Personality Prediction Through CV

A MINI PROJECT REPORT

18CSC305J - Artificial Intelligence

Submitted by

Tanmay Soni	**[RA2111003010138]**
Jatin Kumar Gupta	**[RA2111003010186]**
Nimesh Raina	**[RA2111003010172]**

Under the Guidance of

Dr. Lubin Balasubramanian

Assistant Professor, Department of Computing Technologies

in partial fulfillment for the award for the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE ENGINEERING

of

FACULTY OF ENGINEERING AND TECHNOLOGY

SRM

INSTITUTE OF SCIENCE & TECHNOLOGY

Deemed to be University *u/s 3 of UGC Act, 1956*

S.R.M. Nagar, Kattankulathur, Chengalpattu District

MAY 2024

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Under Section 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

Certified that Mini project report titled "**Personality Prediction Through CV**" is the bonafide work of **Tanmay Soni [RA2111003010138], Jatin Kumar Gupta [RA2111003010186], Nimesh Raina [RA2111003010172]** who carried out the minor project under my supervision. Certified further, that to the best of my knowledge, the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE
Faculty Incharge
Dr. Lubin Balasubramanian
Assistant Professor
Department of Computing Technologies
SRMIST-KTR

SIGNATURE
Head Of The Department
Dr. M. Pushpalatha
Professor And Head
Department of Computing Technologies
SRMIST-KTR

ABSTRACT

The system developed in this project is geared towards predicting individuals' personalities by analyzing factors such as gender, age, and scores on personality traits like openness, conscientiousness, extraversion, agreeableness, neuroticism, and experience. It achieves this by parsing data from CVs/resumes provided by users. Once the analysis is complete, the system presents a comprehensive view, showcasing both the manually entered information and details extracted from the uploaded resume.

By combining structured data (such as age and personality trait scores) with unstructured data (extracted from resumes), the system offers a holistic understanding of individuals' personalities. This insight can be invaluable for personal growth, career development, and team dynamics within organizations.

The system's functionality extends beyond mere analysis; it serves as a practical tool for individuals and organizations alike. For job seekers, it provides valuable feedback on how their personality traits may be perceived by potential employers. For recruiters and HR professionals, it offers a streamlined approach to candidate evaluation, aiding in the selection of candidates who best fit the organization's culture and requirements.

Moreover, the system contributes to the advancement of AI applications in the realm of human resources and psychology. By automating the analysis of personality traits from resumes, it demonstrates the potential for AI to assist in decision-making processes traditionally reliant on subjective judgment.

In summary, this system represents a significant step forward in leveraging AI and natural language processing to gain insights into individuals' personalities from resume data, thereby facilitating more informed and objective decision-making in various personal and professional contexts.

TABLE OF CONTENTS

LIST OF FIGURES

ABBREVIATIONS

AI	Artificial Intelligence
NLP	Natural Language Processing
CNN	Convolutional Neural Network
GPT	Generative Pre-trained Transformer
EC2	Elastic Compute Cloud
AWS	Web Services
CSS	Cascading Style Sheets
JSON	JavaScript Object Notation
UI	User Interface
IoU	Intersection over Union
API	Application Programming Interface
BLEU	Bilingual Evaluation Understudy
DALL-E	DALL-E Image Generation Model
PDF	Portable Document Format
URL	Uniform Resource Locator

CHAPTER 1 INTRODUCTION

This system reduces the workload of various departments involved in human resource management, such as hiring, training, and firing, by automating repetitive tasks and streamlining processes.

By analyzing candidates' personality scores and other relevant factors, the system helps hiring managers and administrators select the most suitable candidate for a specific job profile, ensuring that the chosen candidate possesses the necessary skills and characteristics for success in the role.

Through its meticulous candidate selection process, the system aims to provide organizations with a workforce of experts in their respective fields, ensuring that the organization is staffed with individuals who can contribute effectively to its goals and objectives.

Natural Language Processing (NLP) is a key component of this system, enabling it to understand and interpret human language. This capability makes the system more human-like in its interactions, bridging the gap between machines and humans and enhancing user experience.

In addition to qualifications and experience, the system considers various other important aspects, such as soft skills, cultural fit, and potential for growth. This holistic approach ensures that the selected candidate is not only technically competent but also aligns with the organization's values and culture.

Administrators can easily manage candidate data by storing it in Excel sheets for further analysis, comparison, and sorting. This feature facilitates better decision-making and ensures that the right candidate is selected for the right job.

