

**A Project Report**  
on  
**PERSONALITY PREDICTION SYSTEM**

*Submitted in partial fulfillment of the  
requirement for the award of the degree of*

**Bachelor of Technology in Computer Science and  
Engineering**



**Under The Supervision of  
Dr. Alok Katiyar  
Professor GU**

**Submitted By**

**Asmi Mishra (19SCSE1050010)  
Shishu Pandey (19SCSE1180063)**

**SCHOOL OF COMPUTING SCIENCE AND ENGINEERING  
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING  
GALGOTIAS UNIVERSITY, GREATER NOIDA  
INDIA**

**MAY, 2023**



**SCHOOL OF COMPUTING SCIENCE AND  
ENGINEERING  
GALGOTIAS UNIVERSITY, GREATER NOIDA**

**CANDIDATE'S DECLARATION**

I/We hereby certify that the work which is being presented in the thesis/project/dissertation, entitled **“PERSONALITY PREDICTION SYSTEM”** in partial fulfillment of the requirements for the award of the BACHELOR OF TECHNOLOGY submitted in the School of Computing Science and Engineering of Galgotias University, Greater Noida, is an original work carried out during the period of January, 2023 to May 2023, under the supervision of Dr. Alok katiyar (Professor,GU) , Department of Computer Science and Engineering/Computer Application and Information and Science, of School of Computing Science and Engineering , Galgotias University, Greater Noida

The matter presented in the thesis/project/dissertation has not been submitted by me/us for the award of any other degree of this or any other places.

Asmi Mishra (19SCSE1050010)

Shishu Pandey (19SCSE1050010)

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Dr. Alok Katiyar

Professor, GU

**CERTIFICATE**

The Final Thesis/Project/ Dissertation Viva-Voce examination of Asmi Mishra (19SCSE1050010) and Shishu Pandey (19SCSE1180063) has been held on 15-05-2023 and his/her work is recommended for the award of -Bachelor of technology.

**Signature of Examiner(s)**

**Signature of Supervisor(s)**

**Signature of Program Chair**

**Signature of Dean**

Date: May, 2023

Place: Greater Noida

## **Abstract**

This will enable a more effective way to short list submitted candidate CVs from a large number of applicants providing a consistent and fair CV ranking policy, which can be legally justified. System will rank the experience and key skills required for particular job position. Than system will rank the CV's based on the experience and other key skills which are required for particular job profile. This system will help the HR department to easily shortlist the candidate based on the CV ranking policy. This system will focus not only in qualification and experience but also focuses on other important aspects which are required for particular job position. This system will help the human resource department to select right candidate for particular job profile which in turn provide expert workforce for the organization. Candidate here will register him/herself with all its details and will upload their own CV into the system which will be further used by the system to shortlist their CV. Candidate can also give an online test which will be conducted on personality questions as well as aptitude questions. After completing the online test, candidate can view their own test results in graphical representation with marks. Personality prediction system selects the right and suitable candidate required for desired job profile by grading the resume on certain criteria like experience, skills etc. This system can also be used in other fields where they need to hire candidates by filtering processes in a very less time. Our application uses Spacy module to analyze, summarize and compare to get similarity score between resume and job description.

***Keywords- CV, Shortlisting, Machine learning, Candidate, HR, Big Five Method***

## Table of Contents

<b>Title</b>	<b>Page No.</b>
<b>Candidates Declaration</b>	<b>2</b>
<b>Acknowledgement</b>	<b>3</b>
<b>Abstract</b>	<b>4</b>
<b>Contents</b>	<b>5</b>
<b>List of Table</b>	<b>6</b>
<b>List of Figures</b>	<b>7</b>
<b>Acronyms</b>	<b>8</b>
<b>Chapter 1 Introduction</b>	<b>9</b>
1.1 Introduction	<b>9</b>
1.2 Project objectives	<b>11</b>
<b>Chapter 2 Literature Survey/Project Design</b>	<b>12</b>
<b>Chapter 3 Functionality/Working of Project</b>	<b>14</b>
<b>Chapter 4 Requirement analysis</b>	<b>31</b>
<b>Chapter 5 Results and discussion</b>	<b>44</b>
<b>Chapter 6 Conclusion and Future Scope</b>	<b>52</b>
<b>Reference</b>	<b>53</b>

### List of Table

<b>S.No.</b>	<b>Caption</b>	<b>Page No.</b>
<b>1</b>	<b>Ocean Keywords</b>	<b>16</b>
<b>2</b>	<b>Ocean score and personality type</b>	<b>17</b>

## List of Figures

<b>S.No.</b>	<b>Title</b>	<b>Page No.</b>
<b>1</b>	<b>Big five model</b>	<b>10</b>
<b>2</b>	<b>Solution of problem</b>	<b>14</b>

## **Acronyms**

B.Tech.	Bachelor of Technology
M.Tech.	Master of Technology
BCA	Bachelor of Computer Applications
MCA	Master of Computer Applications
B.Sc. (CS)	Bachelor of Science in Computer Science
M.Sc. (CS)	Master of Science in Computer Science
SCSE	School of Computing Science and Engineering



# CHAPTER-1

## Introduction

### 1.1 Introduction

The word Personality derives from the Latin word persona which refers to a mask worn by actors to act. However, Personality is much more than a mask now, it could possibly determine whether a person is suitable for a particular job profile. It tells us if a human is capable enough to lead, influence and communicate effectively with others. The first step of recruitment is the job application which consists of personal details, experience, and most importantly CV. Companies typically receive thousands of applications per job opening and have a dedicated team of screeners to select qualified candidates. It is very difficult for human beings to manually go through the CV of all applicants. Many candidates get filtered out in the first round itself on the basis of suitability, improper CV, not being skilled enough. Hiring the right candidate is a very difficult task as no candidate is perfect, some might not be skilled enough or some might not have the right personality. Hence, we propose a way in which the process of shortlisting gets streamlined and faster by personality prediction. CVs can reflect upon the professional qualifications of a person but do not reflect upon the personality of a person. Personality is one of the vital factors which suggests how a person would be able to work in a designated role, hence personality analysis and understanding is key. Our objective doing this project is to make the machine more human, and analyze the candidate in such a way that an actual human reviewer would. This paper tries to explore and implement various machine learning algorithms and analyze which one among them provides the best accuracy with a wide array of data provided. We also attempt to visualize the data and form a connection between various factors. The project is based on identifying the personality of an individual using machine learning algorithms and big 5 models. The personality of a human plays a major role in his personal and professional life. Nowadays, many organizations have also started shortlisting the candidates based on their personality as this increase the efficiency of the work because the person is working in what he is good at than what he is forced to do.

The Big Five model is also known as the Five-Factor Model (FFM) and OCEAN model was developed in the early 1980s according to many psychological theories. When the statistical

analysis is applied to personality survey data, some words used to describe the person and these words give a summary of the overall character or personality of the person accurately.

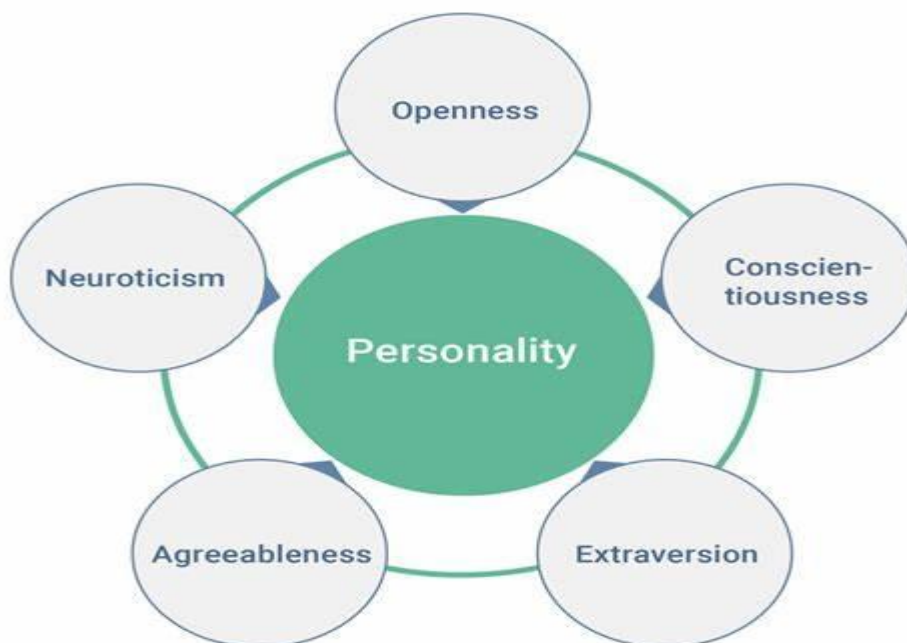
**Open to Experience:** It involves various dimensions, like imagination, sensitivity, attentiveness, preference to variety, and curiosity.

