In [ ]:
```
from sklearn.ensemble import RandomForestClassifier
import pandas as pd
```

In [ ]:
```
df_train = pd.read_csv("./dataset/train_features.csv")
df_test = pd.read_csv("./dataset/test_features.csv")
```

In [ ]:
```
df_train
```

Out[ ]:

	letter_slant	line_slant	letter_size	word_spacing	personality
0	13.980637	1.075874	22.50	13.0	Agreeableness
1	-11.017610	4.004173	18.50	25.0	Agreeableness
2	24.000325	-0.836375	10.50	1.0	Agreeableness
3	0.028662	-2.013475	28.00	1.0	Agreeableness
4	9.966474	0.000000	51.00	25.0	Agreeableness
...	...	...	...	...	...
172	6.982799	-4.124767	33.50	25.0	Openness
173	10.989471	0.000000	137.50	19.0	Openness
174	-6.001505	1.084344	65.00	44.0	Openness
175	15.969017	-0.804346	36.75	14.0	Openness
176	7.011250	0.000000	63.00	32.0	Openness

177 rows × 5 columns

In [ ]:
```
df_test
```

Out[ ]:

	letter_slant	line_slant	letter_size	word_spacing	personality
0	9.971437	0.000000	89.75	24.5	Agreeableness
1	-6.001505	0.000000	85.50	45.0	Agreeableness
2	9.994699	0.000000	37.50	55.0	Agreeableness
3	1.003577	1.937840	34.50	85.0	Agreeableness
4	6.982799	1.145763	193.50	27.0	Agreeableness
5	-1.003075	0.000000	43.00	73.5	Agreeableness
6	4.992081	1.086656	9.00	35.0	Agreeableness
7	4.992081	-1.023764	10.50	38.5	Agreeableness
8	9.994699	0.000000	69.00	110.0	Conscientiousness
9	-4.992081	1.677468	172.00	57.0	Conscientiousness
10	11.017610	-3.887910	58.50	22.0	Conscientiousness
11	17.020526	0.000000	15.00	58.5	Conscientiousness
12	0.028662	-1.988638	33.50	66.0	Conscientiousness
13	1.003075	-1.157451	57.00	47.0	Conscientiousness
14	-1.003075	1.709814	49.00	108.0	Conscientiousness
15	-7.996300	0.550904	66.00	92.0	Conscientiousness
16	13.013958	0.000000	95.00	14.0	Extraversion
17	-11.984631	0.000000	38.00	79.0	Extraversion
18	-7.011250	2.097837	152.50	32.0	Neuroticism
19	-1.003577	0.000000	111.00	20.0	Neuroticism
20	11.017610	-2.227568	62.00	64.0	Neuroticism
21	9.994699	1.086070	76.50	13.0	Neuroticism
22	-3.011281	0.813614	57.25	44.5	Neuroticism
23	2.006539	-1.012488	58.75	137.0	Neuroticism
24	8.024690	1.046404	14.00	51.0	Neuroticism
25	-1.003075	-2.140901	7.00	57.0	Neuroticism
26	-7.011250	2.834111	8.00	22.0	Openness
27	3.987588	-1.041627	11.50	14.0	Openness
28	1.003075	-0.473476	13.50	34.0	Openness
29	-7.996300	1.134422	7.75	43.5	Openness
30	10.989471	0.000000	26.00	31.5	Openness
31	10.011297	0.000000	65.00	81.0	Openness
32	-8.020690	0.000000	92.00	30.0	Openness
33	-13.007505	0.000000	46.50	45.0	Openness
34	-4.992081	0.000000	54.00	13.0	Openness

	letter_slant	line_slant	letter_size	word_spacing	personality
35	12.979579	0.000000	86.50	24.0	Openness
36	-6.001505	3.012788	194.00	32.0	Openness
37	12.986020	0.000000	65.50	46.0	Openness
38	14.988652	2.267955	95.25	62.0	Openness
39	10.989471	-2.073193	61.00	13.0	Openness
40	0.028662	-1.943317	25.00	25.0	Openness
41	4.000882	-0.572939	60.00	100.0	Openness
42	8.024690	-2.202598	43.00	104.0	Openness
43	-4.016182	0.000000	41.50	42.0	Openness

