Predicting if a Bank Client Subscribes to a Deposit

12/14/2020

Team ID: Group 19

Meredith Harrison (Boosting, Literature Review, Writing Report, Review Report and Code)

Jasper Tsai (Logistic Regression and Lasso, Basic Random Forest, Boosting, Review Report and Code)

Ankita Bhat (Variable Seclection with Random Forest, Basic Random Forest, Review Report and Code)

Background

In this project, we analyze the a multivariate bank marketing dataset associated with phone-based marketing campaigns from a Portuguese banking institution. Phone-based marketing campaigns are important for generating business for financial institutions. Marketing campaigns are often tailored to certain clients, however sometimes they are broadly used to target new clients. Technology and statistical methods should be implemented to make marketing campaigns more successful by targeting clients that are more likely to subscribe (Moro et al., 2014).

The bank data set contains 20 attributes and a single binary response. The binary response (yes/no) refers to whether a client will subscribe to a term deposit. The primary interest of this project is to explore relationships between the 20 attributes and the response and identify a model to predict whether clients will subscribe to a term deposit using classification methodologies (Argesti, 2018). In addition, we will determine if social and economic attributes impact a client's decision to subscribe to a term deposit. Finally, we will analyze the impact of the customer-sales representative conversations during the marketing campaign on term deposits.

For our classification analysis, we have selected the following three methods: 1) random forest, 2) boosting, and 3) logistic regression. Random forest was selected for its ability to fit independent trees to nonlinear data (UCD, 2018). Breiman (2001) used random forest on multiple datasets including those with a multitude of categorical and numeric predictors both, similar to our dataset, and found random forest to be successful method. Boosting was selected as it is similar to random forest, but rather than fitting independent trees, boosting fits trees in sequential fashion, using information from the previously fit tree. However, with this methodology, overfitting is often a problem (UCD, 2018). To eliminate overfitting with boosting, we implemented cross-validation (CV) to determine the parameter values for n.trees and shrinkage. For our third method we choose logistic regression for its ability to build models with relatively high accuracy when the number of noise variables is less than the number of explanatory variables (Kirasich et al., 2018). In comparison, Kirasich et al. (2018) reported that when implementing random forest, the true false positive rate increases with an increasing number of explanatory variables. Considering, we don't know which variables are noise and explanatory, we thought the previous three methods would provide an interesting comparison of model performance across the different classification methods. Finally, to complement the main analysis we have conducted variable selection using a random forest. Geneur et al. (2010) used random forest to find important variables for interpretation and to design a parsimonious predictive model, which aligns with our goals of using random forest for variable selection.

Reserach Questions

1. Are all 20 attributes relevant to the predictive model for the client signing up for term deposit?
2. Are the social and economic context attributes relevant to subscribing to long term deposit?
3. Did the duration of a call from a sales representative during the bank's marketing campaign impact whether or not a consumer chose to become a long-term subscriber?

Answering these key questions will be of importance to banking institutions, as it will allow them to tailor their client recruiting efforts based on the results of this project. This will likely improve subscription rates and make the recruiting process more efficient as banks will be able to focus on the attributes that are known to have an effect term subscription. For a more cohesive report, additional literature review was done at the time each of the various methods were introduced and described.

Population and Study Design

The data set considers real campaign marketing data from phone calls of a Portuguese banking institution from May 2008 to November 2010; in total, 41,188 contacts were in the dataset (Moro et al., 2014). All calls were recorded making this representative of clients within the bank. However, the data set only contains a single Portuguese bank, so it's likely not representative of all bank campaigns (particularly those outside of Portugal). However, results of this analysis would be a good starting place for other institutions looking to implement such screening methods.

Analysis

Analysis Plan

Before beginning with our statistical analysis, we first inspected the data missing values, which were represented by "unknown" in the data set. The removal of all unknowns resulted in a total of 30,488 contacts to be used for analysis. Then, we will consider the descriptive statistics for all features in the data set (shown below). In the set-up using the str() function, we determined that the response (y) was binary,

and thus classification was the appropriate statistical method (Argesti, 2018). Duration was removed as it was noted to be extremely influential (for obvious reasons).

Descriptive Analysis

After the removal of missing values, we considered summary statistics for all the features in our dataset, except duration, which was removed in our data set-up. For numerical predictors (i.e., age, campaign, pdays, previous, emp.var.rate, cons.price.idx, cons.conf.idx, euribor3m, and nr.employed), the distribution was summarized in table format using minimum, maximum, median, mean, and SD. For categorical predictors (i.e., job, marital, education, default, housing, loan, contact, month, day_of_week, and poutcome) the n and the associated percent of the total entries (in parentheses) were shown.

```
##
##
## |                                        |Data (N = 30,488)       |
## |:---------------------------------------|:-----------------------|
## |**Client Age**                          |              |
## |   min                        |17                      |
## |   median                     |37                      |
## |   max                        |95                      |
## |   mean (sd)                  |39.03 &plusmn; 10.33    |
## |**Number of Campaign Contacts)**        |              |
## |   min                        |1                       |
## |   median                     |2                       |
## |   max                        |43                      |
## |   mean (sd)                  |2.52 &plusmn; 2.72      |
## |**Number of Days since Previous Campaign** |           |
## |   min                        |0                       |
## |   median                     |999                     |
## |   max                        |999                     |
## |   mean (sd)                  |956.33 &plusmn; 201.37  |
## |**Number of Previous Contacts**         |              |
## |   min                        |0                       |
## |   median                     |0                       |
## |   max                        |7                       |
## |   mean (sd)                  |0.19 &plusmn; 0.52      |
## |**Employment Variation Rate**           |              |
## |   min                        |-3.4                    |
## |   median                     |1.1                     |
## |   max                        |1.4                     |
## |   mean (sd)                  |-0.07 &plusmn; 1.61     |
## |**Consumer Price Index**                |              |
## |   min                        |92.201                  |
## |   median                     |93.444                  |
## |   max                        |94.767                  |
## |   mean (sd)                  |93.52 &plusmn; 0.59     |
## |**Consumer Confidence Index**           |              |
## |   min                        |-50.8                   |
## |   median                     |-41.8                   |
## |   max                        |-26.9                   |
## |   mean (sd)                  |-40.60 &plusmn; 4.79    |
## |**Daily Indicator**                     |              |
## |   min                        |0.634                   |
## |   median                     |4.856                   |
## |   max                        |5.045                   |
## |   mean (sd)                  |3.46 &plusmn; 1.78      |
## |**Number of Employees**                 |              |
## |   min                        |4963.6                  |
## |   median                     |5191                    |
## |   max                        |5228.1                  |
## |   mean (sd)                  |5,160.81 &plusmn; 75.16 |
## |**Job Type**                            |              |
## |   Administration             |8,737 (29)              |
## |   Blue-Collar                |5,675 (19)              |
## |   Entrepreneur               |1,089 (4)               |
## |   Housemaid                  |690 (2)                 |
## |   Management                 |2,311 (8)               |
## |   Retired                    |1,216 (4)               |
## |   Self-Employed              |1,092 (4)               |
## |   Services                   |2,857 (9)               |
## |   Student                    |610 (2)                 |
## |   Technician                 |5,473 (18)              |
## |   Unempolyed                 |738 (2)                 |
```

```
## |**Marital Status**                    |            |
## |   Divorced                 |3,553 (12)            |
## |   Married                  |17,492 (57)           |
## |   Single                   |9,443 (31)            |
## |**Education Level**                   |            |
## |   Basic 4 Year             |2,380 (8)             |
## |   Basic 6 Year             |1,389 (5)             |
## |   Basic 9 Year             |4,276 (14)            |
## |   High School Diploma      |7,699 (25)            |
## |   Illiterate               |11 (0)                |
## |   Professional Course      |4,321 (14)            |
## |   Univerisity Degree       |10,412 (34)           |
## |**Credit in Default**                 |            |
## |   No                       |30,485 (100)          |
## |   Yes                      |3 (0)                 |
## |**Housing Loan**                      |            |
## |   No                       |13,967 (46)           |
## |   Yes                      |16,521 (54)           |
## |**Contact Type**                      |            |
## |   Cellular                 |20,443 (67)           |
## |   Telephone                |10,045 (33)           |
## |**Month of Contact**                  |            |
## |   January                  |0 (0)                 |
## |   February                 |0 (0)                 |
## |   March                    |482 (2)               |
## |   April                    |2,115 (7)             |
## |   May                      |9,733 (32)            |
## |   June                     |3,614 (12)            |
## |   July                     |5,081 (17)            |
## |   August                   |4,673 (15)            |
## |   September                |495 (2)               |
## |   October                  |642 (2)               |
## |   November                 |3,496 (11)            |
## |   December                 |157 (1)               |
## |**Last contact Day of the Week**      |            |
## |   Monday                   |6,279 (21)            |
## |   Tuesday                  |5,955 (20)            |
## |   Wednesday                |6,125 (20)            |
## |   Thursday                 |6,395 (21)            |
## |   Friday                   |5,734 (19)            |
## |**Outcome of Last Campaign**          |            |
## |   Failure                  |3,461 (11)            |
## |   Non-Existent             |25,836 (85)           |
## |   Success                  |1,191 (4)             |
## |**Subscribed (y)**                    |            |
## |   Yes                      |3,859 (13)            |
## |   No                       |26,629 (87)           |
```

Classification Analysis

We used three methods for classification: random forest, logistic regression, and boosting. For extra credit, we implemented variable selection by random forest. A more detailed explanation of each of the analyses is shown in the subheadings below. Before beginning, we fit the dataset (minus duration) with the all unknown values removed into a training and test data set, where 70% of the data set was allocated to training.

