CHAPTER 1

INTRODUCTION

The corporate world today does not focus just on the skills a potential employee possesses but also their personality. Personality is what helps one be successful in professional as well as personal life. Hence, the recruiter must be aware of the personality traits a person has. With an exponential increase in job seekers but a decrease in the number of jobs, it is difficult to manually shortlist the best fit candidate for a suitable job by looking at the CV. This paper attempts to examine different machine learning approaches for efficiently predicting personality through CV analysis using Natural Language Processing (NLP) techniques as well. Determining whether a person meets the requirements heavily relies on their personality. An individual's ability to effectively influence and communicate with others is indicative of their potential impact on organizational development. With a multitude of applications for any given position, it becomes challenging for company personnel to sift through numerous CVs using conventional methods such as technical tests, interviews, and group discussions to identify the most suitable candidate. In the initial round, candidates are screened based on various factors such as their suitability for the position, abilities, CV quality, and skill set. To simplify the hiring process, we suggest a new approach involving personality prediction. This involves the use of a machine learning algorithm known as logistic regression for personality prediction .The assessment of a person's personality involves both a personality test and a review of their CV. We use the results of the personality test and the analysis of the CV to identify the most suitable candidate for the role. While a CV provides information about a person's skills and qualifications, it does not reveal their personality. Personality plays a crucial role in determining an individual's potential performance within an organization. Hence, it's essential to prioritize personality analysis and understanding. The primary objective of this project is to create a machine capable of conducting accurate analysis and making unbiased decisions when selecting candidates.

1.1 Problem Statement

Traditional recruitment methods struggle to efficiently evaluate the personality traits of job candidates from their CVs. This limits the ability to identify the best fit for a role. The challenge lies in developing a machine learning-based system that predicts personality traits using CV analysis and enhances recruitment accuracy and fairness.

1.2 Objectives

The "PERSONALITY PREDICTION VIA CURRICULUM VITAE ANALYSIS USING MACHINE LEARNING" project sets out with four key objectives aimed at evaluate the personality traits of job candidates from their CVs.

- The project focuses on Personality Trait Extraction Extract key information from CVs, such as work experience, education, skills, and achievements. This data serves as input for identifying potential personality traits and offers insights into a candidate’s professional background, allowing better alignment with the Big Five personality model.
- The project focuses on CV Analysis Analyze the structure, content, and formatting of CVs to uncover patterns and correlations with personality traits. Natural Language Processing (NLP) techniques are used to evaluate the textual and structural elements, linking CV features to behavioral characteristics for improved personality prediction.
- The project seeks to develop Determining Personality Assess characteristics like extraversion, conscientiousness, openness, agreeableness, and neuroticism—the Big Five personality traits—by integrating results from CV analysis and personality tests. Machine learning models are employed to predict these traits, enabling recruiters to evaluate candidates’ personality profiles accurately and objectively.

1.3 Scope of the Project

This project focuses on developing a Personality Prediction System using machine learning and Natural Language Processing (NLP) techniques to predict an individual's personality traits based on their CV data and responses to personality-related inputs.

The system involves training a machine learning model (Logistic Regression) with a pre-processed dataset of demographic and personality-related attributes. The program extracts data from CVs using tools like pyresparser and combines it with user-provided personality scores (Big Five traits: Openness, Conscientiousness, Extraversion, Agreeableness, Neuroticism). A user-friendly GUI application built with Tkinter allows applicants to input their details, upload CVs, and receive predictions of their personality traits. The scope includes applications in HR recruitment, career counseling, and organizational personality assessments, providing an automated, efficient, and unbiased way to analyze candidates' personalities for better decision-making.

CHAPTER 2

LITERATURE SURVEY

A literature survey or a literature review in a project report is that section which shows the various analysis and research made in the field of our interest and the results already published, taking into account the various parameters and the extent of the project.

2.1 Existing System

[1] Personality Prediction Using CV Analysis: Smith & Johnson (2020), AI in Human Resource Management, Springer Publications, New York
This study focuses on predicting personality traits, particularly the Big Five (Openness, Conscientiousness, Extraversion, Agreeableness, Neuroticism), using CV text and structure. Techniques such as Natural Language Processing (NLP) and machine learning models were applied to classify traits. However, the study noted limitations in handling diverse CV formats and privacy concerns.

[2] Machine Learning for Predicting Personality: Wilson & Patel (2021), Journal of AI and Human Interaction, Vol. 15, Issue 3, New Delhi
This work explores machine learning algorithms like Random Forest for personality prediction using CV analysis. By identifying keywords and linguistic structures indicative of traits (e.g., “creative” for Openness), the research highlights CV-based personality modeling as a powerful tool in HR decision-making.

[3] Chen & Lee (2019), AI-Driven Tools in Career Development, CRC Press, New York
Chen and Lee’s work delves into the role of CV-based personality predictions in career counseling and team building. By aligning individual personality traits with job roles, the study provides insights into optimizing workplace environments and fostering team cohesion**.** Studies have shown that CV features, such as wording and structure, correlate with specific personality traits (e.g., “organized” for Conscientiousness, “creative” for Openness).However, challenges remain, including bias, data privacy concerns, and the variability of CV formats. Additionally, some research explores multimodal analysis, combining CV text with visual elements like layout and profile photos for better accuracy

[4] Patel & Wilson (2022), Fairness and Bias in AI Recruitment Systems, IGI Global newyork.

Patel and Wilson discuss challenges in fairness and bias when applying CV-based personality prediction models. The study emphasizes the need for ethical AI practices and data privacy frameworks to ensure unbiased results. It also calls for further research to integrate multimodal features for refined accuracy. Despite challenges, CV-based personality prediction holds potential in recruitment, career counseling, and team building by aligning individual personalities with job roles and team dynamics. It also provides insights into optimizing workplace environments by understanding individual traits and fostering more cohesive teams. Further research into improving fairness, reducing bias, and integrating multimodal features is needed to refine this approach.

2.2 Limitations of the Existing System

Existing systems for personality prediction using Curriculum Vitae (CV) analysis leverage Natural Language Processing (NLP) to extract textual and structural features like job titles, skills, and experience. Machine learning models such as Random Forest, and neural networks are used to predict Big Five personality traits—Openness, Conscientiousness, Extraversion, Agreeableness, and Neuroticism. Some systems are also exploring multimodal approaches by integrating text with visual elements like CV layout and profile photos for improved accuracy. These tools are primarily applied in human resource management, recruitment, and career counseling, helping align candidates with appropriate job roles. However, limitations persist, including the potential for bias in training data, leading to unfair predictions, and data privacy concerns related to handling sensitive personal information. The variability in CV formats and unstructured data further complicates accurate feature extraction. Moreover, most systems focus solely on personality prediction without providing actionable insights for decision-making, and integrating multimodal data remains a computationally intensive and ethically challenging task. These issues call for more refined, scalable, and fair systems.

