**INDUSTRY INTERNSHIP REPORT**

**ON**

**“PYTHON WITH MACHINE LEARNING”**

**AT**

**FAME WORLD EDUCATIONAL HUB**

**Lucknow, Uttar Pradesh, India**

**AN INDUSTRY INTERNSHIP REPORT SUBMITTED**

**IN PARTIAL FULFILLMENT OF THE REQUIREMENTS**

**FOR THE AWARD OF DEGREE OF**

**BACHELOR OF ENGINEERING**

**In**

**COMPUTER SCIENCE & ENGINEERING**

**SUBMITTED BY**

SHAKIR HUSSAIN MALIK

Roll No:2020A1R019

**A picture containing text, clipart

Description automatically generated**

**SUBMITTED TO**

**Computer Science & Engineering**

**Model Institute of Engineering and Technology (Autonomous)**

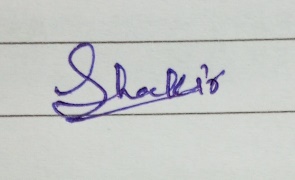
**Jammu, India**

**2022**

**CANDIDATES DECLARATION**



I, **Shakir Hussain Malik, Roll No.: 2020A1R019,** hereby declare that the work which is being presented in the Industry Internship Report entitled, “**Python with Machine Learning**” in partial fulfillment of requirement for the award of degree of B.E. (Computer Science & Engineering) and submitted in the Department of Computer Science & Engineering, Model Institute of Engineering and Technology (Autonomous), Jammu is an authentic record of my own work carried by me at “Fame World Educational Hub, Lucknow, Uttar Pradesh” under the supervision and mentorship of **Ms. Aishwarya Saxena,** Director, Fame World Educational Hub, Lucknow Uttar Pradesh. The matter presented in this report has not been submitted in this or any other University / Institute for the award of B.E. Degree.



*Signature of the Student*  *Dated*: 28/10/2022

**SHAKIR HUSSAIN MALIK**

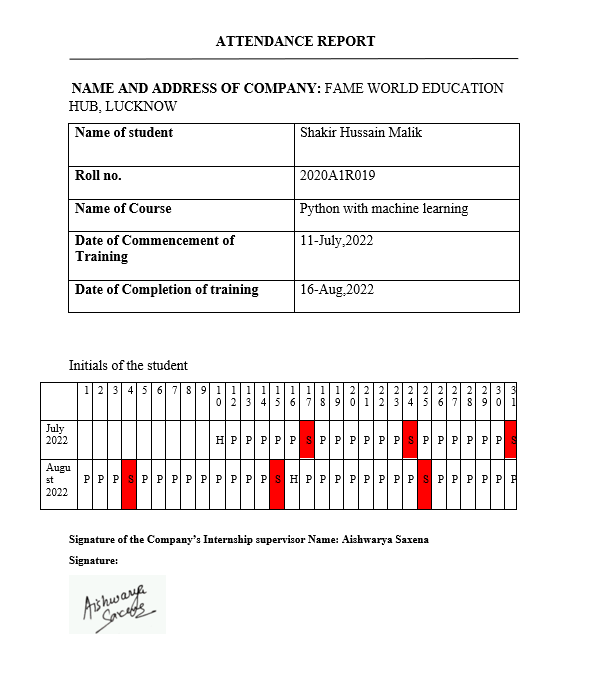
**2020A1R019**

**INTERNSHIP CERTIFICATE**





# ATTENDANCE REPORT



**Computer Science & Engineering**

**Model Institute of Engineering and Technology (Autonomous) Kot Bhalwal, Jammu, India**

***(NAAC “A” Grade Accredited)***



**Ref. No.: 2020A1R019 Date: 28-10-2022**

**CERTIFICATE**

Certified that this Industry Internship Report entitled **“PYTHON WITH MACHINE LEARNING”** is the bonafide work of “**Shakir Hussain Malik, Roll No. 2020A1R019, of 5th Semester, CSE, Model Institute of Engineering and Technology (Autonomous), Jammu”,** who carried out the Industry Internship at “Industry name, Location” work under my mentorship during July, 2022 - Aug, 2022.

**Dr. Mekhla Sharma**

**Assistant Professor,**

**Department of CSE, MIET**

*This is to certify that the above statement is correct to the best of my knowledge.*

**Prof. (Dr.) Ashok Kumar**

**Dean Academic Affairs,**

**Head of the Department,**

**Department of CSE, MIET**

**ACKNOWLEDGEMENTS**



This Summer internship opportunity was a great chance for learning and professional development. I am grateful for having a chance to meet so many wonderful people and professionals who led me through this internship period.

It is my pleasant duty to pay my heartfelt gratitude to Ms. Aishwarya Saxena, who have guided me through the course of this Internship.

I must record my deep sense of gratitude to Prof. (Dr.) Ankur Gupta (Director, MIET) and Prof. (Dr.) Ashok Kumar (Dean Academics & HOD CSE, MIET) for their guidance, constant inspiration and encouragement, and for their keen involvement throughout the course of present work.

Gratitude and thanks, although mean a very small thing to convey my thanks to my parents who have always given me a parental source of love, motivation and strength right from the journey of my life.

Bearing in mind previous I am using this opportunity to express my deepest gratitude and special thanks to the teachers who in spite of being extraordinarily busy with their duties, took time out to hear, guide and keep me on the correct path and allowing me to carry out my project at their esteemed organization and extending during the training.

