

A project report on

PERSONALITY PREDICTION USING MACHINE LEARNING

Submitted in partial fulfillment for the award of the degree of

M.Tech (Software Engineering)

by

RAGUL S (19MIS0006)



VIT[®]

Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

**SCHOOL OF COMPUTER SCIENCE ENGINEERING AND
INFORMATION SYSTEMS**

November, 2023

DECLARATION

I here by declare that the thesis entitled “PERSONALITY PREDICTION USING MACHINE LEARNING” submitted by me, for the award of the degree of M.Tech (Software Engineering) is a record of bonafide work carried out by me under the supervision of Prof Deepa M.

I further declare that the work reported in this thesis has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Place: Vellore

Date:

Signature of the Candidate

CERTIFICATE

This is to clarify that these entitled “PERSONALITY PREDICTION USING MACHINE LEARNING” Submitted by RAGUL S (19MIS0006) School of Computer Science Engineering And Information Systems, Vellore Institute of Technology, Vellore for the award of the degree M.Tech (Software Engineering) is a record of bonafide work carried out by him/her under my supervision.

The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university. The Project report fulfils the requirements and regulations of VELLORE INSTITUTE OF TECHNOLOGY, VELLORE and in my opinion meets the necessary standards for submission.

Signature of the Guide

Signature of the HoD

Internal Examiner

External Examiner

ABSTRACT

Data Science and AI are revolutionizing the planet through technical transformations. We can observe many machine learning applications in our day-to-day lives, but one of the greatest applications of machine learning is to classify individuals based on their personality traits. Every person on the planet is unique and carries a unique personality type. The availability of a high- dimensional data has paved the way for increasing marketing campaigns' effectiveness by targeting specific groups of people. Such personality-based communications are highly effective in increasing the recognition and attractiveness of products and services. Developed a system for personality prediction using personality traits in this project. Daily lot of users using the youtube for the various purpose, we will take the data from the youtube comment page and will classify the personality of the person. From the personality classification, the person can view the type of personality and can improve the personality based upon the results.

ACKNOWLEDGEMENT

It is my pleasure to express with deep sense of gratitude to Prof Deepa M, Associate Professor Sr, School of Computer Science Engineering and Information Systems, Vellore Institute of Technology, for his/her constant guidance, continual encouragement, understanding; more than all, he taught me patience in my endeavor. My association with him is not confined to academics only, but it is a great opportunity on my part of work with an intellectual and expert in the field of Software Engineering.

I would like to express my gratitude to DR.G.VISWANATHAN, Chancellor VELLORE INSTITUTE OF TECHNOLOGY, VELLORE, MR. SANKAR VISWANATHAN, DR. SEKAR VISWANATHAN, MR.G V SELVAM, Vice – Presidents VELLORE INSTITUTE OF TECHNOLOGY, VELLORE, Dr. V. S. Kanchana Bhaaskaran, I/c Vice – Chancellor, DR. Partha Sharathi Mallick, Pro-Vice Chancellor and Dr. S. Sumathy, Dean, School of Computer Science Engineering And Information Systems,, for providing with an environment to work in and for his inspiration during the tenure of the course.

In jubilant mood I express ingeniously my whole-hearted thanks to Dr. Neelu Khare, HoD/Professor, all teaching staff and members working as limbs of our university for their not-self-centered enthusiasm coupled with timely encouragements showered on me with zeal, which prompted the acquirement of the requisite knowledge to finalize my course study successfully. I would like to thank my parents for their support.

It is indeed a pleasure to thank my friends who persuaded and encouraged me to take up and complete this task. At last but not least, I express my gratitude and appreciation to all those who have helped me directly or indirectly toward the successful completion of this project.

Place: Vellore

Date:

Name of the student

TABLE OF CONTENTS

LIST OF FIGURES.....	x
LIST OF TABLES.....	xii
LIST OF ACRONYMS.....	xiii

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND.....	2
1.2 MOTIVATION.....	2
1.3 PROJECT STATEMENT.....	3
1.4 OBJECTIVES.....	3
1.5 SCOPE OF THE PROJECT.....	4
1.6 GENERAL.....	5
1.6.1 MACHINE LEARNING SYSTEM.....	5
1.6.2 JUPYTER.....	5
1.6.3 MACHINE LEARNING.....	6
1.6.4 SCIKIT LEARN.....	6
1.6.5 CLUSTERING.....	7
1.7 CLASSIFICATION.....	7
1.7.1 DIMENSIONALITY REDUCTION.....	8

CHAPTER 2

LITERATURE SURVEY

2.1 SUMMARY OF THE EXISTING WORK.....	11
2.2 CHALLENGES PRESENT IN EXISTING SYSTEM.....	16

CHAPTER 3

REQUIREMENTS

3.1 HARDWARE REQUIREMENTS.....	17
3.2 SOFTWARE REQUIREMENTS.....	17
3.3 GANTT CHART.....	18

CHAPTER 4

ANALYSIS & DESIGN

4.1 PROPOSED METHODOLOGY.....	19
4.2 SYSTEM ARCHITECTURE.....	19
4.3 MODULE DESCRIPTIONS.....	23
4.3.1 DATASET COLLECTION.....	23
4.3.2 DATA PREPROCESSING.....	24
4.3.3 DATA FORMATTING.....	24
4.3.4 DATA CLEANING.....	24
4.3.5 NATURAL LANGUAGE PROCESSING.....	25

4.3.6 FEATURIZATION.....	26
4.3.7 SPLITTING OF DATA.....	27
4.3.8 MODELING OF EVALUATION.....	27
4.4 ALGORITHMS USED.....	29

CHAPTER 5

IMPLEMENTATION & TESTING

5.1 SAMPLE CODE.....	33
5.2 SAMPLE OUTPUT.....	53

CHAPTER 6

RESULTS

6.1 RESULT ANALYSIS.....	63
6.2 EVALUATION METRICS.....	64

CONCLUSION AND FUTURE WORKS.....65

REFERENCES.....66

LIST OF FIGURES

1.1 JUPYTER.....	5
1.2 MACHINE LEARNING.....	6
1.3 CLUSTERING.....	7
1.4 DIMENSIONALITY REDUCTION.....	9
1.5 DIMENSIONALITY REDUCTION(LOGISTIC REGRESSION).....	10
3.1 GANTT CHART.....	18
4.1 SYSTEM ARCHITECTURE.....	19
4.2 USECASE DIAGRAM.....	20
4.3 CLASS DIAGRAM.....	21
4.4 SEQUENCE DIAGRAM.....	21
4.5 ACTIVITY DIAGRAM.....	22
4.6 E-R DIAGRAM.....	22
4.7 DATAFLOW DIAGRAM.....	23
4.8 CONFUSION MATRIX.....	28
4.9 RANDOMFOREST CLASSIFIER.....	30
4.10 DECISION TREE.....	31
5.1 SAMPLE OUTPUT(FRONTEND).....	53
5.2 SAMPLE OUTPUT(FRONTEND).....	53
5.3 SAMPLE OUTPUT(FRONTEND).....	54

5.4 SAMPLE OUTPUT(BACKEND).....	54
5.5 SAMPLE OUTPUT(BACKEND).....	55
5.6 SAMPLE OUTPUT(BACKEND).....	55
5.7 SAMPLE OUTPUT(BACKEND).....	56
5.8 SAMPLE OUTPUT(BACKEND).....	56
5.9 SAMPLE OUTPUT(BACKEND).....	57
5.10 SAMPLE OUTPUT(BACKEND).....	57
5.11 SAMPLE OUTPUT(BACKEND).....	58
5.12 SAMPLE OUTPUT(BACKEND).....	58
5.13 SAMPLE OUTPUT(BACKEND).....	59
5.14 SAMPLE OUTPUT(BACKEND).....	59
5.15 SAMPLE OUTPUT(BACKEND).....	60
5.16 SAMPLE OUTPUT(BACKEND).....	60
5.17 SAMPLE OUTPUT(BACKEND).....	61
5.18 SAMPLE OUTPUT(BACKEND).....	61
5.19 SAMPLE OUTPUT(BACKEND).....	62
5.20 SAMPLE OUTPUT(BACKEND).....	62
6.1 EVALUATION METRICS.....	64

LIST OF TABLES

2.1 SUMMARY OF THE EXISTING WORKS.....	11
--	----

LIST OF ACRONYMS

ML	MACHINE LEARNING
AL	ARTIFICIAL INTELLIGENCE
PP	PERSONALITY PREDICTION
PT	PERSONALITY TRAITS

CHAPTER 1

INTRODUCTION

Introversion, extroversion, intuition, sensing, thinking, feeling, judging, and perceiving are the personality qualities. Differentiation in personality and decision-making are explained by these pairings of personality traits, which cover a wide range of human behaviour. Currently, the model is used by HR professionals to assess new employees and by marketers to comprehend the target markets for their products. In this project, we are utilising the personality model to construct the algorithm. Receptivity to new things: This personality quality, which is also known as intellect or imagination, stands for the readiness to try new things and think beyond the box. Insightfulness, inventiveness, and curiosity are characteristics of this feature.

Conscientiousness is the urge to use caution, diligence, and self-control when pursuing immediate gratification. This quality involves determination, self-control, reliability, and consistency.

Extroversion: The tendency to seek out social interaction and to take energy from other people rather than spending time alone. This quality involves being gregarious, enthusiastic, and self-assured.

Agreeableness: A person's capacity for empathy and cooperation, which are indicators of their ability to relate with others. Inherent in this quality are tact, gentleness, and loyalty.

The propensity for negative personality traits, emotional instability, and self-destructive thought is known as neuroticism. Pessimism, worry, insecurity, and fearfulness are characteristics of this trait.

1.1 BACKGROUND

In today's digital age, data science and artificial intelligence have become instrumental in deciphering human behavior and preferences. One prominent application of these technologies lies in the realm of personality prediction, which has significant implications for targeted marketing and personal development. With the exponential growth of online platforms like YouTube, analyzing user behavior and personalities through their comments has become an invaluable tool for understanding consumer preferences and tailoring marketing strategies. This project focuses on harnessing the power of machine learning and natural language processing to predict users' personalities based on their comments on YouTube. By leveraging high-dimensional data and advanced algorithms, the system aims to categorize individuals into distinct personality types, providing them with valuable insights into their own behavioral tendencies and characteristics. The ultimate goal is to enhance user engagement by tailoring marketing campaigns to specific personality types, thereby increasing the recognition and attractiveness of products and services. Moreover, by offering users personalized insights into their own personalities and suggesting methods for self-improvement, the project seeks to foster personal growth and development within the online community. Through this innovative approach, the project not only contributes to the effectiveness of marketing strategies but also empowers users to better understand themselves and work towards enhancing their personal attributes and qualities.

1.2 MOTIVATION

The motivation behind this project stems from the growing demand for personalized user experiences and targeted marketing strategies in the dynamic realm of digital media. By delving into the personalities and preferences of users through their comments on YouTube, the project aims to create a more engaging and tailored online environment. With a focus on revolutionizing marketing approaches, the project seeks to leverage personality prediction to customize communication strategies, leading to improved conversion rates and heightened brand loyalty. Moreover, the project is driven by the aspiration to empower users

to understand their own personalities better and to encourage personal growth and development within the digital community. By using advanced data science methodologies, the project aims to provide businesses with valuable insights into user preferences and behaviors, enabling data-driven decision-making for more effective marketing and user engagement strategies. Through the alignment of content and services with user personalities, the project ultimately strives to enhance customer satisfaction and foster lasting connections between businesses and their target audience.

1.3 PROJECT STATEMENT:

Developed a system using a machine learning technique known as Random forest and Decision Tree. This system performs a psychometric analysis and predicts the personality of applicants based on the analysis of their YouTube comments. The project involves the use of Natural Language Processing (NLP) techniques to analyze the sentiments and emotional content conveyed in the YouTube comments of users. By leveraging Random Forest and Decision Tree classifiers, the system can effectively estimate emotional aptitude and predict personality traits based on the analysis of users' comments. This approach enables a comprehensive understanding of the emotional aspects and psychometric profiles of the individuals, aiding in the evaluation of their suitability for specific roles or positions.

1.4 OBJECTIVE:

In our project, we suggest employing machine learning algorithms for personality assessment. For decades, psychology researchers have worked to understand personality in a systematic way. After extensive work to develop and validate a widely accepted personality model, researchers have shown connections between general personality traits and many types of behaviour. Relationships have been discovered between personality and psychological disorders This paper attempts to bridge the gap between social media and personality research by using

the information people reveal in their online profiles. Our core research question asks whether social media profiles can predict personality traits. If so, then there is an opportunity to integrate the many results on the implications of personality factors and behaviour into the users' online experiences and to use social media profiles as a source of information to better understand individuals. For example, the friend suggestion system could be tailored to a user based on whether they are more introverted or extraverted.

1.5 SCOPE OF THE PROJECT

Youtube comments can reflect upon the personality qualities of a person. Personality is one of the vital factors which suggests how a person would be able to work in a designated role, hence personality analysis and understanding is key. Our objective doing this project is to make the machine more human, and analyse the candidate in such a way that an actual human reviewer would. Our project's primary goal is to make personality predictions based on youtube comments. Personality has been shown to be relevant to many types of interactions; it has been shown to be useful in predicting job satisfaction, professional and romantic relationship success, and even preference for different interfaces. Until now, to accurately gauge users' personalities, they needed to take a personality test. This made it impractical to use personality analysis in many social media domains. In this paper, we present a method by which a user's personality can be accurately predicted through the publicly available information on their youtube comments.


1.6 GENERAL

1.6.1 MACHINE LEARNING SYSTEM

Machine learning is a branch of artificial intelligence (AI) and computer science which focuses on the use of data and algorithms to imitate the way that humans learn, gradually improving its accuracy. Over the last couple of decades, the technological advances in storage and processing power have enabled some innovative products based on machine learning, such as Netflix's recommendation engine and self-driving cars. Machine learning is an important component of the growing field of data science. Through the use of statistical methods, algorithms are trained to make classifications or predictions, and to uncover key insights in data mining projects. These insights subsequently drive decision making within applications and businesses, ideally impacting key growth metrics. As big data continues to expand and grow, the market demand for data scientists will increase. They will be required to help identify the most relevant business questions and the data to answer them.

1.6.2 JUPYTER

Jupyter, previously known as IPython Notebook, is a web-based, interactive development environment. Originally developed for Python, it has since expanded to support over 40 other programming languages including Julia and R. Jupyter allows for notebooks to be written that contain text, live code, images, and equations. These notebooks can be shared, and can even be hosted on GitHub for free. For each section of this tutorial, you can download a Jupyter notebook that allows you to edit and experiment with the code and examples for each topic. Jupyter is part of the Anaconda distribution; it can be started from the command line using the Jupyter command:



```
$ jupyter notebook
```

Fig 1.1

1.6.3 MACHINE LEARNING

We will now move on to the task of machine learning itself. In the following sections we will describe how to use some basic algorithms, and perform regression, classification, and clustering on some freely available medical datasets concerning breast cancer and diabetes, and we will also take a look at a DNA microarray dataset.

