Machine Learning Kaggle Competition

Ingrid Wijaya

1 Introduction

In this paper, I will conduct a regression analysis using different approaches I attempted for the Kaggle Competition. My goal is to find a statistical model that will perform well given the public test data set and at the same time, generalize well to the private test dataset at the same time. After comparing all the statistical model, I will choose the model that has the lowest Root Mean Squared Error (RMSE), that is also lower than the four benchmark models given the testing data sets. In the following section, I will discuss the data analysis method and statistical model I attempted in this competition.

2 Methodology

To make a fair comparison on which models has the lowest validation set error rate and to make my results reproducible, I will set seed to 10. Since I do not have the full public test dataset to calculate the test error, I will use the validation set approach. I will split my data randomly into 2 parts, 70% training set and 30% validation set. Then, I will fit the model on the training set, and use the fitted model to predict the responses for the observations in the validation set. The resulting validation set error rate obtained will be assessed using RMSE for this regression problem setting, providing as an estimate of the test error rate. The formula for the Root Mean Squared Error used is:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(g_i - \hat{g}_i)^2}$$

In the following subsections, I will briefly discuss the different approaches I explored.

2.1 Linear Regression

As a starting point, I used a simple linear regression model to fit the training data as it is the fundamental starting point for all regression models. It is also much more interpretable as compared to non-linear methods. Also, I think it is a good comparison to other models.

2.2 Bagging

To construct a more powerful prediction model, I will apply one of the tree-based methods, bagging, to the dataset as it has been shown to give great improvements in terms of accuracy in its prediction and also reduces variance by averaging a set of observations. Bagging is a special case of random forest, for this dataset, m = p = 15.

2.3 Random Forest

To further improve my predictions accuracy, I will apply another tree-based method, random forests, as it can provide an improvement over bagged trees by decorrelating the trees. This in turns makes the average of the resulting trees less variable and hence more reliable. For the tuning parameter, m, I chose 3 different values – 3, 4, and 5. This includes the recommended m value when building a random forest of regression tree, that is, p/3 = 15/3 = 5. I set the number of trees to grow to 500 as there is a lot of rows in the data, so more trees are needed.

2.4 Boosting

To further improve my predictions accuracy, I will apply Boosting to the training data set. Unlike the approach in 2.2 and 2.3, boosting approach can prevent potential overfitting by learning slowly, and thus tends to perform better. There are 3 tuning parameters – shrinkage parameter λ, the number of trees B, and the number of d of splits in each tree. To select the best combination of the tuning parameters, I created a triple for-loop. The range of values I chose for λ is from 0.0001 to 0.01. The range of values I chose for B is 500, 1000, 1500, 2000, 2500, and 3000. The range of values I chose for d is 1, 2, 3, and 4.I try to ensure the range is large enough for each tuning parameter so that I can find the best set of tuning parameters that can help improve my predictions.

For each approach, if applicable, I find the best statistical model by finding the model with the lowest validation error rate. Then, I fit that best model on the training dataset, and use the fitted model to predict the responses for the observations in the test dataset. I will then obtain the test error rate. In the next section, I will discuss and analyze the results, particularly, about the validation and test error rates.

3 Results

3.1 Comparison Between Different Approaches

For the linear regression model, there is only one model, and it has a validation error rate of 1.459112.

For the bagged model, there is only one model, and it has a validation error rate of 1.4107, a great improvement from the linear regression model.

For the random forest approach, the validation error rates for the models with m = 3, 4, 5 are 1.398608, 1.395356, and 1.399285 respectively. Since the random forest model with m = 4 has the smallest validation error rates of 1. 395356, I chose that as the model to compare with other approaches. This random forest model shows a slight improvement from the bagged model.

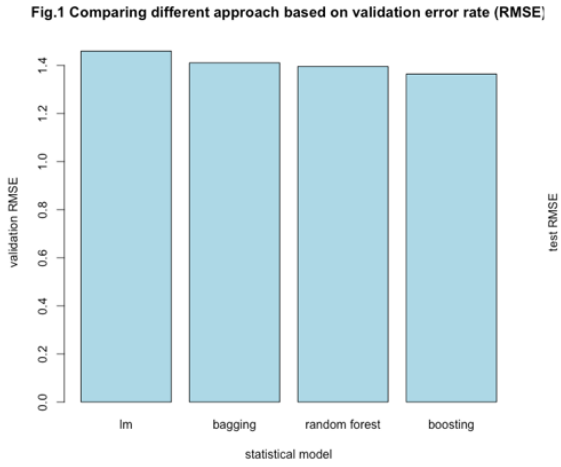
For the boosting approach, the model with the lowest validation error rates of 1.308332 has tuning parameters – $\lambda = 0.0501$, $B = 500$ and $d = 4$. This boosted model shows a major improvement from the random forest approach.

