules with a simple import statement.

Creating a Module

As mentioned before, a module is simply a file. So, to begin creating a module, we simply need to create a file. To do that, we will maximize the project tool window, which is the left pane on PyCharm’s interface. Now we will create a new file by right clicking on our project folder and selecting ‘New’, then clicking on ‘File’. At this point we need to give our file a name followed by the dot py extension (example: ‘convert.py’). Remember the general naming rules still apply; do not add spaces, special characters or reserved names. In our example, we are going to create a module that will have the ability to convert values from the empirical to the metric system.

```
convert.py
```

1. `def mi2KM(miles):`
2. `Return round (miles * 1.609344) #Rounding up the kilometers calculated and returning the value.`
3. `def km2MI(KM):`
4. `return round(KM/1.609344)`

```
#Rounding up the miles calculated and returning the value.
```

That is basically it. You just created a module, and you do not need to worry about anything in this file anymore. Just remember only to add functions for conversions in this file.

Using A Module

To be able to make your new module, just follow the same steps. Only this time, type 'import convert' instead of 'import math.' Note that we left out the file extension '.py' while importing the module.

MyFirstProgram.py

1. Import convert # importing the convert module.
2. Print (convert.km2MI (200)) # printing the result of the conversion.
3. Print (convert.mi2KM (40)) # printing the other result of the conversion.

Program output:

```
C:\Users\...\PycharmProjects\GettingStarted\venv\Scripts\Python.exe  
e C:/Users/.../PycharmProjects/GettingStarted/MyFirstProgram.py
```

```
124
```

```
64
```

```
Process finished with exit code 0
```

Renaming a Module

Renaming a module is fairly simple and will only take one line of code. We use this kind of practice when we know that a certain module name will confuse the developers in a program. Let's say that you are creating a program that uses the label 'math' frequently. It could be a learning platform for kids where they practice math equations, for example. Anyway, you decide that you would like to rename the math module that you are going to import. That can be done by simply adding the word 'as' followed by the new name that you want to use.

1. `Import math as moo # importing the math module and renaming it to moo.`
2. `c = 2 * mo. * 10 # Calculating the circumference of a circle.`
3. `Print(c) # printing the result.`

Program output:

```
C:\Users\...\PycharmProjects\GettingStarted\venv\Scripts\Python.exe  
e C:/Users/.../PycharmProjects/GettingStarted/MyFirstProgram.py
```

```
62.83185307179586
```

```
Process finished with exit code 0
```

Built-In Modules

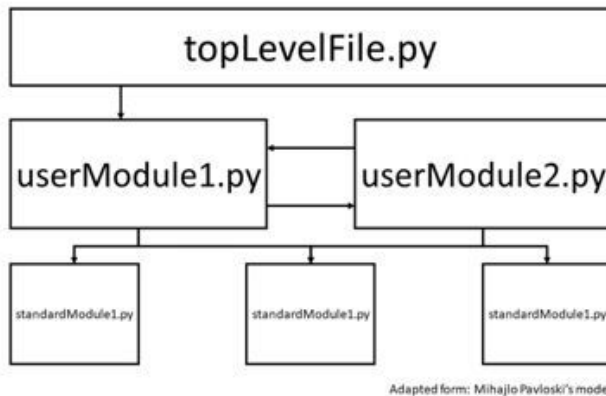
There is a plethora of modules, and it is constantly being updated and built into Python.

As you progress in your programming abilities and build complex programs, these modules will make your life easier. There is absolutely no need to re-invent the wheel if the community has already given you plenty to work with.

One thing that you need to keep in mind is the structure of the program as a whole. Any non-trivial program will probably have three types of modules.

- The top-level file: This is the main file that is used to run all of your script and to launch the main application.
- User-defined modules: These are modules that have been created by the developer for specific use in the application.
- Standard library modules: These are modules that are built into Python and are not part of the executable itself (your program), but pertinent to the standard Python library.

The way complex programs should be built would consist of the top level file, which then calls the user-defined module. The standard Python modules should be called within the user defined ones. The figure below does not constitute the way you must design your program structure, but is



simply an example of how

sophisticated programs operate.

Import from module

There are cases where you do not need to import complete modules, just specific functions in the module. Doing that is simple. We'll just use a slightly different syntax. This imports the function itself as if it were defined in the same file. There is no need for a dot notation in this case.

1. From convert import mi2KM # importing a specific function from another module.

printing the result of our function.

2. Print (mi2KM (200)) # Notice that the function is called as if it were defined in the same file.

Program output:

```
C:\Users\...\PycharmProjects\GettingStarted\venv\Scripts\Python.exe
C:/Users/.../PycharmProjects/GettingStarted/MyFirstProgram.py
```

322

Process finished with exit code 0

Chapter 5 - Machine Learning Datasets

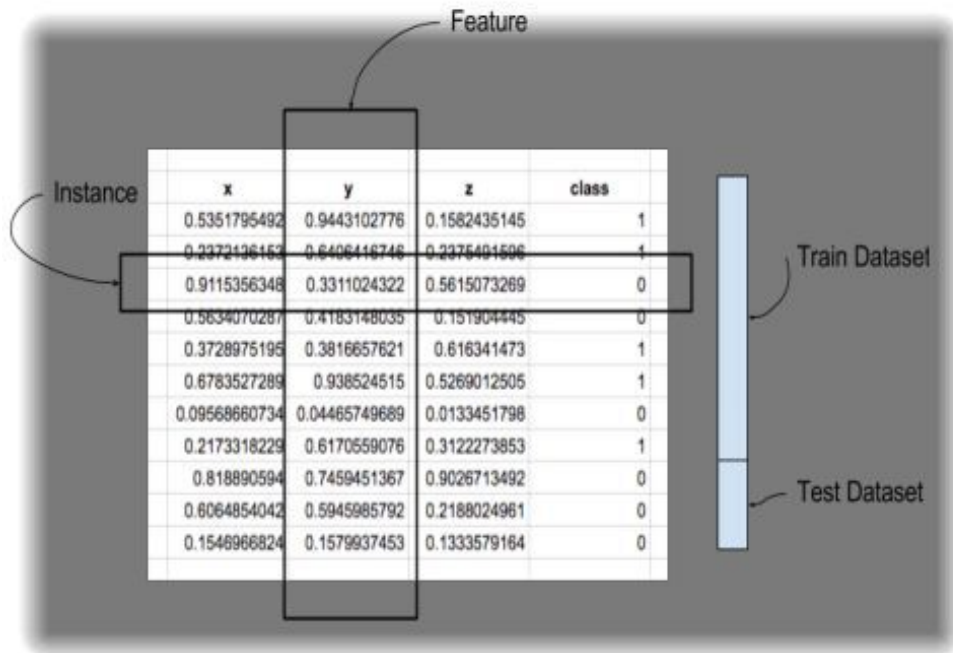


Numerous concepts underpin the general knowledge of machine learning. These components lay the foundation for understanding the field as well as making the processes a go.

Data

As we have seen earlier, machine learning algorithms learn from examples. The most crucial element to note in this phase is the input data and various terms used to describe these sets of data.

In this phase, I would like to explain data in terms of rows and columns, similar to an excel spreadsheet. Data analyzed in this method reflects the traditional structure that is commonly used in machine learning. The image below is a representation of sets of data used in machine learning algorithms.



An image of data sets

Defining the terms

Instance: an instance is a single line (row) of data in a machine learning algorithm. A case is an inference from the realm.

Feature: a feature is a single column of data. An element is a constituent of observation and is also known as the attribute of a data instance. A function may also be called a metric or a parameter. Some of these metrics may be data inputs to an algorithm (the training data, also called the predictors) while others may be outputs (the parameters to be determined or predicted).

Data Type: metrics have different forms of data. Types of data may be actual or integer-valued. In other instances, the type of data may have a clear-cut or standard value. Additional classes may be in the form of strings, times, lengths, etc. When traditional machine learning techniques are used, the various types of data are reduced to real values.

Datasets: a dataset can be defined as a group of instances. Machine learning models usually require a few sets of data for various functions.

Training Dataset: a set of data that is fed into the system to help guide the algorithm is called a training dataset. As the name suggests, they train the system.

Testing Dataset: After training the machine learning model, it is essential to test their performance. The sets of data that are used to evaluate the accuracy of the algorithm are called testing datasets. In other terms, they are called validation or evaluation datasets.

Learning

The focus here is based on machine learning. As seen, machine learning is all about training algorithms to perform tasks. In this perspective, let's take a look at some concept of learning.

Induction: the algorithms in various models of machine learning are and known as induction or inductive learning. Induction is a process of deciding sweeping statements (generalization) from detailed information (training datasets).

Generalization: The notion is also called sweeping statements. The process is vital because the representation set by a machine learning algorithm is required to make forecasts. Such findings are based on specific data instances not present at the training phase.

Over-Learning: over-learning is described as the situation where an algorithm of machine learning observes the features in the training data too much but still fails to perform generalization. This situation is brought about by poor performance on data other than the training dataset. This instance is also known as over-fitting.

Under-Learning: under-learning is a situation whereby machine learning algorithms do not observe the training datasets to completion. The process may be caused by early termination of the training process — under-learning results in good at all but poor performance on all data, not excluding the training dataset. The situation is also described as under-fitting.

Online Learning: online learning methods are instances when the data instance from a domain is used to update the system as the data become available. In online learning, robust processes that are not affected by noisy data are used. These methods can produce models that correspond to the current state domain.

Offline Learning: Offline learning is when a technique is produced on data that is pre-prepared to be used operationally on unlearned data. The process

of training the system can be carefully tuned since the training dataset is known. After the preparation of the model, no further updates are made. This scenario may lower the performance in case there is a change in domain.

Supervised Learning: This is a method of machine learning that is used to generalize problems where a forecast is necessary. A “training process” compares predictions by the model to known answers and makes corrections in the model. **Unsupervised Learning:** as we talk earlier, unsupervised learning is the process of generalization without any training data set being fed to the system. In this technique, naturally occurring structures are observed and utilized to interpret the relationship between the instances.

Modeling

A structure that is created by the process of machine learning is considered a model. The method of producing models is modeling. **Model Selection:** the process of model selection can be described as the process in which configuration and training of a model of machine learning are done. In every given time, the system should present different models to modify, use, or discard. The choice of which algorithm to install in machine learning is also a part of model selection. For a given problem, the algorithms of machine learning will pick a model in which it operates. **Inductive Bias:** a bias is an impartial input in a given model of machine learning. All machine learning models are biased to induce an error in the model. The failure caused is usually used to test the model or to act as a control set of data. Bias can be produced in a model during the configuration phase of the training phase. A machine learning algorithm can have a low or high preference depending on the input data. There are methods to remove high bias in algorithms.

Model Variance: variance the sensitivity of a model towards a training data set. Generally put, a variation is how a model responds to data input for training. A machine learning model can show high or low variance to training datasets. The best way to reduce the difference is by running the machine on a dataset over and over again. This process is done under different parameters and conditions. When variance has diminished the accuracy of the model is enhanced.

Bias-Variance Tradeoff: a model selection process can be regarded as a trade-off of the bias and the variance. The relationship between bias and variance is inverse proportionality. A model with low bias will possess high variation and will require longtime and continuous training to get a usable model. On the other hand, a model with high bias will maintain low variance, which makes it learn faster, but realizes adverse and limited performance.

Chapter 6 - Training Simple Machine Learning

Algorithms for Classification

People are living in one of the most defining periods of human history. However, it is not what is happening now that makes the period so exciting; rather, it is what is coming in the near future. Today, data scientists and developers can use complex algorithms to build powerful data-crunching machines that would have seemed like something out of science fiction only a few years ago.

There are two ways to group the main algorithms, these are:

1. Grouping them by their similarity in function or form
2. Grouping them by their learning style

Grouping algorithms by their similarity in terms of how they work is a useful method of categorization. However, it is important to understand that certain algorithms can easily fit into several different categories.

On the other hand, it is common in artificial intelligence and machine learning to first look at the different learning styles that an algorithm can use. Algorithms have a small number of machine-learning models and learning styles. This way of grouping them is useful because it forces developers to consider the model preparation process and the roles of the input data to find the most appropriate option needed to achieve the best results. The main learning styles in algorithms include supervised learning, unsupervised learning, and semi-supervised learning.

1. Supervised Learning

Supervised learning aims to predict a target/dependent variable from a given set of independent variables or predictors. Using these predictors, developers generate functions that match inputs to desired outputs. The training or learning process continues until the algorithm achieves the desired level of accuracy. Examples of this type of learning model include Logistic Regression, KNN, Random Forest, Decision Tree, and others.

2. Unsupervised Learning

An algorithm using unsupervised learning does not have any outcome or target variable to estimate or predict. It is ideal for clustering or grouping population, which is helpful when it comes to segmenting consumers in different groups to achieve certain objectives. Examples of this type of algorithm are K-means and Apriori algorithm.

3. Reinforcement Learning

Developers use this type of algorithm to train machines to take specific actions. Essentially, they expose their machine to certain environments where it continually trains itself using trial and error, thereby learning from experiences. In the process, the machine gains more knowledge to make accurate decisions. An example of this type of machine learning is the Markov Decision Process.

4. Semi-Supervised Learning

This model falls somewhere between supervised and unsupervised learning. Developers combine some features present in both models to produce the desired outcome. It is one of the most important and useful algorithms for real-world scenarios, where all the information available is a combination of unlabeled and labeled data.

Common Machine Learning Algorithms

Most machine-learning algorithms are applicable to almost any data problem. Some of the commonly used learning algorithms include:

Linear Regression

Initially developed to analyze the relationship between input and output numerical values in statistics, linear regression soon caught the attention of the machine learning community. Developers use it to estimate or predict real values based on continuous variables. These values can include total sales, home costs, number of calls, and more. It works by establishing a relationship between dependent variables and independent variables by fitting a regression line, also called the best-fit line.

A regression line in the form of $y = a + bx$ represents linear regression, where 'y' is the output variable and 'x' is the input variable. The goal,

therefore, is to determine the values of coefficients 'a', which is the intercept, and 'b', which is the slope of the regression line. The aim is to plot a line that is near most of the points, which would minimize the error or distance between the line and the 'y' value of the data point.

A good way to understand this concept is to use the example of a kid asked to arrange his/her classmates based on their weights without measuring their weights or asking them to state their weights. Obviously, the kid will analyze them visually by looking at their build and height and arrange them accordingly. This is a real-life application of linear regression. Essentially, he/she will know that build and height would have a relationship to the weight, which explains the equation above.

There are two types of linear regression. These are:

1. Simple linear regression
2. Multiple linear regression

SLR comprises a single independent variable, while MLR, as the name suggests, has multiple independent variables. To find the best-fit line, one needs to fit a curvilinear or polynomial regression.

Sometimes, instead of a straight line, a curve may represent the relationship between two different variables. Nevertheless, in terms of linear regression, this does not mean that such relationships are non-linear. Rather, this is a type of relationship where an increase in one variable will lead to an increase in the other variable up to a certain level. Afterward, as continues to increase, the other one will decrease.

Therefore, graphing this type of curvilinear relationship will produce an inverted-U. A curvilinear relationship might also happen whereas one variable goes up, the other one goes down up to a certain point, after which, both go up together, which will produce a U-shaped curve.

A real-world example of this type of relationship is the relationship between customer satisfaction and employee satisfaction. When the level of employee satisfaction is high, customer satisfaction tends to go up as well, but up to a certain point. When employees are too cheerful or happy, customers may perceive it to be annoying or fake, which would lower their level of satisfaction.

Polynomial regression fits a non-linear relationship between the independent variable and the corresponding conditional mean of the dependent variable. Developers hypothesize certain relationships as curvilinear. Often, such relationships will have a polynomial term. If they try to fit a linear model to such relationships, the result achieved will not be appropriate. In usual MLR, the assumption is that all independent variables are independent; however, this assumption does not satisfy polynomial regression.

Often, developers use polynomial regression to describe or define this type of non-linear relationships, which might include:

1. The spread of disease epidemics
2. Tissue growth rates
3. Carbon isotope distribution in lake sediments

By using polynomial regression, developers or researchers can fit a wide range of functions under it to fit a wide range of curvature. In addition, this regression model offers the best approximation of the relationships between independent and dependent variables. However, it is important to understand that polynomial regression is extremely sensitive to outliers in the data.

Logistic Regression

Whereas predictions for linear regression are continuous values, those applying to logistic regression are discrete values after employing a transformation function. This model is ideal for binary classification, which refers to data sets where ($y = 1$ or 0), where 1 identifies the default class. Suppose a researcher wants to predict whether something will happen or not. In this case, there are only two possibilities; i.e., it will happen, which he will denote as (1), or it will not happen, which will be (0).

That said, in spite of its name, it is important to understand that this is a classification algorithm, not a regression technique. Many experts, however, insist it is a regression model. Essentially, this algorithm estimates discrete values, such as true/false, yes/no, or 0/1, based on independent variables. For example, if a teacher gives a student a puzzle to solve, the student will either solve it or fail to solve it. In linear regression, the assumption is that data follows a linear function. In the same way, logistic regression

algorithms apply the sigmoid function when it comes to modeling data. Only after reaching the decision threshold does logistic regression become a classification model; therefore, this threshold is a critical aspect of this type of algorithm. The values of precision and recall are the main determinants of the value of the threshold. Ideally, both should be 1; however, this is rarely the case. Therefore, to choose the right threshold value, the following factors come into play:

1. High Recall/Low Precision

In programs where developers want to minimize false negatives without reducing the number of false positives, they often go for a threshold value with a high value for recall and low value for precision. A real-world application of this model is an application to diagnose cancer.

Users do not want the application to classify patients with cancer as healthy without considering the possibility of a wrong diagnosis. This is because further medical issues or procedures can confirm the absence of cancer, but the detection of cancer cannot happen in an already cleared or rejected candidate.

2. Low Recall/High Precision

In programs where developers want to lower the occurrence of false positives without lowering the number of false negatives, they choose this decision or threshold value. For example, if their aim is to classify consumers based on their positive or negative reaction to a certain type of marketing strategy, they want to be sure of a positive reaction. Otherwise, a negative reaction can have a terrible impact on the company's sales and revenue.

Therefore, based on the broad range of categories, logistic regression can be ordinal, multinomial, or binomial. In other words, it can focus on target variables with ordered categories, target variables with more than two possible disordered categories, or target variables with only two possible types of categories.

Decision Tree

This is one of the most popular algorithms used in machine learning. Every time people ask themselves a question before making a choice or decision,

their brain is working like a decision tree. For example, is it raining outside? If yes, I will take an umbrella. Essentially, the aim of this algorithm is to break down information into smaller pieces of data based on a particular feature value, until all variables targeted fall under a single category.

For instance, suppose a developer wants to build a decision tree to determine whether an individual is a child or an adult based on height and weight. He could break down the data points based on a set of values of one of the two characteristics, such as, for a weight greater than 40 kg, the individual is an adult. However, if it is less than that, his subset will contain two kids and one adult, which means he will need to separate it again, until only one class remains, or until all his subsets are pure.

This is a type of supervised machine learning. The aim of this type of algorithm is to solve both classification and regression problems. Decision tree algorithms use the representation of a tree to solve problems, whereby each leaf node represents a class label, while the internal node of the tree represents attributes. Using this algorithm, developers can represent any type of Boolean function on certain attributes.

This type of algorithm makes several assumptions, including:

1. There is a recursive distribution of records based on attribute values
2. The whole training set is the root at the beginning
3. The use of statistical methods to order attributes as the internal node or root
4. Feature values should be categorical

The major challenge in a decision tree algorithm is the process of identifying the attribute for the root node, also referred to as attribute selection.

Chapter 7 - Unsupervised Machine Learning

Now that we have explored a bit about supervised machine learning, it is time to explore other options that you can work with when it comes to machine learning. The first one that we spent some time talking about was supervised learning. Learning is designed in a way where you will show the computer some examples, and then you teach that computer how you would like it to respond based on the given examples. There are going to be a lot of programs where this technique is going to end up working well for you. But, when you think about showing hundreds or thousands of different examples to your computer, it is all going to seem pretty tedious. And then there are times when the program isn't going to be able to learn this way and still give you the expected results. This is where the other two types of machine learning are going to come into play.

This is where you will find unsupervised machine learning is going to come into play. This Phase is going to spend some more time talking about unsupervised machine learning and what it is all about. Unsupervised machine learning is going to be a type of learning that is going to happen if your algorithm makes mistakes and can learn from these mistakes along the way. And the program can do it even without having an associated response to work from.

