**Prediction Accuracy of Different Modeling Approaches on a Supervised Learning Dataset**

**STA 5703 Data Mining Method I Final Project (Spring 2020)**

Prepared By: Mina Sonbol & Justin Kennedy

Table of Contents

[***Executive Summary*** 3](#_Toc38483443)

[Model Testing and Accuracy 3](#_Toc38483444)

[Important Feature Determination 4](#_Toc38483445)

[Impact of Important Features on Accuracy 4](#_Toc38483446)

[***Introduction*** 5](#_Toc38483447)

[***Methodology*** 5](#_Toc38483448)

[Data Preparation and Model Testing 5](#_Toc38483449)

[***Modeling Performance Measures*** 8](#_Toc38483450)

[Logistic Regression with no Interactions 9](#_Toc38483451)

[Logistic Regression with Two-Way Interactions 9](#_Toc38483452)

[Random Forest 9](#_Toc38483453)

[Gradient Boosting 10](#_Toc38483454)

[***Model Selection and Reasoning*** 10](#_Toc38483455)

[***Conclusion*** 12](#_Toc38483456)

[***Appendix*** 13](#_Toc38483457)

[ROC Curves for each Model 13](#_Toc38483458)

[Feature Importance, Ranked by Random Forest Model 14](#_Toc38483459)

[Optimal number of Trees for Gradient Boosting 16](#_Toc38483460)

[Summary Statistics of All Features 17](#_Toc38483461)

# Executive Summary

The purpose of this project was to apply several different modeling techniques to a single dataset and analyze the differences in model efficacy.  The data science discipline is both art and science. For any given dataset multiple models can “work” but this is where the art of data science is applied. While there are models that tend to perform better as a general rule, it is possible to improve accuracy scores with any given model by applying methods of data preparation. Our project included missing value imputation, feature selection and reduction, model creation, model analysis, model optimization and comparison of selected fitted models using various comparison statistics.

## Model Testing and Accuracy

The four basic models that we used were logistic regression without interaction, logistic regression with two-way interactions, random forest and gradient boosting. In general, simple logistic regression had the worst achieved accuracy of .69 and gradient boosting performed the best, with an achieved accuracy of .72. However, we were able to slightly improve the accuracy of each of the models by performing basic data preparation and cleansing. In the end, we selected the gradient boosting model because it presented the highest consistent accuracy and precision.

## Important Feature Determination

The original data set contained 50,000 observations across 80 different features. In general, feature selection is an exercise in balancing *bias* and *variance;* too few features will result in a model that has an increased bias, while too many features will result in increased variance. Additionally, the number of features increases the computational complexity of building any model and so feature reduction, without decreased accuracy, is a desired step. We decided to execute a series of feature reduction methods to try and determine which features were the most critical to the accuracy of the various models. These feature reduction outcomes included stepwise modeling, multicollinearity analysis, and determining the importance of features by utilizing random forest modeling. Ultimately, we ended up with 19 features that we determined to be the most important.

## Impact of Important Features on Accuracy

We performed two broad feature reductions. The first was a reduction to 37 features and the second was a reduction to 19 features. We found that the reduction in features to 37 resulted in a decrease across almost all performance measurements for all models. In fact, there was only one registered improvement in any measurement following the reduction to 37 features, and that was for the true positive rate for our logistic regression model with two-way interactions. However, this trend reversed when we performed the final reduction to 19 features. We saw an improvement to most measurements when modeling using the 19-feature set. Overall, since the models remained at least as accurate, and in some cases improved, we can determine that these 19 most important features have a positive impact on the accuracy of all the models.

# Introduction

To begin our analysis, we took the raw dataset and imputed missing values (using the feature mean) and removed features that contained no data. From this new dataset, we ran the various models and recorded the different statistics appropriate to each model. This established a baseline for us to analyze how feature reduction may or may not affect the accuracy of each individual model. After this baseline was established, we performed the various feature reduction methods one at a time while recording the impact that these feature reduction steps had on the performance of each of our models. We then selected a model combined with our ideal feature set size as the final predictive model.

# Methodology

## Data Preparation and Model Testing

We started by importing the data into a dataframe in R and running a summary() function. This provided us basic information about each feature and was the first step in determining how to perform a basic iteration of data preparation. Since our first step in this project was to establish a baseline with as many features as possible, we decided to initially only remove the features that had zero variance in their values. For example, features feat29, feat55, feat51, feat50, feat49, and feat47 all had values of “0” when the summary() function was ran (see Appendix – Table 1). Because these features would not contribute to the overall performance of the model, we removed them prior to running any modeling. Additionally, we also removed the feature exampleid as it was also a non-contributing feature. After removing the non-contributing features, we then proceeded to identify columns with missing values and then imputing those values. The first step was to identify features with missing values and then create missing value indicators for each identified instance. After the modified dataframe was validated as no longer containing missing values, we then imputed the missing values by using the mean of the feature in question. We then validated that the derived value was an accurate mean by manually comparing to one of the features. Following the validation step, we then applied our mean calculation to the entire dataframe and validated that the resultant dataframe contained zero missing values. The end state dataframe after our initial data preparation and cleansing consisted of 50,000 observations and a feature set of 72.