I perceive this opportunity as a big milestone in my career development. I will strive to use gained skills and knowledge in the best possible way, and I will continue to work on their improvement, in order to attain desired career objectives. Hope to continue cooperation with all of you in the future.

I express my sincere gratitude to FAME WORLD EDUCATIONAL HUB, Lucknow, Uttar Pradesh and Model Institute of Engineering and Technology (Autonomous), Jammu for giving me the opportunity.

At the end thanks to the Almighty for everything.

**SHAKIR HUSSAIN MALIK**

**2020A1R019**

**SELF EVALUATION**



I am a 3rd year B.E. undergraduate student pursuing Computer Science and Engineering at Model Institute of Engineering and Technology, Jammu. I recently completed my internship training with “FAME WORLD EDUCATION HUB”, Lucknow as a Python Development Intern.

During my internship I learned quite a bit about Python programming language and its applications in day-to-day lives. Also, I learned about the Basics of Machine Learning in a very easy and efficient manner.

I was also provided with multiple assignments during my internship, which I always completed on time with full dedication and zeal. I still experienced a learning curve due to this being my first exposure to this kind of work. By the end of my internship, however, I felt comfortable in completing my assigned tasks and even received reviews from team leaders expressing their opinions about my work.

I developed great communication skills with people and this helped me to be a good team member when difficult situations aroused in meeting a deadline or solving a problem. Teamwork is valuable to me because I welcome co-worker insights into these types of challenges.

I totally understand the importance of regular practice and learning conceptual theories while being a CS student. And due to this internship opportunity, I got the chance to learn the topics not only theoretically but practically too. I got a firmer grasp on the coding part and learned a lot of new concepts.

While working as a Python Development Intern at Fame World, I gained a newer kind of experience which is surely going to help me for my future endeavors.

**SHAKIR HUSSAIN MALIK**

**2020A1R019**

**Contents**

Candidates Declaration 2

Internship Certificate 3

Attendance Report 4

Certificate 5

Acknowledgement 6

Self-Evaluation 7

Contents 8

**Chapter 1: Learning**  **10-16**

1.1 Introduction to Python 10

* + 1. Data-types in Python
    2. Variables in Python – Declaration and Use
    3. Operators
    4. Conditional Statements
    5. Functions
  1. Object Oriented Programming in Python **13**
     1. Classes an Objects
     2. Inheritance
     3. Constructors in Python
  2. Python Libraries and Packages **15**
     1. NumPy
     2. Pandas
     3. Matplotlib
     4. Seaborn
     5. Tkinter GUI

1.4 Introduction to Machine Learning **16**

**Chapter 2: Project** (Amazon order status prediction) **18**

2.1 Introduction

2.2 Objective

2.3 Implementation

2.4 Conclusion

**References 25**

**Figure and their references**

|  |  |  |
| --- | --- | --- |
| **S no.** | **Figure no.** | **Page no.** |
| **1** | **Fig-1** | **19** |
| **2** | **Fig-2** | **21** |
| **3** | **Fig-3** | **21** |
| **4** | **Fig-4** | **22** |
| **5** | **Fig-5** | **22** |
| **6** | **Fig-6** | **23** |
| **7** | **Fig-7** | **23** |

**Chapter 1 - Learning**

**1.1 Introduction to Python:**

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built-in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

* + 1. **Data-types in Python:**

Data types are the classification or categorization of data items. It represents the kind of value that tells what operations can be performed on a particular data.

Python has the following data types built-in by default, in these categories:

|  |  |
| --- | --- |
| Text Type: | str |
| Numeric Types: | int, float, complex |
| Sequence Types: | list, tuple, range |
| Mapping Type: | dict (dictionary) |
| Set Types: | set |
| Boolean Type: | bool |

* + 1. **Variables:**

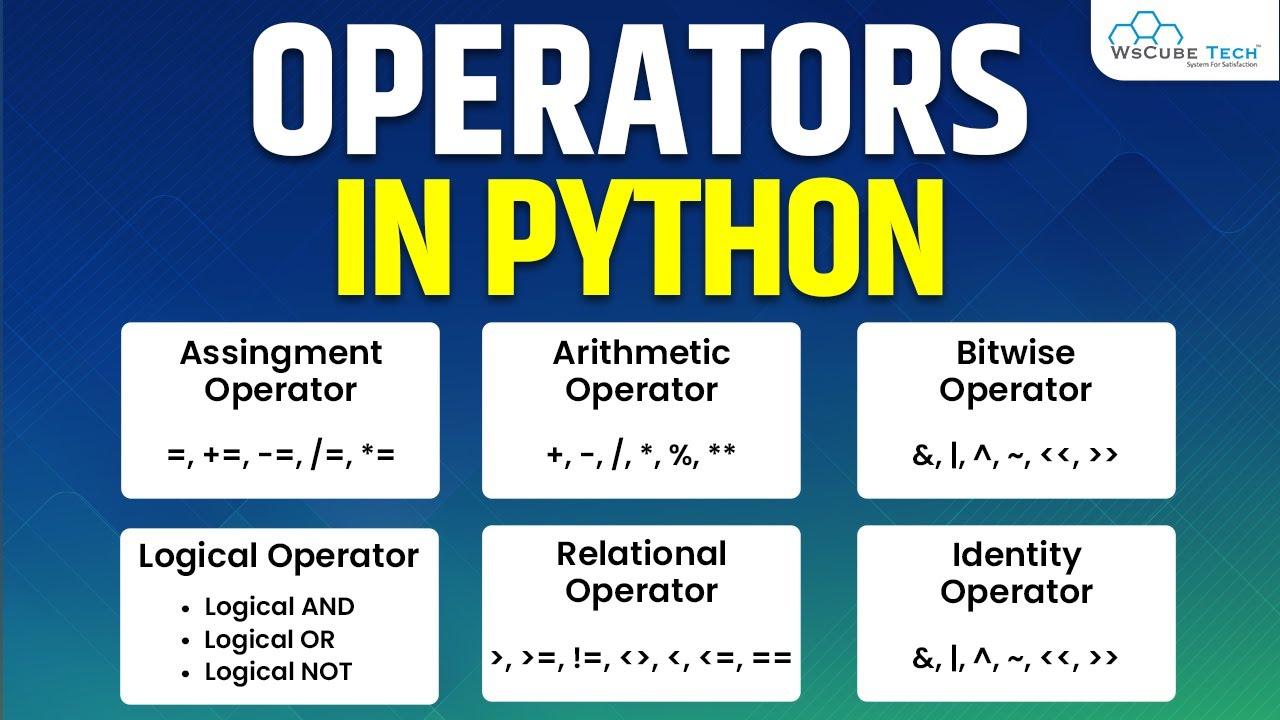
A Python variable is a symbolic name that is a reference or [pointer](https://realpython.com/pointers-in-python/) to an object. Once an object is assigned to a variable, you can refer to the object by that name. But the data itself is still contained within the object.

