Resume Screening using Natural Language Processing and Machine Learning

Importing the required libraries

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
from sklearn.naive_bayes import MultinomialNB
from sklearn.multiclass import OneVsRestClassifier
from sklearn import metrics
from sklearn.metrics import accuracy_score
from pandas.plotting import scatter_matrix
from sklearn.neighbors import KNeighborsClassifier
from sklearn import metrics
```

In [2]:
```
resumeDataSet = pd.read_csv("C:/Users/DELL/Desktop/Minor Project 1/UpdatedResumeDataSet.csv",
resumeDataSet['cleaned_resume'] = ''
```

Exploratory Data Analysis

In [3]:
```
resumeDataSet.head()
```

Out[3]:

	Category	Resume	cleaned_resume
0	Data Science	Skills * Programming Languages: Python (pandas...	
1	Data Science	Education Details \r\nMay 2013 to May 2017 B.E...	
2	Data Science	Areas of Interest Deep Learning, Control Syste...	
3	Data Science	Skills â□¢ R â□¢ Python â□¢ SAP HANA â□¢ Table...	
4	Data Science	Education Details \r\n MCA YMCAUST, Faridab...	

In [4]:
```
resumeDataSet.shape
```

Out[4]:
```
(962, 3)
```

In [5]:
```
resumeDataSet.describe()
```

Out[5]:

	Category	Resume	cleaned_resume
count	962	962	962
unique	25	166	1
top	Java Developer	Technical Skills Web Technologies: Angular JS,...	
freq	84	18	962

In [6]:
```
resumeDataSet.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 962 entries, 0 to 961
Data columns (total 3 columns):
 #   Column          Non-Null Count  Dtype 
---  ------          --------------  ----- 
 0   Category        962 non-null    object
 1   Resume          962 non-null    object
 2   cleaned_resume  962 non-null    object
dtypes: object(3)
memory usage: 22.7+ KB
```

In [7]:
```
resumeDataSet.tail()
```

Out[7]:

	Category	Resume	cleaned_resume
957	Testing	Computer Skills: â□¢ Proficient in MS office (...	
958	Testing	â□□ Willingness to accept the challenges. â□□ ...	
959	Testing	PERSONAL SKILLS â□¢ Quick learner, â□¢ Eagerne...	
960	Testing	COMPUTER SKILLS & SOFTWARE KNOWLEDGE MS-Power ...	
961	Testing	Skill Set OS Windows XP/7/8/8.1/10 Database MY...	

In [8]:
```
resumeDataSet.dtypes
```

Out[8]:
```
Category          object
Resume            object
cleaned_resume    object
dtype: object
```

In [9]:
```
resumeDataSet['Category'].value_counts()
```

Out[9]:
```
Java Developer               84
Testing                      70
DevOps Engineer              55
Python Developer             48
Web Designing                45
HR                           44
Hadoop                       42
Blockchain                   40
ETL Developer                40
Operations Manager           40
Data Science                 40
Sales                        40
Mechanical Engineer          40
Arts                         36
Database                     33
Electrical Engineering       30
Health and fitness           30
PMO                          30
Business Analyst             28
DotNet Developer             28
Automation Testing           26
Network Security Engineer    25
SAP Developer                24
Civil Engineer               24
Advocate                     20
Name: Category, dtype: int64
```

In [10]:
```
resumeDataSet.isnull().sum()
```

Out[10]:
```
Category          0
Resume            0
cleaned_resume    0
dtype: int64
```

In [11]:
```
print ("Displaying the distinct categories of resume -")
print (resumeDataSet['Category'].unique())
```

```
Displaying the distinct categories of resume -
['Data Science' 'HR' 'Advocate' 'Arts' 'Web Designing'
 'Mechanical Engineer' 'Sales' 'Health and fitness' 'Civil Engineer'
 'Java Developer' 'Business Analyst' 'SAP Developer' 'Automation Testing'
 'Electrical Engineering' 'Operations Manager' 'Python Developer'
 'DevOps Engineer' 'Network Security Engineer' 'PMO' 'Database' 'Hadoop'
 'ETL Developer' 'DotNet Developer' 'Blockchain' 'Testing']
```

In [12]:

```
print ("Displaying the distinct categories of resume and the number of records belonging to e
print (resumeDataSet['Category'].value_counts())
```

```
Displaying the distinct categories of resume and the number of records belonging to each cate
gory -
Java Developer               84
Testing                      70
DevOps Engineer              55
Python Developer             48
Web Designing                45
HR                           44
Hadoop                       42
Blockchain                   40
ETL Developer                40
Operations Manager           40
Data Science                 40
Sales                        40
Mechanical Engineer          40
Arts                         36
Database                     33
Electrical Engineering       30
Health and fitness           30
PMO                          30
Business Analyst             28
DotNet Developer             28
Automation Testing           26
Network Security Engineer    25
SAP Developer                24
Civil Engineer               24
Advocate                     20
Name: Category, dtype: int64
```

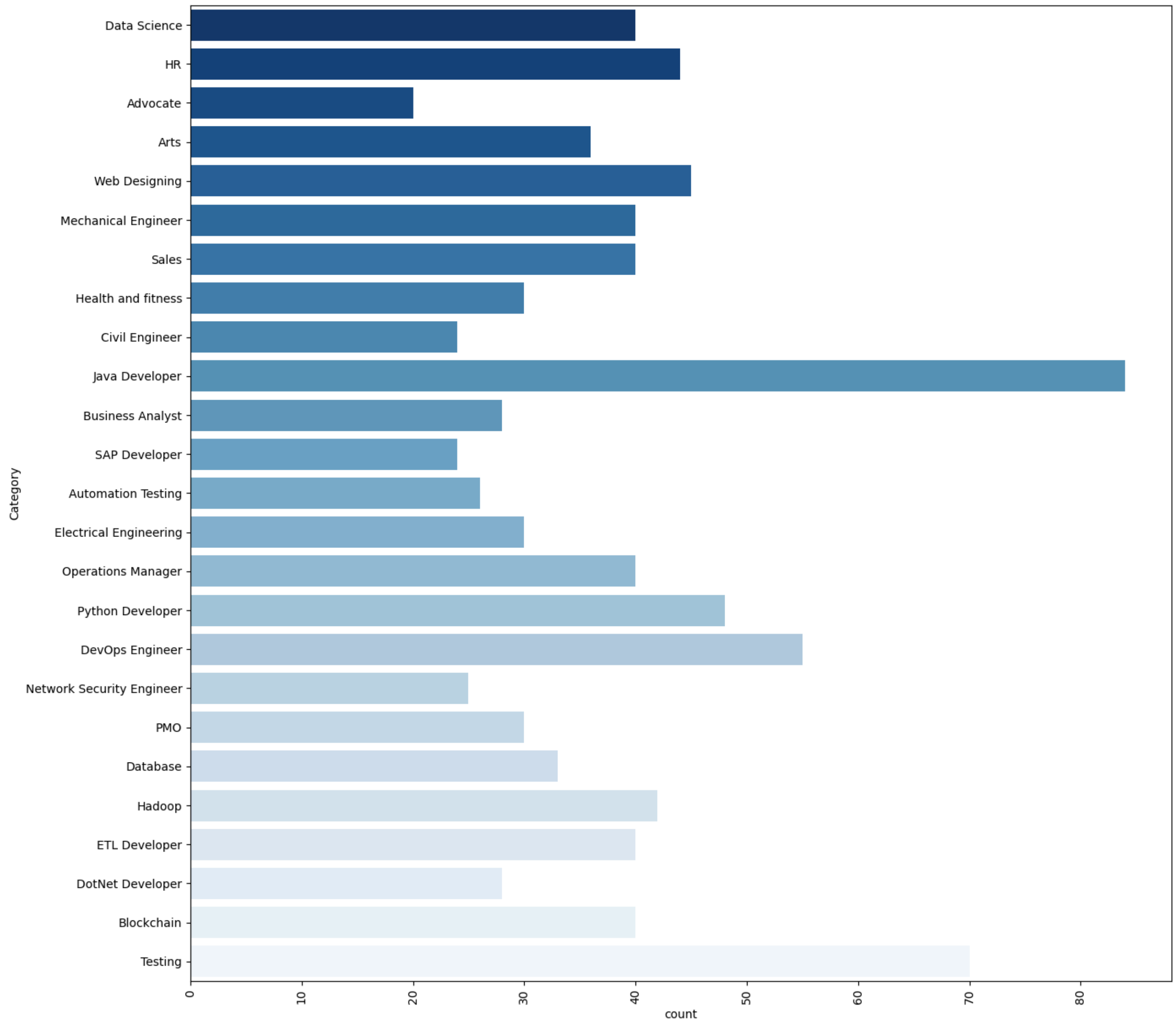
Data Visualization

In [13]:

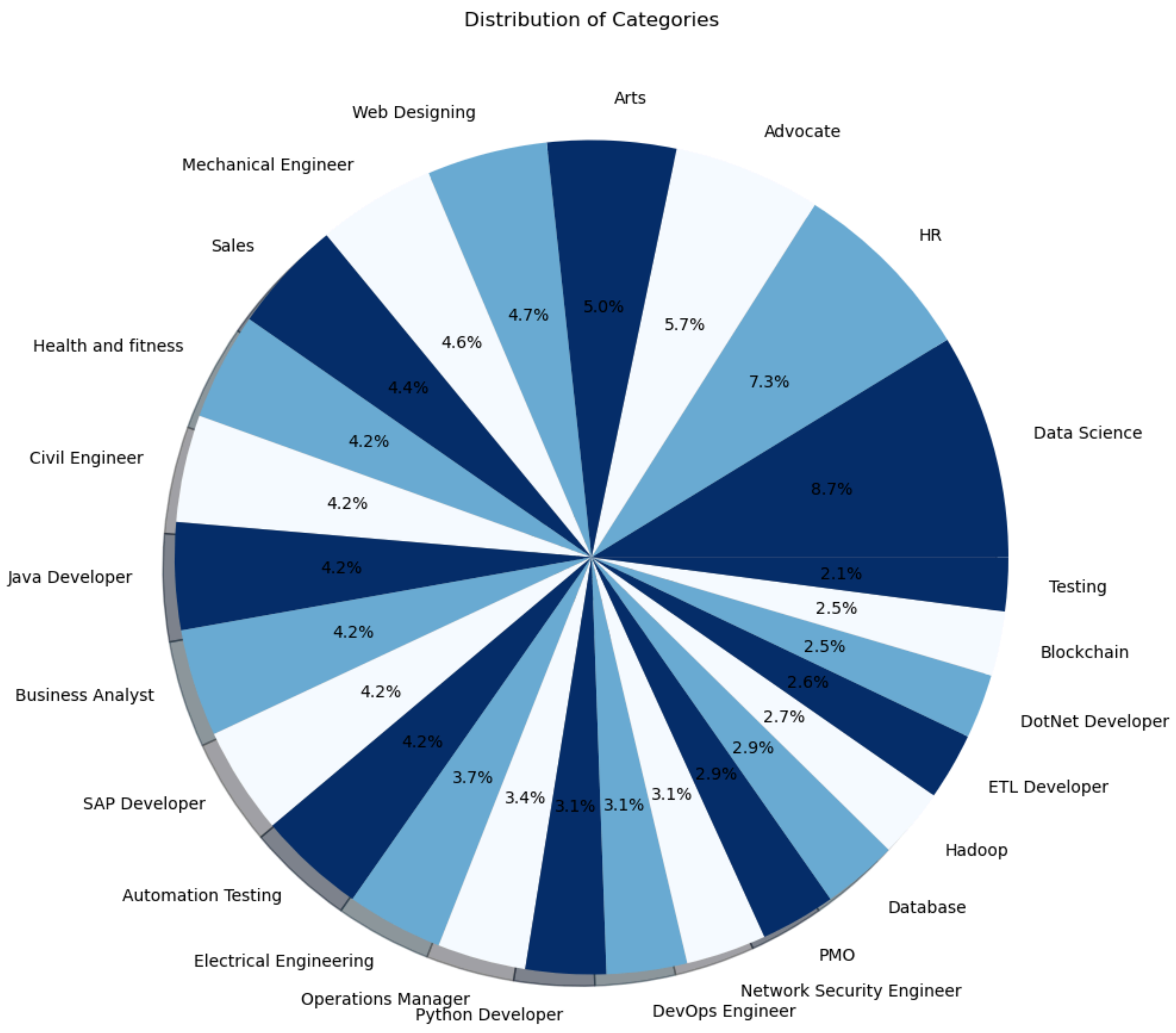
```
import seaborn as sns
plt.figure(figsize=(15,15))
plt.xticks(rotation=90)
sns.countplot(y="Category", data=resumeDataSet,palette="Blues_r")
```

Out[13]:

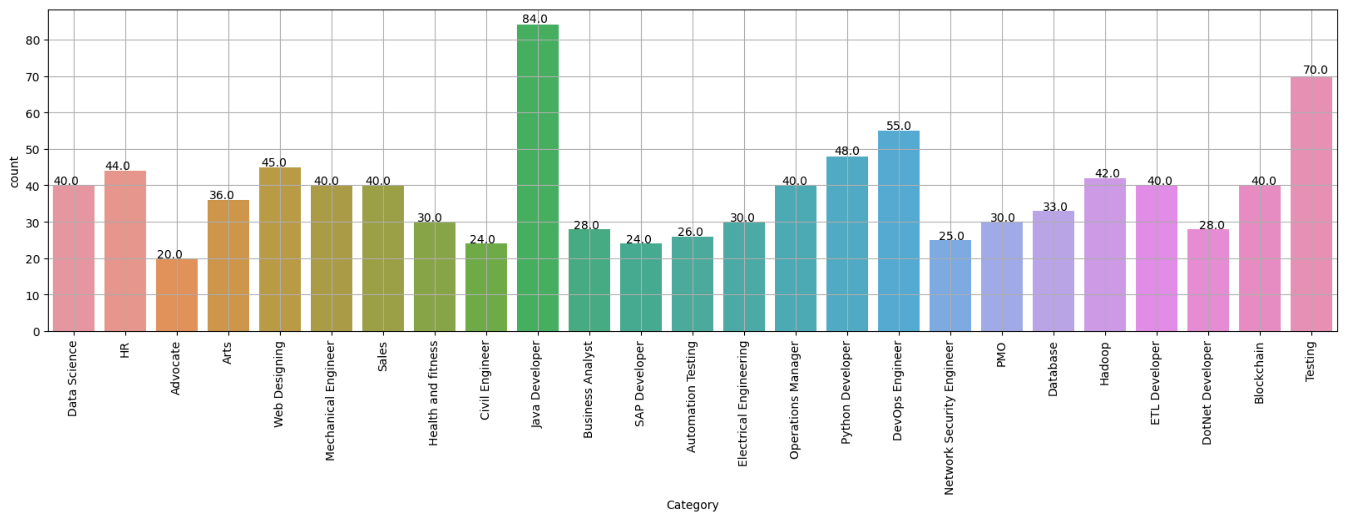
```
<Axes: xlabel='count', ylabel='Category'>
```