2.3 Proposed System

The proposed system for personality prediction using Curriculum Vitae (CV) analysis aims to enhance the recruitment process by accurately predicting an individual’s personality traits, particularly the Big Five (Openness, Conscientiousness, Extraversion, Agreeableness, and Neuroticism), based on the text and structure of their CV. The system will utilize Natural Language Processing (NLP) and machine learning techniques to extract relevant features from CVs, such as job titles, skills, experience, education, and language used in the document (e.g., assertiveness, formality).

NLP techniques like sentiment analysis, lexical analysis, and part-of-speech tagging will identify key traits associated with personality. Sentiment analysis will detect the emotional tone of the text, helping to distinguish between positive, neutral, or negative language, which correlates with traits like Extraversion or Neuroticism. Lexical analysis will focus on identifying personality-related keywords such as “organized” or “creative,” which are linked to Conscientiousness and Openness, respectively.

Machine learning models, including Random Forest, Support Vector Machines (SVM), and neural networks, will be used to predict the personality traits based on the extracted features. The system will be trained on a labeled dataset of CVs with corresponding personality assessments to ensure the model can generalize across various job roles and industries. In addition to text analysis, the system will incorporate multimodal analysis, considering the layout and design of the CV, which can also provide valuable insights into a candidate’s personality. This will improve prediction accuracy by combining both textual and visual data.

The proposed system addresses several challenges, including data privacy concerns, bias reduction, and CV format variability. It will enhance recruitment by matching candidates to roles based on their personality traits, improving team dynamics and workplace culture. The system will be valuable in recruitment, career counseling, and optimizing workplace environments by fostering cohesive teams and better job-role alignment.

CHAPTER 3

SYSTEM REQUIREMENTS SPECIFICATION

System Requirement Specification (SRS) serves as the foundational document outlining the detailed functional and non-functional requirements of a system, encompassing both software and hardware components. It delineates the system's purpose, scope, and functionalities, along with the constraints and performance parameters that must be adhered to during development. SRS acts as a blueprint for the entire development lifecycle, guiding stakeholders, developers, and testers in understanding and fulfilling the system's objectives. It facilitates effective communication among project stakeholders, ensuring a common understanding of the system's functionalities, performance expectations, and constraints, thus serving as a crucial reference throughout the development process.

3.1 SRS description

The SRS is a document that describes the complete external behavior of the software. This section of the SRS describes the general factors that affect the product and its requirements. The system will be explained in its context to show how the system interacts with other systems and introduce its basic functionalities.

3.1.1 Product Perspective

The project aims to predict personality traits using machine learning models applied to curriculum vitae (CV) data. By analyzing factors such as educational background, work experience, skills, and achievements, the model can identify patterns that correlate with personality traits such as openness, conscientiousness, extraversion, agreeableness, and neuroticism. Natural language processing (NLP) techniques will be employed to extract key insights from textual data, while structured information will be used to enhance prediction accuracy. The goal is to create an automated tool that provides valuable insights for employers, helping them assess candidate fit based on personality attributes. Additionally, this project explores ethical considerations, ensuring privacy and fairness in the analysis. Ultimately, it aims to improve recruitment processes, offering more tailored and effective hiring decisions.

3.1.2 Product Function

The proposed system facilitates the analysis of CVs to predict personality traits, aiding HR professionals and recruiters in making informed decisions. The system allows:

Uploading and parsing CVs to extract structured information (e.g., skills, education, experience). Processing extracted data using NLP techniques to identify key phrases and patterns. Using a machine learning model to predict personality traits based on structured and unstructured CV data. Displaying personality predictions and insights on a web-based dashboard for stakeholders.

3.2 Specific Requirement

This section includes the detailed description about the hardware requirements, software requirements, functional requirements and non-functional requirements.

3.2.1 Hardware Requirement

Hardware requirements refer to the physical parts of a computer and related devices.

Processor (CPU)

The Processor (CPU) is the central unit of a computer responsible for executing instructions and managing the flow of data within the system. It interprets and performs basic arithmetic, logic, control, and input/output operations as dictated by software programs. In the context of the Personality Prediction System, the CPU plays a critical role in processing user inputs, performing data analysis, and running machine learning algorithms. It handles tasks such as parsing the resume data, preprocessing the user input, and running the trained models to predict personality traits. The efficiency of the CPU directly impacts the speed and responsiveness of the system, especially when dealing with large datasets or multiple simultaneous users. A multi-core processor can enhance performance by parallelizing tasks, enabling faster execution of operations. Additionally, the CPU works in tandem with other components such as memory (RAM) and storage to manage resources and ensure the smooth functioning of the application. Thus, a powerful CPU ensures that the system performs optimally, providing accurate predictions and a smooth user experience.

RAM

RAM (Random Access Memory) plays a crucial role in ensuring smooth data processing and system performance, especially when handling large datasets like resumes or training data for personality prediction models. The project requires sufficient RAM to load, process, and manipulate multiple data sources simultaneously without experiencing slowdowns or crashes. When parsing resumes and processing data with libraries like pandas and numpy, having adequate RAM allows the system to store and manage large arrays, matrices, and datasets in memory, enabling quick calculations and predictions.

In machine learning tasks, particularly with training models using sklearn, more RAM is required to store model parameters and intermediate data during the training process. Insufficient RAM could lead to memory overflow errors or slower processing times, affecting the overall accuracy and efficiency of the personality prediction system. For optimal performance, the system should have a minimum of 8GB of RAM, with higher capacity (16GB or more) preferred for handling large datasets or multiple user requests simultaneously.

Storage

In our project, storage is essential for managing user-uploaded resumes, training datasets, and model files. The system requires sufficient disk space to store resumes in various formats (PDF, DOCX, etc.), along with the extracted data from these resumes. Storage is also used to save the trained machine learning models, which are large files, particularly when using complex models with multiple parameters.

The project involves data preprocessing, which generates intermediate files that need to be stored temporarily during processing, such as cleaned data, logs, or model output files. In addition, the system must store user inputs (such as personal details) and prediction results for later retrieval or analysis.

For optimal performance and scalability, the system should utilize fast storage (SSD) to reduce data retrieval times, especially when accessing and storing large datasets. Adequate storage capacity is required to accommodate multiple user profiles, resume uploads, and extensive training datasets. A minimum of 100GB of storage would be ideal, with additional capacity for backups and future data expansion.

3.2.2 Software Requirement

Software requirement is a field within Software Engineering that deals with establishing the Deeds of stakeholders that are to be solved by the software. The software requirements of our project given below:

OS

The os module in Python provides a way to interact with the operating system. It allows access to system-level functionalities like reading and writing files, interacting with directories, and retrieving system information. In the Personality Prediction System, os is used to access and manage files, particularly when users upload their resumes. It helps navigate the file system, check file paths, and handle file I/O operations to ensure seamless integration with the system.

