CS5710: Machine Learning

Project Part-2 (Proposal + Increment)

CLASSIFYING PERSONALITY OF TWITTER USERS USING MACHINE LEARNING TECHNIQUES

CRN: 30527

TEAM MEMBERS:

BHAVANA CHERUKUPALLI- 700748205
VARSHA ERROLLA- 700742817
SNIDHI REDDY GUDOOR- 700745499

ABSTRACT

Social media is a platform where people may showcase their identities to the public by sharing intimate information and insights into their lives. We are starting to grasp how some of this knowledge may be used to enhance how users interact with interfaces and one another. Users' personalities are of interest to us. It has been demonstrated that personality matters for a variety of interactions, including the ability to predict work happiness, the success of romantic and professional relationships, and even a preference for interfaces. Users had to take a personality test in the past to determine their personalities with any accuracy. The application of personality profiling in many social media domains became impracticable as a result. In this research, we offer a technique for successfully predicting a user's personality using the information that is made available to the public on their Twitter profile.

We will discuss the types of data gathered, our analysis procedures, and the machine learning approaches that enable us to accurately predict personality. The ramifications for social media design, interface design, and more general areas are then discussed.

LIST OF ACRONYMS AND DEFINITIONS

SNO.	ACRONYM	DEFINITION
1	STT	Speech to Text
2	TTS	Text to Speech
3	API	Application Program Interface
4	UML	Unified Modelling Language
5	JS	JavaScript
6	Gtts	Google Text to Speech
7	HTML	Hyper Text Markup Language
8	XML	Extensible Markup Language
9	Pyttxs	Python Text to Speech
10	JSON	JavaScript Object Notation

CHAPTER 1

INTRODUCTION

1.1 NECESSITY OF PERSONALITY CLASSIFICATION

Users expose a lot about themselves while constructing social networking profiles, both in what they post and how they express it. A user's profile reveals a lot about their personality through their self-description, status updates, photographs, and interests. There are connections between personality and psychological problems, career happiness and performance, and even romantic success.

Using the data people provide in their online accounts, this study aims to close the gap between personality research and social media. The Big Five Personality Index and related research on personality and social media are introduced before we get started. We next go over our experimental setup and approaches for evaluating and measuring data from Twitter profiles.

1.2 TYPES OF BIG FIVE PERSONALITY

- **Conscientiousness:** Reliable, well-organized, and persistent. High achievers, diligent workers, and good planners make up most of the conscientious population.
- **Extroversion:** Friendly, aggressive, and outgoing. Extroverts are outgoing and enthusiastic people who obtain their energy from social interactions.
- **Agreeableness:** cooperative, beneficial, and nurturing. A high agreeableness rating indicates that a person is a peacekeeper who is generally upbeat and trustworthy of others.
- **Neuroticism:** Insecure, sensitive, and anxious. Neurotics tend to have erratic moods, are uptight, and are susceptible to feeling upset.

The capacity to anticipate personality has consequences across a wide range of fields. Personality qualities and achievement in both professional and interpersonal interactions have been linked by existing studies. Insights into personality may be useful for social media technologies that aim to facilitate these interactions. Additionally, earlier research on personality and interfaces shown that users are more responsive to and more trusting of interfaces and information that are provided

from the perspective of their own personality traits (for example, introverts prefer messages delivered from an introvert's perspective). Online marketing and applications can use this information to tailor their message and how it is presented if a user's personality can be inferred from their social media profile.

The Big Five Personality Index and related research on personality and social media are introduced first. We next go over our experimental setup and approaches for evaluating and measuring data from Twitter profiles. We give findings on connections between each profile element and personality aspect to comprehend the relationship between personality and social media accounts. Using this information as a foundation, we outline the machine learning algorithms utilized for classification and demonstrate how we significantly improve upon the baseline categorization on each personality characteristic. We wrap up by talking about the ramifications of this work for social media platforms and for businesses that might use social media to learn more about the individuals they connect with.

One of the most thoroughly studied and well accepted measurements of personality structure in recent years is the "Big Five" model of personality traits. The five personality dimensions of the model—Openness, Conscientiousness, Extroversion, Agreeability, and Neuroticism—were developed. A person has ratings for each of the five variables, as shown in Fig.1.1.

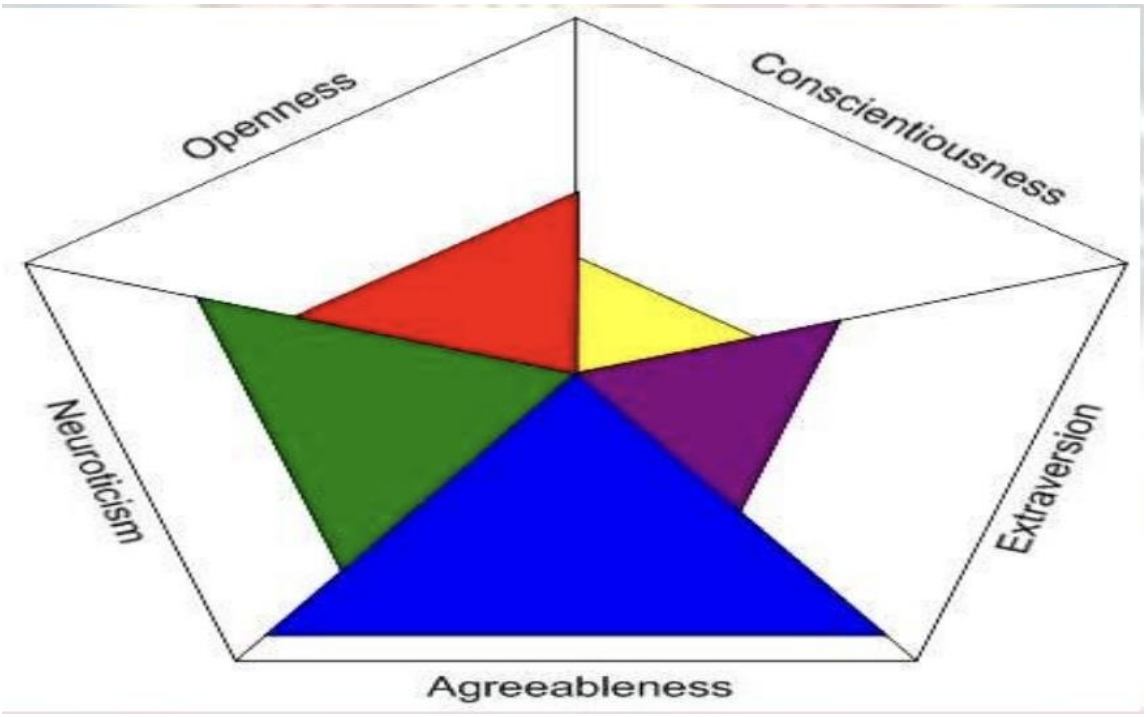


Fig.1.1: Personality Representation of an individual person

Over the past ten years, online social networking has increased significantly. A study of social networking websites in January 2005 found that there were around 115 million members across all websites on the internet. Only Twitter has more than 200 million users today, just over five years after it first launched. Users reveal a lot about themselves while constructing social networking profiles, both in the information they offer and the language they employ. A user's personality can be seen in large part through their profile through self-description, status updates, photographs, and interests.

Researchers in psychology have been trying to develop a methodical understanding of personality for many years. Researchers have established links between common personality qualities and a variety of behaviors after substantial work to create and validate a widely used personality model. Relationships between personality and psychological problems, job performance and job happiness, and even romantic success have been found.

This initiative uses data from users' online profiles to try and close the gap between personality study and social media. Whether social media profiles may foretell personality traits is the central subject of our study. If so, there may be a chance to incorporate the numerous findings regarding the significance of personality traits and behavior into users' online experiences and to use social

media profiles as a source of knowledge to learn more about particular people. If a user is more introverted or extraverted, for instance, the friend suggestion system might be adjusted to that user's preferences.

The content in users' Facebook accounts, as opposed to a "idealized" representation of themselves, reflects their true personalities, according to previous research. Given that we anticipate Twitter to share these traits and boasts a 200 million strong user base, it is the perfect venue for research.

We gave the Big Five Personality Inventory to 279 participants using a Twitter application. We gathered their 2000 most recent tweets (tweets) that were visible to the public during this process. To create a feature set, this information was quantified, aggregated, and subjected to text analysis. Using these data, we were able to develop a model that, to within 11% to 18% of the actual values, can predict personality on each of the five personality characteristics.

There are various applications for personality prediction. Research has already been done that links personality attributes to success in both personal and professional relationships. Personality insights may be useful for social media technologies that aim to facilitate these interactions. Additionally, earlier research on personality and interfaces demonstrated that users are more responsive to and more trusting of interfaces and information that are provided from the perspective of their own personality traits (for example, introverts prefer messages delivered from an introvert's perspective). Online marketing and applications can leverage social media profiles to forecast a user's personality, which they can then use to tailor their message and how it is presented.

The Big Five Personality Index and related research on personality and social media are introduced before we get started. We next go over our experimental setup and approaches for evaluating and measuring data from Twitter profiles. We give findings on connections between each profile element and personality aspect in order to comprehend the relationship between personality and social media accounts. Using this information as a foundation, we outline the machine learning algorithms utilized for classification and demonstrate how we significantly improve upon the baseline categorization on each personality characteristic. We wrap up by talking about the ramifications of this work for social media platforms and for businesses that might use social media to learn more about the individuals they connect with.

Understanding a student's personality can aid educators in assessing a student's growth in the educational setting. Knowing someone's personality allows us to easily pinpoint the traits,

mentalities, emotions, and behaviors that set them apart from others. However, traditional personality assessments need a lot of resources, including interpreters, room, and time, which is typically rather long. Word choice and word placement are only two examples of how a person's personality will influence and be related to various areas of language use.

Twitter is a social networking website that operates online and lets users post and read short messages using its linguistic features.

With the help of Twitter data and a predictive model that uses the naive bayes classifier approach to categorize Twitter users' DISC personality types, this study offers solutions to issues with using traditional personality evaluations, particularly the DISC personality type.

As training data, we created a predictive model using 9,044 tweets from 70 different Twitter accounts. Tweets need to go through a number of preprocessing steps prior to being formed and evaluated as a model. By contrasting the expert's classification findings with the model's classification results, the model's performance was evaluated, and the data prediction accuracy rate came out to be 76.19%.

Users present themselves to the public on social media by sharing intimate information and insights into their life. We are starting to comprehend how some of this data may be applied to enhance how users interact with interfaces and one another. The personality of users is something we're interested in.

Personality has been linked to a variety of interactions; it has been proven to be helpful in predicting job satisfaction, the success of personal and professional relationships, and even a preference for certain interfaces. Users formerly required to take a personality test in order to determine their personalities accurately.

The application of personality profiling in many social media domains became impracticable as a result. In this research, we describe a technique that uses publicly available data from a user's Twitter profile to reliably predict that user's personality.

We will discuss the types of data gathered, our analysis procedures, and the machine learning approaches that enable us to accurately predict personality. The ramifications for social media design, interface design, and more general areas are then discussed.

There are many individuals who can access Twitter, a popular online social media network. Twitter users converse via messages and posts. The communications are known as tweets on it.

Using a study of the tweets the user has shared, this project aims to predict the personality of the user. Numerous methods exist for predicting a person's personality; nevertheless, this study will focus on some of these methods' shortcomings. This project's goal is to give a broad overview of the approaches used to forecast a user's personality and to make comparisons between the outcomes of running the available data through various classifiers.

The purpose of this research is to identify the most accurate models for predicting the personality traits of Machiavellianism, Narcissism, Openness, Conscientiousness, Extraversion, Agreeableness, and Neuroticism based on Twitter usage and language analysis.

By attracting the attention of Internet users, social network platforms play a vital role. Many people are active on social media, according to a team of academics, so they may learn more about the many personalities of people they interact with on a regular basis. Comments, tweets, relationships with other people, and any type of text communication can be used to acquire this information. The user's online behavior on other profiles can be shown by employing these kinds of processed data.

Our study initially focuses on social behavior and linguistic practices on twitter.

Then, we anticipate the user's character by selecting the most advantageous features for each personality dimension. We suggested a structured framework based on two categories.

Based on the dataset we utilized, one is social network analysis (SNA), followed by two classes of linguistic inquiry and word counts (LIWC) and structured programming for linguistic cue extraction (SPLICE). By processing the data, we may examine some of the commonalities based on these categories. The personality traits and feature sets have almost identical relationships. Several fundamental standards make prediction simpler. We arrived at a decision to turn the data into information using many sets of categories, including network size, the six densities of tweets, profession, and the number of connections. We have investigated the predictability of the characteristics created by forecasts made utilizing personality traits.

