Project 1

Salil Khandelwal
MSc Computer Science
301376050

Q1 96.8

Step 1: Created univariate scatterplots for the numerical features and histograms for categorical features(X1, X2, X15). I did this for the training set and testing set. I also looked at the summary statistics for numerical features.

Good idea.

(a) Rationale: This helps us understand the data better. We can see how the data is distributed in both the training set and the testing set. If there are outliers in the data, we can detect them as well.

(b) Results: The distribution for the features are similar in both the training set and the testing set. There are no visible outliers in the data. Examples of the scatterplots and histograms are below in the plot section.

Step 2: Calculated pairwise correlation between numerical features and created a scatterplot matrix for them.

(a) Rationale: This can help us see if any numerical feature has a strong relationship with the response variable Y. This can also help us understand if the features are correlated amongst each other.

(b) Results: Looking at the scatterplots, I don't see any strong correlation between the numerical predictors and the response. Also, I don't see any strong correlation between any 2 pairs of numerical variables except X13 and X14.

Step 3: Handling categorical variables by creating dummy variables using model.matrix

(a) Rationale: The models we will be using in the steps ahead should be made aware of the presence of categorical variables. Certain methods can deal with them implicitly, for example, Random Forests where we simply list them as factor but other requires the input in strictly numerical fashion for instance, neural networks.

(b) Results: We have unpacked the categorical variables and created dummy variables for their levels.

Step 4: Created a baseline model using OLS. Calculated its prediction error using the data splitting process. The given training data was divided in two parts randomly, where one part contains 3/4th of the data which will be used for fitting the model and remaining will be used to calculate the prediction error.

Once or multiple times?

(a) Rationale: Often modeling data is an iterative procedure, where we start model and try to build upon that. OLS is the simplest model we can think of. For a

more robust measure of the prediction error we repeated the process for 20 times to counter the issue of randomness that comes with the data splitting.

(b) Results: We obtained a baseline solution of OLS where the MSPE is 1.56. This is the standard OLS with all the variables in the model.

Step 5: Finding out the important variables in the dataset using classical algorithm (Stepwise using BIC) and modern approach (LASSO with lambda min). I reapplied these algorithms 20 times on different random splits. I noted which variables were occurring more frequently than others and this gives us some sense of importance of variables in the dataset.

(a) Rationale: When we are given a dataset some features may be more important than others and identifying them can give a significant boost in the performance of our model. Removing unnecessary variables from the model can prevent us from increasing the variance of our prediction. I am repeating the process for 20 times on different splits to give a more confident answer to the question of variable importance. Variables that are occurring regularly across 20 splits can be thought of as important.

(b) Results: The variables that were occurring regularly were X4, X12, X2 and X15. Note that the last two are categorical variables, there were regular occurrences of the dummy variables created from these two variables in the variable selection process. This gives us sense of important variables in our model. X6 was also showing up occasionally but was way less as compared to the 4 variables listed above. These variables also appeared to be significant while fitting the OLS. So we have three sets of variables

(i) Full Set
(ii) {X2, X4, X12, X15}
(iii) {X2, X4, X6, X12, X15}

For X2 and X15, we will be including all dummy variables created using them

To feel a bit more reassured about our selection process, I also calculated the prediction error using OLS on these 3 sets by repeating the 20 splits process where the prediction error was calculated on the left out data in each split. The mean of those prediction errors are as follows:

Set	mean MSPE
Full	1.56
{X2, X4, X12, X15}	1.47
{X2, X4, X6, X12, X15}	1.49

FYI, you are using the full data to select variables, and then using the selected variables in the training of machines -- this can lead to target leakage because data used in test sets have already contributed to the selection of variables. This may possibly cause some overfitting. If you wanted to d o this more safely, you would repeat the variable selection process WITHIN each split. You could still record the results in each split and gain a consensus for later reporting.

ariable is resulting in better mean prediction errors.
it the models on these three sets of variables.

ets mentioned above. I experimented with
, Projection Pursuit R
Networks, Random F
s most time consumi
. To tune the method
gbm) I randomly split
or testing and this pr
tuned using grid search over the various hyperparameters
thing was repeated for each set of variables i.e. Full Set, {X2, X4, X12, X15} and
X6, X12, X15}.

A little unclear how tuning was done: ranges of parameter values, process of tuning within method vs. how tuned versions of different methods were compared.

I like that you used both full and reduced sets.