Following the removal of the non-contributing features and imputation of missing values, we ran the initial logistic regression model with no interaction terms included. We ran a no-interaction logistic regression model on three different versions of the original data set: a dataset with minimal cleansing (50,000 observations and 72 features), a dataset with feature reduction from random forest analysis combined with VIF multicollinearity analysis (50,000 observations and 37 features) and a dataset with feature reduction from stepwise modeling (50,000 observations and 19 features). After running the logistic regression modeling on the three different feature sets, we decided that the best feature set to use was the one provided by the stepwise model. The stepwise model reduced our feature set by 73% and improved the overall performance of the logistic regression model without interactions. The formula for this final logistic regression formula, without interaction, was:   
After establishing a baseline feature set for logistic regression with no interactions, we then proceeded to go through the same process with two-way interactions to see if the results were the same. The three features we chose for our two-way interactions were feat13, feat66 and feat40. We chose these features as a combination of analysis of the random forest variance importance plot (see Appendix – Figures 8, 9), the multicollinearity of the VIF function and simple trial and error. What we noticed when we began testing was that simply choosing the three “highest” or “best” features on any single given analysis did not result in the best performing models. This was confirmed when an analysis of the model showed that our selected features for interaction had a statistically significant p-value (<.05) and therefore were collinear with each other. Our final equation for our logistic regression with two-way interaction was:

For the random forest model, we again began with the full dataset of 72 features to get baseline performance prior to running the model on feature-optimized datasets. We did modify the model to provide a ranking of the importance of the features provided to the model. This importance was ranked in two different ways, by the mean decrease in both Gini and accuracy if this feature were removed from the model. We used these measurements as a part of our analysis when determining critical features to include in our reduced feature set (see Appendix – Figures 8,9). Our random forest model ultimately performed best on the dataset with all features included.

Lastly, we performed the same three dataset test using the gradient boosting method. The biggest change we made to the gradient boosting method was the ability to utilize both Out-of-Bag (OOB) and Cross-Validation (CV) estimates to arrive at the optimal number of trees to utilize in the gradient boosting method (see Appendix – Figures 10,11). The final model that we selected for the gradient boosting method was one that utilized 373 trees, which we found by utilizing the CV estimates. This method using 373 trees performed best on the dataset with 19 features.

# Modeling Performance Measures

We measured each of our models by using five standard performance measures: confusion matrix, accuracy, true positive rate (or sensitivity), false positive rate and precision. The definitions of these measurements and the results of our models are listed below.

|  |  |
| --- | --- |
| **Actual Negative** | **Actual Positive** |
| **Predicted Negative** | True Negative (TN) | False Negative (FN) |
| **Predicted Positive** | False Positive (FP) | True Positive (TP) |
| **Accuracy** | (TP+TN)/(TP+TN+FP+FN) | |
| **True Positive Rate (Sensitivity)** | TP/(TP+FN) | |
| **False Positive Rate (Miss Rate)** | FP/(FP+TN) | |
| **Precision (Positive Predictive Value)** | TP/(TP+FP) | |

## Logistic Regression with no Interactions

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Logistic Regression | | All Features | | 37 Features | | 19 Features | |
| Actual Neg. | Actual Pos. | Actual Neg. | Actual Pos. | Actual Neg. | Actual Pos. |
| Confusion Matrix | Pred. Neg | 5358 | 2229 | 5328 | 2242 | 5507 | 2244 |
| Pred. Pos | 2183 | 5229 | 2213 | 5216 | 2034 | 5214 |
| Accuracy | | 0.705 | | 0.703 | | 0.714 | |
| True Positive | | 0.701 | | 0.699 | | 0.699 | |
| False Positive | | 0.289 | | 0.293 | | 0.270 | |
| Precision | | 0.705 | | 0.702 | | 0.719 | |
| AUC | |  | |  | | .792 | |

## Logistic Regression with Two-Way Interactions

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Logistic Regression with Two-Way Interactions | | All Features | | 37 Features | | 19 Features | |
| Actual Neg. | Actual Pos. | Actual Neg. | Actual Pos. | Actual Neg. | Actual Pos. |
| Confusion Matrix | Pred. Neg | 5425 | 2174 | 5091 | 2131 | 5512 | 2259 |
| Pred. Pos | 2116 | 5284 | 2450 | 5327 | 2029 | 5199 |
| Accuracy | | 0.714 | | 0.694 | | 0.714 | |
| True Positive | | 0.708 | | 0.714 | | 0.697 | |
| False Positive | | 0.281 | | 0.325 | | 0.269 | |
| Precision | | 0.705 | | 0.685 | | 0.719 | |
| AUC | |  | |  | | .799 | |

## Random Forest

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Random Forest | | All Features | | 37 Features | | 19 Features | |
| Actual Neg. | Actual Pos. | Actual Neg. | Actual Pos. | Actual Neg. | Actual Pos. |
| Confusion Matrix | Pred. Neg | 4527 | 1771 | 4443 | 1855 | 4482 | 1816 |
| Pred. Pos | 1831 | 4371 | 1925 | 4277 | 1804 | 4398 |
| Accuracy | | 0.711 | | 0.697 | | 0.71 | |
| True Positive | | 0.712 | | 0.697 | | 0.707 | |
| False Positive | | 0.288 | | 0.302 | | 0.287 | |
| Precision | | 0.718 | | 0.705 | | 0.709 | |
| AUC | |  | |  | | .804 | |