To increase the forecast's precision and effectiveness, we have carefully examined a lot of data that is not useful for the next step. Finally, to move forward with the implementation model and explore how well we can predict personality traits from tweets, we have employed a parallel machine learning approach. We researched and selected the best algorithm to increase prediction.

In this study, we make use of information obtained from twitter. Our research has made some contributions to the field of efficient data prediction, including: first, a study of user connections and interactions in social networks; second, the use of XGBoost machine learning to clarify social network features between various categories; and third, the relationship between categories. demonstrating the superior effectiveness of SNA compared to other elements.

Before moving on to the feature selection and training stages, the dataset acquired from my Personality was preprocessed. Using OpenNLP, we performed pre-processing on the dataset. To begin with, we separated every sentence's final word using tokenization, considering punctuation and the grouping of related terms. Then, lowercase letters, symbols, names, URLs, and other characters were eliminated.

The existence and behavior of other users on the social media platform have a direct impact on how individuals behave there. Through the groups, this may have an impact on the production of new information or actions. To comprehend how these behaviors occur and impact the individuals, numerous programs have been created. In our study, the dataset is divided into two categories. The first category is text feature extraction, which is used to analyze the language of Twitter users and includes the features of topic and expression counts. The LIWC (Linguistic Inquiry and Word Count), a tool widely used in psychological research, may be utilized to comprehend human psychology.

The major purpose of LIWC2015 was to analyze files quickly and effectively in numerous languages[9].

SPLICE (Structured Programming for language Cue Extraction), a language analysis tool, is currently in the experimental phase. When fully formed, it can be used to forecast personalities. Utilizing NLTK (Natural Language ToolKit), features are extracted. A nlp library called NLTK includes packages that help computers interpret human communication. The package comprises extraction, stemming, lemmatization, tokenization, and POS tagging.

Human behavior is primarily determined by personality. Interaction and preference patterns are influenced by personality. To determine one's personality, a personality test must be taken. Users can express themselves to a global audience via social media. It is possible to gather personal information about social media users from their posts. According to text provided by Twitter users, this experiment employs text classification to forecast personality. Indonesian and English are the languages spoken. Support Vector Machine, K-Nearest Neighbors, and Naive Bayes are

the classification techniques used. Based on test findings, Naive Bayes performed a little bit better than the competition.

The Big Five model is the only one that properly comprehends the connection between personality and academic behavior.

Using factor analysis of word descriptions of human behavior, this model was developed by numerous separate research teams. These researchers started by examining connections between a wide range of word descriptions that describe personality traits. They employed component analysis to categorize the remaining attributes (using information primarily based on people's estimates, in self-report questionnaires and peer ratings) to identify the fundamental factors underlying personality. They decreased the lists of these descriptors by 5–10 fold.

Ernest Tupes and Raymond Christal presented the first version of the concept in 1961, but it wasn't until the 1980s that it was accepted by academics. J.M. Digman improved his five-factor model of personality in 1990, and Lewis Goldberg expanded it to the highest level of organizational hierarchy. It has been discovered that most well-known personality qualities fall under these five broad areas, which are thought to constitute the fundamental building blocks of all personality traits.

CHAPTER 2

LITERATURE SURVEY

Our way of living depends greatly on our personalities. It shapes the way we act, speak, and react, expresses our preferences, and even has an impact on our mental health.

Personality analysis is a human intuitive skill that is used daily with many people and for a wide variety of purposes. Personality profiling has many practical applications, including mental health screening exams, screening during job interviews, advising authors on the interaction of characters that readers like reading about, and friend recommendations. In this project, a personality profile of a person is developed by the study of material that person has produced, such as an essay, tweet, or blog post. The primary focuses of the research are the types of data collected, text preprocessing strategies, and machine learning algorithms employed to calculate personality ratings. The comparison of various feature vector combinations and machine learning models, as well as the deployment of solutions, have been discussed.

The techniques used in this article enabled accuracy up to 88% [1].

The process of classifying massive collections of unlabeled documents involves assigning each one to a predetermined category. In this effort, a Deep Recurrent Neural Network-based classification scheme for Bangla newspapers was put out. The gathered news data was initially preprocessed. Training data was fitted into the model during the design of the model's architecture. Finally, the accuracy and F1-score on the testing dataset were calculated to assess the model performance. In the classification of Bangla text, the Deep Recurrent Neural Network with BiLSTM achieved 98.33% accuracy, outperforming other well-known classification techniques [2].

Numerous universities in Bangladesh accept thousands of students each year. A significant portion of them graduate with grades that are below average, which has an impact on their jobs. They can take necessary steps to improve their grades by knowing their grades before the final exam. In the context of Bangladesh's private universities, this article has suggested various machine learning methodologies for estimating a student's course grade. Seven different classifiers have been trained using various factors that influence a student's performance, including Support Vector Machine (SVM), K-Nearest Neighbor (KNN), Logistic Regression, Decision Tree, AdaBoost,

Multilayer Perceptron (MLP), and Extra Tree Classifier. These classifiers were used to divide the students' final grades into four quality classes: Fantastic, Good, Bad, and Fail.

To achieve superior results, the outputs of the basis classifiers have then been aggregated using the weighted voting method. And in this case, this study's accuracy was 81.73%, with the base classifiers outperforming the weighted voting classifier [3].

Humans can concentrate their consciousness on a particular event or piece of information by paying attention. The degree of attention indicates how intensely one is paying attention to a given instance or piece of knowledge. A person's level of attention is influenced by their surroundings and mental health. This study included a variety of machine learning methods for estimating the degree of visual concentration in people's attention. Data from survey reports (environmental data) and eyeball data (eyeball movement, reading length, and head tilt while reading an article) were combined to construct an attention level dataset for each participant.

. Eight distinct classifiers, including: To categorize the participant's attention level into three groups—High, Average, and Low—logistic regression, support vector machine (SVM), decision tree, K-nearest neighbor (KNN), adaBoost, multilayer perceptron (MLP), extra tree classifier, and voting classifier were used. The best level of accuracy in predicting these classes was 96%, achieved by the Logistic Regression, surpassing other methods. A 95% accuracy rate was reached using the weighted voting aggregate classifier [4].

The fields of social sciences and natural language processing are increasingly interested in automated personality prediction from social media. The few datasets that are publicly accessible, however, are small and have a narrow range of topics because of expensive labeling costs and privacy concerns. We solve this issue by introducing a sizable dataset collected from Reddit, a source hitherto ignored for personality prediction.

The dataset includes a wealth of features for over 9k users and is tagged with Myers-Briggs Type Indicators (MBTI). We do a preliminary feature analysis, which reveals notable disparities between the MBTI dimensions and poles. The dataset is also used to train and assess benchmark personality prediction models, with macro F1-scores on the individual dimensions ranging from 67% to 82% and an accuracy of 82% for exact or one-off accurate type prediction. These positive results are online with the validity of standardized exams [5].

Human conduct is fundamentally based on personality. An individual's tastes and interactions are influenced by their personality. To determine one's personality, one must take a personality test. In this experiment, content submitted by Twitter users is classified to predict personality. The classifiers Naive Bayes, K-Nearest Neighbors, and Support Vector Machine are used.

According to test results, Naive Bayes fared marginally better than the competition [6].

An individual's personality can be thought of as certain characteristics that influence what they prefer in life. Numerous forms of interactions have been proven to be influenced by personality, including the prediction of recommendations for movies, social interactions, music, criminal activity, and the relationship between personality and job performance. To help people have a better experience with computerized user interfaces and to aid others in researching different personality preferences, social media personality prediction has become one of the most popular topics among researchers and numerous commercial businesses. To predict personality from social media, numerous algorithms have been run. In this experiment, we evaluated how well several classifiers—including Naive Bayes, Random Forest, and others—predicted the personalities of Kaggle users. The user data of Kagglers were taken from the Kaggle Repository, evaluated, and then categorised in the automatic personality prediction based on the user profile and comments[7].

According to psychological studies, specific personality traits and language conduct are related. Statistical natural language processing methods can be used to successfully model this correlation. \ The generality and statistical power of the results are constrained by both variables. In this study, we investigate the potential for large-scale, open-vocabulary personality assessment via social media. We examine the variables that are predictive of different personality types and present a fresh corpus of 1.2 million English tweets that have been gendered and Myers-Briggs personality type annotated.Our research demonstrates that two of the four personality traits can be accurately predicted using social media data [8].

Large amounts of pertinent training data are required for the automatic classification of personality from language, which poses two possible issues. First off, asking the author or speaker about their personality can be intrusive and expensive, especially in delicate situations. Second, difficulties with context or genre may make training resources less effective for broader personality classification. Utilizing outside judges rather than the text's author is one way to address the first problem. In this project, we examine how helpful these personality perceptions

are for teaching a classifier to distinguish between various language genres. We investigate the projection of personality through 11 language elements after dismal cross-training outcomes. While some of the distinctions we uncover are across the genres, others show that personality is in fact displayed differently across contexts. When using resources from several domains for computational personality recognition, it is obvious that caution is required [9].

Although predicting personality is crucial for social applications that enable human-centered activities, previous modeling techniques using users' written text required too much input data to be properly applied in the context of social media. We want to create a model that works for most Twitter users while also significantly reducing the amount of data needed for personality modeling. Regression using Gaussian Processes is combined with Word Embedding features in our model. Our model outperforms state-of-the-art methods in accuracy with 8 times less data, according to an analysis of over 1.3K Twitter users [10].

It has been shown that a lot of facets of people's life are intricately linked to their professions. In this project, we investigate the distinctive traits of significant occupational categories using tweets. We gather several types of user data from various social media networks. We learn about users' skills from their LinkedIn profiles. The extraction of professions from crowd-sourced data uses a soft clustering approach to get beyond the ambiguity of self-reported data. There are eight sorts of jobs that are taken out: marketing, administration, startups, editors, software engineers, public relations, office clerks, and designers.

The World Well-Being Project from the University of Pennsylvania submitted this article as part of the shared work for "CLPsych 2015," which is a system description and report. The shared task's objective was to automatically identify Twitter users who had either post-traumatic stress disorder (PTSD) or depression according to their self-reported health status. Textual elements from Twitter posts are combined with user metadata in our system. We look at various word clustering algorithms to condense the feature space and prevent data sparsity. We investigate the application of linear classifiers with various feature sets as well as a combination of linear ensemble.

This strategy is easily adaptable to other situations because it is 12 agnostics of illness-specific elements, such as lists of medications. Our method produced the best results at 0.1 false positive rates and placed second overall in all tasks for average precision [12].

In many information retrieval activities, including search, query recommendation, automatic summarization, and picture finding, determining the semantic similarity of texts is crucial. Numerous strategies have been put forth, including those based on lexical matching, custom-made patterns, syntactic parse trees, external sources of structured semantic information, and distributional semantics. Furthermore, it is not reasonable to anticipate that all situations and domains will be covered by custom patterns and external sources of structured semantic knowledge. Finally, parse tree-based techniques are limited to texts that are syntactically sound and often only include one phrase [13].

People's personalities influence some of the photographs they publish on social media. In this investigation, we examine the relationship between Twitter profile photos and user personalities. In our primary analysis, we make use of profile pictures from more than 66,000 individuals, whose personalities we infer from their tweets. To make our study more comprehensible, we concentrate on aesthetic and morphological traits of the face while taking into account demographic diversity in personality traits and image aspects. Our findings demonstrate that personality types differ significantly in the type of profile photo they choose, and that these differences can be used to accurately predict personality traits. Users who are more conscientious and pleasant, for instance, show more positive feelings in their profile photographs, whilst those who are more open, on the other hand, choose more aesthetically pleasing photos[14].

We now introduce the CLiPS Stylometry Investigation (CSI) corpus, a fresh Dutch collection of evaluations and articles authored by college students. It is intended to perform a variety of tasks, including the detection of topic, genre, authorship, personality, sentiment, deception, age, and gender. Its projected yearly expansion with new students each year is a significant additional benefit. Currently, there are 749 papers in the corpus, totaling around 305,000 tokens. The average length of a review is 128 tokens, whereas that of an essay is 1126 tokens. The corpus will be made accessible on the CLiPS website (www.clips.uantwerpen.be/datasets), and academic research projects may freely use it. Deception detection involves automatically determining if a text is true or false, in this example by analyzing the author's writing style. Dutch has never been examined for this task. We used the SVM in a supervised machine learning experiment [15].

