STA141A Group 15 Final Project

Collin Chee, Love Chien, Sharon Wong, Yixuan Deng

12/14/2020

Background

We reviewed the research paper Using Data Mining for Bank Direct Marketing: An Application of the CRISP-DM Methodology from which the bank dataset came from. The data mining strategies used in the paper to analyze the dataset are Naïve Bayes, Decision Trees and Support Vector Machines (SVM). Due to the large size of the dataset, the number of variables were reduced using the rattle tool, and missing values were omitted. In this paper, the researchers discovered that SVM produced the best predictive results with a high value of 0.9 of area under its ROC curve.

Citation: S. Moro, R. Laureano and P. Cortez. Using Data Mining for Bank Direct Marketing: An Application of the CRISP-DM Methodology.

Statistical Question of Interest

We will be analyzing the same bank dataset mentioned above to answer what factors are the most important towards whether a person will sign a long term deposit. We will also analyze what model will best predict the final person's outcome. Specifically, we want to know about the people of Portugal.

Two answer these two questions, we will be using Logistic Regression, RandomForest, and AdaBoost to model how likely it is a for a person to sign a long term deposit. The reason these three methods were chosen was because each model can work with contiuous and categorical variables. In this data set, the variable "y" is the outcome with the outcomes "yes" or "no". The Logistic Regression method was chosen due to its flexibility and its ability to measure the direction and significance of predictors. The RandomForest method should also be a good fit because it is collection of many decision trees that can take multiple nodes to come to a final solution. The final method, Adaboost(Extra Credit), is a slight alternative to decision trees and is a boosting that can weigh many small trees to compute a final outcome.

Analysis Plan

We will be using a 70:30 training/test split to build our models. We will be conducting Logistic and RF on the bank-full and the smaller bank dataset and compare their results. However, we will only conduct the Adaboost on the bank data set due to the large amount of time it takes to train on the entire dataset. At the end, we will compare all three methods by providing classification accuracy results and their confusion matrices.

Population of Interest

The population of interest is the entire adult residents of Portugal who have a bank account. This study samples around 45 thousand people who were called by phone and their personal information have also been collected. The sample was also collected from 2008-2010. Although the number of observations is small compared to the entire population which is probably in the millions, 45 thousand is a very large number of observations and should be sufficient enough to represent the entire population.

Descriptive Analysis Plan

We will ook at three predictors we think will be important. The first predictor is the age because it is important to see what the average and median age was when it comes to making financial decisions. We will also be looking at people’s balance and duration.

Inferential Analysis Plan

For Logistic Regression, we will first build a model using the glm function. Then, we will provide a summary of all the predictors and point out which predictors seems to contribute the most towards the final decision. Then, we will provide a confusion matrix and the accuracy of the model

Similarly, we will build a RandomForest model using the rf function. Instead of a summary box, we can use the importance plot to analyze the gini index. This will work analogously like the p values from the logistic plot as we then can see what predictors are the most “important”.

Extra Credit

We will finally use the adaboost method from the fastAdabag package to analyze the smaller bank.csv dataset. As mentioned before, the smaller dataset is used as it takes to long to run on the full dataset. A confusion matrix, sample classifier (separate file only), and accuracy will be displayed.

Results

Descriptive Analysis

```r
bank_full<-read.table("bank-full.csv",head=TRUE,sep = ';')
```

```r
bank_sum3
```

	bank_full[, c(“duration”, “age”, “balance”)] (N = 45,211)
minimum	0
median (IQR)	180 (103.00, 319.00)
mean (sd)	258.16 ± 257.53
maximum	4,918
minimum	18
median (IQR)	39 (33.00, 48.00)
mean (sd)	40.94 ± 10.62
maximum	95
minimum	-8,019
median (IQR)	448 (72.00, 1,428.00)
mean (sd)	1,362.27 ± 3,044.77
maximum	102,127

To read the table, duration's statistics come first, followed by age, and then finally balance.

We notice that the last contact duration amount in seconds is around 260 seconds, and the median was 3 minutes. The average age was around 40 years old with the median being very close at 39 years of age. The average yearly balance was 1362 euros and the median being 448. The negative amount of euros could signifify that people are in debt and owe money to the bank.