This may sound a bit confusing, but it is going to be when you can teach the computer through trial and error, without it having to work with a million examples to make sure it behaves how you would like it to do. With these different algorithms, it is possible they figure out and analyze the patterns in the data based on any provided input from you or the user.

The good news here is that there are going to be a few different algorithm types that you can work with when you decide to choose unsupervised machine learning. The algorithm that you choose to work with is going to take the data that you have, and it will restructure it so that the data can fall into one of your classes.

These classes are nice because they make it easier for you to see the information nice and sorted out, and it makes it so much easier for you to look through the information. There are many times when you will use this machine learning because it can set up your computer, or another device, to

do most of the work of learning, without having a person sit there and writing out all of the instructions. The computer will do some trial and error and figure out how it should act over time.

Let's take a look at an example. If you have a company that has a considerable amount of data that they want to read through, such as data they want to use to make predictions and make decisions about how to act in the future, and you may want to work with the machine learning. You don't want to have one or two people go through this information. It would take up too much time and effort to get this all done. But unsupervised machine learning is going to do the work for you. Search engines also often use unsupervised machine learning.

Working With Unsupervised Learning

Before we get too far into some of the techniques that can be used with unsupervised machine learning, it is essential to understand a little bit about it. When you are working in a real-world environment, when are going to be times your machine, either in artificial intelligence or a robotic role, won't be able to access the optimal answer with the information that you provided. Maybe there isn't even an optimal answer to the question. You want to make sure that this robot or machine can explore the world all on its own, and learn how to do things just by taking a look at the patterns available.

In most cases, unsupervised learning is for learning the structure, or even the probability distribution of data. But what does this mean? We are going to spend some time talking about some of the different examples where you can use unsupervised learning to help get things done.

There are a few different ways you will be able to work with unsupervised learning. Often, it is to help the computer or the machine find the answers or needed results, without you having to explain it out. With unsupervised learning, the computer is going to learn how to behave based on past performance and feedback. This can be nice because you will be able to insert the information that you want, and the computer will do the work.

Density Estimation

You should know at this point that we use the probability density function, or PDF, to tell us the probability that will occur of a random variable. Density estimation is the process of taking samples of data of the random variable and figuring out the probability density function. After you learn the distribution of the variable, you will be able to use machine learning to generate your samples of the variable based on this information.

For example, at a higher level, you could take Shakespeare and learn its distribution. You can then take this information and generate out a text that looks very similar to what you would find with Shakespeare.

Latent Variables

Many times, you will want to know some of the underlying or the hidden causes of the data you are looking at. You can consider these hidden, missing, or latent variable. For example, say that someone gives you a set of documents, but they don't tell you what these documents are. You would be able to use the clustering option with machine learning to find out that there are a few distinct groups of information in the document. The machine learning would be able to tell you this information rather than you having to read through it all. After you have done clustering, you can read through a few of the documents in this data set and find out that maybe one is children's books, one is a romance novel, and so on. There are several different types of clustering that you can work with. You would use this information any time when the data is too big that it doesn't make sense to go through all of it on your own. The clustering process will be able to summarize the data to help you sort it all out.

There are a few different techniques that you can use when it is time to work with machine learning of this kind. Some of the methods or the algorithms that tend to work the best here are going to include:

- Neural networks
- Markov algorithm
- Clustering algorithm

Let's explore each of these topics and learn a bit more about how they are all going to work, and some of the basics of when and how you are going to be able to use them for your own needs with machine learning.

Clustering Algorithms

The first machine learning that we will look at is called the clustering algorithm. With the clustering algorithm, we are going to keep it pretty simple. This method can take our data and then classify it into clusters. Before the program even starts, you get the benefit of picking out how many clusters you would like all the information to fit into. For example, you may decide that you want to combine the data into five different clusters. The program would then go through and divide up all the information you have into five different clusters so that you could look through it.

The beautiful thing about this algorithm is that it is responsible for doing most of the work for you. This is because it is in charge of how many of your data points are going to fit into those clusters that you chose. To keep things organized, we are going to call all of the main clusters that you picked cluster centroids.

So, when you are looking at one of your clusters, and you notice that there are a lot of points inside of it, you can safely assume that all those particular data points have something in common or they are similar. There is some attribute or another that all the data points in one cluster have in common with each other. You can do this several times, creating more divisions as you go through the steps. You could potentially go through this enough time that the centroids will stop changing. This is when you know you are done with the process. There are several reasons why you would want to work with a clustering algorithm to help you get a program started when doing machine learning. First, doing your computations with the help of a clustering algorithm can be easy and cost efficient, especially compared to some of the supervised learning options. If you would like to do a classification problem, the clustering algorithms are efficient at getting it done. With that said, you do need to use some caution here though. This algorithm is not going to do the work of showing predictions for you. If you end up with centroids that are not categorized the right way, then you may end up with a project that is done the wrong way.

Markov Algorithm

The next type of unsupervised machine learning is the Markov algorithm. This is a nice one to use because it can take any or all of the data that you

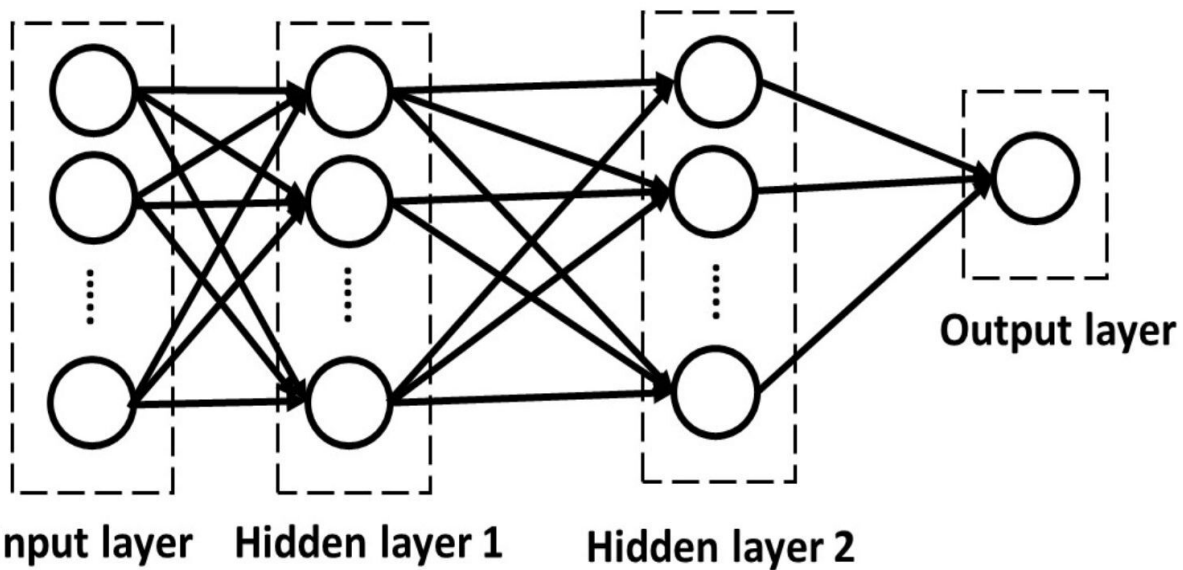
add to the system, and then translates this information so that it can work with another coding language. You will be able to set this up with any or all of the rules that you would like to be present ahead of time, based on how you would like this to work. There are times when doing this is going to be useful because it can take a string of data, and make it more useful when you learn the parameters for how your data is going to behave.

There are a lot of different ways that you can work with this Markov algorithm. One available option is if you decide to do some work with DNA. You would be able to take some of the DNA sequences that you have, and use this particular algorithm to take that information and translate it over to numerical values. When you are working with this on a computer, you will find that reading out numerical values is going to be so much easier compared to looking at a random strand of DNA, and hoping that you will be able to read through it.

Chapter 8 - Learning in Artificial Neural Networks

Definition Of Artificial Neural Network

A widely used approach in machine learning, the employment of artificial neural network is inspired by the brain system of humans. The objective of neural networks is replicating how the human brain learns. The neural network system is an ensemble of input and output layers and a hidden layer that transforms the input layer into useful information to the output layer. Usually, several hidden layers are implemented in an artificial neural network. The figure below presents an



example of a neural network system composed of 2 hidden layers:

Example of an artificial neural network

Before going further and explaining how neural networks work, let's first define what is a neuron. A neuron is simply a mathematical equation expressed as the sum of the weighted inputs. Let's consider $X=\{x_1, x_2, \dots, x_M\}$ a vector of N inputs, the neuron is a linear combination of all inputs defined as follows:

$$F(X=\{x_1, x_2, \dots, x_M\})=w_1x_1+w_2x_2+\dots+w_Mx_M;$$

With w_1, w_2, \dots, w_M is the weights assigned to each input. The function F can also be represented as:

$$F(X) = WX,$$

With W a weight matrix and X a vector of data. The second formulation is very convenient when programming a neural network model. The weights are determined during the training procedure. In fact, training an artificial neural network means finding the optimal weights W that provide the most accurate output.

To each neuron, an activation function is applied the resulted weighted sum of inputs X . The role of the activation function is deciding whether the neuron should be activated or not according to the model's prediction. This process is applied to each layer of the network. In the next sub-phases, we will discuss in details the role and types of activation functions as well as the different types of neural networks.

What Is An Activation Function And Its Role In Neural Network Models?

Activation functions are formulated as mathematical functions. These functions are a crucial component of an artificial neural network model. For each neuron, an activation function is associated. The activation function decides whether to activate the neuron or not. For instance, let's consider the output from a neuron, which is:

$$Y = \sum (\text{weight} * \text{input}) + \text{bias}.$$

The output Y can be of any value. The neuron does not have any information on the reasonable range of values that Y can take. For this purpose, the activation function is implemented in the neural network to check Y values and make a decision on whether the neural connections should consider this neuron activated or not.

There are different types of activation functions. The most instinctive function is the step function. This function sets a threshold and decides to activate or not activate a neuron if it exceeds a certain threshold. In other words, the output of this function is 1 if Y is greater than a threshold and 0 otherwise. Formally, the activation function is:

F='activated' or F=1; if $Y > \text{threshold}$

F='not-activated' or F=0; otherwise.

This activation function can be used for a classification problem where the output should be yes or no (i.e., 0 or 1). However, it has some drawbacks. For example, let's consider a set of several categories (i.e., class1, class2, ..., etc.) to which input may belong to. If this activation function is used and more than one neuron is activated, the output will be 1 for all neurons. In this case, it is hard to distinguish between the classes and decide into which class the input belong to because all neuron outputs are 1. In short, the step function does not support multiple output values and classification into several classes.

Linear activation function, unlike the step function, provides a range of activation values. It computes an output that is proportional to the input. Formally:

$F(X)=WX$, where X is the input.

This function supports several outputs rather than just a 1 or 0 values. This function, because it is linear, does not support backpropagation for model training.

Backpropagation is the process that relies on function derivative or gradient to update the parameters, in particular, the weights. The derivative (i.e., gradient) of the linear activation function is a constant which is equal to W and is not related to changes in the input X. Therefore, it does not provide information on which weights applied to the input can give accurate predictions.

Moreover, all layers can be reduced to one layer when using the linear function. The fact that all layers are using a linear function, the final layer is a linear function of the first layer. So, no matter how many layers are used in the neural network, they are equivalent to the first layer, and there is no point of using multiple layers. A neural network with multiple layers connected with a linear activation function is just a linear regression model that cannot support the complexity of input data.

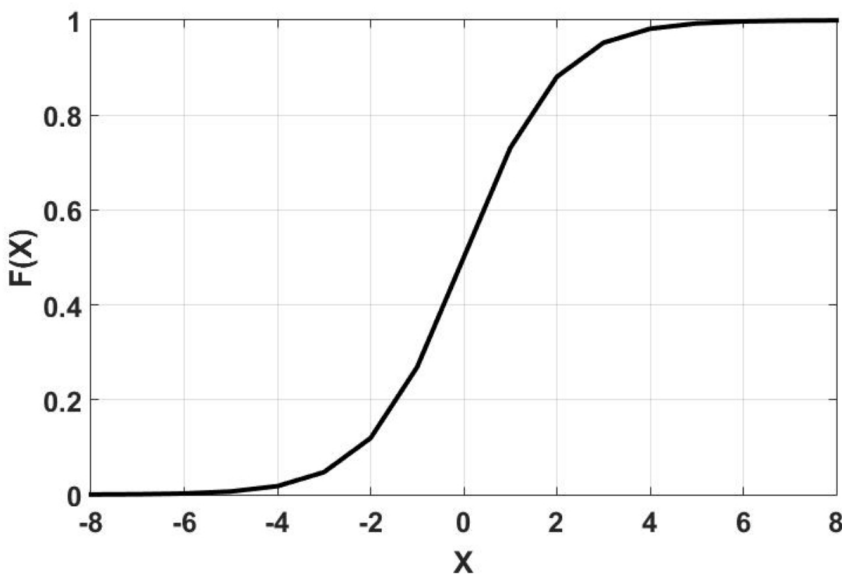
The majority of neural networks use non-linear activation functions because, in the majority of real-world applications, relations between the

output and the input features are non-linear. The non-linear functions allow the neural network to map complex patterns between the inputs and the outputs. They also allow the neural network to learn the complex process that governs complex data or high dimension data such as images, audios, among others. The non-linear functions allow overcoming the drawbacks of linear functions and step functions. They support backpropagation (i.e., the derivative is not a constant and depends on the changes of the input) and stacking several layers (i.e., the combination of non-linear functions is non-linear). Several non-linear functions exist and can be used within a neural network. In this phase, we are going to cover the most commonly used non-linear activation functions in machine learning applications.

The sigmoid function is one of the most used activation functions within an artificial neural network. Formally, a sigmoid function is equal to the inverse of the sum of 1 and the exponential of inputs:

$$F(X) = 1 / (1 + \exp(-X))$$

Outputs of a sigmoid function are bounded by 0 and 1. More precisely, the outputs take any value between 0 and 1 and provide clear predictions. In fact, when the X is greater than 2 or lower than -2, the value of Y is close to the edge of the curve (i.e., closer to 0 or 1).



Sigmoid activation function

The disadvantage of this activation function, as we can see from the figure above, is the small change in the output for input values under -4 and above 4. This problem is called 'vanishing gradient' which means that the gradient is very small on horizontal extremes of the curve. This makes a neural network using the sigmoid function, learning very slow when they approach the edges and computationally expensive.

The tanh function is another activation function used that is similar to the sigmoid function. The mathematical formulation of this function is:

$$F(X)=\tanh(X)=\frac{2}{1+\exp(-2X)}-1.$$

This function is a scaled sigmoid function. Therefore, it has the same characteristics as the sigmoid function. However, the outputs of this function range between -1 and 1, and the gradient are more pronounced than the gradient of the sigmoid function. Unlike the sigmoid function, the tanh function is zero-centered, which makes it very useful for inputs with negative, neutral, and positive values. The drawback of this function, as for the sigmoid function, is the vanishing gradient issue and computationally expensive.

The Rectified Linear Unit function or what is known as ReLU function, is also a widely used activation function, which is computationally efficient. This function is efficient and allows the neural network to converge quickly compared to the sigmoid and tanh function because it uses simple mathematical formulations. ReLU returns X as output if X is positive or 0 otherwise. Formally, this activation function is formulated as

$$F(X)=\max(0,X).$$

This activation function is not bounded and takes values from 0 to +inf. Although it has a similar shape as a linear function (i.e., this function is equal to identity for positive values), the ReLU function has a derivative. The drawback of the ReLU is that the derivative (i.e., the gradient) is 0 when the inputs are negative. This means as for the linear functions, the backpropagation cannot be processed, and the neural network cannot learn unless the inputs are greater than 0. This aspect of the ReLU, gradient equal to 0 when the inputs are negative, is called dying ReLU problem.

To prevent the dying ReLU problem, two ReLU variations can be used, namely the Leaky ReLU function and the Parametric ReLU function. The

Leaky ReLU function returns as output the maximum of X and X by 0.1 . In other words, the leaky ReLU is equal to the identity function when X is greater than 0 and is equal to the product of 0.1 and X when X is less than zero. This function is provided as follows:

$$F(X) = \max(0.1 * X, X)$$

This function has a small positive gradient which 0.1 when X has negative values, which make this function support backpropagation for negative values. However, it may not provide a consistent prediction for these negative values.

The parametric ReLU function is similar to the Leaky ReLU function, that takes the gradient as a parameter to the neural network to define the output when X is negative. The mathematical formulation of this function is as follows:

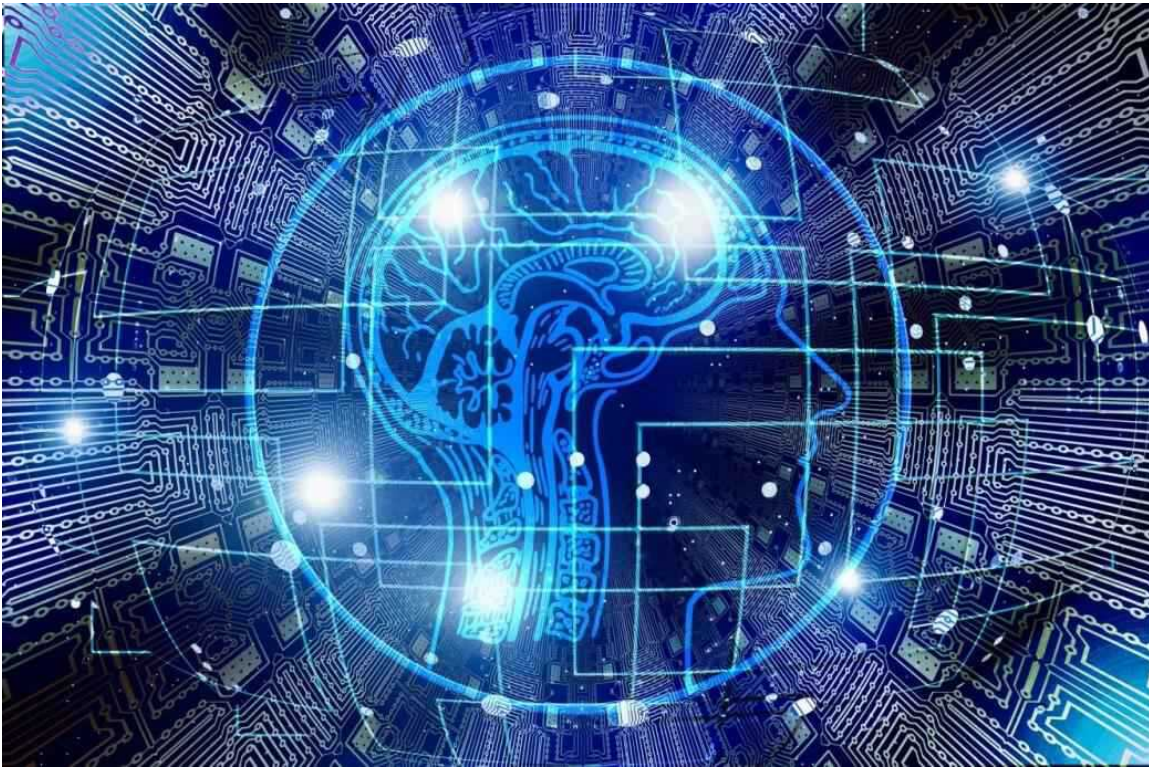
$$F(X) = \max(aX, X)$$

There are other variations of the ReLU function such as the exponential linear ReLU. This function, unlike the other variations of the ReLU the Leaky ReLU and parametric ReLU, has a log curve for negative values of X instead of the linear curves like the Leaky ReLU and the parametric ReLU functions. The downside of this function is it saturates for large negative values of X . Other variations exist which all relies on the same concept of defining a gradient greater than 0 when X has negative values.

The Softmax function is another type of activation function used differently. This function is usually applied only to the output layer when a classification of the inputs into several different classes is needed. In fact, the Softmax function supports several classes and provides the probability of input to belong to a specific class. It normalizes outputs of every category between 0 and 1 then divides by their sum to provide that probability. Given all these activation functions, where each one has its pros and cons, the question now which one should be used in a neural network? The answer is simply having a better understanding of the problem in hand will help guide into a specific activation function, especially if the characteristics of the function being approximated are

known beforehand. For instance, a sigmoid function is a good choice for a classification problem. In case the nature of the function being approximated is unknown, it is highly recommended to start with a ReLu function than try other activation function. Overall, ReLu function works well for a wide range of applications. It is an ongoing research, and you may try your own activation function.