* + 1. **Operators:**

**Python Operators** in general are used to perform operations on values and variables. These are standard symbols used for the purpose of logical and arithmetic operations. In this article, we will look into different types of Python operators.

* OPERATORS: Are the special symbols. E.g. - +, \*, /, etc.
* OPERAND: It is the value on which the operator is applied.

Following are the types of operators in python:

****

* + 1. **Conditional Statements:**

A conditional statement as the name suggests itself, is used to handle conditions in your program. These statements guide the program while making decisions based on the conditions encountered by the program.

Python has 3 key Conditional Statements that you should know:

* *if*statement
* *if-else* statement
* *if-elif-else* ladder

**if Statement:**

The if statement is a conditional statement in python, that is used to determine whether a block of code will be executed or not. Meaning if the program finds the condition defined in the if statement to be true, it will go ahead and execute the code block inside the if statement.

**Syntax:**

**if** condition:

*# execute code block*

**if-else Statement:**

As discussed above, the *if* *statement* executes the code block when the condition is true. Similarly, the *else*statement works in conjuncture with the *if statement*to execute a code block when the defined *if condition*is false.

**Syntax:**

**if** condition:

*# execute code if condition is true*

**else**:

*# execute code if condition if False*

**if-elif-else ladder:**

The *elif* statement is used to check for multiple conditions and execute the code block within if any of the conditions evaluate to be true.

The *elif* statement is similar to the *else*statement in the context that it is optional but unlike the *else*statement, there can be multiple *elif* statements in a code block following an *if* statement.

**Syntax:**

**if** condition1:

*# execute this statement*

**elif** condition2:

*# execute this statement*

.

.

**else**:

*# if non of the above conditions*

*# evaluate to True*

*# execute this statement*

* + 1. **Functions:**

Functions are the basic building block of any python program, defined as the organized block of reusable code, which can be called whenever required.

A function is used to carry out a specific task. The function might require multiple inputs. When the task is done executing, the function can or cannot return one or more values.

There are two types of functions in python:

**Built-in Functions:**

Built-in functions are already defined in python. A user has to remember the name and parameters of a particular function. Since these functions are pre-defined, there is no need to define them again.

**User-Defined Functions:**

These functions are defined by a programmer to perform any specific task or to reduce the complexity of big problems and use that function according to their need.

**Example:**

def sub(x, y):

return x-y

print(sub(5,2))

* 1. **Object Oriented Programming:**

Since Python is an object-oriented programming language, almost everything in Python is an object, with its properties and methods. A Class is an object [constructor](https://www.scaler.com/topics/constructor-in-python/) or a blueprint from which objects are created. It provides a means of bundling data and functionality together.

* + 1. **Classes and Objects;**

Classes are the user-defined blueprints that help us create an object. [Objects](https://www.scaler.com/topics/object-in-python/) are the instances of a particular class. Every other element in Python will be an object of some class, such as the string, dictionary, number (20,30), etc. will be an object of some corresponding built-in class (int, str) in Python.

Creating a class is as easy as creating a function in Python. In function, we start with the *def* keyword while class definitions begin with the keyword *class*.

**Syntax:**

class ClassName:

*# Statement 1*

*# Statement 2*

Objects are different copies of the class with some actual values.

Before creating an object, we should know that it contains such properties to distinguish it from others.

* **State -** The state of an object is determined by the attributes of the object(i.e., the different items we have in the class, which the object inherits.).
* **Behavior -** The behavior is represented by the methods of the object. It shows the difference and similarities of the functionality of an object to other objects.
* **Identity -** Each and every object must be unique on its own. We can make this by giving it a unique name(like obj1, obj3, new\_obj, etc.).

**Syntax:**

[object\_name] = [class\_name](arguments/parameters)

* + 1. **Inheritance:**

Inheritance is the ability to ‘inherit’ features or attributes from already written classes into newer classes we make. These features and attributes are defined data structures and the functions we can perform with them, a.k.a. Methods. It promotes code reusability, which is considered one of the best industrial coding practices as it makes the codebase modular.

