**ASSINGMENT 5**

**Name: Bhuvana Nandhimalla**

**ID: 700744203**

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

import pandas as pd

from sklearn.svm import SVC

from sklearn.naive\_bayes import GaussianNB

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

import time

import warnings

warnings.filterwarnings("ignore")

from sklearn import metrics

data = pd.read\_csv('C:/Users/Pavanisodar/Desktop/glass.csv')

print(data.shape)

X\_train, X\_test = train\_test\_split(data, test\_size=0.2, random\_state=int(time.time()))

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

gauss = GaussianNB()

gauss.fit(X\_train[features].values,X\_train["Type"])

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

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

.format(

X\_test.shape[0],

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

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

))

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

**Description**: Here we applied the naïve bayes approach using the scikit-learn package. Using train test split to create a training and testing portion, and created a dataset with the name glass that is now available. And then used score to evaluate the model on the test portion.

Text

Description automatically generatedGraphical user interface, text

Description automatically generated

**Question 2 : Implement linear SVM method using scikit library Use the same dataset above Use train\_test\_split to create training and testing part Evaluate the model on test part using score and classificationn\_report(y\_true, y\_pred) Which algorithm you got better accuracy? Can you justify why?**

import pandas as pd

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

from sklearn.svm import SVC

from sklearn.metrics import classification\_report, accuracy\_score

glass\_data = pd.read\_csv('glass.csv')

x\_train = glass\_data.drop("Type", axis=1)

y\_train = glass\_data['Type']

# splitting train and test data using train\_test\_split

x\_train, x\_test, y\_train, y\_test = train\_test\_split(x\_train, y\_train, test\_size=0.2, random\_state=0)

# Train the model using the training sets

svc = SVC()

svc.fit(x\_train, y\_train)

y\_pred = svc.predict(x\_test)

# Classification report

qual\_report = classification\_report(y\_test, y\_pred, zero\_division = 0)

print(qual\_report)

print("SVM accuracy is: ", accuracy\_score(y\_test, y\_pred)\*100)

**Description:** Here we use Scikit to implement the linear svm algorithm. The training and testing portions were created using the same dataset and train test split. Use the test section score to evaluate the model.

Graphical user interface, text

Description automatically generated

**Justification:** The accuracy of the Gaussian algorithm is higher. Since our accuracy after training with gaussian is higher than that of svm.

Video Link: https://drive.google.com/file/d/1c7IZjqMQLFdd0PNQwVO-OTQLdy7DZBS9/view?usp=sharing

Source code: <http://localhost:8888/notebooks/Documents/Neural/NNDL_ICP_5.ipynb>

Github: https://github.com/BhuvanaNandhimalla/NeuralNetworks\_Assignment5