CHAPTER 3

SYSTEM ANALYSIS AND DESIGN

3.1 EXISTING SYSTEM

There are various applications for personality prediction. Research has already been done that links personality attributes to success in both personal and professional relationships. Personality insights may be useful for social media technologies that aim to facilitate these interactions. Additionally, earlier research on personality and interfaces demonstrated that users are more responsive to and more trusting of interfaces and information that are provided from the perspective of their own personality traits (for example, introverts prefer messages delivered from an introvert's perspective). Online marketing and applications can customize their message and presentation if a user's personality can be inferred from their social media profile.

3.2 PROPOSED SYSTEM

In this project author is asking to predict human personality by taking 5 features such as Openness, Agreeable, Neuroticism, Extroversion, and Conscientious and this feature can be identified by social media dataset called ‘Twitter Profile’.

Openness means intelligent peoples who express their view in bold or open manner. This user expression can be identified by analyzing his twitter profile and twitter messages, if person is intelligent then he will use open (open words also called as swear words) or bold words in his tweets. By looking for such words we can identify this person as Openness personality. LIWC dictionary contains all open or swear words by applying this dictionary on tweets messages we can predict Openness personality score. If predicted score > 0.1 then this person will put under this category.

Agreeable means peoples who use words such as ‘am, will have and this words also refers as ARTICLES or AUXILIARY VERBS’ etc. will come in this category. MRC dictionary contains all words of this categories and by applying this dictionary on user’s tweets we can predict person category as agreeable.

Neuroticism means peoples in this category is consider as sentiment or emotion, peoples who use words such as ‘ugly, nasty, sad’ etc. will come under this category.

Extroversion means peoples of this category are friendly and person who has many numbers of friends or followers or following in twitter profile will comes under this category.

Conscientious means peoples who express hard working ideas in their post will come under this category.

So, by analyzing above 5 features O (openness), C (Conscientious), E (Extroversion), A (Agreeable), N (Neuroticism) from twitter profile and post we can predict personality of a person.

We will find average of each feature from tweets and then apply Pearson Correlation formula to get scores for all five features. If score > 0.1 for any feature, then person belongs to that category. If person has 0.1 value for more than 1 features, then that person personality belongs to that many categories. For example, same person can be predicted as openness and conscientious etc.

All features' values we will apply using SVM, Random Forest, Naïve Bayes & Logistic Regression algorithms to calculate accuracy of dataset and algorithms.

CHAPTER 4

SYSTEM REQUIREMENTS & SPECIFICATIONS

4.1 ALGORITHM

4.1.1 Support Vector Machine

Support Vector Machine (SVM) is a supervised machine learning algorithm that can be used for both classification and regression challenges. In the SVM algorithm, we plot each data item as a point in n-dimensional space with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyperplane that differentiates the two classes very well. The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane. SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine. Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyperplane. SVM can be of two types:

- Linear SVM: Linear SVM is used for linearly separable data, which means if a dataset can be classified into two classes by using a single straight line, then such data is termed as linearly separable data, and classifier is used called as Linear SVM classifier.
- Non-linear SVM: Non-Linear SVM is used for non-linearly separated data, which means if a dataset cannot be classified by using a straight line, then such data is termed as non-linear data and classifier used is called as Non-linear SVM classifier.

4.1.2 Random Forest

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. Before understanding the working of the random forest, we must investigate the ensemble technique. Ensemble simply means combining multiple models. Thus, a collection of models is used to make predictions rather than an individual model.

Ensemble uses two types of methods:

1. Bagging– It creates a different training subset from sample training data with replacement & the final output is based on majority voting. For example, Random Forest.

2. Boosting– It combines weak learners into strong learners by creating sequential models such that the final model has the highest accuracy. For example, ADA BOOST, XG BOOST

Important Features of Random Forest

1. Diversity- Not all attributes/variables/features are considered while making an individual tree, each tree is different.

2. Immune to the curse of dimensionality- Since each tree does not consider all the features, the feature space is reduced.

3. Parallelization-Each tree is created independently out of different data and attributes. This means that we can make full use of the CPU to build random forests.

4. Train-Test split- In a random forest we don't have to segregate the data for train and test as there will always be 30% of the data which is not seen by the decision tree.

5. Stability- Stability arises because the result is based on majority voting/ averaging.

4.1.3 Naive Bayes

Naive Bayes algorithm is a supervised learning algorithm, which is based on Bayes theorem and used for solving classification problems. It is mainly used in text classification that includes a high-dimensional training dataset. Naïve Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions. It is a probabilistic classifier. The Naïve Bayes algorithm is comprised of two words Naïve and Bayes, which can be described as:

- Naive: It is called Naïve because it assumes that the occurrence of a certain feature is independent of the occurrence of other features. Such as if the fruit is identified on the bases of color, shape, and taste, then red, spherical, and sweet fruit is recognized as an apple...
- Bayes: It is called Bayes because it depends on the principle of Bayes' Theorem Working of Naive Bayes Algorithms

1. Convert the given dataset into frequency tables.

2. Generate Likelihood table by finding the probabilities of given features.

3. Now, use Bayes theorem to calculate the posterior probability.

Applications of Naïve Bayes Classifier:

- It is used for Credit Scoring.

• It is used in medical data classification.

• It can be used in real-time predictions because Naïve Bayes Classifier is an eager learner.

• It is used in Text classification such as Spam filtering and Sentiment analysis.

Types of Naïve Bayes Model: There are three types of Naive Bayes Model, which are given below: • Gaussian: The Gaussian model assumes that features follow a normal distribution. This means if predictors take continuous values instead of discrete, then the model assumes that these values are sampled from the Gaussian distribution.

• Multinomial: The Multinomial Naïve Bayes classifier is used when the data is multinomial distributed. It is primarily used for document classification problems, it means a particular document belongs to which category such as Sports, Politics, education. The classifier uses the frequency of words for the predictors.

• Bernoulli: The Bernoulli classifier works like the Multinomial classifier, but the predictor variables are the independent Booleans variables. Such as if a particular word is present or not in a document. This model is also famous for document classification tasks.

4.1.4 Logistic Regression

Logistic regression, despite its name, is a classification model rather than regression model. Logistic regression is a simple and more efficient method for binary and linear classification problems. It is a classification model, which is very easy to realize and achieves very good performance with linearly separable classes. The logistic regression model, like the Adaline and perceptron, is a statistical method for binary classification that can be generalized to multiclass classification. Scikit-learn has a highly optimized version of logistic regression implementation, which supports multiclass classification task.

• Logistic regression predicts the output of a categorical dependent variable. Therefore, the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1.

• Logistic Regression is much like the Linear Regression except that how they are used. Linear Regression is used for solving Regression problems, whereas Logistic regression is used for solving the classification problems.

• In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1).

• The curve from the logistic function indicates the likelihood of something such as whether the cells are cancerous or not, a mouse is obese or not based on its weight, etc.

• Logistic Regression is a significant machine learning algorithm because it can provide probabilities and classify new data using continuous and discrete datasets.

• Logistic Function (Sigmoid Function):

• The sigmoid function is a mathematical function used to map the predicted values to probabilities.

• The value of the logistic regression must be between 0 and 1, which cannot go beyond this limit, so it forms a curve like the "S" form. The S-form curve is called the Sigmoid function or the logistic function.

• In logistic regression, we use the concept of the threshold value, which defines the probability of either 0 or 1. Such as values above the threshold value tends to 1, and a value below the threshold values tends to 0.

Assumptions for Logistic Regression:

• The dependent variable must be categorical in nature.

• The independent variable should not have multi-collinearity. Type of Logistic Regression: Based on the categories, Logistic Regression can be classified into three types:

• Binomial: In binomial Logistic regression, there can be only two possible types of the dependent variables, such as 0 or 1, Pass or Fail, etc.

• Multinomial: In multinomial Logistic regression, there can be 3 or more possible unordered types of the dependent variable, such as "cat", "dogs", or "sheep"

• Ordinal: In ordinal Logistic regression, there can be 3 or more possible ordered types of dependent variables, such as "low", "Medium", or "High".

4.2 DESIGN

4.2.1 System Architecture

The above Fig.4.1 represents system Architecture which includes various phases after implementing all the above algorithms, accuracy of each algorithm is configured and visualized

using a graph. Such that the optimistic algorithm is further used to get the final prediction of the Twitter users.

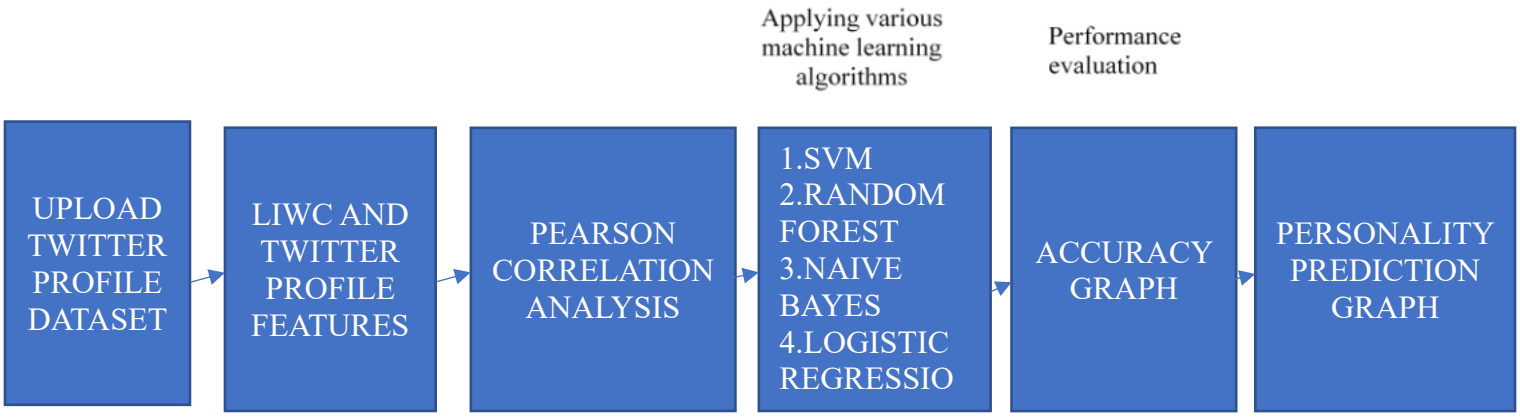


Fig.4.1: System Architecture

4.2.2 Use Case Diagrams

Fig 4.2 illustrates the relationship between the actor and various systems altogether. The first three systems deal with manipulation of data consumed and systems contribute to produce desired results from the extracted data.

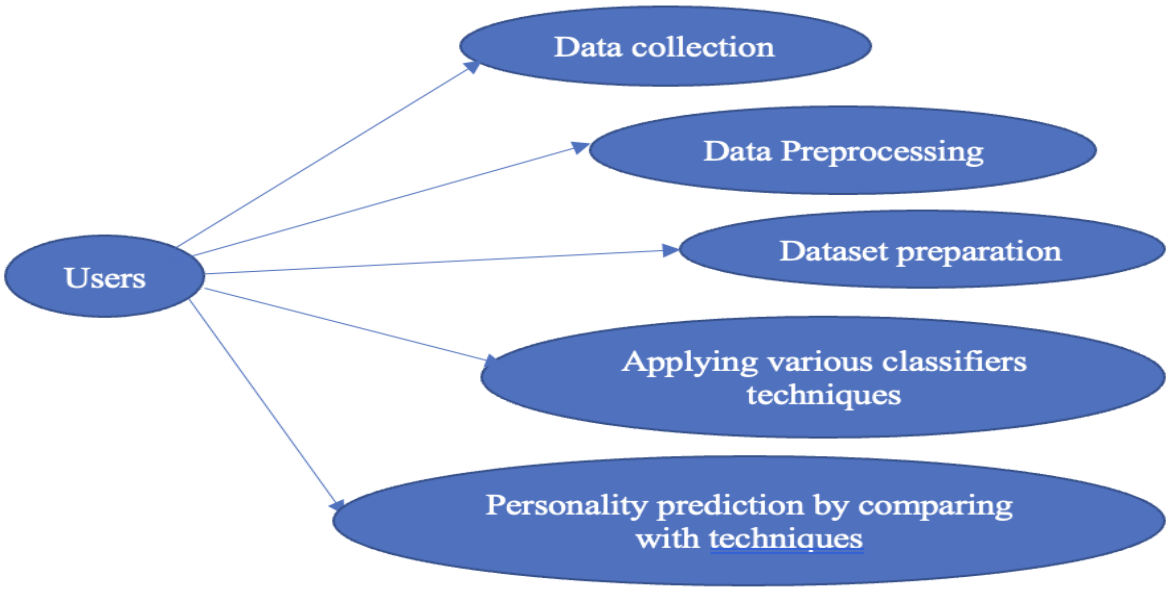


Fig.4.2: Use Case Diagram

4.2.3 Class Diagram

The following Fig.4.3 illustrates a class diagram in the Unified Modeling Language (UML) which is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

