STA141A Final Project

Animay Sharma, Aditya Kallepalli, Charles Chien, Shaumik Pathak

12/14/2020

1. Introduction

1.1 Background

Here, we have a set of marketing data of a banking institution. The data is related with direct marketing campaigns of a Portuguese banking institution. The marketing campaigns were based on phone calls. Often, more than one contact to the same client was required, in order to access if the product (bank term deposit) would be ('yes') or not ('no') subscribed.

Among the datasets provided, we chose to prioritize and utilize the "bank-full.csv" in this report, with 17 different inputs. We will only use the "bank.csv" dataset if it gets computationally difficult with the full dataset.

As mentioned in article (Lopez, Customer segmentation using machine learning 2020) [0], data science and machine learning methods are helpful when it comes to helping companies with customer segmentation. Customer targeting is the process of analyzing customer features to select those customers who are more prone to a target product or service. By making intelligent use of data, companies could make a big difference to their competitors.

Advanced analytics plays a key role when it comes to selecting potentially profitable clients, which allows the design of more effective marketing campaigns. By using the four steps of advanced analytics: descriptive, diagnostic, predictive, and prescriptive, we would be able to answer key questions such as "what happened?", "why did it happen?", "what will happen?", and "how can we make it happen?"

In this report, we would be covering most of those steps in depth. Our primary goal is to build a predictive model to answer a simple yes or no question: to determine whether a client will sign on to a long-term deposit. A model as such would allow banks to save on marketing expense on groups of customers that have a low chance of subscription, and focus on other customers that have a high chance of success. Overall, this would improve the profitability of banks and ultimately decrease marketing deficiencies. We would build 3 different models, compare those candidates, and ultimately find the one that is more suitable and leads to a smaller error.

While our main goal is to build a classification model and assist with bank marketing efforts, we would also like to conduct an exploratory data analysis (EDA) to explore relationships between different input variables. We would report any useful insights along the way, which covers both the "descriptive" and "diagnostic" parts of the four steps of advanced analytics as mentioned in the article.

[0] Lopez, R. (2020). Customer segmentation using machine learning. Retrieved December 12, 2020, from https://www.neuraldesigner.com/blog/customer_segmentation_using_advanced_analytics

1.1.1 Data Description

Here is an example of how our data looks like:

```r
head(bank.data)
```

```
## # A tibble: 6 x 17
##     age job   marital education default balance housing loan  contact   day
##   <dbl> <chr> <chr>   <chr>     <chr>     <dbl> <chr>   <chr> <chr>   <dbl>
## 1    58 mana~ married tertiary  no         2143 yes     no    unknown     5
## 2    44 tech~ single  secondary no           29 yes     no    unknown     5
## 3    33 entr~ married secondary no            2 yes     yes   unknown     5
## 4    47 blue~ married unknown   no         1506 yes     no    unknown     5
## 5    33 unkn~ single  unknown   no            1 no      no    unknown     5
## 6    35 mana~ married tertiary  no          231 yes     no    unknown     5
## # ... with 7 more variables: month <chr>, duration <dbl>, campaign <dbl>,
## #   pdays <dbl>, previous <dbl>, poutcome <chr>, y <chr>
```

This data description is from the UCI machine learning repository for the dataset 'Bank Marketing Data Set'.[1]

1 - age (numeric)

2 - job : type of job (categorical: 'admin.','blue-collar','entrepreneur','housemaid','management','retired','self-employed','services','student','technician','unemployed','unknown')

3 - marital : marital status (categorical: 'divorced','married','single','unknown'; note: 'divorced' means divorced or widowed)

4 - education (categorical: 'basic.4y','basic.6y','basic.9y','high.school','illiterate','professional.course','university.degree','unkno

5 - default: has credit in default? (categorical: 'no','yes','unknown')

6 - housing: has housing loan? (categorical: 'no','yes','unknown')

7 - loan: has personal loan? (categorical: 'no','yes','unknown')

8 - contact: contact communication type (categorical: 'cellular','telephone')

9 - month: last contact month of year (categorical: 'jan', 'feb', 'mar', ..., 'nov', 'dec')

10 - day_of_week: last contact day of the week (categorical: 'mon','tue','wed','thu','fri')

11 - duration: last contact duration, in seconds (numeric).

12 - campaign: number of contacts performed during this campaign and for this client (numeric, includes last contact)

13 - pdays: number of days that passed by after the client was last contacted from a previous campaign (numeric; 999 means client was not previously contacted)

14 - previous: number of contacts performed before this campaign and for this client (numeric)

15 - poutcome: outcome of the previous marketing campaign (categorical: 'failure','nonexistent','success')

Output variable (desired target): y - has the client subscribed a term deposit? (binary: 'yes','no')

[1] Bank Marketing Data Set, [Moro Et Al., 2014] S. Moro, P. Cortez and P. Rita. A Data-Driven Approach to Predict the Success of Bank Telemarketing. Decision Support Systems, Elsevier, 62:22-31, June 2014, archive.ics.uci.edu/ml/datasets/Bank+Marketing.

1.2 Statistical Questions of Interest

To answer the primary scientific question of interest, we would fit our model by 3 different methods. The response will be a binary yes/no variable "has the client subscribed a term deposit?" All other variables

provided will then be our input variables to allow us to build this model. Here, our 3 classification methods are

1. Logistic Regression
2. Random Forest
3. SVM (Extra Method)

For our first method, We would then use both backward and forward stepwise model selection using a likelihood ratio test (LRT) to conduct a heauristic model selection and prune down our model. We would also use AIC and BIC and compare the results of those with LRT. Ultimately, out of all these variable selection methods, we would choose only one logistic regression model that is the most robust among the all.

Then, we will build a Random Forest with tuned mtry and ntree parameters.

Then, we will compare our Logistic Regression and Random Forest along with SVM, particularly their performance when running on the test data set that we separated, and obtain one final candidate to be our ultimate winner. For the purposes of simplicity, we are splitting our original dataset 80-20 (as a rule of thumb) to create our training and testing sets.

2. Analysis Plan

We would first start by conducting an Exploratory Data Analysis (EDA) to evaluate the relationship between different variables to the response variable. This would provide a descriptive story as to answering the most basic question like "what happened?" Then, we would start building our classification model using three methods.

2.1 Population and Study Design

The target population of this analysis would be all customers of banks in Portugal. We believe that sample from this particular Portuguese bank would be representative of all banks in Portugal. To build the models we are using a dataset with 80% of the data. The other 20% will be used to for testing and validating the accuracy of the models. This will help us assess which model is better at categorizing the data.

2.2 Statistical Analysis

2.2.1 Descriptive Analysis

Here is a high level summary of all the variables we have.

```
##       age            job              marital           education
##  Min.   :18.00   Length:45211       Length:45211       Length:45211
##  1st Qu.:33.00   Class :character   Class :character   Class :character
##  Median :39.00   Mode  :character   Mode  :character   Mode  :character
##  Mean   :40.94
##  3rd Qu.:48.00
##  Max.   :95.00
##    default            balance         housing             loan
##  Length:45211       Min.   : -8019   Length:45211       Length:45211
##  Class :character   1st Qu.:    72   Class :character   Class :character
##  Mode  :character   Median :   448   Mode  :character   Mode  :character
```

```
##                     Mean   :  1362
##                     3rd Qu.:  1428
##                     Max.   :102127
##    contact               day           month              duration
##  Length:45211       Min.   : 1.00   Length:45211       Min.   :   0.0
##  Class :character   1st Qu.: 8.00   Class :character   1st Qu.: 103.0
##  Mode  :character   Median :16.00   Mode  :character   Median : 180.0
##                     Mean   :15.81                      Mean   : 258.2
##                     3rd Qu.:21.00                      3rd Qu.: 319.0
##                     Max.   :31.00                      Max.   :4918.0
##     campaign          pdays          previous          poutcome
##  Min.   : 1.000   Min.   : -1.0   Min.   :  0.0000   Length:45211
##  1st Qu.: 1.000   1st Qu.: -1.0   1st Qu.:  0.0000   Class :character
##  Median : 2.000   Median : -1.0   Median :  0.0000   Mode  :character
##  Mean   : 2.764   Mean   : 40.2   Mean   :  0.5803
##  3rd Qu.: 3.000   3rd Qu.: -1.0   3rd Qu.:  0.0000
##  Max.   :63.000   Max.   :871.0   Max.   :275.0000
##       y
##  Length:45211
##  Class :character
##  Mode  :character
##
##
##
```

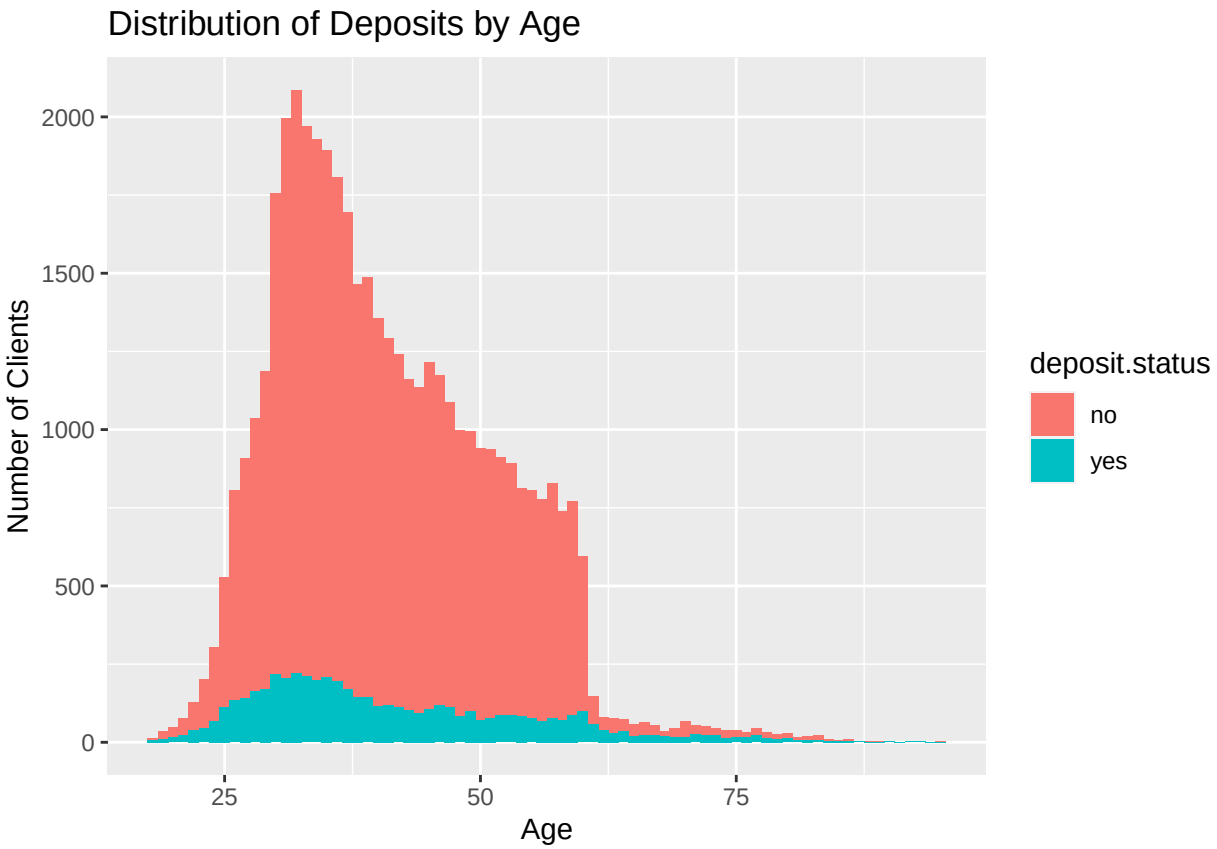
(Note: Stop explore_shiny manually to proceed with remaining code)

A short interpretation of the Exploratory Data Analysis is as follows:

1.) Age: Summary statistics of age are:

```
Mean Age: 40.936
Median Age: 39
25th Percentile Age: 33
75th Percentile Age: 48
```

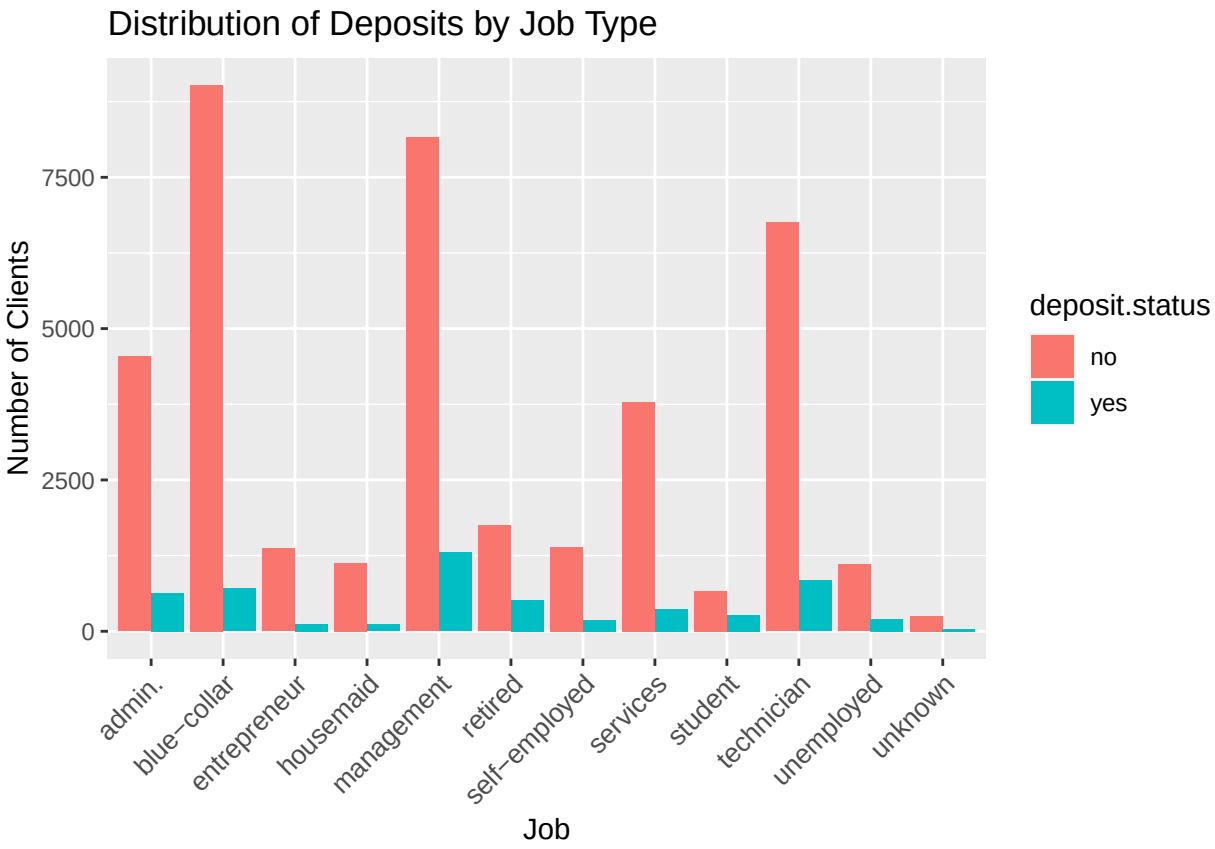
As we can see in the interactive plots when split by target is deselected, we can see that customers who are over the age of 60 are responsible for 33.6% of all deposits made followed by customers under the age of 30 at 18.5%



2.) Jobs: We can see a distribution of the types of jobs that dataset contains:

```
admin.        = 5 171 (11.4%)
blue-collar   = 9 732 (21.5%)
entrepreneur  = 1 487 (3.3%)
housemaid     = 1 240 (2.7%)
management    = 9 458 (20.9%)
retired       = 2 264 (5%)
self-employed = 1 579 (3.5%)
services      = 4 154 (9.2%)
student       = 938 (2.1%)
technician    = 7 597 (16.8%)
```