(a) Rationale: Thus far we have tried to model relationships in our data using OL
Stepwise regression and LASSO. It may be the case that there are complex relati
present between our predictor and the response variables which can't be captured by the methods used previously. Using the methods listed above we can model such complex interactions and hope to get a better prediction.

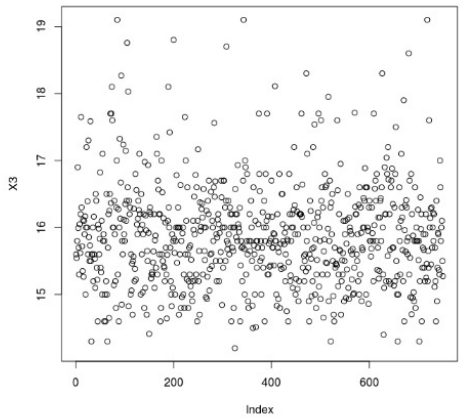
(b) Results: It was observed that the XGBoost model that was built using {X2, X4, X12, X15} was giving the best predictive performance amongst all the methods tried across all the sets of variables. This leads us to believe that the remaining variables outside {X2, X4, X12, X15} are not contributing much to the prediction of the response as the model which was built without them was performing cor
giving the best perfromance used max_de
200 on the variables {X2, X4, X12, X15}. T
accompanying this report as well.

1. Initial Checks: Good study of data. 10/10
2. Modeling: Process was fairly thorough, but lacked information on some parts. Did initial "variable selection" and made good use of this. Some concerns about target leakage. 55/60
3. Model Evaluation: Repeated splitting and comparisons of MSPEs are good. Some uncertainty about use of different splits in different methods. Probably ought to make sure that predicted values look sensible. Might try looking at some residuals in final model to see how it is doing. 27/30
Report total: 92/100 Model Performance: 98.9/100 (#1 in class!!!) Variable bonus: +2 Overall Score: 96.8/100

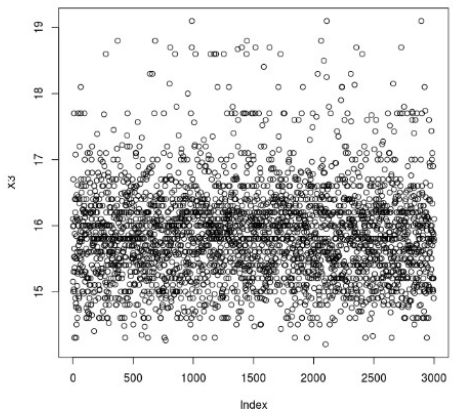
Important Variables:

X2, X4, X12, X15

The plots are given in pages below:



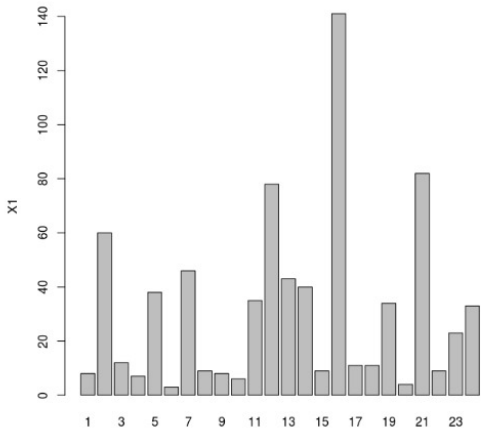
X3 in training set



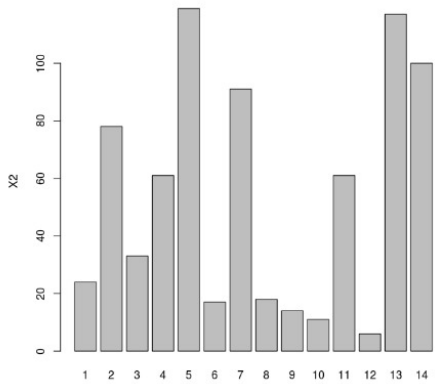
X3 in testing set

The above two plots show distribution of X3 in both the training set and the testing set. The distribution of data looks similar in both. The testing data had 3000 rows and hence looks more dense. Similar plots were drawn for other variables as well. The conclusion from all of them was that the data is distributed similarly across both the datasets.

