

# **Personality Prediction**

**A Project Work**

*Submitted in the partial fulfilment for the award of the degree of*

**BACHELOR OF ENGINEERING**

**IN**

**Computer Science and Engineering**

**specialization in**

**Artificial Intelligence & Machine Learning**

**Submitted by:**

**Priya Yadav(20BCS6126)**

**Hitesh Kumar (20BCS6157)**

**Under the Supervision of:**

**Mrs. Lata Gupta**



**CHANDIGARH  
UNIVERSITY**

Discover. Learn. Empower.

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**APEX INSTITUTE OF TECHNOLOGY**

**CHANDIGARH UNIVERSITY, GHARUAN, MOHALI - 140413,  
PUNJAB**

**November & 2022**



**CHANDIGARH  
UNIVERSITY**

Discover. Learn. Empower.

## ***BONAFIDE CERTIFICATE***

*Certified that this project report **Personality Prediction** is the bonafide work of **Priya Yadav (20BCS6126)** and **Hitesh Kumar (20BCS6157)** who carried out the project work under my/our supervision.*

***SIGNATURE***

***SIGNATURE***

***HEAD OF THE DEPARTMENT***

***SUPERVISOR***

*Submitted for the project viva-voce examination held on*

***INTERNAL EXAMINER***

***EXTERNAL EXAMINER***

## DECLARATION

We **Priya Yadav and Hitesh Kumar** , student of **Bachelor of Engineering in Computer Science and Engineering specialization in Artificial Intelligence and Machine Learning**, session: **2020-24**, Department of Computer Science and Engineering, Apex Institute of Technology, Chandigarh University, Punjab, hereby declare that the work presented in this Project Work entitled **Personality Prediction** is the outcome of our own bona fide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. It contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

## **ACKNOWLEDGMENT**

This project wouldn't be feasible without the support of many people. We immensely obliged to my research supervisor, Mrs. Lata Gupta, for her epitome guidance, and ardent encouragement. I would also like to express our gratitude for her advice and contribution to the project and preparation of this report. I was able to procure invaluable insights during this product development and to explore and learn about many other machine learning fields. I am eternally grateful to my parents and friends for their support and much-needed encouragement throughout this research study.

# **ABSTRACT**

Personality is an important aspect of one's perspective towards their life. It majorly impacts the decision-making and approach to solving the problem. Personality prediction to better accuracy could be very useful for society. There are many papers and research conducted on the usefulness of the data for various purposes like marketing, dating suggestions, organization development, personalized recommendations, and health care to name a few. Particular approaches differ concerning different machine learning algorithms, data sources, and feature sets. The goal of this project is to investigate the predictability of the personality traits of users based on different features and measures of the Big 5 model.

# Table of Contents

Title Page	1
Declaration	2
Abstract	3
Acknowledgement	4
<b>1. Introduction</b>	<b>9</b>
1.1 Introduction	9
1.2 Scope	10
1.3 Project Summary and Purpose	11
1.4 Overview of the Project	12
1.5 Problem Definition	13
<b>2. Technical Literature Survey</b>	<b>14</b>
2.1 Literature review	14
<b>3. System Requirement Study</b>	<b>15</b>
3.1 User Characteristics	15
3.2 Hardware and Software Requirement	16
<b>4. System Analysis</b>	<b>18</b>
4.1 Technical Feasibility	18
4.1.1 Technical Feasibility	19
4.1.2 Operational Feasibility	21
4.1.3 Economic Feasibility	22
4.1.4 Schedule Feasibility	23
4.2 Requirement Definition	

<b>4.3 Study of Current System</b>	25
<b>4.4 Challenges in Personality Prediction</b>	36
<b>4.5 Application of Personality Prediction</b>	40
<b>5. System Design And architecture</b>	41
<b>6. System Testing</b>	43
<b>6.1 Unit testing</b>	45
<b>6.2 Integration Testing</b>	46
<b>6.3 System Testing</b>	50
<b>6.4 Performance Testing</b>	53
<b>6.5 Acceptance testing</b>	58
<b>7. RESULT DISCUSSION</b>	59
<b>8. Conclusion and Future Work</b>	60
<b>9. References</b>	82
<b>10. Annexure A</b>	83
<b>11. Annexure B</b>	84

## List of figures

1. FIG 1	23
2. FIG 2	26
3. FIG 3	26
4. FIG 4	27
5. FIG 5	36
6. FIG 6	36
7. FIG 7	37
8. FIG 8	38
9. FIG 9	39
10. FIG 10	39
11. FIG 11	40
12. FIG 12	40
13. FIG 13	41
14. FIG 14	41
15. FIG 15	42
16. FIG 16	43
17. FIG 17	66
18. FIG 18	67
19. FIG 19	68
20. FIG 20	74
21. FIG 21	75
22. FIG 22	76
23. FIG 23	77
24. FIG 24	78
25. FIG 25	79
26. FIG 26	80
27. FIG 27	81
28. FIG 28	82
29. FIG 29	83



## List of tables

1.	Table I	52
2.	Table II	55
3.	Table III	57
4.	Table IV	58
5.	Table V	58

# INTRODUCTION

## 1.1 Introduction

Personality is a way a person responds to a particular situation. It is a combination of characteristics that make an individual unique. Assessment of personality over the past two decades in various researches has revealed that personality can be defined by five dimensions known as Big Five personality traits. In general, the study of personality is considered as a psychology research based on the survey or questionnaire. But this limits the research data to a smaller number of persons. Hence there is a need for something through which we can increase the number of people involved in surveys and to make the process automated.

Personality identification of a human being by their nature is an old technique. Earlier these were done manually by spending a lot of time to predict the nature of the person. Data mining is primarily used today by companies with a strong consumer focus- retail, financial, communication, and marketing organizations. A dataset is scrapped from the internet and fed to the model, using machine learning techniques to understand the personality of the users to give accurate predictions. But these traditional methods are time consuming and very limited in scale. Our Proposed system will provide information about the personality of the user. Based on the personality traits provided by the user, System will match the personality traits with the data stored in the database. System will automatically classify the user's personality and will match the pattern with the stored data. System will examine the data stored in the database and will match the personality traits of the user with the data in the database. Then the system will detect the personality of the user. Based on the personality traits of the user, the system will provide other features that are relevant to the user's personality. Personality can also affect his/her interaction with the outside world and his/her environment. Personality can also be

used as an additional feature during the recruitment process, career counselling, health counselling, etc. Predicting personality by analysing the behaviour of the person is an old technique. This manual method of personality prediction required a lot of time and resources. Analysing personality based on one's nature was a tedious task and a lot of human effort would be required to do such analysis. Also, this manual analysis did not give accurate results while analysing the personality of a user from their nature and behaviour. Since analysis was done manually, it affects the accuracy of the results as humans prone to be prejudice and generally see the things accordingly

## **1.2 Scope**

We believe our results would build a brand-new reference for future comparative studies. We are going to explore future work diversely. Many users will get to know about their personality based on their personality traits. This system is useful for the social networking sites which help these sites to increase their rating scale. As the results are based on previous data stored in the database, the system will provide appropriate results. There can be a module where users can learn more about their personality and will be able to know the ways to improve areas they are lacking behind.

## **1.3 Project Summary and Purpose**

The primary purpose this project serves is to give users an insight into their personality. A machine learning model was employed for predicting the personality based on the data fed to the model. The dataset was scrapped from the internet that contains personal information of the users. A prototype is developed from where a set question is posed to the user, based on the answers provided we were able to

predict their personality with the accuracy of 82.34%. This project is able to predict the user's personality based on Big 5 model. Hence, this project was able to fulfil the targeted aim set.

## **1.4 Overview of the Project**

Personality prediction is based on identifying the personality of an individual using machine learning algorithms and big 5 models. Personality plays a major role in one's personal and professional life. Nowadays, many organizations have also started shortlisting the candidates based on their personality as this increases the efficiency of the work because the person is working in what he is good at rather than what he is forced to do. When statistical analysis is applied to personality survey data, some person words are used to describe the person and these words give a summary of the overall character or personality of the person accurately. Users can easily identify their personality from this system.

## **1.5 Problem Definition**

We are faced with many challenges which can be overcome by the advancing technology. One such challenge faced by every individual is getting to know their own self better. Knowing your personality helps you better answer the dilemma of how's and why's about yourself. You have a certain way and feel a certain way, but often it's hard to understand why, especially when other people aren't thinking or feeling the same way. Getting insights into your personality helps in career growth by moulding yourself accordingly. It's a vivid example of how technologies

with a transformative potential like the web bring with them each opportunity to get themselves known better by developing a personality prediction product.

Using the personality surveyed dataset, one will get a charming mix of different personality contents and different connected information like gender, age, grading based on user's answers to the question that pave the approach for perceptive analysis.

The dataset consists of user information that they provided during the survey. Survey question posed to the user by the system brings in a lot of information to pre-process and get inferences from, for the prediction. Operating with the dataset can assist you to perceive the challenges related to information provided by the user data processing and additionally find out about classifiers comprehensively. The foremost downside that we tend to be acting on as a beginner is to make a model to predict the personality of the user.

## **Chapter 2**

### **Technology Literature review**

Arsa and Shubhangi, (2015) [1], aimed for developing handwritten- based personality and behaviour identification systems. Supervised techniques Support vector machine and Artificial Neural network were employed into designing this model. They achieved their aim of securing the accuracy of 98.5%. In future, analysis will be performed for multiple lines.

Kedar et al. (2015) proposed [2], By using the supervised techniques ANN and Zernlike and Pseudo-Zernlike, Author builds a personality identification through handwriting analysis and graphology study, which is capable of determining personality of an individual by achieving the accuracy 90%. This system has its potential in personal recruitment in marketing, medicine, and counselling etc.

Ilmini and Fernando (2016) [3], build models to identify personality traits from face image, identification of criminal behaviour in criminology etc. Implementation of Supervised technique ANN and SVM into the model achieved the accuracy of 98.5%. Large dataset can improve the accuracy of the classification. More study on psychology and improved feature extraction phase may improve final results.

Sagadevan et al. (2015) [5], recognized the personality of facebook users from messages based on three Factor Personality (PEN) models. Techniques like supervised, stemming and part of speech tagging (POST) were employed into this system that managed to achieve the accuracy of 95%. In future, use of the higher negative words as cues to detect the psychoticism trait among Facebook users will be implemented.

## **Chapter 3**

### **System Requirement Study**

#### **3.1 Software and Hardware Requirements:**

##### Software Requirements:

- Operating System: Windows 7/8 / 8.1 / 10
- Microsoft Visio (2016)
- Microsoft Word (2016)
- R Studio
- Spyder
- Anaconda
- Google Drive

##### Hardware Requirements:

- Processor Intel i5 or later
- Motherboard Intel Ø Chipset Motherboard.
- 8GB or more
- Cache 512KB
- hard disk 16GB hard disk recommended
- floppy disk drive 1.44MB floppy disk drive
- monitor 1024x720 display
- speed 2.7GHz or more

## **Chapter 4**

### **System Analysis**

#### **4.1 Technical Feasibility**

A feasibility study is a preliminary study that examines information from potential users to determine the resource requirements, costs, benefits, and feasibility of the proposed system. The feasibility study considers various constraints that need to be implemented and operated by the system.

At this stage, the resources required for implementation, such as computer equipment, personnel, and costs, are estimated. The estimated resources are compared to the available resources and a cost-benefit analysis of the system is performed. The feasibility study involves analysing the problem and collecting all relevant information related to the project. The main purpose of the feasibility study is to determine if a project is feasible in terms of economic feasibility, technical feasibility and operational feasibility, and to schedule feasibility.

You need to make sure that the input data required for your project is available. Therefore, we evaluated the feasibility of the system in the following categories:

- Technical feasibility
- Operational feasibility
- Economic Feasibility
- Schedule feasibility

##### **4.1.1 Technical Feasibility**

Technical feasibility assessment is the most difficult part of the feasibility study.



This is because the system is not well designed at this time and it is difficult to access aspects such as performance and cost (depending on the nature of the technology being deployed). There are several aspects to consider in a technical analysis. Understand the different technologies included in the proposed system. Before starting the project,

We need to clarify the technology required to develop a new system. Is the required technology available? Our system is technically feasible since the required tools are easily available. HTML and Streamlit makes the system more user and developer friendly and although all tools seem to be easily available there are challenges too.

#### **4.1.2 Operational feasibility**

Proposed project is beneficial only if it can be turned into information systems that will meet the operating requirements. Simply stated, this test of feasibility asks if the system will work when it is developed and installed. Are there major barriers to Implementation? The proposal was to make a simplified web application. It is simpler to operate and can be used in any webpage, it is free and not costly to operate.

#### **4.1.3 Economic Feasibility**

Economic feasibility seeks to balance the costs of developing and implementing a new system with the benefits of adopting the new system. This feasibility study provides top management with the financial justification for the new system. Here, a simple profitability analysis that reflects a real cost-benefit comparison makes much more sense. In addition, it turns out to be a useful reference for comparing actual costs in the course of the project.

Automation can bring various kinds of intangible benefits. These may help improve product quality, improve information decision making and timeliness, speed up activities, improve operational accuracy, improve documentation and record keeping, and speed up information retrieval.

This is a web-based application. There is no cost to build the application.

#### **4.1.4 Schedule feasibility**

If the project takes too long to complete to be useful, the project will fail. This usually means estimating the time it takes to develop the system and whether it can be completed in a particular time frame using methods such as recovery. Schedule feasibility is a measure of how reasonable a project's schedule is. Given our technical know-how, is the project deadline reasonable? Some projects start with a specific deadline. It needs to be clear whether the deadline is mandatory or desirable. It may deviate slightly from the original schedule set at the start of the project. Application development can be done in a timely manner.

#### **4.2 Requirement Definition**

After a detailed analysis of system problems, understand the requirements for your current system. The requirements required by the system are categorized into functional and non-functional requirements, are listed below:

##### **4.2.1 Functional Requirements**

Functional requirements are features or features that need to be included in the system in order to meet business needs and be accepted by users. Based on this, the functional requirements that the system must meet are:

The system must be able to process new tweets that have been retrieved and stored

in the database.

The system needs to be able to analyse the data and classify the polarity of each tweet.

#### **4.2.2 Non-functional requirements**

Non-functional requirements are a description of system characteristics, properties, attributes, and constraints that may limit the proposed system limits.

Non-functional requirements are basically based on performance, information, economics, control, security efficiency and service.

Based on this, the non-functional requirements are:

- User friendly
- System should provide better accuracy
- To perform with efficient throughout and response time

#### **4.3 Study of Current System**

The approach followed for our current study is as follows:

1. Data collected from online sources was trained using machine learning classification techniques.
2. A prototype of a web browser was created using html and streamlit.

The above approach helped us reach our target.

#### **4.4 Big Personality Traits**

The Big Five Personality traits are the five dimensions or the domains of personality that can be used to analyse or predict the personality of a user. The Big Five Personality Model is the most widely accepted and researched model for predicting the personality of a user. The Big Five Personality traits are found in a variety of

people of different ages, locations and cultures. The Big Five Personality results are very accurate and predict the true personality of a user to a large extent. The Big Five Factors are:

1. Openness to Experience or Imagination Capability.
2. Agreeableness
3. Extraversion
4. Neuroticism or Emotional Stability
5. Conscientiousness

- The Big Five personality traits extraversion (also often spelled extroversion), agreeableness, openness, conscientiousness, and neuroticism.
- Each trait represents a continuum. Individuals can fall anywhere on the continuum for each trait.
- The Big Five remain relatively stable throughout most of one's lifetime.
- They are influenced significantly by both genes and the environment, with an estimated heritability of 50%.
- They are also known to predict certain important life outcomes such as education and health.

## **1. Openness:**

Openness to experience refers to one's willingness to try new things as well as engage in imaginative and intellectual activities. It includes the ability to "think outside of the box." This trait features characteristics such as imagination and insight. People who are high in this trait also tend to have a broad range of interests. They are curious about the world and other people and eager to learn new things and enjoy new experiences. People who are high in this trait tend to be more adventurous and creative. People low in this trait are often much more traditional

and may struggle with abstract thinking.

