### Stats Project 2: Logistic Regression - Term Deposit

-Aniketh Vankina, Vijay Kaniti

### Introduction

This dataset that is being analyzed for this research/project has to do with Bank Marketing. Not specifically on any one bank, but just overall bank trends that is centered towards specific attributes and properties. One specific trend we want to analyze and predict is on term deposit. A term deposit is basically a cash investment that is held at a financial institution, in our case a bank. These deposits accrue interest over the years or used specifically for possible investments. For this task we want to predict whether or not an individual applies a term deposit (yes/no). With the help of various regression techniques, we will identify model(s) that predicts/classifies the best.

### Data Description

As far as the data is concerned, this is one of the more abundant datasets. It has around 41,000 observations coupled with 21 variables. It also has a mix of categorical (12) and continuous variables (9) with some categorical variables having multiple levels. The variables and attributes vary from personal information to finance.

Below are a few instances:

Personal Info related variables

* Age
* Marital
* Job
* Education

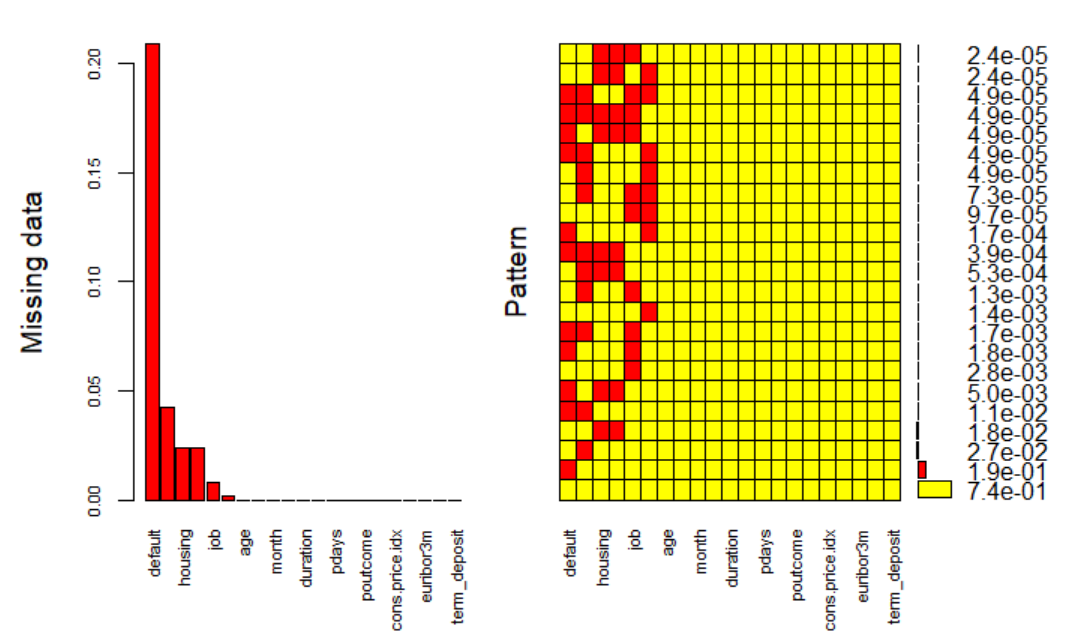
Financial related variables

* Loan
* Consumer Price Index
* Consumer Confidence Index
* Three month rates (Euribor)