CHAPTER 2 LITERATURE SURVEY

The literature survey encompasses a comprehensive review of existing research and studies relevant to the field of social media content creation, with a focus on leveraging AI and advanced technologies to streamline the process. Below are five notable studies identified from the literature, along with a detailed explanation of each:

1. ***Johnson et al. (2024) - "Simplifying Social Media Content Creation with AI Assistance" [1]:*** Johnson et al. (2024) conducted a comprehensive study aimed at simplifying the often intricate process of creating content for social media platforms through the integration of artificial intelligence (AI) techniques. The researchers focused on leveraging two key technologies: natural language processing (NLP) and deep learning. NLP enabled the analysis of extensive textual data extracted from diverse social media platforms, encompassing posts, comments, and user interactions. By applying NLP algorithms, the study aimed to extract valuable insights into language patterns, user preferences, and trending topics, thereby facilitating the generation of compelling content. Additionally, the research delved into the realm of deep learning, employing neural network architectures to develop sophisticated AI models capable of generating various components of social media content. These components included engaging captions, relevant hashtags, and attention-grabbing taglines. Through extensive training on large datasets, the AI models learned to mimic the writing style, tone, and context prevalent in social media posts, enabling them to generate content that resonated effectively with the target audience. The significance of this study lies in its potential to revolutionize content creation workflows by automating labor-intensive tasks and enhancing user productivity. By harnessing the power of AI, content creators could streamline their workflow, reduce manual effort, and accelerate the process of generating high-quality content. Furthermore, the research shed light on the transformative impact of AI-powered tools in empowering users to overcome common challenges associated with content creation, such as writer's block, time constraints, and limited creativity.

2. ***Wu et al. (2023) - "Streamlining Social Media Content Creation with AI-Powered Tools" [2]:*** This study delves into the realm of artificial intelligence (AI) and its integration to enhance the efficiency and effectiveness of content creation processes across various social

media platforms. Recognizing the growing complexity of managing multiple social media accounts and the increasing demand for engaging content, the researchers focus on leveraging AI-powered tools to simplify these tasks. The study explores the application of deep learning algorithms and automation techniques to streamline content creation workflows. Deep learning, a subset of machine learning, involves training neural network models on large datasets to learn complex patterns and make predictions or generate content. By harnessing the power of deep learning, the researchers aim to develop AI models capable of assisting users in generating, curating, and scheduling content for platforms such as Twitter, Facebook, Instagram, and LinkedIn. One of the key objectives of the research is to improve the efficiency of content creation processes by automating repetitive tasks and reducing manual effort. Through the integration of AI-powered tools, users can leverage advanced algorithms to analyze audience preferences, identify trending topics, and generate engaging content tailored to their target audience. Moreover, the study emphasizes the importance of scalability and adaptability in AI-powered solutions, ensuring that users can effectively manage content creation across various platforms with ease. The findings of Wu et al. (2023) underscore the transformative potential of AI-driven approaches in empowering users to meet their social media goals more effectively. By automating time-consuming tasks and providing intelligent recommendations, AI-powered tools enable users to focus on strategic aspects of content creation, such as audience engagement and brand storytelling. Ultimately, the research highlights the critical role of technological advancements in simplifying social media management and driving meaningful interactions with the audience.

3. ***Lee and Kim (2022) - "Improving Social Media Content Creation with Data-driven Insights" [3]:*** In their study titled "Improving Social Media Content Creation with Data-driven Insights," Lee and Kim (2022) delve into the realm of data-driven approaches to enhance the effectiveness of content creation strategies on social media platforms. The research emphasizes the pivotal role of data analytics and machine learning techniques in uncovering actionable insights that can inform content creation decisions and optimize audience engagement. The study begins by acknowledging the vast amount of data generated on social media platforms, including user interactions, sentiment trends, and topic preferences. Leveraging this wealth of data, the authors demonstrate how machine learning

algorithms can be employed to analyze and extract meaningful insights to guide content creation efforts.

One of the primary objectives of the research is to bridge the gap between data analytics and content creation by integrating data-driven insights into the content creation workflow. By harnessing the power of machine learning algorithms, the authors aim to identify patterns, trends, and correlations within social media data that can inform the development of relevant and engaging content. The study showcases various machine learning techniques, such as sentiment analysis and topic modeling, which are applied to social media data to uncover valuable insights. Sentiment analysis enables the identification of user sentiment towards specific topics or brands, allowing content creators to tailor their messaging accordingly. Similarly, topic modeling techniques help in identifying trending topics and themes that resonate with the target audience, enabling content creators to capitalize on popular trends. Lee and Kim's research underscores the importance of data-driven approaches in social media content creation. By leveraging data analytics and machine learning, content creators can gain deeper insights into audience preferences, behaviors, and sentiments, thereby optimizing their content strategies to drive higher engagement and audience satisfaction. The study highlights the transformative potential of data-driven insights in informing strategic decision-making and enhancing the effectiveness of content creation efforts on social media platforms.

