ML_Major_project_digit_classification

May 21, 2024

Importing Dependencies

[ ]:
```
# import all necessary libraries

import pandas as pd
import numpy as np

import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.preprocessing import scale
from sklearn.model_selection import train_test_split

from sklearn import metrics
from sklearn.metrics import confusion_matrix
from sklearn.svm import SVC
```

Data collection & Analysis

[ ]:
```
#loading the Data
df = pd.read_csv("/content/digit_svm.csv")
df
```

[ ]:
```
       label  pixel0  pixel1  pixel2  …  pixel780  pixel781  pixel782
pixel783
0          1       0       0       0  …         0         0         0
0
1          0       0       0       0  …         0         0         0
0
2          1       0       0       0  …         0         0         0
0
3          4       0       0       0  …         0         0         0
0
4          0       0       0       0  …         0         0         0
0
…      …     …     …     …  …       …       …       …
…
41995      0       0       0       0  …         0         0         0
0
```

```
41996      1       0       0       0  …         0         0         0
0
41997      7       0       0       0  …         0         0         0
0
41998      6       0       0       0  …         0         0         0
0
41999      9       0       0       0  …         0         0         0
0

[42000 rows x 785 columns]
```

[ ]:
```
df.shape
```

[ ]:
```
(42000, 785)
```

[ ]:
```
# checking for the null values
df.isnull().sum
```

[ ]:
```
<bound method DataFrame.sum of        label  pixel0  pixel1  pixel2  …
pixel780  pixel781  pixel782  pixel783
0      False   False   False   False  …     False     False     False
False
1      False   False   False   False  …     False     False     False
False
2      False   False   False   False  …     False     False     False
False
3      False   False   False   False  …     False     False     False
False
4      False   False   False   False  …     False     False     False
False
…      …     …     …     …  …        …        …       …
…
41995  False   False   False   False  …     False     False     False
False
41996  False   False   False   False  …     False     False     False
False
41997  False   False   False   False  …     False     False     False
False
41998  False   False   False   False  …     False     False     False
False
41999  False   False   False   False  …     False     False     False
False

[42000 rows x 785 columns]>
```

[ ]:
```
# Checking the
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 42000 entries, 0 to 41999
Columns: 785 entries, label to pixel783
dtypes: int64(785)
memory usage: 251.5 MB
```

[ ]:
```
# Brief analysis of the data
df.describe
```

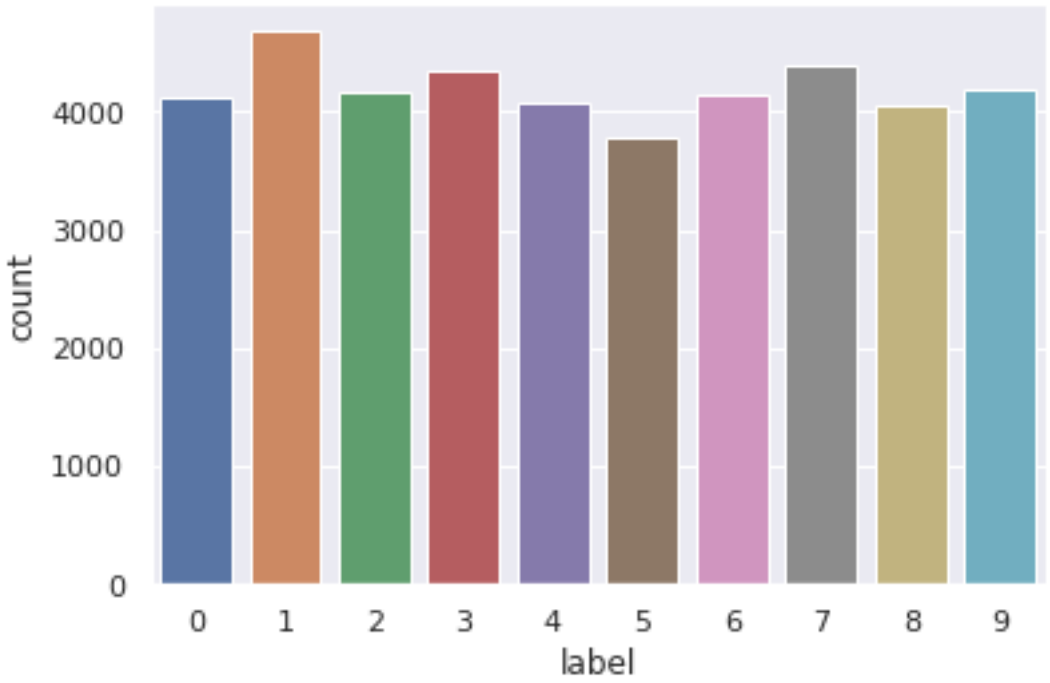
[ ]:
```
<bound method NDFrame.describe of        label  pixel0  pixel1  pixel2  …
pixel780  pixel781  pixel782  pixel783
0          1       0       0       0  …         0         0         0
0
1          0       0       0       0  …         0         0         0
0
2          1       0       0       0  …         0         0         0
0
3          4       0       0       0  …         0         0         0
0
4          0       0       0       0  …         0         0         0
0
…       …      …      …      …  …        …        …        …
…
41995      0       0       0       0  …         0         0         0
0
41996      1       0       0       0  …         0         0         0
0
41997      7       0       0       0  …         0         0         0
0
41998      6       0       0       0  …         0         0         0
0
41999      9       0       0       0  …         0         0         0
0

[42000 rows x 785 columns]>
```

