In [ ]:
```
from sklearn.neighbors import KNeighborsClassifier
import pandas as pd
```

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

In [ ]:
```
df_train.head()
```

Out[ ]:

	letter_slant	line_slant	letter_size	word_spacing	personality
0	backward	upperside	1307.2	small	Agreeableness
1	backward	upperside	932.0	small	Agreeableness
2	forward	upperside	891.4	small	Agreeableness
3	forward	upperside	279.6	small	Agreeableness
4	vertical	upperside	766.0	small	Agreeableness

In [ ]:
```
df_test.head()
```

Out[ ]:

	letter_slant	line_slant	letter_size	word_spacing	personality
0	backward	upperside	1307.2	small	Agreeableness
1	backward	upperside	932.0	small	Agreeableness
2	forward	upperside	891.4	small	Agreeableness
3	forward	upperside	279.6	small	Agreeableness
4	vertical	upperside	766.0	small	Agreeableness

In [ ]:
```
letter_slant_mapping = {'backward': -1, 'forward': 1, 'vertical': 0}
line_slant_mapping = {'lowerside': -1, 'baseline': 0, 'upperside': 1}
word_spacing_mapping = {'small': -1, 'medium': 0, 'large': 1}
```

In [ ]:
```
df_train["letter_slant"] = df_train["letter_slant"].map(letter_slant_mapping)
df_train["line_slant"] = df_train["line_slant"].map(line_slant_mapping)
df_train["word_spacing"] = df_train["word_spacing"].map(word_spacing_mapping)
df_test["letter_slant"] = df_test["letter_slant"].map(letter_slant_mapping)
df_test["line_slant"] = df_test["line_slant"].map(line_slant_mapping)
df_test["word_spacing"] = df_test["word_spacing"].map(word_spacing_mapping)
```

In [ ]:
```
df_train.head()
```

Out[ ]:

	letter_slant	line_slant	letter_size	word_spacing	personality
0	-1	1	1307.2	-1	Agreeableness
1	-1	1	932.0	-1	Agreeableness
2	1	1	891.4	-1	Agreeableness
3	1	1	279.6	-1	Agreeableness
4	0	1	766.0	-1	Agreeableness

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    int64 
 1   line_slant     177 non-null    int64 
 2   letter_size    177 non-null    float64
 3   word_spacing   177 non-null    int64 
 4   personality    177 non-null    object
dtypes: float64(1), int64(3), object(1)
memory usage: 7.0+ KB
None
<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    int64 
 1   line_slant     177 non-null    int64 
 2   letter_size    177 non-null    float64
 3   word_spacing   177 non-null    int64 
 4   personality    177 non-null    object
dtypes: float64(1), int64(3), object(1)
memory usage: 7.0+ KB
None
```

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

Out[ ]:
```
letter_slant    0
line_slant      0
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'>
RangeIndex: 177 entries, 0 to 176
Data columns (total 5 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   letter_slant   177 non-null    int64  
 1   line_slant     177 non-null    int64  
 2   letter_size    177 non-null    float64
 3   word_spacing   177 non-null    int64  
 4   personality    177 non-null    object 
dtypes: float64(1), int64(3), object(1)
memory usage: 7.0+ KB
None
<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    int64  
 1   line_slant     177 non-null    int64  
 2   letter_size    177 non-null    float64
 3   word_spacing   177 non-null    int64  
 4   personality    177 non-null    object 
dtypes: float64(1), int64(3), object(1)
memory usage: 7.0+ 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 [ ]:
```
knn = KNeighborsClassifier(n_neighbors=3, algorithm="brute")
knn_res = knn.fit(x_train, y_train)
```

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

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

```
d:\Subhadip\MCA_Projects\Personality_Prediction\venv\Lib\site-packages\joblib\externals\loky\backend\context.py:136: UserWarning: Could not find the number of physical cores for the following reason:
found 0 physical cores < 1
Returning the number of logical cores instead. You can silence this warning by setting LOKY_MAX_CPU_COUNT to the number of cores you want to use.
  warnings.warn(
  File "d:\Subhadip\MCA_Projects\Personality_Prediction\venv\Lib\site-packages\joblib\externals\loky\backend\context.py", line 282, in _count_physical_cores
    raise ValueError(f"found {cpu_count_physical} physical cores < 1")
```

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

Out[ ]:
```
0.5988700564971752
```

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

Out[ ]:

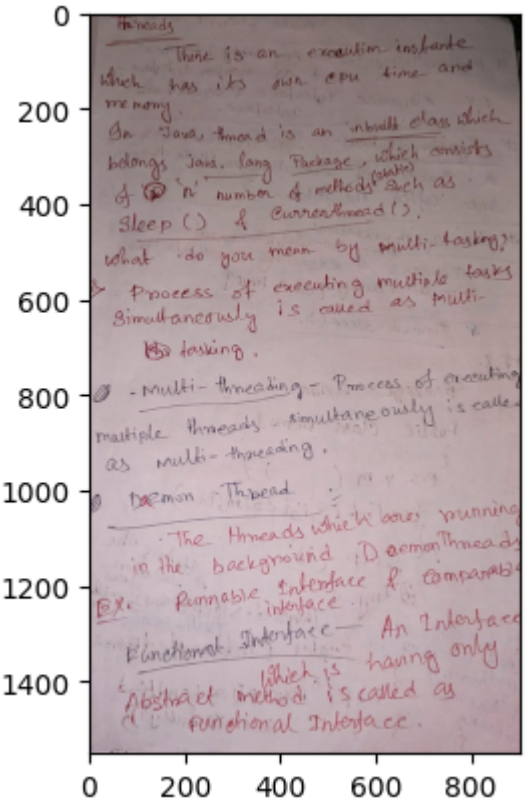
	letter_slant	line_slant	letter_size	word_spacing
0	-1	1	1307.2	-1
1	-1	1	932.0	-1

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

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

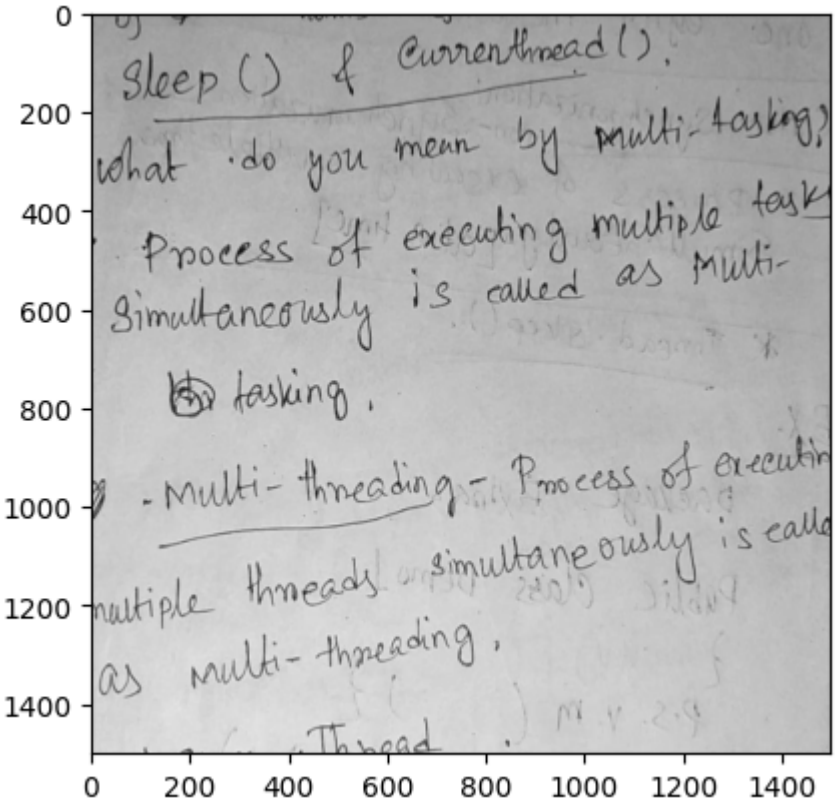
In [ ]:
```
img = cv2.imread(image_path)
plt.imshow(img)
```

Out[ ]:
```
<matplotlib.figure.AxesImage at 0x2a127226c10>
```



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.figure.AxesImage at 0x2a12790ce10>
```



In [ ]:
```
mydataset = {
    'letter_slant': [get_letter_slant(image_path=image_path)[0]],
    'line_slant': [get_line_slant(image_path=image_path)[0]],
    'letter_size': [get_letter_size(image_path=image_path)[0]],
    'word_spacing': [gap_between_words(image_path=image_path)[0]],
}
my_df = pd.DataFrame(mydataset)
my_df["letter_slant"] = my_df["letter_slant"].map(letter_slant_mapping)
my_df["line_slant"] = my_df["line_slant"].map(line_slant_mapping)
my_df["word_spacing"] = my_df["word_spacing"].map(word_spacing_mapping)
my_df
```

Out[ ]:

	letter_slant	line_slant	letter_size	word_spacing
0	1	1	100.9	-1

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

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