In [ ]:
```
print(df_train.info())
print(df_test.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 177 entries, 0 to 176
Data columns (total 5 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   letter_slant   177 non-null    float64
 1   line_slant     176 non-null    float64
 2   letter_size    177 non-null    float64
 3   word_spacing   177 non-null    float64
 4   personality    177 non-null    object 
dtypes: float64(4), object(1)
memory usage: 7.0+ KB
None
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 44 entries, 0 to 43
Data columns (total 5 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   letter_slant   44 non-null     float64
 1   line_slant     44 non-null     float64
 2   letter_size    44 non-null     float64
 3   word_spacing   44 non-null     float64
 4   personality    44 non-null     object 
dtypes: float64(4), object(1)
memory usage: 1.8+ KB
None
```

In [ ]:
```
df_train.isnull().sum()
```

Out[ ]:
```
letter_slant    0
line_slant      1
letter_size     0
word_spacing    0
personality     0
dtype: int64
```

In [ ]:
```
df_test.isnull().sum()
```

Out[ ]:
```
letter_slant    0
line_slant      0
letter_size     0
word_spacing    0
personality     0
dtype: int64
```

In [ ]:
```
df_train.dropna(inplace=True)
df_test.dropna(inplace=True)
```

In [ ]:
```
print(df_train.info())
print(df_test.info())
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 176 entries, 0 to 176
Data columns (total 5 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   letter_slant   176 non-null    float64
 1   line_slant     176 non-null    float64
 2   letter_size    176 non-null    float64
 3   word_spacing   176 non-null    float64
 4   personality    176 non-null    object 
dtypes: float64(4), object(1)
memory usage: 8.2+ KB
None
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 44 entries, 0 to 43
Data columns (total 5 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   letter_slant   44 non-null     float64
 1   line_slant     44 non-null     float64
 2   letter_size    44 non-null     float64
 3   word_spacing   44 non-null     float64
 4   personality    44 non-null     object 
dtypes: float64(4), object(1)
memory usage: 1.8+ KB
None
```

In [ ]:
```
x_train = df_train.drop('personality', axis=1)
y_train = df_train['personality']
x_test = df_test.drop('personality', axis=1)
y_test = df_test['personality']
```

In [ ]:
```
rf = RandomForestClassifier(n_estimators=100, criterion="gini", random_state=42)
rf_res = rf.fit(x_train, y_train)
```

In [ ]:
```
y_pred = rf.predict(x_test)
print(y_pred)
```

```
['Conscientiousness' 'Openness' 'Openness' 'Agreeableness' 'Agreeableness'
 'Agreeableness' 'Openness' 'Openness' 'Neuroticism' 'Openness'
 'Neuroticism' 'Openness' 'Conscientiousness' 'Neuroticism' 'Openness'
 'Agreeableness' 'Openness' 'Conscientiousness' 'Extraversion'
 'Conscientiousness' 'Neuroticism' 'Openness' 'Openness' 'Openness'
 'Openness' 'Neuroticism' 'Neuroticism' 'Openness' 'Openness' 'Openness'
 'Openness' 'Conscientiousness' 'Conscientiousness' 'Neuroticism'
 'Extraversion' 'Openness' 'Agreeableness' 'Openness' 'Openness'
 'Neuroticism' 'Openness' 'Openness' 'Openness' 'Openness']
```

In [ ]:
```
from sklearn.metrics import accuracy_score, confusion_matrix
accuracy = accuracy_score(y_test, y_pred)
```

```
accuracy
```

Out[ ]:
```
0.38636363636363635
```

In [ ]:
```
rf.feature_importances_
```

Out[ ]:
```
array([0.22648692, 0.21817198, 0.31666015, 0.23868096])
```

In [ ]:
```
x_test.head(2)
```

Out[ ]:

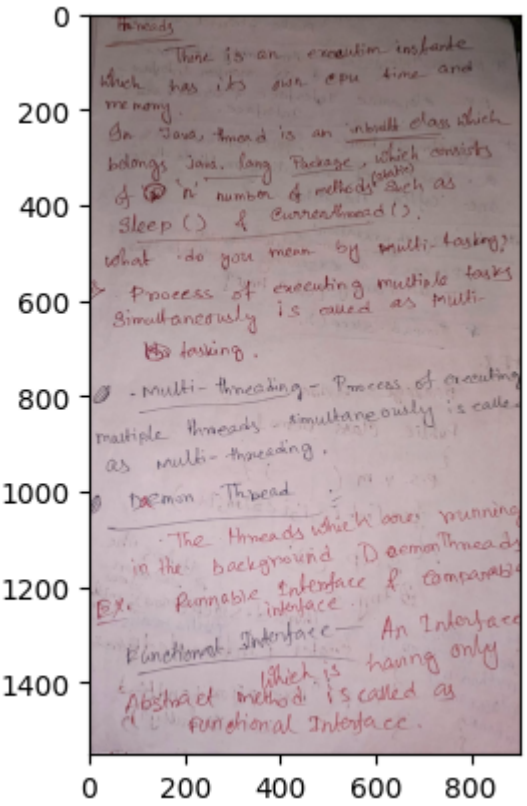
	letter_slant	line_slant	letter_size	word_spacing
0	9.971437	0.0	89.75	24.5
1	-6.001505	0.0	85.50	45.0