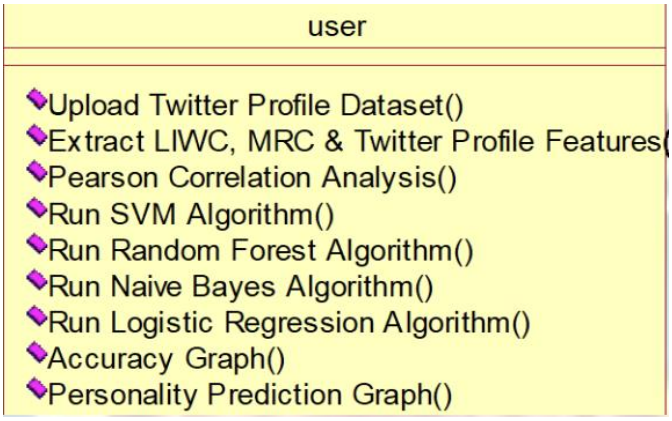
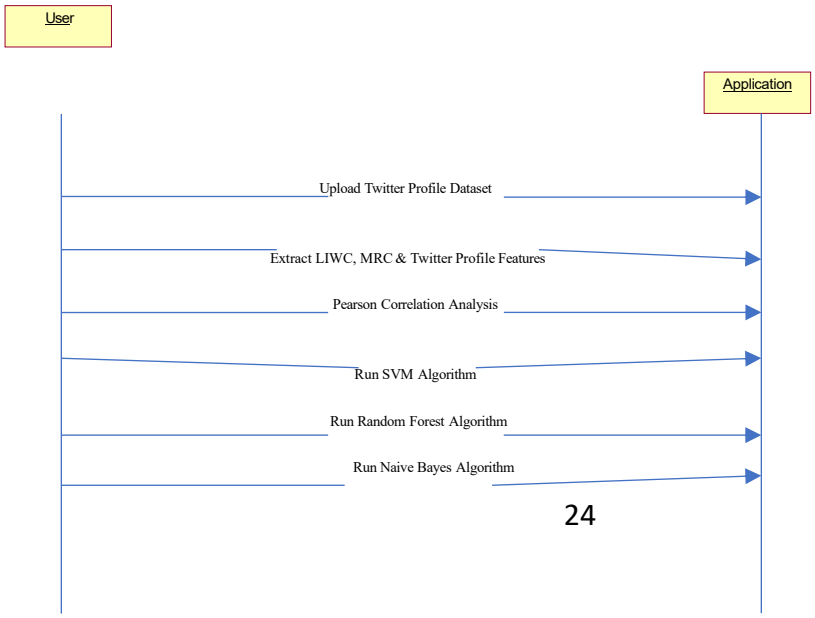


Fig.4.3: Class Diagram

4.2.4 Sequence Diagram

The following Fig.4.4 represents a sequence diagram in Unified Modeling Language (UML) which is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.



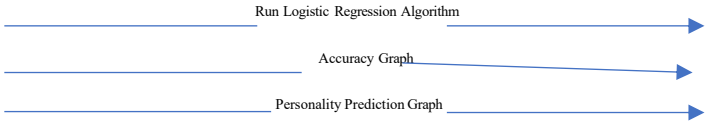


Fig.4.4: Sequence Diagram

4.1.1 Collaboration Diagram

The Fig.4.5 represents collaboration diagram which is used to show the relationship between the objects in a system. Both the sequence and the collaboration diagrams represent the same information but differently. Instead of showing the flow of messages, it depicts the architecture of the object residing in the system as it is based on object-oriented programming. An object consists of several features. Multiple objects present in the system are connected to each other. The collaboration diagram, which is also known as a communication diagram, is used to portray the object's architecture in the system.

The collaborations are used when it is essential to depict the relationship between the objects. Both the sequence and collaboration diagrams represent the same information, but the way of portraying it quite different. The collaboration diagrams are best suited for analyzing use cases. Following are some of the use cases enlisted below for which the collaboration diagram is implemented:

1. To model collaboration among the objects or roles that carry the functionalities of use cases and operations.

2. To model the mechanism inside the architectural design of the system.

3. To capture the interactions that represent the flow of messages between the objects and the roles inside the collaboration.

4. To model different scenarios within the use case or operation, involving a collaboration of several objects and interactions.

5. To support the identification of objects participating in the use case. In the collaboration diagram, each message constitutes a sequence number, such that the top-level message is marked as one and so on. The messages sent during the same call are denoted with the same decimal prefix, but with different suffixes of 1, 2, etc. as per their occurrence.

1: Upload Twitter Profile Dataset
2: Extract LIWC, MRC & Twitter Profile Features 3: Pearson Correlation Analysis
4: Run SVM Algorithm
5: Run Random Forest Algorithm 6: Run Naive Bayes Algorithm
7: Run Logistic Regression Algorithm 8: Accuracy Graph

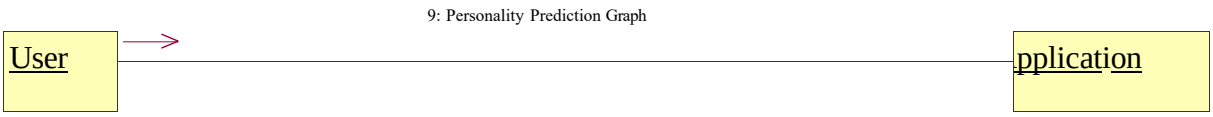


Fig.4.5: Collaboration Diagram

4.3 MODULES

4.3.1 Data Collection

Twitter API is used to collect the efficient number of tweets in this project. While collecting tweets, Turkish language for tweets and username filters are applied.

We need personality cues to predict the level of suitability for each personality traits. To calculate correlations between each personality cues and each personality traits, we need to have actual test results that are answered by different real Twitter users. There were 51 volunteers who are active Twitter users and answered the Big Five Personality test. If test the score of a person is lower than or equal to 15 for a specific trait, this person is not suitable at all for this personality trait; if the score is between 16 and 20, this person is not suitable for this personality trait; if the score is between 21 and 30, this person is suitable for this personality trait; if the score is higher than or equal 31, this person is very suitable for this personality trait.

If the score is between 21 and 30, then that person is suitable for a particular personality trait; if the score is higher than or equal 31, then that person is very suitable for personality trait.

There is a corpus in a previous study that covers the most used Turkish words used in Twitter. It contains 103,000 words in total. We used this corpus to rearrange the LIWC dictionary Since a new dictionary with as many Turkish words as possible should be included to prepare the most suitable dictionary and to obtain the most successful result in the project. After collecting words from this corpus, they should be preprocessed to use.

4.3.2 Data Preprocessing

Collected tweets are in the JSON format. It includes various kinds of information that we might not need during our analysis. We only need the word content in tweets, but they include retweets or links. Besides these, people prefer to use daily language, abbreviations for some words or they misspell some words unintentionally. Before applying word stemming, such irregularities should be resolved. Before using morphological analyzer, some items must be removed from the text of tweet.

If text contains links that must be removed, because this study aimed that personality was assessed by using words not using videos, pictures, or contents. After all unnecessary items are removed, the next step is stemming by using the Zemberek library. There is a class called morphology that provides morphological analysis, morphological ambiguity resolution, and word generation; it also contains its own normalizer and analyzer methods. Normalizer method provides correcting misspelled words and analyzer method provides stemming.

The LIWC dictionary is developed for English words, however it is not available for Turkish words. We used the "google trans" library in python to translate each English word in this dictionary to Turkish. There are 4,328 words that are translated from English to Turkish. After their translation, each word is controlled for correctness of translation and the suitability between the meaning of each word and the groups. There are some words that need to be added or deleted, also some words that need to be added to default dictionary of morphology class.

Commonly used Turkish words were processed by using the Zemberek stemming method. Before esteeming, there were more than one word with the same root that has the inflection. We removed these words and then stemming was performed. As a result of stemming, there are 10.000 words in our corpus. We used this corpus to complete the missing words in the Turkish LIWC dictionary.

4.3.3 Dataset Preparation

Dataset contains frequencies of each word groups and personality test scores of each user. Each tuple has frequencies of 41 different word groups and the scores of 5 different personality traits. There are 51 instances (persons) in our final dataset. The first step of data preparation is normalizing the observed frequencies of word groups. The normalization was done by Equation 1 in which the actual value is divided by total number of grouped words. Before applying normalization, the frequency values are very different from each other, and the dataset has uneven distribution. Normalization was applied as row (tuple)-based.

4.3.4 Machine Learning Technique

Before evaluation of machine learning models, a statistical feature selection was employed. It is a model provided in the "Sci-kit Learn" library of Python. This function takes two parameters: a score function and the number of features that is wanted to be selected. In our project, it uses the Chi2 as a score function and returns the features with the highest scores. By applying this feature selection, only 15 of word groups that provide the highest contribution to personality trait are

used in the prediction phase.

We need to decide the total number of tweets to be used in analysis before applying machine learning models. For this purpose, we prepared four different datasets. Two of them include the latest 25 and 50 tweets of users, the others include the randomly chosen 25 and 50 tweets. We did not prepare datasets more than 50 tweets, because very few people have 50 or more tweets in the dataset. Due to page limits, we could not give all the results of experiments.

As a result of categorization, there are two values for the target feature: "suitable" and "not suitable". The "suitable" category represents the personality scores in the range between [0-21] and the "not suitable" represents the scores between 22 to 50After the categorized (binary) target feature was prepared, classification models were generated. As mentioned, there are five different personality traits that are expected to be predicted. Different classification models are successful for different personality traits.

Since each of them has different sample data distribution. Therefore, each personality trait is predicted by using a different machine learning model. For this purpose, we experimented kNN, decision tree (DT), random forest (RF), AdaBoost, stochastic gradient descent (SGD), gradient boosting (GB) and SVM learning models. The most successful models in our experiments were AdaBoost, SGD, GB and SVM.

In the evaluation of machine learning models, a leave one out cross validation was applied due to having only 51 instances. Because there is a quite small dataset for other training methods. It basically takes only one sample as test sample and uses the rest of samples as the training data. So, each instance in dataset is used as a test data in this validation type.

Support Vector Machine (SVM) is a supervised machine learning algorithm that can be used for both classification and regression challenges. In the SVM algorithm, we plot each data item as a point in n-dimensional space with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper- plane that differentiates the two classes very well. The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane. SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine. Consider the below diagram in which there are two different categories that are classified using a decision

boundary or hyperplane.

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. Before understanding the working of the random forest, we must investigate the ensemble technique. Ensemble simply means combining multiple models. Thus, a collection of models is used to make predictions rather than an individual model.

Naive Bayes algorithm is a supervised learning algorithm, which is based on Bayes theorem and used for solving classification problems. It is mainly used in text classification that includes a high-dimensional training dataset. Naïve Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions. It is a probabilistic classifier.

Logistic regression, despite its name, is a classification model rather than regression model. Logistic regression is a simple and more efficient method for binary and linear classification problems. It is a classification model, which is very easy to realize and achieves very good performance with linearly separable classes. The logistic regression model, like the Adaline and perceptron, is a statistical method for binary classification that can be generalized to multiclass classification. Scikit-learn has a highly optimized version of logistic regression implementation, which supports multiclass classification task.

4.4 SYSTEM REQUIRMENTS

4.4.1 Hardware Requirements

Processor : Intel Core i3

Speed : 1.1 GHZ

Primary Memory : 8 GB RAM

Hard Disk : 500 GB

4.4.2 Software Requirements

Languages Used : Python

Platform : Windows 10

Tools Used : Google Chrome

Additional Tools : IDLE for Python (Version-3.7.4)

4.5 TESTING

4.5.1 Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

4.5.2 Integration testing

Integration tests are designed to test integrated software components to determine if they run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

4.5.3 Functional testing

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing.

4.5.4 System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

4.5.5 White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure, and language of the software, or at least its purpose. It is 27 purposes. It is used to test areas that cannot be reached from a black box level.

