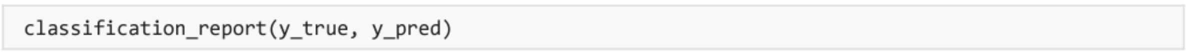
Student ID: 700752115

Student Name: Rumana Shaik

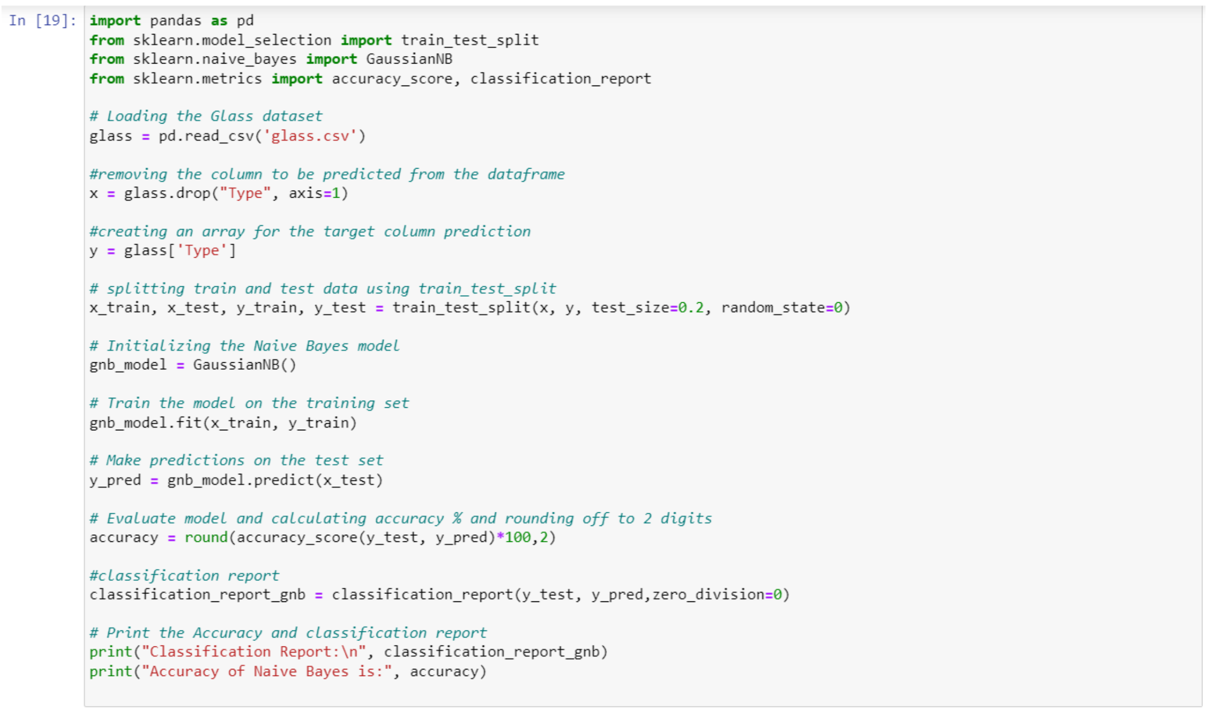
Neural Networks & Deep Learning – ICP - 5

Github link: https://github.com/ShaikRumana301/Neural-Network-DL-ICP-5.git

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

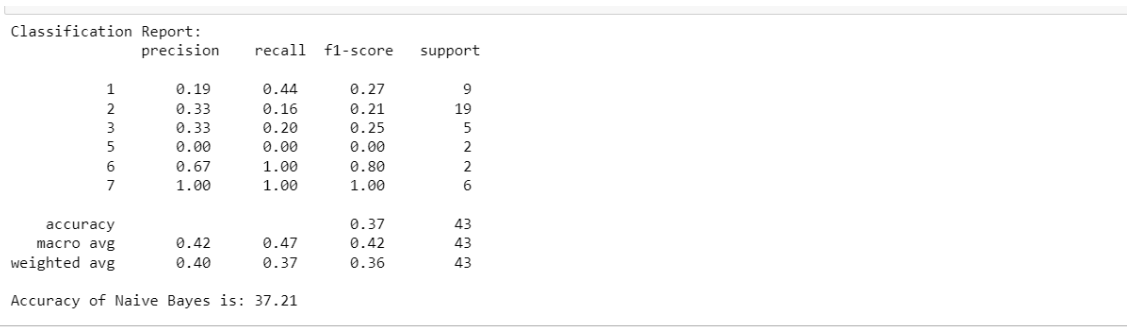


```
classification_report(y_true, y_pred)
```