## Gradient Boosting

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Gradient Boosting | | All Features | | 37 Features | | 19 Features | |
| Actual Neg. | Actual Pos. | Actual Neg. | Actual Pos. | Actual Neg. | Actual Pos. |
| Confusion Matrix | Pred. Neg | 4584 | 1801 | 4506 | 1946 | 4584 | 1714 |
| Pred. Pos | 1714 | 4401 | 1792 | 4256 | 1718 | 4484 |
| Accuracy | | 0.718 | | 0.7 | | 0.725 | |
| True Positive | | 0.709 | | 0.686 | | 0.711 | |
| False Positive | | 0.272 | | 0.285 | | 0.272 | |
| Precision | | 0.719 | | 0.703 | | 0.72 | |
| AUC | |  | |  | | .855 | |

# Model Selection and Reasoning

We used a combination of five different measurements to determine the most effective model for the dataset. These measurements included accuracy, defined as the number of cases predicted correctly (both positive and negative) divided by the total number of cases; true positive rate (or sensitivity), defined as the total number of correctly predicted positive outcomes over all positive outcomes; false positive rate, defined as the number of incorrect positive predictions over the total number of negative outcomes; and precision, defined as the number of accurately predicted positive results over the total number of positive predictions. Additionally, we also included the Area Under the Curve (AUC) measurement each model, which is a representation of how capable the model is of distinguishing between the positive class and the negative class. We only included this measurement for the best performing feature set.

When attempting to determine which model is “best”, consideration needs to be taken into account as to what the model is measuring. For example, certain situations may desire a model that minimizes the false positive rate, such as drug screening or other toxicology applications. Other situations may require a model that maximizes the rate at which the positive outcome is accurately predicted, such environmental testing for toxicants. Simply put, each situation requires an analysis of the task at hand to decide which measurements should be maximized, or minimized, to receive the label of being “the best”.

In this case, the dataset provided was from a sample derived from a particle physics experiment in 2004. At its core, the issue was that when these experiments are conducted it is difficult to accurately determine if the particles belong to one of two classes. To try and solve this problem, 78 features of these particles were provided along with 50,000 observations. In the realm of hard sciences, accuracy tends to be the most important factor, followed closely by precision; precision comes after accuracy because being precisely wrong is of little to no use. Also, AUC is also looked at as a measure of how well a model is fit, but in this case we use it as a supporting measurement since accuracy and precision are more important in a problem such as this.

Using this reasoning, we decided that the best fit model is the gradient boosting model. This model performed best across the 19-feature dataset, but only marginally so as compared to the full 72-feature set. However, because one aspect of effective model creation is achieving maximum results with the minimum number of features, the ability to reduce the number of features by 73% while also increasing accuracy makes the gradient boost model on the reduced feature set the clear winner. This model resulted in an accuracy of .725, a true positive rate of .711, a false positive rate of .272, a precision of .72 and an AUC of .855.

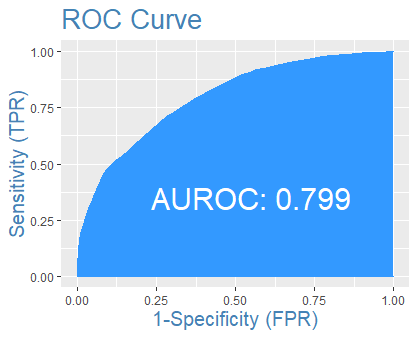
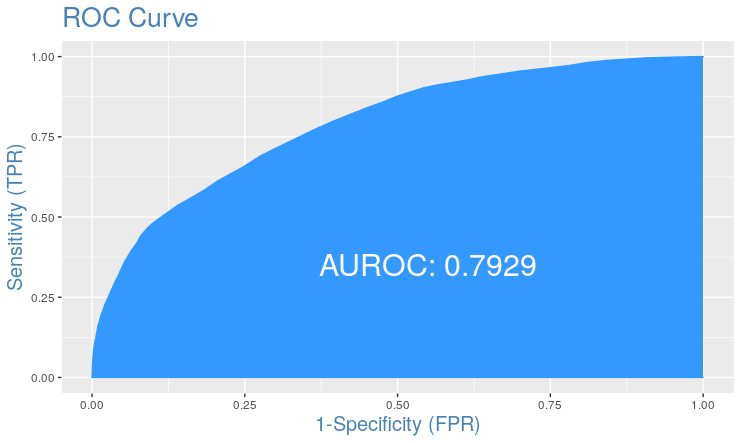
# Conclusion

The process of selecting a model is equal parts art and science. They are two sides of the same coin and attempting to apply both equally to solving a problem will lead to the best results. We conducted this experiment for the express purpose of taking a supervised learning dataset and apply four different modeling techniques to see if we could determine which of these four would be the most effective at building a model that could correctly place any given observation into one of two binary classes. Through a process of data familiarization, feature reduction, multicollinearity analysis, feature importance ranking using random forest modeling, experimentation and analysis, we arrived at the conclusion that the most efficient model was the gradient boosting model ran across an optimized feature set of the 19 most important features.