Inferential Analysis

Logistic Regression

We choose this method because it is a type of supervised learning which fits this data set. In the bank-full and bank data set, we have our response outcome, either "yes" the client has subscribed a term deposit or "no" the client has not subscribed a term deposit. The logistic regression allows us to use create a model based on our training data, which we can then apply to our test data to create predictions. The logistic regression is great because it provides a measure of how appropriate a predictor is (coefficient size) and also its direction of association (positive or negative). Therefore, with logistic regression, we know what predictor variables are the most important in our model and is driving the predictions the most.

```
bank_full$y<-as.factor(bank_full$y) #set up train and test set
```

```
train = bank_full %>%
  sample_frac(0.7)

test = bank_full %>%
  setdiff(train)
```

```
logit_mod <- glm(y ~., data = train, family = 'binomial') #logistic model

summary(logit_mod)
```

```
##
## Call:
## glm(formula = y ~ ., family = "binomial", data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -5.6962  -0.3751  -0.2541  -0.1519   3.2001
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)         -2.512e+00  2.179e-01 -11.525  < 2e-16 ***
## age                 -8.768e-04  2.628e-03  -0.334  0.73864
## jobblue-collar      -3.907e-01  8.743e-02  -4.468 7.89e-06 ***
## jobentrepreneur     -2.592e-01  1.457e-01  -1.779  0.07529 .
## jobhousemaid        -3.981e-01  1.579e-01  -2.522  0.01167 *
## jobmanagement       -1.586e-01  8.775e-02  -1.807  0.07071 .
## jobretired           1.868e-01  1.155e-01   1.617  0.10579
## jobself-employed    -2.607e-01  1.316e-01  -1.981  0.04759 *
## jobservices         -1.885e-01  9.914e-02  -1.901  0.05726 .
## jobstudent           4.273e-01  1.305e-01   3.275  0.00106 **
## jobtechnician       -1.738e-01  8.216e-02  -2.116  0.03437 *
## jobunemployed      -2.579e-01  1.375e-01  -1.875  0.06076 .
## jobunknown         -4.778e-01  2.896e-01  -1.650  0.09898 .
## maritalmarried     -2.233e-01  6.928e-02  -3.224  0.00126 **
## maritalsingle       1.717e-02  7.941e-02   0.216  0.82878
## educationsecondary  1.833e-01  7.793e-02   2.352  0.01865 *
## educationtertiary   3.715e-01  9.044e-02   4.108 3.99e-05 ***
## educationunknown    1.696e-01  1.254e-01   1.353  0.17612
## defaultyes         -7.727e-03  1.908e-01  -0.040  0.96770
## balance             1.148e-05  6.631e-06   1.731  0.08348 .
## housingyes         -6.987e-01  5.227e-02 -13.368  < 2e-16 ***
## loanyes            -5.010e-01  7.238e-02  -6.921 4.47e-12 ***
## contacttelephone   -1.049e-01  8.916e-02  -1.176  0.23949
## contactunknown     -1.578e+00  8.648e-02 -18.246  < 2e-16 ***
## day                 1.191e-02  2.968e-03   4.013 5.99e-05 ***
## monthaug           -7.161e-01  9.469e-02  -7.562 3.97e-14 ***
## monthdec            4.637e-01  2.161e-01   2.146  0.03187 *
## monthfeb           -3.481e-02  1.075e-01  -0.324  0.74596
## monthjan           -1.229e+00  1.475e-01  -8.333  < 2e-16 ***
## monthjul           -7.362e-01  9.135e-02  -8.059 7.69e-16 ***
## monthjun            5.152e-01  1.117e-01   4.615 3.94e-06 ***
## monthmar            1.569e+00  1.431e-01  10.970  < 2e-16 ***
## monthmay           -3.389e-01  8.632e-02  -3.927 8.61e-05 ***
## monthnov           -8.533e-01  1.014e-01  -8.414  < 2e-16 ***
## monthoct            9.077e-01  1.262e-01   7.194 6.27e-13 ***
## monthsep            9.487e-01  1.421e-01   6.678 2.43e-11 ***
## duration            4.153e-03  7.651e-05  54.284  < 2e-16 ***
## campaign           -9.050e-02  1.196e-02  -7.567 3.82e-14 ***
## pdays               2.502e-04  3.579e-04   0.699  0.48455
## previous            6.616e-03  6.411e-03   1.032  0.30215
## poutcomeother       1.057e-01  1.085e-01   0.973  0.33033
## poutcomesuccess     2.300e+00  9.837e-02  23.379  < 2e-16 ***
## poutcomeunknown    -5.939e-02  1.097e-01  -0.541  0.58839
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 22889  on 31647  degrees of freedom
## Residual deviance: 15175  on 31605  degrees of freedom
## AIC: 15261
##
## Number of Fisher Scoring iterations: 6
```