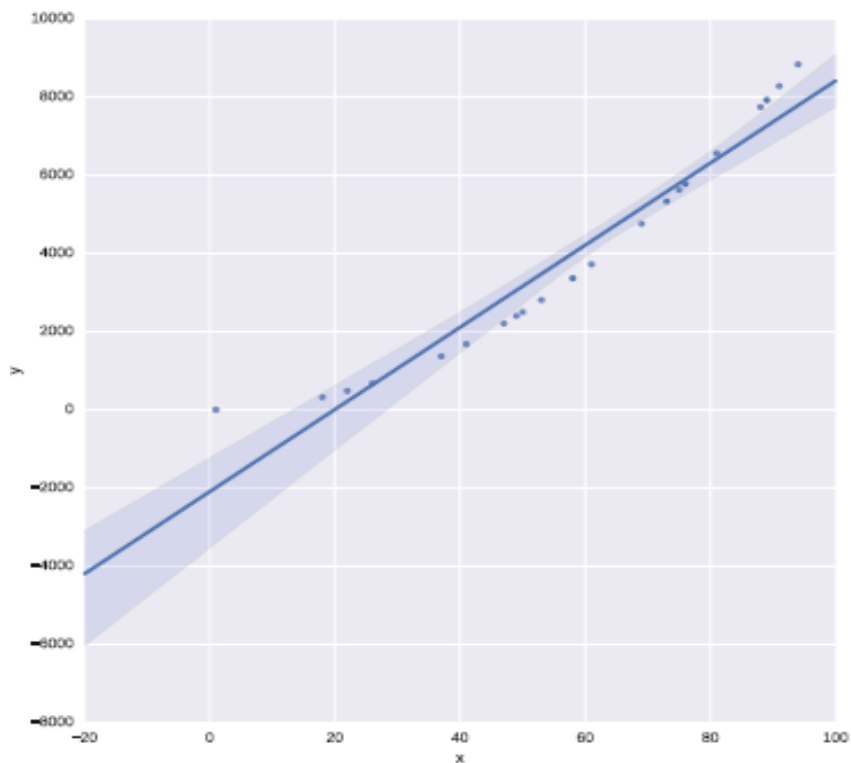


Fig 1.2

1.6.4 SCIKIT-LEARN

SciKit-Learn provides a standardized interface to many of the most commonly used machine learning algorithms, and is the most popular and frequently used library for machine learning for Python. As well as providing many learning algorithms, SciKit-Learn has a large number of convenience functions for common preprocessing tasks (for example, normalization or k-fold cross validation). SciKit-Learn is a very large software library.

1.6.5 CLUSTERING

Clustering algorithms focus on ordering data together into groups. In general clustering algorithms are unsupervised they require no y response variable as input. That is to say, they attempt to find groups or clusters within data where you do not know the label for each sample. SciKit-Learn have many clustering algorithms, but in this section we will demonstrate hierarchical clustering on a DNA expression microarray dataset using an algorithm from the SciPy library. We will plot a visualization of the clustering using what is known as a dendrogram, also using the SciPy library. The goal is to cluster the data properly in logical groups, in this case into the cancer types represented by each sample's expression data. We do this using agglomerative hierarchical clustering, using Ward's linkage method:

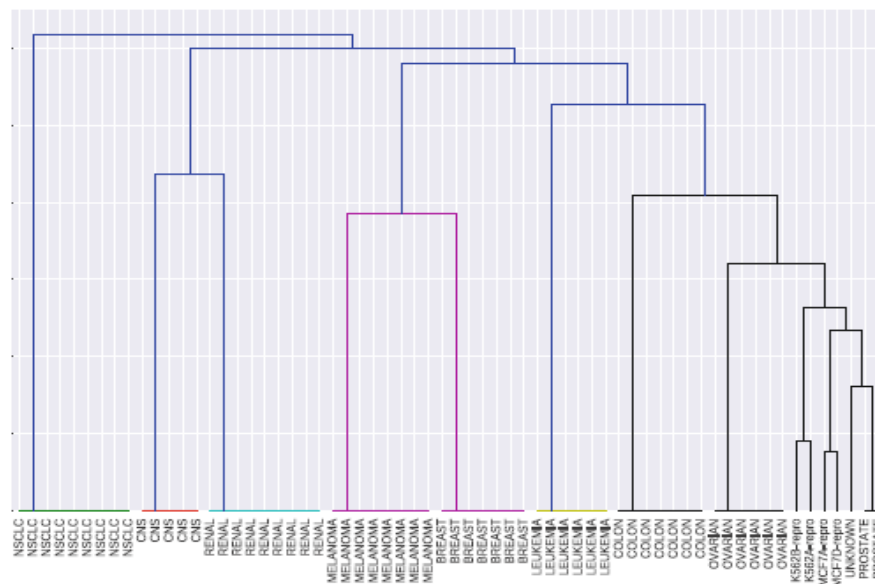


Fig 1.3

1.7 CLASSIFICATION

We analyzed data that was unlabeled, we did not know to what class a sample belonged (known as unsupervised learning). In contrast to this, a supervised problem deals with labelled data where are aware of the discrete

classes to which each sample belongs. When we wish to predict which class a sample belongs to, we call this a classification problem. We will work on the Youtube comments dataset, split it into a training set and a test set, train a Random Forest and test the trained model on an unseen dataset. Random Forest model should be able to predict if a new sample is malignant or benign based on the features of a new, unseen sample. You observed that the Random Forest model performed well in predicting the malignancy of new, unseen samples from the test set. This performance was quantified using various metrics, such as the classification report function, which includes precision, recall, and F1 score (where $F1 = 2 * \text{precision} * \text{recall} / (\text{precision} + \text{recall})$) for each class, along with the support column indicating the count of samples for each class. Random Forest and Decision Trees are powerful tools for classification tasks. They can handle high-dimensional data effectively, making them suitable for complex datasets. However, if the dataset size increases significantly, the training time may also increase notably, affecting scalability. Additionally, in cases where the dataset consists of a large number of features, visualization and plotting of the data can be challenging. To address this issue, the application of dimensionality reduction techniques can be beneficial for visualizing highly dimensional data.

1.7.1 DIMENSIONALITY REDUCTION

Another important method in machine learning, and data science in general, is dimensionality reduction. For this example, we will look at Youtube comments dataset once again. The dataset consists of over 500 samples, where each sample has 30 features. The features relate to text related to comments on youtube, and the features describe the personality traits present in the comments. All features are real values. The target variable is a discrete value and is therefore a classification dataset.

You will recall from the Iris example in Sect. 7.3 that we plotted a scatter matrix of the data, where each feature was plotted against every other feature in the dataset to look for potential correlations (Fig. 3). By examining this plot you could probably find features which would separate the dataset into groups.

Because the dataset only had 4 features we were able to plot each feature against each other relatively easily. However, as the numbers of features grow, this becomes less and less feasible, especially if you consider the gene expression example in Sect. 9.4 which had over 6000 features.

One method that is used to handle data that is highly dimensional is Principle Component Analysis, or PCA. PCA is an unsupervised algorithm for reducing the number of dimensions of a dataset. For example, for plotting purposes you might want to reduce your data down to 2 or 3 dimensions, and PCA allows you to do this by generating components, which are combinations of the original features, that you can then use to plot your data.

PCA is an unsupervised algorithm. You supply it with your data, \mathbf{X} , and you specify the number of components you wish to reduce its dimensionality to. This is known as transforming the data:

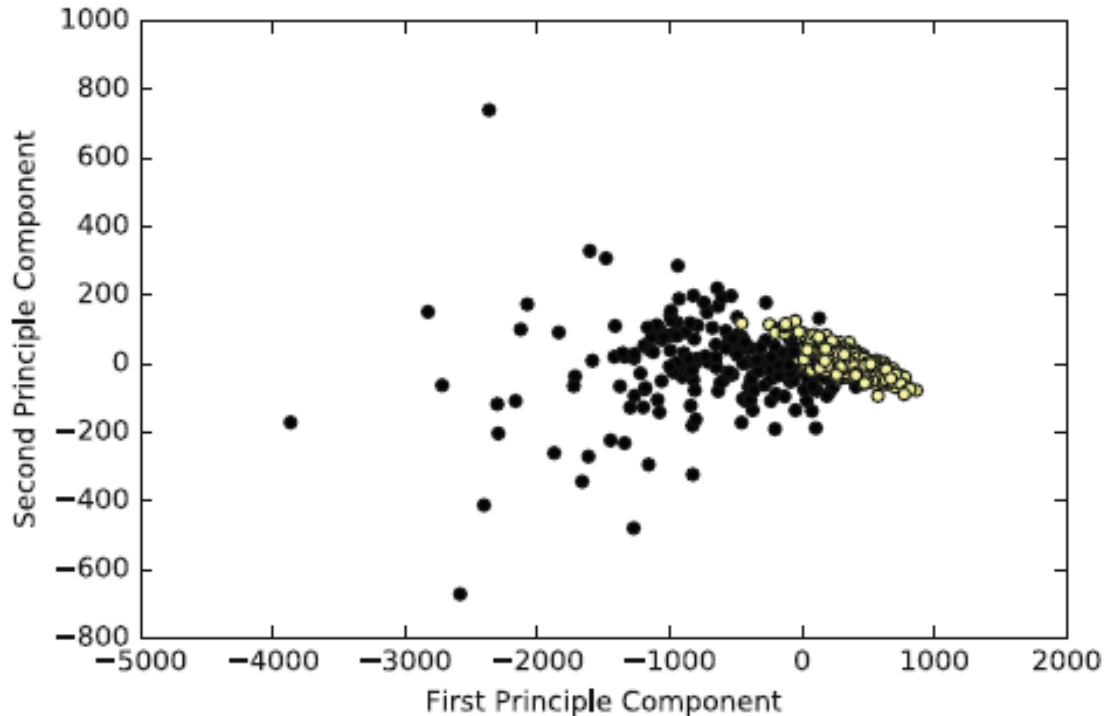


Fig 1.4

Again, you would not use this model for new data in a real world scenario, you would, for example, perform a 10-fold cross validation on the dataset, choosing the model parameters that perform best on the cross validation. This model would be much more likely to perform well on new data. At the very least, you would randomly select a subset, say 30% of the data, as a test set and train the model on the remaining 70% of the dataset. You would evaluate the model based on the score on the test set and not on the training set

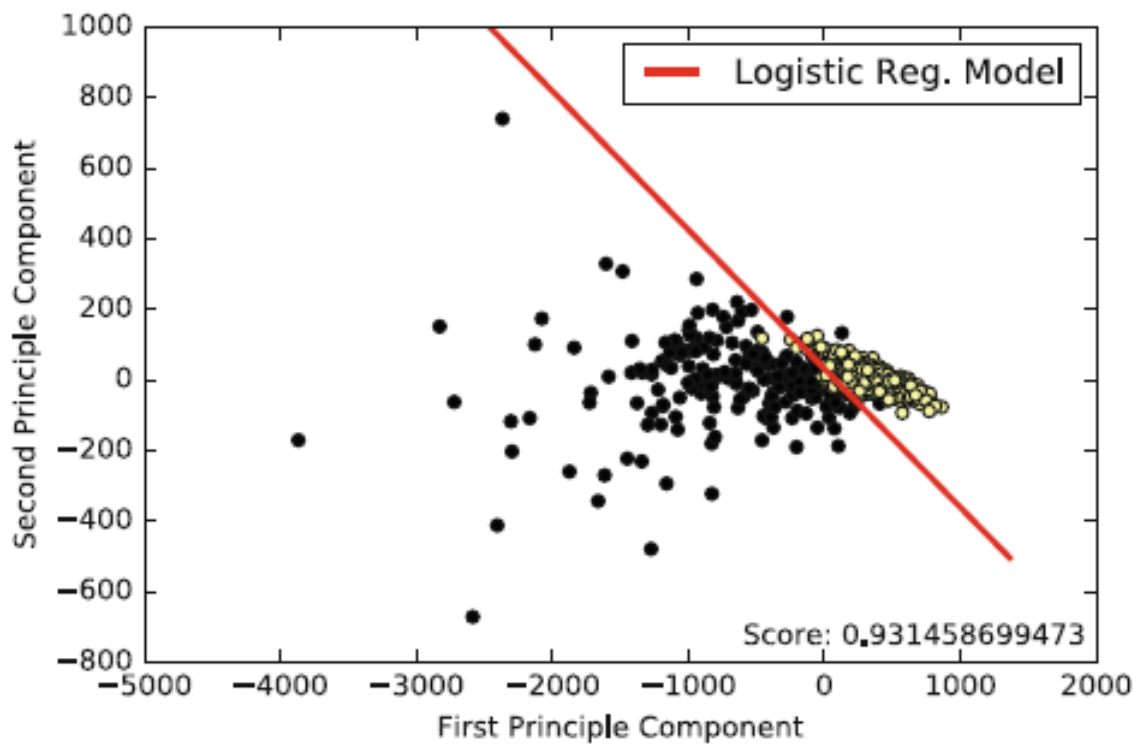


Fig 1.5

CHAPTER 2

LITERATURE SURVEY

2.1 SUMMARY OF THE EXISITING WORKS

S.No	Paper Title	Methodology	Merits	Demerits
1	Personality prediction using Machine Learning YEAR: 2022 AUTHORS: Devesh Agarwal , Mr. M. Karthikeyan	Uses a Youtube comment to collect data on the Big Five personality traits: extraversion, agreeableness, openness, conscientiousness, and neuroticism. The data is then analyzed using Random forest,K-means clustering and Logistic Regression to predict the personality type of the individual.	Rich Data Source, Accessibility, Potential for Insight	YouTube comments may contain noise, irrelevant information, or biased expressions, affecting the quality of the data, ethical considerations, Subjectivity

2	<p>Personality prediction through CV analysis using machine learning techniques</p> <p>YEAR: 2023</p> <p>AUTHORS:Nongmeikapam Thoiba Singh; Abhay Chanana; Darpan Jain; Rajesh Kumar</p>	<p>Uses machine learning algorithms like Random Forest, Naïve Bayes, Support Vector Machine, Decision tree, XG Boost to parse the information in CVs and resumes and to predict the personality of the applicant.</p>	<p>Efficient Data Analysis, Accuracy, Streamlined Recruitment Process</p>	<p>May not be accurate for all candidates</p>
3	<p>Personality Prediction through Resume analysis using Machine Learning</p> <p>YEAR: 2021</p> <p>AUTHORS: Atharva Kulkarni, Tanuj Shankarwar , Siddharth Thorat</p>	<p>The Resume is first analyzed using NLP techniques to extract the keywords that are related to personality traits. These keywords are then used to train a machine learning algorithm, such as Random Forest, to predict the personality of the candidate.</p>	<p>Can be used to screen a large number of candidates quickly and efficiently.</p>	<p>Can be biased, as the personality traits that are considered desirable may vary</p>

4	<p>Personality classification using machine learning</p> <p>YEAR: 2022</p> <p>AUTHORS: H.Vijay, N.Sebastian</p>	<p>The paper combines k-means clustering and random forest algorithms to classify individuals' personalities based on the Big Five model. K-means clusters respondents using unsupervised learning, while the random forest enhances predictive accuracy. This hybrid approach aims to reveal patterns in personalities more effectively, offering practical applications like personalized career guidance. The k-means algorithm categorizes individuals, and the random forest</p>	<p>Enhanced Predictive Accuracy, Revealing Effective Patterns, Insights into Feature Importance</p>	<p>Complexity, Data Sensitivity, Resource Intensive</p>
---	---	---	---	---

		provides insights into feature importance, enhancing overall robustness.		
5	Prediction of Personality Trait using Machine Learning on Online Texts YEAR: 2022 AUTHORS: R. K. Cherukuru, A. Kumar, S. Srivastava and V. Kumar Verma,	The study employs XGBoost and Random forest for predicting MBTI-associated personality traits from social media text, utilizing oversampling to address dataset skewness. Pre-processing techniques, including tokenization, word stemming, stop words elimination, and TF-IDF feature selection, enhance personality exploration.	The use of XGBoost and Random Forest algorithms enhances the accuracy of predicting MBTI-associated personality traits from social media text, and it has a Effective Pre-processing Techniques	The use of advanced algorithms and pre-processing techniques may introduce complexity in the implementation and interpretation of the personality prediction model, requiring expertise in both machine learning and natural language processing
6	A Novel Approach to Predict Personality of a Person YEAR: 2021	Uses decision tree and Naïve Bayes personality prediction tests to determine	Can automate candidate pre-screening	Can be biased

