Linear and Logistic Regression

Name: Alekhya Devi Ranabothu

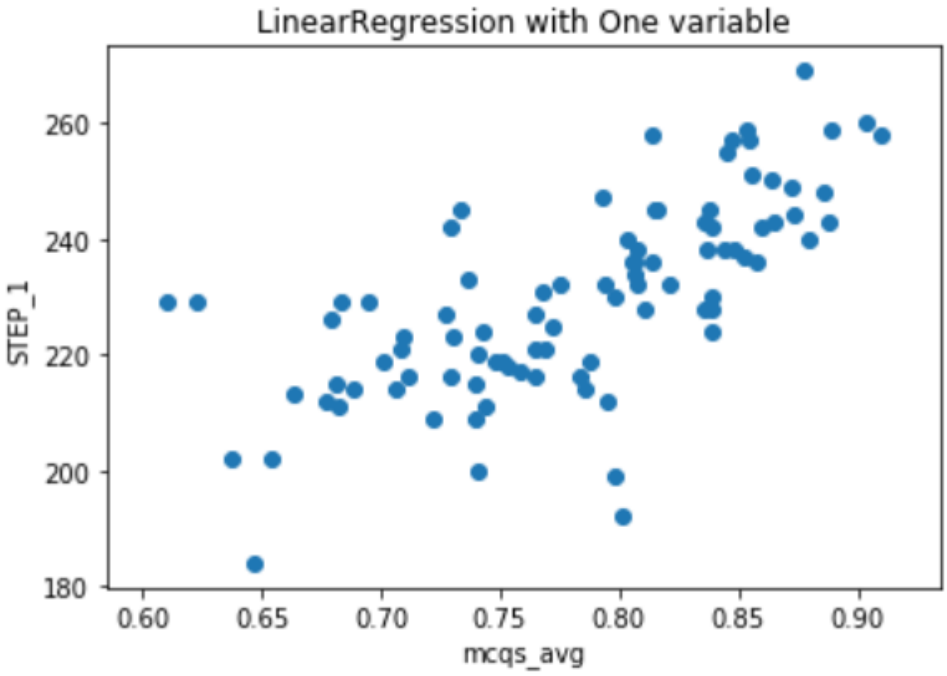
In this assignment Linear Regression and Logistic Regression are implemented from scratch using python. Gradient Descent algorithm is implemented to find out the best parameters.

Linear Regression

- The parameters(thetas) are initialized with zeros.
- Convergence condition is reached when the cost in the previous iteration is equal to the cost in the current iteration.
- The BSOM data is split into training data (80%) and test data (20%) and the metrics on test data are used for performance evaluation.
- Missing values are found in the target variable and these missing values are handled by filling the missing values with mean of the target variable.

1) Linear Regression with One variable

a) Linear Regression is implemented using 'all_mcqs_avg_n20' as independent variable and 'STEP_1' as target variable.
Below is the plot of all the datapoints in the dataset (both training and test points).

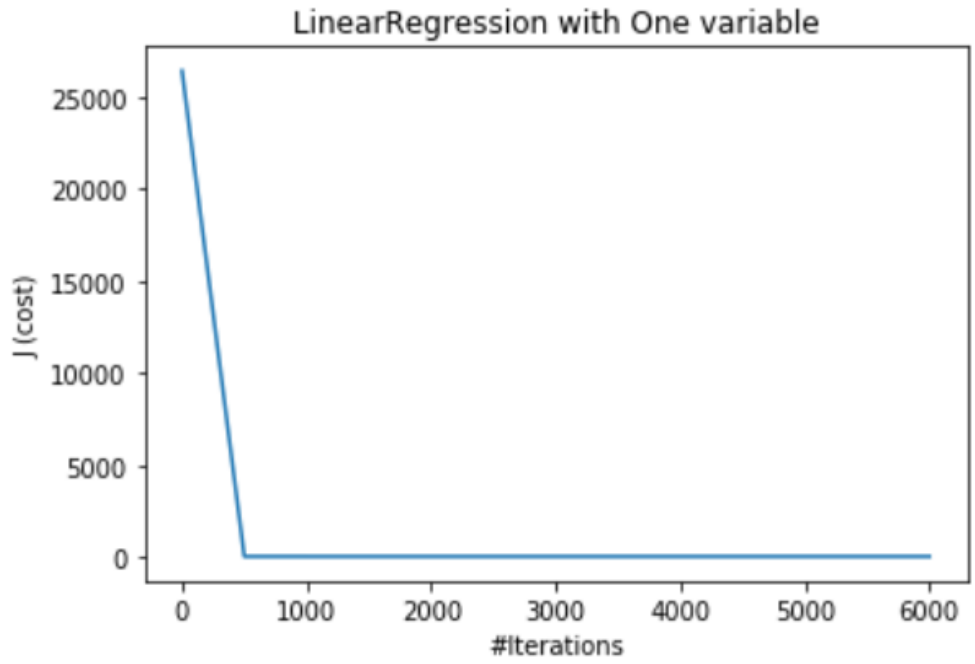


From the above plot, it can be observed that STEP_1 score has linear relationship with 'all_mcqs_avg_n20' score. That means students who has got good STEP_1 score have performed well in 'all_mcqs_avg_n20'.

Now Gradient Descent algorithm is implemented to find optimum parameters, with different learning rates. Here are the optimum parameters with minimum cost for different learning rates on the training data.

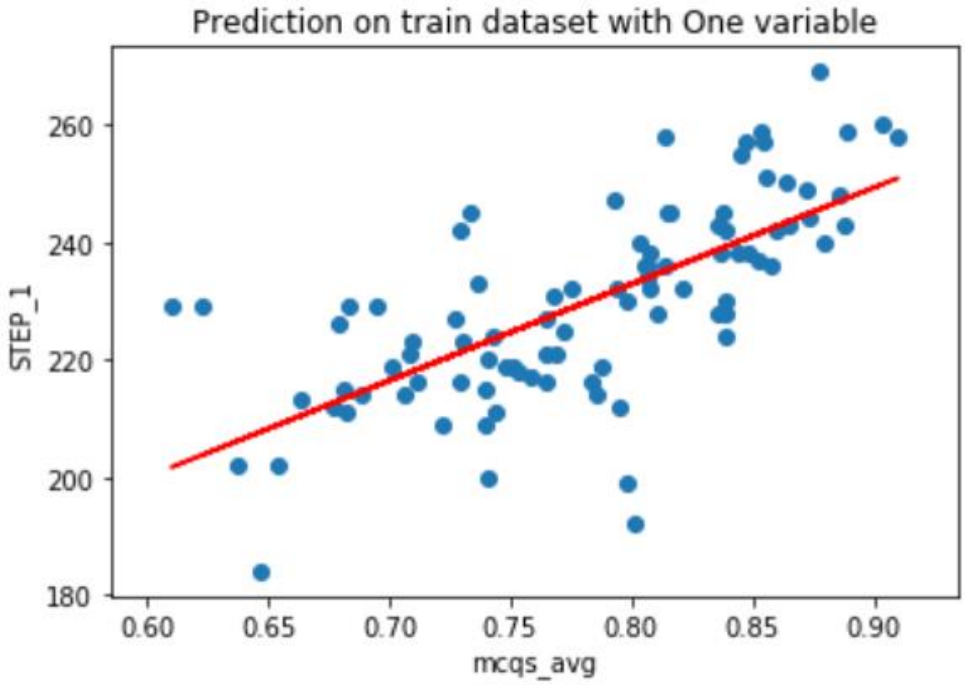
Learning Rate	Θ0	Θ1	Minimum cost	#iterations
0.1	101.60572589	164.03371449	70.62214	40096
0.5	101.60560448	164.03386938	70.62214	8667
0.6	101.60559773	164.033878	70.62214	7276
0.7	101.6055923	164.03388493	70.62214	6277

From the above table, it is observed that for learning rate 0.7, convergence condition is reached fast in a smaller number of iterations compared to other learning rates. So, we take 0.7 as good learning rate for this data. To verify the cost is decreasing with number of iterations and Gradient Descent algorithm is working correctly, a graph between cost function and number of iterations is plotted for learning rate 0.7.

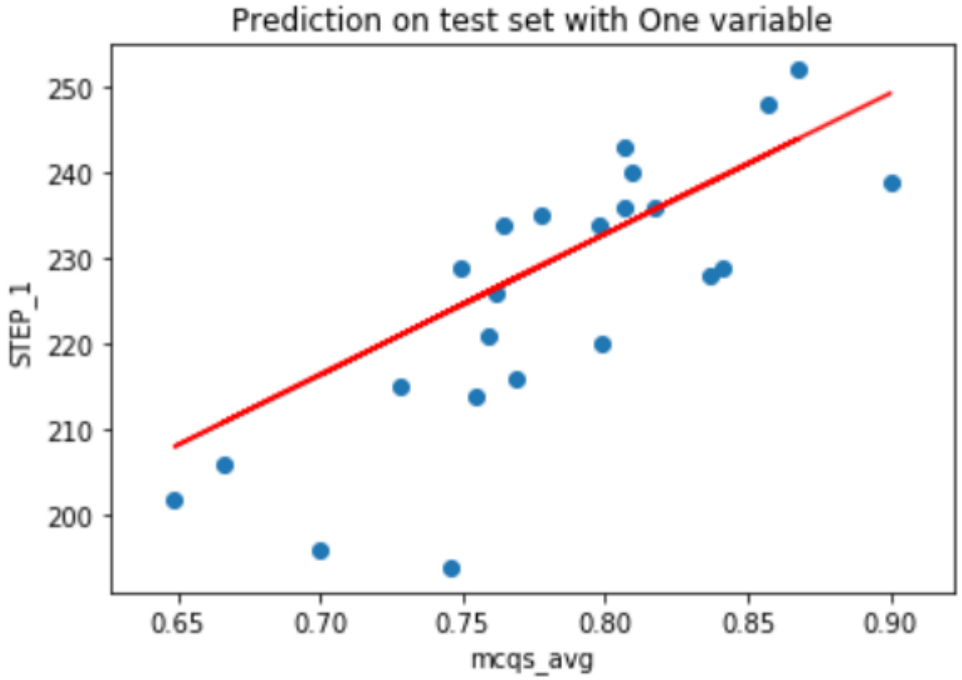


From the above plot, it can be observed that cost is decreased rapidly with number of iterations till less than 1000 iterations and the decrease of the cost is very small after 1000 iterations till the convergence condition is reached.
Hence, we can consider the below parameters as optimum for training data.

Learning Rate: 0.7
Θ0: 101.6055923
Θ1: 164.03388493
Below is the plot of prediction with the selected parameters on training data set. The red color line is the prediction and blue dots are data points.



Now the STEP_1 values are predicted on test data set with the above selected optimum parameters. Below is the prediction on test data set. The color line is the prediction and blue dots are data points.



b) Evaluation metrics such as Mean Squared Error, Pearson Correlation Coefficient and R-Squared are calculated for the test data. Below are the evaluation scores.

Mean Squared Error: 109.8163206
Correlation coefficient: 0.799463055
R-Squared: 0.54653235

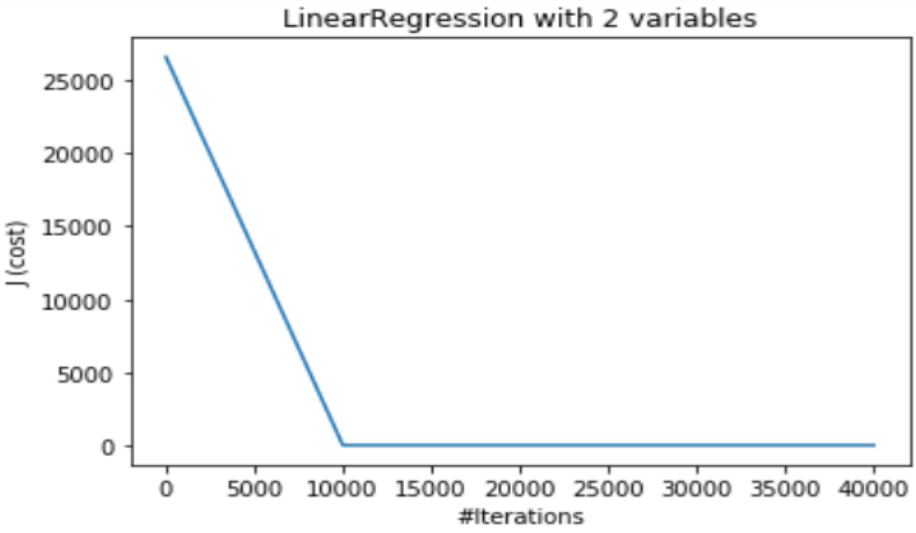
From the value of correlation coefficient and R-Squared it can be observed that the independent variable 'all_mcqs_avg_n20' is positively correlated with the target variable('STEP_1).

2) Linear Regression with Two Variables

a) Linear Regression is performed with two independent variables 'all_mcqs_avg_n20' and 'all_NBME_avg_n4' and target variable 'STEP_1'.
Gradient Descent algorithm is implemented with different learning rates to find optimum parameters.

Learning Rate	Θ0	Θ1	Θ2	Minimum cost	#iterations
0.1	85.06980541	157.57578126	21.31579306	61.256205	265873
0.5	85.06975947	157.5767968	21.31479567	61.256205	57730
0.6	85.06975566	157.57688115	21.31471283	61.256205	48656
0.7	85.06975417	157.57691391	21.31468065	61.256205	41905

From the above table, it is observed that for learning rate 0.7, convergence condition is reached fast in a smaller number of iterations compared to other learning rates. So, we take 0.7 as good learning rate for this data. To verify the cost is decreasing with number of iterations and Gradient Descent algorithm is working correctly, a graph between cost function and number of iterations is plotted for learning rate 0.7.



From the above plot, it can be observed that cost is decreased rapidly with number of iterations till less than 10000 iterations and the decrease of the cost is very small after 10000 iterations till the convergence condition is reached.
Hence, we can consider the below parameters as optimum for training data.

Learning Rate: 0.7
Θ0: 85.06975417
Θ1: 157.57691391
Θ2: 21.31468065

Evaluation metrics such as Mean Squared Error, Pearson Correlation Coefficient and R-Squared are calculated for the test data. Below are the evaluation scores.

Mean Squared Error: 90.35202
Correlation coefficient: 0.8504378
R-Squared: 0.626906

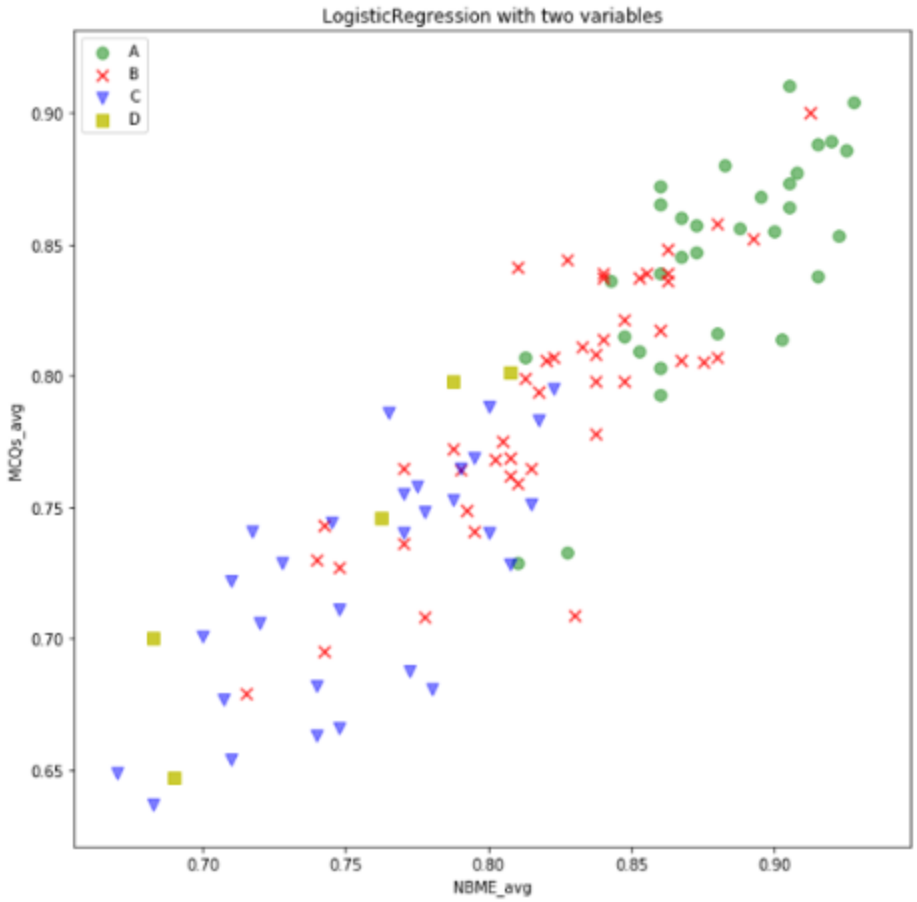
By comparing the metrics of Linear Regression on test data in section 1b and section 2a, it can be observed that the Mean Squared Error is decreased ,Pearson Correlation coefficient and R-Squared score is increased on adding the new variable 'all_NBME_avg_n4' . As the model with less Mean squared error, more correlation coefficient and more R-squared score performs better, we can say that the performance of the model in section 1 is improved on adding the feature 'all_NBME_avg_n4'.

Logistic Regression

- The parameters(thetas) are initialized with zeros.
- Convergence condition is reached when the difference between cost in the previous iteration and the cost in the current iteration is very small (0.00001)
- The BSOM data is split into training data (70%) and test data (30%) and the metrics on test data are used for performance evaluation.
- Missing values in the data are handled by removing the rows containing the missing values.
- While calculating precision, recall and F1 scores, macro averaging is considered.
- Ridge Regularization is used.

3) Logistic Regression with Multiple Variables

a) Logistic Regression is implemented with 2 independent variables ('all_mcqs_avg_n20' and 'all_NBME_avg_n4') and target variable 'LEVEL'. The target variable has 4 classes – A, B, C, D. Below is the plot of all the data points with their labels.



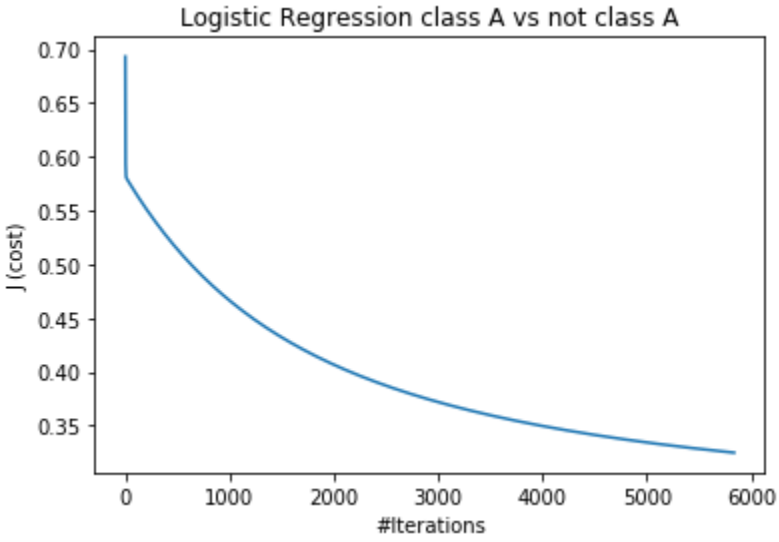
From the above plot, it can be observed that the data is imbalanced and the number of data points in each class are not same or almost same. The number of examples for class D is very less which may lead to underfitting.

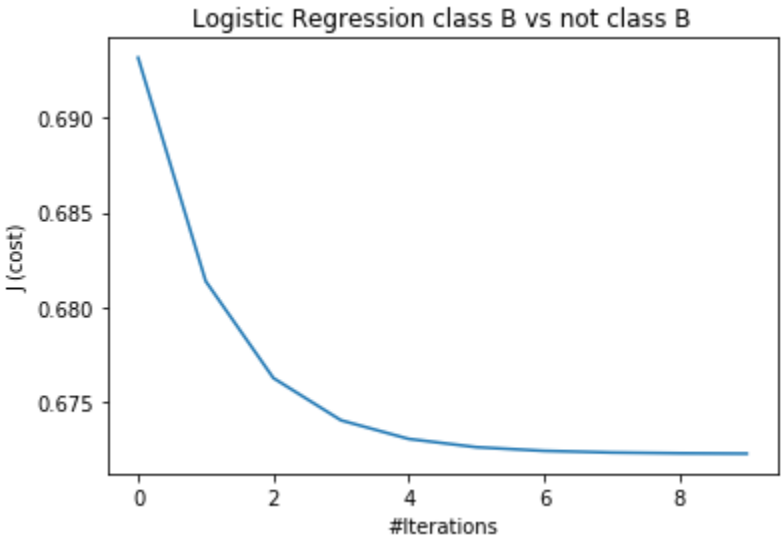
Gradient Descent algorithm is implemented with different learning rates on training data and the learning rate that gives more F1 score (accuracy) is considered as good learning rate to train the data and the parameters obtained after training the data with this optimum learning rate are used to predict the test data.

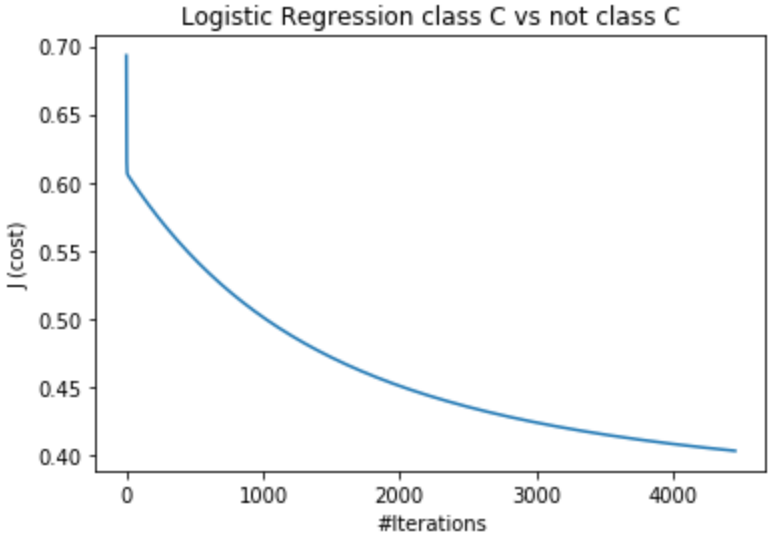
Learning Rate	**F1-score**
0.1	0.44578
0.5	0.52884
0.6	0.529076
0.7	0.51992

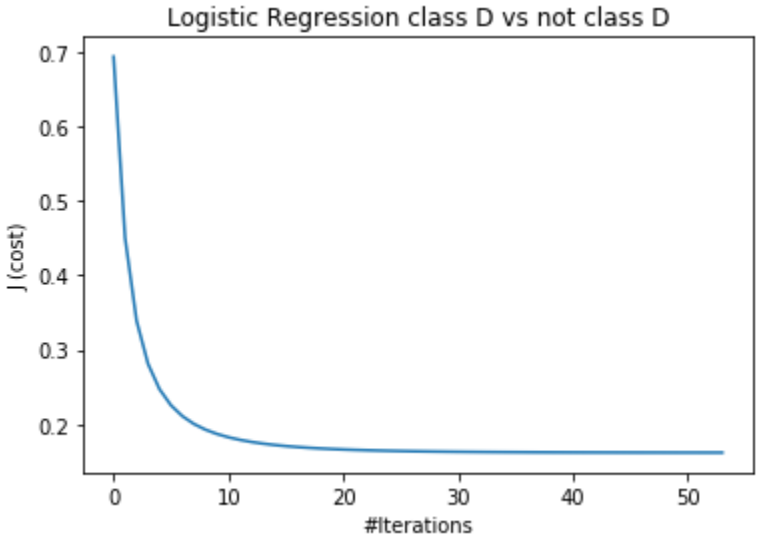
From the above table the F1 scores for learning rates 0.5 and 0.6 are almost same, but the algorithm converges fast with higher learning rate. So, we take 0.6 as good learning rate to apply on training data.

Below are the plots of cost function vs number of iterations for all the 4 classifiers with 0.6 as learning rate, to verify that the cost decreases with the increase in number of iterations.

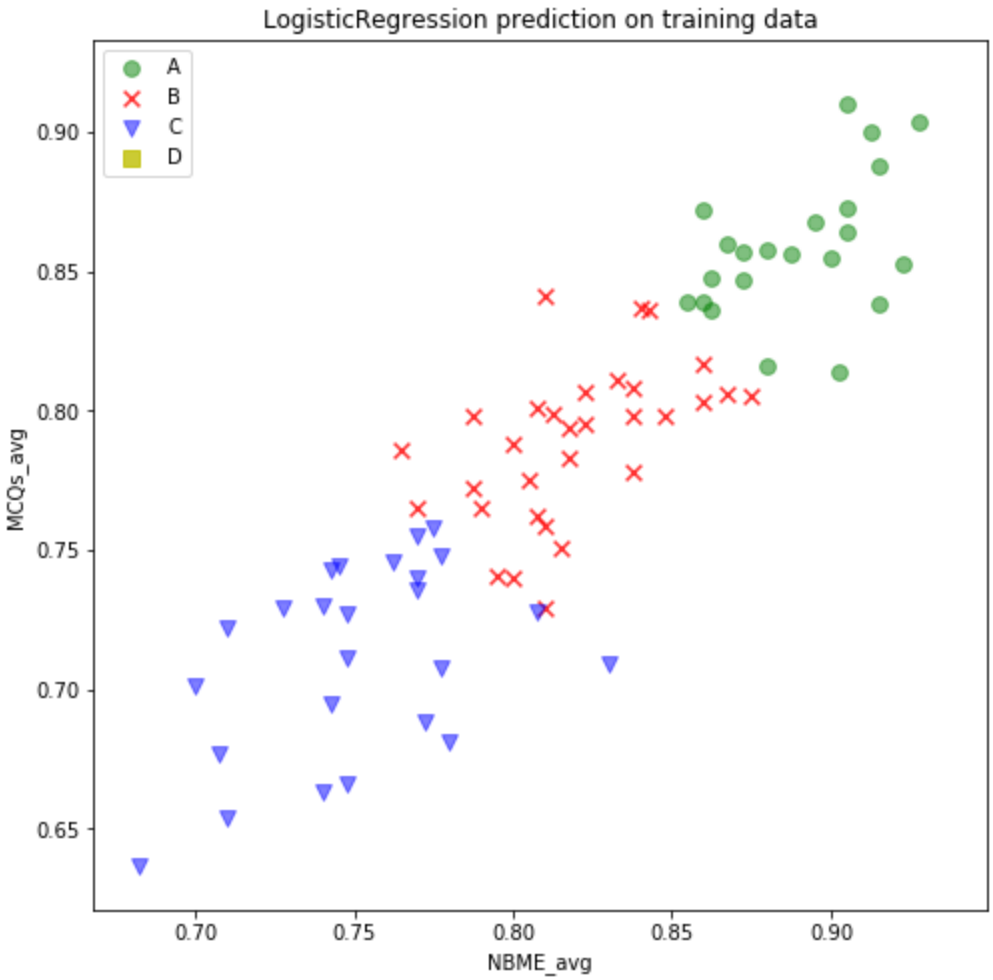








Below is the plot of data points in the training data set with the predicted labels.

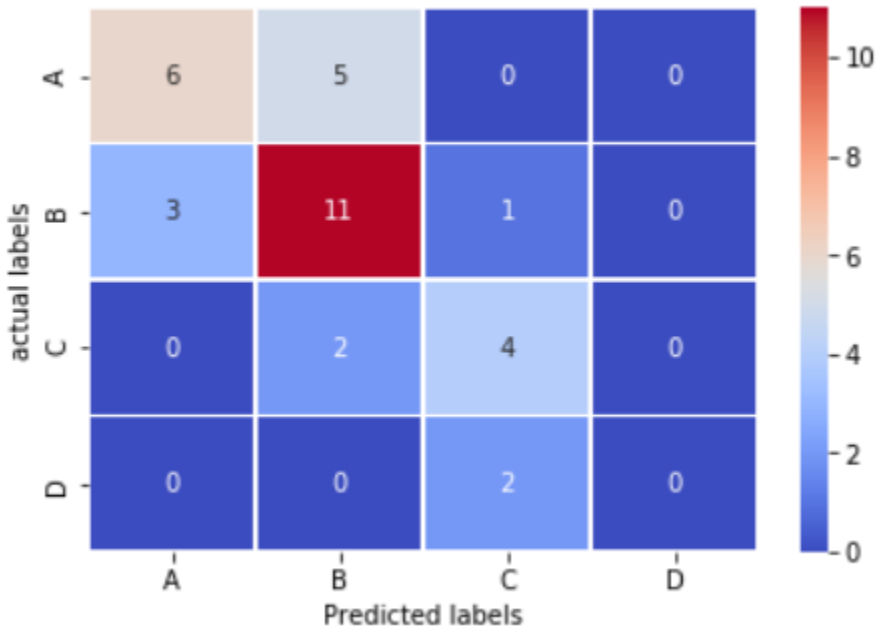


From the above plot it can be observed that class D is not present in the predicted labels. This is because the number of examples of class D are very less to train the data which lead to underfitting.

b) The optimum parameters obtained from training data are used to predict the test data.
Below metrics are used to evaluate the performance of test data.

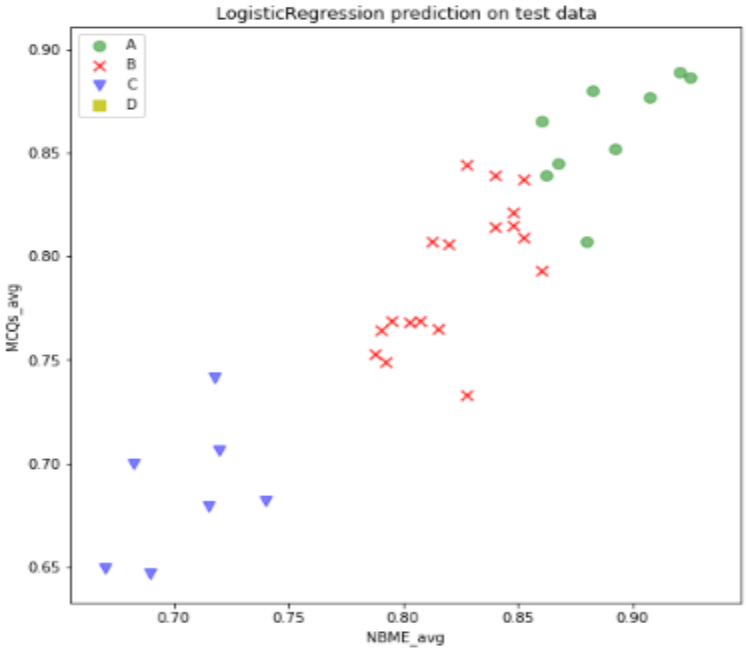
Precision: 0.462301
Recall: 0.486363
F1 score: 0.47051

Confusion Matrix

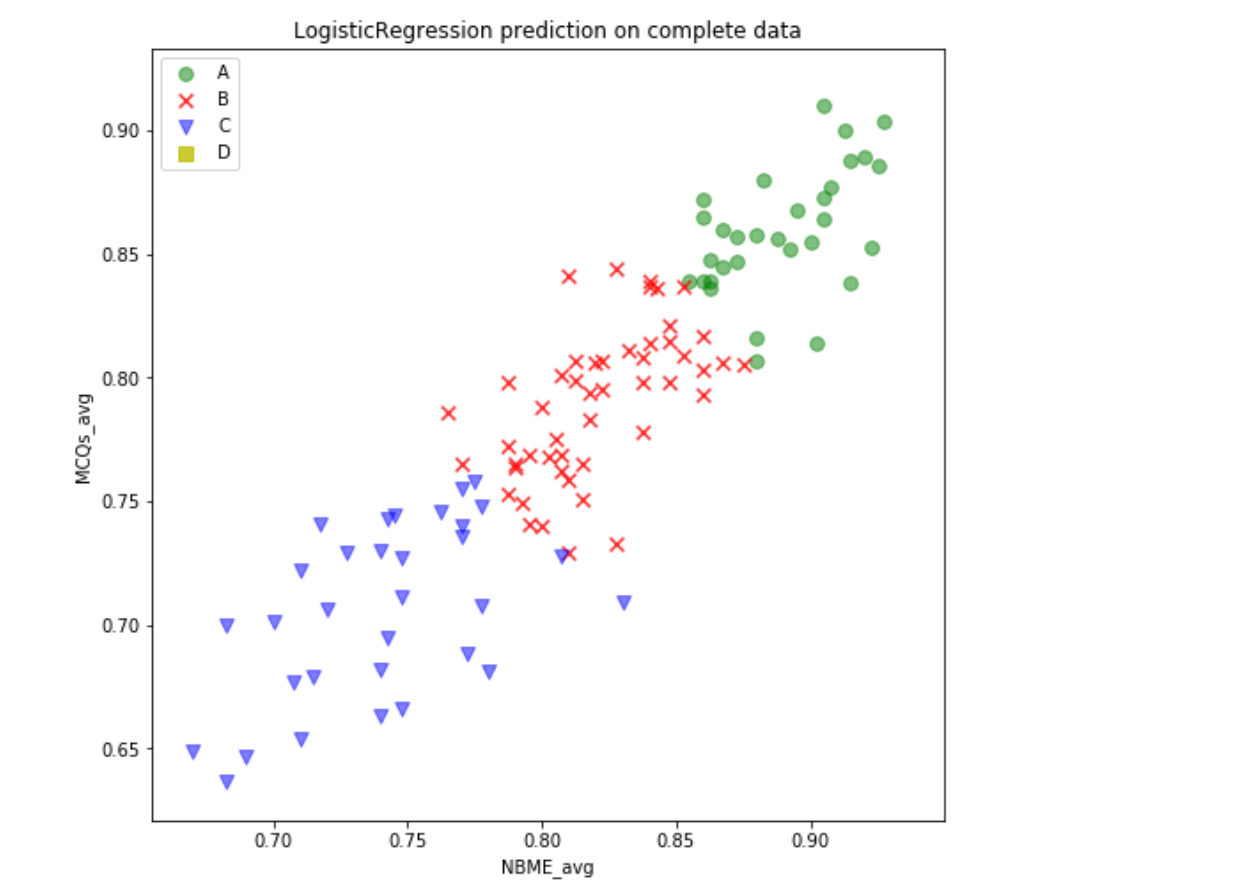


From the confusion matrix it can be observed that the number of correct predictions 21 out of 34. There are no predictions for class D due to underfitting. The data points in class A are not predicted as class C and the points in class C are not predicted as class A, which can also be observed from the plot of training data with predicted labels. In that plot, data points predicted as class A and class C are either side of class B. So, there may be a chance that points in class A can be predicted as class B and vice versa and points in class C can be predicted as class B and vice versa. But the points in class A will not be predicted as class C and vice versa.

Below is the plot for data points in test data with predicted labels.



Below is the plot for all the data points (both training and test) with predicted labels.



Level D points are not predicted due to underfitting. Students with low NBME average and low mcqs average scores are predicted as level C, students with high scores are predicted as level A and students with scores more than the scores of level C and less than scores of level A are predicted as level B.

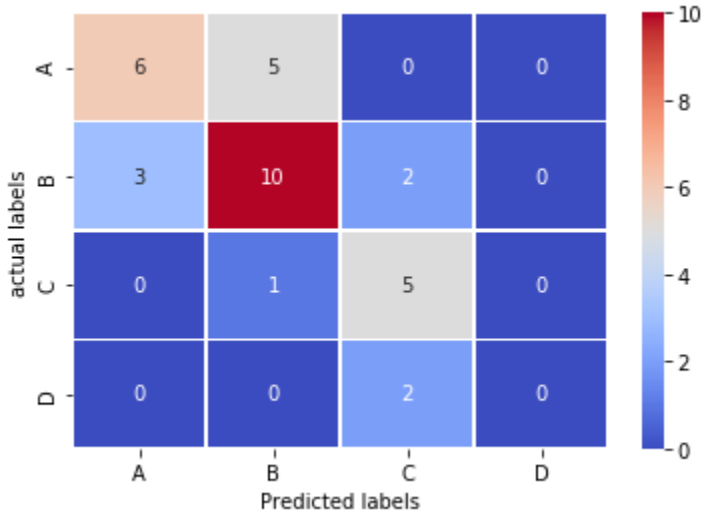
4) Logistic Regression with Feature Scaling and Regularization

a) Feature scaling is applied to the independent variables of training data and test data of the model in section 3 and the optimum coefficients obtained by implementing the Gradient Descent on training data with learning rate 0.6, are used to predict the test data. Below are the evaluation metrics of the test data prediction.

Precision: 0.4618055555555555
Recall: 0.5113636363636364
F1 score: 0.47795698924731184

From these metrics, it is observed that precision score is not improved by feature scaling and recall score is slightly increased and the increase in F1-score (accuracy) is very small.
So, there is no significant improvement in the performance of the model. I believe that one of the reasons for this is, both the features are almost in same scale.

Confusion Matrix:



Out of 11 actual values of class A, 6 values are predicted correctly. For B out of 15, 10 are predicted correctly. For class C, out of 6 ,5 are predicted correctly. For class D, out of 2, none is predicted correctly.

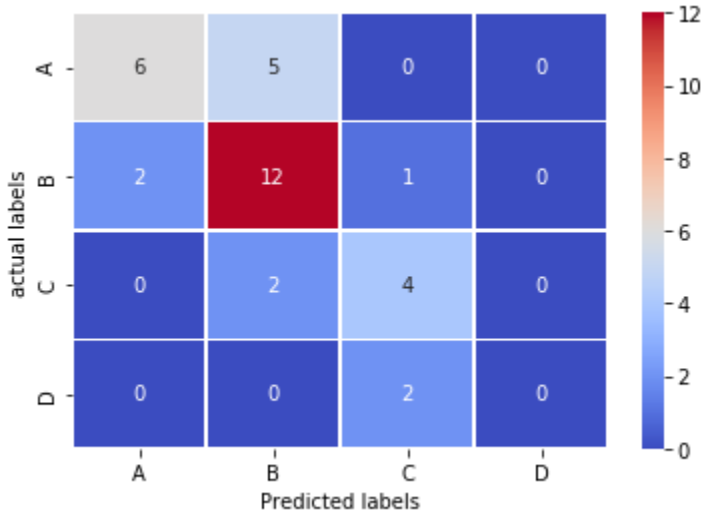
b) Ridge Regularization is applied to the model in section 3 with learning rate 0.6 and 6 different values of regularization parameter on training data and the parameter that gives more accuracy is taken as good regularization parameter. The optimum parameters selected from training the model are used to predict the test data. Below are the results.

Regularization parameter	Accuracy (F1-score)
0.01	0.52945
0.1	0.44878
0.5	0.142201
1	0.142201
10	0.142201
100	0.142201

From the above table, it can be observed that accuracy of the model on training data is more when the regularization parameter is 0.01.
Below are the metrics of test data prediction with the optimum parameters selected from training data, learning rate 0.6 and regularization parameter 0.01.

Precision: 0.4882518796992481
Recall: 0.503030303030303
F1 score: 0.4882114789235532

Confusion Matrix:



Out of 11 actual values of class A, 6 values are predicted correctly. For B out of 15, 12 are predicted correctly. For class C, out of 6 ,4 are predicted correctly. For class D, out of 2, none is predicted correctly.

c) From the metrics in section 4b, it can be observed that precision, recall and f1 score of the model in section 3 have been increased by 1% or 2 %with regularization. So, the performance of the model is increased by regularization. From the metrics in section 4a, it is observed that precision score is not improved by feature scaling and recall score is slightly increased and the increase in F1-score (accuracy) is very small. So, performance of the model is not increased by feature scaling.

5) Best Performance Model

a) In this section, best features are selected among the variables 'all_mcqs_avg_n20', 'all_NBME_avg_n4', 'CBSE_01', 'CBSE_02' to build the best model.
Feature Selection is done by comparing the accuracies (F1-scores) of different combination of features and the combination with best accuracy is selected.
Precision , Recall and F1Score on test data for different combinations of features, before applying feature scaling are evaluated as below.

1) all_mcqs_avg_n20
2) all_NBME_avg_n4
3) CBSE_01
4) CBSE_02

Table 1: Before Feature Scaling

Features selected	Precision	Recall	F1-Score
1,2,3,4	0.1102941	0.25	0.15306122448
1,2,3	0.1102941	0.25	0.15306122448
1,2,4	0.1102941	0.25	0.15306122448
1,3,4	0.3887867	0.3893939	0.207201
2,3,4	0.110294	0.25	0.1530612
1,2	0.462301	0.486363	0.4705128
2,3	0.1102941	0.25	0.15306122448
3,4	0.1102941	0.25	0.15306122448
1,3	0.1102941	0.25	0.15306122448
2,4	0.1102941	0.25	0.15306122448
1,4	0.1102941	0.25	0.15306122448

Precision , Recall and F1Score on test data for different combinations of features, after applying feature scaling are evaluated as below.

Table 2: After Feature Scaling

Features selected	Precision	Recall	F1-Score
1,2,3,4	0.52716503	0.567424242	0.5410714
1,2,3	0.4421977	0.4886363	0.4544956
1,2,4	0.55555	0.584090	0.55975

1,3,4	0.55555	0.584090	0.559751
2,3,4	0.55555	0.584090	0.55975
1,2	0.4618055	0.511363	0.477956
2,3	0.4206349	0.446969	0.4303613
3,4	0.5384469	0.592424	0.544014
1,3	0.415209	0.478030	0.4354707
2,4	0.56964869	0.6068181	0.5816964
1,4	**0.600694**	**0.6234848**	**0.601218**

Precision , Recall and F1Score on test data for different combinations of features, after applying feature scaling and Regularization are evaluated as below.

Table 3: After Feature Scaling and Regularization

Features selected	**Precision**	**Recall**	**F1-Score**
1,2,3,4	0.52716503	0.567424242	0.5410714
1,2,3	0.4421977	0.4886363	0.4544956
1,2,4	0.55555	0.584090	0.559751
1,3,4	0.55555	0.584090	0.559751
2,3,4	0.551948	0.51363	0.499285
1,2	0.519642	0.519696	0.5062271
2,3	0.4206349	0.446969	0.4303613
3,4	0.5384469	0.592424	0.544014
1,3	0.415209	0.478030	0.4354707
2,4	0.54375	0.590151	0.56259600

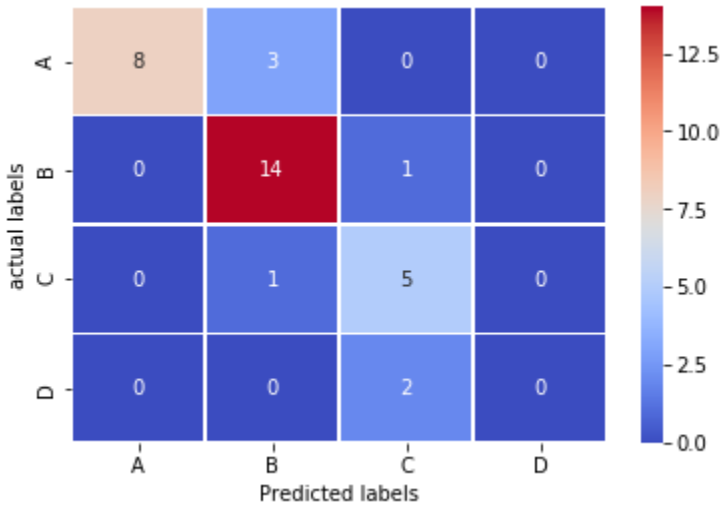
1,4	0.600694	0.6234848	0.601218

By comparing the accuracies of different combinations of features after feature scaling, it is observed that the model with features 'all_mcqs_avg_n20' and 'CBSE_02' is having more accuracy. So, we select these two features to build the best model. And also, it can be observed from Tables1, 2 and 3 that the accuracy of the model is increased after feature scaling, but the accuracy is not increased by applying regularization on feature scaled variables. So, we apply only feature scaling to the selected features to build a best model.

b) Now best model is built with 2 selected features- 'all_mcqs_avg_n20' and 'CBSE_02' and feature scaling is applied. After feature scaling, learning rate of 0.6 is used to train the model to get optimum parameters and these optimum parameters are used to predict the test data. Below are the results of prediction on test data.

Precision: 0.6006944444444444
Recall: 0.6234848484848485
F1 score: 0.6012189564821144

Confusion Matrix:



Out of 11 actual values of class A, 8 values are predicted correctly. For B out of 15, 14 are predicted correctly. For class C, out of 6 ,5 are predicted correctly. For class D, out of 2, none is predicted correctly.

By comparing the precision, recall and f1 scores of the model in section 3, section 4, different combination of features in section 5a, it can be observed that the model built with the features selected in 5a is having more accuracy.