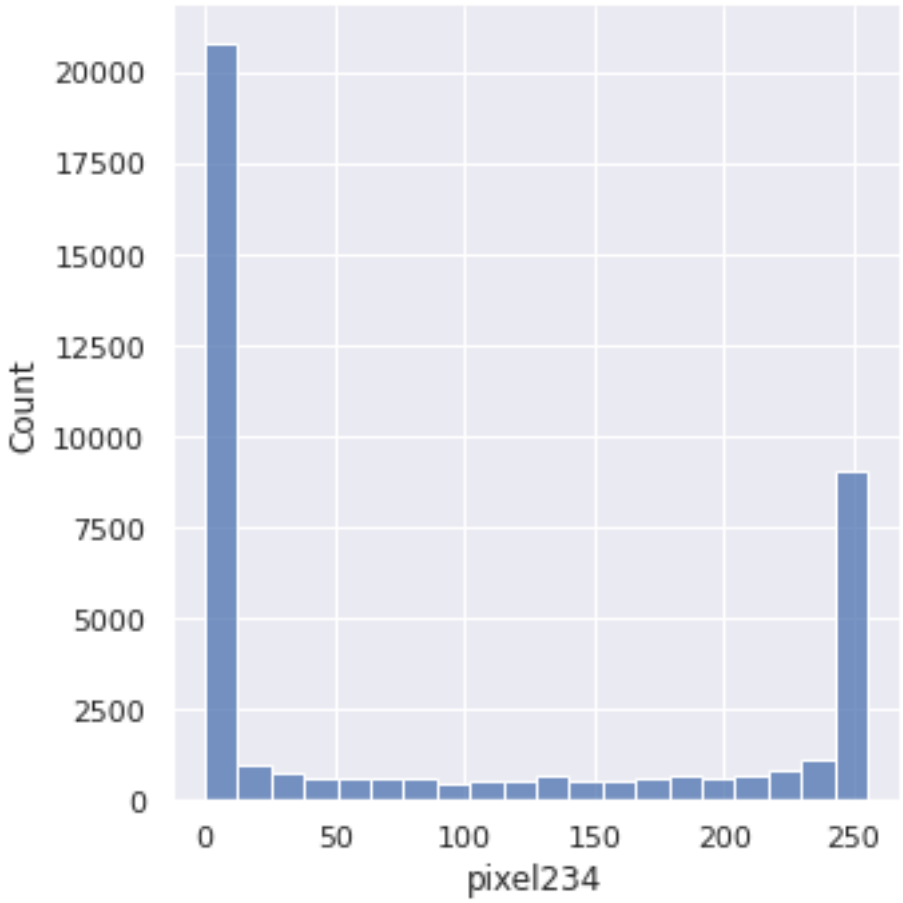
[ ]:
```
# let us check unique entries of label column
np.unique(df['label'])
```