	AUTHORS: Prof. Waheeda Dhokley, Randeria Kaiwan Jehangir, Shaikh Nabeel Rashid, Shaikh Almas Mohd Sarwar.	candidate's personality traits		
7	Smart-Hire Personality Prediction Using ML YEAR: 2023 AUTHORS: Isha Gupta; Manasvi Jain; Prashant Johri	Uses random forest, XGBoost machine learning algorithms to analyze text data, such as social media posts or blog articles, to predict a person's personality traits.	Can be used to target specific demographics for marketing campaigns	May not be accurate for all people
8	Applicant Personality Prediction System Using Machine Learning YEAR: 2021 AUTHORS: M. Karnakar; Haseeb Ur Rahman; A B Jai Santhosh; NageswaraRao Sirisala	Uses random forest, decision tree, and svm to parse the information in CVs and resumes, and personality tests to predict the personality of the applicant.	Can help organizations find the right candidates for the job more efficiently	Implementing and fine-tuning multiple algorithms may require significant computational resources and expertise, potentially limiting practical applicability in certain contexts

9	<p>Mental Health and Personality Determination using Machine Learning</p> <p>YEAR: 2022</p> <p>AUTHORS: Kimaya Raut; Jui Patil; Siddhi Wade; Jisha Tinsu</p>	<p>Uses Decision tree and XGBoost to train on textual comments collected from users via a web application.</p>	<p>Can predict mental health and personality with high accuracy</p>	<p>Implementing and interpreting the results of multiple algorithms may introduce complexity, requiring expertise in machine learning and data analysis</p>
10	<p>Personality trait prediction based on game character design using machine learning approach</p> <p>YEAR: 2017</p> <p>AUTHORS: L. K. P. Suryapranata, G. P. Kusuma, Y. Heryadi, B. S. Abbas, Lukas and A. S. Ahmad</p>	<p>Uses Decision tree and Naïve bayes to predict personality traits based on the game character design chosen by the player.</p>	<p>Can predict personality traits of game players</p>	<p>Accuracy may not be high for all personality traits</p>

Fig 2.1

2.2 CHALLENGES PRESENT IN EXISTING SYSTEM:

- Methods have performance limitations because of wide range of variations in data and amount of data is limited.
- Less accuracy

CHAPTER 3

REQUIREMENTS

3.1 HARDWARE REQUIREMENTS

The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system. They are used by software engineers as the starting point for the system design. It shows what the system does and not how it should be implemented

PROCESSOR : INTEL I5

RAM : 4GB

HARDDISK : 50GB

3.2 SOFTWARE REQUIREMENTS

The software requirements document is the specification of the system. It should include both a definition and a specification of requirements. It is a set of what the system should do rather than how it should do it. The software requirements provide a basis for creating the software requirements specification. It is useful in estimating cost, planning team activities, performing tasks and tracking the team's and tracking the team's progress throughout the development activity.

PYTHON IDE : ANACONDA JUPYTER NOTEBOOK

BACK END : JUPYTER NOTEBOOK

FRONT END : DJANGO FRAME

PROGRAMMING LANGUAGE : PYTHON

3.3 GANTT CHART

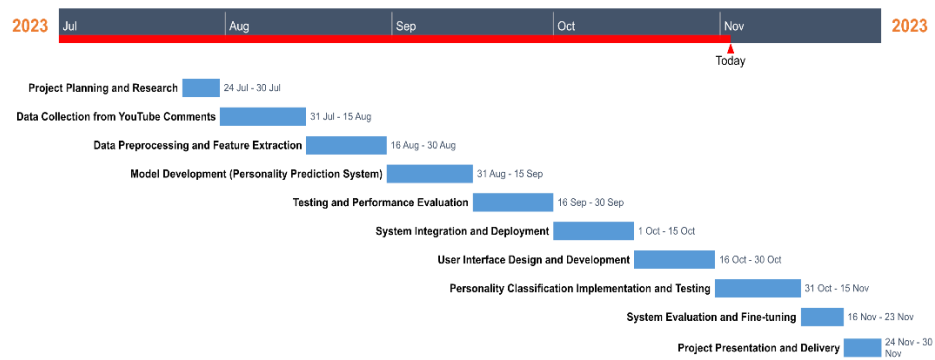


Fig 3.1

CHAPTER 4

ANALYSIS & DESIGN

4.1 PROPOSED METHODOLOGY

One of the major challenges for the project will be the collection of input datasets for the algorithm. The dataset for testing the algorithm is collected from the youtube comment page. This is done by taking out a certain amount of comments and predicting the personality classification. Then, the collected information is fed to the personality classification algorithm. Finally, the algorithm evaluates the data on the basis of the personality traits and displays the result. Then we can come to know the Persoonality type of the Users correctly. From this we come to know the nature of the users.

4.2 SYSTEM ARCHITECTURE

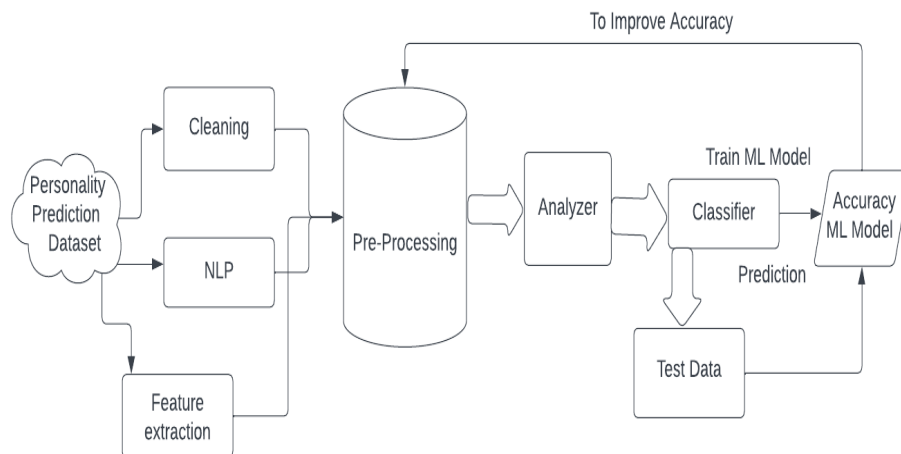


Fig 4.1

USECASE DIAGRAM:

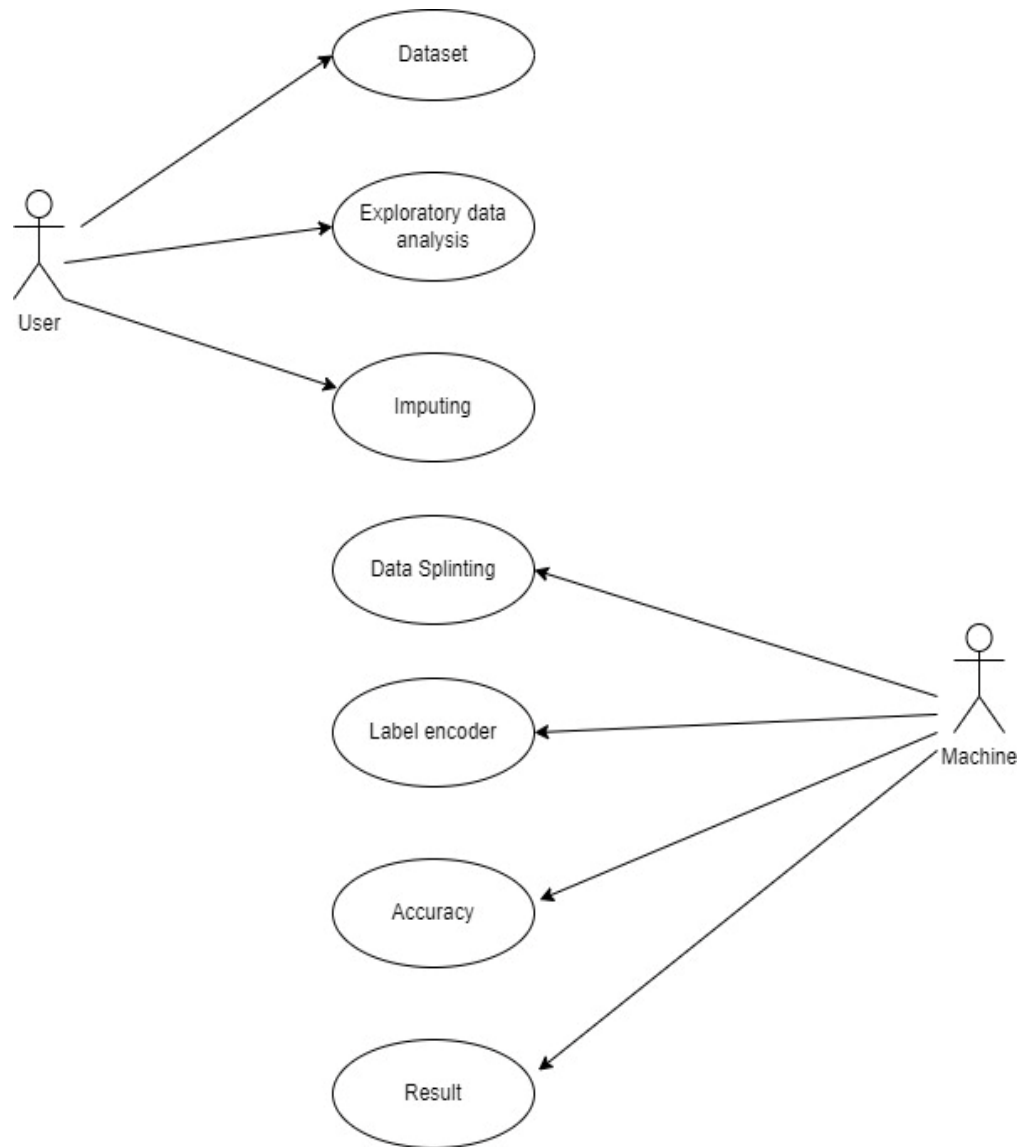


Fig 4.2

CLASS DIAGRAM:

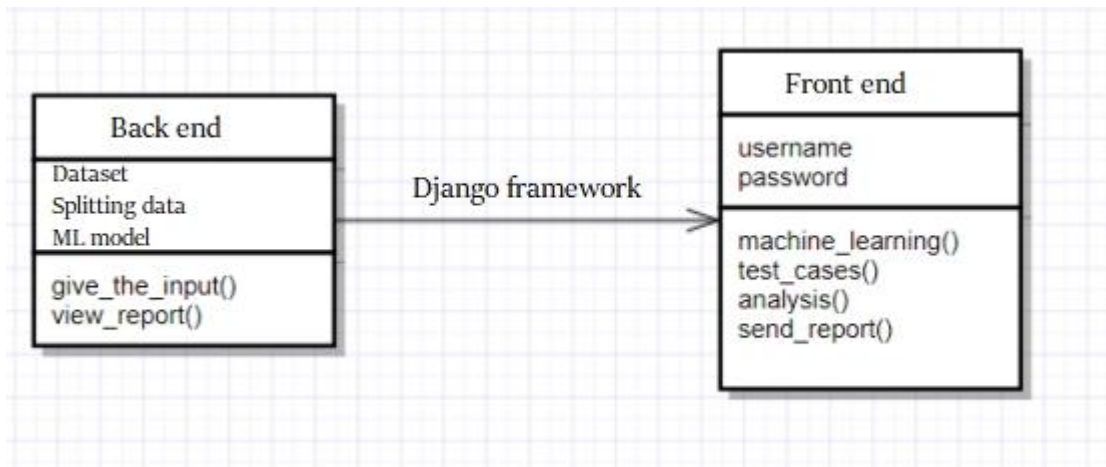


Fig 4.3

SEQUENCE DIAGRAM:

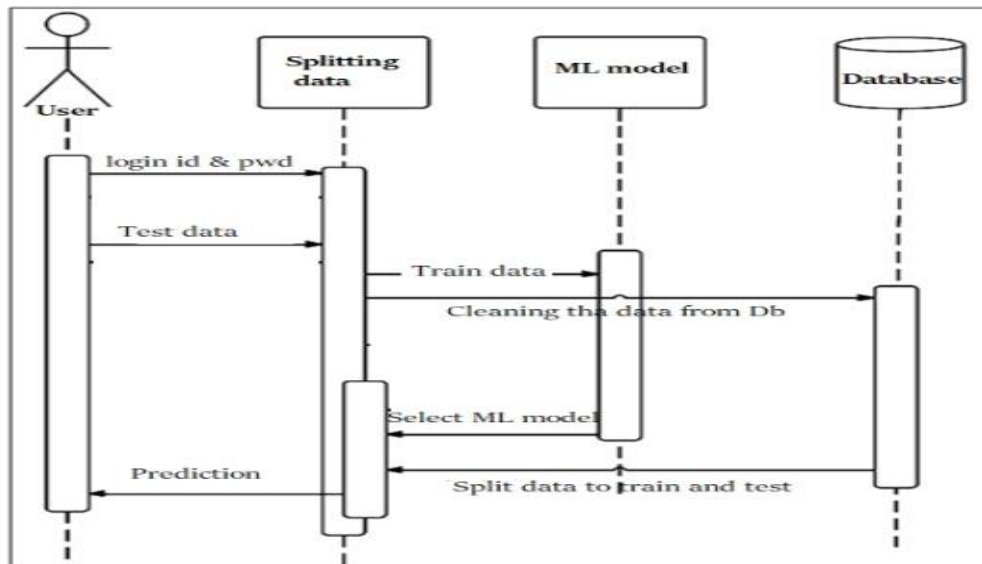


Fig 4.4

ACTIVITY DIAGRAM:

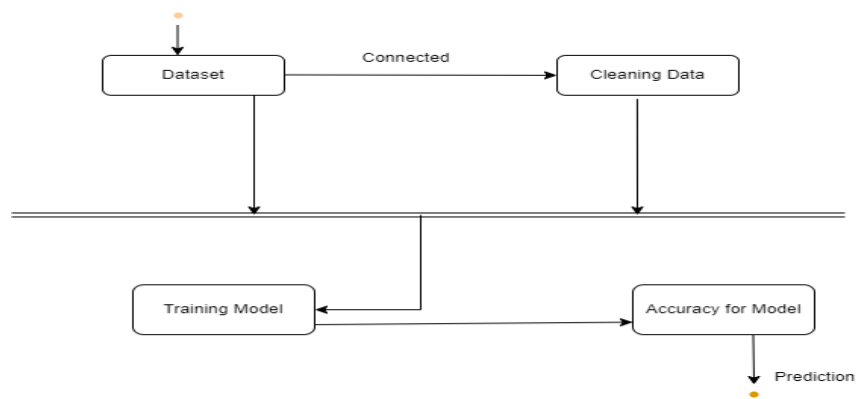


Fig 4.5

E-R DIAGRAM:

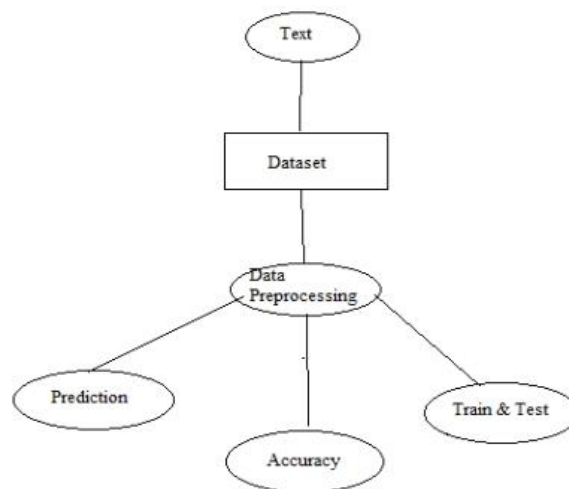


Fig 4.6

DATAFLOW DIAGRAM:

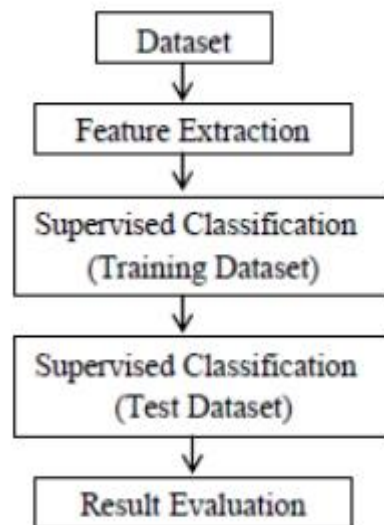


Fig 4.7

4.3 MODULE DESCRIPTION

4.3.1 DATASET COLLECTION

The public dataset of youtube comments is obtained from UCI Machine Learning Repository. The dataset considered in the current research is available on kaggle, a machine learning repository. This study finds that there are only 5,574 labelled comments in the dataset, with 4827 of values belong to legitimate comments while the other 747 values belong to invalid comments. Nonetheless, this dataset consists of two named columns starting with the message labels followed by strings of text comments and three unnamed columns. It's time for a data analyst to pick up the baton and lead the way to machine learning implementation. The job of a data analyst is to find ways and sources of collecting relevant and comprehensive data, interpreting it, and analysing results with the help of statistical techniques. The type of data depends on what you want to predict. There is no exact answer to the question "How much data

is needed?” because each machine learning problem is unique. In turn, the number of attributes data scientists will use when building a predictive model depends on the attributes’ predictive value.

