

Personality Prediction System

Asmi Mishra¹, Shishu Pandey², Dr. Alok katiyar³

SCSE, Galgotias University

Greater Noida, UP, India

¹Asmi_mishra.scsebtech@galgotiasuniversity.edu.in

²Shishu_pandey.scsebtech@galgotiasuniversity.edu.in

³alok.katiyar@galgotiasuniversity.edu.in

Abstract—

This will enable a more effective way to short list submitted candidate CVs from a large number of applicants providing a consistent and fair CV ranking policy, which can be legally justified. System will rank the experience and key skills required for particular job position. Than system will rank the CV's based on the experience and other key skills which are required for particular job profile. This system will help the HR department to easily shortlist the candidate based on the CV ranking policy. This system will focus not only in qualification and experience but also focuses on other important aspects which are required for particular job position. This system will help the human resource department to select right candidate for particular job profile which in turn provide expert workforce for the organization.

Candidate here will register him/herself with all its details and will upload their own CV into the system which will be further used by the system to shortlist their CV. Candidate can also give an online test which will be conducted on personality questions as well as aptitude questions. After completing the online test, candidate can view their own test results in graphical representation with marks.

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

I. INTRODUCTION

The word Personality comes from the Latin word ‘persona’ which refers to a mask worn by actors to act. However, Personality is much more than a mask now, it could possibly determine whether a person is suitable for a particular job profile. It tells us if a human is capable enough to lead, influence and communicate effectively with others. The first step of recruitment is the job application which consists of personal details, experience, and most importantly CV. Companies typically receive thousands of applications per job opening and have a dedicated team of screeners to select qualified candidates. It is very difficult for human beings to manually go through the CV of all applicants. Many candidates get filtered out in the first round itself on the basis of suitability, improper CV, not being skilled enough. Hiring the right candidate is a very difficult task as no candidate is perfect, some might not be skilled enough or some might not have the right personality. Hence, we propose a way in which the process of shortlisting gets streamlined and faster by personality prediction.

A system proposed by Kessler et al. [12] separates applicants based on their relevance by analyzing unstructured text documents (job offers) and categorizes.

CVs can reflect upon the professional qualifications of a person but do not reflect upon the personality of a person. Personality is one of the vital factors which suggests how a person would be able to work in a designated role, hence personality analysis and understanding is key. Our objective doing this project is to make the machine more human, and analyze the candidate in such a way that an actual human reviewer would.

This paper tries to explore and implement various machine learning algorithms and analyze which one among them provides the best accuracy with a wide array of data provided. We also attempt to visualize the data and form a connection between various factors.

II. LITERATURE SURVEY

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

A set of unique differences that are manipulated by the development of an individuals’ personal memories, values, social relationships, attitude, skills and habits. Characteristics revealed in a certain pattern of behavior in a different bunch of situations form a personality trait. Finding an individual personality trait and intelligence from his or her face plays a crucial role in interpersonal relationships but it is non-reliable. Results show personality traits such as social interaction, capacity of mutual respect, creativity, and many other traits cannot be just estimated by personal interactions [2].

Liden et al. published The General Factor of Personality: The interrelations among the Big Five personality factors (Openness, Conscientiousness, Extraversion, Agreeableness, and Neuroticism) were analyzed in this paper to test for the existence of a GFP. The meta-analysis provides evidence for a GFP at the highest hierarchical level and that the GFP had a substantive component as it is related to supervisor-rated job performance were concluded by this paper. However, it is also realized that it is important to note that the existence of a GFP did not mean that other personality factors that were lower in the hierarchy lost their relevance [14].

Another XML-based multi-agent recommender system, proposed by Pasquale De Meo [13] uses rich user profiles for support. The proposed system was a multi-agent recommender system which used XML for exploiting user profiles to enhance recruitment services in a personalized manner. Another approach uses NLP for standardizing resumes through a modelling language approach. Despite in great usage, these techniques have disparities related to structure, inconsistent CV formats and contextual information. Additionally, the applicants may show themselves in a wellbehaved manner as an online questionnaire's responses can be manipulated for personality inference. Also, social networking sites contain data that is usually irrelevant for recruitment and thus shall not consist sufficient supplementary information regarding the candidate [3].