From this summary, we can see that the most important feature that will determine whether the client will or will not subscribe a term deposit is the outcome of the previous marketing campaign, specifically if the previous marketing campaign outcome was categorized as “success”, which has a coefficient of 2.3, then the client will likely subscribe a term deposit. Other important features that have a large influence on the outcome is last contact month, specifically if the last contact was in March, which has a coefficient of 2.154. Other important months are January, June, October, and September with a coefficient of -1.09, 1.11, 1.75 and 1.24 respectively. In addition, if the client is unemployed, which has a coefficient of -1.2 or the type of contact communication is unknown, which has a coefficient of -1.512, the client is likely to not subscribe a3 term deposit.

```
logit.conf=confusionMatrix(predictions$g, test$y) #confusion matrix
logit.conf
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    no   yes
##        no  11690  1035
##        yes   298   540
##
##                Accuracy : 0.9017
##                  95% CI : (0.8966, 0.9067)
##     No Information Rate : 0.8839
##     P-Value [Acc > NIR] : 1.756e-11
##
##                   Kappa : 0.3991
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9751
##             Specificity : 0.3429
##          Pos Pred Value : 0.9187
##          Neg Pred Value : 0.6444
##              Prevalence : 0.8839
##          Detection Rate : 0.8619
##    Detection Prevalence : 0.9382
##       Balanced Accuracy : 0.6590
##
##        'Positive' Class : no
##
```

```
(logit.conf$table[1,1] + logit.conf$table[2,2])/sum(logit.conf$table) #classification accuracy
```

```
## [1] 0.9017179
```

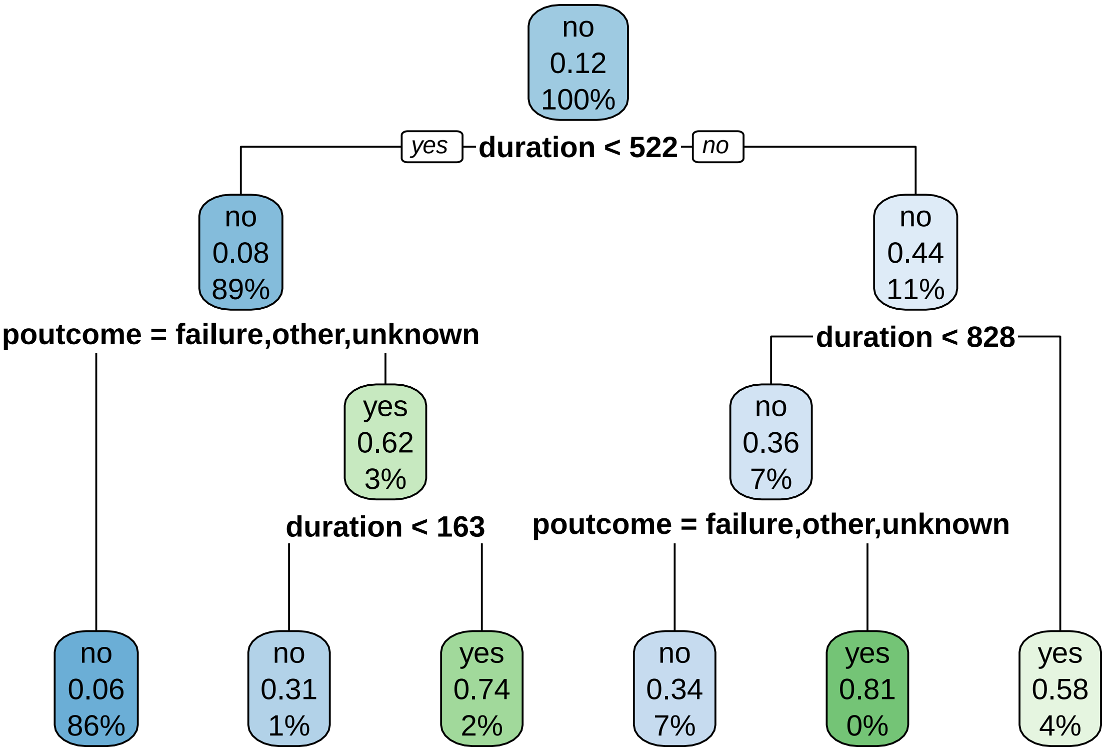
From our confusion matrix, we can see that our model has an 90% accuracy rate of predicting whether the client will say "yes" or "no" to subscribing to a term desposit, which means that our model is pretty good.

Decision Tree analysis for bank-full dataset

Here is an example of a single classification tree using the entire dataset.