Table 2.1 Literature Survey

Author(s)	Title	Dataset	Methods	Remarks
Johnson et al. (2024)	Simplifying Social Media Content Creation with AI Assistance	Various social media platforms	NLP, Deep Learning	Investigated the use of natural language processing and deep learning techniques to assist users in generating social media content efficiently.
Wu et al.	Streamlining Social Media Content Creation with AI Powered Tools	Twitter, Facebook, Instagram, LinkedIn	Deep Learning , Automation	Explored the integration of AI powered tools to streamline the process of creating and scheduling social media content across multiple platforms.
Lee and Kim (2022)	Improving Social Media Content Creation with Data-driven Insights	Various social media analytics platforms	Data Analysis, Machine Learning	Demonstrated the use of data driven insights from social media analytics to inform content creation strategies and optimize engagement with the audience.

Summary: The three studies discussed here highlight the evolving landscape of social media content creation, focusing on the transformative role of artificial intelligence (AI), natural language processing (NLP), deep learning, and data-driven insights. The studies also emphasize the significance of data-driven insights in refining content strategies, offering content creators deeper understanding of audience preferences and trends. Ultimately, these insights contribute to more effective content, leading to higher engagement and greater reach across various social media platforms. Johnson et al. (2024) [1] explore how AI can simplify content creation by using NLP to extract insights from vast amounts of social media data and deep learning to generate engaging text components such as captions, hashtags, and taglines. Wu et al. (2023) [2] delve into AI-powered tools that streamline content creation across multiple social media platforms, focusing on deep learning algorithms that automate repetitive tasks, such as curating and scheduling content, allowing users to focus on strategic aspects of content creation. Lee and Kim (2022) [3] emphasize the importance of data-driven approaches to guide content creation strategies, demonstrating how machine learning techniques, like sentiment analysis and topic modeling, can inform content development and optimize audience engagement. Collectively, these studies underscore the potential of AI and advanced technologies to revolutionize social media content creation, enabling users to overcome common challenges and drive meaningful engagement with their audience.

CHAPTER 3 EXISTING SYSTEM

The existing methodology for personality prediction encompasses a structured approach centered around the development and implementation of a training model encapsulated within a comprehensive 'train_model' class. This class serves as the backbone for training the personality prediction model, leveraging logistic regression as the underlying statistical method.

Within the 'train_model' class:

The 'train' method orchestrates the intricate process of model training by meticulously parsing and analyzing data sourced from a CSV file. This method intelligently constructs a logistic regression model, meticulously considering seven distinctive features: gender, age, and the quintet of personality traits.

Subsequently, the 'test' method emerges as a pivotal component, facilitating the seamless prediction of personality traits for individual candidates. This method ingeniously processes an array of essential values, including gender, age, and the comprehensive spectrum of personality traits, to deliver accurate predictions.

Main Method:

The central hub of the system, the main method, serves as the gateway to the entire prediction process. It masterfully initializes a Tkinter object, laying the foundation for the intuitive and user-friendly interface that characterizes the system.

In addition to its pivotal role in interface design, the main method ingeniously engineers a button titled "Predict Personality," strategically positioned to seamlessly trigger the subsequent stages of the prediction pipeline.

Predict_Person Method:

This method emerges as a cornerstone in the prediction journey, spearheading the user interaction phase with finesse and precision. It meticulously engineers a new top-level window, meticulously configuring its dimensions and attributes to ensure optimal user experience.

Furthermore, the Predict_Person method ingeniously integrates functionalities that empower users to input vital data crucial for personality prediction, fostering a seamless and intuitive user experience that transcends conventional limitations.

OpenFile Method:

Serving as the proverbial bridge between user interaction and data acquisition, the OpenFile method emerges as a critical component in the prediction process. It deftly attempts to access the designated directory, leveraging default settings and file types to ensure seamless data retrieval.

Furthermore, the OpenFile method showcases unparalleled adaptability by dynamically updating the label of the "choose file" button within the Predict_Person method, seamlessly synchronizing user actions with system responses.

