PERSONALITY RECOGNITION FOR CANDIDATE SCREENING

Submitted in partial fulfillment of the requirements

of the degree of

Bachelor of Engineering in Computer Engineering

By

PIYUSH PATIL	**18102A0036**
SALONI GOYAL	**18102A0037**
TANYA DWIVEDI	**18102A0028**

Under the Guidance of

Prof. Suvarna Bhat

Department of Computer Engineering

Vidyalankar Institute of Technology
Wadala(E), Mumbai-400437

University of Mumbai

2021-22

CERTIFICATE OF APPROVAL

This is to certify that the project entitled

PERSONALITY RECOGNITION FOR CANDIDATE SCREENING

is a bonafide work of

Piyush Patil	**18102A0036**
Saloni Goyal	**18102A0037**
Tanya Dwivedi	**18102A0028**

submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of

Undergraduate in **Computer Engineering**.

Guide	Head of Department	Principal
Prof. Suvarna Bhat	Dr. Sachin Bojewar	Dr. S.A. Patekar

Project Report Approval for B. E.

This project report entitled **PERSONALITY RECOGNITION FOR CANDIDATE SCREENING** by

1. Piyush Patil	***18102A0036***
2. Saloni Goyal	***18102A0037***
3. Tanya Dwivedi	***18102A0028***

is approved for the degree of ***Bachelor of Engineering in Computer Engineering.***

Examiners

1.----------

2.----------

Date:

Place:

Declaration

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Sr. No.	Name of Student	Roll No.	Signature
1.	Piyush Patil	18102A0036	
2.	Saloni Goyal	18102A0037	
3.	Tanya Dwivedi	18102A0028	

Date:

ACKNOWLEDGMENTS

We are honored to present “Personality Recognition for Candidate Screening” as our B.E Final Year Project. We are using this opportunity to express our profound gratitude to our principal “Dr.S.A.Patekar ” for providing us with all proper facilities and support.

We express our deepest gratitude towards our HOD “Dr.Sachin Bojewar” for his valuable and timely advice during the various phases in our project. We would like to thank our project guide “Prof. Suvarna Bhat” for support, patience, and faith in our capabilities and for giving us flexibility in terms of working and reporting schedules. Finally, we would like to thank everyone who have helped us directly or indirectly in our project.

We would also like to thank our staff members and lab assistant for permitting us to use computer in the lab as when required. We thank our college for providing us with excellent facilities that helped us to complete and present this project.

ABSTRACT

We perform Personality Recognition for the development of a platform for analysis and examination of emotions and behavior of job candidates through personality traits recognition. Personality traits can be considered an important factor for working in a professional environment. Evaluation of such traits at a preliminary stage can prove to be beneficial in a working medium. We have decided to explore textual inputs as well as video input for developing an ensemble model that gathers the information from text responses and integrates them with a provided standard and displays a clear and understandable way of assessing candidates' interest and enthusiasm. Hence, this research suggested an empirical technique to compare Machine Learning models such as Support Vector Machine, Naïve Bayes, Decision Tree, Random Forest, Logistic Regression, Recurrent Neural Networks to discover the optimum personality recognition performance along with Natural Language Processing (NLP), and Affective Computing Methods and find better accuracy of classification algorithms than previous studies by merging Big Five dataset and MBTI dataset. Five human personality traits named as Extraversion (EXT), Neuroticism (NEU), Agreeableness (AGR), Conscientiousness (CON), and Openness (OPN) used for experimental analysis. The result showed that Support Vector Machine and Logistic Regression performed better overall other models across all metrics with an average accuracy score 78.01% for EXT, 79.63% for AGR, 58.91% for NEU, 74.21% for CON, and 81.56% for OPN traits. However, the Naive Bayes algorithm resulted in overall lower performance. For video recognition, we used Convolutional Neural Networks and identified live facial expressions (Happy, sad, Disgust, Surprise, Neutral and Angry) with 65.86% accuracy.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

Emotion recognition through text input is a demanding assignment that surpasses conventional sentiment analysis. Besides detecting basic responses such as neutral, positive, or negative, the objective is to pin down a set of emotions characterized by a higher gradient. Many subtleties are factored in to perform an accurate detection of human emotions where context-dependency is of prime importance.

There are different methods of tackling natural language processing problems, but they are majorly classified as rule-based and learning-based techniques. Rule-Based approaches target more on pattern-identification and are largely based on grammar analysis and sentence structure, whereas learning-based approaches prioritize probabilistic modelling and likelihood maximization. This study prominently focuses on learning-based methods, from conventional classifiers to more advanced neural network architectures. In our study, we have chosen text mining in order to streamline unstructured data and to recognize personality traits based on the ”Big Five” model and MBTI dataset.

Emotion recognition and personality traits classification are two different fields of studies based on different theoretical foundations. However, they use similar learning-based methods. The main aim is to provide a broader assessment of the user's emotions, since it can only be understood through the learning of a person’s characteristics, and analysing personality traits would provide a new key to comprehending human emotion.

Psychology researchers mainly believe that there are five categories, or core factors, also called BIG 5 that determine one’s personality. To mention this model, the acronym OCEAN (for openness, conscientiousness, Extraversion, agreeableness, and neuroticism) is generally used. Due to its popularity and clarity, we have chosen to use this precise model. Openness is a measure that describes an emotion, intellectual curiosity, willingness to try new things, general awareness, reaction to surroundings, etc. Conscientiousness is a scale that elaborates on how people control and regulate their senses and behaviour. Extraversion indicates how people function and flourish in their general surroundings and react around others. Agreeableness reflects individual decisions in general concern for social harmony. Neuroticism measures negative emotions, mental stability, anger, anxiety, etc.

Judgment of personality is key when it comes to the prediction of characteristics of an individual which differentiate them from other individuals. Hence, getting a deeper knowledge of a person's personality types is extremely valuable for indicating their physical and mental wellbeing. Myers-Briggs Type Indicator (MBTI) is an indicator of an individual's perception of the world and decision-making process. It classifies an individual's personality based on Extraversion(E) or Introversion(I), Sensing(S) or Intuition(N), Thinking(T) or Feeling(F), and Judging(J) or Perceiving(P). Information from these personality tests helps companies better comprehend the extent of their employees' strengths, weaknesses, and their capability to perceive and process information. As mentioned in [1] , Data obtained from MBTI assessments helps businesses build stronger organizations as it guides them to assemble efficient teams, facilitate communication between the team and the manager, motivate employees, solve conflicts and develop leadership.

1.2 OBJECTIVE

The main goal is to build a tool capable of recognizing the personality traits of a candidate, given a text and video containing his answers to pre-established personal questions with the support of statistical learning methods. Getting a general sense of a candidate's educational background, career history, and interest in the company should help you weed out unqualified candidates. Primary checks include looking at qualifications such as work experience, academic background, skills, knowledge base, traits and behaviours that indicate a person's behaviour, as well as competency. With traditional recruiting methods, recruiters struggle to evaluate candidates accurately. The purpose of our model is to determine if the candidate has the basic skills, aptitude and enthusiasm to meet the requirements of the job. In candidate selection, various tools are used to assess a candidate's suitability for the job, including interviews, skills tests, psychometric tests, group discussions, and reference checks. It should be noted that the purpose of our model is to narrow down the most qualified candidates who should be treated to a traditional interview and remaining steps of candidate selection.

CHAPTER 2

LITERATURE SURVEY

2.1 SURVEY OF EXISTING SYSTEMS

Sr. No	Authors & Title of Paper	Features
1.	Md Abdur Rahman, Asif Al Faisal, Tayeba Khanam, Mahfida Amjad, and Md Saeed Siddik. Personality detection from text using convolutional neural network [2]	It presents an empirical method to determine the best way of personality detection by making comparisons between several activation functions named sigmoid, tanh, and leaky ReLU. Python 2.7 is used for coding of the model, along with Google's word2vec embeddings and Mairesse features. The experimental analysis uses five personality traits, namely EXT, NEU, AGR, CON, and OPN. Calculating the average F1-score of the functions sigmoid, tanh, and leaky ReLU gives 33.11%, 47.25%, and 49.07% respectively.
2.	Kasula Chaithanya Pramodh and Y Vijayalata. Automatic personality recognition of authors using big five factor model. [3]	Datasets stream-of-consciousness and MyPersonality are used. MyPersonality dataset has a mixture of 250 users who are updating around 10,000 Facebook Status Updates. Natural Language Toolkit has been used for their model and their F1-scores are given as 0.665, 0.632, 0.625, 0.624 and 0.637 for the traits OPN, CON, EXT, AGR and NEU respectively. .
3.	Raad Bin Tareaf, Seyed Ali Alhosseini, Philipp Berger, Patrick Hennig, and Christoph Meinel. Towards automatic personality prediction using facebook likes metadata. [4]	proposes a model that can properly distinguish between a religious individual and a non-believer across 83% of circumstances, between individuals of Asian and European decent in 87% of situations, and between emotionally stable and emotionally unstable individual across 81% of circumstances. The presented analysis is based on a MyPersonality dataset containing over 738,000 users who have granted their Facebook activities, data from other social networks, egocentric networks, demographic

		characteristics. Regression trees, linear regression, k-nearest neighbors, and neural networks were some of the different algorithms used.
4.	Rapha el Lederman Anatoli de Bradk e, Ma el Fabien and St ephane Reynal. Multimodal emotion recognition. [5]	This paper explored different art models including text, audio, and video in multimodal emotion recognition. Under text input, the dataset was a study by Pennebaker and King consisting of 2,468 essays. Bag-of-words and Word2Vec embedding techniques were used for preprocessing. Personality scores were assessed by the Big Five Inventory. Different Classification models were tested against each other such as Multinomial Naive Bayes, Support Vector Machines, Recurrent Neural Networks, and LSTM.
5.	Navonil Majumder, Soujanya Poria, Alexander Gelbukh, and Erik Cambria. Deep learning-based document modeling for personality detection from text. [6]	Deep Convolutional Neural Network or DCNN were used by the researchers to classify between personality traits on the basis of the Big Five model. The dataset used included the stream-of-consciousness essay dataset by James Pennebaker and Laura King. The model also utilizes Google's word2vec embeddings along with Mairesse features.
6.	Abir Abyaa, Mohammed Khalidi Idrissi, and Samir Bennani. Predicting the learner's personality from educational data using supervised learning. [7]	Supervised learning algorithms have been used for the classification of personality according the Big Five model. The algorithms used were SVM (Support Vector Machines), RF (Random Forest), kNN (k-Nearest Neighbors), NB (Naïve Bayes), J48, LR (logistic regression), and bagging. The dataset used was collected from 48 students over a 10-week term. Data included was of four types: educational data, pre and post-survey responses, EMA, and sensor data. Positives results of 62.5%, 50%, 62.5% were achieved by Extraversion SVM, Random Forest, and logistic regression respectively.