**Conscientiousness:** This trait is used to describe the carefulness and diligence of the person. It is the quality that describes how organized and efficient a person is.

**Extraversion:** It is the trait that describes how the best candidates can interact with people that is how good are his/her social skills.

**Agreeableness:** It is a quality that analyses the individual behavior based on the generosity, sympathy, cooperativeness and ability to adjust with people.

**Neuroticism:** This trait usually describes a person to have mood swings and has extreme expressive power.



Big Five Model

## **1.2 Project Objectives**

The objectives of the project are as stated below:

- To develop a system to provide a more effective way of short-listing the candidates.
- To determine the key skill characteristic by defining each expert's preferences and ranking decisions.
- To automate the process of requirement specifications and applicant's ranking.
- To conduct online aptitude and personality test.
- To produce ranking decisions that would have relatively higher consistency than those of human experts

## **CHAPTER-2**

### **LITERATURE SURVEY**

In 2014 an Integrated E-Recruitment System for Automated Personality Mining and Applicant Ranking was proposed by Faliagka et al. an automated candidate ranking was implemented by this system. It was based on objective criteria that the candidate's details would be extracted from the candidate's LinkedIn profile. The candidates' personality traits were automatically extracted from their social presence using linguistic analysis. The candidate's rank was derived from individual selection criteria using Analytical Hierarchy Process (AHP), while their weight was controlled by the recruiter (admin). The limitations of the system were that senior positions that required expertise and certain qualifications were screened inconsistently [1].

Liden et al. published The General Factor of Personality: The interrelations among the Big Five personality factors (Openness, Conscientiousness, Extraversion, Agreeableness, and Neuroticism) were analyzed in this paper to test for the existence of a GFP. The meta-analysis provides evidence for a GFP at the highest hierarchal level and that the GFP had a substantive component as it is related to supervisor-rated job performance were concluded by this paper. However, it is also realized that it is important to note that the existence of a GFP did not mean that other personality factors that were lower in the hierarchy lost their relevance [2].

Kalghatgi et al. presented a Neural Network Approach based on the Big Five Test to predict the personality of individuals depending on tweets published on Twitter by extracting meta-attributes from tweets. Which are used to analyze ones social behaviour. The authors followed a four-step process which is Data Collection from tweets, Preprocessing, Transformation and Classification. Although neural networks are used to predict personality there are limitations such as countering fake information, automatic analysis of tweets and relying on just Twitter is not enough to predict someones personality but only user behaviour and trends [3]

Allan Robey et al proposed a system to reduce the load on the Human Resource department of companies by having two sides: organization and candidate oriented. The authors claim that the proposed system will be more effective to shortlist CVs from a large pool making sure that the

ranking is fair and legal. The main difference between the existing system and the proposed system is that instead of just scanning the CVs, the authors propose to conduct an aptitude test and a personality test for personality prediction [4]

Juneja Afzal Ayub Zubeda et al worked on a project to rank CVs using Natural Language Processing and Machine Learning. The system ranks CVs in any format according to the company's criteria. The authors propose to consider candidates Github and LinkedIn profile as well to get a better understanding making it easier for the company to find a suitable match based on skillsets, ability and most importantly, personality [5]

Md. Tanzim Reza and Md. Sakib Zaman analyzed CV of individuals using Natural Language Processing and Machine Learning by first converting CVs to HTML and then reverse engineering to HTML code following which, segment finalization and qualification feature extraction has been done. The model extracts data from a CV and segments them based on the values. They have classified the CVs using multivariate logistic regression. However, the size of the dataset was very less.[6]

## Chapter-3

### Functionality/Working of Project

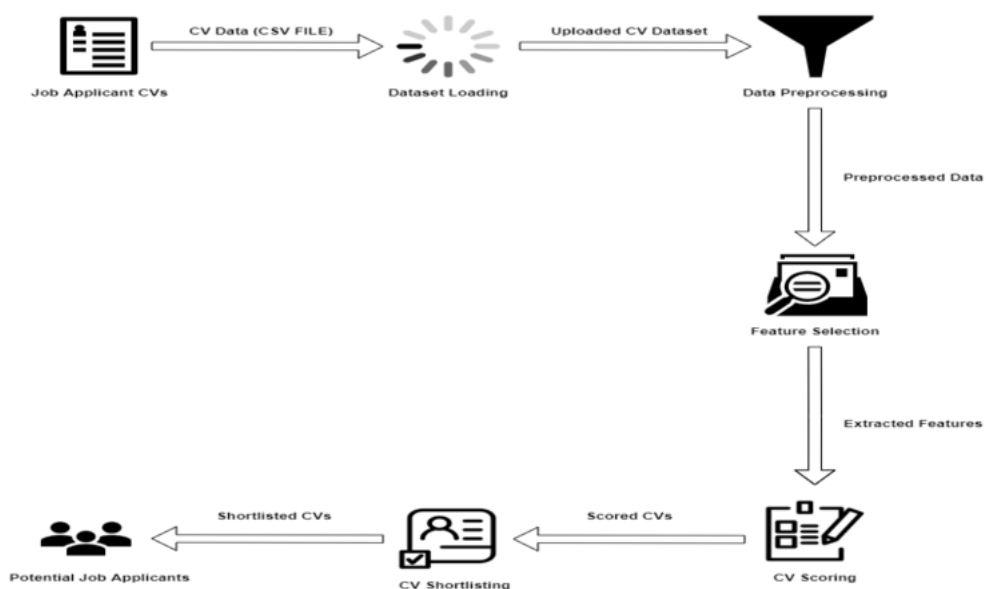
#### 3.1 Problem Statement

There is a huge workload on the human resource department to select the right candidate for a particular job profile which in turn would provide experts workforce for the organization from a large pool of candidates.

#### 3.2 Solution

The proposed system will enable a more effective way to short list submitted candidate CVs from a large number of applicants providing a consistent and fair CV ranking policy. This can be legally justified. System will rank the experience and key skills required for a particular job position than system will rank the CV's based on the experience and other key skills which are required for particular job profile. This system will help the HR department to easily shortlist the candidate based on the CV ranking policy

Figure 2 : Solution of the problem



### 3.3 Working of the system

Dataset As manual data collection is time-consuming, we collected candidate resumes through a lot of websites and personal interaction with potential job seekers taking the total count to 708 CVs. The collected CVs were in PDF and DOCx format. Methodology The objective of our paper is to predict the personality of a person based on their score of openness, extraversion, agreeableness, neuroticism and conscientiousness. For achieving this, we needed a way to calculate the scores directly from every CV. Our approach as shown in Fig.1 was to parse the entire resume and search for keywords relating to the Big Five Test. The Big Five Personality Traits model is based on findings from several independent researchers, and it dates back to the late 1950s. But the model as we know it now began to take shape in the 1990s. Lewis Goldberg, a researcher at the Oregon Research Institute, is credited with naming the model "The Big Five." It is now considered to be an accurate and respected personality scale, which is routinely used by businesses and in psychological research. The Big Five Personality Traits Model measures five key dimensions of people's personalities:

Openness: sometimes called "Intellect" or "Imagination," this measures your level of creativity, and your desire for knowledge and new experiences.

Conscientiousness: this looks at the level of care that you take in your life and work. If you score highly in conscientiousness, you'll likely be organized and thorough, and know how to make plans and follow them through. If you score low, you'll likely be lax and disorganized.

Extraversion/Introversion: this dimension measures your level of sociability. Are you outgoing or quiet, for instance? Do you draw energy from a crowd, or do you find it difficult to work and communicate with other people?

Agreeableness: this dimension measures how well you get on with other people. Are you considerate, helpful and willing to compromise? Or do you tend to put your needs before others'?

Natural Reactions: sometimes called "Emotional Stability" or "Neuroticism," this measure emotional reactions. Do you react negatively or calmly to bad news? Do you worry obsessively about small details, or are you relaxed in stressful situations? For parsing CVs, we have used pyresparser- a simple resume parser used for extracting important features such as name, email id, description, skills from CVs. Pyresparser supports PDF and DOCx files. The parsed data is then stored in a CSV file.