Classification tree

```
# classification tree
ct<-rpart(y~.,data=bank_full)
# plot
rpart.plot(ct)
```



Confusion matrix:

```r
# prediction
pred<-predict(ct,bank_full,type='class')
# confusion matrix
confuse<-table(bank_full$y,pred)
confuse
```

```
##      pred
##          no   yes
##   no  38904  1018
##   yes  3444  1845
```

```r
#variance
var1<-var(predict(ct)[,1])
```

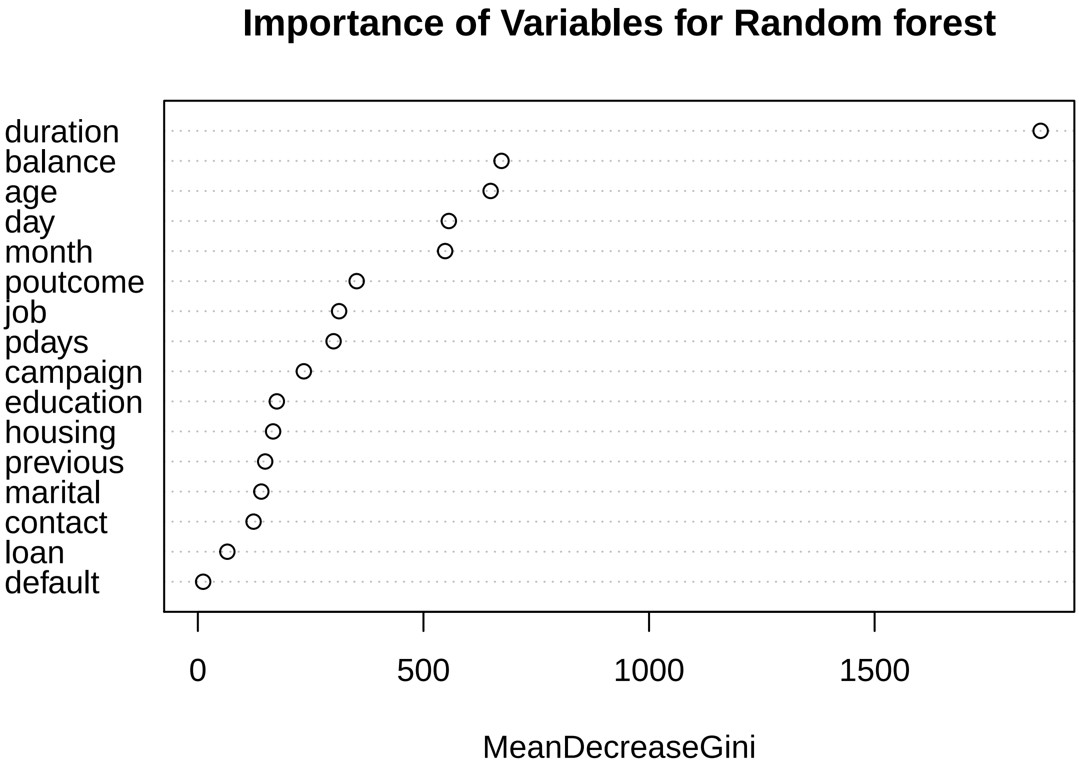
Accuracy is 90.13%.

Variance of this specific classification tree on the full dataset is 0.0245366

RandomForest

Now, we use RandomForest. This allows us to build multiple decision trees that have random number of predictors. Although the one classification tree above may show a good accuracy, it is very sensitive towards just the full dataset itself. RandomForest will allows many different decision trees that produce class predictions. Each tree will have an equal amount of say, and the class with the most votes will be the model's prediction.

```
# model building
train$y <- as.factor(train$y)
rf<-randomForest(y~.,data=train,ntree=100)
# importance plot
varImpPlot(rf,main='Importance of Variables for Random forest')
```



Confusion matrix:

```
pred<-predict(rf,test,type='class')
confuse<-table(test$y,pred)
confuse
```

```
##      pred
##          no   yes
##   no  11639   349
##   yes   905   670
```

```
# variance
var3<-var(predict(rf,type='prob')[,1])
```