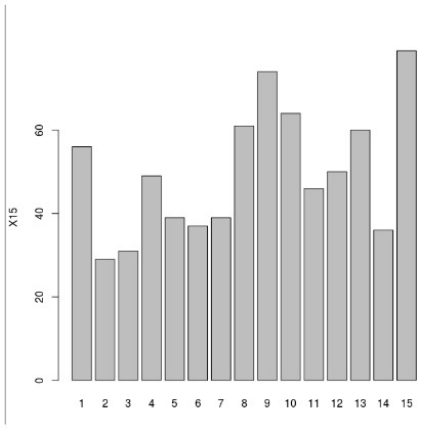
The histograms for three categorical variables look as follows in the training set:



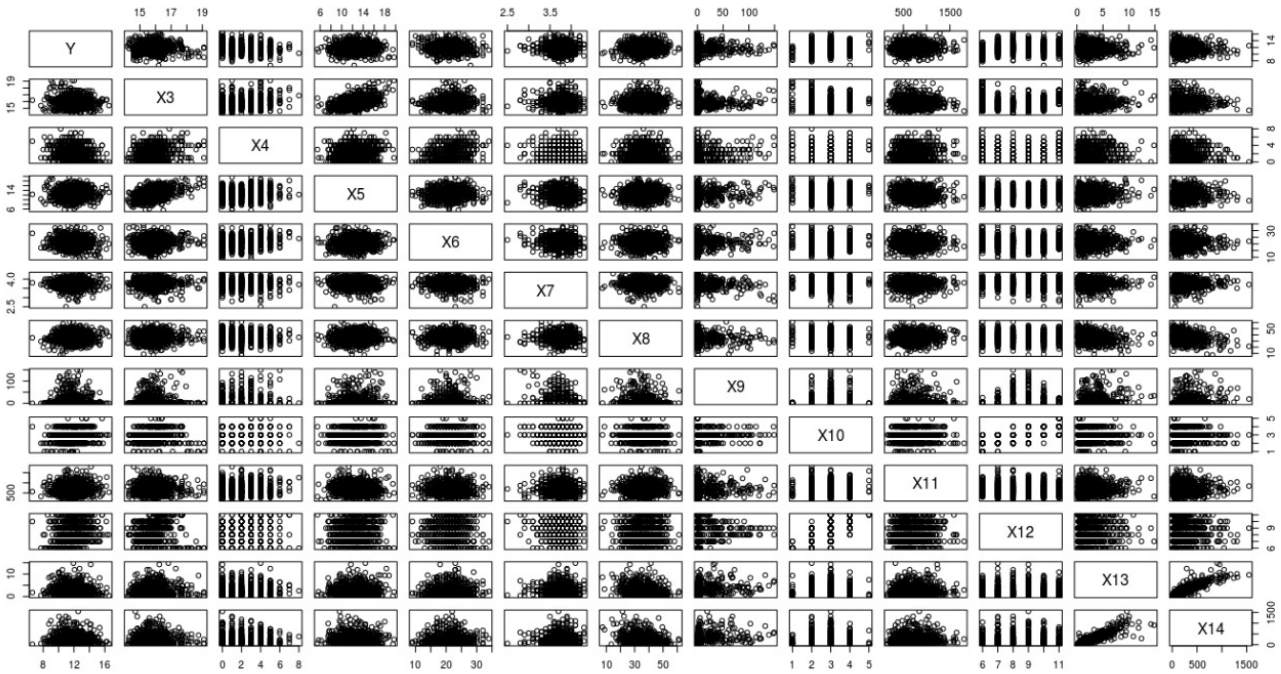
X1 has 24 levels



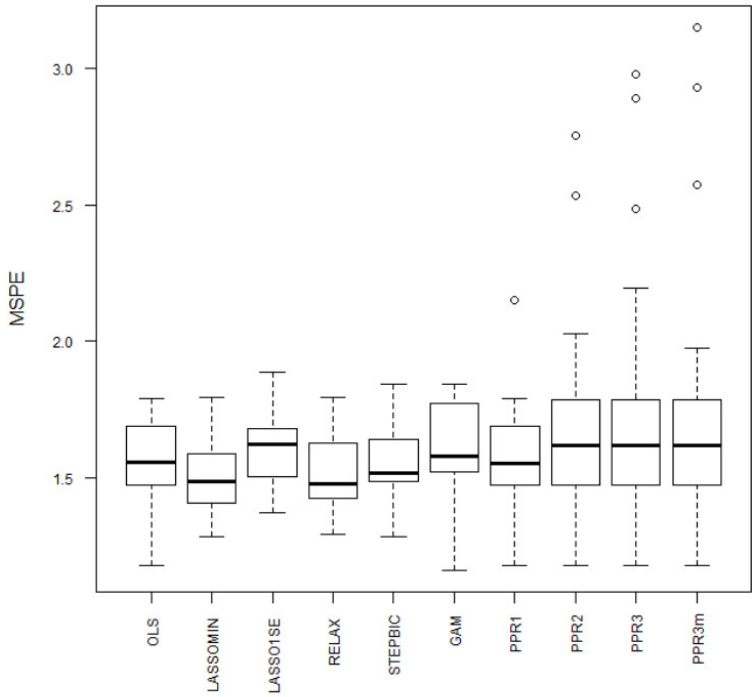