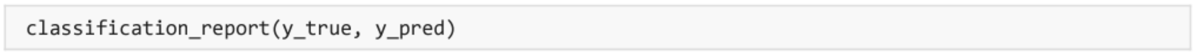
```
In [19]: import pandas as pd
         from sklearn.model_selection import train_test_split
         from sklearn.naive_bayes import GaussianNB
         from sklearn.metrics import accuracy_score, classification_report

         # Loading the Glass dataset
         glass = pd.read_csv('glass.csv')

         #removing the column to be predicted from the dataframe
         x = glass.drop("Type", axis=1)

         #creating an array for the target column prediction
         y = glass['Type']

         # splitting train and test data using train_test_split
         x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=0)

         # Initializing the Naive Bayes model
         gnb_model = GaussianNB()

         # Train the model on the training set
         gnb_model.fit(x_train, y_train)

         # Make predictions on the test set
         y_pred = gnb_model.predict(x_test)

         # Evaluate model and calculating accuracy % and rounding off to 2 digits
         accuracy = round(accuracy_score(y_test, y_pred)*100,2)

         #classification report
         classification_report_gnb = classification_report(y_test, y_pred,zero_division=0)

         # Print the Accuracy and classification report
         print("Classification Report:\n", classification_report_gnb)
         print("Accuracy of Naive Bayes is:", accuracy)
```

Output:

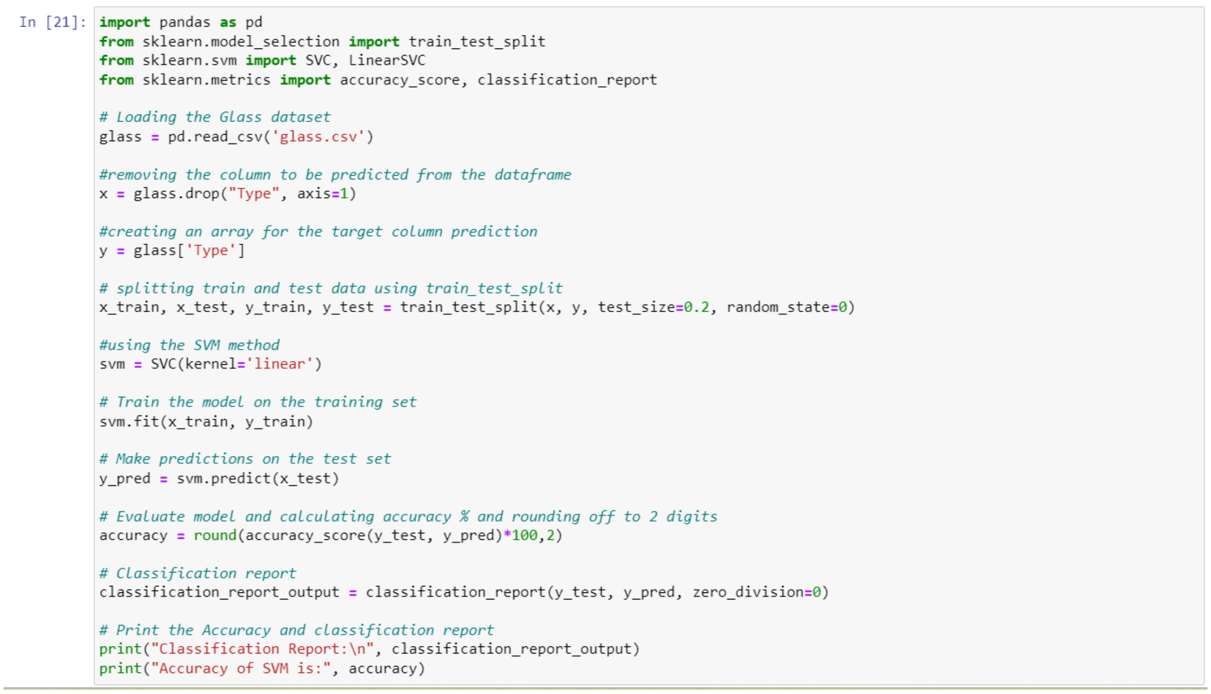


```
Classification Report:
               precision    recall  f1-score   support

           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

Accuracy of Naive Bayes is: 37.21
```

