Neural Networks & Deep Learning: ICP5

1. Implement Naïve Bayes method using scikit-learn library

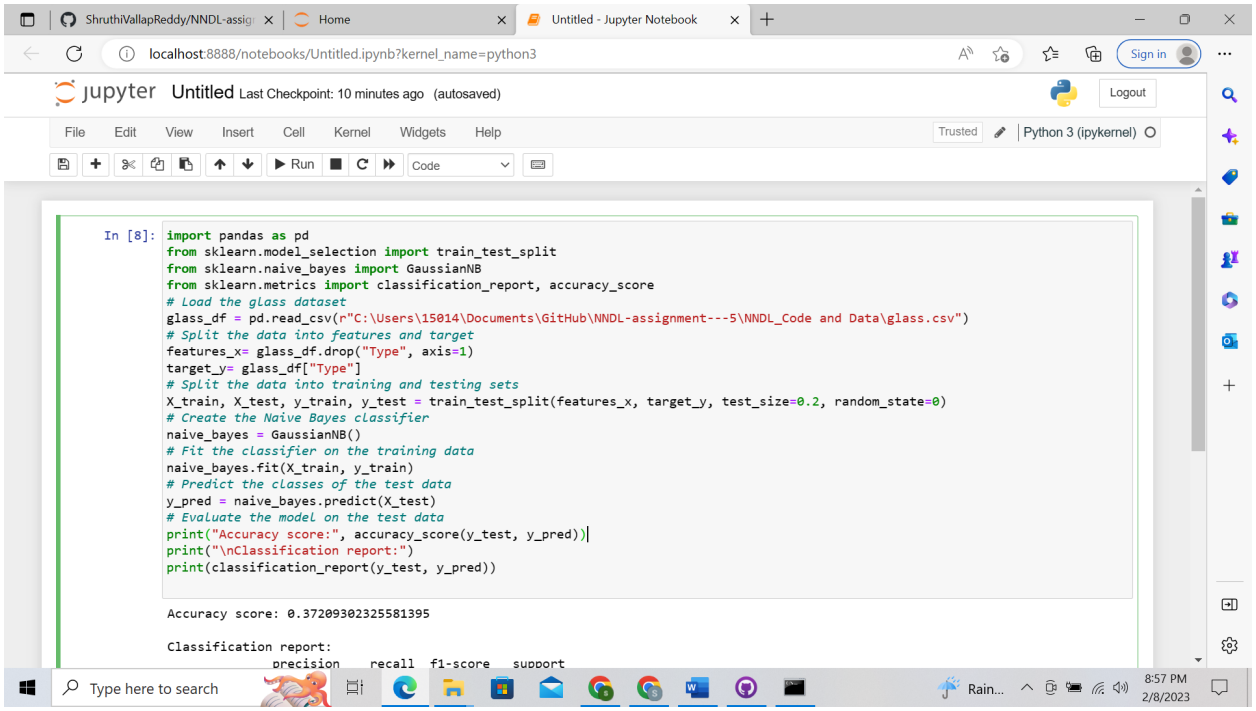
Use dataset available with name glass

Use train_test_split to create training and testing part

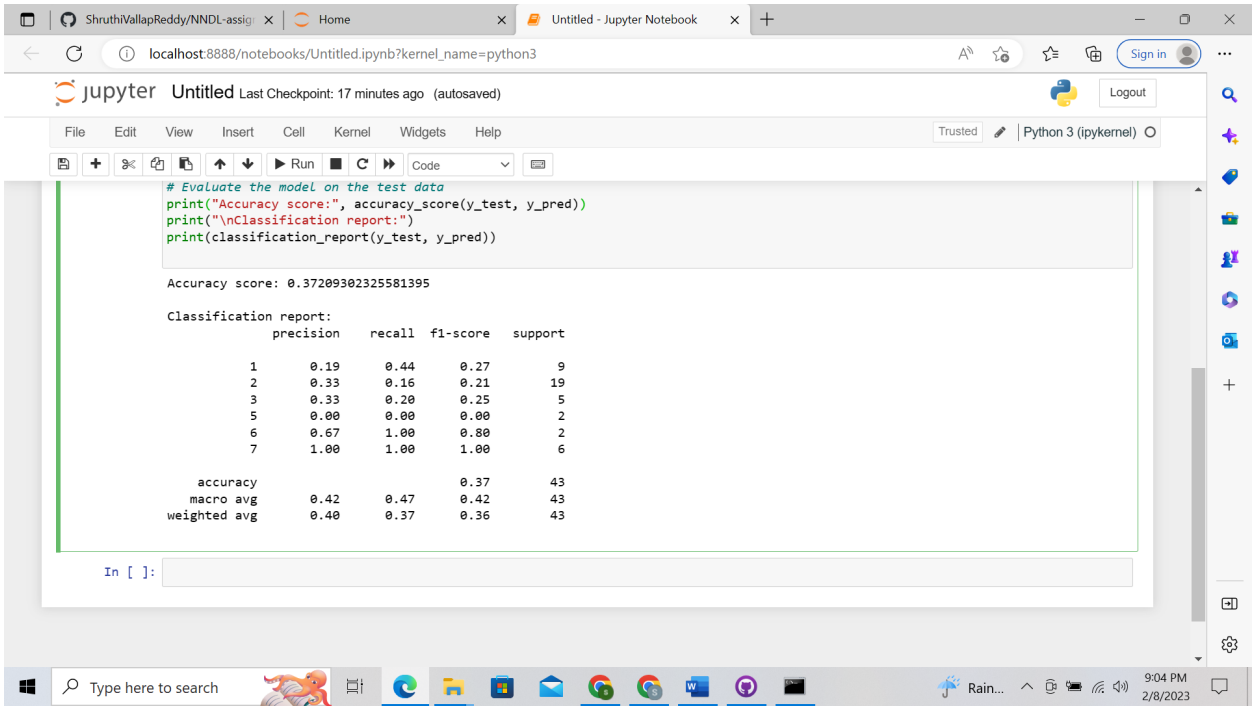
Evaluate the model on test part using score and