## **2. Conscientiousness:**

Conscientiousness describes a person's ability to regulate their impulse control in order to engage in goal-directed behaviours. It measures elements such as control, inhibition, and persistence of behaviour. Standard features of this dimension include high levels of thoughtfulness, good impulse control, and goal-directed behaviours.<sup>1</sup> Highly conscientious people tend to be organized and mindful of details. They plan ahead, think about how their behaviour affects others, and are mindful of deadlines.

## **3. Agreeableness:**

Agreeableness refers to how people tend to treat relationships with others. Unlike extraversion which consists of the pursuit of relationships, agreeableness focuses on people's orientation and interactions with others (Ackerman, 2017). This personality dimension includes attributes such as trust, altruism, kindness, affection, and other prosocial behaviours. <sup>1</sup> People who are high in agreeableness tend to be more cooperative while those low in this trait tend to be more competitive and sometimes even manipulative.

## **4. Extraversion:**

Extraversion reflects the tendency and intensity to which someone seeks interaction with their environment, particularly socially. It encompasses the comfort and assertiveness levels of people in social situations. Extraversion (or extroversion) is characterized by excitability, sociability, talkativeness, assertiveness, and high amounts of emotional expressiveness.<sup>1</sup> People who are high in extraversion are outgoing and tend to gain energy in social situations. Being around other people helps them feel energized and excited. People who are low in extraversion (or introverted) tend to be more reserved and have less energy to expend in social

settings. Social events can feel draining and introverts often require a period of solitude and quiet in order to "recharge."

## **Machine Learning**

Machine learning (ML) is the study of computer algorithms that improve automatically through experience and by the use of data. It is seen as a part of artificial intelligence. Machine learning algorithms build a model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to do so. Machine learning algorithms are used in a wide variety of applications, such as in medicine, email filtering, and computer vision, where it is difficult or unfeasible to develop conventional algorithms to perform the needed task. Machine learning-based text classifiers are a kind of supervised machine learning paradigm that requires you to train classifiers with labelled training data before applying them to actual classification tasks. Training data is usually an extracted part of the original data and is manually annotated. After proper training, you can apply them to the actual test data. Naive Bayes is a statistical classifier and Support Vector Machine is a kind of vector space classifier. The Naive Bayes (NB) Statistical Text Classification Scheme can be visualized as a two-class text classification problem, a positive class and a negative class, and can be adapted for use in emotion classification problems. A Support Vector Machine (SVM) is a type of vector space model-based classifier that requires a text document to be converted to a feature vector before it can be used for classification. Text documents are usually converted to multidimensional vectors.

## **Supervised learning**

It is the most common sub branch of machine learning today. Typically, new machine learning practitioners will begin their journey with supervised learning algorithms. Therefore, the first of this three-post series will be about supervised learning.

Supervised machine learning algorithms are designed to learn by example. The name “supervised” learning originates from the idea that training this type of algorithm is like having a teacher supervise the whole process.

When training a supervised learning algorithm, the training data will consist of inputs paired with the correct outputs. During training, the algorithm will search for patterns in the data that correlate with the desired outputs. After training, a supervised learning algorithm will take in new unseen inputs and will determine which label the new inputs will be classified as based on prior training data. The objective of a supervised learning model is to predict the correct label for newly presented input data. At its most basic form, a supervised learning algorithm can be written simply as:

$$Y = f(x)$$

Where  $Y$  is the predicted output that is determined by a mapping function that assigns a class to an input value  $x$ . The function used to connect input features to a predicted output is created by the machine learning model during training.

Supervised learning can be split into two subcategories: Classification and regression.

## Classification

During training, a classification algorithm will be given data points

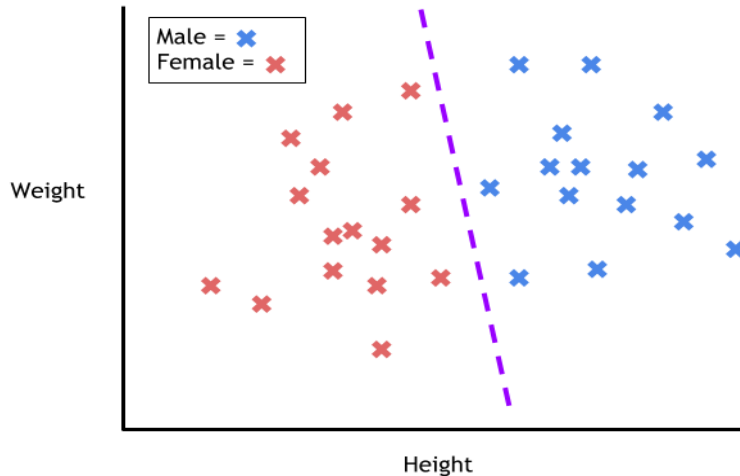


FIG I

with an assigned category. The job of a classification algorithm is to then take an input value and assign it a class, or category, that it fits into based on the training data provided.

The most common example of classification is determining if an email is spam or not. With two classes to choose from (spam, or not spam), this problem is called a binary classification problem. The algorithm will be given training data with emails that are both spam and not spam. The model will find the features within the data that correlate to either class and create the mapping function mentioned earlier:  $Y=f(x)$ . Then, when provided with an unseen email, the model will use this function to determine whether or not the email is spam.

Classification problems can be solved with a numerous number of algorithms.



Whichever algorithm you choose to use depends on the data and the situation. Here are a few popular classification algorithms:

- Linear Classifiers
- Support Vector Machines
- Decision Trees
- K-Nearest Neighbour
- Random Forest

## Regression

Regression is a predictive statistical process where the model attempts to find the important relationship between dependent and independent variables. The goal of a regression algorithm is to predict a continuous number such as sales, income, and test scores. The equation for basic linear regression can be written as so:

$$\hat{y} = w[0] * x[0] + w[1] * x[1] + \dots + w[i] * x[i] + b$$

Where  $x[i]$  is the feature(s) for the data and where  $w[i]$  and  $b$  are parameters which are developed during training. For simple linear regression models with only one feature in the data, the formula looks like this:

$$\hat{y} = wx + b$$

Where  $w$  is the slope,  $x$  is the single feature and  $b$  is the y-intercept. Familiar? For simple regression problems such as this, the model's predictions are represented by the line of best fit. For models using two features, the plane will be used. Finally, for a model using more than two features, a hyperplane will be used.

Imagine we want to determine a student's test grade based on how many hours they studied the week of the test. Let's say the plotted data with a line of best fit looks like this:

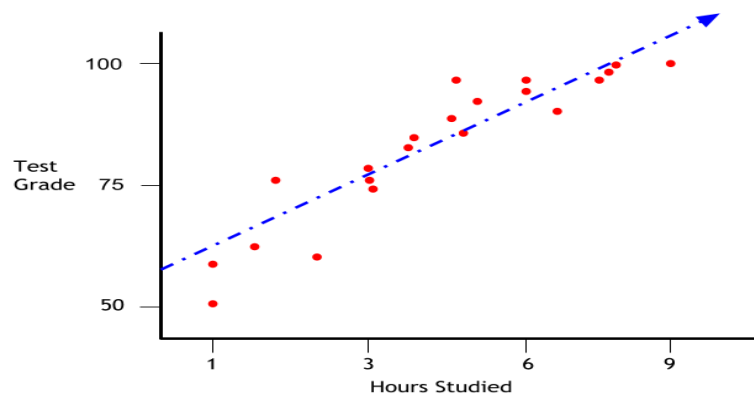


FIG II

There is a clear positive correlation between hours studied (independent variable) and the student's final test score (dependent variable). A line of best fit can be drawn through the data points to show the models predictions when given a new input. Say we wanted to know how well a student would do with five hours of studying. We can use the line of best fit to predict the test score based on other student's performances.

There are many different types of regression algorithms. The three most common

are listed below:

- Linear Regression
- Logistic Regression
- Polynomial Regression

### Simple Regression Example

First, we will import the needed libraries and then create a random dataset with an increasing output.

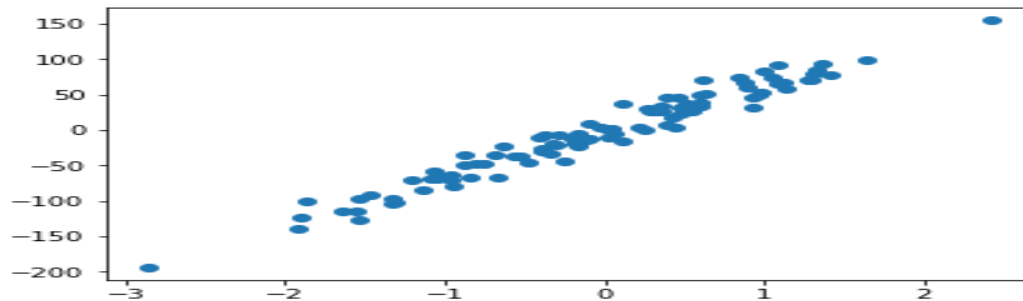


FIG III

We can then place our line of best fit onto the plot along with all of the data points.

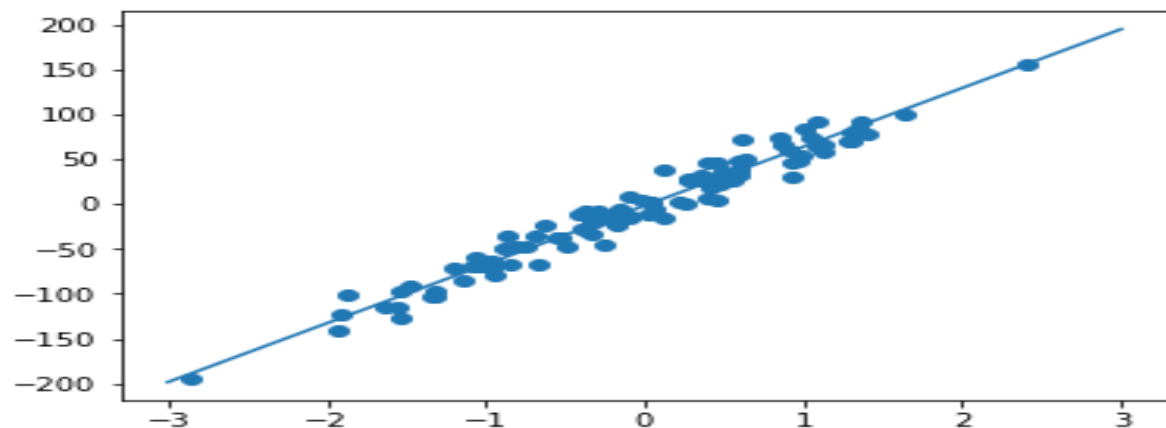


FIG IV

In middle school, we all learned that the equation for a linear line is  $y = mx + b$ . We can now create a function called “predict” that will multiply the slope ( $w$ ) with the new input ( $x$ ). This function will also use the intercept ( $b$ ) to return an output value. After creating the function, we can predict the output values when  $x = 3$  and when  $x = -1.5$ .

### 1.1.3 DATA PREPROCESSING

There are seven significant steps in data pre-processing in Machine Learning:

1. Acquire the dataset to build and develop Machine Learning models, you must first acquire the relevant dataset. This dataset will be comprised of data gathered from multiple and disparate sources which are then combined in a proper format to form a dataset. Dataset formats differ according to use cases.

2. Import all the crucial libraries Since Python is the most extensively used and also the most preferred library by Data Scientists around the world, we'll show you how to import Python libraries for data pre-processing in Machine Learning. Read more about Python libraries for Data Science [here](#). The predefined Python libraries can perform specific data pre-processing jobs. The three core Python libraries used for this data pre-processing in Machine Learning are:

- NumPy is the fundamental package for scientific calculation in Python. Hence, it is used for inserting any type of mathematical operation in the code. Using NumPy, you can also add large multidimensional arrays and matrices in your code.
- Pandas Pandas is an excellent open-source Python library for data manipulation and analysis. It is extensively used for importing and managing the datasets. It

packs in high-performance, easy-to-use data structures and data analysis tools for Python.

- Matplotlib is a Python 2D plotting library that is used to plot any type of charts in Python. It can deliver publication-quality figures in numerous hard copy formats and interactive environments across platforms (IPython shells, Jupyter notebook, web application servers, etc.).

3. Import the dataset in this step, you need to import the dataset/s that you have gathered for the ML project at hand. However, before you can import the dataset/s, you must set the current directory as the working `19 directory. Once you've set the working directory containing the relevant dataset, you can import the dataset using the “read\_csv()” function of the Pandas library. This function can read a CSV file (either locally or through a URL) and also perform various operations on it. The read\_csv() is written as: data\_set= pd.read\_csv('Dataset.csv').

4. Identifying and handling the missing values in data pre-processing, it is pivotal to identify and correctly handle the missing values, failing to do this, you might draw inaccurate and faulty conclusions and inferences from the data. Needless to say, this will hamper your ML project. Basically, there are two ways to handle missing data:

- Deleting a particular row – In this method, you remove a specific row that has a null value for a feature or a particular column where more than 75% of the values are missing. However, this method is not 100% efficient, and it is recommended that you use it only when the dataset has adequate samples. You must ensure that after deleting the data, there remains no addition of bias.

- Calculating the mean – This method is useful for features having numeric data like age, salary, year, etc. Here, you can calculate the mean, median, or mode of a particular feature or column or row that contains a missing value and replace the result for the missing value. This method can add variance to the dataset, and any loss of data can be efficiently negated. Hence, it yields better results compared to

the first method (omission of rows/columns). Another way of approximation is through the deviation of neighbouring values. However, this works best for linear data.

5. Encoding the categorical data Categorical data refers to the information that has specific categories within the dataset. In the dataset cited above, there are two categorical variables – country and purchased. Machine Learning models are primarily based on mathematical equations. Thus, you can intuitively understand that keeping the categorical data in the equation will cause certain issues since you would only need numbers in the equations.

6. Splitting the dataset Every dataset for Machine Learning model must be split into two separate sets – training set and test set. ` 20 Training set denotes the subset of a dataset that is used for training the machine learning model. Here, you are already aware of the output. A test set, on the other hand, is the subset of the dataset that is used for testing the machine learning model. The ML model uses the test set to predict outcomes.

7. Feature scaling Feature scaling marks the end of the data pre-processing in Machine Learning. It is a method to standardize the independent variables of a dataset within a specific range. In other words, feature scaling limits the range of variables so that you can compare them on common grounds.

## **Python**

Python is a popular object-oriented programming language having the capabilities of a high-level programming language. It's easy to learn syntax and portability capability makes it popular these days. The following facts give us the introduction to Python. Python was developed by Guido van Rossum at Sichtung Mathematics Centrum in the Netherlands. It was written as the successor of the programming language named 'ABC'. Its first version was released in 1991. The name Python

was picked by Guido van Rossum from a TV show named Monty Python's Flying Circus. It is an open-source programming language which means that we can freely download it and use it to develop programs. It can be downloaded from [www.python.org](http://www.python.org). Python programming language is having the features of Java and C both. It is having the elegant 'C' code and on the other hand, it is having classes and objects like java for object-oriented programming. It is an interpreted language, which means the source code of a Python program would be first converted into bytecode and then executed by a Python virtual machine. Why to learn "Python? Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently whereas other languages use punctuation, and it has fewer syntactical constructions than other languages. Python is a MUST for students and working professionals to become a great Software Engineer specially when they are working in Web Development Domain. I will list down some of the key advantages of learning Python.

Python is Interpreted – Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.

Python is Interactive – You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

Python is Object-Oriented – Python supports Object-Oriented style or technique of programming that encapsulates code within objects` 23

Python is a Beginner's Language – Python is a great language for the beginner level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games

Characteristics of Python Following are important characteristics of Python Programming

- It supports functional and structured programming methods as well as OOP.
- It can be used as a scripting language or can be compiled to byte-code for building large applications.

- It provides very high-level dynamic data types, supports dynamic type checking.
- It supports automatic garbage collection.