Random Forest

A random forest was fit with the randomForest from the randomForest package using all of the predictors except duration. The randomForest model was fit using 500 trees. Following creation of the random forest, a confusion matrix and accuracy were calculated for comparison of results acorss analysis type. Then, out-of-bag error and test error were calculated. A randomForest model was fit to both the training and test data. Finally for ease of comparison across models, we calculated accuracy and area under the curve (AUC). All code and output is shown in the results section.

Boosting

A gradient boosting machine (i.e., boosting) was fit using the gbm function from the gbm package. Cross-validation was used to determine the correct parameter values for the number of trees (i.e, n.trees) and shrinkage, which are arguments that are used in the gbm function. Specifically, k-fold cross validation (CV) was used where the number of folds was set to 5. We tested the following n.tree values: 1000, 2000, and 3,000 in increments , and we tested the following shrinkage values 0.01, 0.02, and 0.03.

Logistic Regression

Before beginning the analysis, balance was assessed using the table function. The data was determined to be unbalanced. A logistic model was fit with the glm function from the base R package using the specification “family = binomial.” Considering the number of predictors (i.e., 20), we believed multicollinearity might be an issue. Thus, we evaluated multicollinearity for all 20 predictors using a standardized GVIF, where GVIF values > 10 indicated multicollinearity (Fox and Monette, 1992). The feature with the greatest GVIF value above the determined threshold was dropped. Then, GVIF was re-calculated, and the feature with the next greatest GVIF was dropped. The process was completed when all the remaining features had GVIF below the threshold of 10. The remaining features were considered for the analysis.

All the predictors except those that were excluded after previously checking multicollinearity were included in the model. Are under the curve (AUC) curve was created to display the model sensitivity and specificity simultaneously. Next, with hopes of improving our model, we fit a LASSO to obtain a subset of predictors that minimized our prediction error. To set up the LASSO, a model matrix was created for training and test datasets and used for CV and identifying the best lambda. For CV comparison, we evaluated AUC. All code and output is shown in the results section.

Extra Credit

Variable Selection using Random Forest

The random forest function resulted in two subsets of variables, the first being important variables that include some redundancy, which is important for interpretation. The second subset is much smaller and corresponds to a model to avoid redundancy in future predictions. VSURF, an R package for variable selection using random forests, follows a two-step strategy that addressed both subsets by breaking them down into three processing steps. The first processing step is the thresholding step, which gets rid of the variables that had negative importance, where negative importance is defined as that removal of variables that would improve the performance of the model (Geneur, 2010). The second processing step is the interpretation step, which identifies important that are highly related to the response variable. Variables were ranked by importance and unimportant variables were eliminated. Finally, the third step and second of the two step strategy is the prediction step. The objective of this step is to find the variables that would reduce redundancy. The results and code are shown below.

Results

Random Forest

After running the random forest using the training dataset with 500 trees, 4 variables were tried at each split and our out-of-bag (OOB) error estimate was 5.81%.The OOB tested the optimal number of variables at each split. Based on “Random Forest Plot 1” shown below, we could have chosen a smaller number of trees for the random forest. The graph levels off at about 25 trees, so we could have chosen to run 100 trees, instead of 500 trees, which would greatly decrease the computation time without sacrificing model performance.

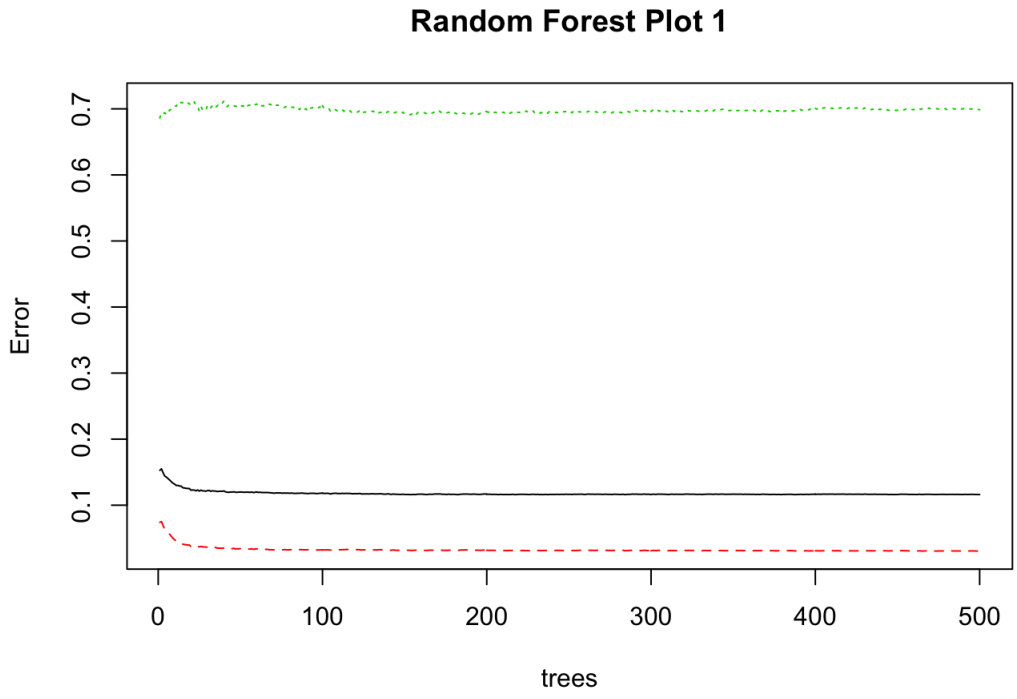
After testing mtry values using cross-validation on the training and test datasets, we created “Random Forest Plot 2,” depicting m on the x-axis and error on the y-axis, and both OOB and test error are shown on the plot. From “Random Forest Plot 2” we can conclude, the correct number of mtries is two since that has the lowest OOB error. Next model importance was shown in “Random Forest Plot 3”, and this showed slight differences in variable importance based on accuracy versus the Gini Index. In both graphs, the higher the value, the greater the importance, and the top three most important variables (i.e., age, job and euribor3m) were the same under both criteria. Finally, the AUC derived from this model was 0.799, and the curve is shown below in “Random Forest AUC.”

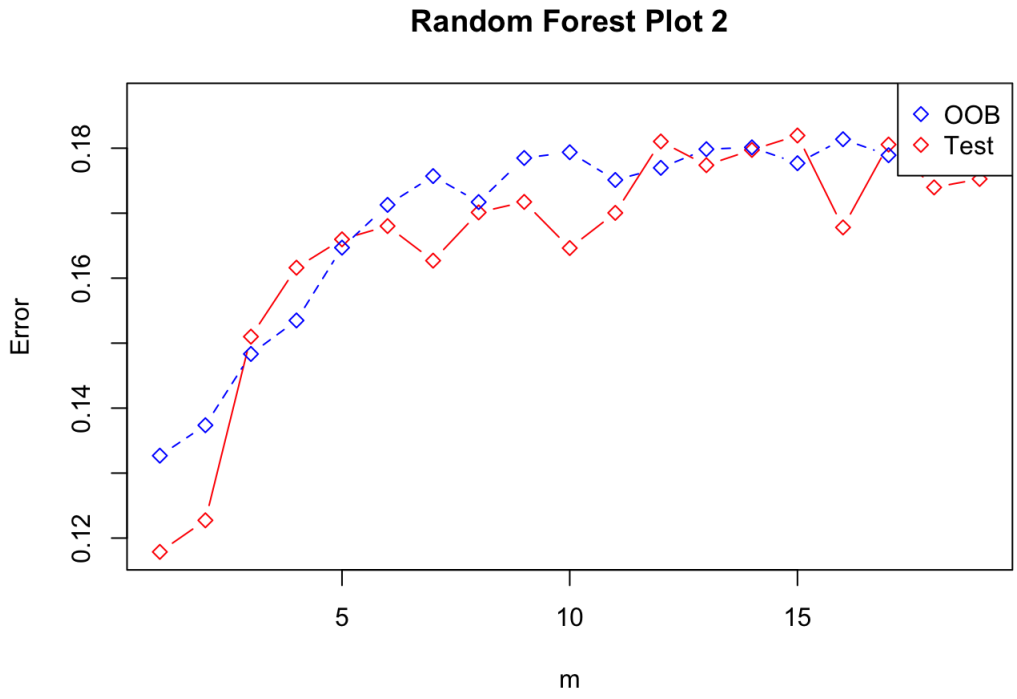
Next, we re-ran the random forest using the variables that were selected in variable selection using VSURF (i.e., pdays, euribor3m, and poutcome) and the optimal mtry (i.e., 2) that was found using CV in the basic random forest model. A confusion matrix, accuracy, and AUC were all calculated. Area under the curve was calculated before and after the prediction step. The AUC from prior to the prediction step was 0.799. The AUC associated with the final prediction step was 0.724. When we compare “Best Random Forest Plot 1” to “Best Random Forest Plot 3,” we notice we could have used a much fewer number of trees for the final prediction step. This occurred because we only had 3 predictors, and thus the trees were quite simplistic reducing the number required. It is interesting that when ee compare the variable importance plot before and after the prediction step, that the top three variables change. This occurs do to reasons described under the results in the extra credit section. Ultimately, the model selected by VSURF resulted in a lower AUC, but this may have occurred due to the model being too parsimonious.

```
##
## Call:
##  randomForest(formula = y ~ ., data = Data, importance = T, subset = train)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 4
##
##         OOB estimate of  error rate: 11.62%
## Confusion matrix:
##        no yes class.error
## no  18039 572  0.03073451
## yes  1907 823  0.69853480
```