Chapter 9 - The Perceptron



Defining Perceptron

A perceptron is a layer of the neural network that is capable of doing the following.

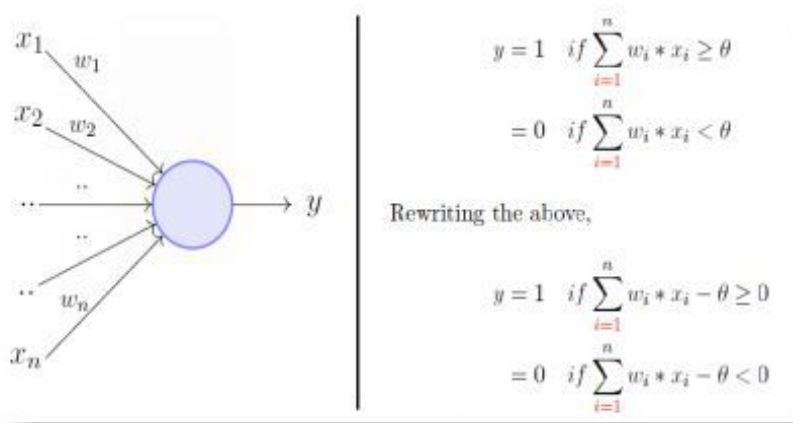
Configuring arithmetic values to the core of the data pool and estimating the sum of the positive return to the value of 1 for positive (+) outcomes.

In a different point of view, a perceptron can also be considered as a model for computation that draws a boundary (a line in two-dimensional cases) to isolate two classes in a given space.

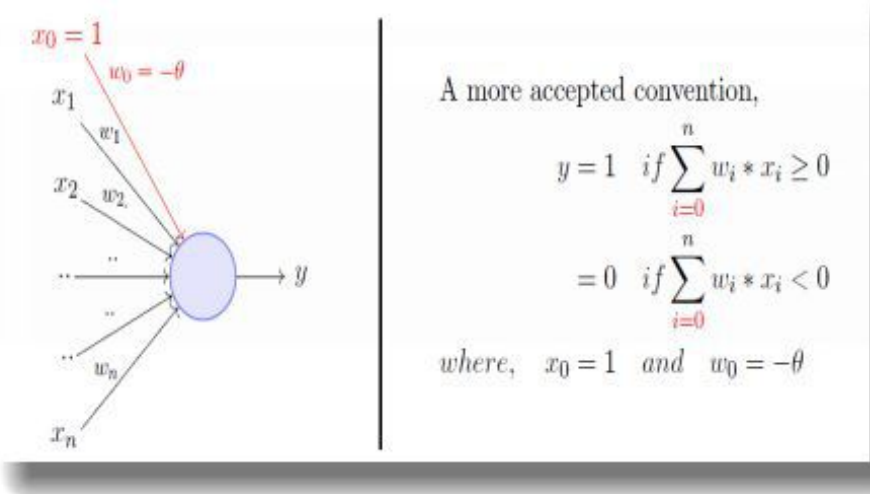
Initially, the weights and the bias which represents the classification hyperplane is usually unknown. In this case, unsystematic weights and bias are allocated to the model. The random assignment leads to misclassification of points.

The Perceptron Trick

The objective of the perceptron trick is decreasing the amount of misclassified points. The reduction can be made by sliding the line over the space. In other words, it is altering the expression of the hyperplane. For all the wrongly categorized points, we adjust the weights and prejudice to move the hyperplane closer to the points that are improperly classified. After some time, the algorithm will rectify the problem to classify the points correctly. However, in this phase, I choose to look at the perceptron differently. The approach taken is to identify perceptron in the context of the machine learning algorithm. Here, the perceptron is defined as an algorithm that predicts whether a given feature belongs to a specific class or not. From this angle, the algorithm is considered a binary classifier that is used for supervised machine learning. In most cases, the feature to be studied is always represented by a number with both magnitude and direction. As a classifier, the perceptron is used for linear classification and representation of data in graphical forms. The graphical representation model can take either two-dimension or the multi-dimension way. The perceptron identifies a feature, weighs its sum, and give an outcome of value that is 1 if the weighted summation is higher than the result from other function. The equation supports this statement in the image below

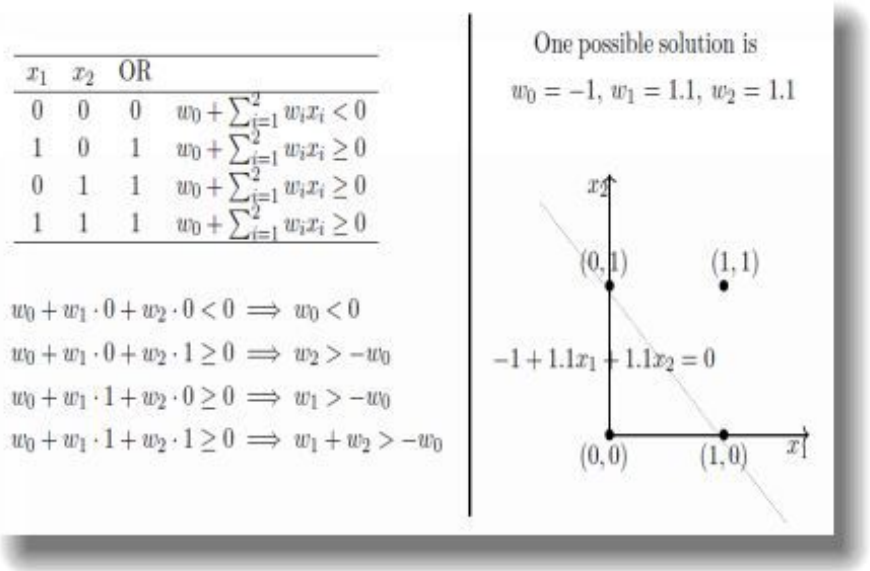


When the equation edge is substituted (second equation in the image above) and made to be constant with the weight of a given parameter, the equation in the image below is arrived at.



A single unit of a perceptron algorithm can be used to perform a divisible linear equation. The only group works by weighting or assigning a weight to both Boolean and positive metrics, with a corresponding partiality.

Using a Perceptron to Perform an OR Function



In the image above, conditions have been defined in line with the OR function. Such terms are meant to meet the optimum model, which requires the summation of weighted data to be zero (0) or more than it, for a given outcome of value one (1). The summations of the weights have been done a

straight line that separates the positive (+) values from negative (-) values established.

The Minsky proposal set out the standards of learning the features by the use of sample data. The proposed models have been described below.

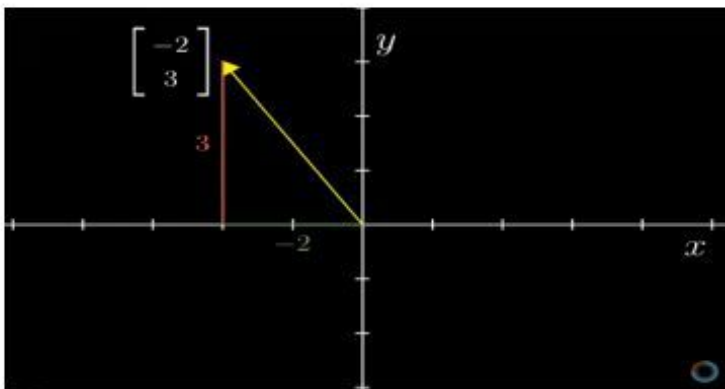
Basic Components of Linear Algebra

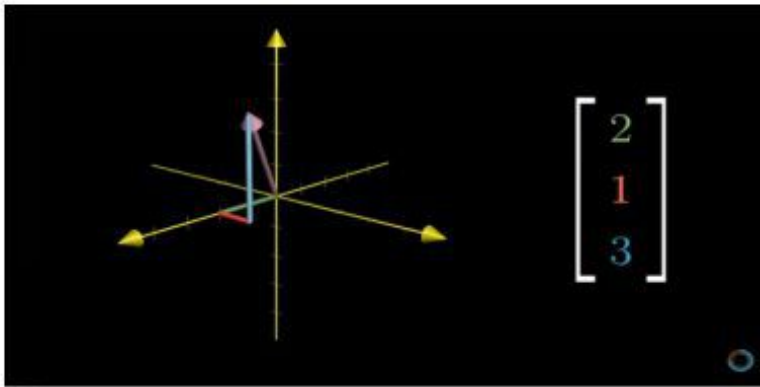
Vector

The vector value can represent a lot of things depending on the type of data and the user. Thus, there is more than one way of describing a vector. It may be a feature in space with both directing and magnitude, a structure or a database used to keep large amounts of data, etc. for this study; we will look at a vector as a line with an arrow being the head and the other point without an arrow as the origin. This is not the best definition. However, I want you just to grab the end. The images used in this presentation are mined from 3Blue1Brown.

Representing a Vector

A vector can be expressed in a two-dimension and a three-dimension plane, depending on the choice of the user and the function. Below is a two-dimensionally described vector (the first image) and a 3-D one (the second image).





Calculating the Dot Product of Two Vectors

Given two vectors $(n+1, w)$ and $(n+1, x)$, their dot product is calculated using the formula below. For substitution of value, the formula I written in a matrix form.

$$\begin{aligned} \mathbf{w} &= [w_0, w_1, w_2, \dots, w_n] \\ \mathbf{x} &= [1, x_1, x_2, \dots, x_n] \\ \mathbf{w} \cdot \mathbf{x} &= \mathbf{w}^T \mathbf{x} = \sum_{i=0}^n w_i * x_i \end{aligned}$$

From the equation, the vectors $/w/$ and $/x/$ are represented by a line with an arrow. The dimension of the vectors is the $(n+1)$ value. The outcome of the computation represents the dot product of vectors $/w/$ and $/x/$. This result shows the extent to which a single vector goes to the direction of the other vector. A perceptron creates a line that separates the positive value from the negative ones. This line is called the separating boundary. For the function in our example, this is represented as, $w \cdot x = 0$.

The Angle Between Two Vectors

The dot product mentioned above can be calculated using a different method. However, certain conditions must be met

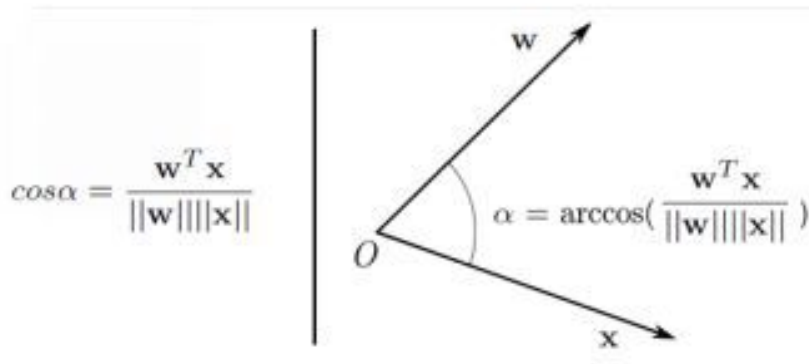
- I. The angle between the vectors must be known.
- II. The magnitude of the vectors must be known.

When conditions are met, the computation of the dot product is done by

$$\mathbf{w}^T \mathbf{x} = \|\mathbf{w}\| \|\mathbf{x}\| \cos \alpha$$

substituting in the formula in the image below.

The reverse method can be used to determine the angle between the two vectors. This can be done by first identifying the vector, getting its magnitude and the dot product.



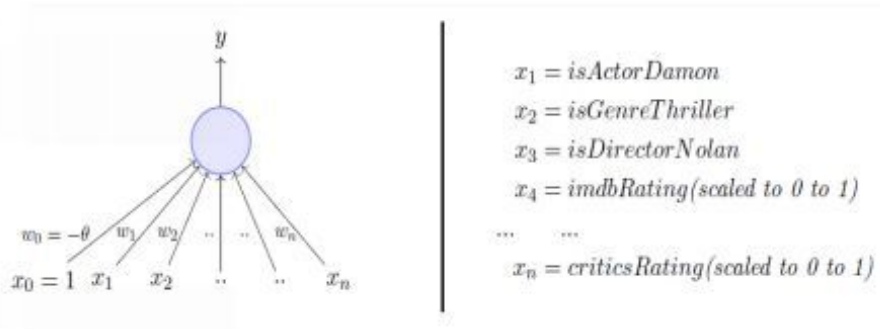
The diagram illustrates the relationship between the dot product and the angle between two vectors. On the left, the formula $\cos \alpha = \frac{\mathbf{w}^T \mathbf{x}}{\|\mathbf{w}\| \|\mathbf{x}\|}$ is shown. On the right, two vectors \mathbf{w} and \mathbf{x} originate from a common point O . The angle between them is labeled α . Below the angle, the formula $\alpha = \arccos\left(\frac{\mathbf{w}^T \mathbf{x}}{\|\mathbf{w}\| \|\mathbf{x}\|}\right)$ is provided.

The image below can guide in the substitution of values into the equation.

From the equation and a mathematical point of view, we know that when the cosine of an angle is 0, then the lines making the angle are perpendicular to each other. Taking from this, we can conclude that when

the dot product of vectors w and x is 0, then line w is perpendicular to line x . that is the relationship between the dot product and the angle between two vectors.

Problem Representation Using the Perceptron



The set example is going to be used to determine if a person will settle on a movie, data and inputs that are indicated on the diagram. The input data provided in this diagram have both real (+) and negative values (-). The actual values represent the movies watched by the person in question. We will use the data to see how the perceptron learning algorithm estimates weight in a dataset. For simplicity and clear visualization, we are going to use a 2-D representation.

Using the Perceptron Learning Algorithm

The main objective here is identifying the vector that can separate the real (+) from the negative (-) inputs, say vector (w), as used in the example.

We will use the set algorithm below.

Algorithm: Perceptron Learning Algorithm

```
P ← inputs with label 1;
N ← inputs with label 0;
Initialize w randomly;
while !convergence do
  Pick random  $\mathbf{x} \in P \cup N$  ;
  if  $\mathbf{x} \in P$  and  $\mathbf{w} \cdot \mathbf{x} < 0$  then
    |  $\mathbf{w} = \mathbf{w} + \mathbf{x}$  ;
  end
  if  $\mathbf{x} \in N$  and  $\mathbf{w} \cdot \mathbf{x} \geq 0$  then
    |  $\mathbf{w} = \mathbf{w} - \mathbf{x}$  ;
  end
end
//the algorithm converges when all the
inputs are classified correctly
```

The first step is adjusting the vector (w) with any other vector. After the initialization, we restate the examples provided in the dataset, in this instance, the metrics are P , U , and N . the examples represent both real and negative standards.

Proceeding the substitution phase, an input x (randomly picked vector) corresponds to the value P , the dot product to be determined in w , x should be stated as a value that is either equal to zero (0) or greater than zero. The actual value is not actually, outstanding since the perceptron will only give a yes or no answer.

In another instance, when vector x corresponds to the value N , the calculated dot product gives a

```

while !convergence do
  Pick random  $x \in P \cup N$  ;
  if  $x \in P$  and  $w \cdot x < 0$  then
     $w = w + x$  ;
  end
  if  $x \in N$  and  $w \cdot x \geq 0$  then
     $w = w - x$  ;
  end
end
end

```

zero (0) outcome.

The statements above can be represented in one IF situation in a while loop as;

The deduction from these representations can give two outcomes

The first outcome is when vector x corresponds to the P-value, and the dot product of the two vectors (w, x) is less than zero ($w \cdot x < 0$).

The second outcome is found when the vector x corresponds to the value of N, and the dot product of the two vectors (w, x) is more significant than zero ($w \cdot x > 0$).

Assessing the Viability of the Update Rule Above

The basic rule of perceptron has helped us to determine that vector x corresponds to the value P. Therefore,; the remaining part of the task is making the equation to be $w \cdot x > 0$. The equation can also be interpreted to mean that when vector x corresponds to P, then the angle between the two vectors (w, x) is supposed to be less than ninetydegrees. Why this inference? The cosine of the angle between the two vectors (w, x) is the same as the dot product of the two vectors. Thereby,

$$\cos\alpha = \frac{\mathbf{w}^T \mathbf{x}}{\|\mathbf{w}\| \|\mathbf{x}\|} \quad \Bigg| \quad \cos\alpha \propto \mathbf{w}^T \mathbf{x}$$

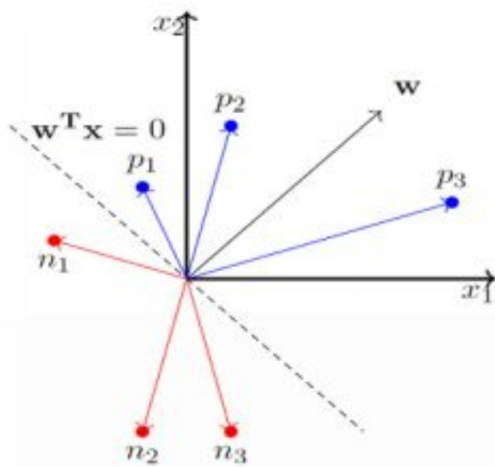
So if $\mathbf{w}^T \mathbf{x} > 0 \Rightarrow \cos\alpha > 0 \Rightarrow \alpha < 90$

Similarly, if $\mathbf{w}^T \mathbf{x} < 0 \Rightarrow \cos\alpha < 0 \Rightarrow \alpha > 90$

that

angle should be less than ninety degrees.

From the above interpretation, we get that we don't have to know the vector w given that it makes an angle that is less than ninety degrees with a given feature vector. In this case, the vector must be a positive value. On the other hand, vector w should make an angle that is greater than ninety degrees with a feature vector that has a negative value.



This represented, as shown

below.

At this point, we can comfortably say that the angle between vectors x and w should be a value less than ninety, for the given case that x corresponds to the value of P .

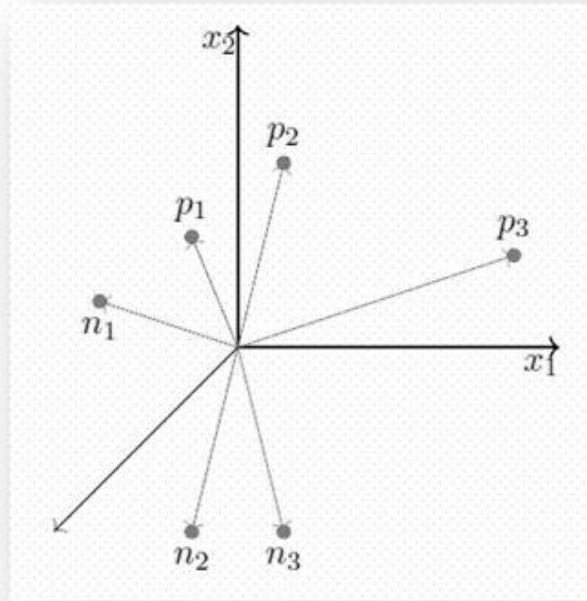
In the same way, we can conclude that the angle made between vectors w and x should be a value that is more than ninety degrees for a given condition that the vector x corresponds to the value of N .

Moving forward, we take a look at why the update worked. The diagram below presents the explanation as to why the upgrade works. The valuation may be a bit way-out, but the general

$ \begin{aligned} &(\alpha_{new}) \text{ when } \mathbf{w}_{new} = \mathbf{w} + \mathbf{x} \\ &\cos(\alpha_{new}) \propto \mathbf{w}_{new}^T \mathbf{x} \\ &\quad \propto (\mathbf{w} + \mathbf{x})^T \mathbf{x} \\ &\quad \propto \mathbf{w}^T \mathbf{x} + \mathbf{x}^T \mathbf{x} \\ &\quad \propto \cos\alpha + \mathbf{x}^T \mathbf{x} \\ &\cos(\alpha_{new}) > \cos\alpha \end{aligned} $	$ \begin{aligned} &(\alpha_{new}) \text{ when } \mathbf{w}_{new} = \mathbf{w} - \mathbf{x} \\ &\cos(\alpha_{new}) \propto \mathbf{w}_{new}^T \mathbf{x} \\ &\quad \propto (\mathbf{w} - \mathbf{x})^T \mathbf{x} \\ &\quad \propto \mathbf{w}^T \mathbf{x} - \mathbf{x}^T \mathbf{x} \\ &\quad \propto \cos\alpha - \mathbf{x}^T \mathbf{x} \\ &\cos(\alpha_{new}) < \cos\alpha \end{aligned} $
--	--

picture is seen.