- It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

Applications of Python As mentioned before, Python is one of the most widely used languages over the web. I'm going to list a few of them here:

- Easy-to-learn – Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly
- Easy-to-read – Python code is more clearly defined and visible to the eyes
- Easy-to-maintain - Python's source code is fairly easy-to-maintain
- A broad standard library – Python's bulk of the library is very portable and cross platform compatible on UNIX, Windows, and Macintosh
- Interactive Mode – Python has support for an interactive mode which allows interactive testing and debugging of snippets of code
- Portable – Python can run on a wide variety of hardware platforms and has the same interface on all platforms

24 GUI Programming – Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix

Scalable – Python provides a better structure and support for large programs than shell scripting

NUMPY is one of the most powerful Python libraries. It is used in the industry for array computing. This article will outline the core features of the NumPy library. It will also provide an overview of the common mathematical functions in an easy-to-follow manner. NumPy is gaining popularity and is being used in a number of production systems. PANDAS is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series. It is free software released under the three-clause BSD license. The name is derived from the term "panel data", an econometrics term for data sets that include observations over multiple time periods for the same individuals.

- Data Frame object for data manipulation with integrated indexing.
- Tools for reading and writing data between in-memory data structures and different file formats.



- Data alignment and integrated handling of missing data.
- Reshaping and pivoting of data sets.
- Dataset merging and joining.

## **DJANGO**

Django is a high-level Python web framework that enables rapid development of secure and maintainable websites. Built by experienced developers, Django takes care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel. Django helps you write software that is:

- Complete
- Versatile
- Secure
- Scalable
- Maintainable
- portable `

## **SKLEARN**

Scikit-learn is probably the most useful library for machine learning in Python. The sklearn library contains a lot of efficient tools for machine learning and statistical modelling including classification, regression, clustering and dimensionality reduction. Components of scikit-learn:

- Supervised learning algorithms
- Cross-validation
- Unsupervised learning algorithms
- Various toy datasets
- Feature extraction.

## **MATPLOTLIB**

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. Matplotlib is one of the most popular Python packages used for data visualization. It is a cross-platform library for

making 2D plots from data in arrays. It provides an object-oriented API that helps in embedding plots in applications using Python GUI toolkits such as PyQt, WxPython or Tkinter. It can be used in Python and IPython shells, Jupyter notebook and web application servers also. Pickle in Python is primarily used in serializing and deserializing a Python object structure. In other words, it's the process of converting a Python object into a byte stream to store it in a file/database, maintain program state across sessions, or transport data over the network. You can pickle objects with the following data types:

- Booleans
- Integers
- Floats
- Complex numbers
- (normal and Unicode) Strings
- Tuples
- Lists
- Sets
- Dictionaries that obtain pickable object

It is always necessary to have interactive environments to create software applications and this fact becomes very important when you work in the fields of Data Science, engineering, and scientific research. The Python Spyder IDE has been created for the same purpose. In this article, you will be learning how to install and make use of Spyder or the Scientific Python and Development IDE.

Before moving on, let's take a look at all the topics that are discussed over here:

- What is Python Spyder IDE?

- Features of Spyder
- Python Spyder IDE Installation
- Creating a file/ Starting a Project
- Writing the Code
- Variable Explorer
- File Explorer
- Configuring Spyder
- Help

What is Python Spyder IDE?

Spyder is an open-source cross-platform IDE. The Python Spyder IDE is written completely in Python. It is designed by scientists and is exclusively for scientists, data analysts, and engineers. It is also known as the Scientific Python Development IDE and has a huge set of remarkable features which are discussed below.

Features of Spyder

Some of the remarkable features of Spyder are:

- Customizable Syntax Highlighting
- Availability of breakpoints (debugging and conditional breakpoints)
- Interactive execution which allows you to run line, file, cell, etc.
- Run configurations for working directory selections, command-line options, current/ dedicated/ external console, etc
- Can clear variables automatically (or enter debugging)
- Navigation through cells, functions, blocks, etc can be achieved through the Outline Explorer
- It provides real-time code introspection (The ability to examine what

functions, keywords, and classes are, what they are doing and what information they contain)

- Automatic colon insertion after if, while, etc
- Supports all the IPython magic commands
- Inline display for graphics produced using Matplotlib
- Also provides features such as help, file explorer, find files, etc

### Python Spyder IDE Installation (Installing with Anaconda — Recommended)

The Python Spyder IDE comes as a default implementation along with Anaconda Python distribution. This is not just the recommended method but also the easiest one. Follow the steps given below to install the Python Spyder IDE:

- Go to the official Anaconda website using the following link:  
<https://www.anaconda.com>
- Click on the Download option on the top right as shown below:



FIG V

- Choose the version that is suitable for your OS and click on Download.

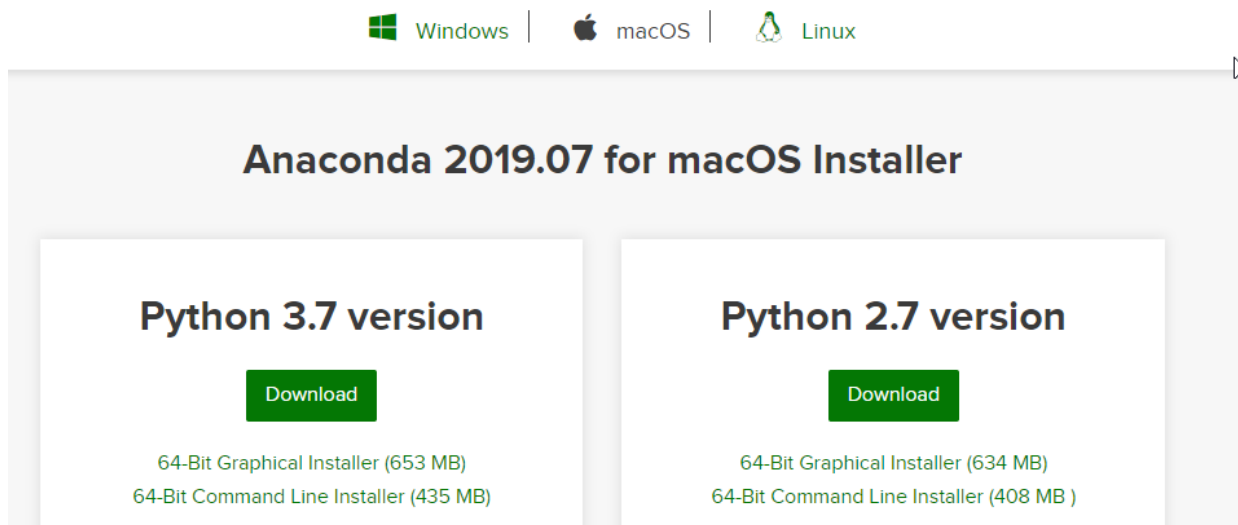


FIG VI

- Once the installer is downloaded, you can see a dialog box for the Setup. Complete the Setup and click on Finish.
- Then, search for Anaconda Navigator in the search bar of your system and launch Spyder. Once launched, you will see a screen similar to the one below:

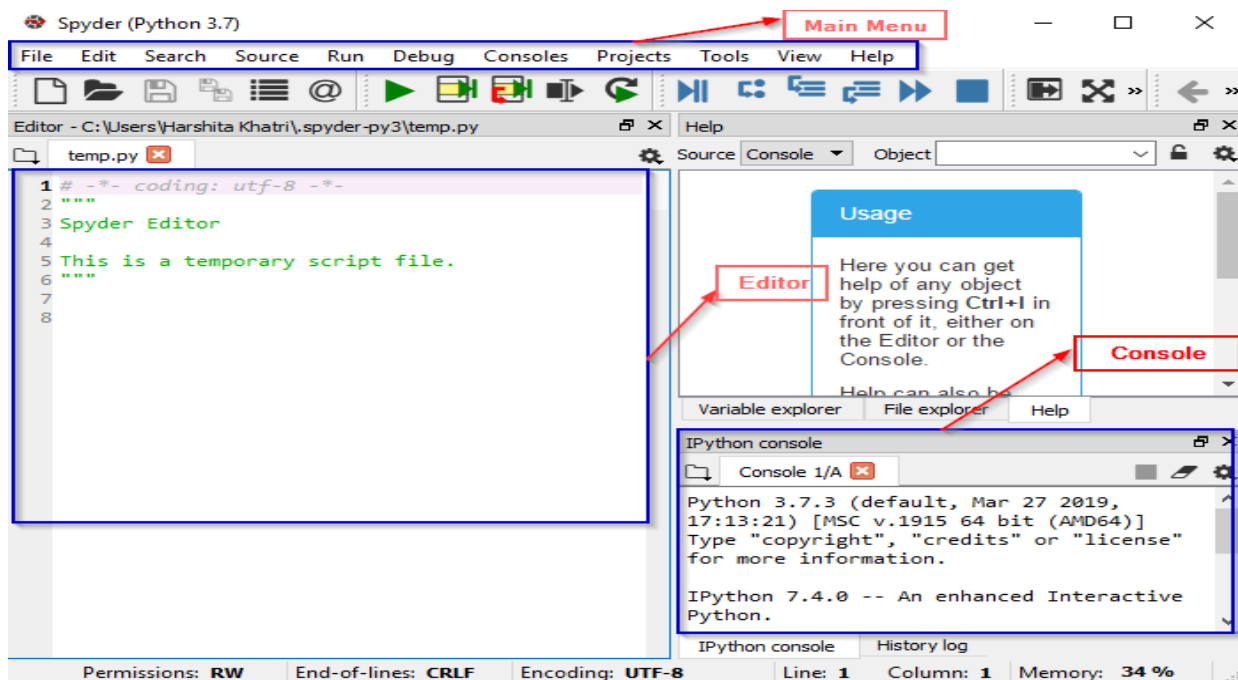


FIG VII

Writing the code:

Writing code in Spyder becomes very easy with its multi-language code editor and a number of powerful tools. As mentioned earlier, the editor has features such as syntax highlighting, real-time analysis of code, style analysis, on-demand completion, etc. When you write your code, you will also notice that it gives a clear call stack for methods suggesting all the arguments that can be used along with that method.

Take a look at the example below:

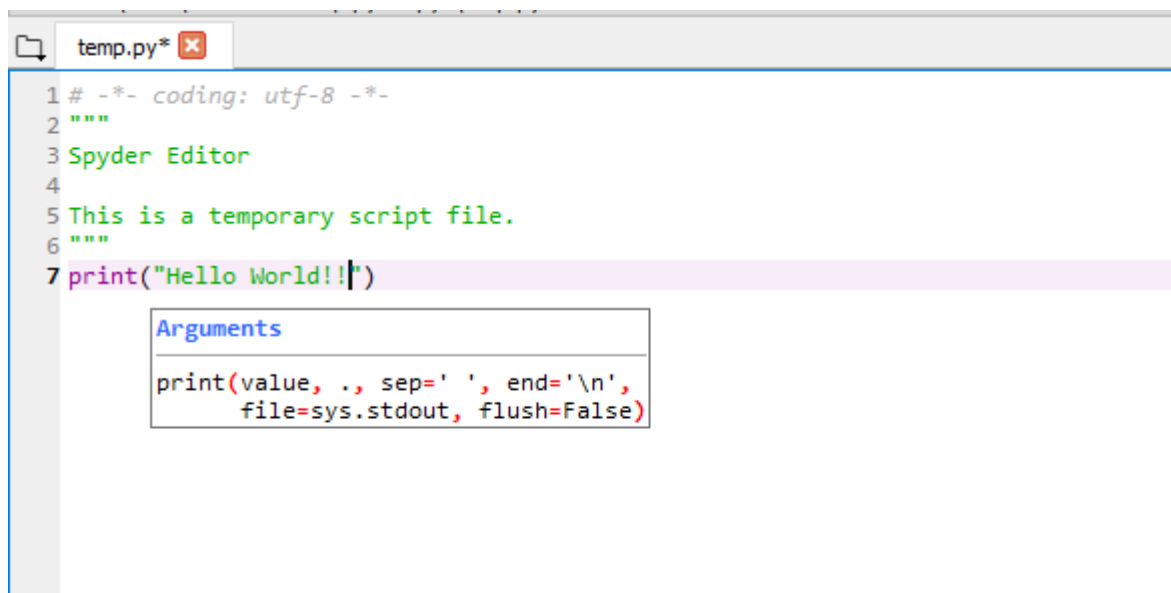


FIG VIII

In the above example, you can notice that the editor is showing the complete syntax of the *print* function. Not just this, in case you have made an error in any line, you will be notified about it before the line number with a message describing what the issue is. Take a look at the image below:

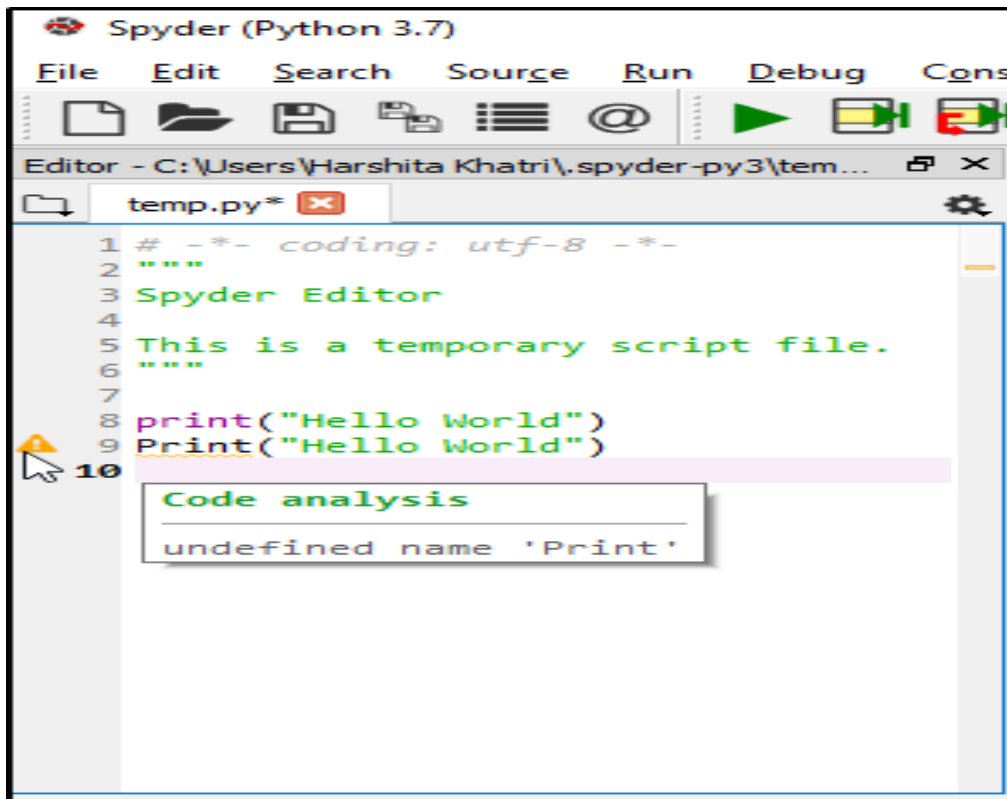


FIG IX

To run any file, you can select the *Run* option and click on run. Once executed, the output will be visible on the Console as shown in the image below:

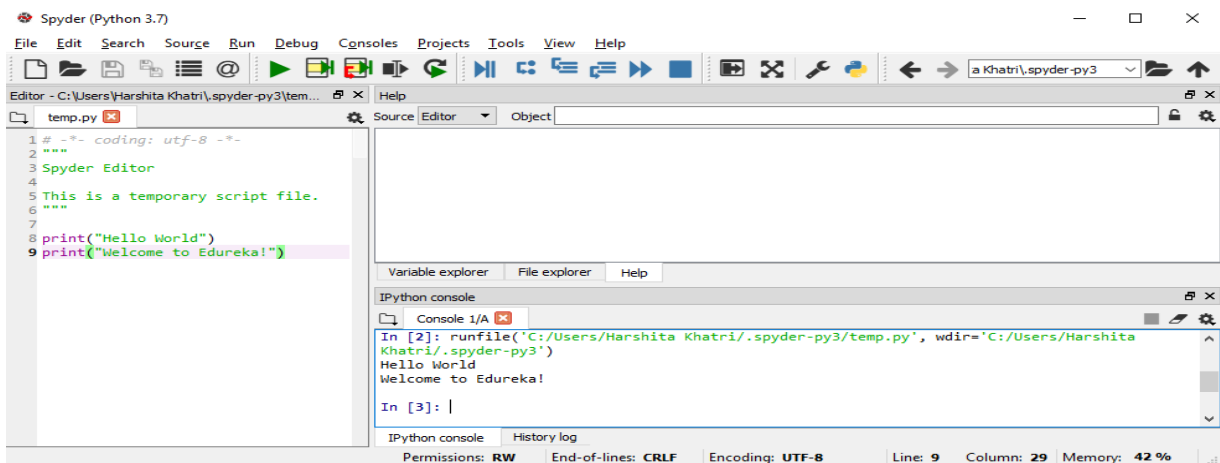


FIG X

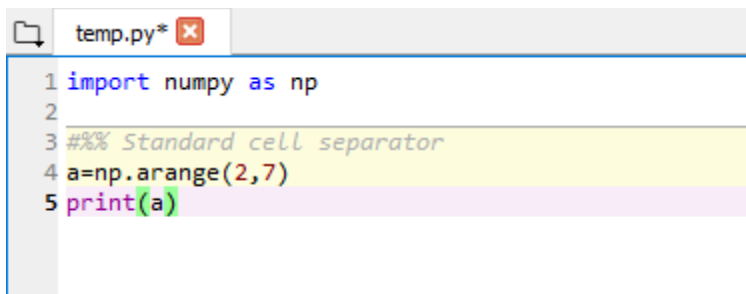
Code cells:

You can define code cells easily using the following:

Type	Description
<code>###</code>	Standard cell separator
<code># %%</code>	Standard cell separator, when the file has been edited with Eclipse
<code># &lt;codecell&gt;</code>	IPython notebook cell separator

FIG XI

For example, when you use the Standard cell separator, you will see that the code has been separated as follows:



The screenshot shows a code editor window titled 'temp.py'. The code is as follows:

```
1 import numpy as np
2
3 ### Standard cell separator
4 a=np.arange(2,7)
5 print(a)
```

The code is color-coded: 'import' is blue, 'numpy' is green, 'as' is blue, 'np' is green, '###' is green, 'Standard cell separator' is green, 'a=' is blue, 'np.' is green, 'arange' is green, '(2,7)' is green, 'print' is blue, and '(a)' is green.

