Neural Network Deep Learning

Assignment – 5

Name: Kishor Kumar Andekar

Student ID: 700744713

Github Link : https://github.com/kishorreyansh/Neural-Network-Deep-Learning/tree/main/Assignment-5

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)

   In the below code snippet, we are implementing Naïve Bayes method using scikit-learn library:

   Importing the given "glass.csv" into a variable called glass_dataframe. Splitting the data using the train_test_split() function, such that 1/4 of the data is reserved as a test subset. Train the model using the GaussianNB() function, predict the values using the predict() function. Evaluate the model and generate the classification report of the predicted set using classification_report(), and then calculate the Naïve Bayes accuracy using accuracy_score().

naivebayes.py svm.py

Assignment 5 > naivebayes.py > ...

```python
# 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

# pip install -U scikit-learn
from sklearn.model_selection import train_test_split
# Importing a Naive Bayes library
from sklearn.naive_bayes import GaussianNB
# Importing a Metrics library to Use classification_report and accuracy_score functions
from sklearn.metrics import classification_report, accuracy_score

# Importing the datasets from glass.csv
glass_dataframe = pd.read_csv('glass.csv')

# Dropping Type Column and assigning other columns to variable X
X = glass_dataframe.drop("Type",axis=1)
Y = glass_dataframe[['Type']]

X_Train, X_Test, Y_Train, Y_Test = train_test_split(X,Y,test_size=1/4, random_state = 0)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS D:\UCM\Kishor\Neural Network Deep Learning\Assignments\Assignment 5>
```