4.3.2 DATA PREPROCESSING

The purpose of preprocessing is to convert raw data into a form that fits machine learning. Structured and clean data allows a data scientist to get more precise results from an applied machine learning model. The technique includes data formatting, cleaning, and sampling.

4.3.3 DATA FORMATTING

The importance of data formatting grows when data is acquired from various sources by different people. The first task for a data scientist is to standardize record formats. A specialist checks whether variables representing each attribute are recorded in the same way. Titles of products and services, prices, date formats, and addresses are examples of variables. The principle of data consistency also applies to attributes represented by numeric ranges.

4.3.4 DATA CLEANING

This set of procedures allows for removing noise and fixing inconsistencies in data. A data scientist can fill in missing data using imputation techniques, e.g. substituting missing values with mean attributes. A specialist also detects outliers — observations that deviate significantly from the rest of distribution. If an outlier indicates erroneous data, a data scientist deletes or corrects them if possible. This stage also includes removing incomplete and useless data objects.

a) DATA ANONYMIZATION

Sometimes a data scientist must anonymize or exclude attributes representing sensitive information

b) DATA SAMPLING

Big datasets require more time and computational power for analysis. If a dataset is too large, applying data sampling is the way to go. A data scientist uses this technique to select a smaller but representative data sample to build and run models much faster, and at the same time to produce accurate outcomes. Pre-processing is the first stage in which the unstructured data is converted into more structured data. Since keywords in youtube comments are prone to be replaced by symbols. In this study, the stop word list remover for English language have been applied to eliminate the stop words in the youtube comments.

4.3.5 NATURAL LANGUAGE PROCESSING(NLP)

The input given by the user is processed through a number of stages to understand what the user is trying to say. Natural language processing (NLP) is the ability of a program to make use of the natural language spoken by a human and comprehend it's meaning. NLP is the study of the computational treatment of natural (human) language. The development of NLP is challenging because computers are used to of getting a highly structured input whereas natural language is highly complex and ambiguous with different linguistic structures and complex variables. NLP has various stages as follows:

- Tokenization (lexical analysis), also referred as segmentation involves breaking up a sentence or paragraph into tokens or individual words, numbers or meaning full phrases. Tokens can be thought of as a small part like a word is a token in a sentence and a sentence is a token in a paragraph. The words are separated with the help of word boundaries. English is space delimited hence, word boundaries are the space between ending of one word and starting of the next one. Example: "I am suffering with fever!" The output after tokenisation would be: ['I' , 'am' , 'suffering' , 'with' , ' fever']
- Syntactic analysis involves analysis of words for grammar and putting the words together in a manner which can show their relationship. This can be done with a data structure such as a parse tree or syntax tree. The tree is constructed with the rules of grammar of the language. If the input can be produced using the syntax tree the input

is found to have correct syntax. For example the string “I pick that have to” will be considered incorrect syntax.

- Semantic analysis picks up the dictionary meaning of words and tries to understand the actual meaning of the sentence. It is the process of mapping syntactic structures with the actual or text independent meaning of the words. Strings like “hot winter” will be disregarded.
- Pragmatic analysis: Pragmatic investigation manages outside word information, which implies learning the outer to the archives and additionally inquiries. Pragmatics analysis that centers around what was portrayed reinterpreted by what it really implied, inferring the different parts of language that require true learning.

4.3.6 FEATURIZATION

Featurization is a way to change some form of data (text data, graph data, time-series data...) into a numerical vector. *Featurization* is different from feature engineering. Feature engineering is just transforming the numerical features somehow so that the machine learning models work well. In feature engineering, features are already in the numerical form. Whereas in Featurization data not need to be in the form of numerical vector. The machine learning model cannot work with row text data directly. In the end, machine learning models work with numerical (categorical, real...) features. So it is import to change some type of data into numerical vector so that we can leverage the whole power of linear algebra (making the decision boundary between data points) and statistics tools with other types of data also. Feature extraction and selection is important for the discrimination of youtube comments. For this phases TFIDF will be used. TFIDF is the often-weighting method used to in the Vector Space Model, particularly in IR domain including text mining. It is a statistical method to measure the important of a word in the document to the whole corpus. The term frequency is simply calculated in proportion to the number of occurrences a word appears in the document and usually normalized in positive quadrant between 0 and 1 to eliminate bias towards lengthy documents.

4.3.7 SPLITTING OF DATA

After cleaning the data, data is normalized in training and testing the model. When data is splitted then we train algorithm on the training data set and keep test data set aside. This training process will produce the training model based on logic and algorithms and values of the feature in training data. Basically aim of feature extraction is to bring all the values under same scale. A dataset used for machine learning should be partitioned into three subsets — training, test, and validation sets.

Training Set : A data scientist uses a training set to train a model and define its optimal parameters — parameters it has to learn from data.

Test Set: A test set is needed for an evaluation of the trained model and its capability for generalization. The latter means a model's ability to identify patterns in new unseen data after having been trained over a training data. It's crucial to use different subsets for training and testing to avoid model over fitting, which is the incapacity for data

4.3.8 MODELING EVALUATION

PERFORMANCE METRICS:

Data was divided into two portions, training data and testing data, both these portions consisting 70% and 30% data respectively. All these two algorithms were applied on same dataset using Enthought Canaopy and results were obtained.

$$\text{Accuracy} = (\text{TP} + \text{TN}) / (\text{P} + \text{N})$$

Predicting accuracy is the main evaluation parameter that we used in this work. Accuracy can be defied using equation. Accuracy is the overall success rate of the algorithm.

CONFUSION MATRIX:

It is the most commonly used evaluation metrics in predictive analysis mainly because it is very easy to understand and it can be used to compute other essential

metrics such as accuracy, recall, precision, etc. It is an NxN matrix that describes the overall performance of a model when used on some dataset, where N is the number of class labels in the classification problem.

All predicted true positive and true negative divided by all positive and negative. True Positive (TP), True Negative (TN), False Negative (FN) and False Positive (FP) predicted by all algorithms are presented in table.

True positive (TP) indicates that the positive class is predicted as a positive class, and the number of sample positive classes was actually predicted by the model.

False negative indicates (FN) that the positive class is predicted as a negative class, and the number of negative classes in the sample was actually predicted by the model.

False positive (FP) indicates that the negative class is predicted as a positive class, and the number of positive classes of samples was actually predicted by the model.

True negative (TN) indicates that the negative class is predicted as a negative class, and the number of sample negative classes was actually predicted by the modellization.

Actual	Negative (0)	True Negative (TN)	False Positive (FP)
	Positive (1)	False Negative (FN)	True Positive (TP)
		Negative (0)	Positive (1)
		Predicted	

Fig 4.8

4.4 ALGORITHMS USED

4.4.1 RANDOM FOREST ALGORITHM:

Random forest is a Supervised Machine Learning Algorithm that is used widely in Classification and Regression problems. It builds decision trees on different samples and takes their majority vote for classification and average in case of regression.

One of the most important features of the Random Forest Algorithm is that it can handle the data set containing continuous variables as in the case of regression and categorical variables as in the case of classification. It performs better results for classification problems.

STEPS INVOLVED IN RANDOM FOREST ALGORITHM:

STEP 1: In Random forest n number of random records are taken from the data set having k number of records.

STEP 2: Individual decision trees are constructed for each sample.

STEP 3: Each decision tree will generate an output

STEP 4: Final output is considered based on Majority Voting or Averaging for Classification and regression respectively.

RANDOM FOREST CLASSIFIER:

Random forests are recently proposed statistical inference tools, deriving their predictive accuracy from the nonlinear nature of their component decision tree members and the power of groups. Random forest committees provide more than just predictions; model information on data proximities can be exploited to provide random forest features. Variable importance measures show which variables are closely

associated with a chosen response variable, while partial dependencies indicate the relation of important variables to said response variable.

Random forest algorithm is a supervised learning algorithm that is developed to solve the problems of regression and classification. So, the main advantage of decision trees is that they can handle both numerical and categorical data. Like other conventional algorithms decision tree algorithm creates a training model and that training model is used to predict the value or class of the target label/variable but here this is done by learning decision rules inferred from previous training dataset. This algorithm makes use of tree structure in which the internal nodes also known as decision node refers to an attribute and each internal node has two or more leaf nodes which corresponds to a class label. The topmost node known as root node corresponds to the best predictor i.e. best attribute of the dataset. This algorithm splits the whole data-frame into parts or subsets and simultaneously a random forest is developed and the end result of this is a tree with leaf nodes, internal nodes and a root node. As the tree becomes more deep and more complex, then the model becomes more and more fit

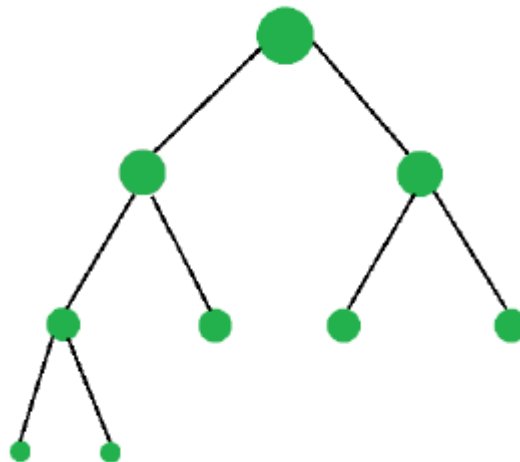


Fig 4.9

4.4.2 DECISION TREE:

A Decision Tree is a supervised machine learning algorithm that is used for classification and regression analysis. It is a graphical representation of all the possible solutions to a decision based on certain conditions. It works by partitioning the data into subsets based on a series of binary questions, with each question representing a node in the tree. The answers to these questions lead to the classification of the data into different categories.

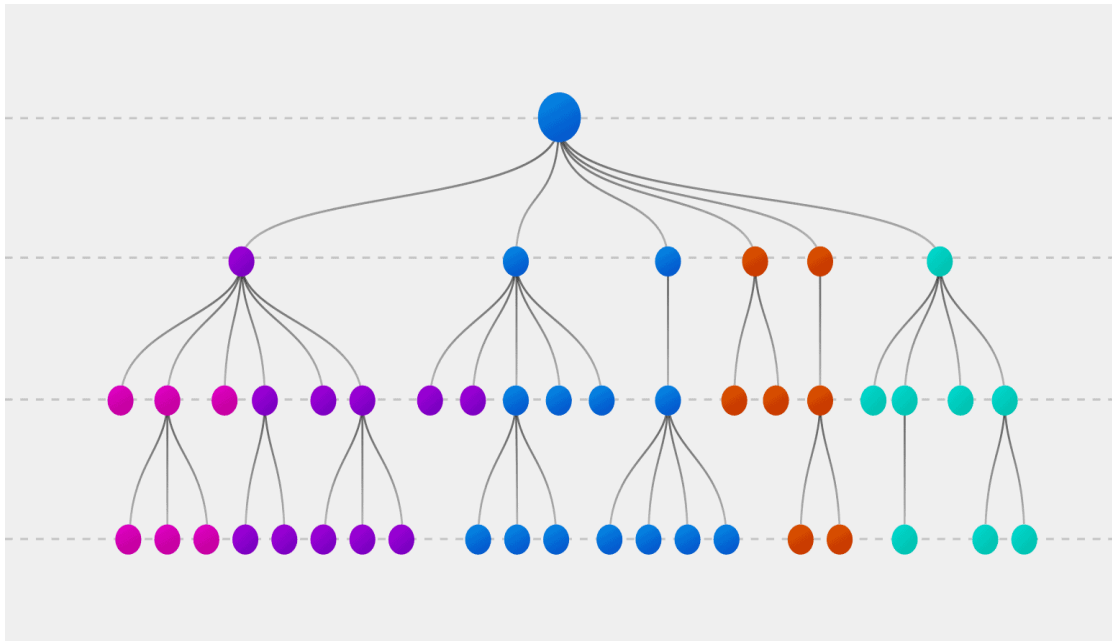


Fig 4.10

The decision tree algorithm is particularly useful when dealing with data with multiple input variables or attributes. It begins by selecting the most important attribute, also known as the root node, and then divides the data into two or more subsets based on that attribute. This process is repeated recursively for each subset until a decision or classification is made. The final result is a tree with decision nodes and leaf nodes,

where the decision nodes represent the attributes or features, and the leaf nodes represent the class labels or output variables. Decision trees are popular because they are easy to understand and interpret. They also have the ability to handle both continuous and categorical data and can be used for both classification and regression tasks. However, decision trees can be prone to overfitting, where the model fits the training data too closely and fails to generalize well to new data. Techniques such as pruning, ensemble methods, and boosting can be used to reduce overfitting and improve the accuracy of the decision tree algorithm.

