A Project Report

on

Personality Prediction Using Machine Learning

Submitted in partial fulfillment of the requirements

for the award of the degree of

BACHELOR OF TECHNOLOGY

in

Computer Science & Engineering

by

M. SUDHARSHINI (184G1A0599)

B. SAI SREEKANTH (184G1A0577)

K. RAJU (174G1A0563)

K. YASWANTH KUMAR (194G5A0512)

Under the Guidance of

Mr. M. Narasimhulu, M.Tech., (Ph.D)

Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY ANANTAPURAMU

(Affiliated to JNTUA & Approved by AICTE)

(Accredited by NAAC with 'A' Grade & Approved by NBA(EEE, ECE & CSE))

Rotarypuram Village, B K Samudram Mandal, Ananthapuramu-515701.

2021-2022

SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY ANANTAPURAMU

(Affiliated to JNTUA & Approved by AICTE)

(Accredited by NAAC with 'A' Grade & Approved by NBA(EEE, ECE & CSE))

Rotarypuram Village, B K Samudram Mandal, Ananthapuramu-515701.

Certificate

This is to certify that the project report entitled Personality Prediction Using Machine Learning is the bonafide work carried out by **M. Sudharshini** bearing Roll Number 184G1A0599**, B. Sai Sreekanth** bearing Roll Number 184G1A0577**, K. Raju** bearing Roll Number 174G1A0563 and **K. Yaswanth Kumar** bearing Roll Number 194G5A0512 in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science & Engineering** during the academic year 2021-2022.

Signature of the Guide

Mr. M. Narasimhulu, M.Tech., (Ph.D)
Assistant Professor

Head of the Department

Mr. P. Veera Prakash, M.Tech., (Ph.D)
Assistant Professor

Date:

EXTERNAL EXAMINER

Place: Rotarypuram

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of people who made it possible, whose constant guidance and encouragement crowned our efforts with success. It is a pleasant aspect that we now have the opportunity to express my gratitude for all of them.

It is with immense pleasure that we would like to express my indebted gratitude to my Guide **Mr. M. Narasimhulu**, **Assistant professor, Computer Science & Engineering**, who has guided me a lot and encouraged me in every step of the project work. We thank him for the stimulating guidance, constant encouragement and constructive criticism which have made possible to bring out this project work.

We are very much thankful to **Mr. P. Veera Prakash, Assistant Professor & Head of the Department, Computer Science & Engineering,** for his kind support and for providing necessary facilities to carry out the work.

We wish to convey my special thanks to **Dr. G. Bala Krishna, Principal** of **Srinivasa Ramanujan Institute of Technology** for giving the required information in doing our project work. Not to forget, we thank all other faculty and non-teaching staff, and my friends who had directly or indirectly helped and supported us in completing our project in time.

We also express our sincere thanks to the Management for providing excellent facilities**.**

Finally, we wish to convey our gratitude to our family who fostered all the requirements andfacilities that we need.

Project Associates

184G1A0599

184G1A0577

174G1A0563

194G5A0512

Declaration

We, Ms. M. Sudharshini with reg no: 184G1A0599, Mr. B. Sai Sreekanth with reg no: 184G1A0577, Mr. K. Raju with reg no: 174G1A0563, Mr. K. Yaswanth Kumar with reg no: 194G5A0512 students of SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY, Rotarypuram, hereby declare that the dissertation entitled "PERSONALITY PREDICTION USING MACHINE LEARNING" embodies the report of our project work carried out by us during IV year Bachelor of Technology under the guidance of Mr. M. Narasimhulu, Department of CSE, SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY, and this work has been submitted for the partial fulfillment of the requirements for the award of the Bachelor of Technology Degree.

The results embodied in this project have not been submitted to any other University of Institute for the award of any Degree or Diploma.

M. SUDHARSHINI	Reg no: 184G1A0599
B. SAI SREEKANTH	Reg no: 184G1A0577
K. RAJU	Reg no: 174G1A0563
K. YASWANTH KUMAR	Reg no: 194G5A0512

CONTENTS

List of Figures

List of Abbreviations

XGBoost	--	Extreme Gradient Boosting
SVM	--	Support Vector Machine
MBTI	--	Myers Briggs Type Indicator
ASM	--	Attribute Selection Measure
RAM	--	Random Access Memory
SDLC	--	Software Development life cycle
SRS	--	Software Requirement Specification
DDS	--	Design Document Specification
UAT	--	User Acceptance Testing
UML	--	Unified Modeling Language
ER	--	Entity-relationship
DFD	--	Data Flow Diagram

ABSTRACT

Personality refers to a characteristic pattern of thoughts, behaviour, and feelings that makes a person unique. Asking users to fill a questionnaire to get their personality insights could be inaccurate because the users are conscious and try to take a careful approach when filling the survey. However, when it comes to social media, users do not take any consideration before posting their opinions on social media. Therefore, the data obtained from social media could be precious to determine the user personality type. In this paper, we propose a way to analyse the user's data posted on social media by using machine learning algorithms, Random forest, Decision Tree, SVM, and XGBoost, in order to predict user personality type. Moreover, this research helps to analyse the empirical relation between the user's data posted on social media and the user's personality.

In this work, we used The Myer-Briggs Type Indicator (MBTI) introduced by Swiss psychiatrist Carl Jung. MBTI is based on sixteen personality types, and they act as a valuable reference point to understand a person's unique personality. The technique of combining these machine learning algorithms gave accurate results than the traditional naive Bayes classification and other algorithms. Results of this study can help bloggers and social media users to know what type of personality they are showing on the social media with the data they posted on the internet.

Keywords: Decision Tree, Random forest, XGBoost, performance Analysis.

1. INTRODUCTION

Personality is characterized as the trademark sets of cognitions, behaviours and passionate examples that advance from biological and ecological components. It mirrors the people as they contrast in thoughts, behaviour and sentiments. Personality characteristics are constant in world as they give a sense of high and low of explicit attributes in an individual on consistent quality instead of displaying particular personality. In order to predict distinct personality of each individual we use a machine learning model i.e., the Myers-Briggs Type Indicator (MBTI). This Project follows the principle of MBTI as a guideline's so that it helps to identify the personality of the user based on the following personality dimensions: Introvert (I) and Extrovert (E), Sensation(A) and Intuition(N), Thinking(T) and Feeling(F), Perceiving(P) and Judging(J).

The coalescence of the above four types of personality dimensions will result in sixteen types of personality such as "INFJ" or "ENFP" etc. In our model we have used algorithms like SVM, Random Forest, Decision tree, XGBoost. We have taken the dataset from the source Kaggle. In our model we first import the dataset from Kaggle. Then feed it into data analysis to check whether there are any missing or null values present in the dataset and the analysed data is then fed into data pre- processing to clean the data. After cleaning process, the data is sent to feature engineering and finally by comparing the algorithms with each other we choose the best algorithm to our model that can predict the personality of each individual.

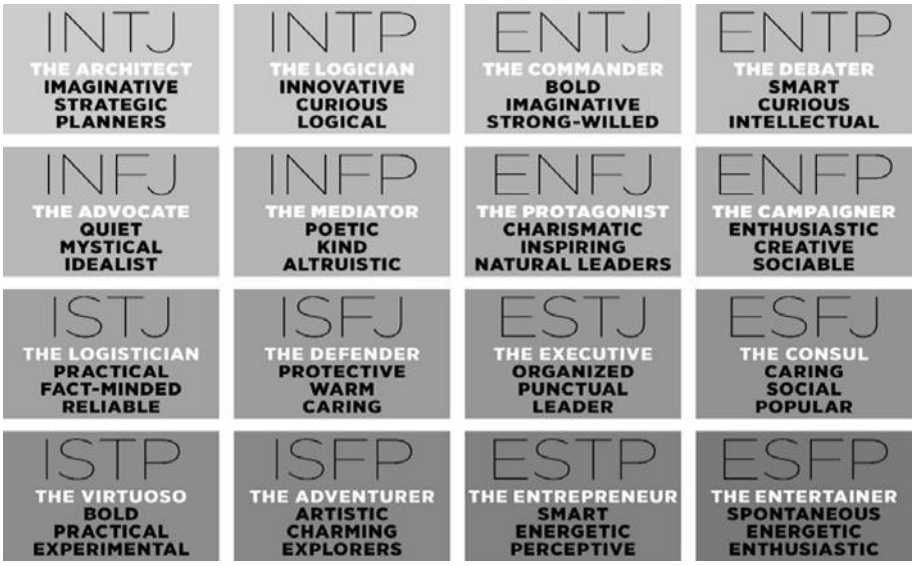


Fig.1.1 Myers Briggs 16 personality types

The current era of the internet is witnessing a huge growth of electronic media such as social websites, and a massive amount of new data is being created every minute. With the growing popularity of social media, it has become possible to disseminate this information at a rapid rate. Millions of posts are published every day on social networking sites like Facebook, Twitter, Instagram, and many others. Among these social networking sites, Facebook is the only fantastic real-time social networking tool that can be a great source of rich information for data mining. In early 2018, a whistle-blower revealed a piece of astonishing information regarding the presidential election campaign in 2016. The information involved an organization called Cambridge Analytical that harvested Facebook Profiles of 50 million users in the United States and performed a detailed analysis of their data to determine the success rates during the elections. Although it was a direct security breach by the organization on the user's data, we can conclude that user's data on social media (Facebook) can help determine the outcome of a particular business objective.

Data mining is a process of going through small or large datasets to identify hidden patterns and structures to solve different problems through data analysis and machine learning algorithms. It is a multidisciplinary field that uses statistics, Artificial intelligence, databases, and machine learning to find the insights of the dataset. In other words, data mining is knowledge discovery, pattern or data analysis, information harvesting, and others. Insights revealed by data mining techniques can be used in different fields, such as market analysis, biogenetics, etc. Data mining tools and techniques help various organizations to predict future trends by analyzing the past or current data. In data mining, different association rules are created by analyzing the data and extracting useful information out of it. To get the valuable insights from a data, many data mining techniques are applied.

The phenomenon of Big Data has changed the business world like never before. The most important part of this transformation is the strong emergence of analytics to support the shift in modern enterprises from a process-centric viewpoint to one that is more data-centric and data-driven. The data that surrounds the enterprise is being harnessed into information that informs, supports and drives decision making in a timely, repeatable manner. This biannual KPMG global CFO survey report, Being the Best: Inside the Intelligent Finance Function, brings data & analytics

(D&A) concepts into play at the outset, saying of today's finance function leaders: "Their biggest challenges lie in creating the efficiencies needed to gather and process basic financial data and continue to deliver traditional finance outputs while at the same time redeploying their limited resources to enable higher-value business decision support activities."

1.1 Objective of the Project:

The purpose of this project is to predict personality types as one of the sixteen categories of Myers Briggs personality types (MBTI) based on the correlation between people's writing styles and their psychological personalities. We believe that social media gives people the platform to express themselves freely and openly and hence those posts can be an indicator of their personality type. We acknowledge the fact that all personality types are equal.

Myers Briggs 16 personality types:

ISTJ	The Inspector
ISTP	The Crafter
ISFJ	The Protector
ISFP	The Artist
INFJ	The Advocate
INFP	The Mediator
INTJ	The Architect
INTP	The Thinker
ESTP	The Persuader
ESTJ	The Director
ESFP	The Performer
ESFJ	The Caregiver
ENFP	The Champion
ENFJ	The Giver
ENTP	The Debater
ENTJ	The Commander

2. LITERATURE SURVEY

[1] Firoj Alam, Evgeny A. Stepanov, Giuseppe Riccardi, "Personality Traits Recognition on Social Network - Facebook", AAAI Technical Report WS-13-01 Computational Personality Recognition.

For the natural and social interaction it is necessary to understand human behavior. Personality is one of the fundamental aspects, by which we can understand behavioral dispositions. It is evident that there is a strong correlation between users' personality and the way they behave on online social network (e.g., Facebook). This paper presents automatic recognition of Big-5 personality traits on social network (Facebook) using users' status text. For the natural and social interaction it is necessary to understand human behavior. Personality is one of the fundamental aspects, by which we can understand behavioral dispositions. It is evident that there is a strong correlation between users' personality and the way they behave on online social network (e.g., Facebook). This paper presents automatic recognition of Big-5 personality traits on social network (Facebook) using users' status text. For the automatic recognition we studied different classification methods such as SMO (Sequential Minimal Optimization for Support Vector Machine), Bayesian Logistic Regression (BLR) and Multinomial Naïve Bayes (MNB) sparse modeling. Performance of the systems had been measured using macro-averaged precision, recall and F1; weighted average accuracy (WA) and un-weighted average accuracy (UA). Our comparative study shows that MNB performs better than BLR and SMO for personality traits recognition on the social network data.

