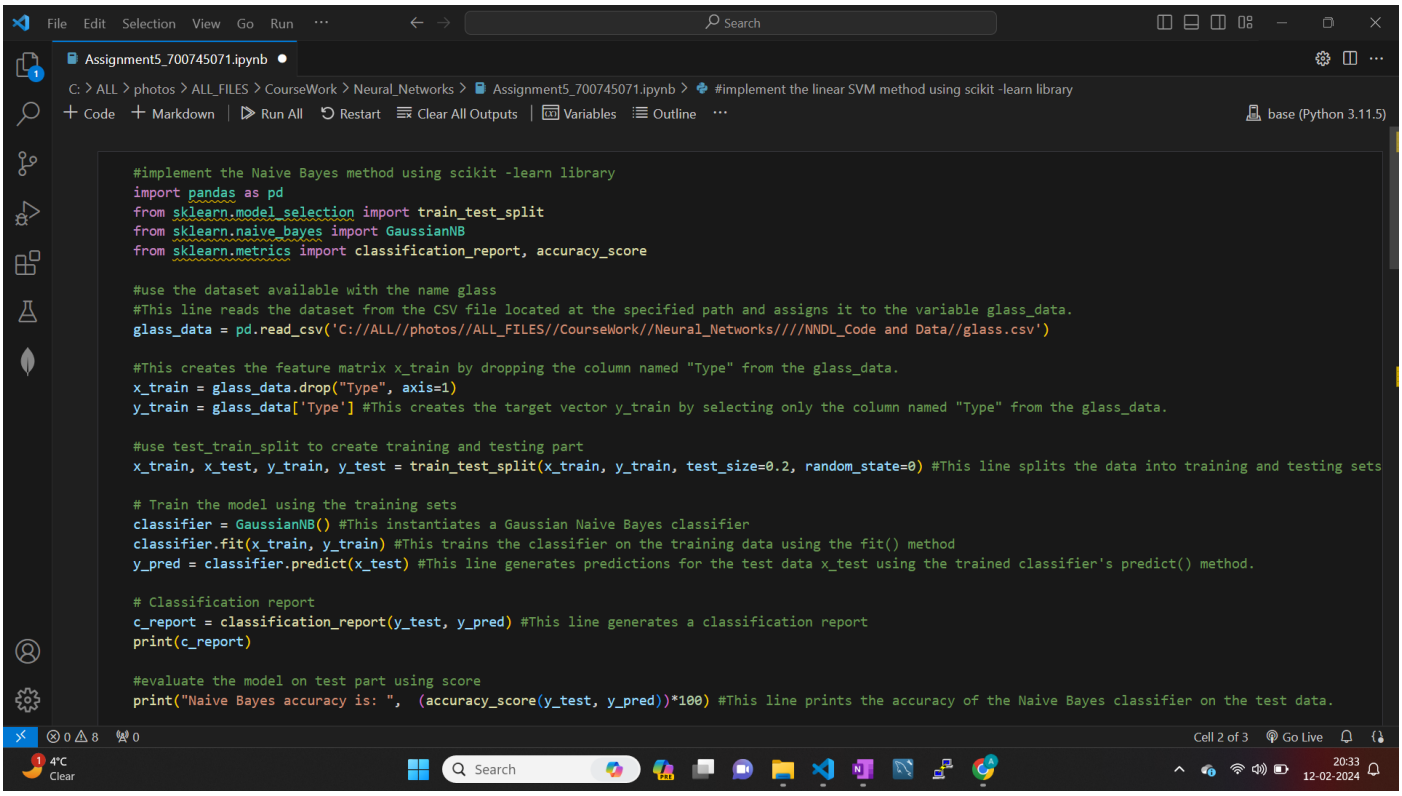
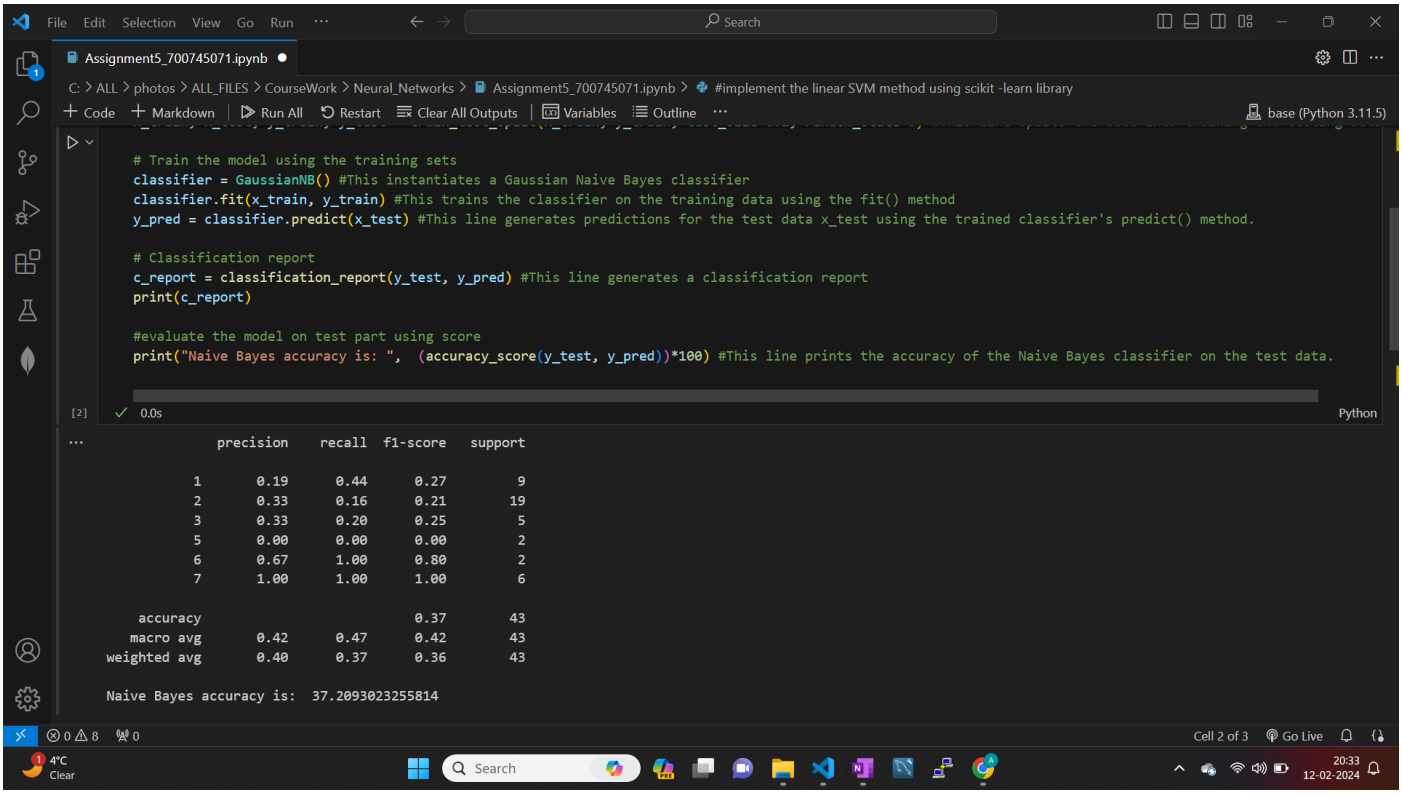
Github link: https://github.com/naraanjali/Assignment5.git

Vido link: https://drive.google.com/file/d/1T6_7FTk90wksZ893utgxmYFsyLR9dDTW/view?usp=sharing

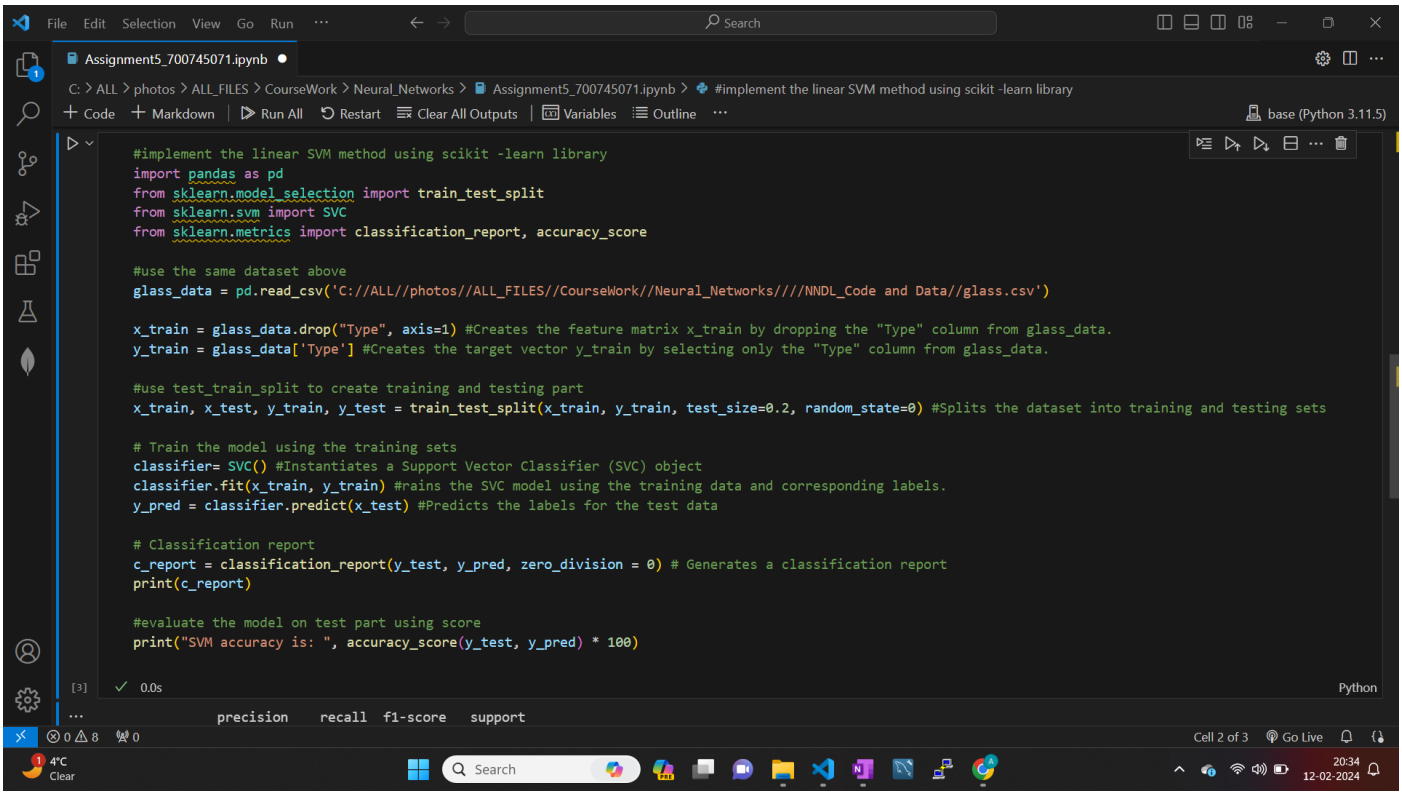


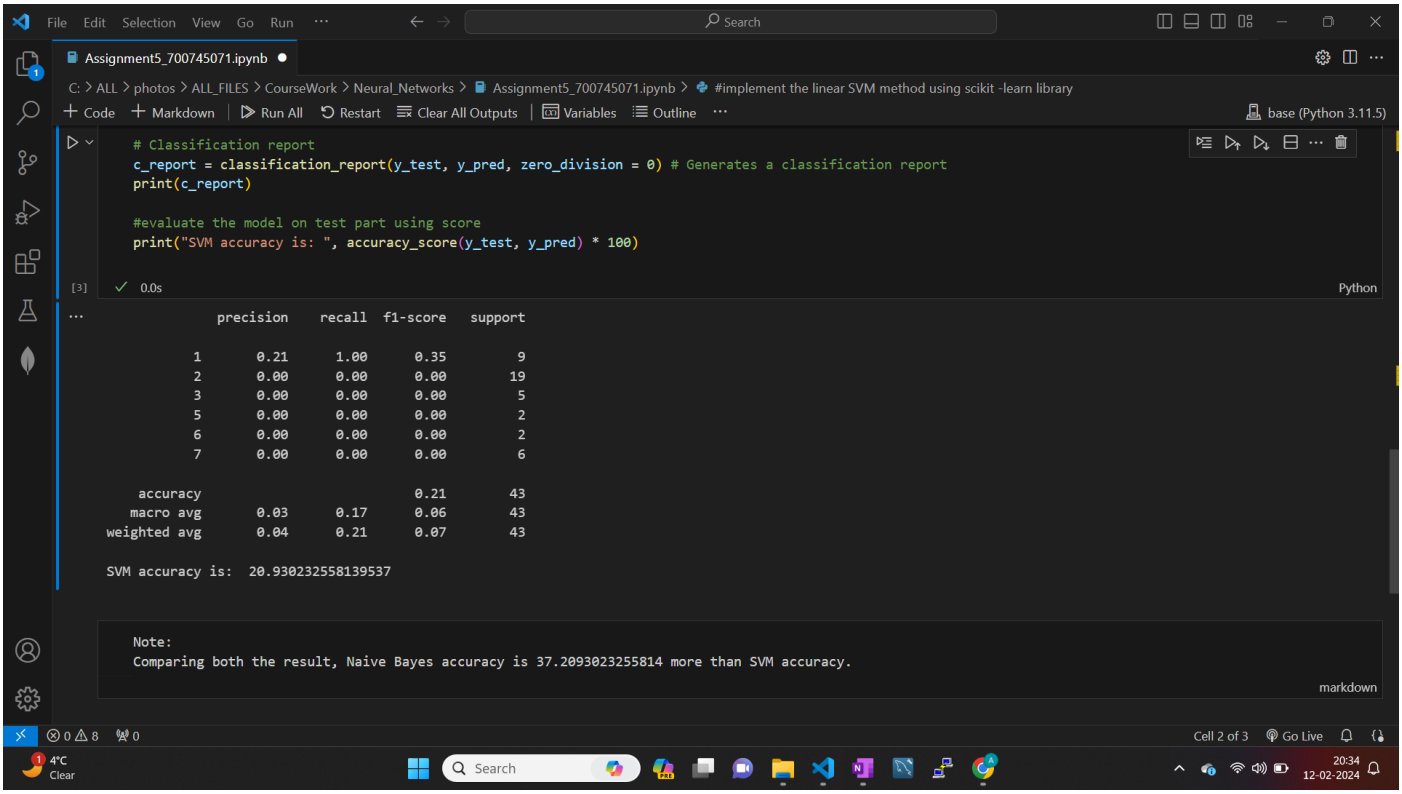
```
#implement the Naive Bayes method using scikit -learn library
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import classification_report, accuracy_score

#use the dataset available with the name glass
#This line reads the dataset from the CSV file located at the specified path and assigns it to the variable glass_data.
glass_data = pd.read_csv('C://ALL//photos//ALL_FILES//CourseWork//Neural_Networks////NNDL_Code and Data//glass.csv')

#This creates the feature matrix x_train by dropping the column named "Type" from the glass_data.
x_train = glass_data.drop("Type", axis=1)
y_train = glass_data['Type'] #This creates the target vector y_train by selecting only the column named "Type" from the glass_data.

#use test_train_split to create training and testing part
x_train, x_test, y_train, y_test = train_test_split(x_train, y_train, test_size=0.2, random_state=0) #This line splits the data into training and testing sets

# Train the model using the training sets
classifier = GaussianNB() #This instantiates a Gaussian Naive Bayes classifier
classifier.fit(x_train, y_train) #This trains the classifier on the training data using the fit() method
y_pred = classifier.predict(x_test) #This line generates predictions for the test data x_test using the trained classifier's predict() method.