In this research paper, a system is proposed, which automates the eligibility check and estimates the emotional intelligence by leveraging the potentials of the data found in the test scripts. Various attributes of the test are processed for evaluating the candidate's personality in the system. The professional eligibility of a candidate is checked based on the entries in the online CV submitted by the applicants. Credibility is assured from the mandatory declaration of the users and also resolves the standardization issue. The prime intention is the reduction in the time spent on the initial recruitment phases keeping the end-goal of making the procedure more effective at a higher stage. Overriding of the decision-making capabilities of employers does not take place by the system. Rather, the proposed system helps in removing the time-consuming phases and shortens the tedious process.

III. WORKING

Dataset

As manual data collection is time-consuming, we collected candidate resumes through a lot of websites and personal interaction with potential job seekers taking the total count to 708 CVs. The collected CVs were in PDF and DOCx format.

Methodology

The objective of our paper is to predict the personality of a person based on their score of openness, extraversion, agreeableness, neuroticism and conscientiousness. For achieving this, we needed a way to calculate the scores directly from every CV. Our approach as shown in Fig.1 was to parse the entire resume and search for keywords relating to the Big Five Test.

For predicting the personality of a prospective candidates, we have used various machine learning algorithms like Logistic regression, Naïve Bayes, k-NN, SVM, and Random forest.

Description of flow and methods in the system:

1. **train_model class:** It contains two method which train the model and predict the result by giving the various values.
 - a. **train method:** It read the dataset for training the model from a csv file and build a model using Logistic Regression. It uses different 7 values for training the model

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

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

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

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

2. **main method:** We start with creating an object of train_model class and train the model by calling train method of class. Then we initialize a variable with Tk object and design the landing page of system using labels and button. A button with name Predict Personality is designed which calls predict_person method.

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

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

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

3. **predict_person method:** We withdraw the root tkinter window and create a new toplevel window and configure its size and attributes. We label the heading of window followed by various labels and their entries. For selecting of a resume file, user needs to press choose file button which then calls Openfile method that takes an argument of button. In predict_person method, various entries are taken for predicting the personality. Submit button pass all the values to prediction_result.
4. **OpenFile method:** It tries to open the directory with default address name and file types and except if file not chosen. After try except block, the method changes the name of choose file button in predict_person method with the base name of file so that user can know about the chosen file.
5. **prediction_result method:** This method firstly closes the previous tkinter window which was used to take the data from user. After this, it calls test method of model object and stores the result returned by method. After this it parse all the information from resume and stores in a variable followed by a try except block which try to delete name and validate mobile number from fetched information from resume. Then it prints all the data submitted by user on console. After this, the method popup a full screen window which shows all the parsed information and predicted personality on GUI window along with the definition of each personality characteristic's definition.
6. **check_type method:** It converts various strings and numbers into desired format and converts lists and tuples in string.

IV. CODE

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

class train_model:

    def train(self):
        data = pd.read_csv('training_dataset.csv')
        array = data.values

        for i in range(len(array)):

            if array[i][0] == "Male":
                array[i][0] = 1
            else:
                array[i][0] = 0

        df = pd.DataFrame(array)

        maindf = df[[0,1,2,3,4,5,6]]
```

```

        mainarray=maindf.values

        temp=df[7]
        train_y =temp.values

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

    def test(self, test_data):
    try:
        test_predict=list()
        for i in test_data:
            test_predict.append(int(i))
        y_pred = self.mul_lr.predict([test_predict])
        return y_pred
    except:
        print("All Factors For Finding Personality Not Entered!")

def check_type(data):
if type(data)==str or type(data)==str:
    return str(data).title()
if type(data)==list or type(data)==tuple:
    str_list=""
for i,item in enumerate(data):
    str_list+=item+", "
    return str_list    else:  return str(data)

def prediction_result(top, aplcnt_name, cv_path, personality_values):
    "after applying a job"    top.withdraw()
    applicant_data={"Candidate Name":aplcnt_name.get(),

"CV Location":cv_path}

    age = personality_values[1]