**Table 1: – Ocean Keyword**

<b>Openness</b>	<b>Conscientiousness</b>	<b>Extraversion</b>	<b>Agreeableness</b>	<b>Neuroticism</b>
Imaginative	Thoughtful	Cheerful	Trustworthy	Calm
Insightful	Goal oriented	Sociable	Altruism	Strong hearted
Curious	Ambitious	Talkative	Kind	Collected
Creative	Organised	Assertive	Affectionate	Balanced
Outspoken	Mindful	Outgoing	Cooperative	Peaceful
Straightforward	Vigilant	Energetic	Empathetic	Tranquil
Direct	Control	Extroverted	Modest	Strong will
Receptive	Discipline	Friendly	Sympathetic	Emotionally stable
Open minded	Reliable	Enthusiastic	Compliant	Serene
Adventurous	Responsible	Outspoken	Tendermindedness	Resilient

Table 1 given above houses various keywords of OCEAN. Each trait is associated with a set of 10 keywords that relate to it. There are many Natural Language Processing (NLP) libraries like Natural Language Toolkit (NLTK), TextBlob, SpaCY which could help us in parsing the resume data. We have used SpaCY- an open-source software library for advanced natural language processing and is helpful to handle large amounts of text data.

The Phrase Matcher class in spaCY is highly efficient in matching large sequences of tokens in documents

The keywords in Table 1 will be matched by the mentioned class. Using Phrase Matcher class, our algorithm searches for the keywords and gives a score from a range of 0-10 according to the occurrence of OCEAN keywords in ones CV. After assigning scores as shown in the table below, the algorithm labels each data point as dependable, extraverted, lively, responsible, or serious. Thus, we output a CSV file with degrees of the Big five traits as all the columns. Each datapoint



has been labelled as either dependable, extraverted, lively, responsible, or serious as given in Table 2 below

Table 2: – OCEAN Score and Personality type

Openness	Conscientiousness	Extraversion	Agreeableness	Neuroticism	Personality
6	4	7	5	4	Extraverted
4	6	4	4	7	Serious
5	6	4	7	4	Lively
7	4	5	4	5	Dependable
5	7	6	6	3	responsible

### 3.4 Model training and testing

For predicting the personality of a prospective candidates, we have used various machine learning algorithms like Logistic regression, Naïve Bayes, k-NN, SVM, and Random forest. Before training our model, we label encoded the Personality column of our dataset. Our final dataset had 708 rows and 6 rows. Using the sklearn library, we have used 70% of our data for training purposes and 30% for testing the results. For predicting the personality of a prospective candidate, we have used various machine learning algorithms like Logistic Regression, Naive Bayes, Random Forest, Support Vector Machine (SVM) and KNN.

#### Logistic Regression

It is an algorithm analogous to Linear Regression, except it predicts whether something is True or False. It is a popular algorithm for solving classification problems like Binary Classification (Pass/Fail, Rain/No Rain). Logistic regression is one of the most popular Machine Learning

algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables.

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 similar to the Linear Regression except that how they are used. Linear Regression is used for solving Regression problems, whereas Logistic regression is used for solving the classification problems.

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 has the ability to provide probabilities and classify new data using continuous and discrete datasets.

Logistic Regression can be used to classify the observations using different types of data and can easily determine the most effective variables used for the classification.

## **Naïve Bayes**

In probability, Bayes theorem is used to compute the conditional probability. The theorem forms the basis of the Naïve Bayes classifier, a classification algorithm that assumes strong independence assumptions between the features. According to the algorithm, each feature in the problem makes an equal and independent contribution to the outcome. 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 on the basis of the probability of an object. Some popular examples of Naïve Bayes Algorithm are spam filtration, Sentimental analysis, and classifying articles.

Why is it called Naïve Bayes?

The Naïve Bayes algorithm is comprised of two words Naïve and Bayes, Which can be described as:

Naïve: 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. Hence each feature individually contributes to identify that it is an apple without depending on each other.

Bayes: It is called Bayes because it depends on the principle of Bayes' Theorem.

## **kNN**

kNN stands for k-nearest neighbours, a supervised machine learning algorithm capable of solving both regression and classification problems. Intuitively we can think of the proverb Birds of the same feather flock together as similar to kNN. The algorithm assumes that similar data points usually occur in close proximity. K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique.

K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.

K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm.

K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.

K-NN is a non-parametric algorithm, which means it does not make any assumption on underlying data.

It is also called a lazy learner algorithm because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.

KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.

Example: Suppose, we have an image of a creature that looks similar to cat and dog, but we want to know either it is a cat or dog. So for this identification, we can use the KNN algorithm, as it works on a similarity measure. Our KNN model will find the similar features of the new data set to the cats and dogs images and based on the most similar features it will put it in either cat or dog category.

## **SVM**

Support vector machine is a supervised machine learning algorithm used to deal with data for classification and regression analysis. The goal of SVM is to find a hyperplane in N-dimensional space ( N- number of features) that can easily classify the data points. 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.

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.

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.

## Random Forest

Random forest is another ensemble technique used for classification and regression tasks. It uses multiple decision trees to produce the output. Bagging or bootstrap aggregation are used to train the random forest algorithm's "forest."

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.

As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

The below diagram explains the working of the Random Forest algorithm:

PlayNext

Unmute

Current Time 0:00

/

Duration 18:10

Loaded: 0.37%

Fullscreen

Fullscreen

Backward Skip 10s Play Video Forward Skip 10s

Note: To better understand the Random Forest Algorithm, you should have knowledge of the Decision Tree Algorithm.

Assumptions for Random Forest

Since the random forest combines multiple trees to predict the class of the dataset, it is possible that some decision trees may predict the correct output, while others may not. But together, all

the trees predict the correct output. Therefore, below are two assumptions for a better Random forest classifier:

There should be some actual values in the feature variable of the dataset so that the classifier can predict accurate results rather than a guessed result.

The predictions from each tree must have very low correlations.

Why use Random Forest?

Below are some points that explain why we should use the Random Forest algorithm:

<="" li="">

It takes less training time as compared to other algorithms.

It predicts output with high accuracy, even for the large dataset it runs efficiently.

It can also maintain accuracy when a large proportion of data is missing.

How does Random Forest algorithm work?

Random Forest works in two-phase first is to create the random forest by combining N decision tree, and second is to make predictions for each tree created in the first phase.

The Working process can be explained in the below steps and diagram:

Step-1: Select random K data points from the training set.

Step-2: Build the decision trees associated with the selected data points (Subsets).

Step-3: Choose the number N for decision trees that you want to build.

Step-4: Repeat Step 1 & 2.

Step-5: For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes.

The working of the algorithm can be better understood by the below example:

Example: Suppose there is a dataset that contains multiple fruit images. So, this dataset is given to the Random forest classifier. The dataset is divided into subsets and given to each decision tree. During the training phase, each decision tree produces a prediction result, and when a new data point occurs, then based on the majority of results, the Random Forest classifier predicts the final decision.

After training our model on all of the algorithms, we realised that our predictions turned out to be rather poor. Even our best models could only find an accuracy of about 30 percent.

Another factor was that our training and testing datasets had very different distributions. While our training data was a little imbalanced, the testing data was even more imbalanced. But when we put ourselves in the shoes of an employer, we realise that he would want to hire someone who is responsible and lively more than anything else. Thus our problem now turns into a binary classification problem (1- responsible or lively 0-others)

### **3.5 Methodology and algorithms**

- Lexical analysis
- Syntactic analysis
- Semantic analysis
- Disclosure integration
- Pragmatic analysis

#### **1) Lexical analysis**

Lexical Analysis is the first phase of the compiler also known as a scanner. It converts the High level input program into a sequence of Tokens.

Lexical Analysis can be implemented with the Deterministic finite Automata.

The output is a sequence of tokens that is sent to the parser for syntax analysis

How Lexical Analyzer works-

Input preprocessing: This stage involves cleaning up the input text and preparing it for lexical analysis. This may include removing comments, whitespace, and other non-essential characters from the input text.

Tokenization: This is the process of breaking the input text into a sequence of tokens. This is usually done by matching the characters in the input text against a set of patterns or regular expressions that define the different types of tokens.

Token classification: In this stage, the lexer determines the type of each token. For example, in a programming language, the lexer might classify keywords, identifiers, operators, and punctuation symbols as separate token types.

Token validation: In this stage, the lexer checks that each token is valid according to the rules of the programming language. For example, it might check that a variable name is a valid identifier, or that an operator has the correct syntax.