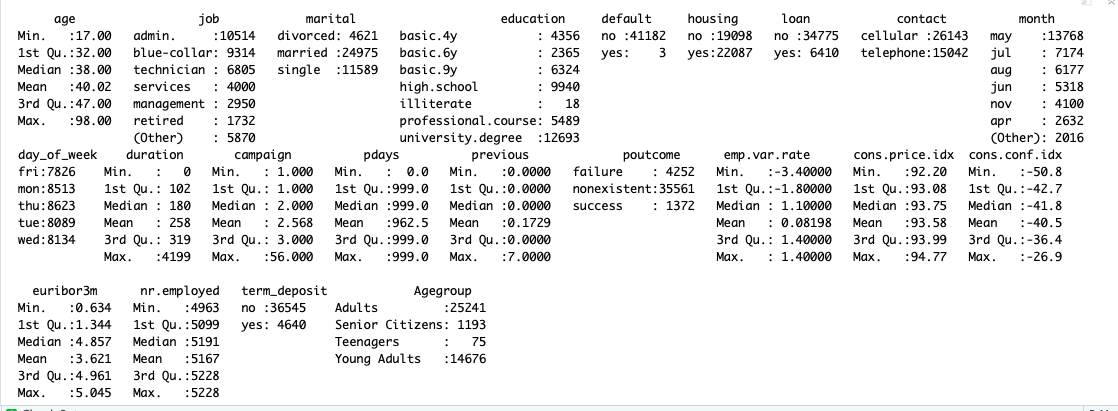
### Exploratory Analysis

For the EDA portion of this project, we spent a significant amount of time to get the gist of the overall details. Below points highlight the path we took to optimizing the data without overlooking any susceptible points:

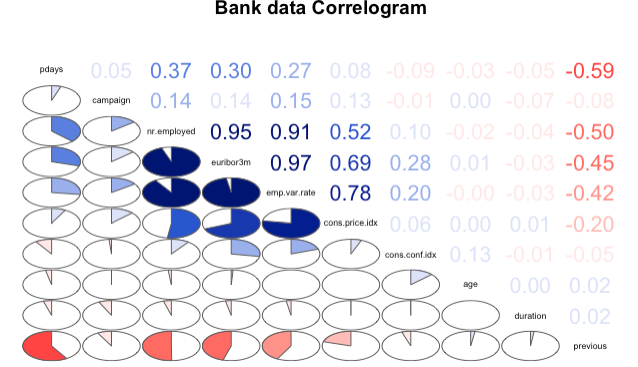
* Unknown/NA values and how they were treated:
  + Missing Data Plot (see figure below):
  + We decided to replace these unknown variables with NA and then impute the NA values using the Mice function in R. Mainly with the means and recurring observations for the imputation.



* Response variable **y** was renamed to **term\_deposit.**
* Below are the summary statistics for the data we worked with. Outliers and such were addressed in Objective One.



* **Age** was also recreated as a categorical variable called Age group that split into four categories: Teenagers, Young Adults, Adults, and Senior Citizens. This was mainly done for interpretation purposes.
* We also performed a correlation matrix to check whether variables are correlated with each other (The matrix is shown below). Some were identified as “trash” and did not correlate very well with the other explanatory variables. Those were either removed or left in the model for the feature selection to remove.



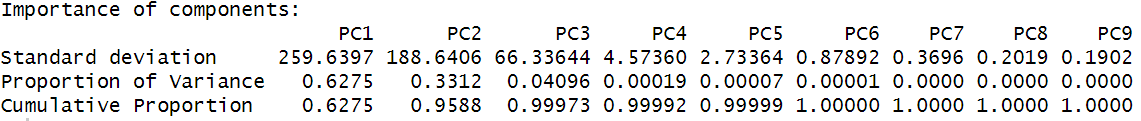
* euribor3m and nr.employed are highly correlated (0.95)
* emp.var.rate and euribor3m are highly correlated (0.97)
* emp.var.rate and nr.employed are highly correlated (0.91)
* The distribution of age is slightly right skewed. More number of observations fall underage group between 22 and 60. Distribution of those values looks normal.
* The distribution of **duration** is highly right skewed, "duration" and the response variable are strongly associated. The longer the duration then there is a chance that person subscribes for a term deposit. Duration variable may not available to predict observation as we will not know how long customer will be talking in phone.But for now we included in our model as it shows significant relation with response variable.
* The distribution of **campaigns** is right skewed and looks like the greater number of campaigns the less chances the person subscribes for a term deposit.
* The **pdays** variable is the number of days that passed by after the client was last contacted from a previous campaign. Most of the pdays have values of 999 which means these customers were never contacted in the past.
* We can remove **emp.var.rate** and **nr.employed** from our analysis due to multicollinearity.
* 65.1% of customers whose **previous** outcome was “Success” subscribed to term deposit.
* The **default**, **housing**, and **loan** predictors shows no significance with response variable.
* From the Chi-square test of each categorical variable predictors like job, education, contact, month, and marital shows significance with response variable.

