Question 2 - Sklearn Handwritten Classifier using Naive Bayes Assumption

In [1]:

```
# Import pandas library.
import pandas as pd

# Reading dataset.
df = pd.read_csv("A_Z_Handwritten_Data.csv")

# print(df)
```

In [2]:

```
X = df.iloc[:,1:]
X = X//128
# X

Y = df.iloc[:,0]
# Y
```

In [3]:

```
from sklearn.model_selection import train_test_split

# Split dataset into training dataset and test dataset
X_train,X_test,y_train,y_test = train_test_split(X,Y,test_size=0.2)
```

In [4]:

```
# # Import Gaussian Naive Bayes model.
# from sklearn.naive_bayes import GaussianNB

# # Create a Gaussian Classifier.
# gnb = GaussianNB()

# # Train the model using the training data set.
# gnb.fit(X_train, y_train)

# # Predict the response for test dataset.
# y_pred = gnb.predict(X_test)

# # Accuracy : 51.14%
```

In [5]:

```
# Import Multinomial Naive Bayes model.
from sklearn.naive_bayes import MultinomialNB

# Create a Multinomial Classifier.
clf = MultinomialNB()

# Train the model using the training data set.
clf.fit(X_train, y_train)

# Predict the response using test dataset.
y_pred = clf.predict(X_test)

# Accuracy : 71.09%
```

In [6]:

```
# # Import Multinomial Naive Bayes model
# from sklearn.naive_bayes import BernoulliNB

# # Create a Bernoulli Classifier.
# Br = BernoulliNB()

# # Train the model using the training data set.
# Br.fit(X_train, y_train)

# # Predict the response using test dataset.
# y_pred = Br.predict(X_test)

# # Accuracy : 70.34%
```

In [7]:

```
# Import metrics module for calculating accuracy.
from sklearn import metrics

# Printing accuracy.
print("Accuracy: ",metrics.accuracy_score(y_test, y_pred))
```

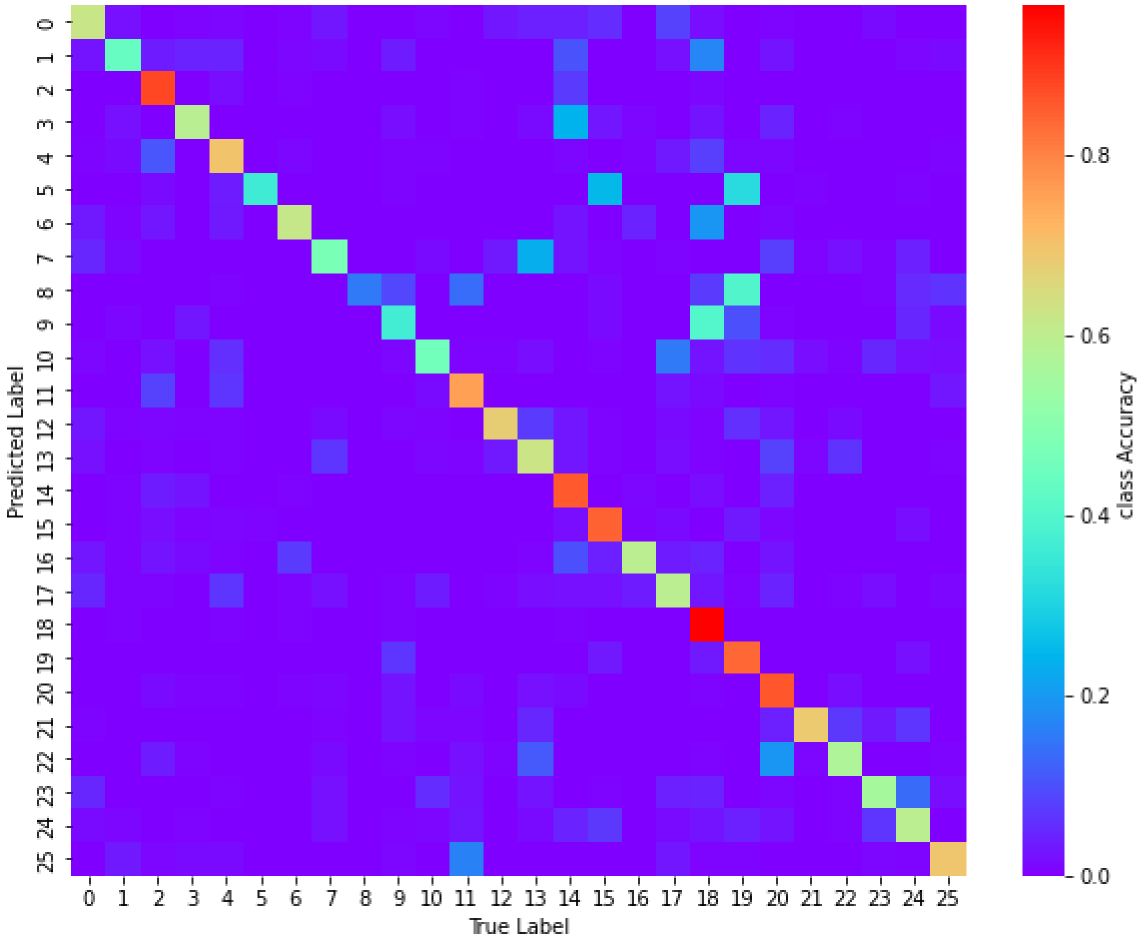
```
Accuracy:  0.7096791515639683
```

In [8]:

```
# Import confusion_matrix module for creating confusion matrix.
from sklearn.metrics import confusion_matrix

# Creating confusion matrix.
conf_matrix = confusion_matrix(y_test, y_pred)
```

In [19]:

```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# Storing the confusion matrix in output.xlsx.
cm = pd.DataFrame(conf_matrix)
# df.to_excel(excel_writer = "output.xlsx")

# Plot confusion matrix
cm = np.array([ cm[i]/sum(cm[i]) for i in range(len(cm))])
plt.figure( figsize=(10,8))
sns.heatmap(cm, cmap='rainbow',cbar_kws={'label': 'class Accuracy '})
plt.xlabel('True Label')
plt.ylabel('Predicted Label')
plt.show()
```



In [29]:

```
# A = { y:{ key:0 for key in range()} for y in range(26)}
# A
```

In [30]:

```
# a = [ i for i in range(10)]
# a
```

In [ ]:

```
```