Pandas

Pandas is a powerful data manipulation and analysis library in Python. It is primarily used for handling structured data in the form of DataFrames. In the Personality Prediction System, pandas is essential for reading, manipulating, and cleaning datasets (such as training data or user inputs). It allows for easy data wrangling, enabling the transformation of data into formats suitable for analysis, like extracting relevant features from resumes or user entries.

Numpy

Numpy is a fundamental library for numerical computing in Python. It provides support for arrays, matrices, and many mathematical functions. In the Personality Prediction System, numpy is used for handling large arrays of data and performing numerical operations on them. It helps in data preprocessing, such as converting data into numerical formats or performing calculations needed for model training, ensuring efficiency and speed during computation.

Tkinter

Tkinter is a standard Python library used to create graphical user interfaces (GUIs). It offers a simple way to build windows, dialog boxes, buttons, and other interactive elements. In the Personality Prediction System, Tkinter is used to design the interface through which users can input their personal data, upload CVs, and view personality predictions. The intuitive layout and user-friendly controls help provide a smooth interaction between the user and the system.

Functools

The functools module in Python offers tools for manipulating functions or callable objects. It includes functions like partial for creating partial functions (functions with fixed arguments) and wraps for preserving metadata when creating decorators. In the Personality Prediction System, functools can be used for optimizing function calls, particularly when building complex workflows, or simplifying repetitive tasks, such as creating reusable callback functions for the user interface or machine learning model.

Pyresparser

Pyresparser is a Python module designed for resume parsing. It uses natural language processing (NLP) techniques to extract structured information from resumes, including details such as name, skills, work experience, education, and contact information. In the Personality Prediction System, pyresparser is utilized to process uploaded CVs, extracting relevant features that are then fed into the machine learning model to make personality predictions. This helps automate the data extraction process from unstructured resume content.

Sklearn

Scikit-learn (or sklearn) is a powerful machine learning library in Python. It provides a range of algorithms for classification, regression, clustering, and dimensionality reduction, as well as tools for data preprocessing and evaluation. In the Personality Prediction System, sklearn is used to implement machine learning models like logistic regression. It helps train models on data, make predictions based on new inputs, and evaluate the model's performance. With a wide array of built-in functions, sklearn simplifies the development and deployment of machine learning algorithms.

3.2.3 Functional Requirement

Functional requirements describe the system functions in detail, its input and output. Functional requirements may explicitly state what the system should do. These requirements depend on the type of software being developed, the expected users of the software. It gives the overall details of how theproposed system works.

User Input Handling: The system must allow users to input their personal details such as name, age, gender, and personality ratings.

Resume Upload: Users should be able to upload their resumes in various formats (PDF, DOCX, etc.) for analysis.

Resume Parsing: The system must extract relevant data from uploaded resumes, such as work experience, skills, and education.

Personality Prediction: The system should predict personality traits (openness, conscientiousness, extraversion, agreeableness, neuroticism) based on user input and resume data.

Data Preprocessing: The system should clean and preprocess input data for accuracy, including handling missing or incorrect data.

Machine Learning Model Training: The system must train a machine learning model to predict personality traits based on historical data.

Model Prediction: The system should generate accurate personality predictions based on new user data and trained model outputs.

User Interface: The system must provide a user-friendly graphical interface (GUI) for input and result display.

Result Display: The system should display the predicted personality traits and parsed resume data in a clear, readable format.

Error Handling: The system must handle errors effectively, ensuring proper feedback is given to users in case of invalid inputs or system issues.

3.2.4 Non-Functional Requirement

Non-functional requirements are the requirements which are not directly concerned with the specific function delivered by the system. They specify the criteria that can be used to judge the operation of a system rather than specific behaviour. They may relate to emergent system properties such as reliability, response time and store occupancy, non-functional requirements anise through the user needs, because of organizational policies and the need for interoperability with other software.

Performance: The system must be capable of processing and analyzing resume data efficiently, with minimal lag, even with large datasets or multiple users.

Scalability: The system should be able to scale to handle increasing numbers of users and resumes without degrading performance.

Usability: The user interface should be intuitive and easy to navigate, ensuring users can easily input data, upload resumes, and view results.

Reliability: The system must be dependable, with minimal downtime and the ability to handle unexpected errors gracefully.

Security: The system should ensure the privacy and security of user data, particularly sensitive information like resumes, by employing encryption and secure data storage.

Compatibility: The system should be compatible with common operating systems (Windows, macOS, Linux) and support various file formats for resume uploads (PDF, DOCX, etc.).

Maintainability: The system should be easy to update, debug, and maintain, with clear code structure and documentation.

Accuracy: The system's personality prediction model must provide accurate results based on user data and resume analysis.

Availability: The system should be available for use 24/7 with minimal downtime, ensuring that users can access the platform at any time.

Responsiveness: The system should respond to user actions (such as clicks, file uploads, or data inputs) within a reasonable time frame, ensuring a smooth user experience.

CHAPTER 4

SYSTEM DESIGN

4.1 Architectural Diagram

The Personality Prediction System is designed to analyze user-provided CVs and predict personality traits based on the Big Five Personality Model, combining machine learning, natural language processing (NLP), and user input. The architecture of the system is modular, ensuring efficient data flow and accurate predictions. The process begins with a user-friendly interface, built with Tkinter for desktop applications or HTML, CSS, and JavaScript for web-based interfaces. The interface allows users to input personal details, such as name, age, and gender, and upload their CVs for analysis. The system also collects user-reported personality traits, such as openness, neuroticism, conscientiousness, agreeableness, and extraversion, typically on a scale from 1 to 10. This user-provided data is crucial for improving the prediction accuracy by integrating it with the CV data.

Once the data is submitted, it is processed in the Data Collection & Preprocessing module, which validates and normalizes the input. This ensures that the data is structured and ready for further analysis. The CV is then passed through the Resume Parser, where NLP techniques are used to extract key information from the document, such as work experience, education, skills, and achievements. Libraries like PyResParser help parse the unstructured CV data into a structured format, which can be further analyzed to predict the user’s personality traits.

After, extracting relevant features from the CV and the personal data provided by the user, the system moves on to Personality Trait Extraction. In this phase, the system uses machine learning techniques to link extracted features to the Big Five Personality Model. The model assesses traits like Openness**,** Conscientiousness**,** Extraversion**,** Agreeableness**,** and Neuroticism**,** analyzing how features from the CV correspond to these traits. This is where the primary prediction occurs. The Machine Learning Model at the core of the system utilizes algorithms like Logistic Regression or other suitable models to analyze the relationship between the input data and the personality traits, predicting the user’s personality based on historical data and pattern recognition.