		However only naïve Bayes and bagging yielded promising results: 57.14% for Openness to experience. SVM and bagging gave uplifting outcome of 50% for agreeableness. In case of neuroticism, only SVM gave acceptable results of 57.14%.
7.	Waiel Tinwala and Shristi Rauniyar. Big five personality detection using deep convolutional neural networks. [8]	This paper proposed a model for personality detection using deep convolutional neural networks. The dataset used is essays collated by James Pennebaker and Laura King which are based on the Big Five Model also known as the five-factor model or the OCEAN model. The document-level feature extraction was carried out using Google's word2vec embeddings and Mairesse features. The data processed is then fed to a deep convolutional network (DCN) and a binary classifier is used for classifying the presence/absence of the personality trait. Function of tanh is best used for traits Extraversion, Neuroticism, and Agreeableness giving F1- scores of 61.2%, 66.33% and 62.67% respectively. Sigmoid is best used for Openness and Conscientiousness providing F1-scores of 69.71% and 67.46% respectively.
8.	KN Pavan Kumar and Marina L Gavrilova. Personality traits classification on twitter. [9]	study proposes a personality traits classification system, which can incorporate the language-based features, that are formulated upon count-based vectorization (TF-IDF) and the technique of GloVe word embedding, with a collection of prediction system that consists of gradient-boosted decision trees and a Support Vector Machine classifier. The given combination helps to reliably estimate the certain personality traits using the most recent 50 tweets from the given user's profile. The proposed system's performance gets validated on a giant, publicly available

		Twitter MBTI Personality Dataset and is compared favorably with other different state-of-the-art techniques.
9.	Mehendale, N. Facial emotion recognition using convolutional neural networks (FERC). [21]	In this paper, we propose a novel technique called facial emotion recognition using convolutional neural networks (FERC). The FERC is based on two-part convolutional neural network (CNN): The first-part removes the background from the picture, and the second part concentrates on the facial feature vector extraction. In FERC model, expressional vector (EV) is used to find the five different types of regular facial expression. Supervisory data were obtained from the stored database of 10,000 images (154 persons). It was possible to correctly highlight the emotion with 96% accuracy, using a EV of length 24 values.
10.	Zhang, Hongli & Jolfaei, Alireza & Alazab, Mamoun. (2019). A Face Emotion Recognition Method Using Convolutional Neural Network and Image Edge Computing. [22]	In this paper, they avoided the complex process of explicit feature extraction in traditional facial expression recognition by using a face expression recognition method based on a convolutional neural network (CNN) and an image edge detection is proposed. Firstly, the facial expression image is normalized, and the edge of each layer of the image is extracted in the convolution process. The extracted edge information is superimposed on each feature image to preserve the edge structure information of the texture image. The experimental results show that the proposed algorithm can achieve an average recognition rate of 88.56% with fewer iterations, and the training speed on the training set is about 1.5 times faster than that on the contrast algorithm.

CHAPTER 3

SYSTEM DESIGN

3.1 PROJECT PLANNING AND MANAGEMENT

3.1. Project Development Lifecycle:

- Initiation:
  - We got the idea of this project in our 4th semester and we started our studies and information gathering on this project from then till 2nd week of 7th semester.
- System Concept Development:
  - In Our Application, we provide companies and institutions a platform to provide an additional candidate screening feature depending on the candidate's personality.
- Planning:
  - By group discussion and initial planning, we understood most of the requirements of our projects and then we started to plan accordingly.
- Requirement Analysis:
  - Here we gather all the requirements of every user which is required to develop the system. Also, we gathered software requirements we needed to develop the intended system. Then we create a final requirement document list.
- Design:
  - We transform the detail requirement into a complete system, design documents and focus on how to deliver required functionality. We design final UML diagrams of each for our system.
- Development:
  - We prepare test cases, procedure, testing, coding, compiling, refining a program then performing test readiness review.
- Integration and Test:
  - We demonstrate that the developed system confirms the requirement as specified in the functional requirement document conducted by quality assurance, end user, automated testing and analysis.

- Implementation:
  - We implemented the system in production environment, resolution problem identity in integration and test phases.
- Operation and Maintenance:
  - We describe the task to operate and maintain the information system in product.
- Disposition:
  - Here we describe how the system works and create a user manual for the end user. The importance is given to proper implementation of data.

3.2 DATASET

For personality recognition using text. the dataset that we have chosen is from Kaggle [10] that is based on the Myers Briggs Type Indicator [11] and the Neo Personality Inventory Costa Jr and McCrae [12] also called the Big 5. The Myers Briggs Type Indicator (or MBTI for short) is a personality type system that divides personality into 16 distinct types across 4 major divisions that is Introversion (I) – Extraversion (E), Intuition (N) – Sensing (S), Thinking (T) – Feeling (F), Judging (J) – Perceiving (P). Depending on these attributes' personalities can be coded in a four-letter term for example - ISTJ, ISTP, ESFJ. This dataset contains over 8600 rows of data, on each row is a person's MBTI type (4 letter MBTI code/type) and a written entry of each of the last 50 things they have posted (Each entry separated by 3 pipe characters). The second dataset used is the MyPersonality Project dataset [13] which consists of 2400 stream-of-consciousness texts labeled with a personality from Pennebaker and King [14] and used by Mairesse [15] that examines five personality traits that are openness, conscientiousness, extraversion, agreeableness, and neuroticism. Major traits of both the personality indicators exhibit a correlation among them explained in Furnham [16] and hence can be used together on the dataset for increased accuracy. Judging-Perceiving dimensions exhibit similarity with Conscientiousness; Thinking-Feeling dimensions for the MBTI is equivalent to Agreeableness; Introversion-Extraversion is analogous with Extraversion and Openness with the Sensing-Intuitive dimension. Only Neuroticism appeared to be correlated with a variety of MBTI dimensions and somewhat inconsistent across all of them. The dimension remarkably missing from the MBTI is Neuroticism.

For the facial expression recognition, the data consists of 48x48 pixel grayscale images of faces. The faces have been automatically registered so that the face is more or less centred and occupies about the same amount of space in each image. The task is to categorize each face based on the emotion shown in the facial expression into one of seven categories (0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, 6=Neutral). The training set consists of 28,709 examples and the public test set consists of 3,589 examples.

3.3 HARDWARE AND SOFTWARE REQUIREMENTS

Hardware used:

1. Laptop : 8 GB RAM, 1 GB free memory, i7 Core Processor, Windows 10

Software used:

1. Google colab : To program the Machine Learning Code
2. Flask : Used for development of the Webpage
3. Visual Studio Code : Used for Python Programming

Dependencies:

1. Numpy : The name “Numpy” stands for “Numerical Python”. It is the commonly used library. It is a popular machine learning library that supports large matrices and multi- dimensional data. It consists of in-built mathematical functions for easy computations. Even libraries like TensorFlow use Numpy internally to perform several operations on tensors.
2. Pandas : Pandas are an important library for data scientists. It is an open-source machine learning library that provides flexible high-level data structures and a variety of analysis tools. It eases data analysis, data manipulation, and cleaning of data. Pandas support operations like Sorting, Re-indexing, Iteration, Concatenation, Conversion of data, Visualizations, Aggregations, etc.
3. Matplotlib : This library is responsible for plotting numerical data. And that’s why it is used in data analysis. It is also an open-source library and plots high-defined figures
4. Pickle : Python pickle module is used for serializing and de-serializing a Python object structure. Any object in Python can be pickled so that it can be saved on disk.
5. Seaborn : Seaborn is a library for making statistical graphics in Python. It builds on top of matplotlib and integrates closely with pandas’ data structures. Seaborn helps you explore and understand your data.
6. Scikit-learn : It is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistence interface in Python.

3.4 METHODOLOGY

The following is the proposed system's operation procedure: (i) Merging dataset and re-sampling (ii) Pre-processing and Text-vectorization (iii) Personality Classification based on text data using Machine learning models (iv) Comparing the efficiency of the models with other classifiers (v) Various evaluation metrics

The publicly available dataset of Big five personality traits is acquired from the website. The dataset is of 2467 rows, where every row represents individual user. Each user's essays are included along with that user's Big 5 personality type (e.g., openness, agreeableness). We merged this Big 5 dataset with the MBTI dataset to increase the efficiency of the Machine learning model. MBTI dataset is also publicly available on Kaggle. This dataset consists of 8675 rows, where the user's past social media posts have been given with the MBTI personality type (e.g., ENTP, ISJF). We merged the data based on the correlation derived from the research paper of Adrian Furnham 1996 Furnham [1996]. As a result, a labeled dataset comprising a total of 11142 records.

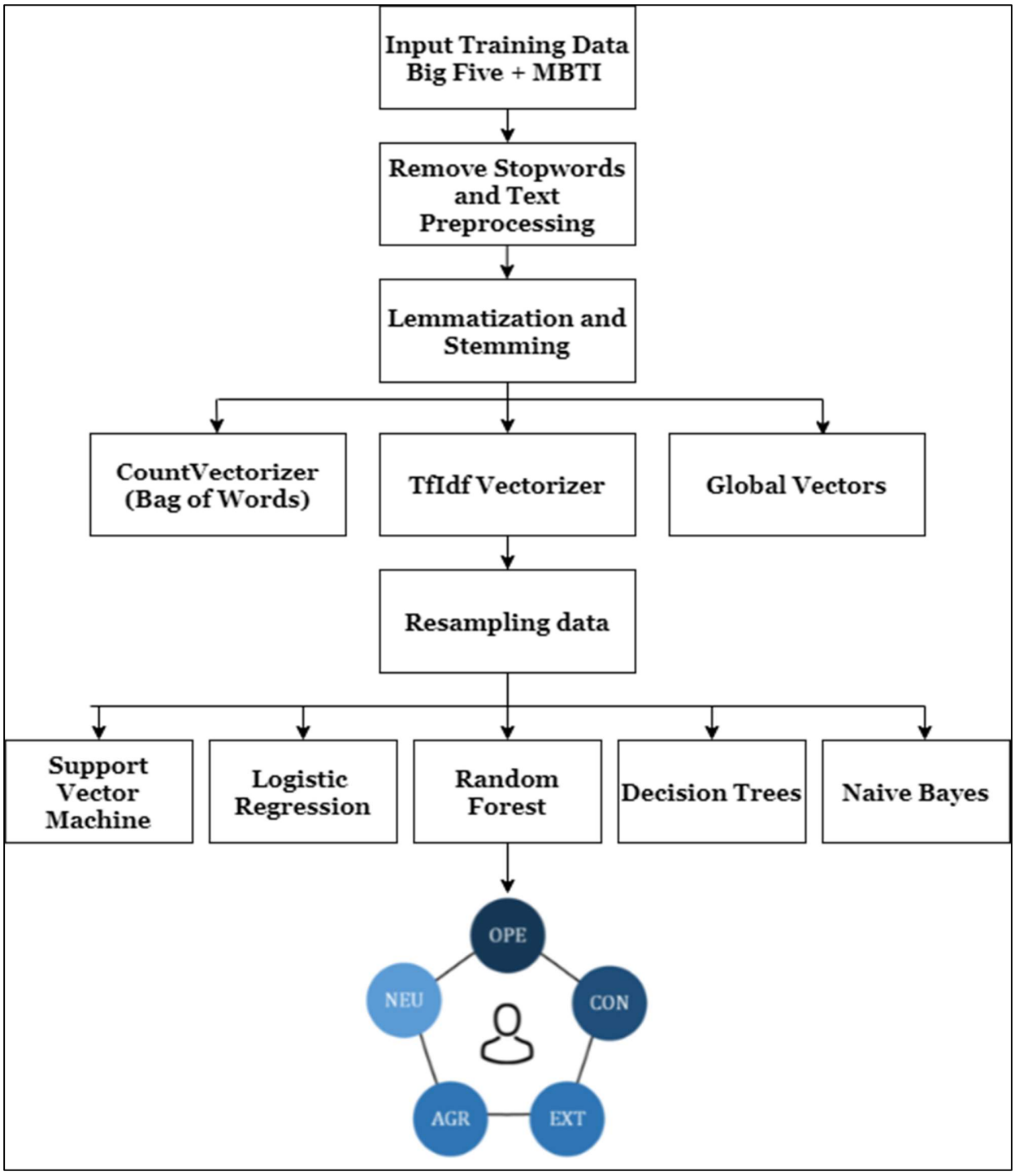


Figure 1: Proposed text based personality recognition framework

3.4.1 Text Preprocessing

Removal of Stop words

Stop words, such as articles, prepositions, pronouns, conjunctions, and others, are the most common words in any language and do not provide any significant information to the text. “the”, “a”, “an”, “so”, “what” are some of the stopwords in English. Elimination of these stopwords is the first step in text preprocessing. Stop words are present in abundance and hence can hamper the results while training the dataset. By keeping unwanted words out of our corpus, we can focus more on the high-level information essential for training our dataset.