Prediction_Result Method:

This method represents the culmination of the prediction journey, seamlessly synthesizing user inputs, model predictions, and parsed resume information into a coherent and insightful output. It orchestrates the graceful transition from data input to result visualization, fostering a sense of clarity and transparency in the prediction process.

Additionally, the Prediction_Result method serves as a beacon of innovation by harnessing the power of GUI visualization to present user-submitted data, parsed resume information, and predicted personality traits in a visually appealing and comprehensible format.

Check_Type Method:

The unsung hero of data preprocessing, the Check_Type method operates behind the scenes, diligently ensuring data integrity and compatibility across diverse formats and data types. It showcases unparalleled versatility by seamlessly converting strings and numbers into the requisite format, thereby laying the groundwork for accurate and reliable personality predictions.

Furthermore, the Check_Type method embodies a spirit of adaptability, seamlessly transforming lists and tuples into strings to ensure seamless integration with the broader prediction pipeline, thereby underscoring its indispensable role in the prediction journey.

CHAPTER4 PROBLEMSTATEMENT

Problem Definition:

Crafting compelling content for social media platforms has emerged as a formidable challenge amidst the rapid expansion of digital communication channels. With the proliferation of diverse platforms and their distinct requirements, alongside the relentless demand for sustained audience engagement, content creators and businesses grapple with the arduous task of consistently producing captivating text, relevant hashtags, and visually striking imagery. The multifaceted nature of this challenge is further compounded by resource constraints, time limitations, and varying degrees of creative inspiration.

Problem Statement:

The project endeavors to address the pressing need for a system capable of predicting personality traits based on user-provided data. By harnessing machine learning algorithms, the system aims to analyze demographic information and responses to personality-related queries, offering insights into the user's unique personality type.

Problem Description:

In response to the complex landscape of social media content creation, this project seeks to develop an intuitive system that empowers users to predict personality traits effortlessly. Through the utilization of pre-trained machine learning models, notably logistic regression, the system will leverage user-provided data to deliver personalized personality assessments. Implemented in Python, the project will integrate essential libraries such as tkinter for seamless GUI development and scikit-learn for robust machine learning model implementation.

CHAPTER 5 METHODOLOGY

The methodology for advancing personality prediction through CV focuses on leveraging machine learning techniques to refine predictions. The project integrates diverse features, ensemble methods, and neural networks to enhance accuracy and generalization. It also emphasizes data augmentation, domain-specific customization, and evaluation techniques for improved performance. User interface enhancement and feedback incorporation ensure usability and continuous improvement while addressing ethical considerations is crucial

Goals:

Refine Predictions: Enhance prediction accuracy and generalization.

Diverse Feature Integration: Incorporate language use and education background for better predictions.

Algorithm Enhancement: Utilize ensemble methods and deep learning for improved accuracy.

Data Diversity: Increase dataset diversity through synthetic data or perturbations.

Customization: Tailor the model for specific job domains.

Evaluation Enhancement: Implement cross-validation and hyperparameter tuning for better performance.

User Interface Improvement: Enhance user interface for better usability.

Feedback Incorporation: Incorporate user feedback for continuous improvement.

Ethical Considerations: Address bias, privacy, and fairness.

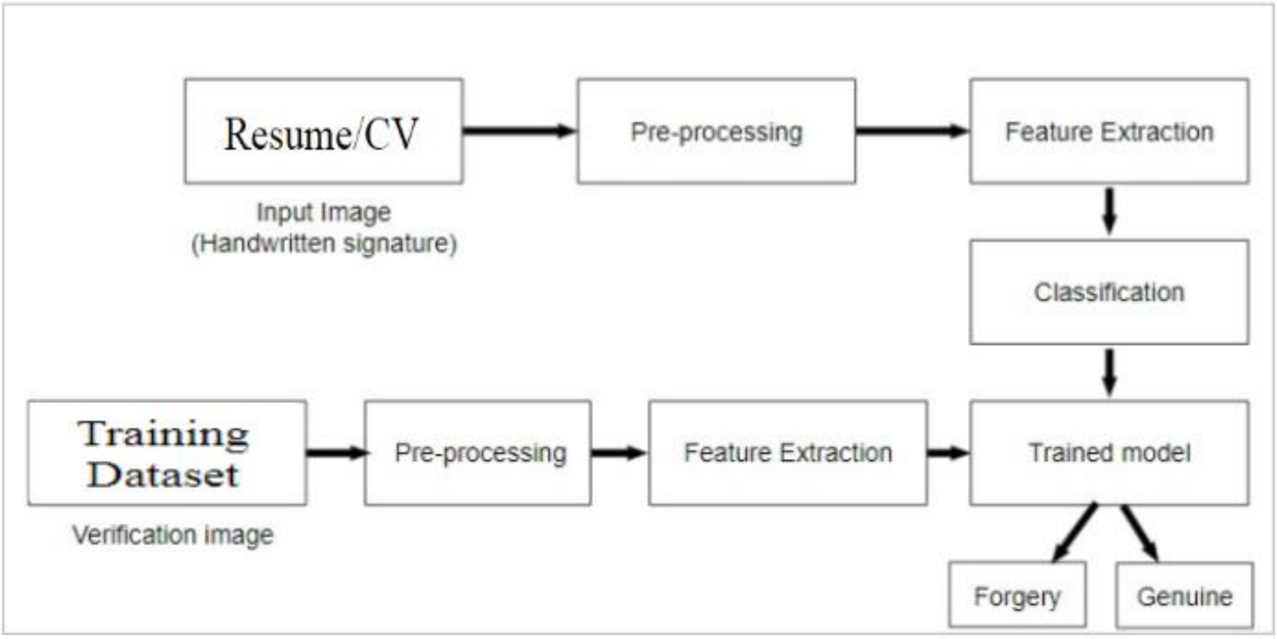
Modules Used:

Os, pandas, numpy, tkinter, functools, pyresparser, sklearn, nltk, scaPy.

Improved Version:

The improved version includes:
Enhanced features like language use and education background.
Utilization of advanced algorithms such as ensemble methods or deep learning.
Data augmentation to increase dataset diversity.
Domain-specific customization for tailored predictions.
Improved evaluation using cross-vyaaalidation and hyperparameter tuning.
User interface enhancement for improved usability.
Incorporation of a feedback loop for continuous improvement.
Ensuring ethical use by addressing bias, privacy, and fairness.

CHAPTER 6 SYSTEM ARCHITECTURE AND DESIGN



Architecture of the Project

Description of Modules

Explanation of Each Module:

os: This module facilitates interaction with the operating system, offering functionality for file manipulation, path handling, and environment variable management.

pandas: Providing data structures and tools for data analysis, pandas is invaluable for tasks like data manipulation, preprocessing, and analysis.

numpy: numpy enables the handling of large, multi-dimensional arrays and offers an array of mathematical functions for numerical computing tasks.

tkinter: tkinter is a GUI toolkit for Python, empowering developers to create desktop applications with graphical components such as buttons, menus, and text boxes.

functools: functools furnishes higher-order functions and operations on callable objects, aiding in the creation and manipulation of functions.

pyresparser: pyresparser is a Python library tailored for parsing and extracting information from resumes (CVs) using natural language processing (NLP) techniques.

sklearn: Also known as scikit-learn, this module is a powerful machine learning library, offering tools for data mining, analysis, and modeling.

nltk: The Natural Language Toolkit is a comprehensive library for natural language processing (NLP), providing utilities for tasks like tokenization, stemming, and parsing.

scaPy: scaPy is a Python library designed for network packet manipulation and analysis, enabling tasks such as packet creation, sniffing, and dissection.
matplotlib: matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. It offers a wide range of plotting functions and customization options.

seaborn: seaborn is a statistical data visualization library based on matplotlib. It provides a high-level interface for creating attractive and informative statistical graphics.

gensim: gensim is a robust library for topic modeling and document similarity analysis. It offers implementations of various algorithms such as Latent Dirichlet Allocation (LDA) and Word2Vec.

spaCy: spaCy is a modern and efficient natural language processing (NLP) library that provides tools for tokenization, part-of-speech tagging, named entity recognition, and dependency parsing.

TensorFlow: TensorFlow is an open-source machine learning framework developed by Google. It provides tools for building and training deep learning models, as well as deployment options for production environments.

Keras: Keras is a high-level neural networks API, written in Python and capable of running on top of TensorFlow, Theano, or Microsoft Cognitive Toolkit (CNTK). It offers a user-friendly interface for building and training deep learning models.

OpenCV: OpenCV (Open Source Computer Vision Library) is a powerful library for computer vision tasks such as image and video processing, object detection, and feature extraction.

CHAPTER 7 IMPLEMENTATION AND RESULTS

1. Dataset Loading:

Use the pandas library to load the dataset from a CSV file into a DataFrame.

Ensure that the dataset contains features such as demographic information (age, gender) and responses to personality-related questions, along with the target variable 'Personality_Type' indicating the predicted personality type.

