

Spring 2024 CS5720

Neural Networks & Deep Learning - Assignment 5

Name: Sravya Reddy Pilli

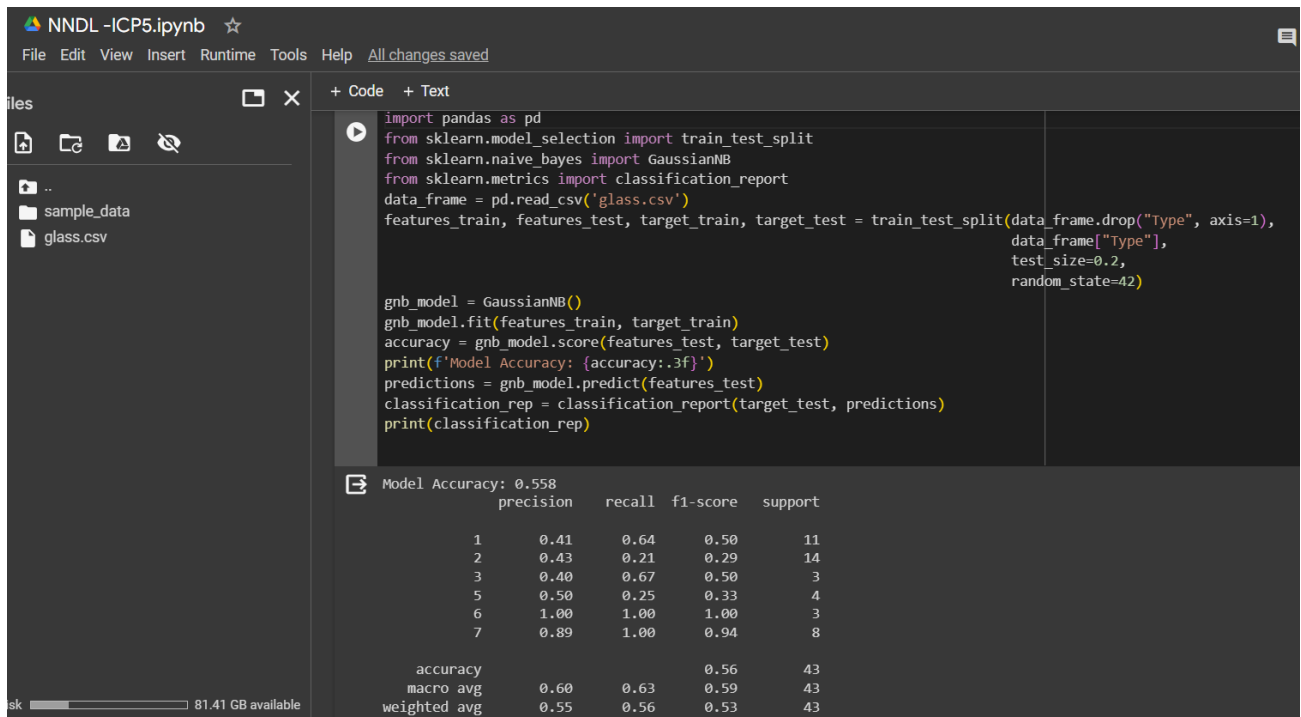
Student Id: 700747154

Github link : <https://github.com/09sravyareddy/NNDL-ICP5>

Recording Link :

<https://drive.google.com/file/d/1Z3I68YmTT4uEOLpTiFmN60upwksnrIJW/view?usp=sharing>

- 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.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import classification_report
data_frame = pd.read_csv('glass.csv')
features_train, features_test, target_train, target_test = train_test_split(data_frame.drop("Type", axis=1),
                                                                            data_frame["Type"],
                                                                            test_size=0.2,
                                                                            random_state=42)

gnb_model = GaussianNB()
gnb_model.fit(features_train, target_train)
accuracy = gnb_model.score(features_test, target_test)
print(f'Model Accuracy: {accuracy:.3f}')
predictions = gnb_model.predict(features_test)
classification_rep = classification_report(target_test, predictions)
print(classification_rep)
```

Model Accuracy: 0.558

	precision	recall	f1-score	support
1	0.41	0.64	0.50	11
2	0.43	0.21	0.29	14
3	0.40	0.67	0.50	3
5	0.50	0.25	0.33	4
6	1.00	1.00	1.00	3
7	0.89	1.00	0.94	8
accuracy			0.56	43
macro avg	0.60	0.63	0.59	43
weighted avg	0.55	0.56	0.53	43

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 classification_report(y_true, y_pred)

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.svm import LinearSVC
from sklearn.metrics import classification_report

glass_data = pd.read_csv('glass.csv')

features_train, features_test, labels_train, labels_test = train_test_split(
    glass_data.drop(columns=["Type"]), glass_data["Type"], test_size=0.2, random_state=123)
svc_model = LinearSVC(dual=False)
svc_model.fit(features_train, labels_train)

model_accuracy = svc_model.score(features_test, labels_test)
print(f'Model Accuracy: {model_accuracy:.3f}')

predicted_labels = svc_model.predict(features_test)
detailed_report = classification_report(labels_test, predicted_labels)
print(detailed_report)
```

```
Model Accuracy: 0.698
      precision    recall  f1-score   support

     1       0.67       0.36       0.47        11
     2       0.54       1.00       0.70        13
     3       0.00       0.00       0.00         3
     5       1.00       0.33       0.50         3
     6       1.00       0.50       0.67         2
     7       1.00       1.00       1.00        11
```

```
accuracy          0.70         43
macro avg         0.70         0.53         0.56         43
weighted avg      0.71         0.70         0.65         43
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision and F-score are ill-defined ar
_warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision and F-score are ill-defined ar
_warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision and F-score are ill-defined ar
_warn_prf(average, modifier, msg_start, len(result))
```


2)