**Types of Inheritance in Python:**

1. **Single Inheritance:**

Single Inheritance is the simplest form of inheritance where a single child class is derived from a single parent class. Due to its candid nature, it is also known as Simple Inheritance.

1. **Multiple Inheritance**:

In multiple inheritance, a single child class is inherited from two or more parent classes. It means the child class has access to all the parent classes' methods and attributes.

1. **Multilevel Inheritance:**

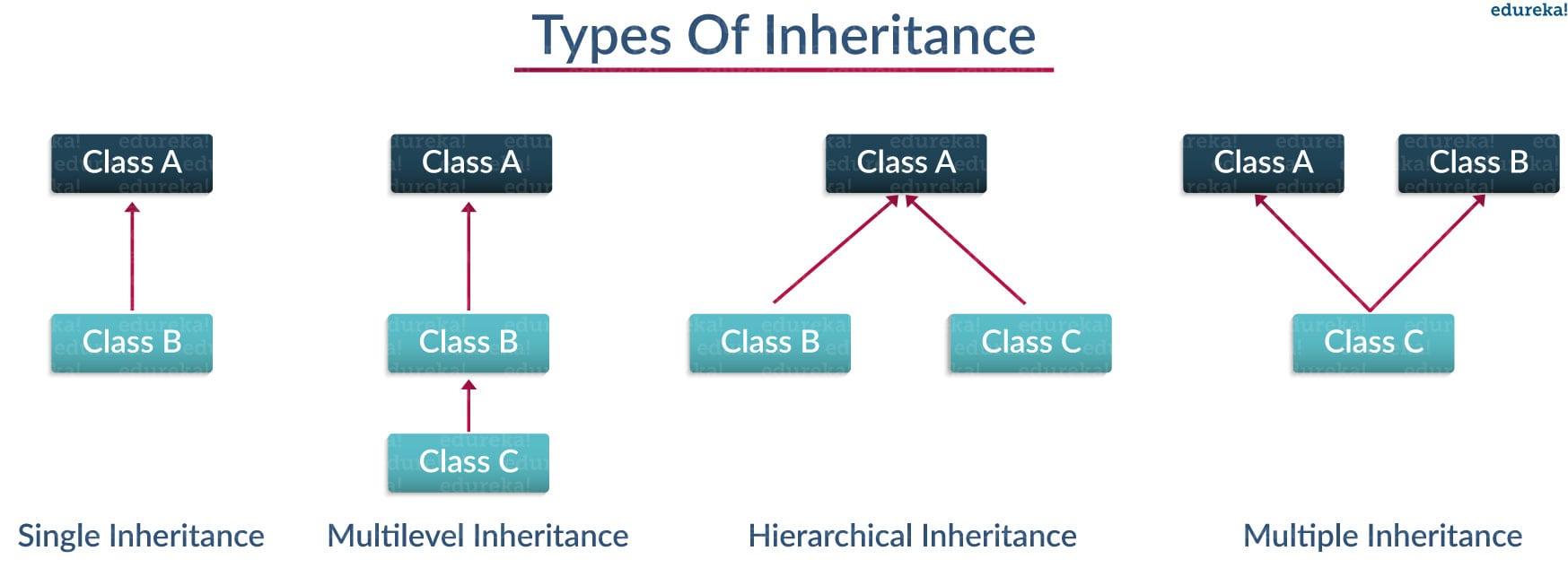
In multilevel inheritance, we go beyond just a parent-child relation. We introduce grandchildren, great-grandchildren, grandparents, etc. We have seen only two levels of inheritance with a superior parent class/es and a derived class/es, but here we can have multiple levels where the parent class/es itself is derived from another class/es.

1. **Hierarchical Inheritance:**

Hierarchical Inheritance is the right opposite of multiple inheritance. It means that, there are multiple derived child classes from a single parent class.

1. **Hybrid Inheritance:**

Hybrid Inheritance is the mixture of two or more different types of inheritance. Here we can have many relationships between parent and child classes with multiple levels.



* + 1. **Constructors:**

A constructor is a unique function that gets called automatically when an object is created of a class. The main purpose of a constructor is to initialize or assign values to the data members of that class. It cannot return any value other than none.

**Syntax:**

def \_\_init\_\_(self):

*# initializations*

**Types of Constructors:**

* **Parameterized Constructor:** When the constructor accepts arguments along with **self**, it is known as parameterized constructor.
* **Non-Parameterized Constructor:** When the constructor doesn't accept any arguments from the object and has only one argument, self, in the constructor, it is known as a non-parameterized constructor.
* **Default Constructor**: When you do not write the constructor in the class created, Python itself creates a constructor during the compilation of the program.
  1. **Libraries and Packages:**

**Package:** The package is a simple directory having collections of modules. This directory contains Python modules and also having [*\_\_init\_\_.py*](https://www.geeksforgeeks.org/__init__-in-python/) file by which the interpreter interprets it as a Package. The package is simply a namespace. The package also contains sub-packages inside it.

* + 1. **NumPy:**

NumPy is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more.

* + 1. **Pandas:**

Pandas is an open source Python package that is most widely used for data science/data analysis and machine learning tasks. It is built on top of another package named [Numpy](https://www.activestate.com/products/python/python-packages/), which provides support for multi-dimensional arrays. As one of the most popular data wrangling packages, Pandas works well with many other [data science](https://www.activestate.com/products/python/python-data-science/) modules inside the Python ecosystem, and is typically included in every Python distribution.