[ ]:
```
array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

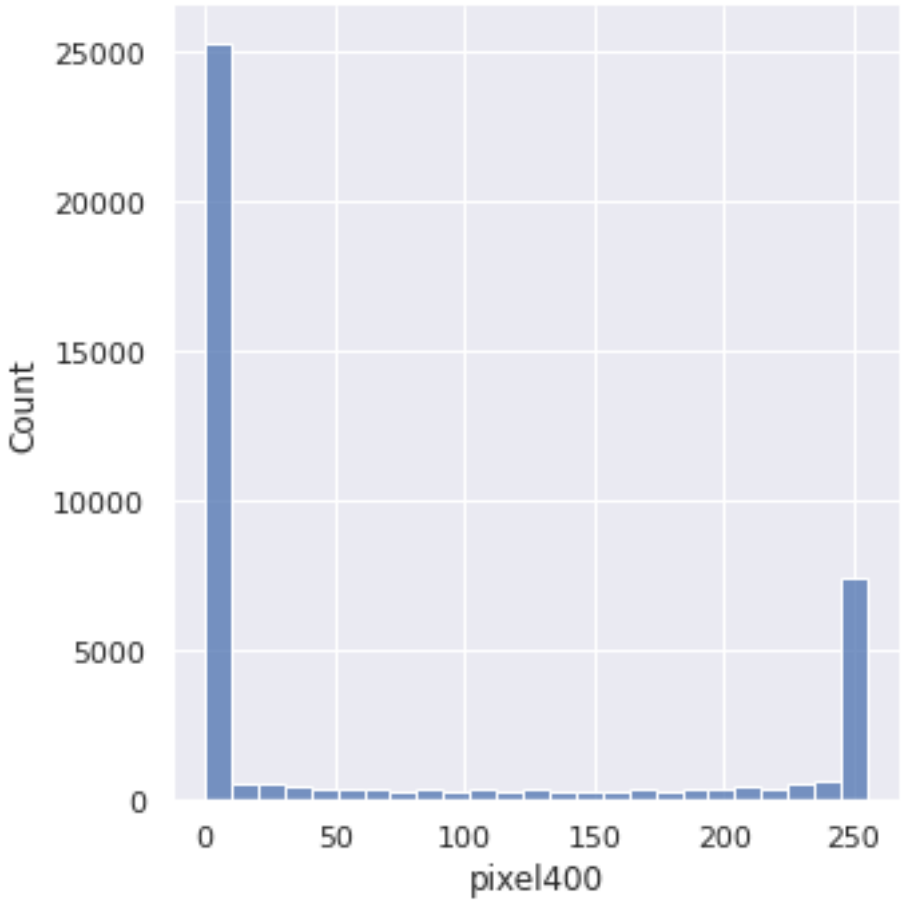
[ ]:
```
sns.countplot(x='label',data=df)
plt.show()
```



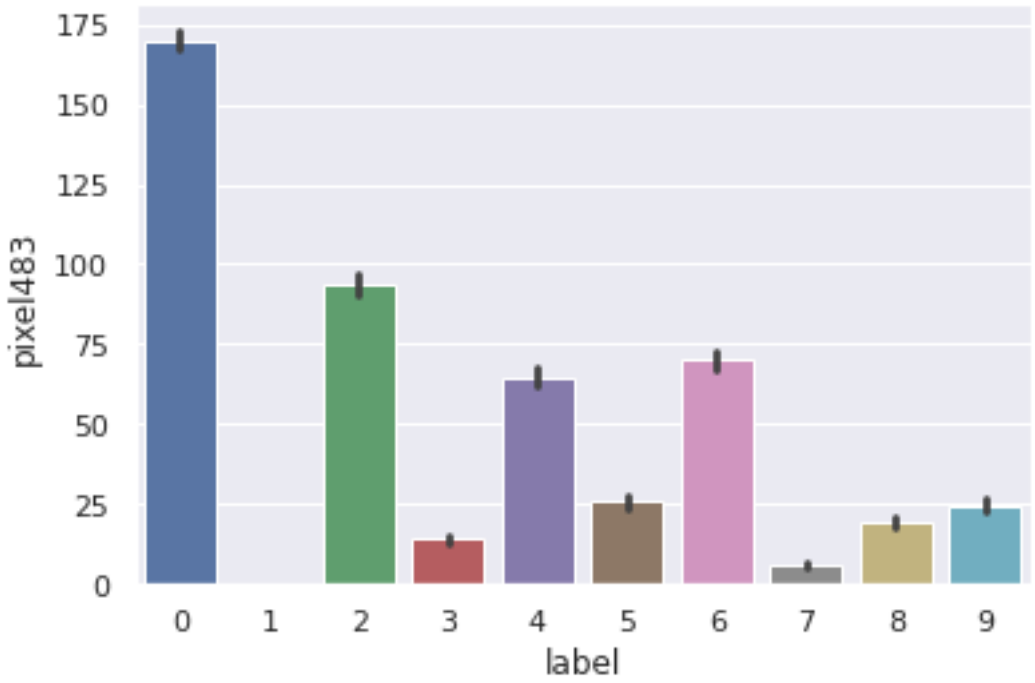
[ ]:
```
sns.displot(df['pixel234'])
plt.show()
```



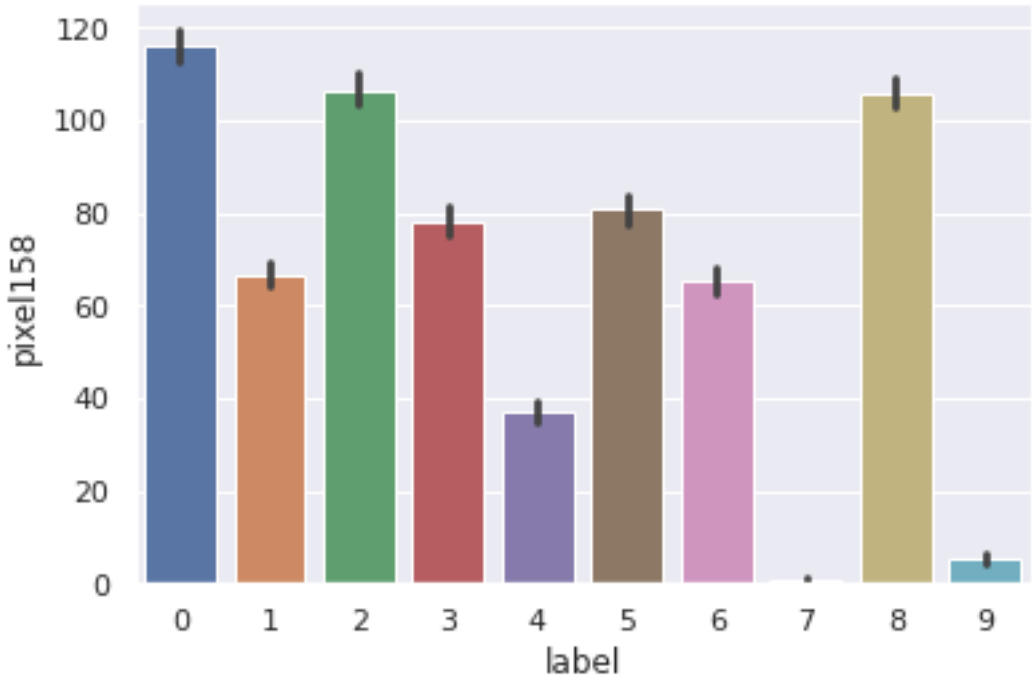
[ ]:
```
sns.displot(df['pixel400'])
plt.show()
```



[ ]:
```
sns.barplot(x='label', y='pixel483', data=df)
plt.show()
```

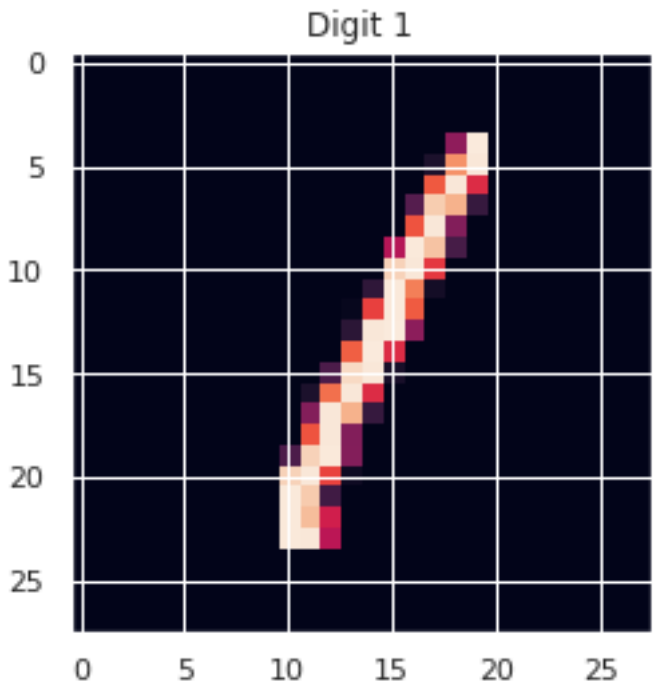


[ ]:
```
sns.barplot(x='label', y='pixel158', data=df)
plt.show()
```



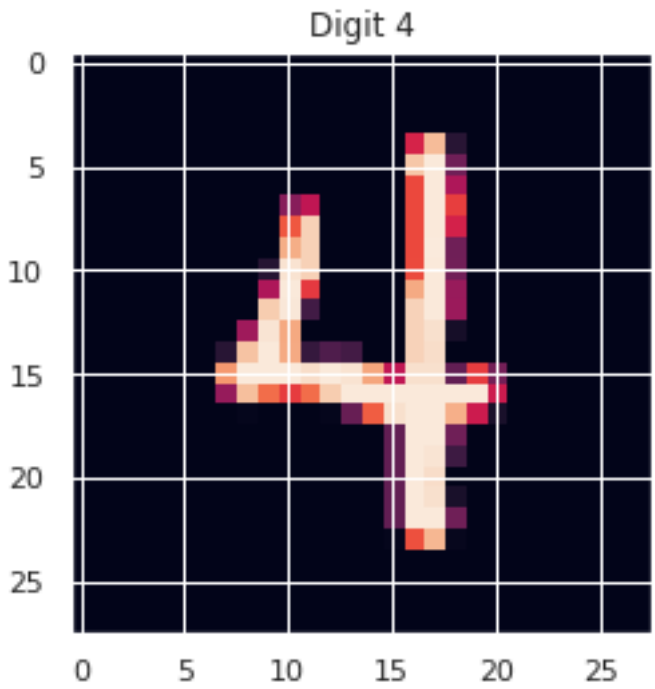
[ ]:
```
one = df.iloc[32456, 1:]
