EXECUTIVE SUMMARY

For this project, we analysed a data set that was obtained for ISEN-613 project to develop important insights between different features and to create a predictive model to estimate new value of target for given values of predictors.

First of all, given the training data, we started with fitting various regression models to it.

Our aim is to find the best model which can give the best prediction accuracy. For model fitting, we tried different methods. Among those were Tree Regression (Boosting & Random Forest), Multiple Linear Regression, Subset Selection, Ridge Regression, and Lasso Regression.

We evaluated all these models based on three error checking methods.

- Training Error – By fitting the model to the entire data set.
- Splitting the datasets into two part, fitting the model on one part and finding error on another.
- Developed algorithm in R to find cross validation error for models for which cross validation method is not readily available. Plus, used cross validation techniques directly for some methods where cross validation is already available.

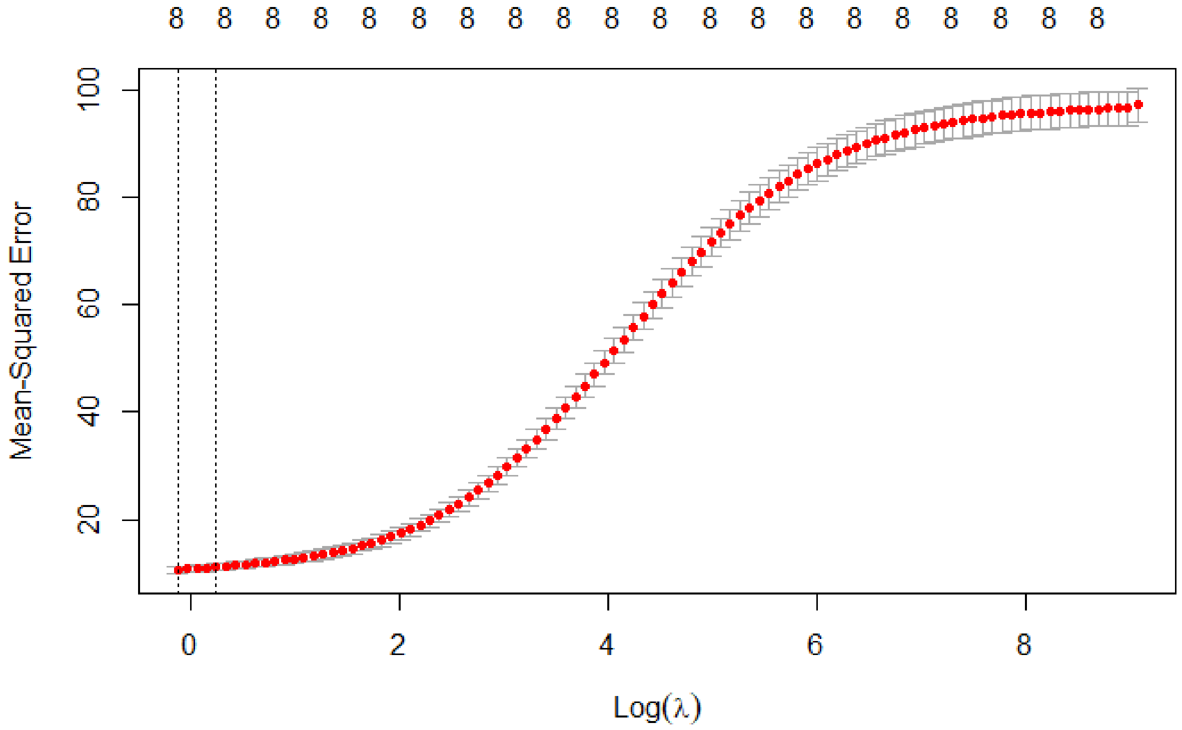
Based on the error rates discussed above we have selected the best model. Later, we verified this by checking error using test data.

After fitting and analyzing each model, we found that Boosting produces good accuracy. Whereas, Multiple Linear Regression, Lasso Regression, Ridge Regression produce less reliable result.