classification_report(y_true, y_pred)

Ans



1)Load the glass dataset using pd.read_csv() function and store in the variable.
2)Split the data in to features and target using drop() function
3)Split the dataset into training and testing sets using train_test_split() function.
4)Created the Naive Bayes Classifier using GuassianNB().
5)Fit the classifier on the training data using fit() function.
6)Predict the test data using predict() function.
7)Calculated the accuracy score using accuracy_score() function.
8)Generated the Classification report using classification_report() function.


ShruthiVallapReddy/NNDL-assig
Home
Untitled - Jupyter Notebook
localhost:8888/notebooks/Untitled.ipynb?kernel_name=python3
Sign in
jupyter Untitled Last Checkpoint: 17 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
# Evaluate the model on the test data
print("Accuracy score:", accuracy_score(y_test, y_pred))
print("\nClassification report:")
print(classification_report(y_test, y_pred))
Accuracy score: 0.37209302325581395
Classification report:
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
accuracy 0.37 43
macro avg 0.42 0.47 0.42 43
weighted avg 0.40 0.37 0.36 43
In [ ]:
Type here to search
Rain...
9:04 PM
2/8/2023

2. Implement linear SVM method using scikit library

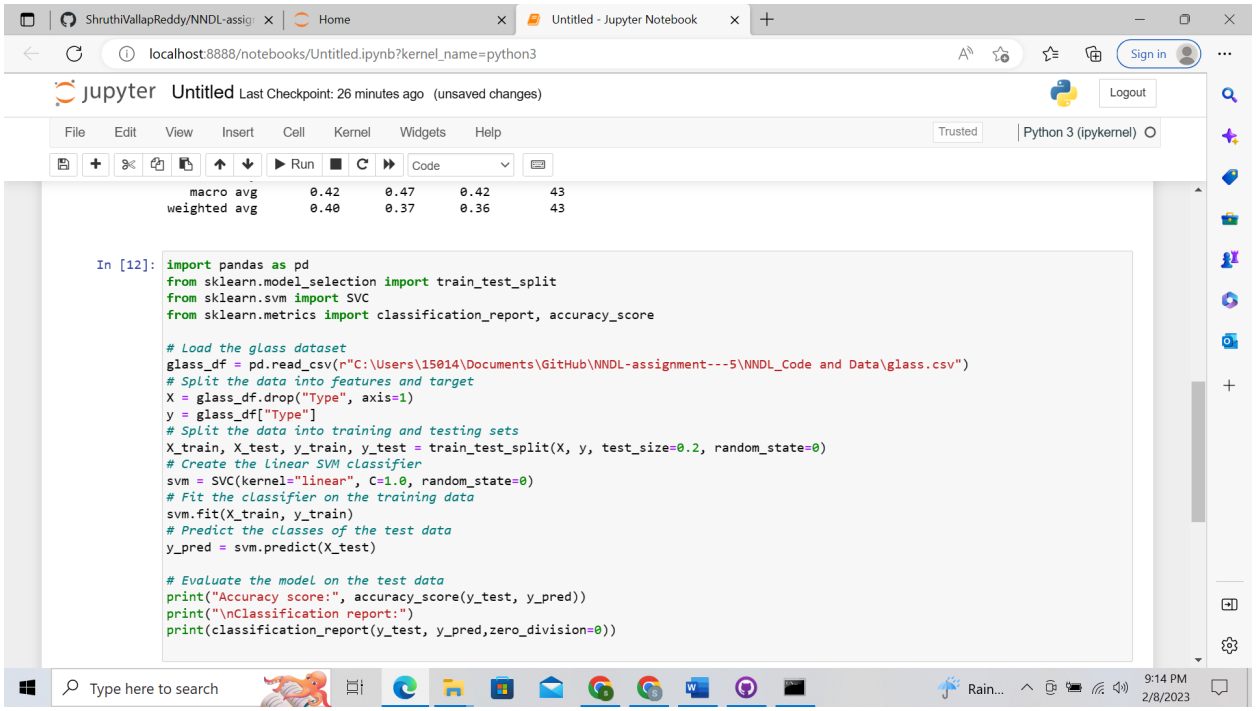
Use the same dataset above