4.5.6 Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

4.5.7 Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

CHAPTER 5

SOURCE CODE

```
from tkinter import messagebox
from tkinter import *
from tkinter import simpledialog
import tkinter
from tkinter import filedialog
import matplotlib.pyplot as plt
import numpy as np
from tkinter.filedialog import askopenfilename import
numpy as np
import pandas as pd
from sklearn import
*
from sklearn.metrics import confusion_matrix
from sklearn.model_selection import train_test_split
from sklearn import svm
from sklearn.metrics import accuracy_score  from
sklearn.metrics import classification_report
from sklearn.ensemble import RandomForestClassifier import json
```

```
import os
import re
from scipy.stats import pearsonr
import numpy as np
import pandas as pd
from sklearn.naive_bayes import GaussianNB
from sklearn.linear_model import LogisticRegression  #all packages import in above ines main =
tkinter.Tk()
main.title("Predicting Personality") #designing main screen
main.geometry("1300x1200")
global filename
mrc = []   #list of MRC and LIWC words
liwc = []
emotion = []
global
openness
global
agreeable
global conscientious
global extroversion
global neuroticism
global open_count
global agree_count
global ext_count
global neu_count
global cons_count
global X, Y, X_train, X_test, y_train, y_test global
svm_acc, random_acc, naive_acc, lra_acc def
traintest(train):
    X = train.values[:,
    0:4] Y =
    train.values[:, 5]
    X_train, X_test, y_train, y_test = train_test_split( X,
    Y, test_size = 0.2, random_state = 0)
    return X, Y, X_train, X_test, y_train, y_test
def generateModel(): #method to read dataset values which contains all five features data global X, Y,
    X_train, X_test, y_train, y_test
```

```
    train = pd.read_csv("dataset.txt")
    X, Y, X_train, X_test, y_train, y_test = traintest(train)
with open("LIWC.dic", "r") as file:  #reading LIWC dictinary for line
    in file:
        line = line.strip('\n')
        line = line.strip()
        liwc.append(line.lower())
with open("MRC.txt", "r") as file: #reading MRC dictionary for
    line in file:
        line = line.strip('\n')
        line = line.strip()
        mrc.append(line.lower())
with open("emotions.txt", "r") as file: #reading emotion word for line
    in file:
        line = line.strip('\n')
        line = line.strip()
        emotion.append(line.lower())
def opennessFunction(words): #calculate number of openness words from tweets count = 0.0
    for i in range(len(liwc)):
        if words.find(liwc[i]) != -1:
            count = count + 1
    if count > 0:
        count = count/float(len(liwc))
    return count
def agreeableFunction(words): #calculate number of agreeable words from tweets count = 0.0
    for i in range(len(mrc)):
        if words.find(mrc[i]) != -1:
            count = count + 1
    if count > 0:
        count =
    count/float(len(mrc)) return
    count
def neuroticismFunction(words): #calculate number of emotion words from tweets count = 0.0
    for i in range(len(emotion)):
        if words.find(emotion[i]) != -1:
            count = count + 1
```

```
    if count > 0:
        count = count/float(len(emotion))
    return count
def pearson(feature,retweet,followers,mention,hashtag,following): #perason calculation pearson_value = 0;
    x = [feature,retweet,followers]
    y = [mention,hashtag,following]
    pearson_value, _ = pearsonr(x, y) return
    pearson_value
def upload(): #function to upload tweeter profile global
    filename
    filename = filedialog.askdirectory(initialdir=".") pathlabel.config(text=filename)
    text.delete('1.0', END)
    text.insert(END,filename+" loaded\n");
def extractFeatures(): #extract features from tweets
    global openness
    global agreeable
    global
    conscientious
    global extroversion
    global neuroticism
    openness = 0.0
    agreeable = 0.0
    conscientious = 0.0
    extroversion = 0.0
    neuroticism = 0.0
    text.delete('1.0', END)
    for root, dirs, files in os.walk(filename):
      for fdata in files:
        with open(root+"/"+fdata, "r") as file:
            data = json.load(file)
            textdata = data['text'].strip('\n')
            textdata = textdata.replace("\n"," ")
            textdata = re.sub('\W+',' ', textdata)
            retweet = data['retweet_count']
            followers = data['user']['followers_count']
            density = data['user']['listed_count']
            following = data['user']['friends_count']
```

```
            replies = data['user']['favourites_count']
            hashtag = data['user']['statuses_count']
            username = data['user']['screen_name']
            words = textdata.split(" ")
            text.insert(END,"Username : "+username+"\n"); text.insert(END,"Tweet
            Text : "+textdata); text.insert(END,"Retweet Count : "+str(retweet)+"\n")
            text.insert(END,"Following : "+str(following)+"\n")
            text.insert(END,"Followers : "+str(followers)+"\n")
            text.insert(END,"Density : "+str(density)+"\n") text.insert(END,"Hashtag
            : "+str(hashtag)+"\n") text.insert(END,"Tweet Words Length :
            "+str(len(words))+"\n\n")
  def pearsonFunction(): #calculating pearson for each feature value text.delete('1.0',
      END)
      global open_count
      global agree_count
      global ext_count
      global neu_count
      global cons_count
      global openness
      global agreeable
      global
      conscientious
      global extroversion
      global neuroticism
      open_count = 0.0
      agree_count = 0.0
      ext_count = 0.0
      neu_count = 0.0
      cons_count = 0.0
    headers = "Openness,Agreeable,Neuroticism,Extroversion,Conscientious,class\n"
  text.insert(END,"Username\t\tOpenness\tAgreeable\tNeuroticism\tExtroversion\tConscie
ntious\n")
      for root, dirs, files in os.walk(filename): for
        fdata in files:
          with open(root+"/"+fdata, "r") as file:
            data = json.load(file)
            textdata = data['text'].strip('\n')
```

```
            textdata = textdata.replace("\n"," ")
            textdata = re.sub('\W+',' ', textdata)
            retweet = data['retweet_count']
            followers = data['user']['followers_count']
            density = data['user']['listed_count']
            following = data['user']['friends_count']
            replies = data['user']['favourites_count']
            hashtag = data['user']['statuses_count']
            username = data['user']['screen_name']
            words = textdata.split(" ")
            openness = opennessFunction(textdata.lower())#use open swear words in tweets agreeable
            = agreeableFunction(textdata.lower()) #use agreeable words in tweets neuroticism =
            neuroticismFunction(textdata.lower()) #sentiment
            extroversion = following/hashtag              #friendly
            conscientious = followers/hashtag  #hardwork and reliable
            openness = pearson(openness,retweet,hashtag,followers,hashtag,following) agreeable =
            pearson(agreeable,retweet,following,followers,hashtag,following) neuroticism =
            pearson(neuroticism,retweet,density,followers,hashtag,following) extroversion =
            pearson(extroversion,retweet,replies,followers,hashtag,following) conscientious =
Pearson(conscientious,retweet,retweet,followers,hashtag,following) classlabel = 0
            max = 0
            if  openness  >
               max:  max  =
               openness
               classlabel = 1
            if  agreeable  >
               max:  max  =
               agreeable
               classlabel = 2
            if  neuroticism  >
               max:  max  =
               neuroticism
               classlabel = 3
            if  extroversion  >
               max:  max  =
               extroversion
               classlabel = 4
```

```
            if    conscientious    >
                max:      max      =
                conscientious
                classlabel = 5
            values = str(openness)+","+str(agreeable)+","+str(neuroticism)+","+str(extroversion)+","+str(consci
entious)+","+str(classlabel)+"\n"
        headers+=values
            if openness > 0.1:
                open_count = open_count + 1
            if agreeable > 0.1:
                agree_count = agree_count + 1
            if neuroticism > 0.1:
                neu_count = neu_count + 1
            if extroversion > 0.1:
                ext_count = ext_count + 1
            if conscientious > 0.1:
                cons_count = cons_count + 1
            #print('Pearsons correlation: %.3f' % corr)
            text.insert(END,username+"\t\t"+str(round(openness,4))+"\t
"+str(round(agreeable,4))+"\t              "+str(round(neuroticism,4))+"\t
"+str(round(extroversion,4))+"\t                  "+str(round(conscientious,4))+"\n")
        f = open("dataset.txt", "w")
        f.write(headers)
        f.close()
        generateModel()
    def prediction(X_test, cls):  #prediction done here y_pred
        = cls.predict(X_test)
        for i in range(len(X_test)):
          print("X=%s, Predicted=%s" % (X_test[i], y_pred[i])) return
        y_pred
    # Function to calculate accuracy
    def cal_accuracy(y_test, y_pred, details): cm
        = confusion_matrix(y_test, y_pred)
        accuracy = accuracy_score(y_test,y_pred)*100
        text.insert(END,details+"\n\n") text.insert(END,"Accuracy :
        "+str(accuracy)+"\n\n")
        text.insert(END,"Report : "+str(classification_report(y_test, y_pred))+"\n") text.insert(END,"Confusion
```

```
    Matrix : "+str(cm)+"\n\n\n\n\n")
    return
  accuracy def
  runSVM():
    global svm_acc
    global X, Y, X_train, X_test, y_train, y_test
    text.delete('1.0', END)
    cls = svm.SVC(C=2.0,gamma='scale',kernel = 'rbf', random_state = 2)
    cls.fit(X_train, y_train)
    text.insert(END,"Prediction Results\n\n")
    prediction_data = prediction(X_test, cls)
    svm_acc = cal_accuracy(y_test, prediction_data,'SVM Accuracy, Classification Report &
Confusion Matrix')
  def runRandomForest():
    global random_acc
    global X, Y, X_train, X_test, y_train, y_test
    text.delete('1.0', END)
    cls = RandomForestClassifier(n_estimators=1,max_depth=0.9,random_state=None) cls.fit(X_train, y_train)
    text.insert(END,"Prediction Results\n\n")
    prediction_data = prediction(X_test, cls)
    random_acc = cal_accuracy(y_test, prediction_data,'Random Forest Algorithm Accuracy,
Classification Report & Confusion Matrix')
  def
    runNaiveBayes():
    global naive_acc
    global X, Y, X_train, X_test, y_train, y_test
    text.delete('1.0', END)
    cls = GaussianNB()
    cls.fit(X_train, y_train)
    text.insert(END,"Prediction Results\n\n")
    prediction_data = prediction(X_test, cls)
    naive_acc = cal_accuracy(y_test, prediction_data,'Naive Bayes Algorithm Accuracy, Classification Report &
Confusion Matrix')
  def runLRA():
    global
    lra_acc
    global X, Y, X_train, X_test, y_train, y_test
```

```
    text.delete('1.0', END)
    cls = LogisticRegression(C=1e5, solver='lbfgs', multi_class='multinomial') cls.fit(X_train,
    y_train)
    text.insert(END,"Prediction  Results\n\n")
    prediction_data = prediction(X_test, cls)
    lra_acc = cal_accuracy(y_test, prediction_data,'Logistic Regression Algorithm Accuracy,
Classification Report & Confusion Matrix')
  def graph():
    height = [svm_acc,random_acc,naive_acc,lra_acc]
    bars = ('SVM Accuracy', 'Random Forest Accuracy','Naive Bayes Accuracy','Logistic Regression
Accuracy')
    y_pos = np.arange(len(bars))
    plt.bar(y_pos, height)
    plt.xticks(y_pos, bars)
    plt.show()
  def personalityGraph():
    height = [open_count, agree_count,neu_count,ext_count,cons_count]
    bars = ('Openness', 'Agreeable','Neuroticism','Extroversion','Conscientious') y_pos =
    np.arange(len(bars))
    plt.bar(y_pos,
    height)
    plt.xticks(y_pos,
    bars) plt.show()
  font = ('times', 16, 'bold')
  title = Label(main, text='Predicting Personality from Twitter')
  title.config(bg='brown', fg='white')
  title.config(font=font)
  title.config(height=3, width=120)
  title.place(x=0,y=5)
  font1 = ('times', 13, 'bold')
  uploadButton = Button(main, text="Upload Twitter Profile Dataset", command=upload)
  uploadButton.place(x=50,y=100)
  uploadButton.config(font=font1) pathlabel
  = Label(main)
  pathlabel.config(bg='brown', fg='white')
  pathlabel.config(font=font1)
  pathlabel.place(x=360,y=100)
```

```
  extractButton = Button(main, text="Extract LIWC, MRC & Twitter Profile Features",
command=extractFeatures)
  extractButton.place(x=50,y=150)
  extractButton.config(font=font1)
  pearsonButton = Button(main, text="Pearson Correlation Analysis",
command=pearsonFunction)
  pearsonButton.place(x=470,y=150)
  pearsonButton.config(font=font1)
  runsvm = Button(main, text="Run SVM Algorithm", command=runSVM) runsvm.place(x=740,y=150)
  runsvm.config(font=font1)
  runrandomforest = Button(main, text="Run Random Forest Algorithm", command=runRandomForest)
  runrandomforest.place(x=950,y=150)
  runrandomforest.config(font=font1)
  runnb = Button(main, text="Run Naive Bayes Algorithm", command=runNaiveBayes)
  runnb.place(x=50,y=200)
  runnb.config(font=font1)
  lra = Button(main, text="Run Logistic Regression Algorithm", command=runLRA) lra.place(x=330,y=200)
  lra.config(font=font1)
  graph = Button(main, text="Accuracy Graph", command=graph)
  graph.place(x=650,y=200)
  graph.config(font=font1)
  pgraph = Button(main, text="Personality Prediction Graph",
command=personalityGraph)
  pgraph.place(x=820,y=200)
  pgraph.config(font=font1)
  font1 = ('times', 12, 'bold')
  text=Text(main,height=30,width=150)
  scroll=Scrollbar(text)
  text.configure(yscrollcommand=scroll.set)
  text.place(x=10,y=250) text.config(font=font1)
  main.config(bg='brown')
  main.mainloop()
```

CHAPTER 6

EXPERIMENTAL RESULTS

In Fig. 6.1 click on ‘Upload Twitter Profile Dataset’ button to upload dataset. The dataset here consists of a folder. The folder contains multiple text files which contains the twitter user information. The data in the text file consists of username, followers, tweets posted and other related information of the user. So, the tweets folder consists of multiple similar text files of various users. We can also add some more text files into the tweets folder for more better results.