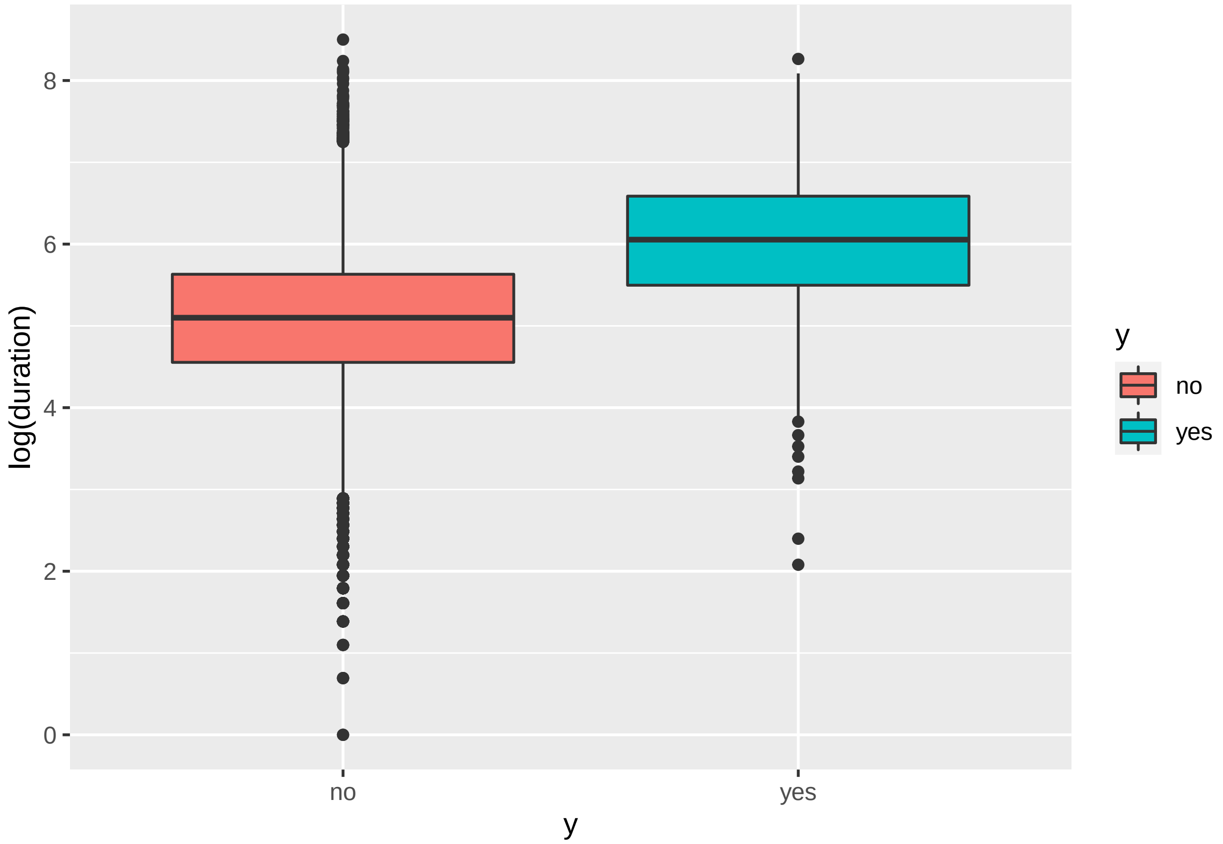
Accuracy is 90.75%.

According to the importance plot, the most powerful predictors are duration, balance, and age. Although the accuracy for a single classification tree had a similar accuracy, this does not mean that the single classification is better because the tree was a fit for the entire data set. This means that for future data, it may be too overfit to do well with future data. The random forest may not be completely accurate, but it should do a better job modeling any future data.

Comparing the RandomForest Model vs. the Logistic Model, the logistic model will be a little more flexible compared to the RandomForest as the logistic model computes the probability of an outcome being close to 1 (yes) or 0 (no). The RF model is more strict as the results are strictly yes or no. In this data, the logistic regression model edges out the RF model by one percent. Nevertheless, both models do a good job of predicting the correct outcome as most people do say no.

Which type of clients will sign a long-term deposit

```r
# data descriptive plot
bank_full%>%ggplot(aes(x=y,y=log(duration),fill=y))+geom_boxplot()
```



Data descriptive plot would also reveal that obviously difference of duration, the most important predictor, is really existed between `yes` and `no` for response variable. According to the boxplot, longer duration is, more probability the client would sign on to a long-term deposit.

Now we apply the logistic and randomForest method to the bank.csv dataset because we want to be able to compare our method with the Adaboost method (Extra Credit). We are using a smaller dataset because it takes too long to run the Adaboost method on a very large dataset.