X2 has 14 levels



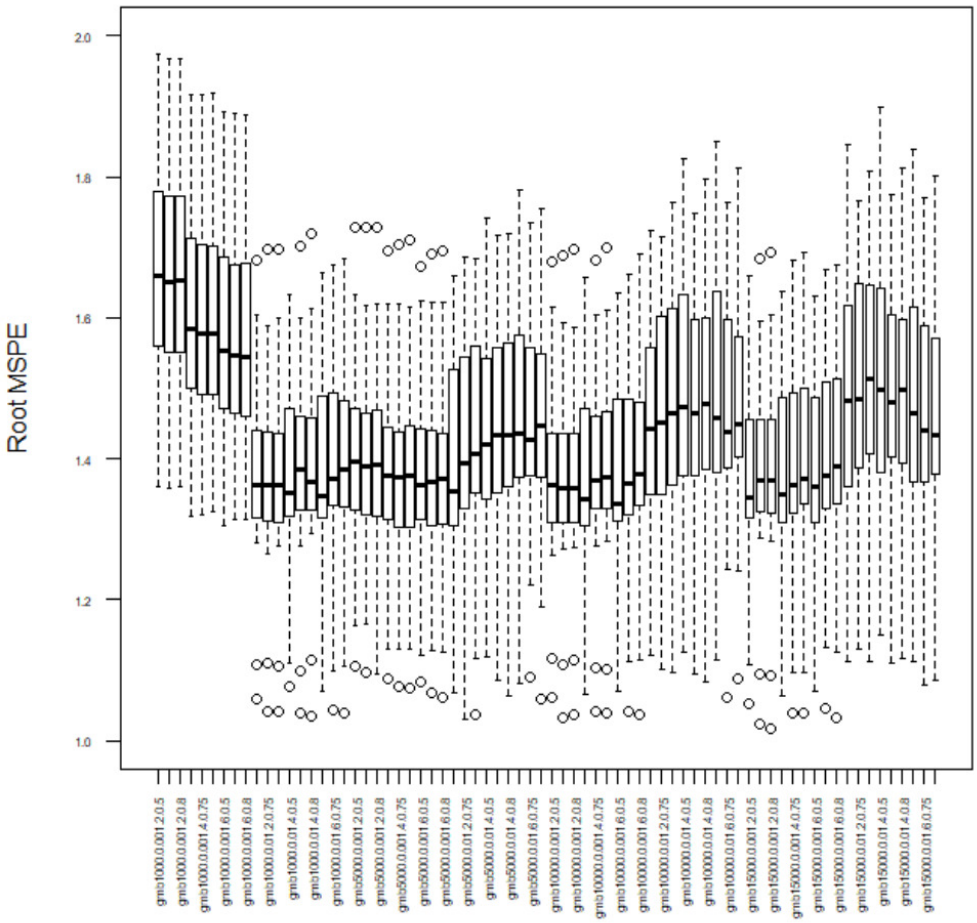
X15 has 15 levels



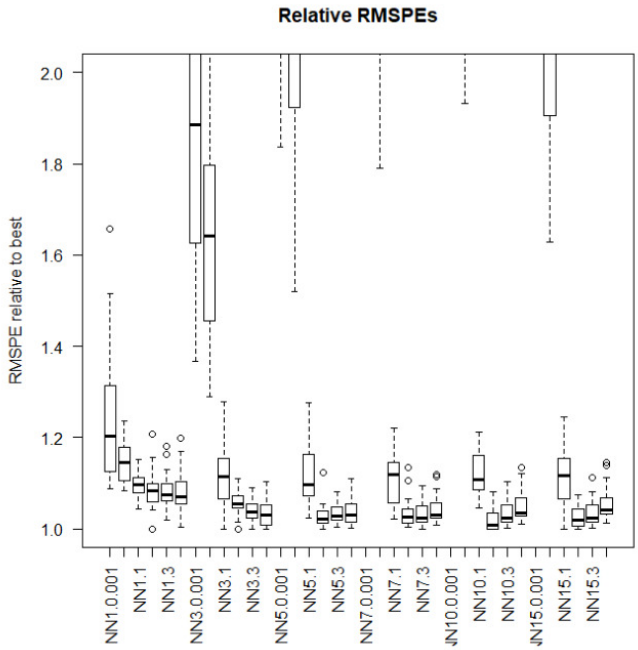
Scatterplot matrix between every pair of numerical variables



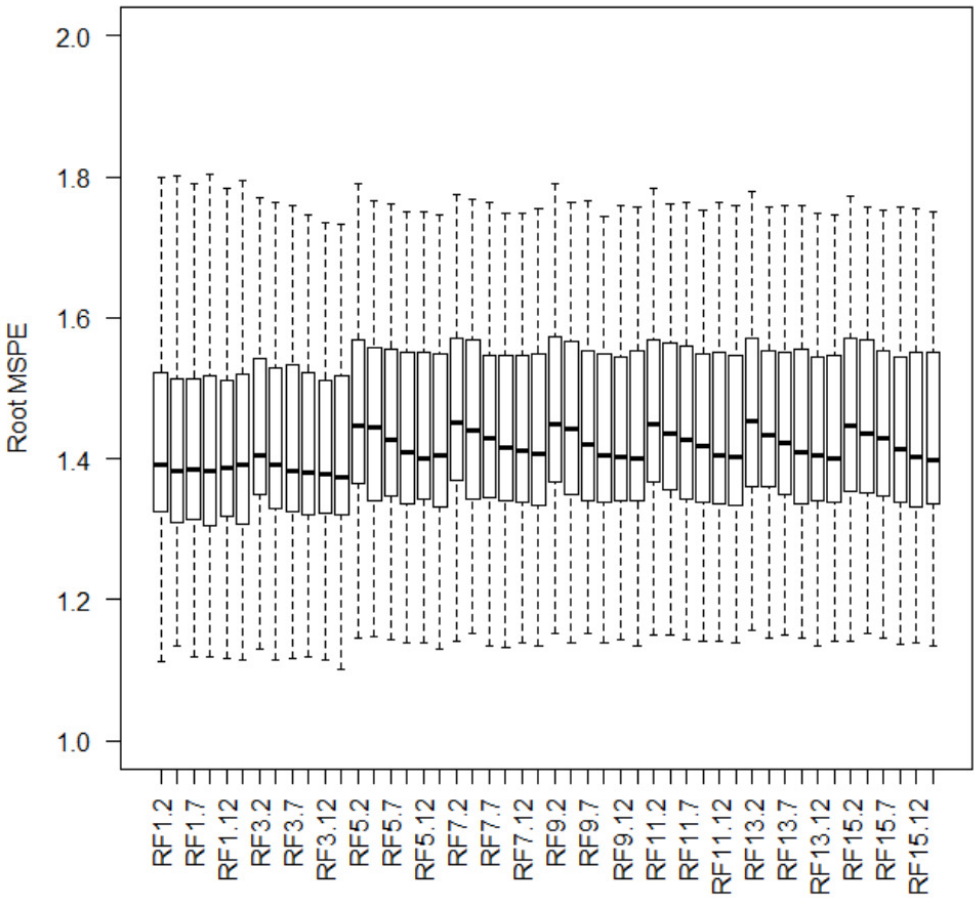
Comparison of performance amongst various methods on the full set, these boxplots were created by repeating the 20 splits for calculating the prediciton error. This is for full set but similar plots were made for other 2 sets as well.