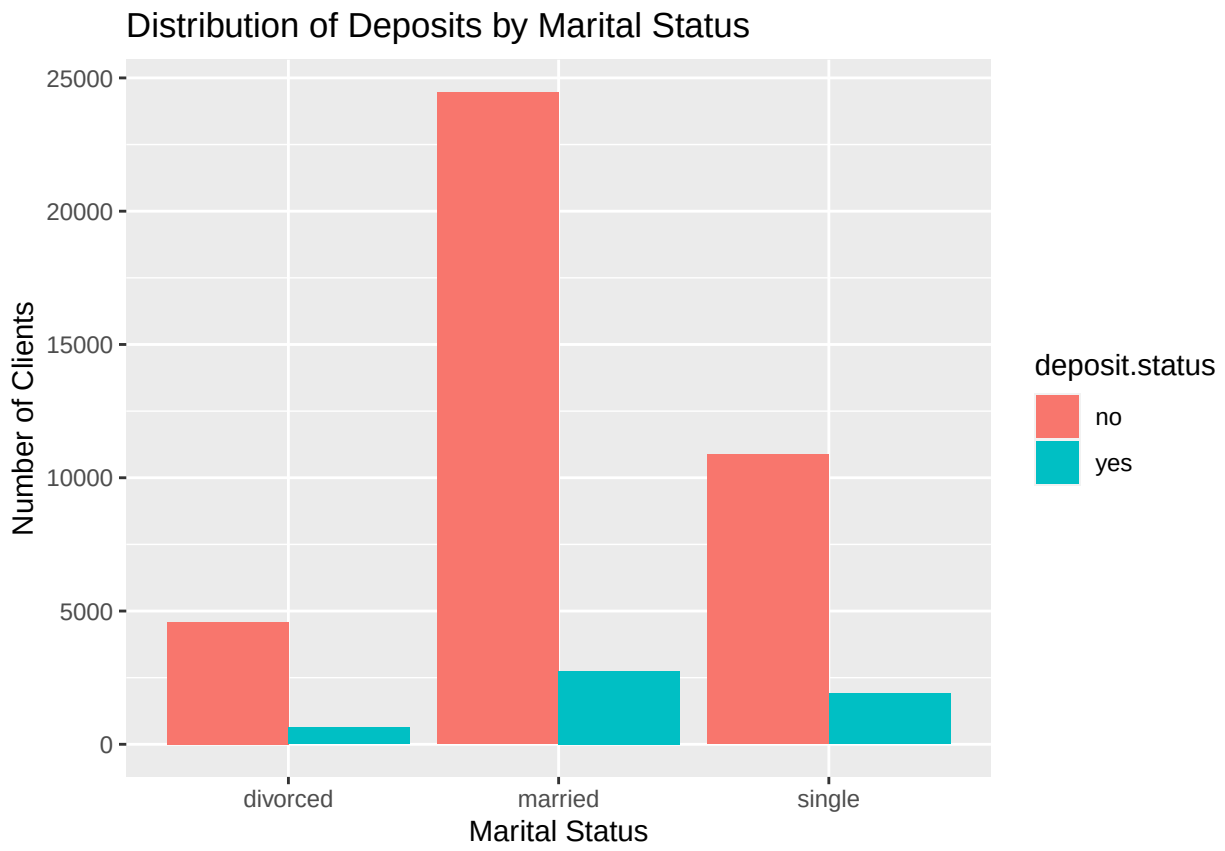
Most of the deposits are made by customers who work in management. Nearly 25% of customers who work as managers deposit money with the bank, followed by technicians at around 17%.



3.) Marital Status: Distribution of the martial status of customers is as follows:

```
divorced = 5 207 (11.5%)
married  = 27 214 (60.2%)
single   = 12 790 (28.3%)
```

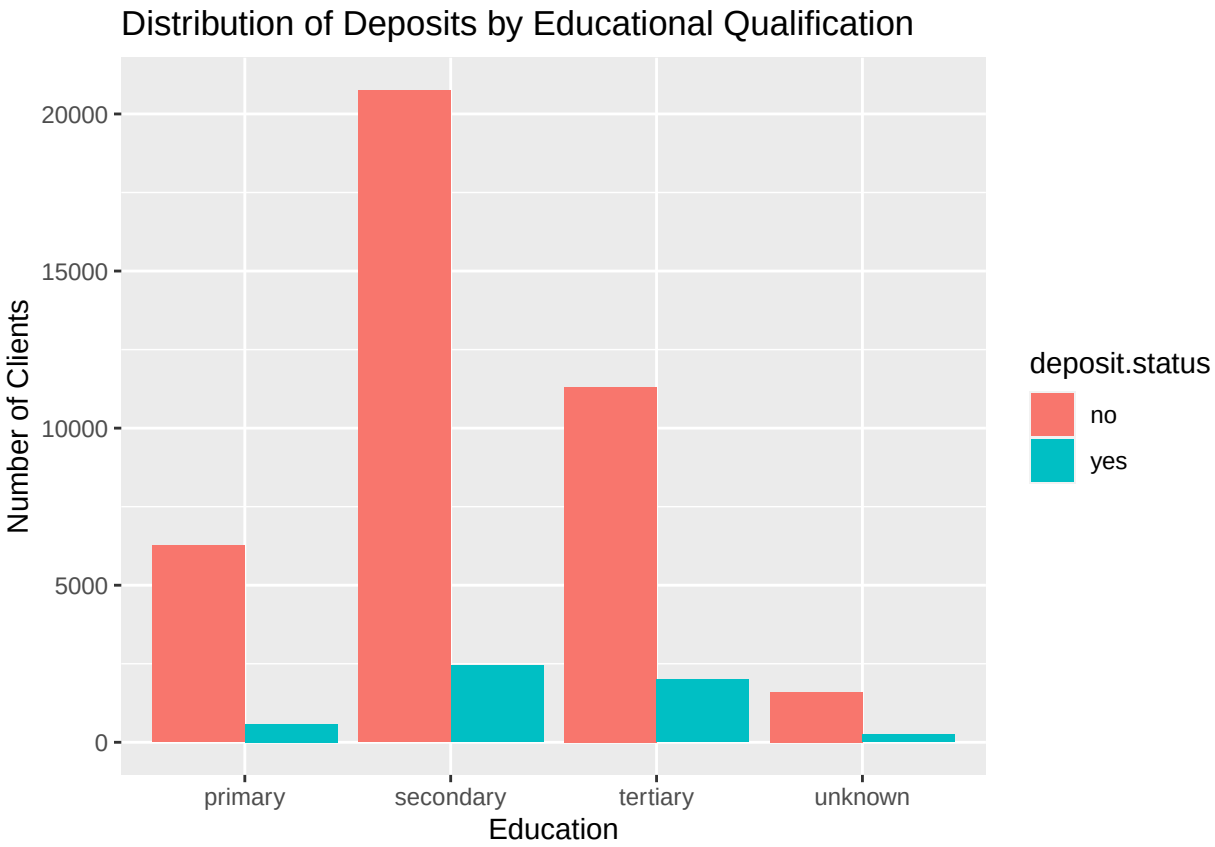
Married couples who deposit money in the bank account for 61.3% of deposits made, followed by 36.2% of single customers making deposits, followed by 11.8% of divorced customers make deposits.



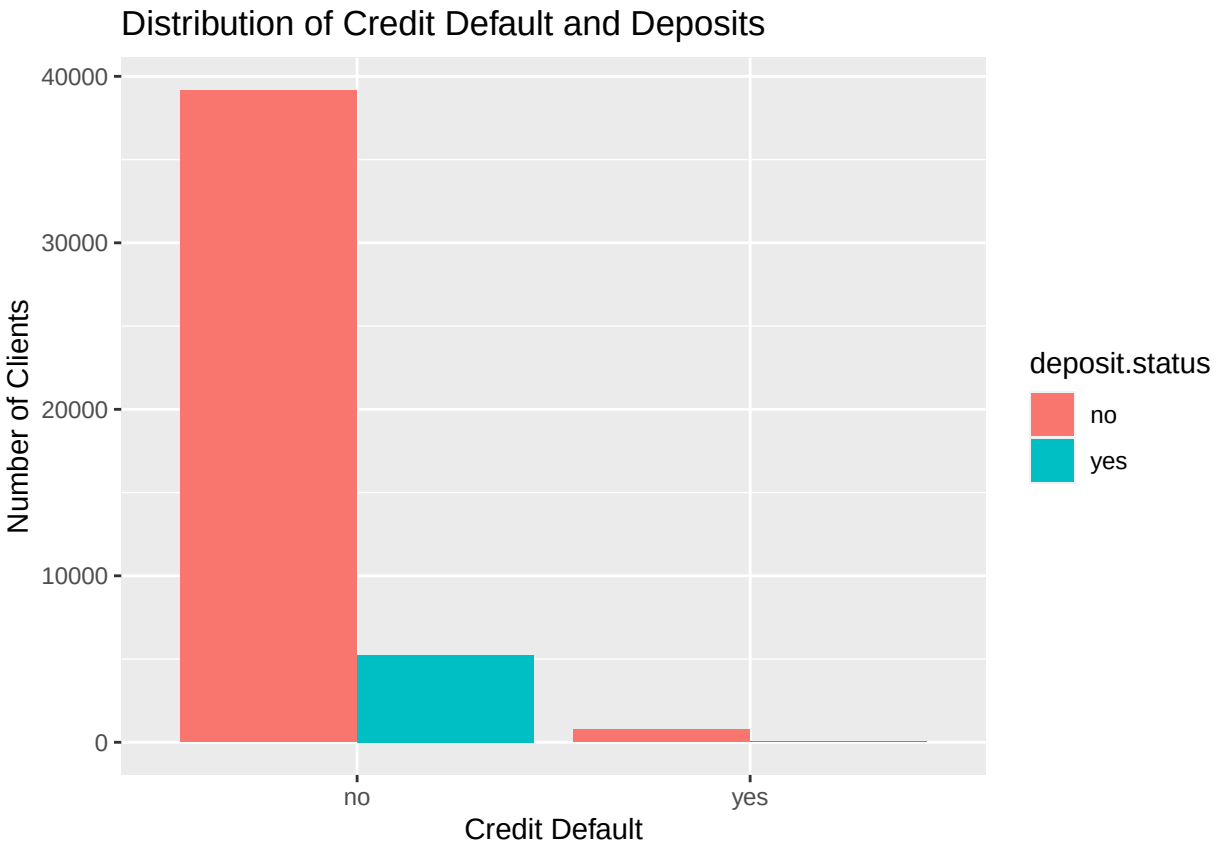
4.) Education: Summary statistics for educational distribution of the bank's customers are as follows:

```
primary = 6 851 (15.2%)
secondary = 23 202 (51.3%)
tertiary  = 13 301 (29.4%)
unknown   = 1 857 (4.1%)
```

46.3% of secondary educated customers are likely to make deposits, as compared to just 11.2% of primary educated customers and 37.7% tertiary educated customers make deposits.



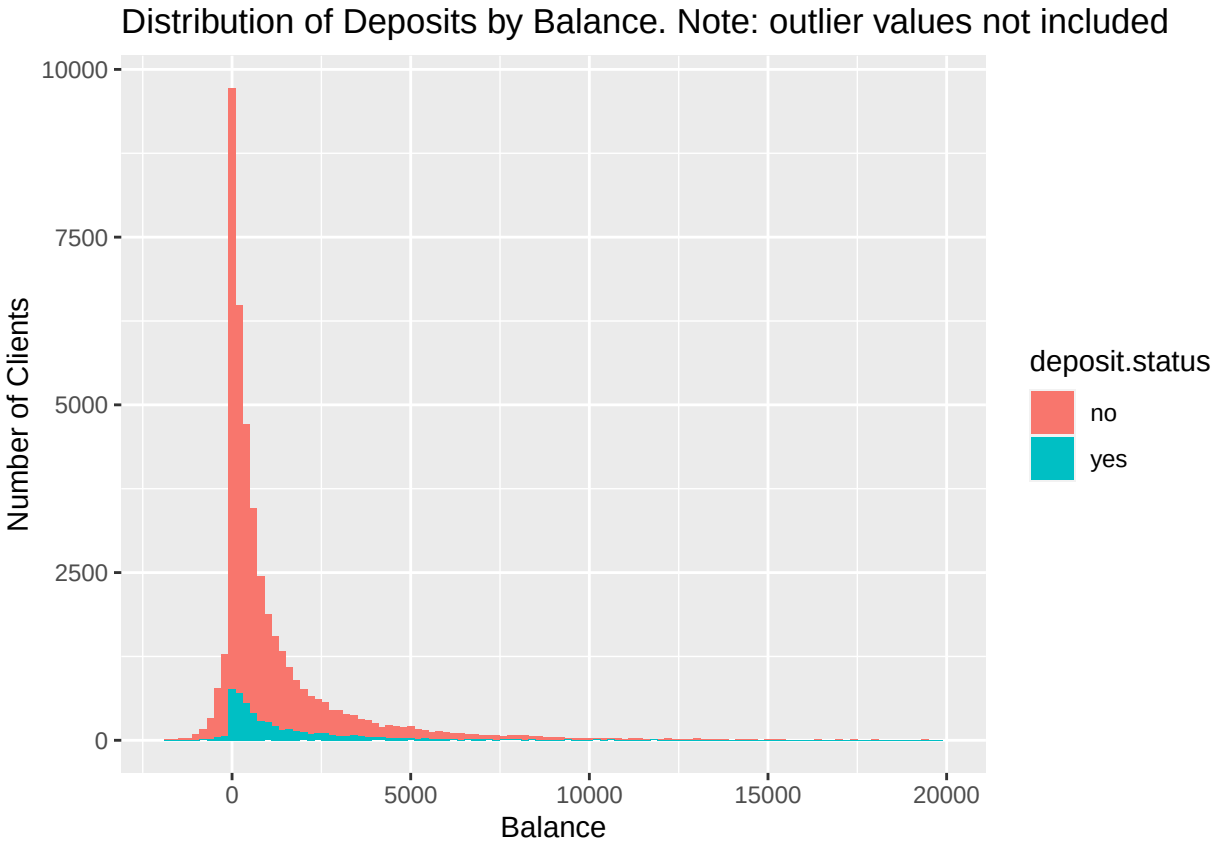
5.) Defaults: 98.2% of customers do not default on credit payments. 11.8% of customers who do not default on payments make deposits as compared to 6.4% of customers who default on credit.