Handling imbalanced dataset

As shown in Figure 2, the dataset is unevenly distributed in all 5 types, mentioned as follows: S/N Trait: S=7466 and N=1194, T/F Trait: T=4685 and F=3975, I/E Trait: I=6664 and E=1996, J/P Trait: J=5231 and P=3249. The outcome of any algorithm applied on a skewed or unbalanced classified dataset always favors the large and smaller classes that are passed over for prediction. So, we used resampling methods such as Oversampling and Undersampling which basically make the dataset equally distributed.

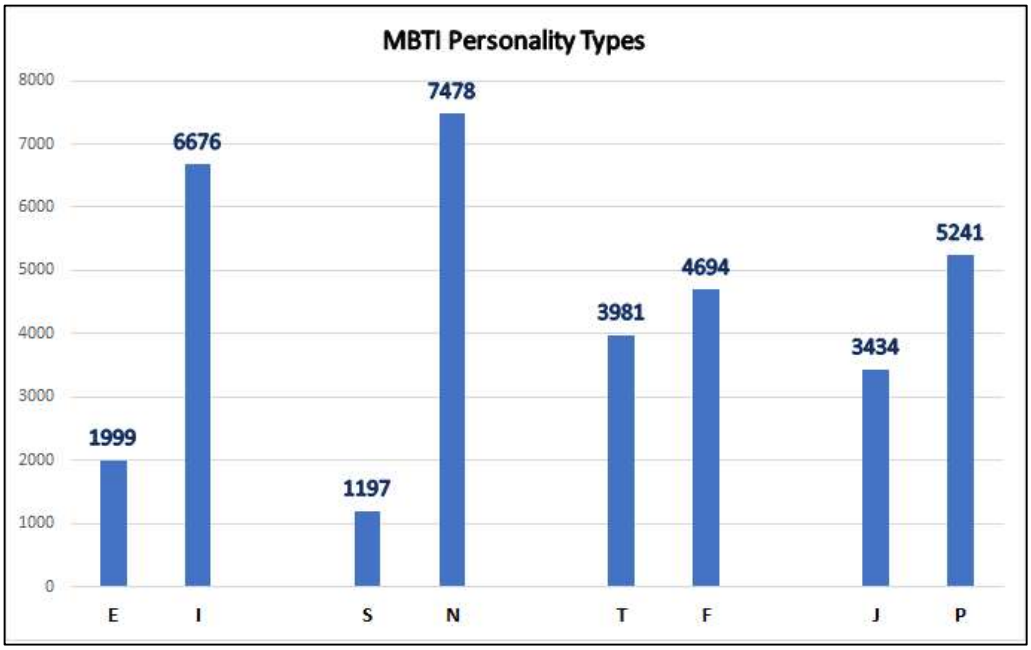


Figure 2: Imbalanced Dataset 1

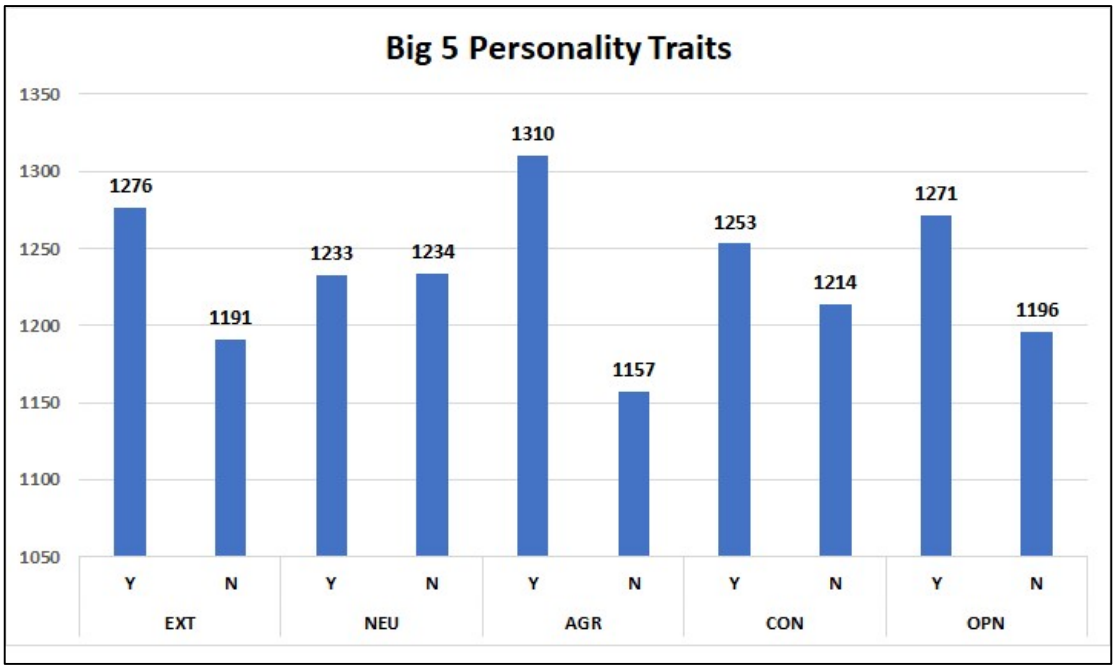


Figure 3: Imbalanced Dataset 2

- Undersampling

The minority classes can present information that is pertinent to the outcome but the biased distribution of the classes can lead to ineffective results. Undersampling is the removal of random samples from the majority class. Under-sampling minimizes the size of the data, requiring less time for learning.

The drawback is that deleting majority classes may result in the majority class losing useful information.

- Oversampling

Another strategy used to add minority cases to the dataset in order to achieve a balance is over-sampling, which involves duplicating the current minority samples. This increases the data size but provides impetus to the minority classes.

3.4.2 Text Vectorization

CountVectorizer

It's a technique for converting a given text into a vector or matrix based on the frequency (count) of each word in the corpus. It considers how many times a word appears in a text (multiplicity), ignoring grammatical subtleties and even word order. Count Vectorizer generates a matrix with a column for each unique word and a row for each text sample from the document. The count of the words in that particular text sample is the value of each cell. The frequency of a particular word in a text is proportional to the importance of the term in the text.

Term frequency–inverse document frequency (TFIDF)

TFIDF technique is of two sections that are term frequency and inverse of document frequency. The term frequency is measured as a percentage of the total number of words in a single document. Term frequency does not consider the importance of words only the frequency. Some words can be most frequently present but are of little significance and hence can alter the results. Each word is given a weight based on its frequency in a corpus using inverse document frequency. This measure is generated by dividing the total number of documents by the number of documents that include a specific word.

$$TF(t) = \frac{Number\ of\ times\ term\ t\ appears\ in\ a\ document}{Total\ number\ of\ terms\ in\ the\ document}$$

$$IDF(t) = log_e\left(\frac{Total\ number\ of\ documents}{Number\ of\ documents\ with\ term\ t\ in\ it}\right)$$

3.4.3 Modelling

Support Vector Machines

SVMs are popular linear classifiers that essentially splits the data plotted on a multidimensional graph with a hyperplane. Ideally, the features we use in this classifier must correlate linearly with all the classes in a way that the classifier linearly segments the space to include all records with the same trait labels. For each label, we train a single-label One Vs All classifier that employs the subset of features that are most linearly related to the classes they are assigned to. Finding the right hyper-parameter can be difficult, but it can be done by experimenting with various combinations and observing which ones work best. The method involves the creation of a grid of hyperparameters and trying all of its various combinations, and hence is called Grid search. As mentioned in GeeksforGeeks [18], GridSearchCV is a built-in feature that uses a dictionary to specify the parameters that can be used to train a model. The parameter grid is described as a dictionary, with the keys being the parameters and the values being the settings to be tested.

Decision Tree

By constructing a decision tree, the decision tree classifier builds the classification model. Each node in the tree represents a test on an attribute, and each branch descending from that node represents one of the attribute's possible values as given in KDnuggets [19]. By learning simple decision rules inferred from past data, the purpose of employing a Decision Tree is to develop a training model that can be used to predict the class or value of the target variable (training data). We start from the root of the tree when using Decision Trees to forecast a class label for a record. The values of the root attribute and the record's attribute are compared. We follow the branch that corresponds to that value and jump to the next node based on the comparison.

Naïve Bayes

Naïve Bayes works fairly well for text-based classification problems. The classifier makes predictions based on the learning of distribution of posterior probability. We train the Na¨ıve Bayes model with a subset of features listed in the previous section. It turns out that the Na¨ıve Bayes classifier yields the best result for one of the personality traits. This algorithm has a simple and intuitive design, and is a good benchmark for classification purposes.

$$P(A/B) = \frac{P(B/A) * P(A)}{P(B)}$$

Logistic Regression

Logistic regression is one of the classification methods that has its roots in statistics but is now widely utilized in machine learning. It's a statistical technique for assessing a data collection in which one or more independent variables influence the outcome. Logistic regression is used to create the best-fitting model and characterize the relationship of the dependent and independent variables. In logistic regression [20], the Sigmoid function is used in logistic regression to map predicted values to probabilities. This method converts any real value to a number between 0 and 1. At each point, this function does have a non-negative derivative and only one inflection point.

Random Forest

Random forest is a decision tree-based model, it predicts the result by averaging several results from multiple pre-built decision trees. Each decision tree has different structures learned from the training input. In our project, multiple features are in use, but we have do not know how to weigh the features to get the best results for the classification. So, we experimented with the Random Forest model with the training set containing the features we choose to use and use the rest of the labeled data for validation.

3.4.4 Video Input

The task is to categorize each face based on the emotion shown in the facial expression into one of seven categories (0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, 6=Neutral). The training set consists of 28,709 examples and the public test set consists of 3,589 examples.