Use train_test_split to create training and testing part

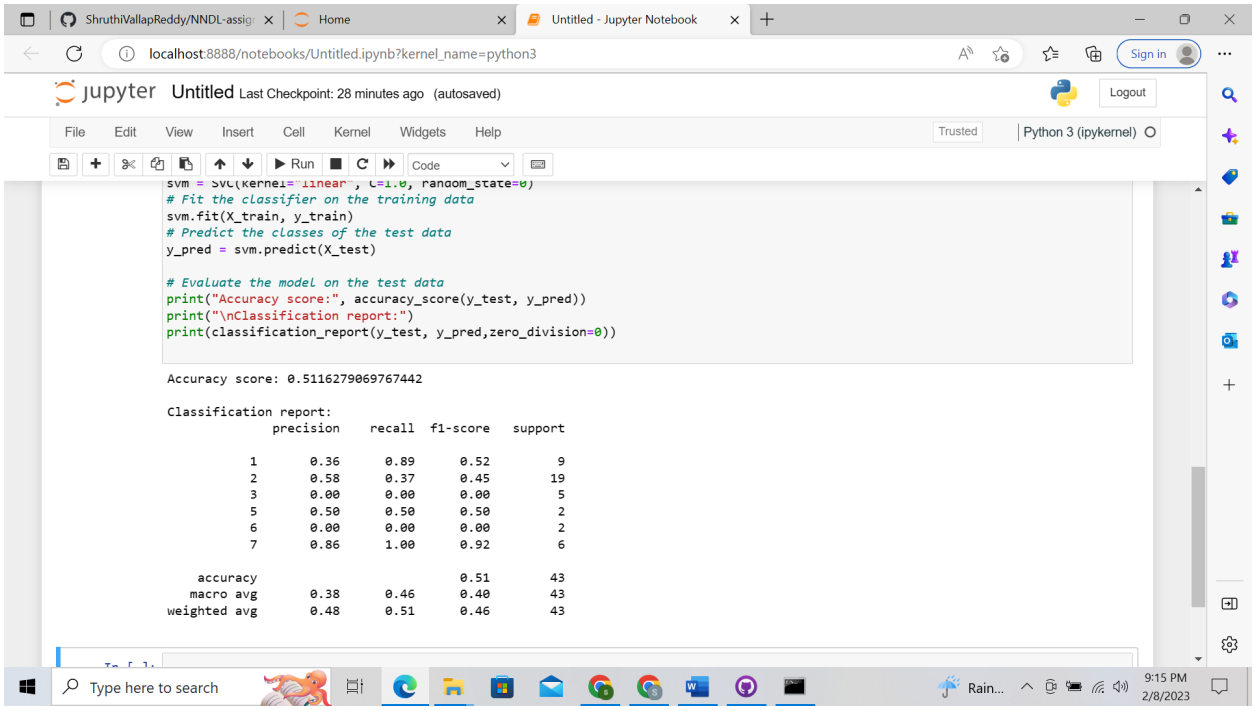
Evaluate the model on test part using score and

classification_report(y_true, y_pred)

Ans



1)Load the glass dataset using pd.read_csv() function and store in the variable.
2)Split the data in to features and target using drop() function
3)Split the dataset into training and testing sets using train_test_split() function.
4)Created the Linear SVM classifier using SVC(kernel=”linear”).
5)Fit the classifier on the training data using fit() function.
6)Predict the test data using predict() function.
7)Calculated the accuracy score using accuracy_score() function.
8)Generated the Classification report using classification_report() function.



Which algorithm you got better accuracy? Can you justify why?
Ans
1) Based on the accuracy scores , the linear SVM method has a better accuracy score compared to the Naive Bayes method.
2)The accuracy score of 0.51 for the linear SVM method indicates that it correctly predicted the target class for 51% of the instances in your data.
3) On the other hand, the accuracy score of 0.37 for the Naive Bayes method indicates that it correctly predicted the target class for only 37% of the instances in your data.

Git hub repo link : https://github.com/ShruthiVallapReddy/NNDL-assignment---5.git