**Library:** The library is having a collection of related functionalities of codes that allows you to perform many tasks without writing your code. It is a reusable chunk of code that we can use by importing it into our program, we can just use it by importing that library and calling the method of that library with a period(.). However, it is often assumed that while a package is a collection of modules, a library is a collection of packages.

* + 1. **Matplotlib:**

Matplotlib is a python library used to create 2D graphs and plots by using python scripts. It has a module named pyplot which makes things easy for plotting by providing feature to control line styles, font properties, formatting axes etc. It supports a very wide variety of graphs and plots namely - histogram, bar charts, power spectra, error charts etc.

* + 1. **Seaborn:**

Seaborn is an amazing visualization library for statistical graphics plotting in Python. It provides beautiful default styles and color palettes to make statistical plots more attractive. It is built on the top of [matplotlib](https://www.geeksforgeeks.org/python-introduction-matplotlib/) library and also closely integrated to the data structures from [pandas](https://www.geeksforgeeks.org/introduction-to-pandas-in-python/).

* + 1. **Tkinter GUI:**

Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit.

* 1. **Introduction to Machine Learning:**

Machine learning is a branch of [artificial intelligence (AI)](https://www.ibm.com/in-en/cloud/learn/what-is-artificial-intelligence) and computer science which focuses on the use of data and algorithms to imitate the way that humans learn, gradually improving its accuracy.

Machine learning is an important component of the growing field of data science. Through the use of statistical methods, algorithms are trained to make classifications or predictions, uncovering key insights within data mining projects. These insights subsequently drive decision making within applications and businesses, ideally impacting key growth metrics. As big data continues to expand and grow, the market demand for data scientists will increase, requiring them to assist in the identification of the most relevant business questions and subsequently the data to answer them.

* + 1. **Machine Learning Algorithms:**

There are three main groups of Algorithms in Machine Learning:

**Supervised learning:** In supervised learning, the AI model is trained based on the given input and its expected output, i.e., the label of the input. The model creates a mapping equation based on the inputs and outputs and predicts the label of the inputs in the future based on that mapping equation.

Let’s suppose we have to develop a model that differentiates between a cat and a dog. To train the model, we feed multiple images of cats and dogs into the model with a label indicating whether the image is of a cat or a dog. The model tries to develop an equation between the input images and their labels. After training, the model can predict whether an image is of a cat or a dog even if the image is previously unseen by the model.

**Unsupervised learning:** In unsupervised learning, the AI model is trained only on the inputs, without their labels. The model classifies the input data into classes that have similar features. The label of the input is then predicted in the future based on the similarity of its features with one of the classes.

Suppose we have a collection of red and blue balls and we have to classify them into two classes. Let’s say all other features of the balls are the same except for their color. The model tries to find the dissimilar features between the balls on the basis of how the model can classify the balls into two classes. After the balls are classified into two classes depending on their color, we get two clusters of balls, one of blue color and one of red color

**Reinforcement learning:** In reinforcement learning, the AI model tries to take the best possible action in a given situation to maximize the total profit. The model learns by getting feedback on its past outcomes.

Consider the example of a robot that is asked to choose a path between **A** and **B**. In the beginning, the robot chooses either of the paths as it has no past experience. The robot is given feedback on the path it chooses and learns from this feedback. The next time the robot gets into a similar situation, it can use feedback to solve the problem. For example, if the robot chooses path **B** and gets a reward, i.e., positive feedback, this time the robot knows that it has to choose path **B** to maximize its reward.

**Chapter 2 – Project**

**Amazon: order status prediction**

**2.1 Introduction**

Amazon is the largest online marketplace and it simplifies the selling process to a large extent. There are two kinds of platforms for sellers on Amazon called Seller Central and Vendor Central. Depending on whether you’re selling directly to **Amazon (Seller Central) or directly to consumers through Amazon (Vendor Central)**, you can accordingly create an account. Note that you have to pay a fee when you use these accounts.

A survey found that most Amazon sellers have monthly sales at least **$1,000** in 2019. It was estimated that super-sellers could make up to **$3,000,000** annually. Amazon is one of the largest E-com companies in the world. It has truly changed the way we shop online. Starting an FBA business has become extremely popular.

**2.2 Objective**

* The project aims is to develop a Machine learning model for which would predict the order status (DILVERED TO BUYER OR RETURNED TO SELLER)
* The seller stores the item. Company picks up the item from the seller location and dispatch it to you.
* The company allows us to track the order and also if the item qualified, a free replacement will be available.
* This model helps us predicting the status of the order weather the order is being returned to seller or delivered to the customer.

**2.3 Implementation**

The aim of the project is to predict the status of order. We prepare a dataset by the keyword amazon seller (The dataset is obtained from the Kaggle <https://www.kaggle.com/>)Which gives the list of the buyer and the seller with other features like order no., shipping no., features of the product etc.