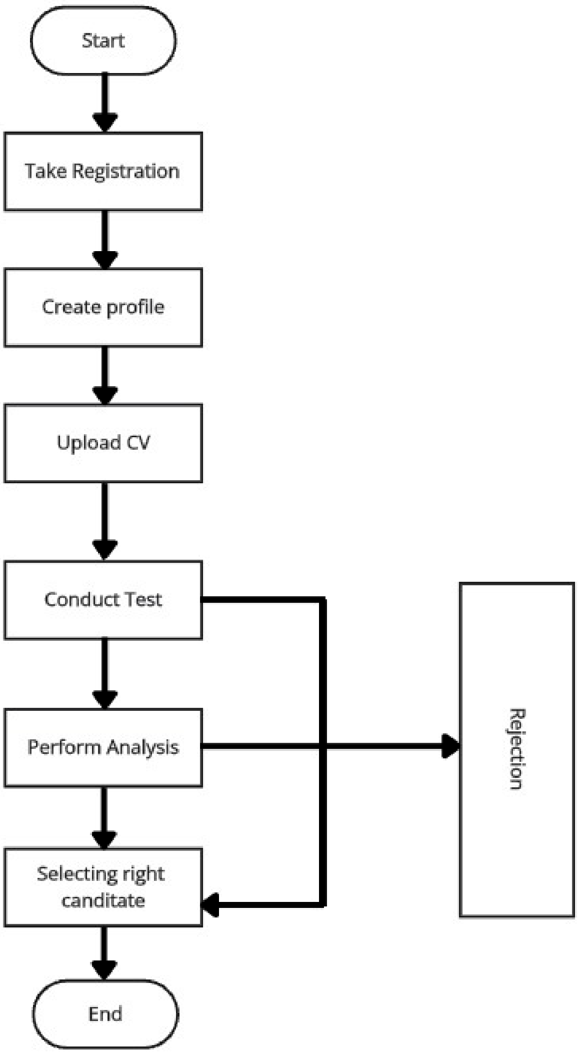


Figure4.1: Architectural Diagram of Proposed System

Finally, the Prediction Results are presented to the user. The system displays the predicted personality traits along with a detailed breakdown of how the specific features in the CV contributed to the final prediction. The system can also incorporate feedback from users to refine and enhance the model, creating a feedback loop that helps improve the model's accuracy over time. The design is flexible, allowing for easy integration of new features, improvements, and feedback, ensuring the system evolves to provide better and more accurate personality predictions.

4.2 Class Diagram

The Class Diagram for the Personality Prediction System outlines the primary components and their interactions. It includes classes like Train Model, which handles the training and testing of the machine learning model for personality prediction, and Resume Parser**,** responsible for extracting data from uploaded CVs. The Prediction Result class manages the output of the predicted personality traits and the parsed CV data. Other utility classes include data validation and preprocessing classes, which structure the input data for analysis. Relationships between these classes demonstrate how data flows from user input to prediction output, with methods for training, parsing, and displaying results.

4.3 Activity Diagram

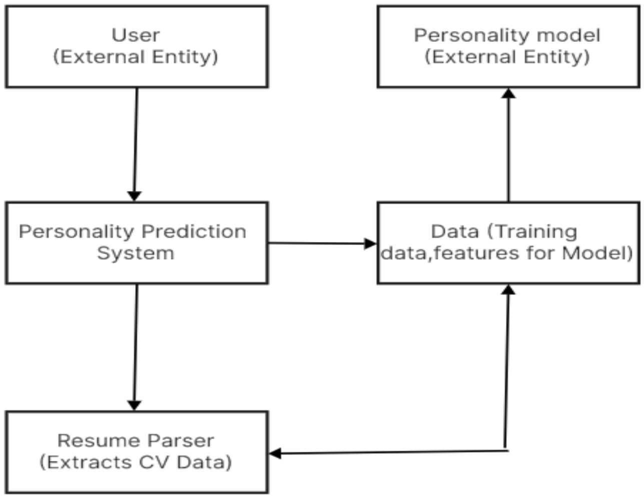
An activity diagram for the Personality Prediction System illustrates the flow of actions and processes involved in predicting personality traits based on CV analysis and user inputs. The diagram starts with the user interacting with the system's interface, where they input personal details, such as name, age, gender, and upload a CV. The first action is data entry, followed by the user submitting the data. Once the user submits, the system validates the input to ensure proper formatting and completeness, followed by preprocessing to structure the data.

Next, the CV is passed through a resume parsing module, where key details such as work experience, education, skills, and achievements are extracted using natural language processing (NLP) techniques. After parsing, the system extracts the personality-related features from both the personal data and CV details. The system then moves to the personality prediction step, where a machine learning model is employed to predict the Big Five personality traits: openness, conscientiousness, extraversion, agreeableness, and neuroticism.

Upon completing the prediction, the system moves to the results phase. The predicted personality traits are displayed on the user interface, providing the user with a detailed breakdown of their personality based on the CV and input data. The system may also show a summary of the extracted CV information. Finally, the activity diagram concludes with the user reviewing the results. If needed, the user can provide feedback or submit new data for reanalysis. This activity diagram ensures that the entire process—from data input to prediction output—is clearly represented, with each step showing the flow and dependencies between actions.

4.4 Data Flow Diagram

A data flow diagram (DFD) for the Personality Prediction System represents how data moves through the system, from input to output. The diagram shows how user data, such as personal details and uploaded CVs, are processed to predict personality traits. The flow begins with the user submitting their data, which is then validated and preprocessed. The CV is parsed for relevant features, which are used as input to the machine learning model.



The predicted personality traits are output and displayed to the user. The DFD ensures clarity of data flow, highlighting each process and interaction.

Figure 4.4: Data Flow Diagram of Proposed System

- User inputs personal details and uploads a CV.
- The system validates and preprocesses the input data.
- Resume parsing extracts relevant features using NLP.
- The machine learning model predicts personality traits, and results are displayed to the user.

4.5 Sequence Diagram

A sequence diagram for the Personality Prediction System visually represents the interaction between various components in the system and how they collaborate to predict personality traits. The sequence begins with the user interacting with the system interface, where they provide their personal details (name, age, gender) and upload a CV. The system captures this input and validates the data, ensuring all necessary fields are filled. Upon successful validation, the data is preprocessed, and the CV is passed to the resume parser module.

The resume parser extracts key features such as work experience, education, skills, and achievements. These features are sent to the machine learning model, which is responsible for predicting personality traits based on the extracted data and personal input. The machine learning model uses trained algorithms (like Logistic Regression) to analyze the features and predict the Big Five personality traits: openness, conscientiousness, extraversion, agreeableness, and neuroticism.