naivebayes.py svm.py

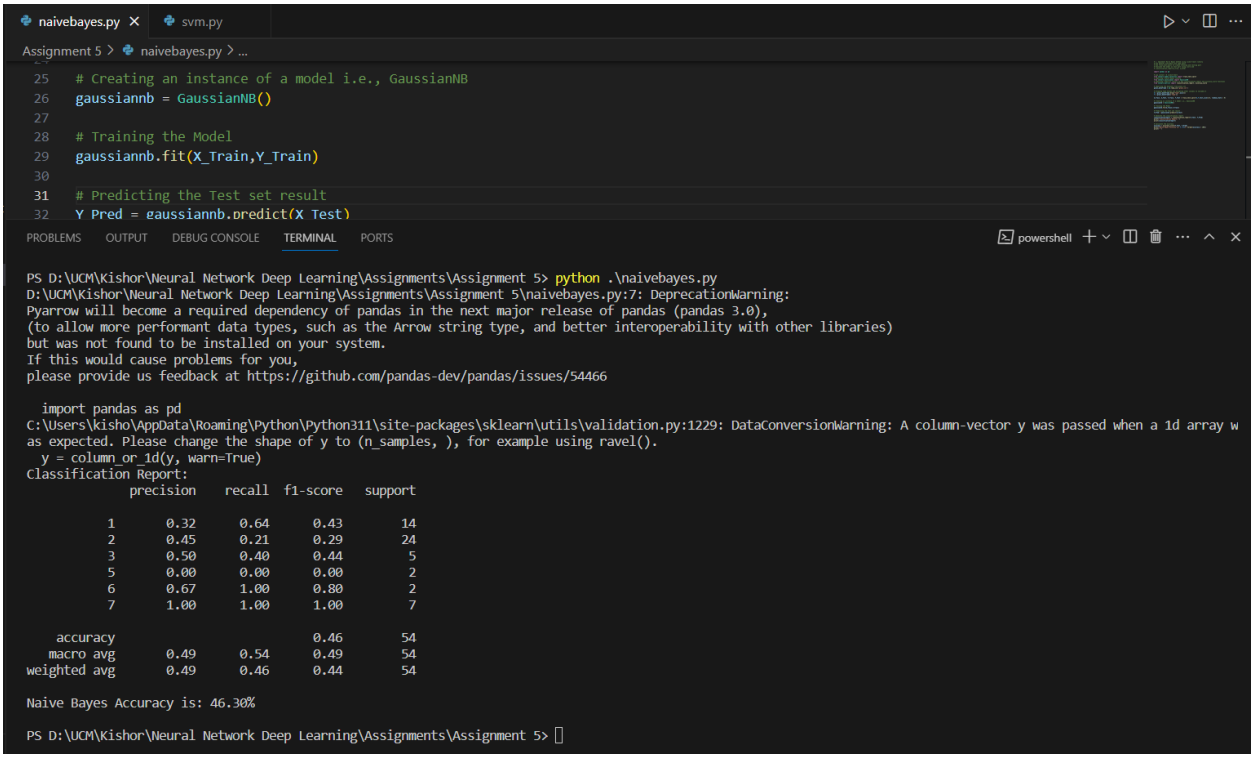
Assignment 5 > naivebayes.py > ...

```python
# Creating an instance of a model i.e., GaussianNB
gaussiannb = GaussianNB()

# Training the Model
gaussiannb.fit(X_Train,Y_Train)

# Predicting the Test set result
Y_Pred = gaussiannb.predict(X_Test)

# Generate the classification report
classificationreport = classification_report(Y_Test, Y_Pred)
print("Classification Report: ")
print(classificationreport)

# Calculate the accuracy
accuracy = accuracy_score(Y_Test, Y_Pred)
print("Naive Bayes Accuracy is: {:.2f}%".format(accuracy * 100))
print(" ")
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS D:\UCM\Kishor\Neural Network Deep Learning\Assignments\Assignment 5>

Output:



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)

   In the below code snippet, we are implementing Linear SVM method using scikit library:

   Importing the given “glass.csv” into a variable called glass_dataframe. Splitting the data using the train_test_split() function, such that 1/4 of the data is reserved as a test subset. Train the model using the SVC() function, predict the values using the predict() function. Evaluate the model and generate the classification report of the predicted set using classification_report(), and then calculate the Naïve Bayes accuracy using accuracy_score().

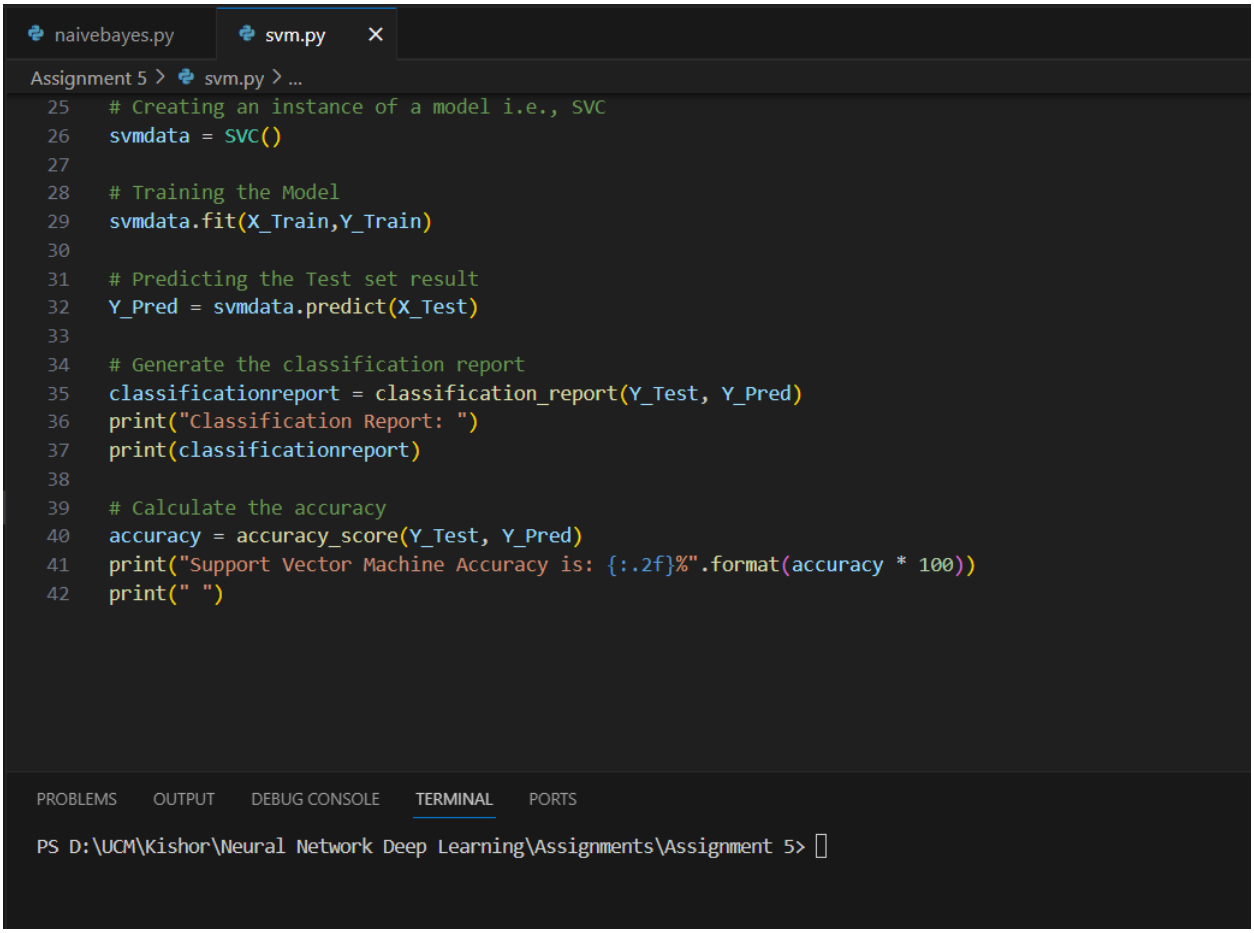
naivebayes.py | svm.py

Assignment 5 > svm.py > ...

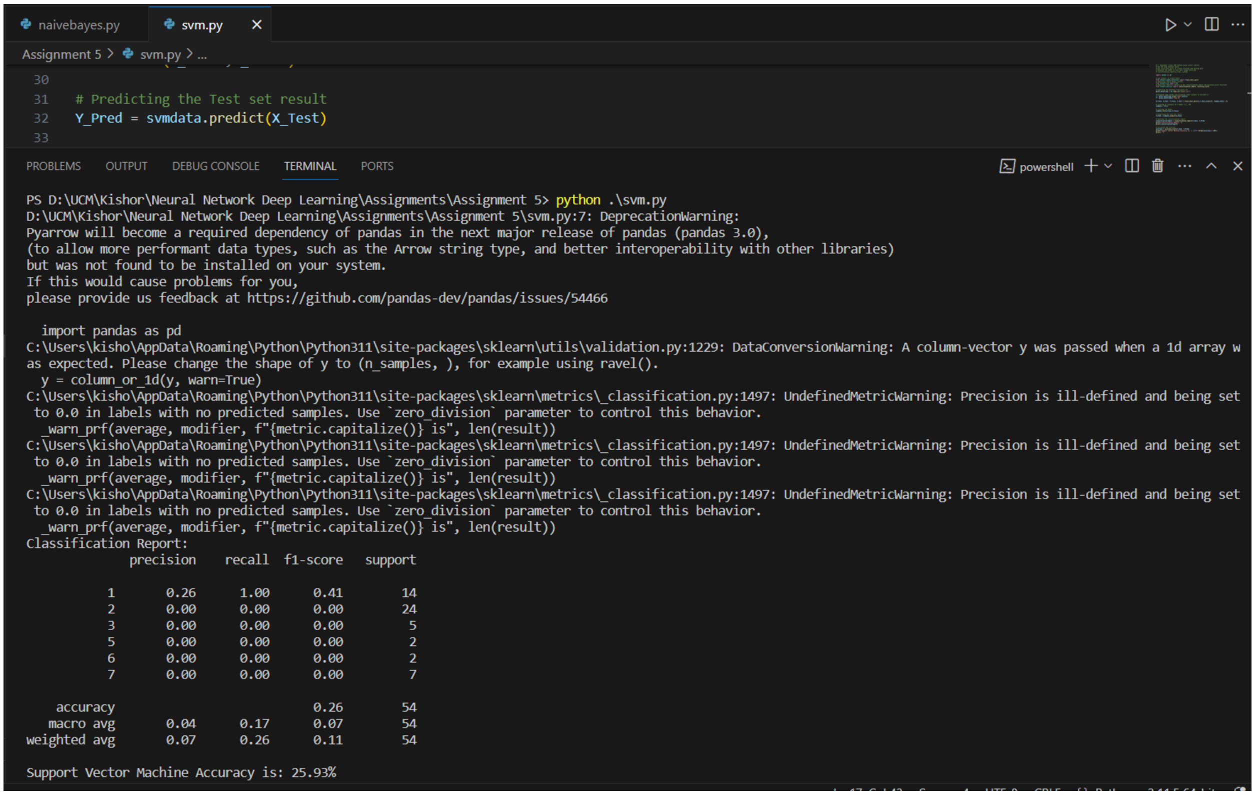
```python
# 2. Implement linear SVM method using scikit library
# Use the same dataset above
# Use train_test_split to create training and testing part
# Evaluate the model on test part using score and
# classification_report(y_true, y_pred)