We worked in a team of four for this project. We had a brainstorming session followed by some raw model fitting to select which machine learning methods to be taken for this project. Then, we selected four best methods and allocated to each member of group. Thereafter, each of us tried to tune their respective allocated models. Finally, with a group discussion session we ranked all the models and submitted our top 3 ranked models and one best model.

Ridge Regression

Ridge regression is preferred over Ordinary Least Square Regression techniques, as it shrinks the coefficients of the unimportant variables. For Ridge regression, an important parameter is lambda, as it controls the number of parameters that are added to the model.
We can find the value of lambda using cross validation. We can check the plot of cross validation error vs lambda values:



From the plot we get the log value of lambda for which cross validation error is the lowest. Hence we get the value of lambda as 0.88.

We can also check the coefficients given by this Ridge regression model.

Intercept	X1	X2	X3	X4	X5	X6	X7	X8
6.17	-6.21	-0.01	0.05	-0.05	2.87	-0.03	22.3	0.23

From the above table we can see that X7, X1, and X5 are important variables as their coefficients are large as compared to other variables.

Training error (MSE):
Model is fit using the entire data set of 550 observations and we find MSE (mean squared error) corresponding to this entire set. We can use this formula : (mean(yhat - y)^2) to find training MSE.

Validation Set Approach:

We do not have a test set with us, so if we the data is fitted entirely on the data set available to us, it may lead to overfitting. Hence, we divide the data into 2 parts, one being the training set and other being the validation set (550 observations were split into 2 sets). We fit the model using the training set and then check the MSE using the validation set.

K-Fold Cross validation:

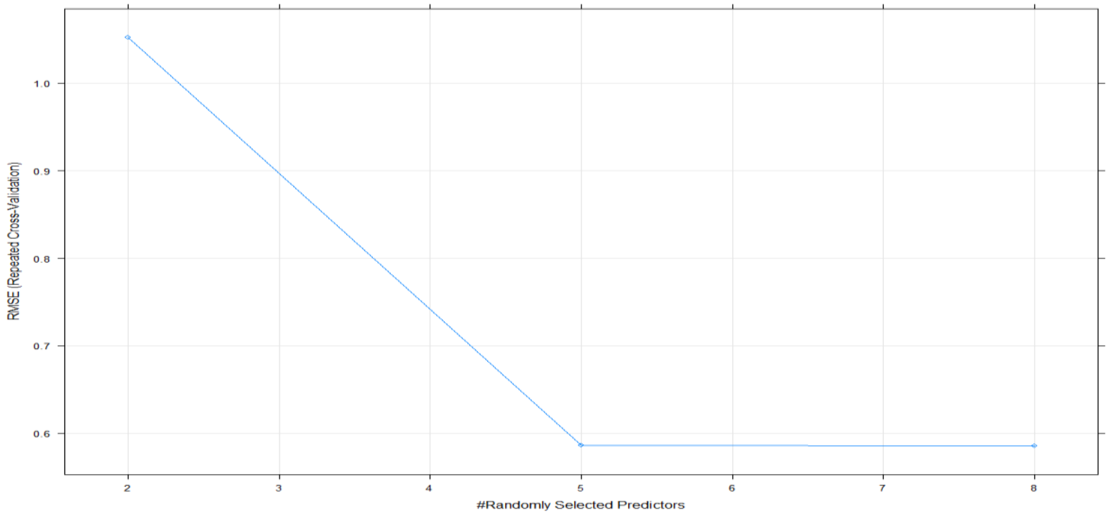
The Validation Set approach has a disadvantage as it randomly splits the data set into 2 parts, hence the estimated error that we get from each split is different. For this purpose we use cross validation, in this case we use K-Fold Cross validation with K = 10. For techniques like Random Forest, functions to perform cross validation are not readily available, hence we tried to develop a general function that can perform cross validation for various techniques. We did this by looping over K folds, and averaging out the mean squared error obtained for each set. (We are attaching the same for your perusal in the R file).

