Marcelo Jimenez

Report – C3T2

Objectives

One of the objectives of the survey was to find out which of two brands of computers our customers prefer. Two different decision tree classification methods were made to predict the preferred brand Stochastic Gradient Boosting (GBM)and Random Forest (RF). Both methods used a 10-fold cross-validation and were repeated 3 times with 3 types of metrics, Kappa, accuracy, and ROC. These will help me determine which algorithm worked best for this classification.

GBM Results

Since I specified a two Class Summary (A specialized function for 2 class data to measure performance) in the control parameters, the command returns the area under the ROC curve. ROC takes into account the Rate of True Positives and the Rate of False Positives, an ROC of 1.0 means 100% accurate predictions.

Changing the interaction depth allows for greater accuracy in the case of GBM. Also, by running GBM_Fit3 I get the results from the other two fits as well.

Accuracy is the percentage of correctly classifies instances out of all instances, Kappa is similar however it is normalized at the baseline of random chance (useful when the classes have an imbalanced split like 70-30 split).

Random Forest Results

Using this Random Forest command, it yields ROC numbers which don't change between iterations due to the nature of Random Forest. Furthermore, between the three iterations, feature importance scores dont seem to change that much.

Here are the accuracy and kappa results for both models:

Model	Iteration	Accuracy	Kappa
GBM	1	0.7255701	0.41986
GBM	2	0.8829461	0.756453
GBM	3	0.9007271	0.792767
RF	1	0.9157691	0.821467
RF	2	0.9124015	0.813718
RF	3	0.9136579	0.816536

