DATA ANALYSIS PROJECT

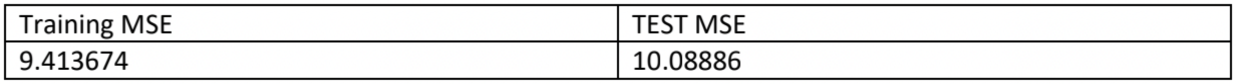
BY:
SAMUEL ZOWAM
PRABHA PADAMSINGH KANYAL

TEST RESULT AND IMPROVED MODEL

SECTION 6: EVALUATING TEST RESULT

For assessing each model, we were interested in the estimates of the test MSE to see how each method would perform on new data. However, to select the best model, we were interested in identifying the method that produced the lowest amount of test mean square error (MSE) using cross validation technique. After selecting the best model based on an approximation of the test MSE, we re–estimated our model parameters on the full training data and then made predictions of the response variable, given the actual test data. The true test MSE was found to be 10.08886.

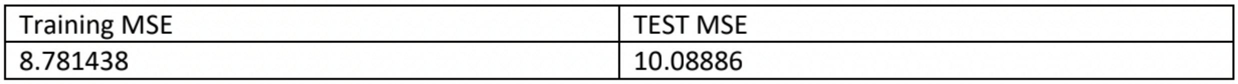
We did not quite expect our true test MSE to be close to the approximated value, and we also thought the true test MSE could either be smaller or larger. More so, since we used validation set approach for the test MSE approximation, we acknowledged that the cross-validation estimate could be relatively different from the true test MSE due to the variability in the approach - since the split is random and the test error is also dependent on where the split occurs. However, based on the approximation result which we found to be 9.413674, we thus conclude that the cross validation estimates of test MSE for our best model and the true test MSE are almost identical. For ease of reference, both values are presented below:



Training MSE	TEST MSE
9.413674	10.08886

Table 1

In comparison to the training error (which measures the accuracy of our best model in predicting the training data), we found the test MSE to be larger. Again, this was expected. To calculate the training MSE, we re–estimated our model parameters on the full training data and then made predictions of the response variable, given the same training data. For ease of reference, both values are presented below:



Training MSE	TEST MSE
8.781438	10.08886

Table 2

SECTION 7: MODEL IMPROVEMENT

At the start of the project, we selected our models based on model properties alone. For instance, we assumed trees regression not to outperform other advanced or more complicated approaches in terms of prediction accuracy (which was in fact the goal of the project).

However, upon taking a closer look at our data, we observed that the variable X5 has only two unique values (7 and 3.5), and decided it was a good reason to think of the variable X5 as a categorical variable. Knowing that Decision Trees can accommodate categorical variables easily and more efficiently without having to create dummy variables, we decided to use Trees Regression in our improved model case.

First, we fit a regression tree to the training data set. The plotted tree is shown below as well as the training MSE obtained. In terms of fitting the model, having the variable X5 coded as a continuous variable (factor) did not change our outcome.