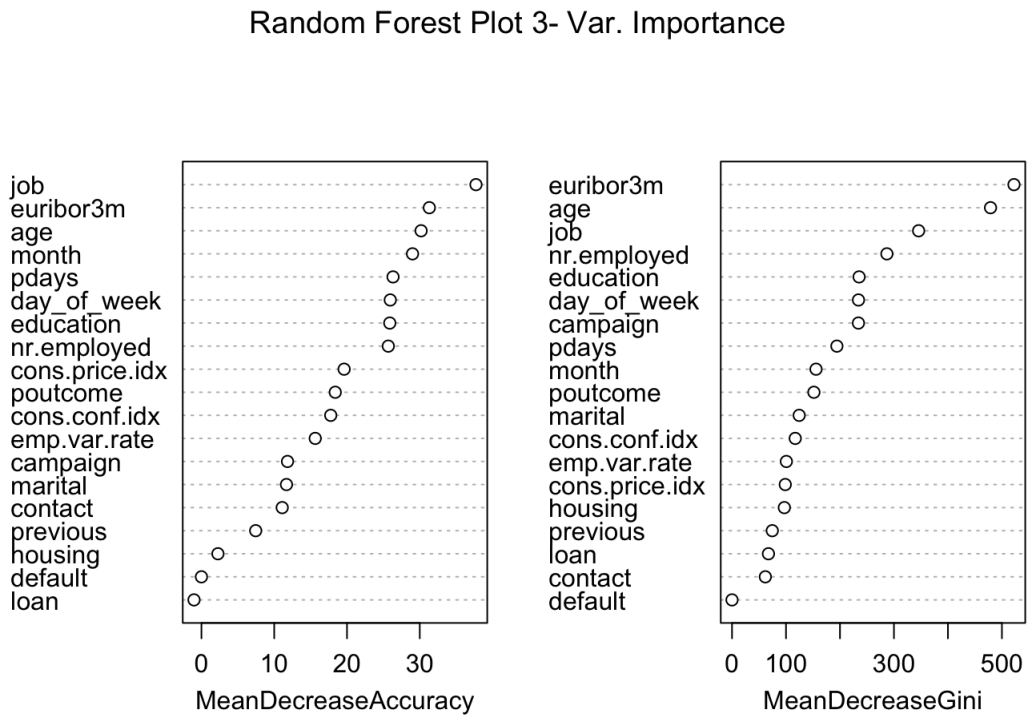
```
##        no yes class.error
## no  18039 572  0.03073451
## yes  1907 823  0.69853480
```

```
## [1] 0.8838084
```

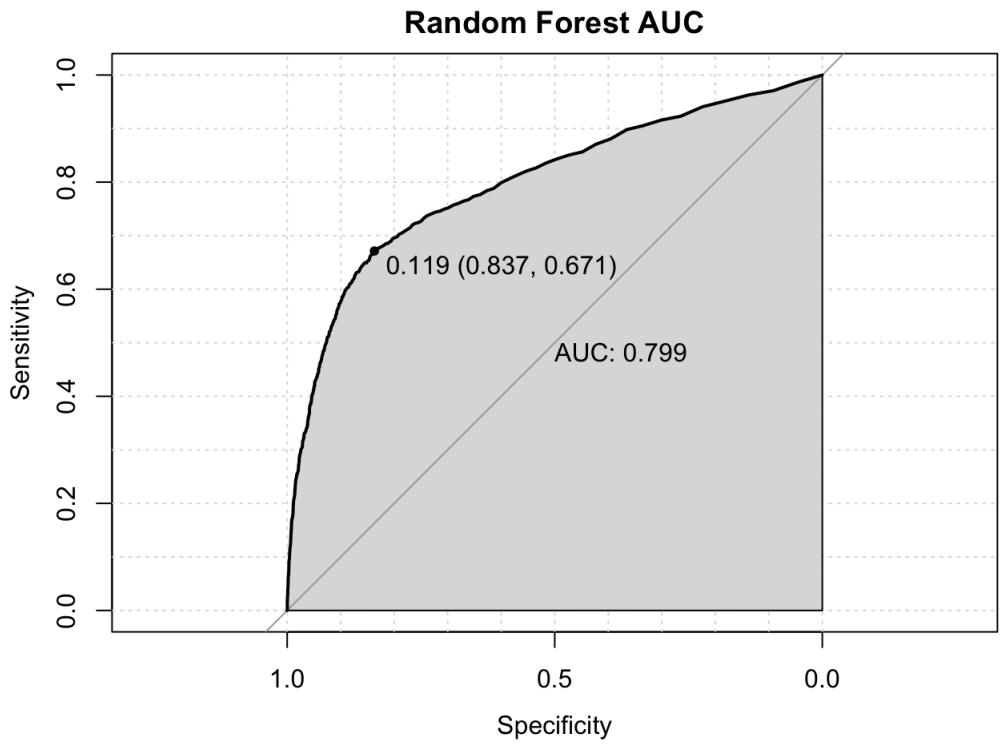




```
##        test.err   oob.err
##  [1,] 0.1178741 0.1326816
##  [2,] 0.1227340 0.1373647
##  [3,] 0.1509988 0.1483333
##  [4,] 0.1616101 0.1534896
##  [5,] 0.1659726 0.1646924
##  [6,] 0.1680144 0.1712763
##  [7,] 0.1626949 0.1757053
##  [8,] 0.1701170 0.1717082
##  [9,] 0.1717321 0.1784888
## [10,] 0.1646287 0.1793767
## [11,] 0.1700454 0.1751082
## [12,] 0.1810319 0.1769766
## [13,] 0.1773868 0.1798452
## [14,] 0.1797317 0.1801266
## [15,] 0.1819531 0.1776870
## [16,] 0.1678052 0.1813788
## [17,] 0.1805597 0.1789191
## [18,] 0.1739652 0.1872146
## [19,] 0.1752608 0.1797481
```




```
##       no   yes
## 13 0.994 0.006
## 14 0.990 0.010
## 15 0.988 0.012
## 17 0.988 0.012
## 19 0.992 0.008
## 39 1.000 0.000
```