Then, I find the test error rates for each model by submitting my predictions onto Kaggle. The test error rates for the linear regression, bagged, random forest, and, boosted model are 1.31123, 1.26061, 1.25125, and 1.20241.

Model	Linear	Bagging	Random Forest	Boosting
Validation Error Rate	1.459112	1.4107	1. 395356	1. 308332
Test Error Rate	1. 31123	1. 26061	1. 25125	1. 20241

Table 1. Summary of validation and test error rate from each model.

Table 1 shows that we can see the same pattern of improvement, with the test error rate as with the validation error rate amongst the model.



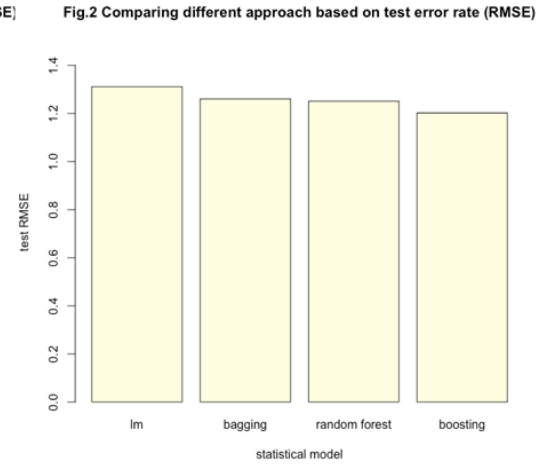


Fig. 1 shows that the boosted model has the lowest validation error rate and thus I chose it as my final model. Furthermore, Fig. 2 shows that the boosted model once again has the lowest test error rate, agreeing with the results I found by using the validation set approach. In the next subsection, I will discuss more about the final model.

3.2 Final model

I chose boosting as my final model since it has the lowest validation and test error rates. In other words, it outperformed other models that I have explored.

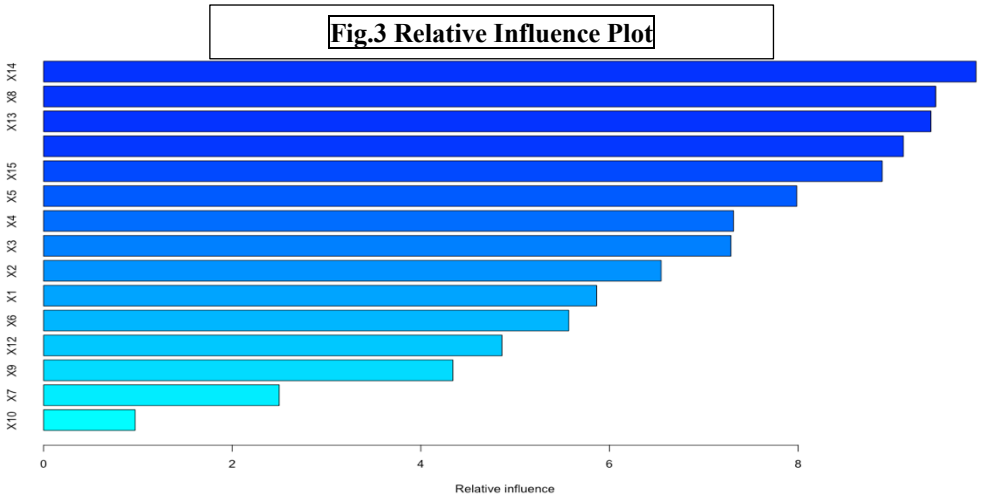


Fig. 3 shows that the two most important predictors in the boosted model is X14 and X8.

All in all, the models that I explored in this report is just a small portion of regression model that can predict the test data set. There are various other models and improvements that can be done to the approaches I explore, such as by tuning other parameters, to increase the accuracy of predictions.