1. Take the video stream and convert it to frames. Use Haar cascades to identify and draw bounding box around faces.
2. Pass the Region of Interest (Face patch) to the pretrained model. Use Forward feed to detect the expression.
3. Take the output of the model and add the label to the image and return the frame as a 'jpeg' file.
4. Use flask to publish the resulting frames onto the website. (HTML file has the template for video output) The output is deployed on 'http://localhost:5000/' web address.

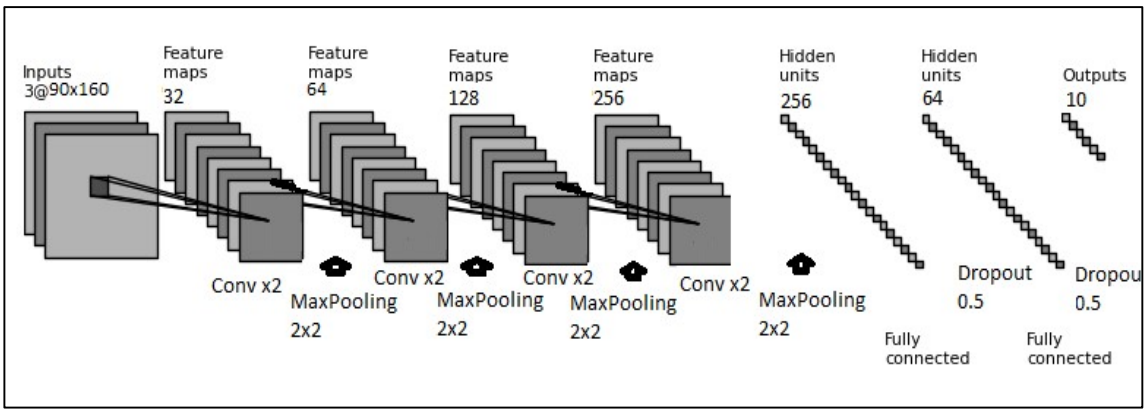


Figure 4 : CNN Model Architecture

Haar Cascades

Here we will work with face detection. Initially, the algorithm needs a lot of positive images (images of faces) and negative images (images without faces) to train the classifier. Then we

need to extract features from it. For this, Haar features shown in the below image are used. They are just like our convolutional kernel. Each feature is a single value obtained by subtracting sum of pixels under the white rectangle from sum of pixels under the black rectangle. Now, all possible sizes and locations of each kernel are used to calculate lots of features. Then we apply each feature on all the training images. For each feature, it finds the best threshold which will classify the faces to positive and negative. Obviously, there will be errors or misclassifications. We select the features with minimum error rate, which means they are the features that most accurately classify the face and non-face images.

Convolutional Neural Networks

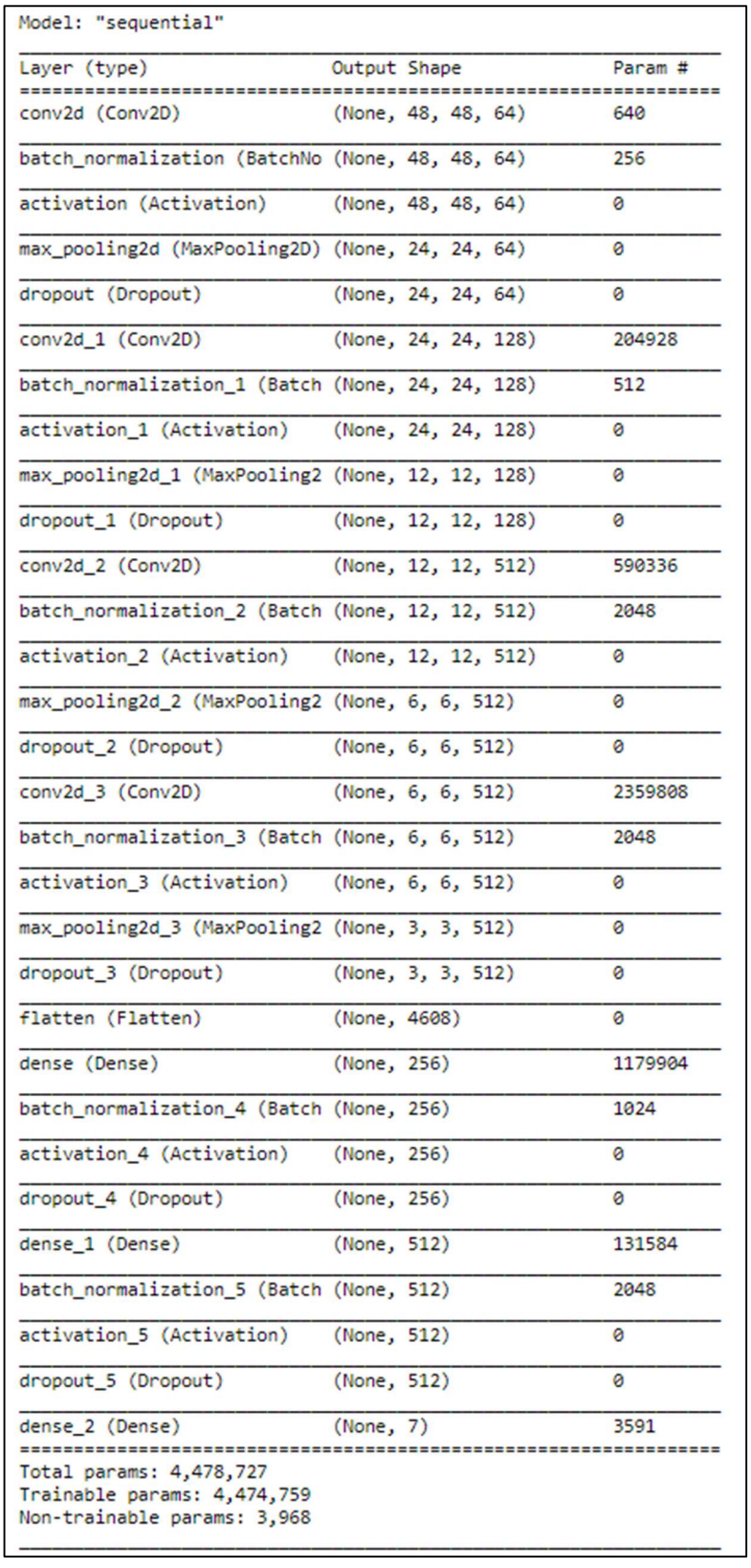
A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, ConvNets have the ability to learn these filters/characteristics.

Pooling layer

Similar to the Convolutional Layer, the Pooling layer is responsible for reducing the spatial size of the Convolved Feature. This is to decrease the computational power required to process the data through dimensionality reduction. Furthermore, it is useful for extracting dominant features which are rotational and positional invariant, thus maintaining the process of effectively training of the model.

Classification — Fully Connected Layer (FC Layer)

Adding a Fully-Connected layer is a (usually) cheap way of learning non-linear combinations of the high-level features as represented by the output of the convolutional layer. The Fully-Connected layer is learning a possibly non-linear function in that space.



```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 48, 48, 64)        640
_________________________________________________________________
batch_normalization (BatchNo (None, 48, 48, 64)        256
_________________________________________________________________
activation (Activation)      (None, 48, 48, 64)        0
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 24, 24, 64)        0
_________________________________________________________________
dropout (Dropout)            (None, 24, 24, 64)        0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 24, 24, 128)       204928
_________________________________________________________________
batch_normalization_1 (Batch (None, 24, 24, 128)       512
_________________________________________________________________
activation_1 (Activation)    (None, 24, 24, 128)       0
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 12, 12, 128)       0
_________________________________________________________________
dropout_1 (Dropout)          (None, 12, 12, 128)       0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 12, 12, 512)       590336
_________________________________________________________________
batch_normalization_2 (Batch (None, 12, 12, 512)       2048
_________________________________________________________________
activation_2 (Activation)    (None, 12, 12, 512)       0
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 6, 6, 512)         0
_________________________________________________________________
dropout_2 (Dropout)          (None, 6, 6, 512)         0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 6, 6, 512)         2359808
_________________________________________________________________
batch_normalization_3 (Batch (None, 6, 6, 512)         2048
_________________________________________________________________
activation_3 (Activation)    (None, 6, 6, 512)         0
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 3, 3, 512)         0
_________________________________________________________________
dropout_3 (Dropout)          (None, 3, 3, 512)         0
_________________________________________________________________
flatten (Flatten)            (None, 4608)              0
_________________________________________________________________
dense (Dense)                (None, 256)               1179904
_________________________________________________________________
batch_normalization_4 (Batch (None, 256)               1024
_________________________________________________________________
activation_4 (Activation)    (None, 256)               0
_________________________________________________________________
dropout_4 (Dropout)          (None, 256)               0
_________________________________________________________________
dense_1 (Dense)              (None, 512)               131584
_________________________________________________________________
batch_normalization_5 (Batch (None, 512)               2048
_________________________________________________________________
activation_5 (Activation)    (None, 512)               0
_________________________________________________________________
dropout_5 (Dropout)          (None, 512)               0
_________________________________________________________________
dense_2 (Dense)              (None, 7)                 3591
=================================================================
Total params: 4,478,727
Trainable params: 4,474,759
Non-trainable params: 3,968
_________________________________________________________________
```

Figure 5 : Sequential Model

3.5 ANALYSIS

3.5.1 Process Model

We intend to use the Waterfall Model in the development of our project. The reasons for using this model in our project are given below.

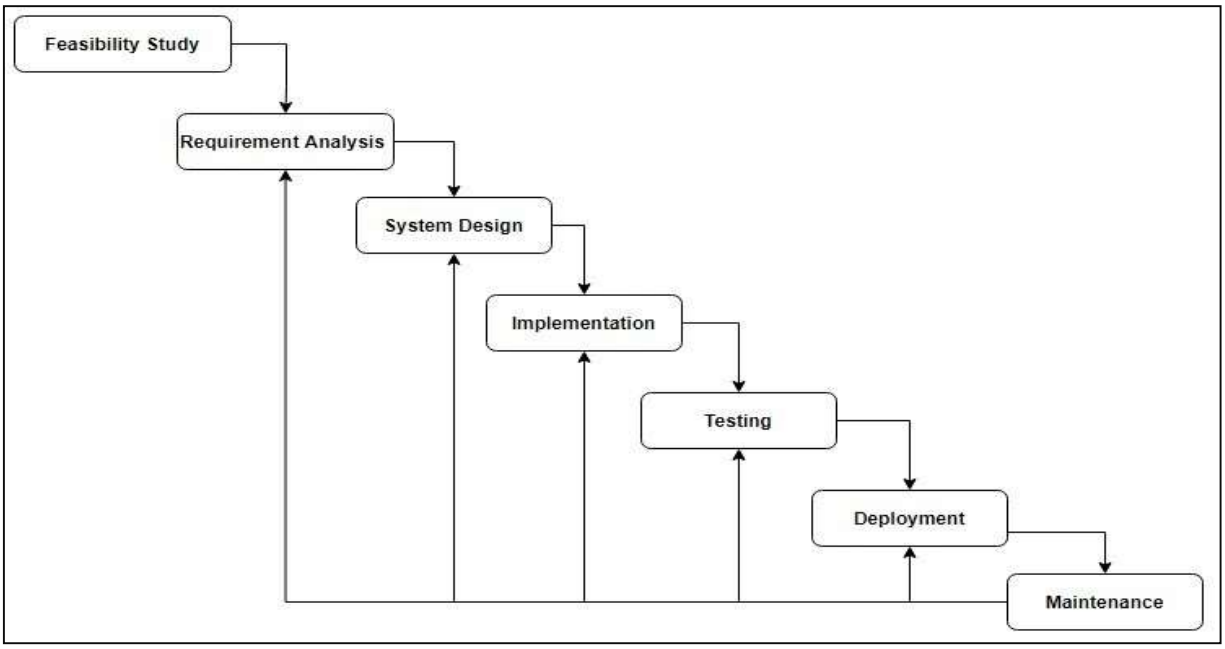


Fig 6: Waterfall Model

1. In waterfall, development of one phase starts only when the previous phase is complete. Because of this nature, each phase of the waterfall model is quite precise well defined. Since the phases fall from a higher level to lower level, like a waterfall, it is named as the waterfall model.
2. Since the requirements are stable and not changing frequently, the Waterfall Model is best suited for the project.
3. It enables the monitoring and departmentalization. A timeline can be set with deadlines for each development step, and a product can proceed through the development process model phases one by one.

