PERSONALITY PREDICTION THROUGH CV

END TERM REPORT

by

USMAN ALI

GitHub: https://github.com/Usmanali07/Personality-Prediction-Using-CV

OBJECTIVES

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.

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 peoples 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.

Description of Methods and Flow in the System:

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

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

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

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

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

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

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

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

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

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

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

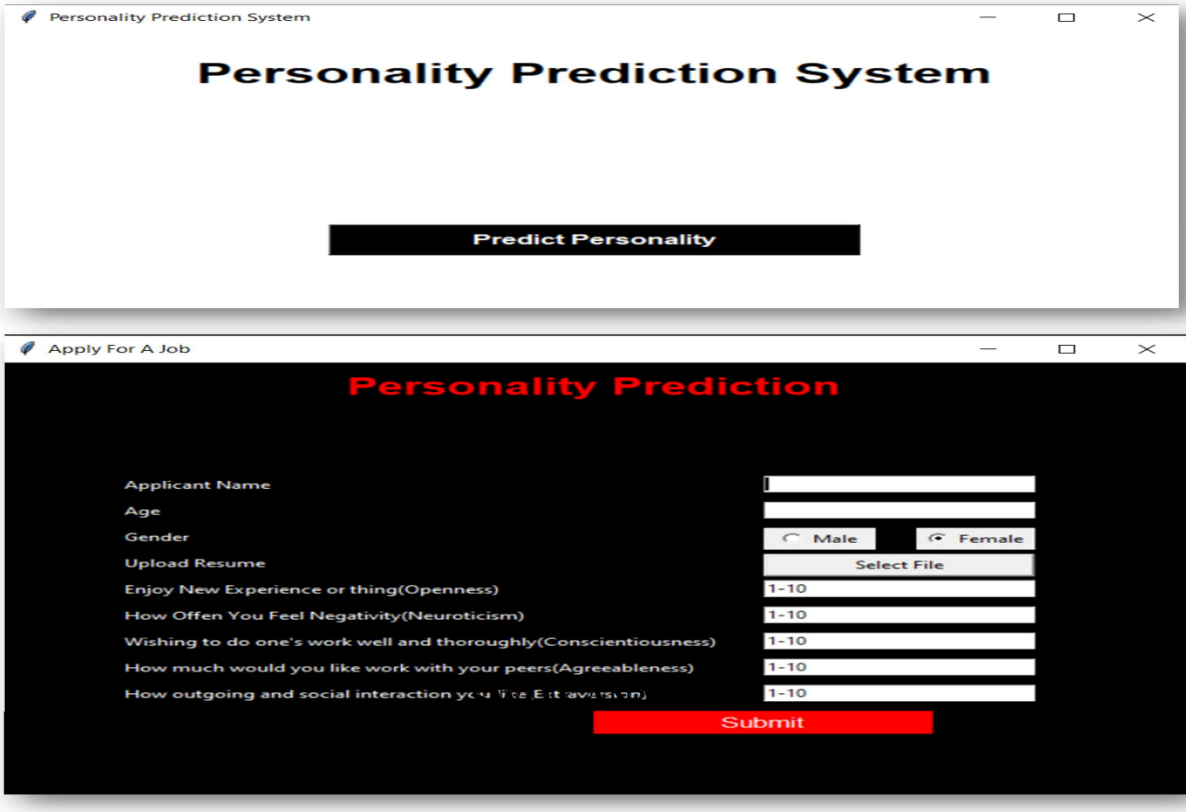
5. **prediction_result method:** This method firstly closes the previous tkinter window which was used to take the data from user. After this, it calls test method of model object and stores the result returned by method. After this it parse all the information from resume and stores in a variable followed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

```python
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)
```

6. **check_type method:** It converts various strings and numbers into desired format and converts lists and tuples in string.

IMPLEMENTATION

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.



SWOT

Strengths:

- Interactive and easy to use.
- Extract all the important features of resume in
- seconds Easily predict the personality of applicant

Weakness:

- It does not store the predicted
- personality data. Bulk of CV cannot be parsed in one go.

Opportunities:

- It can be extended for commercial uses
- It can be made more interactives where we can easily handle bulk data and represent it.
- It can improve the training model for various addition features that help us to predict more accurate result.
- Instead of direct asking the five characteristic values we can add questionaries' which ask some multiple-choice questions and auto calculate the various values.

Threats:

- There is no security added in the app yet that gives different rights to different users.
- There are a lot of companies in the world and their hiring system is different from sector to sector, so it needs to do changes with company to company requirement which can be complicated and expensive to maintains different views of application.