Analysis using the smaller bank.csv dataset (To compare with Adaboost later)

```
bank<-read.table("bank.csv",head=TRUE,sep = ';')
```

```
bank$y<-as.factor(bank$y) #training and testing set for smaller dataset
```

```
train_bank = bank %>%
  sample_frac(0.7)

test_bank = bank %>%
  setdiff(train_bank)
```

```
logit_mod_bank <- glm(y ~., data = train_bank, family = 'binomial')

summary(logit_mod_bank)
```

```
##
## Call:
## glm(formula = y ~ ., family = "binomial", data = train_bank)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.0108  -0.3860  -0.2556  -0.1461   3.1563
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)         -2.591e+00  7.273e-01  -3.563 0.000367 ***
## age                 -9.853e-03  8.597e-03  -1.146 0.251733
## jobblue-collar      -4.879e-01  2.859e-01  -1.706 0.087951 .
## jobentrepreneur     -3.643e-01  5.052e-01  -0.721 0.470913
## jobhousemaid        -5.900e-01  4.919e-01  -1.199 0.230359
## jobmanagement       -8.503e-03  2.824e-01  -0.030 0.975980
## jobretired           3.606e-01  3.807e-01   0.947 0.343544
## jobself-employed    -2.878e-01  4.366e-01  -0.659 0.509791
## jobservices         -1.636e-01  3.222e-01  -0.508 0.611732
## jobstudent           3.345e-01  4.351e-01   0.769 0.442065
## jobtechnician       -1.448e-01  2.671e-01  -0.542 0.587761
## jobunemployed       -9.254e-01  5.376e-01  -1.721 0.085211 .
## jobunknown           6.639e-01  7.478e-01   0.888 0.374710
## maritalmarried      -3.888e-01  2.080e-01  -1.869 0.061566 .
## maritalsingle       -2.133e-01  2.406e-01  -0.886 0.375404
## educationsecondary -1.565e-01  2.403e-01  -0.651 0.514762
## educationtertiary    5.650e-02  2.747e-01   0.206 0.837061
## educationunknown    -6.203e-01  4.366e-01  -1.421 0.155429
## defaultyes           7.822e-01  4.928e-01   1.587 0.112426
## balance             -4.948e-06  2.031e-05  -0.244 0.807519
## housingyes          -2.979e-01  1.649e-01  -1.807 0.070728 .
## loanyes             -5.681e-01  2.393e-01  -2.374 0.017588 *
## contacttelephone    -9.882e-02  2.810e-01  -0.352 0.725040
## contactunknown      -1.762e+00  2.753e-01  -6.399 1.56e-10 ***
## day                  2.252e-02  9.849e-03   2.287 0.022213 *
## monthaug           -4.052e-02  3.057e-01  -0.133 0.894562    
## monthdec            4.672e-01  7.995e-01   0.584 0.558991    
## monthfeb            3.771e-01  3.570e-01   1.056 0.290748    
## monthjan           -1.131e+00  4.777e-01  -2.368 0.017863 *  
## monthjul           -6.042e-01  3.122e-01  -1.935 0.052980 .  
## monthjun            1.074e+00  3.643e-01   2.947 0.003204 ** 
## monthmar            2.085e+00  4.728e-01   4.411 1.03e-05 ***
## monthmay           -2.027e-01  2.846e-01  -0.712 0.476364    
## monthnov           -6.221e-01  3.336e-01  -1.865 0.062217 .  
## monthoct            1.604e+00  3.997e-01   4.013 6.01e-05 ***
## monthsep            5.615e-01  5.566e-01   1.009 0.313064    
## duration            4.395e-03  2.467e-04  17.811  < 2e-16 ***
## campaign           -7.399e-02  3.287e-02  -2.251 0.024366 *  
## pdays               9.341e-04  1.151e-03   0.812 0.416847    
## previous           -1.801e-03  4.302e-02  -0.042 0.966597    
## poutcomeother       4.234e-01  3.122e-01   1.356 0.175035    
## poutcomesuccess     2.530e+00  3.433e-01   7.371 1.69e-13 ***
## poutcomeunknown     1.118e-01  3.791e-01   0.295 0.768028    
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 2287.3  on 3164  degrees of freedom
## Residual deviance: 1526.3  on 3122  degrees of freedom
## AIC: 1612.3
## 
## Number of Fisher Scoring iterations: 6
```

As we can see from the summary, the same predictors are the most significant.

```
logit.conf_bank=confusionMatrix(predictions_bank$gb, test_bank$y)
logit.conf_bank
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction   no  yes
##        no  1172   92
##        yes   34   58
##                                          
##                Accuracy : 0.9071         
##                  95% CI : (0.8904, 0.922)
##     No Information Rate : 0.8894         
##     P-Value [Acc > NIR] : 0.01909        
##                                          
##                   Kappa : 0.4315         
##                                          
##  Mcnemar’s Test P-Value : 3.815e-07      
##                                          
##             Sensitivity : 0.9718         
##             Specificity : 0.3867         
##          Pos Pred Value : 0.9272         
##          Neg Pred Value : 0.6304
##              Prevalence : 0.8894
##          Detection Rate : 0.8643
##    Detection Prevalence : 0.9322
##       Balanced Accuracy : 0.6792
##
##        'Positive' Class : no
##
```