Output generation: In this final stage, the lexer generates the output of the lexical analysis process, which is typically a list of tokens. This list of tokens can then be passed to the next stage of compilation or interpretation.

## **2) Syntactic analysis**

Syntactic analysis or parsing or syntax analysis is the third phase of NLP. The purpose of this phase is to draw exact meaning, or you can say dictionary meaning from the text. Syntax analysis checks the text for meaningfulness comparing to the rules of formal grammar. For example, the sentence like “hot ice-cream” would be rejected by semantic analyzer.

Parsing, syntax analysis, or syntactic analysis is the process of analyzing a string of symbols, either in natural language, computer languages or data structures, conforming to the rules of a formal grammar. The term parsing comes from Latin *pars* (orationis), meaning part (of speech).[1]

The term has slightly different meanings in different branches of linguistics and computer science. Traditional sentence parsing is often performed as a method of understanding the exact meaning of a sentence or word, sometimes with the aid of devices such as sentence diagrams. It usually emphasizes the importance of grammatical divisions such as subject and predicate.

Within computational linguistics the term is used to refer to the formal analysis by a computer of a sentence or other string of words into its constituents, resulting in a parse tree showing their syntactic relation to each other, which may also contain semantic and other information (p-values).[citation needed] Some parsing algorithms may generate a parse forest or list of parse trees for a syntactically ambiguous input.[2]

The term is also used in psycholinguistics when describing language comprehension. In this context, parsing refers to the way that human beings analyze a sentence or phrase (in spoken language or text) "in terms of grammatical constituents, identifying the parts of speech, syntactic relations, etc." [1] This term is especially common when discussing which linguistic cues help speakers interpret garden-path sentences.



Within computer science, the term is used in the analysis of computer languages, referring to the syntactic analysis of the input code into its component parts in order to facilitate the writing of compilers and interpreters. The term may also be used to describe a split or separation.

### **3) Semantic analysis**

Semantic Analysis is a subfield of Natural Language Processing (NLP) that attempts to understand the meaning of Natural Language. Understanding Natural Language might seem a straightforward process to us as humans. However, due to the vast complexity and subjectivity involved in human language, interpreting it is quite a complicated task for machines. Semantic Analysis of Natural Language captures the meaning of the given text while taking into account context, logical structuring of sentences and grammar roles.

In order to understand the meaning of a sentence, the following are the major processes involved in Semantic Analysis:

1. Word Sense Disambiguation
2. Relationship Extraction

Word Sense Disambiguation:

In Natural Language, the meaning of a word may vary as per its usage in sentences and the context of the text. Word Sense Disambiguation involves interpreting the meaning of a word based upon the context of its occurrence in a text.

For example, the word ‘Bark’ may mean ‘the sound made by a dog’ or ‘the outermost layer of a tree.’

Likewise, the word ‘rock’ may mean ‘*a stone*’ or ‘*a genre of music*’ – hence, the accurate meaning of the word is highly dependent upon its context and usage in the text.

Thus, the ability of a machine to overcome the ambiguity involved in identifying the meaning of a word based on its usage and context is called Word Sense Disambiguation.

Relationship Extraction:

Another important task involved in Semantic Analysis is Relationship Extracting. It involves firstly identifying various entities present in the sentence and then extracting the relationships between those entities.

#### **4) Disclosure integration**

While processing a language there can arise one major ambiguity known as referential ambiguity. Referential ambiguity is the ambiguity that can arise when a reference to a word cannot be determined. For example,

Ram won the race.

Mohan ate half of a pizza.

He liked it.

In the above example, “He” can be Ram or Mohan. This creates an ambiguity. The word “He” shows dependency on both sentences. This is known as *disclosure integration*. It means when an individual sentence relies upon the sentence that comes before it. Like in the above example the third sentence relies upon the sentence before it. Hence the goal of this model is to remove referential ambiguity.

#### **5) Pragmatic analysis**

The pragmatic analysis means handling the situation in a much more practical or realistic manner than using a theoretical approach. As we know that a sentence can have different meanings in various situations. For example, The average is 18.

The average is 18. (average may be of sequence)

The average is 18. (average may be of a vehicle)

The average is 18. (average may be of a mathematical term)

We can see that for the same input there can be different perceptions. To interpret the meaning of the sentence we need to understand the situation. To tackle such problems we use pragmatic analysis. The pragmatic analysis tends to make the understanding of the language much more clear and easy to interpret.

### Description of flow and methods in the system:

1. train\_model class: It contains two method which train the model and predict the result by giving the various values.

a. train method: It read the dataset for training the model from a csv file and build a model using Logistic Regression. It uses different 7 values for training the model.

```
self.mul_lr = linear_model.LogisticRegression(multi_class='multinomial',
                                              solver='newton-cg',
                                              max_iter =1000)

self.mul_lr.fit(mainarray, train_y)
```

b. test method: It predict the personality of a person by passing an array of values that contains gender, age and other 5 personality characteristics.

```
test_predict=list()
for i in test_data:
    test_predict.append(int(i))
y_pred = self.mul_lr.predict([test_predict])
return y_pred
```

2. main method: We start with creating an object of train\_model class and train the model by calling train method of class. Then we initialize a variable with Tk object and design the

landing page of system using labels and button. A button with name Predict Personality is designed which calls predict\_person method.

```
if __name__ == "__main__":
    '''initialize system with training model'''
    model = train_model()
    model.train()

    root = Tk()
    root.geometry('700x500')
    root.configure(background='white')
    root.title("Personality Prediction System")
    titleFont = font.Font(family='Helvetica', size=25, weight='bold')
    homeBtnFont = font.Font(size=12, weight='bold')
    lab=Label(root, text="Personality Prediction System", bg='white', font=titleFont, pady=30).pack()
    b2=Button(root, padx=4, pady=4, width=30, text="Predict Personality", bg='black', foreground='white',
    root.mainloop()
```

3. predict\_person method: We withdraw the root tkinter window and create a new toplevel window and configure its size and attributes. We label the heading of window followed by various labels and their entries. For selecting of a resume file, user needs to press choose file button which then calls Openfile method that takes an argument of button. In predict\_person method, various entries are taken for predicting the personality. Submit button pass all the values to prediction\_result.

4. OpenFile method: It tries to open the directory with default address name and file types and except if file not chosen. After try except block, the method changes the name of choose file button in predict\_person method with the base name of file so that user can know about the chosen file.

```

name = filedialog.askopenfilename(initialdir="C:/Users/Batman/Documents/Programming/tkinter/",
                                  filetypes=(("Document", "*.docx"), ("PDF", "*.pdf"), ('All files', '*')),
                                  title = "Choose a file."
                                  )
try:
    filename=os.path.basename(name)
    loc=name
except:
    filename=name
    loc=name
b4.config(text=filename)
return

```

5. prediction\_result method: This method firstly closes the previous tkinter window which was used to take the data from user. After this, it calls test method of model object and stores the result returned by method. After this it parse all the information from resume and stores in a variable followed by a try except block which try to delete name and validate mobile number from fetched information from resume. Then it prints all the data submitted by user on console. After this, the method popup a full screen window which shows all the parsed information and predicted personality on GUI window along with the definition of each personality characteristic's definition.

6. check\_type method: It converts various strings and numbers into desired format and converts lists and tuples in string.

```
def check_type(data):  
    """Check datatype of string and convert and return"""  
    if type(data)==str or type(data)==str:  
        return str(data).title()  
    if type(data)==list or type(data)==tuple:  
        str_list=""  
        for i,item in enumerate(data):  
            str_list+=item+", "  
        return str_list  
    else: return str(data)
```

## Chapter-4

### Requirement Analysis

#### Dependencies of system:

##### 1) Pandas

Pandas is a Python package that provides fast, flexible, and expressive data structures designed to make working with "relational" or "labeled" data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, real world data analysis in Python. Additionally, it has the broader goal of becoming the most powerful and flexible open source data analysis / manipulation tool available in any language. It is already well on its way towards this goal.