one = one.values.reshape(28,28)
plt.imshow(one)
plt.title("Digit 1")
```

[ ]:
```
Text(0.5, 1.0, 'Digit 1')
```



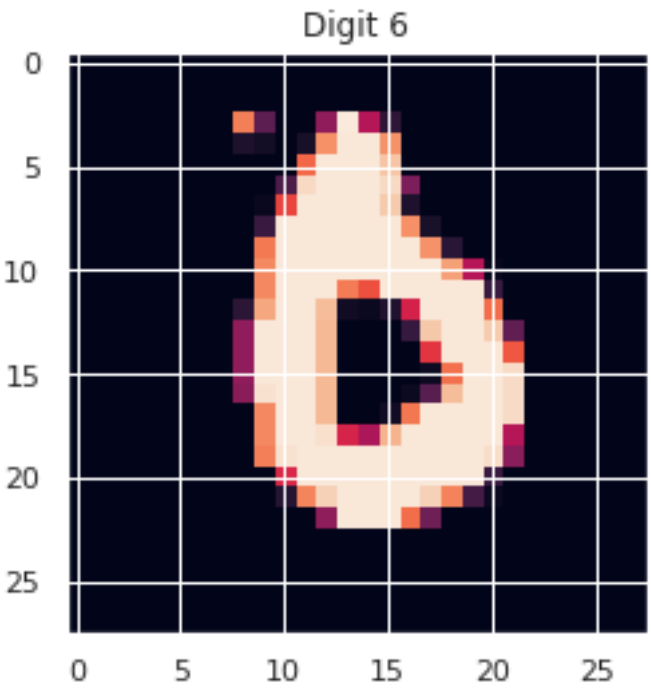
[ ]:
```
four = df.iloc[10600, 1:]
four = four.values.reshape(28,28)
plt.imshow(four)
plt.title("Digit 4")
```

[ ]:
```
Text(0.5, 1.0, 'Digit 4')
```



[ ]:
```
six = df.iloc[35006, 1:]
six = six.values.reshape(28,28)
plt.imshow(six)
plt.title("Digit 6")
```

[ ]: Text(0.5, 1.0, 'Digit 6')



Separating the Data and Labels

[ ]:
```
# splitting into X and y
X = df.drop("label", axis = 1)
y = df['label']
```

[ ]:
```
X
```

[ ]:
```
       pixel0  pixel1  pixel2  pixel3  …  pixel780  pixel781  pixel782
pixel783
0           0       0       0       0  …         0         0         0
0
1           0       0       0       0  …         0         0         0
0
2           0       0       0       0  …         0         0         0
0
3           0       0       0       0  …         0         0         0
0
4           0       0       0       0  …         0         0         0
0
…         …       …       …       …  …         …         …         …
…
41995       0       0       0       0  …         0         0         0
0
41996       0       0       0       0  …         0         0         0
```