6.) Balance: Summary statistics of balance is as follows:

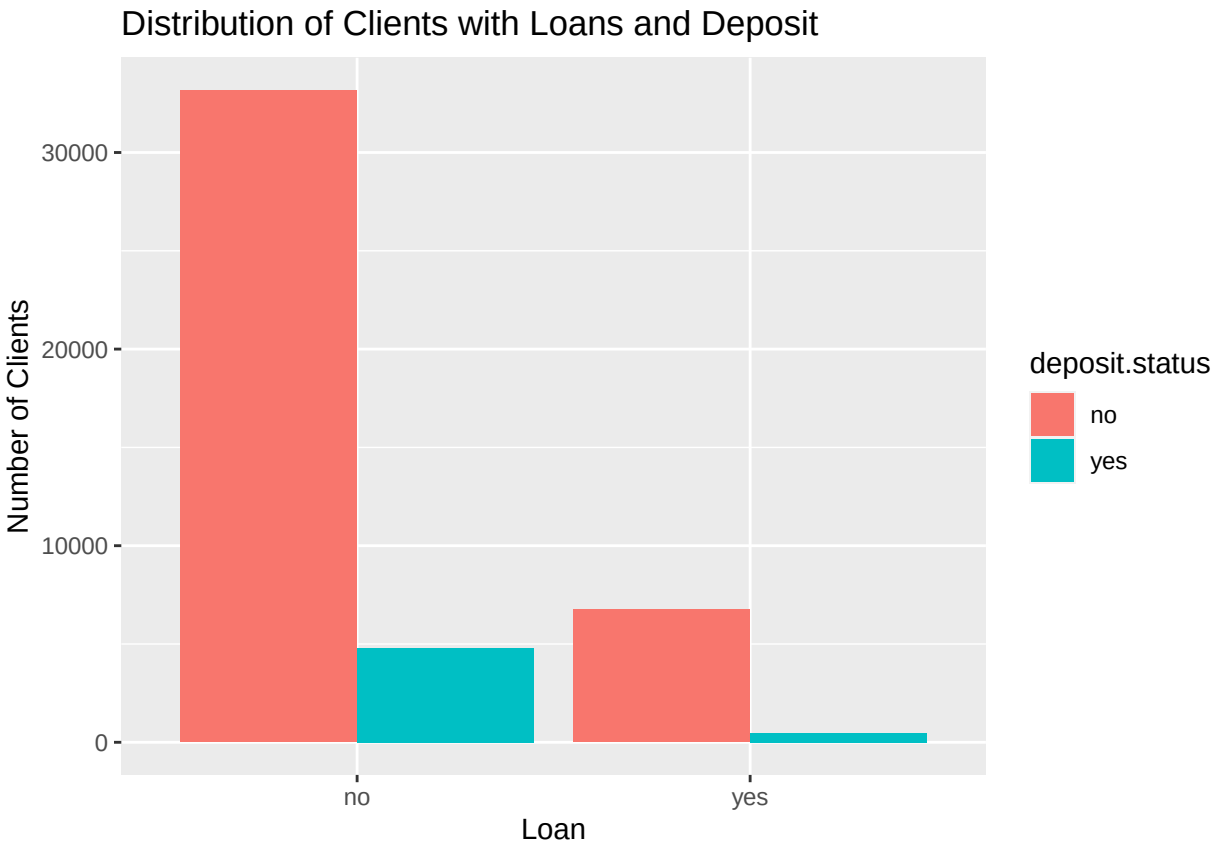
Minimum balance: -8019 Maximum balance: 102,127 25th percentile balance: 72 75th percentile balance: 1428 median balance: 448 mean: 1362

From the distribution plot we see that customers with a balance of 3000-6000 account for most customers who deposit money with the bank.



7.) Housing Loan 55.6% of the data do have housing loans, which leaves the remaining 44.4% of the data that does not have housing loans. Of those who had a housing loan, Only 36.6% have deposited with the bank, while of those who did not have a housing loan 63.4% have deposited with the bank. Looking at this we can say that those without a housing loan are more likely to deposit with the bank from the marketing channels.

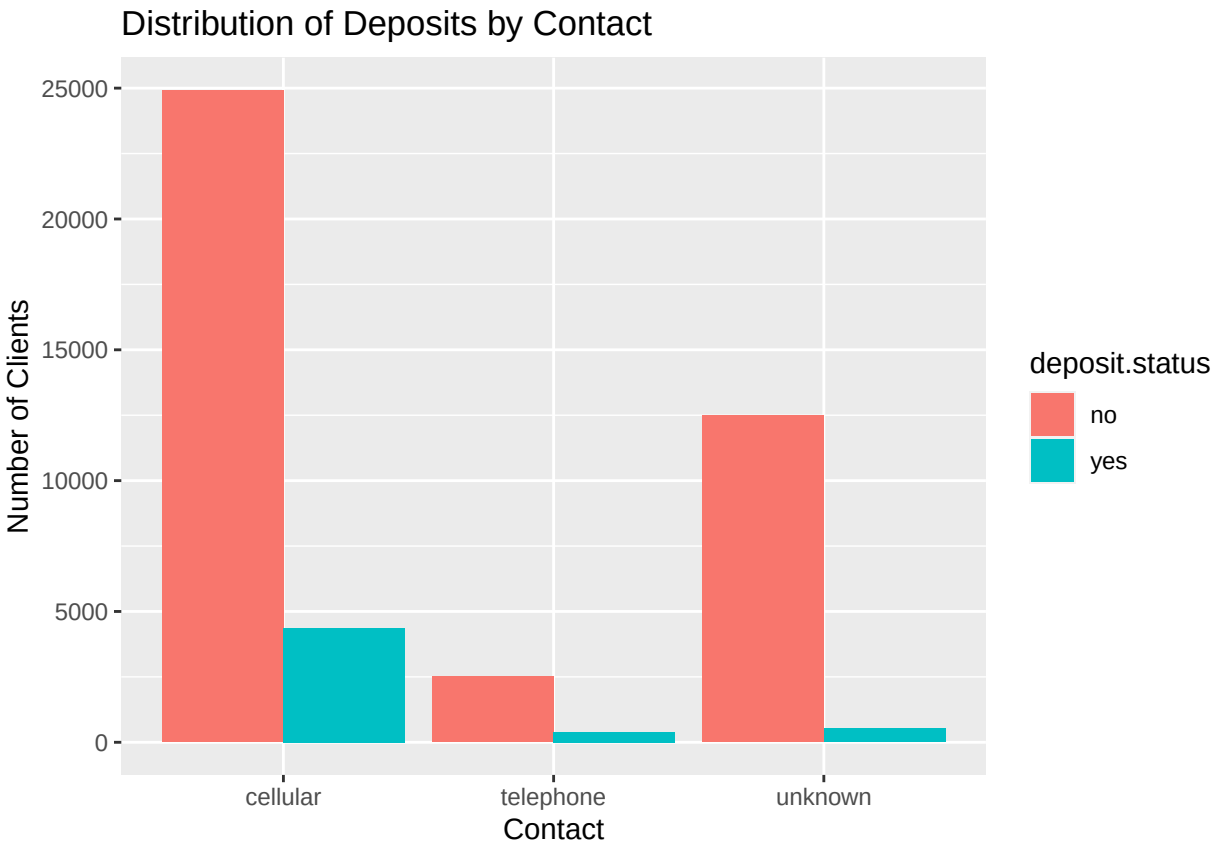
8.) Personal loan Looking at the data for personal loans we can observe that 6.7% of those who said they had a personal loan deposited with the bank while 12.7% percent of those who did not have a personal loan deposited with the bank. We may see this feature in our Logistic regression models as the difference between those that have a loan and do not have a loan is very apparent.



9.) Contact: Summary statistics for method of contact are as follows:

```
cellular  = 29 285 (64.8%)
telephone = 2 906 (6.4%)
unknown   = 13 020 (28.8%)
```

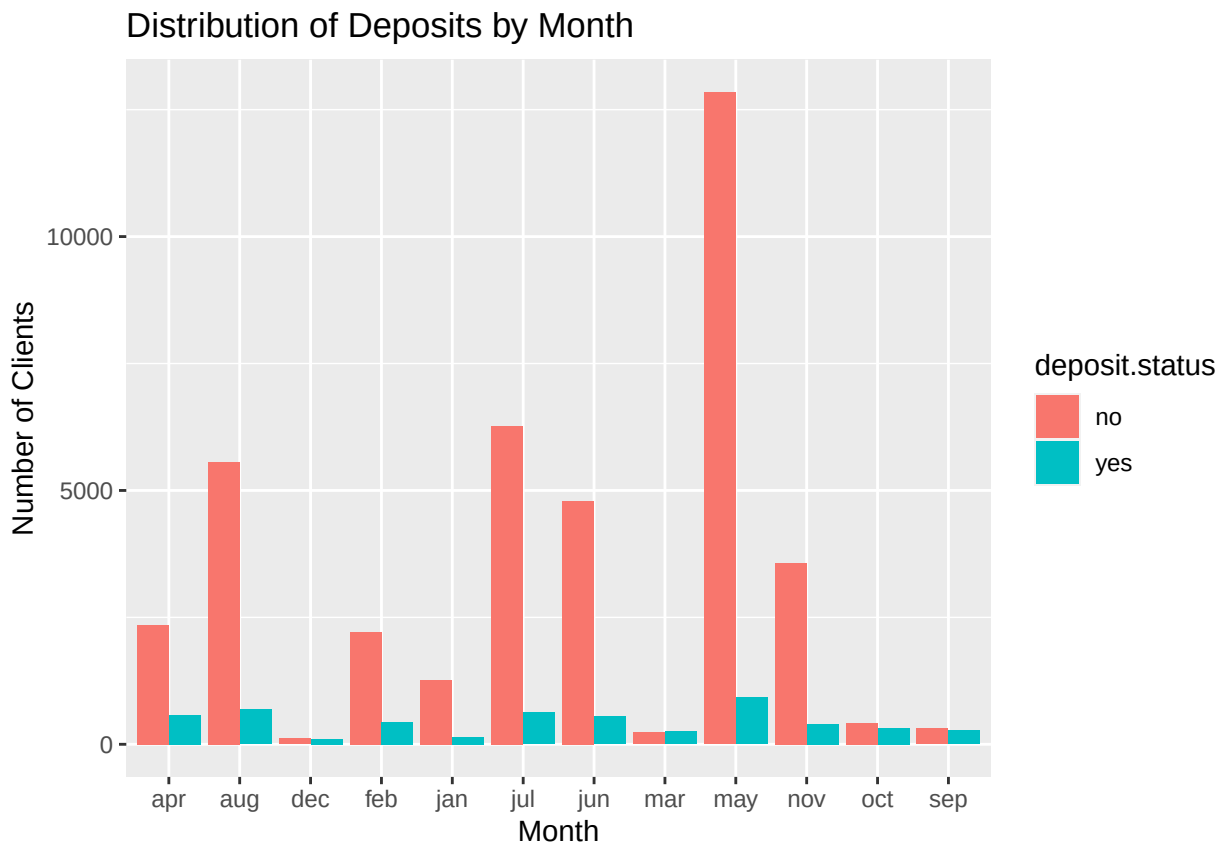
82.6% of customers contacted by cellular will end up making a deposit with the bank. The value for telephone contact is just 7.4%.



10.) Month: Summary Statistics for month of contact are as follows:

```
apr      = 2 932 (6.5%)
aug      = 6 247 (13.8%)
dec      = 214 (0.5%)
feb      = 2 649 (5.9%)
jan      = 1 403 (3.1%)
jul      = 6 895 (15.3%)
jun      = 5 341 (11.8%)
mar      = 477 (1.1%)
may      = 13 766 (30.4%)
nov      = 3 970 (8.8%)
```

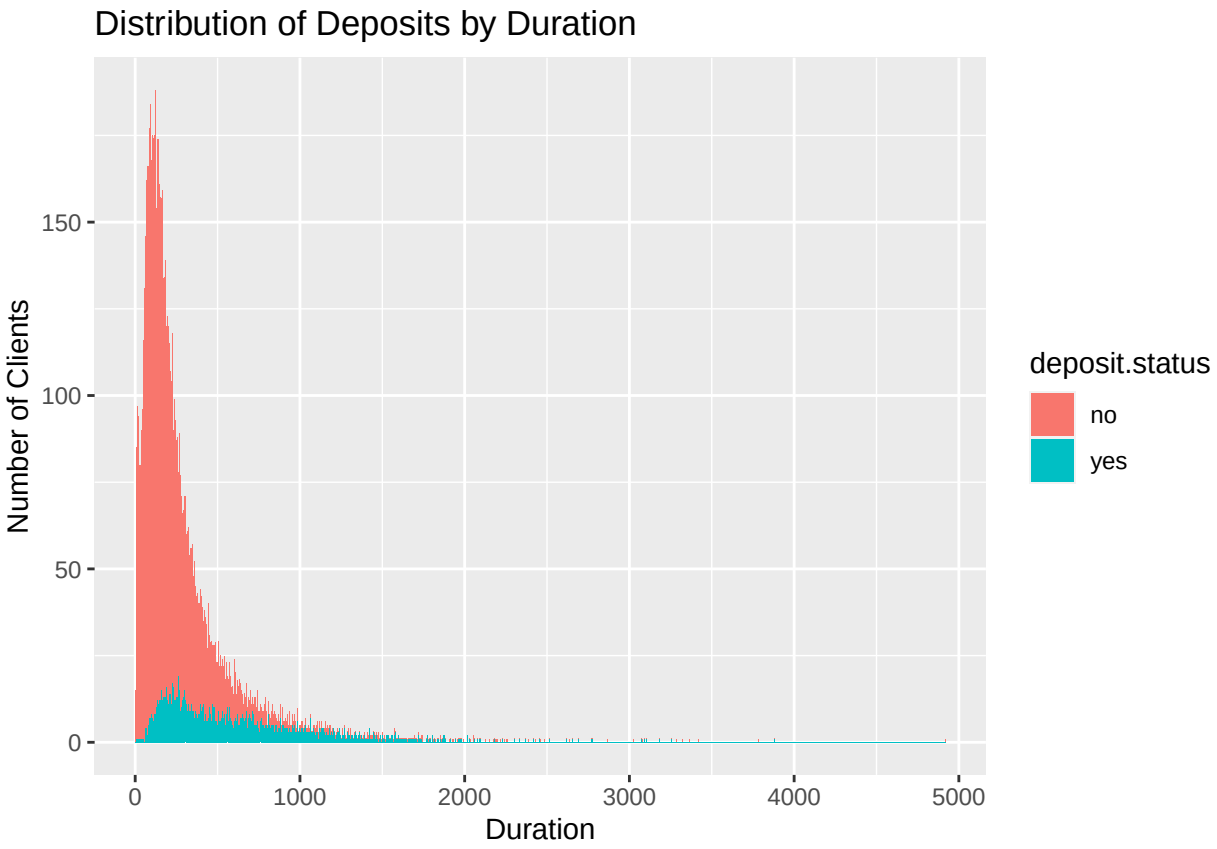
The month of May has the highest hit rate of customers making a deposit which is about 20%. However, this can be attributed to the higher volume of contact made.