Training MSE	8.32
Validation MSE	12.2
Cross Validation error (MSE)	10.32

Random Forest

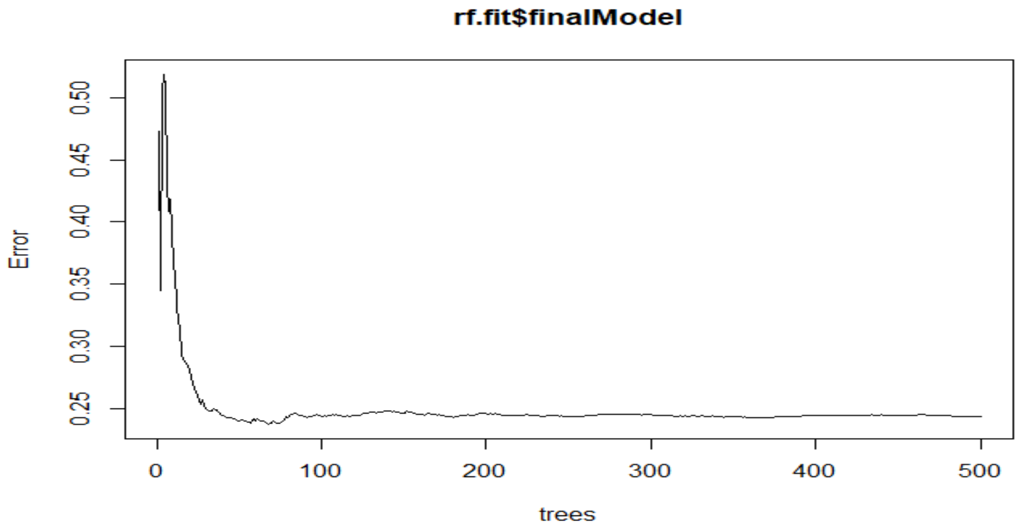
From, scatter plot of Y1 (response) with each individual predictor (eg: X1, X2, …), we can see that they do not follow linear relationships, hence linear models are not a good fit for this data. So we are trying tree-based methods. Random Forest, Bagging, Boosting is preferred over normal regression trees as it aggregates different trees for improving the prediction accuracy. Random Forest is preferred over Bagging as it decorrelates the underlying trees.

In the case, of Random Forest we randomly select a subset of predictors (m) which are suitable for the split. We can choose the value of ‘m’ for our model. Hence, we can tune the value of parameter ‘m’ by using cross validation. We have plot cross-validation error of the model for different values of m, as can be seen below:

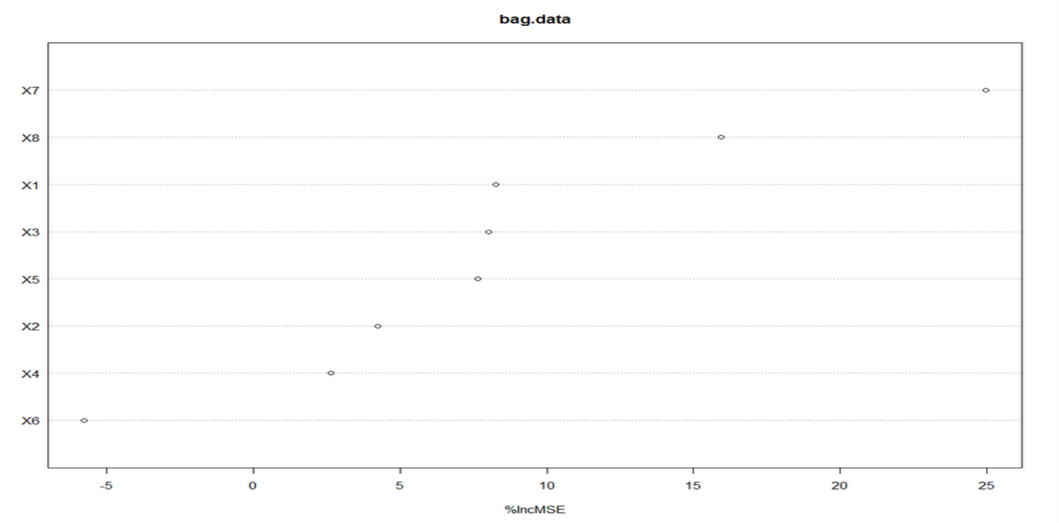


As can be seen from the plot above, we are getting the least value of cross validation error for m = 5 for our data set. (There is a slight increase in cross validation error after m = 5).

Next important parameter for this model is “ntree” (that is the number of trees that model will collectively use). Similar to the value of “m”, we will tune this parameter using cross validation. Below plot of cross validation error can be found for different values of “ntree”.



From the above plot we can see that, smallest cross validation error is corresponding to ntree = 80 for our data set.



From the above plot we can see that X7 is the important variable, this is because, after removing this variable there is a high increase in MSE.

Training error (MSE):
Model is fit using the entire data set of 550 observations and we find MSE (mean squared error) corresponding to this entire set. We can use this formula : (mean(yhat - y)^2) to find training MSE.

Validation Set Approach:
We do not have a test set with us, so if we the data is fitted entirely on the data set available to us, it may lead to overfitting. Hence, we divide the data into 2 parts, one being the training set and other being the validation set (550 observations were split into 2 sets). We fit the model using the training set and then check the MSE using the validation set.

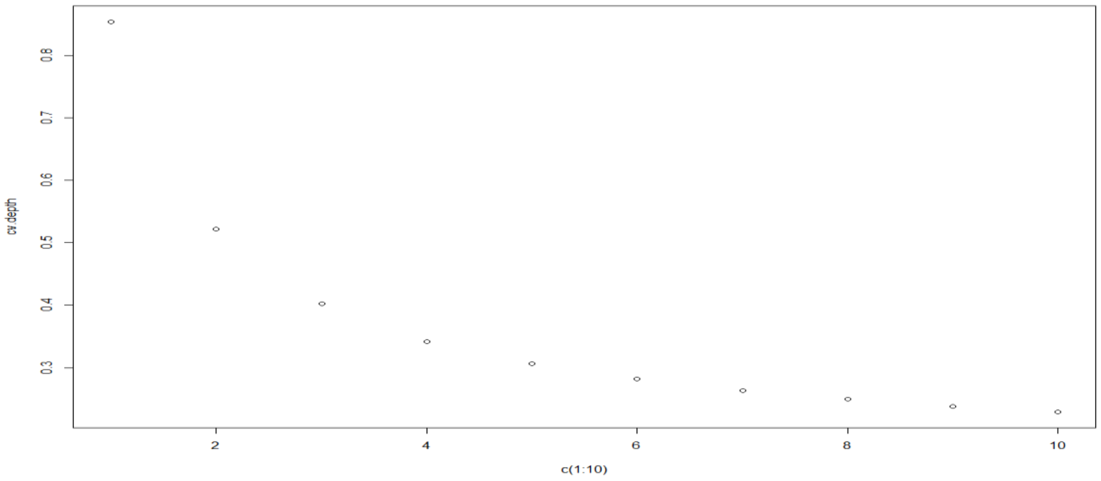
K-Fold Cross validation:
The Validation Set approach has a disadvantage as it randomly splits the data set into 2 parts, hence the estimated error that we get from each split is different. For this purpose we use cross validation, in this case we use K-Fold Cross validation with K = 10. For techniques like Random Forest, functions to perform cross validation are not readily available, hence we tried to develop a general function that can perform cross validation for various techniques. We did this by looping over K folds, and averaging out the mean squared error obtained for each set. (We are attaching the same for your perusal in the R file).