##### Main Features

Here are just a few of the things that pandas does well:

Easy handling of missing data (represented as `NaN`, `NA`, or `NaT`) in floating point as well as non-floating point data

Size mutability: columns can be inserted and deleted from `DataFrame` and higher dimensional objects

Automatic and explicit data alignment: objects can be explicitly aligned to a set of labels, or the user can simply ignore the labels and let `Series`, `DataFrame`, etc. automatically align the data for you in computations

Powerful, flexible group by functionality to perform split-apply-combine operations on data sets, for both aggregating and transforming data

Make it easy to convert ragged, differently-indexed data in other Python and NumPy data structures into `DataFrame` objects

Intelligent label-based slicing, fancy indexing, and subsetting of large data sets

Intuitive merging and joining data sets

Flexible reshaping and pivoting of data sets

Hierarchical labeling of axes (possible to have multiple labels per tick)

Robust IO tools for loading data from flat files (CSV and delimited), Excel files, databases, and saving/loading data from the ultrafast HDF5 format

Time series-specific functionality: date range generation and frequency conversion, moving window statistics, date shifting and lagging

## **2) Numpy**

NumPy is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, fourier transform, and matrices. NumPy was created in 2005 by Travis Oliphant. It is an open source project and you can use it freely. NumPy stands for Numerical Python.

Why Use NumPy?

In Python we have lists that serve the purpose of arrays, but they are slow to process. NumPy aims to provide an array object that is up to 50x faster than traditional Python lists. The array object in NumPy is called ndarray, it provides a lot of supporting functions that make working with ndarray very easy. Arrays are very frequently used in data science, where speed and resources are very important.

## **3) Tkinter**

Python offers multiple options for developing GUI (Graphical User Interface). Out of all the GUI methods, tkinter is the most commonly used method. It is a standard Python interface to the Tk GUI toolkit shipped with Python. Python with tkinter is the fastest and easiest way to create the GUI applications. Creating a GUI using tkinter is an easy task.

To create a tkinter app:

Importing the module – tkinter

Create the main window (container)

Add any number of widgets to the main window

Apply the event Trigger on the widgets.

Importing tkinter is same as importing any other module in the Python code. Note that the name of the module in Python 2.x is 'Tkinter' and in Python 3.x it is 'tkinter'.



There are two main methods used which the user needs to remember while creating the Python application with GUI.

`Tk(screenName=None, baseName=None, className='Tk', useTk=1)`: To create a main window, `tkinter` offers a method `'Tk(screenName=None, baseName=None, className='Tk', useTk=1)'`. To change the name of the window, you can change the `className` to the desired one. The basic code used to create the main window of the application is: `m=tkinter.Tk()` where `m` is the name of the main window object

`mainloop()`: There is a method known by the name `mainloop()` is used when your application is ready to run. `mainloop()` is an infinite loop used to run the application, wait for an event to occur and process the event as long as the window is not closed.

`m.mainloop()`

`tkinter` also offers access to the geometric configuration of the widgets which can organize the widgets in the parent windows. There are mainly three geometry manager classes class.

`pack()` method:It organizes the widgets in blocks before placing in the parent widget.

`grid()` method:It organizes the widgets in grid (table-like structure) before placing in the parentwidget.

`plce()` method:It organizes the widgets by placing them on specific positions directed by the programmer.

There are a number of widgets which you can put in your `tkinter` application. Some of the major widgets are explained below:

**Button:** To add a button in your application, this widget is used.

The general syntax is:

`w=Button(master, option=value)`

`master` is the parameter used to represent the parent window.

There are number of options which are used to change the format of the Buttons. Number of options can be passed as parameters separated by commas. Some of them are listed below.

`activebackground`: to set the background color when button is under the cursor.

activeforeground: to set the foreground color when button is under the cursor.

bg: to set the normal background color.

command: to call a function.

font: to set the font on the button label.

image: to set the image on the button.

width: to set the width of the button.

height: to set the height of the button.

Canvas: It is used to draw pictures and other complex layout like graphics, text and widgets.

The general syntax is:

w = Canvas(master, option=value) master is the parameter used to represent the parent window.

There are number of options which are used to change the format of the widget. Number of options can be passed as parameters separated by commas. Some of them are listed below.

bd: to set the border width in pixels.

bg: to set the normal background color.

cursor: to set the cursor used in the canvas.

highlightcolor: to set the color shown in the focus highlight.

width: to set the width of the widget.

height: to set the height of the widget.

CheckBox: To select any number of options by displaying a number of options to a user as toggle buttons. The general syntax is:

w = CheckBox(master, option=value)

There are number of options which are used to change the format of this widget. Number of options can be passed as parameters separated by commas. Some of them are listed below.

Title: To set the title of the widget.

activebackground: to set the background color when widget is under the cursor.

activeforeground: to set the foreground color when widget is under the cursor.

bg: to set the normal backgrouSteganography

Break

Secret Code:

Attach a File:nd color.

command: to call a function.

font: to set the font on the button label.

image: to set the image on the widget.

Entry:It is used to input the single line text entry from the user.. For multi-line text input, Text widget is used.

The general syntax is:

w=Entry(master, option=value)master is the parameter used to represent the parent window.

There are number of options which are used to change the format of the widget. Number of options can be passed as parameters separated by commas. Some of them are listed below.

bd: to set the border width in pixels.

bg: to set the normal background color.

cursor: to set the cursor used.

command: to call a function.

highlightcolor: to set the color shown in the focus highlight.

width: to set the width of the button.

height: to set the height of the button.

Frame: It acts as a container to hold the widgets. It is used for grouping and organizing the widgets. The general syntax is:

w = Frame(master, option=value)master is the parameter used to represent the parent window.

There are number of options which are used to change the format of the widget. Number of options can be passed as parameters separated by commas. Some of them are listed below.

highlightcolor: To set the color of the focus highlight when widget has to be focused.

bd: to set the border width in pixels.

bg: to set the normal background color.

cursor: to set the cursor used.

width: to set the width of the widget.

height: to set the height of the widget.

Label: It refers to the display box where you can put any text or image which can be updated any time as per the code.

The general syntax is:

w=Label(master, option=value) master is the parameter used to represent the parent window.

There are number of options which are used to change the format of the widget. Number of options can be passed as parameters separated by commas. Some of them are listed below.

bg: to set the normal background color.

bg to set the normal background color.

command: to call a function.

font: to set the font on the button label.

image: to set the image on the button.

width: to set the width of the button.

height” to set the height of the button.

Listbox: It offers a list to the user from which the user can accept any number of options.

The general syntax is: w = Listbox(master, option=value)

master is the parameter used to represent the parent window.

There are number of options which are used to change the format of the widget. Number of options can be passed as parameters separated by commas. Some of them are listed below.

highlightcolor: To set the color of the focus highlight when widget has to be focused.

bg: to set the normal background color.

bd: to set the border width in pixels.

font: to set the font on the button label.

image: to set the image on the widget.

width: to set the width of the widget.

height: to set the height of the widget.

MenuButton: It is a part of top-down menu which stays on the window all the time. Every menubutton has its own functionality. The general syntax is: `w = MenuButton(master, option=value)` master is the parameter used to represent the parent window.

There are number of options which are used to change the format of the widget. Number of options can be passed as parameters separated by commas. Some of them are listed below.

activebackground: To set the background when mouse is over the widget.

activeforeground: To set the foreground when mouse is over the widget.

bg: to set the normal background color.

bd: to set the size of border around the indicator.

cursor: To appear the cursor when the mouse over the menubutton.

image: to set the image on the widget.

width: to set the width of the widget.

height: to set the height of the widget.

highlightcolor: To set the color of the focus highlight when widget has to be focused.

Menu: It is used to create all kinds of menus used by the application.

The general syntax is: `w = Menu(master, option=value)` master is the parameter used to represent the parent window.

There are number of options which are used to change the format of this widget. Number of options can be passed as parameters separated by commas. Some of them are listed below.

title: To set the title of the widget.

activebackground: to set the background color when widget is under the cursor.

activeforeground: to set the foreground color when widget is under the cursor.

bg: to set the normal background color.

command: to call a function.

font: to set the font on the button label.

image: to set the image on the widget.

**Message:** It refers to the multi-line and non-editable text. It works same as that of Label.

The general syntax is: `w = Message(master, option=value)` master is the parameter used to represent the parent window.

There are number of options which are used to change the format of the widget. Number of options can be passed as parameters separated by commas. Some of them are listed below.

**bd:** to set the border around the indicator.

**bg:** to set the normal background color.

**font:** to set the font on the button label.

**image:** to set the image on the widget.

**width:** to set the width of the widget.

**height:** to set the height of the widget.

**RadioButton:** It is used to offer multi-choice option to the user. It offers several options to the user and the user has to choose one option.

The general syntax is: `w = RadioButton(master, option=value)`

There are number of options which are used to change the format of this widget. Number of options can be passed as parameters separated by commas. Some of them are listed below.