For the case that vector x corresponds to P , we will have to add the values of vectors x and w . the addition raises the cosine of (alpha) function. This also means that the value of alpha is reduced (which is the angle between two vectors). The diagram below indicates how we can learn the vector w , which meets with a real object to make an angle that is less than ninety



degrees and an angle
than ninety degrees with negative-value objects.

greater

Proof of Convergence

There is no precise explanation to make people believe that the vector studied will converge in the end. This is due to the varying conditions within different sets of data. Even though it looks that there might be cases where the vector fails to converge, the truth is that the algorithm is proven to converge at some point. Many types of research have studied the patterns and deduced these proofs. Videos demonstrating this can be found in YouTube and other platforms.

Chapter 10 - Execution and Repetitive Tasks

This phase deals with if-else statement and different types of loop structures that help in repetitive tasks in programming.

If Structure

So far, the Python programs we've seen have been instruction-based. In a program, computer instructions are executed sequentially. Instructions cannot be skipped, nor repeated backward. The first programs were all like this. For example, to make a light come on ten times, repeat ten lines of instructions to make the light come on.

In order to make the program flexible, early programming languages added the function of "jump." With jump instructions, we can jump to any line in the program during execution and continue down. For example, to repeat execution, jump to a line that has already been executed. Programmers frequently jump forward and backward in their programs, for convenience. As a result, the program runs in a sequence that looks like a tangle of noodles, hard to read and prone to error.

Programmers have come to realize that the main function of a jump is to execute a program selectively or repeatedly. Computer experts have also argued that with the grammatical results of "selection" and "loop," "jump" is no longer necessary. Both structures change the flow of program execution and the order in which instructions are executed. Programming languages have entered a structured age. Compared with the "Spaghetti Program" brought about by the "jump," the structured program becomes pleasing to the eye. In modern programming languages, the "jump" syntax has been completely abolished.

Let's start with a simple example of a choice structure: if a house sells for more than \$200,000, the transaction rate 3% or 4%. We write a program using a selection structure.

Below is the program code:

```
Price = 340000
```

```
If
  Expected price > 200000:
    Fixed tax = 0.04
```

```
Else:
```

```
  Fixed tax = 0.03
```

```
  Print (fixed tax)
```

```
#prints 0.03
```

```
Output is:
```

```
0.04
```

In this process, there's an if that we haven't seen before. In fact, the function of this sentence is easy to understand. If the total price exceeds 200,000, then the expected tax is 4%; otherwise, the transaction rate is 3%. The keywords, if and else, each have a line of code attached to them, with a four-space indentation at the beginning of the dependent code. The program will eventually choose whether to execute the if dependent code or the else dependent code, depending on whether the condition after the if holds. In short, the if structure branches off from the program.

If and else can be followed by more than one line:

```
Price = 340000
```

```
If
```

```
  Total > 200000:
```

```
    Print ("rate is above 200,000")
```

```
    Fixed tax = 0.04
```

```
Else:
```

```
  Print ("rate is below 200,000")
```

```
  Fixed tax = 0.03
```

```
  Print (fixed tax)
```

```
# the result is 0.04
```

```
Output is:
```

```
Rate is above 200,000
```

As you can see, code that is also an if or else has four spaces indented. Keywords if and else are like two bosses, standing at the head of the line. There's a little brother standing back from the boss. The boss only by the terms of winning, standing behind his younger brother, has a chance to appear. The last line of the print statement also stands at the beginning of the line, indicating that it is on an equal footing with it and else. The program doesn't need conditional judgment; it always executes this sentence.

Else is not necessary; we can only write if program. For example:

```
price= 340000
```

```
if
```

```
price > 200000:
```

```
Print ("total price over $200,000")
```

Without else, it is effectively equivalent to an empty else. If the condition after it doesn't hold, then the computer doesn't have to do anything.

Stand Back

Python features indentation to indicate the dependencies of your code. As we show, the design for indenting code relationships is derived from the ABC language. For comparison, let's look at how C is written:

```
If (price > 0) {
```

```
Selling= 1;
```

```
Buying= 2;
```

```
}
```

The program means that if the variable price is greater than 0, we will do the two assignments included in the parentheses. In C, a curly brace is used to represent a block of code that is subordinate to the if. The average programmer also adds indentation to the C language to distinguish the dependencies of instructions. But indenting is not mandatory. The following does not indent the code, in the C language can also be normal execution, and above the results of the program run no difference:

In Python, the same program must be written in the following form:

```
If price> 0:
```

Selling= 1

Buying= 2

In Python, the parentheses around 0 are removed, the semicolon at the end of each statement is removed, and the curly brackets around the block are also removed. There's more—the colon (:) and the indentation of four spaces in front of 1 and 2. By indenting, Python recognizes that both statements are subordinate to the if. Indenting in Python is mandatory in order to distinguish between dependencies. The following procedure will have an entirely different effect:

```
if price > 0:
```

```
    selling = 1
```

```
    buying = 2
```

Here, only selling that is 1 is subordinate to if, and the second assignment is no longer subordinate to if. In any case, buying will be assigned to 2.

It should be said that most mainstream languages today, such as C, C++, Java, JavaScript, mark blocks with curly braces, and indentation is not mandatory. This syntax is derived from the popularity of the C language. On the other hand, while indenting is not mandatory, experienced programmers write programs in these languages with indenting in order to make them easier to read. Many editors also have the ability to indent programs automatically. Python's forced indentation may seem counterintuitive, but it's really just a matter of enforcing this convention at the syntactic level so that programs look better and are easier to read. This way of writing, with four spaces indented to indicate affiliation, is also seen in other Python syntax constructs.

If Nesting and Elif

And then back to the choice of structure. The choice of structure frees the program from the tedium of command-and-control permutations. A program can have a branching structure inside of it. Depending on the conditions, the same program can work in a volatile environment. With Elif Syntax and nested use of if, programs can branch in a more colorful way.'

The next program uses the elif structure. Depending on the condition, the program has three branches:

```
Result= 1
```

```
If result > 0:# Condition 1. Since I is 1, this part will perform.
```

```
Print ("positive result")
```

```
Result= result + 1
```

```
Elif result == 0:
```

```
# Condition 2. This part is not executed.
```

```
Print ("result is 0")
```

```
Result= result*10
```

```
Else:
```

```
# Condition 3. This part is not executed.
```

```
Print ("negative result")
```

```
Result= result - 1
```

There are three blocks, led by if, Elif, and else. Python first detects the condition of the if, skips the block that belongs to the if the condition is false, and executes the else block if the condition of the Elif is still false. The program executes only one of three branches, depending on the condition. Since the result has a value of 1, only the if part is executed in the end. In the same way, you can add more elif between it and else to branch your program.

We can also nest an if structure inside another if structure:

```
Result = 5
```

```
If result> 1: # this condition holds, execute the Internal Code
```

```
Print ("result bigger than 1")
```

```
Print ("nice")
```

```
If result > 2: # nested if structure, the condition holds.
```

```
Print ("result bigger than 2")
```

```
Print ("It's good than before")
```

After making the first if judgment, if the condition holds, the program runs in sequence and encounters the second if construct. The program will continue to judge and decide whether to execute based on the conditions. The second subsequent block is indented four more spaces relative to the if to become the “little brother.” Programs that further indent are subordinate to the inner if. In general, with the if construct, we branch the program. Depending on the conditions, the program will take a different path.

For Loop

Loops are used to iterate through blocks of code. In Python, loops are either for or while. Let’s start with the for loop. From the selection structure in phase 2.3, we have seen how to use indentation to represent the membership of a block. Loops use similar notation. Programs that are part of a loop and need to be repeated are indented, such as:

```
for input in [4,6,8,"love"]:
```

```
    print(c) #prints each element in the list in turn
```

The loop is to take an element from the list [4,6,8, “love”] one at a time, assign it to c, and execute the line belongs to the program for, which calls the print() function to print out the element. As you can see, one of the basic uses of for is to follow in with a sequence:

For Element in Sequence

The number of elements in a sequence determines the number of iterations. There are three elements in the example, so print() will be executed three times. That is, the number of repetitions of the for loop is fixed. The for loop, in turn, takes elements from the sequence and assigns them to the variable immediately after the for (a) in the example above. Therefore, even though the statements executed are the same, the effect of the same statement changes after three executions because the data also changes.

One of the conveniences of a for loop is to take an element from a sequence, assign it to a variable, and use it in a membership program. But sometimes, if we simply want to repeat a certain number of times and don’t want to create a sequence, we can use the range() function provided by Python:

```
for result in range(3):
```

```
print("This is crazy") # print "This is crazy" Three Times
```

The 3-way range() function in the program indicates the number of times you want to repeat. As a result, the program that belongs to for is executed five times. Here, after the for loop, there is still the variable I, which counts for each loop:

```
for result in range(7):
```

```
print(result, "This is crazy") # prints the sequence number and "This is crazy"
```

As you can see, the count provided by range() in Python also starts at 0, the same as the index of the table. We also saw a new use of print(), which is to specify multiple variables in parentheses, separated by commas. The function print() prints them all.

Let's look at a practical example of a for loop. We have used Tuples to record the yearly interest rate on a mortgage.

```
thisisinteresttuple= (0.06, 0.07, 0.08, 0.1, 0.3)
```

If there is a 200,000-dollar mortgage, and the principal is unchanged, then the annual interest to pay is how much? Using the for loop:

```
price= 200000
```

```
for interest in thisisinteresttuple:
```

```
debt= price* interest
```

```
Print ("you need to pay ", debt)
```

While Loop

There is also a loop structure in Python, the while loop. The use of the While is:

```
check= 0
```

```
while check< 20:
```

```
print(check)
```

```
check= check+ 1
```

```
# prints from 0 to 19
```

A while is followed by a condition. If the condition is true, the while loop continues to execute the statements that belong to it. The program will only

stop if the condition is false. In the while membership, we change the variable I that participates in conditional judgments until it becomes 10 so that the loop is terminated before the condition is met. This is a common practice for the while loop. Otherwise, if the while condition is always true, it becomes an infinite loop.

Chapter 11 - Variables and Strings

A software application consists of two fundamental parts: Logic and Data. Logic consists of the functionalities that are applied on data to accomplish a particular task. Application data can be stored in memory or hard disk. Files and databases are used to store data on hard disk. In memory, data is stored in the form of variables. Definition of Variable

Variable in programming is a memory location used to store some value. Whenever you store a value in a variable, that value is actually being stored at physical location in memory. Variables can be thought of as reference to physical memory location. The size of the memory reserved for a variable depends upon the type of value stored in the variable.

Creating a Variable

It is very easy to create a variable in Python. The assignment operator “=” is used for this purpose. The value to the left of the assignment operator is the variable identifier or name of the variable. The value to the right of the operator is the value assigned to the variable. Take a look at the following code snippet.

```
Name = 'Mike' # A string variable
Age = 15 # An integer variable
Score = 102.5 # A floating type variable
Pass = True # A Boolean variable
```

In the script above we created four different types of variables. You can see that we did not specify the type of variable with the variable name. For instance we did not write “string Name” or “int Age”. We only wrote the variable name. This is because Python is a loosely typed language. Depending upon the value being stored in a variable, Python assigns type to the variable at runtime. For instance when Python interpreter interprets the

line “Age = 15”, it checks the type of the value which is integer in this case. Hence, Python understands that Age is an integer type variable.

To check type of a variable, pass the variable name to “type” function as shown below: `type(Age)`

You will see that the above script, when run, prints “int” in the output which is the type of Age variables

Python allows multiple assignment which means that you can assign one value to multiple variables at the same time. Take a look at the following script:

```
Age = Number = Point = 20 #Multiple Assignment
print (Age)
print (Number)
print (Point)
```

In the script above, integer 20 is assigned to three variables: Age, Number and Point. If you print the value of these three variables, you will see 20 thrice in the output.

For any programming language, the basic part is to store the data in memory and process it. No matter what kind of operation we are going to perform, we must have the object of operation. It is difficult for a skillful woman to cook without rice. In Python language, constants and variables are the main ones. Both of them are identification codes used by program designers to access data contents in memory.

The biggest difference between the two is that the contents of variables will change with the execution of the program, while the contents of constants are fixed forever. In the process of program execution, it is often necessary to store or use some data. For example, if you want to write a program to calculate the mid-term exam results, you must first input the students' results, and then output the total score, average score and ranking after calculation. This phase describes how to store and access this data.

Variable Naming and Assignment

In a program, program statements or instructions tell the computer which Data to access and execute step by step according to the instructions in the program statements. These data may be words or numbers. What we call

variable is the most basic role in a programming language, that is, a named memory unit allocated by the compiler in programming to store changeable data contents. The computer will store it in "memory" and take it out for use when necessary. To facilitate identification, it must be given a name. We call such an object "variable." For example:

```
> > firstsample = 3
```

```
> > > second sample = 5
```

```
> > > result = firstsample + secondsample
```

In the above program statement, firstsample, secondsample, result are variables, and number 3 is the variable value of firstsample. Since the capacity of memory is limited, in order to avoid wasting memory space, each variable will allocate memory space of different sizes according to requirements, so "Data Type" is used to regulate it.

Variable Declaration And Assignment

Python is an object-oriented language, all data are regarded as objects, and the method of an Object reference is also used in variable processing. The type of variable is determined when the initial value is given, so there is no need to declare the data type in advance. The value of a variable is assigned with "=" and beginners easily confuse the function of the assignment operator (=) with the function of "equal" in mathematics. In programming languages, the "=" sign is mainly used for assignment.

The syntax for declaring a variable is as follows:

```
variable name = variable value
```

e.g. number = 10.

The above expression indicates that the value 10 is assigned to the variable number. In short, in Python language, the data type does not need to be declared in advance when using a variable, which is different from that in C language, which must be declared in advance before using a variable. Python interpretation and operation system will automatically determine the data type of the variable according to the value of the variable given or set. For example, the data type of the above variable number is an integer. If the content of the variable is a string, the data type of the variable is a string.

Variable Naming Rules

For an excellent programmer, readability of program code is very important. Although variable names can be defined by themselves as long as they conform to Python's regulations, when there are more and more variables, simply taking variables with letter names such as abc will confuse people and greatly reduce readability. Considering the readability of the program, it is best to name it according to the functions and meanings given by variables. For example, the variable that stores height is named "Height" and the variable that stores weight is named "Weight." Especially when the program scale is more significant, meaningful variable names will become more critical. For example, when declaring variables, in order to make the program readable, it is generally used to start with lowercase letters, such as score, salary, etc. In Python, variable names also need to conform to specific rules. If inappropriate names are used, errors may occur during program execution. Python is a case-sensitive language. In other words, number and Number are two different variables. Variable names are not limited in length. Variable names have the following limitations: the first character of a variable name must be an English letter, underlined "_" and cannot be a number. Subsequent characters can match other upper- and lower-case English letters, numbers, underlined "_," and no space character is allowed. You cannot use Python's built-in reserved words (or keywords). Although Python version 3. X supports foreign language variable names; it is recommended that you try not to use words to name variables. On the one hand, it is more troublesome to switch input methods when inputting program code. On the other hand, the reading of program code will not be smooth. The so-called reserved word usually has special meaning and function, so it will be reserved in advance and cannot be used as a variable name or any other identifier name.

The following is an example of a valid variable name: pageresponse

fileName4563

level

Number_

dstance

The following is an example of an invalid variable name:

2_sample

for

\$ levelone

The user name learning classroom uses the help () function to query Python reserved word. The help () function is Python's built-in function. If you are not sure about the method and property usage of a specific object, you can use the help () function to query.

The Python reserved words mentioned above can be viewed by using the help () function. As long as "help ()" is executed, the help interactive mode will be entered. In this mode, the instructions to be queried will be input, and the relevant instructions will be displayed.

We can continue to input the instructions we want to query in help mode. When we want to exit help interactive mode, we can input Q or quit. You can also take parameters when entering the help () command, such as help (" keywords "), Python will directly display help or description the information without entering help interactive mode.

Although Python uses dynamic data types, it is very strict in data processing, and its data type is "strong type." For example:

```
>>> firstsample = 5
>>> secondsample = "45"
>>> print (firstsample + secondsample) #
```

shows that TypeError variable firstsample is of numeric type and variable secondsample is of string type.

Some programming languages will convert the type unconsciously and automatically convert the value A to the string type, so firstsample + secondsample will get 545. Python language prohibits different data types from operating, so executing the above statement obviously Indicates information about the wrong type.

There is a difference between "strongly typed" and "weakly typed" in the data types of strong and weak type programming languages in small classrooms. One of the trade-offs is the safety of data type conversion. The strong type has a strict inspection for data type conversion.

Different types of operations must be explicitly converted, and programs will not automatically convert. For example, Python and Ruby prefer durable types.

However, most weak type programming languages adopt Implicit Conversion. If you don't pay attention to it, unexpected type conversion will occur, which will lead to wrong execution results.

Conclusion

Practice makes perfect. When learning a new language like Chinese or Spanish, experts recommend you use it every day in the form of speaking and going through an exercise or so. It's no different with a programming language. The more you practice, the more the basic syntax will become second nature to you, and you'll instinctively know when to use concepts like conditionals and loops. There are lots of resources (which you find in the appendix) that provide exercises and sample programs you can work on right away.

Write by hand

When you're taking notes (and you should take notes), write them out by hand. Studies show that the process of taking physical pen to physical paper facilitates the best long-term memory retention. Writing by hand includes writing code by hand, and then checking it on your computer, so you know for sure if it's accurate. Outlining code and ideas this way can help you stay organized, as well, before starting actually to build a program.

Find other beginners

Learning to code by yourself can get annoying and frustrating. One of the best ways to learn and improve is to find others who are in the same phase as you. You can bounce ideas off each other, help out on projects, and more. If you don't know anyone in your immediate circle, you can check out groups online and find local events through Meetups and Facebook. Always exercise caution and employ safe practices when first meeting people you only know online. Stick to public places during daylight hours, and don't go anywhere alone with someone you don't know well until you feel comfortable.

Try explaining Python out loud

Sometimes explaining something you just learned to someone is the best way to really cement it into your mind. It allows you to reframe concepts into your own words. You don't even have to talk to a real person; it can be an inanimate object. This is such a common practice among programmers that it's known as "rubber duck debugging," which references talking to a rubber duck about bugs in a program. Pick a topic in Python like conditionals or variables, and see if you can explain it. If you have trouble

or realize there's a gap in your knowledge, it's time to go back and keep learning.

Check out other languages

This is about Python, so obviously we believe that should be your priority, but getting to know a little bit about other languages can be very helpful, too. It will definitely make you a better programmer in the future. Checking out other languages can help you discover what typical architecture is built into every language as well as the differences between them and Python. Even if you just read about other languages and never write much code in anything besides Python, you'll be improving your knowledge and skill.

Have a plan for when you get stuck

When you get stuck while coding, take a step back. What are you trying to get the code to do? What have you tried? And what's happening? Write the answers down and be as specific and detailed as possible. Then, you can go to someone else for help, and you won't have to spend a ton of time trying to explain the problem. The answers are also really useful just for your own thought process. Take a close look at any error messages you're getting. Work your way backward to try and spot any mistakes.

Another response to getting stuck is just to start over. If your code is really long, it can be discouraging to start from scratch, but that means you don't have to go through the whole thing, picking it apart and wearing out your eyes. Starting over may actually be easier.

Take a break

Whether you choose to begin again or go through the code with a fine-toothed comb, you should take breaks. When you work on a problem for too long, your brain gets stuck in a groove, and it's difficult to come up with new solutions. Do something that doesn't use the exact same muscles as coding. Exercise your body instead, take a long shower, lie down for a nap, or bake some cookies. Einstein would often come up with solutions to his problems while he played the violin, and who doesn't want to think a little bit like Einstein?

PYTHON DATA SCIENCE

**THE ULTIMATE STEP BY STEP
BEGINNER'S GUIDE WITH TOOLS
AND PRINCIPLES FOR CODING AND
DATA ANALYSIS PLUS
HANDS-ON PROJECTS TO HAVE A
SUCCESSFUL PRACTICAL
PROGRAMMING FROM SCRATCH**

Adrienne Hawkes Matthes and John Tacke

Introduction

The emergence of data science is recent, and these new uses of data have often had difficulty finding suitable tools. Indeed, the data scientist must be a good developer while remaining a good data analyst. He has had to opt for a tool that would combine this demand with more and more strong development and automation (all the more so with the arrival of artificial intelligence and connected objects), with the need for a toolbox suitable for data applications.