Summary: we present our baseline study to automatically recognize BIG-5 personality traits on the social network data (Facebook status messages). We explored different classification methods. In this study we observed that MNB sparse model performs better than SMO and BLR. We report system performances using macro-averaged precision, recall, F1, and accuracy (WA). Future directions of this study include integrating syntactic, semantic and statistical features; studying feature selection and classifier combination methods, which may lead to provide more information to recognize personality traits.

[2] Basant Agarwal, "Personality Detection from Text: A Review", International Journal of Computer System (ISSN: XXXX – XXXX), Volume 01– Issue 01, September, 2014.

Social media usage has been on an ever increasing exponential rise. Usage of social media sites, such as Twitter and Facebook, for social interaction has also become a popular trend. It is estimated that on an average, around 6,000 tweets are tweeted on Twitter every second. With people spending on an average 35 minutes on Facebook each day, it is also estimated that there are about 317,000 status updates on Facebook per minute. These vast volumes of data have powerful information locked within them. This data can be analyzed and several purposes. The use of such social media data for predicting user personality is common. Prediction models have been successfully built that can predict several user attributes - age, gender, personality traits, occupation, political orientation etc. Standards in personality models such as the Big Five model, DISC and the Myers-Briggs Type Indicator have been the basis for all such personality prediction. A user's social media data can thus be used to predict his/her personality. The main objective of this work is to review the work carried out for personality prediction using social media data

Summary: Social media is being most widely used for social interaction and communication. A user's behaviour on social platforms is reflected in his/her tweets, status updates, comments, interests etc., which in turn reveals traits of his/her personality. A user's personality is correlated with their online behaviour and hence what user's share/write on social media can be used to extract information needed to identify his/her behaviour.

[3] Nadeem Ahmad, Jawaid Siddique, "Personality Assessment using Twitter Tweets", 21st International Conference on Knowledge Based and Intelligent Information and Engineering Systems, KES2017, 6-8 September 2017, Marseille, France.

Social media has seen an exponential growth in the last few years and has become an easy and popular method - both for social interaction and information distribution. People use social media to express themselves to the world. These expressions characterize the behavior of a user and human behavior is a reflection of his/her personality. Thus, there is a strong interconnection between a user's

personality and their online behavior on social networks. This implies that posts made by a user can be analyzed to obtain pieces of information which can be used to recognize his/her personality traits. Of all the human attributes, personality has been considered the most difficult to understand. It is vital as it can be utilized to define the unique characteristics of a person. A central part of human conduct/behavior, personality is made up of emotions, conduct, patterns of thoughts, feelings, which makes a person one of a kind. An individual's preference of things (book, music, films etc.,) and interactions with other people are influenced by his/her personality.

Summary: It has been possible to predict user attributes such as age, gender, political orientation etc., using social media and models have been successfully built to achieve this. Work of several researchers has also focused on using social media data for predicting personality which has been successfully demonstrated. It is thus possible to automatically predict personality using data from social media which is more simpler, cost effective and efficient compared to the manual methods.

[4] Shuotian Bai a, Sha Yuana, Bibo Haoa and Tingshao Zhu , "Predicting personality traits of microblog users", Web Intelligence and Agent Systems: An International Journal 12 (2014) 249–265 249 DOI 10.3233/WIA-140295 IOS Press.

Personality can be defined as a set of characteristics which makes a person unique. Psychological theory suggests that people's behavior is a reflection of personality. Therefore, it is feasible to predict personality through behavior. Conventional personality assessment is performed by self-report inventory. Participants need to fill in a tedious inventory to get their personality scores. In the large-scale investigation, every returned inventory needs manual computation, which costs much manual efforts and cannot be done in real time. In order to avoid these shortages, this research aims to objectively predict the Big-Five personality from the usage records of Sina Microblog. Since its initial launch in December, 2005, Sina Microblog has been the leading microblogging service provider in China. Millions of users upload and download resources via microblogging status everyday. Therefore, by conducting an online user survey of 444 active users, this paper analyzes the relation modes between personality and online behavior. Furthermore, this research proposes multi-task regression and incremental regression to predict the BigFive personality from online behaviors. The results indicate that correlation factors are

significant between different personality dimensions. Besides, our training data set is reliable enough and multi-task regression performs better than other modeling algorithms.

Summary: This research leaves some blanks to be desired in future work. First, we will continue to collect users' data in Sina Microblog, and invite more participants to get a larger dataset. To achieve this, some interesting functions will be added to the platform application. In feature, once the participant finish the inventory, the application will give some feedback information to the user, such as the advice for psychological care or friend recommendation on identical personality.

[5] **Yago Saez , Carlos Navarro , Asuncion Mochon and Pedro Isasi "A System for Personality and Happiness Detection", International Journal of Artificial Intelligence and Interactive Multimedia, vol. 2, no. 5, 2014.**

This work proposes a platform for estimating personality and happiness. Starting from Eysenck's theory about human's personality, authors seek to provide a platform for collecting text messages from social media (WhatsApp), and classifying them into different personality categories. Researchers have tried to obtain information about the personality of human beings through direct means such as the EPQ-R questionnaire, but they have also used indirect methods. Because personality is considered to be stable over time and throughout different situations, specialized psychologists are able to infer the personality profile of a subject by observing the subject's behaviour. One of the sources of knowledge about the behaviour of individuals is written text. According to research in this field, it is reasonable to expect that different individuals will have different ways of expressing themselves through the written word, and these differences will correspond to their individual.

3. SYSTEM ANALYSIS & FEASIBILITY STUDY

3.1 Existing Method:

In the existing system, implementation of machine learning algorithms is bit complex to build due to the lack of information about the data visualization. Mathematical calculations are used in existing system for model building this may takes the lot of time and complexity. To overcome all this, we use machine learning packages available in the scikit-learn library.

Disadvantages:

- High complexity.
- Time consuming.

3.2 Proposed System:

Proposed several machine learning models to classify the personality through the posts, but none have adequately addressed this misdiagnosis problem. Also, similar studies that have proposed models for evaluation of such performance classification mostly do not consider the heterogeneity and the size of the data Therefore, we propose a Decision Tree, Random Forest, and XGBoost to classify the personality.

Advantages:

- Highest accuracy.
- Reduces time complexity.

3.3 Block Diagram:

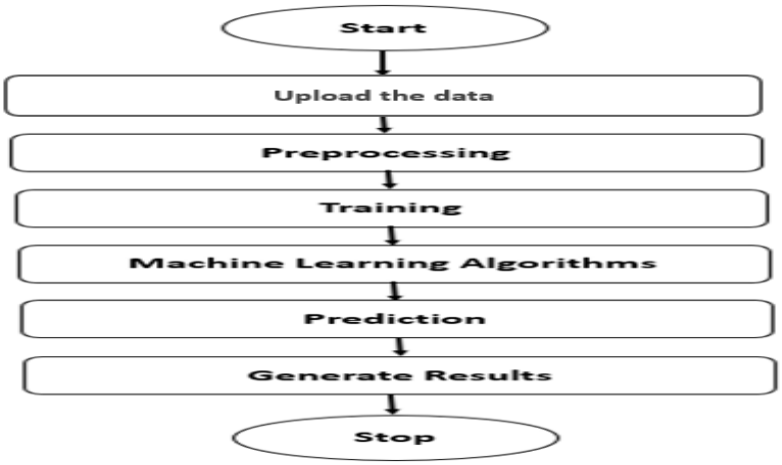


Fig.3.1 Block Diagram

3.4 Architecture:

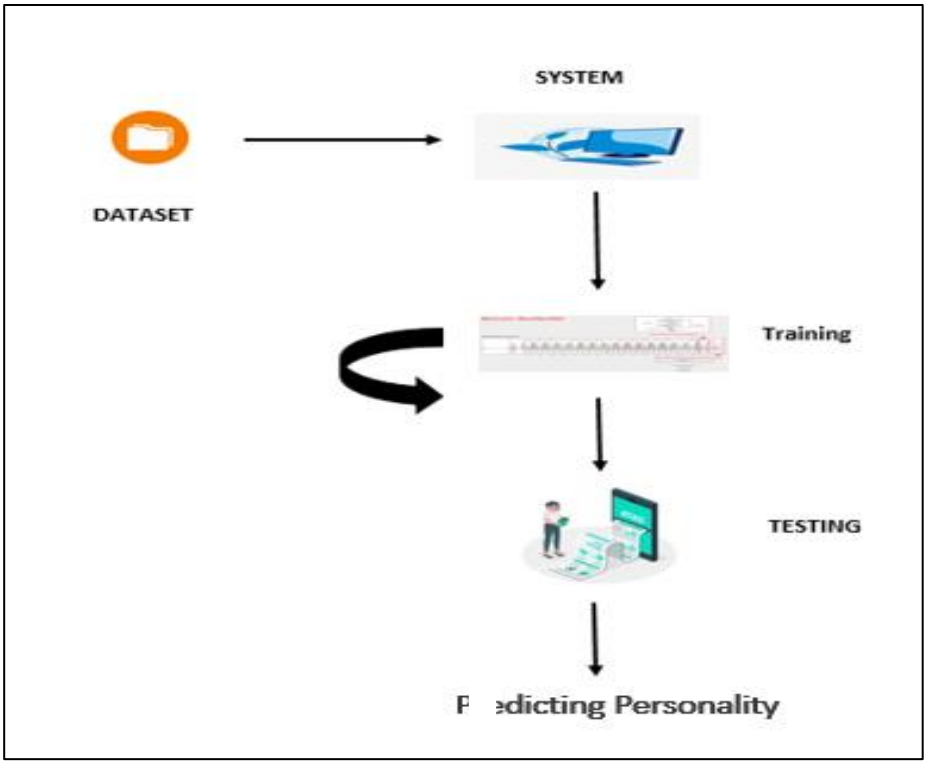


Fig.3.2 Architecture of Personality Prediction system

3.4.1 Architecture of MBTI Classifier

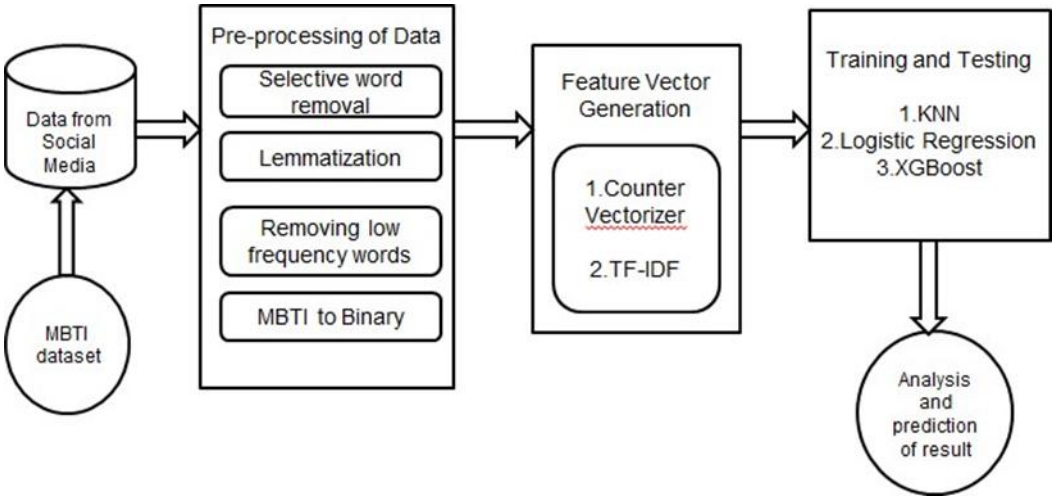


Fig.3.3 Architecture of MBTI Classifier

4. METHODOLOGY AND ALGORITHMS

4.1 Machine Learning:

Machine Learning is the area of study which enables machines to learn without being explicitly programmed. It is seen as a subset of artificial intelligence. Machine learning algorithms build a mathematical model of sample data, known as "training data". A Machine Learning system learns from historical data, builds the prediction models, and whenever it receives new data, predicts the output for it.

Machine learning gives a system the ability to learn automatically and improve its recommendations using data alone, with no additional programming needed. Because retailers generate enormous amounts of data, machine learning technology quickly proves its value. When a machine learning system is fed data—themore, the better— it searches for patterns. Going forward, it can use the patterns it identifies within the data to make better decisions. The accuracy of predicted output depends upon the amount of data, as the huge amount of data helps to build a better model which predicts the output more accurately.