**activebackground:** to set the background color when widget is under the cursor.

**activeforeground:** to set the foreground color when widget is under the cursor.

**bg:** to set the normal background color.

**command:** to call a function.

**font:** to set the font on the button label.

**image:** to set the image on the widget.

**width:** to set the width of the label in characters.

**height:** to set the height of the label in characters.

**Scale:** It is used to provide a graphical slider that allows to select any value from that scale.

The general syntax is: `w = Scale(master, option=value)` master is the parameter used to represent the parent window.

There are number of options which are used to change the format of the widget. Number of options can be passed as parameters separated by commas. Some of them are listed below.

cursor: To change the cursor pattern when the mouse is over the widget.

activebackground: To set the background of the widget when mouse is over the widget.

bg: to set the normal background color.

orient: Set it to HORIZONTAL or VERTICAL according to the requirement.

from\_: To set the value of one end of the scale range.

to: To set the value of the other end of the scale range.

image: to set the image on the widget.

width: to set the width of the widget.

Scrollbar: It refers to the slide controller which will be used to implement listed widgets.

The general syntax is: `w = Scrollbar(master, option=value)` master is the parameter used to represent the parent window.

There are number of options which are used to change the format of the widget. Number of options can be passed as parameters separated by commas. Some of them are listed below.

width: to set the width of the widget.

activebackground: To set the background when mouse is over the widget.

bg: to set the normal background color.

bd: to set the size of border around the indicator.

cursor: To appear the cursor when the mouse over the menubutton.

Text: To edit a multi-line text and format the way it has to be displayed.

The general syntax is: `w =Text(master, option=value)`

There are number of options which are used to change the format of the text. Number of options can be passed as parameters separated by commas. Some of them are listed below.

highlightcolor: To set the color of the focus highlight when widget has to be focused.

insertbackground: To set the background of the widget.

bg: to set the normal background color.

font: to set the font on the button label.  
image: to set the image on the widget.  
width: to set the width of the widget.  
height: to set the height of the widget.

TopLevel: This widget is directly controlled by the window manager. It don't need any parent window to work on. The general syntax is: `w = TopLevel(master, option=value)`

There are number of options which are used to change the format of the widget. Number of options can be passed as parameters separated by commas. Some of them are listed below.

bg: to set the normal background color.  
bd: to set the size of border around the indicator.  
cursor: To appear the cursor when the mouse over the menubutton.  
width: to set the width of the widget.  
height: to set the height of the widget.

SpinBox: It is an entry of 'Entry' widget. Here, value can be input by selecting a fixed value of numbers. The general syntax is: `w = SpinBox(master, option=value)`

There are number of options which are used to change the format of the widget. Number of options can be passed as parameters separated by commas. Some of them are listed below.

bg: to set the normal background color.  
bd: to set the size of border around the indicator.  
cursor: To appear the cursor when the mouse over the menubutton.  
command: To call a function.  
width: to set the width of the widget.  
activebackground: To set the background when mouse is over the widget.  
disabledbackground: To disable the background when mouse is over the widget.  
from\_: To set the value of one end of the range.



to: To set the value of the other end of the range.

PannedWindow: It is a container widget which is used to handle number of panes arranged in it. The general syntax is: `w = PannedWindow(master, option=value)` master is the parameter used to represent the parent window.

There are number of options which are used to change the format of the widget. Number of options can be passed as parameters separated by commas. Some of them are listed below.

bg: to set the normal background color.

bd: to set the size of border around the indicator.

cursor: To appear the cursor when the mouse over the menubutton.

width: to set the width of the widget.

height: to set the height of the widget.

#### **4) functools**

Functools module is for higher-order functions that work on other functions. It provides functions for working with other functions and callable objects to use or extend them without completely rewriting them. This module has two classes – partial and partialmethod.

Partial class

A partial function is an original function for particular argument values. They can be created in Python by using “partial” from the functools library. The `__name__` and `__doc__` attributes are to be created by the programmer as they are not created automatically.

Partialmethod class

It is a method definition of an already defined function for specific arguments like a partial function. However, it is not callable but is only a method descriptor. It returns a new partialmethod descriptor.

#### **5) pyresparser**

A simple resume parser used for extracting information from resumes.

##### **Features**

Extract name, Extract email, Extract mobile numbers, Extract skills, Extract total experience, Extract college name, Extract degree, Extract designation, Extract company names.

## 6) Sklearn

scikit-learn comes with a few standard datasets, for instance the iris and digits datasets for classification and the diabetes dataset for regression.

In the following, we start a Python interpreter from our shell and then load the iris and digits datasets. Our notational convention is that \$ denotes the shell prompt while >>> denotes the Python interpreter prompt:

```
$ python
```

```
>>> from sklearn import datasets
```

```
>>> iris = datasets.load_iris()
```

```
>>> digits = datasets.load_digits()
```

A dataset is a dictionary-like object that holds all the data and some metadata about the data. This data is stored in the .data member, which is a n\_samples, n\_features array. In the case of supervised problems, one or more response variables are stored in the .target member. More details on the different datasets can be found in the dedicated section

## 7) nltk

Natural language processing (NLP) is an interdisciplinary subfield of linguistics, computer science, and artificial intelligence concerned with the interactions between computers and human language, in particular how to program computers to process and analyze large amounts of natural language data. The goal is a computer capable of "understanding" the contents of documents, including the contextual nuances of the language within them. The technology can then accurately extract information and insights contained in the documents as well as categorize and organize the documents themselves.

Challenges in natural language processing frequently involve speech recognition, natural-language understanding, and natural-language generation.

## 8) ScaPy

Scapy is a packet manipulation tool for computer networks,[3][4] originally written in Python by Philippe Biondi. It can forge or decode packets, send them on the wire, capture them, and match requests and replies. It can also handle tasks like scanning, tracerouting, probing, unit tests, attacks, and network discovery.

Scapy provides a Python interface into libpcap or native raw sockets, in a similar way to that in which Wireshark provides a view and capture GUI. It differs by supporting packet injection, custom packet formats and scripting. While it is a command-line only tool, it can still interface with a number of other programs to provide visualisation including Wireshark, GnuPlot for providing graphs, graphviz or VPython for interactive displaying, etc.

Starting with 2.4.0, Scapy supports Python 2.7 and 3.4+. It must not be confused with "scapy3k" (now renamed Kamene), which is an outdated independent and unmaintained fork.

## Chapter-5

### Results and discussions

We run the code given below:

```
import os import pandas as pd import numpy as np from tkinter import * from tkinter import
filedialog import tkinter.font as font from functools import partial from pyresparser import
ResumeParser
from sklearn import datasets, linear_model
class train_model:

    def train(self):
        data =pd.read_csv('training_dataset.csv')
        array = data.values
        for i in range(len(array)):
            if array[i][0]=="Male":
                array[i][0]=1
            else:
                array[i][0]=0
        df=pd.DataFrame(array)
        maindf =df[[0,1,2,3,4,5,6]]
        mainarray=maindf.values
        temp=df[7]
        train_y =temp.values

        self.mul_lr =
        linear_model.LogisticRegression(multi_class='multinomial', solver='newton-cg',max_iter
        =1000)
        self.mul_lr.fit(mainarray, train_y)

    def test(self, test_data):
        try:
            test_predict=list()
```

```

for i in test_data:
    test_predict.append(int(i))
y_pred = self.mul_lr.predict([test_predict])
    return y_pred
except:
    print("All Factors For Finding Personality Not Entered!")
def check_type(data):
if type(data)==str or type(data)==str:
    return str(data).title()
if type(data)==list or type(data)==tuple:
    str_list=""
for i,item in enumerate(data):
    str_list+=item+", "
    return str_list else: return str(data)
def prediction_result(top, aplcnt_name, cv_path, personality_values):
    "after applying a job" top.withdraw()
    applicant_data={"Candidate Name":aplcnt_name.get(),
"CV Location":cv_path}

    age = personality_values[1]

    print("\n##### Candidate Entered Data #####\n")
    print(applicant_data, personality_values)

    personality = model.test(personality_values)
    print("\n##### Predicted Personality #####\n") print(personality)
    data = ResumeParser(cv_path).get_extracted_data()
try:
    del data['name']
if len(data['mobile_number'])<10:
    del data['mobile_number']