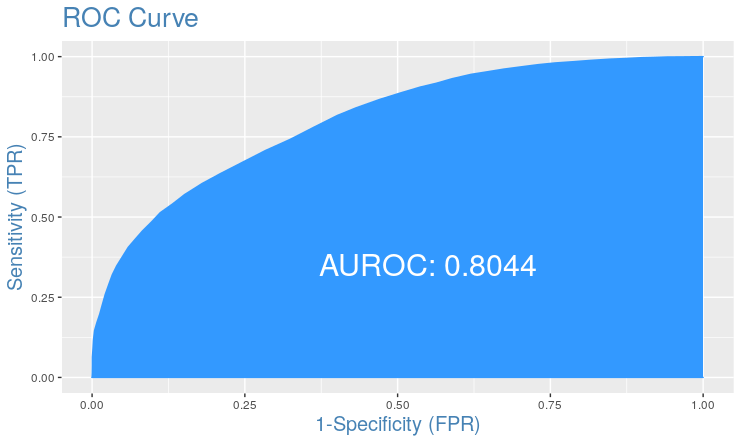
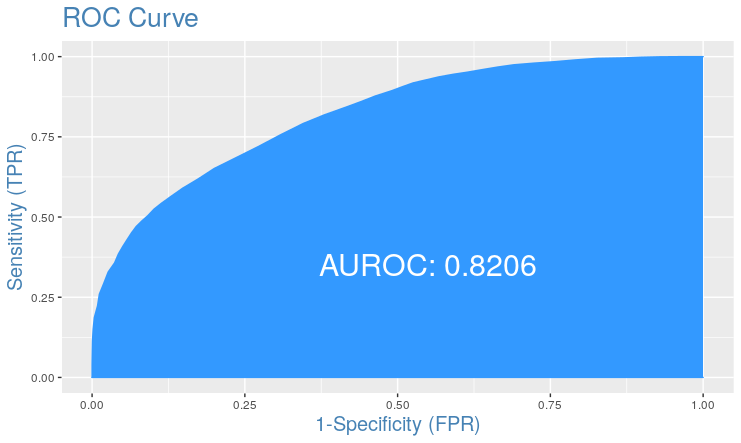
# Appendix

## ROC Curves for each Model

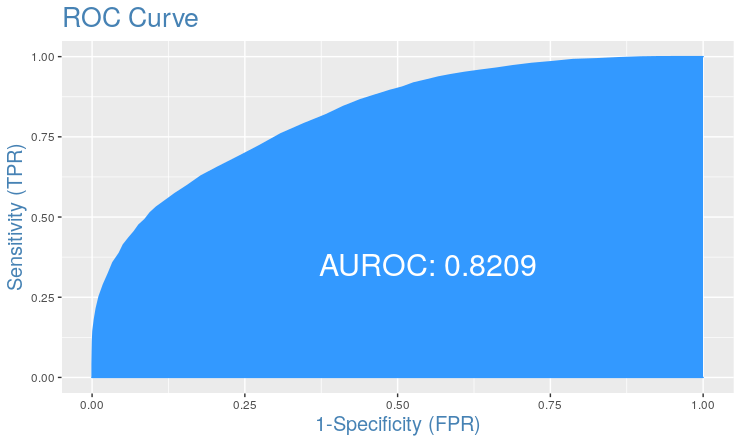
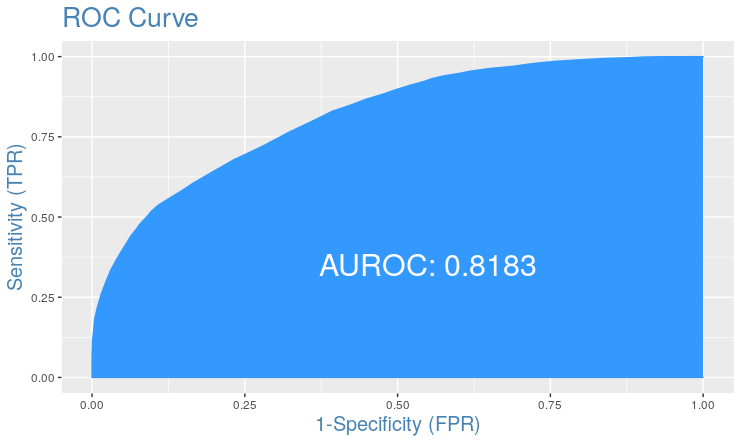
**Figure 1- Logistic Regression without Interaction Figure 2 - Logistic Regression with Two-Way Interaction**



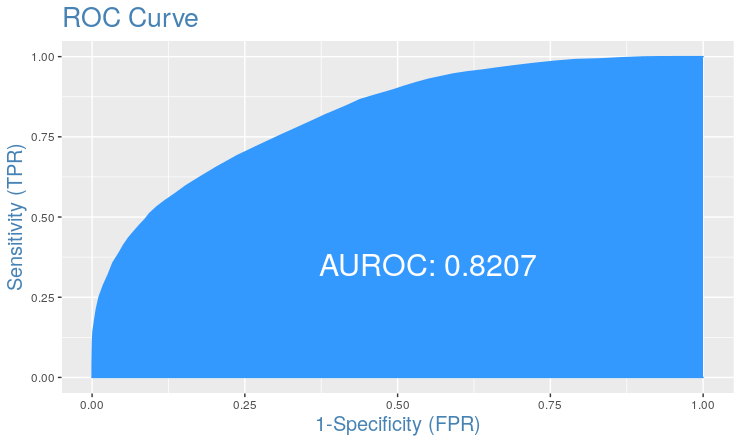
**Figure 3- Random Forest Figure 4- Boosting, 500 Trees**