FIG XII

Variable Explorer:

The Variable Explorer shows all the global objects references such as modules, variables, methods, etc of the current IPython Console. Not just this, you can also interact with these using various GUI based editors.



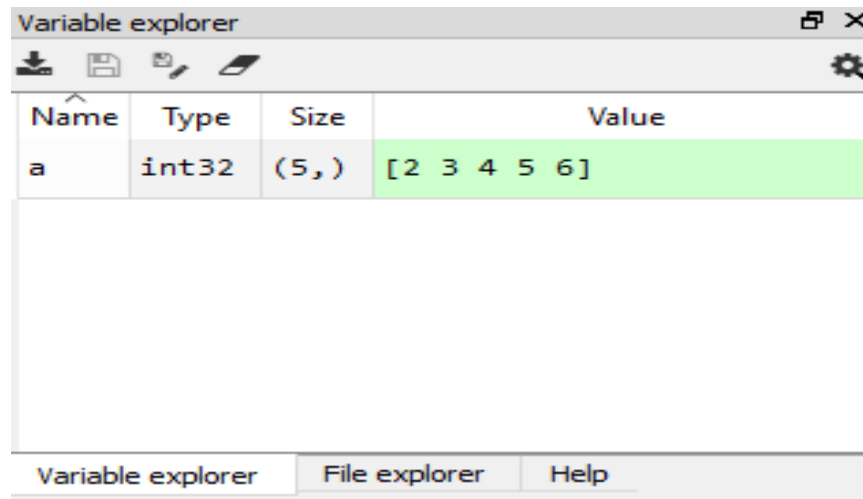


FIG XIII

### File Explorer:

The File Explorer is basically a filesystem and directory browser that allows you to browse, open, and perform other management tasks on the files and folders. You can make use of the context menus functions for operating with them.

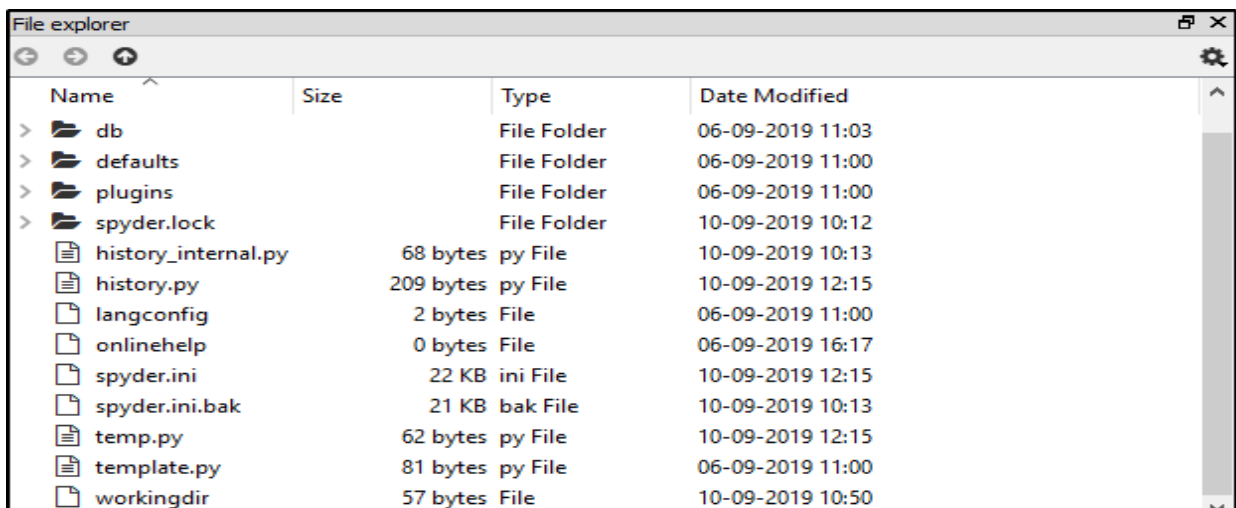


FIG XIV

## Configuring Spyder:

Python Spyder IDE can be configured conveniently using the options present in the preferences menu. You can change anything such as themes, syntax colors, font size, etc. To do this, navigate to the Tools menu and then select the Preferences option. You will see the following window which will allow you to configure Spyder according to your choice:

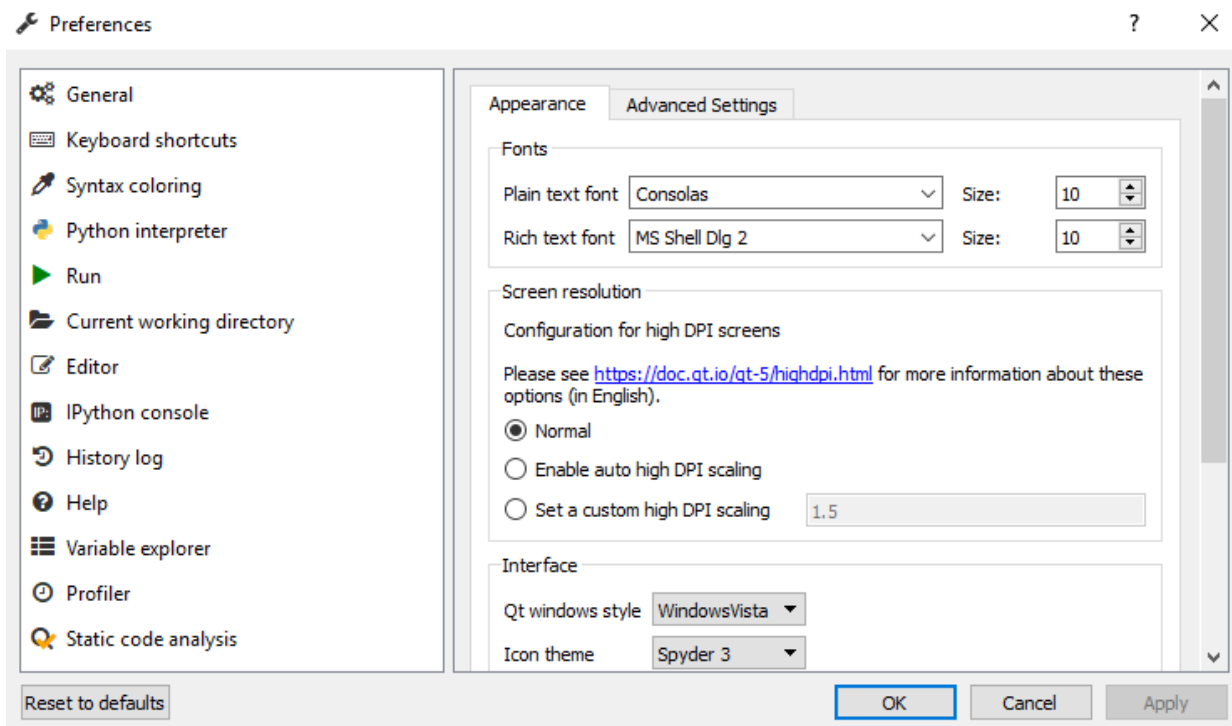


FIG XV

## Help:

The *help* pane allows you to find and display documentation of any object you desire. When you choose the *help* option, you will be able to see the following options:

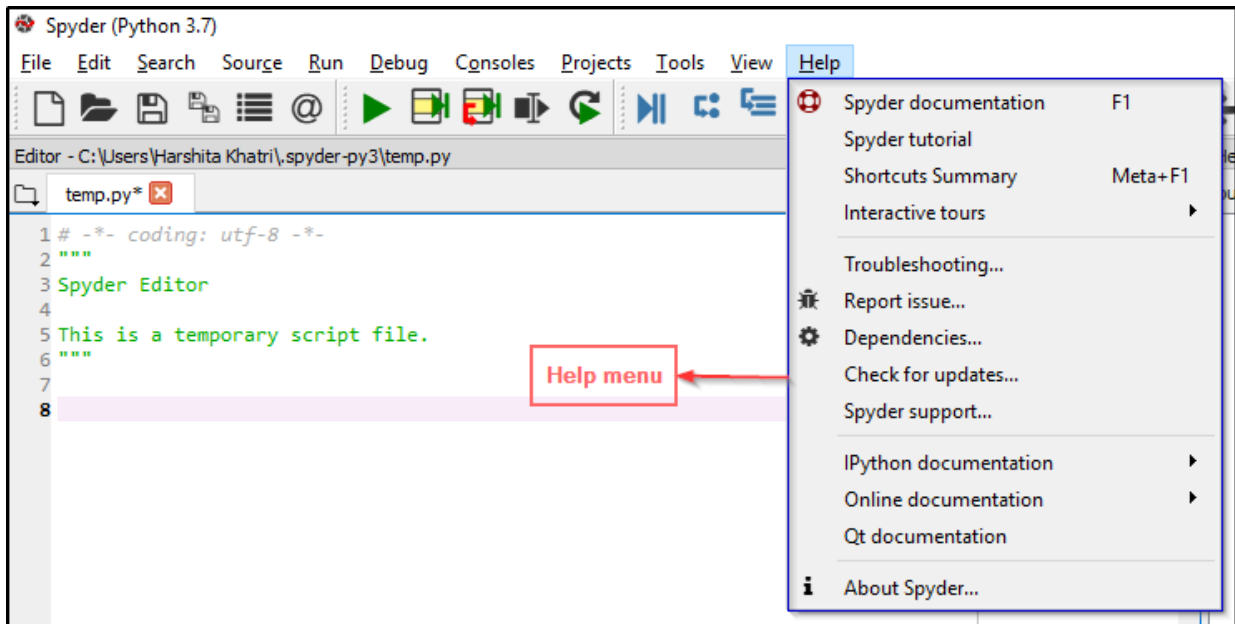


FIG XVI

As you can see, it has many options that will help you solve any issue you encounter while using the Python Spyder IDE.

## HTML Language

HTML is a markup language that web browsers use to interpret and compose text, images, and other material into visual or audible web pages. Default characteristics for every item of HTML markup are defined in the browser, and these characteristics can be altered or enhanced by the web page designer's additional use of CSS. Many of the text elements are found in the 1988 ISO technical report TR 9537 *Techniques for using SGML*, which in turn covers the features of early text formatting languages such as that used by the RUNOFF command developed in the early 1960s for the CTSS (Compatible Time-Sharing System) operating system: these formatting commands were derived from the commands used by typesetters to manually format documents. However, the SGML concept of generalized markup is based on elements (nested annotated ranges with attributes) rather than merely print effects, with also the separation of structure and markup; HTML has been progressively moved in this direction with CSS.

## Features of HTML Language

- Hypertext is what makes travel between the internet possible, especially if you haven't memorized all the URLs on the whole internet. It's essential to guide users around your website and act as a door to know that different rooms exist and can travel between them.
- A web document is a web page. All web pages are possible because you can create the content you want the user to see and then wrap them up in snug code that tells the computer how to format the whole thing.
- HTML supports Polyfill, which is a piece of code that lets you use different technology *natively*. That means that you can mimic a future API and give fallback functionality to older browsers. You can use polyfill libraries according to your needs and do things no one has ever done before.

## Package

The performance of the R language can be improved by user-written packages developed in C, C++, and Java in general.

For specific statistical methods, graphing (ggplots), import / export functions, reporting tools (knitr, sweave), etc. R has a core set of packages. It is provided by installing over 7,801 additional packages including Comprehensive R Archive Network (CRAN), Bio Conductor, Omega Hat, GitHub and more. The Job View page on the CRAN website uses R to offer a variety of packaged jobs (finance, genetics, high-performance computing, machine learning, medical imaging, social sciences, spatial statistics, and more). Is provided. R is also used by the Food and Drug Administration (FDA) to analyse data from medical research. Some R package resources include Crantastic, an open site for evaluating and investigating all CRAN packages, and R-Forge, a

central platform for collective improvement of R packages, R-related software, and projects included. R-Forge also hosts the development of various unreleased beta and CRAN packages. For the analysis of genomic data, the Bioconductor project offers many R packages such as Affymetrix and object-oriented cDNA microarray data processing, and has begun offering tools for researching high-throughput sequencing techniques for next-generation data high throughput sequencing technique.

## **WHAT IS KAGGLE?**

Kaggle is an online community platform for data scientists and machine learning enthusiasts. Kaggle allows users to collaborate with other users, find and publish datasets, use GPU integrated notebooks, and compete with other data scientists to solve data science challenges. The aim of this online platform (founded in 2010 by Anthony Goldbloom and Jeremy Howard and acquired by Google in 2017) is to help professionals and learners reach their goals in their data science journey with the powerful tools and resources it provides. As of today (2021), there are over 8 million registered users on Kaggle.

One of the sub-platforms that made Kaggle such a popular resource is their competitions. In a similar way that Hacker Rank plays that role for software developers and computer engineers, “Kaggle Competitions” has significant importance for data scientists; you can learn more about them in our Kaggle Competition Guide and learn how to analyse a dataset step-by-step in our Kaggle Competition Tutorial. In data science competitions like Kaggle’s or Data Camp’s, companies and organizations share a big amount of challenging data science tasks with generous rewards in which data scientists, from beginners to experienced, compete on their completion. Kaggle also provides the Kaggle Notebook, which, just like Data Camp Workspace, allows you to edit and run your code for data

science tasks on your browser, so your local computer doesn't have to do all the heavy lifting and you don't need to set up a new development environment on your own.

Kaggle provides powerful resources on cloud and allows you to use a maximum of 30 hours of GPU and 20 hours of TPU per week. You can upload your datasets to Kaggle and download others' datasets as well. Additionally, you can check other people's datasets and notebooks and start discussion topics on them. All your activity is scored on the platform and your score increases as you help others and share useful information. Once you start earning points, you will be placed on a live leader board of 8 million Kaggle users.

Kaggle is suitable for different groups of people, from students interested in data science and artificial intelligence to the most experienced data scientists in the world. If you are a beginner, you can take advantage of the courses provided by Kaggle. By joining this platform, you will be able to progress in a community of people of various levels of expertise, and you will have the chance to communicate with many highly experienced data scientists. As you earn Kaggle points and medals, which are proof of your progress, it is quite possible that you may even end up attracting head-hunters and recruiters, and unlock new job opportunities.

Last but not least, when applying for jobs in data science, mentioning your Kaggle experience definitely makes a positive impact. It goes without saying that all these benefits also apply to highly experienced data scientists. No matter how experienced you are, this platform offers continuous learning and improvement possibilities, and, of course, the cash rewards that can come with the competitions are just as interesting.

