Machine learning – Assignment 2

[redacted] [redacted]

Initial settings: Training and Test data split is in ratio 80:20

1. Linear Regression with One Variables (50 points):

a. Can you demonstrate linear regression using 'all_mcqs_avg_n20' and 'STEP_1'? Note, here 'STEP_1' is the target variable.

Preprocessing:

>Data is Normalized with Z-normalization.
>Initial theta as random in range (0,1)
>Replacing missing values in 'STEP_1' with mean of the values.

Demonstration: The Model initial settings are alpha=0.1 and iterations =1000.The Final model obtained has intercept= -0.02170789 and theta_1= 0. 7006554257799749.The Figure 1 shows the visual representation of Linear regression model.

The Model obtained with the algorithm matches with the sklearn toolkit model which can be observed in the Figure 1.

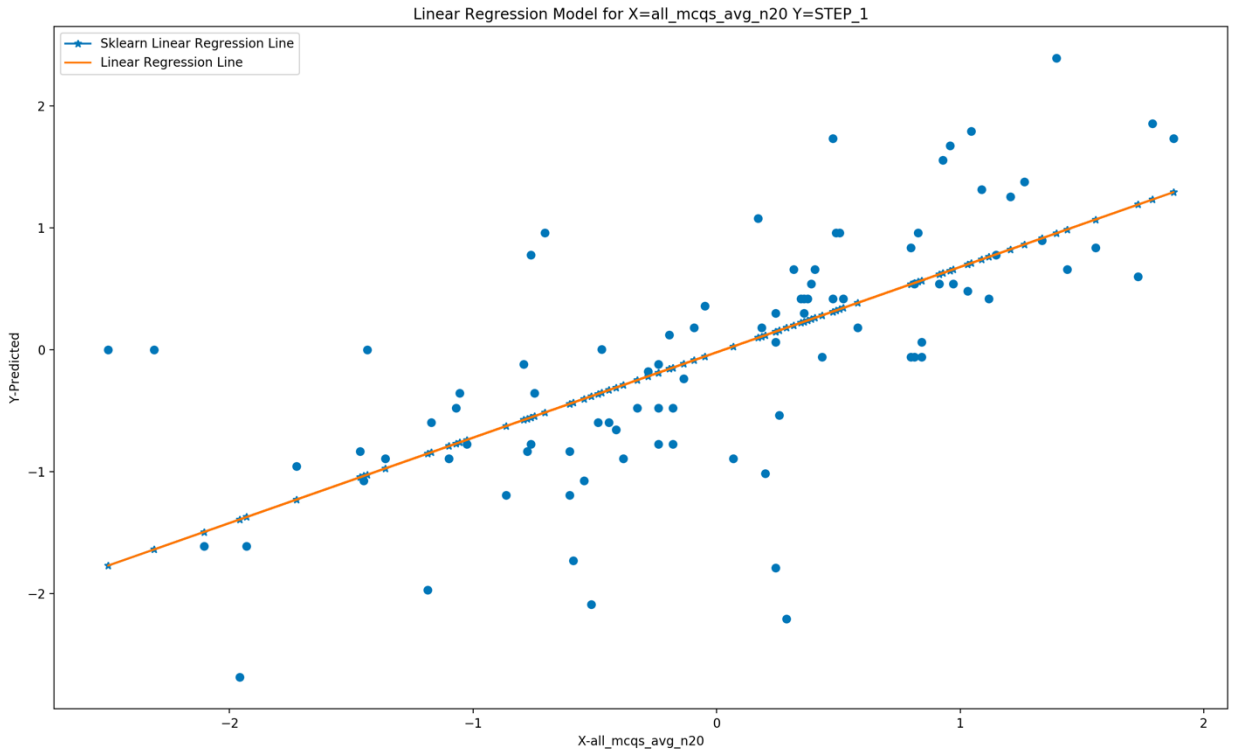


Figure 1:Linear regression Model

Residual vs fit plot: The plot is used to detect non-linearity, unequal error variances, and outliers. Since the data points are widespread across horizontal axis near to zero residual the Figure 2 describe the Linear regression model is appropriate for the given data.

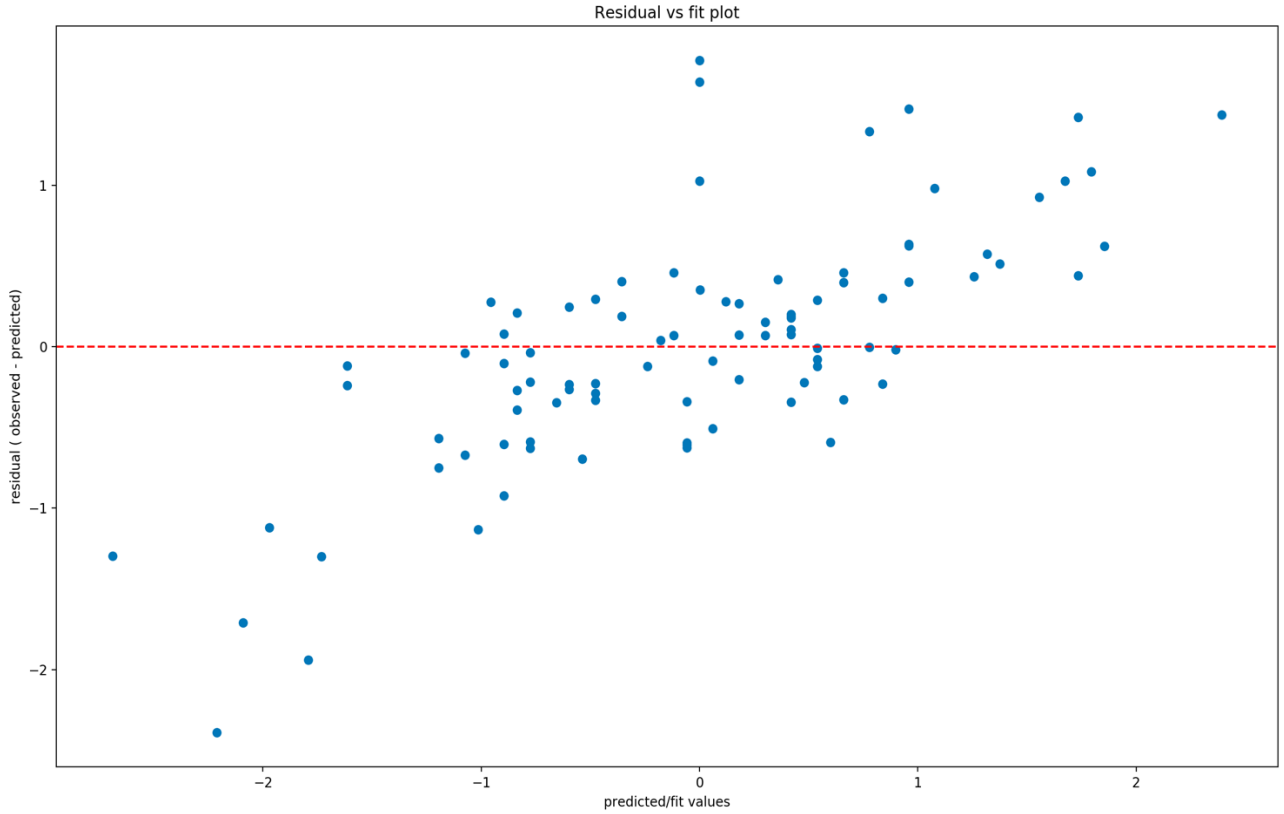


Figure 2: Residual vs Fit Plot

b. Evaluate performance using metrics (such as Mean Squared Error, Pearson correlation coefficient and R2). You may also use graphs for explaining your observations.

