**BACKGROUND OF PERSONALITY PERCEPTION**

The Big Five Personality Traits model is based on findings from several independent researchers, and it dates back to the late 1950s. But the model as we know it now began to take shape in the 1990s. Lewis Goldberg, a researcher at the Oregon Research Institute, is credited with naming the model "The Big Five." It is now considered to be an accurate and respected personality scale, which is routinely used by businesses and in psychological research.

The Big Five Personality Traits Model measures five key dimensions of people's personalities:

* Openness: sometimes called "Intellect" or "Imagination," this measures your level of creativity, and your desire for knowledge and new experiences.
* Conscientiousness: this looks at the level of care that you take in your life and work. If you score highly in conscientiousness, you'll likely be organized and thorough, and know how to make plans and follow them through. If you score low, you'll likely be lax and disorganized.
* Extraversion/Introversion: this dimension measures your level of sociability. Are you outgoing or quiet, for instance? Do you draw energy from a crowd, or do you find it difficult to work and communicate with other people?
* Agreeableness: this dimension measures how well you get on with other people. Are you considerate, helpful and willing to compromise? Or do you tend to put your needs before others'?
* Natural Reactions: sometimes called "Emotional Stability" or "Neuroticism," this measure emotional reactions. Do you react negatively or calmly to bad news? Do you worry obsessively about small details, or are you relaxed in stressful situations?

**DEPENDENCIES OF SYSTEM**

Python Modules/Libraries:

1. OS: For accessing the files and data from internal storage.

2. Pandas: For accessing and manipulating datasheets.

3. Numpy: For working on arrays and other data manipulation.

4. Tkinter: For building the GUI.

5. Functools: Tools for Manipulating Functions. Purpose: Functions that operate on other functions.

6. Pyresparser: Module for extracting information from resume.

7. Sklearn: It features various classification, regression and clustering algorithms. We used sklearn to make the model learn on various characteristic values using logical regression.

**DESCRIPTION**

The system built in this project predicts personality of people by using their gender, age, score of openness, conscientiousness, extraversion, agreeableness, neuroticism and experience. It parses all the data from CV/resume and on the result page, it shows all the information from the entered data and uploaded resume. This system uses logistic regression for training the model and pyresparser module for parsing the information from resume which is built using nltk and spaCy module in python.

**CONTRIBUTION TO KNOWLEDGE**

This system can be used in many business parts/areas that may require expert candidates. This system will reduce the workload of the (workers in general/hiring, training, and firing department). This system will help the (related to workers in general) to select the right candidate for the desired job profile, which in turn provide the expert (all the workers in a company or country) for the organization. Admin can easily shortlist a candidate based on their personality scores and select the appropriate candidate for a particular job profile.

Using Natural Language Processing (NLP) can be defined as a process that enables a machine to become more like a human, because of this deeply cutting the distance between machines and humans. This system will focus not only on qualification and inexperience but also focuses on other important aspects, which are needed/demanded for a particular job position. Admin can store the data in excel sheet for further comparison and sorting of data.

**Source Code/APPENDIX**

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()

print(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("Personality Prediction")

#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 new ways of doing things(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."