2. Data Preprocessing:

Handle missing values: Check for missing values in the dataset and decide on a strategy for handling them (e.g., imputation or removal).

Encode categorical variables: If the dataset contains categorical variables, encode them into numerical format using techniques such as one-hot encoding or label encoding.

Split the dataset: Split the dataset into features (X) and the target variable (y) for model training.

3. Model Training:

Select a machine learning algorithm suitable for multi-class classification tasks such as logistic regression.

Split the dataset into training and testing sets using train_test_split from sklearn.model_selection.

Train the model using the training set features (X_train) and corresponding target variable (y_train).

Fit the model to the training data using the fit method.

4. Model Evaluation:

Use the trained model to make predictions on the test set features (X_test).

Compare the predicted personality types with the actual personality types from the test set.

Evaluate the model's performance using metrics such as accuracy_score from sklearn.metrics.

5. Results Analysis:

Analyze the model's performance metrics to assess its effectiveness in predicting personality types.

Examine any misclassifications or areas where the model may need improvement.

Interpret the results and draw conclusions about the feasibility and accuracy of personality prediction through CV.

CHAPTER 8 CONCLUSION AND FUTURE ENHANCEMENT

Conclusion:

Summarizing the key findings and outcomes of the project.

Highlighting the effectiveness of the developed model in predicting personality traits from CVs/resumes.

Emphasize the significance of leveraging AI-driven approaches for enhancing the efficiency and accuracy of the candidate selection process.

Reiterate the potential benefits of integrating personality prediction systems into organizational hiring practices.

Future Enhancements

Looking ahead, there are numerous opportunities for advancing the performance and capabilities of the personality prediction model:

Advanced Machine Learning Techniques: Delve into the integration of state-of-the-art machine learning algorithms or ensemble methods to elevate predictive accuracy and robustness.

Feature Expansion and Refinement: Explore avenues for enriching the feature set or refining existing feature extraction methodologies to capture more nuanced aspects of personality traits.

Scalability and Adaptability: Devise strategies to ensure the scalability and adaptability of the model to evolving hiring requirements and industry dynamics, enabling its continued relevance and effectiveness.

By embarking on these future enhancements and embracing emerging methodologies and technological advancements, the personality prediction system aims to remain at the forefront of character assessment, facilitating informed decision-making and fostering personalized interactions in various domains.

CHAPTER 9 REFERENCES

- Stillwell, and Thore Graepel.
- "Mining Personality Traits from Microblogs: A Psychometric Evaluation of a Twitter-Based Personality Predictor" by H. Andrew Schwartz, Johannes C. Eichstaedt, Margaret L. Kern, Lukasz Dziurzynski, Stephanie M. Ramones, Megha Agrawal, Achal Shah, Michal Kosinski, David Stillwell, Martin E.P. Seligman, and Lyle H. Ungar.
- "Personality Prediction Based on Twitter Stream" by Marcelo Medeiros Eler, Fernanda Lima, Solange Oliveira Rezende, and Daniel Santana de Freitas.
- "Personality Prediction System through CV Analysis Using Machine Learning" by Archana R. Lomte, Prof. S. S. Sambare, Prof. A. V. Deorankar.
- "Personality Trait Classification Using LinkedIn Profiles" by Pramod K. Pisharady, D. S. Guru, and Sowmya Kamath S.
- "Automatic Personality Assessment Through Social Media Language" by Austin H. K. Lee, Shuyang Li, Matthew P. Purver, and Yla Tausczik.
- Predicting Personality Traits of Applicants for Personnel Selection: Evidence from Job Descriptions" by Isaac J. Sabat, Anu Subramanian, Rama Akkiraju, Prakruthi Anantharaju, and Chandan K. Reddy.