4. The waterfall model progresses through easily understandable and explainable phases and thus it is easy to use.
5. It is easy to manage due to the rigidity of the model – each phase has specific deliverables and a review process.
6. Suited for smaller projects where requirements are well defined also Project is completely dependent on project team with minimum client intervention.
7. All the requirements are documented beforehand and Any changes in the system is made during the process of the development.

In this model, phases are processed and completed one at a time and they do not overlap.

3.5.2 Feasibility Analysis

- **Technical Feasibility:** Technical feasibility focuses on the technical resources (software and hardware) available to the organization and also helps to determine whether the technical team is capable of converting the ideas into working systems. The software required for our project are mainly open-source and readily available (i.e. Google colab, Python, sklearn, T-5 transformer etc.). No extra hardware is in use for our project except the laptop. Some specifics can be uncovered in detail during implementation.

- **Economic Feasibility:** This assessment typically involves a cost/ benefits analysis of the project. The project will be developed in minimal cost as most of the software required are open-source hence does not involve any upfront cost, the main cost incurred will be of the computer system.

- **Legal Feasibility:** This assessment investigates whether any aspect of the proposed project conflicts with legal requirements like zoning laws, data protection acts, or social media laws. The project does not involve any legal concerns since all the licenses and laws will be respected and included in the project.

- **Operational Feasibility:** This assessment involves undertaking a study to analyze and determine whether—and how well—the organization's needs can be met by completing the project. The main objective of this project is to generate varied types of questions according to user needs and with minimal human intervention.

- **Resource Feasibility:** The number of people working for the project and the technical resources provided are deemed to be of adequate nature after due analysis and hence the resource feasibility for the project is good.

3.5.3 Timeline Chart

Task Name	Start Date	End Date	Duration (Days)
Analysis	01-Mar-21	30-Apr-21	60
Requirements	01-May-21	30-Jun-21	60
Design	01-Jul-21	15-Aug-21	45
Development: ML Model	16-Aug-21	15-Oct-21	60
Development: User Interface	16-Oct-21	30-Nov-21	45
Testing & Integration	01-Dec-21	15-Dec-21	15
Final Documentation	15-Mar-22	31-Mar-22	15

Table 1: Timeline Chart

CHAPTER 4

SYSTEM IMPLEMENTATION

4.1. IMPLEMENTATION: UI DEVELOPMENT

UI Development :

In this project, Flask framework has been used for the UI development. The text area web page of the project takes the required inputs from the user in order to recognize personality traits. The user inputs required is the experience of leadership by the interviewee in the last company. Once the user inputs these fields, the text is preprocessed, trained on machine learning models and it is given to respective algorithm. Personality traits are displayed on the next web page after clicking submit button. The next web page gives introduction of each personality traits and gives the result of user's personality. It also shows the effect on the work based on the predicted personality. The facial expression page will open after we click on Start on video analysis page. Then it starts the camera and capture our image frame by frame and gives the live facial expression.

4.2 SOURCE CODES

- DATASET MERGING

```
df_essays = pd.read_csv('essays.csv', encoding='cp1252', delimiter=',', quotechar='"')

# for every essay, we replace the personalitiy categories
# of the essay wich are "y" and "n" with "1" and "0"
for e in df_essays.columns[2:7]:
    df_essays[e] = df_essays[e].replace('n', '0')
    df_essays[e] = df_essays[e].replace('y', '1')
    # not sure if we need this line: furter investigation possible:
    df_essays[e] = pd.to_numeric(df_essays[e])

df_kaggle = pd.read_csv('mbti_1.csv',  skiprows=0 )

def mbti_to_big5(mbti):
    mbti = mbti.lower()
    cEXT, cNEU, cAGR, cCON, cOPN = 0,np.NaN,0,0,0

    ## IN MBTI, extrovert or introvert
    ## correlates with Extroversion
    if mbti[0] == "i":
        cEXT = 0
    elif mbti[0] == "e":
        cEXT = 1

    ## IN MBTI, Feeler or Thinker
    ## correlates with Agrreableness
```

```
    if mbti[2] == "t":
        cAGR = 0
    elif mbti[2] == "f":
        cAGR = 1

    ## IN MBTI, Judger or Perceiver
    ## correlates with Conscientiousness
    if mbti[3] == "p":
        cCON = 0
    elif mbti[3] == "j":
        cCON = 1

    ## IN MBTI, Intuition or Sensing
    ## correlates with Openness
    if mbti[1] == "n":
        cOPN = 1
    elif mbti[1] == "s":
        cOPN = 0

    return cEXT, cNEU, cAGR, cCON, cOPN

df_kaggle["cEXT"] =   df_kaggle.apply(lambda x: mbti_to_big5(x.type)[0], 1)
df_kaggle["cNEU"] =   df_kaggle.apply(lambda x: mbti_to_big5(x.type)[1], 1)
df_kaggle["cAGR"] =   df_kaggle.apply(lambda x: mbti_to_big5(x.type)[2], 1)
df_kaggle["cCON"] =   df_kaggle.apply(lambda x: mbti_to_big5(x.type)[3], 1)
df_kaggle["cOPN"] =   df_kaggle.apply(lambda x: mbti_to_big5(x.type)[4], 1)
```

- TEXT PREPROCESSING

```
import re
import nltk
nltk.download('stopwords')
nltk.download('punkt')
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer, WordNetLemmatizer
from nltk.tokenize import word_tokenize, sent_tokenize
from wordcloud import WordCloud

import gensim
from gensim.utils import simple_preprocess
from gensim.parsing.preprocessing import STOPWORDS

stop_words = stopwords.words("english")
stop_words.remove('')

def preprocess(text):
    corpus = []

    for token in gensim.utils.simple_preprocess(text):
        if token not in gensim.parsing.preprocessing.STOPWORDS and len(token)>2 and token not
in stop_words:
            corpus.append(token)
    return corpus

essays_kaggle['clean'] = essays_kaggle['TEXT'].apply(preprocess)
essays_kaggle['clean_text'] = essays_kaggle['clean'].apply(lambda x:" ".join(x))
```

```
## BAG OF WORDS

from sklearn.feature_extraction.text import CountVectorizer
bow_vectorizer = CountVectorizer()

# create vectors from our words
train_x_vectors = bow_vectorizer.fit_transform(train_x)
test_x_vectors = bow_vectorizer.transform(test_x)
# # now that's a big thing :-O

## TFIDF VECTORIZER

from sklearn.feature_extraction.text import TfidfVectorizer

cv = TfidfVectorizer()
train_x_vectors_tf = cv.fit_transform(train_x)
test_x_vectors_tf = cv.transform(test_x)

# for evaluation save some data for later:
evaluation = []
evaluation_tf = []
data = len(essays_kaggle)
vec_name = "MBTI"

## SVM

from sklearn import svm
name = "svm"

print("training Extraversion cEXT using SVM...")
clf_svm_cEXT = svm.SVC(kernel='linear')
```

```
clf_svm_cEXT.fit(train_x_vectors, train_y_cEXT)
evaluation.append([data, vec_name, name, "cEXT", clf_svm_cEXT.score(test_x_vectors,
test_y_cEXT)])
print("cEXT score: ", clf_svm_cEXT.score(test_x_vectors, test_y_cEXT))
```

SAME TRAINING FOR ALL OTHER PERSONALITY TRAITS FOR BOW AND OTHER MACHINE LEARNING MODELS.

SVM - TFIDF

```
print("training Extraversion cEXT using SVM...")
clf_svm_cEXT = svm.SVC(kernel='linear')
clf_svm_cEXT.fit(train_x_vectors_tf, train_y_cEXT)
evaluation_tf.append([data, vec_name, name, "cEXT", clf_svm_cEXT.score(test_x_vectors_tf,
test_y_cEXT)])
print("cEXT score: ", clf_svm_cEXT.score(test_x_vectors_tf, test_y_cEXT))
```

SAME TRAINING FOR ALL OTHER PERSONALITY TRAITS FOR TFIDF AND OTHER MACHINE LEARNING MODELS.

- HYPERPARAMETER TUNING

```
from sklearn.svm import SVC

from sklearn.model_selection import GridSearchCV
param_grid = {'C':[100,1000], 'gamma':[0.001,0.0001]}

grid = GridSearchCV(SVC(), param_grid, verbose=2)

grid.fit(train_x_vectors, train_y_cEXT)

grid.score(test_x_vectors, test_y_cEXT)

## SVM

from sklearn import svm
name = "svm"

print("training Extraversion cEXT using SVM...")
clf_svm_cEXT = svm.SVC(C=1000, gamma=0.001)
clf_svm_cEXT.fit(train_x_vectors, train_y_cEXT)
evaluation.append([data, vec_name, name, "cEXT", clf_svm_cEXT.score(test_x_vectors,
test_y_cEXT)])
print("cEXT score: ", clf_svm_cEXT.score(test_x_vectors, test_y_cEXT))

SAME TRAINING FOR ALL OTHER PERSONALITY TRAITS FOR BOW AND TFIDF

from sklearn.model_selection import RandomizedSearchCV
random_grid = {'C':[100,1000], 'gamma':[0.001,0.0001]}
random_cv = RandomizedSearchCV(SVC(), random_grid, verbose=2)
random_cv.fit(train_x_vectors_tf, train_y_cEXT)
```