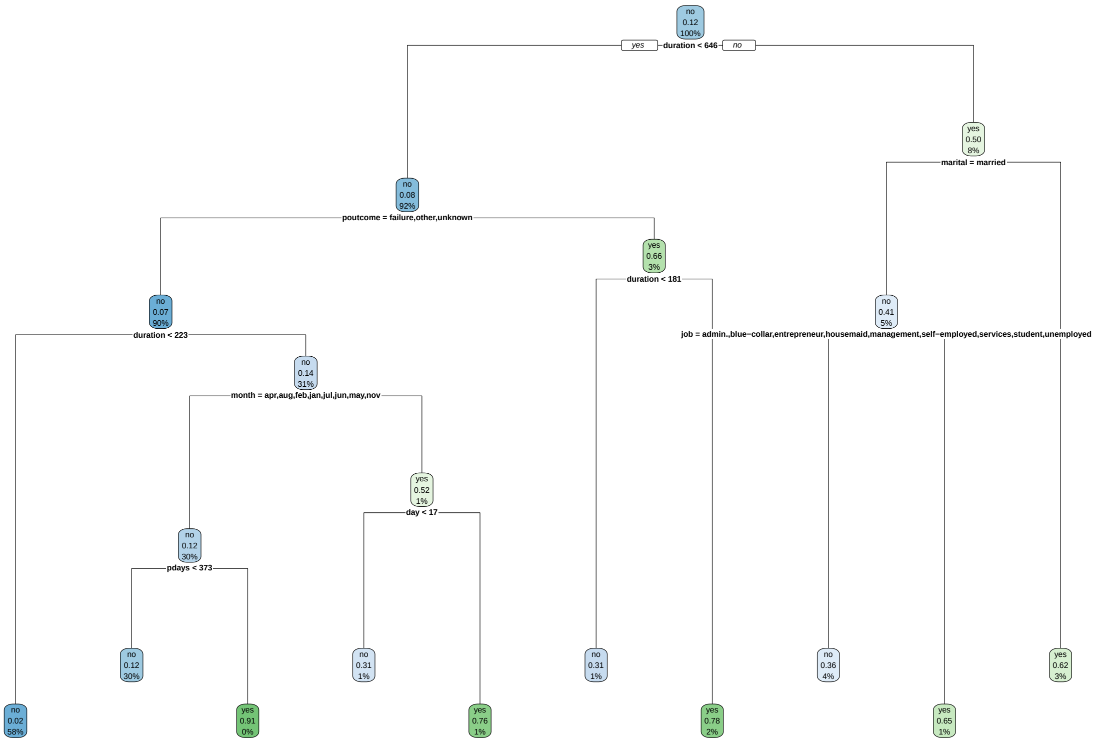
```
#confusion matrix and accuracy

(logit.conf_bank$table[1,1] + logit.conf_bank$table[2,2])/sum(logit.conf_bank$table)
```

```
## [1] 0.9070796
```

Classification tree

```
bank <- read.table("bank.csv", head = T, sep = ";")
library(rpart)
library(rpart.plot)
# classification tree
ct<-rpart(y~.,data=bank)
# plot
rpart.plot(ct)
```



Confusion matrix:

```
# prediction
pred<-predict(ct,bank,type='class')
# confusion matrix
confuse<-table(bank$y,pred)
confuse
```

```
##      pred
##         no  yes
##   no  3901   99
##   yes  303  218
```

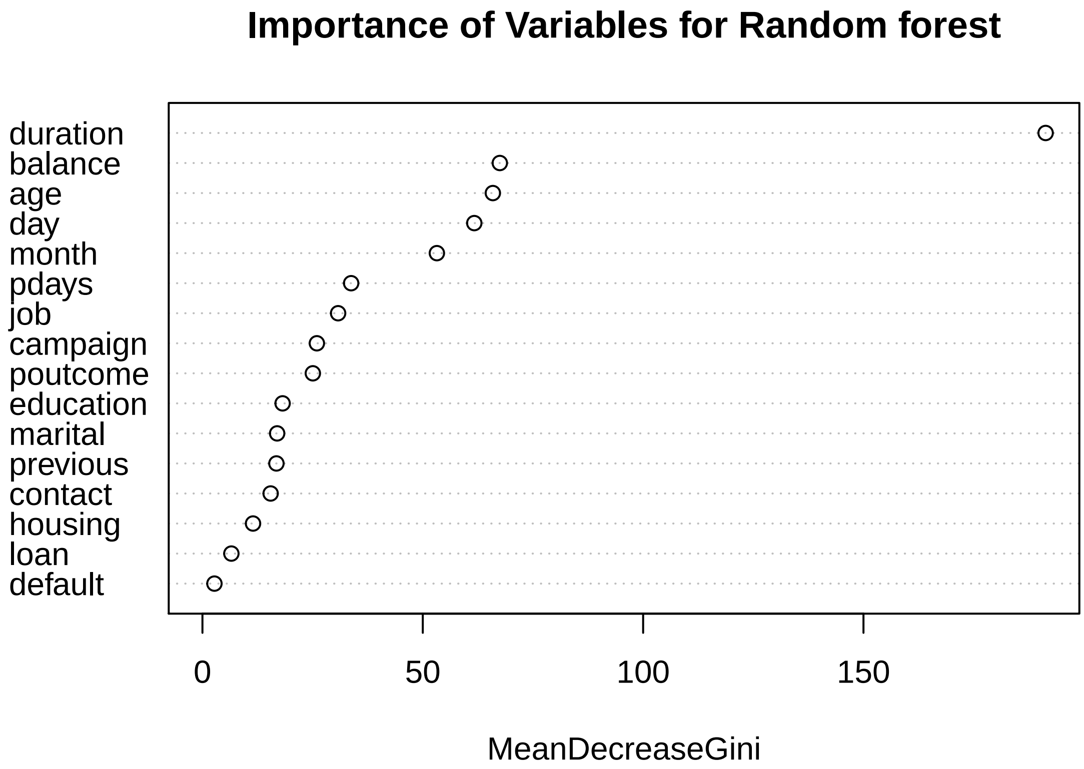
```
#variance
var1<-var(predict(ct)[,1])
```