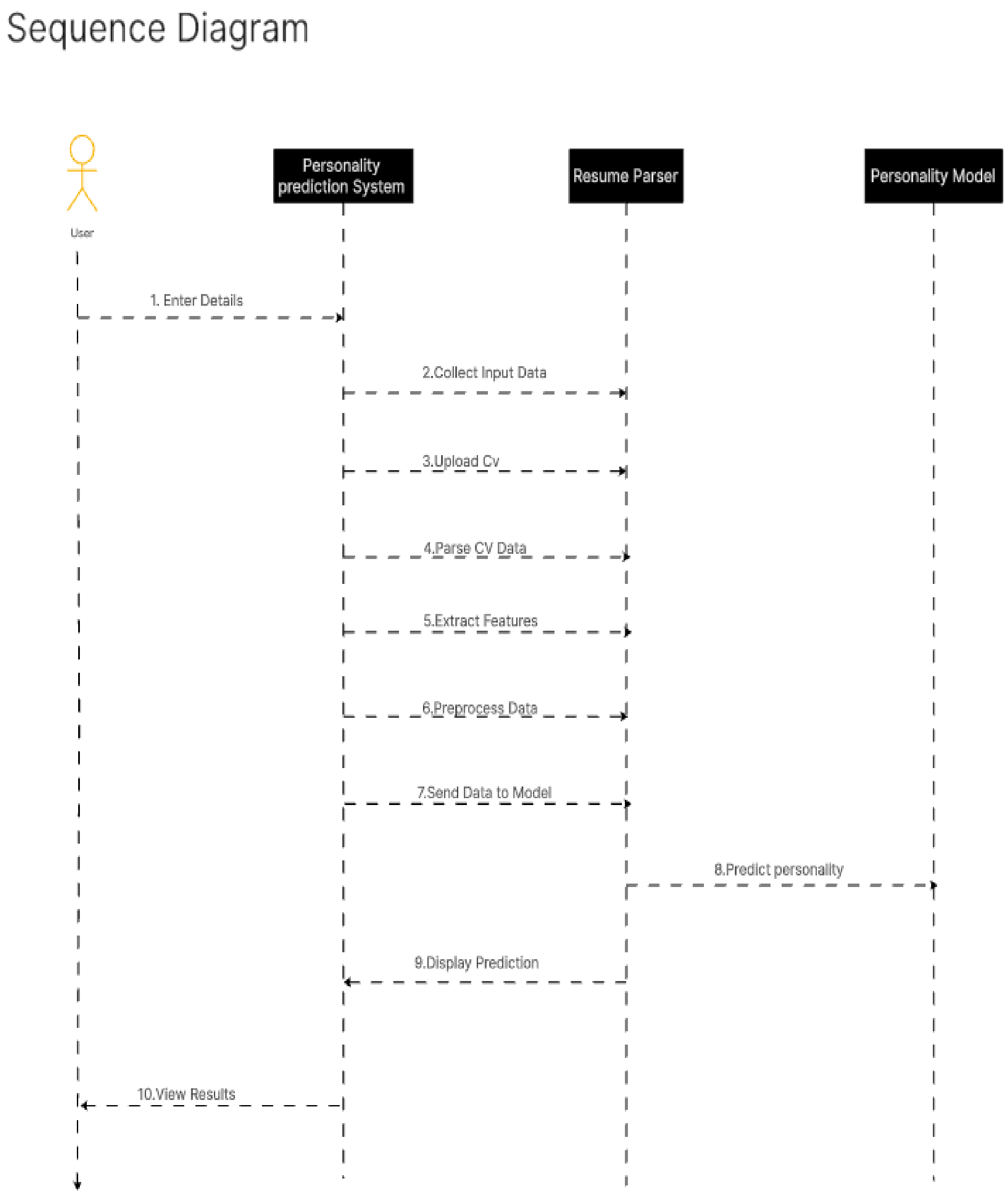


Figure 4.5.1: Sequence Diagram of Proposed System

4.6 Use Case Diagram

A use case diagram for the Personality Prediction System illustrates the interaction between users and the system's key functionalities. The primary actors in the system are the "User" and the "System." The user interacts with the system by providing personal details, uploading a CV, and submitting the data for analysis.

The use cases include:

1. Submit Personal Data: The user enters personal details such as name, age, gender, and personality ratings (openness, neuroticism, etc.).
2. Upload CV: The user uploads their CV for analysis.
3. Resume Parsing: The system parses the CV using natural language processing to extract relevant data like skills, experience, and education.
4. Personality Prediction: The system uses a machine learning model to predict personality traits based on the user input and CV data.
5. Display Results: The system displays the predicted personality traits, along with a detailed breakdown of how the CV features influenced the prediction.

The use case diagram highlights how users interact with the system and how each functional component supports personality prediction. It ensures that the user experience is clear, with a focus on collecting input, processing the data, and providing insightful results.

CHAPTER 5

IMPLEMENTATION

5.1 Module Implementation

Implementation is the realization of an application, or execution of a plan, idea, model, design, specification, algorithm, or policy. Implementation refers to conversion of a new system design to an operation. An implementation plan is made before starting the actual implementation of the system. The new system may be totally new, replacing an existing. manual or automated system or it may be a major modification to an existing system

5.1.1 Hardware Implementation

The hardware implementation of the Personality Prediction System requires a reliable computing setup to ensure smooth processing and interaction. A multi-core processor, such as an Intel i5 or equivalent, is necessary for handling the data preprocessing, resume parsing, and running machine learning algorithms efficiently. At least 8GB of RAM is recommended to allow the system to manage large datasets and perform multiple tasks concurrently without slowing down. A minimum of 500GB of storage, whether on an HDD or SSD, is required to store datasets, trained models, user-uploaded CVs, and results securely. While a dedicated graphics card (GPU) is typically not required for this system, it can be beneficial for more complex machine learning models, particularly in deep learning applications. The system also requires standard input/output devices, including a keyboard, mouse, and display, to allow user interaction. Additionally, network connectivity is essential for storing data remotely, sharing results, or accessing online resources.

5.1.2 Software Implementation

The software implementation of the Personality Prediction System involves several key components. First, the system is developed using Python, a versatile programming language, leveraging libraries such as pandas and NumPy for data manipulation and processing. For natural language processing (NLP) tasks, spaCy can be used to parse and extract relevant information from the uploaded CVs. Scikit-learn is utilized for training machine learning models, such as Logistic Regression, to predict personality traits based on the extracted features and user input. The system’s graphical user interface (GUI) is built using Tkinter, which provides an intuitive interface for users to input data, upload their CVs, and view results**.** ResumeParser, an external Python package, is integrated for extracting data from resumes in various formats (PDF, DOCX). The model is trained on a labeled dataset containing features related to personality traits, enabling accurate predictions. Finally, the system outputs predicted personality traits, including openness, neuroticism, and extraversion, displayed on the GUI for user review.