**Figure 5- Boosting, 500 Trees, 3CV Figure 6 -Boosting, 193 Trees**

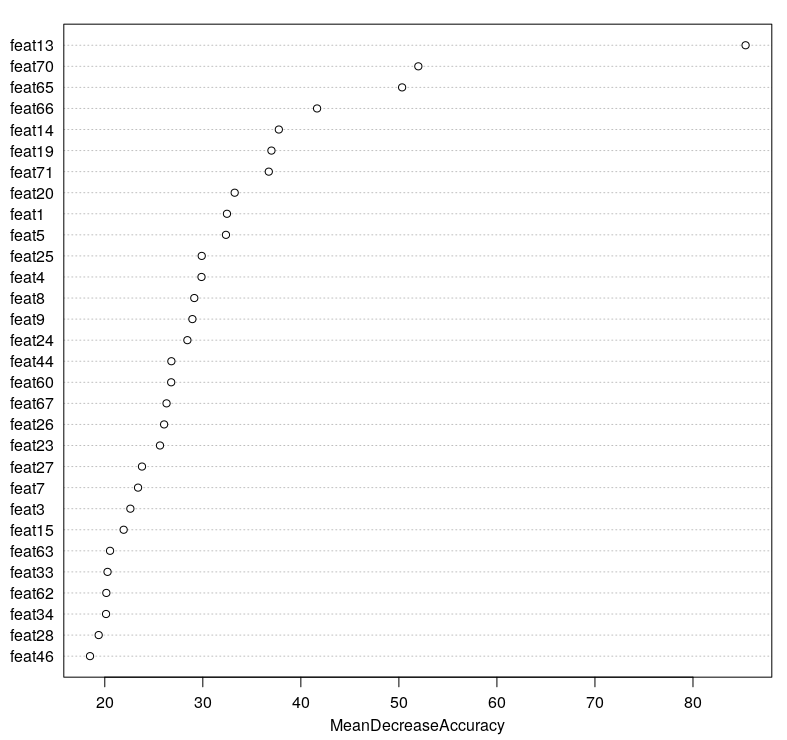
 

**Figure 7- Boosting, 393 Trees**

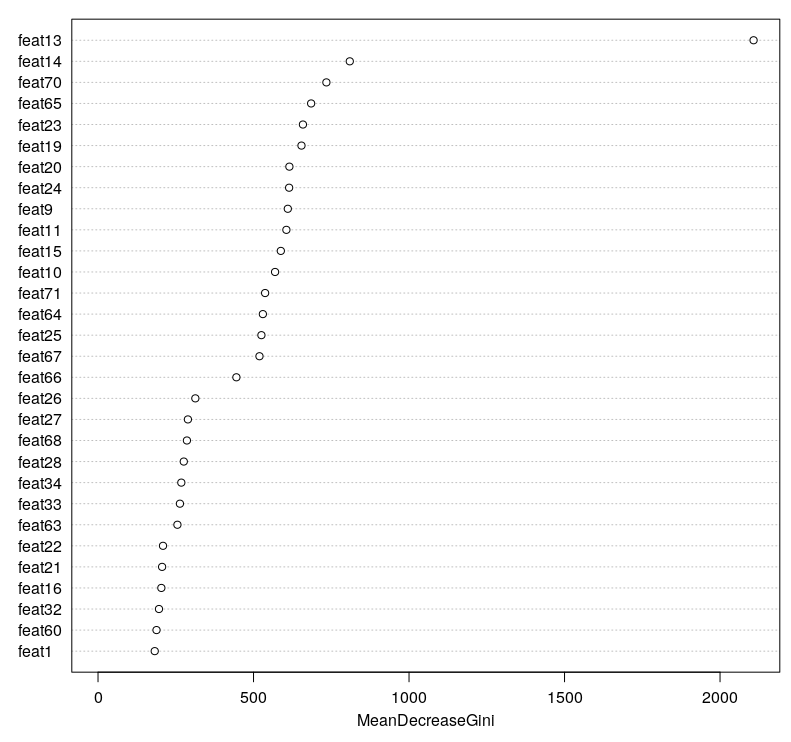


## Feature Importance, Ranked by Random Forest Model

**Figure 8- Feature Importance Measured by Mean Decrease in Accuracy, Random Forest Model**

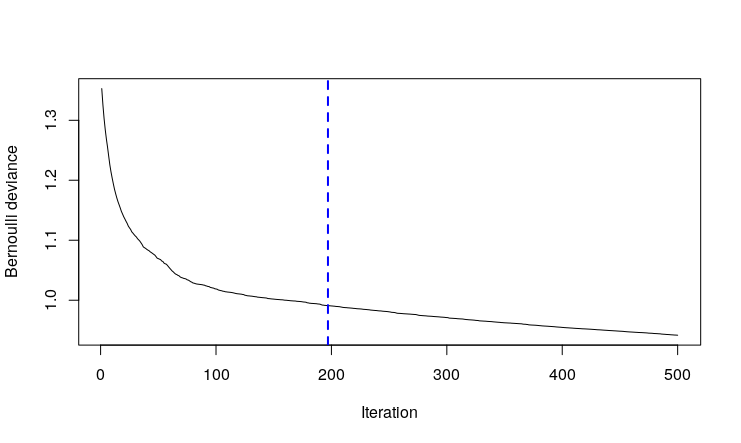


**Figure 9- Feature Importance Measured by Mean Decrease in Gini, Random Forest Model**

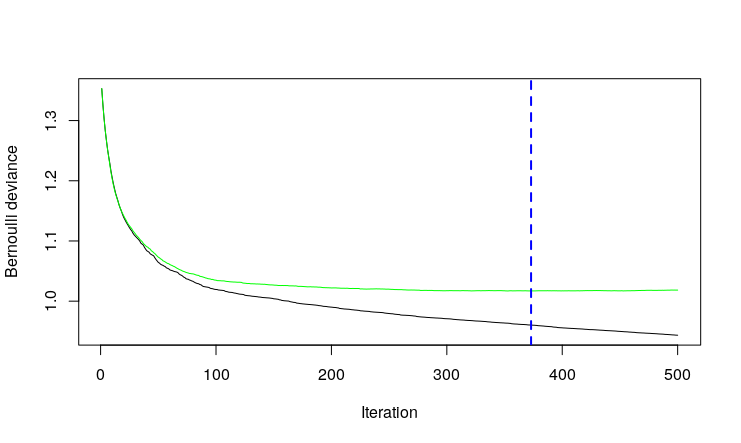


## Optimal number of Trees for Gradient Boosting

**Figure 10- Bernoulli deviance decrease with increased number of trees, using OOB. Blue line represents optimal number of trees at 197**



**Figure 11-Bernoulli deviance decrease with number of trees, using Cross Validation. Blue line represents optimal number at 373**