11.) Duration: Summary statistics for contact duration are as follows:

```
min|max  = 0 | 4 918
q05|q95  = 35 | 751
q25|q75  = 103 | 319
median   = 180
mean     = 258.1631
```

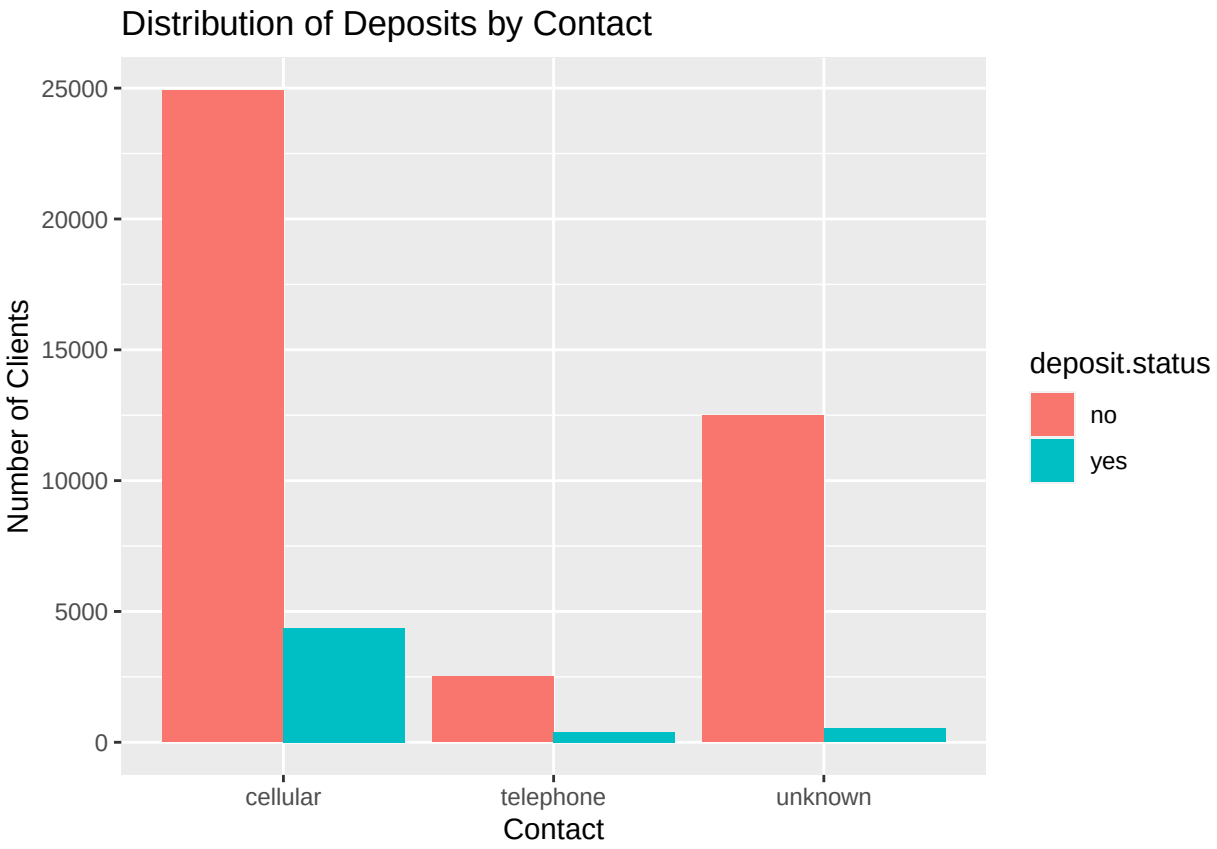
There is a likelihood of 59.8% of customers who are engaged on call for more than 1000 seconds to make a depsoit. The average call lasts for about 258 seconds.



12.) Contact: Summary statistics for number of times contacted are as follows:

```
min|max  = 1 | 63
q05|q95  = 1 | 8
q25|q75  = 1 | 3
median   = 2
mean     = 2.763841
```

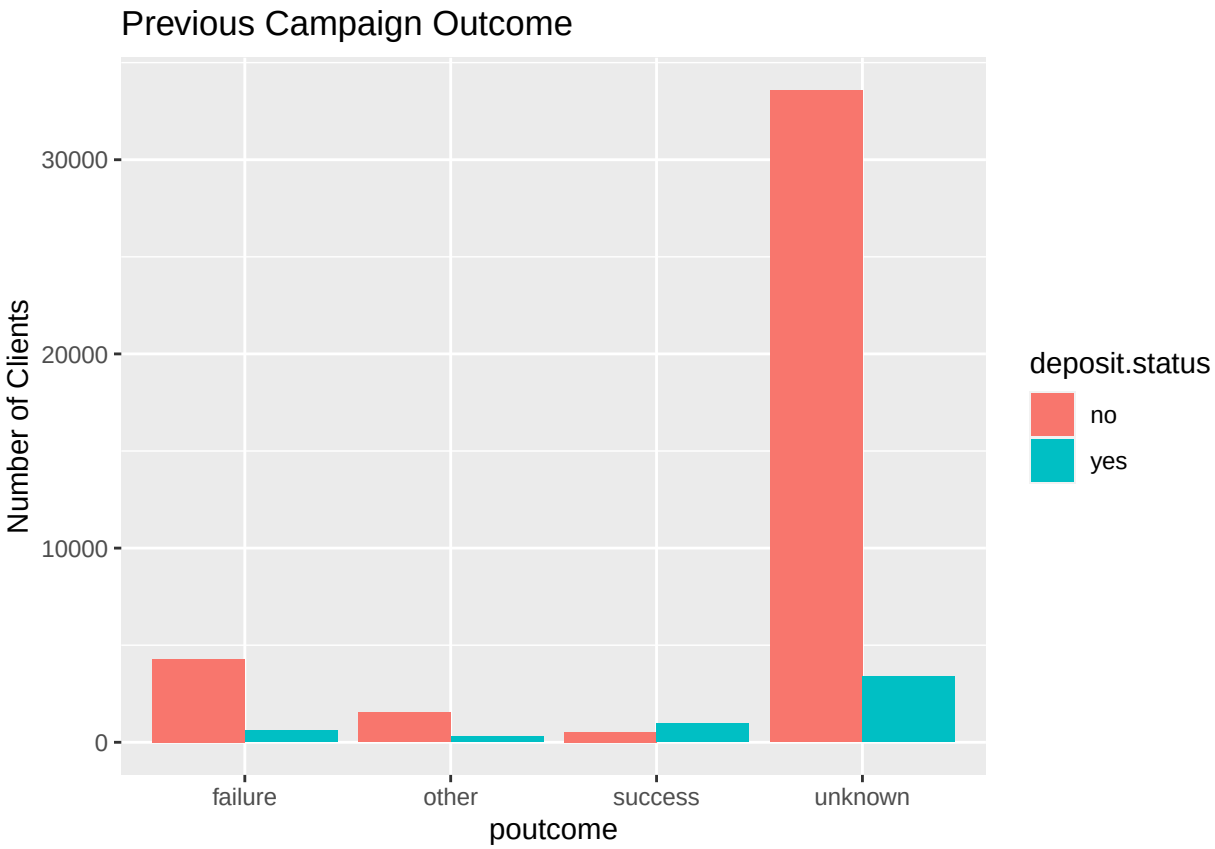
Customers with less than 3 times of contact are most likely to make a deposit.



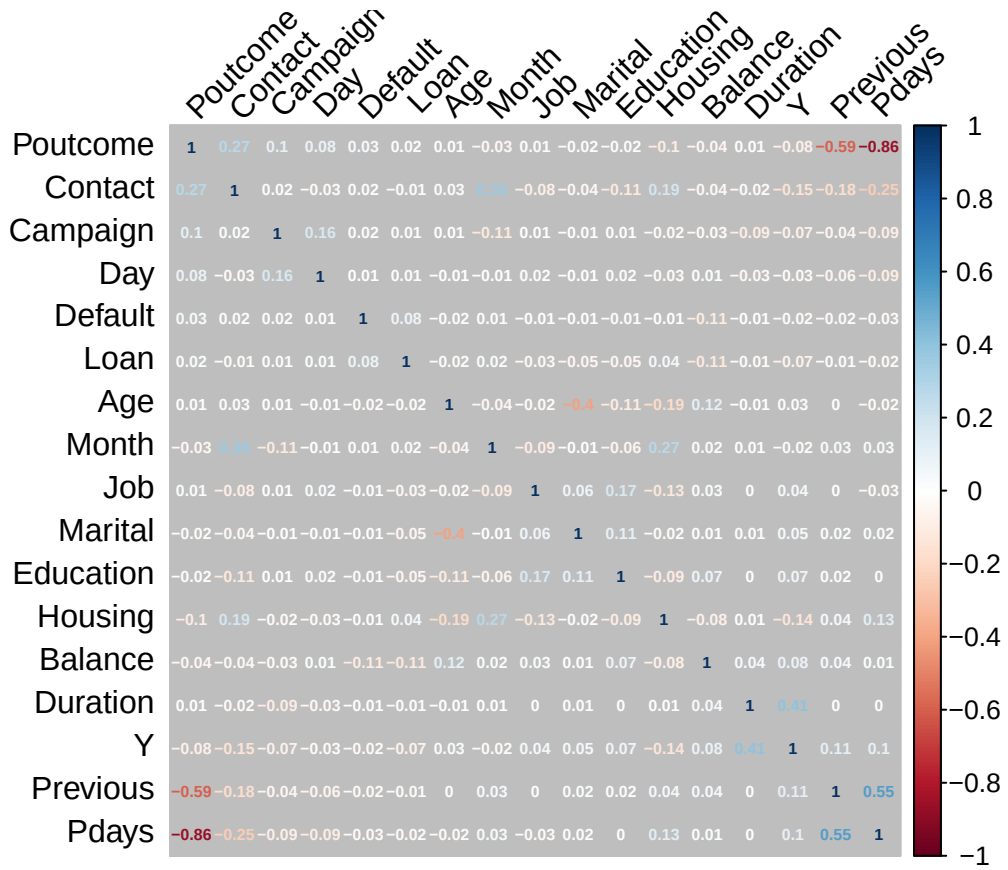
13.) Previous Outcome: Summary statistics for previous ourcome is as follows:

```
failure = 4 901 (10.8%)
 other   = 1 840 (4.1%)
 success = 1 511 (3.3%)
 unknown = 36 959 (81.7%)
```

The company needs to keep a better record of previous outcome since nearly 82% of previous outcomes are unknown. This can be useful for the bank to understand since 18.5% cases with a succesful previous outcome make a deposit.


Previous Campaign Outcome
Number of Clients
30000
20000
10000
0
failure
other
success
unknown
poutcome
deposit.status
no
yes

2.2.2 Correlation Analysis



The above correlogram is ordered in a first principle component manner. We can see that variables pday and previous contact method are highly correlated. Hence, we might be able to remove these two variables since they bring about multicollinearity to any model we might make in the predictions later. We can further test the variable selection by backward/forward selection in section 3 of the analysis.

2.2.3 Prediction Models

To answer our question of interest, building a predictive categorical model, we choose to utilize three different methods.

1. Logistic regression Particularly we are using different stepwise methods (backward, forward, both) and different selection criteria (Likelihood-Ratio Test, AIC, BIC) to conduct variable selection for building the most effective model.

We chose logistic regression as one of the modeling technique we would perform due to a couple of factors. It is relatively easy to implement on categorical data and the parameters can explain the significance of each predictor variable. However, we still do acknowledge some of the down sides to this modeling technique. Colinearity of the variables can affect the accuracy of the model and proper selection of features is required. However, those might be potentially addressed by our stepwise selection methods.

2. Random Forest The reason we chose random forest for this predictive model was because random forest is much better than logistic regression at handling multi-colinearity. Random Forest is a collection of decision trees and average/majority vote of the forest is selected as the predicted output. It is less prone to

overfitting than regular Decision trees, and gives a more generalized solution (Varghese, Comparative study on Classic Machine learning Algorithms 2019). It is also usually more robust and accurate than decision trees [2]. The random forest model can also deal with categorical data better than the logistic regression can. It usually comes up with a robust and accurate model that can handle large varieties of input data with binary, categorical, and continuous features, which we do have in this bank dataset. While Random Forest can give a probability over the prediction and supports implicit feature selection and derives feature importance, it is not a well descriptive model over the prediction [3] (Varghese, Comparative Study on Classic Machine learning Algorithms , Part-2 2018). A logistic regression model does a better job deriving the significance of each of the predictor variables which would give us better incites on the data.

[2] Varghese, D. (2019, May 10). Comparative study on Classic Machine learning Algorithms. Retrieved December 14, 2020, from https://towardsdatascience.com/comparative-study-on-classic-machine-learning-algorithms-24f9ff6ab222

[3] Varghese, D. (2018, December 11). Comparative Study on Classic Machine learning Algorithms , Part-2. Retrieved December 14, 2020, from https://medium.com/@dannymvarghese/comparative-study-on-classic-machine-learning-algorithms-part-2-5ab58b683ec0

3. SVM (Extra Method) Support Vector Machine(SVM) is a supervised learning model that uses classification algorithms for group classification problems. An SVM finds an optimal hyperplane as the solution to the learning problem.

Added in the end for extra credit. See below sections for more information on the analysis and comparison of SVM against Logistic and Random Forest.

Finally, we will compare these and see how the models compare to each other, and select a ultimate winner between the two. Our criteria for selecting the model will be based on the accuracy of the model and the AUC of the model.

[4] Evgeniou, Theodoros & Pontil, Massimiliano. (2001). Support Vector Machines: Theory and Applications. 2049. 249-257. 10.1007/3-540-44673-7_12.

3. Extra-Credit Methods

First Method: Correlogram & Explore Function

Correlogram for preliminary variable selection and the explore function for interactive EDA.

Second Method: Stepwise Variable Selection for Logistic Regression

We have used (Forward, Backward, Both) Variable Selection using AIC, BIC, and Likelihood Ratio Test (LRT) for our Logistic Regression Model. Specifically, we have computed results for:

1. LRT Forward - at 5% significance level, dropped variables *age*, *default*, *pdays*, *previous*.
2. LRT Backward - same dropped variables as above.
3. AIC Forward - dropped variables *previous*, *pdays*, *default*, *age*.
4. AIC Backward - same dropped variables as above.
5. AIC Both - same dropped variables as above.
6. BIC Forward - dropped variables *previous*, *default*, *age*, *pdays*, and *job*
7. BIC Backward - same dropped variables as above.

8. BIC Both - same dropped variables as above.

Note, we tried adding interaction variables but it was too computationally complicated for R and our dataset, and leaded to stack overflow. Ultimately, we see that all all 3 stepwise selections (backward, forward, both) using AIC yield the exact same model. We have dropped pdays, default, age, and previous. Our final logistic regression model would be:

y ~ duration + poutcome + month + contact + housing + job + campaign + loan + marital + day + education + balance

The reason we think AIC is the most robust and appropriate and do not use LRT nor BIC is because firstly, a Likelihood Ratio Test only provides a "heauristic model selection," meaning it's main purpose is not for variable selection but just a quick and convinient way of selecting variables. We chose to select variables at a 5% significance level, but this selection is in fact trivial. If we chose a different alpha level such as 1% or 10%, our variables would be different. Therefore, it is an inferior choice as compared to AIC and BIC.

While AIC and BIC are both robust parameters, for the purposes of answering this statistical question of interest, we choose AIC over BIC. BIC usually penalizes complicated models more than AIC. Naturally, AIC would create a more complicated model (with more predictors) yet it is actually better for prediction. The Akaike information criterion is known to be efficient in the sense that its prediction performance is asymptotically equivalent to the best offered by the candidate models (Ding, Tarokh, & Yang, Bridging AIC and BIC: a new criterion for autoregression 2016)[4]. When the true model is not in the candidate model set the AIC is efficient, in that it will asymptotically choose whichever model minimizes the mean squared error of prediction/estimation. The BIC is not efficient under these circumstances (Vrieze, Model selection and psychological theory: a discussion of the differences between the Akaike information criterion (AIC) and the Bayesian information criterion (BIC) 2012) [5]. Models built using BIC would be less complicated, which would be better for inference, but worse than AIC in terms of predictive performance, which is contradicting our main goal here.

[5] Ding, J., Tarokh, V., & Yang, Y. (2016, August 24). Bridging AIC and BIC: A new criterion for autoregression. Retrieved December 13, 2020, from https://arxiv.org/abs/1508.02473

[6] Vrieze, S. (2012, June). Model selection and psychological theory: A discussion of the differences between the Akaike information criterion (AIC) and the Bayesian information criterion (BIC). Retrieved December 13, 2020, from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3366160/

Third Method: Support Vector Machine(SVM) Method

Added in the end for extra credit. See below sections for more information on the analysis and comparison of SVM against Logistic and Random Forest.

Finally, we will compare these and see how the models compare to each other, and select a ultimate winner between the two. Our criteria for selecting the model will be based on the accuracy of the model and the AUC of the model.