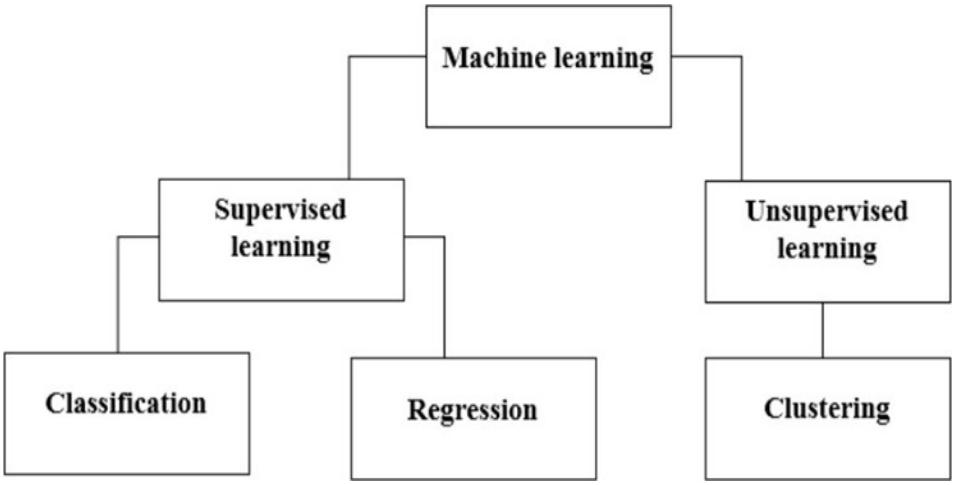


Fig.4.1 Types of Machine Learning

4.1.1 Supervised Learning:

Supervised learning is a process of providing input data as well as correct output data to the machine learning model. The aim of a supervised learning algorithm is to find a mapping function to map the input variable(x) with the output variable(y). Supervised learning is the type of machine learning in which machines

are trained using well "labeled" trained data. Once the training process is completed, the model is tested on the basis of test data (a subset of the training set), and then it predicts the output.

In supervised learning, models are trained using labelled dataset, where the model learns about each type of data. Once the training process is completed, themodel is tested on the basis of test data (a subset of the training set), and then it predicts the output. The working of Supervised learning can be easily understood by the below example and diagram:

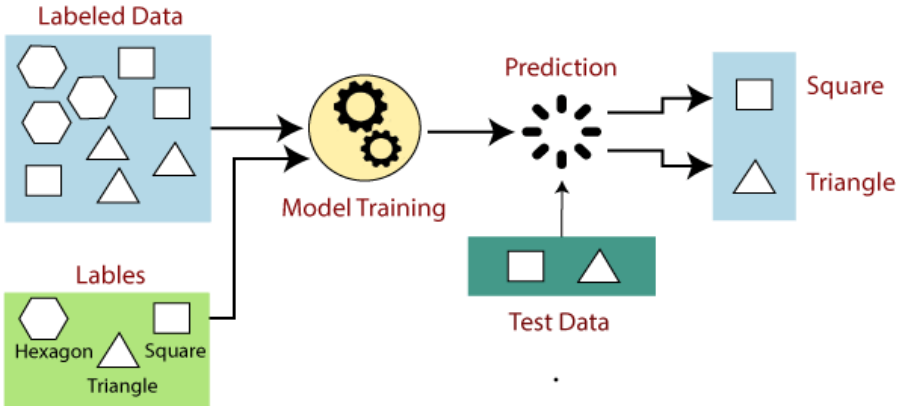


Fig.4.2 Process of any ML algorithm

Suppose we have a dataset of different types of shapes which includes square, rectangle, triangle, and Polygon. Now the first step is that we need to train the model for each shape.

If the given shape has four sides, and all the sides are equal, then it will be labelled as a Square.

- If the given shape has three sides, then it will be labelled as a triangle.
- If the given shape has six equal sides then it will be labelled as hexagon.

Now, after training, we test our model using the test set, and the task of the model is to identify the shape. The machine is already trained on all types of shapes, and when it finds a new shape, it classifies the shape on the bases of a number of sides, and predicts the output. Supervised learning can be further divided into two types:

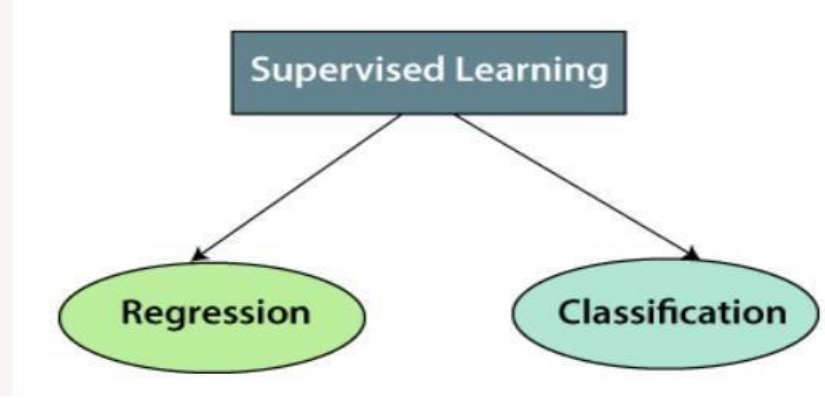


Fig.4.3 Types of Supervised learning

4.1.1.1 Regression

Regression is a supervised learning technique which helps in finding the correlation between variables and enables us to predict the continuous output variable based on the one or more predictor variables. It is mainly used for prediction**.** Regression analysis helps us to understand how the value of the dependent variable is changing corresponding to an independent variable when other independent variables are held fixed. It predicts continuous/real values such as price**,** etc. Some real-world examples for regression are predicting the sales based on input parameters etc.

4.1.1.2 Classification

Classification is supervised learning. It can be performed on both structured and unstructured data. Classification is the process of finding a model that helps to separate the data into different categorical classes. In this process, data is categorized under different labels according to some parameters given in input and then the labels are predicted for the data.

Steps Involved in Supervised Learning

1. First Determine the type of training dataset
2. Collect/Gather the labelled training data.
3. Split the training dataset into training dataset, test dataset, and validation dataset.
4. Determine the input features of the training dataset, which should have enough knowledge so that the model can accurately predict the output.
5. Determine the suitable algorithm for the model, such as support vector machine, decision tree, etc.

6. Execute the algorithm on the training dataset. Sometimes we needvalidation sets as the control parameters, which are the subset of training datasets.
7. Evaluate the accuracy of the model by providing the test set. If the model predicts the correct output, which means our model is accurate.

4.1.2 Unsupervised Learning

Unsupervised learning is a machine learning technique in which models are not supervised using training dataset. Models itself find the hidden patterns and insights from the given data. It mainly deals with the unlabelled data. Unsupervised learning cannot be directly applied to a regression or classification problem because unlike supervised learning, we have the input data but no corresponding output data. The goal of unsupervised learning is to find the underlying structure of dataset, group that data according to similarities, and represent that dataset in a compressed format.

4.2 Algorithms Used

4.2.1 Decision Tree

Decision tree is a flowchart-like tree structure where an internal node represents feature (or attribute), the branch represents a decision rule, and each leaf node represents the outcome. The topmost node in a decision tree is known as the root node. It learns to partition on the basis of the attribute value. It partitions the tree in recursively manner call recursive partitioning. This flowchart-like structure helps you in decision making. It's visualization like a flowchart diagram which easily mimics the human level thinking. That is why decision trees are easy to understand and interpret.

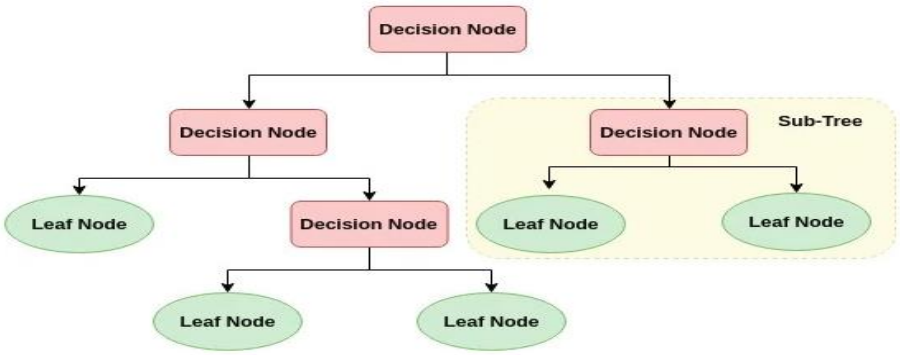


Fig.4.4 Decision Tree Algorithm

The basic idea behind any decision tree algorithm is as follows:

1. Select the best attribute using Attribute Selection Measures (ASM) to split the records.
2. Make that attribute a decision node and breaks the dataset into smaller subsets.
3. Starts tree building by repeating this process recursively for each child until one of the conditions will match:
    - All the tuples belong to the same attribute value.
    - There are no more remaining attributes.
    - There are no more instances.

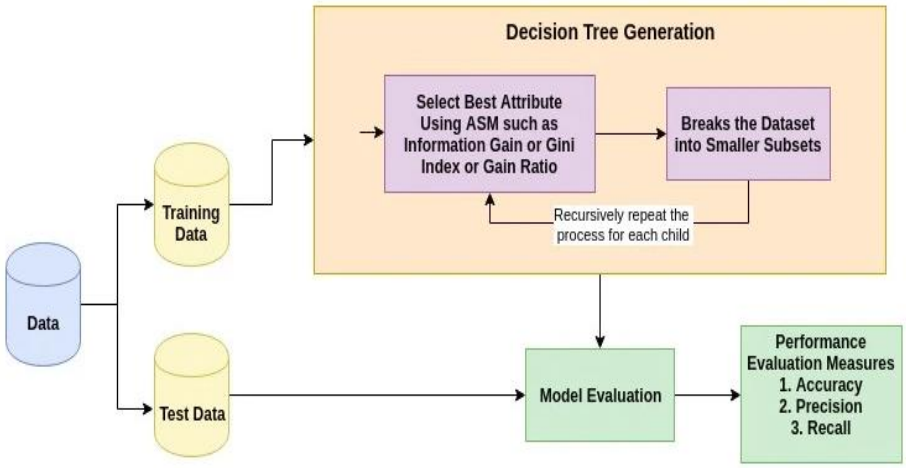


Fig.4.5 Working of Decision Tree

4.2.2 XGBOOST:

XGBoost stands for “Extreme Gradient Boosting”. XGBoost is an optimized distributed gradient boosting library designed to be highly efficient, flexible and portable. It implements Machine Learning algorithms under the Gradient Boosting framework. It provides a parallel tree boosting to solve many data science problems in a fast and accurate way.

Boosting

Boosting is an ensemble learning technique to build a strong classifier from several weak classifiers in series. Boosting algorithms play a crucial role in dealing with bias-variance trade-off. Unlike bagging algorithms, which only controls for high variance in a model, boosting controls both the aspects (bias & variance) and is considered to be more effective.

Below are the few types of boosting algorithms:

- AdaBoost (Adaptive Boosting)
- Gradient Boosting
- XGBoost
- CatBoost
- Light GBM

XGBoost:

XGBoost stands for eXtreme Gradient Boosting. It became popular in the recent days and is dominating applied machine learning and Kaggle competitions for structured data because of its scalability. XGBoost is an extension to gradient boosted decision trees (GBM) and specially designed to improve speed and performance.

4.2.3 Random Forest Classifier:

A random forest is a machine learning technique that's used to solve regression and classification problems. It utilizes ensemble learning, which is a technique that combines many classifiers to provide solutions to complex problems.

A random forest algorithm consists of many decision trees. The 'forest' generated by the random forest algorithm is trained through bagging or bootstrap aggregating. Bagging is an ensemble meta-algorithm that improves the accuracy of machine learning algorithms.

The (random forest) algorithm establishes the outcome based on the predictions of the decision trees. It predicts by taking the average or mean of the output from various trees. Increasing the number of trees increases the precision of the outcome.

A random forest eradicates the limitations of a decision tree algorithm. It reduces the over fitting of datasets and increases precision. It generates predictions without requiring many configurations in packages (like Scikit-learn).

Features of a Random Forest Algorithm:

- It's more accurate than the decision tree algorithm.
- It provides an effective way of handling missing data.

- It can produce a reasonable prediction without hyper-parameter tuning.
- It solves the issue of over fitting in decision trees.
- In every random forest tree, a subset of features is selected randomly at the node's splitting point.

Decision trees are the building blocks of a random forest algorithm. A decision tree is a decision support technique that forms a tree-like structure. An overview of decision trees will help us understand how random forest algorithms work.

A decision tree consists of three components: decision nodes, leaf nodes, and a root node. A decision tree algorithm divides a training dataset into branches, which further segregate into other branches. This sequence continues until a leaf node is attained. The leaf node cannot be segregated further.

The nodes in the decision tree represent attributes that are used for predicting the outcome. Decision nodes provide a link to the leaves. The following diagram shows the three types of nodes in a decision tree.