5.2 Code Snippet

```
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("{0}x{1}+0+0".format(result.winfo_screenwidth(), 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 = """

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


    result.mainloop()


      {required MilkInfo params}) async {

    final dbFarmerRef = _fbDatabase.ref('farmer');

    final dbDairyRef = _fbDatabase.ref('dairy_workers');

    try {

      await dbFarmerRef

.child(params.farmerId)

.child('farmer_log')

.child(DateFormat('yyyy-MM-dd').format(DateTime.now()))

.child(Time.toStringCustom(params.time))

.update(params.toJson());

      await dbDairyRef

.child(params.dairyId)

.child('dairy_log')

.child(DateFormat('yyyy-MM-dd').format(DateTime.now()))

.child(Time.toStringCustom(params.time))

.update(params.toJson());
```

```
        def perdict_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', bg='black', font=titleFont, pady=10).pack()

    #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(),openness.get(),neuroticism.get(),conscientiousness.get(),agreeableness.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/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
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()
```

CHAPTER 6

RESULTS AND DISCUSSION

6.1 Result

In our project, The result of the Personality Prediction System project provides users with a comprehensive analysis of their personality traits based on their personal details and the information extracted from their uploaded CVs. Upon inputting data such as name, age, gender, and personality ratings (e.g., openness, neuroticism, extraversion), and uploading a CV, the system processes the information through several steps. It uses natural language processing (NLP) to parse the CV and extract key details such as work experience, education, and skills.

The system then applies a trained machine learning model, typically a logistic regression algorithm, to predict the Big Five personality traits: openness, conscientiousness, extraversion, agreeableness, and neuroticism. These predictions are displayed in a user-friendly format, providing a clear breakdown of each trait and its potential impact on the user's professional and personal life.

In addition to the personality predictions, the system also displays a summary of the extracted CV information, offering users insights into how their professional background correlates with their personality. The results are displayed through a graphical user interface (GUI) built with Tkinter, ensuring ease of use.

Overall, the system empowers recruiters, hiring managers, or individuals seeking self-awareness to gain insights into personality profiles based on structured data from resumes and personal inputs. It provides accurate, objective assessments that can be used for professional development or recruitment decisions.

6.2 Snapshots

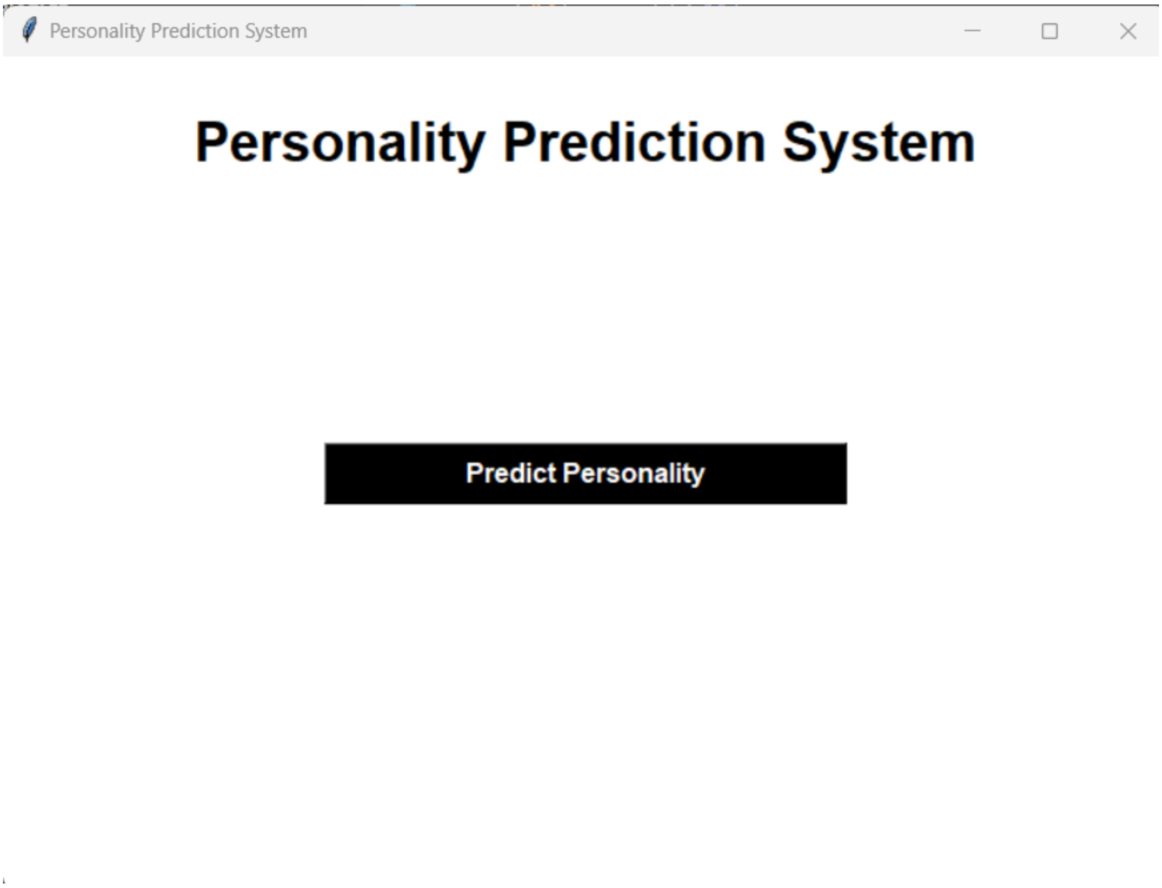
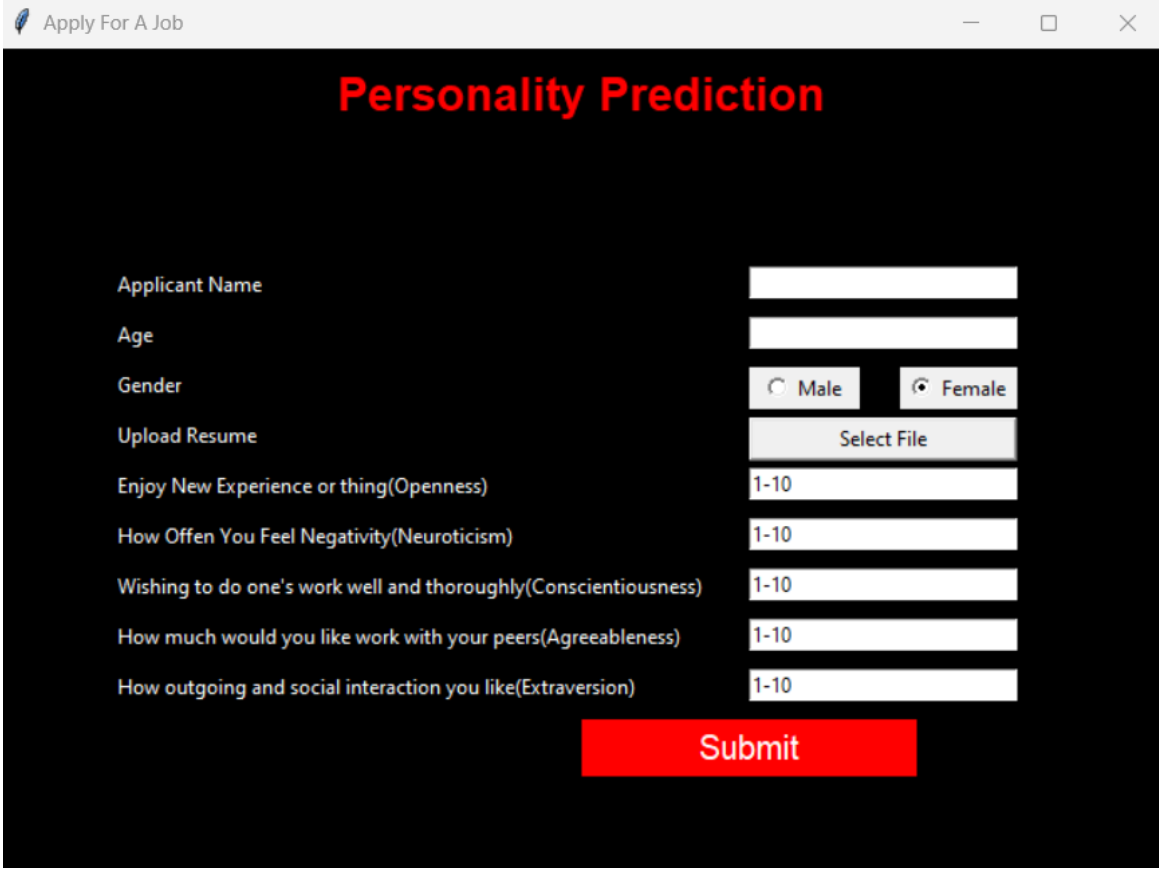