* From the given dataset, default, housing, and job values are missing for a major number of observations.
* **Multicollinearity Check:** Removed **euribor3m** and **emp.var.rate**. Below is the VIF figure after removing these variables is shown below.
  + 
* We calculated VIF and anything above 10 was removed.

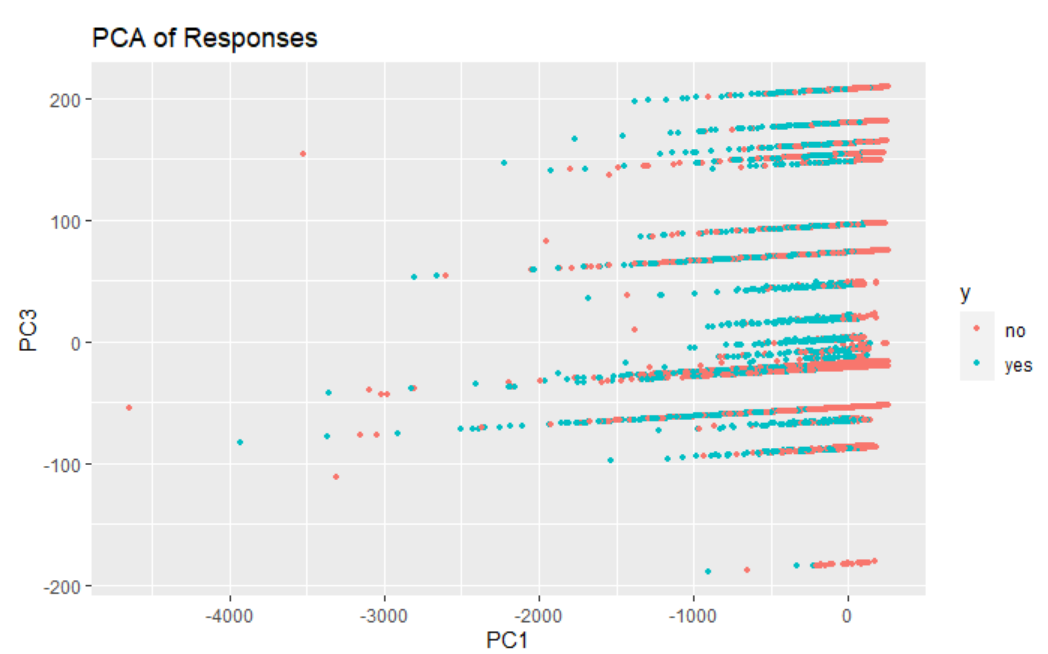
PCA Analysis:

We prepared the data set with all continuous variables from data set and verified if we can detect any relation between variables. Scree plot is show in Appendix

Cumulative proportion of PC3 is 99.9% which explains all variables.



When we plotted PC1 and PC3, we can see some very slight separation between response variable No and Yes observations.



### Objective 1: Simple Logistic Regression Model

The overall goal for this objective is to identify the key variables that help with predicting term deposits. The direction that we decided to go to achieve our goal, is to delve deeper into the dataset so we can check which predictors played the most significant role with helping with prediction accuracy. Additionally, we want to check the likelihood of the effect of the predictors on the response.

For our simpler model we have chosen a logistic regression method to predict term deposit.

The following are the assumptions that should be satisfied to implement logistic regression.