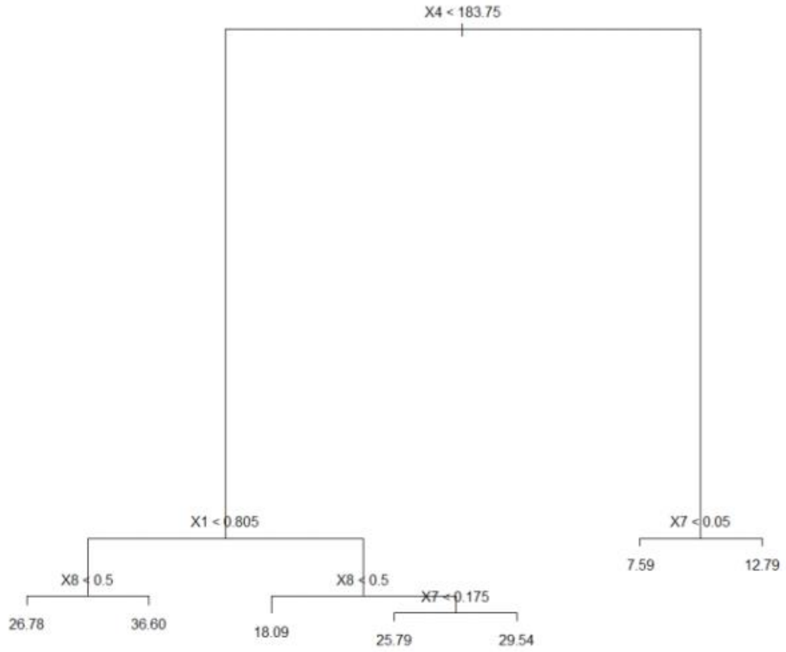
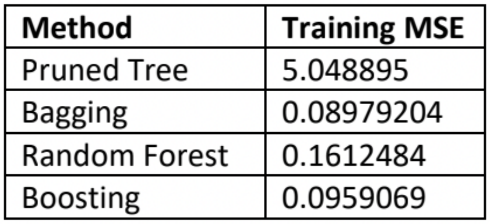


Fig 1: Regression tree fit to training data

The grown tree has 7 terminal nodes. Interpreting the tree; for X4 < 183.75, and X1< 0.805 and X8 < 0.5, we have one region (terminal node). For X4 < 183.75, and X1< 0.805 and X8 > 0.5, we have another region or terminal node. Similarly, for X4 > 183.75 and X7 < 0.005, we have another region. Following this logic, the entire result can as well be interpreted. The training error (training MSE) of the tree is 5.048895.

However, to achieve high prediction accuracy (since decision trees suffer from high variance), we also analyzed the data using tree pruning approach, bagging, random forest and boosting. Using Cross validation to determine the optimal level of tree complexity, we find the optimal tree size (or number of terminal nodes) to be 7 which is in fact what we had in our original tree. Hence, as expected, we get identical results when we pruned the tree given that the original tree itself is not large.

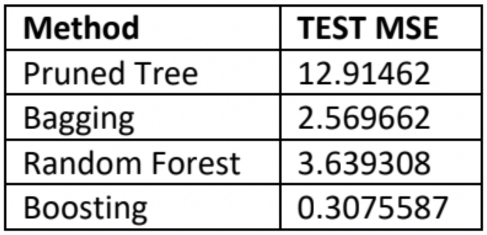
For Random Forest we set the random sample of predictors, m to be 4. For Boosting we set the number of trees, B as 5000; the shrinkage parameter, lambda as 0.01 and the number of splits, d as 4. We compared the results from each approach and then determined which variables were most important. The results are presented below:



Method	Training MSE
Pruned Tree	5.048895
Bagging	0.08979204
Random Forest	0.1612484
Boosting	0.0959069

Table 3

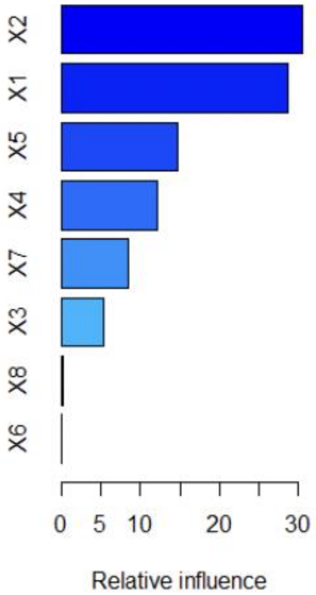
With the known test data, we directly calculated the test error (Test MSE) of all four trees-improvement approaches and selected the Boosting approach as our new Best Model since it had the lowest actual test MSE.



Method	TEST MSE
Pruned Tree	12.91462
Bagging	2.569662
Random Forest	3.639308
Boosting	0.3075587

Table 4

The relative influence of each variable is summarized below:



Relative influence of each predictor

Decision trees also proved to handle collinearity better, which was an evident issue in our previous regression models.

REFERENCE

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An introduction to statistical learning (Vol. 112, p.
18). New York: springer.