Performance evaluation is done by making prediction on test data.

Analysis:

Algorithm designed:

Mean squared error: 0.29938112680039525 indicates the error in prediction by model is around 29.9%.

Pearson correlation coefficient: **0.8117732425771855** indicates there is **strong correlation between observed** values and predicted values of the model developed.

R2: **0.6449564548985776** indicates the strength of the observed values and the model is 64.49%. Means 64.49% of observed outcomes are corrected predicted by the model.

Metrics for predictions from sklearn:

Mean squared error: 0.2993811268003953

Pearson correlation coefficient: **0.81177324**

R2: **0.6449564548985776**

2. Linear Regression with Two Variables (10 points):
a. Does adding 'all_NBME_avg_n4' as input improve the performance of the previous model? Please use evaluation metrics or graphs to compare the performance of Question 1 and 2.

Anaysis: Adding **'all_NBME_avg_n4' did not improve the performance of the previous model**.

Metrics:
Mean squared error: **0.3275302321329594** indicates the error in prediction by model is around 32.7% which is higher than the value from model with one feature. Since the mean squared error is increases r and R2 value we can observe decrease in 'r' and 'R2' values.
Pearson correlation coefficient: **0.8074534859268194** indicates there is **strong correlation between observed** values and predicted values of the model developed however it is lower than the model with one variable.
R2: **0.6115737288212244**indicates the strength of the observed values and the model is 61.11%. Means 61.11% of observed outcomes are corrected predicted by the model.

The above metrics can be visualized with figure 3.

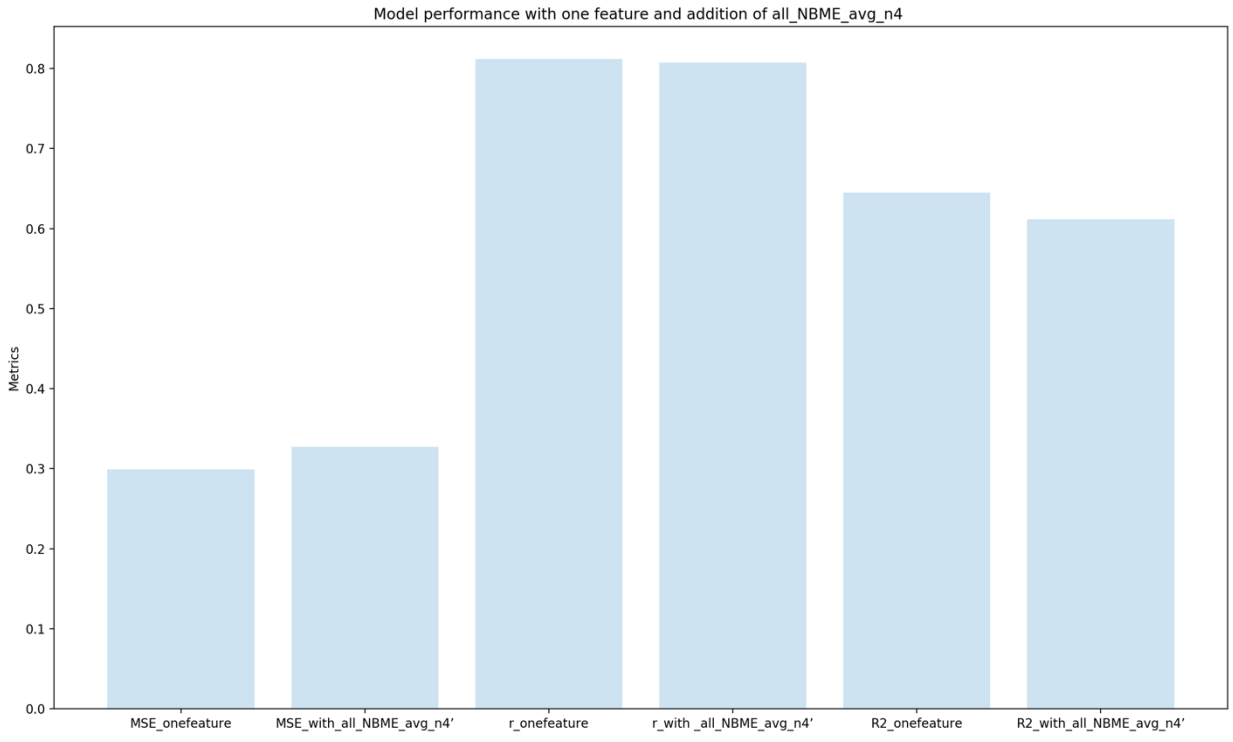


Figure 3:Bar plot showing increase in MSE followed by decrease in 'Pearson correlation coefficient' and "R2" by adding new feature to the Model.

3. Logistic Regression with Multiple Variables (50 points):
a. Can you demonstrate logistic regression using 'all_mcqs_avg_n20', 'all_NBME_avg_n4' and 'LEVEL'? Note, here 'LEVEL' is the target variable.

Preprocessing:
Given data set is highly imbalanced. Data of Level B is higher in number and Level D is lower in number so chances of not having Level D record in test data is higher, which can result in incorrect metric evaluation.

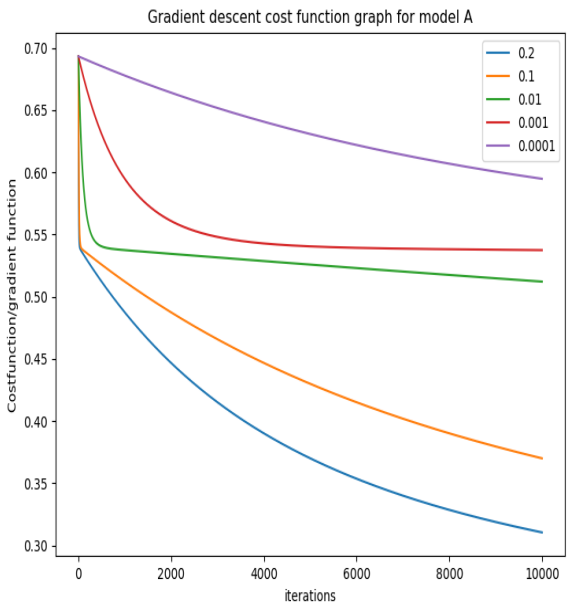
So, upscaling the data such that the test data can get all Levels of data records during data split. However, upscaling might result in overfit as few data samples that are duplicated in train data can be available in test data as well. But risk of model giving false high performance is considered as priority. So, **upscaling the data for logistic regression model demonstration**. With upscaling the data to more frequent label and removing of missing 'LEVEL' records, the size of the data is increased from 115 to 184.