```
In [15]: from matplotlib.gridspec import GridSpec
         targetCounts = resumeDataSet['Category'].value_counts()
         targetLabels  = resumeDataSet['Category'].unique()
         plt.figure(1, figsize=(25,25))
         the_grid = GridSpec(2, 2)
         cmap = plt.get_cmap('Blues_r')
         colors = [cmap(i) for i in np.linspace(0, 1, 3)]
         plt.subplot(the_grid[0, 1], aspect=1, title='Distribution of Categories')
         source_pie = plt.pie(targetCounts, labels=targetLabels, autopct='%1.1f%%', shadow=True, color
         plt.show()
```



In [16]:
```python
plt.figure(figsize=(20,5))
plt.xticks(rotation=90)
ax=sns.countplot(x="Category", data=resumeDataSet)
for p in ax.patches:
    ax.annotate(str(p.get_height()), (p.get_x() * 1.01 , p.get_height() * 1.01))
plt.grid()
```



Data Preprocessing

In [17]:
```python
import re
def cleanResume(resumeText):
    resumeText = re.sub('http\S+\s*', ' ', resumeText)  # remove URLs
    resumeText = re.sub('RT|cc', ' ', resumeText)  # remove RT and cc
    resumeText = re.sub('#\S+', '', resumeText)  # remove hashtags
    resumeText = re.sub('@\S+', '  ', resumeText)  # remove mentions
    resumeText = re.sub('[%s]' % re.escape("""!"#$%&'()*+,-./:;<=>?@[\]^_`{|}~"""), ' ', resu
    resumeText = re.sub(r'[^\x00-\x7f]',r' ', resumeText)
    resumeText = re.sub('\s+', ' ', resumeText)  # remove extra whitespace
    return resumeText