Many avenues have been explored, in particular with the software R which continues to be a reference in data science but that could seem too oriented towards statistics for more developmentally oriented data scientists. Many other tools for setting up data science processes have been developed (most proprietors such as Matlab or SAS), but it turns out that Python (which combines powerful language and extremely simple) has managed to draw its pin.

The first real advance was the creation of NumPy (Numerical Python) package, which is still today the cornerstone of the Python ecosystem for data science. On the other hand, setting up data-driven Python environments with Anaconda has also enabled a whole community to come together around Python and data. Finally, IPython and its notebooks (now Jupyter) have completed the ecosystem to provide data scientists with a very simple language but one which is extremely comprehensive for data science. This global environment resulted in the development of many packages and APIs, which today make Python the best tool to automate data science treatments.

The Current Evolution

In recent years, Python has taken an extremely important place in the world of data processing. While in the early 2010s, it seemed clear that in the world of open-source data processing tools, the software R was going to carve out the lion's share, a significant change has taken place since a few years. The emergence of Python as a language related to data science, machine learning, deep learning, and artificial intelligence is extremely fast.

Grace, an extremely active community under the PyData banner and frequent and numerous events (PyCon, PyData, JupyterCon, SciPyCon ...), language development took an unexpected turn. While we could hear in

2015 developers say that from the point of view of machine learning the development of Python was modeled on that of R with a few months late. Today, it is R who begins to model his developments in the field of machine learning, deep learning, and big data, on packages developed in Python. In 2018, KDnuggets, an influential blog in the world of data science, even surveyed thousands of data scientists around the world who, for the first time, and showed more users of Python than of R.

The Python adventure in data science is therefore recent but only because it is a language that adapts perfectly to the approach led by a data scientist, which would be: "better in programming than a statistician and better in statistics than a developer. "

The near future of Python is above all the abandonment of version 2 and the general version 3. The other big current development concerns the use of interactive interfaces with Python as a communication language with more and more advanced APIs. We'll talk a little bit further about Jupyter's widgets that allow you to develop interactively and build interfaces in a few lines.

Python is increasingly used as a language to communicate with other environments. So Python can communicate with Apache Spark through PySpark medium, or with deep learning ecosystems such as TensorFlow. Calculations are no longer done in the Python engine but much more in engine devices using distributed infrastructures or GPU-based computations (Graphical Process Units). This trend is only beginning with the massification of data and requests for real-time treatments are ever more common.

Python cannot answer these challenges alone but, combined with other tools, it allows the data scientist to manage the entire ecosystem of data processing, be it big, small, and smart.

Chapter 1 - What Is Data Science And Why Should You Learn It Using Python

This guidebook has taken some time to look through a lot of the different parts that come with data analysis. We took a look at what data analysis is all about, how to work with the Python language and why it is such a good thing for the data analysis, and even some of the basics of Machine Learning and why this should be a part of our process.

by all of this in mind, it is now time for us to move on to some of the other things that we can do when working on this process. We are going to explore some of the best algorithms and models that we can use to complete our data analysis with the help of the Python language. There are so many different algorithms that we can choose from, and all of them are going to be great options to get the work done. With this in mind, let's dive right in and see what some of the best algorithms and models are for completing your business data analysis with Python.

Neural Networks

It is hard to have a discussion about Machine Learning and data analysis without taking some time to talk about neural networks and how these forms of coding are meant to work. Neural networks are a great addition to any Machine Learning model because they can work similarly to the human brain. When they get the answer right, they can learn from that, and some of the synapses that bring it all together will get stronger. The more times that this algorithm can get an answer right, the faster and more efficient it can become with its job as well.

With neural networks, each of the layers that you go through will spend a bit of time at that location, seeing if there is any pattern. This is often done with images or videos so it will go through each layer of that image and see whether or not it can find a new pattern. If the network does find one of these patterns, then it is going to instigate the process that it needs to move over to the following layer. This is a process that continues, with the neural network going through many layers until the algorithm has created a good idea of what the image is and can give an accurate prediction.

There are then going to be a few different parts that can show up when we reach this point, and it depends on how the program is set up to work. If the algorithm was able to go through the process above and could sort through all of the different layers, then it is going to make a prediction. If the prediction it provides is right, the neurons in the system will turn out stronger than ever. This is because the program is going to work with artificial intelligence to make the stronger connections and associations that we need to keep this process going. The more times that our neural network can come back with the correct answer, the more efficient this neural network will become in the future when we use it.

If the program has been set up properly, it is going to make the right prediction that there is a car in the picture. The program can come up with this prediction based on some of the features that it already knows belongs to the car, including the color, the number on the license plate, the placement of the doors, the headlights, and more.

When you are working with some of the available conventional coding methods, this process can be really difficult to do. You will find that the neural network system can make this a really easy system to work with.

For the algorithm to work, you would need to provide the system with an image of the car. The neural network would then be able to look over the picture. It would start with the first layer, which would be the outside edges of the car. Then it would go through some other layers that help the neural network understand if any unique characteristics are present in the picture that outlines that it is a car. If the program is good at doing the job, it is going to get better at finding some of the smallest details of the car, including things like its windows and even wheel patterns.

There could potentially be a lot of different layers that come with this one, but the more layers and details that the neural network can find, the more accurately it will be able to predict what kind of car is in front of it. If your neural network is accurate in identifying the car model, it is going to learn from this lesson. It will remember some of these patterns and characteristics that showed up in the car model and will store them for use later. The next time that they encounter the same kind of car model, they will be able to make a prediction pretty quickly.

When working with this algorithm, you are often going to choose one and use it, when you want to go through a large number of pictures and find some of the defining features that are inside of them. For example, there is often a big use for this kind of thing when you are working with face recognition software. All of the information wouldn't be available ahead of time with this method. And you can teach the computer how to recognize the right faces using this method instead. It is also one that is highly effective when you want it to recognize different animals, define the car models, and more.

As you can imagine, there are several advantages that we can see when we work with this kind of algorithm. One of these is that we can work with this method, and we won't have to worry as much about the statistics that come with it. Even if you need to work with the algorithm and you don't know the statistics or don't have them available, the neural network can be a great option to work with to ensure that any complex relationship will show up.

Naïve Bayes

We can also work with an algorithm that is known as the Naïve Bayes algorithm. This is a great algorithm to use any time that you have people who want to see some more of the information that you are working on, and who would like to get more involved in the process, but they are uncertain about how to do this, and not to understand the full extent of what you are doing. It is also helpful if they want to see these results before the algorithm is all the way done.

As you work through some of the other algorithms on this page and see what options are available for handling the data, you will notice that they often take on hundreds of thousands of points of data. This is why it takes some time to train and test the data, and it can be frustrating for those on the outside to find out they need to wait before they can learn anything about the process. Showing information to the people who make the decisions and the key shareholders can be a challenge when you are just getting started with the whole process.

This is where the Naïve Bayes algorithm comes in. It is able to simplify some of the work that you are doing. It will usually not be the final algorithm that you use, but it can often give a good idea to others outside of the process about what you are doing. It can answer questions, puts the

work that you are doing in a much easier to understand the form, and can make sure that everyone will be on the same page.

Clustering Algorithms

One of the best types of algorithms that you can work with is going to be the clustering algorithm. There are a variety of clustering algorithms out there to focus on, but they are going to help us ensure that the program can learn something on its own, and will be able to handle separating the different data points that we have. These clustering algorithms work best when you can keep things simple. It takes some of the data that you are working with and then makes some clusters that come together. Before we start with the program though, we can choose the number of clusters that we want to fit the information too.

The number of clusters that you go with is going to depend on what kind of information you are working with as well. If you just want to separate your customers by gender, then you can work with just two clusters. If you would like to separate the customers by their age or some other feature, then you may need some more clusters to get this done. You can choose the number of clusters that you would like to work with.

The nice thing that comes with the clustering algorithms is that they will handle most of the work of separating and understanding the data for you. This is because the algorithm is in charge of how many points of data go into each of the clusters you choose whether there are two clusters, or twenty, that you want to work with. When you take a look at one of these clusters, you will notice that with all of the points inside it is safe to assume that these data points are similar or share something important. Therefore they fell into the same cluster with one another.

Once we can form some of these original clusters, it is possible to take each of the individual ones and divide them up to get some more sets of clusters because this can sometimes provide us with more insights. We can do this a few times, which helps to create more division as we go through the steps. In fact, it is possible to go through these iterations enough times that the centroids will no longer change. This is a sign that it is time to be done with the process.

Support Vector Machines

Another option that we need to work with is known as the support vector machine or SVM. When we work with this one, it is important to take all of the items in our data set, and then work on plotting them into one n-dimensional space, rather than having them all over the place. N is going to be the number of features that should show up in this algorithm along with the rest of our information. We then have the option to take the value of all these features and translate them over to the value that is in your coordinates. From here, we determine where the hyperplane is because this will show us the differences that are there between our various classes.

You may notice while working on this kind of algorithm that more than one support vector is going to show up. Many of these are easy to ignore because they are just the coordinates of individual observations that are seen. You can then use the SVM as a frontier that can separate them into classes. The two support vectors that we need to focus on will be the hyperplane and the line.

Chapter 2 – Statistics and Probability

As you start to work with the process of machine learning, it is essential to know that there is going to be a friendly relationship that ends up showing between this process, and what is called the probability theory. Machine learning is a pretty broad field to work with, and this means that it doesn't work on its own, but also with some other fields at the same time. The fields you will be able to work with often depend on the kind of project.

One thing that you are going to notice when you start with machine learning is that it can merge with statistics and probability. It is so crucial for a lot of the projects that you choose to start on to learn how these three different areas are going to work together.

Now, there are a few different methods that you can utilize with the probability and statistics, and all of them are important to the learning process that you need to see happen here. The first thing to consider is picking out the right algorithm. And as you go through this guidebook, you will find that there are a lot of different algorithms that you can use, including supervised, unsupervised, and also reinforced learning algorithms. However, not all of the algorithms are going to work with every project that you have.

When you pick out one of the algorithms to work with (and we will talk about quite a few of these in this guidebook), there are a few things you need to balance out together, including the number of parameters that you need, the complexity, the training time that you can work with, and the accuracy. As we spend more time with machine learning, you will find that each project you need to focus on will ask for a specific combination of these factors, so consider that ahead of time.

When you decide to work with the ideas of statistics and the probability theory, you will be better prepared to pick out the parameters that are right for your specific program, the strategies for validation, and you can then use all of these to pick out the algorithm for this project. This is also going to be helpful to use when you want to figure out the amount of uncertainty that is present in that algorithm, and then you can determine if there is a level of trust that you should have for any predictions.

As you can imagine here, both of these two topics are going to be very useful when working on any project with machine learning, and they will do wonders when you want to understand what is going on with any project. This phase will look at the different topics that come with both statistics and the probability theory, and how you can use them on any project.

What Are The Random Variables?

Now, the first topic in statistics is random variables. With probability theory, these random variables are going to be expressed with the “X” symbol, and it is the variable that has all its possible variables come out as numerical outcomes that will come up during one of your random experiments. With random variables, there are going to be either continuous or discrete options. This means that sometimes, your random variables will be functions that will map outcomes to the real value inside their space. We will look at a few examples of this one to help it make sense later on.

We are going to start with an example of a random variable by throwing a die. The random variable that we are going to look at is going to be represented by X, and it will rely on the outcome that you will get once the die is thrown. The choices of X that would come naturally here is going to go through to map out the outcome denoted as 1 to the value of i.

What this means is that if X equals 1, you will map the event of throwing a one on your die to being the value of i. You would be able to map this out with any number that is on the die, and it is even possible to take it to the next step and pick out some mappings that are a bit strange. For example, you could map out Y to make it the outcome of 0. This can be a hard process to do, and we aren't going to spend much time on it, but it can help you to see how it works. When we are ready to write out this one, we would have the probability, which is shown as P of outcome 1 of random variable X. it would look like the following:

$PX(i)$ or $(x=i)$

Distribution

Now that we have looked a bit at the random variables, it is time to look a bit at the idea of a probability distribution, and how it works with machine

learning. What is meant here is that we need to take a look at the outcomes and figure out the probability that they are going to happen, or a random variable to happen. To make this even easier, we are going to use this distribution to figure out how likely it is that we are going to get a specific number.

Let's say that you are working with a die. There are six sides on it, and you have a random probability of one of the numbers showing up each time that you throw it. We can use the distribution to figure out how likely it is that with a particular throw, we will get a five or a two or one of the other numbers.

To help us get started with this one, it helps to have an example. We will need to let the X , which is our random variable, but the outcome that we will see on the die when we throw it. We are also going to start this experiment using the assumption that the die is perfectly capable of being used, with no tricks, and it isn't loaded. This ensures that the sides all end up with the same probability of showing up each time that you do a throw. The probability distribution you will work with here to figure out how probable it is that one number will show up includes

$$P(X=1) = P(X=2) = \dots = P(X=6) = 1/6$$

In this example, it matches up to what we did with the random variables. It does have a different meaning. Your probability distribution is more about the spectrum of events that can happen, while our random variable example is all about which variables are there. With the probability theory, the $P(X)$ part is going to note that we are working with our probability distribution of the random variable X .

While you take a look at this example, you may notice that your distribution could have just one variable, or there could be two or more of these variables that show up at the same time. When this occurs, you will name it a joint distribution. To figure out this probability, you will need to figure out the variables on their own and combine them to see the results.

In able to see how this is going to work when it comes to two or more variables, let's have X be the random variable and the one that will be defined by the outcome you can get any time you throw the die. And then you can use Y to show us the random variable that will tell you what results occur if you decide to flip a coin. For this one, to make things easier, we are

going to assign the heads side of the coin 1, and the tails side is going to be 0. This is used to help us figure out the probability distribution for each variable on their own and together.

We are going to denote this joint distribution as $P(X, Y)$ and the probability of X as having an outcome of a and Y as having an outcome of b as either $P(X = a, Y = b)$ or $P(X, Y(a, b))$.

Conditional Distribution

The next thing we need to bring into the mix with machine learning and statistics is the idea of the conditional distribution. When we already know what the random variable distribution is all about, possibly because we already know the value of the second random variable, then we can base the probability of one event on the outcome we can get with that second event. So, you will find that when you use this distribution, you will have the random variable be known as X when $X = a$ given that the variable of Y is going to be $Y = b$. When these are true, the following formula is going to help you to define and figure out what the variable is for both of the situations:

$$P(X = a | Y = b) = P(X = a, Y = b) / P(Y = b).$$

As you work through machine learning, there are going to be a few times when you may need to use conditional distributions. These can be useful tools depending on the system that you are designing, especially if you need to have the program reason with uncertainty.

Independence

And finally, when working with statistics and probability during machine learning, we need to consider independence. One of the variables that you can work with here is to figure out how much independence is inside the problem. When you work with these random variables, you are going to find out that they are going to be independent of what the other random variables are, as long as the distribution that you have doesn't change if you take a new variable and try to add it into that equation.

To make this one work a bit better, you are going to need to work with a few assumptions in concerns to the data you are using with machine learning. This makes it a bit easier when you already know about

independence. An excellent example to help us understand what this is all about is a training sample that uses j and I , and are independent of any underlying space when the label of sample I is unaffected by the features sample j . No matter what one of the variables turns out, the other one is not going to see any change or be affected if they are independent.

Think back to the example of the die and the coin flip. It doesn't matter what number shows up on the die. The coin is going to have its result. And the same can be said the other way around as well. The X random variable is always going to be independent of the Y variable. It doesn't matter the value of Y , but the following code needs to be true for it:

$$P(X) = P(X|Y).$$

In the case above, the values that come up for X and Y variables are dropped because, at this point, the values of these variables are not going to matter that much. But with the statement above, it is true for any value that you provide to your X or Y , so it isn't going to matter what values are placed in this equation.

This phase went over a few of the things that you can do with the help of probability theory and statistics when you are working on machine learning. You can experiment with some of these to get the hang of what you can do using them, and then we can learn a few more algorithms that you can use later on.

Chapter 3 - Work With Numpy

These are the following example demonstrates how to work with numpy arrays:

```
import numpy as np
x = np.array([10, 20, 30]) # Create an array of rank 1
print(type(x))           # Returns the object type
print(x.shape)           # Returns the size
print(x[0], x[1], x[2]) # Accessing array elements using their indexes
x[0] = 40                 # Change array element at index 0
print(x)                  # Returns the array values
y = np.array([[10, 20, 30], [40, 50, 60]]) # Create an array of rank 2
print(y.shape)            # Returns "(2, 3)"
print(y[0, 0], y[0, 1], y[1, 0]) # Returns "10 20 40"
```

The code should return the following after execution:

```
<class 'numpy.ndarray'>
(3,)
10 20 30
[40 20 30]
(2, 3)
10 20 40
```

We created an array named `x` with 3 values. The `type()` method returns the class of the object, which is a numpy `ndarray`. The `shape ()` method returns the number of elements in the array, which is 3. We have used array indexes to access the elements of the array. The first element is stored at index 0. This means `x [0]` returns the first element of the array. We have also used an index to change the value of the first element of the array.

For the case of the second array, the `shape ()` method returned (2, 3). The reason is that we have an array of rank 2, each array having 3 elements.

NumPy provides us with several other functions that we can use to create arrays. These are demonstrated below:

```
import numpy as np
x = np.zeros((2,2)) # To create an array of zeros
print(x)
y = np.ones((1,2)) # To create an array of ones
print(y)
z = np.full((2,2), 7) # To create a constant array
print(z)
a = np.eye(2) # To create an identity matrix, 2x2
print(a)
b = np.random.random((2,2)) # Create an array of random values
print(b)
```

The code will print the following:

```
[[ 0.  0.]
 [ 0.  0.]]
[[ 1.  1.]]
[[7 7]
 [7 7]]
[[ 1.  0.]
 [ 0.  1.]]
[[ 0.88325492  0.54391249]
 [ 0.83040157  0.50402765]]
```

Ndarray Object

The Ndarray (N-dimensional) array is a very important object in NumPy. It is simply a collection of items that belong to the same type. To access the elements stored in the collection, we can use a zero-based index.

All elements in the ndarray are assigned an equal amount of memory. Every element in the array belongs to the data-type object (dtype). An item obtained from the ndarray through a slicing operation is represented through a Python object that belongs to one of the array scalar types.

We can create an instance of the ndarray class using a number of array creation routines. We create a basic ndarray using an array function as shown below:

```
numpy.array
```

The function creates a ndarray from any object that exposes an array interface or from any method returning an array as shown below:

```
numpy.array(object, dtype = None, copy = True, order = None, subok = False, ndmin = 0)
```

This is a description of the parameters used in the above constructor:

- object- An object that exposes the array interface method should return an array, or a sequence (nested).
- dtype- this is the required data type of the array. It is an optional parameter.
- copy- The default setting is that the object will be copied. It is an optional parameter.
- order- This can be C (for row major) or F (for column major) or A (for Any). A is the default value.
- subok- by default, the returned array is made a base class array. If set to true, the sub-classes are passed through.
- ndmin- for setting the minimum dimensions of the resultant array.

Here is an example:

```
import numpy as np
myarray = np.array([10,20,30])
print(myarray)
```

We have just created an array of 3 elements. The code gives the following output:

```
[10 20 30]
```

We can also use the same mechanism to create an array with more than one dimension. For example:

```
import numpy as np
myarray = np.array([[10, 20], [30, 40]])
print(myarray)
```

The code will return the following:

```
[[10 20]
 [30 40]]
```

Here is how we create an array with minimum dimensions:

```
import numpy as np
myarray = np.array([10, 20, 30, 40, 50], ndmin = 2)
print(myarray)
```

The output from the code should be as follows:

```
[[10 20 30 40 50]]
```

Loading Numpy Arrays from Text

With numpy, we can create an array by loading the data from an existing sample. In fact, in data analysis, most data is loaded from text files. The loading of the data can be done using `loadtxt ()` or `genfromtxt ()` methods.

Let us demonstrate this using an example:

```
x, y = np.loadtxt('data.txt',
                 skiprows=1,
                 unpack=True)
```

In the above code, we have used the `loadtxt ()` method to load the dataset into the environment. The function takes the name of the file with the data as the argument. We also have other arguments, with the first one skipping the first row in the file since it is the header marking the columns. The `unpack = True` argument returns the columns as separate arrays. What this means is that the values in the first column will be returned to `x` while the second column will be returned to `y`.