The link of the site from where we can download the dataset:

<https://www.kaggle.com/datasets/pranalibose/amazon-seller-order-status-prediction>

For reference purpose the Image of the dataset:

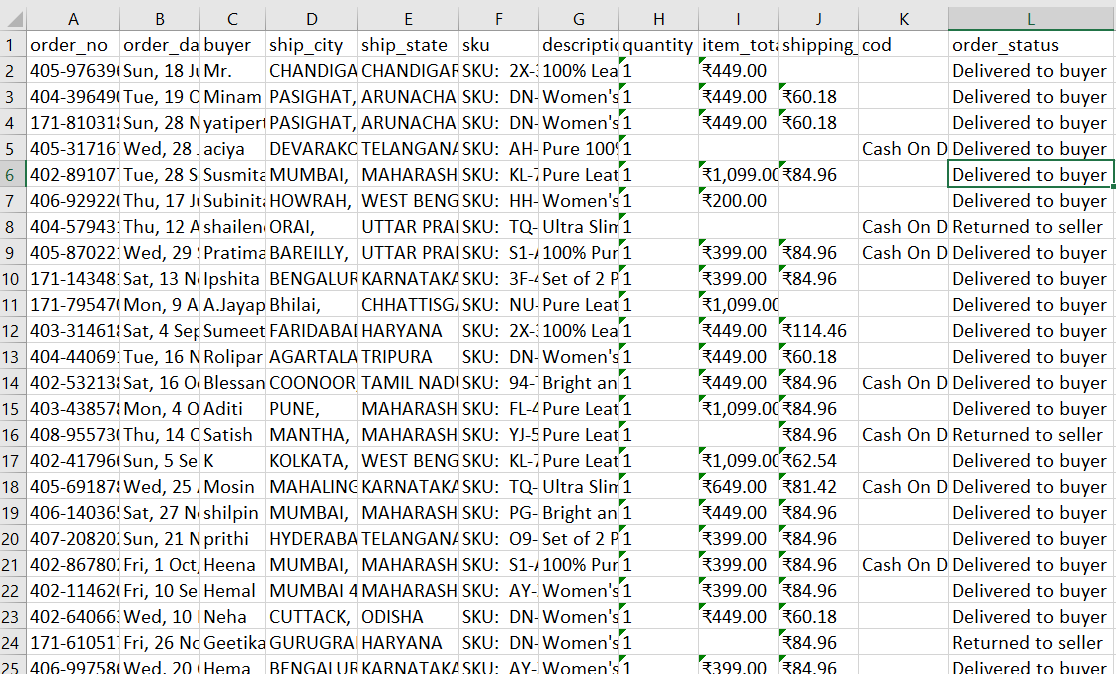


Fig-1

**2.3.1 Data Understanding**

**Independent features**

* order\_no - Unique Amazon Order Number
* order\_date - Date on which the order was placed
* buyer - Name of the buyer
* ship\_city - Delivery Address City
* ship\_state - Delivery Address State
* sku - Unique identifier of a product
* description - Product description
* quantity - Number of units ordered
* item\_total - Total amount paid by the buyer
* shipping\_fee - Charges borne by Boss Leathers to ship the item
* cod - Mode of payment: Cash on delivery or not

**Labeled/targeted feature**

* order\_status - Status of the order

**Understanding the data**

* There are 171 datapoints and 12 features
* order\_no and date are unique
* One buyer made at most 3 orders
* City,state, sku and description need further analysis
* The ordered quantity is at most 4 but median is 1
* Item\_total and shipping\_fee should be converted to float
* Cash on delivery has only 1 value, so it seems like boolean
* Order status is target variable, and has 2 classes.
* item\_total, shipping\_fee and cod has missing values, we should deal with them
* Order\_data should be date dtype
* Item\_total and Shipping\_fee should be float or int
* Cod should be Boolean
* Order\_status is also should be Boolean

**2.3.2 Data Cleaning**

**Checking for null values**

msno.matrix(data)

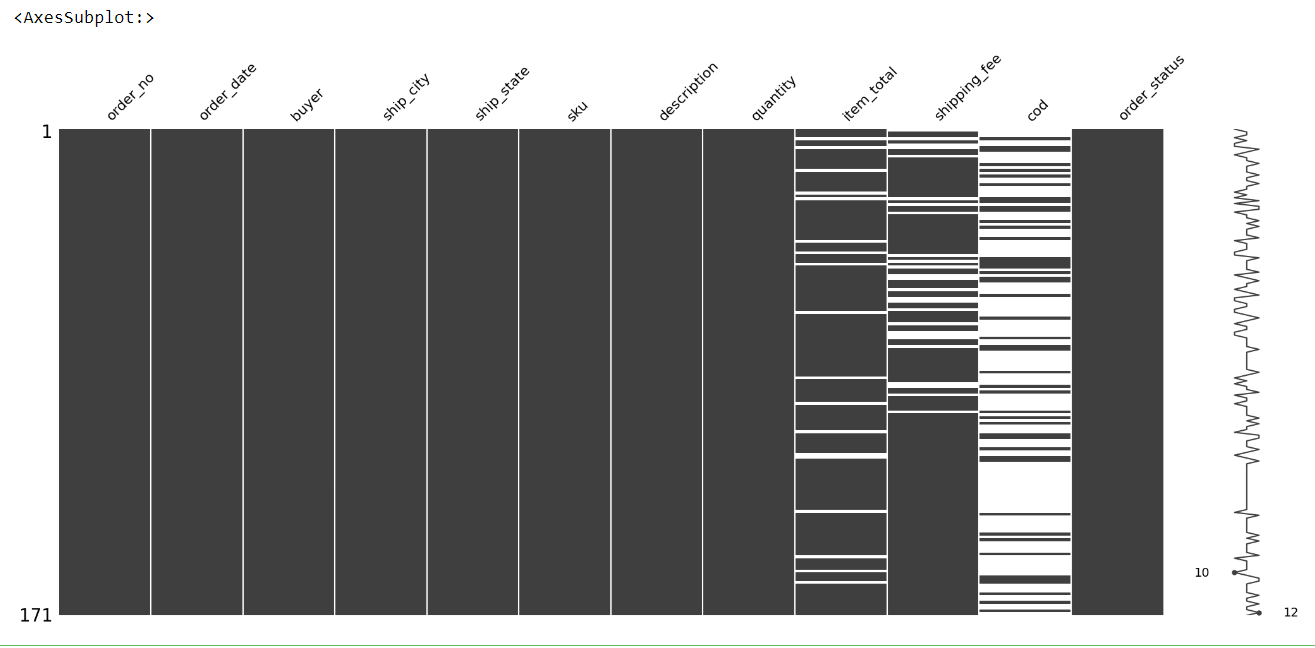
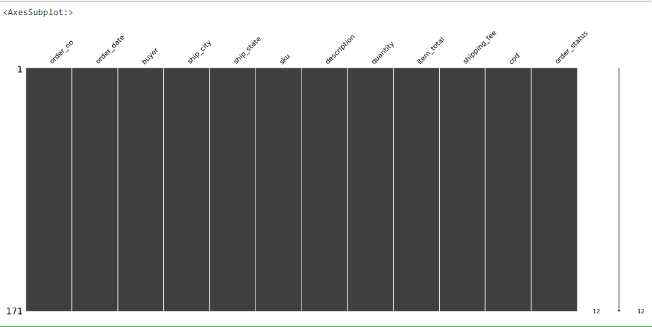