CHAPTER 5

IMPLEMENTATION & TESTING

5.1 SAMPLE CODE

Front End:

Index:

```
{% load static %}

<!DOCTYPE html>
<!--
Template Name: Shiphile
Author: <a href="https://www.os-templates.com/">OS Templates</a>
Author URI: https://www.os-templates.com/
Copyright: OS-Templates.com
Licence: Free to use under our free template licence terms
Licence URI: https://www.os-templates.com/template-terms
-->
<html lang="">
<!-- To declare your language - read more here:
https://www.w3.org/International/questions/qa-html-language-
declarations -->
<head>
<title> Personality with Disorder Prediction </title>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0,
maximum-scale=1.0, user-scalable=no">
<link href="{% static 'layout/styles/layout.css' %}" rel="stylesheet"
type="text/css" media="all">
</head>
<body id="top">
<!--
#####
##### -->
<!--
#####
##### -->
<!--
#####
##### -->
<!-- Top Background Image Wrapper -->
<div class="bgded overlay" style="background-image:url('{% static
'images/demo/backgrounds/01.jpg' %}');">
```

```

<!--
#####
##### -->

<!--
#####
##### -->

<!--
#####
##### -->

<!--
#####
##### -->

<div id="pageintro" class="hoc clear">
  <!--
#####
##### -->
  <article>
    <h3 class="heading"> Personality with Disorder
Prediction using Machine Learning </h3>
    <p> Sentimental analysis is used to predict the personality along
with the disorder of the person using Machine learning</p>

    <footer>
      <form action='input' method="POST" enctype="multipart/form-data"
class="login100-form validate-form">
        {% csrf_token %}

        <ul class="nospace inline pushright">
          <li>
            <div class="wrap-input100 validate-input" data-validate =
"Valid email is required: ex@abc.xyz">
              <input class="input100" type="text" name="name" style="color:
black;">
              <span class="focus-input100"></span>
              <span class="label-input100">Name</span>
            </div>
            <input class="input100" type="int" name="password"
style="color: black;">
            <span class="focus-input100"></span>
            <span class="label-input100">Password</span>

```

```

        <footer><button type="submit" class="btn" href="#"> LOGIN
</button></footer>
        </ul>
        </form>

        <!--
#####
##### -->
        </div>
        <!--
#####
##### -->
</div>
<!-- End Top Background Image Wrapper -->
<!--
#####
##### -->
<!--
#####
##### -->
<!--
#####
##### -->
<!--
#####
##### -->
<!--
#####
##### -->
<!--
#####
##### -->
<!--
#####
##### -->
<!--
#####
##### -->
<!--
#####
##### -->
<a id="backtotop" href="#top"><i class="fas fa-chevron-up"></i></a>

```

```

<!-- JAVASCRIPTS -->
<script src="{% static 'layout/scripts/jquery.min.js' %}"></script>
<script src="{% static 'layout/scripts/jquery.backtotop.js'
%}"></script>
<script src="{% static 'layout/scripts/jquery.mobilemenu.js'
%}"></script>
</body>
</html>

```

INPUT:

```

{% load static %}

<!DOCTYPE html>
<!--
Template Name: Shiphile
Author: <a href="https://www.os-templates.com/">OS Templates</a>
Author URI: https://www.os-templates.com/
Copyright: OS-Templates.com
Licence: Free to use under our free template licence terms
Licence URI: https://www.os-templates.com/template-terms
-->
<html lang="">
<!-- To declare your language - read more here:
https://www.w3.org/International/questions/qa-html-language-
declarations -->
<head>
<title> Personality with Disorder Prediction </title>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0,
maximum-scale=1.0, user-scalable=no">
<link href="{% static 'layout/styles/layout.css' %}" rel="stylesheet"
type="text/css" media="all">
</head>
<body id="top">
<!--
#####
##### -->

<!--
#####
##### -->
<!-- Top Background Image Wrapper -->
<div class="bgded overlay" style="background-image:url('{% static
'images/demo/backgrounds/01.jpg' %}');">

```



```

<!--
#####
##### -->

<!--
#####
##### -->
<div id="pageintro" class="hoc clear">
  <!--
#####
##### -->
  <article>
    <h3 class="heading"> Personality with Disorder
Prediction using Machine Learning </h3>
    <p> Sentimental analysis is used to predict the personality along
with the disorder of the person using Machine learning</p>

  </article>
  <!--
#####
##### -->
  <form action='output' method="POST" enctype="multipart/form-data"
class="login100-form validate-form">
    {% csrf_token %}

    <ul class="nospace inline pushright">
      <li>
        <div class="wrap-input100 validate-input" data-validate =
"Valid email is required: ex@abc.xyz">
          <input class="input100" type="text" name="text" style="color:
black;">
          <span class="focus-input100"></span>

          <span class="label-input100">Text</span>
        </div>
        <select class="input100" type = 'text' name="algo"
style="color: black;">
          <option value='dt'> Decision Tree </option>
          <option value='rf'> Random Forest </option>

        </select>
        <span class="label-input100">Algorithm</span>

```

```

        <footer><button type="submit" class="btn"
href="#">Predict</button></footer>
    </ul>
</form>

</div>
<!--
#####
##### -->
</div>
<!-- End Top Background Image Wrapper -->

<!--
#####
##### -->
<a id="backtotop" href="#top"><i class="fas fa-chevron-up"></i></a>
<!-- JAVASCRIPTS -->
<script src="{% static 'layout/scripts/jquery.min.js' %}"></script>
<script src="{% static 'layout/scripts/jquery.backtotop.js'
%}"></script>
<script src="{% static 'layout/scripts/jquery.mobilemenu.js'
%}"></script>
</body>
</html>

```

OUTPUT:

```

{% load static %}

<!DOCTYPE html>
<!--
Template Name: Shiphile
Author: <a href="https://www.os-templates.com/">OS Templates</a>
Author URI: https://www.os-templates.com/
Copyright: OS-Templates.com
Licence: Free to use under our free template licence terms
Licence URI: https://www.os-templates.com/template-terms
-->
<html lang="">
<!-- To declare your language - read more here:
https://www.w3.org/International/questions/qa-html-language-
declarations -->
<head>
<title> Personality with Disorder Prediction </title>

```

```

<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0,
maximum-scale=1.0, user-scalable=no">
<link href="{% static 'layout/styles/layout.css' %}" rel="stylesheet"
type="text/css" media="all">
</head>
<body id="top">
<!--
#####
##### -->
<!--
#####
##### -->
<!--
#####
##### -->
<!-- Top Background Image Wrapper -->
<div class="bgded overlay" style="background-image:url('{% static
'images/demo/backgrounds/01.jpg' %}');">
  <!--
#####
##### -->

  <!--
#####
##### -->

  <!--
#####
##### -->

  <!--
#####
##### -->

  <div id="pageintro" class="hoc clear">
    <!--
#####
##### -->

    <article>
      <h3 class="heading"> Predicted Personality with Disorder </h3>

      <footer><a class="btn" href="#">{{out}}</a></footer>
    </article>
    <!--
#####
##### -->
  </div>

```

```

<!--
#####
##### -->
</div>
<!-- End Top Background Image Wrapper -->
<!--
#####
##### -->
<!--
#####
##### -->
<!--
#####
##### -->

<!--
#####
##### -->
<!--
#####
##### -->

<!--
#####
##### -->

<!--
#####
##### -->

<!--
#####
##### -->

<!--
#####
##### -->

<!--
#####
##### -->

<!--
#####
##### -->
<a id="backtotop" href="#top"><i class="fas fa-chevron-up"></i></a>
<!-- JAVASCRIPTS -->
<script src="{% static 'layout/scripts/jquery.min.js' %}"></script>
<script src="{% static 'layout/scripts/jquery.backtotop.js'
%}"></script>
<script src="{% static 'layout/scripts/jquery.mobilemenu.js'
%}"></script>
</body>
</html>

```

BACK END:

```
import pandas as pd
```

```
import numpy as np
```

```
import matplotlib.pyplot as plt
```

```
#list of useful imports that I will use
```

```
%matplotlib inline
```

```
import os
```

```
import matplotlib.pyplot as plt
```

```
import pandas as pd
```

```
import numpy as np
```

```
import seaborn as sns
```

```
import random
```

```
from sklearn.feature_extraction.text import TfidfTransformer
```

```
from sklearn.feature_extraction.text import TfidfVectorizer
```

```
from sklearn.feature_extraction.text import CountVectorizer
```

```
from sklearn.metrics import confusion_matrix
```

```
from sklearn import metrics
```

```
from sklearn.metrics import roc_curve, auc
```

```
from nltk.stem.porter import PorterStemmer
```

```

from nltk.corpus import stopwords

from nltk.tokenize import word_tokenize

# Imputer

#from sklearn.preprocessing import Imputer

from sklearn.impute import SimpleImputer

#imputer = SimpleImputer(missing_values=np.nan, strategy='mean')

from sklearn_pandas import DataFrameMapper

data1 = pd.read_csv(r'C:\Users\ragzv\Music\Personality_pred\dataset.csv')

data1

data = data1[['type', 'posts', 'Disorder']]

data.head()

data.shape

#Check the data

data.info()

#Check the missing values in the data

data.isnull().sum()

data.dropna(inplace=True)

data.isnull().sum()

data['Disorder'].value_counts()

data['Disorder']=data['Disorder'].replace({'bipolar_disorder, depression, ptsd,
seasonal_affective_disorder':'bipolar_disorder', 'suicide_(attempt), anxiety, depression,
eating, panic, schizophrenia, bipolar_disorder,
ptsd':'adhd,suicide_(attempt),anxiety,stress','suicide_(ideation), depression, anxiety,
stress':'adhd,suicide_(attempt),anxiety,stress','suicide_(ideation),
stress':'adhd,suicide_(attempt),anxiety,stress','suicide_(attempt), suicide_(ideation),
depression':'adhd,suicide_(attempt),anxiety,stress','stress, stress_(stressor_and_subjects)
':'suicide_(attempt),anxiety,stress','adhd, anxiety, autism,
bipolar_disorder':'adhd,suicide_(attempt),anxiety,stress','eating, eating_(recovery)':'eating,
depression','depression, eating':'eating,
depression','anxiety':'suicide_(attempt),anxiety,stress','anxiety,

```

```

borderline_personality_disorder, bipolar_disorder, opiate_addiction, self_harm, aspergers,
autism, alcoholism, opiate_usage, schizophrenia,
suicide_(ideation)': 'bipolar_disorder, anxiety', 'depression, trauma, bipolar_disorder, ptsd,
psychosis, eating, self_harm, rape_(survivors), panic, anxiety_(social),
suicide_(ideation)': 'bipolar_disorder, anxiety', 'stress': 'suicide_(attempt), anxiety, stress', 'postp
artum_depression': 'eating, depression', 'depression_(symptoms)': 'eating,
depression', 'borderline_personality_disorder': 'borderline_personality_disorder', 'suicide_(i
deation)': 'suicide_(attempt), anxiety, stress', 'eating': 'eating,
depression', 'aggression': 'cyberbullying, aggression', 'cyberbullying': 'cyberbullying, aggression',
adhd': 'adhd, suicide_(attempt), anxiety, stress', 'adhd, anxiety, bipolar_disorder, depression,
eating, ocd, ptsd, schizophrenia,
seasonal_affective_disorder': 'ptsd', 'ptsd': 'adhd, suicide_(attempt), anxiety, stress', 'adhd, suici
de_(attempt), anxiety, stress': 'adhd, suicide_(attempt), anxiety, stress, ptsd', 'self_harm': 'self_ha
rm, self_esteem', 'self_esteem': 'self_harm, self_esteem', 'bipolar_disorder': 'bipolar_disorder, a
nxiety', 'eating_(recovery)': 'eating, depression', 'depression, substance_use, sleep_disorder,
eating': 'eating, depression', 'stress,
stress_(stressors_and_subjects)': 'suicide_(attempt), anxiety, stress', 'suicide_(ideation),
imminent_death, depression, loneliness': 'suicide_(attempt), anxiety, stress', 'life_satisfaction,
depression': 'depression', 'sentiment': 'self_harm, self_esteem', 'cognitive_distortion': 'mental_
health_(combined), cognitive_distortion', 'mental_health_(combined)': 'mental_health_(comb
ined), cognitive_distortion', 'antisocial_behavior': 'self_harm, self_esteem', 'self_harm, self_este
em': 'self_harm, self_esteem, antisocial_behavior'})

```

```

data['Disorder'].value_counts()

```

```

from sklearn.utils import resample

```

```

# Separate majority and minority classes

```

```

df1 = data[data['Disorder'] == 'borderline_personality_disorder']

```

```

df2 = data[data['Disorder'] == 'suicide_(attempt), anxiety, stress']

```

```

# Downsample majority class and upsample the minority class

```

```

df1_upsampled = resample(df1, replace=True, n_samples=800, random_state=123)

```

```

df2_downsampled = resample(df2, replace=True, n_samples=800, random_state=123)

```

```

# Combine minority class with downsampled majority class

```

```

data1 = pd.concat([df1_upsampled, df2_downsampled])

```

```

# Display new class counts

data1['Disorder'].value_counts()

data1.head(5)

data1['type'].value_counts()

from sklearn.utils import resample

# Separate majority and minority classes

df1 = data1[data1['type']=='INFP']
df2 = data1[data1['type']=='INFJ']
df3 = data1[data1['type']=='INTP']
df4 = data1[data1['type']=='INTJ']
df5 = data1[data1['type']=='ENTP']


# Downsample majority class and upsample the minority class

df1_upsampled = resample(df1, replace=True,n_samples=150,random_state=123)
df2_downsampled = resample(df2, replace=True,n_samples=150,random_state=123)
df3_upsampled = resample(df3, replace=True,n_samples=150,random_state=123)
df4_downsampled = resample(df4, replace=True,n_samples=150,random_state=123)
df5_upsampled = resample(df5, replace=True,n_samples=150,random_state=123)


# Combine minority class with downsampled majority class

df_upsampled = pd.concat([df1_upsampled, df2_downsampled,df3_upsampled,
df4_downsampled,df5_upsampled])

# Display new class counts

df_upsampled['type'].value_counts()

# shuffle the DataFrame rows

data2= df_upsampled.sample(frac = 1)

```



```

import re

def decontracted(phrase):

    # specific

    phrase = re.sub(r"won't", "will not", phrase)

    phrase = re.sub(r"can't", "can not", phrase)


    # general

    phrase = re.sub(r"n't", " not", phrase)

    phrase = re.sub(r"\'re", " are", phrase)

    phrase = re.sub(r"\s", " is", phrase)

    phrase = re.sub(r"\d", " would", phrase)

    phrase = re.sub(r"\ll", " will", phrase)

    phrase = re.sub(r"\t", " not", phrase)

    phrase = re.sub(r"\ve", " have", phrase)

    phrase = re.sub(r"\m", " am", phrase)

    return phrase

stopwords= ['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', "you're", "you've", \
            "you'll", "you'd", 'your', 'yours', 'yourself', 'yourselves', 'he', 'him', 'his', 'himself', \
            'she', "she's", 'her', 'hers', 'herself', 'it', "it's", 'its', 'itself', 'they', 'them', 'their', \
            'theirs', 'themselves', 'what', 'which', 'who', 'whom', 'this', 'that', "that'll", 'these', \
            'those', \
            'am', 'is', 'are', 'was', 'were', 'be', 'been', 'being', 'have', 'has', 'had', 'having', 'do', \
            'does', \
            'did', 'doing', 'a', 'an', 'the', 'and', 'but', 'if', 'or', 'because', 'as', 'until', 'while', 'of', \
            'at', 'by', 'for', 'with', 'about', 'against', 'between', 'into', 'through', 'during', 'before', \
            'after', \

```

```

    'above', 'below', 'to', 'from', 'up', 'down', 'in', 'out', 'on', 'off', 'over', 'under', 'again',
    'further',\

    'then', 'once', 'here', 'there', 'when', 'where', 'why', 'how', 'all', 'any', 'both', 'each',
    'few', 'more',\

    'most', 'other', 'some', 'such', 'only', 'own', 'same', 'so', 'than', 'too', 'very', \

    's', 't', 'can', 'will', 'just', 'don', "don't", 'should', "should've", 'now', 'd', 'll', 'm', 'o', 're',
    \

    've', 'y', 'ain', 'aren', "aren't", 'couldn', "couldn't", 'didn', "didn't", 'doesn', "doesn't",
    'hadn',\

    'hadn't', 'hasn', "hasn't", 'haven', "haven't", 'isn', "isn't", 'ma', 'mightn', "mightn't",
    'mustn',\

    'mustn't', 'needn', "needn't", 'shan', "shan't", 'shouldn', "shouldn't", 'wasn', "wasn't",
    'weren', "weren't", \

    'won', "won't", 'wouldn', "wouldn't"]

```

```

data2['posts'].head(5)

print("printing some random reviews")

print(7, data2['posts'].values[7])

print(234, data2['posts'].values[234])

print(17, data2['posts'].values[17])

# Combining all the above students

from tqdm import tqdm

def preprocess_text(text_data):

    preprocessed_text = []

    # tqdm is for printing the status bar

    for sentence in tqdm(text_data):

        sent = decontracted(sentence)

        sent = sent.replace('\r', ' ')

        sent = sent.replace('\n', ' ')

        sent = sent.replace('\\"', ' ')

```

```

sent = re.sub('[^A-Za-z0-9]+', '', sent)

sent = ' '.join(e for e in sent.split() if e.lower() not in stopwords)

preprocessed_text.append(sent.lower().strip())

return preprocessed_text

preprocessed_text = preprocess_text(data2['posts'].values)

print("printing some random reviews")

print(7, preprocessed_text[7])

print(234, preprocessed_text[234])

print(17, preprocessed_text[17])

from sklearn.preprocessing import LabelEncoder

data2['type'].value_counts()

data2.head(10)

data2.tail(10)

data2.shape

x = data2[['posts']]

le = LabelEncoder()

y = le.fit_transform(data2['type'])

y1= le.fit_transform(data2['Disorder'])

y1 = np.array(y1)

y = np.array(y)

from sklearn.model_selection import train_test_split

#Breaking into Train and test

X_train, X_test, y_train, y_test = train_test_split(preprocessed_text, y,
test_size=0.3,stratify=y ,random_state=42)

X_train

X_test