4. Results

Method 1. Logistic Regression & Model Selection

Here is a summary of the logistic regression model with all 16 of the input variables. However, we would like to start with stepwise variable selection first to prune down this model. We will then interpret that final pruned model.

```
##
```

```
## Call:
## glm(formula = as.factor(y) ~ ., family = binomial(link = "logit"),
##     data = lr.bank.data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.9357  -0.3735  -0.2524  -0.1483   3.3580
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -2.580e+00  2.055e-01 -12.557  < 2e-16 ***
## age                -1.169e-03  2.471e-03  -0.473 0.636205
## jobblue-collar     -2.791e-01  8.098e-02  -3.447 0.000568 ***
## jobentrepreneur    -2.874e-01  1.382e-01  -2.080 0.037535 *
## jobhousemaid       -4.974e-01  1.525e-01  -3.262 0.001106 **
## jobmanagement      -1.509e-01  8.185e-02  -1.844 0.065157 .
## jobretired          3.026e-01  1.095e-01   2.764 0.005716 **
## jobself-employed   -3.109e-01  1.258e-01  -2.472 0.013427 *
## jobservices        -2.383e-01  9.439e-02  -2.525 0.011575 *
## jobstudent          3.788e-01  1.229e-01   3.083 0.002047 **
## jobtechnician      -1.829e-01  7.694e-02  -2.377 0.017438 *
## jobunemployed      -1.674e-01  1.235e-01  -1.355 0.175428
## jobunknown         -5.370e-01  2.733e-01  -1.965 0.049415 *
## maritalmarried     -2.095e-01  6.564e-02  -3.191 0.001416 **
## maritalsingle       5.311e-02  7.500e-02   0.708 0.478862
## educationsecondary  2.204e-01  7.299e-02   3.020 0.002531 **
## educationtertiary   3.960e-01  8.472e-02   4.674 2.95e-06 ***
## educationunknown    2.834e-01  1.167e-01   2.429 0.015131 *
## defaultyes         -1.001e-01  1.848e-01  -0.542 0.587900
## balance             1.802e-05  5.693e-06   3.164 0.001554 **
## housingyes         -6.757e-01  4.896e-02 -13.802  < 2e-16 ***
## loanyes            -4.421e-01  6.703e-02  -6.595 4.26e-11 ***
## contacttelephone   -1.684e-01  8.501e-02  -1.980 0.047653 *
## contactunknown     -1.678e+00  8.237e-02 -20.372  < 2e-16 ***
## day                 9.040e-03  2.805e-03   3.223 0.001268 **
## monthaug           -6.987e-01  8.762e-02  -7.973 1.54e-15 ***
## monthdec            6.788e-01  1.982e-01   3.424 0.000617 ***
## monthfeb           -1.192e-01  1.000e-01  -1.191 0.233568
## monthjan           -1.279e+00  1.379e-01  -9.277  < 2e-16 ***
## monthjul           -7.959e-01  8.581e-02  -9.275  < 2e-16 ***
## monthjun            5.186e-01  1.046e-01   4.959 7.09e-07 ***
## monthmar            1.517e+00  1.372e-01  11.055  < 2e-16 ***
## monthmay           -3.801e-01  8.059e-02  -4.716 2.40e-06 ***
## monthnov           -8.524e-01  9.386e-02  -9.082  < 2e-16 ***
## monthoct            8.668e-01  1.213e-01   7.145 9.00e-13 ***
## monthsep            9.407e-01  1.320e-01   7.127 1.03e-12 ***
## duration            4.242e-03  7.245e-05  58.548  < 2e-16 ***
## campaign           -8.254e-02  1.110e-02  -7.437 1.03e-13 ***
## pdays              -5.715e-06  3.373e-04  -0.017 0.986482
## previous            8.936e-03  6.514e-03   1.372 0.170126
## poutcomeother       2.593e-01  1.001e-01   2.591 0.009562 **
## poutcomesuccess     2.318e+00  9.200e-02  25.193  < 2e-16 ***
## poutcomeunknown    -2.922e-02  1.036e-01  -0.282 0.777845
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 26136  on 36168  degrees of freedom
## Residual deviance: 17212  on 36126  degrees of freedom
## AIC: 17298
##
## Number of Fisher Scoring iterations: 6
```

1. Backward Stepwise Model Selection Using LRT

Perform feature selection using likelihood ratio test (comparing a certain coefficient vs. it to be zero) to prune down model at alpha = 0.05. We find the largest p value at each step to drop the variable. After 5 steps, we have dropped age, default, pdays, previous from our model. All variables at the fifth run have p value significant at 5% significance level, so we stop our test. The final model is

y ~ job + marital + education + balance + housing + loan + contact + day + month + duration + campaign + poutcome.

2. Forward Stepwise Model Selection Using LRT

Perform feature selection using likelihood ratio test (comparing a certain coeffeicient vs. it to be zero) to prune down model at alpha = 0.05. We find the smallest p value at each step to add the variable. After 12 steps, we have added job, marital, education, housing, loan, contact, day, month, duration, campaign, and poutcome to our model. All variables at the thirdteenth run have p value insignificant at 5% significance level, so we stop our test. The final model is (y) ~ duration + poutcome + month + contact + housing + job + campaign + loan + marital + education + day + balance. We have left out age, default, pdays, previous.

Not surprisingly, backward and forward stepwise using LRT yield same results. We have excluded age, default, pdays, and previous from our model, and the model selected from LRT is

y ~ job + marital + education + housing + loan + contact + day + month + duration + campaign + poutcome + balance.

3. Forward Selection Using AIC - (Less penalize on larger models, so creates more complicated model, but better predictive performance)

According to this criteria, we excluded previous, pdays, default, and age. Our model is

y ~ duration + poutcome + month + contact + housing + job + campaign + loan + marital + education + day + balance.

4. Forward Selection Using BIC - (More penalize on larger models, builds simpler models, less complicated and better for inference)

According to this criteria, we excluded previous, default, age, pdays, and job. Our model is

y ~ duration + poutcome + month + contact + housing + campaign + loan + marital + education + balance + day.

5. Backward Selection Using AIC

According to this criteria, we excluded previous, pdays, default, and age. Our model is

y ~ duration + poutcome + month + contact + housing + job + campaign + loan + marital + education + day + balance. Which is the same as Forward Selection using AIC.

6. Backward Selection Using BIC

According to this criteria, we excluded previous, default, age, pdays, job. Our model is

y ~ marital + education + balance + housing + loan + contact + day + month + duration + campaign + poutcome. Which is the same as Forward Selection using BIC.

7. Both Directions Using AIC

Here, we exclude pdays, default, age, and previous. Our model is

y ~ duration + poutcome + month + contact + housing + job + campaign + loan + marital + day + education + balance.

8. Both Directions Using BIC

Here, we exclude previous, default, age, pdays, and job. Our model is

y ~ duration + poutcome + month + contact + housing + campaign + loan + marital + education + balance + day.

Building a Confusion Matrix with our final logistic regression model

Again, our final model is the one with variables selected from AIC (reason explained in previous section).

y ~ duration + poutcome + month + contact + housing + job + campaign + loan + marital + day + education + balance

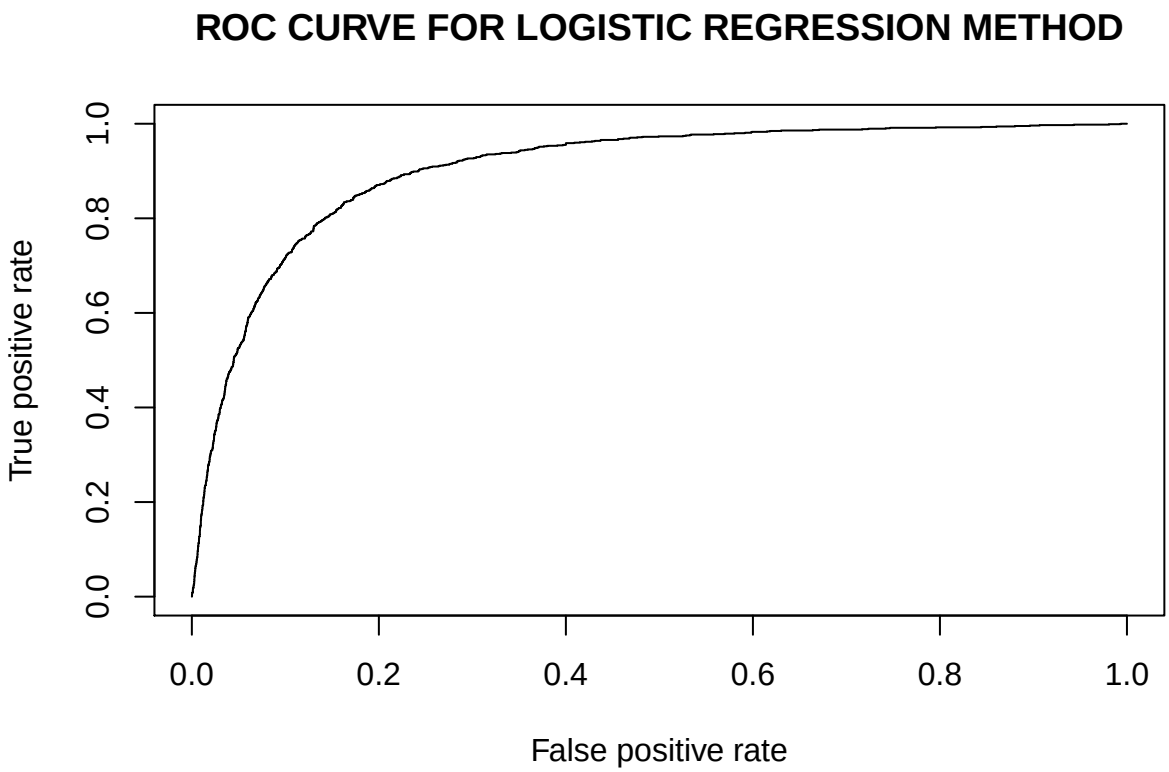
Here is a brief summary of our final logistic model. The stars on the very right column indicates the significance of the variable. Here, R automatically created dummy variables for categorical variables with more than one category. As we can see, many of the variables that are left after variable selection are significant, with only a few insignificant variables (those which have a blank on the very right column, for instance poutcomeunknown and maritalsingle.) Furthermore, to briefly interpret the estimate of each variable, it means each one unit change in (the variable) will increase the log odds of our response variable by (the estimate). For instance, each one unit change in duration will increase the log odds of subscribing by 4.242e-03, and with the p-value and three stars on the very right, we know that it is somewhat significant in determining our response. Estimates with a positive value would be associated with y = 1 (yes) and estimates with a negative value vice versa. For instance, more likely to subscribe in December than in August.

Additionally, the difference between null deviance and residual deviance tells us that the model is a good fit. Greater the difference better the model. Here, we have somewhat a good model, as null deviance is the value when we only have intercept in our equation with no variables, and residual deviance is the value when we take all of our variables into account.

```
##
## Call:
## glm(formula = y ~ duration + poutcome + month + contact + housing +
##     job + campaign + loan + marital + day + education + balance,
```

```
##     family = binomial(link = "logit"), data = lr.bank.data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.9318  -0.3736  -0.2524  -0.1483   3.3539
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -2.614e+00  1.457e-01 -17.943  < 2e-16 ***
## duration            4.242e-03  7.245e-05  58.554  < 2e-16 ***
## poutcomeother       2.706e-01  9.951e-02   2.719 0.006550 **
## poutcomesuccess     2.319e+00  8.916e-02  26.015  < 2e-16 ***
## poutcomeunknown    -5.336e-02  6.436e-02  -0.829 0.407074
## monthaug           -6.991e-01  8.753e-02  -7.987 1.38e-15 ***
## monthdec            6.821e-01  1.982e-01   3.442 0.000578 ***
## monthfeb           -1.166e-01  9.987e-02  -1.168 0.242835
## monthjan           -1.277e+00  1.378e-01  -9.262  < 2e-16 ***
## monthjul           -7.951e-01  8.573e-02  -9.275  < 2e-16 ***
## monthjun            5.204e-01  1.045e-01   4.978 6.43e-07 ***
## monthmar            1.518e+00  1.371e-01  11.071  < 2e-16 ***
## monthmay           -3.785e-01  8.053e-02  -4.700 2.61e-06 ***
## monthnov           -8.525e-01  9.333e-02  -9.134  < 2e-16 ***
## monthoct            8.676e-01  1.212e-01   7.158 8.18e-13 ***
## monthsep            9.421e-01  1.320e-01   7.138 9.46e-13 ***
## contacttelephone   -1.717e-01  8.409e-02  -2.042 0.041122 *
## contactunknown     -1.680e+00  8.230e-02 -20.411  < 2e-16 ***
## housingyes         -6.718e-01  4.852e-02 -13.846  < 2e-16 ***
## jobblue-collar     -2.798e-01  8.091e-02  -3.459 0.000543 ***
## jobentrepreneur    -2.933e-01  1.381e-01  -2.125 0.033615 *
## jobhousemaid       -5.049e-01  1.520e-01  -3.320 0.000899 ***
## jobmanagement      -1.526e-01  8.176e-02  -1.867 0.061947 .
## jobretired          2.783e-01  9.832e-02   2.831 0.004644 **
## jobself-employed   -3.140e-01  1.257e-01  -2.497 0.012515 *
## jobservices        -2.390e-01  9.436e-02  -2.533 0.011314 *
## jobstudent          3.904e-01  1.207e-01   3.234 0.001219 **
## jobtechnician      -1.835e-01  7.693e-02  -2.385 0.017096 *
## jobunemployed      -1.692e-01  1.235e-01  -1.370 0.170747
## jobunknown         -5.425e-01  2.730e-01  -1.987 0.046941 *
## campaign           -8.217e-02  1.109e-02  -7.413 1.24e-13 ***
## loanyes            -4.438e-01  6.686e-02  -6.638 3.18e-11 ***
## maritalmarried     -2.050e-01  6.535e-02  -3.138 0.001702 **
## maritalsingle       6.591e-02  7.035e-02   0.937 0.348782
## day                 8.970e-03  2.804e-03   3.198 0.001382 **
## educationsecondary  2.229e-01  7.259e-02   3.071 0.002132 **
## educationtertiary   4.009e-01  8.398e-02   4.774 1.81e-06 ***
## educationunknown    2.815e-01  1.167e-01   2.413 0.015807 *
## balance             1.795e-05  5.669e-06   3.166 0.001544 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 26136  on 36168  degrees of freedom
## Residual deviance: 17214  on 36130  degrees of freedom
```

```
## AIC: 17292
##
## Number of Fisher Scoring iterations: 6
```



Here are our AUC - ROC plots. AUC - ROC curve is a performance measurement for classification models. ROC is a probability curve and AUC represents degree or measure of separability. It tells how much model is capable of distinguishing between classes. Higher the AUC, the better the model accuracy.

The ROC and corresponding Area Under Curve (AUC): 0.9038654

Therefore, our model achieved also 90% of accuracy as well, signifying that we have successfully pruned down the variables that are important.