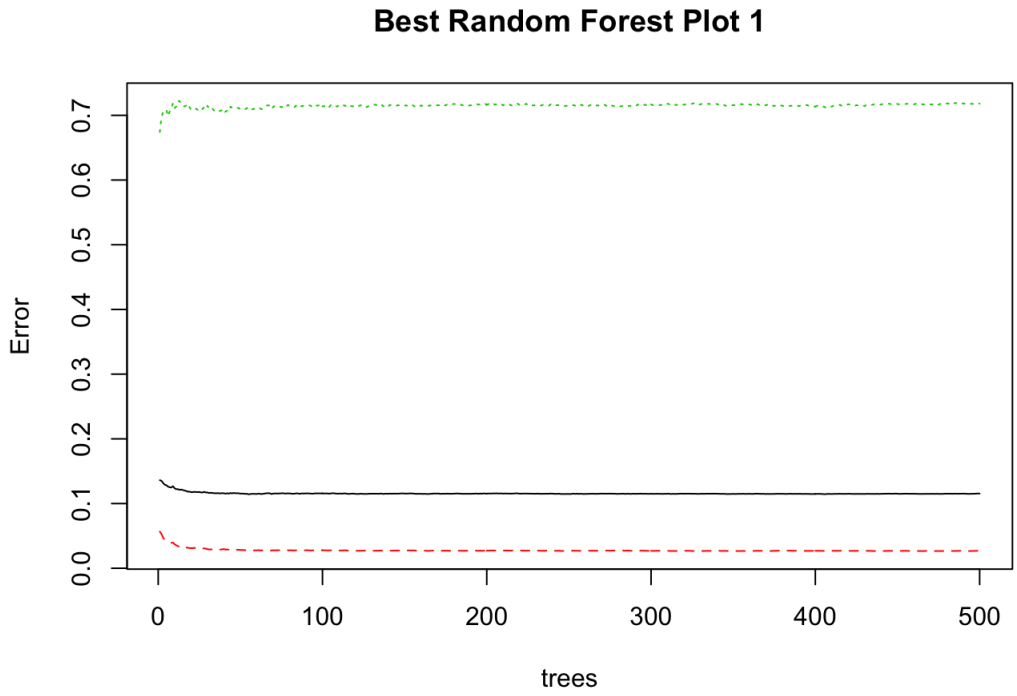


Random Forest using VSURF (extra credit) Results

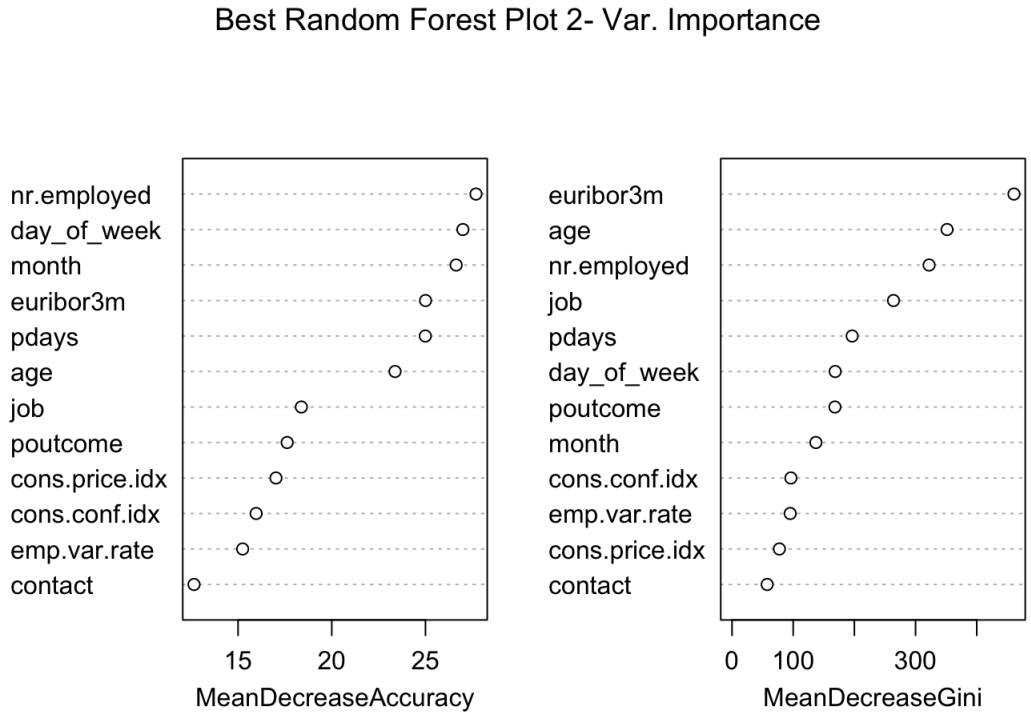
```
##
## Call:
##  randomForest(formula = y ~ ., data = Data2, importance = T, subset = train)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 11.55%
## Confusion matrix:
##        no yes class.error
## no  18107 504  0.02708076
## yes  1960 770  0.71794872
```

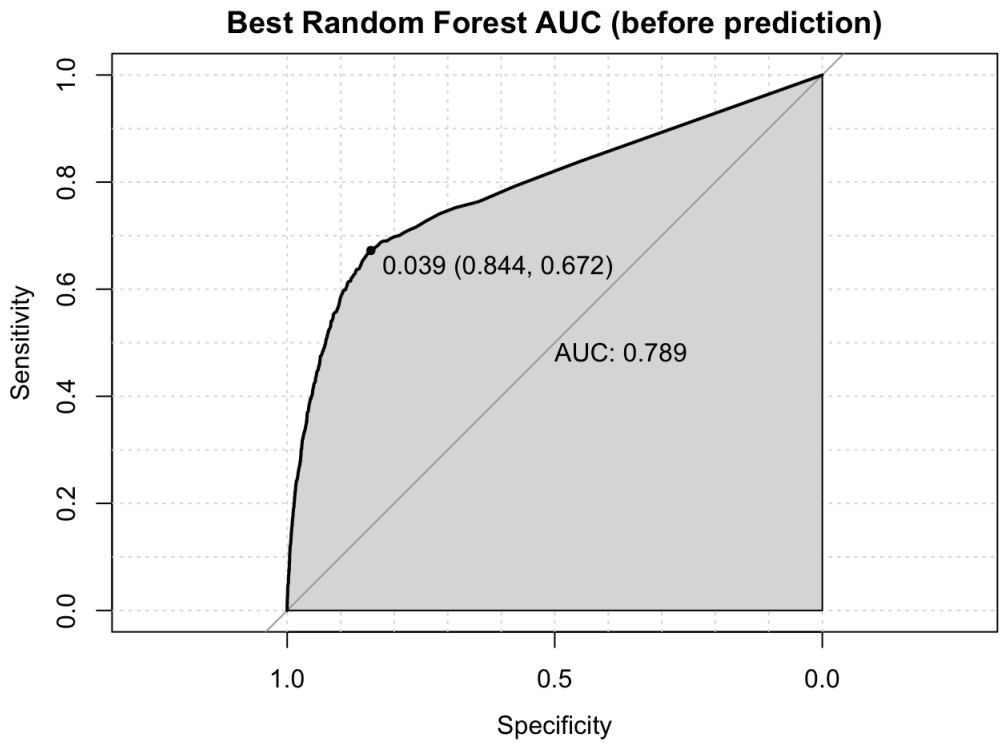
```
##        no yes class.error
## no  18107 504  0.02708076
## yes  1960 770  0.71794872
```

```
## [1] 0.8845106
```



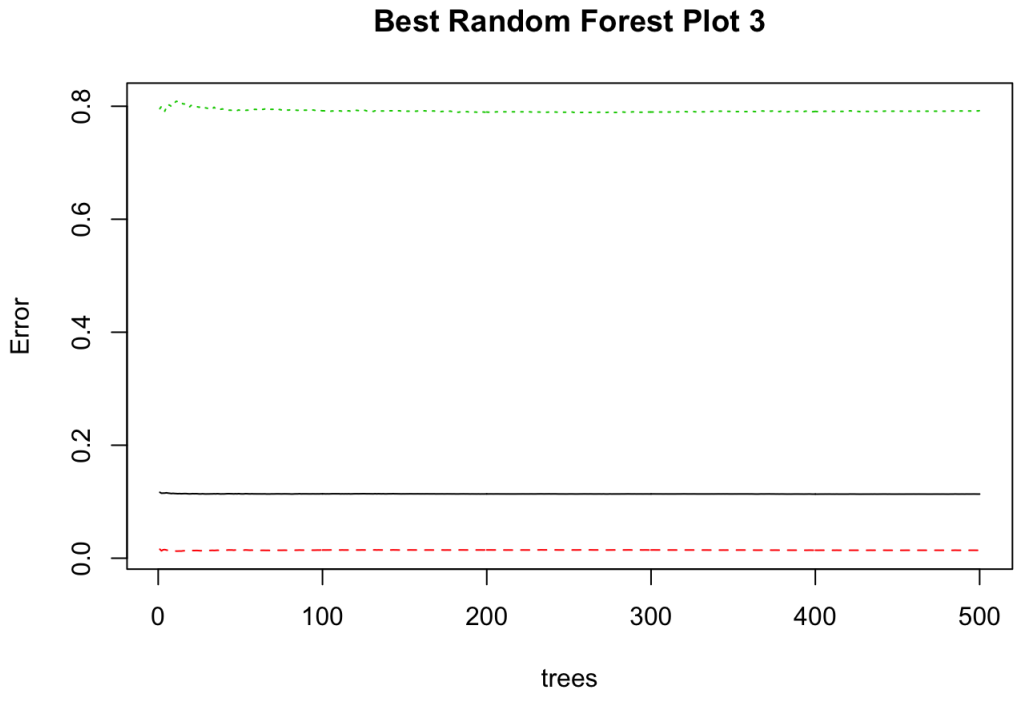

```
##       no   yes
## 13 1.000 0.000
## 14 0.998 0.002
## 15 1.000 0.000
## 17 1.000 0.000
## 19 1.000 0.000
## 39 1.000 0.000
```