1. Response variable should be dichotomous.
   * Term deposit variable holds yes or no
2. Observations should be independent of each other.
   * We do not have much information about the people to whom the calls were made. We can assume independence.
3. There are no high intercorrelations among predictors.
   * We performed EDA and removed predictors which are highly correlated.
4. No influential values or outliers (Figure XXX)

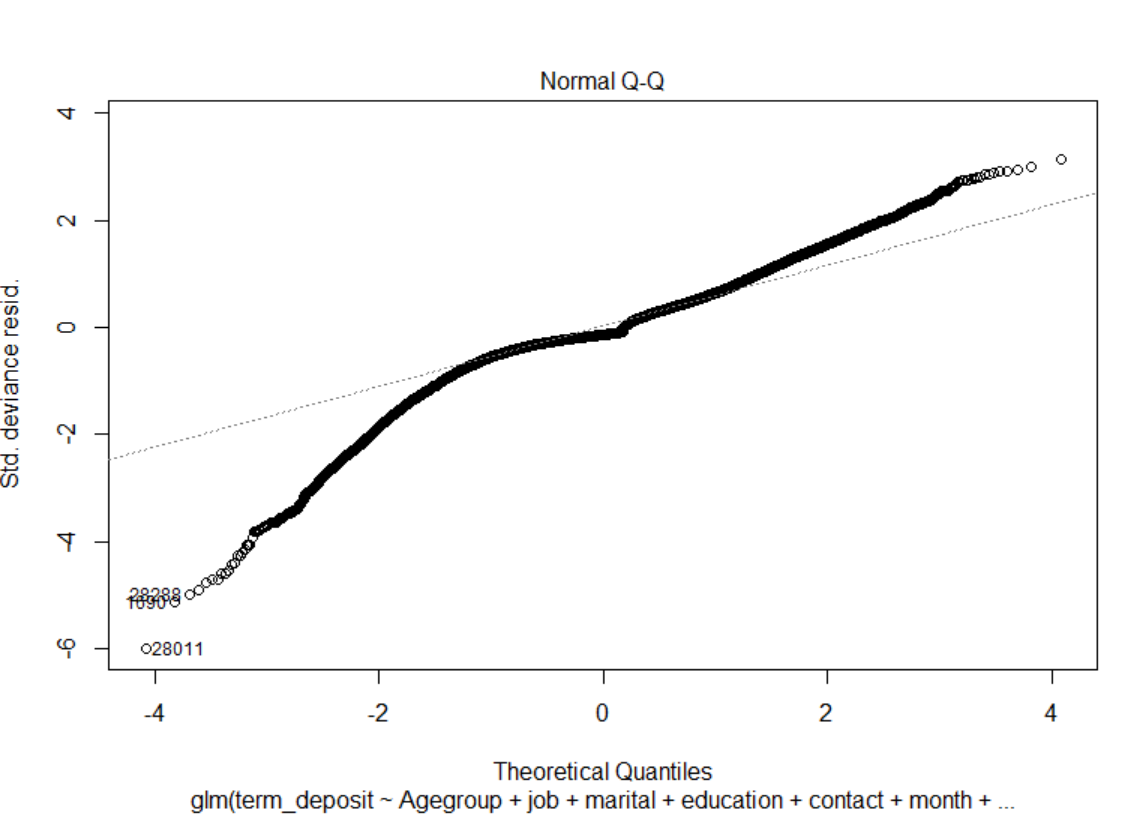
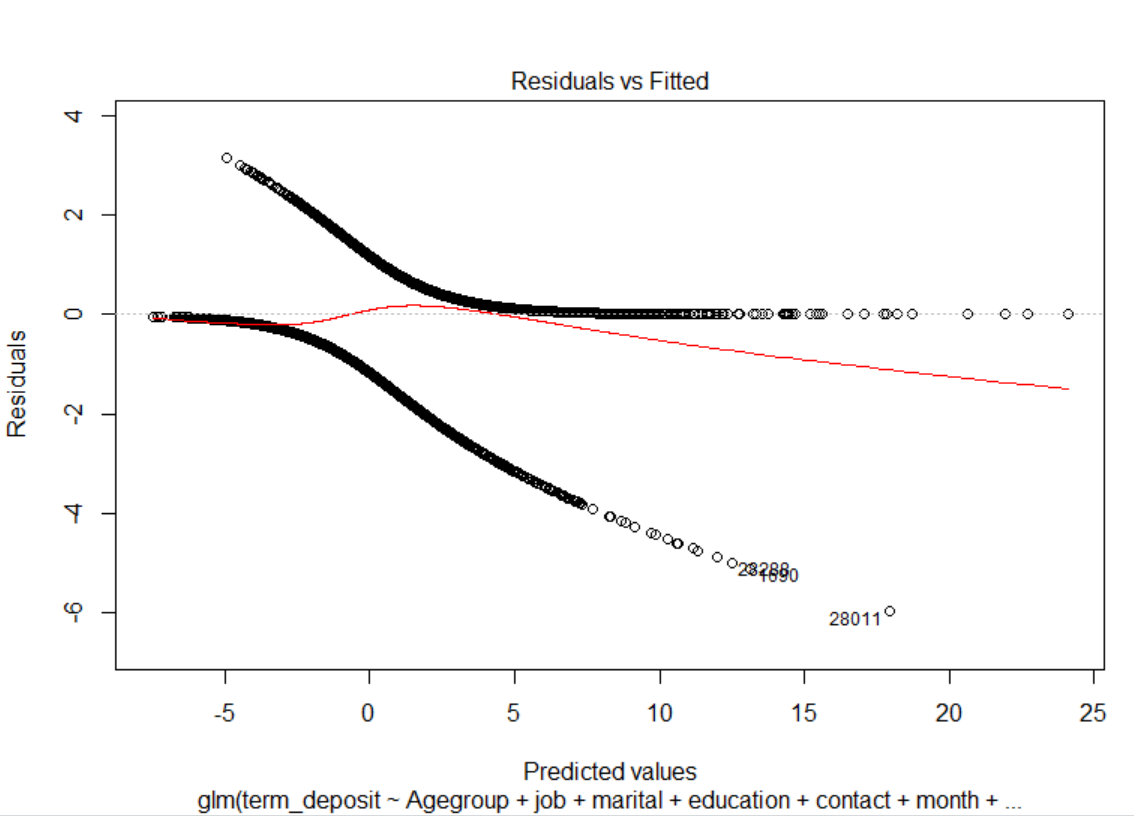
* As per our initial EDA and further analysis of the simple model, we found a few outliers that we removed. An important note is that the model prediction metrics did not drastically change. This will be explored in the later objective.