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

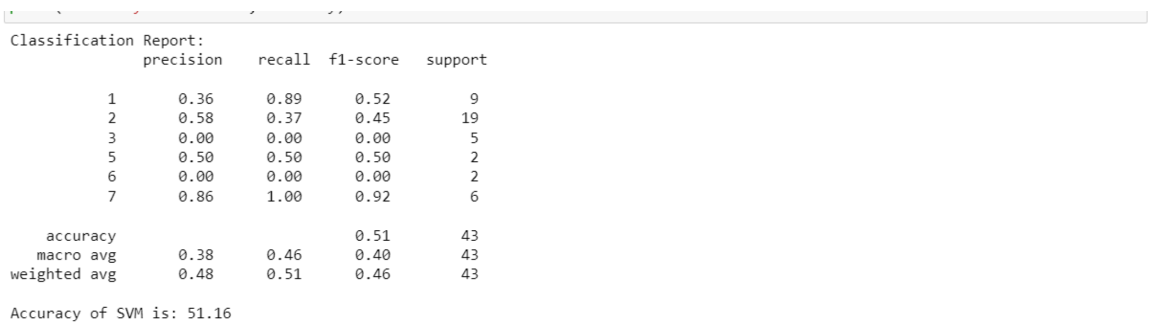


```
classification_report(y_true, y_pred)
```



```
In [21]: import pandas as pd
         from sklearn.model_selection import train_test_split
         from sklearn.svm import SVC, LinearSVC
         from sklearn.metrics import accuracy_score, classification_report

         # Loading the Glass dataset
         glass = pd.read_csv('glass.csv')

         #removing the column to be predicted from the dataframe
         x = glass.drop("Type", axis=1)

         #creating an array for the target column prediction
         y = glass['Type']

         # splitting train and test data using train_test_split
         x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=0)

         #using the SVM method
         svm = SVC(kernel='linear')

         # Train the model on the training set
         svm.fit(x_train, y_train)

         # Make predictions on the test set
         y_pred = svm.predict(x_test)

         # Evaluate model and calculating accuracy % and rounding off to 2 digits
         accuracy = round(accuracy_score(y_test, y_pred)*100,2)

         # Classification report
         classification_report_output = classification_report(y_test, y_pred, zero_division=0)

         # Print the Accuracy and classification report
         print("Classification Report:\n", classification_report_output)
         print("Accuracy of SVM is:", accuracy)
```

Output:



```
Classification Report:
               precision    recall  f1-score   support

           1       0.36      0.89      0.52         9
           2       0.58      0.37      0.45        19
           3       0.00      0.00      0.00         5
           5       0.50      0.50      0.50         2
           6       0.00      0.00      0.00         2
           7       0.86      1.00      0.92         6

    accuracy                           0.51        43
   macro avg       0.38      0.46      0.40        43
weighted avg       0.48      0.51      0.46        43

Accuracy of SVM is: 51.16
```

- Which algorithm you got better accuracy? Can you justify why?

Ans: When we compare the Naïve Bayes and linear SVM both read the glass.csv data set and x_train has all the data except for type column data and y_train had only type column data x_train and y_train has been divided into 80:20 ratio. Declared naive Bayes method in Code 1 and linear SVM method for Code 2, to fit and train data. Following the training, predictions are made, and the results are stored in y_pred, with the actual data available in y_test.

After running both methods and comparing the accuracy, we can say that Linear SVM has a better accuracy score than Gaussian Naive Bayes. The reason for this could be because Linear SVM is better suited for complex and non-linearly separable datasets. The SVM algorithm tries to find the maximum margin between the data points and the decision boundary, which makes it robust to outliers and non-linearities, where it finds the hyper plane. Additionally, SVM allows for more flexibility in terms of choosing different kernel functions to tackle different types of data distributions. Gaussian Naive Bayes, on the other hand, assumes independence between the features and is not as robust as SVM for complex datasets.

The dataset size and structure, if the data has many features and is easy to interpret, Naive Bayes can be more effective. The data distribution, Naive Bayes assumes that the features are independent, if the features in the data are not independent, the performance of Naive Bayes can decrease.

The accuracy of the model can also depend on the choice of hyperparameters, if the hyperparameters of the Naïve Bayes algorithm were not optimized, this could result in a lower accuracy score. It is important to keep in mind that accuracy is not the only evaluation metric that should be considered, other metrics such as precision, recall, and score should also be considered to get a comprehensive evaluation on performance.

.