You may have data that is separated by commas. You can still use the `loadtxt()` method to load such data but add arguments like `delimiter` and

dtype to the method. Here is another example demonstrating how to load data:

```
array2 = np.genfromtxt('myfile.txt',  
                      skip_header=1,  
                      filling_values=-999)
```

In the above example, we have used the `genfromtxt ()` method to load the data. However, we have provided a way of dealing with missing values. This has been done using the `filling_values` argument. We have specified that all values marked as `MISISNG` will be replaced with `-999`, which will become their value.

Saving NumPy Arrays

After working with your arrays, it is possible for you to save them in a file. To save an array into a text file, you should use the `savetxt()` function. This is demonstrated below:

```
import numpy as np  
array1 = np.arange(0.1, 5.2, 7.4)  
np.savetxt('dat.out', array1, delimiter=',')
```

The `np.arange()` function used above will create an array with evenly spaced values.

NumPy Broadcasting

With broadcasting, NumPy can work with arrays of different shapes to perform arithmetic operations. A good example is when you have a relatively smaller and another relatively larger array. Your aim is to use the smaller array to perform operations on the larger array. Examples of such operations include multiplication, sum, etc. This can be accomplished by use of the broadcasting mechanism.

However, a number of rules apply to the use of broadcasting. The good thing with this is that these rules are simple.

First, for you to be able to do broadcasting, the dimensions of the arrays must be compatible. For the dimensions to be compatible, they must be equal. Here is an example:

```
import numpy as np
```

```

# Initialize `a`
a = np.ones((2,3))
# Check the shape of `a`
print(a.shape)
# Initialize `b`
b = np.random.random((2,3))
# Check the shape of `b`
print(b.shape)
# Add `a` and `b`
print(a + b)

```

The code should return something close to the following:

```

(2, 3)
(2, 3)
[[ 1.25839265  1.07494823  1.71289399]
 [ 1.80906205  1.91956748  1.13260726]]

```

Two dimensions can also be compatible if one of them is 1. Here is an example:

```

# Import `numpy`
import numpy as np
# Initialize `a`
a = np.ones((3, 4))
# Check the shape of `a`
print(a.shape)
# Initialize `b`
b = np.arange(4)
# Check the shape of `b`
print(b.shape)
# Subtract `a` and `b`
print(a - b)

```

The code returns something close to this:

```
(3, 4)
(4,)
[[ 1.  0. -1. -2.]
 [ 1.  0. -1. -2.]
 [ 1.  0. -1. -2.]]
```

In case the dimensions are found not compatible, you will get a `ValueError`.

You can only broadcast arrays together when they are compatible in all dimensions. Here is an example:

```
# Import `numpy`
import numpy as np
# Initialize `a` and `b`
a = np.ones((3, 4))
b = np.random.random((5, 1, 4))
# Add `a` and `b`
print (a + b)
```

The code prints the following:

```
[[[ 1.78102557  1.74977271  1.49652607  1.09286643]
 [ 1.78102557  1.74977271  1.49652607  1.09286643]
 [ 1.78102557  1.74977271  1.49652607  1.09286643]]

 [[ 1.9372402  1.28990963  1.84075993  1.67310304]
 [ 1.9372402  1.28990963  1.84075993  1.67310304]
 [ 1.9372402  1.28990963  1.84075993  1.67310304]]

 [[ 1.5057764  1.43491567  1.72935755  1.78274838]
 [ 1.5057764  1.43491567  1.72935755  1.78274838]
 [ 1.5057764  1.43491567  1.72935755  1.78274838]]

 [[ 1.92863798  1.15736223  1.55748743  1.802319  ]
 [ 1.92863798  1.15736223  1.55748743  1.802319  ]
 [ 1.92863798  1.15736223  1.55748743  1.802319  ]]

 [[ 1.27827926  1.41860554  1.99675455  1.23437043]
 [ 1.27827926  1.41860554  1.99675455  1.23437043]
 [ 1.27827926  1.41860554  1.99675455  1.23437043]]]
```

The a and b seem to be different in terms of dimensions, but it was possible for us to add them together. The reason is that the two are compatible in all dimensions. This makes the two arrays good for broadcasting. In short, if you need to do broadcasting, you have to pay attention to both the shape and dimensions of your arrays.

Chapter 4 – Work With Pandas

Pandas is built on NumPy and they are meant to be used together. This makes it extremely easy to extract arrays from the data frames. Once these arrays are extracted, they can be turned into data frames themselves. Let's take a look at an example:

```
In: import pandas as pd
import numpy as np
marketing_filename = 'regression-datasets-marketing.csv'
marketing = pd.read_csv(marketing_filename, header=None)
```

In this phase, we are uploading data to a data frame. Next, we're going to use the "values" method to extract an array that is of the same type as those contained inside the data frame.

```
In: marketing_array = marketing.values
marketing_array.dtype
Out: dtype('float64')
```

We can see that we have a float type array. You can anticipate the type of the array by first using the "dtype" method. This will establish which types are being used by the specified data frame object. Do this before extracting the array. This is how this operation would look:

```
In: marketing.dtypes
Out: 0float64
1int64
```

```
2float64
3int64
4float64
5float64
6float64
7float64
8int64
9int64
10int64
11float64
12float64
13float64
dtype: object
```

Matrix Operations

This includes matrix calculations, such as matrix to matrix multiplication. Let's create a two-dimensional array.

This is a two dimensional array of numbers from 0 to 24. Next, we will declare a vector of coefficients and a column that will stack the vector and its reverse. Here's what it would look like:

```
In: coefs = np.array([1., 0.5, 0.5, 0.5, 0.5])
coefs_matrix = np.column_stack((coefs,coefs[::-1]))
print (coefs_matrix)
```

Out:

```
[[1. 0.5]
 [0.5 0.5]
 [0.5 0.5]
 [0.5 0.5]
 [0.5 1.]]
```

Now we can perform the multiplication. Here's an example of multiplying the array with the vector:

```
In: np.dot(M,coefs)
```

```
Out: array([5.,20.,35.,50.,65.])
```

Here's an example of multiplication between the array and the coefficient vectors:

```
In: np.dot(M,coefs_matrix)
```

```
Out: array([[5.,7.],  
[20.,22.],  
[35.,37.],  
[50.,52.],  
[65.,67.]])
```

In both of these multiplication operations, we used the "np.dot" function in order to achieve them. Next up, let's discuss slicing and indexing.

Slicing and Indexing

Indexing is great for viewing the nd-array by sending an instruction to visualize the slice of columns and rows or the index.

Let's start by creating a 10x10 array. It will initially be two-dimensional.

```
In: import numpy as np
```

```
M = np.arange(100, dtype=int).reshape(10,10)
```

Next let's extract the rows from 2 to 8, but only the ones that are evenly numbered.

```
In: M[2:9:2,:]
```

```
Out: array([[20, 21, 22, 23, 24, 25, 26, 27, 28, 29],  
[40, 41, 42, 43, 44, 45, 46, 47, 48, 49],  
[60, 61, 62, 63, 64, 65, 66, 67, 68, 69],  
[80, 81, 82, 83, 84, 85, 86, 87, 88, 89]])
```

Now let's extract the column, but only the ones from index 5.

```
In: M[2:9:2,5:]
```

```
Out: array([[25, 26, 27, 28, 29],  
[45, 46, 47, 48, 49],  
[65, 66, 67, 68, 69],  
[85, 86, 87, 88, 89]])
```

We successfully sliced the rows and the columns. But what happens if we try a negative index? Doing so would reverse the array. Here's how our array would look when using a negative index.

```
In: M[2:9:2,5::-1]
```

```
Out: array([[25, 24, 23, 22, 21, 20],  
[45, 44, 43, 42, 41, 40],  
[65, 64, 63, 62, 61, 60],  
[85, 84, 83, 82, 81, 80]])
```

keep in mind that this process is However, only a way of viewing the data. If you want to use these views further by creating new data, you cannot make any modifications to the original arrays. If you do, it can lead to some negative side effects. In that case, you want to use the “copy” method. This will create a copy of the array which you can modify however you wish. Here's the code line for the copy method:

```
In: N = M[2:9:2,5:].copy()
```

Chapter 5 - Learning About Functions

So far, we have learned quite a lot of things. If you have already started to lose track of all the knowledge, you shouldn't be alarmed. It is only natural for everyone to find themselves in such a situation when they are in the learning process. No one is perfect and that is what makes us all human beings, right?

We have seen dictionaries and learned they are nothing like the ones we use to learn new words and meanings. We have learned about a rather funny thing called tuples and understood that they are essentially a list with parentheses and do not allow anyone to add, remove, or modify values. We have gone initially through some functions too, but now it is time for us to start looking into functions a little more closely.

Understanding the Concept of Function

Take a moment or two here and engage your mind a little. Think about it and try to come up with some vague idea of what functions truly are.

Functions are either user-defined or pre-defined. In either case, their job is to organize codes within a recallable function name. There are tons of pre-defined functions available within Python. We have already been using some of these again and again.

We already have a decent idea about functions that are built-in and pre-defined. These include and are not limited to `input()`, `print()`, and so many more. However, let us now look at how to create our function.

Let us begin by a traditional approach and write a block of code that welcomes the user with a friendly greeting. We will store this as a function named "welcome_message" so that we can call on this function later on.

```
def welcome_message():
    print("Hello and welcome")
    print("Hope you have a great time")
print("Begin")
welcome_message()
print("End")
```

Let us begin learning and see what is happening in the block of code above. Firstly, for us to create our function, we need to define it first. The 'def' is a keyword that Python will look at and immediately understand that you are about to 'define' a new function. Next, we will need to name the function. While you can always name the function as you please, it is highly recommended and encouraged that you use names that are easy to understand and have a descriptive name. If we were to name this function anything other than `welcome_message`, we may know what it is as we wrote it, but for any other programmer out there, they may not understand.

Whenever you create a function, you need to use parentheses. You do not have to pass any information through it so leave them as they are. Now, we need to add the colon mark.

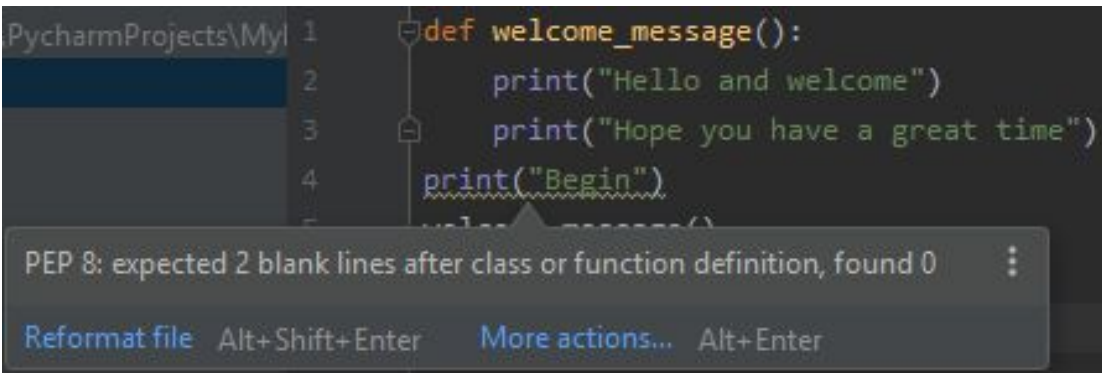
What happens when you use a colon at the end of a statement? Your cursor gets indented in the following line. That means your cursor will be slightly far from the actual starting point. This is to denote to the programmer that he/she is about to type something that would hold value for a command or a statement above it. In this case, we are trying to define the function.

Let us then use two print commands and place our greeting messages. That is it! You have now created your very first function. You can now recall it as many times as you like. However, should you try to call on this function a line or two before the 'def' command, Python will have no idea what you're talking about. Why? That has everything to do with the fact that Python reads a program line by line. By the time it arrives on the line where you called a function, it would check with the lines and not find anything relatable as the actual 'def' step was carried out in a step following this.

After this, let us now use our function and see how it works. Remember, the function holds two printable messages for our users. For our reference, we will now create a 'begin' and an 'end' message. This would allow us and the programmer to know where the regular messages are and where the function lies. Use your function with empty parentheses between the two print commands as shown above. If you like, you can remove these print commands and just type in your function number to see the results.

A quick tip for all! If you come across the annoying wiggly lines, simply hover your mouse over it and you will find out what the expected or

suggested solution is. In this case, if you remove the two-line spaces, you should see a suggestion saying this:



```
PycharmProjects\My 1 def welcome_message():
2     print("Hello and welcome")
3     print("Hope you have a great time")
4     print("Begin")
5     welcome_message()

PEP 8: expected 2 blank lines after class or function definition, found 0
Reformat file Alt+Shift+Enter More actions... Alt+Enter
```

Whenever you define a function, you will always be required to leave at least two blank lines before proceeding on with the codes.

Now, let us run the program and you should see all the messages and our function in action. Python initiated the sequence and first read the definition. This is where Python only understood for itself what the function was. The actual program was executed when Python reached line six where our print("Begin") message started. In the next line, we placed our function and this is where Python recalled what it had just learned. It quickly carried out the set of codes we defined within the function and executed the same. Lastly, it executed the last line before finishing the program.

This is how functions are created and used. Now, we can use this function as many times as we like within the same file. Note that you cannot use this newly created function if you were to open a new file or work on an older file where you did not define this function.

When things start to get tougher for you in your programming future, remember to create your functions and use them where applicable. They will save you quite a lot of time and help you in places as well. These are used when certain actions or operations need to be carried out every now and then.

Using Various Functions

Python was created with simplicity in mind. It was also created to minimize the work and maximize the output. If you use the codes and the functions wisely, you will surely be making the most out of this programming language. It is also noticeable that most of the things you learn about

Python and its functions, parameters, methods, and such will help you learn other languages quicker, so do pay attention.

Parameters

Our eagle-eyed readers may have noticed something about the function we just created a few minutes ago. Unlike most of the functions, we did not pass any information through the parentheses at all. Why that happens is something we will come to know about once we understand exactly what parameters are in Python.

Parameters are used as place-holders for receiving information. These are what we, as well as users, provide to the program in order for it to work more accurately. There are some cases and functions where arguments are not required for them to do their basic operation. However, if you provide an argument to these functions, they will provide you a more specific output. Of course, it does depend on the availability of the said parameter. You cannot force a function to do something it is not designed to do.

Now, let us look at our function. It is certainly missing something. If you currently print the `welcome_user` function, it would say everything but will not contain the name of the user at all. Surely, it would look a lot nicer for us if we could somehow use this function to use the name of the user and add it to the greeting.

Luckily, we can do just that! For that, we first need to add the 'name' parameter in the first line where we began defining our function. Simply type name between the parentheses and you will see the text turn grey. This confirms that the word has been added as a parameter. Now, we wish to print the name of this user along with the greetings we have defined within the function. For this example, let us assume that the user is named Fred.

```
def welcome_message(name):  
    print("Begin")  
    print("Hello and welcome {name}!")  
    print("Hope you have a great time")  
    print("End")  
welcome_message('Fred')
```

Begin

Hello and welcome Fred!

Hope you have a great time

End

Finally! We have a name to add to our greetings. You can add another line of code by using our function and passing a different name now. See what happens then.

When we set a parameter for a function and then call upon the function without providing it an argument or the bit of information that goes between the parentheses, it will provide us with an error, except for a few.

Now, let us make our function a little more dynamic and add another parameter. Let us add a parameter that allows the program to print out the last name of the user. Now, our code should look something like this:

```
def welcome_message(name, last_name):  
    print("Hello and welcome {name} {last_name}!")  
    print("Hope you have a great time")  
print("Begin")  
welcome_message('Fred', 'William')  
print("End")
```

The point to learn here, apart from being able to add parameters, is the fact that 'Fred' and 'William' are being used in a specific order. Should you type it the other way around, Python will print these as they are. This is because of their position concerning the defined parameters. The first value Python reads here, it will automatically link it with the first parameter. This can cause a little confusion, especially if the last name becomes the first name.

These arguments are called as positional arguments. To further show their importance, let us remove the last name from the argument above.

```
print("Begin")  
welcome_message('Fred')  
print("End")
```

Traceback (most recent call last):

Begin

File "C:/Users/Smith/PycharmProjects/MyFirstGo/PosArg.py", line 7, in
<module>

```
welcome_message('Fred')
```

TypeError: welcome_message() missing 1 required positional argument:
'last_name'

So, the system does not allow us to continue as we have removed an element. This time, type in the last name first followed by the first name and see if it makes any difference. When you run the program now, you should be able to see this:

```
print("Begin")
```

```
welcome_message('Fred', 'William')
```

```
print("End")
```

Begin

Hello and welcome William Fred!

Hope you have a great time

End

Now, the sequence kind of worked. The only issue is that it has gotten the name wrong. Now, the last name is being portrayed and printed as the first name. That is rather embarrassing, isn't it?

The above errors either state that we are missing one required positional argument or show that we placed the wrong name in the wrong place. Positional arguments are such arguments whose position matters a lot. If you miss out on the position altogether, you will end up with an error. If you type in something else, as we did in our last example, you will produce incorrect results. To correct it, simply provide the last name after the first name.

Chapter 6 - Developing a Machine Learning Model with Python

The benefits of machine learning are the models that make predictions and the predictions themselves. To succeed in this field, developers need to understand how to deliver accurate, reliable, and consistent predictions, which requires a systematic approach to building an ML model. Essentially, it depends on the mathematics applied in harnessing and quantifying uncertainty.

The main steps involved in developing a powerful machine-learning model are:

1. Define the problem to solve or objectives to meet
2. Adjusting one's mindset
3. Choosing a systematic process
4. Picking a tool, such as Python
5. Choosing the right datasets to work on and practicing the process
6. Building an ML portfolio

Python is one of the fastest-growing and most popular ML platforms. When it comes to developing an ML model with Python, the first step is to define one's problem or objectives. Using the fanciest and most powerful algorithms is meaningless if one is focusing on the wrong problem. Therefore, it is important to think deeply about one's problem or objectives before starting. This is one of the most important steps when it comes to developing a good ML model.

When defining a new problem, one needs to use a simple framework that will help one understand the motivation and elements of the problem. This involves answering a few important questions, such as:

1. What is the main problem or objective?
2. Why do I need to solve this problem?

3. How should I go about solving it?

Describing the problem as though one were describing it to a friend is a good way to begin. Doing this will help one identify areas that one needs to focus on or address. In addition, it will provide the foundation for a specific sentence description one will use to analyze the problem. For example, "I need a model that will identify which emails need a response."

Having described the problem, one needs to think about why one needs to solve it, which means considering the motivation for solving it, the benefits of solving it, and the possible solutions, including the type of lifetime one expects from the chosen solution.

The final step is to determine how one would solve the problem or achieve one's objectives manually. This involves identifying the data one needs to gather, how to prepare it, and how to design a model to load the data and solve the problem. This step may include experiments and prototypes one will need to use.

They are helpful because they can help identify uncertainties and knowledge about a potentially useful domain to explore. They can also help identify problems that simply need a manually implemented solution, in addition to uncovering hidden but valuable knowledge.

Installing the Python and SciPy Platforms

People typically install python packages from either PyPI or Conda package repositories. Every version of Python comes with its own set of packages; however, it is usually good practice to isolate them within a Python environment using the reticulate package so that updating a package for one project does not affect other projects. Reticulate functions, on the other hand, create Python environments and create packages within them.

Many professionals recommend using the Anaconda Python distribution to install Python. For computational modeling and scientific computing, developers need additional packages that are not available in the standard Python library. These packages will allow them to use specialized numerical methods and create plots. These packages are numpy, pandas, scipy, and matplotlib, and are the building blocks for computational work with Python.

Anaconda is one of the most popular Python distributions, providing the Python interpreter and a list of packages and applications. Generally, installing Python is easy; however, installing additional packages can be a bit difficult. The Anaconda Python distribution is available for free download for Linux operation systems, OS X, and Windows. For OS X and Windows, users can download either the next based installer or the graphical installer. For beginners, the graphical installer is a better choice.

Once they have installed the Python distribution of their choice, users should download a testing program, such as Spyder, and execute it. If it produces the desired results, chances are that the installation of Python and other packages, including SciPy, was successful.

SciPy, or scientific Python, provide useful functionalities. It is an open-source Python-based library used in scientific computing, mathematics, technical computing, and engineering. It is available for download from the web. Once downloaded, users simply need to unpack, compile, and install it using a custom installation path.

It contains many different sub-packages, including image manipulation; file input and output, signal processing, special function, numerical integration, linear algebra operation, statistics, and random numbers, and others.