resumeDataSet['cleaned_resume'] = resumeDataSet.Resume.apply(lambda x: cleanResume(x))
```

In [18]:
```
resumeDataSet.head()
```

Out[18]:

	Category	Resume	cleaned_resume
0	Data Science	Skills * Programming Languages: Python (pandas...	Skills Programming Languages Python pandas num...
1	Data Science	Education Details \r\nMay 2013 to May 2017 B.E...	Education Details May 2013 to May 2017 B E UIT...
2	Data Science	Areas of Interest Deep Learning, Control Syste...	Areas of Interest Deep Learning Control System...
3	Data Science	Skills â□¢ R â□¢ Python â□¢ SAP HANA â□¢ Table...	Skills R Python SAP HANA Tableau SAP HANA SQL ...
4	Data Science	Education Details \r\n MCA YMCAUST, Faridab...	Education Details MCA YMCAUST Faridabad Haryan...

In [19]:
```
import nltk
nltk.download('stopwords')
nltk.download('punkt')
from nltk.corpus import stopwords
import string
from wordcloud import WordCloud
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\DELL\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\DELL\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
```

In [20]:
```
oneSetOfStopWords = set(stopwords.words('english')+['``',"''"])
totalWords =[]
Sentences = resumeDataSet['Resume'].values
cleanedSentences = ""
for i in range(0,160):
    cleanedText = cleanResume(Sentences[i])
    cleanedSentences += cleanedText
    requiredWords = nltk.word_tokenize(cleanedText)
    for word in requiredWords:
        if word not in oneSetOfStopWords and word not in string.punctuation:
            totalWords.append(word)
wordfreqdist = nltk.FreqDist(totalWords)
mostcommon = wordfreqdist.most_common(50)
print("Most Commonly used words : \n")
for i in mostcommon:
    print(i)
```

```
Most Commonly used words :