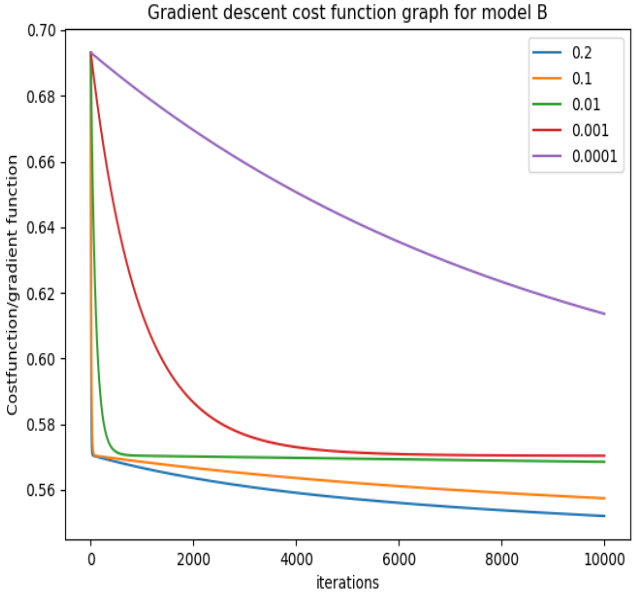
Demonstration:

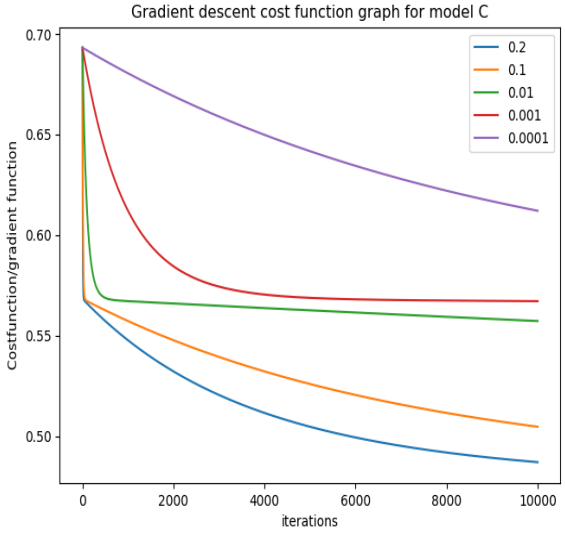
Demonstrating the model from initial best parameter selection to visualization of Logistic regression Model.

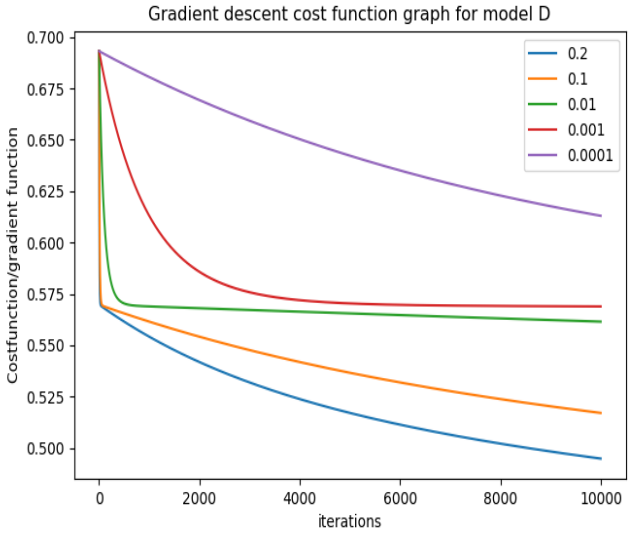
Parameter selection:

Determining the best parameters for the model based on cost function. The goal is to find the values of model parameter (alpha/learning rate) for which Cost Function is minimum. From below four figures it is evident that for each model A, B, C, D the cost function is tending towards lowest for alpha value 0.2.

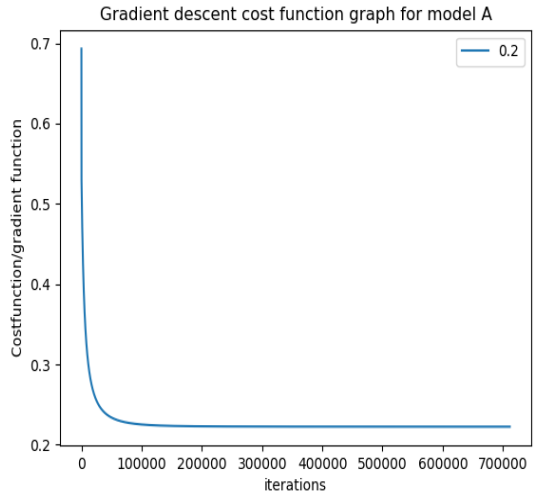


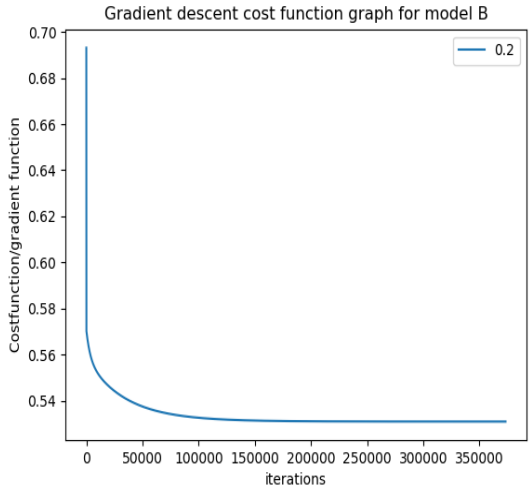


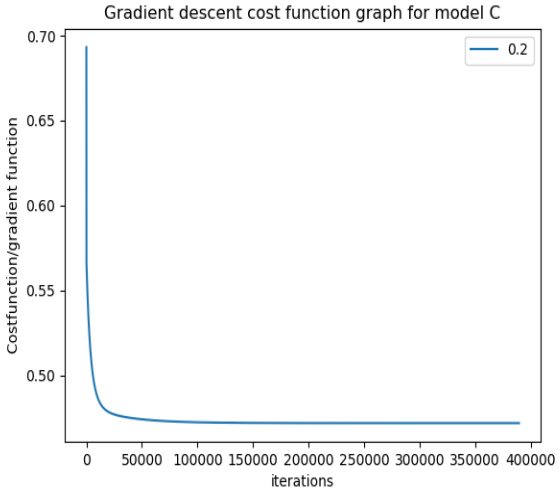


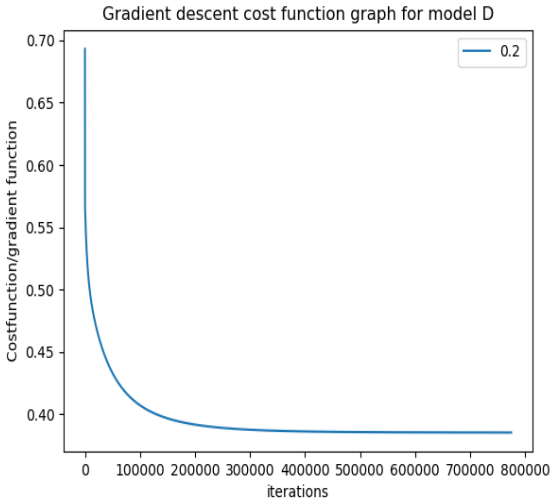


Note: Since alpha value is small and data values are not scaled and with tolerance of 0.0000001 and initial theta as 0 , the algorithm stops for maximum iteration 10000 so while building the logistic regression model we increase iteration to 1000000 which meet convergence condition for iterations around 720000 for Model A,370000 for Model B,380000 for Model C,780000 for model D which is shown in below four Figures .So **model is set with alpha = 0.1,intial thetas = 0(count depends on number of independent variables given),tolerance =0.00001 and iterations for A,B,C,D as mentioned above**.









Logistic Regression Model parameters and associated model coefficients:

Regression Model A: alpha = 0.2, initial thetas = [0,0,0] for [‘bias term x0’, ‘all_mcqs_avg_n20’, ‘all_NBME_avg_n4’], tolerance =0.0000001, iterations=750000

Coefficient of Model A, [[-46.14588212] [19.61508295] [34.58081207]].

Regression Model B: alpha = 0.2, initial thetas = [0,0,0] for [‘bias term x0’,‘all_mcqs_avg_n20’, ‘all_NBME_avg_n4’],tolerance =0.0000001,iterations=390000

Coefficient of Model B, [[-7.28605781] [-15.6319688] [22.70452266]]

Regression Model C: alpha = 0.2, initial thetas = [0,0,0] for ['bias term x0','all_mcqs_avg_n20', 'all_NBME_avg_n4'],tolerance =0.0000001,iterations=400000

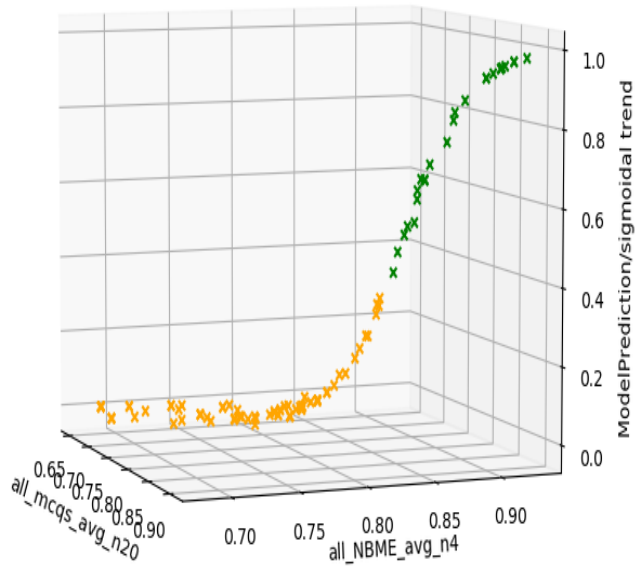
Coefficient of Model C, [[11.11798356][-20.74492184][4.51528918]]

Regression Model D: alpha = 0.2, initial thetas = [0,0,0] for ['bias term x0','all_mcqs_avg_n20', 'all_NBME_avg_n4'],tolerance =0.0000001,iterations=800000

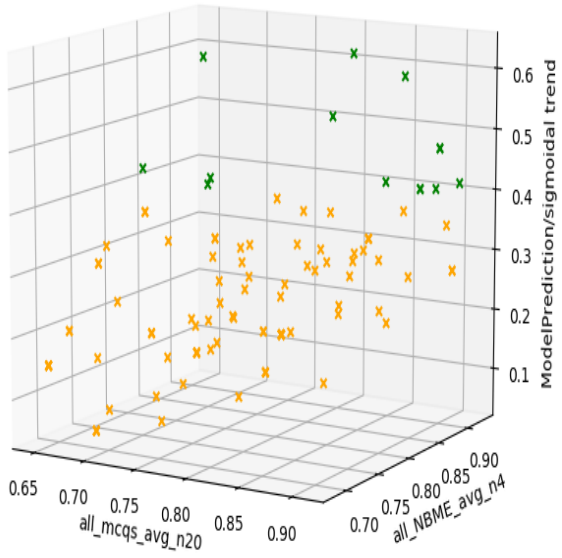
Coefficient of Model D, [[17.73540533] [46.266766] [-68.79908734]]

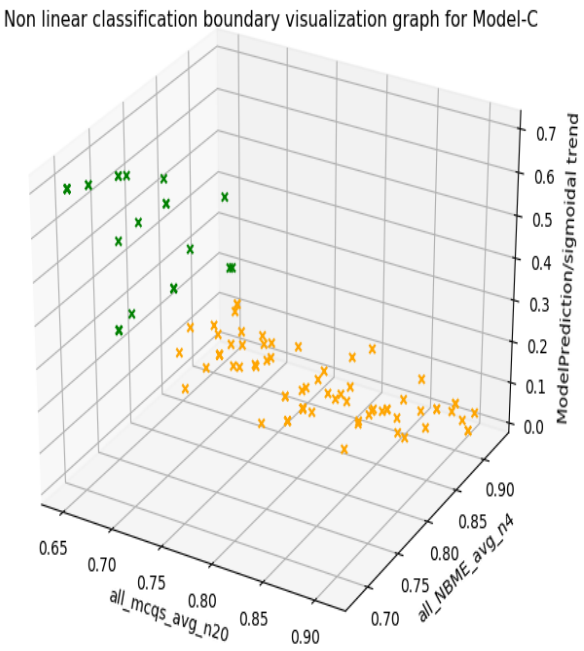
The 3D plots below shows Logistic regression trend of the model A, B,C and D with features along X and Y axis and Logistic Model trend (sigmoid for given theta) along Z axis: For Model -A is can be observed that it can predict A vs not A more clearly with the sigmoidal trend .For Model B,Model D the trends of sigmoidal is scattered however we can observe a clear boundary decision can be made between labels '0' and '1' for D compared to B .For Model C it is neither randomly scattered like Model B and D nor completely structured like Model A.Hence Predictions accuracy from Model A and C can be better compared to Model D and B.

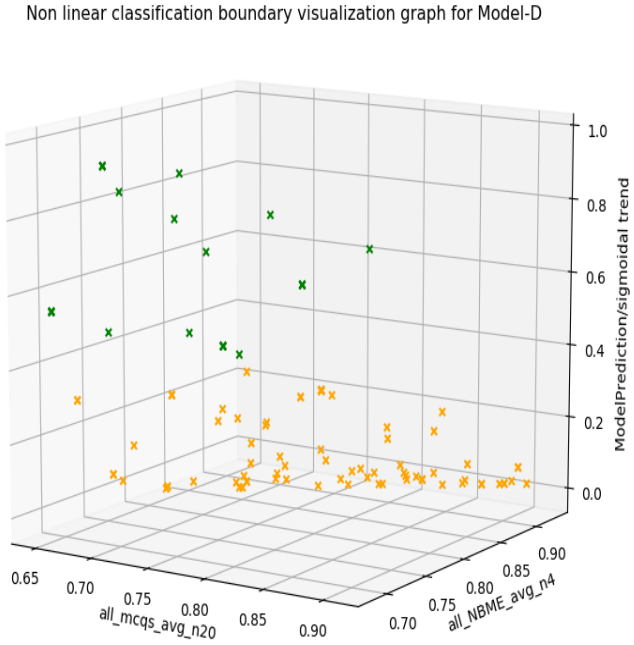
Non linear classification boundary visualization graph for Model-A



Non linear classification boundary visualization graph for Model-B







b. Evaluate performance using metrics (such as confusion matrix, precision, recall and F1 scores). You may also use graphs for explaining your observations.

Evaluating the performance on test data in comparison with train accuracy. Combined performance of Logistic regression model for all four classification is as follow.

Analysis :From the Figure 4 ,it can be analyzed that the model is having 0.69 precision indicating , 69% of positive values (classification as Level A,B,C D)predicted is correct and 0.64 recall means, the model predicts 64% of total positive values and F1-Score of 61%.Both the precision and recall are nearly moderate .It can be **analyzed that the classes are moderately handled by the model means the model is neither poor nor very good w.r.t training data**. However, performance of the Model is more reliable based on metrics from unseen data.

Now we get true Model metrics from Figure 5 metrics of test/unseen data. The precision recall and F1-Score are nearly 48%,51% and 48% on the new data. Since these metrics are less compared to metrics on train data from Figure 4, we can say the model is not overfitting However the **performance of the classifier is slightly poor on unseen data**.

```
Algorithm confusion matrix
 [[33  0  3  0]
 [14  9  6  6]
 [ 0  1 20 14]
 [ 0  0  8 33]]
algorithm Precision,recall,f-score train data
 (0.6913274273872427, 0.6375290360046457, 0.6132159845053975, None)
              precision    recall  f1-score   support

           0       0.70      0.92      0.80        36
           1       0.90      0.26      0.40        35
           2       0.54      0.57      0.56        35
           3       0.62      0.80      0.70        41

   micro avg       0.65      0.65      0.65       147
   macro avg       0.69      0.64      0.61       147
weighted avg       0.69      0.65      0.62       147
```

Figure 4:Confusion matrix and precision, recall, F1-score values of the trained Model.

```
Algorithm confusion matrix
 [[7 0 1 2]
 [5 4 1 1]
 [0 4 4 3]
 [0 0 2 3]]
algorithm Precision,recall,f-score test data
 (0.4791666666666667, 0.5068181818181818, 0.4767600820232394, None)
              precision    recall  f1-score   support

           0       0.58      0.70      0.64        10
           1       0.50      0.36      0.42        11
           2       0.50      0.36      0.42        11
           3       0.33      0.60      0.43         5

   micro avg       0.49      0.49      0.49        37
   macro avg       0.48      0.51      0.48        37
weighted avg       0.50      0.49      0.48        37
```

Figure 5: Confusion matrix and precision, recall, F1-score values of the Model w.r.t unseen data.

4. Regularization and Feature Scaling (20 points):

a. Does Feature Scaling improve the performance for the model in Question 3?

Solution: No. feature scaling did not improve the performance of the model. Z-score feature scaling been done here.
It is observed from Figure 7 a bar graph plot for F1score on raw and scaled data.

Analysis: From figure 6 there is **no high variance in distribution of data** for given features, hence the scaling did not improve the Performance

(115, 201)

	all_mcqs_avg_n20	all_NBME_avg_n4
count	115.000000	115.000000
mean	0.781339	0.812565
std	0.068596	0.067605
min	0.610000	0.615000
25%	0.738000	0.770000
50%	0.793000	0.815000
75%	0.838500	0.861250
max	0.910000	0.927500

Figure 6:Distribution of data for "all_mcqs_avg_n20", "all_NBME_avg_n4"

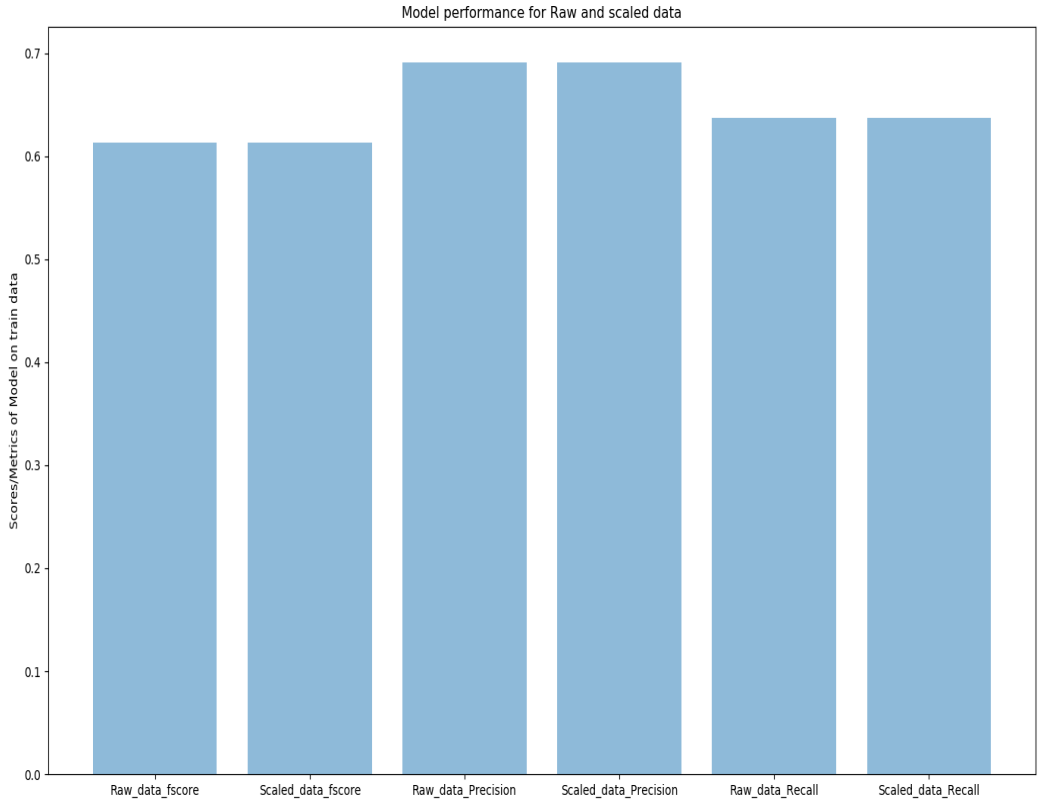


Figure 7: Bar plot of F1Score for Raw and scaled data.

b. Does regularization improve the performance for the model in Question 3? Test at least 5 different regularization values to support your answer.

Solution: **Yes, Model performance is improved** at one penalty value. We are performing Ridge regularization with different penalty value as evidence to the solution.

Analysis: **Regularization means adding or increase bias if our model suffers from higher variance which works better on unseen data rather than known data**. It can be observed from Figure 8 **the model performance did improve on unseen data for lambda = 0.0001 and no improvement on training data** as shown in figure 7**.** Decision on improvement with regularization technique can be reliable only from unseen data. Figure 8 can be taken as evidence for the same.

Note: Here lambda value zero mean no regularization.

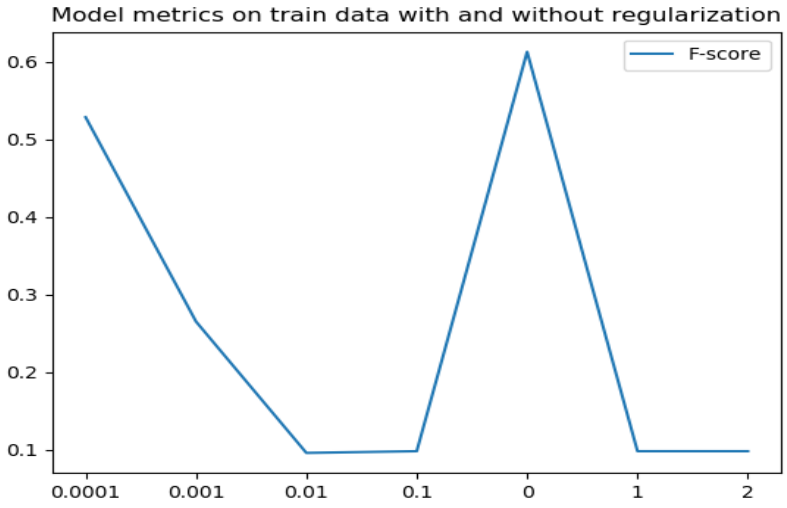


Figure 7:Model scores for different penalty values on train data

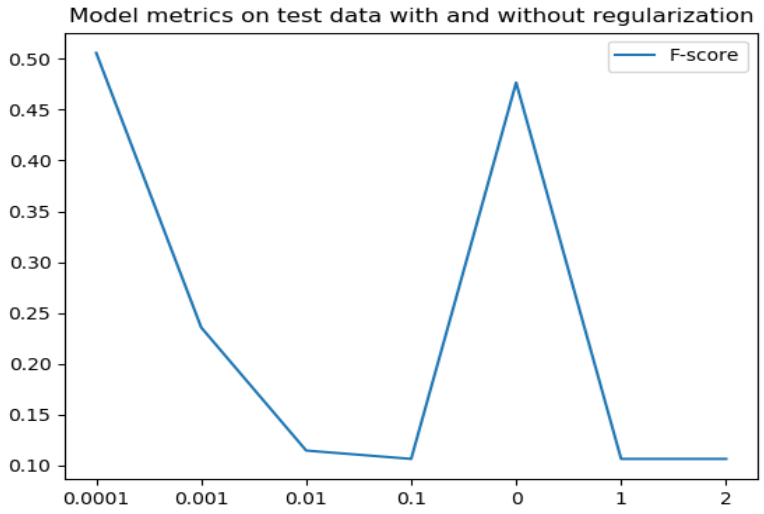


Figure 8: Model -score for different penalty values on train data

c. Evaluate performance for each case using metrics (such as confusion matrix, precision, recall and F1 scores).

Scaling Evaluation:
Scaling did not improve the model performance it can analyzed with Precision, recall and F1 scores of test data with scaling and without scaling.
The metrics from Figure 9 is similar to metrics from Figure 5 without scaling.

```
Algorithm confusion matrix
 [[7 0 1 2]
 [5 4 1 1]
 [0 4 4 3]
 [0 0 2 3]]
algorithm Precision,recall,f-score test data
 (0.4791666666666667, 0.5068181818181818, 0.47676008202323994, None)
              precision    recall  f1-score   support

           0       0.58      0.70      0.64        10
           1       0.50      0.36      0.42        11
           2       0.50      0.36      0.42        11
           3       0.33      0.60      0.43         5

   micro avg       0.49      0.49      0.49        37
   macro avg       0.48      0.51      0.48        37
weighted avg       0.50      0.49      0.48        37
```

Figure 9: Confusion matrix and precision, recall, F1-score values of the Model w.r.t unseen data without scaling

Regularization Evaluation: Analysis for statement "Regularization improved model performance" is validated with below metrics. The Model with regularization has following Precision, Recall, F1-score.

>**lambda 0.0001:**
Train data: (0.5306840450796981, 0.5481562137049942, 0.5289879109538552)
Test Data: (0.5232371794871795, 0.5022727272727273, 0.505947634237108)
> **lambda 0.001:**
Train Data: (0.20789473684210524, 0.37857142857142856, 0.2655483870967742)
Test Data: (0.19015151515151515, 0.31363636363636366, 0.23575757575757578)
> **lambda 0.01:**
Train Data: (0.05952380952380952, 0.25, 0.09615384615384615)
Test Data: ((0.07432432432432433, 0.25, 0.11458333333333333)
>**lambda 0.1:**
Train Data: (0.061224489795918366, 0.25, 0.09836065573770492)
Test Data: (0.06756756756756757, 0.25, 0.10638297872340427)
>**lambda 0:** No regularization
Train Data: (0.6913274273872427, 0.6375290360046457, 0.6132159845053975)
Test Data: (0.4791666666666667, 0.5068181818181818, 0.47676008202323994)

>**lambda 1:**

Train Data: (0.061224489795918366, 0.25, 0.09836065573770492)
Test Data: (0.06756756756756757, 0.25, 0.10638297872340427)

>**lambda 2:**

Train Data: (0.061224489795918366, 0.25, 0.09836065573770492)
Test Data: (0.06756756756756757, 0.25, 0.10638297872340427)

Analysis: Since the model improve at lambda = 0.0001 so we evaluate our model with 0.0001. Precision is around 0.53068 which says it detects 53% of true data of all the predicted values. Recall 0.548 means it predict 54% true data of total actual true values. So, over all model performance is 52% (F1-Score) which is considered as moderate model. **Model has Moderate performance with and without regularization on training data**. Since evaluating regularization on unseen data can give us appropriate process for performance evaluation. we analyze the metric of regularized test data w.r.t test data of unregularized model. Without regularization over all model F1-Score is 47.6% which can be classified as poor classifier with lambda =0.0001 and the overall model F1-score is 50.59% which is classified as **slightly moderate but not poor**.

5. Build the best performance model (20 points):

a. **Select features from input variables 'all_mcqs_avg_n20', 'all_NBME_avg_n4', 'CBSE_01', and 'CBSE_02' to build the best performance logistic regression model to predict target variable 'LEVEL'. Also use feature scaling and regularization techniques if that improves the model performance.**

Solution: Feature set 1: 'all_mcqs_avg_n20','CBSE_01','CBSE_02' feature set for best model performance.

To build best performance logistic model we start with best feature selection. Figure 10 shows the correlation between the all four features. Size of each square indicate the degree of correlation and each color indicate intensity within same degree of correlation (red indicate low and blue indicate high). Initially based on this correlation plot we try to eliminate the features with high correlation. Feature 'all_mcqs_avg_n20', 'all_NBME_avg_n4' are highly correlated we try not to consider these features together with any combination set.

Feature set 1:

We can consider 'CBSE_01','CBSE_02' and 'all_mcqs_avg_n20' by eliminating 'all_NBME_avg_n4' (it has high correlation with remaining three features)

Feature set 2:

CBSE_01', 'CBSE_02' – This could be possible best features as correlation is lower among all.

Feature set 3:

'CBSE_01', 'all_mcqs_avg_n20' - This could be another possible best feature as correlation is lower among all

Feature set 4:

'CBSE_02' and 'all_mcqs_avg_n20' – This pair appear little higher correlation when compared with feature 3 and feature 4.

We try to experiment for above four feature set because since the we're not aware of correlation factor of combination with Dependent variable or 'Level'. We choose the best model based on F1-Score

performance on training data. Later we can evaluate the best model performance on test data for reliable metrics.

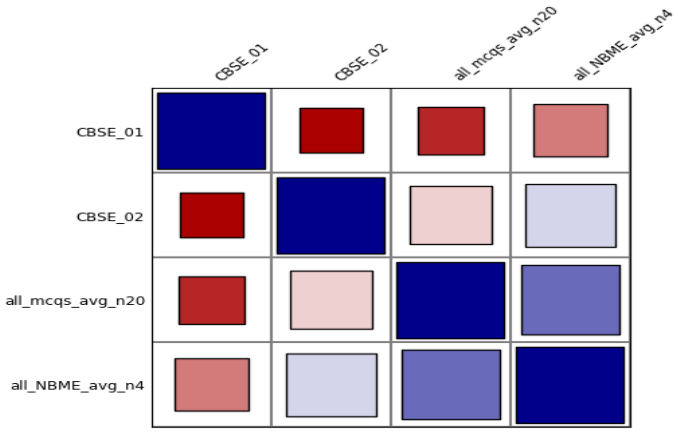


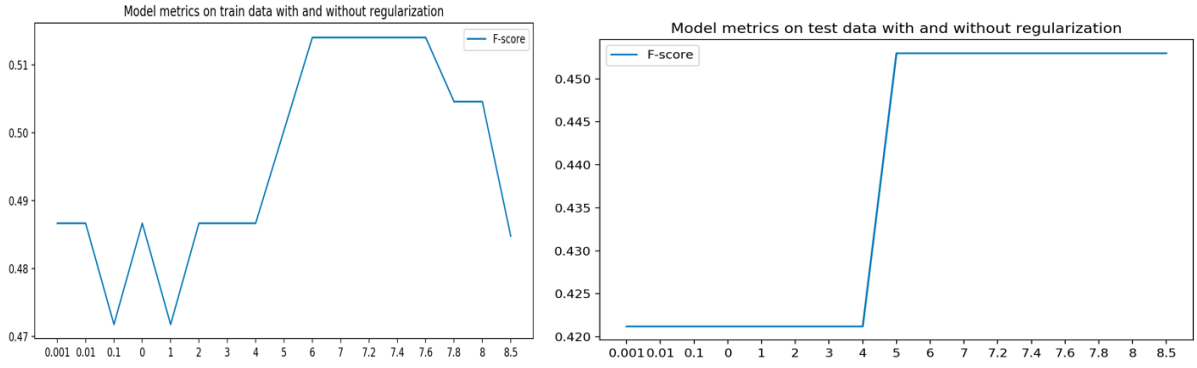
Figure 10:Correlation between features

Best Model building:

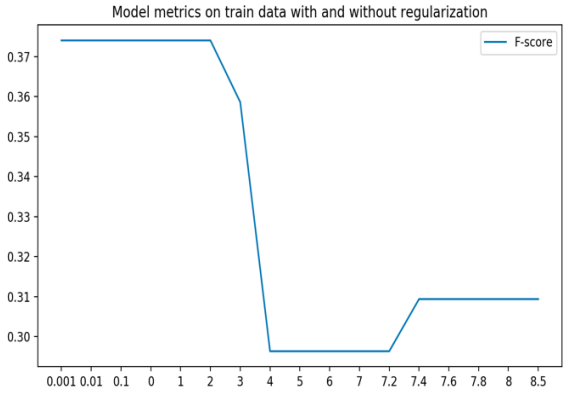
Without scaling the f1-score of the model is around 10% to 25% for above feature sets as data distribution has high variance. So, post scaling and cost function analysis over several iterations, alpha and iterations are fixed for optimal value of alpha =0.001 and iterations=10000.
Now we apply regularization and see performance of each feature set and select best among them. From below figures, in general for all features there is improvement in performance with regularization except for feature 2. Since majority of feature set, performance is affected by regularization. We use **regularization technique as a part of best model building process**.

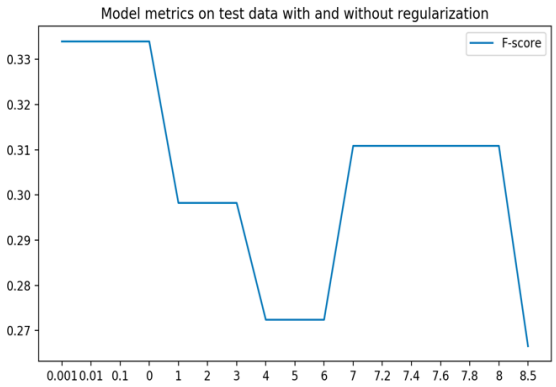
Below are the 8 plots for different feature sets for different values of penalty. From below plots we can say the best score for the model is obtained for feature set 1.