## **Description of HTML Example**

`<!DOCTYPE>`: It defines the document type or it instruct the browser about the version of HTML.

`<html >`: This tag informs the browser that it is an HTML document. Text between html tag describes the web document. It is a container for all other elements of HTML except `<!DOCTYPE>`

`<head>`: It should be the first element inside the `<html>` element, which contains the metadata (information about the document). It must be closed before the body tag opens.

`<title>`: As its name suggested, it is used to add title of that HTML page which appears at the top of the browser window. It must be placed inside the head tag and should close immediately. (Optional)

`<body>`: Text between body tag describes the body content of the page that is visible to the end user. This tag contains the main content of the HTML document.

`<h1>`: Text between `<h1>` tag describes the first level heading of the webpage.

`<p>`: Text between `<p>` tag describes the paragraph of the webpage.

## **Brief History of HTML**

In the late 1980's, a physicist, Tim Berners-Lee who was a contractor at CERN, proposed a system for CERN researchers. In 1989, he wrote a memo proposing an internet-based hypertext system.

Tim Berners-Lee is known as the father of HTML. The first available description of HTML was a document called "HTML Tags" proposed by Tim in late 1991. The latest version of HTML is HTML5, which we will learn later in this tutorial.

## **HTML Versions**

Since the time HTML was invented, there are lots of HTML versions in market, the brief introduction about the HTML version is given below:

### **HTML 1.0:**

The first version of HTML was 1.0, which was the barebones version of HTML language, and it was released in 1991.

### **HTML 2.0:**

This was the next version which was released in 1995, and it was the standard language version for website design. HTML 2.0 was able to support extra features such as form-based file upload, form elements such as text box, option button, etc.

### **HTML 3.2:**

The HTML 3.2 version was published by W3C in early 1997. This version was capable of creating tables and providing support for extra options for form elements. It can also support a web page with complex mathematical equations. It became an official standard for any browser till January 1997. Today it is practically supported by most of the browsers.

### **HTML 4.01:**

HTML 4.01 version was released in December 1999, and it is a very stable version of HTML language. This version is the current official standard, and it provides added support for stylesheets (CSS) and scripting ability for various multimedia elements.

### **HTML5:**



HTML5 is the newest version of Hypertext Markup language. The first draft of this version was announced in January 2008. There are two major organizations one is W3C (World Wide Web Consortium), and another one is WHATWG (Web Hypertext Application Technology Working Group) which are involved in the development of the HTML 5 version, and still, it is under development.

### **Features of HTML**

- 1) It is a very easy and simple language. It can be easily understood and modified.
- 2) It is very easy to make an effective presentation with HTML because it has a lot of formatting tags.
- 3) It is a markup language, so it provides a flexible way to design web pages along with the text.
- 4) It facilitates programmers to add a link on the web pages (by html anchor tag), so it enhances the interest of browsing the user.
- 5) It is platform-independent because it can be displayed on any platform like Windows, Linux, and Macintosh, etc.
- 6) It facilitates the programmer to add Graphics, Videos, and Sound to the web pages which makes it more attractive and interactive.
- 7) HTML is a case-insensitive language, which means we can use tags either in lower-case or upper-case.

### **Streamlit**

It is an open-source python framework for building web apps for Machine Learning and Data Science. We can instantly develop web apps and deploy them easily using Streamlit. Streamlit allows you to write an app the same way you write a

python code. Streamlit makes it seamless to work on the interactive loop of coding and viewing results in the web app.

## **st.bar\_chart**

Display a bar chart.

This is just syntax-sugar around `st. altair_chart`. The main difference is this command uses the data's own column and indices to figure out the chart's spec. As a result, this is easier to use for many "just plot this" scenarios, while being less customizable.

If `st.bar_chart` does not guess the data specification correctly, try specifying your desired chart using `st. altair_chart`.

---

### Function signature

---

```
st.bar_chart(data=None, *, x=None, y=None, width=0, height=0,  
             use_container_width=True)
```

---

### Parameters

data ( <i>pandas.DataFrame</i> , <i>pandas.Styler</i> , <i>pyarrow.Table</i> , <i>numpy.ndarray</i> , <i>snowflake.snowpark</i> <i>.dataframe.DataFrame</i> , <i>Iterable</i> , or <i>dict</i> )	Data to be plotted. Pyarrow tables are not supported by Streamlit's legacy DataFrame serialization (i.e. with <code>config.dataFrameSerialization = "legacy"</code> ). To use pyarrow tables, please enable pyarrow by changing the config setting, <code>config.dataFrameSerialization = "arrow"</code> .
x ( <i>str</i> or <i>None</i> )	Column name to use for the x-axis. If <i>None</i> , uses the data index for the x-axis. This argument can only be supplied by keyword.
y ( <i>str</i> , <i>sequence of str</i> , or <i>None</i> )	Column name(s) to use for the y-axis. If a sequence of strings, draws several series on the same chart by melting your wide-format table into a long-format table behind the scenes. If <i>None</i> , draws the data of all remaining columns as data series. This argument can only be supplied by keyword.
width ( <i>int</i> )	The chart width in pixels. If 0, selects the width automatically. This argument can only be supplied by keyword.
height ( <i>int</i> )	The chart height in pixels. If 0, selects the height automatically. This argument can only be supplied by keyword.
use_container_width ( <i>bool</i> )	If <i>True</i> , set the chart width to the column width. This takes precedence over the width argument. This argument can only be supplied by keyword.

TABLE I

## **st.line\_chart**

Display a line chart.

This is syntax-sugar around `st.altair_chart`. The main difference is this command uses the data's own column and indices to figure out the chart's spec. As a result, this is easier to use for many "just plot this" scenarios, while being less customizable.

If `st.line_chart` does not guess the data specification correctly, try specifying your desired chart using `st.altair_chart`.

## **st.area\_chart**

Display an area chart.

This is just syntax-sugar around `st.altair_chart`. The main difference is this command uses the data's own column and indices to figure out the chart's spec. As a result, this is easier to use for many "just plot this" scenarios, while being less customizable.

If `st.area_chart` does not guess the data specification correctly, try specifying your desired chart using `st.altair_chart`.

## st.radio

Display a radio button widget.

---

### Function signature

---

```
st.radio(label, options, index=0, format_func=special_internal_function, key=None,
        help=None, on_change=None, args=None, kwargs=None, *, disabled=False,
        horizontal=False, label_visibility="visible")
```

---

### Parameters

label ( <i>str</i> )	A short label explaining to the user what this radio group is for. For accessibility reasons, you should never set an empty label ( <code>label=""</code> ) but hide it with <code>label_visibility</code> if needed. In the future, we may disallow empty labels by raising an exception.
options ( <i>Sequence, numpy.ndarray, pandas.Series, pandas.DataFrame, or pandas.Index</i> )	Labels for the radio options. This will be cast to <code>str</code> internally by default. For <code>pandas.DataFrame</code> , the first column is selected.
index ( <i>int</i> )	The index of the preselected option on first render.
format_func ( <i>function</i> )	Function to modify the display of radio options. It receives the raw option as an argument and should output the label to be shown for that option. This has no impact on the return value of the radio.

key ( <i>str or int</i> )	An optional string or integer to use as the unique key for the widget. If this is omitted, a key will be generated for the widget based on its content. Multiple widgets of the same type may not share the same key.
help ( <i>str</i> )	An optional tooltip that gets displayed next to the radio.
on_change ( <i>callable</i> )	An optional callback invoked when this radio's value changes.
args ( <i>tuple</i> )	An optional tuple of args to pass to the callback.
kwargs ( <i>dict</i> )	An optional dict of kwargs to pass to the callback.
disabled ( <i>bool</i> )	An optional boolean, which disables the radio button if set to True. The default is False. This argument can only be supplied by keyword.
horizontal ( <i>bool</i> )	An optional boolean, which orients the radio group horizontally. The default is false (vertical buttons). This argument can only be supplied by keyword.
label_visibility ( <i>"visible"</i> or <i>"hidden"</i> or <i>"collapsed"</i> )	The visibility of the label. If "hidden", the label doesn't show but there is still empty space for it above the widget (equivalent to label=""). If "collapsed", both the label and the space are removed. Default is "visible". This argument can only be supplied by keyword.

---

TABLE II
----------

### **st.select\_slider**

Display a slider widget to select items from a list.

This also allows you to render a range slider by passing a two-element tuple or list as the value.

The difference between `st.select_slider` and `st.slider` is that `select_slider` accepts any data type and takes an iterable set of options, while `slider` only accepts numeric or date/time data and takes a range as input.

### **st.text\_area**

Display a multi-line text input widget.

---

Function signature

---

```
st.text_area(label, value="", height=None, max_chars=None, key=None, help=None,
             on_change=None, args=None, kwargs=None, *, placeholder=None, disabled=False,
             label_visibility="visible")
```

---

Parameters

label ( <i>str</i> )	A short label explaining to the user what this input is for. For accessibility reasons, you should never set an empty label (label="") but hide it with label_visibility if needed. In the future, we may disallow empty labels by raising an exception.
value ( <i>object</i> )	The text value of this widget when it first renders. This will be cast to str internally.
height ( <i>int or None</i> )	Desired height of the UI element expressed in pixels. If None, a default height is used.
max_chars ( <i>int or None</i> )	Maximum number of characters allowed in text area.
key ( <i>str or int</i> )	An optional string or integer to use as the unique key for the widget. If this is omitted, a key will be generated for the widget based on its content. Multiple widgets of the same type may not share the same key.
help ( <i>str</i> )	An optional tooltip that gets displayed next to the textarea.
on_change ( <i>callable</i> )	An optional callback invoked when this text_area's value changes.
args ( <i>tuple</i> )	An optional tuple of args to pass to the callback.
kwargs ( <i>dict</i> )	An optional dict of kwargs to pass to the callback.



placeholder ( <i>str</i> or <i>None</i> )	An optional string displayed when the text area is empty. If <i>None</i> , no text is displayed. This argument can only be supplied by keyword.
disabled ( <i>bool</i> )	An optional boolean, which disables the text area if set to <i>True</i> . The default is <i>False</i> . This argument can only be supplied by keyword.
label_visibility ( <i>"visible"</i> or <i>"hidden"</i> or <i>"collapsed"</i> )	The visibility of the label. If <i>"hidden"</i> , the label doesn't show but there is still empty space for it above the widget (equivalent to <i>label=""</i> ). If <i>"collapsed"</i> , both the label and the space are removed. Default is <i>"visible"</i> . This argument can only be supplied by keyword.

TABLE III

## st.title

Display text in title formatting.

Each document should have a single `st.title()`, although this is not enforced.

### Function signature

`st.title(body, anchor=None)`

### Parameters

body ( <i>str</i> )	The text to display.
---------------------	----------------------

anchor ( <i>str</i> )	The anchor name of the header that can be accessed with #anchor in the URL. If omitted, it generates an anchor using the body. TABLE IV
-----------------------	--

## **st.header**

Function signature	
st.header(body, anchor=None)	
Parameters	
body ( <i>str</i> )	The text to display.
anchor ( <i>str</i> )	The anchor name of the header that can be accessed with #anchor in the URL. If omitted, it generates an anchor using the body. TABLE V

## **4.2 ALGORITHM**

### **4.2.1 LOGISTIC REGRESSION**

Logistic regression is a supervised learning classification algorithm used to predict

the probability of a target variable. The nature of target or dependent variable is dichotomous, which means there would be only two possible classes. o Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables. o Logistic regression predicts the output of a categorical dependent variable. Therefore, the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, True or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1. o Logistic Regression is much similar to the Linear Regression except that how they are used. Linear Regression is used for solving Regression problems, whereas Logistic regression is used for solving the classification problems. o In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1). o the curve from the logistic function indicates the likelihood of something such as whether the cells are cancerous or not, a mouse is obese or not based on its weight, etc. o Logistic Regression is a significant machine learning algorithm because it has the ability to provide probabilities and classify new data using continuous and discrete datasets. o Logistic Regression can be used to classify the observations using different types of data and can easily determine the most effective variables used for the classification. The below image shows the logistic function: Fig(5). Logistic Regression Curve ` 36 Types of Logistic Regression

Generally, logistic regression means binary logistic regression having binary target variables, but there can be two more categories of target variables that can be predicted by it. Based on the number of categories, Logistic regression can be divided into following types –

1. Binary or Binomial In such a kind of classification, a dependent variable will have only two possible types either 1 and 0. For example, these variables may represent success or failure, yes or no, win or loss etc.

2. Multinomial In such a kind of classification, dependent variables can have 3 or more possible unordered types or the types having no quantitative significance. For example, these variables may represent “Type A” or “Type B” or “Type C”.

3. Ordinal In such a kind of classification, dependent variables can have 3 or more possible ordered types or the types having a quantitative significance. For example, these variables may represent “poor” or “good”, “very good”, “Excellent” and each category can have the scores like 0,1,2,3.

Logistic Regression Assumptions Before diving into the implementation of logistic regression, we must be aware of the following assumptions about the same –

- In case of binary logistic regression, the target variables must be binary always and the desired outcome is represented by the factor level .
  - There should not be any multi-collinearity in the model, which means the independent variables must be independent of each other.
  - We must include meaningful variables in our model.
  - We should choose a large sample size for logistic regression.
- Regression Models
- Binary Logistic Regression Model – The simplest form of logistic regression is binary or binomial logistic regression in which the target or dependent variable can have only 2 possible types either 1 or 0.
  - Multinomial Logistic Regression Model – Another useful form of logistic regression is multinomial logistic regression in which the target or dependent variable can have 3 or more possible unordered types i.e., the types having no quantitative significance.
- Logistic Regression Predicted Probabilities (Technical Interlude)
- Logistic regression models the probability of the default class (e.g., the first class). For example, if we are modelling people’s sex as male or female from their height, then the first class could be male and the logistic regression model could be written as the probability of male given a person’s height, or more formally:  $P(\text{sex}=\text{male} \mid \text{height})$  Written another way, we are modelling the probability that an input (X) belongs to the default class (Y=1), we can write this

formally as:  $P(X) = P(Y=1|X)$  We're predicting probabilities. I thought logistic regression was a classification algorithm. Note that the probability prediction must be transformed into a binary value (0 or 1) in order to actually make a probability prediction. More on this later when we talk about making predictions. Logistic regression is a linear method, but the predictions are transformed using the logistic function. continuing on from above, the model can be stated as:

$$p(X) = e^{(b_0 + b_1 * X)} / (1 + e^{(b_0 + b_1 * X)})$$

$$\ln(p(X) / 1 - p(X)) = b_0 + b_1 * X$$

This is useful because we can see that the calculation of the output on the right is linear again (just like linear regression), and the input on the left is a log of the probability of the default class. This ratio on the left is called the odds of the default class (it's historical that we use odds, for example, odds are used in horse racing rather than probabilities). Odds are calculated as a ratio of the probability of the event divided by the probability of not the event, e.g.,  $0.8/(1-0.8)$  which has the odds of 4. So, we could instead write:  $\ln(\text{odds}) = b_0 + b_1 * X$  Because the odds are log transformed, we call this left-hand side the log-odds or the probit. It is possible to use other types of functions for the transform (which is out of scope, but as such it is common to refer to the transform that relates the linear regression equation to the probabilities as the link function, e.g., the probit link function. We can move the exponent back to the right and write it as:  $\text{odds} = e^{(b_0 + b_1 * X)}$  All of this helps us understand that indeed the model is still a linear combination of the inputs, but that this linear combination relates to the log-odds of the default class.

### **Naïve Bayes Classifier (NB):**

The naive Bayes classifier is the simplest and most commonly used classifier. The naive Bayes classification model calculates the posterior probabilities of a class based on the distribution of words in the document. The model works

with feature extractions from the BOW that ignore the position of words in the document. Use Bayes' theorem to predict the probability that a particular feature set will belong to a particular label.

### **Training:**

In this phase, we need to generate training data (words that can occur in positive / negative train data files).

Calculates the label for each label and the word in the dictionary and saves the result.

You now have a probability that corresponds to each defined word on the label.

Test

- Find the mood of a specific test data file Create a feature set (x) of the test data file.

Find the test sentence for each document

Decision1 =  $\log P(x| \text{label} = \text{pos}) + \log P(\text{label} = \text{pos})$

Calculate in the same way as

Decision2 =  $\log P(x| \text{label} = \text{neg}) + \log P(\text{label} = \text{neg})$

Compare decision I to calculate negative or positive emotions.