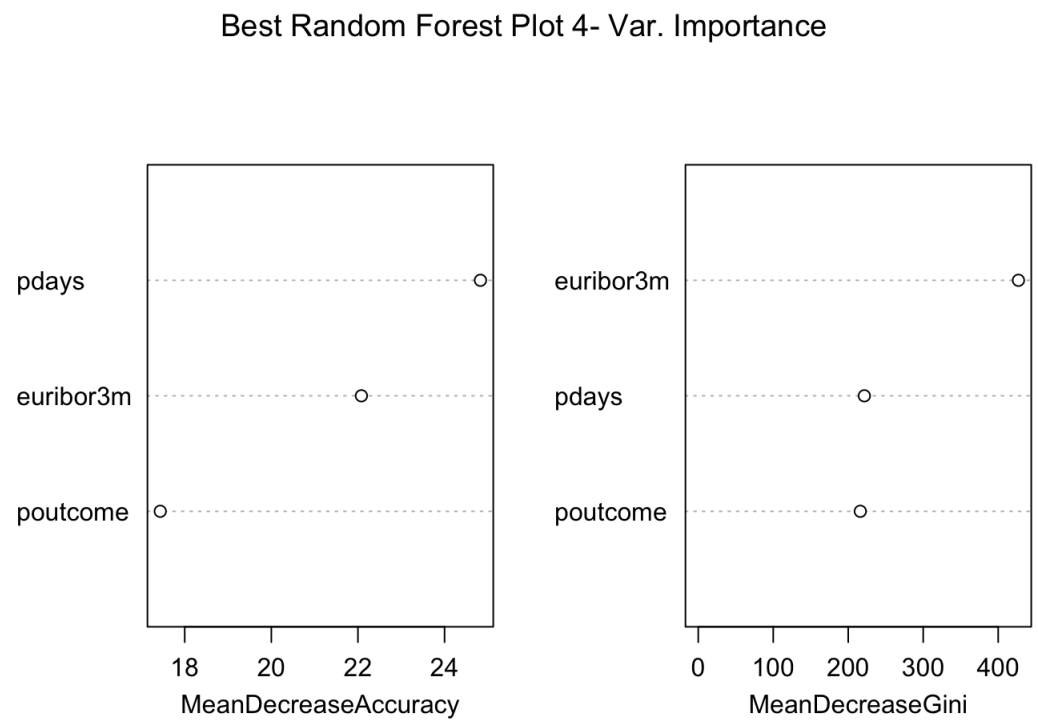


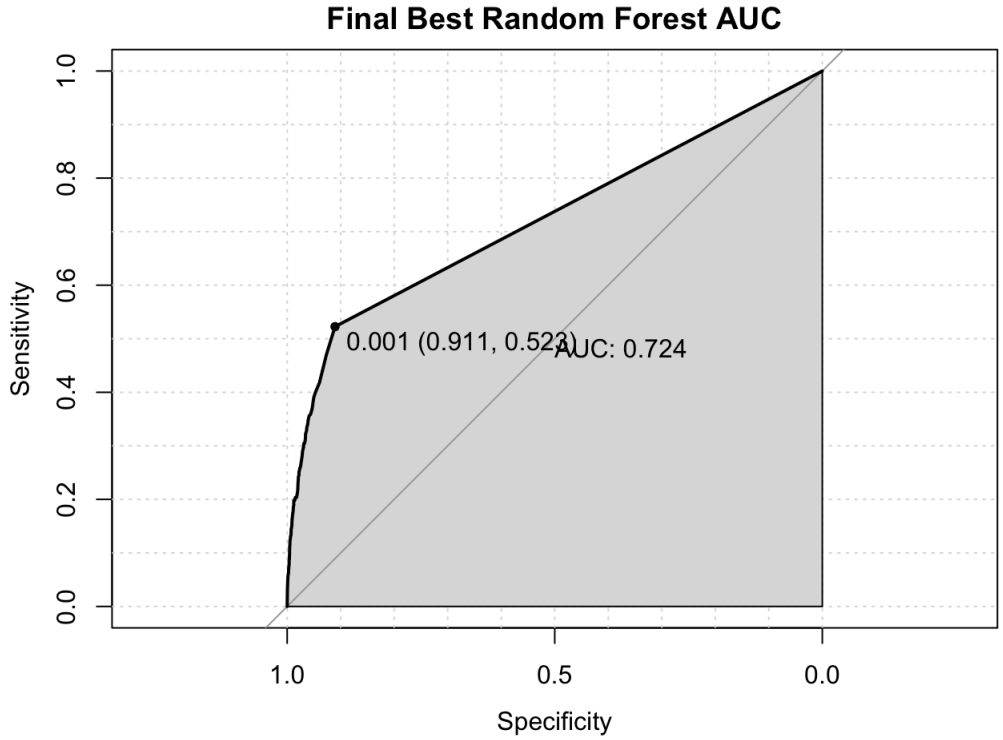

```
##
## Call:
##  randomForest(formula = y ~ ., data = Data3, importance = T, subset = train)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 1
##
##         OOB estimate of  error rate: 11.35%
## Confusion matrix:
##        no yes class.error
## no  18351 260  0.01397023
## yes  2163 567  0.79230769
```

```
## [1] 0.8864292
```



```
##    no yes
## 13  1   0
## 14  1   0
## 15  1   0
## 17  1   0
## 19  1   0
## 39  1   0
```





Logistic Regression

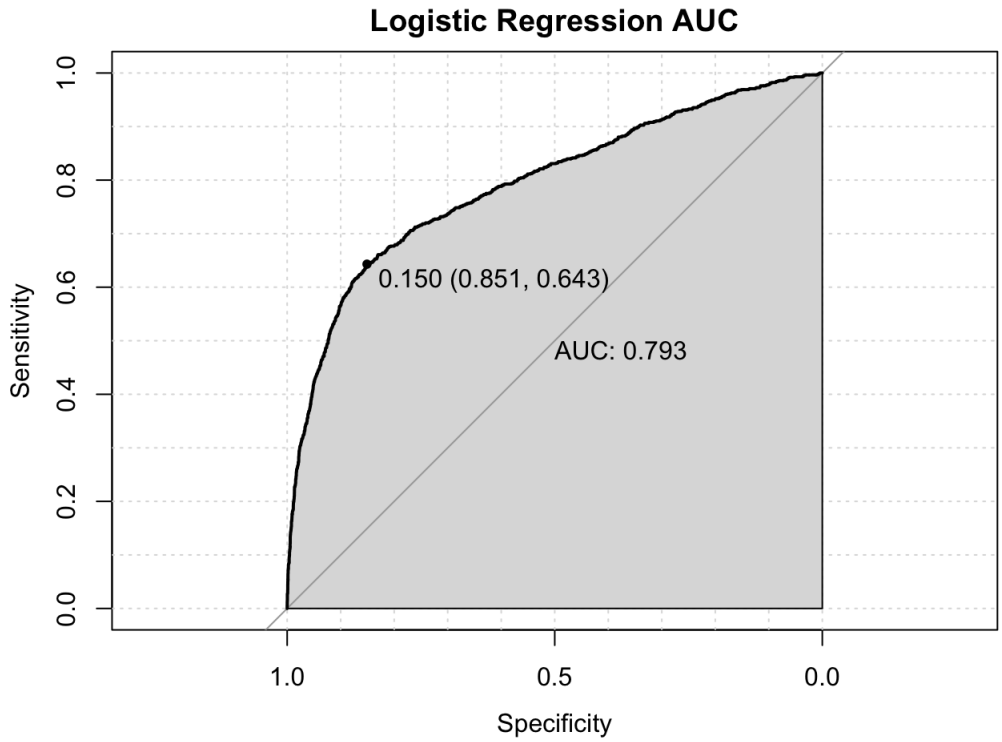
We found our dataset was heavily unbalanced, with a greater number of “no” (n = 26,629) responses than “yes” (n = 3,859) for our response variable y. Recognizing the unbalance, we relied on AUC rather than accuracy as an indicator of model performance because accuracy is a very poor metric of unbalanced data. Instead, we rely on ROC and AUC, which are a measure of how good the model is at distinguishing between various classes (Yadav, 2020).

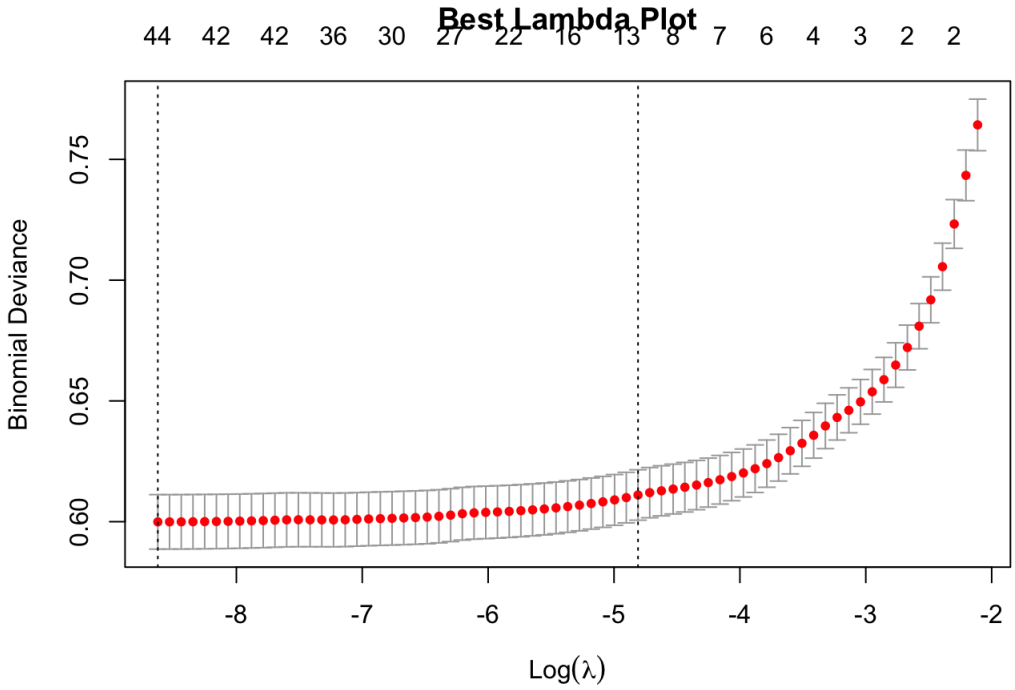
Based on the results of GVIF, the variables nr.employes and emp.var.rate were dropped from the logistic model due to multicollinearity. The results from this model are shown in summary(log.model.2). We can see not all of these predictors are significant from the summary output.

