# COMP – 8745 Project Report

Kazi Ashraf Moinuddin

U00727147

# Task 1

## 5-Fold Cross Validation Results:

### Neural Net:

```
-------- Average Scores for Layer (25, 25)
Avg Precision:  0.39
Avg Recall:  0.39
Avg f1 score:  0.39
-------- Average Scores for Layer (25, 25, 25)
Avg Precision:  0.39
Avg Recall:  0.39
Avg f1 score:  0.39
-------- Average Scores for Layer (30, 30, 30)
Avg Precision:  0.39
Avg Recall:  0.39
Avg f1 score:  0.40
-------- Average Scores for Layer (40, 40, 40)
Avg Precision:  0.40
Avg Recall:  0.39
Avg f1 score:  0.39
-------- Average Scores for Layer (40, 40, 40, 40)
Avg Precision:  0.40
Avg Recall:  0.40
Avg f1 score:  0.40
-------- Average Scores for Layer (50, 50, 50, 50)
Avg Precision:  0.40
Avg Recall:  0.40
Avg f1 score:  0.38
-------- Average Scores for Layer (100, 100, 100, 100)
Avg Precision:  0.40
Avg Recall:  0.41
Avg f1 score:  0.40
```

### Naive Bayes:

```
Avg Precision:  0.38
Avg Recall:  0.38
Avg f1 score:  0.38
```

### Logistic Regression

**Using l1 regularization**

```
----------- Average Score for C = 0.01 -----------
Avg Precision:  0.20
Avg Recall:  0.20
Avg f1 score:  0.20
```

```
---------- Average Score for C = 0.1 ----------
Avg Precision:  0.36
Avg Recall:  0.36
Avg f1 score:  0.36
---------- Average Score for C = 1 ----------
Avg Precision:  0.44
Avg Recall:  0.44
Avg f1 score:  0.44
---------- Average Score for C = 10 ----------
Avg Precision:  0.40
Avg Recall:  0.40
Avg f1 score:  0.40
---------- Average Score for C = 100 ----------
Avg Precision:  0.39
Avg Recall:  0.39
Avg f1 score:  0.39
```

**Using l2 regularization**

```
---------- Average Score for C = 0.01 ----------
Avg Precision:  0.42
Avg Recall:  0.42
Avg f1 score:  0.42
---------- Average Score for C =  0.1 ----------
Avg Precision:  0.43
Avg Recall:  0.43
Avg f1 score:  0.43
---------- Average Score for C =  1 ----------
Avg Precision:  0.44
Avg Recall:  0.44
Avg f1 score:  0.44
---------- Average Score for C =  10 ----------
Avg Precision:  0.42
Avg Recall:  0.42
Avg f1 score:  0.42
---------- Average Score for C =  100 ----------
Avg Precision:  0.40
Avg Recall:  0.40
Avg f1 score:  0.40
```

## AdaBoosting:

```
---------- Average Score for n_estimators = 40 ----------
Avg Precision:  0.41
Avg Recall:  0.41
Avg f1 score:  0.41
---------- Average Score for n_estimators =  50 ----------
Avg Precision:  0.41
Avg Recall:  0.41
Avg f1 score:  0.41
```

```
---------- Average Score for n_estimators =  60 ----------
Avg Precision:  0.41
Avg Recall:  0.41
Avg f1 score:  0.41
---------- Average Score for n_estimators =  70 ----------
Avg Precision:  0.42
Avg Recall:  0.42
Avg f1 score:  0.42
---------- Average Score for n_estimators =  80 ----------
Avg Precision:  0.41
Avg Recall:  0.41
Avg f1 score:  0.41
---------- Average Score for n_estimators =  90 ----------
Avg Precision:  0.41
Avg Recall:  0.41
Avg f1 score:  0.41
---------- Average Score for n_estimators =  100 ----------
Avg Precision:  0.42
Avg Recall:  0.42
Avg f1 score:  0.42
```

## SVM:

**Using gaussian kernel**

```
---------- Average Score for C: 0.01 ----------
Avg Precision:  0.21
Avg Recall:  0.21
Avg f1 score:  0.21
---------- Average Score for C: 0.1 ----------
Avg Precision:  0.21
Avg Recall:  0.21
Avg f1 score:  0.21
---------- Average Score for C: 1 ----------
Avg Precision:  0.21
Avg Recall:  0.21
Avg f1 score:  0.21
---------- Average Score for C: 10 ----------
Avg Precision:  0.21
Avg Recall:  0.21
Avg f1 score:  0.21
---------- Average Score for C: 100 ----------
Avg Precision:  0.21
Avg Recall:  0.21
Avg f1 score:  0.21
```

**Using Poly Kernel**

```
---------- Average Score for degree: 1 ----------
Avg Precision:  0.38
Avg Recall:  0.38
Avg f1 score:  0.38
```

```
---------- Average Score for degree: 2 ----------
Avg Precision:  0.20
Avg Recall:  0.20
Avg f1 score:  0.20
---------- Average Score for degree: 3 ----------
Avg Precision:  0.20
Avg Recall:  0.20
Avg f1 score:  0.20
---------- Average Score for degree: 5 ----------
Avg Precision:  0.20
Avg Recall:  0.20
Avg f1 score:  0.20
---------- Average Score for degree: 10 ----------
Avg Precision:  0.20
Avg Recall:  0.20
Avg f1 score:  0.20
```

# Task 2

## Neural Net:

**hidden_layer_sizes=(100,100,100,100)**

```
Avg Precision:  0.40
Avg Recall:  0.40
Avg f1 score:  0.39
```

## Naïve Bayes:

```
Avg Precision:  0.38
Avg Recall:  0.38
Avg f1 score:  0.38
```

## Logistic Regression:

**L1 Regularization**

```
Avg Precision:  0.44
Avg Recall:  0.44
Avg f1 score:  0.44
```

**L2 Regularization**

```
Avg Precision:  0.44
Avg Recall:  0.44
Avg f1 score:  0.44
```

## AdaBoosting:

**N_estimators = 100**

```
Avg Precision:  0.42
Avg Recall:  0.42
Avg f1 score:  0.42
```

## SVM:

**C = 500, kernel = poly**

```
Avg Precision:  0.34
Avg Recall:  0.34
Avg f1 score:  0.34
```

The scores do not really change for any algorithms. In some of the algorithms, the accuracy decreases. The detail results are provided on the ipython notebook.

# Task 3

## Neural Net:

```
Accuracy : 0.46846846846846846
---------- Scores for test data ------------
             precision    recall  f1-score   support

        1.0       0.56      0.66      0.60       199
        2.0       0.47      0.41      0.44       200
        3.0       0.41      0.39      0.40       200
        4.0       0.39      0.46      0.42       200
        5.0       0.51      0.44      0.47       200

avg / total       0.47      0.47      0.47       999
```

## Naïve Bayes:

```
Accuracy : 0.5005005005005005
---------- Scores for test data ------------
             precision    recall  f1-score   support

        1.0       0.50      0.65      0.57       199
        2.0       0.49      0.33      0.39       200
        3.0       0.43      0.41      0.42       200
        4.0       0.44      0.52      0.48       200
        5.0       0.65      0.59      0.62       200

avg / total       0.50      0.50      0.50       999
```

## Logistic Regression:

```
Accuracy : 0.5325325325325325
---------- Scores for test data ------------
             precision    recall  f1-score   support

        1.0       0.54      0.76      0.64       199
        2.0       0.56      0.40      0.46       200
        3.0       0.46      0.38      0.42       200
        4.0       0.47      0.50      0.48       200
        5.0       0.62      0.62      0.62       200

avg / total       0.53      0.53      0.52       999
```

## AdaBoosting:

```
Accuracy : 0.46646646646646645
---------- Scores for test data ------------
             precision    recall  f1-score   support

        1.0       0.58      0.60      0.59       199
        2.0       0.41      0.38      0.39       200
        3.0       0.38      0.34      0.36       200
        4.0       0.39      0.39      0.39       200
        5.0       0.54      0.62      0.58       200

avg / total       0.46      0.47      0.46       999
```

## SVM:

```
Accuracy : 0.3993993993993994
---------- Scores for test data ------------
             precision    recall  f1-score   support

        1.0       0.49      0.89      0.63       199
        2.0       0.52      0.06      0.11       200
        3.0       0.00      0.00      0.00       200
        4.0       0.31      0.92      0.47       200
        5.0       0.90      0.13      0.23       200

avg / total       0.44      0.40      0.29       999
```

- All the algorithms managed to report except SVM around 45% accuracy.
- SVM the lowest performing algorithm with around 40% accuracy on the test data
- Logistic regression with l2 regularization and C =1 is the best performing algorithm on the test data, achieving 53% accuracy.
- The hardest reviews to predict seem to be 3.0 and 4.0

# Task 4

- Using feature selection algorithms like grid search could be useful but is time consuming.
- Using deep neural networks could be useful also.
- Probabilistic systems seem to work much better than non-probabilistic systems as both naïve bayes and logistic regression showed promising results.