**Spring 2024: CS5720**

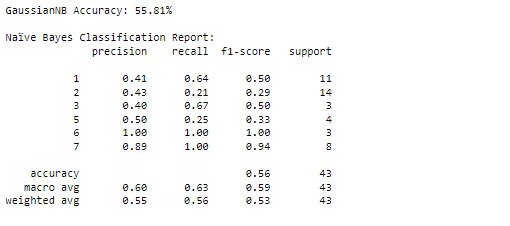
**Neural Networks and Deep Learning - ICP-5**

**Pranay Rao Guddity (700745063)**

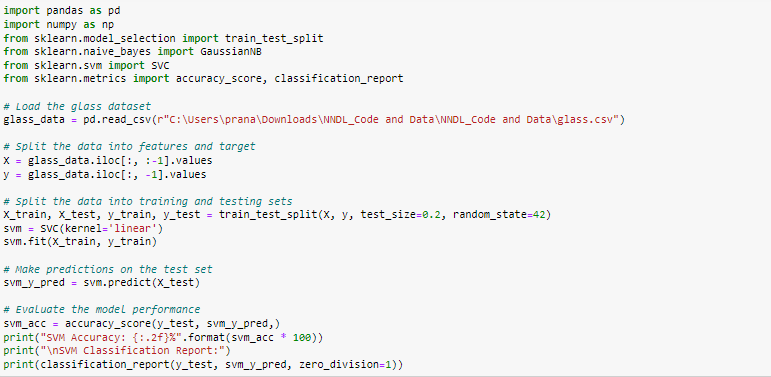
**GitHub Link:** https://github.com/raopranay1999/-Neural-Network-Assignment-5.git

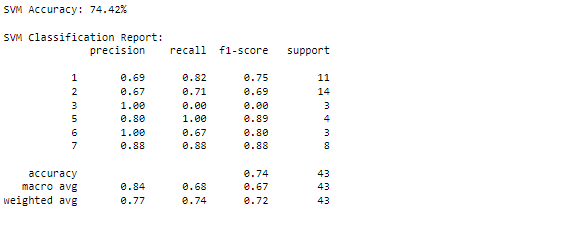
1. Naive Bayes Implement Naive Bayes method using scikit-learn library. Use dataset available with name glass. Use train\_test\_split to create training and testing part. Evaluate the model ion test part using score and classification\_report(y\_true, ypred)

****

****

1. SVM Method 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)

****

****

Which algorithm you got better accuracy? Can you justify why? The findings indicate that the SVM approach produced an accuracy of 74.42% and the GaussianNB accuracy is 55.81%. This suggests that, given the same test size and random state, the SVM Algorithm performs admirably in this situation. The following are some supporting factors for these: GNB is a basic probabilistic classifier that is easy to learn and comprehend, and it is based on the Bayes theorem. It applies strong independence assumptions between the features, i.e., it assumes that the characteristics are conditionally independent given the class. This approach is most commonly used when the characteristics in a text classification problem are represented by words or n-grams. Given that GNB handles high-dimensional data well, it's a good option if you have a lot of features. SVM, on the other hand, is a more sophisticated method that makes use of the concept of margins to identify the optimal decision border between classes. Because SVM can handle complex non-linear relationships between features, it is often used for image classification or classification problems with little data. SVM is also more resistant to overfitting than GNB when