The confusion matrix is,

```
##
## pred    0    1
##    0 7797  689
##    1  195  361
```

and the corresponding classification accuracy is 0.902234

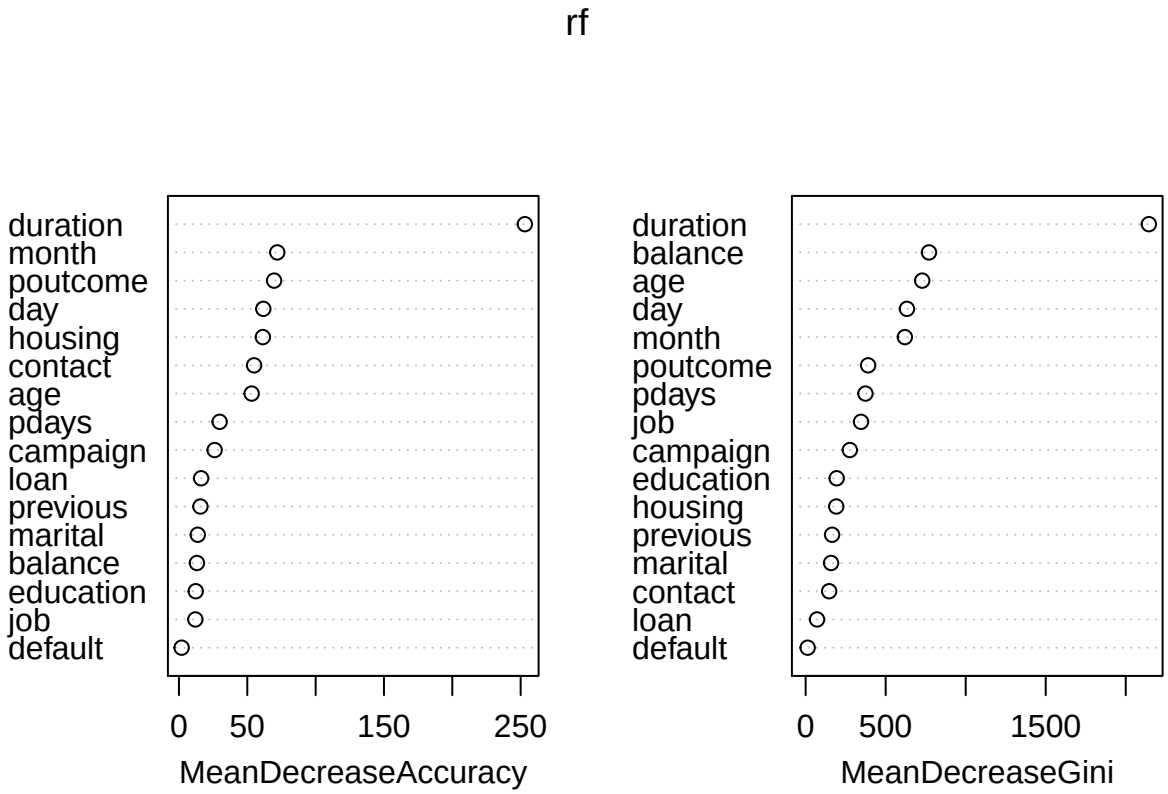
This is a very robust classification accuracy since it correctly for roughly 90% of our classifications.

Now, we will move on to our second method, Random Forest and see how it compares.

Method 2. Random Forest

The second method that we choose is Random Forest. A Random Forest is a collection of decision trees and average/majority vote of the forest is selected as the predicted output. The reason that we choose it is because a Random Forest model will be less prone to over fitting than a decision tree, and gives a more generalized solution. Random Forest is more robust and accurate than decision trees.

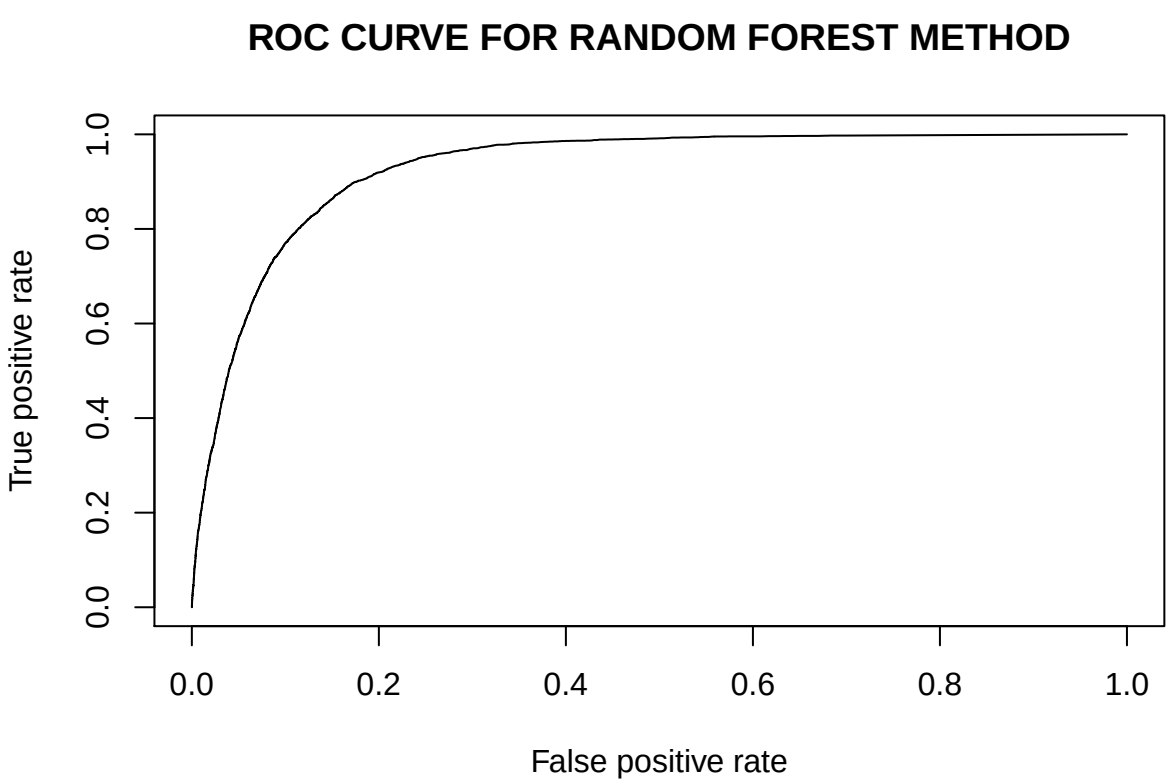
Particularly, we would first tune the model so that we find our ideal amount of ntree and mtry (Use mtry from 1 to p and then use CV to find minimize OOB error). Typically, the rule of thumb is to choose sqrt(p) for classification RF's for mtry. Then, we will investigate the importance of variables in our model.



We used a manual tuning method (comparing between common choices 100, 200, 300, 400, and 500) to find the optimal ntree value for our random forest model, which ended up being 500. The OOB for ntree = 500 was the lowest at 9.45% as well as the confusion matrix classification error was also the lowest compared to the values for the other candidate ntree values.

As for mtry, we have ran the tuneRF function, which yielded an optimal result of 4. This corresponds to our rule of thumb of sqrt(16) which equals to 4.

Then, we built our RF using such parameters. From the importance plots, we see that duration is the most important (has the most predictive power) according to both "MeanDecreaseAccuracy" and "MeanDecreaseGini" whereas default are the worst in both cases. This somewhat corresponds with our results obtained from the stepwise variable seleciton process.

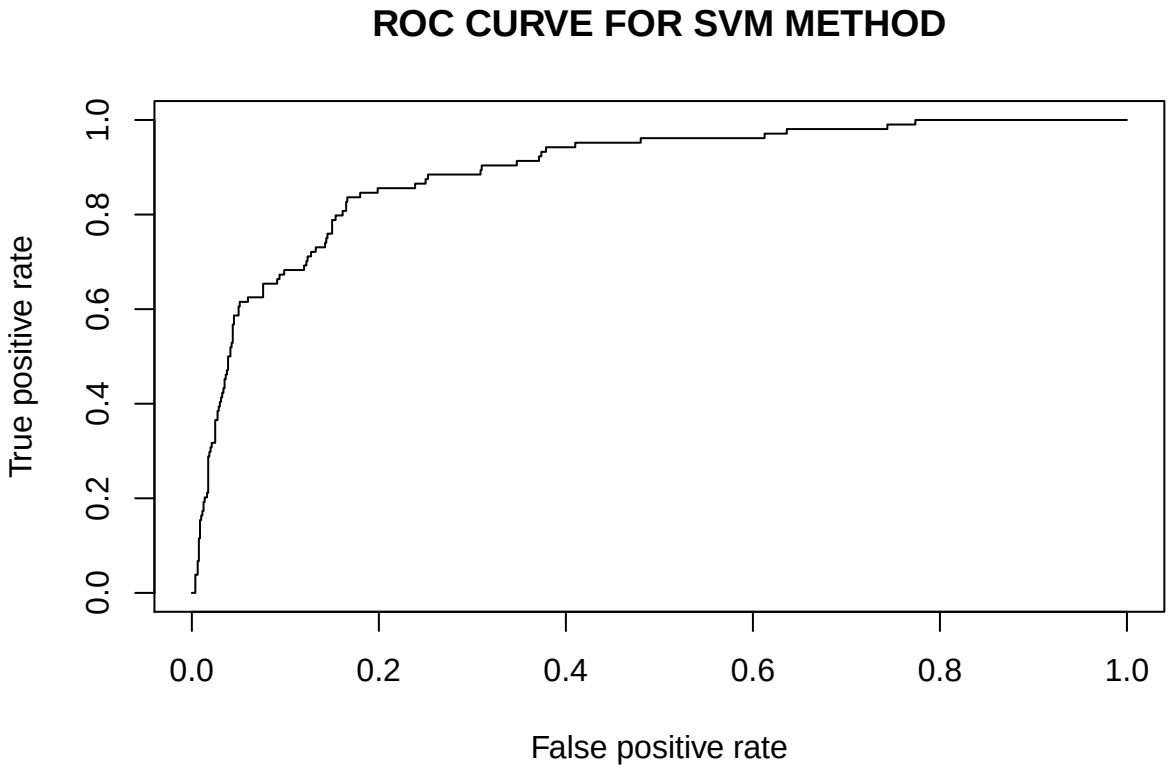


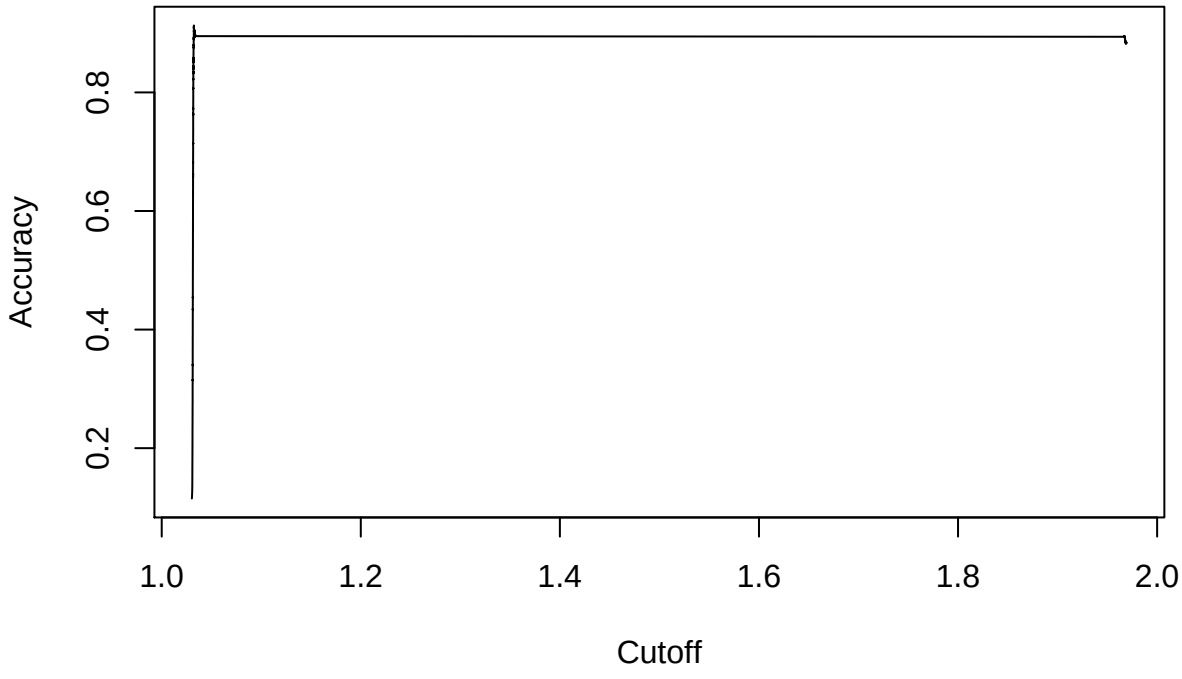
Furthermore, the corresponding confusion matrix is below, and the corresponding classification accuracy is 0.906326, very close to what we get in our logistic model after AIC variable selection.

```
##
## prediction_rf    0    1
##             0 7765  620
##             1  227  430
```

However, the AUC that we computed from the ROC plot had a rather significant increase to 0.9278294. This is almost 3% higher than what we have obtained from our logistic model. Now, we will move on to SVM and see how it compares.

Method 3. Support Vector Machine (SVM) - Extra Method





Support Vector Machine (SVM) Interpretation

Support Vector Machine is a very popular classification technique used in Machine Learning. It is widely used due to having a significant accuracy. However, this type of model is commonly used for relatively smaller datasets. We mainly decided to use SVM becuase of the significant accuracy that is usually obtained from this model.

For the SVM model, we decided to use a relatively smaller subset of the actual dataset. This was because it is computationally costly to run SVM and since this was a very huge dataset, it made more sense to use the subset of the dataset which we named "bank_small". The bank_small dataset is a 10% sample of the larger bank_full dataset. While using the bank_full dataset with a radial kernel, we found that it was taking extremely long for the process to finish. So, we split the bank_small model into a 80-20 format (train/test) and changed the kernel type to linear since our data is largely linear. The process finished fairly quickly after that.

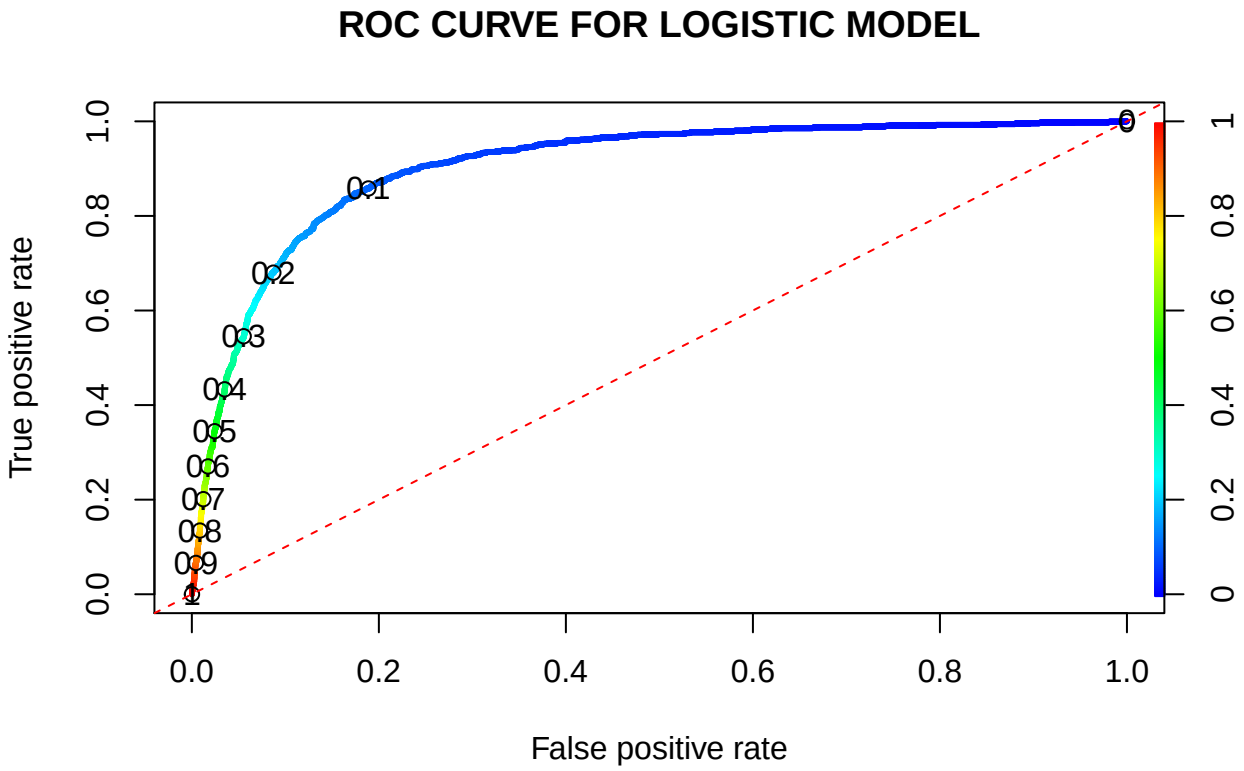
The accuracy that we obtained from running this model on the "bank_small" is 0.8938053. We can see that this is less than the accuracy of the logistic regression model that we trained which was 0.902234. Additionally, we found that it was also less than the classification accuracy of the random forest model which is 0.906326. The reduced accuracy of the SVM model could be a result of using a smaller dataset (10% sample of full dataset) when compared to logistic regression and random forest methods.