Training MSE	0.12
Validation MSE	0.62
Cross Validation error (MSE)	0.28

Boosting

From, scatter plot of Y1 (response) with each individual predictor (eg: X1, X2, …), we can see that they do not follow linear relationships, hence linear models are not a good fit for this data. So we are trying tree-based methods. Random Forest, Bagging, Boosting is preferred over normal regression trees as it aggregates different trees for improving the prediction accuracy.

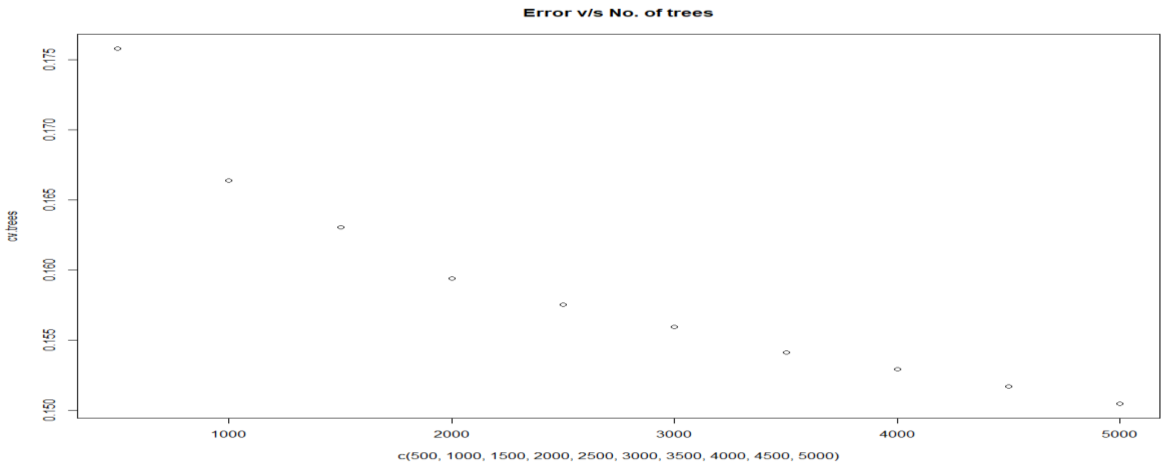
In case of boosting we improve upon the existing tree by using the information from that tree. Instead of developing large trees with all predictors we develop many small trees which learn from residual of previous tree. We select interaction depth, "d", for our model. This controls the complexity of the boosted ensemble. Hence, we can tune the value of parameter 'd' by using cross validation. We have plot cross-validation error of the model for different values of d, as can be seen below:



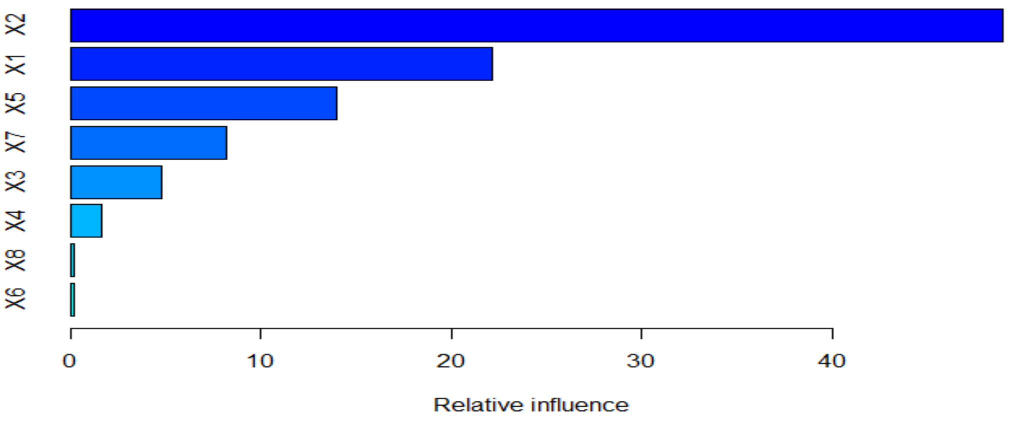
As we can see from the above plot, that cross validation error decreases as "d" increases. However, we can see that after d = 4, there is not a significant decrease in the error hence in order to reduce the complexity of the boosted ensemble we choose d = 4.