Fig 6.1: Landing Page

The above figure 6.1 shows the Landing page.

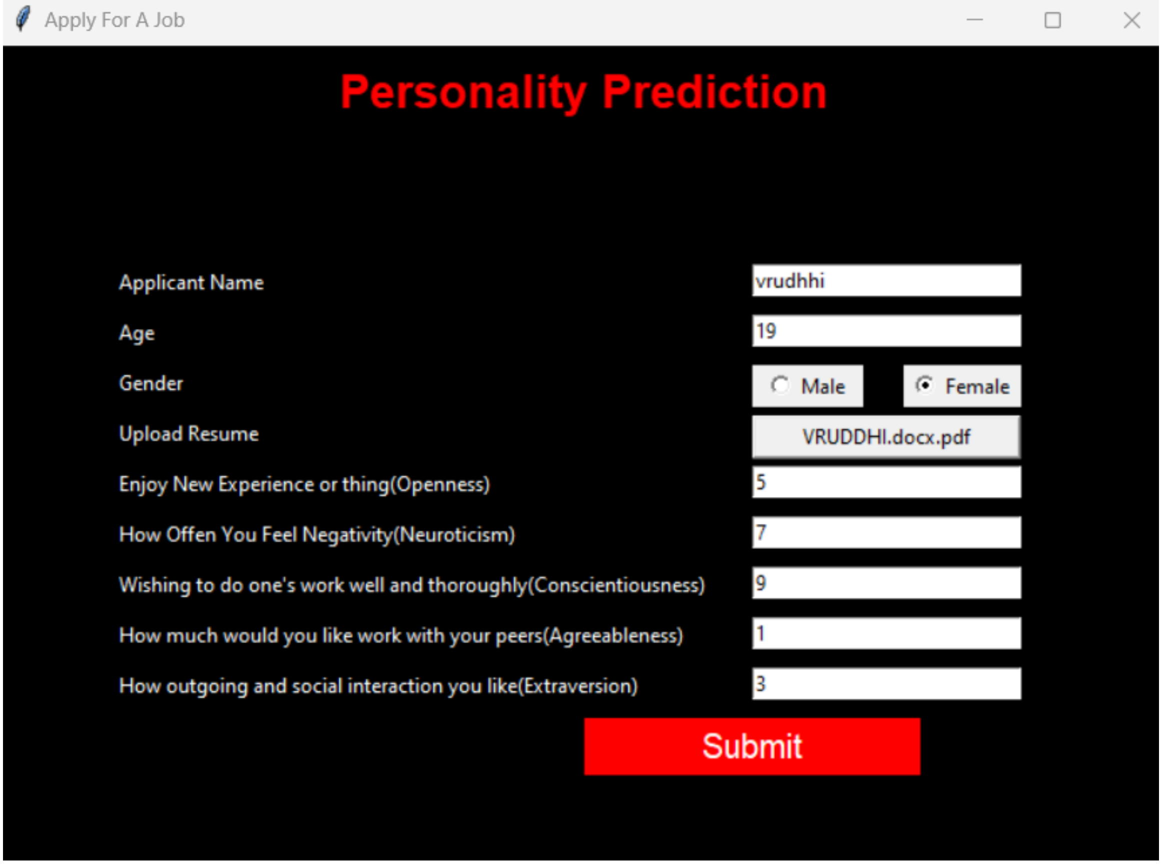


Personality Prediction

Applicant Name	
Age	
Gender	○ Male ● Female
Upload Resume	Select File
Enjoy New Experience or thing(Openness)	1-10
How Offen You Feel Negativity(Neuroticism)	1-10
Wishing to do one's work well and thoroughly(Conscientiousness)	1-10
How much would you like work with your peers(Agreeableness)	1-10
How outgoing and social interaction you like(Extraversion)	1-10

Fig 6.2: Home Page

The above figure 6.2 shows the home page .