Fig-2(Graph taken from code)

**Filling of missing values**

msno.matrix(data)

 fig-3(Graph taken from code)

* + 1. **Data Exploring**

**Sale vs state**.

Shows the sale in different state and quantity of sale as well.

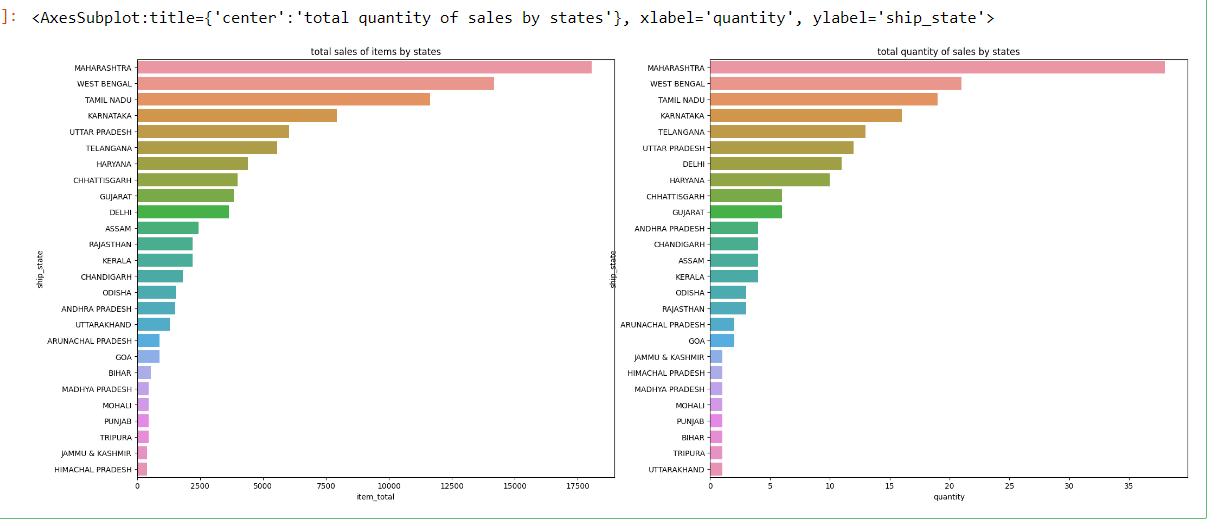


Fig-4(Graph taken from code)

**Sale vs payment method and return to seller.**

Graph used to analyses the relation between payment method and total items.



Fig-5(graph taken from code)