Shrinkage parameter is another important parameter for Boosting, however it depends on 'B' (the number of trees).

Distribution is another important parameter for the model, we select that as "gaussian" in this case as we are performing regression analysis.



As we can see from the above plot, that cross validation error decreases as “n.trees” (“B” value) increases. However, we can see that after n.trees = 4000, there is not a significant decrease in the error hence in order to reduce the overfitting of the model we choose n.trees = 4000.



Above figure shows us the relative influence of different predictor variables. We can see that X2 and X1 are the 2 most important variables.

Training error (MSE):

Model is fit using the entire data set of 550 observations and we find MSE (mean squared error) corresponding to this entire set. We can use this formula : (mean(yhat - y)^2) to find training MSE.

Validation Set Approach:

We do not have a test set with us, so if we the data is fitted entirely on the data set available to us, it may lead to overfitting. Hence, we divide the data into 2 parts, one being the training set and other being the validation set (550 observations were split into 2 sets). We fit the model using the training set and then check the MSE using the validation set.

K-Fold Cross validation:

The Validation Set approach has a disadvantage as it randomly splits the data set into 2 parts, hence the estimated error that we get from each split is different. For this purpose we use cross validation, in this case we use K-Fold Cross validation with K = 10. For techniques like Random Forest, functions to perform cross validation are not readily available, hence we tried to develop a general function that can perform cross validation for various techniques. We did this by looping over K folds, and averaging out the mean squared error obtained for each set. (We are attaching the same for your perusal in the R file).

Training MSE	0.04
Validation MSE	0.21
Cross Validation error (MSE)	0.15

Comparison

Of all the different regression models we tried, we now compare the 3 best models. We compare the cross validation errors of the 3 models to select the best model. Cross validation error can be considered as a good estimate of the test error. Hence, in absence of test data, we can consider cross validation error. Cross validation error for the 3 models is given below:

Model	Cross Validation Error
Ridge Regression	10.32
Random Forest	0.28
Boosting	0.15

As we can see from the above table that cross validation error is lowest for Boosting. Hence we conclude that Boosting is our best model. (The code for this model is given in the R file).

Hence, for this particular data set we can say that of the 3 models we are comparing tree based methods are better than Ridge regression method. Moreover, Boosting is better than random forest as it improves the existing tree by slowly learning based on the available information.

Analysis on Test Set

As a part of Phase 2 of the project we were provided with the test data set. There are 218 observations and 9 columns (8 predictors and 1 response variable) in the test data set. In training data set there were 550 observations.

Our best model is Boosting with n_trees (B) as 4000 and interaction.depth (d) as 4. The reasons for considering this as the best model is discussed above.

Hence, we find the test error by finding mean of the difference between predicted values (found using test ‘x’ values) and actual response values (test data set).

Test error (mean squared error) = mean((yhat – ytest)^2)

After employing the above formula on the test data set and using our best model,

Test error – 0.248

Test Error (MSE)	**0.248**

Improvement Upon Best Method

As discussed above, Boosting is our best model. We would like to improve this model, by changing parameters such as n_trees (B) and interaction.depth (d).
We found the test error (MSE) for different values of these parameters. And we will find the parameters corresponding to lowest test error.
We did this and got the best parameters as n_trees = 7000 and interaction.depth = 4.
Test error at these parameters (MSE) = mean((yhat – ytest)^2)
After employing the above formula on the test data set and using our improved model,
Test error – 0.230

Test Error (MSE)	**0.230**

Hence, test error corresponding to our improved model is 0.230.
Hence the best model according to our analysis is Boosting with optimum parameters n_trees = 7000 and interaction.depth = 4.

Submitted by:

- Niraj Kishore Dere
- Meet Snehal Shukla
- Dixeet Gaurang Purohit
- Aayush Sharma