[**SEP 785 Machine Learning**](https://avenue.cllmcmaster.ca/d2l/home/648292)

Student name: Mahya Karamad Kasmaei – 400609404

Dr. Jeff Fortuna

**Report on final project**

I have used one of the datasets from UCI (Wine Quality). There were 2 datasets (Red and White wines) which I concatenated into one dataset and used the type of wine (RED & WHITE) as labels rather than using 10 different qualities. The shape of the data is (5320, 13) which means **5320 Samples** with 13 features and 2 labels. (Red, White)

* Step1: **Importing**.

I have imported both datasets using **Pandas** and gave them flag (Red wine = 0), (White wine = 1) by adding a new column named “Wine Type”. Then I concatenated them into one dataset with axis=0 means that they have been concatenated by rows.

* Step 2: **Preprocessing**.

Dataset was checked in duplicates; null data and it was normalized before the training process. Dataset was Separated in features and labels to avoid overfitting. wine type column is used for labels. **20** percent of data is used for testing data and the remaining is used for training. (**4256** samples for training and **1064** samples for testing)

* Step 3: **Model selection**

A graph with a line

Description automatically generated**SVM** is the first algorithm used for training. By using **Cross-validation** with 5 folds, the average accuracy of training was shown to be **0.99 %**. I also used “Grid Search” for choosing the best parameters. The results are shown to be as follows:

Training time in SVM: 0.4790 seconds

Testing time in SVM: 0.0036 seconds

Confusion matrix: [288 1]

[2 773]

precision: 99.87 %

recall: 99.74 %

accuracy: 99.71 %

**KNN** is the second algorithm used for training. I used 5 neighbors and 10 for leaf size. The results are shown to be as follows:

A graph with a line

Description automatically generatedTraining time in KNN: 0.0100 seconds

Testing time in KNN: 0.0725 seconds

confusion matrix: [288 1]

[7 768]

precision: 99.86 %

recall: 99.09 %

accuracy: 99.24%

**Naive Bayes** is the third algorithm used for training. The results are shown to be as follows:

A graph with a line

Description automatically generatedTraining time in NB: 0.0019 seconds

Testing time in NB: 0.0010 seconds

confusion matrix :[283 6]

[15 760]

precision: 99.21%

recall: 98.06%

accuracy: 98.02%

**Conclusion:**

**Cross validation** gives me average accuracy and **gride search** helped to find the best parameters for algorithm so that the results show better percentage in precision, accuracy and recall considering that longer **computational** time in SVM.

The **ROC** curve in SVM seems the best because the classifier has enough skill in distinguishing classes and the best threshold has been chosen for it.

**Confusion matrix** is also better in SVM. As can be observed, the number of **False Negative** and **False Positive** are limited in SVM which means that the majority amount of test samples is predicted correctly.