```
0
41997       0       0       0       0  …         0         0         0
0
41998       0       0       0       0  …         0         0         0
0
41999       0       0       0       0  …         0         0         0
0

[42000 rows x 784 columns]
```

[ ]:
```
y
```

[ ]:
```
0        1
1        0
2        1
3        4
4        0
        ..
41995    0
41996    1
41997    7
41998    6
41999    9
Name: label, Length: 42000, dtype: int64
```

Data Standardization

Train Test Split

[ ]:
```
# train test split
X_train, X_test, y_train, y_test = train_test_split(X, y, train_size=0.2)
```

[ ]:
```
print('X_train shape:',X_train.shape)
print('y_train shape:',y_train.shape)
print('X_test shape:',X_test.shape)
print('y_test shape:',y_test.shape)
```

```
X_train shape: (8400, 784)
y_train shape: (8400,)
X_test shape: (33600, 784)
y_test shape: (33600,)
```

[ ]:
```
X_test
```

[ ]:
```
       pixel0  pixel1  pixel2  pixel3  …  pixel780  pixel781  pixel782
pixel783
19685       0       0       0       0  …         0         0         0
0
35105       0       0       0       0  …         0         0         0
```

```
0
4879        0       0       0       0  …         0         0         0
0
31939       0       0       0       0  …         0         0         0
0
22035       0       0       0       0  …         0         0         0
0
…       …     …     …     …  …        …        …       …
…
16911       0       0       0       0  …         0         0         0
0
6401        0       0       0       0  …         0         0         0
0
16138       0       0       0       0  …         0         0         0
0
37011       0       0       0       0  …         0         0         0
0
5318        0       0       0       0  …         0         0         0
0

[33600 rows x 784 columns]
```

[ ]:
```
X_train
```

[ ]:
```
       pixel0  pixel1  pixel2  pixel3  …  pixel780  pixel781  pixel782
pixel783
22676       0       0       0       0  …         0         0         0
0
3867        0       0       0       0  …         0         0         0
0
21838       0       0       0       0  …         0         0         0
0
16249       0       0       0       0  …         0         0         0
0
6366        0       0       0       0  …         0         0         0
0
…       …     …     …     …  …        …        …       …
…
18450       0       0       0       0  …         0         0         0
0
24313       0       0       0       0  …         0         0         0
0
17791       0       0       0       0  …         0         0         0
0
24491       0       0       0       0  …         0         0         0
0
33625       0       0       0       0  …         0         0         0
```