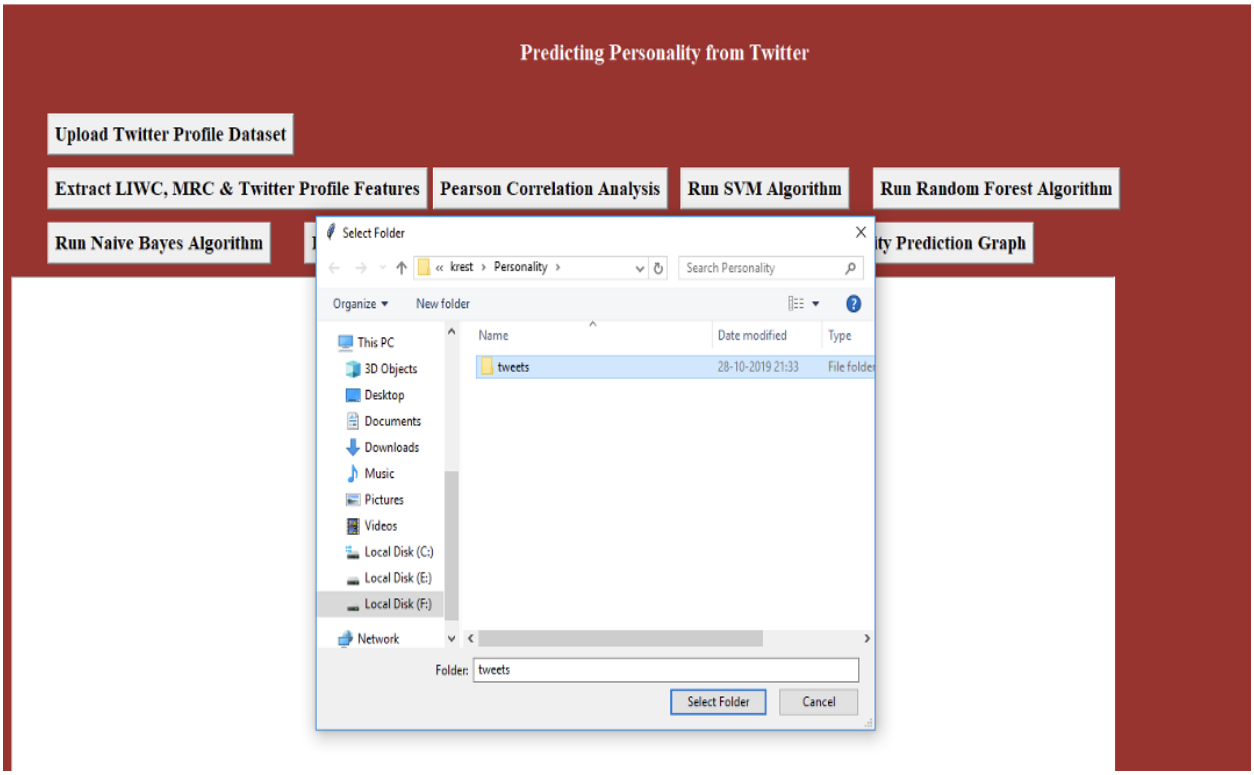


Fig.6.1: Upload Twitter Profile Dataset

In Fig. 6.2 click on 'Extract LIWC Features' button to extract features from tweets and profile. LIWC stands for Linguistic Inquiry and word count. It is a transparent text analysis program that counts words in psychologically meaningful categories. LIWC's design has made it a favorite for psychologists, but it also finds use in marketing, twitter analysis, mental health diagnostics and much more. Psychologists across the world have developed LIWC dictionaries in their native languages. As of writing, languages supported include Arabic, Chinese, Dutch, English, French, German, Italian, Portuguese, Russian, Serbian, Spanish, and Turkish.

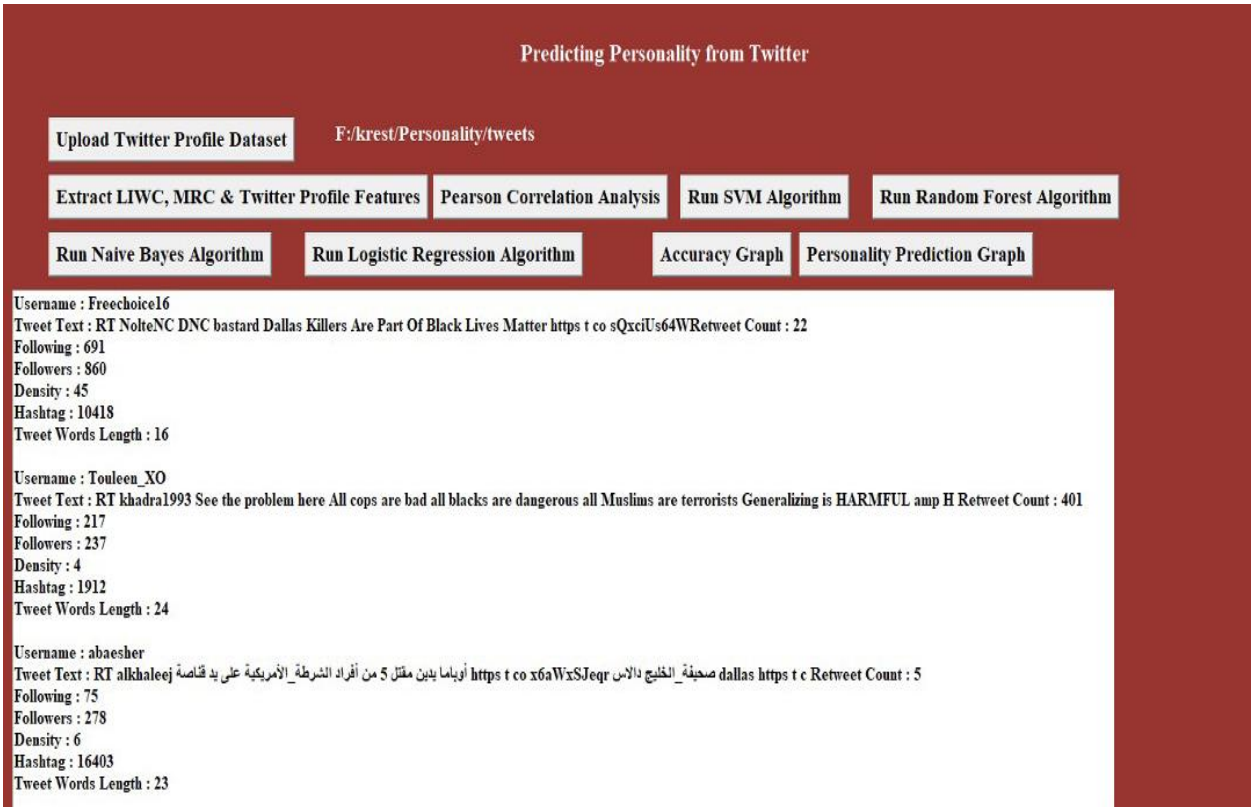
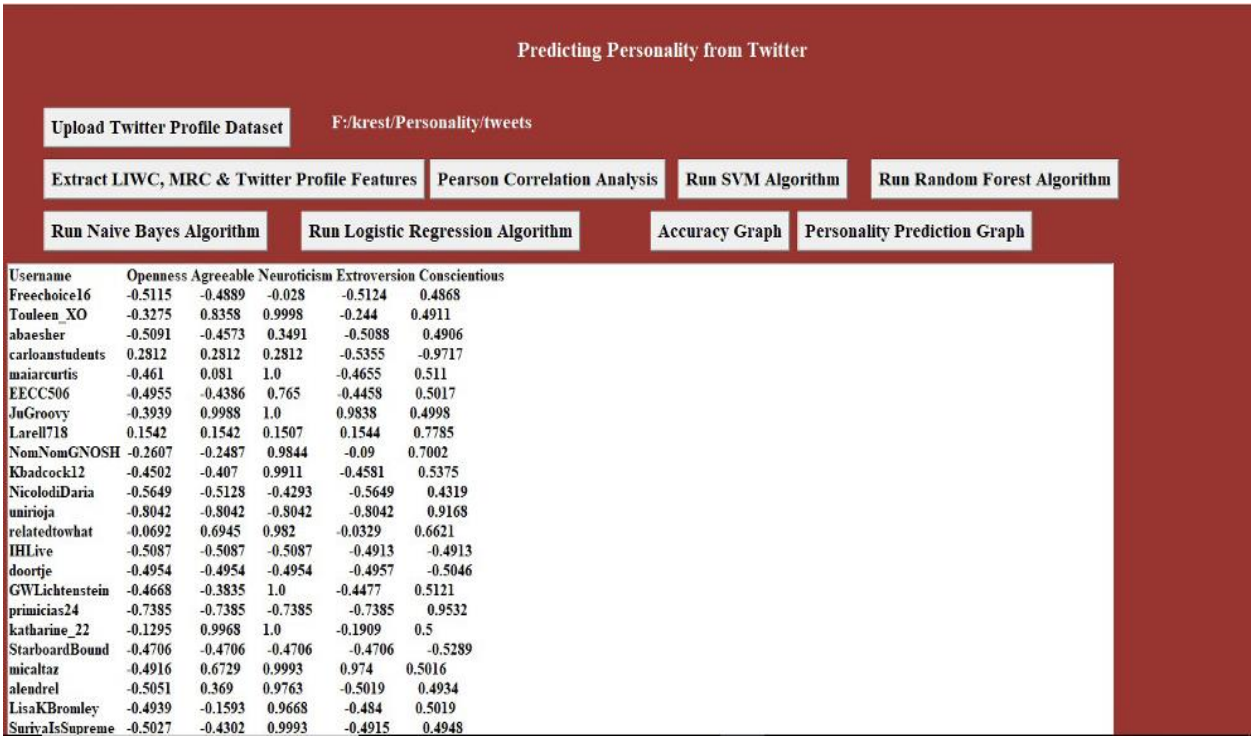


Fig.6.2: Extract LIWC Features

In Fig.6.3, Click on "Pearson Correlation Analysis' to calculate scores for all five features. It represents no of followers, following, density, Hashtag and mainly the tweet first columns contain username and rest of the columns are for features score. These details are retrieved for all the users who had done the tweets. It is the test statistics that measures the statistical relationship, or association, between two continuous variables. It is known as the best method of measuring the

association between variables of interest because it is based on the method of covariance. It is the test statistics that measures the statistical relationship, or association, between two continuous variables. It is known as the best method of measuring the association between variables of interest because it is based on the method of covariance.