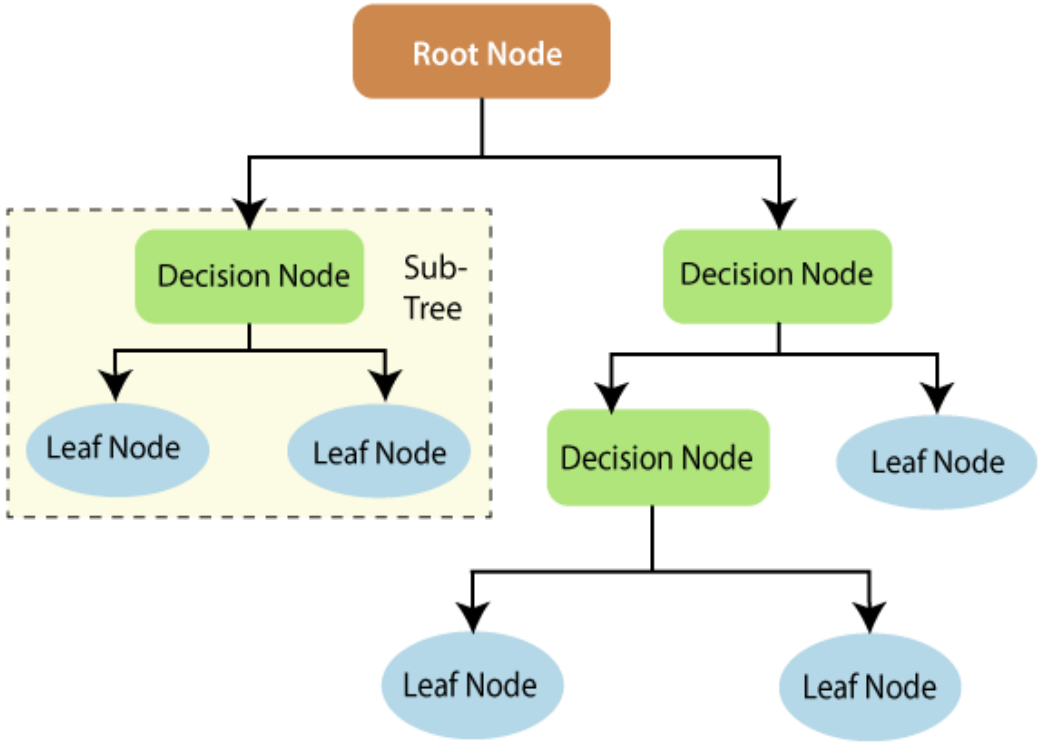


Fig.4.6 Types of nodes in Decision Tree

The information theory can provide more information on how decision trees work. Entropy and information gain are the building blocks of decision trees. An

overview of these fundamental concepts will improve our understanding of how decision trees are built.

Entropy is a metric for calculating uncertainty. Information gain is a measure of how uncertainty in the target variable is reduced, given a set of independent variables.

The information gain concept involves using independent variables (features) to gain information about a target variable (class). The entropy of the target variable (Y) and the conditional entropy of Y (given X) are used to estimate the information gain. In this case, the conditional entropy is subtracted from the entropy of Y.

Information gain is used in the training of decision trees. It helps in reducing uncertainty in these trees. A high information gain means that a high degree of uncertainty (information entropy) has been removed. Entropy and information gain are important in splitting branches, which is an important activity in the construction of decision trees.

Let's take a simple example of how a decision tree works. Suppose we want to predict if a customer will purchase a mobile phone or not. The features of the phone form the basis of his decision. This analysis can be presented in a decision tree diagram.

The root node and decision nodes of the decision represent the features of the phone mentioned above. The leaf node represents the final output, either *buying* or *not buying*. The main features that determine the choice include the price, internal storage, and Random Access Memory (RAM). The decision tree will appear as follows.

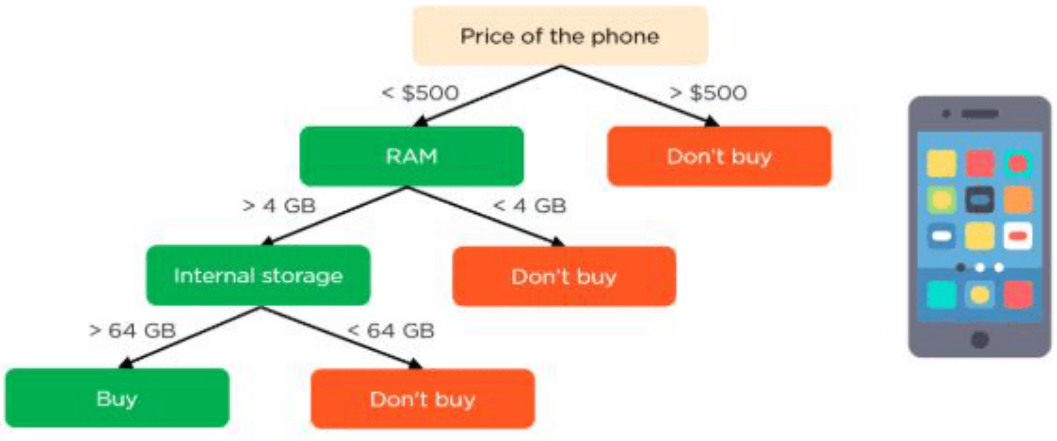


Fig.4.7 Example of Decision Tree Algorithm

Applying decision trees in random forest

The main difference between the decision tree algorithm and the random forest algorithm is that establishing root nodes and segregating nodes is done randomly in the latter. The random forest employs the bagging method to generate the required prediction.

Bagging involves using different samples of data (training data) rather than just one sample. A training dataset comprises observations and features that are used for making predictions. The decision trees produce different outputs, depending on the training data fed to the random forest algorithm. These outputs will be ranked, and the highest will be selected as the final output.

Our first example can still be used to explain how random forests work. Instead of having a single decision tree, the random forest will have many decision trees. Let's assume we have only four decision trees. In this case, the training data comprising the phone's observations and features will be divided into four root nodes.

The root nodes could represent four features that could influence the customer's choice (price, internal storage, camera, and RAM). The random forest will split the nodes by selecting features randomly. The final prediction will be selected based on the outcome of the four trees.

The outcome chosen by most decision trees will be the final choice. If three trees predict *buying*, and one tree predicts *not buying*, then the final prediction will be *buying*. In this case, it's predicted that the customer will buy the phone.

4.2.4 Support Vector Machine

The objective of the support vector machine algorithm is to find a hyper plane in an N-dimensional space (N — the number of features) that distinctly classifies the data points.

Possible hyper planes :

To separate the two classes of data points, there are many possible Hyper planes that could be chosen. Our objective is to find a plane that has the maximum margin, i.e. the maximum distance between data points of both classes. Maximizing the margin distance provides some reinforcement so that future data points can be

classified with more confidence.

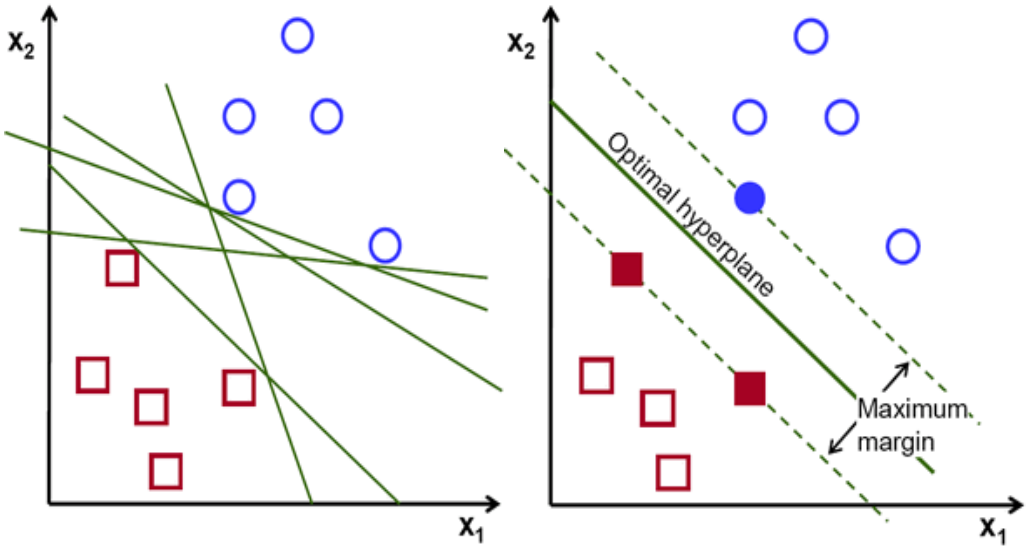


Fig.4.8 Hyper planes in SVM

Hyper planes and Support Vectors

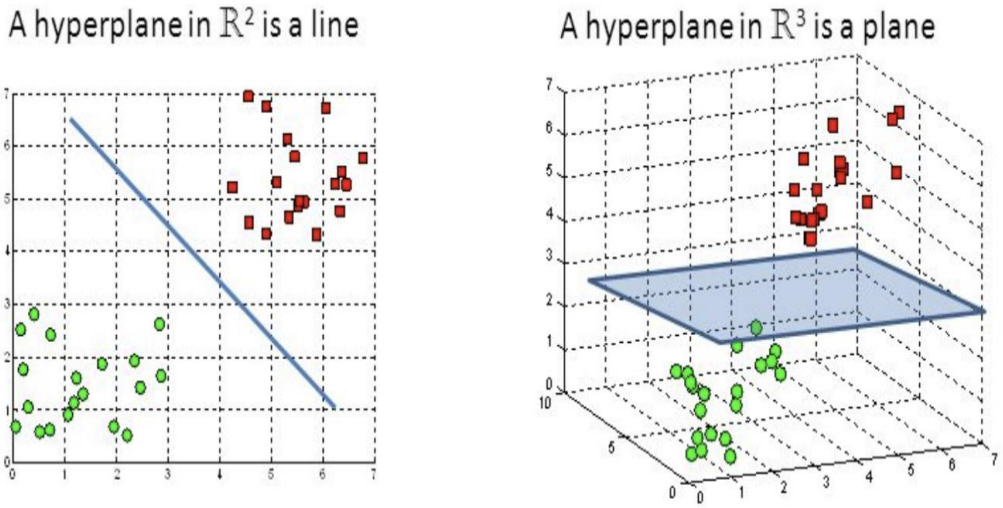


Fig.4.9 Hyper planes in a line and plane

Hyper planes in 2D and 3D feature space

Hyper planes are decision boundaries that help classify the data points. Data points falling on either side of the hyper plane can be attributed to different classes. Also, the dimension of the hyper plane depends upon the number of features. If the number of input features is 2, then the hyper plane is just a line. If the number of input features is 3, then the hyper plane becomes a two-dimensional plane. It becomes

difficult to imagine when the number of features exceeds 3.

Support Vectors

Support vectors are data points that are closer to the hyper plane and influence the position and orientation of the hyper plane. Using these support vectors, we maximize the margin of the classifier. Deleting the support vectors will change the position of the hyper plane. These are the points that help us build our SVM.

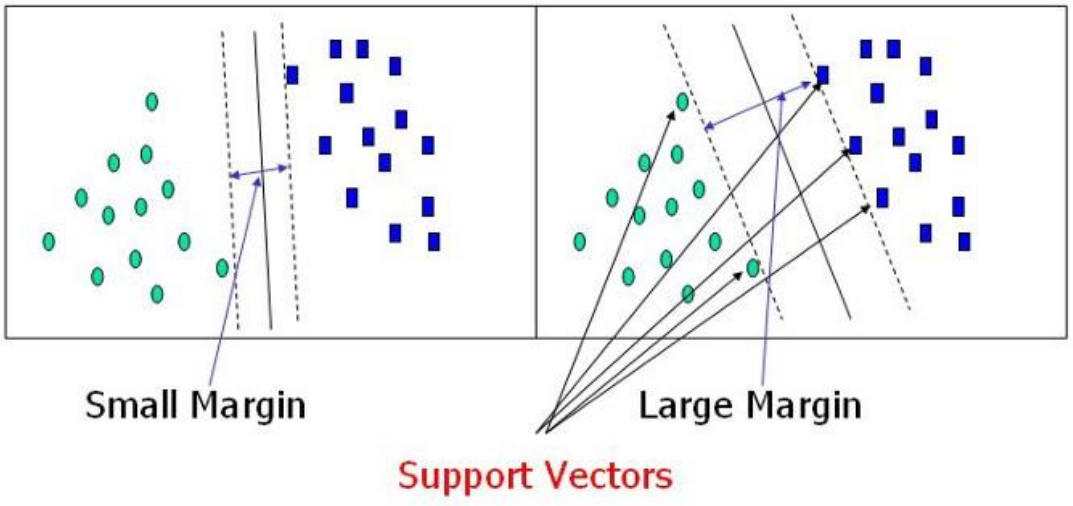


Fig.4.10 Support Vectors in SVM

Large Margin Intuition

In logistic regression, we take the output of the linear function and squash the value within the range of [0,1] using the sigmoid function. If the squashed value is greater than a threshold value (0.5) we assign it a label 1, else we assign it a label 0. In SVM, we take the output of the linear function and if that output is greater than 1, we identify it with one class and if the output is -1, we identify is with another class. Since the threshold values are changed to 1 and -1 in SVM, we obtain this reinforcement range of values ([-1, 1]) which acts as margin.