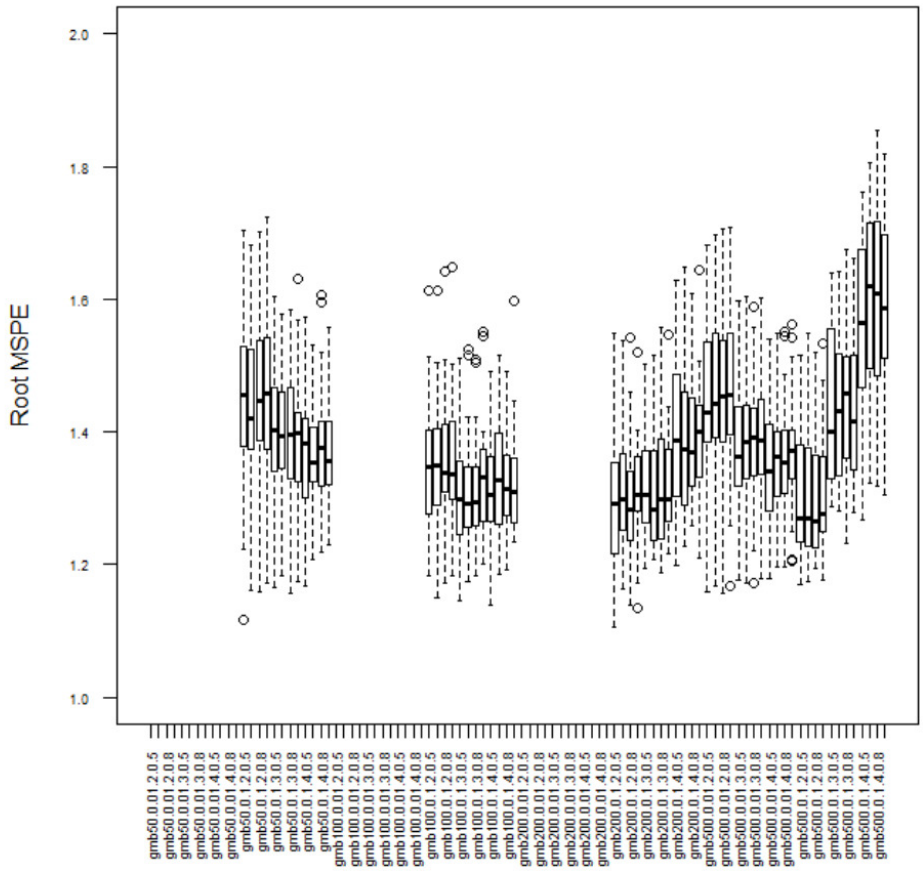
Plot used to determine the best set of tuning parameter for boosting method using gbm. Depiction of the tuning process over grid of parameters. This plot was obtained for the set {X2, X4, X12, X15}



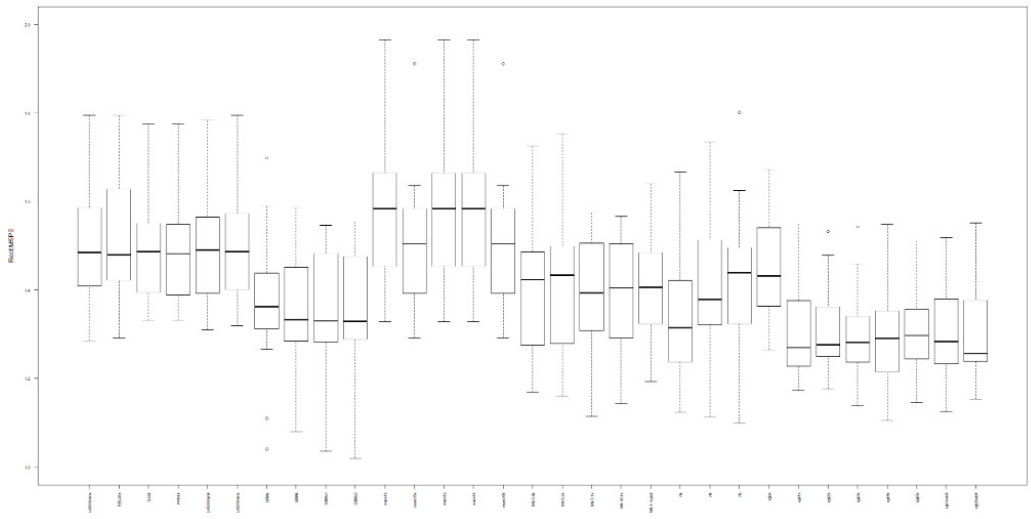
Plot used to determine the best set of tuning parameter using Neural Networks.



Plot used to determine the best set of tuning parameter for Random Forest.



Plot used to determine the best set of tuning parameter for boosting method using xgboost.



This plot contains the best tuned model across all the sets of variables used in this project. From this plot I was able to arrive at my final model.

In the plots above some y labels say Root MSPE, in reality it is MSPE and not root MSPE.