- FACIAL EXPRESSION RECOGNITION - DATASET GENERATION AND CNN MODEL

```
"""# category images"""

for expression in os.listdir("train/"):
    print(str(len(os.listdir("train/" + expression))) + " " + expression + " images")

"""# Generate Training and Validation Batches """

img_size = 48
batch_size = 64

datagen_train = ImageDataGenerator(horizontal_flip=True)
train_generator = datagen_train.flow_from_directory("train/",
                                                    target_size = (img_size,img_size),
                                                    color_mode = 'grayscale',
                                                    batch_size = batch_size,
                                                    class_mode = 'categorical',
                                                    shuffle = True)

datagen_validation = ImageDataGenerator(horizontal_flip=True)
validation_generator = datagen_validation.flow_from_directory("test/",
                                                    target_size = (img_size,img_size),
                                                    color_mode = 'grayscale',
                                                    batch_size = batch_size,
                                                    class_mode = 'categorical',
                                                    shuffle = True)

"""# Create CNN model"""
```

```
model = Sequential()

# 1 - conv layer
model.add(Conv2D(64, (3,3), padding='same', input_shape=(48,48,1)))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.25))

# 2 - conv layer
model.add(Conv2D(128, (5,5), padding='same'))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.25))

# 3 - conv layer
model.add(Conv2D(512, (3,3), padding='same'))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.25))

# 4 - conv layer
model.add(Conv2D(512, (3,3), padding='same'))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.25))
```

```
model.add(Flatten())

# FC 1
model.add(Dense(256))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(Dropout(0.25))

# FC 2
model.add(Dense(512))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(Dropout(0.25))

model.add(Dense(7, activation='softmax'))

opt = Adam(lr = 0.0005)

model.compile(optimizer = opt,
        loss = 'categorical_crossentropy',
        metrics = ['accuracy'])
model.summary()

"""# Train and Evaluate Model"""

epochs = 15
steps_per_epoch = train_generator.n//train_generator.batch_size
validation_steps = validation_generator.n//validation_generator.batch_size

checkpoint = ModelCheckpoint("model_weight.h5",
                    monitor = 'val_accuracy',
```

```
                    save_weights_only = True,
                    mode = 'max',
                    verbose=1)

reduce_lr = ReduceLROnPlateau(monitor = 'val_loss',
                     factor=0.1,
                     patience=2,
                     min_lr=0.00001,
                     mode='auto')

callbacks = [checkpoint, reduce_lr]

history = model.fit(x=train_generator,
                steps_per_epoch=steps_per_epoch,
                epochs=epochs,
                validation_data=validation_generator,
                validation_steps=validation_steps,
                callbacks=callbacks)

model_json = model.to_json()
with open ("model.json","w") as json_file:
   json_file.write(model_json)
```

- USER INTERFACE CODE

```
from tkinter import E
from flask import Flask, render_template, request, url_for, jsonify, Response
import pickle
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfVectorizer
import pandas as pd
import re
from  Video.camera import VideoCamera

app = Flask(__name__)

cNEU = pickle.load( open("Text\pickles\cNEU_project.p", "rb"))
cEXT = pickle.load( open("Text\pickles\cEXT_project (1).p", "rb"))
cAGR = pickle.load( open("Text\pickles\cAGR_project (1).p", "rb"))
cCON = pickle.load( open("Text\pickles\cCON_project (1).p", "rb"))
cOPN = pickle.load( open("Text\pickles\cOPN_project (1).p", "rb"))

with open("Text/pickles/tfidf_vectorizer_project (1).p", 'rb') as f:
    tfidf_transformer = pickle.load(f)

with open('Text/pickles/bow_vectorizer_project.p', 'rb') as f1:
    bow_transformer = pickle.load(f1)

def predict_personality(text):
    sentences = re.split("(?<=[.!?]) +", text)
    text_vector_31 = tfidf_transformer.transform(sentences)
    text_vector_32 = bow_transformer.transform(sentences)
    EXT = cEXT.predict(text_vector_31)
    NEU = cNEU.predict(text_vector_32)
```

```
    AGR = cAGR.predict(text_vector_31)
    CON = cCON.predict(text_vector_31)
    OPN = cOPN.predict(text_vector_31)
    return [EXT[0], NEU[0], AGR[0], CON[0], OPN[0]]

@app.route('/')
def hello_world():
    return render_template("index.html")

@app.route('/text.html')
def hello_world2():
    return render_template("text.html")

@app.route('/text.html', methods=['GET','POST'])
def pred():
    if request.method == 'POST':
        message = request.form['message']
        sentences = re.split("(?<=[.!?]) +", message)
        text_vector_31 = tfidf_transformer.transform(sentences)
        text_vector_32 = bow_transformer.transform(sentences)
        EXT = cEXT.predict(text_vector_31)
        NEU = cNEU.predict(text_vector_32)
        AGR = cAGR.predict(text_vector_31)
        CON = cCON.predict(text_vector_31)
        OPN = cOPN.predict(text_vector_31)
        pred = [EXT[0], NEU[0], AGR[0], CON[0], OPN[0]]

        return render_template("text.html", predictions=pred, mes=message)

@app.route('/text_predict.html', methods=['GET','POST'])
def hello_world3():
```

```
        return render_template("text_predict.html", predictions=pred, mes=message)


@app.route('/video_start.html')
def hello_video2():
    return render_template("video_start.html")



@app.route('/vid_index.html', methods=['GET','POST'])
def hello_video():
    return render_template("vid_index.html")

def gen(camera):

    while True:
        frame = camera.get_frame()
        yield (b'--frame\r\n'
               b'Content-Type: image/jpeg\r\n\r\n' + frame + b'\r\n\r\n')

@app.route('/video_feed')
def video_feed():
    return Response(gen(VideoCamera()),
                    mimetype='multipart/x-mixed-replace; boundary=frame')



if __name__ == "__main__":
    app.run(debug=True)
```

CHAPTER 5

RESULTS AND DISCUSSIONS

5.1 RESULTS

We processed with the Big five dataset for personality traits classification for candidate screening. We initially used the CountVectorizer (Bag of Words) method for the vectorization the words on the five personality traits dataset and then trained using Support vector machines, Logistic Regression, Naive Bayes algorithm, Random Forest, and Decision Trees models. In addition, we applied hyperparameter tuning for support vector machines such as GridsearchCV to get better accuracy. We got the better accuracy of NEU traits by hyperparameter tuning. The comparison of the result is given below:

	SVM	DT	NB	LogR	RF	SVM(Hyperparameter Tuning)
cEXT	55.46	51.81	50.41	55.66	53.84	55.26
cNEU	51.61	54.45	51.82	53.23	56.27	58.91
cAGR	52.83	50.41	50.61	53.41	54.04	54.04
cCON	50.81	51.41	53.03	53.03	56.27	53.44
cOPN	57.08	52.42	53.03	56.03	59.11	58.90

Table 2 : Data Essays:Vectorizer-Bag-of-words

Then, we used the TfIdf Vectorizer that takes not only the frequency but also the importance of words into account when analyzing a corpus. With the TfIdf vectorizer, we observed that accuracy increased for all the classification traits by around an increase of 2%. The results with Recurrent Neural networks are also not up to the mark.

	SVM	DT	NB	LogR	RF	SVM(Hyperparameter Tuning)
cEXT	53.84	55.66	49.79	55.06	53.64	55.66
cNEU	58.09	49.59	52.83	57.89	56.47	55.87
cAGR	56.07	51.61	50.61	55.87	49.79	56.07
cCON	54.64	51.01	52.22	55.66	52.83	53.44
cOPN	60.72	53.23	53.03	60.52	62.14	60.92

Table 3 : Data Essays:Vectorizer-TfIdf

Nevertheless, based on the Big 5 dataset, we could not achieve high accuracy. Following the research of personality traits in humans and getting a sense of the concept from the paper of Furnham [1996], we merged both five personality traits dataset and the MBTI dataset. By the help of CountVectorizer and TfIdf vectorizer, we convert words into vectors. Then, we trained the data on different machine learning models. By merging the MBTI dataset with the Big five personality, we achieved an increase of accuracy by around 20%.

	SVM	DT	NB	LogR	RF	SVM(Hyperparameter Tuning)
cEXT	72.31	72.45	69.71	75.91	71.42	78.86
cNEU	N/A	N/A	N/A	N/A	N/A	N/A
cAGR	70.39	68.28	54.82	72.11	71.01	75.81
cCON	68.05	64.61	57.37	69.62	59.62	72.22
cOPN	77.47	76.08	75.63	79.97	78.66	81.15

Table 4 : Data Essays+MBTI:Vectorizer-CountVectorizer

	SVM	DT	NB	LogR	RF	SVM(Hyperparameter Tuning)
cEXT	**78.01**	70.48	68.56	77.65	72.05	77.47
cNEU	N/A	N/A	N/A	N/A	N/A	N/A
cAGR	78.06	66.62	55.65	**79.63**	70.74	76.62
cCON	73.17	65.09	60.25	**74.21**	61.73	71.46
cOPN	**82.18**	76.08	74.18	81.56	79.47	81.29

Table 5 : Data-Essays + MBTI :Vectorizer- TfIdf

Now, goal is to estimate likely performance of a model on out-of-sample data, so we applied kFold cross validation on the whole data to reduce the chances of overfitting of the data or high variance. We can acquire a more accurate estimate of out-of-sample accuracy and a more "efficient" use of data with kfold cross validation (every observation is given for both training and testing).With the cross-validation, we get results as minimum 55.03% and maximum cross validation score as 84.29% for the Extraversion trait with 10 folds. The average cross validation score is 77.08% for EXT, 81.47% for OPN, 78.80% for AGR, 74.22% for CON that we achieved with the kfold cross validation on the complete dataset.

SVM and Logistic Regression are the most accurate and exact algorithms for our proposed research. Using all criteria, it produced great results for all attributes. SVM and Logistic Regression obtained maximum accuracy (82.18%) for cOPN trait. It has the best performance in all four aspects and all metrics. There are only four MBTI traits and five traits in the Big five personality traits dataset. So, we could not achieve better accuracy NEU trait. We achieved 58.91% accuracy for NEU traits based on the Big five personality traits dataset but better accuracy for the remaining four traits with the help of the MBTI dataset.

The comparison of present work with the previous research work Rahman et al. [2019] and Majumder et al. [2017] of personality traits classification (Extraversion, Conciousness, Agreeableness, Neuroticism, Openness) is given below:

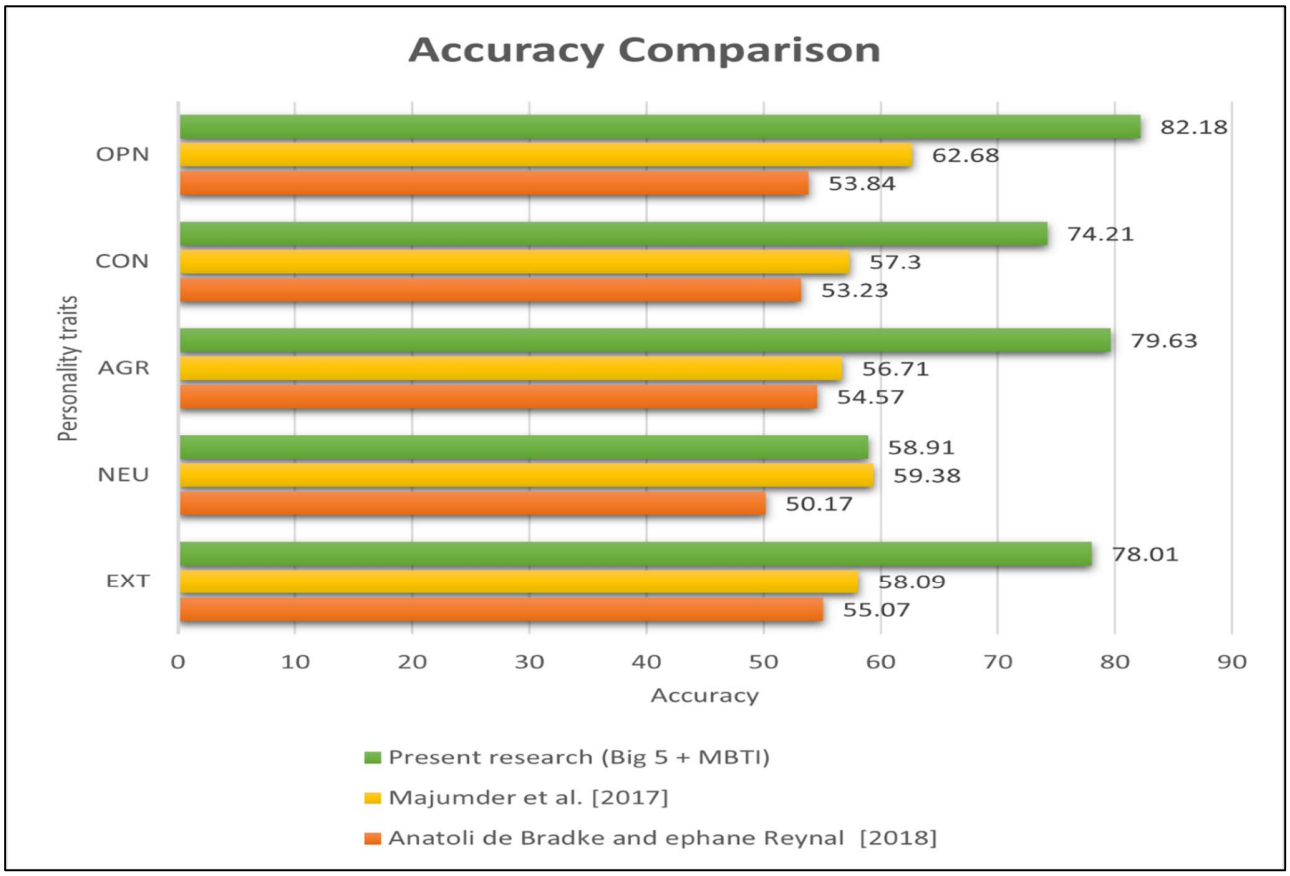


Figure 6 : Comparison of accuracy with present work

5.2. SNAPSHOTS OF RESULTS

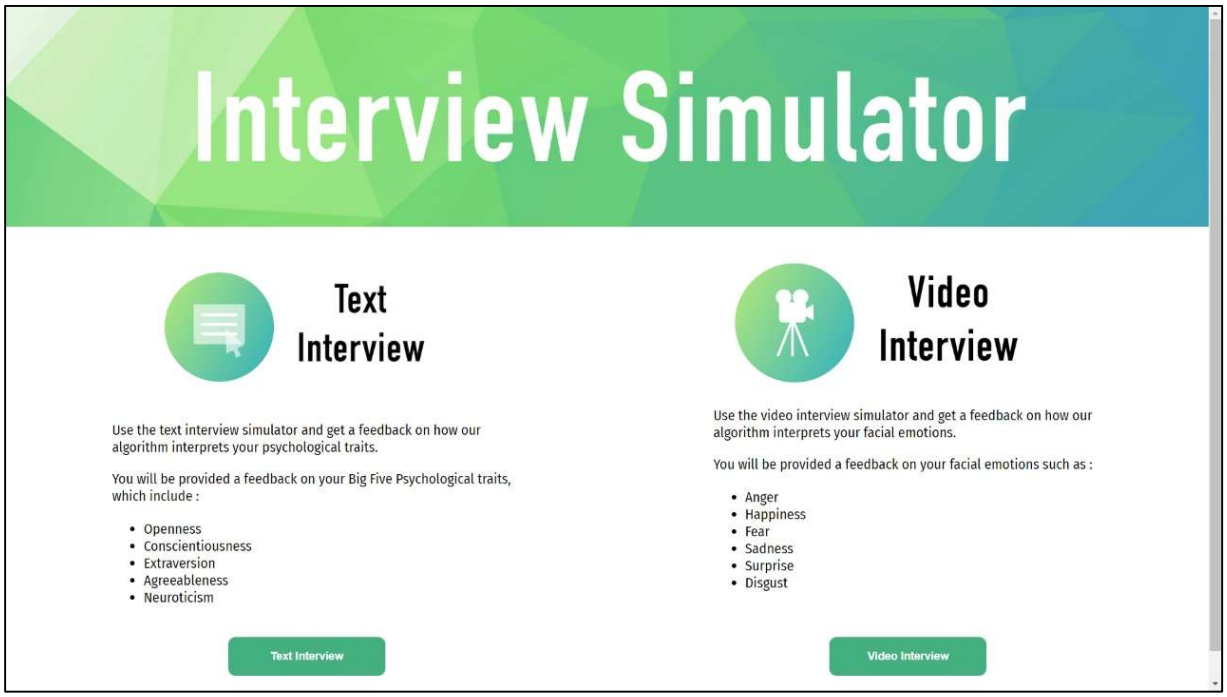


Figure 7 : User Interface 1

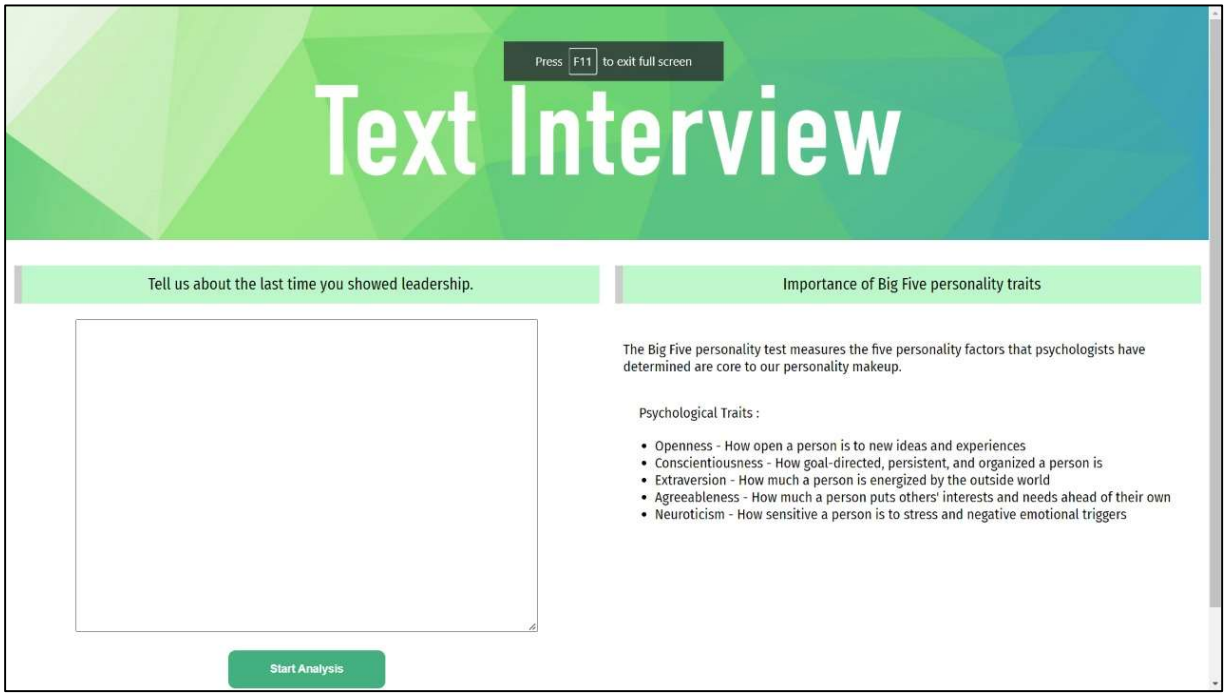


Figure 8 : User Interface 2

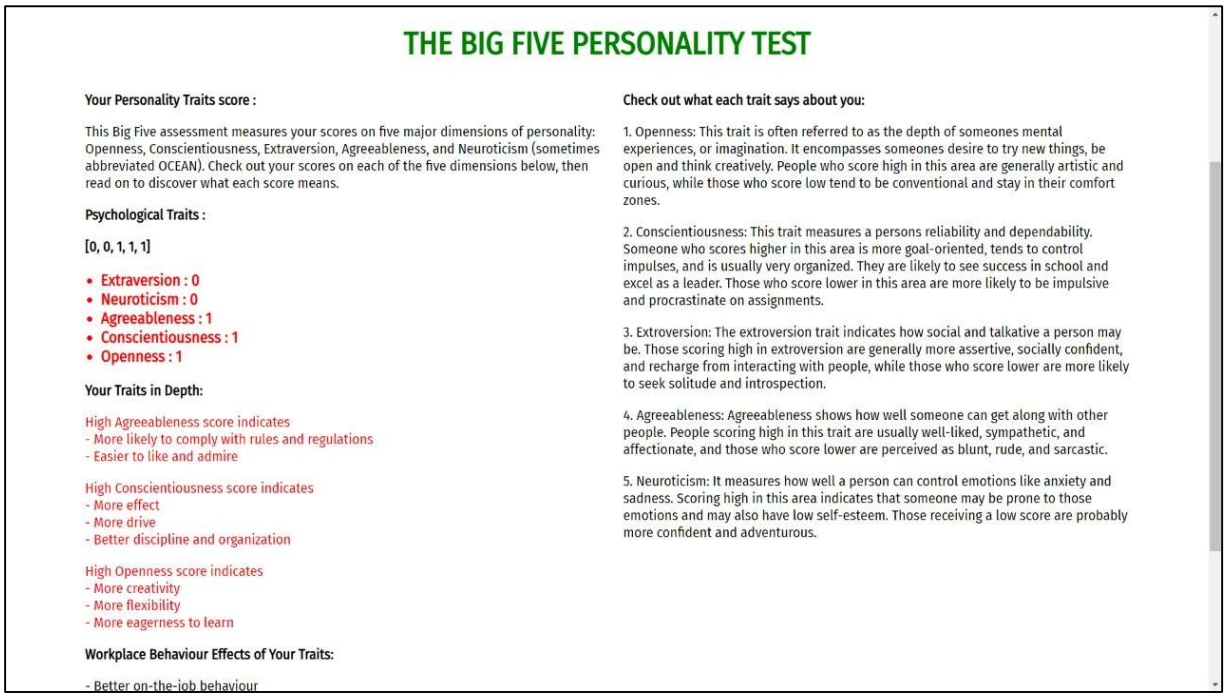


Figure 9 : User Interface 3

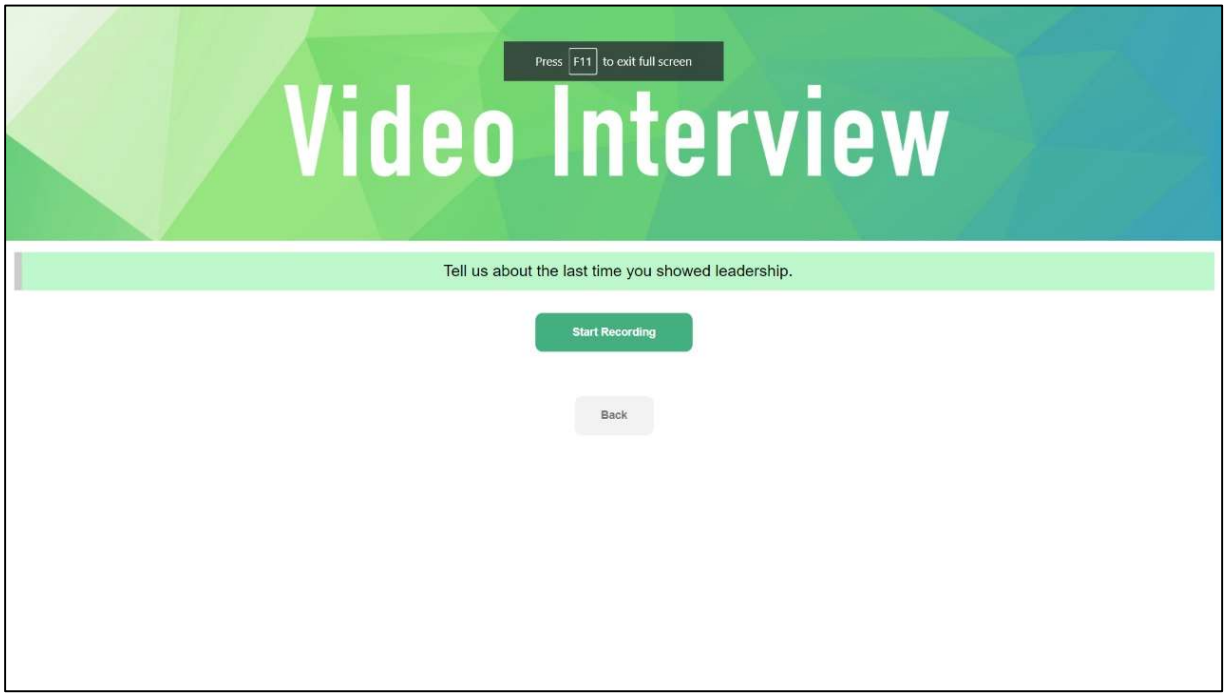


Figure 10 : User Interface 4

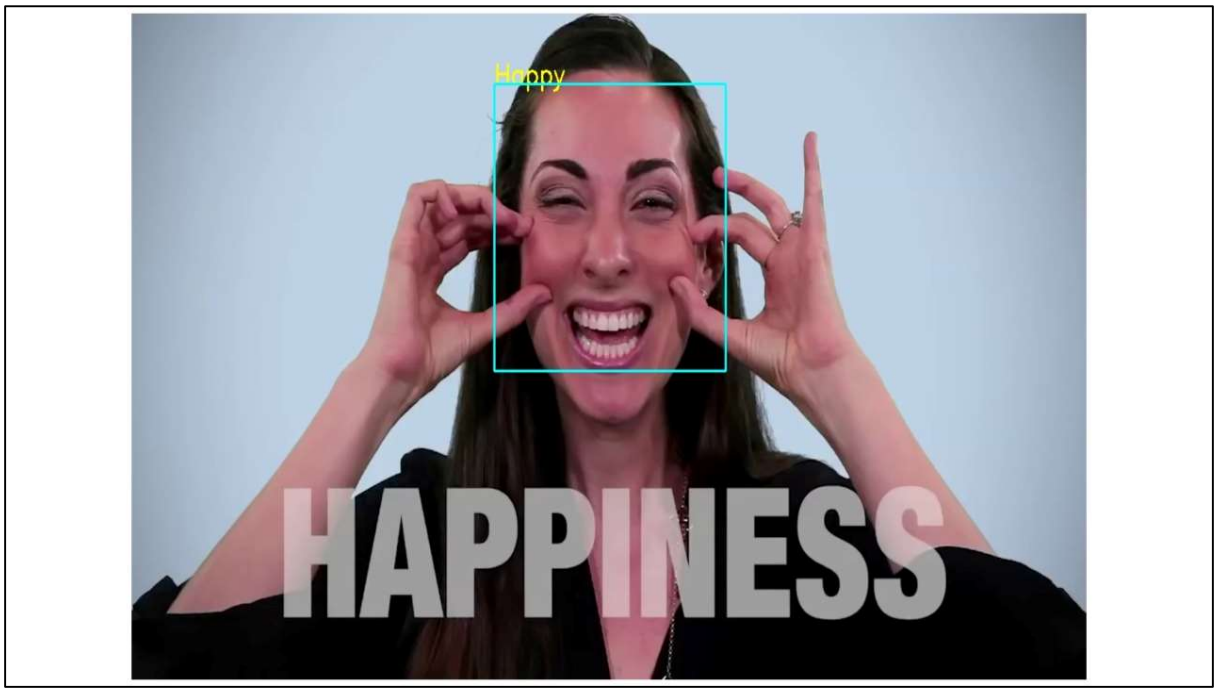


Figure 11 : User Interface 5

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

6.1 CONCLUSION

Using several machine learning models, this study provided a method for determining the optimum personality characteristic identification methodology. We are creating model which can help recruiters to analyse the emotions of candidates as candidate screening method which will eventually save a lot of time of recruiter and will get alternative to traditional interview process. This research will surely help interviewers and companies in the online interview process where company could not able analyse the emotions and personality traits of interviewee. This method consists of data filtering, data preprocessing, data merging, re-sampling of the data, and classification modeling. First, we merged the Big 5 dataset and the MBTI dataset. Following that, Countvectorizer and TfIdf vectorizer was used to vectorize the processed data and then trained with Support Vector Machine, Logistic Regression, Random Forest, Decision Trees, and Naive Bayes models.

For comparative experimental analysis, five personality traits were used: EXT, NEU, AGR, CON, and OPN. Support Vector Machine and Logistic Regression outperformed other machine learning models, according to the comparative results. We observed Logistic Regression and Support Vector Machine model gives higher accuracy for the binary classification of the text data into personality traits with an accuracy score 78.01% for EXT, 79.63% for AGR, 58.91% for NEU, 74.21% for CON, and 81.56% for OPN traits. However, the Naive Bayes algorithm resulted in overall lower performance. For the Facial Expression analysis, we got 65.58% accuracy using Convolutional Neural Networks based on the images that are categorized into seven emotions (Happy, Sad, Angry, Surprise, Disgust, Neutral, Fear).

6.2 FUTURE SCOPE

For future work, we plan to investigate the impact of deep learning techniques with neural networks for better accuracy. The personality traits recognition can be extended to audio and video emotion recognition with help of Convolutional Neural Networks and Speech recognition methods.

REFERENCES

[1] Elena Bajic. How the mbti can help you build a stronger company— forbes, 2015. URL https://www.forbes.com/sites/elenabajic/2015/09/28/ how-the-mbti-can-help-you-build-a-stronger-company/?sh=51754881d93c. [Online; accessed 1st-September-2021].

[2] Md Abdur Rahman, Asif Al Faisal, Tayeba Khanam, Mahfida Amjad, and Md Saeed Siddik. Personality detection from text using convolutional neural network. In 2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT), pages 1–6. IEEE, 2019.

[3] Kasula Chaithanya Pramodh and Y Vijayalata. Automatic personality recognition of authors using big five factor model. In 2016 IEEE International Conference on Advances in Computer Applications (ICACA), pages 32–37. IEEE, 2016.