**SKU and Buyer vs Return to seller.**

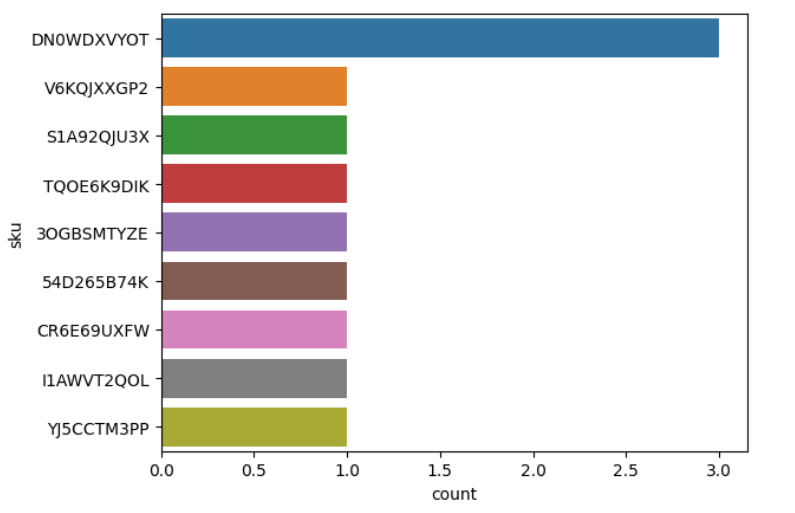
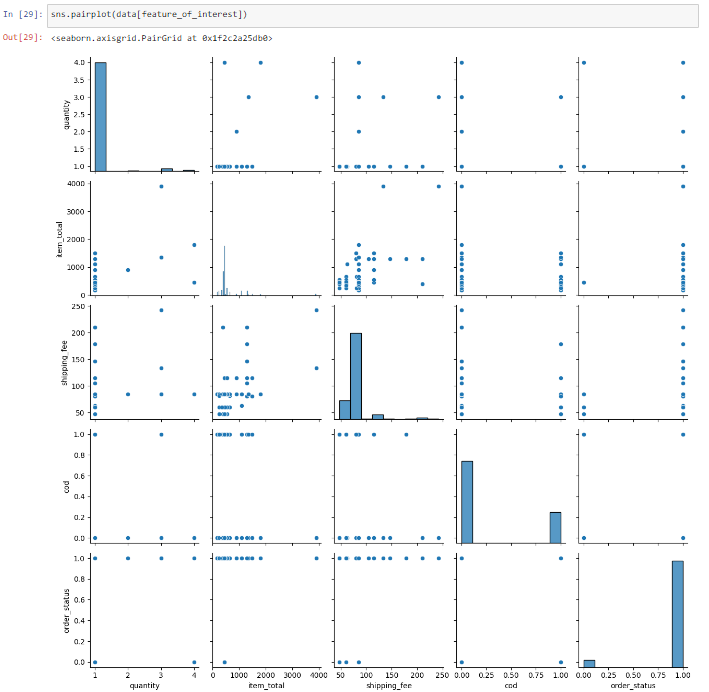


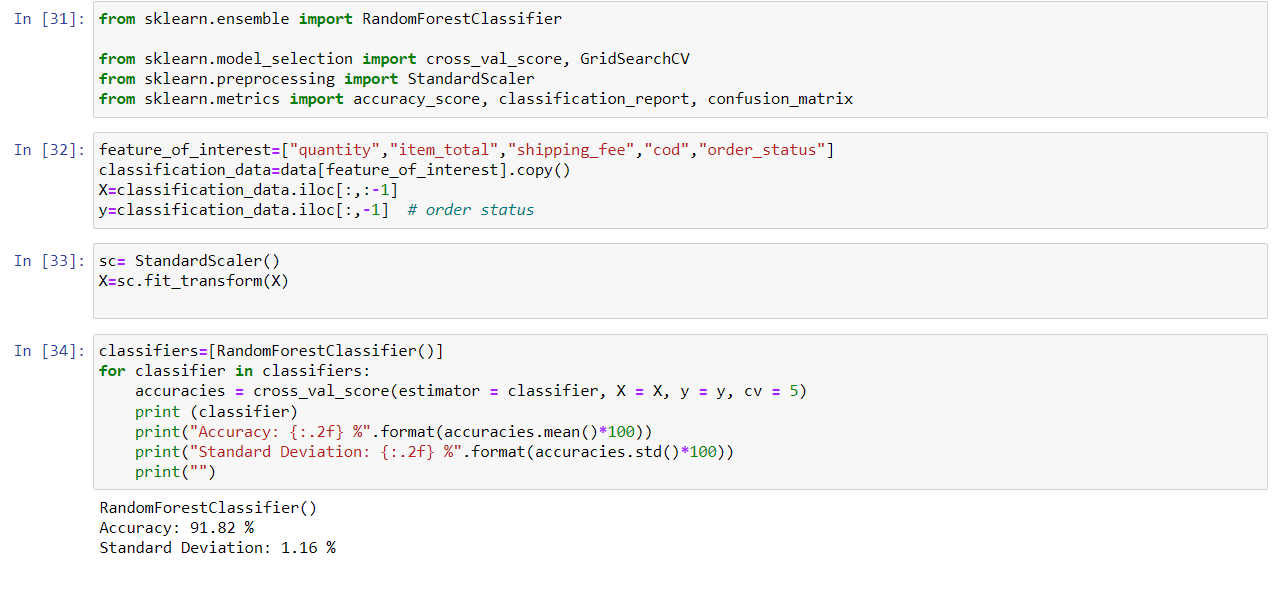
Fig-6(graph taken from code)

**Correlation Analysis**

Used to check kind of relation between the different attributes weather it is linear, logistic or anything else. Here it’s a random relation between attributes.

 Fig-7(graph taken from code)

* + 1. **Classification Models**



At last, we use random classifier model to train and predict order status. As shown above the accuracy of the model is 91.82%.

* 1. **Conclusion**

Through this project I have build the machine learning project to predict the status of the order and sale of the different items by analysing the different graphs and on the bases of that can make business plan accordingly and avoid loss and earn more and more profit.

**References**

* For different kind of query and doubt:
* <https://www.quora.com/search?q=what%20is%20the%20dilevery%20model%20for%20online%20apps>
* For study and research purpose:
* <https://www.amazon.in/gp/help/customer/display.html?nodeId=GJK6YT2NCPCAV7DC>
* For the dataset and project detail:
* <https://www.kaggle.com/>
* For the code and graph:
* <https://www.kaggle.com/datasets/pranalibose/amazon-seller-order-status-prediction>