Cost Function and Gradient Updates

In the SVM algorithm, we are looking to maximize the margin between the data points and the hyper plane. The loss function that helps maximize the margin is hinge loss.

$$c(x, y, f(x)) = \begin{cases} 0, & \text{if } y * f(x) \geq 1 \\ 1 - y * f(x), & \text{else} \end{cases}$$

$$c(x, y, f(x)) = (1 - y * f(x))_+$$

Hinge loss function (function on left can be represented as a function on the right)The cost is 0 if the predicted value and the actual value are of the same sign. If they are not, we then calculate the loss value. We also add a regularization parameter the cost function. The objective of the regularization parameter is to balance the margin maximization and loss. After adding the regularization parameter, the cost functions looks as below.

$$min_w \lambda \parallel w \parallel^2 + \sum_{i=1}^{n} (1 - y_i \langle x_i, w \rangle)_+$$

Loss function for SVM

Now that we have the loss function, we take partial derivatives with respect to the weights to find the gradients. Using the gradients, we can update our weights.

$$\frac{\delta}{\delta w_k} \lambda \parallel w \parallel^2 = 2\lambda w_k$$

$$\frac{\delta}{\delta w_k} \left(1 - y_i \langle x_i, w \rangle\right)_+ = \begin{cases} 0, & \text{if } y_i \langle x_i, w \rangle \geq 1 \\ -y_i x_{ik}, & \text{else} \end{cases}$$

Gradients

When there is no misclassification, i.e. our model correctly predicts the class of our data point, we only have to update the gradient from the regularization parameter.

$$w = w - \alpha \cdot (2\lambda w)$$

Gradient Update — No misclassification

When there is a misclassification, i.e. our model make a mistake on the prediction of the class of our data point, we include the loss along with the regularization parameter to perform gradient update.

5. SOFTWARE DEVELOPMENT LIFE CYCLE

Software Development Life Cycle (SDLC) is a process used by the software industry to design, develop and test high quality softwares. The SDLC aims to produce a high-quality software that meets or exceeds customer expectations, reaches completion within times and cost estimates. SDLC is the acronym of Software Development Life Cycle. It is also called as Software Development Process. SDLC is a framework defining tasks performed at each step in the software development process.

ISO/IEC 12207 is an international standard for software life-cycle processes. It aims to be the standard that defines all the tasks required for developing and maintaining software. SDLC is a process followed for a software project, within a software organization. It consists of a detailed plan describing how to develop, maintain, replace and alter or enhance specific software. The life cycle defines a methodology for improving the quality of software and the overall development process.

A typical Software Development Life Cycle consists of the following stages:

Stage 1: Planning and Requirement Analysis

Requirement analysis is the most important and fundamental stage in SDLC. It is performed by the senior members of the team with inputs from the customer, the sales department, market surveys and domain experts in the industry. This information is then used to plan the basic project approach and to conduct product feasibility study in the economical, operational and technical areas.

Planning for the quality assurance requirements and identification of the risks associated with the project is also done in the planning stage. The outcome of the technical feasibility study is to define the various technical approaches that can be followed to implement the project successfully with minimum risks.

Stage 2: Defining Requirements

Once the requirement analysis is done the next step is to clearly define and document the product requirements and get them approved from the customer or the market analysts. This is done through an SRS (Software Requirement Specification)

document which consists of all the product requirements to be designed and developed during the project life cycle.

Stage 3: Designing the Product Architecture

SRS is the reference for product architects to come out with the best architecture for the product to be developed. Based on the requirements specified in SRS, usually more than one design approach for the product architecture is proposed and documented in a DDS - Design Document Specification.

This DDS is reviewed by all the important stakeholders and based on various parameters as risk assessment, product robustness, design modularity, budget and time constraints, the best design approach is selected for the product.

A design approach clearly defines all the architectural modules of the product along with its communication and data flow representation with the external and third party modules (if any). The internal design of all the modules of the proposed architecture should be clearly defined with the minutest of the details in DDS.

Stage 4: Building or Developing the Product

In this stage of SDLC the actual development starts and the product is built. The programming code is generated as per DDS during this stage. If the design is performed in a detailed and organized manner, code generation can be accomplished without much hassle.

Developers must follow the coding guidelines defined by their organization and programming tools like compilers, interpreters, debuggers, etc. are used to generate the code. Different high level programming languages such as C, C++, Pascal, Java and PHP are used for coding. The programming language is chosen with respect to the type of software being developed.

Stage 5: Testing the Product

This stage is usually a subset of all the stages as in the modern SDLC models, the testing activities are mostly involved in all the stages of SDLC. However, this stage refers to the testing only stage of the product where product defects are reported, tracked, fixed and retested, until the product reaches the quality standards defined in the SRS.

Stage 6: Deployment in the Market and Maintenance

Once the product is tested and ready to be deployed it is released formally in the appropriate market. Sometimes product deployment happens in stages as per the business strategy of that organization. The product may first be released in a limited segment and tested in the real business environment (UAT- User acceptance testing). Then based on the feedback, the product may be released as it is or with suggested enhancements in the targeting market segment. After the product is released in the market, its maintenance is done for the existing customer base.

SDLC Models:

There are various software development life cycle models defined and designed which are followed during the software development process. These models are also referred as Software Development Process Models". Each process model follows a Series of steps unique to its type to ensure success in the process of software development.

Following are the most important and popular SDLC models followed in the industry:

- Waterfall Model
- Iterative Model
- Spiral Model
- V-Model
- Big Bang Model

The Waterfall Model was the first Process Model to be introduced. It is also referred to as a linear-sequential life cycle model. It is very simple to understand and use. In a waterfall model, each phase must be completed before the next phase can begin and there is no overlapping in the phases.

The Waterfall model is the earliest SDLC approach that was used for software development. The waterfall Model illustrates the software development process in a linear sequential flow. This means that any phase in the development process begins only if the previous phase is complete. In this waterfall model, the phases do not overlap.

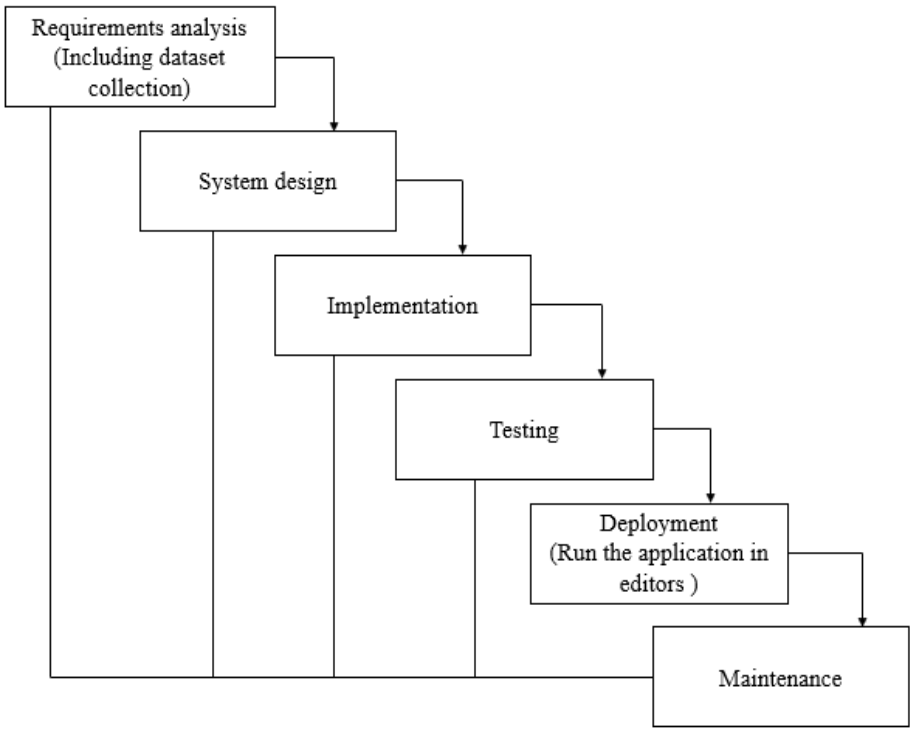


Fig.5.1 Waterfall Model

- **Requirement Gathering and analysis** − All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification document.
- **System Design** − the requirement specifications from first phase are studied in this phase and the system design is prepared. This system design helps in specifying hardware and system requirements and helps in defining the overall system architecture.
- **Implementation** − with inputs from the system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality, which is referred to as Unit Testing.
- **Integration and Testing** − All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.
- **Deployment of system** − Once the functional and non-functional testing is done; the product is deployed in the customer environment or released into the market.

- **Maintenance** − There are some issues which come up in the client environment. To fix those issues, patches are released. Also, to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

5.1 Feasibility Study

The feasibility of the project is analysed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- ECONOMICAL FEASIBILITY
- TECHNICAL FEASIBILITY
- SOCIAL FEASIBILITY

5.1.1 Economic feasibility:

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus, the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

5.1.2 Technical feasibility:

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

5.1.3 Social feasibility:

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

6. SYSTEM REQUIREMENTS SPECIFICATION

6.1 Functional and non-functional requirements

Requirement's analysis is very critical process that enables the success of a system or software project to be assessed. Requirements are generally split into two types: Functional and non-functional requirements.

6.1.1 Functional Requirements

These are the requirements that the end user specifically demands as basic facilities that the system should offer. All these functionalities need to be necessarily incorporated into the system as a part of the contract. These are represented or stated in the form of input to be given to the system, the operation performed and the output expected. They are basically the requirements stated by the user which one can see directly in the final product, unlike the non-functional requirements.

Examples of functional requirements:

1) Authentication of user whenever he/she logs into the system
2) System shutdown in case of a cyber-attack
3) A verification email is sent to user whenever he/she register for the first time on some software system.

6.1.2 Non-functional requirements:

These are basically the quality constraints that the system must satisfy according to the project contract. The priority or extent to which these factors are implemented varies from one project to other. They are also called non-behavioral requirements. They basically deal with issues like:

- Portability
- Security
- Maintainability
- Reliability
- Scalability
- Performance
- Reusability
- Flexibility

Examples of non-functional requirements:

1) Emails should be sent with a latency of no greater than 12 hours from such an activity.
2) The processing of each request should be done within 10 seconds
3) The site should load in 3 seconds whenever of simultaneous users are > 10000

6.2 Software and Hardware Requirements

Hardware:

Operating system	: Windows 10
RAM	: 8 GB
Hard disc or SSD	: More than 500 GB
Processor	: Intel 3rd generation or high or Ryzen with 8 GB Ram

Software:

Software's	: Python 3.6 or high version
IDE	: PyCharm.
Framework	: Flask

7. SYSTEM DESIGN

7.1 Input Design

In an information system, input is the raw data that is processed to produce output. During the input design, the developers must consider the input devices such as PC, MICR, OMR, etc. Therefore, the quality of system input determines the quality of system output.

Well-designed input forms and screens have following properties:

- It should serve specific purpose effectively such as storing, recording, and retrieving the information.
- It ensures proper completion with accuracy.
- It should be easy to fill and straightforward.
- It should focus on user's attention, consistency, and simplicity.
- All these objectives are obtained using the knowledge of basic design principles regarding
    - What are the inputs needed for the system?
    - How end users respond to different elements of forms and screens.

7.1.1 Objectives for Input Design

The objectives of input design are

- To design data entry and input procedures
- To reduce input volume
- To design source documents for data capture or devise other data capture methods
- To design input data records, data entry screens, user interface screens, etc.
- To use validation checks and develop effective input controls.

7.2 Output Design

The design of output is the most important task of any system. During output design, developers identify the type of outputs needed, and consider the necessary output controls and prototype report layouts.

7.2.1 Objectives of Output Design

The objectives of input design are:

- To develop output design that serves the intended purpose and eliminates the production of unwanted output.
- To develop the output design that meets the end user's requirements.
- To deliver the appropriate quantity of output.
- To form the output in appropriate format and direct it to the right person.
- To make the output available on time for making good decisions.

7.3 Modules

7.3.1 User

7.3.1.1 View Home page:

Here user view the home page of the personality prediction web application.

7.3.1.2 View Upload page:

In the about page, users can learn more about the personality prediction.

7.3.1.3 Input Model:

The user must provide input values for the certain fields in order to get results.

7.3.1.4 View Results:

User view's the generated results from the model.

7.3.1.5 View Score:

Here user have ability to view the score in %

7.3.2 System

7.3.2.1 Working on dataset

System checks for data whether it is available or not and load the data in csv files.

7.3.2.2 Pre-processing

Data need to be pre-processed according the models it helps to increase the accuracy of the model and better information about the data.

7.3.2.3 Training the data

After pre-processing the data will split into two parts as train and test data before training with the given algorithms.

7.3.2.4 Model Building

To create a model that predicts the personality with better accuracy, this module will help user.