# Classification report
c_report = classification_report(y_test, y_pred) #This line generates a classification report
print(c_report)

#evaluate the model on test part using score
print("Naive Bayes accuracy is: ",  (accuracy_score(y_test, y_pred))*100) #This line prints the accuracy of the Naive Bayes classifier on the test data.
```



```
# Train the model using the training sets
classifier = GaussianNB() #This instantiates a Gaussian Naive Bayes classifier
classifier.fit(x_train, y_train) #This trains the classifier on the training data using the fit() method
y_pred = classifier.predict(x_test) #This line generates predictions for the test data x_test using the trained classifier's predict() method.

# Classification report
c_report = classification_report(y_test, y_pred) #This line generates a classification report
print(c_report)

#evaluate the model on test part using score
print("Naive Bayes accuracy is: ",  (accuracy_score(y_test, y_pred))*100) #This line prints the accuracy of the Naive Bayes classifier on the test data.
```

```
              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

Naive Bayes accuracy is:  37.2093023255814
```


File Edit Selection View Go Run
Search
Assignment5_700745071.ipynb
C: > ALL > photos > ALL_FILES > CourseWork > Neural_Networks > Assignment5_700745071.ipynb > #implement the linear SVM method using scikit -learn library
Code Markdown Run All Restart Clear All Outputs Variables Outline
base (Python 3.11.5)
#implement the linear SVM method using scikit -learn library
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
#use the same dataset above
glass_data = pd.read_csv('C://ALL//photos//ALL_FILES//CourseWork//Neural_Networks////NNDL_Code and Data//glass.csv')
x_train = glass_data.drop("Type", axis=1) #Creates the feature matrix x_train by dropping the "Type" column from glass_data.
y_train = glass_data['Type'] #Creates the target vector y_train by selecting only the "Type" column from glass_data.
#use test_train_split to create training and testing part
x_train, x_test, y_train, y_test = train_test_split(x_train, y_train, test_size=0.2, random_state=0) #Splits the dataset into training and testing sets
# Train the model using the training sets
classifier= SVC() #Instantiates a Support Vector Classifier (SVC) object
classifier.fit(x_train, y_train) #rains the SVC model using the training data and corresponding labels.
y_pred = classifier.predict(x_test) #Predicts the labels for the test data
# Classification report
c_report = classification_report(y_test, y_pred, zero_division = 0) # Generates a classification report
print(c_report)
#evaluate the model on test part using score
print("SVM accuracy is: ", accuracy_score(y_test, y_pred) * 100)
[3] 0.0s
Python
precision recall f1-score support
Cell 2 of 3
Go Live
4°C Clear
Search
20:34 12-02-2024


File Edit Selection View Go Run
Search
Assignment5_700745071.ipynb
C: > ALL > photos > ALL_FILES > CourseWork > Neural_Networks > Assignment5_700745071.ipynb > #implement the linear SVM method using scikit -learn library
Code Markdown Run All Restart Clear All Outputs Variables Outline
base (Python 3.11.5)
# Classification report
c_report = classification_report(y_test, y_pred, zero_division = 0) # Generates a classification report
print(c_report)
#evaluate the model on test part using score
print("SVM accuracy is: ", accuracy_score(y_test, y_pred) * 100)
[3] 0.0s
Python
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
6 0.00 0.00 0.00 2
7 0.00 0.00 0.00 6
accuracy 0.21 43
macro avg 0.03 0.17 0.06 43
weighted avg 0.04 0.21 0.07 43
SVM accuracy is: 20.930232558139537
Note:
Comparing both the result, Naive Bayes accuracy is 37.2093023255814 more than SVM accuracy.
markdown
Cell 2 of 3
Go Live
4°C Clear
Search
20:34 12-02-2024