```

```

except: pass

print("\n##### Resume Parsed Data #####\n")
for key in data.keys():
    if data[key] is not None:
        print('{} : {}'.format(key,data[key]))
        result=Tk() # result.geometry('700x550')
        result.overrideredirect(False)

        result.geometry("{}x{}+0+0".format(result.winfo_screen
        width(), result.winfo_screenheight())) result.configure(background='White')
        result.title("Predicted Personality")

        #Title
        titleFont = font.Font(family='Arial', size=40, weight='bold')
        Label(result, text="Result - Personality Prediction", foreground='green', bg='white',
        font=titleFont, pady=10,
        anchor=CENTER).pack(fill=BOTH)

        Label(result, text = str('{} : {}'.format("Name:", aplcnt_name.get())).title(),
        foreground='black', bg='white', anchor='w').pack(fill=BOTH)
        Label(result, text = str('{} : {}'.format("Age:", age)), foreground='black', bg='white',
        anchor='w').pack(fill=BOTH) for key in data.keys(): if data[key] is not None:
        Label(result, text = str('{} :
        {}'.format(check_type(key.title()),check_type(data[key]))),
        foreground='black', bg='white', anchor='w', width=60).pack(fill=BOTH)
        Label(result, text = str("perdicted personality: "+personality).title(), foreground='black',
        bg='white', anchor='w').pack(fill=BOTH)

        quitBtn = Button(result, text="Exit", command =lambda:
        result.destroy()).pack()

```

```
terms_mean = "" # Openness:
```

People who like to learn new things and enjoy new experiences usually score high in openness. Openness includes traits like being insightful and imaginative and having a wide variety of interests.

```
# Conscientiousness:
```

People that have a high degree of conscientiousness are reliable and prompt. Traits include being organised, methodic, and thorough.

```
# Extraversion:
```

Extraversion traits include being; energetic, talkative, and assertive (sometime seen as outspoken by Introverts). Extraverts get their energy and drive from others, while introverts are self-driven get their drive from within themselves.

```
# Agreeableness:
```

As it perhaps sounds, these individuals are warm, friendly, compassionate and cooperative and traits include being kind, affectionate, and sympathetic. In contrast, people with lower levels of agreeableness may be more distant.

```
# Neuroticism:
```

Neuroticism or Emotional Stability relates to degree of negative emotions. People that score high on neuroticism often experience emotional instability and negative emotions.

Characteristics typically include being moody and tense. ""

```
Label(result, text = terms_mean, foreground='green',  
bg='white', anchor='w', justify=LEFT).pack(fill=BOTH)  
result.mainloop()
```

```
def predict_person(): ""Predict Personality""
```

```
# Closing The Previous Window root.withdraw()
```

```
# Creating new window top = Toplevel() top.geometry('700x500')  
top.configure(background='black')
```

```

top.title("Apply For A Job")

#Title
titleFont = font.Font(family='Helvetica', size=20,
weight='bold')
lab=Label(top, text="Personality Prediction",
foreground='red',
pady=10).pack()
bg='black', font=titleFont,
#Job_Form
job_list=('Select Job', '101-Developer at TTC', '102-Chef at Taj', '103-Professor at MIT')
job = StringVar(top)
job.set(job_list[0])
l1=Label(top, text="Applicant Name",
foreground='white', bg='black').place(x=70, y=130) l2=Label(top, text="Age",
foreground='white', bg='black').place(x=70, y=160) l3=Label(top,
text="Gender", foreground='white', bg='black').place(x=70, y=190)
l4=Label(top, text="Upload Resume",
foreground='white', bg='black').place(x=70, y=220) l5=Label(top, text="Enjoy
New Experience or thing(Openness)", foreground='white',
bg='black').place(x=70, y=250)
l6=Label(top, text="How Offen You Feel Negativity(Neuroticism)", foreground='white',
bg='black').place(x=70, y=280) l7=Label(top, text="Wishing to do one's work well and
thoroughly(Conscientiousness)", foreground='white', bg='black').place(x=70, y=310)
l8=Label(top, text="How much would you like work with your peers(Agreeableness)",
foreground='white', bg='black').place(x=70, y=340) l9=Label(top, text="How outgoing and
social interaction you like(Extraversion)", foreground='white',
bg='black').place(x=70, y=370)

sName=Entry(top) sName.place(x=450, y=130, width=160) age=Entry(top)
age.place(x=450, y=160, width=160)

```



```

gender = IntVar()
R1 = Radiobutton(top, text="Male", variable=gender, value=1, padx=7) R1.place(x=450,
y=190)
R2 = Radiobutton(top, text="Female", variable=gender, value=0, padx=3)
R2.place(x=540, y=190) cv=Button(top, text="Select File", command=lambda:
OpenFile(cv)) cv.place(x=450, y=220, width=160) openness=Entry(top)
openness.insert(0,'1-10') openness.place(x=450, y=250, width=160)
neuroticism=Entry(top) neuroticism.insert(0,'1-10') neuroticism.place(x=450, y=280,
width=160) conscientiousness=Entry(top) conscientiousness.insert(0,'1-10')
conscientiousness.place(x=450, y=310, width=160) agreeableness=Entry(top)
agreeableness.insert(0,'1-10') agreeableness.place(x=450, y=340, width=160)
extraversion=Entry(top) extraversion.insert(0,'1-10') extraversion.place(x=450,
y=370, width=160) submitBtn=Button(top, padx=2, pady=0, text="Submit", bd=0,
foreground='white', bg='red', font=(12)) submitBtn.config(command=lambda:
prediction_result(top,sName,loc,(gender.get(),age.get(),open
ness.get(),neuroticism.get(),conscientiousness.get(),agreeabl eness.get(),extraversion.get()))
submitBtn.place(x=350, y=400, width=200)

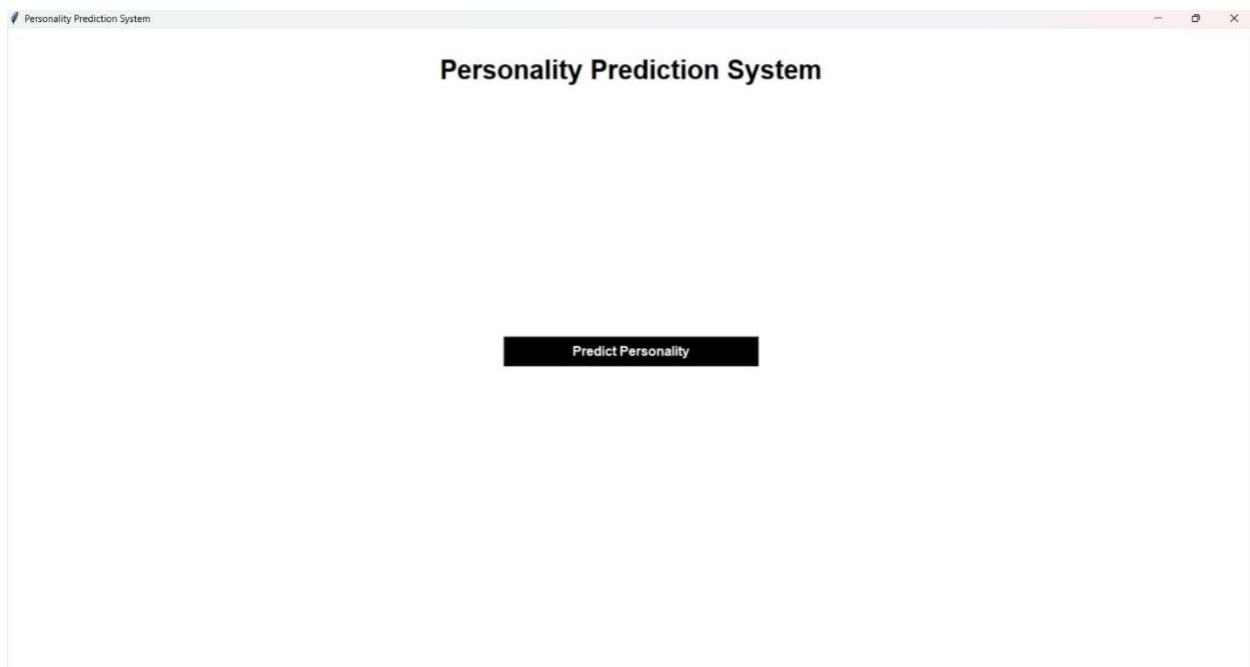
top.mainloop()
def OpenFile(b4): global loc;
name = filedialog.askopenfilename(initialdir="C:/Users/Batman/Do
cuments/Programming/tkinter/",
filetypes
=(("Document", "*.docx*"),("PDF", "*.pdf*"),('All files', '*')),
title = "Choose a file.")
try:
filename=os.path.basename(name)
loc=name except: filename=name loc=name b4.config(text=filename)
return
if __name__ == "__main__": model = train_model()
model.train()

```

```
root = Tk() root.geometry('700x500') root.configure(background='white')
root.title("Personality Prediction System") titleFont =
font.Font(family='Helvetica', size=25, weight='bold') homeBtnFont =
font.Font(size=12, weight='bold') lab=Label(root, text="Personality Prediction System",
bg='white', font=titleFont, pady=30).pack()
b2=Button(root, padx=4, pady=4, width=30, text="Predict Personality",
bg='black', foreground='white', bd=1, font=homeBtnFont,
command=perdict_person).place(relx=0.5, rely=0.5, anchor=CENTER) root.mainloop()
```

After running the code, it will take us to the login page of personality prediction system.