APPENDIX I

CODING AND TESTING

Coding

Main.py

```
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
```

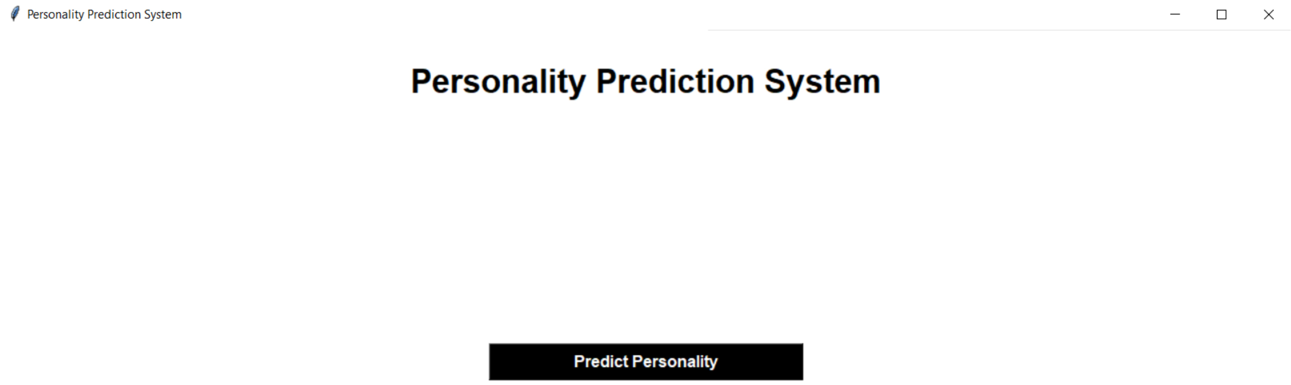
```
            self.mul_lr = linear_model.LogisticRegression(multi_class='multinomial', solver='newton-cg',max_iter =1000)
            self.mul_lr.fit(mainarray, train_y)

        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 = """
      # 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 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(),conscientio
usness.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:
```

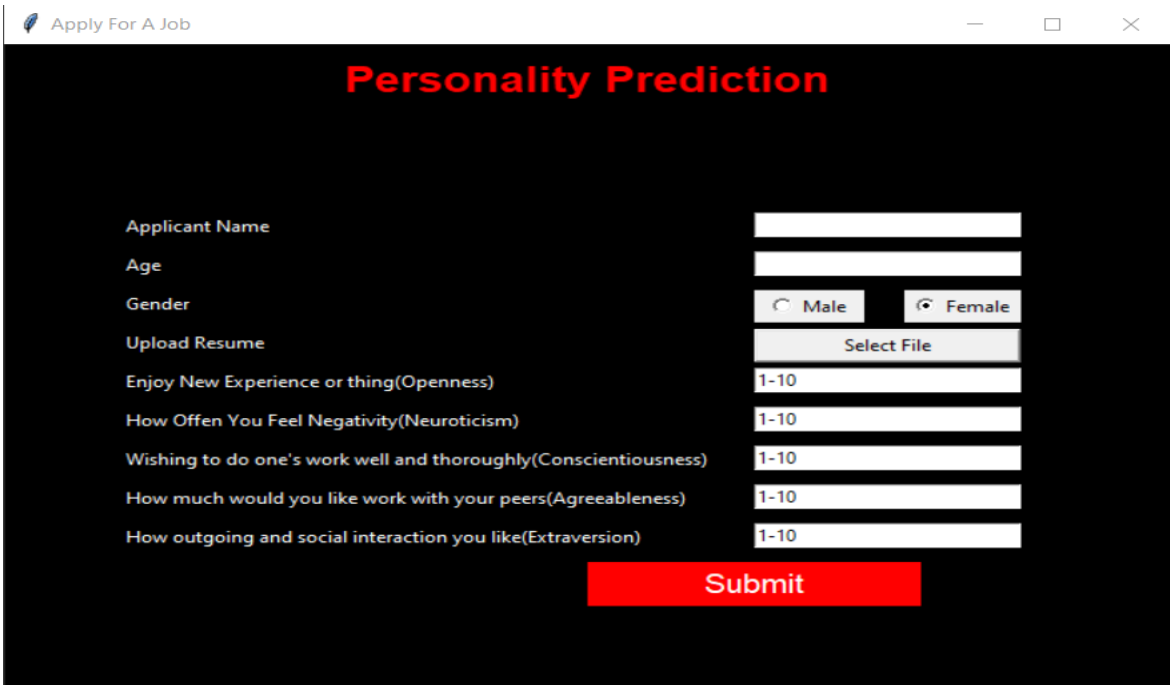
```
        filename=name
        loc=name
    b4.config(text=filename)
    return

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()
```