import pandas as pd

# pip install -U scikit-learn
from sklearn.model_selection import train_test_split
# Importing a Naive Bayes library
from sklearn.svm import SVC
# Importing a Metrics library to Use classification_report and accuracy_score functions
from sklearn.metrics import classification_report, accuracy_score

# Importing the datasets from glass.csv
glass_dataframe = pd.read_csv('glass.csv')

# Dropping Type Column and assigning other columns to variable X
X = glass_dataframe.drop("Type",axis=1)
Y = glass_dataframe[['Type']]

X_Train, X_Test, Y_Train, Y_Test = train_test_split(X,Y,test_size=1/4, random_state = 0)

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS D:\UCM\Kishor\Neural Network Deep Learning\Assignments\Assignment 5>
```


naivebayes.py
svm.py
Assignment 5 > svm.py > ...
25 # Creating an instance of a model i.e., SVC
26 svmdata = SVC()
27
28 # Training the Model
29 svmdata.fit(X_Train,Y_Train)
30
31 # Predicting the Test set result
32 Y_Pred = svmdata.predict(X_Test)
33
34 # Generate the classification report
35 classificationreport = classification_report(Y_Test, Y_Pred)
36 print("Classification Report: ")
37 print(classificationreport)
38
39 # Calculate the accuracy
40 accuracy = accuracy_score(Y_Test, Y_Pred)
41 print("Support Vector Machine Accuracy is: {:.2f}%".format(accuracy * 100))
42 print(" ")
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
PS D:\UCM\Kishor\Neural Network Deep Learning\Assignments\Assignment 5>

Output:



```
naivebayes.py    svm.py
Assignment 5 > svm.py > ...
30
31    # Predicting the Test set result
32    Y_Pred = svmdata.predict(X_Test)
33

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS        powershell

PS D:\UCM\Kishor\Neural Network Deep Learning\Assignments\Assignment 5> python .\svm.py
D:\UCM\Kishor\Neural Network Deep Learning\Assignments\Assignment 5\svm.py:7: DeprecationWarning:
Pyarrow will become a required dependency of pandas in the next major release of pandas (pandas 3.0),
(to allow more performant data types, such as the Arrow string type, and better interoperability with other libraries)
but was not found to be installed on your system.
If this would cause problems for you,
please provide us feedback at https://github.com/pandas-dev/pandas/issues/54466

  import pandas as pd
C:\Users\kisho\AppData\Roaming\Python\Python311\site-packages\sklearn\utils\validation.py:1229: DataConversionWarning: A column-vector y was passed when a 1d array w
as expected. Please change the shape of y to (n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)
C:\Users\kisho\AppData\Roaming\Python\Python311\site-packages\sklearn\metrics\_classification.py:1497: UndefinedMetricWarning: Precision is ill-defined and being set
 to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
C:\Users\kisho\AppData\Roaming\Python\Python311\site-packages\sklearn\metrics\_classification.py:1497: UndefinedMetricWarning: Precision is ill-defined and being set
 to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
C:\Users\kisho\AppData\Roaming\Python\Python311\site-packages\sklearn\metrics\_classification.py:1497: UndefinedMetricWarning: Precision is ill-defined and being set
 to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
Classification Report:
              precision    recall  f1-score   support

           1       0.26      1.00      0.41        14
           2       0.00      0.00      0.00        24
           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         7

    accuracy                           0.26        54
   macro avg       0.04      0.17      0.07        54
weighted avg       0.07      0.26      0.11        54

Support Vector Machine Accuracy is: 25.93%
```

CONCLUSION:

The Naïve Bayes Method in this case has an accuracy of 46.30%, while the SVM classifier has a 25.93% accuracy. Better performance is usually indicated by a higher accuracy. Thus, the Naïve Bayes classifier seems to be superior in this instance based only on accuracy.