7.3.2.5 Generated Score

Here user view the score in %

7.3.2.6 Generate Results

We train the machine learning algorithm and calculate the personality prediction.

7.4 UML Diagrams

UML stands for Unified Modelling Language. UML is a standardized general-purpose modelling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object-oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modelling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modelling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modelling of large and complex systems.

The UML is a very important part of developing objects-oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

GOALS:

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modelling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modelling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations,

frameworks, patterns and components.

7. Integrate best practices.

7.4.1 Use Case Diagram

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis.

Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases.

The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

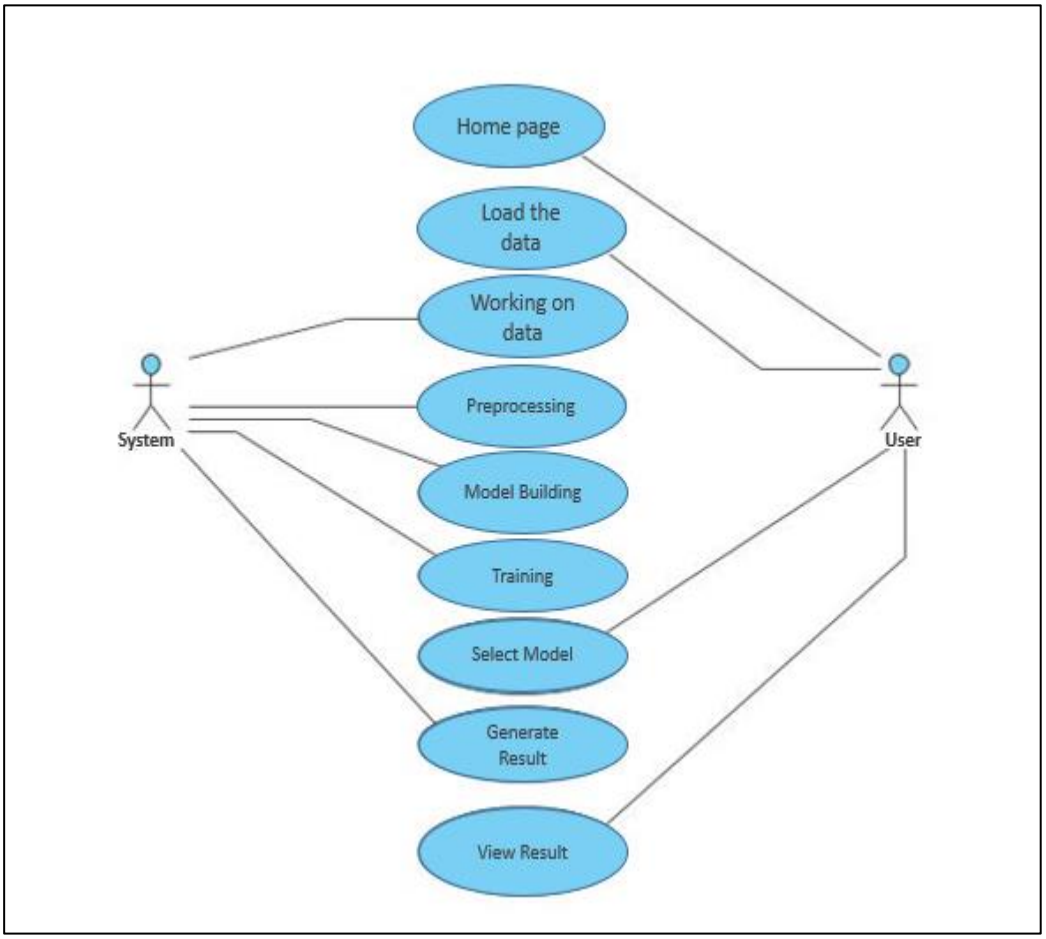


Fig.7.1 Use case Diagram

7.4.2 Class Diagram

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the

relationships among the classes. It explains which class contains information.

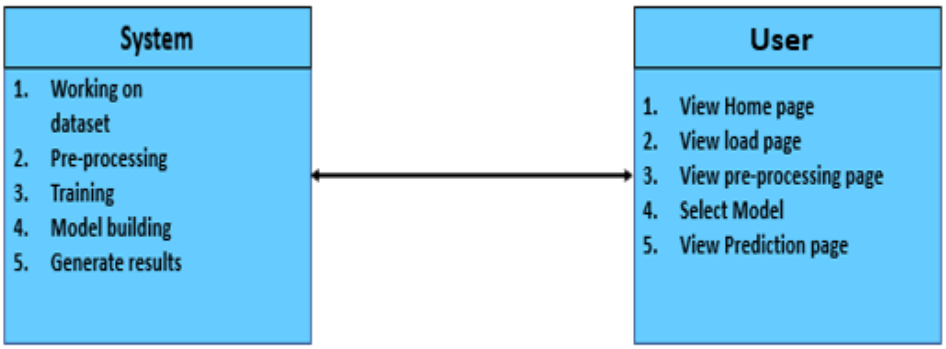


Fig.7.2 Class Diagram

7.4.3 Sequence Diagram

- A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order.
- It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

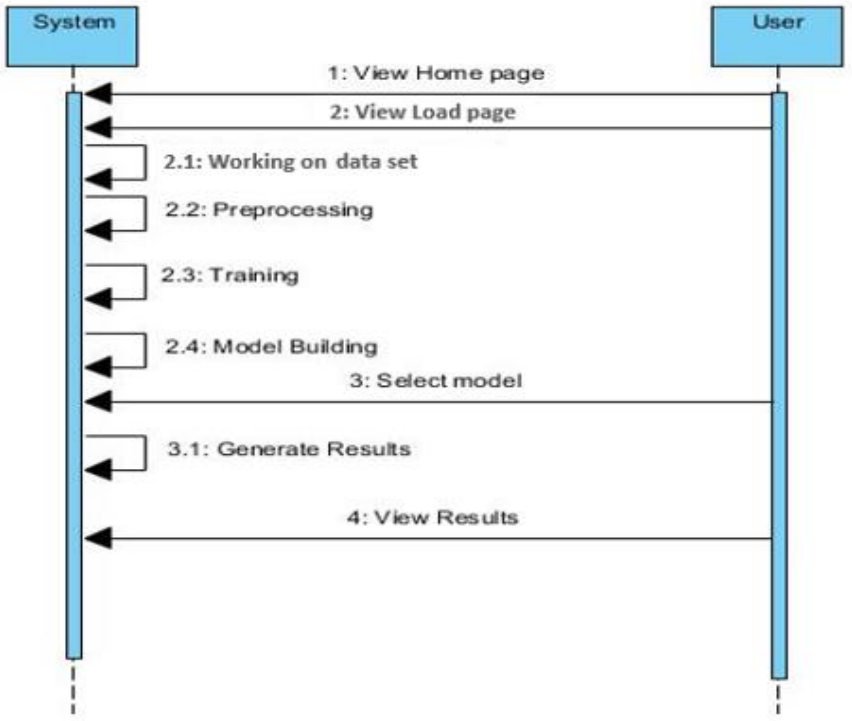


Fig.7.3 Sequence Diagram

7.4.4 Collaboration Diagram

In collaboration diagram the method call sequence is indicated by some numbering technique as shown below. The number indicates how the methods are called one after another. We have taken the same order management system to describe the collaboration diagram. The method calls are similar to that of a sequence diagram. But the difference is that the sequence diagram does not describe the object organization whereas the collaboration diagram shows the object organization.

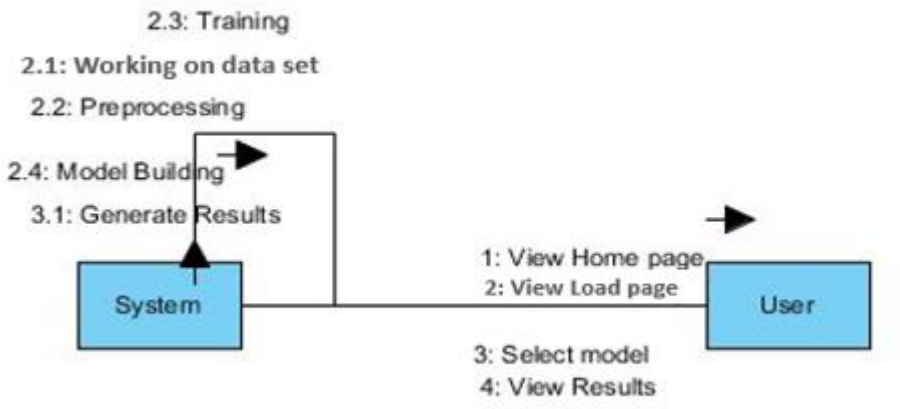


Fig.7.4 Collaboration Diagram

7.4.5 Deployment Diagram

Deployment diagram represents the deployment view of a system. It is related to the component diagram. Because the components are deployed using the deployment diagrams. A deployment diagram consists of nodes. Nodes are nothing but physical hardware's used to deploy the application.

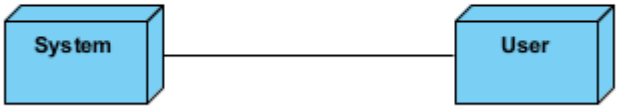


Fig.7.5 Deployment Diagram

7.4.6 Activity Diagram

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the

Unified Modelling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

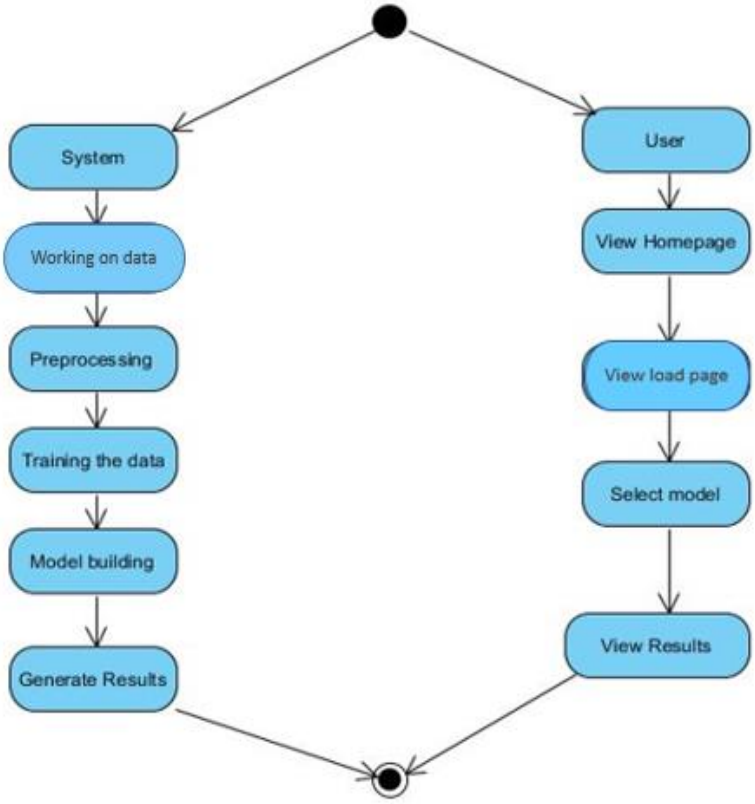


Fig.7.6 Activity Diagram

7.4.7 Component Diagram

A component diagram, also known as a UML component diagram, describes the organization and wiring of the physical components in a system. Component diagrams are often drawn to help model implementation details and double-check that every aspect of the system's required function is covered by planned development.

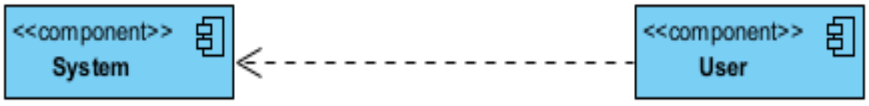


Fig.7.7 Component Diagram

7.4.8 ER Diagram

An Entity–relationship model (ER model) describes the structure of a database with the help of a diagram, which is known as Entity Relationship Diagram (ER Diagram). An ER model is a design or blueprint of a database that can later be implemented as a database. The main components of E-R model are: entity set and relationship set.

An ER diagram shows the relationship among entity sets. An entity set is a group of similar entities and these entities can have attributes. In terms of DBMS, an entity is a table or attribute of a table in database, so by showing relationship among tables and their attributes, ER diagram shows the complete logical structure of a database. Let's have a look at a simple ER diagram to understand this concept.