Loading the Dataset

Before a developer can begin an ML project, he/she must load his/her dataset. The most common format for ML data is CSV files. There are several considerations when it comes to loading an ML dataset from these types of files. The first consideration is the file header. If the data lacks a file header, one may need to name each file attribute manually. However, if it has a file header, it will help in naming each column of data. In both cases, one needs to specify whether one's data has a file header before loading.

It is also important to consider whether the CSV file contains comments, usually indicated using a hash sign at the start of a line. If it has comments, depending on the method used to load the dataset, one may have to indicate the characters to expect to identify a comment line. Other considerations when it comes to loading CSV files are quotes and delimiters.

Each process for loading ML datasets is standalone, which means users have the ability to copy and paste it into their projects. Python's standard library, for example, provides the function `reader` and the module `CSV` to load CSV files. Once loaded, users can convert their datasets to a NumPy array and employ it for machine learning.

Another way to load datasets is by using the NumPy and `numpy.loadtxt` function, which assumes that all data has the same format, and there is no header row. Developers can also load their CSV files using Pandas, which is an easy and flexible process that returns a data-frame that they can immediately begin plotting and summarizing.

ML algorithms learn from data; therefore, it is important to provide them with the right data for the problem one wants to solve. It is also important to ensure the data is in a useful format and scale, in addition to having meaningful features. The more disciplined users are in their handling of ML datasets, the more consistent and accurate predictions they are likely to achieve. The process of preparing data for an ML algorithm involves three main steps, which are:

1. Data selection
2. Data preprocessing
3. Data transformation

The first step involves choosing the subset of all available data; however, the data selected should address the problem one wants to solve. The second step is about considering how one will use the data and getting it into a workable format. This involves three common preprocessing steps, which are formatting, cleaning, and sampling the data.

The third step is to transform the preprocessed data. Here, knowledge of the problem at hand and the specific algorithm one is working with will have an influence on this step. The main processes involved in this step are scaling, decomposition of attributes, and attribute aggregation.

Data preparation is one of the most important processes in an ML project. It involves a lot of analysis, exploration, and iterations; however, learning how to do it properly will help one succeed in this field.

Summarizing the Dataset

People today have access to tons of information due to the rise of the internet. In fact, this information bombards people from different sources, such as social media, news media, emails, and many other sources. People wish there were someone to summarize the most important information for them. Actually, machine learning is getting there.

Using the latest technological advances in the field of ML, developers can now group large datasets by different variables and use summary functions on each group. Python's rich Pandas library, for example, is an excellent analysis tool designed to help developers summarize their datasets.

To demonstrate the simplicity and effectiveness of such an analysis tool, assume one has a dataset containing mobile phone usage records with 1,000 entries from the phone log spanning 6 months. One can simply load a CSV file containing this data using Panda's Data Frame function, which will create several columns in the file, including date, duration, item, and month, network, and network type.

Having loaded this data into Python, the calculation of different statistics for columns will be very simple. These calculations might include standard deviations, minimum, maximum, mean, and more. For example:

1. How many rows are in the dataset
2. `Data['item'].count()`
3. `Out[40]:1000`

Unless one has very specific requirements, the need to create and use custom functions is minimal or not necessary. The standard Pandas package comes with a wide range of quickly calculable basic statistics, including sum, count, median, mean, min, max, prod, abs, mode, skew, var, std, cumsum, sem, quantile, and more. Developers can quickly apply these functions to gain summary data for each group, which is a very helpful function.

In addition, they can include or exclude different variables from each summary requirement or combine more than one variable to perform more detailed and complex queries.

Chapter 7 - Identifying the Nearest Neighbors

Before we dive in, let's take time to understand how to verify whether the classification model is working to our satisfaction. At one point in the phase, we learn the confusion matrix.

A confusion matrix refers to the arrangement of the actual versus the predicted class tables. Say, for example, that you have a two-class problem, that is, your y can take either value, T, or F. Assume you have trained a classifier to predict your y . You have predicted the value of y from

		Predicted	
		T	F
Actual	T	TP	FN
	F	FP	TN

your model. Then you can complete your confusion matrix using like this:

In the above table, it is important to list the results from the test set. Keep in mind that you know the class labels in the test set; thus, you can compare the classification model output with the actual class label.

- Below TP, which stands for True Positive, it has a count of all those records within the test set whose labels feature T, and where the model is predicted T.
- Below the FN, which stands for False Negative, you have a count of all the records whose actual label is T, but the algorithm is predicted as N.
- FP stands for False Positive, where the actual label is F, but the algorithm is predicted as T.
- TN stands for True Negative, where the algorithm predicted both the label and the original class label as F.

With the above knowledge about the confusion matrix, you can extract performance metrics using the measure of the quality of our classification model.

Accuracy is defined as the ration of a correct prediction to the total number of predictions. From the confusion matrix, you know that the sum of TP and TN totals the number of correct predictions.

Accuracy from the training set is normally very optimistic. It is important for someone to consider the test set's accuracy value to define the true performance of the model.

With the above knowledge, the first classification problem that you will consider is the k-nearest neighbor, shortened as KNN. Before jumping to details of KNN, let's consider a simple classification algorithm known as the rote classifier algorithm. The rote classifier memorizes the whole training data; it loads all the data within the memory. You need to conduct classification on an unseen new training instance. It will try to march the new training instance using any of the training instances within the memory. It will match each attribute within the training instance. In case it discovers a match, it predicts the class label of the test insurance as the class label of the matched training instance.

So far, you need to know that this classifier will fail in case the test instance isn't similar to any of the training instances loaded into the memory.

KNN resembles the rote classifier, except that instead of searching for an exact match, it applies a similarity measure. Resembling the rote classifier, KNN loads all the training sets into the memory. When it requires to classify a test instance, it determines the distance between the test instance and all the training instances. Using the above distance, it selects K's closest instances within the training set. Next, the prediction for the test set is defined based on the number of classes of the K nearest neighbors.

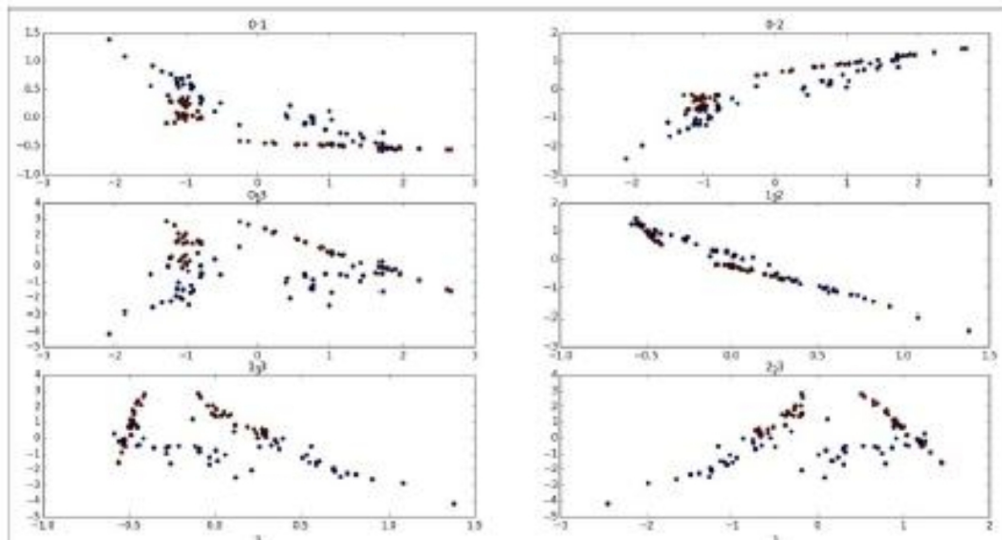
For example, if you have a two-class classification problem and you select the K value as three, and if the test record's three nearest neighbors have classes, 1,1, and 0, it will categorize the test instance as 1, which the popular.

KNN is part of the family algorithms known as instance-based learning. Besides that, the decision to categorize a test instance is considered last; it is also known as the lazy learner.

For this algorithm, you will generate some data by making use of the scikit's `make_classification` method. This will produce a matrix of four columns and 100 instances:

The `get_data` function internally requests `make_classification` to produce test data for any classification task.

It's always a great practice to visualize the data before beginning to feed it into any algorithm. The



`plot_data`

function generates a scatter plot between all the variables.

In the above diagram, you have plotted all the variable combinations. The top two charts demonstrate combinations between the 0th and 1st column, followed by 0th and 2nd. The points are also colored using their class labels. This presents a hint of how much information is available for variable combinations to perform a classification task.

How to Achieve It

You will need to divide your dataset preparation and model training into two different methods: `get_train_test` to find the train and test data, and build your model. Finally, you will apply the `test_model` to verify the usefulness of the model.

How it Operates

You must start by calling `get_data` and plotting it using `plot_data`.

It is also important to separate a section of the training data for the testing that is required to determine the model. Next, invoke the `get_train_test` method to achieve the same.

Within the `get_train_test`, you need to train the test split size, which is the standard 80/20. Next, apply 80 percent of the data to train your model. Now, combine both `x` and `y` to a single matrix before the split by using NumPy's `column_stack` method.

Armed with the following train and test sets, you should be ready to create a classifier. You must trigger the build model using the training set, class labels `y`, and attributes `x`. This function also accepts `K`, the number of neighbors, as a parameter, with a default value of two.

We apply the scikit-learn's KNN execution, `KNeighborClassifier`. Then build an object of the classifier and call its `fit` method to create the model.

You should be ready to test how good the module is by leveraging on the training data. You can pass the training data (`x` and `y`) and model to the `test_model` function.

You know the actual class labels (`y`). Now invoke the `predict` function using `x` to get the predicted labels. Next, print some of the model evaluation metrics. You can then begin with printing the accuracy of the model, follow it up using a confusion matrix, and then display the output of a function known as the `classification_report`. Scikit-learn's metrics module offers a function called `classification_report`, which can display different model evaluation metrics.

Let's explore the model metrics:

```
Model evaluation on training set
-----

Model accuracy - 91.25%

Confusion Matrix
=====
array([[40,  0],
       [ 7, 33]])
None

Classification Report
-----
```

	precision	recall	f1-score	support
0.0	0.85	1.00	0.92	40
1.0	1.00	0.82	0.90	40
avg / total	0.93	0.91	0.91	80

As you can see, the accuracy score is 91.25 percent. You will not repeat the definition accuracy; you can refer to the introduction part.

Let's consider the confusion matrix. The top-left cell represents a true positive cell. You should see that there are no false negatives, but there are seven false positives.

Lastly, you have precision, remember an F1 score, and support within the classification report. Let's consider their definitions:

Precision refers to the ratio of the true positive and the sum of the true positive and false positive.

Accuracy refers to the ration of the true positive and the total of the true positive and false negative.

An F1 score describes the harmonic mean of precision and sensitivity.

You will learn more about the above metric. For now, let's assume that we have high precision and recall values.

It is good to understand that we have about 91 percent accuracy for our model, but the actual test will be when it is performed on the test data. Consider the metrics for the test data:

```
Model evaluation on test set
=====

Model accuracy = 95.00%

Confusion Matrix
-----
array([[ 9,  1],
       [ 0, 10]])
None

Classification Report
=====
              precision    recall  f1-score   support

 0.0         1.00      0.90      0.95         10
 1.0         0.91      1.00      0.95         10

 avg / total         0.95      0.95      0.95         20
```

It is good to understand that our model contains 95 percent accuracy for the test data, which is a sign of a great job in a fitting model.

Let's look deeper into the model that we have created.

First, the function `get_params` is invoked. This function outputs all the parameters that are passed to the model. Let's define each of the parameters.

The first parameter describes the underlying data structure applied by the KNN implementation. Because each record within the training set has to be compared against another record, brute force implementation may require heavy resources. Therefore, you can decide either `kd_tree` or `ball_tree` as the data structure. A brute will apply the brute force method for looping through all the records for each record.

Leaf size represents the parameter passes to the `kd_tree` or `ball_tree` method.

Metric represents the distance measure applied to determine the neighbors. The p -value of two decreases the Minkowski to Euclidean distance.

Lastly, there is a weight parameter. KNN selects the class label of the test instance, depending on the class label of its K nearest neighbors. The highest vote decides the class label for the test instance. However, if you set the weights to distance, then each neighbor is assigned a weight that is inversely proportional to its distance. Therefore, to choose the class label of a test set, weighted voting is done instead of simple voting.

Chapter 8 - Deep Learning vs Machine Learning

Before we begin, it is important that you remind yourself of the basic definitions or explanations of these two subjects. Machine learning is a branch of artificial intelligence that uses algorithms to teach machines how to learn. Further from the algorithms, the machine learning models need input and output data from which they can learn through interaction with different users.

When building such models, it is always advisable to ensure that you build a scalable project that can take new data when applicable and use it to keep training the model and boost its efficiency. An efficient machine learning model should be able to self-modify without necessarily requiring your input, and still provide the correct output. It learns from structured data available and keeps updating itself.

Deep learning is a class of machine learning that uses the same algorithms and functions used in machine learning. However, deep learning introduces layered computing beyond the power of algorithms. Algorithms in deep learning are used in layers, with each layer interpreting data in a different way. The algorithm network used in deep learning is referred to as artificial neural networks.

The name artificial neural networks give us the closest iteration of what happens in deep learning frameworks. The goal here is to try and mimic the way the human brain functions, by focusing on the neural networks. Experts in deep learning sciences have studied and referenced different studies on the human brain over the years, which has helped spearhead research into this field.

Problem Solving Approaches

Let's consider an example to explain the difference between deep learning and machine learning.

Say you have a database that contains photos of trucks and bicycles. How can you use machine learning and deep learning to make sense of this data? At first glance, what you will see is a group of trucks and bicycles. What if

you need to identify photos of bicycles separately from trucks using these two frameworks?

To help your machine learning algorithm identify the photos of trucks and bicycles based on the categories requested, you must first teach it what these photos are about. How does the machine learning algorithm figure out the difference? After all, they almost look alike.

The solution is in a structured data approach. First, you will label the photos of bicycles and trucks in a manner that defines different features that are unique to either of these items. This is sufficient data for your machine learning algorithm to learn from. Based on the input labels, it will keep learning and refine its understanding of the difference between trucks and bicycles as it encounters more data. From this simple illustration, it will keep searching through millions of other data it can access to tell the difference between trucks and bicycles.

How Do We Solve This Problem In Deep Learning?

The approach in deep learning is different from what we have done in machine learning. The benefit here is that in deep learning, you do not need any labeled or structured data to help the model identify trucks from bicycles.

The artificial neural networks will identify the image data through the different algorithm layers in the network. Each of the layers will identify and define a specific feature in the photos. This is the same method that our brains use when we try to solve some problems.

Generally, the brain considers a lot of possibilities, ruling out all the wrong ones before settling on the correct one. Deep learning models will pass queries through several hierarchical processes to find the solution. At each identification level, the deep neural networks recognize some identifiers that help in distinguishing bicycles from trucks.

This is the simplest way to understand how these two systems work. Both deep learning and machine learning however, might not necessarily be applicable methods to tell these photos apart. As you learn about the differences between these two fields, you must remember that you have to define the problem correctly, before you can choose the best approach to implement in solving it. You will learn how to choose the right approach at

a later stage in your journey into machine learning, which has been covered in the advanced books in this series.

From the example illustrated above, we can see that machine learning algorithms need structured data to help them tell the difference between trucks and bicycles. From this information, they can then produce the correct output after identifying the classifiers.

In deep learning, however, your model can identify images of the trucks and bicycles by passing information through several data processing layers in its framework. There is no need for structured data. To make the correct prediction, deep learning frameworks depend on the output provided at every data processing layer. This information then builds up and presents the final outcome. In this case, it rules out all possibilities to remain with the only credible solution.

From our illustrations above, we have learned some important facts that will help you distinguish deep learning from machine learning as you learn over the years. We can summarize this in the following points:

- Data presentation

The primary difference between machine learning and deep learning is evident in the way we introduce data into the respective models. With machine learning models, you will almost always need to use structured data. However, in deep learning, the networks depend on artificial neural network layers to identify unique features that help to identify the data.

- Algorithms and human intervention

The emphasis of machine learning is to learn from interacting with different inputs and use patterns. From such interaction, machine learning models can produce better output the longer it learns, and the more interaction it receives.

To aid this cause, you must also try to provide as much new data as possible.

When you realize that the output presented is not what you needed, you must retrain the machine learning model to deliver a better output. Therefore, for a system that should work without human intervention, you will still have to be present from time to time.

In deep learning, your presence is not needed. All the nested layers within the neural networks process data at different levels. In the process, however, the model might encounter errors and learn from them.

This is the same way that the human brain works. As you grow up, you learn a lot of important life skills through trial and error. By making mistakes, your brain learns the difference between positive and negative feedback, and you strive to achieve positive results whenever you can.

To be fair, even in deep learning, your input will still be required. You cannot confidently assume that the output will always be perfect. This particularly applies when your input data is insufficient for the kind of output you demand from the model.

The underlying factor here is that both machine learning and deep learning must all use data. The quality of data you have will make a lasting impact on the results you get from these models. Speaking of data, you cannot just use any data you come across. To use either of these models effectively, you must learn how to inspect data and make sure you are using the correct format for the model you prefer.

Machine learning algorithms will often need labeled, structured data. For this reason, they are not the best option if you need to find solutions to sophisticated problems that need massive chunks of data.

In the example we used to identify trucks from bicycles, we tried to solve a very simple issue in a theoretical concept. In the real world, however, deep learning models are applied in more complex models. If you think about the processes involved, from the concepts to hierarchical data handling and the different number of layers that data must pass through, using deep learning models to solve simple problems would be a waste of resources.

While all these classes of AI need data to help in conducting the intelligence we require, deep learning models need significantly wider access to data than machine learning algorithms. This is important because deep learning algorithms must prove beyond a reasonable doubt that the output is perfect before it is passed. Deep learning models can easily identify differences and concepts in the data processing layers for neural networks only when they have been exposed to millions of data points. This helps to rule out all other possibilities. In the case of machine learning,

however, the models can learn through criteria that are already predetermined.

Different Use Cases

Having seen the difference between machine learning and deep learning, where can these two be applied in the real world? Deep learning is a credible solution in case you deal with massive amounts of data. In this case, you will need to interpret and make decisions from such data, hence you need a model that is suitable given your resource allocation.

Deep learning models are also recommended when dealing with problems that are too complicated to solve using machine learning algorithms. Beyond this, it is important to realize that deep learning models usually have a very high resource demand. Therefore, you should consider deep learning models when you have the necessary financial muscle and resource allocation to obtain the relevant programs and hardware.

Machine learning is a feasible solution when working with structured data that can be used to train different machine learning algorithms. There is a lot of learning involved before the algorithms can perform the tasks requested.

You can also use machine learning to enjoy the benefits of artificial intelligence without necessarily implementing a full-scale artificial intelligence model.

Machine learning algorithms are often used to help or speed up automation processes in businesses and industrial processes. Some common examples of machine learning models in use include advertising, identity verifiers, information processing, and marketing. These should help your business position itself better in the market against the competition.

Chapter 9 - Other Basics of the Python Code

Before we move on to some more of the coding that we need to focus on, it is important for us to take a closer look at some of the different basics that come with this kind of coding. Many of the basics that we are going to take a look at in this guidebook are going to be similar to what we may find in other coding languages, so if you have worked with others, some of these are going to seem a little bit familiar. With that in mind though, we are going to explore a few of the basics that come with Python, and learn how we can add these into our codes to make them perform.

First on the list is the Python keywords. These are going to be the words that the Python code will reserve because they are meant to tell the compiler command. You want to make sure that these are not used anywhere else in the code, and it is important that the programmer is able to use these in the proper manner.

For example, if you ever try to use these keywords in the wrong part of the code, or in the wrong manner, then it is going to result in an error. These keywords are there to make sure that the compiler has the commands that it needs which makes it easier for the compiler to know the best way to react to your code. Remember that these keywords are going to be very important to the code that you are writing, so make sure that they are used in the right area.

The second thing that we need to take a look at when working on our codes in Python code is the method that you have to name the identifiers. Your identifiers are going to be important to your code, and there are actually quite a few of these that you are able to work with. For example, as we go through this guidebook you may find them named as functions, entities, variables, and even classes.

When you are going through some of the rules of naming these identifiers, you will find that they are going to use the same rules and same information for each one, no matter what type it is. This is going to make the process of naming these so much easier overall. So, let's dive right in and figure out what some of the rules for naming these identifiers.