[4] Raad Bin Tareaf, Seyed Ali Alhosseini, Philipp Berger, Patrick Hennig, and Christoph Meinel. Towards automatic personality prediction using facebook likes metadata. In 2019 IEEE 14th International Conference on Intelligent Systems and Knowledge Engineering (ISKE), pages 714–719. IEEE, 2019.

[5] Rapha el Lederman Anatoli de Bradk e, Ma el Fabien and St ephane Reynal. Multimodal emotion recognition. Projet Fil Rouge 2018-2019, 2018. URL https://www.overleaf.com/project/5c06f9e12aee1927b458fc4a.

[6] Navonil Majumder, Soujanya Poria, Alexander Gelbukh, and Erik Cambria. Deep learning-based document modeling for personality detection from text. IEEE Intelligent Systems, 32(2):74–79, 2017.

[7] Abir Abyaa, Mohammed Khalidi Idrissi, and Samir Bennani. Predicting the learner's personality from educational data using supervised learning. In Proceedings of the 12th International Conference on Intelligent Systems: Theories and Applications, pages 1–7, 2018.

[8] Waiel Tinwala and Shristi Rauniyar. Big five personality detection using deep convolutional neural networks. 2021.

[9] KN Pavan Kumar and Marina L Gavrilova. Personality traits classification on twitter. In 2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), pages 1–8. IEEE, 2019.

[10] Kaggle. Myers-briggs personality type dataset, 2017. URL https://www. kaggle.com/datasnaek/mbti-type. [Online; accessed 16-August-2021].

[11] Isabel Briggs Myers. The myers-briggs type indicator: Manual (1962). 1962.

[12] Paul T Costa Jr and Robert R McCrae. The Revised Neo Personality Inventory (neo-pi-r). Sage Publications, Inc, 2008.

[13] Michal Kosinski, Sandra C Matz, Samuel D Gosling, Vesselin Popov, and David Stillwell. Facebook as a research tool for the social sciences: Opportunities, challenges, ethical considerations, and practical guidelines. American psychologist, 70(6):543, 2015.

[14] James W Pennebaker and Laura A King. Linguistic styles: language use as an individual difference. Journal of personality and social psychology, 77(6): 1296, 1999.

[15] Franc¸ois Mairesse, Marilyn A Walker, Matthias R Mehl, and Roger K Moore. Using linguistic cues for the automatic recognition of personality in conversation and text. Journal of artificial intelligence research, 30:457–500, 2007.

[16] Adrian Furnham. The big five versus the big four: the relationship between the myers-briggs type indicator (mbti) and neo-pi five factor model of personality. Personality and Individual Differences, 21(2):303–307, 1996. ISSN 0191-8869. doi: https://doi.org/10.1016/0191-8869(96)00033-5. URL https://www.sciencedirect.com/science/article/pii/0191886996000335

[17] GeeksforGeeks. Svm hyperparameter tuning using gridsearchcv ml— geeksforgeeks,

[18] 2019. URL https://en.wikipedia.org/wiki/Support-vector machine# Bayesian SVM. [Online; accessed 2nd-September-2021].

[19] KDnuggets. Algorithms, decision trees, explained. URL https://www. kdnuggets.com/2020/01/decision-tree-algorithm-explained.html. [Online; accessed 3rd-September-2021].

[20] Ashwin Raj. Perfect recipe for classification using logistic regression, 2020. URL https://towardsdatascience.com/ the-perfect-recipe-for-classification-using-logistic-regression-f8648e267592. [Online; accessed 3rd-September-2021].

[21] Mehendale, N. Facial emotion recognition using convolutional neural networks (FERC). SN Appl. Sci. 2, 446 (2020). https://doi.org/10.1007/s42452-020-2234-1

[22] Zhang, Hongli & Jolfaei, Alireza & Alazab, Mamoun. (2019). A Face Emotion Recognition Method Using Convolutional Neural Network and Image Edge Computing. IEEE Access. PP. 1-1. 10.1109/ACCESS.2019.2949741.