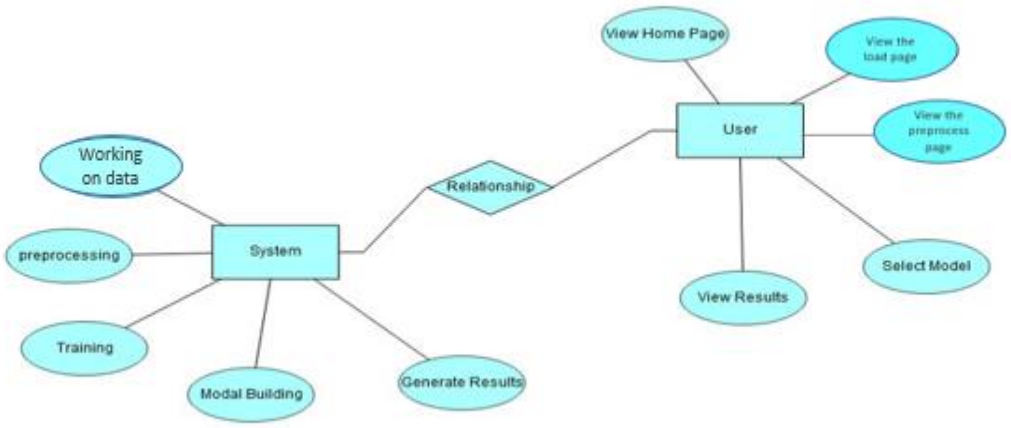


Fig.7.8 ER Diagram

7.4.9 DFD Diagram

A Data Flow Diagram (DFD) is a traditional way to visualize the information flows within a system. A neat and clear DFD can depict a good amount of the system requirements graphically. It can be manual, automated, or a combination of both. It shows how information enters and leaves the system, what changes the information and where information is stored. The purpose of a DFD is to show the scope and boundaries of a system as a whole. It may be used as a communications tool between a systems analyst and any person who plays a part in the system that acts as the starting point for redesigning a system.

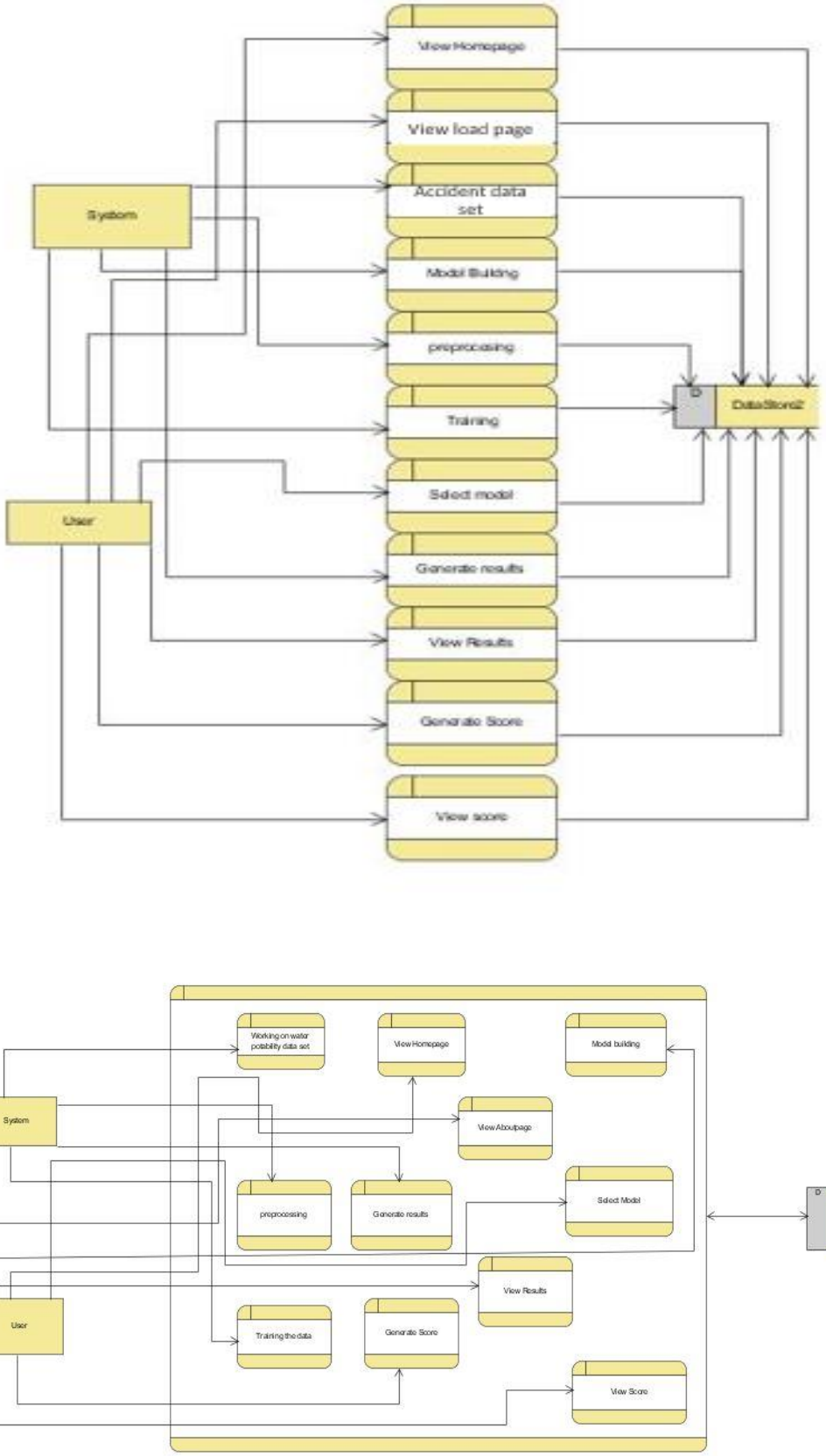


Fig.7.9 DFD Diagram

8. TESTING

8.1 Introduction

The main objective of testing is to uncover a host of errors, systematically and with minimum effort and time. Stating formally, we can say,

Testing is a process of executing a program with the intent of finding an error

- A successful test is one that uncovers an as yet undiscovered error.
- A good test case is one that has a high probability of finding error, if it exists.

The first approach is what is known as Black box testing and the second approach is White box testing. We apply white box testing techniques to ascertain the functionalities top-down and then we use black box testing techniques to demonstrate that everything runs as expected.

8.1.1 Black-Box Testing

This technique of testing is done without any knowledge of the interior workings of the application. The tester is oblivious to the system architecture and does not have access to the source code. Typically, while performing a black-box test, a tester will interact with the system‘s user interface by providing inputs and examining the outputs without knowing how and where the inputs are worked upon.

- Well suited and efficient for large code segments
- Code access is not required
- Clearly separates user‘s perspectives from the developer‘s perspective through visibly defined roles.

8.1.2 White-Box Testing

White-box testing is the detailed investigation of internal logic and structure of the code. It is also called ―glass testing‖ or ―open-box testing‖. In order to perform white box testing on an application, a tester needs to know the internal workings of the code.

The tester needs to look inside the source code and find out which part of the code is working inappropriately.

In this, the test cases are generated on the logic of each module. It has been uses to

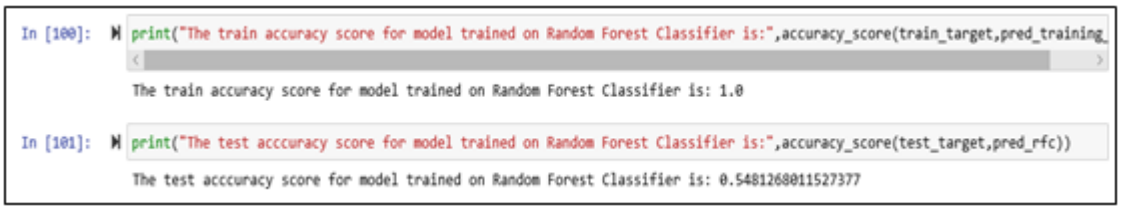
generate the test cases in the following cases:

- Guarantee that all independent modules have been executed.
- Execute all logical decisions and loops.
- Execute through proper plots and curves.

8.2 Performance Evaluation

Score method : It is a kind of method used to evaluate the performance of the model. Performance evaluation is made for this project using Score method of the sklearn library of Python.

The score method is applied for all the algorithms as follows:



Accuracy of the models of the algorithms are as follows:

Random Forest accuracy: 54.8%

XGBoost classifier accuracy: 66.5%

9. OUTPUT SCREEN SHOTS WITH DESCRIPTION

9.1 Home Page

Here user view the home page of personality prediction web application.

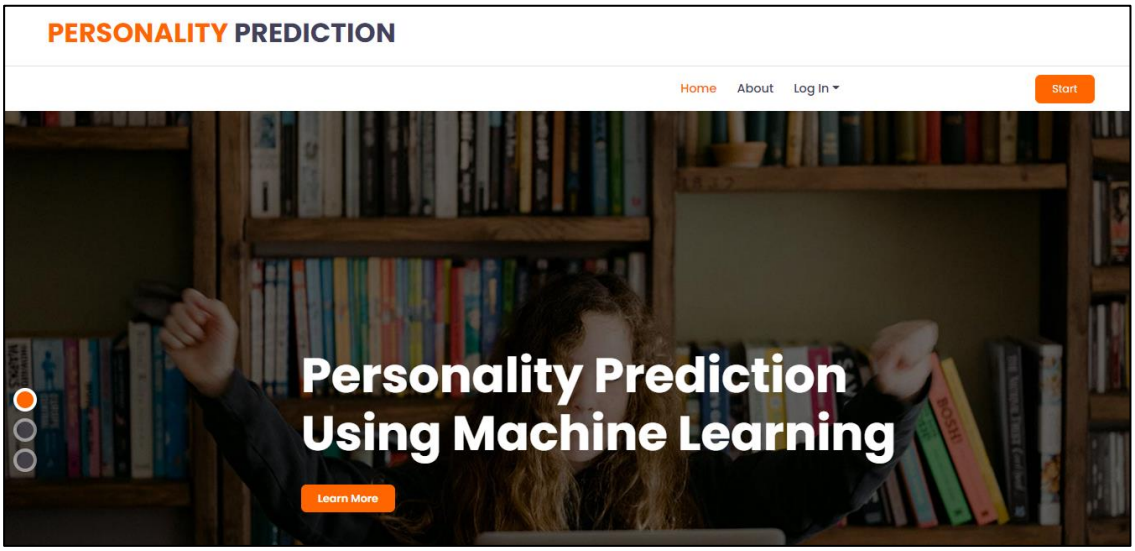


Fig.9.1 Home Page

9.2 About

Here we can read about our project.

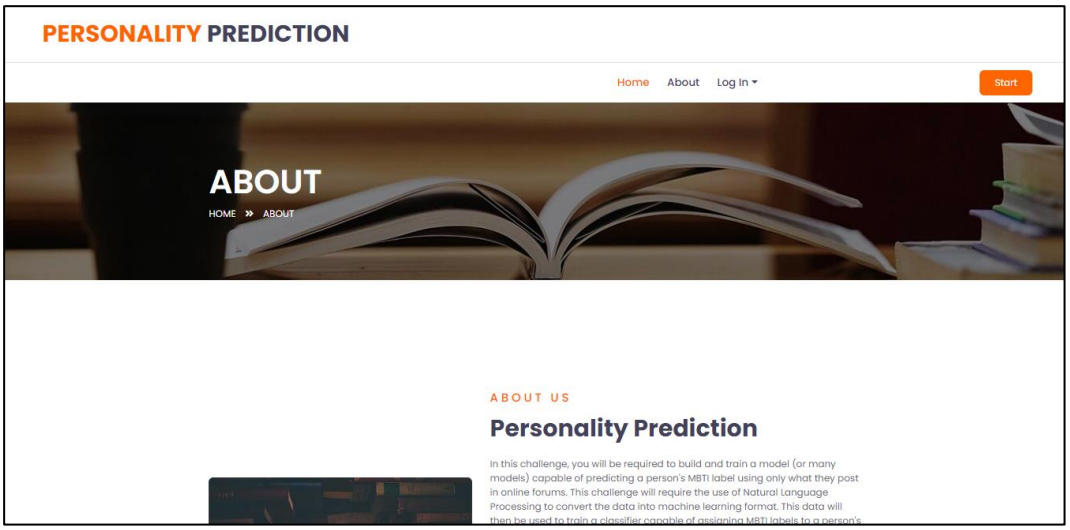


Fig.9.2 About page

9.3 Registration

In this page user can register him/herself by entering the details

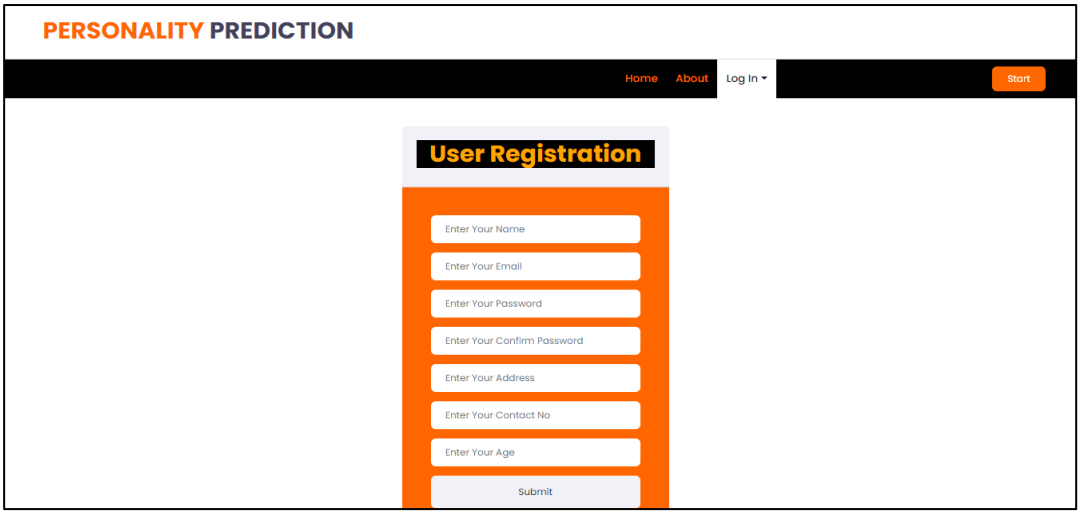


Fig.9.3 Registration page

9.4 Login

Here the user can log in with the valid credentials.