The AUC associated with our log.model.2 was 0.793. With the LASSO model fitted on the best lambda from CV, the AUC associated with LASSO was 0.800. This indicated based on AUC, we prefer the logistic model over the LASSO. This is not surprising considering the goal of implementing the LASSO was to develop a better model. However, the difference between the two methods was quite small.

```
##
##    no   yes
## 26629  3859
```

```
##
## Call:
## glm(formula = y ~ age + job + marital + education + default +
##     housing + loan + contact + month + day_of_week + campaign +
##     pdays + previous + poutcome + cons.price.idx + cons.conf.idx +
##     euribor3m, family = binomial, data = train.data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2275  -0.4261  -0.3508  -0.2406   3.0587
##
## Coefficients:
##                                Estimate Std. Error z value Pr(>|z|)
## (Intercept)                  -3.812e+01  5.045e+00  -7.555 4.18e-14 ***
## age                          -4.079e-04  2.863e-03  -0.142 0.886732
## jobblue-collar               -2.326e-01  9.357e-02  -2.486 0.012929 *
## jobentrepreneur              -2.325e-01  1.507e-01  -1.543 0.122828
## jobhousemaid                 -9.301e-02  1.727e-01  -0.539 0.590157
## jobmanagement                -5.198e-02  9.686e-02  -0.537 0.591462
## jobretired                    2.012e-01  1.271e-01   1.583 0.113519
## jobself-employed             -4.027e-02  1.304e-01  -0.309 0.757435
## jobservices                  -1.931e-01  1.001e-01  -1.928 0.053803 .
## jobstudent                    3.549e-01  1.304e-01   2.722 0.006495 **
## jobtechnician                 2.561e-02  7.874e-02   0.325 0.745019
## jobunemployed                -7.958e-03  1.422e-01  -0.056 0.955370
## maritalmarried                6.596e-02  7.913e-02   0.834 0.404513
## maritalsingle                 6.448e-02  8.923e-02   0.723 0.469931
## educationbasic.6y             3.246e-02  1.478e-01   0.220 0.826177
## educationbasic.9y            -7.453e-02  1.135e-01  -0.656 0.511583
## educationhigh.school         -3.055e-02  1.097e-01  -0.278 0.780733
## educationilliterate           8.160e-01  8.875e-01   0.919 0.357837
## educationprofessional.course -7.395e-02  1.192e-01  -0.621 0.534855
## educationuniversity.degree    1.098e-02  1.098e-01   0.100 0.920389
## defaultyes                   -8.713e+00  1.387e+02  -0.063 0.949900
## housingyes                   -3.808e-02  4.692e-02  -0.812 0.416983
## loanyes                      -4.350e-02  6.489e-02  -0.670 0.502601
## contacttelephone             -5.336e-01  7.826e-02  -6.818 9.23e-12 ***
## monthaug                     -4.806e-02  1.185e-01  -0.406 0.685071
## monthdec                      3.203e-01  2.352e-01   1.362 0.173252
## monthjul                      2.490e-01  1.101e-01   2.260 0.023794 *
## monthjun                      2.212e-01  1.058e-01   2.091 0.036484 *
## monthmar                      1.031e+00  1.381e-01   7.469 8.08e-14 ***
## monthmay                     -6.258e-01  8.592e-02  -7.284 3.25e-13 ***
## monthnov                     -3.538e-02  1.105e-01  -0.320 0.748742
## monthoct                      2.377e-01  1.416e-01   1.679 0.093211 .
## monthsep                     -1.695e-02  1.514e-01  -0.112 0.910874
## day_of_weekmon               -2.498e-01  7.654e-02  -3.264 0.001100 **
## day_of_weekthu                7.516e-02  7.301e-02   1.029 0.303257
## day_of_weektue                2.265e-02  7.515e-02   0.301 0.763067
## day_of_weekwed                1.317e-01  7.439e-02   1.770 0.076736 .
## campaign                     -4.435e-02  1.246e-02  -3.559 0.000373 ***
## pdays                        -1.052e-03  2.513e-04  -4.187 2.83e-05 ***
## previous                     -1.991e-02  7.096e-02  -0.281 0.779043
## poutcomenonexistent           5.087e-01  1.098e-01   4.632 3.62e-06 ***
## poutcomesuccess               8.638e-01  2.449e-01   3.527 0.000420 ***
## cons.price.idx                4.287e-01  5.550e-02   7.723 1.14e-14 ***
## cons.conf.idx                 3.480e-02  6.078e-03   5.725 1.03e-08 ***
## euribor3m                    -5.492e-01  2.046e-02 -26.846  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 16322  on 21340  degrees of freedom
## Residual deviance: 12835  on 21296  degrees of freedom
## AIC: 12925
##
## Number of Fisher Scoring iterations: 10
```

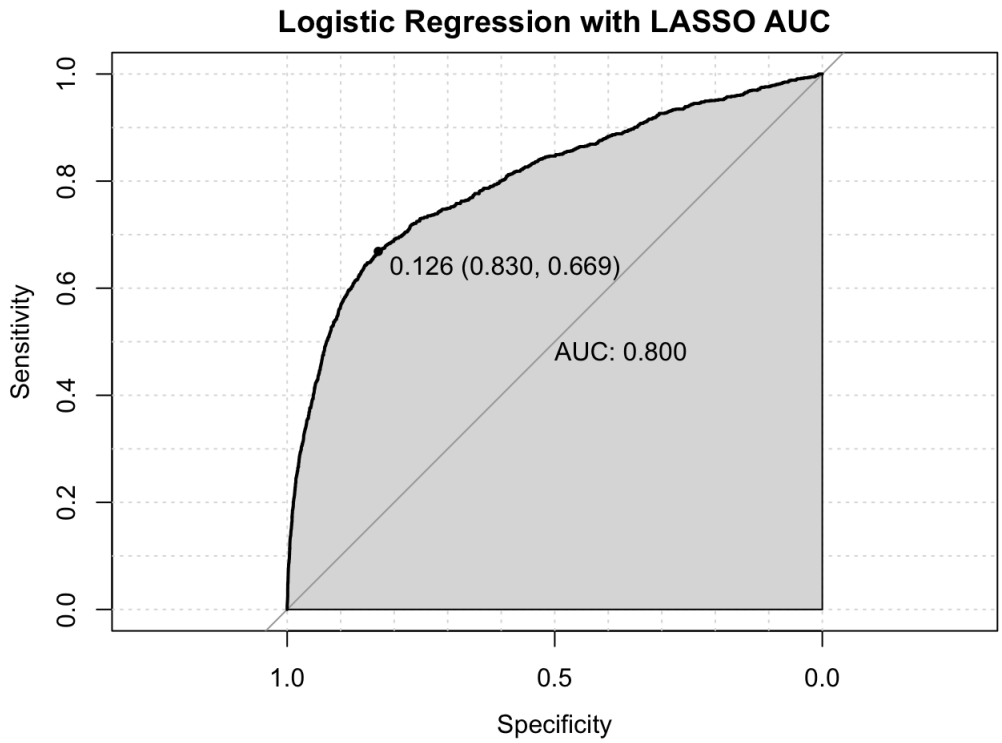

Logistic Regression AUC
Sensitivity
1.0
0.8
0.6
0.4
0.2
0.0
0.150 (0.851, 0.643)
AUC: 0.793
1.0
0.5
0.0
Specificity


Best Lambda Plot
44 42 42 36 30 27 22 16 13 8 7 6 4 3 2 2
Binomial Deviance
0.75
0.70
0.65
0.60
-8
-7
-6
-5
-4
-3
-2
Log(λ)

```
## 53 x 1 sparse Matrix of class "dgCMatrix"
##                                        1
## (Intercept)                   -1.386710e+02
## age                           -7.199508e-04
## jobblue-collar                -1.902842e-01
## jobentrepreneur               -2.117844e-01
## jobhousemaid                  -6.709199e-02
## jobmanagement                 -3.213044e-02
## jobretired                     1.922606e-01
## jobself-employed              -3.616758e-02
## jobservices                   -1.720696e-01
## jobstudent                     3.224017e-01
## jobtechnician                  3.606874e-02
## jobunemployed                  .
## jobunknown                     .
## maritalmarried                 3.059918e-02
## maritalsingle                  3.041132e-02
## maritalunknown                 .
## educationbasic.6y              3.000863e-02
## educationbasic.9y             -6.205749e-02
## educationhigh.school           .
## educationilliterate            8.233714e-01
## educationprofessional.course -4.095127e-02
## educationuniversity.degree     2.946341e-02
## educationunknown               .
## defaultunknown                 .
## defaultyes                    -1.041640e+00
## housingunknown                 .
## housingyes                    -3.208501e-02
## loanunknown                    .
## loanyes                       -2.589111e-02
## contacttelephone              -6.873026e-01
## monthaug                       3.557811e-01
## monthdec                       2.017114e-01
## monthjul                       1.010340e-01
## monthjun                      -3.874768e-01
## monthmar                       1.267014e+00
## monthmay                      -4.624590e-01
## monthnov                      -4.599142e-01
## monthoct                      -4.531804e-02
## monthsep                       7.341762e-02
## day_of_weekmon                -2.544778e-01
## day_of_weekthu                 8.868905e-02
## day_of_weektue                 4.259978e-03
## day_of_weekwed                 1.354307e-01
## campaign                      -3.626881e-02
## pdays                         -9.621303e-04
## previous                      -4.029711e-02
## poutcomenonexistent            4.486622e-01
## poutcomesuccess                9.047363e-01
## emp.var.rate                  -1.191556e+00
## cons.price.idx                 1.451660e+00
## cons.conf.idx                  1.212125e-02
## euribor3m                      3.260355e-01
## nr.employed                    1.714393e-04
```

```
## [1] 0.888816
```

```
##            Length Class     Mode
## a0            71   -none-    numeric
## beta        3692   dgCMatrix S4
## df            71   -none-    numeric
## dim            2   -none-    numeric
## lambda        71   -none-    numeric
## dev.ratio     71   -none-    numeric
## nulldev        1   -none-    numeric
## npasses        1   -none-    numeric
## jerr           1   -none-    numeric
## offset         1   -none-    logical
## classnames     2   -none-    character
## call           5   -none-    call
## nobs           1   -none-    numeric
```