APPENDIX II SCREENSHOTS



Personality Prediction System

Personality Prediction System



Apply For A Job

Personality Prediction

Applicant Name

Age

Gender ○ Male ● Female

Upload Resume

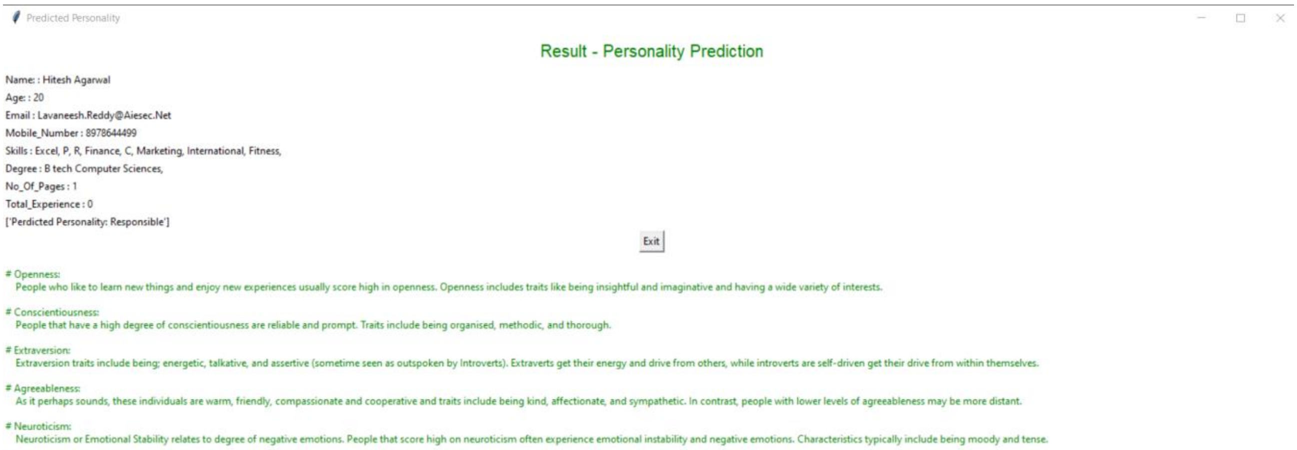
Enjoy New Experience or thing(Openness) 1-10

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



Result - Personality Prediction

Name: : Hitesh Agarwal

Age: : 20

Email : Lavaneesh.Reddy@Aiesec.Net

Mobile_Number : 8978644499

Skills : Excel, P, R, Finance, C, Marketing, International, Fitness,

Degree : B tech Computer Sciences,

No_Of_Pages : 1

Total_Experience : 0

['Perdicted Personality: Responsible']

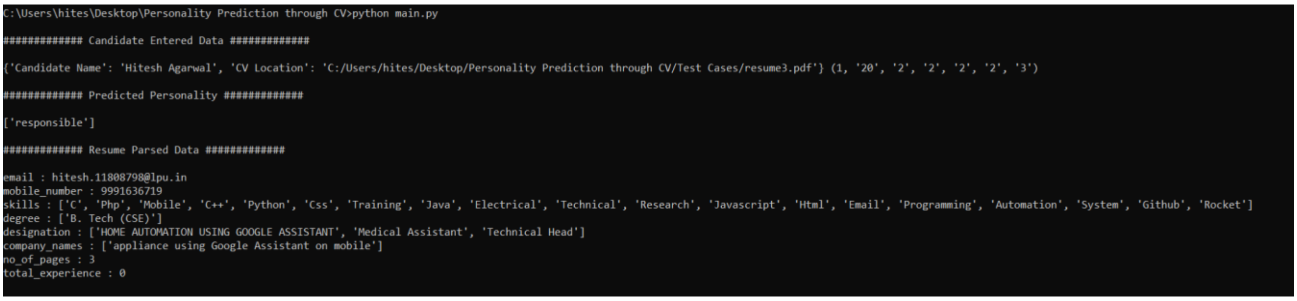
Openness:
People who like to learn new things and enjoy new experiences usually score high in openness. Openness includes traits like being insightful and imaginative and having a wide variety of interests.

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.



```
C:\Users\hites\Desktop\Personality Prediction through CV>python main.py

############# Candidate Entered Data #############

{'Candidate Name': 'Hitesh Agarwal', 'CV Location': 'C:/Users/hites/Desktop/Personality Prediction through CV/Test Cases/resume3.pdf'} (1, '20', '2', '2', '2', '2', '3')

############# Predicted Personality #############

['responsible']

############# Resume Parsed Data #############

email : hitesh.11808798@lpu.in
mobile_number : 9991636719
skills : ['C', 'Php', 'Mobile', 'C++', 'Python', 'Css', 'Training', 'Java', 'Electrical', 'Technical', 'Research', 'Javascript', 'Html', 'Email', 'Programming', 'Automation', 'System', 'Github', 'Rocket']
degree : ['B. Tech (CSE)']
designation : ['HOME AUTOMATION USING GOOGLE ASSISTANT', 'Medical Assistant', 'Technical Head']
company_names : ['appliance using Google Assistant on mobile']
no_of_pages : 3
total_experience : 0
```