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

    personality = model.test(personality_values)
    print("\n##### Predicted Personality #####\n")    print(personality)
    data = ResumeParser(cv_path).get_extracted_data()

```

```

    try:
        del data['name']
    if len(data['mobile_number'])<10:
        del data['mobile_number']
    except:    pass

    print("\n##### Resume Parsed Data #####\n")

    for key in data.keys():
    if data[key] is not None:
        print('{} : {}'.format(key,data[key]))

    result=Tk() # result.geometry('700x550')
    result.overrideredirect(False)

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

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

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

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

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

```

Conscientiousness:

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

Extraversion:

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

Agreeableness:

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

Neuroticism:

Neuroticism or Emotional Stability relates to degree of negative emotions. People that score high on neuroticism often experience emotional instability and negative emotions. Characteristics typically include being moody and tense. """

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

```
result.mainloop()
```

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

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

```
# Creating new window top = Toplevel() top.geometry('700x500')
top.configure(background='black')
top.title("Apply For A Job")
```

```
#Title
```

```
titleFont = font.Font(family='Helvetica', size=20,
weight='bold')
```

```
lab=Label(top, text="Personality Prediction",
foreground='red', 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(),open
ness.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()

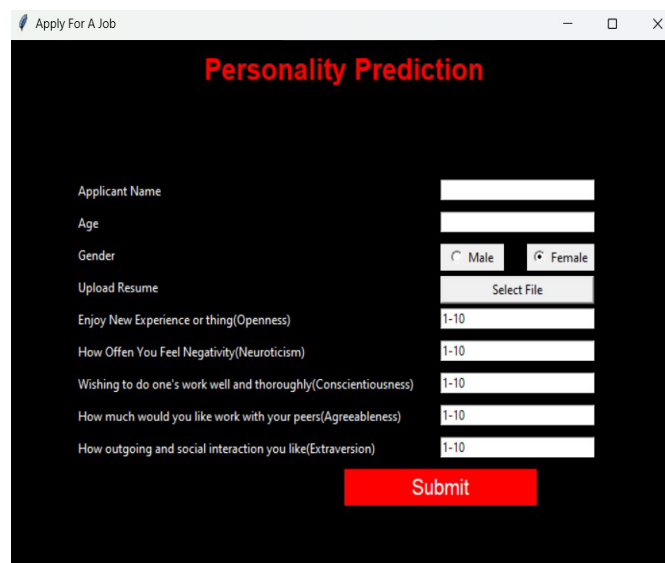
```

V. RESULT

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



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



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

```

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

{'Candidate Name': 'Asmi Mishra', 'CV Location': 'C:/Users/Asus/Desktop/Asmi_Mishra.pdf.docx'} (0, '21', '10', '2', '10', '10', '7')

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

['dependable']
    
```

Output 1

Candidate Entered Data

```
{'Candidate Name': 'Shishu Pandey', 'CV Location': 'C:/Users/Asus/Desktop/Shishu.resume.docx'} (1, '20', '10', '10', '10', '10', '10')
```

Predicted Personality