```

```

y_train.shape

y_test

#Featuraization:- TF-IDF

import pickle

from sklearn import preprocessing

tfidf = TfidfVectorizer(ngram_range=(1,2), min_df=10)
tfidf.fit(X_train)

import pickle

filename = r'C:\Users\ragzv\Music\FRONT END\new_tfidf.pkl'

pickle.dump(tfidf, open(filename, 'wb'))# fit has to happen only on train data


# we use the fitted CountVectorizer to convert the text to vector
X_train_tfidf =tfidf.transform(X_train)
X_test_tfidf = tfidf.transform(X_test)


#Normalize Data
X_train_tfidf = preprocessing.normalize(X_train_tfidf)
print("Train Data Size: ",X_train_tfidf.shape)


#Normalize Data
X_test_tfidf = preprocessing.normalize(X_test_tfidf)
print("Test Data Size: ",X_test_tfidf.shape)


#Random Forest with TF-IDF
from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import roc_auc_score

from sklearn.model_selection import GridSearchCV

from sklearn.metrics import roc_curve


dept = [1, 5, 10, 50, 100, 500, 1000]

n_estimators = [20, 40, 60, 80, 100, 120]

```

```

param_grid={'n_estimators':n_estimators , 'max_depth':dept}

clf = RandomForestClassifier()

model = GridSearchCV(clf,param_grid,scoring='accuracy',n_jobs=-1,cv=3)

model.fit(X_train_tfidf,y_train)

print("optimal n_estimators",model.best_estimator_.n_estimators)

print("optimal max_depth",model.best_estimator_.max_depth)

optimal_max_depth = model.best_estimator_.max_depth

optimal_n_estimators = model.best_estimator_.n_estimators

from sklearn.metrics import accuracy_score

#training our model for max_depth=100,n_estimators = 120

clf = RandomForestClassifier(max_depth = optimal_max_depth,n_estimators =
optimal_n_estimators)

clf.fit(X_train_tfidf,y_train)


pred_test =clf.predict(X_test_tfidf)

test_accuracy = accuracy_score(y_test, pred_test)

pred_train = clf.predict(X_train_tfidf)

train_accuracy =accuracy_score(y_train,pred_train)


print("AUC on Test data is " +str(accuracy_score(y_test,pred_test)))

print("AUC on Train data is " +str(accuracy_score(y_train,pred_train)))


print("-----")

```

```

# Code for drawing seaborn heatmaps

class_names = ['INFP personality No borderline_personality_disorder','INFJ with
borderline_personality_disorder ','INTP No borderline_personality_disorder','INTJ No
borderline_personality_disorder','ENTP With borderline_personality_disorder']

df_heatmap = pd.DataFrame(confusion_matrix(y_test, pred_test.round()),
index=class_names, columns=class_names )

fig = plt.figure( )

heatmap = sns.heatmap(df_heatmap, annot=True, fmt="d")

all_model_result = pd.DataFrame(columns=['METHOD', 'Classifier' , 'Train-Accuracy', 'Test-
Accuracy' ])

new = ['TFIDF ','Random forest-Classfier',train_accuracy, test_accuracy]

all_model_result.loc[0] = new

from sklearn.tree import DecisionTreeClassifier

from sklearn.metrics import accuracy_score,confusion_matrix

from sklearn.model_selection import GridSearchCV

dept = [1, 5, 10, 50, 100, 500,800, 1000]

min_samples = [5, 10, 100, 500]

param_grid={'min_samples_split':min_samples , 'max_depth':dept}

clf = DecisionTreeClassifier()

model = GridSearchCV(clf,param_grid,scoring='accuracy',n_jobs=-1,cv=3)

model.fit(X_train_tfidf,y_train)

print("optimal min_samples_split",model.best_estimator_.min_samples_split)

print("optimal max_depth",model.best_estimator_.max_depth)

optimal_max_depth = model.best_estimator_.max_depth

```

```

optimal_min_samples_split = model.best_estimator_.min_samples_split

#Testing AUC on Test data

dt = DecisionTreeClassifier(max_depth =500,min_samples_split =5)

dt.fit(X_train_tfidf,y_train)


#predict on test data and train data

y_predtestd = dt.predict(X_test_tfidf)
y_predtraind = dt.predict(X_train_tfidf)

pred_test =dt.predict(X_test_tfidf)
test_accuracy = accuracy_score(y_test, pred_test)
pred_train = dt.predict(X_train_tfidf)
train_accuracy =accuracy_score(y_train,pred_train)

print('*'*35)

#accuracy on training and testing data

print('the accuracy on testing data',accuracy_score(y_test,y_predtestd))
print('the accuracy on training data',accuracy_score(y_train,y_predtraind))
train0 = accuracy_score(y_train,y_predtraind)
test0 = accuracy_score(y_test,y_predtestd)

```

```

print('*'*35)

# Code for drawing seaborn heatmaps

class_names = ['INFP personality No borderline_personality_disorder','INFJ with
borderline_personality_disorder ','INTP No borderline_personality_disorder','INTJ No
borderline_personality_disorder','ENTP With borderline_personality_disorder']

df_heatmap = pd.DataFrame(confusion_matrix(y_test, pred_test.round()),
index=class_names, columns=class_names )

fig = plt.figure( )

heatmap = sns.heatmap(df_heatmap, annot=True, fmt="d")

new = ['TFIDF ','DECISION TREE',train0, test0]

all_model_result.loc[1] = new

all_model_result

```


5.2 SAMPLE OUTPUT

FRONT END:

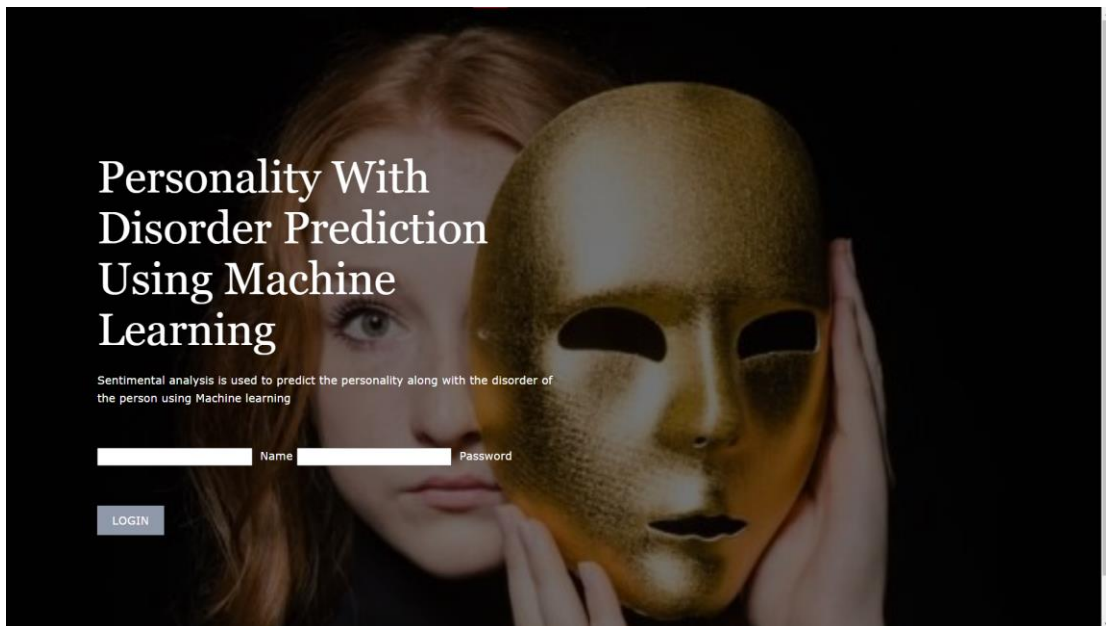


Fig 5.1

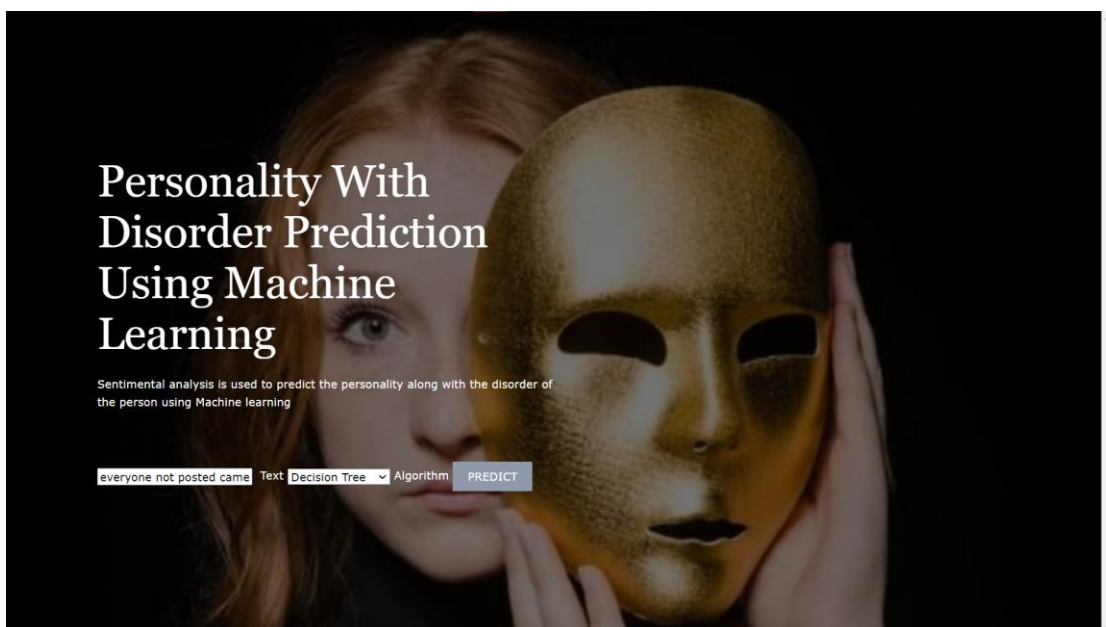


Fig 5.2

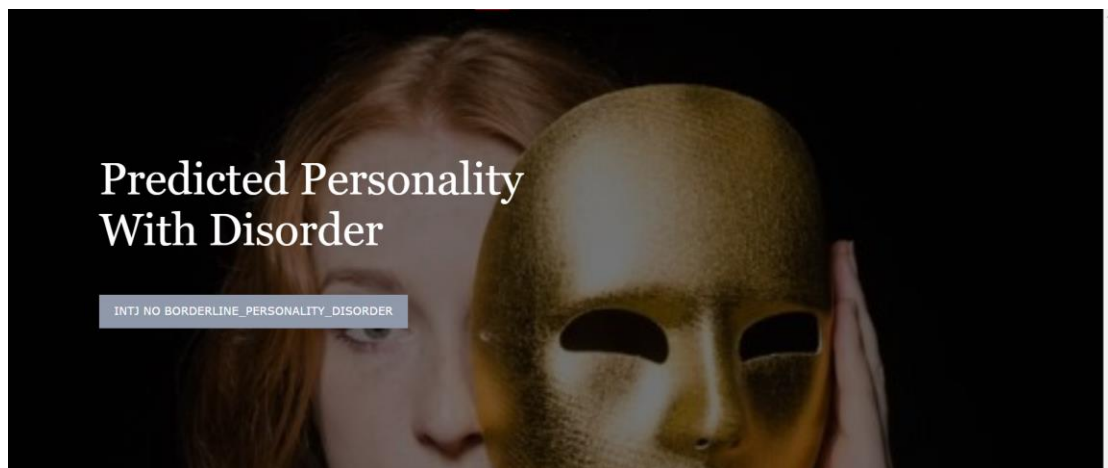


Fig 5.3

BACKEND:

```
In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

# list of useful imports that I will use
import matplotlib inline
import os

import matplotlib.pyplot as plt
import pandas as pd

import numpy as np

import seaborn as sns
import random
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.feature_extraction.text import CountVecorizer
from sklearn.metrics import confusion_matrix
from sklearn import metrics
from sklearn.metrics import roc_curve, auc
from nltk.stem.porter import PorterStemmer

from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
# Inputer
# from sklearn.preprocessing import Inputer

from sklearn.input import SimpleInputer
# Inputer = SimpleInputer(missing_value=np.nan, strategy='mean')
from sklearn_pandas import DataFrameMapper

In [2]: data1 = pd.read_csv(r"C:\Users\ragz\Music\Personality_pred\dataset.csv")

In [3]: data1

Out[3]:
```

	type	posts	Unnamed: 2	Unnamed: 3	Unnamed: 4	Unnamed: 5	Unnamed: 6	Unnamed: 7	Unnamed: 8	Unnamed: 9	...	Unnamed: 92	Unnamed: 93
0	ENTP	"I'm finding the lack of me in these posts ver..."	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN
1	INTP	"Good one _____ https://www.youtube.com/watch?v=... "	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN
2	INTJ	"Dear INTP, I enjoyed our conversation the o..."	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN
3	ENTJ	"You're fired!! That's another day mission..."	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN
4	INTJ	"18/17 @@@ Science is not perfect. No scien..."	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN

Fig 5.4

```

5413 ENFP 'I certainly see how enfp's can be fascinated ... NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN ... NaN NaN
5414 ENTP '1. originality, be opinionated on intellectua... NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN ... NaN NaN
5415 INFJ 'This is how I felt at first as well. A good r... NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN ... NaN NaN
5416 INFP 'yes >X< because i said so|||cuddling is my l... NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN ... NaN NaN
5417 INTJ 'Exactly, and as an INTJ, whose Se is his infe... NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN ... NaN NaN

5418 rows x 102 columns

```

In [4]: `data = data[['type', 'posts', 'Disorder']]`

In [5]: `data.head()`

Out[5]:

	type	posts	Disorder
0	ENTP	'I'm finding the lack of me in these posts ver...	borderline_personality_disorder
1	INTP	'Good one ____ https://www.youtube.com/wat...	bipolar_disorder, depression, ptsd, seasonal_a...
2	INTJ	'Dear INTP, I enjoyed our conversation the o...	adhd, anxiety, bipolar_disorder, depression, e...
3	ENTJ	'You're fired.!!!That's another silly misconce...	na
4	INTJ	'18/37 @ @ Science is not perfect. No scien...	postpartum_depression

In [6]: `data.shape`

Out[6]: (5418, 3)

In [7]: `#Check the data
data.info()`

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5418 entries, 0 to 5417
Data columns (total 3 columns):
#   Column      Non-Null Count  Dtype
---  ---
0    type        5417 non-null   object
1    posts       5417 non-null   object
2    Disorder    5403 non-null   object
dtypes: object(3)
memory usage: 127.1+ KB

```

In [8]: `#Check the missing values in the data
data.isnull().sum()`

Fig 5.5

```

In [8]: #Check the missing values in the data
data.isnull().sum()

Out[8]:
type        1
posts       1
Disorder    15
dtype: int64

In [9]: data.dropna(inplace=True)

C:\Users\ragzv\AppData\Local\Temp\ipykernel_13648\1368182302.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
data.dropna(inplace=True)

In [10]: data.isnull().sum()

Out[10]:
type        0
posts       0
Disorder     0
dtype: int64

In [11]: data['Disorder'].value_counts()

Out[11]:
borderline_personality_disorder    831
suicide_ideation                  781
depression                        778
na                                 566
depression, ptsd                  556
eating                            238
suicide_(attempt)                 185
mental_health_(combined)          168
bipolar_disorder, depression, ptsd, seasonal_affective_disorder    129
adhd, anxiety, bipolar_disorder, depression, eating, ocd, ptsd, schizophrenia, seasonal_affective_disorder    117
postpartum_depression             115
schizophrenia                     112
bipolar_disorder                   90
suicide_(attempt), anxiety, depression, eating, panic, schizophrenia, bipolar_disorder, ptsd

```

Fig 5.6