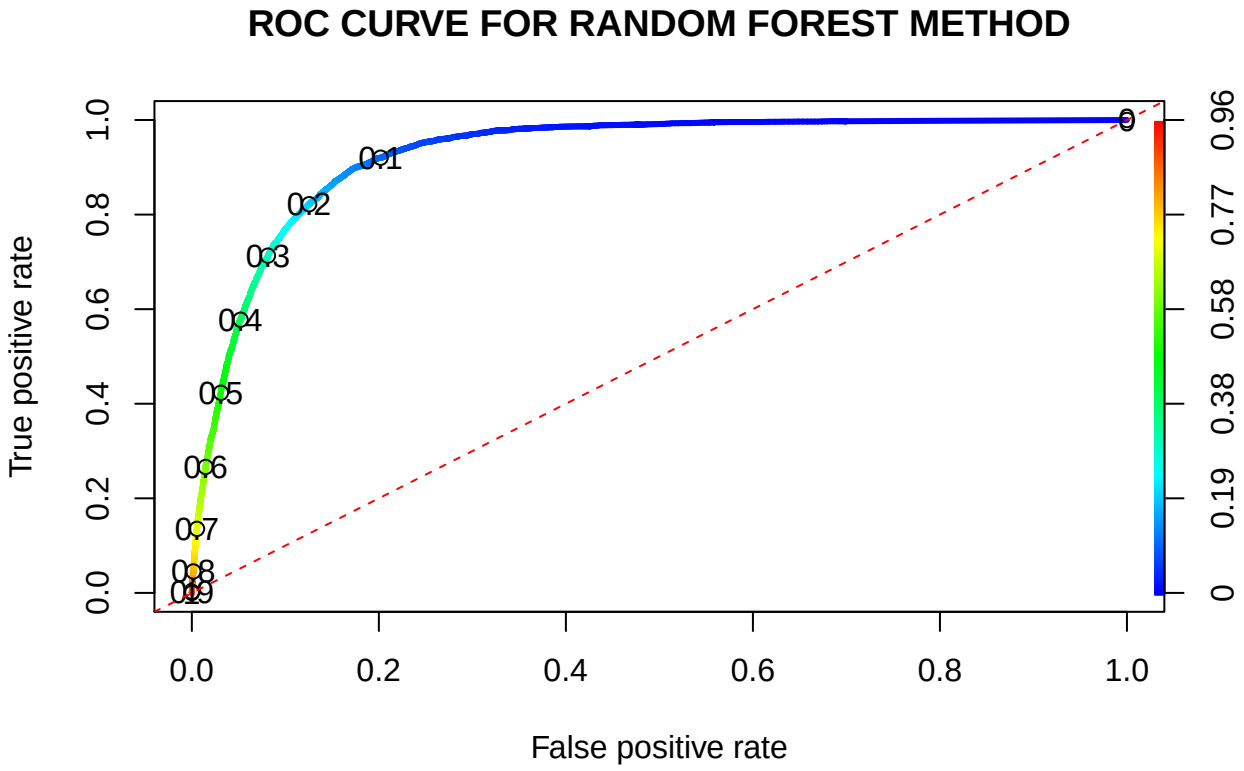
Moreover, from doing additional computation, we found out the Area Under the Curve for the ROC curve for each of the models. This area told us how much each of the models is capable of distinguishing between each of the classes. The AUC we got for the SVM model is 0.8915625 as compared to the Logistic Regression AUC which was 0.9038654 and Random Forest 0.9278294.

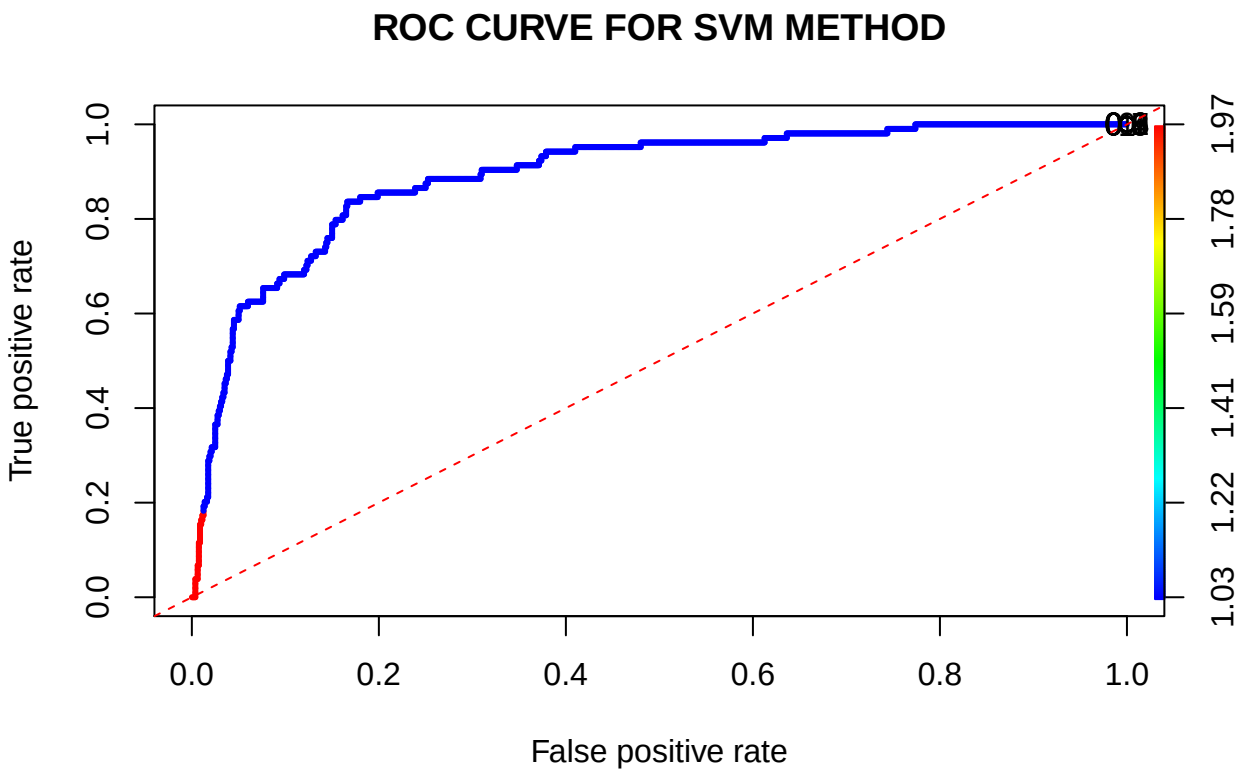
By looking at all these scores and comparing them with the different model that we had, we came to a conclusion that for this particular dataset, SVM is not the best model.

Final Comparison between Logistic Regression, Random Forest, and SVM

Method	Accuracy	AUC
Random Forest	0.9063260	0.9278294
Logistic Regression	0.9022340	0.9038654
SVM	0.8938053	0.8915625


ROC CURVE FOR LOGISTIC MODEL
True positive rate
False positive rate
0.0
0.2
0.4
0.6
0.8
1.0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
2


ROC CURVE FOR RANDOM FOREST METHOD
True positive rate
False positive rate
0.0
0.2
0.4
0.6
0.8
1.0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
0.96
0.77
0.58
0.38
0.19



After comparing all of the models by checking the accuracy as well as the AUC, we conclude that the random forest model is most suitable for the prediction model that is required to predict clients that will make a deposit and those who will not make a deposit. The random forest model and the logistic regression model have very similar accuracies with the random forest model having an accuracy of approximately 0.904 while the logistic regression model having an accuracy of approximately 0.902. This means that the random forest method has better predictive capabilities than the logistic regression model. We can also compare the AUC's of the model and see that the logistic regression model has a lower AUC than the random forest model. The logistic regression model has an AUC of 0.904 and the random forest model has an AUC of 0.926 which makes the random forest model better. Therefore, we will choose the Random Forest Model to use for our prediction model.