Username	Openness	Agreeable	Neuroticism	Extroversion	Conscientious
Freechoice16	-0.5115	-0.4889	-0.028	-0.5124	0.4868
Touleen_XO	-0.3275	0.8358	0.9998	-0.244	0.4911
abaesher	-0.5091	-0.4573	0.3491	-0.5088	0.4906
carloanstudents	0.2812	0.2812	0.2812	-0.5355	-0.9717
maiarcurtis	-0.461	0.081	1.0	-0.4655	0.511
EECC506	-0.4955	-0.4386	0.765	-0.4458	0.5017
JuGroovy	-0.3939	0.9988	1.0	0.9838	0.4998
Larell718	0.1542	0.1542	0.1507	0.1544	0.7785
NomNomGNOSH	-0.2607	-0.2487	0.9844	-0.09	0.7002
Kbadcock12	-0.4502	-0.407	0.9911	-0.4581	0.5375
NicolodiDaria	-0.5649	-0.5128	-0.4293	-0.5649	0.4319
unirioja	-0.8042	-0.8042	-0.8042	-0.8042	0.9168
relatedtowhat	-0.0692	0.6945	0.982	-0.0329	0.6621
IHLive	-0.5087	-0.5087	-0.5087	-0.4913	-0.4913
doortje	-0.4954	-0.4954	-0.4954	-0.4957	-0.5046
GWLichtenstein	-0.4668	-0.3835	1.0	-0.4477	0.5121
primicias24	-0.7385	-0.7385	-0.7385	-0.7385	0.9532
katharine_22	-0.1295	0.9968	1.0	-0.1909	0.5
StarboardBound	-0.4706	-0.4706	-0.4706	-0.4706	-0.5289
micaltaz	-0.4916	0.6729	0.9993	0.974	0.5016
alendrel	-0.5051	0.369	0.9763	-0.5019	0.4934
LisaKBromley	-0.4939	-0.1593	0.9668	-0.484	0.5019
SuriyaIsSupreme	-0.5027	-0.4302	0.9993	-0.4915	0.4948

Fig.6.3: Pearson Correlation Analysis

In Fig.6.4 click on 'Run SVM Algorithm' button to get SVM Accuracy which is 87.5%. Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning. The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

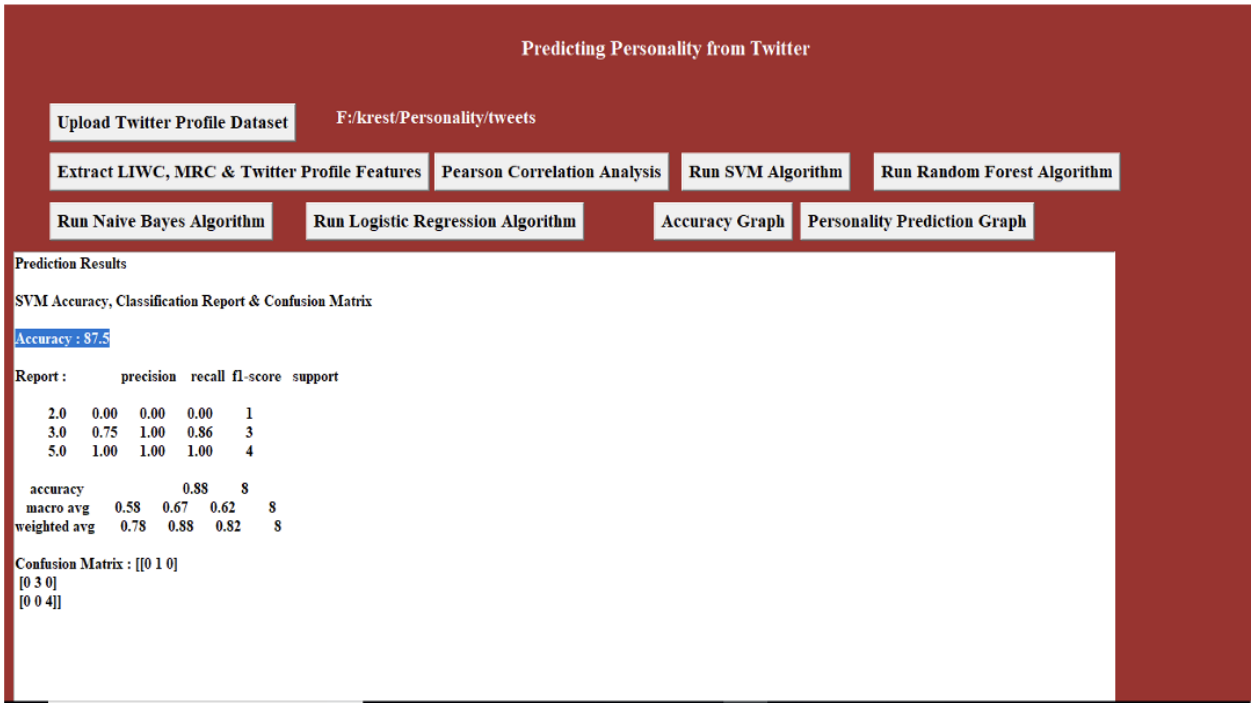


Fig.6.4: Run SVM algorithm.

In Fig.6.5 click on 'Run Random Forest Algorithm' button to get Random Forest Accuracy which is 37.5%. 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.

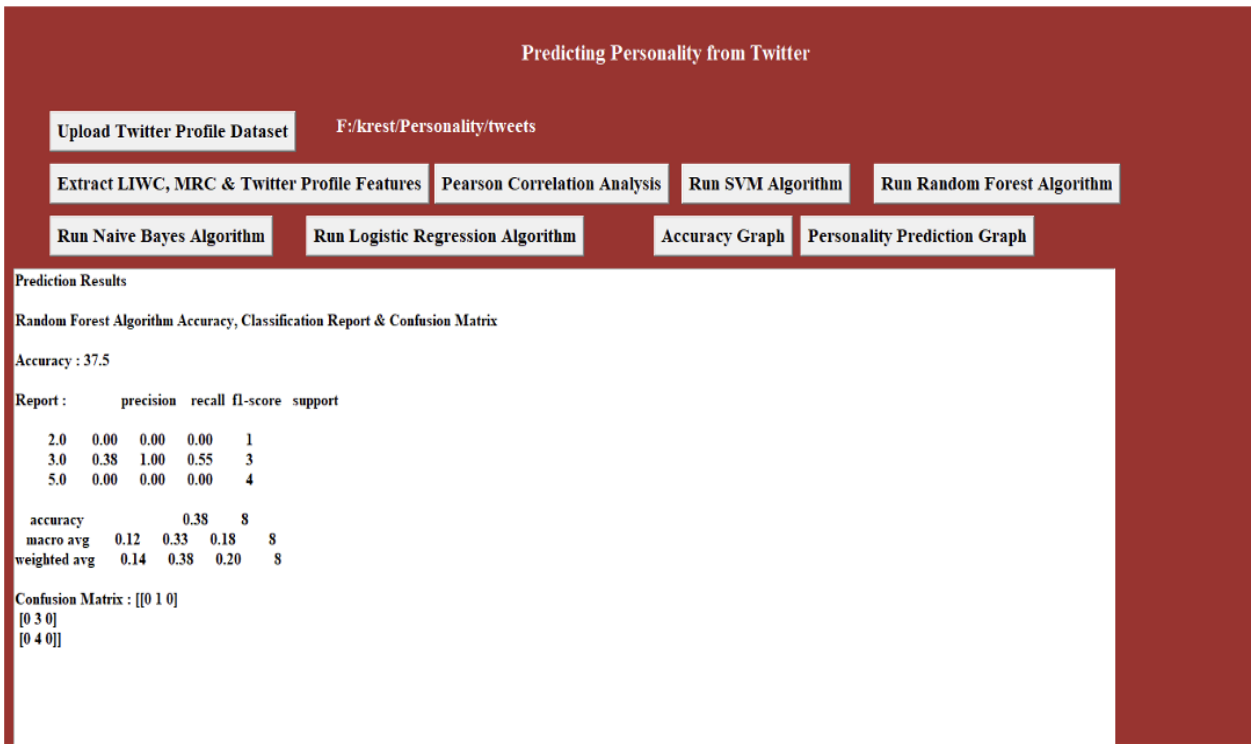


Fig.6.5: Run Random Forest Algorithm

In Fig.6.6 click on 'Run Naïve Bayes Algorithm' button to get Naive Bayes Accuracy which is 87.5%. Naïve Bayes algorithm is a supervised learning algorithm, which is based on Bayes theorem and used for solving classification problems. It is mainly used in text classification that includes a high-dimensional training dataset. Naïve Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions. It is a probabilistic classifier, which means it predicts based on the probability of an object. Some popular examples of Naïve Bayes Algorithm are spam filtration, Sentimental analysis, and classifying articles.

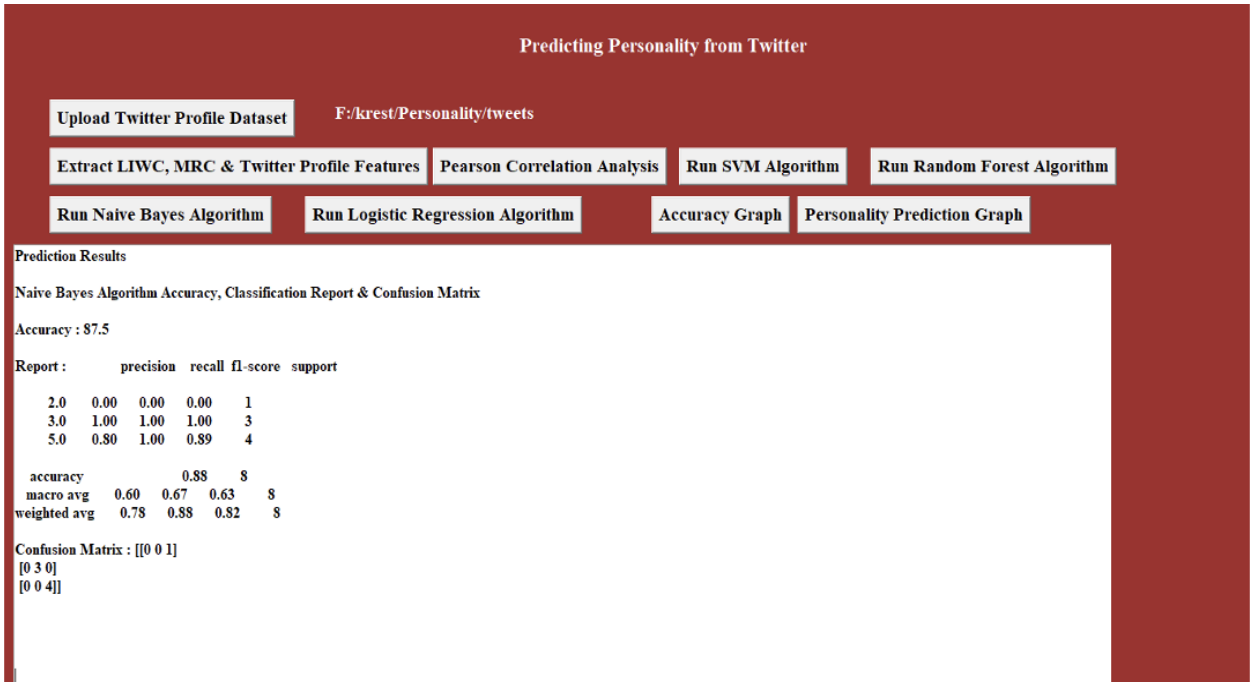


Fig.6.6: Run Naïve Bayes Algorithm

In Fig.6.7 click on 'Run Logistic Regression Algorithm' button to get Logistic Regression Accuracy which is 62.5%. This type of statistical model is often used for classification and predictive analytics. Logistic regression estimates the probability of an event occurring, such as voted or didn't vote, based on a given dataset of independent variables. Since the outcome is a probability, the dependent variable is bounded between 0 and 1. In logistic regression, a logit transformation is applied on the odds—that is, the probability of success divided by the probability of failure. This is also commonly known as the log odds, or the natural logarithm of odds.