The first rule to remember is when you name these identifiers. You have many options when you are naming your identifiers. For example, you can

rely on both uppercase and lowercase letters with naming, as well as any number and the underscore symbol. You can also combine any of these together. One thing to remember here is that you can't start the name with a number and there shouldn't be any spaces between the words that you write out. So, you can't write out 3words as a name, but you can write out words3 or threewords.

When you pick out the identifier name, you can follow the rules above, and try to pick out a name that you can remember. Later on, when writing the code, you will need to pull it back up, and if you give it a name that is difficult to remember, you could run into problems or raise errors because things aren't showing up the way that you want. Outside of these rules, you will be fine naming the identifiers anything that you want.

Another question that a new programmer may have when working with this kind of coding is what the control flow of Python is all about. In any kind of language that you are working with while coding, the control flow is important to make sure that you are writing out the commands in the right manner. There are going to be a few string types in the code that you may need to write out so that the compiler knows what you want to have happened.

If you end up writing out some of your code wrongs, and the compiler is not able to make sense of it, then there will be an error that comes up when you try to run the code.

We will take a look at a few different examples of writing code in this guidebook, and it will be easy from those examples to really learn how the control flow works and how you can follow this when writing some of your own codes as well. Next on the list is going to be the Python statements. These are pretty simple ideas when we work in this language, but they are basically just the strings of code that you will write out, and that you want the compiler to list out on the screen when it executes that part of the code. It would be the Hello, World! Part of that code that we worked on before.

As long as you take the time to write out these statements in the proper manner, and put them after the command that tells the compiler to print them out on the screen, then you will find that these are going to show up on the screen. The statements can be as long or as short as you would like

and will vary based on the type of message that you would like to see show up on the screen.

Another thing that we need to take a look at when it comes to working with our own Python code is the comments. These are going to be unique to each language because they will ensure that we have the ability to write out a little kind of message in our code, one that other programmers are going to be able to see and read through, but are designed to not mess with the code at all.

The compiler can see the comment and knows to just skip over that part rather than adding it into the mix.

As you are writing out the code, you may find that there are times you want to include a little note or a little explanation of what you are writing inside the code.

These are little notes that you and other programmers are able to read through in the code and can help explain out what you are doing with that part of the code. Any comment that you write out in Python will need to use the # symbol ahead of it.

This tells the compiler that you are writing out a comment and that it should move on to the next block of code.

You can add in as many of these comments as you need to explain the code that you are writing and to help it make sense. You could have one very another line if you would like, but you should try not to add in too many or you may make a mess of the code that you have. But as long as the # symbol is in front of the statement, you can write out as many of these comments as you would like and your compiler will just skip out of them.

And finally, we are going to take a quick look at the operators and how these work with the Python language. Operators are simple, but there are several different types of these so we still need to spend some time learning how these are going to work. As we mentioned, there are a few different types that we are able to work with, and each one is designed to make different parts of the code work the way that you would like. For example, you could pull out the arithmetic operators that help you to multiply, divide, add, and subtract different parts of your code. You could work with the

assignment operators that are responsible for putting a specific value over to the variable, allowing the compiler to know how to treat this part.

You can also work with what are known as the comparison operators because these are going to allow you to look at two or more parts of your code, and then make a determination on whether they are similar and how the computer is supposed to react based on the information that you have given and the conditions that are set with that code.

Working with the Python language has a lot of different parts that need to come together in order to make it work.

But when we are able to follow some basics that come with this phase and that help us to learn this language, we will find that working with the Python coding language is a great experience to get your coding done.

How to Work with Variables in Python

Another topic that we need to spend some time looking at is how to work with some of the variables in Python. These variables are basically anything that is able to hold onto a value that we are able to change when we want. A variable is often just a box that is going to hold onto some value that we would like. These may seem simple, but we will see that it is something that is important to work with. This ensures that we can pull out a specific value later on in the code as needed.

These variables are going to be good to learn because they are going to be stored as a piece of the memory of the code. When it is needed inside of the compiler, you will just need to bring them up.

This means that any variable that you decide to create is going to be found in the memory, in specific locations, throughout the computer, which is going to ensure that when you execute the kind of code that you want to write, they will be easier to find. You can just tell the compiler what to do, and work from there.

With some of this information in mind, it is time for us to dive right in a bit more and see how to assign a value over to the variable, and how to make this whole process work for us.

Chapter 10 - Modules

A module is defined as an object of Python, which has certain attributes, to which we can give an arbitrary name and also link them. A module allows us to organize the code, to make easier the understanding and use of our code to third party people and even to ourselves.

In other words, we can define a module as a simple file with a .py extension that is capable of defining functions, variables, classes and also include executable code.

Why Should You Use Modules?

Modules allow us to organize the elements and components inside our codes in an easier way, providing us with a big package of variables that are auto contained. Names that are defined on a superior level in a module file automatically will become an attribute of the object of the imported module.

Another advantage of using modules is that they let us reuse the code, using data services and linking individual files to broaden our program.

The main reason why we think that the modules are a very useful tool when it comes to programming is that they are really helpful to organize and reuse our code. This is very important when we talk about OOP (Object-Oriented Programming) since on that mode, the modularization and reusage are very popular. Since Python is a programming language oriented for that, it comes very user-friendly.

Imagine that you want to create an application or a program, more complex than what we have been doing until now. For it, you are going to need one of the codes to complement. Here is when you see the real benefit of the modules since you will be able to simply add one of the old codes to the complex application you want to do.

In modules, we will also have modularization. It is based on dividing our codes into several tiny pieces of codes, so that, at the moment of making the complex program or application, it won't have hundreds and hundreds of lines of codes that could be annoying and hard to read. Instead, the code will be separated into tiny files.

How Do We Create a Module on Python?

Creating a module is something very easy that anyone can do, all that needs to be done is to create a file with the .py extension, then, that file will be stored on a folder of your preference; this is known as import.

Python, on its default library, has a big amount of modules, we can observe them on the official manual. You can find it on the following link: <http://docs.python.org/modindex.html> .

In case we want to create a module of our own, you will have to do the following. We will make a program on which we will create a module that could be used later.

```
modules.py X
1  def calculator(a, b, option):
2      if(option=="+"):
3          return a + b
4      elif (option=="-"):
5          return a - b
6      elif (option=="*"):
7          return a * b
8      elif (option==" / "):
9          return a / b
10     elif (option=="^"):
11         return a ^ b
12     else:
13         return "Bad option"
14
```

The module syntax is

as follows:

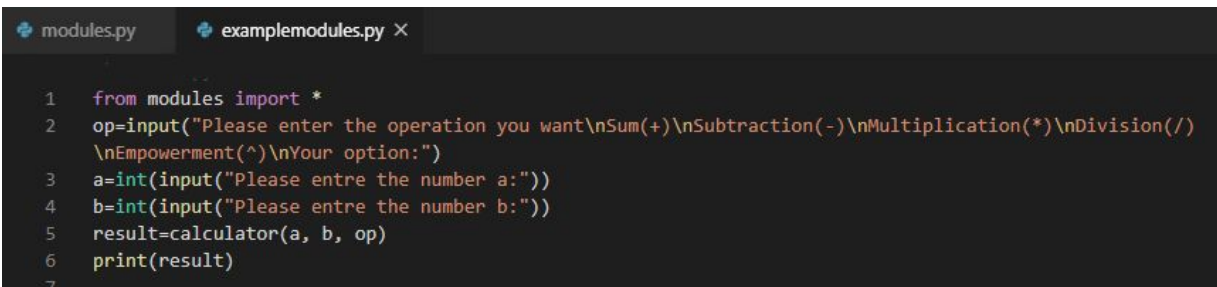
As you could see, the syntax is really simple, since it is pretty much like creating a function. After we created it, we must be able to import it from another program, to do that, we will use the import statement.

Import Statement

A module can contain definitions of a function and even statements, which can be executable. With this, it is possible to initialize a module since they execute only when our module is on the import statement.

Modules are capable of importing other modules, that is why people use to put the import type statements at the beginning of each since with the names of our imported modules, they will locate on a space named global; function that modules have for importing.

With the help of the last example, we can manage to import the module created and use the functions that we defined there.



```
modules.py  examplemodules.py X
1  from modules import *
2  op=input("Please enter the operation you want\nSum(+)\nSubtraction(-)\nMultiplication(*)\nDivision(/)\nEmpowerment(^)\nYour option:")
3  a=int(input("Please entre the number a:"))
4  b=int(input("Please entre the number b:"))
5  result=calculator(a, b, op)
6  print(result)
7
```

As you see in this example, we created the op variable, who takes the task of storing a string, which will specify the option that the users choose. Then, two variables would be initialized, a and b; they will store the value of the operators we are going to use to perform the mathematical operations.

Afterward, the result variable will store the value that the function calculator returns, according to the operators and the type of operation that the users want. The function calculator comes from the module that we have imported.

When the Python interpreter finds the import statement, it imports the module, as long as it is located on the full search path. The search path is nothing but a list where all the directories that Python accesses before importing any module are located.

How to Import a Module?

For being able to import a module, we just have to follow some instructions and steps that are performed at the moment of the execution:

We look for the module through the module search path, compile to byte code, and lastly, we execute the byte-code of our module to build then an

object that defines it.

How can I search for a module through Search Path?

To search for a module, our search system compounds of the concatenation of paths; these can be seen on the directory “Home” of our program. After this, the environment PYTHONPATH will be located from left to right, and that is how we will find the directory of default libraries.

Namespaces in Modules

As you know, modules are files. Python creates a module object in which all the names that we assigned in that module-file will be contained. What does that mean? This means that namespaces are just places where all the names that later become attributes are created.

What Are the Attributes?

Attributes are the names that have been assigned to a value considered of a higher level on a module file, which does not belong to a function or class.

Chapter 11 - Applications of Big Data Analysis

The applications of Big Data and Big Data Analytics are benefitting both small and big companies across various industrial domains. In this phase, we are going to explore more in detail such applications.

eCommerce

Over 2.6 billion and counting active social media users include customers and potential customers for every company out there. The race is on to create more effective marketing and social media strategies, powered by machine learning, aimed at providing enhanced customer experience to turn prospective customers into raving fans. The process of sifting through and analyzing a massive amount of data has not only become feasible, but it's easy now. The ability to bridge the gap between execution and significant Data Analysis has been supplemented by artificial intelligence marketing solutions.

Artificial Intelligence (AI) marketing can be defined as a method of you using artificial intelligence consonants like machine learning on available customer data to anticipate customer's needs and expectations while significantly improving the customer's journey. Marketers can boost their campaign performance and return on investment read a little to no extra effort in the light of big data insights provided by artificial intelligence marketing solutions. The key elements that make AI marketing as powerful are:

- Big data - A marketing company's ability to aggregate and segment a huge dump of data with minimal manual work is referred to as Big Data. The marketer can then leverage the desired medium to ensure the appropriate message is being delivered to the target audience at the right time.
- Machine learning - Machine learning platforms enable marketers to identify trends or common occurrences and gather effective insights and responses, thereby deciphering the root cause and probability of recurring events.

- Intuitive platform – Super fast and easy to operate applications are integral to AI marketing. Artificial intelligence technology is capable of interpreting emotions and communicating like a human, allowing AI-based platforms to understand open form content like email responses and social media.

Predictive Analysis

All artificial intelligence technology-based solutions are capable of extracting information from data assets to predict future trends. AI technology has made it possible to model trends that could be determined only retroactively. These predictive analysis models can be reliably used in decision-making and to analyze customers' purchase behavior. The model can successfully determine when the consumer is more likely to purchase something new or reorder an old purchase. The marketing companies are now able to reverse engineer customer's experiences and actions to create more lucrative marketing strategies. For example, FedEx and Sprint are using predictive analytics to identify customers who are at potential risk of deflecting to the competitor.

Smart Searches

Only a decade ago, if you type in "women's flip flops" on Nike, the probability of you finding what you were looking for would be next to zero. But today's search engines are not only accurate but also much faster. This upgrade has largely been brought on by innovations like "semantic search" and "natural language processing" that enable search engines to identify links between products and provide relevant search results, recommend similar items, and auto-correct typing errors. The artificial intelligence technology and big data solutions can rapidly analyze user search patterns and identify key areas that the marketing companies should focus on.

In 2015, Google introduced the first Artificial Intelligence-based search algorithm called "RankBrain." Following Google's lead, other major e-commerce websites, including Amazon has incorporated big Data Analysis and artificial intelligence into their search engines to offer smart search experience for their customers, who can find desired products even when they don't know exactly what they're looking for. Even small e-commerce stores have access to Smart search technologies like "Elasticsearch." The

data-as-a-service companies like "Indix" allow companies to learn from other larger data sources to train their product search models.

Recommendation Engines

Recommendation engines have quickly evolved into fan favorites and are loved by the customers just as much as the marketing companies. "Apple Music" already knows your taste in music better than your partner, and Amazon always presents you with a list of products that you might be interested in buying. This kind of discovery aide that can sift through millions of available options and hone in on an individual's needs are proving indispensable for large companies with huge physical and digital inventories.

In 1998, Swedish computational linguist, Jussi Karlgren, explored the practice of clustering customer behaviors to predict future behaviors in his report titled "Digital bookshelves." The same here, Amazon implemented collaborative filtering to generate recommendations for their customers. The gathering and analysis of consumer data paired with individual profile information and demographics, by the predictive analysis based systems allow the system to continually learn and adapt based on consumer activities such as likes and dislikes on the products in real-time. For example, the company "Sky" has implemented a predictive analysis based model that is capable of recommending content according to the viewer's mode. The smart customer is looking for such an enhanced experience not only from their Music and on-demand entertainment suppliers but also from all other e-commerce websites.

Product Categorization and Pricing

E-commerce businesses and marketing companies have increasingly adopted artificial intelligence in their process of categorization and tagging of the inventory. The Marketing companies are required to deal with awful data just as much, if not more than amazingly organized, clean data. This bag of positive and negative examples serves as training resources for predictive analysis based classification tools. For example, different detailers can have different descriptions for the same product, such as sneakers, basketball shoes, trainers, or Jordan's, but the AI algorithm can identify that these are all the same products and tag them accordingly. Or if

the data set is missing the primary keyword like skirts or shirts, the artificial intelligence algorithm can identify and classify the item or product as skirts or shirts based solely on the surrounding context.

We are familiar with the seasonal rate changes of the hotel rooms, but with the advent of artificial intelligence, product prices can be optimized to meet the demand with a whole new level of precision. The machine learning algorithms are being used for dynamic pricing by analyzing customer's data patterns and making near accurate predictions of what they are willing to pay for that particular product as well as their receptiveness to special offers. This empowers businesses to target their consumers with high precision and calculated whether or not a discount is needed to confirm the sale. Dynamic pricing also allows businesses to compare their product pricing with the market leaders and competitors and adjust their prices accordingly to pull in the sale. For example, "Airbnb" has developed its dynamic pricing system, which provides 'Price Tips' to the property owners to help them determine the best possible listing price for their property. The system takes into account a variety of influencing factors such as geographical location, local events, property pictures, property reviews, listing features, and most importantly, the booking timings and the market demand. The final decision of the property owner to follow or ignore the provided 'price tips' and the success of the listing are also monitored by the system, which will then process the results and adjust its algorithm accordingly.

Customer Targeting and Segmentation

For the marketing companies to be able to reach their customers with a high level of personalization, they are required to target increasingly granular segments. The artificial intelligence technology can draw on the existing customer data and train Machine learning algorithms against "gold standard" training sets to identify common properties and significant variables. The data segments could be as simple as location, gender, and age, or as complex as the buyer's persona and past behavior. With AI, Dynamics Segmentation is feasible which accounts for the fact that customers' behaviors are ever-changing, and people can take on different personas in different situations.

Sales and Marketing Forecast

One of the most straightforward artificial intelligence applications in marketing is in the development of sales and marketing forecasting models. The high volume of quantifiable data such as clicks, purchases, email responses, and time spent on webpages serve as training resources for the machine learning algorithms. Some of the leading business intelligence and production companies in the market are Sisense, Rapidminer, and Birst. Marketing companies are continuously upgrading their marketing efforts, and with the help of AI and machine learning, they can predict the success of their marketing initiatives or email campaigns. Artificial intelligence technology can analyze past sales data, economic trends as well as industrywide comparisons to predict short and long-term sales performance and forecast sales outcomes. The sales forecasts model aid in the estimation of product demand and to help companies manage their production to optimize sales.

Programmatic Advertisement Targeting

With the introduction of artificial intelligence technology, bidding on and targeting program based advertisement has become significantly more efficient. Programmatic advertising can be defined as “the automated process of buying and selling ad inventory to an exchange which connects advertisers to publishers.” To allow real-time bidding for inventory across social media channels and mobile devices as well as television, artificial intelligence technology is used. This also goes back to predictive analysis and the ability to model data that could only be determined retroactively. Artificial intelligence is able to assist the best time of the day to serve a particular ad, the probability of an ad turning into sales, the receptiveness of the user, and the likelihood of engagement with the ad.

Programmatic companies can gather and analyze visiting customers’ data and behaviors to optimize real-time campaigns and to target the audience more precisely. Programmatic media buying includes the use of "demand-side platforms" (to facilitate the process of buying ad inventory on the open market) and "data management platforms" (to provide the marketing company an ability to reach their target audience). In order to empower the marketing rep to make informed decisions regarding their prospective customers, the data management platforms are designed to collect and

analyze the big volume of website "cookie data." For example, search engine marketing (SEM) advertising practiced by channels like Facebook, Twitter, and Google. To efficiently manage huge inventory of the website and application viewers, programmatic ads provide a significant edge over competitors. Google and Facebook serve as the gold standard for efficient and effective advertising and are geared to words providing a user-friendly platform that will allow non-technical marketing companies to start, run and measure their initiatives and campaigns online.

Visual Search and Image Recognition

Leaps and bounds of the advancements in artificial intelligence-based image recognition and analysis technology have resulted in uncanny visual search functionalities. With the introduction of technology like Google Lens and platforms like Pinterest, people can now find results that are visually similar to one another using the visual search functionality. The visual search works in the same way as traditional text-based searches that display results on a similar topic. Major retailers and marketing companies are increasingly using the visual search to offer an enhanced and more engaging customer experience. Visual search can be used to improve merchandising and provide product recommendations based on the style of the product instead of the consumer's past behavior or purchases.

Conclusion

Data Science helps an individual predict and make decisions by taking advantage of prescriptive analytics, machine learning, and predictive causal analytics. The importance of data science can be summarized using the following points:

- Prescriptive analytics - If you require a model that has the intelligence and capability to make its own decisions, then prescriptive analytics is the best to use.

This new field delivers advice, it doesn't just predict, but it also recommends different prescribed actions and related outcomes. The best example to illustrate this is the Google self-driving car. Data that is collected by the vehicle is used to train cars. You can further mine this data by using algorithms to reveal intelligence. This will allow your car to make decisions such as when to turn, which path to take, as well as when to speed up or slow down.

- Machine learning for pattern discovery - Let us say that you don't have resources that you can apply to make predictions, it will require you to determine the hidden patterns in the data set to predict correctly. The most popular algorithm applied in pattern discovery is Clustering. Assume that you work in a telephone company and you want to determine a network by installing towers in the region. Therefore, you may use the clustering technique to determine the tower location that will make sure all users have the maximum signal strength.

- Make predictions with machine learning - If you desire to build a model that can predict the future trend of a company, then machine learning algorithms are the best to go with. This falls under supervised learning; it is called supervised because data is already present that you can use to train machines.

- Predictive causal analytics - If you require a model that can help predict chances of a given event happening in the future, you need to use predictive causal analytics.

This book discusses important concepts in data science such as artificial neural networks, machine learning, decision trees, clustering, big data

analytics, applications of machine learning, and many more interesting topics.

The next step is to start with some of the different programs and codes that we took some time to explore in this guidebook so that you can get more experience with the Python code. There are a lot of different things that you are able to do when it comes to the Python language, and being able to bring all of this together and getting it to work for your needs is going to be very important. This guidebook took some of the time that was needed to go over some of the basics of the Python language, as well as some of the more complex parts, and put it all together to help you see some results.

In addition to looking at all of the cool things that you are able to do with the Python language, we also spent some time in this guidebook looking at how to work with machine learning and data analysis. While there are many things that you are able to do with the help of Python and its various libraries and features, you will find that it is often used to help out with those two topics. Many companies and individuals want to learn how to use Python so that they can really see results with machine learning and data analysis, and this guidebook will help you to get started with some of that in no time.