5. Code Appendix

```
#setup
set.seed(123)
library(readr)
library(tidyverse)
library(fastDummies)
library(knitr)
library(plyr)
library(dplyr)
library(explore)
library(corrplot)
library(ROCR)
```

```
library(cutpointr)
library(caret)
library(randomForest)
require(caTools)
library(rpart)
library(pROC)
library(ROCR)
library(e1071)



#import dataset
bank.data <- read_delim("datasets/bank-full.csv",";", escape_double = FALSE, trim_ws = TRUE)
# small datset for computationaly intensive tasks
bank_small <- read_delim("datasets/bank.csv", ";", escape_double = FALSE, trim_ws = TRUE)
#clean NA vaules
bank.data = na.omit(bank.data)
bank_small = na.omit(bank_small)
#look at sample of our dataset
#head(bank.data)

# modifying data set
#cat_data = data.frame(bank.data$job,bank.data$marital,bank.data$education)
#bin_cat_data = dummy_cols(cat_data)
#bin_cat_data = bin_cat_data %>% select(4:22)

#yesno_data = data.frame(bank.data$default,bank.data$housing,bank.data$loan,bank.data$y)
#yesno_data$bank.data.default <- revalue(yesno_data$bank.data.default, c("yes"=1))
#yesno_data$bank.data.default <- revalue(yesno_data$bank.data.default, c("no"=0))
#yesno_data$bank.data.housing <- revalue(yesno_data$bank.data.housing, c("yes"=1))
#yesno_data$bank.data.housing <- revalue(yesno_data$bank.data.housing, c("no"=0))
#yesno_data$bank.data.loan <- revalue(yesno_data$bank.data.loan, c("yes"=1))
#yesno_data$bank.data.loan <- revalue(yesno_data$bank.data.loan, c("no"=0))
#yesno_data$bank.data.y <- revalue(yesno_data$bank.data.y, c("yes"=1))
#yesno_data$bank.data.y <- revalue(yesno_data$bank.data.y, c("no"=0))
#remaining_data = bank.data %>% select(1,6,9,10,11,12,13,14,15,16)
#master_bin_data = cbind(bin_cat_data,yesno_data,remaining_data)
#head(master_bin_data)

#create test and train dataset, 80-20 split
bank.train = bank.data %>%
  sample_frac(0.8)

bank.test = bank.data %>%
  setdiff(bank.train)

#turning yes and no into 1 and 0
bank.test$y <- ifelse(bank.test$y=='yes',1,0)
bank.train$y <- ifelse(bank.train$y=='yes',1,0)
head(bank.data)
summary(bank.data)
deposit.status = bank.data$y
```

```
# #Age
# ggplot(bank.data,aes(x=bank.data$age,fill=deposit.status)) + geom_histogram(binwidth=1) +
#   labs(y= "Number of Clients", x="Age", title = "Distribution of Deposits by Age")
# age.desc = bank.data %>% group_by(y) %>% summarise(age.mean = mean(age), .groups = 'drop')
#
# #Job
# ggplot(bank.data, aes(x=bank.data$job,fill=deposit.status)) + geom_bar(position = position_dodge())+
#   labs(y= "Number of Clients", x="Job", title = "Distribution of Deposits by Job Type")+
#   theme(axis.text.x = element_text(size = 10, angle = 45, hjust=1, vjust=1))
#
# #Marital Status
# ggplot(bank.data, aes(x=bank.data$marital,fill=deposit.status)) + geom_bar(position = position_dodge(
#   labs(y= "Number of Clients", x="Marital Status", title = "Distribution of Deposits by Marital Statu
#
# #Education
# ggplot(bank.data, aes(x=bank.data$education,fill=deposit.status)) + geom_bar(position = position_dodg
#   labs(y= "Number of Clients", x="Education", title = "Distribution of Deposits by Educational Qualif
#
# #Credit Default
# ggplot(bank.data, aes(x=bank.data$default,fill=deposit.status)) + geom_bar(position = position_dodge(
#   labs(y= "Number of Clients", x="Credit Default", title = "Distribution of Credit Default and Deposi
#
# #Housing Loan
# ggplot(bank.data, aes(x=bank.data$housing,fill=deposit.status)) + geom_bar(position = position_dodge(
#   labs(y= "Number of Clients", x="Contact", title = "Distribution of Client Having a Housing Loan and
#
# #Contact
# ggplot(bank.data, aes(x=bank.data$contact,fill=deposit.status)) + geom_bar(position = position_dodge(
#   labs(y= "Number of Clients", x="Contact", title = "Distribution of Deposits by Contact")
#
# #Loans
# ggplot(bank.data, aes(x=bank.data$loan,fill=deposit.status)) + geom_bar(position = position_dodge())+
#   labs(y= "Number of Clients", x="Loan", title = "Distribution of Clients with Loans and Deposit")
#
# #month
# ggplot(bank.data, aes(x=bank.data$month,fill=deposit.status)) + geom_bar(position = position_dodge())
#   labs(y= "Number of Clients", x="Month", title = "Distribution of Deposits by Month")
#
# #Day
# ggplot(bank.data, aes(x=bank.data$day,fill=deposit.status)) + geom_bar(position = position_dodge())+
#   labs(y= "Number of Clients", x="Day of Week", title = "Distribution of Deposits by Day of Week")
#
# #Previous Campaign Outcome
# ggplot(bank.data, aes(x=bank.data$poutcome,fill=deposit.status)) + geom_bar(position = position_dodge
#   labs(y= "Number of Clients", x="poutcome", title = "Previous Campaign Outcome")
#
# ##Interactive EDA (NOTE: SET TARGET TO y. use install.packages("explore"))
# #explore_shiny(bank.data)
#Age
ggplot(bank.data,aes(x=bank.data$age,fill=deposit.status)) + geom_histogram(binwidth=1) +
  labs(y= "Number of Clients", x="Age", title = "Distribution of Deposits by Age")
age.desc = bank.data %>% group_by(y) %>% summarise(age.mean = mean(age), .groups = 'drop')
#Job
```

```r
ggplot(bank.data, aes(x=bank.data$job,fill=deposit.status)) + geom_bar(position = position_dodge())+
  labs(y= "Number of Clients", x="Job", title = "Distribution of Deposits by Job Type")+
  theme(axis.text.x = element_text(size = 10, angle = 45, hjust=1, vjust=1))
#Marital Status
ggplot(bank.data, aes(x=bank.data$marital,fill=deposit.status)) + geom_bar(position = position_dodge())
  labs(y= "Number of Clients", x="Marital Status", title = "Distribution of Deposits by Marital Status"
#Education
ggplot(bank.data, aes(x=bank.data$education,fill=deposit.status)) + geom_bar(position = position_dodge(
  labs(y= "Number of Clients", x="Education", title = "Distribution of Deposits by Educational Qualifica
#Credit Default
ggplot(bank.data, aes(x=bank.data$default,fill=deposit.status)) + geom_bar(position = position_dodge())
  labs(y= "Number of Clients", x="Credit Default", title = "Distribution of Credit Default and Deposits
#Balance
#bank.data %>% explore(balance)
ggplot(bank.data,aes(x=balance,fill=deposit.status)) + geom_histogram(binwidth=200) +
  labs(y= "Number of Clients", x="Balance", title = "Distribution of Deposits by Balance. Note: outlier

#balance.desc = bank.data %>% group_by(y) %>% summarise(balance.mean = mean(balance), .groups = 'drop')
#Housing Loan
ggplot(bank.data, aes(x=bank.data$housing,fill=deposit.status)) + geom_bar(position = position_dodge())
  labs(y= "Number of Clients", x="Contact", title = "Distribution of Client Having a Housing Loan and D
#Loans
ggplot(bank.data, aes(x=bank.data$loan,fill=deposit.status)) + geom_bar(position = position_dodge())+
  labs(y= "Number of Clients", x="Loan", title = "Distribution of Clients with Loans and Deposit")
#Contact
ggplot(bank.data, aes(x=bank.data$contact,fill=deposit.status)) + geom_bar(position = position_dodge())
  labs(y= "Number of Clients", x="Contact", title = "Distribution of Deposits by Contact")
#month
ggplot(bank.data, aes(x=bank.data$month,fill=deposit.status)) + geom_bar(position = position_dodge())+
  labs(y= "Number of Clients", x="Month", title = "Distribution of Deposits by Month")
#Age
ggplot(bank.data,aes(x=bank.data$duration,fill=deposit.status)) + geom_histogram(binwidth=1) +
  labs(y= "Number of Clients", x="Duration", title = "Distribution of Deposits by Duration")
duration.desc = bank.data %>% group_by(y) %>% summarise(duration.mean = mean(duration), .groups = 'drop
#Contact
ggplot(bank.data, aes(x=bank.data$contact,fill=deposit.status)) + geom_bar(position = position_dodge())
  labs(y= "Number of Clients", x="Contact", title = "Distribution of Deposits by Contact")
#Previous Campaign Outcome
ggplot(bank.data, aes(x=bank.data$poutcome,fill=deposit.status)) + geom_bar(position = position_dodge()
  labs(y= "Number of Clients", x="poutcome", title = "Previous Campaign Outcome")
bank.num = data.frame(as.numeric(as.factor(bank.data$age)),
                      as.numeric(as.factor(bank.data$job)),
                      as.numeric(as.factor(bank.data$marital)),
                      as.numeric(as.factor(bank.data$education)),
                      as.numeric(as.factor(bank.data$default)),
                      as.numeric(as.factor(bank.data$balance)),
                      as.numeric(as.factor(bank.data$housing)),
                      as.numeric(as.factor(bank.data$loan)),
                      as.numeric(as.factor(bank.data$contact)),
                      as.numeric(as.factor(bank.data$day)),
                      as.numeric(as.factor(bank.data$month)),
                      as.numeric(as.factor(bank.data$duration)),
                      as.numeric(as.factor(bank.data$campaign)),
```

```
                      as.numeric(as.factor(bank.data$pdays)),
                      as.numeric(as.factor(bank.data$previous)),
                      as.numeric(as.factor(bank.data$poutcome)),
                      as.numeric(as.factor(bank.data$y)))

colnames(bank.num) = c("Age", "Job", "Marital", "Education", "Default", "Balance", "Housing", " Loan",

bank.num %>%
  cor() %>%
  corrplot(method = "number",
           tl.srt = 45,
           bg = "grey",
           order = "FPC",
           tl.col = "black",
           number.cex = 0.5)
#make copy of data
lr.bank.data <- bank.train

#create logistic model with all variables
logistic.model <- glm(as.factor(y)~., binomial(link = "logit"),lr.bank.data)

#summary of logistic model
summary(logistic.model)
#perform feature selection using likelihood ratio test (comparing a certain coeffeicient vs. it to be z
b_first_run <- drop1(glm(as.factor(y)~., binomial,lr.bank.data),test="LRT")
#Find largest p value (larger p value indicates insignificance) to be age, so we drop age from our seco
b_second_run <- drop1(glm(as.factor(y)~job+marital+education+default+balance+housing+loan+contact+day+m
#Find largest p value to be default, so drop default from our third run
b_third_run <- drop1(glm(as.factor(y)~job+marital+education+balance+housing+loan+contact+day+month+dura
#Find largest p value to be pdays, so drop pdays from our fourth run
b_fourth_run <- drop1(glm(as.factor(y)~job+marital+education+balance+housing+loan+contact+day+month+dur
#Find largest p value to be previous, so drop previous from our fifth run
b_fifth_run <- drop1(glm(as.factor(y)~job+marital+education+balance+housing+loan+contact+day+month+dura
#Stop. All p values significant at alpha = 0.05. Drawback = cutoff level is trivial choice.
#Final model: (y) ~ job + marital + education + balance + housing + loan + contact + day + month + dura
#Dropped age, default, pdays, previous.
#perform feature selection using likelihood ratio test (comparing a certain coeffeicient vs. it to be z
f_first_run <- add1(glm(as.factor(y)~1, binomial, lr.bank.data),
     scope = ~age+job+marital+education+default+balance+housing+loan+contact+day+month+duration+campaig
     test = "LRT")
#Find smallest p value (small p value indicates significance) to be duration, so we add duration to our
f_second_run <- add1(glm(as.factor(y)~duration, binomial, data = lr.bank.data),
     scope = ~.+age+job+marital+education+default+balance+housing+loan+contact+day+month+campaign+pdays
     test = "LRT")
#Find smallest p value (small p value indicates significance) to be poutcome, so we add poutcome to our
f_third_run <- add1(glm(as.factor(y)~duration+poutcome, binomial, data = lr.bank.data),
     scope = ~.+age+job+marital+education+default+balance+housing+loan+contact+day+month+campaign+pdays
     test = "LRT")
#Find smallest p value (small p value indicates significance) to be month, so we add month to our fourt
f_fourth_run <- add1(glm(as.factor(y)~duration+poutcome+month, binomial, data = lr.bank.data),
     scope = ~.+age+job+marital+education+default+balance+housing+loan+contact+day+campaign+pdays+previ
     test = "LRT")
#Find smallest p value (small p value indicates significance) to be contact, so we add contact to our f
```

```
f_fifth_run <- add1(glm(as.factor(y)~duration+poutcome+month+contact, binomial, data = lr.bank.data),
     scope = ~.+age+job+marital+education+default+balance+housing+loan+day+campaign+pdays+previous,
     test = "LRT")
#Find smallest p value (small p value indicates significance) to be housing, so we add housing to our s
f_sixth_run <- add1(glm(as.factor(y)~duration+poutcome+month+contact+housing, binomial, data = lr.bank.
     scope = ~.+age+job+marital+education+default+balance+loan+day+campaign+pdays+previous,
     test = "LRT")
#Find smallest p value (small p value indicates significance) to be job, so we add job to our seventh r
f_seventh_run <- add1(glm(as.factor(y)~duration+poutcome+month+contact+housing+job, binomial, data = lr
     scope = ~.+age+marital+education+default+balance+loan+day+campaign+pdays+previous,
     test = "LRT")
#Find smallest p value (small p value indicates significance) to be campaign, so we add campaign to our
f_eighth_run <- add1(glm(as.factor(y)~duration+poutcome+month+contact+housing+job+campaign, binomial, da
     scope = ~.+age+marital+education+default+balance+loan+day+pdays+previous,
     test = "LRT")
#Find smallest p value (small p value indicates significance) to be loan, so we add loan to our ninth r
f_ninth_run <- add1(glm(as.factor(y)~duration+poutcome+month+contact+housing+job+campaign+loan, binomial
     scope = ~.+age+marital+education+default+balance+day+pdays+previous,
     test = "LRT")
#Find smallest p value (small p value indicates significance) to be marital, so we add marital to our t
f_tenth_run <- add1(glm(as.factor(y)~duration+poutcome+month+contact+housing+job+campaign+loan+marital,
     scope = ~.+age+education+default+balance+day+pdays+previous,
     test = "LRT")
#Find smallest p value (small p value indicates significance) to be education, so we add education to o
f_eleventh_run <- add1(glm(as.factor(y)~duration+poutcome+month+contact+housing+job+campaign+loan+marita
     scope = ~.+age+default+balance+day+pdays+previous,
     test = "LRT")
#Find smallest p value (small p value indicates significance) to be day, so we add day to our twelve ru
f_twelve_run <- add1(glm(as.factor(y)~duration+poutcome+month+contact+housing+job+campaign+loan+marital
     scope = ~.+age+default+balance+pdays+previous,
     test = "LRT")
#Find smallest p value (small p value indicates significance) to be day, so we add balance to our third
f_thirdteenth_run <- add1(glm(as.factor(y)~duration+poutcome+month+contact+housing+job+campaign+loan+ma
     scope = ~.+age+default+pdays+previous,
     test = "LRT")
#Stop. All p values significant at alpha = 0.05. Drawback = cutoff level is trivial choice.
#Final mode: (y) ~ duration + poutcome + month + contact + housing + job + campaign + loan + marital +
#Dropped age, default, pdays, previous.
f_AIC <- step(glm(y~1, binomial, lr.bank.data),
     scope = ~age+job+marital+education+default+balance+housing+loan+contact+day+month+duration+campaig
     direction = "forward")
f_AIC
#previous, pdays, default, age
# y ~ duration + poutcome + month + contact + housing + job + campaign + loan + marital + education + d
f_BIC <- step(glm(y~1, binomial, lr.bank.data),
     scope = ~age+job+marital+education+default+balance+housing+loan+contact+day+month+duration+campaig
     direction = "forward",
     k = log(dim(lr.bank.data)[1]))
f_BIC
#previous, default, age, pdays, job
#y ~ duration + poutcome + month + contact + housing + campaign + loan + marital + education + balance
b_AIC <- step(glm(y~., binomial, lr.bank.data),
     direction = "backward")
```

```r
#pdays, default, age, previous
# y ~ job + marital + education + balance + housing + loan + contact + day + month + duration + campaig
b_AIC
b_BIC <- step(glm(y~., binomial, lr.bank.data),
     direction = "backward",
     k = log(dim(lr.bank.data)[1]))
b_BIC
#previous, default, age, pdays, job
#y ~ marital + education + balance + housing + loan + contact + day + month + duration + campaign + pou

both_AIC <- step(glm(y~1, binomial, lr.bank.data),
     scope = ~age+job+marital+education+default+balance+housing+loan+contact+day+month+duration+campaig
     direction = "both")
both_AIC
#pdays, default, age, previous
#y ~ duration + poutcome + month + contact + housing + job + campaign + loan + marital + day + educatio
both_BIC <- step(glm(y~1, binomial, lr.bank.data),
     scope = ~age+job+marital+education+default+balance+housing+loan+contact+day+month+duration+campaig
     direction = "both",
     k = log(dim(lr.bank.data)[1]))
both_BIC
#y ~ duration + poutcome + month + contact + housing + campaign + marital + loan + day
#y ~ duration + poutcome + month + contact + housing + campaign + loan + marital + education + balance
#previous, default, age, pdays, job
summary(glm(y ~ duration + poutcome + month + contact + housing + job + campaign + loan + marital + day
#Build logistic model with selected significant variables using AIC stepwise variable selection
logistic.model.AIC <- glm(y ~ duration + poutcome + month + contact + housing + job + campaign + loan +

pred = format(round(predict(logistic.model.AIC, newdata = bank.test, type = "response")))
conf = table(pred, as.factor(bank.test$y))
#Tuning parameters
pred2 = predict(logistic.model.AIC, newdata = bank.test, type = "response")
pred.tune.log = prediction(pred2, bank.test$y)
perf.tune.log = performance(pred.tune.log, measure = "tpr", x.measure = "fpr")
#accuracy.log = performance(pred.tune.log, measure = "acc", x.measure = "cutoff")
#plot(accuracy.log)
plot(perf.tune.log, main="ROC CURVE FOR LOGISTIC REGRESSION METHOD")
auc.log = as.numeric(performance(pred.tune.log, measure = "auc")@y.values)
#Classification accuracy
log.acc = (conf[1,1]+conf[2,2])/(sum(conf))
conf
#create random forest model with all variables, mtry = sqrt(p) for classification

#Find ntree

#control1 = trainControl(method = "repeatedcv", number=10, repeats=3, search="grid")
#tuning.grid = expand.grid(mtry = sqrt(dim(bank.train)[2]-1))
#model.list = list()
#for (ntree in c(100,200,300,400,500)){
#  set.seed(123)
#  tune.fit = randomForest(as.factor(y)~., data = bank.train, metric=metric, tuneGrid = tuning.grid, tr
#  index.tune = toString(ntree)
#  model.list[[index.tune]] = tune.fit
```

```
#  }
#tuning.results = resamples(model.list)
#summary(tuning.results)
#dotplot(tuning.results)

#Finding mtry

#ideal <- tuneRF(bank.train, as.factor(bank.train$y), ntreeTry=500, stepFactor=2, improve=0.05,
#        trace=FALSE, plot=FALSE, doBest=TRUE)

#Final Model with 500 ntree and 4 mtry

rf <- randomForest(as.factor(y)~ ., data = bank.train, importance = TRUE, mtry = 4, ntree = 500)
#importance of variables
importance_of_variables <- importance(rf)
#plot of importance of variables
varImpPlot(rf)
#duration is the most important variable, default is the least important

prediction_rf <- format(predict(rf, newdata = bank.test, type = "response"))
conf_rf <- table(prediction_rf, as.factor(bank.test$y))

predictions = as.vector(rf$votes[,2])
pred=prediction(predictions,bank.train$y)

perf_AUC=performance(pred,"auc") #Calculate the AUC value
AUC=perf_AUC@y.values[[1]]

perf_ROC=performance(pred,"tpr","fpr") #plot the actual ROC curve
plot(perf_ROC, main="ROC CURVE FOR RANDOM FOREST METHOD")

#Classification accuracy
classification_accuracy_rf <- (conf_rf[1,1]+conf_rf[2,2])/(sum(conf_rf))
conf_rf
split = sample.split(bank_small$y,SplitRatio = 0.80)
train.svm = subset(bank_small, split == TRUE)
test.svm = subset(bank_small, split == FALSE)
train.svm$y = as.numeric(as.factor(train.svm$y))
test.svm$y = as.numeric(as.factor(test.svm$y))

svmmodel = svm(y~., data = train.svm, kernel="linear", probability=TRUE, cost = 1, gamma=0.02)
pred.svm =format(round( predict(svmmodel,test.svm)))
svm.pred1 = predict(svmmodel,test.svm)
svm.pred = prediction((predict(svmmodel,test.svm)),test.svm$y)
svm.perf = performance(svm.pred, "tpr",  "fpr")
svm.perf.acc = performance(svm.pred, 'acc')
auc.svm = performance(svm.pred, 'auc')
auc.svm=auc.svm@y.values[[1]]
plot(svm.perf, main = "ROC CURVE FOR SVM METHOD")
plot(svm.perf.acc)
conf.svm = table(pred.svm, as.factor(test.svm$y))
acc.svm = (conf.svm[1,1]+conf.svm[2,2])/(sum(conf.svm))
comparison.table = data.frame(Method = c('Random Forest', 'Logistic Regression', 'SVM'), Accuracy = NA,
```

```r
comparison.table$Accuracy = c(classification_accuracy_rf, log.acc, acc.svm)
comparison.table$AUC = c(AUC, auc.log, auc.svm)
kable(comparison.table)

plot(perf.tune.log, lwd = 3, colorize = TRUE,
     main = "ROC CURVE FOR LOGISTIC MODEL",
     print.cutoffs.at = seq(0,1,by=0.1))
abline(0,1, col = "red", lty = 2)

plot(perf_ROC, lwd = 3, colorize = TRUE,
     main = "ROC CURVE FOR RANDOM FOREST METHOD",
     print.cutoffs.at = seq(0,1, by=0.1))
abline(0,1,col="red", lty = 2)

plot(svm.perf, lwd = 3, colorize = TRUE,
     main = "ROC CURVE FOR SVM METHOD",
     print.cutoffs.at = seq(0,1, by=0.1))
abline(0,1,col = "red", lty = 2)
```