```
0

[8400 rows x 784 columns]
```

[ ]:
```
y_train
```

[ ]:
```
22676    7
3867     9
21838    8
16249    5
6366     1
        ..
18450    2
24313    2
17791    8
24491    0
33625    8
Name: label, Length: 8400, dtype: int64
```

[ ]:
```
y_test
```

[ ]:
```
19685    7
35105    0
4879     7
31939    9
22035    0
        ..
16911    0
6401     0
16138    6
37011    4
5318     2
Name: label, Length: 33600, dtype: int64
```

Model Evaluation

Accurancy Score

[ ]:
```
clf= SVC(kernel = 'rbf',random_state=0)
```

[ ]:
```
#traning the the (train_x,train_y) data set
clf.fit(X_train,y_train)
```

[ ]:
```
SVC(C=1.0, break_ties=False, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='scale', kernel='rbf',
    max_iter=-1, probability=False, random_state=0, shrinking=True, tol=0.001,
    verbose=False)
```

[ ]:
```
#predicting the values from the test_x Data set
pred_y = clf.predict(X_test)
```

[ ]:
```
pred_y
```

[ ]:
```
array([7, 0, 7, …, 6, 4, 2])
```

[ ]:
```
y_test
```

[ ]:
```
19685    7
35105    0
4879     7
31939    9
22035    0
        ..
16911    0
6401     0
16138    6
37011    4
5318     2
Name: label, Length: 33600, dtype: int64
```

[ ]:
```
# To check the accuracy_score and Confusion_matrix we import the␣
 ↪(confusion_matrix,accuracy_score) Libraries
from sklearn.metrics import confusion_matrix,accuracy_score
```

[ ]:
```
#confusion matrix checking
confusion_matrix(y_test,pred_y)
```

[ ]:
```
array([[3270,    0,    5,    2,    9,    8,   16,    0,   11,    2],
       [   0, 3683,   27,    8,    8,    3,    3,    6,    9,    7],
       [  16,    6, 3126,   13,   19,    6,   10,   26,   31,    3],
       [   7,   12,   62, 3270,    0,   61,    4,   25,   42,   17],
       [   6,    7,   11,    0, 3120,    0,   13,    4,    2,   85],
       [  10,    9,    5,   59,    9, 2901,   28,    3,   11,    5],
       [  21,    4,    6,    0,    9,   20, 3270,    0,    6,    0],
       [   5,   22,   30,    4,   22,    1,    1, 3419,    6,   66],
       [  10,   20,   13,   26,   19,   25,   18,    8, 3068,   20],
       [  13,    9,   11,   28,   65,   11,    0,   49,   18, 3136]])
```

[ ]:
```
#accuracy score checking
scr=accuracy_score(y_test,pred_y)
```

[ ]:
```
z=y_test-pred_y
z
```

[ ]:
```
19685    0
35105    0
4879     0
31939    0
22035    0
        ..
16911    0
6401     0
16138    0
37011    0
5318     0
Name: label, Length: 33600, dtype: int64
```

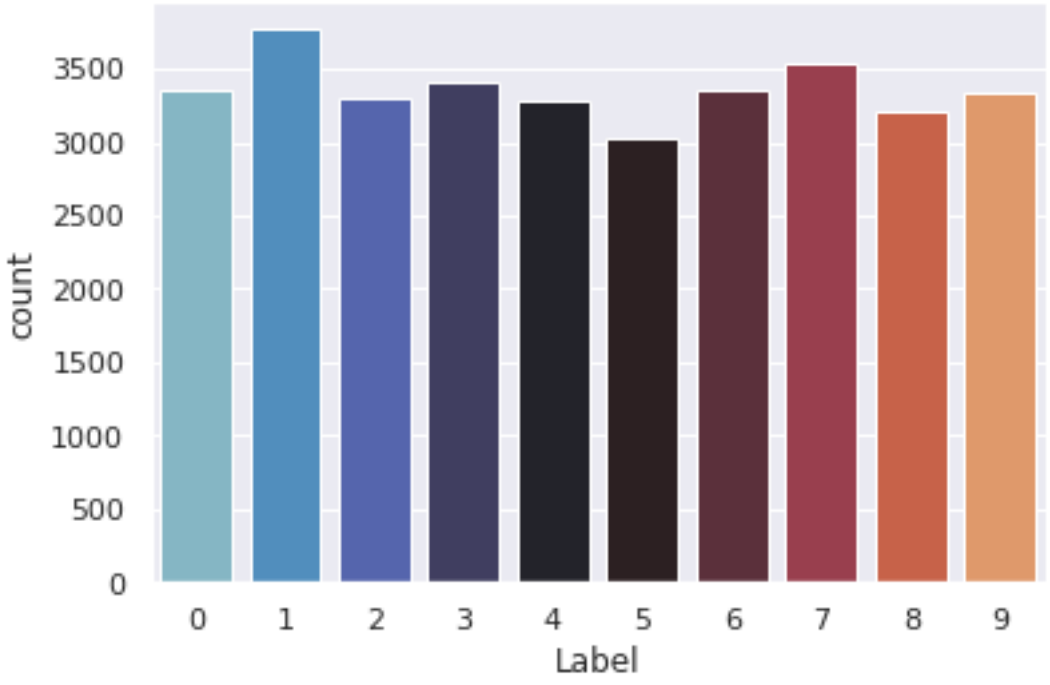
[ ]:
```
print("The accuracy score is " ,scr*100,"%")
```

```
The accuracy score is  96.02083333333333 %
```

Making a Predictive System

[ ]:
```
test_predict = clf.predict(X_test)
```

[ ]:
```
# Plotting the distribution of prediction
a = {'ImageId': np.arange(1,test_predict.shape[0]+1), 'Label': test_predict}
data_to_export = pd.DataFrame(a)
sns.countplot(x=data_to_export['Label'], palette = 'icefire')
plt.show()
```



[ ]:
```
# Exporting the predicted values
data_to_export.to_csv(path_or_buf='output.csv', index=True)
```