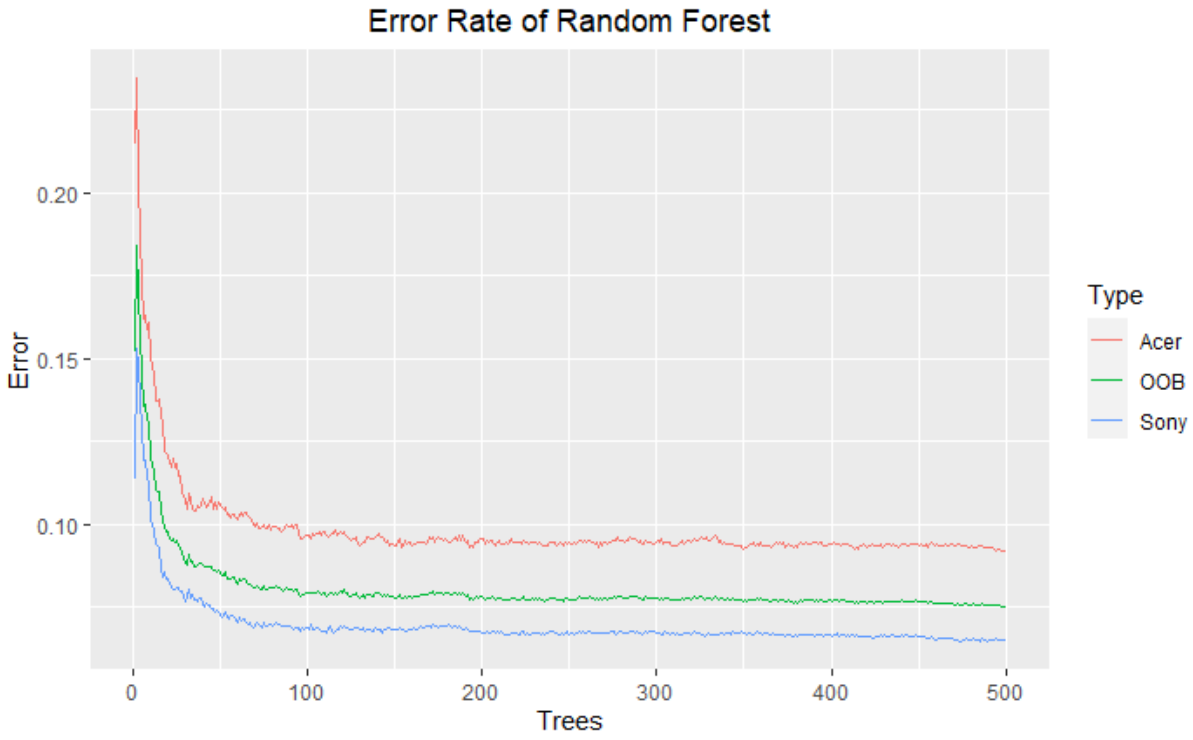
GGplot and Random Forest

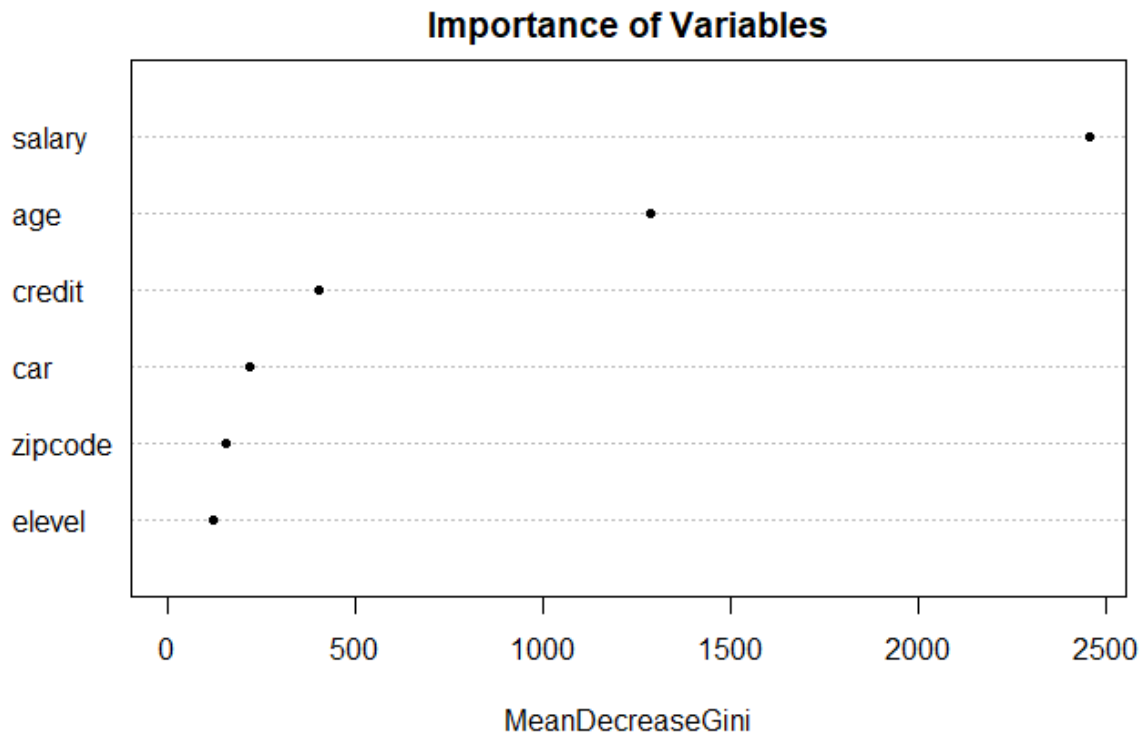
The ggplot is based on err.rate matrix, a matrix calculated when constructing the model using randomForest(). It contains columns for the OOB (out of bag) error rate, Acer error rate, Sony error rate (how frequent those two get missed classified). Each row of the matrix represents the error rate after certain iterations of the random forest, so first row is the error rates after making the first tree, the 50th row shows the error rates after making the 50th tree and so on.

oob.error.data is created to transform the data into something ggplot can understand and plot.

Then using ggplot I can graph the matrix and evaluate whether the number of trees I selected are enough to stabilize the error rates.

Here are the results:

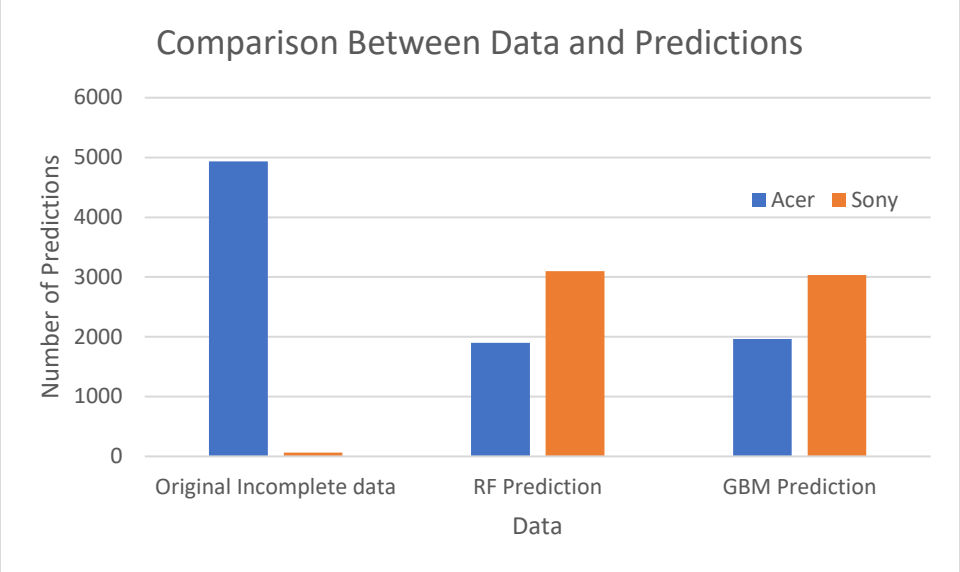





```
        OOB estimate of  error rate: 7.78%
Confusion matrix:
     Acer Sony class.error
Acer 3393  351   0.0937500
Sony  419 5735   0.0680858
```

The number of trees need to stabilize the error rate is around 200 trees and it gives the most importance to the salary and age features out of them all. Furthermore, after comparing the training data with the test data we see the random forest model has an error rate of around 7.8%, making it a very reliable model.

Lastly, this simple chart compares the incomplete data with both the model predictions and concludes that Sony is the preferred brand between Blackwell customers.


Comparison Between Data and Predictions
6000
5000
4000
3000
2000
1000
0
Number of Predictions
Acer
Sony
Original Incomplete data
RF Prediction
GBM Prediction
Data