```
['dependable']
```

Output 2

VI. REFERENCES

- [1] Sudhir Bagade, Jayashree Rout, Pooja Yede, Personality Evaluation and CV Analysis using Machine Learning Algorithm. Article ID155678,21December 2018
- [2] Gayatri Vaidya, Pratima Yadav, Reena Yadav,Prof.Chandana Nighut, “Personality Prediction By Discrete Methodology” ,in Personality traits, IOSR Journal of Engineering (IOSRJEN), ISSN (e): 2250-3021, ISSN (p): 2278- 8719,Volume 14, PP 10-13
- [3] Automated CV Processing along with Psychometric Analysis in Job Recruiting Process Firoz Ahmed, Mehrin Anannya 2, Tanvir Rahman 3, Risala Tasin Khan4Institute of Information Technology, May 2015.
- [4] Janet E. Squires, Leslie Hayduk, Alison M. Hutchinson, Lisa A Cranley, Mark Gierl, Greta G. Cummings, Peter G. Norton, and Carole A. Estabrooks, “A Protocol for Advanced Psychometric Assessment of Surveys”, Nursing Research and Practice Volume 2013 (2013), Article ID156782,20 December 2012.
- [5] Nathan C. Popkins: The Five-Factor Model: Emergence of a Taxonomic Model for Personality Psychology”. December 2018
- [6] M. Kalghatgi, M Ramannavar, and Dr. N. S. Sidnal, Neural Network approach to personality prediction based on the Big-Five Model in IJIRAE, vol2 issue 8, August 2015, pp56-63.
- [7] A..Robey, K. Shukla, K. Agarwal, K. Joshi, Professor S.
- [8] Joshi Personality prediction system through CV Analysis, in IRJET vol 6, issue 02, February2019
- [9] Justindhas Y, Sabari Mutheeswari Mohanraj, Shivani R, "A Synoptic Survey on Personality Prediction System using MBTI", 2022 3rd International Conference on Electronics and Sustainable Communication Systems (ICESC), pp.950-955, 2022
- [10] N. &. M. S. Iyer, “Implementation of Secure Hash Algorithm-1 using FPGA,” International Journal of Information and Computation Technology, Volume III, Issue 3, pp. 50-60, 2018.
- [11] Tommy Tandra, Hendro, Derwin Suhartono, Rini Wongso, and Yen Lina Prasetyo, “Personality Prediction System from Facebook Users”, 2nd International Conference on Computer Science and Computational Intelligence 2017, ICCSCI 2017, 13-14 October 2017, Bali, Indonesia, Vol. No. 10.1016/j.procs.2017.10.016

- [12] Kessler, R., Béchet, N., Torres-Moreno, J.-M., Roche, M., and El-Bèze, M., “Job Offer Management: How Improve the Ranking of Candidates”, in Foundations of Intelligent Systems, 431-441.doi:10.1007/978-3-642-04125-9_46
- [13] Shaza D. Rihan, Ahmed Khalid and Saife Eldin F. Osman, “A Performance Comparison of Encryption Algorithms AES and DES”, International Journal of Engineering Research and Technology (IJERT) ISSN : 2278-0181 Vol. 4 Issue 12, December-2015.
- [14] Pasquale De Meo, Giovanni Quattrone, Giorgio Terracina, and Domenico Ursino, “An XML-based Multi-Agent System for Supporting Online Recruitment Services”, IEEE Transactions on Systems, Man, And Cybernetics, Vol. 37, No. 4, 2017.
- [15] Carolyn Winslow, Xiaoxiao Hu, Seth Kaplan, Yi Li, “Accentuate the Positive: Which Discrete Positive Emotions Predict Which Work Outcomes?”, The Psychologist-Manager Journal – 2017, The Psychologist-Manager Journal, 20(2), 74- 89
- [16] Muhammad Fahim Uddin, Soumita Banerjee, Jeongkyu Lee, “Personality Based Recommendation Engine (PBRE)”, 2016 IEEE 17th International Conference on Information Reuse and Integration (IRI), Electronic ISBN: 978-1-5090-3207-5, Print on Demand(PoD) ISBN: 978-1-5090-3208-2
- [17] Chamorro-Premuzic, T., & Furnham, A. (2017). Personality and music: Can traits explain how people use music in everyday life? *British Journal of Psychology*, 98, 175- 185.
- [18] Iftexhar Naim, M. Iftexhar Tanveer, Daniel Gildea and Mohammed (Ehsan) Hoque, “Automated Analysis and Prediction of Job Interview Performance”, IEEE Transactions on Affective Computing, Volume: 9 , Issue: 2 , April-June 1, 2018
- [19] Muhammad Fahim Uddin, Jeongkyu Lee, “Proposing Stochastic Probability-based Math Model and Algorithms Utilizing Social Networking and Academic Data for Good Fit Students Prediction”, IEEE International Conference – 2018, DOI 10.1007/s13278-017-0448-z
- [20] Vishnu M. Menon and Rahulnath H A, “A Novel Approach to Evaluate and Rank Candidates in a Recruitment Process by Estimating Emotional Intelligence through Social Media Data“, in Big Five Model, 2016 International Conference on Next Generation Intelligent Systems (ICNGIS), 978-1-5090- 0870-4/16 2018