On landing page, 'Predict Personality' button pops up a new window for taking various inputs from user and submit it prediction model which will predict the personality.



After clicking on, predict personality we'll forwarded to page where we have to fill all the details.

Apply For A Job

## Personality Prediction

Applicant Name

Age

Gender  Male  Female

Upload Resume

Enjoy New Experience or thing(Openness)

How Offen You Feel Negativity(Neuroticism)

Wishing to do one's work well and thoroughly(Conscientiousness)

How much would you like work with your peers(Agreeableness)

How outgoing and social interaction you like(Extraversion)

After filling every details we have to click on submit button and then output will be printed

### OUTPUT 1:

```
##### Candidate Entered Data #####
{'Candidate Name': 'Asmi Mishra', 'CV Location': 'C:/Users/Asus/Desktop/Asmi_Mishra.pdf.docx'} (0, '21', '10', '2', '10', '10', '7')
##### Predicted Personality #####
['dependable']
```

---

### OUTPUT 2:

```
##### Candidate Entered Data #####
{'Candidate Name': 'Shishu Pandey', 'CV Location': 'C:/Users/Asus/Desktop/Shishu.resume.docx'} (1, '20', '10', '10', '10', '10', '10')
##### Predicted Personality #####
['dependable']
```

## Chapter-6

### CONCLUSION AND FUTURE SCOPE

In this paper, we have used various Machine Learning Algorithms such as Logistic Regression, Naive Bayes, Random Forest, SVM and KNN for Personality prediction using CV Analysis. Using pyresparser, spaCy and Phrase Matcher we were able to predict the personalities of various candidates. The results indicate Random Forest has the maximum accuracy of 0.71 however due to lack of available data the accuracy is much lesser than it was anticipated. The proposed system can be used by various companies in order to streamline the recruitment process by considering the personality of potential candidates. Future work can also be done to increase the efficiency and performance of the proposed system in order to predict personality using CV analysis more accurately.

In pattern matching and string matching methods, the regular expression approach played an important role in the feature extraction process to extract important parameters from CV data such as work experience, skills, education background, certifications. The weighted average method was used to score the CV performance based on job title specification.

Artificial Intelligence provides adaptive learning and performs real-time operations on input data. It saves time and human efforts for both the hired employee and the recruiter who is recruiting the employee. The proposed application offers a fair approach to applicant evaluation for a particular job opening. Automated CV analysis can be used in a variety of industries, including IT, law, medicine, government, retail, and health. This project work further can be extended to all file extensions as input CV data in the future.

## REFERENCES

- [1] Sudhir Bagade, Jayashree Rout, Pooja Yede, Personality Evaluation and CV Analysis using Machine Learning Algorithm. Article ID155678,21December 2018
  
- [2] Gayatri Vaidya, Pratima Yadav, Reena Yadav,Prof.Chandana Nighut, “Personality Prediction By Discrete Methodology” ,in Personality traits, IOSR Journal of Engineering (IOSRJEN), ISSN (e): 2250-3021, ISSN (p): 2278- 8719,Volume 14, PP 10-13
  
- [3] Automated CV Processing along with Psychometric Analysis in Job Recruiting Process Firoz Ahmed, Mehrin Anannya 2, Tanvir Rahman 3, Risala Tasin Khan4Institute of Information Technology, May 2015.
  
- [4] Janet E. Squires, Leslie Hayduk, Alison M. Hutchinson, Lisa A Cranley, Mark Gierl, Greta G. Cummings, Peter G. Norton, and Carole A. Estabrooks, “A Protocol for Advanced Psychometric Assessment of Surveys\”, Nursing Research and Practice Volume 2013 (2013), Article ID156782,20 December 2012.
  
- [5] Nathan C. Popkins: The Five-Factor Model: Emergence of a Taxonomic Model for Personality Psychology”. December 2018
  
- [6] M. Kalghatgi, M Ramannavar, and Dr. N. S. Sidnal, Neural Network approach to personality prediction based on the Big-Five Model in IJIRAE, vol2 issue 8, August 2015, pp56-63.
  
- [7] A..Robey, K. Shukla, K. Agarwal, K. Joshi, Professor S.

- [8] Joshi Personality prediction system through CV Analysis, in IRJET vol 6, issue 02, February2019
- [9] Justindhas Y, Sabari Mutheeswari Mohanraj, Shivani R, "A Synoptic Survey on Personality Prediction System using MBTI", 2022 3rd International Conference on Electronics and Sustainable Communication Systems (ICESC), pp.950-955, 2022
- [10] N. &. M. S. Iyer, "Implementation of Secure Hash Algorithm-1 using FPGA," International Journal of Information and Computation Technology, Volume III, Issue 3, pp. 50-60, 2018.
- [11] Tommy Tandra, Hendro, Derwin Suhartono, Rini Wongso, and Yen Lina Prasetio, "Personality Prediction System from Facebook Users", 2nd International Conference on Computer Science and Computational Intelligence 2017, ICCSCI 2017, 13-14 October 2017, Bali, Indonesia, Vol. No. 10.1016/j.procs.2017.10.016
- [12] Kessler, R., Béchet, N., Torres-Moreno, J.-M., Roche, M., and El-Bèze, M., "Job Offer Management: How Improve the Ranking of Candidates", in Foundations of Intelligent Systems, 431-441.doi:10.1007/978-3-642-04125-9\_46
- [13] Shaza D. Rihan, Ahmed Khalid and Saife Eldin F. Osman, "A Performance Comparison of Encryption Algorithms AES and DES", International Journal of Engineering Research and Technology (IJERT) ISSN : 2278-0181 Vol. 4 Issue 12, December-2015.
- [14] Pasquale De Meo, Giovanni Quattrone, Giorgio Terracina, and Domenico Ursino, "An XML-based Multi-Agent System for Supporting Online Recruitment Services", IEEE Transactions on Systems, Man, And Cybernetics, Vol. 37, No. 4, 2017.
- [15] Carolyn Winslow, Xiaoxiao Hu, Seth Kaplan, Yi Li, "Accentuate the Positive: Which

Discrete Positive Emotions Predict Which Work Outcomes?”, *The Psychologist-Manager Journal* – 2017, *The Psychologist-Manager Journal*, 20(2), 74- 89

- [16] Muhammad Fahim Uddin, Soumita Banerjee, Jeongkyu Lee, “Personality Based Recommendation Engine (PBRE)”, 2016 IEEE 17th International Conference on Information Reuse and Integration (IRI), Electronic ISBN: 978-1-5090-3207-5, Print on Demand(PoD) ISBN: 978-1-5090-3208-2
  
- [17] Chamorro-Premuzic, T., & Furnham, A. (2017). Personality and music: Can traits explain how people use music in everyday life? *British Journal of Psychology*, 98, 175-185.
  
- [18] Iftekhar Naim, M. Iftekhar Tanveer, Daniel Gildea and Mohammed (Ehsan) Hoque, “Automated Analysis and Prediction of Job Interview Performance”, *IEEE Transactions on Affective Computing*, Volume: 9 , Issue: 2 , April-June 1, 2018
  
- [19] Muhammad Fahim Uddin, Jeongkyu Lee, “Proposing Stochastic Probability-based Math Model and Algorithms Utilizing Social Networking and Academic Data for Good Fit Students Prediction”, *IEEE International Conference – 2018*, DOI 10.1007/s13278-0170448-z
  
- [20] Vishnu M. Menon and Rahulnath H A, “A Novel Approach to Evaluate and Rank Candidates in a Recruitment Process by Estimating Emotional Intelligence through Social Media Data“, in *Big Five Model*, 2016 International Conference on Next Generation Intelligent Systems (ICNGIS), 978-1-5090- 0870-4/16 2018