## Summary Statistics of All Features

**Table 1-Summary Statistics of all Features**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Feature | 1st Qu. | 3rd Qu. | Max. | Mean | Median | Min. | NA's | Skewness |
| exampleid | 12501 | 37500 | 50000 | 25000 | 25000 | 1.00 | 0.00 | 0.00 |
| feat1 | 0.00 | 0.00 | 2.64 | 0.16 | 0.00 | 0.00 | 0.00 | 2.83 |
| feat10 | 0.25 | 1.02 | 5.28 | 0.67 | 0.60 | 0.00 | 0.00 | 0.66 |
| feat11 | -0.81 | 0.17 | 1.00 | -0.28 | -0.45 | -1.00 | 0.00 | 0.61 |
| feat12 | -1.00 | 1.00 | 1.00 | -0.01 | 0.00 | -1.00 | 0.00 | 0.02 |
| feat13 | -0.28 | 0.36 | 1.00 | 0.01 | 0.00 | -1.00 | 0.00 | -0.01 |
| feat14 | 0.00 | 0.00 | 1.00 | 0.00 | 0.00 | -1.00 | 0.00 | 0.01 |
| feat15 | 0.00 | 0.00 | 1.00 | 0.00 | 0.00 | -1.00 | 0.00 | -0.32 |
| feat16 | 0.00 | 1.00 | 9.00 | 0.86 | 1.00 | 0.00 | 0.00 | 1.50 |
| feat17 | 0.00 | 0.00 | 4.00 | 0.17 | 0.00 | 0.00 | 0.00 | 2.46 |
| feat18 | 0.00 | 0.00 | 0.50 | 0.08 | 0.00 | 0.00 | 0.00 | 1.91 |
| feat19 | 0.80 | 1.39 | 6.08 | 1.12 | 1.07 | 0.00 | 0.00 | 0.68 |
| feat2 | 0.00 | 0.00 | 3.43 | 0.08 | 0.00 | 0.00 | 0.00 | 4.42 |
| feat20 | -0.81 | 0.81 | 4.51 | 0.00 | 0.50 | -2.45 | 34202 | NA |
| feat21 | -0.90 | 0.88 | 1.00 | -0.07 | -0.51 | -1.00 | 34202 | NA |
| feat22 | -0.87 | -0.40 | 0.00 | -0.62 | -0.68 | -1.00 | 34202 | NA |
| feat23 | 0.54 | 1.20 | 4.51 | 0.93 | 0.82 | 0.00 | 0.00 | 1.24 |
| feat24 | 0.47 | 1.05 | 4.79 | 0.83 | 0.71 | 0.00 | 0.00 | 1.41 |
| feat25 | 0.25 | 0.69 | 6.45 | 0.52 | 0.46 | 0.00 | 0.00 | 1.87 |
| feat26 | 0.00 | 0.81 | 1.00 | 0.40 | 0.35 | 0.00 | 0.00 | 0.22 |
| feat27 | 0.00 | 0.16 | 0.25 | 0.09 | 0.09 | 0.00 | 0.00 | 0.22 |
| feat28 | 0.00 | 0.00 | 1.00 | 0.01 | 0.00 | 0.00 | 0.00 | 7.86 |
| feat29 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 30062 | NA |
| feat3 | 0.00 | 0.00 | 1.00 | -0.05 | 0.00 | -1.00 | 0.00 | -1.56 |
| feat30 | 0.00 | 1.00 | 1.00 | 0.40 | 0.00 | 0.00 | 0.00 | 0.41 |
| feat31 | -1.00 | 1.00 | 1.00 | 0.00 | 0.00 | -1.00 | 0.00 | 0.00 |
| feat32 | 0.00 | 625 | 10000 | 362 | 0.00 | 0.00 | 0.00 | 6.00 |
| feat33 | 0.00 | 6.10 | 75.00 | 4.37 | 3.81 | 0.00 | 0.00 | 2.63 |
| feat34 | 0.00 | 0.19 | 0.53 | 0.11 | 0.11 | 0.00 | 0.00 | 0.55 |
| feat35 | 0.00 | 0.00 | 1.00 | 0.02 | 0.00 | 0.00 | 0.00 | 6.29 |
| feat36 | 0.00 | 0.00 | 1.00 | 0.02 | 0.00 | 0.00 | 0.00 | 6.36 |
| feat37 | 0.00 | 0.00 | 7.57 | 0.00 | 0.00 | -6.47 | 0.00 | 0.27 |
| feat38 | 0.00 | 0.00 | 9.44 | 0.00 | 0.00 | -10.77 | 0.00 | 1.06 |
| feat39 | 0.00 | 0.00 | 14.76 | 0.00 | 0.00 | -17.32 | 0.00 | -1.77 |
| feat4 | 0.00 | 0.00 | 1.00 | 0.00 | 0.00 | -1.00 | 0.00 | 0.00 |
| feat40 | 0.00 | 0.00 | 1.94 | 0.00 | 0.00 | 0.00 | 0.00 | 18.29 |
| feat41 | 0.00 | 0.00 | 3.55 | 0.00 | 0.00 | 0.00 | 0.00 | 27.96 |
| feat42 | 0.00 | 0.00 | 6.19 | 0.01 | 0.00 | 0.00 | 0.00 | 24.13 |