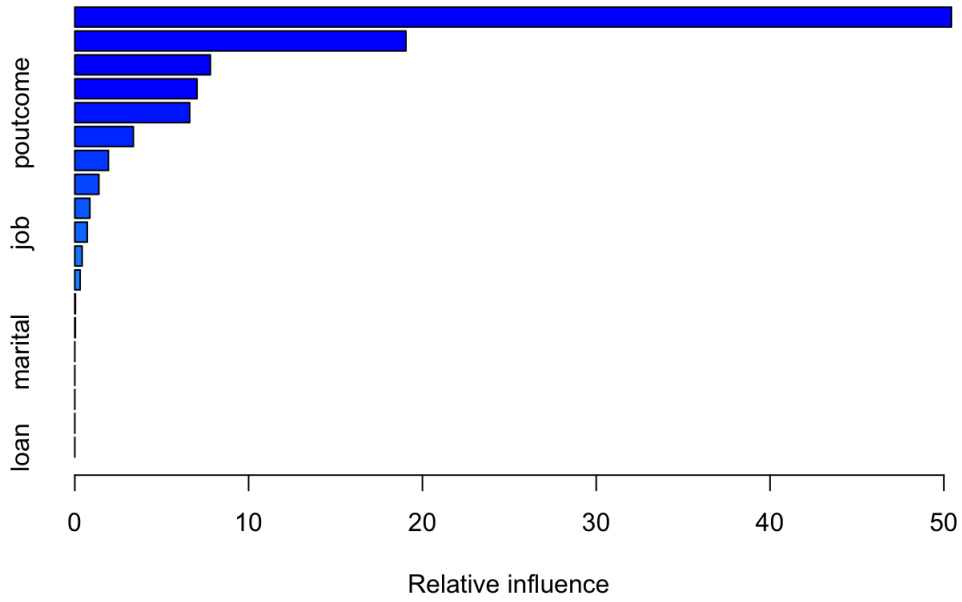
Boosting

The summary from the initial gbm model is shown in summary(gbm.model). The AUC associated with this was 0.798. However, after using CV to find the correct number of n.trees and shrinkage, we calculated the following mean accuracies based on the five folds and parameter combinations:

CROSS-VALIDATTED K-FOLDS

Model	n.trees	shrinkage	Accuracy	Rank
1a	1000	0.01	0.885654	1
1b	1000	0.02	0.884257	2
1c	1000	0.03	0.884024	3*
2a	2000	0.01	0.884024	3*
2b	2000	0.02	0.883326	6
2c	2000	0.03	0.882394	8
3a	3000	0.01	0.884024	3*
3b	3000	0.02	0.883093	7
3c	3000	0.03	0.881928	9

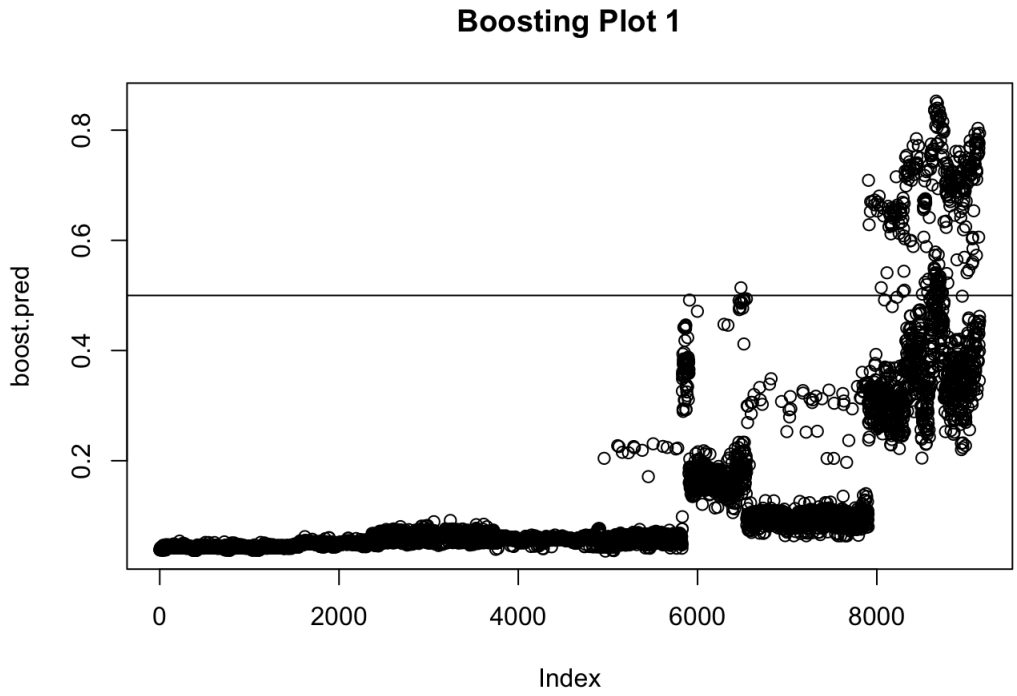
From the ranking shown in the table above, we can see the best parameter combination was n.trees = 1000 and shrinkage = 0.01, which were the parameter combination that were fit to the initial gbm model, so the gbm was not re-ran. The final AUC derived from boosting is shown in “Boosting AUC” below.



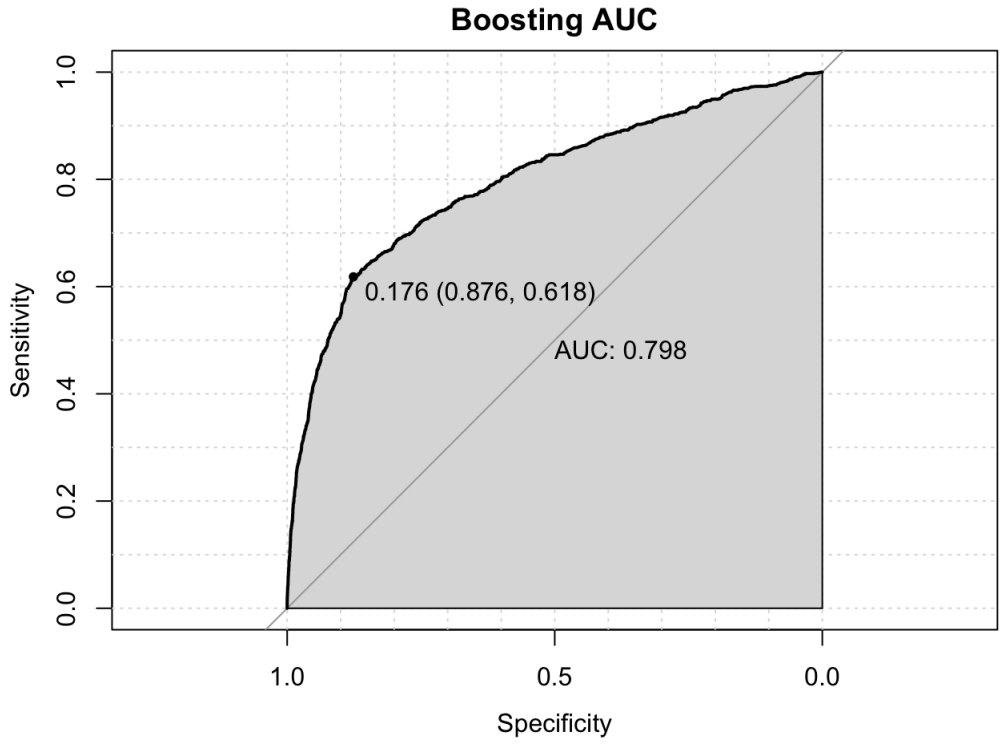
nr.employed
pdays
month
euribor3m
poutcome
cons.conf.idx
emp.var.rate
contact
age
job

1-10 of 19 rows | 1-1 of 3 columns

Previous 1 2 Next




```
##
## pred.out    0    1
##         0 7917  899
##         1  101  230
```



```
## [1] "Mean accuracy of cv for n.tree = 1000 shrinkage = 0.01"
## [1] 0.8856544
## [1] "Mean accuracy of cv for n.tree = 1000 shrinkage = 0.02"
## [1] 0.8842571
## [1] "Mean accuracy of cv for n.tree = 1000 shrinkage = 0.03"
## [1] 0.8840242
## [1] "Mean accuracy of cv for n.tree = 2000 shrinkage = 0.01"
## [1] 0.8840242
## [1] "Mean accuracy of cv for n.tree = 2000 shrinkage = 0.02"
## [1] 0.8833256
## [1] "Mean accuracy of cv for n.tree = 2000 shrinkage = 0.03"
## [1] 0.882394
## [1] "Mean accuracy of cv for n.tree = 3000 shrinkage = 0.01"
## [1] 0.8840242
## [1] "Mean accuracy of cv for n.tree = 3000 shrinkage = 0.02"
## [1] 0.8830927
## [1] "Mean accuracy of cv for n.tree = 3000 shrinkage = 0.03"
## [1] 0.8819283
```

Extra Credit

The thresholding step got rid of the features default and loan. From the next step, the following 12 variables in this order were identified– euribor3m, month, emp.var.rate, nr.employed, cons.price.idx, cons.conf.idx, contact, job, pdays, day_of_week, age, and poutcome. The greatest reduction of error occured between the first and second steps and can be seen between the thresholding and interpretation step. This can be seen in the graph on the top right and bottom left, respectively.