### **Natural Language Processing**

Natural Language Processing (NLP) is a field of computer science, artificial intelligence, and linguistics that deals with the interaction of computers with human (natural) language.

This approach uses the public Opinion Lexicon library, which provides emotional polarity values for each term displayed in the document. In this lexicon, each term t that occurs in the opinion lexicon is associated with three

numerical scores  $obj(t)$ ,  $pos(t)$ , and  $neg(t)$  that represent the purpose of the term, positive and negative polarities, respectively increase. These three scores are calculated by combining the results produced by eight cubic classifiers. Opinion Lexicon is a large English vocabulary database. Nouns, verbs, adjectives, and adverbs are each grouped into a set of cognitive synonyms that represent a particular concept. You can also download the opinion dictionary for free. The structure of Opinion Lexicon makes it a useful tool for computational linguistics and natural language processing. Group words based on meaning. This is just a set of one or more synonyms. This approach uses semantics to understand the language.

An important NLP task to help extract emotions from sentences:

- Extract some of the sentences that reflect the mood.
- Understand the structure of the sentence.

Various tools to help with text processing Data Basically, positive and negative ratings are taken from the Opinion Lexicon according to their panoramic speech tags, and then by counting the total positive and negative ratings, which class (That is, it determines emotional polarity based on whether the positive or negative) received the highest rating.

#### **4.5 Application of Personality Prediction:**

Applications Using Website Reviews:

Nowadays the internet has a huge collection of reviews and comments on almost anything. This includes product reviews, policy issue reviews, service reviews, and more. Therefore, there is a need for a personality analysis

system capable of drawing emotions about a particular product or service. This will help us automate the delivery of reviews or ratings for products, articles, and more. Send. This will meet the needs of users and vendors.

#### Applications as a Sub-component Technology:

It has been observed that nowadays people tend to check the reviews of products available online before buying them. And for many companies, online reviews determine the success or failure of their products.

#### Applications in Business Intelligence:

As such, psychoanalysis plays an important role in business. Businesses also want to harness personality from online reviews to improve their products and, therefore, their reputation and contribute to customer satisfaction.

#### Applications across Domains:

Recent research in sociology and other fields such as medicine and sports also benefit from personality analysis showing trends in human emotions, especially on social media.

#### Applications in smart home:

Smart home is said to be the technology of the future. In the future, the whole house will be networked and people can control any part of the house with a tablet. Recently, a lot of research has been done on the Internet of Things (IOT). personality analysis will also find its place in the IoT.

For example, depending on the personality of the occupants, the house can change its surroundings to create a soothing and peaceful environment.



personality analysis can also be used in trend forecasting. By monitoring audience personality traits, important data regarding sales trends and customer satisfaction can be extracted.

## Chapter 5

### Flow of program

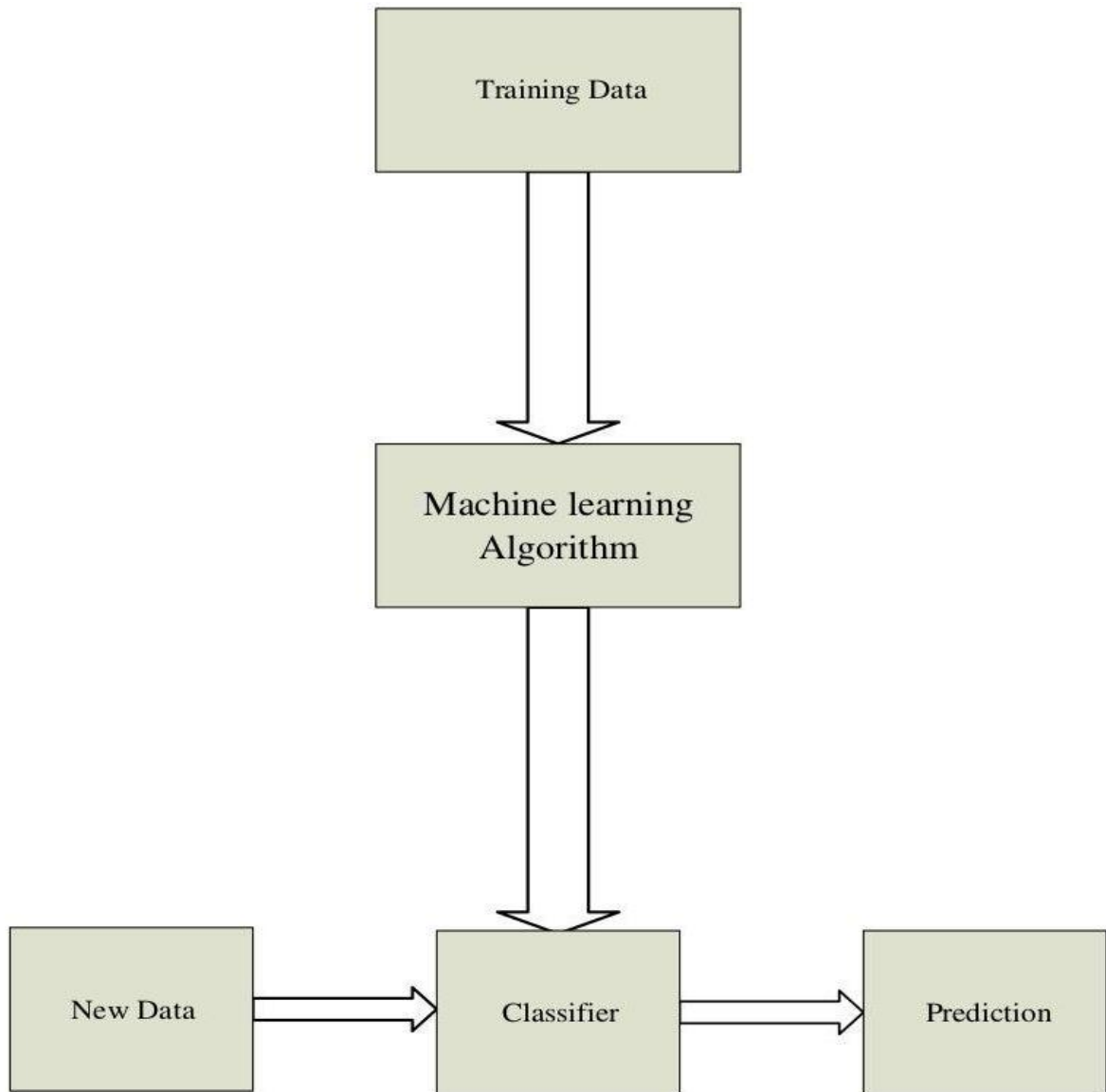


Fig XVII

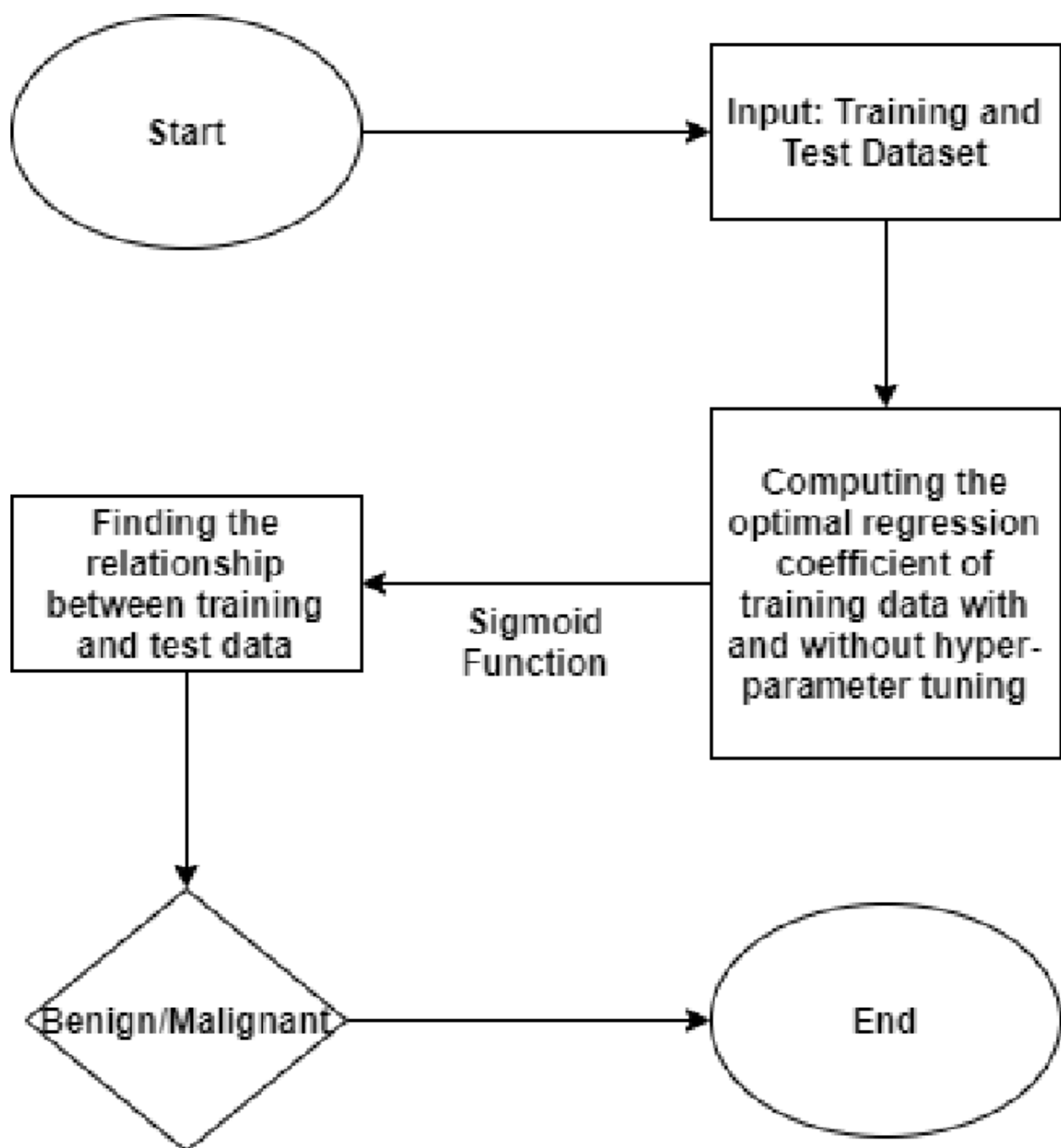


FIG XVIII

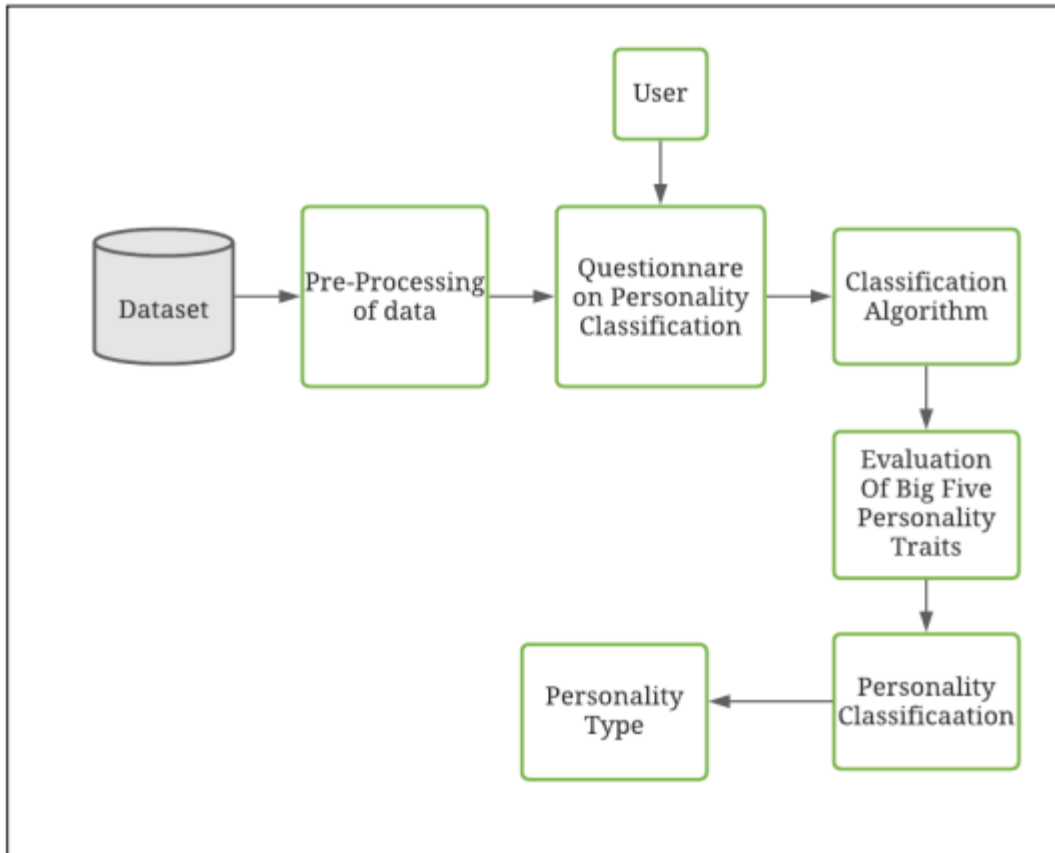


FIG XIX

# **Chapter 6**

## **System Testing**

Testing is the process of evaluating a system or its components to determine whether it meets specified requirements. This activity leads to actual, expected results and the difference between them, i.e., tests that run a system to identify deficiencies, flaws or missing requirements. with actual wishes or requirements.

### **Testing Strategies**

To ensure that the system is error free, the different levels of testing strategies applied at different stages of software development are

#### **6.1. Unit testing:**

Unit testing is done to test the modules against the detailed design. The input to the process is usually modules compiled from the coding process. Each module is assembled into a larger unit during unit testing. Tests were performed on each design and coding phase of the project. We perform module interface testing to ensure proper flow of information into and out of the program unit during testing. We ensure that temporarily stored data maintains its integrity during algorithm execution by examining local data structures. Finally, all error handling paths are also checked.

## **6.2. Integration Testing:**

Testing the combined parts of an application to determine if they work properly with each other is an integration test. This testing can be done using two different methods

### **6.2.1 Top-down Integration testing:**

In Top-Down integration testing, the top-level modules are tested first, and then the level modules lower is tested gradually.

### **6.2.2 Bottom-up Integration testing**

Testing can be performed starting from smallest and lowest level modules and proceeding one at a time. When bottom level modules are tested attention turns to those on the next level that use the lower-level ones they are tested individually and then linked with the previously examined lower-level modules. In a comprehensive software development environment, bottom-up testing is usually done first, followed by top-down testing.

## **6.3. System Testing**

We usually perform system testing to find errors resulting from unanticipated interaction between the subsystem and system components. Software must be tested to detect and rectify all possible errors once the source code is generated before delivering it to the customers. For finding errors, a series of test cases must be developed which ultimately uncover all the possibly existing errors. Different

software techniques can be used for this process. These techniques provide systematic guidance for design tests that Exercise the Internal logic of the software components, Exercise the input and output domains of a program to uncover errors in program function, behaviour and performance. We test the software using two methods: White Box testing: Internal program logic is exercised using this test case design techniques. Black Box testing: Software requirements are exercised using this test case design techniques. Both techniques help in finding the maximum number of errors with minimal effort and time.

#### **6.4 Performance Testing:**

This is done to test the run-time performance of the software in the context of an integrated system. These tests are run throughout the testing process. For example, the performance of individual modules is accessed during white-box testing in unit tests.

#### **6.5 Acceptance Testing**

The main purpose of this test is to determine if the application meets the intended specifications and customer requirements. This test uses two different methods.

##### **6.5.1 Alpha Test**

This test is the first phase of the test and is conducted between teams. Unit tests, integration tests, and system tests are collectively called alpha tests. In this phase, the application checks for the following

- Spelling Mistakes.

- Broken link.

The application is tested on the lowest spec computer to test for load time and latency issues.