| feat43 | 0.00 | 1.00 | 1.00 | 0.60 | 1.00 | 0.00 | 0.00 | -0.41 |
| feat44 | 0.00 | 0.78 | 2.88 | 0.43 | 0.00 | 0.00 | 14469 | NA |
| feat45 | 0.00 | 0.88 | 1.00 | 0.37 | 0.00 | 0.00 | 14469 | NA |
| feat46 | -0.59 | 0.00 | 0.00 | -0.27 | 0.00 | -1.00 | 14469 | NA |
| feat47 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | NA |
| feat48 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | NA |
| feat49 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | NA |
| feat5 | 0.00 | 0.00 | 2.72 | 0.13 | 0.00 | 0.00 | 0.00 | 3.24 |
| feat50 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | NA |
| feat51 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | NA |
| feat52 | 0.00 | 10.00 | 10.00 | 3.72 | 0.00 | 0.00 | 0.00 | 0.53 |
| feat53 | 0.00 | 0.13 | 0.25 | 0.06 | 0.00 | 0.00 | 0.00 | 0.95 |
| feat54 | 0.00 | 0.00 | 1.00 | 0.01 | 0.00 | 0.00 | 0.00 | 8.49 |
| feat55 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 18602 | NA |
| feat56 | 0.00 | 0.00 | 1.00 | 0.00 | 0.00 | -1.00 | 0.00 | 0.00 |
| feat57 | 0.00 | 464 | 10000 | 218.80 | 0.00 | 0.00 | 0.00 | 7.08 |
| feat58 | 0.00 | 4.40 | 75.00 | 2.66 | 0.00 | 0.00 | 0.00 | 3.33 |
| feat59 | 0.00 | 0.14 | 0.54 | 0.07 | 0.00 | 0.00 | 0.00 | 1.24 |
| feat6 | 0.00 | 0.00 | 3.05 | 0.05 | 0.00 | 0.00 | 0.00 | 5.76 |
| feat60 | 0.00 | 0.69 | 3.04 | 0.35 | 0.00 | 0.00 | 0.00 | 1.35 |
| feat61 | 0.00 | 0.82 | 1.00 | 0.31 | 0.00 | 0.00 | 0.00 | 0.67 |
| feat62 | -0.43 | 0.00 | 0.00 | -0.22 | 0.00 | -1.00 | 0.00 | -1.18 |
| feat63 | 0.00 | 0.00 | 22.00 | 0.00 | 0.00 | -22.00 | 0.00 | -0.04 |
| feat64 | 0.00 | 0.01 | 0.10 | 0.01 | 0.01 | 0.00 | 0.00 | 2.75 |
| feat65 | 0.67 | 1.28 | 5.63 | 0.97 | 0.92 | 0.00 | 0.00 | 0.16 |
| feat66 | -1.00 | 1.00 | 1.00 | 0.00 | 0.00 | -1.00 | 0.00 | 0.01 |
| feat67 | 0.75 | 0.97 | 1.00 | 0.78 | 0.90 | 0.00 | 0.00 | -1.87 |
| feat68 | 0.00 | 0.57 | 1.00 | 0.16 | 0.00 | -1.00 | 0.00 | -0.10 |
| feat69 | -1.00 | 1.00 | 1.00 | 0.01 | 0.00 | -1.00 | 0.00 | -0.01 |
| feat7 | 0.00 | 0.00 | 1.00 | -0.04 | 0.00 | -1.00 | 0.00 | -2.23 |
| feat70 | 0.00 | 0.00 | 1.00 | 0.00 | 0.00 | -1.00 | 0.00 | 0.01 |
| feat71 | 0.00 | 0.00 | 0.91 | 0.00 | 0.00 | -0.91 | 0.00 | -0.01 |
| feat72 | 0.00 | 0.00 | 1.00 | 0.05 | 0.00 | 0.00 | 0.00 | 3.48 |
| feat73 | 0.00 | 0.00 | 3.43 | 0.07 | 0.00 | 0.00 | 0.00 | 5.02 |
| feat74 | 0.00 | 0.00 | 1.00 | -0.01 | 0.00 | -1.00 | 0.00 | -1.42 |
| feat75 | 0.00 | 0.00 | 1.00 | 0.00 | 0.00 | -1.00 | 0.00 | -0.04 |
| feat76 | 0.00 | 0.00 | 3.00 | 0.09 | 0.00 | 0.00 | 0.00 | 3.46 |
| feat77 | 0.00 | 0.00 | 0.39 | 0.00 | 0.00 | 0.00 | 0.00 | 12.11 |
| feat78 | 0.00 | 0.00 | 1.00 | 0.07 | 0.00 | 0.00 | 0.00 | 3.22 |
| feat8 | 0.00 | 0.00 | 1.00 | 0.00 | 0.00 | -1.00 | 0.00 | 0.06 |
| feat9 | 0.52 | 1.11 | 6.70 | 0.85 | 0.79 | 0.00 | 0.00 | 0.86 |
| target | 0.00 | 1.00 | 1.00 | 0.50 | 0.00 | 0.00 | 0.00 | 0.01 |