('Details', 484)
('Exprience', 446)
('months', 376)
('company', 330)
('description', 310)
('1', 290)
('year', 232)
('January', 216)
('Less', 204)
('Data', 200)
('data', 192)
('Skill', 166)
('Maharashtra', 166)
('6', 164)
('Python', 156)
('Science', 154)
('I', 146)
('Education', 142)
('College', 140)
('The', 126)
('project', 126)
('like', 126)
('Project', 124)
('Learning', 116)
('India', 114)
('Machine', 112)
('University', 112)
('Web', 106)
('using', 104)
('monthsCompany', 102)
('B', 98)
('C', 98)
('SQL', 96)
('time', 92)
('learning', 90)
('Mumbai', 90)
('Pune', 90)
('Arts', 90)
('A', 84)
('application', 84)
('Engineering', 78)
('24', 76)
('various', 76)
('Software', 76)
('Responsibilities', 76)
('Nagpur', 76)
('development', 74)
('Management', 74)
('projects', 74)
('Technologies', 72)
```

In [21]:
```
wc = WordCloud(colormap='magma').generate(cleanedSentences)
wc = WordCloud().generate(cleanedSentences)
plt.figure(figsize=(15,15))
plt.imshow(wc, interpolation='bilinear')
plt.axis("off")
```

Out[21]:
```
(-0.5, 399.5, 199.5, -0.5)
```

Skill Details
Details January
year monthsCompany
Machine Learning
Details company
Education Details
monthsCompany Details
Deep Learning
Company Details
Exprience Less
British Council
year months
experience
Pvt Ltd
Maharashtra State
Exprience months
Data Science
project
year

Encoding Labels

In [36]:
```python
from sklearn.preprocessing import LabelEncoder
var_mod = ['Category']
le = LabelEncoder()
for i in var_mod:
    resumeDataSet[i] = le.fit_transform(resumeDataSet[i])
```

In [37]:
```python
resumeDataSet.head()
```

Out[37]:

	Category	Resume	cleaned_resume
0	6	Skills * Programming Languages: Python (pandas...	Skills Programming Languages Python pandas num...
1	6	Education Details \r\nMay 2013 to May 2017 B.E...	Education Details May 2013 to May 2017 B E UIT...
2	6	Areas of Interest Deep Learning, Control Syste...	Areas of Interest Deep Learning Control System...
3	6	Skills â□¢ R â□¢ Python â□¢ SAP HANA â□¢ Table...	Skills R Python SAP HANA Tableau SAP HANA SQL ...
4	6	Education Details \r\n MCA YMCAUST, Faridab...	Education Details MCA YMCAUST Faridabad Haryan...

Splitting the dataset

In [30]:
```python
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from scipy.sparse import hstack
```

In [32]:
```python
requiredText = resumeDataSet['cleaned_resume'].values
requiredTarget = resumeDataSet['Category'].values
word_vectorizer = TfidfVectorizer(
    sublinear_tf=True,
    stop_words='english',
    max_features=1500)
word_vectorizer.fit(requiredText)
WordFeatures = word_vectorizer.transform(requiredText)
print ("Featurization Done successfully using TF-IDF")
```

```
Featurization Done successfully using TF-IDF
```

In [33]:
```
X_train,X_test,y_train,y_test = train_test_split(WordFeatures,requiredTarget,random_state=0,
print(X_train.shape)
print(X_test.shape)
```

```
(769, 1500)
(193, 1500)
```

Model Selection

In [34]:
```
clf = OneVsRestClassifier(KNeighborsClassifier())
clf.fit(X_train, y_train)
prediction = clf.predict(X_test)
```

In [35]:
```
print('Accuracy of KNeighbors Classifier on training set: {:.2f}'.format(clf.score(X_train, y
print('Accuracy of KNeighbors Classifier on test set: {:.2f}'.format(clf.score(X_test, y_test
print("\n Classification report for classifier %s:\n%s\n" % (clf, metrics.classification_repo
```

```
Accuracy of KNeighbors Classifier on training set: 0.99
Accuracy of KNeighbors Classifier on test set: 0.99

 Classification report for classifier OneVsRestClassifier(estimator=KNeighborsClassifier()):
              precision    recall  f1-score   support

           0       1.00      1.00      1.00         3
           1       1.00      1.00      1.00         3
           2       1.00      0.80      0.89         5
           3       1.00      1.00      1.00         9
           4       1.00      1.00      1.00         6
           5       0.83      1.00      0.91         5
           6       1.00      1.00      1.00         9
           7       1.00      1.00      1.00         7
           8       1.00      0.91      0.95        11
           9       1.00      1.00      1.00         9
          10       1.00      1.00      1.00         8
          11       0.90      1.00      0.95         9
          12       1.00      1.00      1.00         5
          13       1.00      1.00      1.00         9
          14       1.00      1.00      1.00         7
          15       1.00      1.00      1.00        19
          16       1.00      1.00      1.00         3
          17       1.00      1.00      1.00         4
          18       1.00      1.00      1.00         5
          19       1.00      1.00      1.00         6
          20       1.00      1.00      1.00        11
          21       1.00      1.00      1.00         4
          22       1.00      1.00      1.00        13
          23       1.00      1.00      1.00        15
          24       1.00      1.00      1.00         8

    accuracy                           0.99       193
   macro avg       0.99      0.99      0.99       193
weighted avg       0.99      0.99      0.99       193
```

In [ ]: