Project Report - Group 9

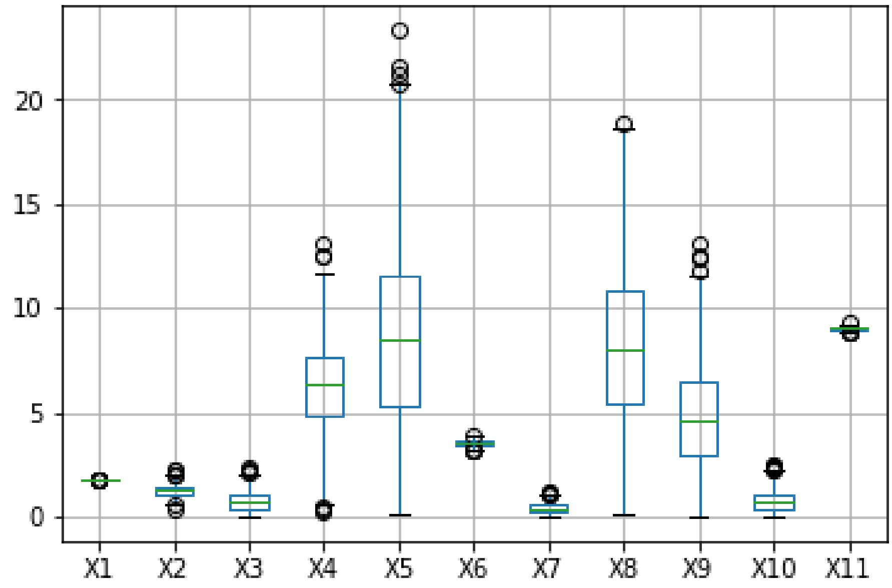
Yashswi Jain
Ram Milan Verma
Koshlendra Dubey

Before beginning with the data pre-processing, we have generated g-9-w.csv file comprising of knocked off values to simulate missing data.

STEP 1 : Data Pre-processing:

1. Identifying the outliers and clipping the values

Before imputation, since imputing the values with the mean, in the presence of the outliers would have an adverse effect, we have decided to clip off the outliers. It is clear from the Box Plot below that the given dataset has outliers which needs to treated appropriately.



2. Data Imputation

The missing values are imputed with the mean value i.e. each missing data point of a specific feature is replaced with the global feature mean.

3. Feature Scaling

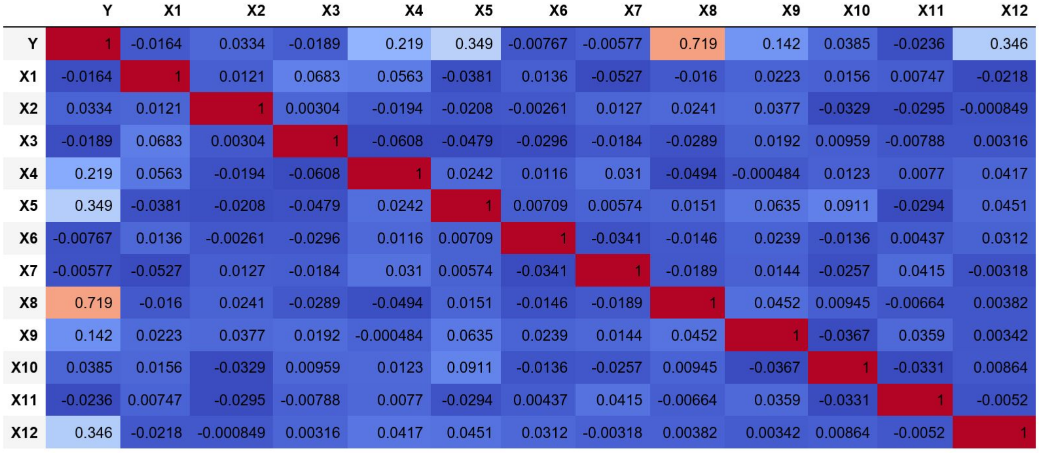
Since many of the models are sensitive to feature scaling, we have scaled the data points so that they have ‘0’ mean and unit standard variance.

STEP 2 : Exploratory Data Analysis:

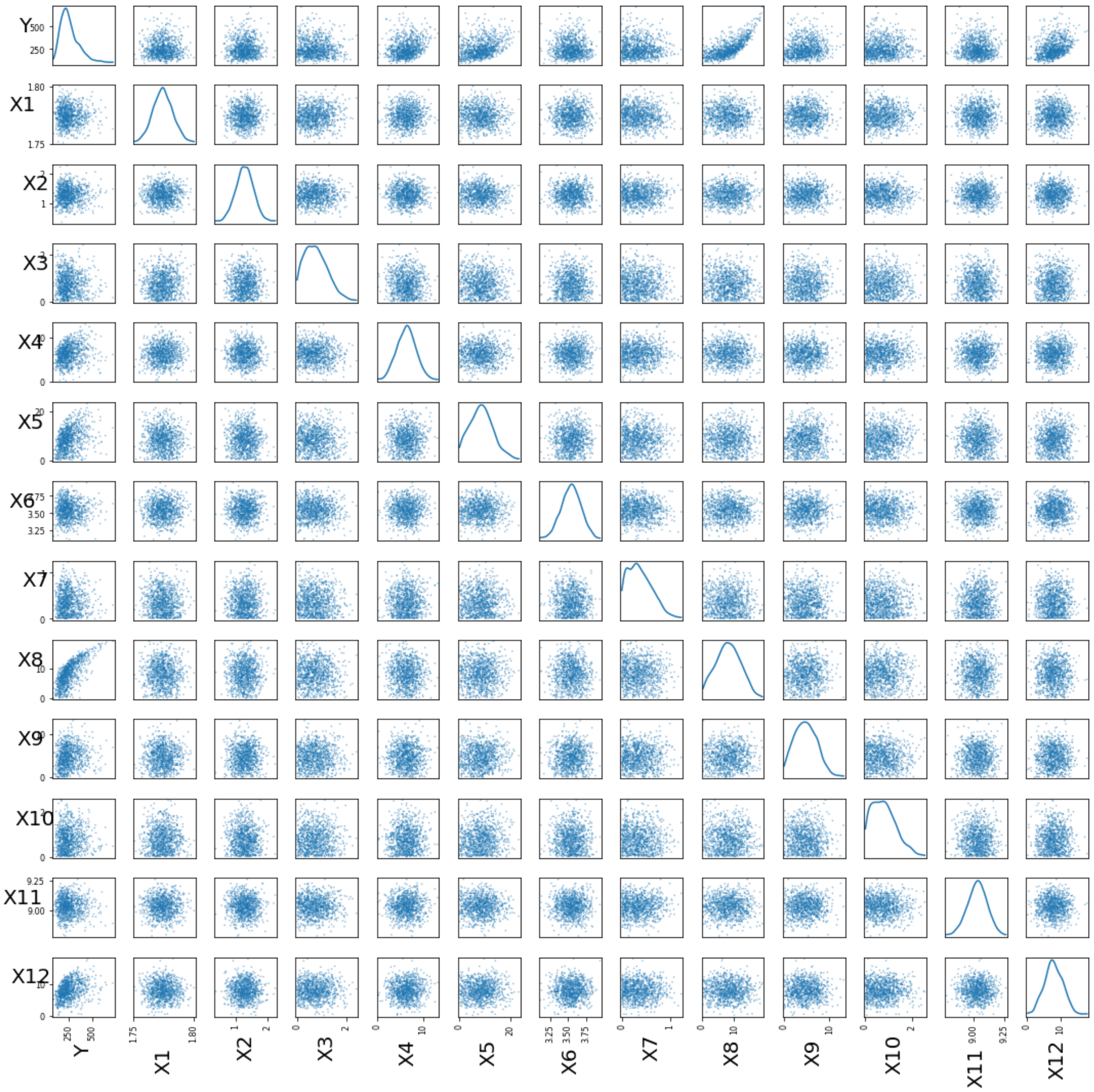
1. Visualising each of the 12 predictor variable against the response variable:

   This step is essential to decide if we need to do some feature engineering if we find some obvious functional relationship(other than linear relationship) between the predictors and the response variable.

   Correlation Matrix:



	Y	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12
Y	1	-0.0164	0.0334	-0.0189	0.219	0.349	-0.00767	-0.00577	0.719	0.142	0.0385	-0.0236	0.346
X1	-0.0164	1	0.0121	0.0683	0.0563	-0.0381	0.0136	-0.0527	-0.016	0.0223	0.0156	0.00747	-0.0218
X2	0.0334	0.0121	1	0.00304	-0.0194	-0.0208	-0.00261	0.0127	0.0241	0.0377	-0.0329	-0.0295	-0.000849
X3	-0.0189	0.0683	0.00304	1	-0.0608	-0.0479	-0.0296	-0.0184	-0.0289	0.0192	0.00959	-0.00788	0.00316
X4	0.219	0.0563	-0.0194	-0.0608	1	0.0242	0.0116	0.031	-0.0494	-0.000484	0.0123	0.0077	0.0417
X5	0.349	-0.0381	-0.0208	-0.0479	0.0242	1	0.00709	0.00574	0.0151	0.0635	0.0911	-0.0294	0.0451
X6	-0.00767	0.0136	-0.00261	-0.0296	0.0116	0.00709	1	-0.0341	-0.0146	0.0239	-0.0136	0.00437	0.0312
X7	-0.00577	-0.0527	0.0127	-0.0184	0.031	0.00574	-0.0341	1	-0.0189	0.0144	-0.0257	0.0415	-0.00318
X8	0.719	-0.016	0.0241	-0.0289	-0.0494	0.0151	-0.0146	-0.0189	1	0.0452	0.00945	-0.00664	0.00382
X9	0.142	0.0223	0.0377	0.0192	-0.000484	0.0635	0.0239	0.0144	0.0452	1	-0.0367	0.0359	0.00342
X10	0.0385	0.0156	-0.0329	0.00959	0.0123	0.0911	-0.0136	-0.0257	0.00945	-0.0367	1	-0.0331	0.00864
X11	-0.0236	0.00747	-0.0295	-0.00788	0.0077	-0.0294	0.00437	0.0415	-0.00664	0.0359	-0.0331	1	-0.0052
X12	0.346	-0.0218	-0.000849	0.00316	0.0417	0.0451	0.0312	-0.00318	0.00382	0.00342	0.00864	-0.0052	1



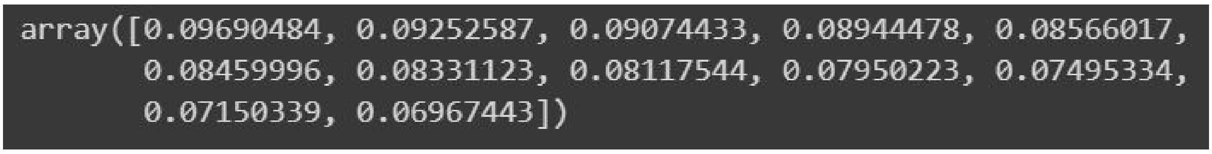
Observation : Only one predictor variable ‘x8’ had an approximate linear relationship with the response variable. Rest all of them didn’t have an apparently identifiable relationship with the response variable.

Q1: Ans: This problem involves regression. Hence, we have tried fitting different regression models and picked the one with the least RMSE as a candidate for out best model proposal.

We have analysed the following models:

1. **PCA followed by Multiple Linear Regression:**

PCA is one of the methods that can be used for model selection since it can identify variables that contribute to the maximum variance.



```
array([0.09690484, 0.09252587, 0.09074433, 0.08944478, 0.08566017,
       0.08459996, 0.08331123, 0.08117544, 0.07950223, 0.07495334,
       0.07150339, 0.06967443])
```

The figure above shows the variance captured by the transformed components i.e. PC1,....,PC12. Notice the above picture was generated to enable the transformed components to capture of 99% of the variance and it had 12 components. This implies non of the 12 predictors can be neglected during feature selection.

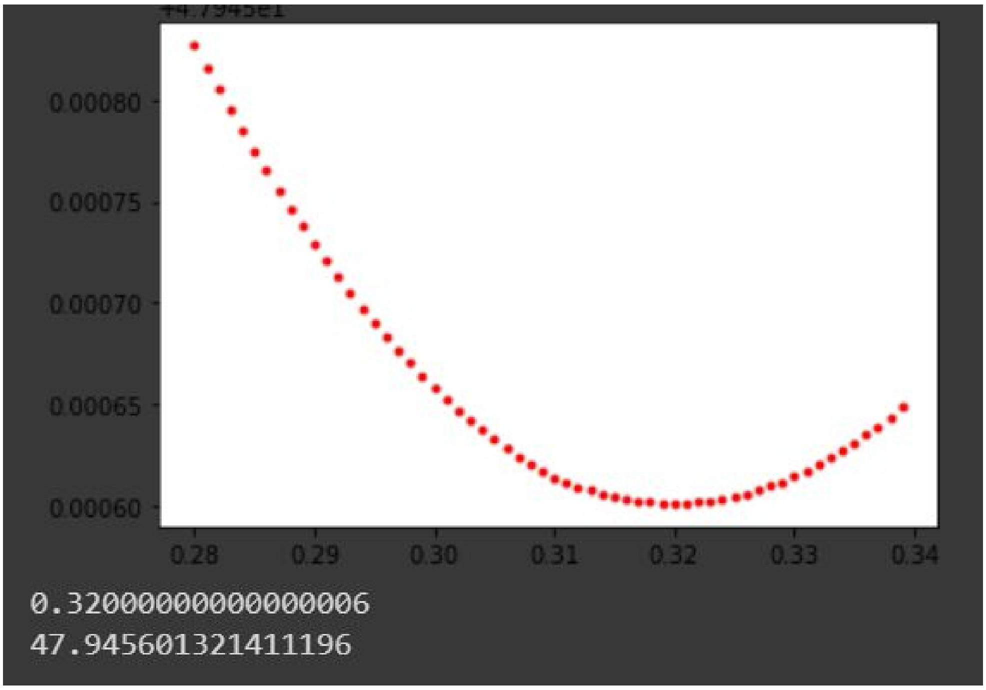
Observation: After PCA, it was observed that almost all the 12 predictor variables in the transformed space had almost an equal share in the total variance, hence, efforts for feature selection were futile if we want to capture more than 95% of the variance, since all the 12 were selected anyways.

<u>Observed RMSE:</u> 47.963

2. Ridge Regression:

Ridge regression is helpful when we want to reduce the effect of the dominant predictor variable. But, it was apparent from the PCA that the dataset didn’t have a clear dominant variable. Hence, it was expected that ridge won’t be of much help.

We have performed 10-fold CV to tune the hyperparameters and have taken the minimum possible average RMSE corresponding to the tuned hyper-parameter.



```
0.32000000000000006
47.945601321411196
```

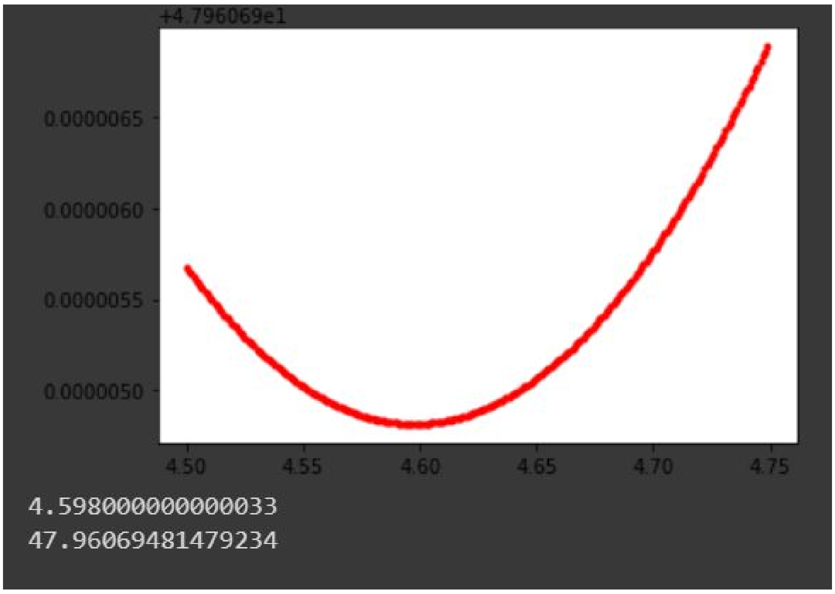
The above is a plot of alpha vs the average RMSE value.

Tuned Hyperparameter Value : 0.32

Observed RMSE: 47.9457

3. LASSO:

Lasso had the added benefit of feature selection.



```
4.598000000000033
47.96069481479234
```

The above is a plot of alpha vs the average RMSE value.

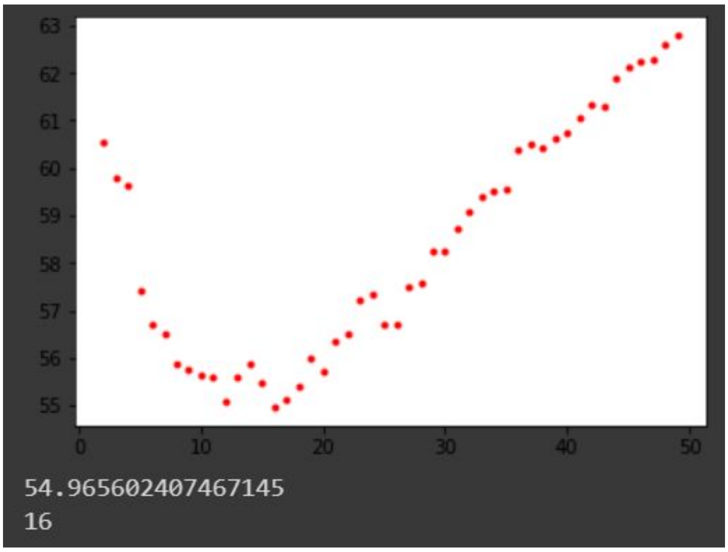
Tuned Hyperparameter Value : 4.598

Observed RMSE: 47.9607

4. CART:

Although a decision tree regressor has many hyper-parameters that can be tuned, we have only tweaked one i.e. minimum samples per leaf node. Also, depth is decided to grow without any constraints unless the minimum number of samples per leaf constraint is not violated.

Below plot shows the relationship between minimum samples per leaf node vs the average rmse value obtained.



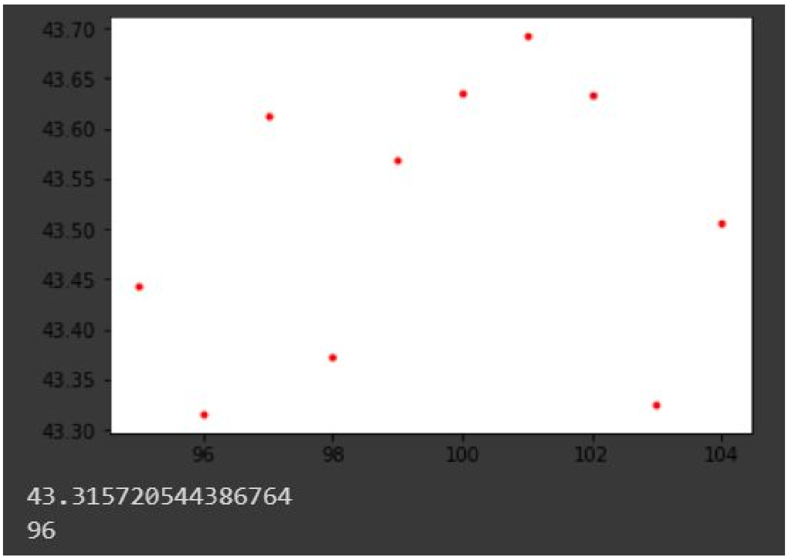
```
54.965602407467145
16
```

Tuned Hyperparameter : 16

Observed RMSE = 54.9657

5. BAGGING:

Since, bagging builds upon a simple CART we have taken the CART with min. 16 observations per leaf which was the optimum for the CART regressor. But, surprisingly keeping the base to be chosen by the optimiser yields better results. Further, we have tried tweaking the number of estimators for obtaining the best possible RMSE.



```
43.315720544386764
96
```

The above plot shows the variance of the avg. RMSE with number of estimators in BAGGING.

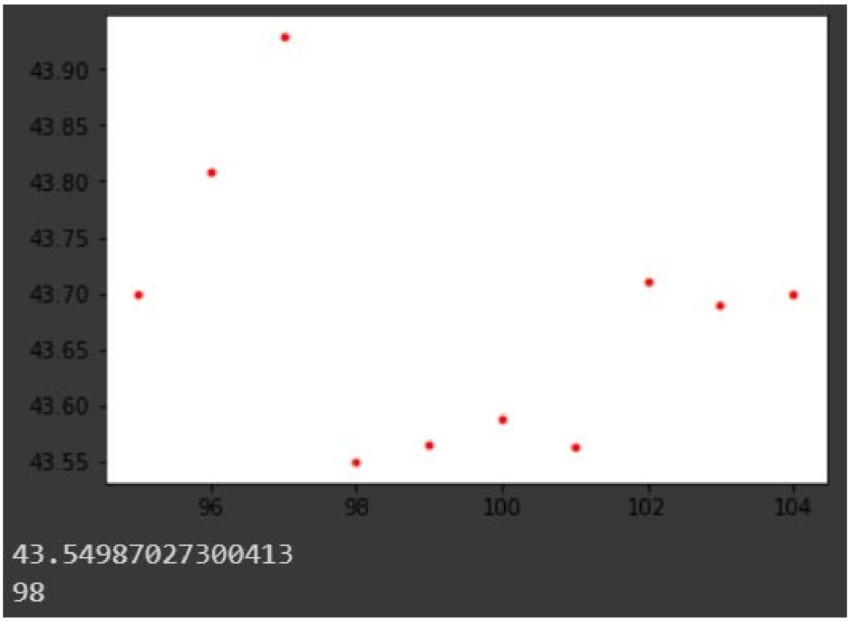
Tuned Hyperparameter : 96 (Number of estimators)

Observed RMSE: 43.3157

6. Random Forest:

Random Forest is particularly useful when the number of features is very large since it randomly samples the observations as well as the features before each split.

Again, we are tweaking just the minimum number of observations per leaf node and taking best possible RMSE value.



```
43.54987027300413
98
```

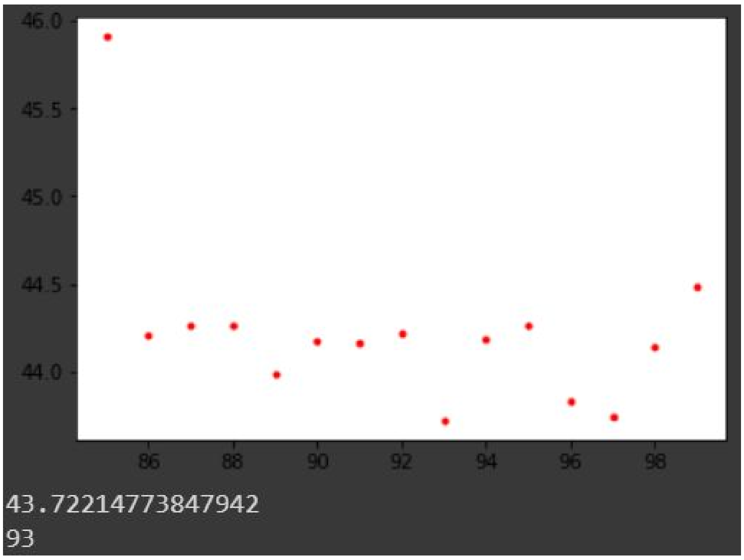
The above plot shows the variation of average RMSE vs the tweaked hyperparameter(in the x-axis).

Tuned Hyperparameter : 98

Observed RMSE : 43.5498

7.AdaBoost:

Average RMSE vs Number of estimators PLOT



```
43.72214773847942
93
```

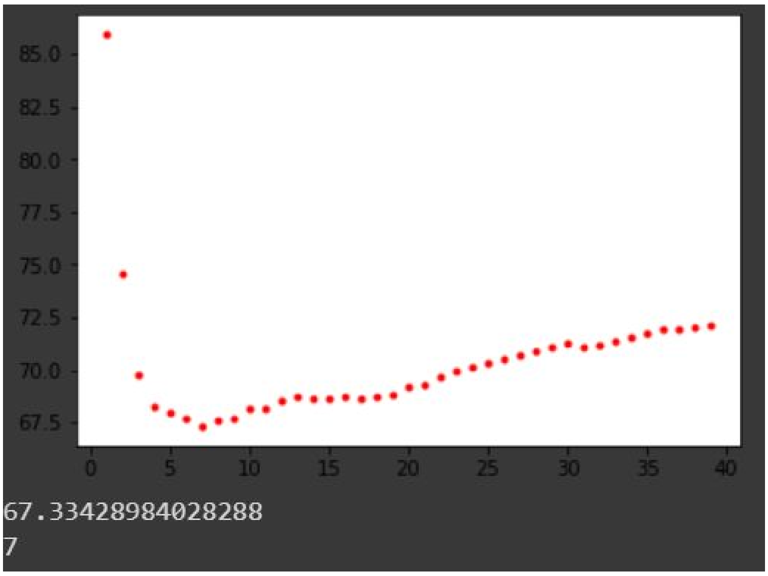
Here, we are tweaking the number of estimators for the AdaBoost regressor.

Tuned Hyperparameter = 93

Observed RMSE = 43.72214

8. KNN Regression:

Below is the plot of K vs Average RMSE



```
67.33428984028288
7
```

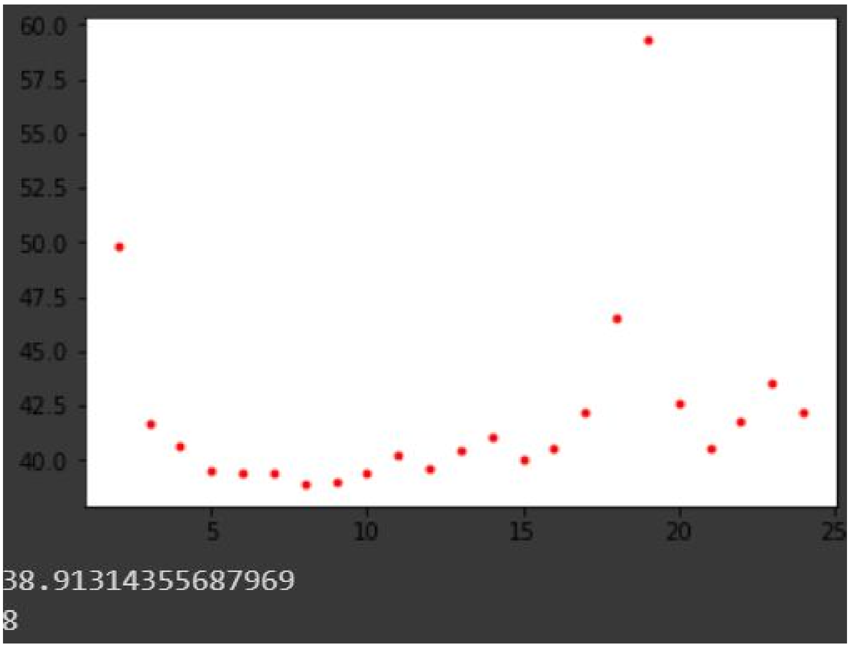
Tuned Parameter = number of neighbours = 7

Minimum RMSE = 67.33

9. MLP Regressor:

Here, we have only used ANN i.e. a network with a single hidden layer since deep neural network requires large amount of data and we don’t have a large dataset.

The hyperparameter that we are tuning is number of nodes in the hidden layer.



```
38.91314355687969
8
```

Above is a plot of Number of nodes in the hidden layer vs the obtained RMSE

Tuned Hyperparameter = 8

Observed RMSE = 38.9131

Caveat: This might not be the RMSE that one might observe by running the same code since the k-fold CV is a random process and the convergence of the SGD to the same minima isn't guaranteed.

10. ElasticNet:

Elastic Net combines both L1 and L2 penalty terms and requires two hyperparameters to be tweaked.

Tuned Hyperparameters: alpha = 0.03, l1_ratio = 1.2

Observed RMSE = 53.295

Synopsis of the analysis:

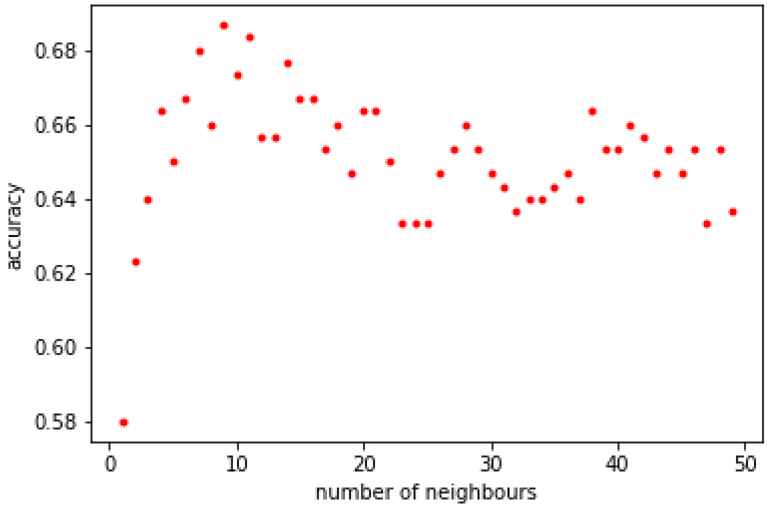
Model	Obtained RMSE
PCA- MLR	47.963
Ridge	47.9457
LASSO	47.9607
CART	54.9657
BAGGING	43.3157
Random Forest	43.5498
AdaBoost	43.7221
MLP Regresor	38.9131
ElasticNet	53.295
KNN Regression	67.33

Conclusion: *MLP Regressor* comes out to be the model with the minimum average RMSE. Hence, we propose it as the best model.

Q2: ANS: This problem involves classification. Hence, we have tried fitting different classification models and picked the one with the highest mean accuracy as a candidate for out best model proposal.

We have tried the following models:

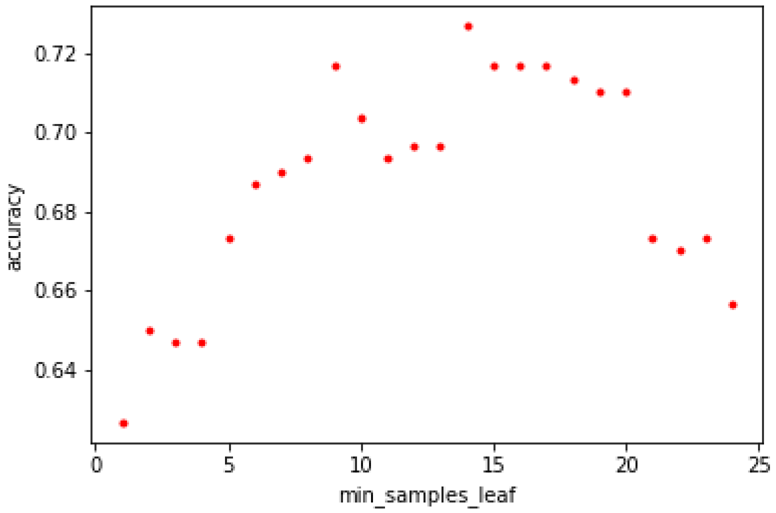
1. **KNN Classification**



Tuned parameter = number of neighbours= 9

Maximum Accuracy=68.67%

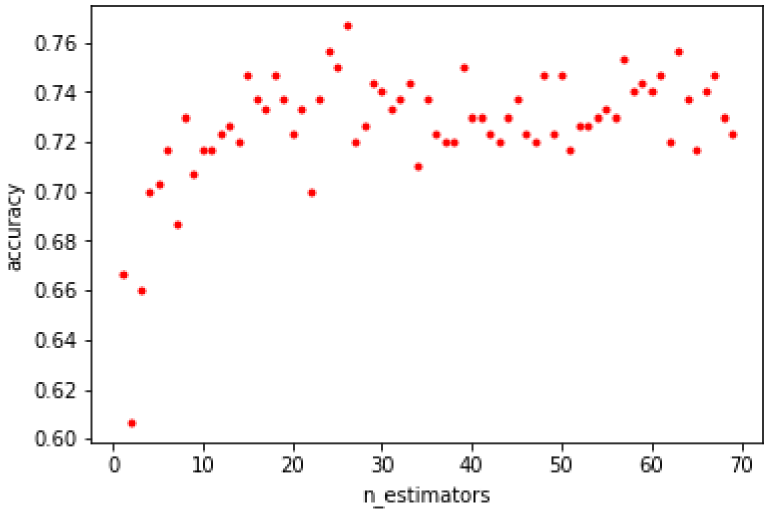
2. CART



Tuned Parameter = minimum number of samples per leaf = 14

Maximum accuracy = 72.666%

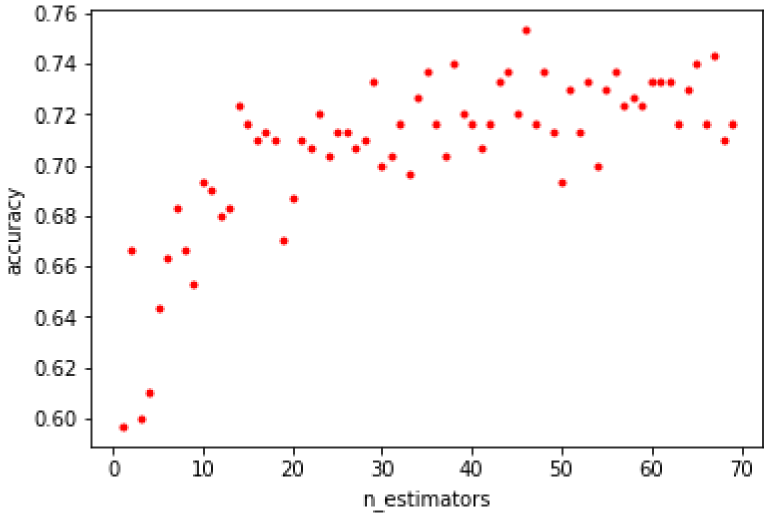
3. BAGGING



Tuned Parameter = Number of estimators = 26

Maximum accuracy = 76.67%

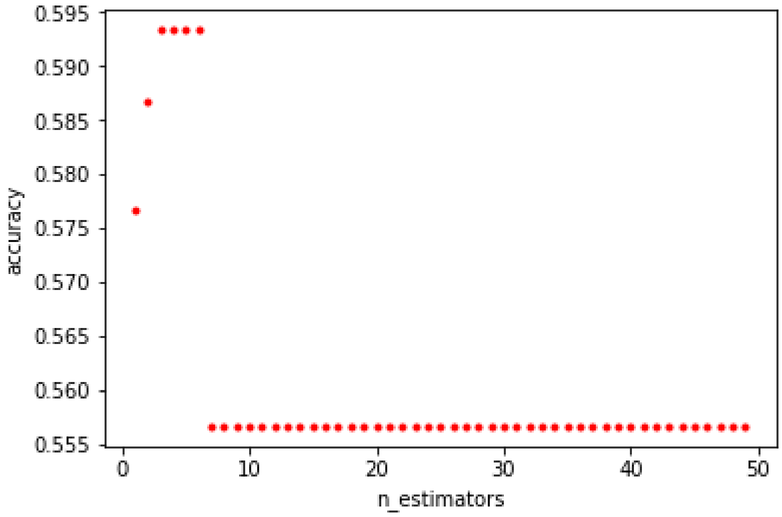
4. **Random Forest**



Tuned Parameter = Number of estimators = 46

Maximum accuracy = 75.44%

5. **AdaBoost**



Tuned Parameter = Number of estimators = 4

Maximum accuracy = 59.33%

6. ANN

Tuned parameter = number of neuron in hidden layer = 9

Maximum accuracy = 75.6 %

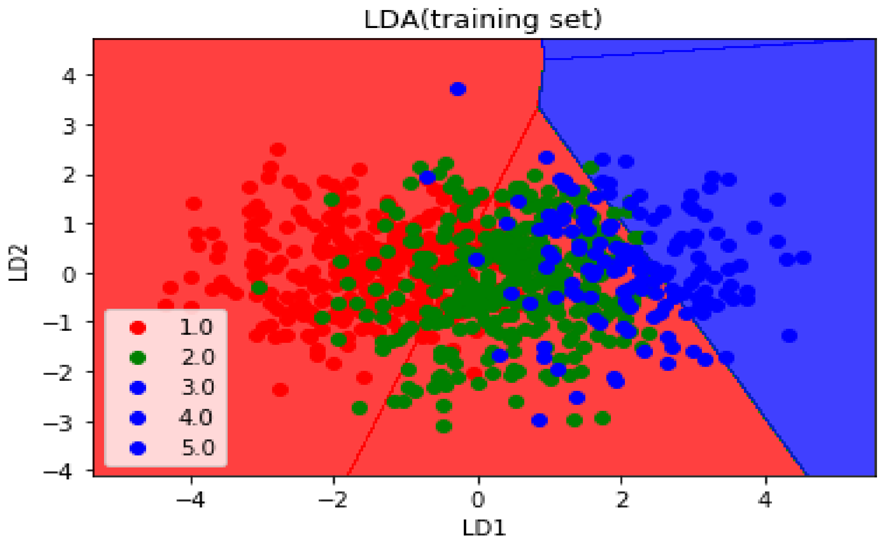
P.S.- on running the code ,the accuracy is fluctuating between 69% to 76% , because of the random splitting of the data and the convergence of the algorithm to the same minima isn't guaranteed as well

(ANN having 9 neuron in the hidden layer gives the best result)

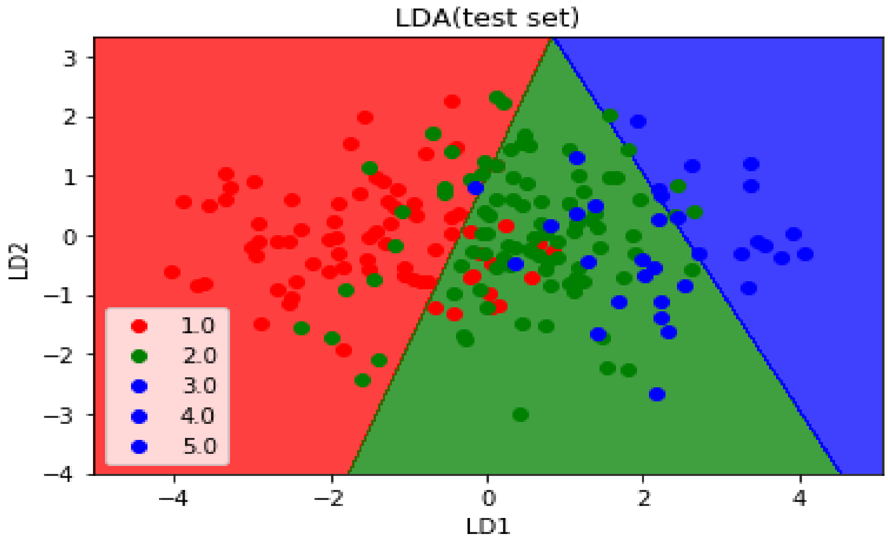
7. LDA

Maximum accuracy =69%

Decision Boundary for LDA :-
(Training set)



(Testing Set)



8. **QDA**

Maximum accuracy = 68 %

Model	Obtained Accuracy
KNN Classification	68.67 %
ANN	77.5 %
LDA	69.0 %
CART	72.67%
BAGGING	76.67%
AdaBoost	59.33%
QDA	68.0 %
Random Forest	75.44%

Conclusion: Since the accuracy of ANN, Bagging and Random Forest are very close, it will be unfair to say that ANN is clearly the best model. Hence, all three of them can be proposed. But, if forced to pick one we will go with ANN.