**Table 2-Feature Importance for Bagging and Cross Validation as Measured by Gradient Boosting**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| OOB Feature | OOB Importance | CV Feature | CV Importance | Similar Rank |
| feat13 | 38.153 | feat13 | 37.9167 | yes |
| feat14 | 7.311 | feat63 | 7.3024 | no |
| feat63 | 7.245 | feat65 | 7.0871 | no |
| feat65 | 6.244 | feat14 | 6.6168 | no |
| feat71 | 5.270 | feat71 | 5.2567 | yes |
| feat66 | 4.722 | feat66 | 4.7177 | yes |
| feat1 | 2.878 | feat70 | 2.1369 | no |
| feat15 | 1.833 | feat1 | 2.1200 | no |
| feat20 | 1.697 | feat15 | 1.7491 | no |
| feat19 | 1.693 | feat20 | 1.7105 | no |
| feat5 | 1.625 | feat4 | 1.5111 | no |
| feat70 | 1.527 | feat5 | 1.4126 | no |
| feat4 | 1.327 | feat19 | 1.3324 | no |
| feat8 | 1.148 | feat11 | 1.2486 | no |
| feat11 | 1.093 | feat9 | 1.1421 | no |
| feat10 | 1.012 | feat8 | 1.1395 | no |
| feat23 | 0.983 | feat23 | 0.9495 | yes |
| feat24 | 0.918 | feat10 | 0.9474 | no |
| feat9 | 0.892 | feat64 | 0.9417 | no |
| feat64 | 0.872 | feat18 | 0.8180 | no |
| feat18 | 0.721 | feat24 | 0.7689 | no |
| feat27 | 0.599 | feat16 | 0.6945 | no |
| feat16 | 0.580 | feat67 | 0.6506 | no |
| feat53 | 0.550 | feat31 | 0.6081 | no |
| feat25 | 0.530 | feat68 | 0.5999 | no |
| feat31 | 0.518 | feat75 | 0.5669 | no |
| feat67 | 0.424 | feat34 | 0.5094 | no |
| feat75 | 0.419 | feat27 | 0.4978 | no |
| feat74 | 0.408 | feat74 | 0.4743 | yes |
| feat33 | 0.381 | feat25 | 0.4726 | no |
| feat22 | 0.363 | feat53 | 0.3989 | no |
| feat68 | 0.340 | feat33 | 0.3438 | no |
| feat26 | 0.338 | feat77 | 0.3172 | no |
| feat77 | 0.333 | feat73 | 0.2943 | no |
| feat59 | 0.312 | feat22 | 0.2879 | no |
| feat34 | 0.307 | feat26 | 0.2820 | no |
| feat6 | 0.304 | feat6 | 0.2657 | yes |
| feat73 | 0.300 | feat28 | 0.2609 | no |
| feat2 | 0.293 | feat2 | 0.2457 | yes |
| feat7 | 0.252 | feat59 | 0.2366 | no |
| feat44 | 0.252 | feat12 | 0.2352 | no |
| feat54 | 0.242 | feat32 | 0.2050 | no |
| feat62 | 0.228 | feat72 | 0.2006 | no |
| feat28 | 0.224 | feat3 | 0.1959 | no |
| feat12 | 0.218 | feat62 | 0.1874 | no |
| feat58 | 0.198 | feat7 | 0.1775 | no |
| feat45 | 0.179 | feat37 | 0.1750 | no |
| feat21 | 0.167 | feat46 | 0.1552 | no |
| feat78 | 0.163 | feat21 | 0.1497 | no |
| feat32 | 0.156 | feat61 | 0.1488 | no |
| feat39 | 0.140 | feat45 | 0.1316 | no |
| feat37 | 0.124 | feat58 | 0.1313 | no |
| feat72 | 0.117 | feat78 | 0.1284 | no |
| feat42 | 0.114 | feat39 | 0.1181 | no |
| feat40 | 0.109 | feat44 | 0.1172 | no |
| feat61 | 0.102 | feat40 | 0.1128 | no |
| feat3 | 0.101 | feat54 | 0.1059 | no |
| feat60 | 0.094 | feat57 | 0.1012 | no |
| feat57 | 0.090 | feat56 | 0.0914 | no |
| feat56 | 0.073 | feat42 | 0.0780 | no |
| feat46 | 0.069 | feat38 | 0.0681 | no |
| feat38 | 0.045 | feat17 | 0.0557 | no |
| feat41 | 0.021 | feat60 | 0.0418 | no |
| feat17 | 0.021 | feat41 | 0.0247 | no |
| feat30 | 0.015 | feat76 | 0.0161 | no |
| feat69 | 0.014 | feat36 | 0.0146 | no |
| feat76 | 0.012 | feat30 | 0 | no |
| feat35 | 0 | feat35 | 0 | yes |
| feat36 | 0 | feat43 | 0 | no |
| feat43 | 0 | feat52 | 0 | no |
| feat52 | 0 | feat69 | 0 | no |