Accuracy is 91.11%.

Variance of classification tree is 0.0310437

Random forest

```
# model building
train_bank$y <- as.factor(train_bank$y)
rf<-randomForest(y~.,data=train_bank,ntree=100)
# importance plot
varImpPlot(rf,main='Importance of Variables for Random forest')
```



Confusion matrix:

```
pred<-predict(rf,test_bank,type='class')
confuse<-table(test_bank$y,pred)
confuse
```

```
##      pred
##         no  yes
##   no  1168   38
##   yes  104   46
```

```
# variance
var3<-var(predict(rf,type='prob')[,1])
```

Accuracy is 89.53%.

Besides, variance of Bagging tree is also calculated as 0.0302178

According to the three tree based model, the most predictor is still duration.

AdaBoost (Extra Credit)

We also used the AdaBoost method to analyze the data AdaBoost works great with multiple parameters. This boosting method will make numerous one node trees, also known as “stumps” or weak classifiers, and weigh them differently to come to a conclusion of whether the outcome is yes or no. The key differeces between AdaBoost and RandomForest are that Adaboost only consists of small stumps and each stump's weights are different. In RandomForest, each tree is weighed the same.

Because there are so many paramters, and that each weak classifier depends on the previously bootstrapped data set, Adaboost will be able to create hundreds of different iterations of weak classifiers. Unfortunately, we could only use AdaBoost on the bank.csv file because it takes way too long to run on the full dataset.

```
test_adaboost <- adaboost(y~., data=train_bank,500) #use adaboost function from fastAdaboost package


pred <- predict(test_adaboost,newdata= test_bank)
```

According to sources online, an estimate of 100-1000 iterations is considered "safe". We can also fetch a single weak decision tree classifier by using the get_tree function. Here is the code for 7th weak tree classifier which is considered part of the strong classifier. The summary for tree will be shown in a separate document in the zip folder as it is very long.

```
tree <- get_tree(test_adaboost, 7)
```

```
confusion_matrix <- table(pred$class,test_bank$y)
confusion_matrix
```

```
##
##         no  yes
##   no  1172   98
##   yes   34   52
```

```
accuracy <- (confusion_matrix[1,1] + confusion_matrix[2,2])/ sum(confusion_matrix)
accuracy
```

```
## [1] 0.9026549
```

Comparison Between Each Method

RandomForest Accuracy - 89.53%

Logistic Regression Accuracy - 90.708%

Adaboost Accucracy - 90.2655%

Conclusion

All three methods show a very similar classification accuracy as all three accuracies hovered around 90%. When comparing Logistic and RandomForest methods wtih the full datasets, the classification accuracy for both methods was also very close to 90%. We were suprised by how well each method modeled the data.

Some key differences between the Logistic and the RandomForest Models is which variables have a more significant effect on the final outcome. For the Logistic Model, predictors such as campaign outcome, last contact month, and employment play a huge factor in the final predictions. In RandomForest, predictors such as last contact duration, age, and current balance contribute more towards a person's decision to saying yes or no. It is hard to analyze which predictor stands out the most in the AdaBoost method as each iteration will start with a random predictor, and the following predictors' weights will depend on the previous predictor's weights. However, in all three models, they have very strong classification accuracies and they all mainly predict that an average person would say "no" to signing long term deposit.