```

suicide_attempt), anxiety, depression, eating, panic, schizophrenia, bipolar_disorder, ptsd
88
anxiety, borderline_personality_disorder, bipolar_disorder, opiate_addiction, self_harm, aspergers, autism, alcoholism, opiate_
usage, schizophrenia, suicide_ideation)      84
antisocial_behavior
82
depression, trauma, bipolar_disorder, ptsd, psychosis, eating, self_harm, rape_survivors, panic, anxiety_social, suicide_ideation)
80
cyberbullying
77
aggression
69
self_harm
44
ptsd
44
eating, depression
36
adhd
22
borderline_personality_disorder
22
suicide_attempt, suicide_ideation, depression
14
anxiety
13
self_esteem
12
cognitive_distortion
10
adhd, anxiety, autism, bipolar_disorder
8
eating, eating_recovery
7
stress
6
depression_symptoms
4
depression, eating
4
stress, stress_stressor_and_subjects
3
suicide_ideation, stress
2
suicide_ideation, depression, anxiety, stress
1
sentiment
1
depression, substance_use, sleep_disorder, eating
1
life_satisfaction, depression
1
suicide_ideation, imminent_death, depression, loneliness
1

```

Fig 5.7

```

1
eating_recovery
1
Name: Disorder, dtype: int64

In [12]: data['Disorder']=data['Disorder'].replace({'bipolar_disorder, depression, ptsd, seasonal_affective_disorder':'bipolar_disorder',
data['Disorder'].value_counts()

< [REDACTED] >

C:\Users\ragzv\AppData\Local\Temp\ipykernel_13648\2421200266.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
data['Disorder']=data['Disorder'].replace({'bipolar_disorder, depression, ptsd, seasonal_affective_disorder':'bipolar_disorder',
n','suicide_attempt), anxiety, depression, eating, panic, schizophrenia, bipolar_disorder, ptsd':'adhd,suicide_attempt),anxiety,depression,
ty,stress','suicide_ideation), depression, anxiety, stress':'adhd,suicide_attempt),anxiety,stress','suicide_ideation), stress':'adhd,suicide_attempt),anxiety,stress',
s':'adhd,suicide_attempt),anxiety,stress','suicide_attempt), suicide_ideation), depression':'adhd,suicide_attempt),anxiety,
stress','stress, stress_stressor_and_subjects':'suicide_attempt),anxiety,stress','adhd, anxiety, autism, bipolar_disorder,
n':'adhd,suicide_attempt),anxiety,stress','eating, eating_recovery':'eating, depression','depression, eating':'eating, depression',
n':'adhd,suicide_attempt),anxiety,stress','anxiety, borderline_personality_disorder, bipolar_disorder, opiate_addiction, self_harm, aspergers, autism, alcoholism, opiate_usage, schizophrenia, suicide_ideation':'bipolar_disorder,anxiety','depression, trauma, bipolar_disorder, ptsd, psychosis, eating, self_harm, rape_survivors, panic, anxiety_social, suicide_ideation':'bipolar_disorder,anxiety','stress':'suicide_attempt),anxiety,stress','postpartum_depression':'eating, depression','depression_symptoms':'eating, depression','borderline_personality_disorder':'borderline_personality_disorder','suicide_ideation':'suicide_attempt),anxiety,stress','eating':'eating, depression','aggression':'cyberbullying,aggression','cyberbullying':'cyberbullying,aggression','adhd':'adhd,suicide_attempt),anxiety,stress','adhd, anxiety, bipolar_disorder, depression, eating, ocd, ptsd, schizophrenia, seasonal_affective_disorder':'ptsd','ptsd':'adhd,suicide_attempt),anxiety,stress','adhd,suicide_attempt),anxiety,stress':'adhd,suicide_attempt),anxiety,stress,ptsd','self_harm':'self_harm,self_esteem','self_esteem':'self_harm,self_esteem','bipolar_disorder':'bipolar_disorder,anxiety','eating_recovery':'eating, depression','depression, substance_use, sleep_disorder, eating':'eating, depression','stress, stress_stressor_and_subjects':'suicide_attempt),anxiety,stress','suicide_ideation), imminent_death, depression, loneliness':'suicide_attempt),anxiety,stress','life_satisfaction, depression':'depression','sentiment':'self_harm,self_esteem','cognitive_distortion':'mental_health(combined),cognitive_distortion','mental_health(combined)':'mental_health(combined),cognitive_distortion','antisocial_behavior':'self_harm,self_esteem','self_harm,self_esteem':'self_harm,self_esteem',antisocial_behavior'))

Out[12]:
borderline_personality_disorder      853
suicide_attempt),anxiety,stress      804
depression                          779
na                                   566
depression, ptsd                    556
eating, depression                   406
bipolar_disorder,anxiety            254
suicide_attempt                     185
adhd,suicide_attempt),anxiety,stress 179
mental_health(combined),cognitive_distortion 178
cyberbullying,aggression            146
self_harm,self_esteem               139
bipolar_disorder                    129
ptsd                                117
schizophrenia                       112
Name: Disorder, dtype: int64

```

Fig 5.8


```
In [22]: # Combining all the above students
from tqdm import tqdm
def preprocess_text(text_data):
```

Fig 5.11

```
Out[26]:
```

INTP	150
INFJ	150
INTJ	150

Fig 5.12

```
In [26]: data2['type'].value_counts()
Out[26]:
INTP    150
INFJ    150
INTJ    150
ENTP    150
INFP    150
Name: type, dtype: int64
```

```
In [27]: data2.head(10)
Out[27]:
```

	type	posts	Disorder
831	INTP	"Well... I didn't pull the trigger. "shrug"...	suicide_(attempt),anxiety,stress
413	INTP	https://youtu.be/WzRsyLz4bTo?list=PLBfYHem6W5...	borderline_personality_disorder
3282	INFJ	"I) ENTPs, cannot deny the chemistry and ease ...	borderline_personality_disorder
3018	INTP	"Discussing only go out of my way to make...	borderline_personality_disorder
1073	INFJ	Well, I'm a Brit, she's an American and she as...	suicide_(attempt),anxiety,stress
2406	INTP	"What exactly am I doing? I asked a question.....	suicide_(attempt),anxiety,stress
3711	INTJ	"The MBTI was supposed to take Jung's theory a...	suicide_(attempt),anxiety,stress
3406	INTP	"I'm small, pointless even. Why yes. Why not...	suicide_(attempt),anxiety,stress
2906	ENTP	"I keep forgetting that Hannibal is K-villain-...	borderline_personality_disorder
628	INFJ	Please check out this site. It will absolutely...	suicide_(attempt),anxiety,stress

```
In [28]: data2.tail(10)
Out[28]:
```

	type	posts	Disorder
3090	INTP	"Display your collection of fancy probiotic yo...	borderline_personality_disorder
3836	INFP	"Lavender. Chamomile lavender tea, lavender o...	suicide_(attempt),anxiety,stress
3551	INTP	"Maybe you could ask people on here to talk ab...	borderline_personality_disorder
3808	INFP	"I don't really relate to most of it, to be ho...	borderline_personality_disorder
3369	INFJ	"I just discovered these two on YouTube and am...	borderline_personality_disorder
633	INTP	"Your big 5 results are the equivalent of INFP...	suicide_(attempt),anxiety,stress
29	INFJ	it could be pyrolunia.. you know.. it is an on...	suicide_(attempt),anxiety,stress
1383	INTJ	"INTJ used logic, it's super effective! Welco...	borderline_personality_disorder
3601	INFJ	"Why. LOL Thank you YvY lol i find the bigg...	borderline_personality_disorder
3243	INFP	"< goes down to the kitchen to check, gets sid...	suicide_(attempt),anxiety,stress

```
In [29]: data2.shape
Out[29]: (750, 3)
```

Fig 5.13

```
In [30]: x = data2[['posts']]

In [31]: le = LabelEncoder()
y = le.fit_transform(data2['type'])
y1= le.fit_transform(data2['Disorder'])
y1 = np.array(y1)
y = np.array(y)

In [32]: from sklearn.model_selection import train_test_split
#Breaking into Train and test
X_train, X_test, y_train, y_test = train_test_split(preprocessed_text, y, test_size=0.3, stratify=y, random_state=42)

In [33]: X_train
Out[33]:
['number 1 cleaning houses mom offered job along several friends family hires joining peace corp looking teach english power slacking 4 minutes classes started sleep well foxy 3 nice meet puncool sad told pun unsure hi hello 3 rock also sleep well mo rfy 3 ghostly hearts 3 https www youtube com watch v m0p7nm rmny song placed credits entertaining movie guess movie win morni ng folks waves leave hated place called vertigo wait leaving come back https www youtube com watch v 98u9qumq 2k rolleyes 3 u 2 scrabble championships dyslexicon wink would like call upon zombiefishy witness like call bono extraordinaire got go thinki ng take care family thank sure come around us best asked us makes decisions chose reach 3 good heart always not need convince think highly saying wish good good matters george washington common man amongst uncommon founders pleb power cool not hate lo ng run wanted brush ego since hurt time made mistakes kept constantly making thank 3 ask not think less think best things peo ple sympathy thing ask even short fault not make unless everyone ups forgives tries talking could say something hoped end ent re end hating worked long run since basically ignored every time attempted speak ninja slips away delivers fishy pun jail hey o 3 saying hi leave not much busy part year p nice come unwind cool tongue ninja giggles ninja ninja understand ninja uh not really working writing guys ninja waves excitedly 3 ninja relaxed ninja nod ninja ninja nod ninja never getting chair well ha lf alive technically stay sober long enough able correlate common average estimations every person ever waves back work later oh hush stayed behind snow could get heli p heygo goes stuff holy fuck quoted saying bounces world shaking drinks copious amou nts coffee miss everyone else morning morn no plans exactly except bleargh workity work probably able fly later awake waves 3 3 harsh times moment cannot sure always send 3s self need hugs well good hear things not downers least everyone else miss who le family 3 things getting situated cleaning jobs finding able writing sittings things going well end',
'https www youtube com watch v qok9iale14c https www youtube com watch v ispfzagb4a https www youtube com watch v uikh0nfeq mo https www youtube com watch v d8zcib 59ys https www youtube com watch v nt6mb8n24 g https www youtube com watch v vlgmyv3q

In [34]: X_test
Out[34]:
['og88pgiofhh0sktupk5klcksk really crazy flirty not problems articulating words easier social extroverted not sure already s aid people take happiness good time shallow stupid almost feel opinions people thought biggest fears life probably unfortunat ely ultimately attributed making fool anything get zone focus gather much 1 reggae rock oldies punk whatever gets going 2 kam ikaze impossibles 3 moonshine supervillains 4 weezer pepper passafire impossibles swellers 5 like royals yes way tend take th ings people say constantly mull yes rude say took advantage group put would awesome could update thread ends truly hope get b ack seems like truly love freind whatever would say good luck someone bringing past failures one worst things person hard sel f hard forgive past failures second fe current situation facing made curious fe works heard said fe demonstrates socially ac ceptable behavior seeks take care create intense stare told told think freak people speak someone else stares not ever bother wonder possibly infj not read whole thread friend intp connect get along part idea though trying get see different not sure a lready said friend told feel much think others much standard socially acceptable behavior someone not intense yessir friggin awesome not mind asking wat band u u message u want obsessed recently passafire descendants streetlight manifesto four year st rong larry flask swellers kind different know scroll entp forum decided steal post saksham makes guys happy small big could a nything start singing moshing always struggled self esteem issues never fit felt like always different everyone therefore fel t everything wrong would cry hours childhood remember tortured guilt whenever got trouble small big always overly anxious dis appointed sometimes could not fun personally not like assholes although intp friend love work hairstylist emotionally drainin g job get home sit kinda vegg love job making people happy feel good hate spotlight ill thinking thinking put spot go blank t
```

Fig 5.14


```
In [35]: y_train.shape
Out[35]: (525,)

In [36]: y_test
Out[36]: array([1, 1, 2, 1, 2, 1, 3, 0, 3, 4, 4, 1, 4, 0, 1, 4, 1, 1, 2, 3, 0, 3,
                2, 4, 2, 2, 0, 1, 3, 3, 3, 4, 2, 3, 2, 2, 0, 2, 1, 3, 4, 2, 2, 0,
                3, 0, 4, 1, 1, 1, 2, 4, 2, 3, 3, 4, 4, 1, 4, 1, 2, 0, 3, 0, 0, 0,
                0, 4, 4, 4, 1, 0, 1, 2, 1, 3, 0, 0, 2, 2, 4, 4, 3, 1, 4, 3, 1, 4,
                2, 0, 4, 3, 0, 0, 2, 4, 3, 0, 1, 1, 0, 3, 2, 4, 4, 1, 3, 1, 3,
                3, 3, 1, 4, 3, 2, 4, 1, 4, 3, 4, 0, 1, 1, 4, 2, 4, 2, 1, 3, 3, 0,
                2, 3, 4, 4, 3, 0, 4, 4, 1, 2, 1, 4, 1, 1, 3, 3, 4, 0, 3, 2, 0,
                1, 0, 4, 2, 2, 1, 1, 2, 3, 0, 0, 4, 0, 2, 0, 4, 3, 1, 1, 4, 2,
                2, 4, 3, 0, 0, 2, 0, 0, 0, 2, 2, 2, 1, 2, 3, 1, 0, 2, 1, 2, 3,
                1, 3, 3, 0, 0, 3, 2, 4, 2, 0, 3, 4, 1, 2, 0, 1, 4, 0, 0, 4, 2,
                2, 3, 1, 3, 3])
```

Featurization:- TF-IDF

```
In [37]: import pickle
from sklearn import preprocessing

tfidf = TfidfVectorizer(ngram_range=(1,2), min_df=10)
tfidf.fit(X_train)
import pickle
filename = r'C:\Users\ragzv\Music\FRONT END\new_tfidf.pkl'
pickle.dump(tfidf, open(filename, 'wb'))# fit has to happen only on train data

# we use the fitted CountVectorizer to convert the text to vector
X_train_tfidf = tfidf.transform(X_train)
X_test_tfidf = tfidf.transform(X_test)

#Normalize Data
X_train_tfidf = preprocessing.normalize(X_train_tfidf)
print("Train Data Size: ",X_train_tfidf.shape)

#Normalize Data
X_test_tfidf = preprocessing.normalize(X_test_tfidf)
print("Test Data Size: ",X_test_tfidf.shape)

Train Data Size: (525, 5258)
Test Data Size: (225, 5258)
```

Random Forest with TF-IDF

```
In [38]: from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import roc_auc_score
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import roc_curve
```

Fig 5.15

```
In [38]: from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import roc_auc_score
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import roc_curve

dept = [1, 5, 10, 50, 100, 500, 1000]
n_estimators = [20, 40, 60, 80, 100, 120]

param_grid={'n_estimators':n_estimators, 'max_depth':dept}
clf = RandomForestClassifier()
model = GridSearchCV(clf,param_grid,scoring='accuracy',n_jobs=-1,cv=3)
model.fit(X_train_tfidf,y_train)
print("optimal n_estimators",model.best_estimator_.n_estimators)
print("optimal max_depth",model.best_estimator_.max_depth)
optimal_max_depth = model.best_estimator_.max_depth
optimal_n_estimators = model.best_estimator_.n_estimators

optimal_n_estimators 120
optimal_max_depth 5

In [39]: from sklearn.metrics import accuracy_score
#training our model for max_depth=100,n_estimators = 120
clf = RandomForestClassifier(max_depth = optimal_max_depth,n_estimators = optimal_n_estimators)
clf.fit(X_train_tfidf,y_train)

pred_test =clf.predict(X_test_tfidf)
test_accuracy = accuracy_score(y_test, pred_test)
pred_train = clf.predict(X_train_tfidf)
train_accuracy =accuracy_score(y_train,pred_train)

print("AUC on Test data is " +str(accuracy_score(y_test,pred_test)))
print("AUC on Train data is " +str(accuracy_score(y_train,pred_train)))

print("-----")