Feature set 1: 'all_mcqs_avg_n20','CBSE_01','CBSE_02'

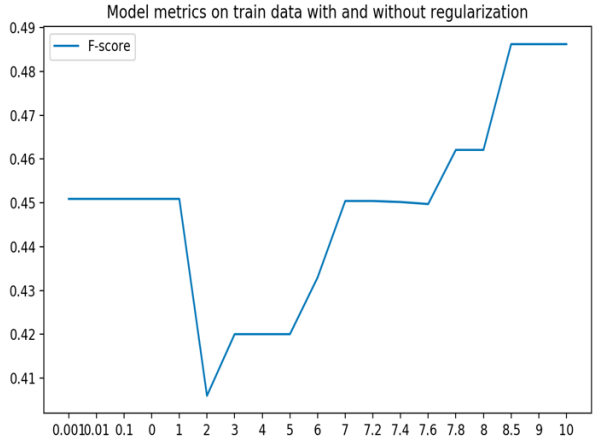


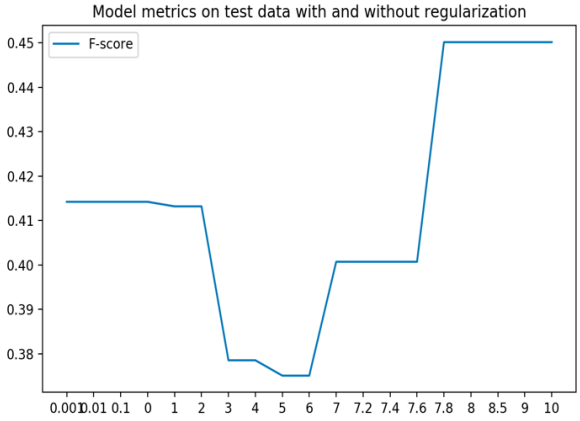
Feature set 2: CBSE_01', 'CBSE_02':



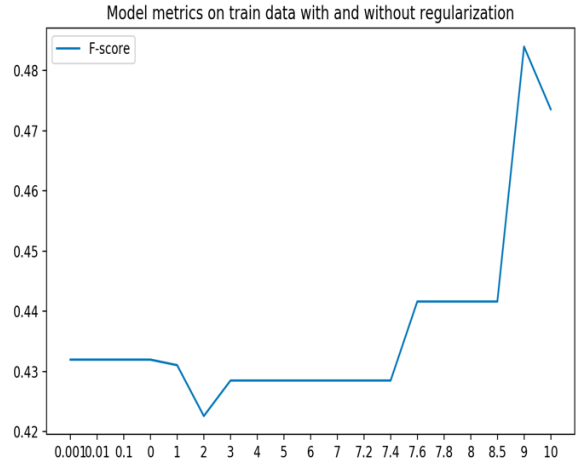


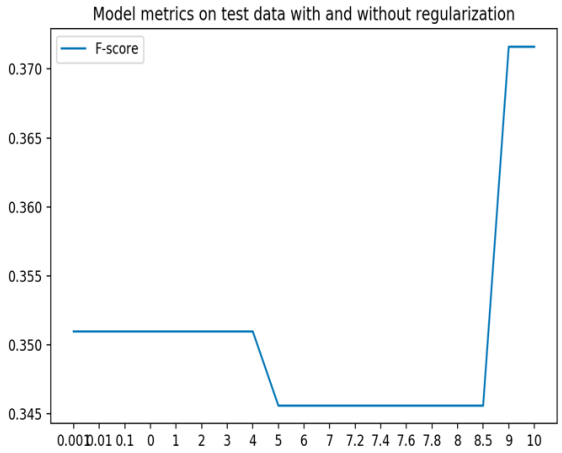
Feature set 3: all_mcqs_avg_n20', 'CBSE_01:





Feature set 4: 'CBSE_02' and 'all_mcqs_avg_n20'





b. Compare model performances using metrics (such as confusion matrix, precision, recall and F1 scores)

Analysis: Model performance evaluation on test data for Feature set'all_mcqs_avg_n20','CBSE_01','CBSE_02'.
From Figure below the F1-score on training data the model able to predict 65% precision,58% recall and f1-score 51% indicating model is average performer. However, the f1-score on test data is 45% resulting in model as poor classifier on given data.

```
train data
Algorithm confusion matrix
 [[33  0  3  0]
 [17  4  4 10]
 [ 1  1 10 23]
 [ 0  0  0 41]]
algorithm Precision,recall,f-score train data
 (0.6473370429252783, 0.5791666666666666, 0.5140698881328567, None)
              precision    recall  f1-score   support

           0       0.65      0.92      0.76        36
           1       0.80      0.11      0.20        35
           2       0.59      0.29      0.38        35
           3       0.55      1.00      0.71        41

   micro avg       0.60      0.60      0.60       147
   macro avg       0.65      0.58      0.51       147
weighted avg       0.64      0.60      0.52       147

test data
Algorithm confusion matrix
 [[10  0  0  0]
 [ 8  1  1  1]
 [ 1  0  3  7]
 [ 0  0  0  5]]
algorithm Precision,recall,f-score test data
 (0.6652327935222672, 0.5909090909090908, 0.45296934865900385, None)
              precision    recall  f1-score   support

           0       0.53      1.00      0.69        10
           1       1.00      0.09      0.17        11
           2       0.75      0.27      0.40        11
           3       0.38      1.00      0.56         5

   micro avg       0.51      0.51      0.51        37
   macro avg       0.67      0.59      0.45        37
weighted avg       0.71      0.51      0.43        37
```

Figure a: Best Model with best features question 3a

```
train data
Algorithm confusion matrix
 [[33  0  3  0]
 [20  4  3  8]
 [ 0  3 12 20]
 [ 7  0  0 34]]
algorithm Precision,recall,f-score train data
 (0.5841205837173579, 0.5507694541231126, 0.4977501384781793, None)
              precision    recall  f1-score   support

           0       0.55      0.92      0.69        36
           1       0.57      0.11      0.19        35
           2       0.67      0.34      0.45        35
           3       0.55      0.83      0.66        41

   micro avg       0.56      0.56      0.56       147
   macro avg       0.58      0.55      0.50       147
weighted avg       0.58      0.56      0.51       147

test data
Algorithm confusion matrix
 [[9 0 1 0]
 [7 1 0 3]
 [3 0 3 5]
 [2 0 0 3]]
algorithm Precision,recall,f-score test data
 (0.6128246753246753, 0.46590909090909094, 0.3805779569892473, None)
              precision    recall  f1-score   support

           0       0.43      0.90      0.58        10
           1       1.00      0.09      0.17        11
           2       0.75      0.27      0.40        11
           3       0.27      0.60      0.37         5

   micro avg       0.43      0.43      0.43        37
   macro avg       0.61      0.47      0.38        37
weighted avg       0.67      0.43      0.38        37
```

Figure metrics of best Model for features of

We can make comparison of best model with features in question 3a(“all_mcqs_avg_n20", "all_NBME_avg_n4”) with initial parameter values set to model best model with feature set 1.For given initial model parameters it is evident from Figure b above the feature set 1 model has best performance. The precision recall and F1-score of models from question 3a is lower than the model from question 5a.