### **6.5.2 Beta Testing**

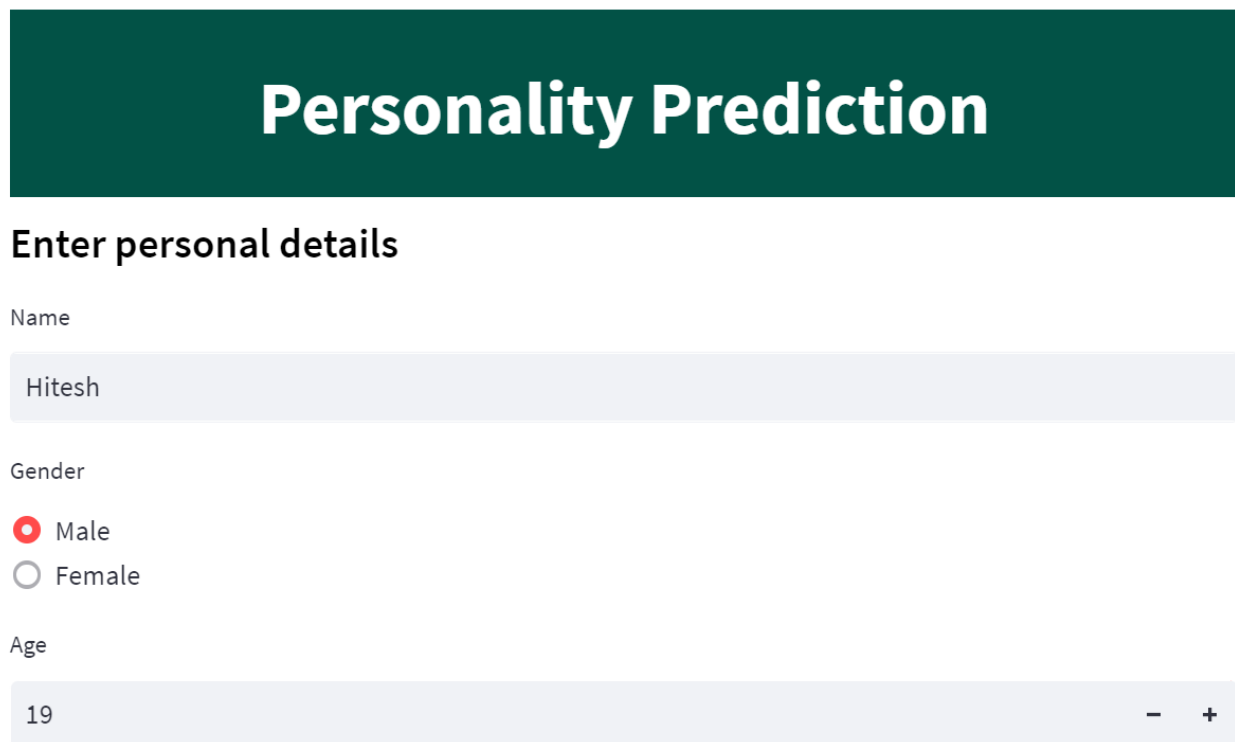
In the Beta Testing, a sample audience tests the application and provides feedback to the project team. The project team receives feedback and can fix the problem before releasing the software to real use



## CHAPTER 7

### RESULT DISCUSSION

Finally, we conclude that our classification approach provides an improvement in accuracy by using even the simplest features and a small amount of data set. However, there are still a number of things we would like to consider as future work which we mention in the next section.



The image shows a web interface titled "Personality Prediction" in a dark green header. Below the header, the section "Enter personal details" is displayed. It contains three input fields: "Name" with the text "Hitesh", "Gender" with radio buttons for "Male" (selected) and "Female", and "Age" with a numeric input field showing "19" and minus/plus buttons.

## Personality Prediction

### Enter personal details

Name

Hitesh

Gender

☒ Male

☐ Female

Age

19 - +

FIG XX

This interface is the first phase of getting started with the process of predicting personality. Here, the user will be asked to share their personal details like name, age and gender. These details will further help as data is fed to the machine learning model running at the back of the interface for analysis.

## Instructions

i). It test is divided in 6 sections.

ii). To answer the following question select the option.

1=Strongly Disagree

2=Disagree

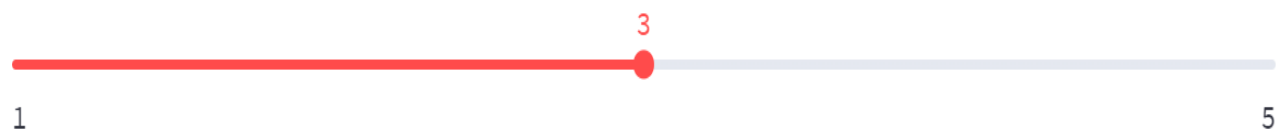
3=Netural

4=Agree

5=Strongly Agree

## Section A

Ques 1. I have excellent ideas



Ques 2. I have a vivid imagination.

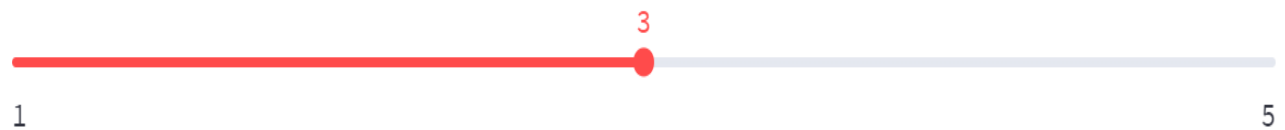


FIG XXI

This interface will display instructions to use the prototype. So, the user's questionnaire is divided into six sections for personality analysis. Users will get five choices that are strongly disagree, disagree, neutral, agree, and strongly agree to answer questions. Section A :This section is designed to judge your 'openness' graph.

## Section B

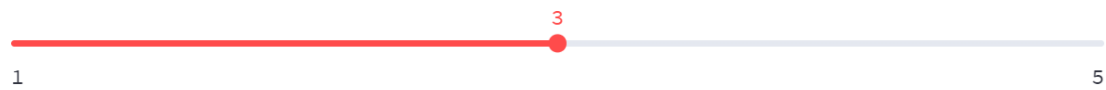
Ques 5. I am relaxed most of the time



Ques 6. I not worry about things.



Ques 7. I did not get stressed out easily



Ques 8. I not get upset easily..

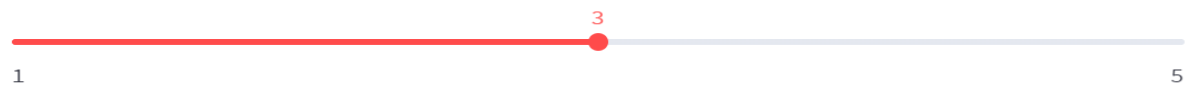


FIG XXII

This interface displays section B questionnaires where users will answer their ‘neuroticism’ personality.

## Section C

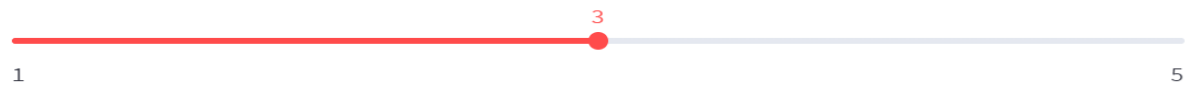
Ques 9. I am always prepared.



Ques 10. I pay attention to details.



Ques 11. I follow a schedule



Ques 12. I am exacting in my work..

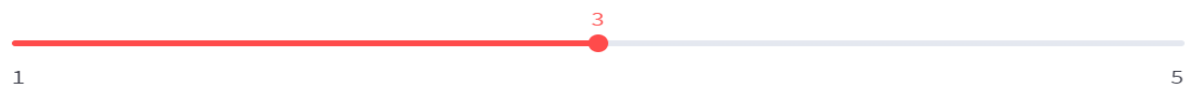
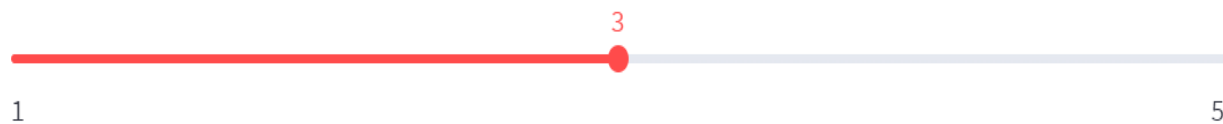


FIG XXIII

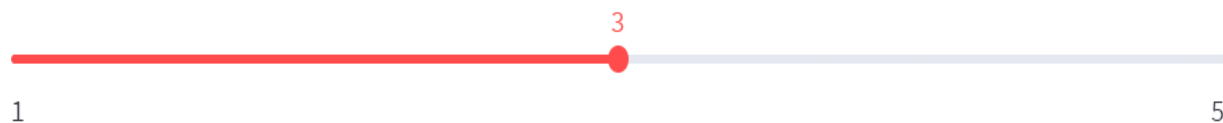
This section will pose questions related to ‘Conscientiousness’ to the user.

## Section D

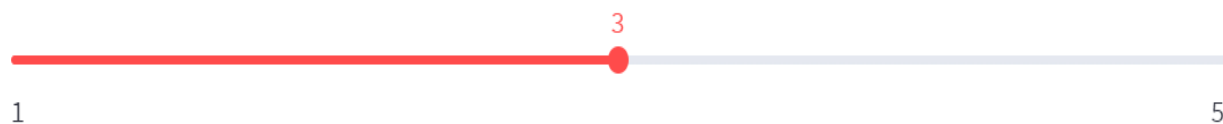
Ques 13. I am interested in people.



Ques 14. I sympathize with others feelings



Ques 15. I did not insult people



Ques 16. I am interested in other people problems

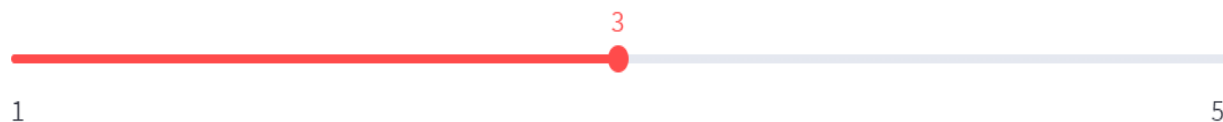


FIG XXIV

This interface will ask questions related to ‘Agreeableness’ to the user.

Section E



FIG XXV

This section will be posing questions judging your ‘Extraversion’ trait.

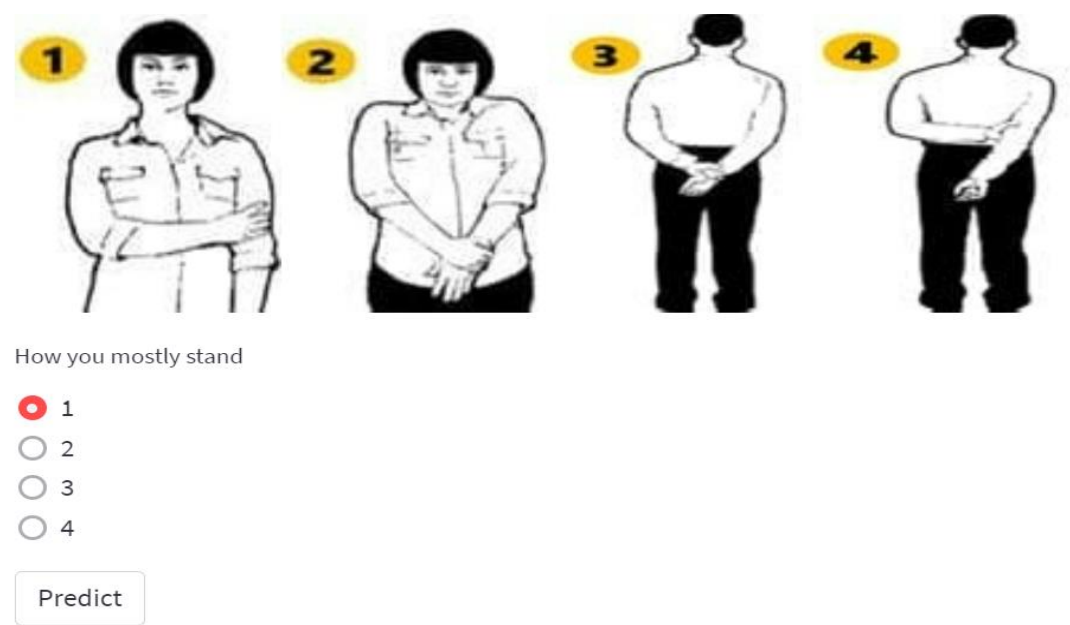


FIG XXVI

This is an image-based question that is posed to users to get their behavioural concise.

Your personality types is ['serious']

When you take up something, you like to analyze it and make your conclusions so that everything is as per your liking. Overall, you have a unique personality. You can be analytical and emotional as per the situation.

You are a defensive person. You trying to comfort yourself, especially in a room full of a crowd.

Your scores

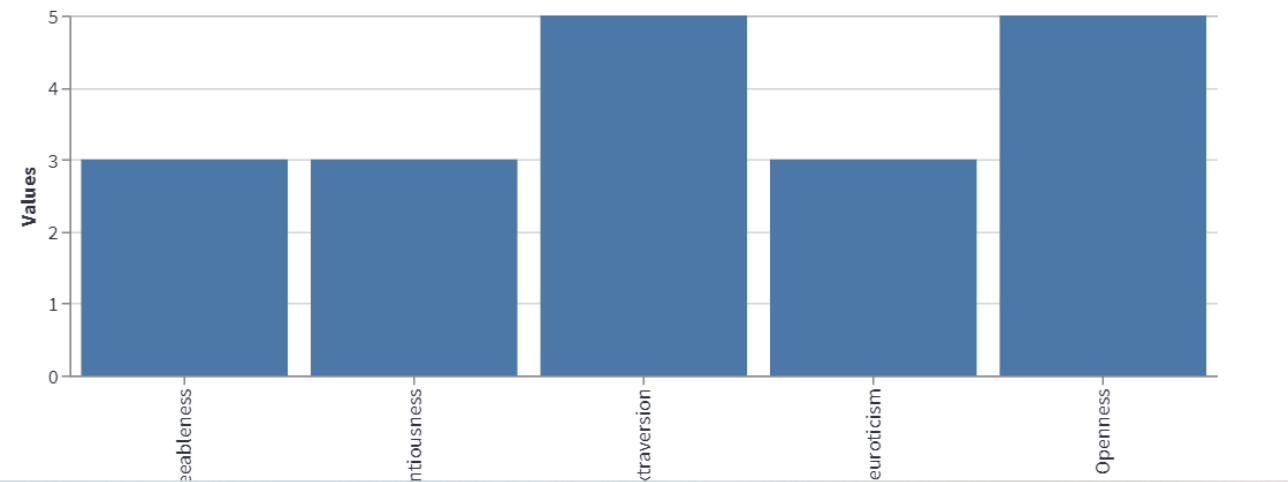


FIG XXVII

This image shows a detailed study of your personality. Here the personality type ‘serious’

Openness graph

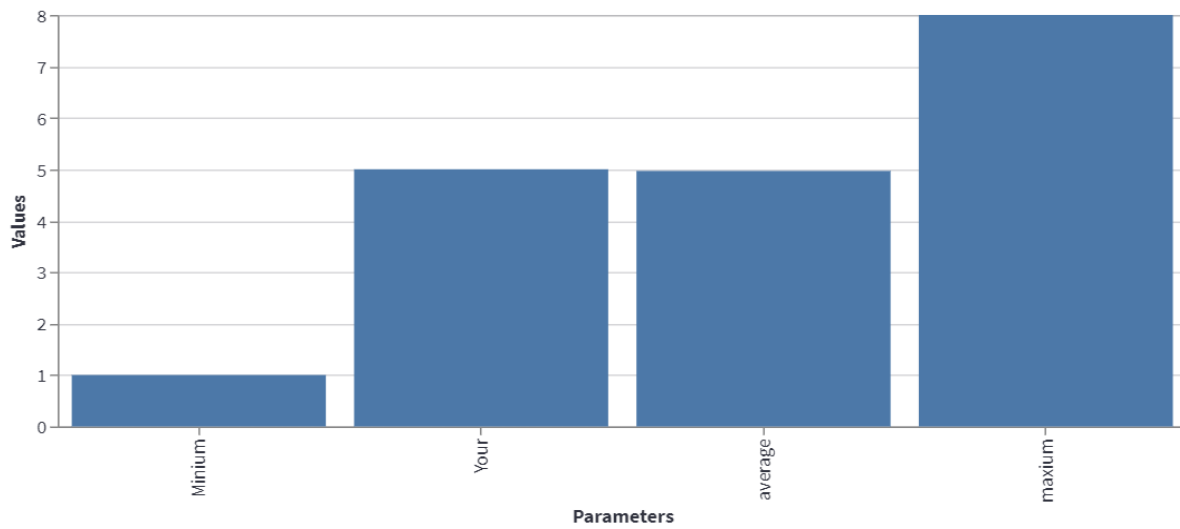


FIG XXVIII

This graph shows the detailed graph on the openness trait to let you have deeper insight.

