**NAME: SATWIK REDDY MOPURU**

**ENROLLMENT NUMBER: 700740060**

**1.Implement Naïve Bayes method using scikit-learn library**

Use dataset available with name glass

Use train\_test\_split to create training and testing part

Evaluate the model on test part using score and

**2. Implement linear SVM method using scikit-learn**

Use the same dataset above

Use train\_test\_split to create training and testing part

Evaluate the model on test part using score and

classification\_report(y\_true, y\_pred)

Which algorithm you got better accuracy? Can you justify why?

CODE:

**import** numpy **as** np

**import** pandas **as** pd

**import** urllib

**import** sklearn

**from** sklearn.svm **import** SVC

**from** sklearn.naive\_bayes **import** BernoulliNB

**from** sklearn.naive\_bayes **import** GaussianNB

**from** sklearn.naive\_bayes **import** MultinomialNB

**from** sklearn.model\_selection **import** train\_test\_split

**from** sklearn **import** metrics

**from** sklearn.metrics **import** accuracy\_score

**import** time

**import** warnings

warnings**.**filterwarnings("ignore")

*# read the data*

data **=** pd**.**read\_csv('C:/Users/shrin/OneDrive/Desktop/NNDL/NNDL\_Code and Data/NNDL\_Code and Data/glass.csv')

print(data**.**shape)

X\_train, X\_test **=** train\_test\_split(

data, test\_size**=**0.2, random\_state**=**int(time**.**time()))

*# features columns*

features **=** ["RI", "Na", "Mg", "Al", "Si", "K", "Ca", "Ba", "Fe"]

*# Naïve Bayes Classifier*

gauss **=** GaussianNB()

*# train the classifier*

gauss**.**fit(

X\_train[features]**.**values,

X\_train["Type"]

)

*# make predictions*

y\_pred **=** gauss**.**predict(X\_test[features])

print("Naïve Bayes\nTotal number of points: {}\nMislabeled points : {}\nAccuracy {:05.2f}%"

**.**format(

X\_test**.**shape[0],

(X\_test["Type"] **!=** y\_pred)**.**sum(),

100 **\*** (1 **-** (X\_test["Type"] **!=** y\_pred)**.**sum() **/** X\_test**.**shape[0])

))

print("\n")

*# Naïve Bayes Classifier performance*

print(metrics**.**classification\_report(X\_test["Type"], y\_pred))

*# Linear Support Vector Classification*

svc\_linear **=** SVC(kernel**=**'linear')

*# train linear SVM model*

svc\_linear**.**fit(

X\_train[features]**.**values,

X\_train["Type"]

)

Y\_pred **=** svc\_linear**.**predict(X\_test[features])

*# Linear SVM Model performance*

acc\_svc **=** round(svc\_linear**.**score(

X\_test[features]**.**values, X\_test["Type"]) **\*** 100, 2)

print("Linear SVM accuracy is:", acc\_svc)

*# Support vector classifier (SVC) with the radial basis function kernel (RBF)*

svc\_rbf **=** SVC(kernel**=**'rbf')

svc\_rbf**.**fit(

X\_train[features]**.**values,

X\_train["Type"]

)

*# model predictions*

Y\_pred **=** svc\_rbf**.**predict(X\_test[features])

*# SVM RBF Model performance*

acc\_svc **=** round(svc\_rbf**.**score(

X\_test[features]**.**values, X\_test["Type"]) **\*** 100, 2)

print("SVM RBF model accuracy is:", acc\_svc)

print("\n")

print(metrics**.**classification\_report(X\_test["Type"], Y\_pred))

(214, 10)

Naïve Bayes

Total number of points: 43

Mislabeled points : 21

Accuracy 51.16%

precision recall f1-score support

1 0.46 0.93 0.62 14

2 0.57 0.22 0.32 18

3 0.50 0.50 0.50 2

5 1.00 0.25 0.40 4

6 1.00 0.50 0.67 2

7 0.50 0.67 0.57 3

accuracy 0.51 43

macro avg 0.67 0.51 0.51 43

weighted avg 0.59 0.51 0.47 43

Linear SVM accuracy is: 60.47

SVM RBF model accuracy is: 41.86

precision recall f1-score support

1 0.00 0.00 0.00 14

2 0.42 1.00 0.59 18

3 0.00 0.00 0.00 2

5 0.00 0.00 0.00 4

6 0.00 0.00 0.00 2

7 0.00 0.00 0.00 3

accuracy 0.42 43

macro avg 0.07 0.17 0.10 43

weighted avg 0.18 0.42 0.25 43

Firstly, the libraries were imported such as pandas, sklearn.svm, sklearn.naive\_bayes. Then, read the provided dataframe from read() function and printed. The dataframe is splitted into train & test. Now feature the coloumns of the dataframe. Implement the naïve bayes classifier & train it. Now, the naïve bayes classifier is predicted and the accuracy is printed. Now, the naïve bayes classifier performance is printed. Implement the linear support vector classification & train it. Now, the performance and the accuracy is printed.

Then Support vector classifier (SVC) with the radial basis function kernel (RBF) is trained and predicted. Now, the performance and accuracy is printed.

By Comparing both the Naïve Bayes method and the linear SVM method, the accuracy of Naïve Bayes method is more with 58.14% This is because as they both are parameter optimization. Naïve bayes treats them as independent, whereas SVM looks at the interactions between them to a certain degree, as long as you’re using a non-linear kernel.