In [ ]:
```
import os
import copy
from package.features import *
import cv2
from matplotlib import pyplot as plt
```

In [ ]:
```
image_path = input("Enter image path: ")
```

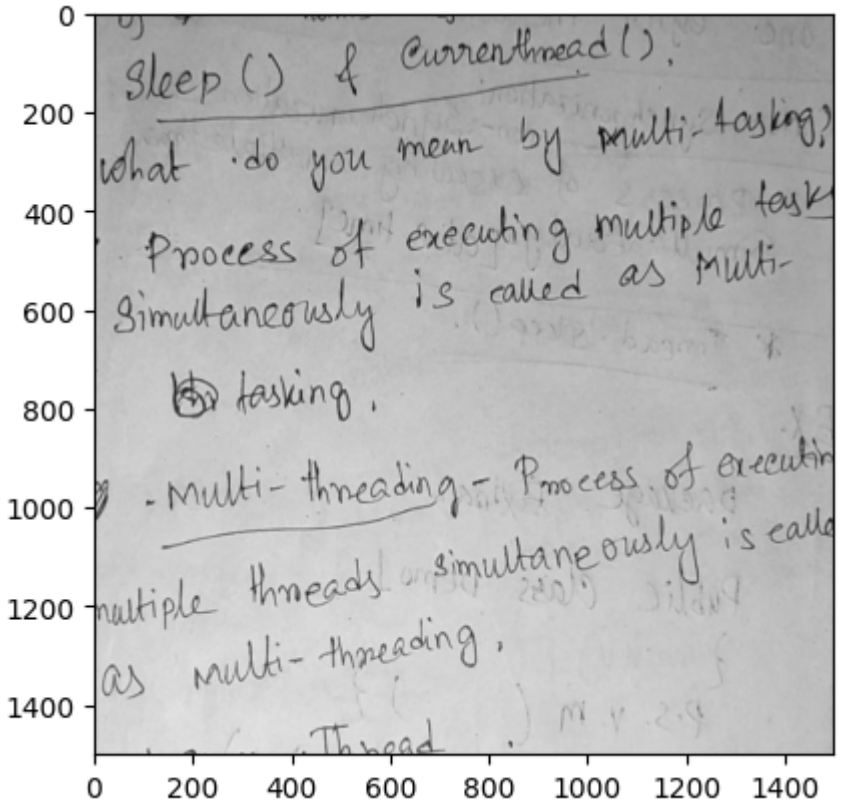
In [ ]:
```
img = cv2.imread(image_path)
plt.imshow(img)
```

Out[ ]:
```
<matplotlib.image.AxesImage at 0x22639db41d0>
```



In [ ]:
```
img = auto_crop_image(image_path)
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
gray = cv2.medianBlur(gray, 3)
thresh = cv2.adaptiveThreshold(gray, 255, cv2.ADAPTIVE_THRESH_MEAN_C, cv2.THRESH_BINARY_INV, 2
dilate = cv2.dilate(thresh, (5, 5), iterations=10)
plt.imshow(gray, cmap="gray")
```

Out[ ]:
```
<matplotlib.image.AxesImage at 0x2263c7d1510>
```



In [ ]:
```
mydataset = {
    'letter_slant': [get_letter_slant(image_path=image_path)[1]],
    'line_slant': [get_line_slant(image_path=image_path)[2]],
    'letter_size': [get_letter_size(image_path=image_path)[0]],
    'word_spacing': [gap_between_words(image_path=image_path)[2]],
}
my_df = pd.DataFrame(mydataset)
my_df
```

Out[ ]:

	letter_slant	line_slant	letter_size	word_spacing
0	4.994576	-5.76311	100.9	7.0

In [ ]:
```
my_pred = rf.predict(my_df)
my_pred
```

Out[ ]:
```
array(['Openness'], dtype=object)
```