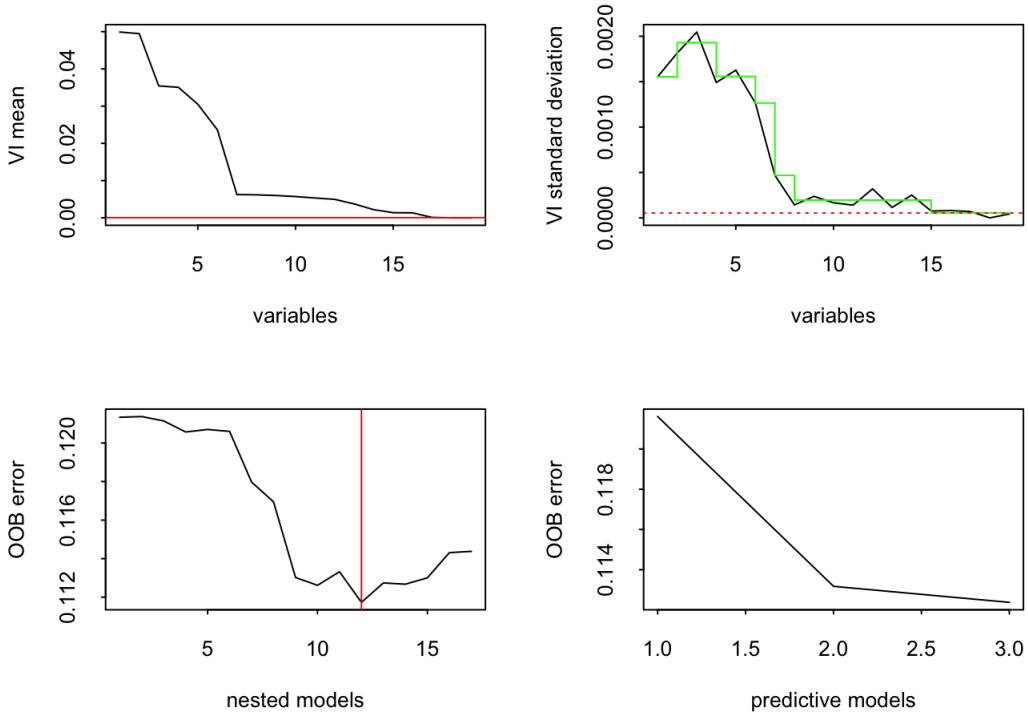
The third step step lept us with the following three variables : 1) euribor3m, 2) pdays, and 3) poutcome. This tells us that for prediction purposes for future campaigns, the only variables that need to be used are these three. This is shown in the graph in the bottom right corner. VSURF, random forests are typically built using ntree = 2000 trees. However due to processing power, size of the dataset and memory space required, we were not able to run code with that default. Cross-referencing with other readings on defining parameters, we confirmed that for large datasets, trees had to be around 500 or more, and went with that metric. In terms of the mtry value for classification model purposes, the value is the square root of the number of predictor values, rounded down.

```
## Thresholding step
## Estimated computational time (on one core): 3394.7 sec.
##
  |
  |                                                                      |   0%
  |
  |=                                                                     |   2%
  |
  |===                                                                   |   4%
  |
  |====                                                                  |   6%
  |
  |======                                                                |   8%
  |
  |=======                                                               |  10%
  |
  |========                                                              |  12%
  |
  |==========                                                            |  14%
  |
  |===========                                                           |  16%
  |
  |=============                                                         |  18%
  |
  |==============                                                        |  20%
  |
  |===============                                                       |  22%
  |
  |=================                                                     |  24%
  |
  |==================                                                    |  26%
  |
  |====================                                                  |  28%
  |
  |=====================                                                 |  30%
  |
  |======================                                                |  32%
  |
```

```
  |
  |========================                                              |  34%
  |
  |=========================                                             |  36%
  |
  |===========================                                           |  38%
  |
  |============================                                          |  40%
  |
  |==============================                                        |  42%
  |
  |===============================                                       |  44%
  |
  |================================                                      |  46%
  |
  |==================================                                    |  48%
  |
  |===================================                                   |  50%
  |
  |====================================                                  |  52%
  |
  |======================================                                |  54%
  |
  |=======================================                               |  56%
  |
  |========================================                              |  58%
  |
  |=========================================                             |  60%
  |
  |===========================================                           |  62%
  |
  |============================================                          |  64%
  |
  |=============================================                         |  66%
  |
  |==============================================                        |  68%
  |
  |================================================                      |  70%
  |
  |=================================================                     |  72%
  |
  |==================================================                    |  74%
  |
  |====================================================                  |  76%
  |
  |=====================================================                 |  78%
  |
  |======================================================                |  80%
  |
  |========================================================              |  82%
  |
  |=========================================================             |  84%
  |
  |==========================================================            |  86%
  |
  |============================================================          |  88%
  |
  |=============================================================         |  90%
  |
  |==============================================================        |  92%
  |
  |================================================================      |  94%
  |
  |=================================================================     |  96%
  |
  |==================================================================    |  98%
  |
  |======================================================================| 100%
## Interpretation step (on 17 variables)
## Estimated computational time (on one core): between 1683 sec. and  11565.5 sec.
##
  |
  |                                                                      |   0%
```

```
  |
  |====                                                                  |   6%
  |
  |========                                                              |  12%
  |
  |============                                                          |  18%
  |
  |================                                                      |  24%
  |
  |====================                                                  |  29%
  |
  |=========================                                             |  35%
  |
  |=============================                                         |  41%
  |
  |=================================                                     |  47%
  |
  |=====================================                                 |  53%
  |
  |=========================================                             |  59%
  |
  |=============================================                         |  65%
  |
  |=================================================                     |  71%
  |
  |=====================================================                 |  76%
  |
  |=========================================================             |  82%
  |
  |=============================================================         |  88%
  |
  |=================================================================     |  94%
  |
  |======================================================================| 100%
## Prediction step (on 12 variables)
## Maximum estimated computational time (on one core): 3989.1 sec.
##
  |
  |                                                                      |   0%
  |
  |======                                                                |   8%
  |
  |============                                                          |  17%
  |
  |==================                                                    |  25%
  |
  |========================                                              |  33%
  |
  |==============================                                        |  42%
  |
  |====================================                                  |  50%
  |
  |==========================================                            |  58%
  |
  |================================================                      |  67%
  |
  |======================================================                |  75%
  |
  |============================================================          |  83%
  |
  |==================================================================    |  92%
  |
  |======================================================================| 100%
```

```
##
##  VSURF computation time: 2.8 hours
##
##  VSURF selected:
##  17 variables at thresholding step (in 60 mins)
##  12 variables at interpretation step (in 1.5 hours)
##  3 variables at prediction step (in 18 mins)
```




```
##  [1] "varselect.thres"     "varselect.interp"    "varselect.pred"
##  [4] "nums.varselect"      "imp.varselect.thres" "min.thres"
##  [7] "imp.mean.dec"        "imp.mean.dec.ind"    "imp.sd.dec"
## [10] "mean.perf"           "pred.pruned.tree"    "err.interp"
## [13] "sd.min"              "err.pred"            "mean.jump"
## [16] "nmin"                "nsd"                 "nmj"
## [19] "overall.time"        "comput.times"        "RFimplem"
## [22] "ncores"              "clusterType"         "call"
```

```
##  [1] 18 12 15 19 16 17  8  2 11 13  1 14  4 10  3  9  6
```

```
##  [1] 18 12 15 19 16 17  8  2 11 13  1 14
```

```
## [1] 18 11 14
```

Discussion

Finally, we will conclude the report by comparing results across all 4 methodologies that were presented above and answer the three questions that fueled our analyses. Logistic regression was performed without and with LASSO, but since the LASSO logistic model was superior, those results are presented in the table below.

Method	AUC	Rank
Random Forest	0.799	2
Random Forest (s)	0.724	4
Boosting	0.798	3
Logistic (w/LASSO)	0.800	1

A greater AUC is an indicator of the best performing model. Based results shown in the above table, the superior model based on AUC was Logistic (w/LASSO). Although the AUC were very similar between the random forest and Logistic (w/LASSO), we would prefer the Logistic (w/LASSO) because it is a simpler model that is less computationally intensive with results that are easier to interpret.

Answers to Research Questions

1. No, all 20 attributes are not relevant to the model. We had some issues with multicollinearity among the social and economic indicator, which indicated correlation and that all variables were not relevant to the model. Further, when we looked at the results from the straight Logistic Regression, we saw many of the predictors were not significant. These results were further emphasized by the variable importance plots that were shown across the various classifcation methods.

2. Yes, some social and economic indicators were relevant to determining whether a client subscribed to a term deposit. In nearly all the methods, euribor3m, which was a “Daily indicator” was highly ranked among the variable importance plot. This is not surprising considering clients are most likely to subscribe when the economy is doing well.
3. As indicated by the description of the dataset, duration was extremely influential on subscribing to a long-term deposit. It was recommended to be only used as a benchmark. Preliminary analyses that were not included in this report showed duration to be extremely influential on variable importance plots. Thus, duration was dropped from the dataset before completing random forest, logistic regression, and boosting. However, it is important to note for banking institutions that longer call durations were associated with increased campaign success.

Literature Cited

Agresti, Alan. 2018. An introduction to categorical data analysis. John Wiley & Sons.

Breiman, L. 2001. Random Forests. Statistics Department. University of California, Berkely.

Fox, J., and G. Monette. 1992. Generalized Collinearity Diagnostics. Journal of the American Statistical Association 87(417):178–83.

Genuer, R., J.M. Poggi, and C. Tuleau-Malot. 2010. Variable selection using random forests. Pattern Recognition Letters. 31(14):2225-2236.

Kirasich, K., T. Smith, and B. Sadler. 2018. Random Forest vs Logistic Regression: Binary Classification for Heterogeneous Datasets," SMU Data Science Review, Vol. 1 : No. 3 , Article 9.

Moro, S., P. Cortez, and P. Rita. 2014. A Data-Driven Approach to Predict the Success of Bank Telemarketing. Decision Support Systems, Elsevier, 62:22-31.

UCD. 2018. Gradient Boosting Machines. UC Davis Business Analytics R Programming Guide. 2018.

Yadav, D. 2020. Weighted Regression for an Imbalanced Dataset.Towards Data Science.