Extraversion graph

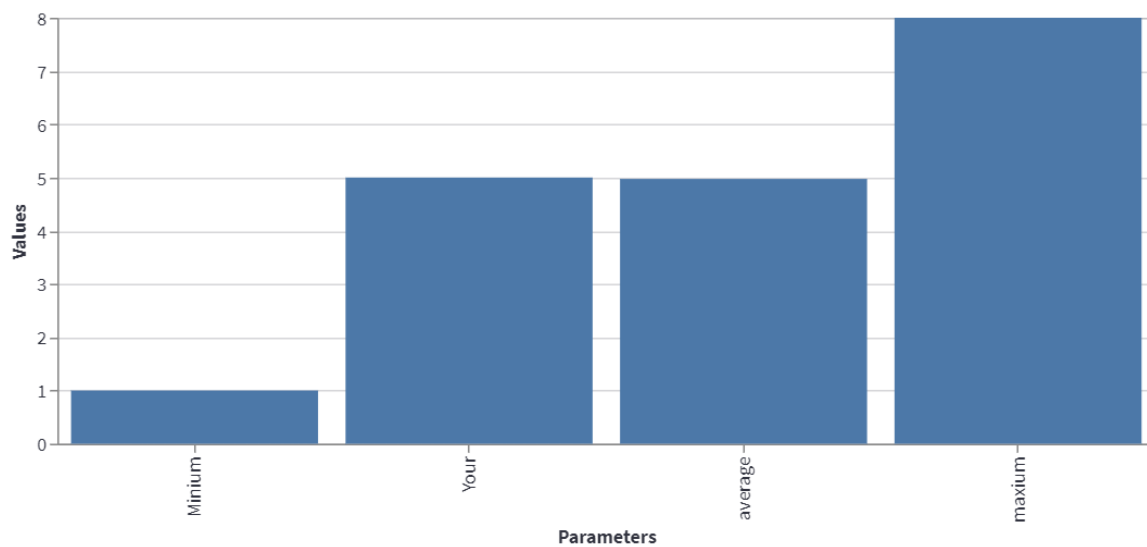


FIG XXIX

This graph gives user's a deep insight into their extraversion trait.

## **Chapter 8**

### **CONCLUSION & FUTURE WORK**

#### **8.1 Conclusion**

You have a certain way and feel a certain way, but often it's hard to understand why, especially when other people aren't thinking or feeling the same way. Getting insights into your personality helps in career growth by moulding yourself accordingly.

Knowing your personality helps you better answer the dilemma of how's and why's about yourself. This project aims to categorize an individual's personality under the Big 5 personality based on analysis of the user's answers using machine learning algorithms with the accuracy of 84.23% . An interactive interface is developed to take user's input for the analysis to give valuable predictions. This product not only lets you know yourself better but also helps in an individual's career growth.

#### **8.2 Future Work**

We believe our results would build a brand-new reference for future comparative studies. We are going to explore future work diversely, like alternative network structures to extract different options. We expect to refine our work by adding new features like not only letting people know their weak points but also giving them an idea on how to improve themselves to excel in their respective field.

Still requires a fair amount of work. Here, we provide a ray of light for the direction of future research.

- Future Interpretation: The proposed approach is currently unable to interpret future. Predicting the future of an individual based on their personality is a unique concept to be ever proposed .Our future goal is to run through algorithms to come up with future predictions based on the individual's personality.

This limitation can be overcome by a thorough study of the basics of knowledge-



driven personality analysis.

The main goal of this approach is to predict the future of an individual based on their personality.

- Multilingualism: Due to the lack of multilingual lexicons, it is currently not possible to develop multilingual personality analysers. Further investigation can be done to make the classifier language independent. The authors proposed a sentiment analysis system using a support vector machine. A similar approach can be applied to the system to make it language independent.
- Analysis of facial emotions to predict personality based on facial impressions.
- Judgment of neutrality
- Possible improvements can be made to data collection and analysis methods
- Possible improvements such as sophisticated data and more accurate algorithms you can do future research.

## References

- [1] Asra S., Shubhangi D. C., Personality Trait Identification Using Unconstrained Cursive and Mood Invariant Handwritten Text. *International Journal of Education and Management Engineering*, 2015, 5(5), 20
- [2] Kedar S., Nair V., Kulkarni S., Personality identification through handwriting analysis: a review. *Int. J. Adv. Res. Comput. Sci. Softw. Eng*, 2015, 5(1)
- [3] Ilimini K., Fernando T. G. I., Persons' personality traits recognition using machine learning algorithms and image processing techniques. *Advances in Computer Science: an International Journal*, 2016, 5(1), 40-44
- [4] Liu L., Preotiuc-Pietro D., Samani Z. R., Moghaddam M. E., Ungar L. H., Analyzing Personality through Social Media Profile Picture Choice. In *Tenth international AAAI conference on web and social media (ICWSM)*, 2016, 211-220.
- [5] Sagadevan S., Malim N. H. A. H., Husin M. H., Sentiment Valences for Automatic Personality Detection of Online Social Networks Users Using Three Factor Model. *Procedia Computer Science*, 2015, 72, 201-208
- [7] jagran
- [8] Abood N. Big five traits: a critical review. *Gadjah Mada Int J Business*. 2019
- [9] Acheampong FA, Nunoo-Mensah H, Chen W. Transformer models for text-based emotion detection: a review of BERT-based approaches. *Artif Intell Rev*. 2021.

## Annexure A

```
import numpy as np
import pandas as pd
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
import warnings
import pickle
import altair as alt
warnings.filterwarnings("ignore")

data = pd.read_csv('C:\\Users\\Hp\\OneDrive\\Projects\\personality\\train_copy.csv')
data = np.array(data)

X = data[:, 1:-1]
y = data[:, -1]
# print(X,y)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=0)
log_reg = LogisticRegression()
log_reg.fit(X_train, y_train)

pickle.dump(log_reg,open('C:\\Users\\Hp\\OneDrive\\Projects\\personality\\model.pkl', 'wb'))
model =
pickle.load(open('C:\\Users\\Hp\\OneDrive\\Projects\\personality\\model.pkl', 'rb'))
```

## Annexure B

```
import streamlit as st
import pickle
import numpy as np
import time
from PIL import Image
import sklearn.preprocessing as pre
import plotly.figure_factory as ff

def predict_forest(age,ope,neu,con, agr,ext):
    input=np.array([[age,ope,neu,con, agr,ext]]).astype(np.float64)
    prediction=model.predict(input)
    #pred='{0:.{1}f}'.format(prediction[0][0], 2)
    return prediction

def main():

    html_temp = """
    <div style="background-color:#025246 ;padding:10px">
    <h1 style="color:white;text-align:center;">Personality Prediction </h1>
    </div>
    """

    st.markdown(html_temp, unsafe_allow_html=True)

    html_temp1 = """
    <h4 style="color:black;">Enter personal details </h4>
```



```
st.markdown(html_temp4, unsafe_allow_html=True)
```

```
a=[0,0,0,0,0]
```

```
a[0] = st.slider('Ques 1. I have excellent ideas', 1, 5, 3)
```

```
a[1] = st.slider('Ques 2. I have a vivid imagination.', 1, 5, 3)
```

```
a[2] = st.slider('Ques 3. I have a rich vocabulary', 1, 5, 3)
```

```
a[3] = st.slider('Ques 4. Spend time reflecting on things.', 1, 5, 3)
```

```
html_tempB = """
```

```
<h4 align="left">Section B</h4>
```

```
"""
```

```
st.markdown(html_tempB, unsafe_allow_html=True)
```

```
b=[0,0,0,0,0]
```

```
b[0] = st.slider('Ques 5. I am relaxed most of the time', 1, 5, 3)
```

```
b[2] = st.slider('Ques 6. I not worry about things.', 1, 5, 3)
```

```
b[3] = st.slider('Ques 7. I did not get stressed out easily', 1, 5, 3)
```

```
b[4] = st.slider('Ques 8. I not get upset easily..', 1, 5, 3)
```

```
html_tempC = """
```

```
<h4 align="left">Section C</h4>
```

```
"""
```

```
st.markdown(html_tempC, unsafe_allow_html=True)
```

```
c=[0,0,0,0,0]
```

```
c[0] = st.slider('Ques 9. I am always prepared.', 1, 5, 3)
```

```
c[1] = st.slider('Ques 10. I pay attention to details.', 1, 5, 3)
```

```
c[3] = st.slider('Ques 11. I follow a schedule', 1, 5, 3)
```

```
c[4] = st.slider('Ques 12. I am exacting in my work..', 1, 5, 3)
```

```
html_tempD = ""
<h4 align="left">Section D<h4>
""

st.markdown(html_tempD, unsafe_allow_html=True)
d=[0,0,0,0,0]
d[0] = st.slider('Ques 13. I am interested in people.', 1, 5, 3)
d[1] = st.slider('Ques 14. I sympathize with others feelings', 1, 5, 3)
d[3] = st.slider('Ques 15. I did not insult people', 1, 5, 3)
d[4] = st.slider('Ques 16. I am interested in other people problems', 1, 5, 3)
```

```
html_tempE = ""
<h4 align="left">Section E<h4>
""

st.markdown(html_tempE, unsafe_allow_html=True)
e=[0,0,0,0,0]
e[0] = st.slider('Ques 17. I am the life of the party.', 1, 5, 3)
e[1] = st.slider('Ques 18. I feel comfortable around people', 1, 5, 3)
e[2] = st.slider('Ques 19. I start conversations', 1, 5, 3)
e[3] = st.slider('Ques 20. I do talk a lot.', 1, 5, 3)
```

### Other tests

```
html_tempF = ""
<h4 align="left">Section F<h4>
""

st.markdown(html_tempF, unsafe_allow_html=True)
pen1 = Image.open(r'C:\Users\Hp\OneDrive\Projects\personality\Image\pen1.jpg')
pen2 = Image.open(r'C:\Users\Hp\OneDrive\Projects\personality\Image\pen2.jpg')
```

```
pen3 = Image.open(r'C:\Users\Hp\OneDrive\Projects\personality\Image\pen3.jpg')
pen4 = Image.open(r'C:\Users\Hp\OneDrive\Projects\personality\Image\pen4.jpg')
st.image([pen1,pen2,pen3,pen4],width=288)
pen_input= st.radio('How you hold pen ',[1,2,3,4])
```

```
stand1 =
Image.open(r'C:\Users\Hp\OneDrive\Projects\personality\Image\stand1.jpg')
stand2 =
Image.open(r'C:\Users\Hp\OneDrive\Projects\personality\Image\stand2.jpg')
stand3 =
Image.open(r'C:\Users\Hp\OneDrive\Projects\personality\Image\stand3.jpg')
stand4 =
Image.open(r'C:\Users\Hp\OneDrive\Projects\personality\Image\stand4.jpg')
st.image([stand1,stand2,stand3,stand4],width=150)
stand_input= st.radio('How you mostly stand ',[1,2,3,4])
```

##Calculating the sum

```
ope=0
```

```
neu=0
```

```
con=0
```

```
agr=0
```

```
ext=0
```

```
for i in range(0,4):
```

```
    ope= ope + a[i]
```

```
    neu= neu + b[i]
```

```
    con= con +c[i]
```

```
    agr= agr+ d[i]
```



```
ext= ext+ e[i]
```

```
##Normalize the data
```

```
ope=(int)((ope-4)*(9-1)/16)+1
```

```
neu=(int)((neu-4)*(9-1)/16)+1
```

```
con=(int)((con-4)*(9-1)/16)+1
```

```
agr=(int)((agr-4)*(9-1)/16)+1
```

```
ext=(int)((ext-4)*(9-1)/16)+1
```

```
X_ope = data[:,2]
```

```
X_neu = data[:,3]
```

```
X_con = data[:,4]
```

```
X_agr = data[:,5]
```

```
X_ext = data[:,6]
```

```
##Calculating the average
```

```
avg_ope=0
```

```
avg_neu=0
```

```
avg_con=0
```

```
avg_agr=0
```

```
avg_ext=0
```

```
for i in range(0,1024):
```

```
    avg_ope=avg_ope+X_ope[i]
```

```
    avg_neu=avg_neu+X_neu[i]
```

```
    avg_con=avg_con+X_con[i]
```

```
    avg_agr=avg_agr+X_agr[i]
```

```
    avg_ext=avg_ext+X_ext[i]
```

```
avg_ope=avg_ope/1024
```

```
avg_neu=avg_neu/1024
```

```
avg_con=avg_con/1024
```

```
avg_agr=avg_agr/1024
```

```
avg_ext=avg_ext/1024
```

```
if st.button("Predict"):
```

```
    output=predict_forest(age,ope,neu,con, agr,ext)
```

```
    st.success('Your personality types is {}'.format(output))
```

```
    if(pen_intput==1):
```

```
        st.write("When you take up something, you like to analyze it and make your  
conclusions so that everything is as per your liking. Overall, you have a unique  
personality. You can be analytical and emotional as per the situation.")
```

```
    if(stand_intput==1):
```

```
        st.write("You are a defensive person. You trying to comfort yourself,  
especially in a room full of a crowd.")
```

```
    if(stand_intput==2):
```

```
        st.write("You are feeling vulnerable and nervous but have to seem not to be  
confident in public.")
```

```
    if(stand_intput==3):
```

```
        st.write("You are not concerned with what someone else has to say or do. It  
show you are self-confidence and pride in one own's identity.")
```

```
    if(stand_intput==4):
```

```
        st.write("You are feeling small and lacking confidence. You are displaying  
submissiveness and lack of power.")
```

"Your scores"

```
source = pd.DataFrame({'Values': [ope,neu,con,agr,ext],'Parameters':  
['Openness','Neuroticism','Conscientiousness','Agreeableness','Extraversion']})  
bar_chart =  
alt.Chart(source).mark_bar().encode(y='Values:Q',x='Parameters:O',)  
st.altair_chart(bar_chart, use_container_width=True)
```

"Openness graph"

```
source = pd.DataFrame({'Values': [ope,avg_ope,1,8],'Parameters':  
['Your','average','Minium','maxium']})  
bar_chart =  
alt.Chart(source).mark_bar().encode(y='Values:Q',x='Parameters:O',)  
st.altair_chart(bar_chart, use_container_width=True)
```

"Neuroticism graph"

```
source = pd.DataFrame({'Values': [ope,avg_neu,1,9],'Parameters':  
['Your','average','Minium','maxium']})  
bar_chart =  
alt.Chart(source).mark_bar().encode(y='Values:Q',x='Parameters:O',)  
st.altair_chart(bar_chart, use_container_width=True)
```

"Conscientiousness graph"

```
source = pd.DataFrame({'Values': [ope,avg_con,1,9],'Parameters':  
['Your','average','Minium','maxium']})  
bar_chart =  
alt.Chart(source).mark_bar().encode(y='Values:Q',x='Parameters:O',)  
st.altair_chart(bar_chart, use_container_width=True)
```

"Agreeableness graph"

```
source = pd.DataFrame({'Values': [ope,avg_agr,1,8],'Parameters':  
['Your','average','Minium','maxium']})  
bar_chart =  
alt.Chart(source).mark_bar().encode(y='Values:Q',x='Parameters:O',)  
st.altair_chart(bar_chart, use_container_width=True)
```

"Extraversion graph"

```
source = pd.DataFrame({'Values': [ope,avg_ext,1,8],'Parameters':  
['Your','average','Minium','maxium']})  
bar_chart =  
alt.Chart(source).mark_bar().encode(y='Values:Q',x='Parameters:O',)  
st.altair_chart(bar_chart, use_container_width=True)
```

```
if __name__=='__main__':  
    main()
```