In the given data set there are more “No’s” than “Yes’s”. This might lead to imbalance in the data set and if we build a model, it may not yield good results. So, we performed oversampling (**SMOTE**) technique to balance the training data set.

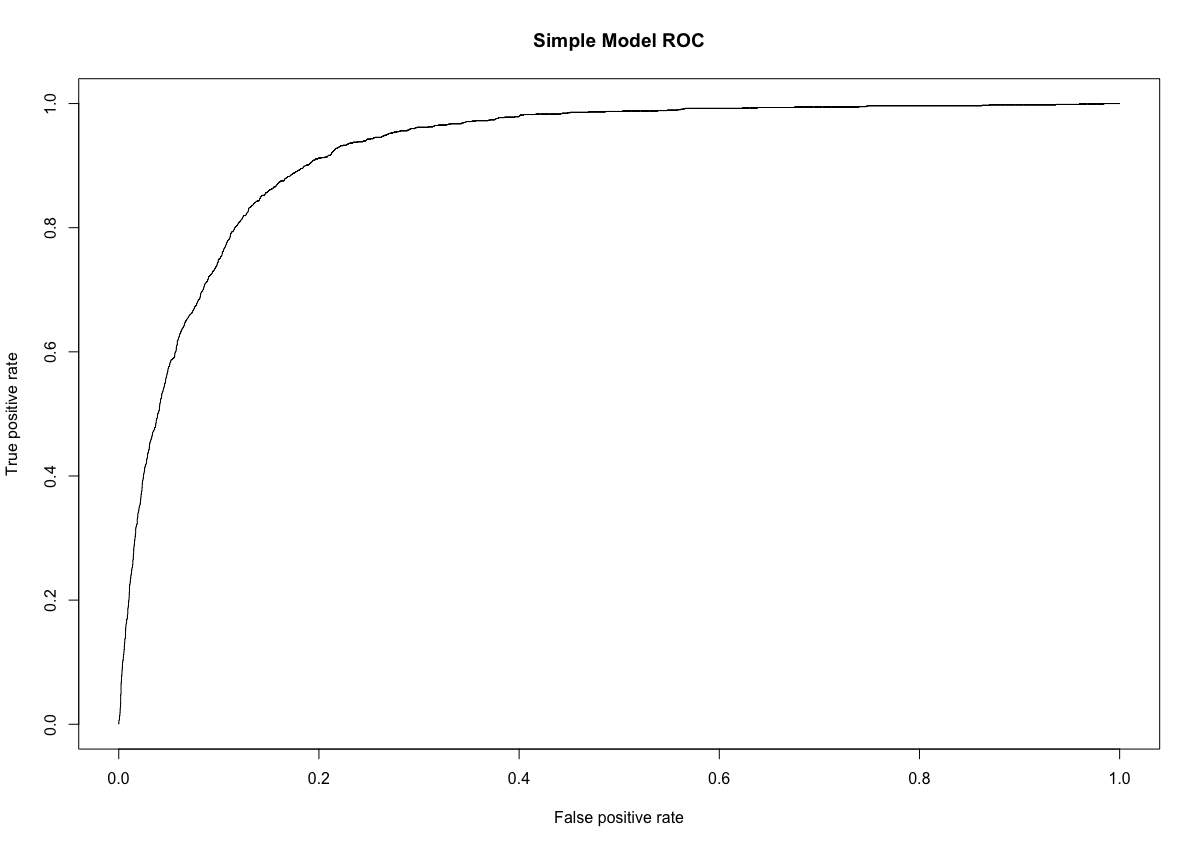
We performed a simple logistic regression model using below parameters.

*simple.log = glm(term\_deposit ~ Agegroup + job + marital + education + contact + month + day\_of\_week + poutcome + campaign + duration + pdays + previous + cons.price.idx + cons.conf.idx , family = "binomial", data = smote\_train)*

**Plots**

****

Plot ROC here for simple model

****

Variable Selection:

After performing Chi square test and correlation graphs the above modeling is done based on our manual intuition from results of the explanatory variables. We built logistic regression models using LASSO and Stepwise feature selection technique. They are compared with the simple logistic model, stated above, in objective two.

* Stepwise selection which adds on an independent variable each time but constantly retests the significance of the predictor already in the model.
* LASSO regression reduced model complexity by penalizing the model coefficients to zero.

Interpretation:

* Holding all other explanatory variables fixed, odds of a person who is a **teenager** (Age<20) subscribing to a term-deposit is **2.78** times higher than a person whose age is greater than 20. Confidence interval is [1.22,6.31].
* Holding all other explanatory variables fixed, odds of a person who is **unemployed** subscribing to a term-deposit is **2.82** times higher than a person having a job. Confidence interval is [2.22,3.50].
* Holding all other explanatory variables fixed, odds of a person who is **in high school** subscribing to a term-deposit is **1.63** times higher than a person who is having education. Confidence interval is [1.34,1.99]
* Holding all other explanatory variables fixed, odds of a person who is contacted in the month of **March** subscribing to a term-deposit is **4.8** times higher than a person who is having education. Confidence interval is [3.55,6.65]
* For every 1 unit increase in **consumer price index** (measure of inflation) the odds of person subscribing to term deposit will increase by multiplicative factor of **2.82** holding other explanatories fixed. Confidence interval is [2.53,3.17]
* For every 1 unit increase in **consumer confidence index** (how optimistic consumers are regarding about their financial situation) the odds of person subscribing to term deposit will increase by multiplicative factor of **1.02** holding other explanatories fixed. Confidence interval is [1.01,1.03]

##### Conclusion:

* In conclusion, our simple model was highly interpretable, and the predictors were chosen based on intuition from our EDA. Our final models could have been either the stepwise or lasso model. The main reasoning in choosing a simple model is to see how well the variables we hand-picked compare to that of a feature selection models.