# Code for drawing seaborn heatmaps
class_names = ['INFP personality No borderline_personality_disorder','INFP with borderline_personality_disorder','INTP No border
df_heatmap = pd.DataFrame(confusion_matrix(y_test, pred_test.round()), index=class_names, columns=class_names)
fig = plt.figure()
heatmap = sns.heatmap(df_heatmap, annot=True, fmt="d")

AUC on Test data is 0.8088888888888889
AUC on Train data is 0.9828571428571429
-----

INFP personality No borderline_personality_disorder 37 1 5 1 1
```

Fig 5.16

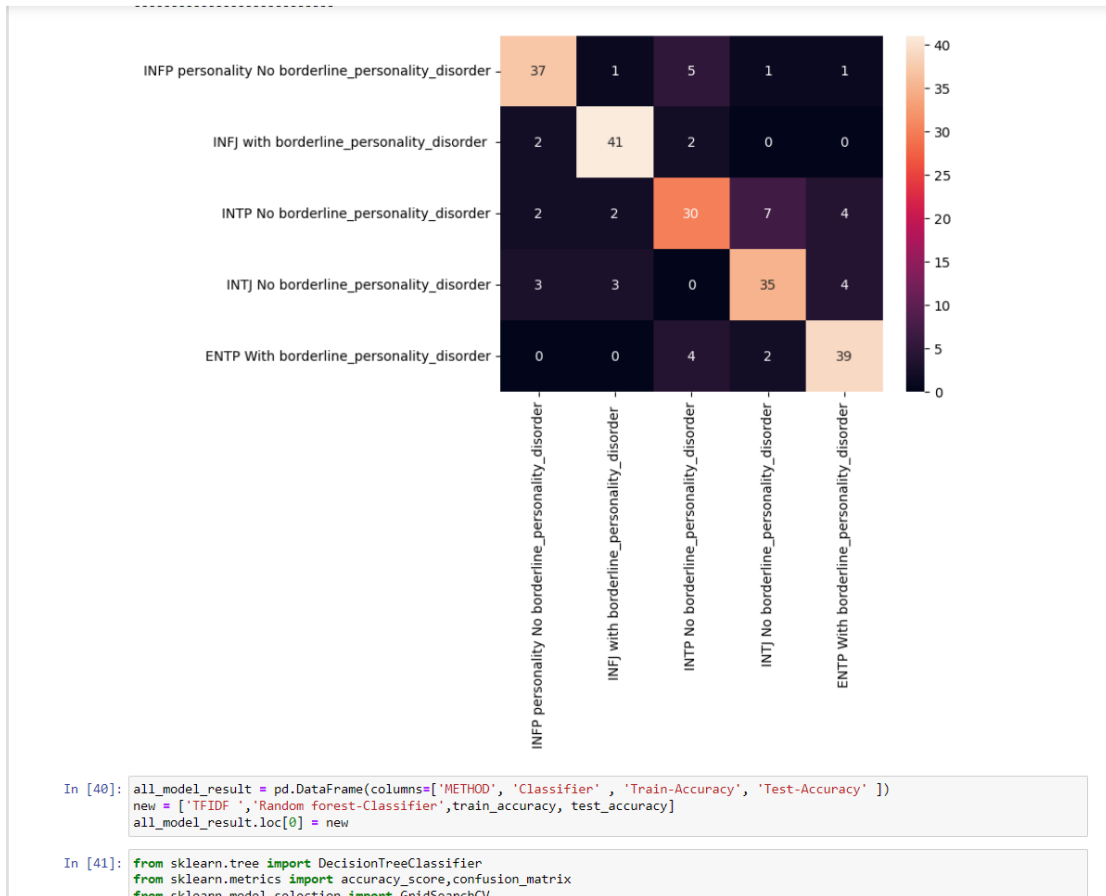


Fig 5.17

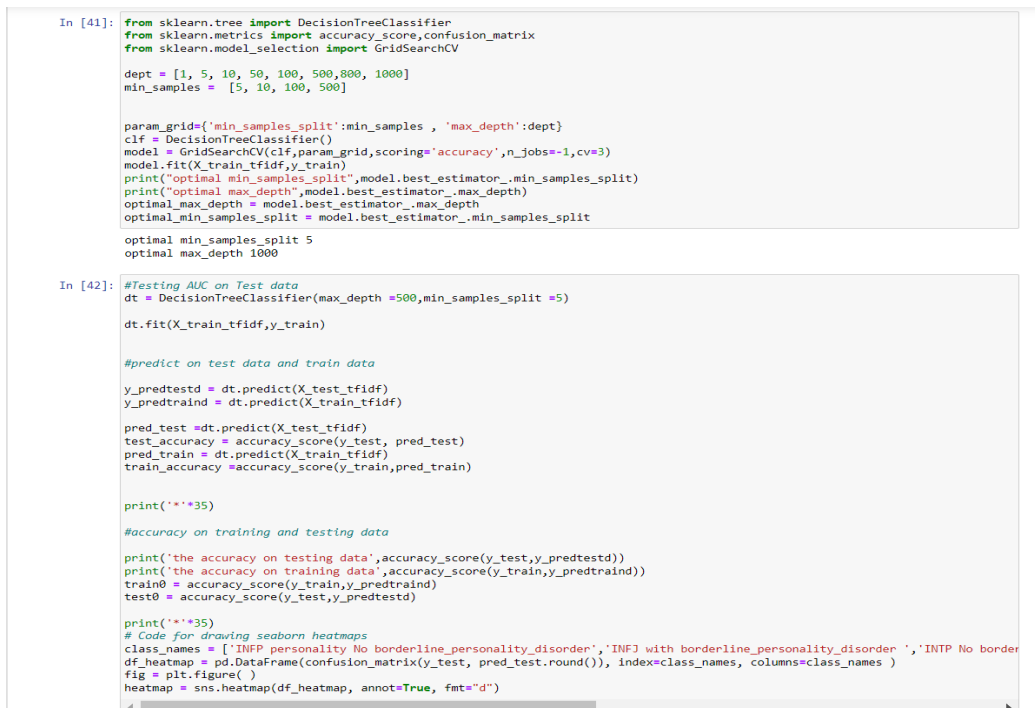


Fig 5.18

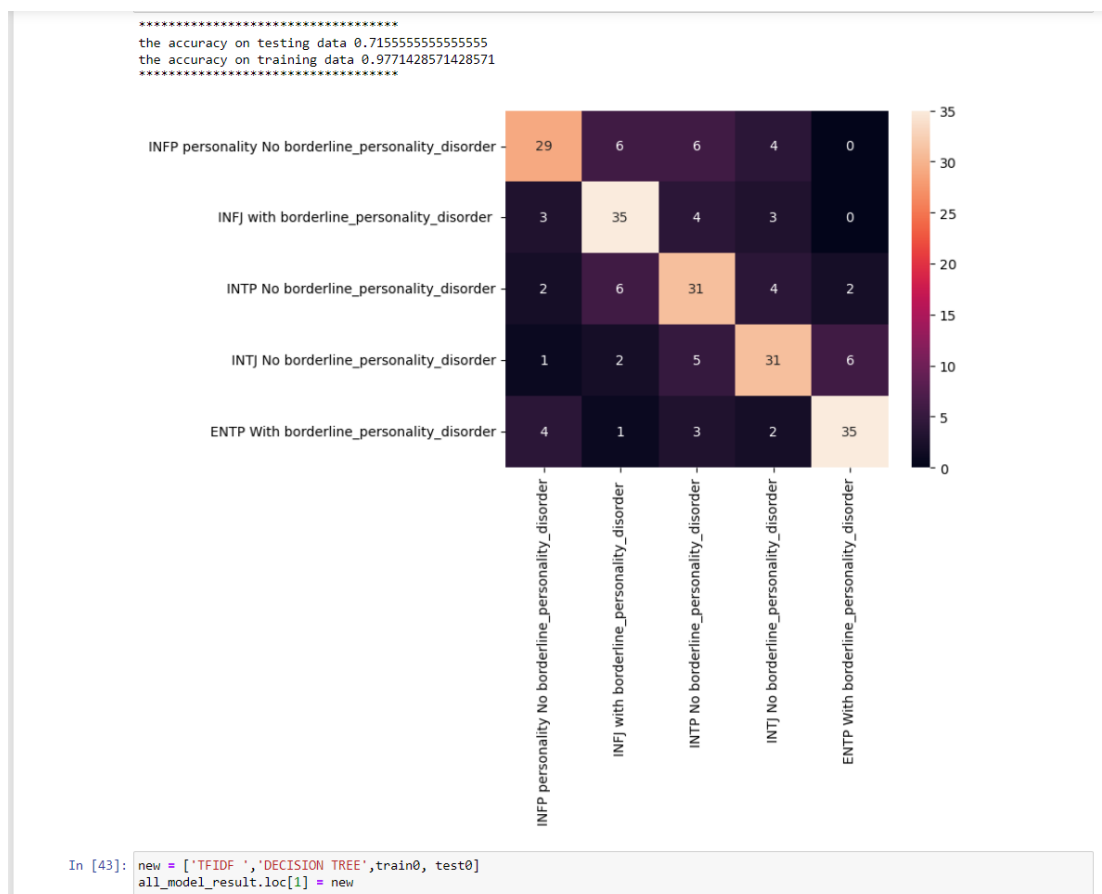


Fig 5.19

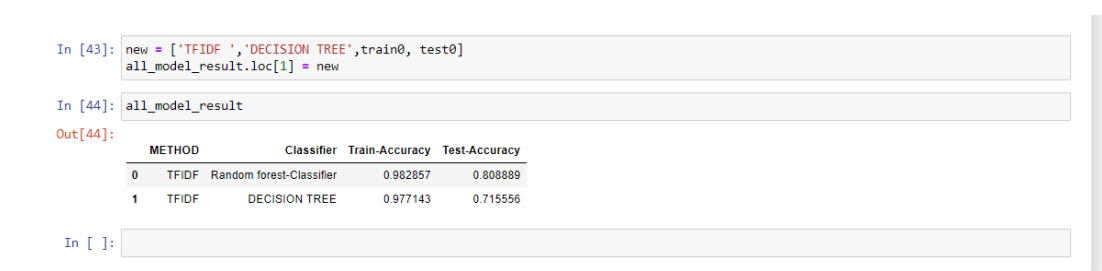


Fig 5.20

CHAPTER 6

RESULTS

6.1 RESULT ANALYSIS

The project's implementation of a personality prediction system based on YouTube comment data yielded promising results, showcasing the successful classification of users into distinct personality types. Through the analysis, dominant personality traits prevalent within the YouTube user base were identified, providing valuable insights for tailored marketing strategies and effective communication approaches. Users who engaged with the system reported an increased understanding of their own behavioral patterns and preferences, fostering heightened self-awareness. Leveraging the personality insights gained, the project highlighted the potential for enhanced personalization in marketing campaigns, enabling targeted approaches that resonate with specific personality groups and thereby improving engagement and response rates. Moreover, the system's recommendations for personality development were well-received, as users expressed a keen interest in utilizing the insights to guide their personal growth and foster positive changes in their communication styles and behavior. These findings underscore the practical significance of the project, emphasizing its potential to facilitate both individual self-improvement and the optimization of marketing strategies through personality-based communications on digital platforms like YouTube.

6.2 EVALUATION METRICS

Actual	Negative (0)	True Negative (TN)	False Positive (FP)
	Positive (1)	False Negative (FN)	True Positive (TP)
		Negative (0)	Positive (1)
		Predicted	

Fig 6.1

It is the most commonly used evaluation metrics in predictive analysis mainly because it is very easy to understand and it can be used to compute other essential metrics such as accuracy, recall, precision, etc. It is an NxN matrix that describes the overall performance of a model when used on some dataset, where N is the number of class labels in the classification problem. All predicted true positive and true negative divided by all positive and negative. True Positive (TP), True Negative (TN), False Negative (FN) and False Positive (FP) predicted by all algorithms are presented in table.

True positive (TP) indicates that the positive class is predicted as a positive class, and the number of sample positive classes was actually predicted by the model.

False negative indicates (FN) that the positive class is predicted as a negative class, and the number of negative classes in the sample was actually predicted by the model.

False positive (FP) indicates that the negative class is predicted as a positive class, and the number of positive classes of samples was actually predicted by the model. True negative (TN) indicates that the negative class is predicted as a negative class, and the number of sample negative classes was actually predicted by the modellization we mentioned above.

CONCLUSIONS AND FUTURE WORK

CONCLUSION:

In conclusion, My project is developing a system for personality prediction using machine learning and personality traits is a valuable and timely endeavor. The ability to classify individuals based on their personality traits has numerous applications in various fields, including marketing, social media, and even climate science. By leveraging machine learning algorithms, your system can help users better understand their personality types and make improvements based on the results. This approach has the potential to enhance user experiences, increase the effectiveness of marketing campaigns, and contribute to a more personalized and tailored interaction with technology.

FUTURE WORK:

Looking ahead, the project opens up several avenues for future exploration and development. One potential direction involves refining the existing personality prediction system to incorporate a broader range of online data sources beyond YouTube comments, such as social media interactions and browsing behavior, to generate more comprehensive personality profiles. Further research could also focus on the integration of advanced natural language processing techniques and sentiment analysis to enhance the accuracy of personality classifications and provide more nuanced insights into users' emotional tendencies and linguistic preferences.

REFERENCES

1. Agarwal, D., & Karthikeyan, M. (2022, April). Personality Prediction Using Machine Learning. *International Research Journal of Modernization in Engineering Technology and Science*, 04(04), 212.
2. N. T. Singh, A. Chanana, D. Jain and R. Kumar, "Personality prediction through CV analysis using machine learning techniques," 2023 Third International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT), Bhilai, India, 2023, pp. 1-6, doi: 10.1109/ICAECT57570.2023.10117883.
3. Atharva Kulkarni , Tanuj Shankarwar , Siddharth Thorat, 2021, Personality Prediction Via CV Analysis using Machine Learning, *INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT)* Volume 10, Issue 09 (September 2021).
4. H. Vijay and N. Sebastian, "Personality Prediction using Machine Learning," 2022 International Conference on Computing, Communication, Security and Intelligent Systems (IC3SIS), Kochi, India, 2022, pp. 1-6, doi: 10.1109/IC3SIS54991.2022.9885425.
5. R. K. Cherukuru, A. Kumar, S. Srivastava and V. Kumar Verma, "Prediction of Personality Trait using Machine Learning on Online Texts," 2022 International Conference for Advancement in Technology (ICONAT), Goa, India, 2022, pp. 1-8, doi: 10.1109/ICONAT53423.2022.9725910.
6. Dhokley, Waheeda and Jehangir, Randeria and Sarwar, Shaikh and Rashid, Shaikh, A Novel Approach to Predict Personality of a Person (May 7, 2021). *Proceedings of the 4th International Conference on Advances in Science & Technology (ICAST2021)*, Available at SSRN: <https://ssrn.com/abstract=3868694>
7. I. Gupta, M. Jain and P. Johri, "Smart-Hire Personality Prediction Using ML," 2023 International Conference on Disruptive Technologies (ICDT), Greater Noida, India, 2023, pp. 381-385, doi: 10.1109/ICDT57929.2023.10151367.

8. M. Karnakar, H. U. Rahman, A. B. J. Santhosh and N. Sirisala, "Applicant Personality Prediction System Using Machine Learning," 2021 2nd Global Conference for Advancement in Technology (GCAT), Bangalore, India, 2021, pp. 1-4, doi: 10.1109/GCAT52182.2021.9587693.
9. K. Raut, J. Patil, S. Wade and J. Tinsu, "Mental Health and Personality Determination using Machine Learning," 2022 7th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 2022, pp. 1231-1236, doi: 10.1109/ICCES54183.2022.9836013.
10. L. K. P. Suryapranata, G. P. Kusuma, Y. Heryadi, B. S. Abbas, Lukas and A. S. Ahmad, "Personality trait prediction based on game character design using machine learning approach," 2017 International Conference on Innovative and Creative Information Technology (ICITech), Salatiga, Indonesia, 2017, pp. 1-5, doi: 10.1109/INNOCIT.2017.8319139.