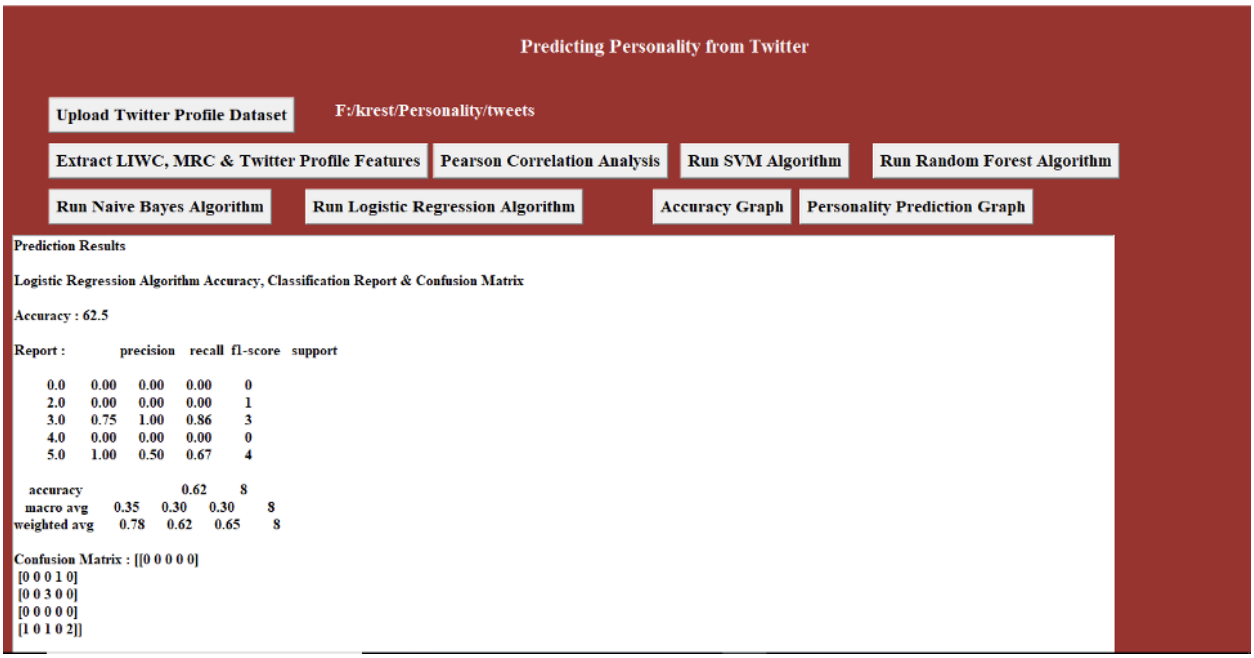


Fig.6.7: Run Logistic Regression Algorithm

In Fig.6.8 click on 'Accuracy Graph' button to get the accuracy of each algorithm. Here x-axis represents algorithm and y- axis represents the accuracy of algorithms, in our case we have four algorithms which are SVM algorithm, Random Forest Algorithm, Naive Bayes Algorithm and Logistic Regression Algorithm. Here, each algorithm has its own methodology and accuracy respectively. In the following graph we can see that both SVM and Naïve Bayes has high and same accuracy which is 87.5. Logistic Regression has accuracy of 62.5 whereas Random Forest has the least accuracy which is 37.5.

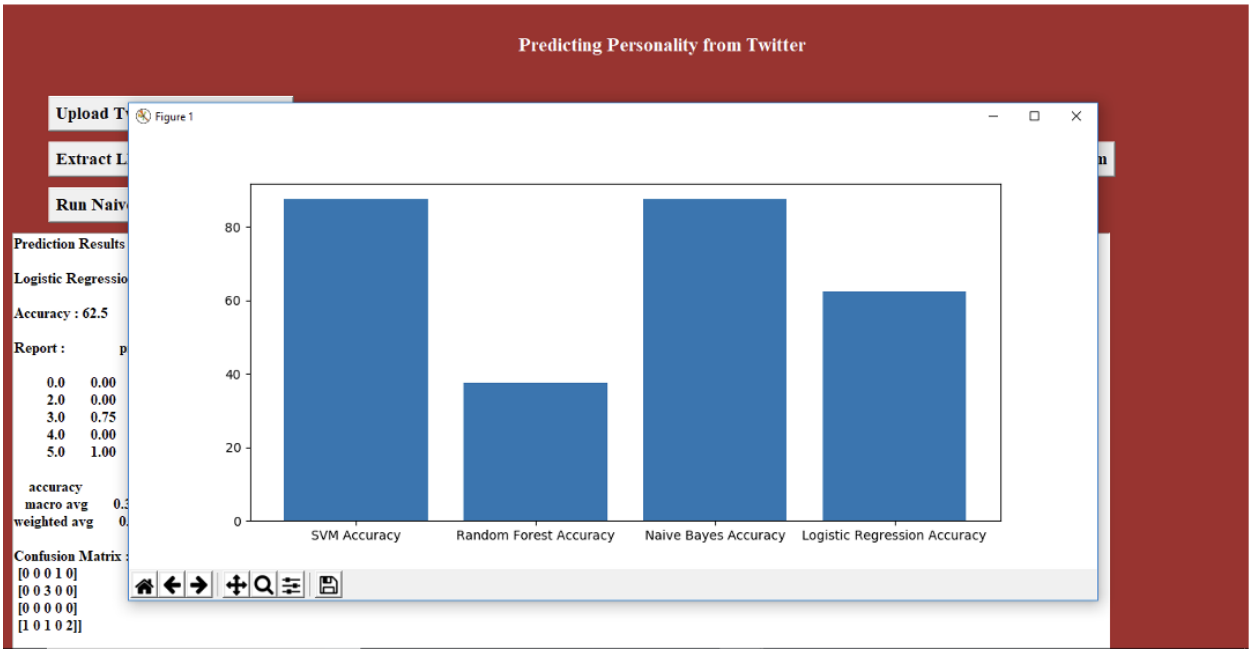


Fig.6.8: Accuracy Graph

In Fig.6.9 click on 'Personality Prediction Graph' to get number of people in each category graph. In this graph x-axis represents feature category name and y-axis represents number of peoples in that category. It shows the bar graph considering the no of users who belongs to the respective personality mentioned. Openness refers to the person who has broad range of interests. Conscientiousness refers to the person who has high levels of thoughtfulness, good impulse control. Extraversion refers to being sociable. Agreeable refers to being kind and affectionate. Neuroticism refers to individuals who have high mood swings, sadness.

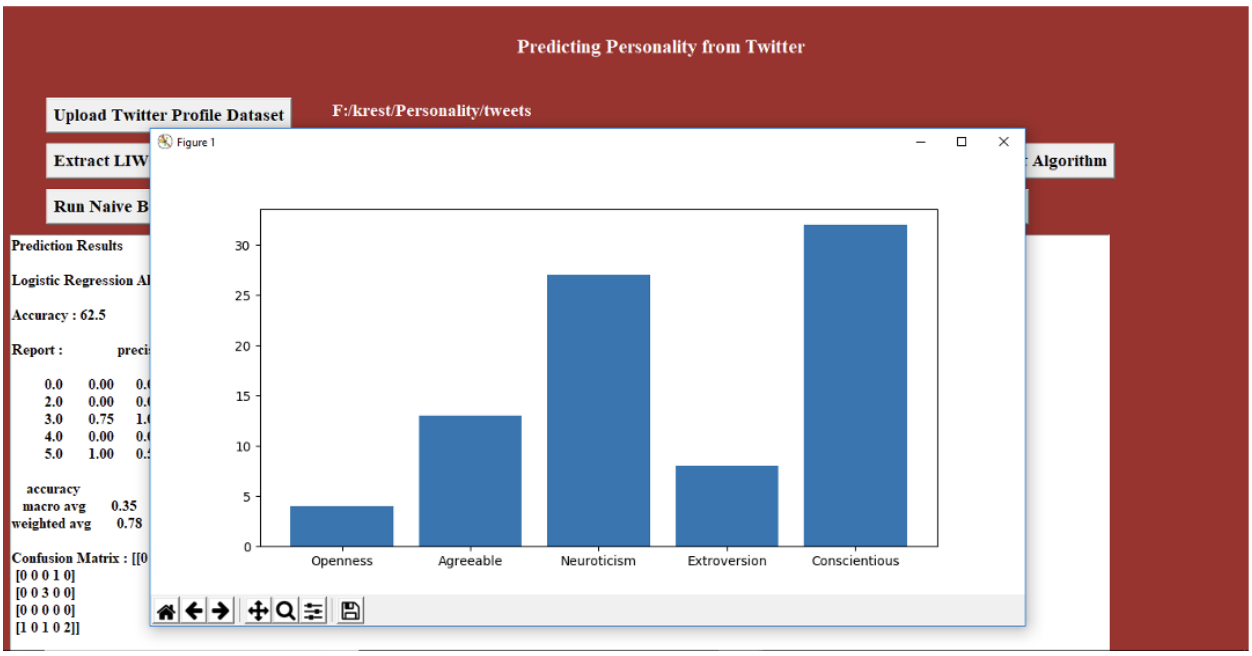


Fig.6.9: Personality Prediction Graph

CHAPTER 7

CONCLUSION AND FUTURE ENHANCEMENT

7.1 CONCLUSION

In this project, we have shown a users' Big Five Personality traits can be predicted from the public information they share of Twitter. Our subject completed a personality test and through the Twitter

API, we collected publicly accessible information from their profiles. With the ability to guess a user's personality traits, many opportunities are opened for personalizing interfaces and information.

7.2 FUTURE ENHANCEMENT

As future works, the number of users can be increased to obtain more consistent models. The current classification might be replaced with more classes for a better representation of the different score levels of each personality traits, that could eventually yield more successful results. This project analyzes only tweets of a user. If other social media environments such as blogs, Instagram, etc. can be integrated for analysis, prediction results are going to be more successful than current ones. These improvements will increase the success of prediction models. Analyzing the personality scores between friends can create an open space for research. We can begin to answer more sophisticated questions about how to present trusted, socially sophisticated questions about how to present trusted, socially relevant, and well-presented information to users.

REFERENCES

[1]. Rahman, S., Chakraborty, P.: Bangla document classification using a deep recurrent neural network with BiLSTM. In: Proceedings of International Conference on Machine Intelligence and Data Science Applications (2020)

[2]. Zulfikar, M.S., Kabir, N., Biswas, A.A., Chakraborty, P., Rahman, M.M.: Predicting students' performance of the private universities of Bangladesh

using machine learning approaches. Int. J. Adv. Comput. Sci. Appl. 11(3), (2020).

[3]. Chakraborty, P., Yousuf, M.A., Rahman, S.: Predicting level of focus of human's attention using machine learning approaches.In: Proceedings of International Conference on Trends in Computational and Cognitive Engineering (2021).

[4]. Ben Verhoeven and Walter Daelemans. 2021. CLiPS Stylometry Investigation (CSI) corpus: A Dutch corpus for the detection of age, gender, personality, sentiment, and deception in text.

[5]. Liu, L., et al. 2020 Analyzing Personality through Social Media Profile Picture Choice.Tenth International AAAI Conference on Web and Social Media.

[6]. 25 Tweets to Know You: A New Model to Predict Personality with Social MediaPierre-Hadrien Arnoux, Anbang Xu, Neil Boyette, Jalal Mahmud, Rama Akkiraju, Vibha Sinha472(2020). Personality classification based on Twitter text using Naïve Bayes, KNN, and SVM.

[7].Plank, B., Hovy, D.: Personality traits on twitter–or–how to get 1,500 personality tests in a week.

[8]. Chaudhary, S., Sing, R., Hasan, S.T., Kaur, I.: A comparative study of classifiers for Myers-Brigg personality prediction model. IRJET (2021).

[9]. Kenter, T., and de Rijke, M. 2019.. In Proceedings of the 24th ACM Conference on Information and Knowledge Management. 1411-1420: ACM. Personality classification based on Twitter text using Naïve Bayes, KNN and SVM.

[10]. Liu, L., et al. 2019. Analyzing Personality through Social Media Profile Picture Choice.Tenth International AAAI Conference on Web and Social Media.

[11]. Ben Verhoeven and Walter Daelemans. 2018. CLiPS Stylometry Investigation (CSI) corpus: A Dutch corpus for the detection of age, gender, personality, sentiment and deception in text.

[12]. Hu, T., et al. 2018. What the Language You Tweet Says About Your Occupation Tenth International AAAI Conference on Web and Social Media.Hansen Andrew Schwartz.

[13]. Azucar D., Marengo D., & Settanni M. (2018). Predicting the big 5 personality traits from digital footprints on social media: A meta–analysis. Personality and Individual Differences.

[14]. Bleidorn W., & Hopwood C. J. (2019). Using machine learning to advance personality assessment and theory. Personality and Social Psychology Review, 23().

[15]. Dahlke J. A., & Wiernik B. M. (2019). Psychmeta: An R package for psychometric meta–analysis. Applied Psychological Measurement

[16]. Haig B. D. (2020). Big data science: A philosophy of science perspective. In Woo S. E., Tay L., & Proctor R. W. (Eds.), Big data in psychological research (pp. 15–33). Washington, DC: American Psychological Association.

[17]. Hinds J., & Joinson A. (2019). Human and computer personality prediction from digital footprints. Current Directions in Psychological Science.

[18]. Alam F, Stepanov EA, Riccardi G (2018) Personality traits recognition on social network-Facebook. In: International conference on weblogs and social media, Cambridge.

[19]. Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J (2020) Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems.

[20]. Bleidorn W., Hopwood C. J., Wright A. G. (2018). Using big data to advance personality theory. Current Opinion in Behavioral Sciences.