

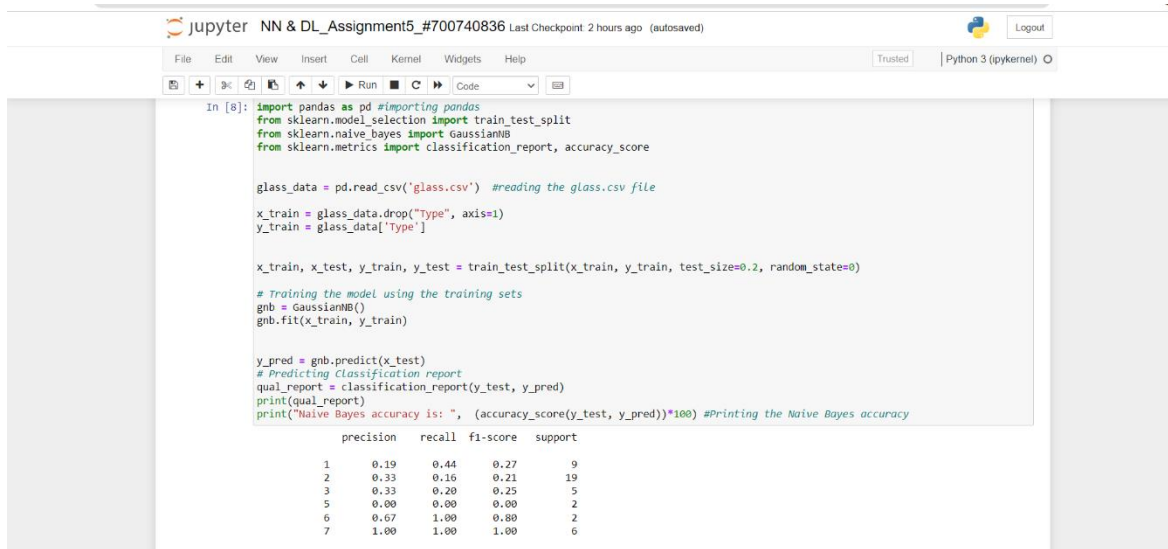
Neural Networks & Deep Learning

Assignment-5

Name : Ajay Mohan Reddy Veluru

ID :700740836

Que:1



The screenshot shows a Jupyter Notebook interface with the title 'jupyter NN & DL_Assignment5_#700740836'. The code in the cell imports pandas as 'pd', sklearn.model_selection as 'train_test_split', sklearn.naive_bayes as 'GaussianNB', and sklearn.metrics as 'classification_report' and 'accuracy_score'. It reads 'glass.csv', drops the 'Type' column, and splits the data. A Gaussian Naive Bayes model is trained and used to predict the test set. The output shows a classification report and the Naive Bayes accuracy.

```
In [8]: import pandas as pd #importing pandas
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import classification_report, accuracy_score

glass_data = pd.read_csv('glass.csv') #reading the glass.csv file

x_train = glass_data.drop("Type", axis=1)
y_train = glass_data["Type"]

x_train, x_test, y_train, y_test = train_test_split(x_train, y_train, test_size=0.2, random_state=0)

# Training the model using the training sets
gnb = GaussianNB()
gnb.fit(x_train, y_train)

y_pred = gnb.predict(x_test)
# Predicting classification report
qual_report = classification_report(y_test, y_pred)
print(qual_report)
print("Naive Bayes accuracy is: ", (accuracy_score(y_test, y_pred))*100) #Printing the Naive Bayes accuracy
```

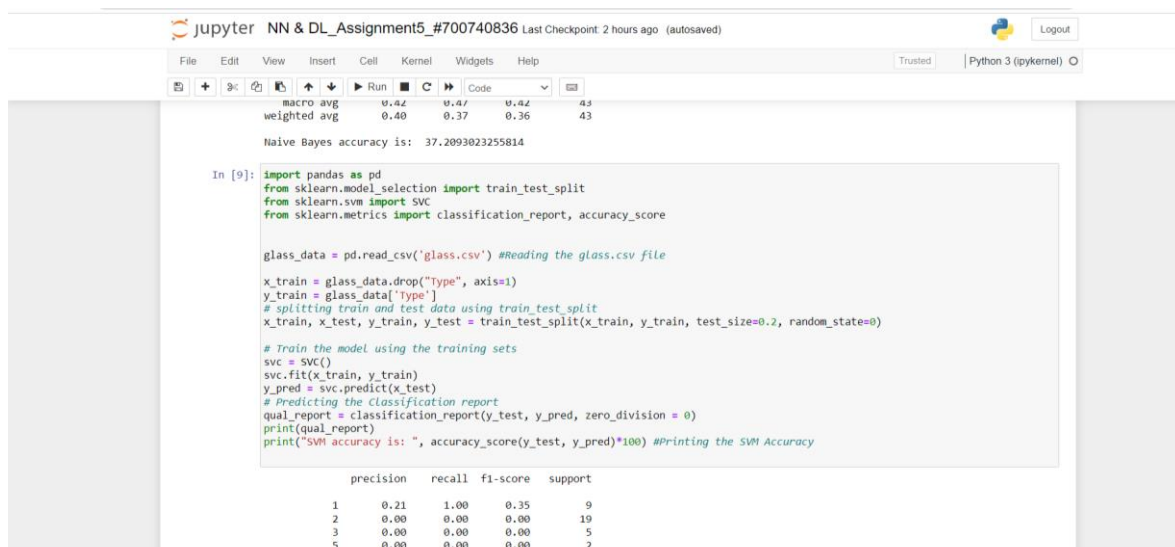
	precision	recall	f1-score	support
1	0.19	0.44	0.27	9
2	0.33	0.16	0.21	19
3	0.33	0.20	0.25	5
5	0.00	0.00	0.00	2
6	0.67	1.00	0.80	2
7	1.00	1.00	1.00	6

-Importing pandas as pd here.

- Reading the glass.csv file giving the type.

-After using the training sets we will get the navie bayes accuracy.

Que -2



The screenshot shows a Jupyter Notebook interface with the title 'jupyter NN & DL_Assignment5_#700740836'. The code imports pandas as 'pd', sklearn.model_selection as 'train_test_split', sklearn.svm as 'SVC', and sklearn.metrics as 'classification_report' and 'accuracy_score'. It reads 'glass.csv', drops the 'Type' column, and splits the data. An SVM model is trained and used to predict the test set. The output shows a classification report and the SVM accuracy.

```
In [9]: 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') #Reading the glass.csv file

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)
# Predicting the 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) #Printing the SVM Accuracy
```

	precision	recall	f1-score	support
1	0.21	1.00	0.35	9
2	0.00	0.00	0.00	19
3	0.00	0.00	0.00	5
5	0.00	0.00	0.00	2

-In this one we are doing the same thing Using the Scikit package implementing the SVM Technique.

-Utilizing the same data set as before.

-To establish a training and testing component, using the train split method here.

-Everything is same according to the implementation part expect the fact that we are using the linear SVC class from sklearn.svm

Comparison: In order to compare the both methods performance, We can simply do that with the help of the accuracy scores achieved from the algorithms. In this case, Navie bayes is better than the SVM Method because of its accuracy achieved. However, It all depends on the data given to the algorithms.

Github Link: https://github.com/Ajay-Reddy564/NN-DL_Assignment5_-700740836

Video Link : <https://www.loom.com/share/55f83a2da12f4067b19beec56888b411>