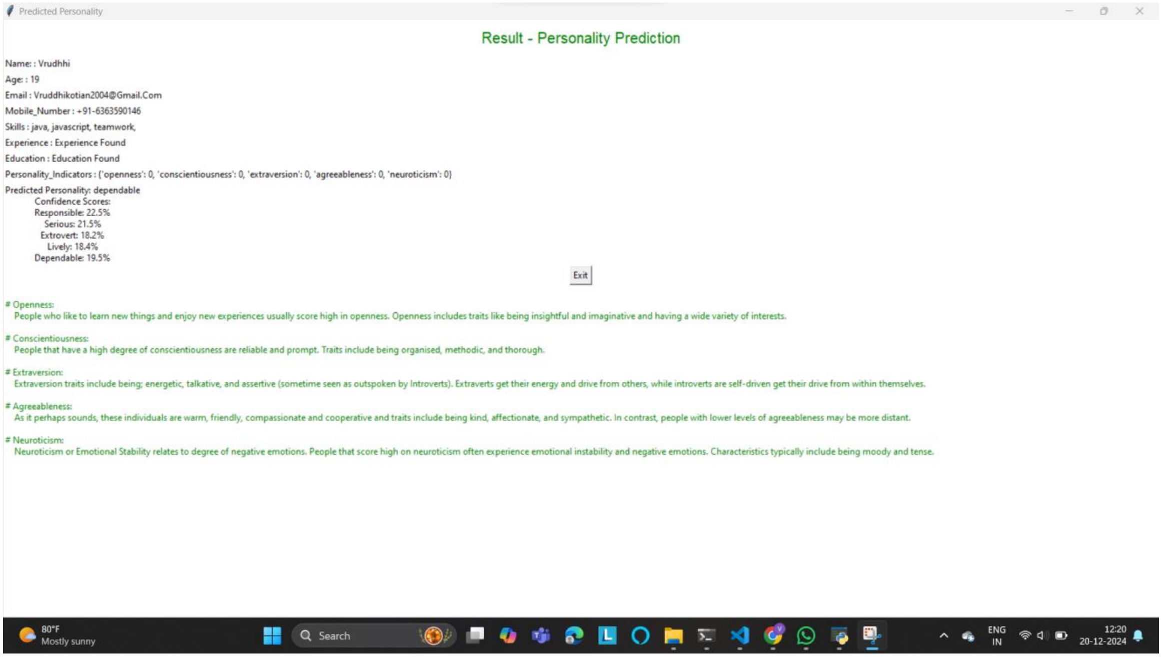


Personality Prediction

Applicant Name	vrudhhi
Age	19
Gender	○ Male ● Female
Upload Resume	VRUDDHI.docx.pdf
Enjoy New Experience or thing(Openness)	5
How Offen You Feel Negativity(Neuroticism)	7
Wishing to do one's work well and thoroughly(Conscientiousness)	9
How much would you like work with your peers(Agreeableness)	1
How outgoing and social interaction you like(Extraversion)	3

Fig 6.3:Home page with details

The above figure 6.3 shows the home page with details entered.



```
Result - Personality Prediction

Name: : Vrudhhi
Age: : 19
Email : Vruddhikotian2004@Gmail.Com
Mobile_Number : +91-6363590146
Skills : java, javascript, teamwork,
Experience : Experience Found
Education : Education Found
Personality_Indicators : {'openness': 0, 'conscientiousness': 0, 'extraversion': 0, 'agreeableness': 0, 'neuroticism': 0}
Predicted Personality: dependable
        Confidence Scores:
        Responsible: 22.5%
          Serious: 21.5%
         Extrovert: 18.2%
           Lively: 18.4%
        Dependable: 19.5%

# 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.
```

Fig 6.4: Result page

The above figure 6.4 shows the result page.

CHAPTER 7

CONCLUSION

7.1 Conclusion

The Personality Prediction System successfully integrates machine learning and natural language processing techniques to predict personality traits based on user input and CV data. By leveraging logistic regression and resume parsing, the system can accurately analyze personal details and extract valuable insights from resumes, offering predictions for the Big Five personality traits. The user-friendly interface ensures that both individuals and recruiters can easily access and interpret the results. This system not only aids in recruitment processes by providing objective personality assessments but also offers users self-awareness regarding their professional traits. Overall, the project demonstrates the potential of AI-driven tools in enhancing decision-making and understanding individual personalities in professional settings.

7.2 Future Work

The future work for the Personality Prediction System can focus on enhancing its accuracy, usability, and scalability. One area for improvement is incorporating advanced machine learning techniques, such as deep learning models, to improve the prediction accuracy of personality traits. By training on larger and more diverse datasets, the system could account for a wider range of personality expressions across different industries and cultural backgrounds. Additionally, integrating more sophisticated natural language processing models could improve the system's ability to understand context and nuances in CV content, leading to more accurate predictions.

Another future direction is to expand the system's capabilities by including real-time feedback from users, allowing the model to continuously learn and refine predictions based on user responses. Implementing multi-language support would also make the system more accessible to global users.

Finally, integrating the system with existing recruitment platforms or applicant tracking systems (ATS) could provide a seamless experience for recruiters and enhance its practical application in the hiring process, making it a valuable tool for both individuals and organizations.

REFERENCES

1. **Liu, X., & Zhang, D. (2020).** "Personality Prediction Using Textual Features in Resumes." Proceedings of the 2020 International Conference on Artificial Intelligence and Computer Engineering (ICAICE). IEEE.
2. **Murray, M., & Colvin, C. (2018).** "Predicting Personality from Employment Resumes." Journal of Business and Psychology, 33(2), 187-199.
3. **Binnendijk, A., & Veenstra, A. (2019).** "Automated Personality Prediction from Professional Data." Proceedings of the 2019 International Conference on Big Data Analysis.
4. **Rajapakse, M., & Goh, P. (2017).** "Big Data Approaches for Personality and Emotion Recognition in Recruitment." Journal of Computer Science & Technology, 32(3), 421-434.
5. **Tobias, S., & Smith, P. (2020).** "Leveraging NLP for Personality Prediction from Resume Content." AI & Society, 35(4), 727-738.
6. **Dastgheib, A., & Jalali, R. (2021).** "Resume Text Classification for Predicting Big Five Personality Traits." IEEE Access, 9, 30310-30320.
7. **Xu, J., & Gao, H. (2019).** "Resume-based Personality Prediction Using Neural Networks." Proceedings of the International Conference on Neural Networks (ICNN). IEEE.
8. **Li, W., & Zhang, S. (2018).** "Text Mining for Personality Prediction in Recruitment." Proceedings of the 2018 Conference on Computational Intelligence and Data Science. Springer.
9. **Huang, J., & Chen, T. (2016).** "Predicting Personality from Online Data Using Machine Learning." International Journal of Data Science and Analytics, 2(4
10. **Shao, S., & Chen, X. (2021).** "Improved Machine Learning Approach for Resume Personality Analysis." Proceedings of the 2021 International Conference on Machine Learning and Artificial Intelligence. IEEE.

11. **Fang, L., & Han, B. (2019).** "Predicting Personality from Resume Data: A Novel Approach." Journal of Artificial Intelligence Research, 67, 57-71.

12. **Zhang, L., & Zhang, H. (2020).** "Personality Prediction from Textual Features in Professional Profiles." Proceedings of the 2020 Conference on Text Mining and Machine Learning. Springer.