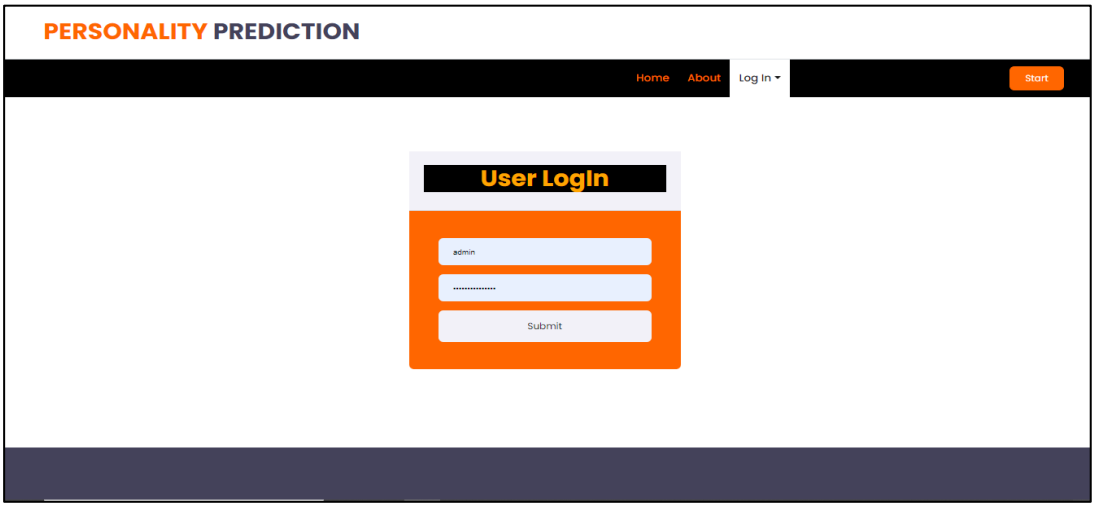


Fig.9.4 Login Page

9.5 User home

After successful login user will enter into the user home page

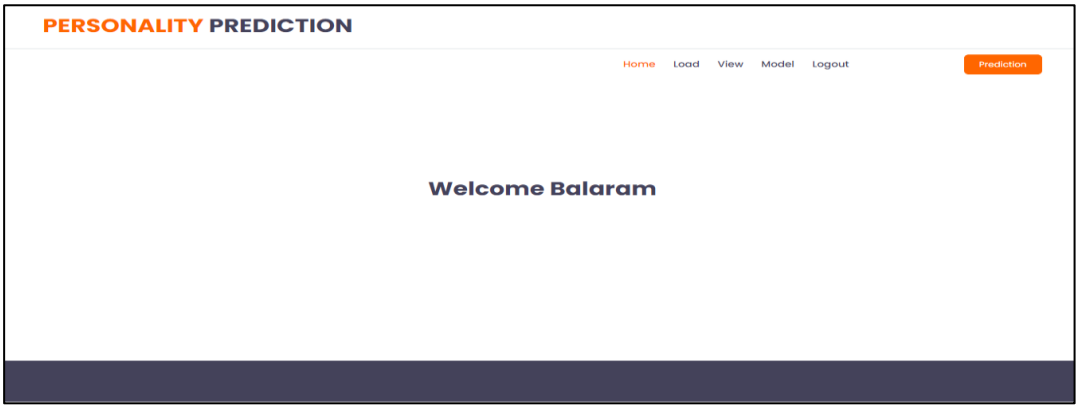


Fig.9.5 User home page

9.6 Load

In the load page, users can load the student dataset.

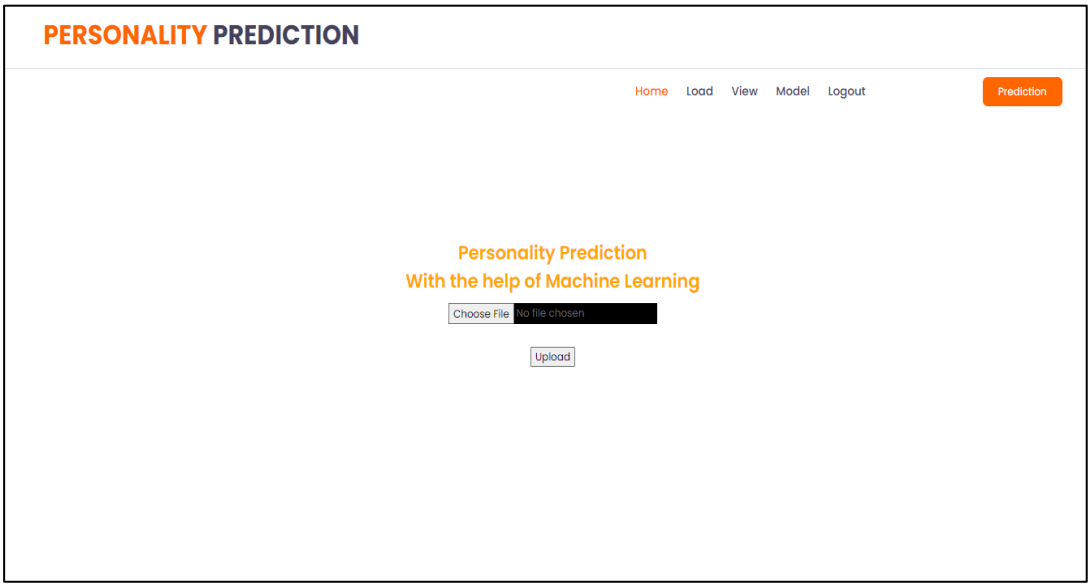


Fig.9.6 Load page

9.7 View:

Here we can see the uploaded data set.

Fig.9.7 View Page

9.8 Model:

Here we can train our data using different algorithm.

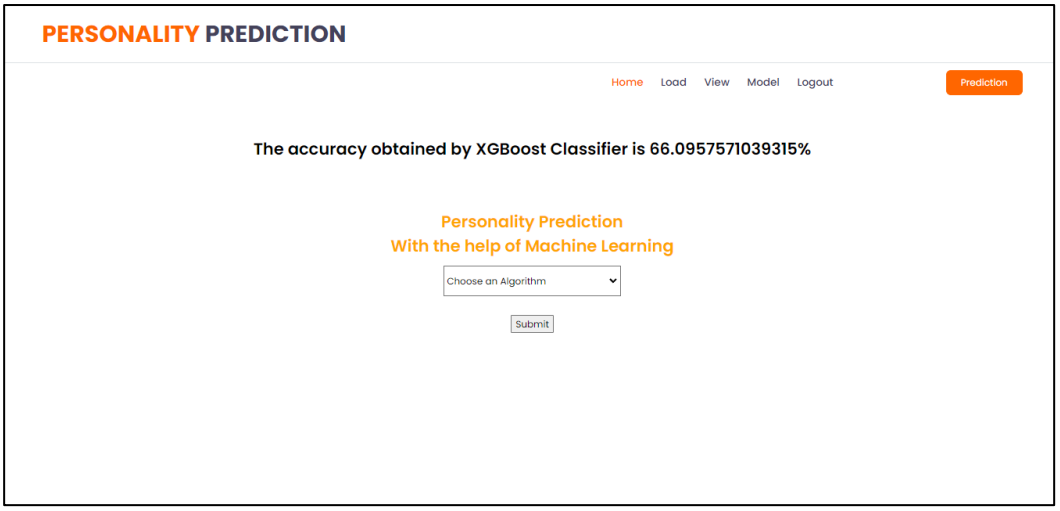


Fig.9.8 Model Page

9.9 Prediction

This page show the detection result of the personality prediction data.

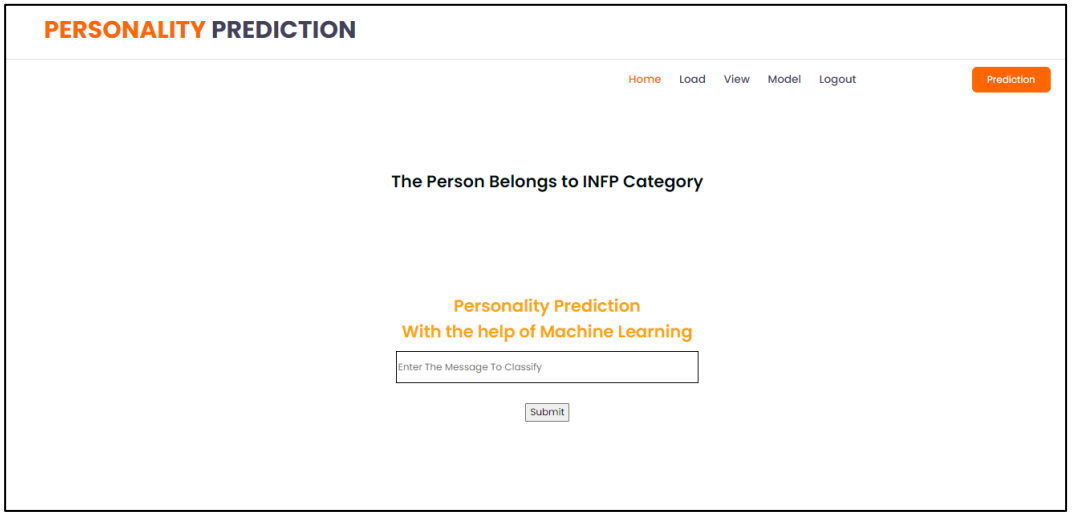


Fig.9.9 Prediction Page

10. CONCLUSION

In this proposed work, the system is able to gather the tweets of the user when given his twitter handle. The algorithms used are able to process the tweets and generate the required results. Finally the system is able to provide an efficiently accurate result by giving the personality type of the user.

By performing the analysis using all the algorithms, XGBoost provided the high accuracy for the trained model and predicts the personality of a person more accurately compared to the Linear regression and Random Forest algorithms.

The established system can be made more efficient by conducting more testing and feeding the system with a much accurate dataset. Once the system is highly accurate it can be used in corporations or even by the government to analyze the personalities of concerned individuals. The system can also be used in the crime sector. The currently implemented system of personality analysis can be extended and features like gender detection, age detection etc., can be added to it.

In this paper, Gradient boosting techniques to predict the personality is proposed. This technique converts weak learners to strong learners, but it also shows the right intuition of the data. Extreme Gradient Boosting (Xgboost) is a perfect algorithm for datasets ranging from medium to large. If the dataset contains millions of rows, this algorithm would not be suitable because of its use of ensemble trees. The depth and iterations are required to be increased in that case and the memory could be easily choked.

11. REFERENCES

[1] Firoj Alam, Evgeny A. Stepanov, Giuseppe Riccardi, "Personality Traits Recognition on Social Network - Facebook", AAAI Technical Report WS-13-01 Computational Personality Recognition (Shared Task)

[2] Basant Agarwal, "Personality Detection from Text: A Review", International Journal of Computer System (ISSN: XXXX – XXXX), Volume 01– Issue 01, September, 2014.

[3] Nadeem Ahmad, Jawaid Siddique, "Personality Assessment using Twitter Tweets", 21st International Conference on Knowledge Based and Intelligent Information and Engineering Systems, KES2017, 6-8 September 2017, Marseille, France.

[4] Shuotian Bai, Sha Yuana, Bibo Haoa and Tingshao Zhu, "Predicting personality traits of microblog users", Web Intelligence and Agent Systems: An International Journal 12 (2014) 249–265 249 DOI 10.3233/WIA-140295 IOS Press.

[5] Vanshika Varshney, Aman Varshney, Tameem Ahmad, Asad M. Khan, "Recognising Personality Traits using Social Media", IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI-2017) 978-1-5386-0814-2/17/$31.00 ©2017 IEEE.