### Objective 2

The task for this objective is to compare the multiple other models to see how it compares with our simple model from the prior objective:

* LDA Model (With just continuous predictors)
* More complex logistic model that was derived from the simple model by adding interaction terms.
* Stepwise Logistic Model (Ran on all predictors)
* LASSO Logistic Model (Ran on all predictors) (Figure 4, Models)
* KNN (Checked importance of predictors bar graph to pick the best ones, we included all in this optional model)

Note: Cutoff was set at 0.5 for optimal performance (Figure 5, Models)

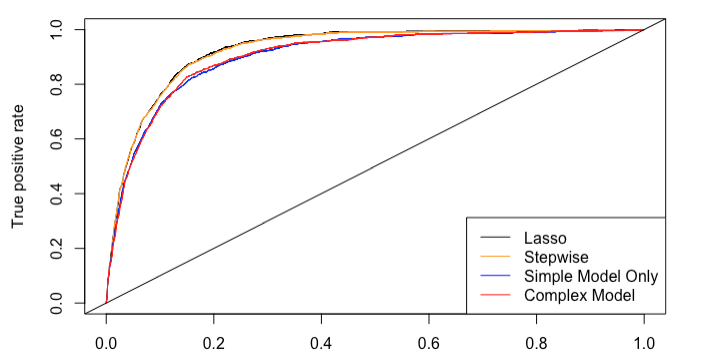
The complex model that we derived included interaction terms that we justified in our EDA. The goal was to obtain good classification scores for the models involved. The predictors with multicollinearity were accounted for and removed from the simple and complex model. In the tables below shows the metrics of the model’s predictive capabilities on the test sets. List of predictors that we included in these models is below the table.

#### Main Analysis Content

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model | AIC | Accuracy | Sensitivity | Specificity | AUC |
| Simple Model | 15460 | 87.47% | 89.22% | 73.88% | 90.33% |
| Complex Model | 15644 | 87.33% | 89.09% | 73.67% | 90.41% |
| Stepwise Model | 13315 | 88.21% | 87.41% | 83.71% | 92.59% |
| LASSO Model | 17520 | 88.25% | 87.16% | 84.22% | 92.74% |
| LDA | N/A | 86.67% | 88.79% | 75.14% | 90.54% |
| KNN | N/A | 88.53% | 90.47% | 73.08% | N/A |

A brief overview on the performance metrics we observed as per the above table:

* Accuracy: This metric is telling us the classification rate for the model, meaning how well it predicts yes/no for term deposits.
* AIC: Mainly telling us which model better, lower AIC is roughly tells us which is better model. Its not be all or tell all.
* Sensitivity: True positive rate aka tells us the true rate if they submit a term deposit or not.
* Specificity: True negative rate aka tells us those that do not make a term deposit.
* AUC: Telling us the performance of the ROC curve or the index accuracy. Higher the area the better the model.



As per the above table, you can see that the Logistic Regression model with the LASSO feature selection technique performed the best, followed closely by the logistic model using the stepwise feature selection. Our simple highly interpretable model performed pretty well.

The model that did not perform as expected is the complex logistic regression model, we added interaction terms of highly correlated variables to check if it made a difference (**duration\*poutcome, cons.price\*cons.conf, agegroup\*marital**). The complex model performed just as well as the simple model.

The LDA model performed slightly better than the simple model but it did have higher specificity compared to that of simple and complex (Figure 1, LDA). The AUC was stellar for all of the models, with exception of KNN (Figure 2 KNN).

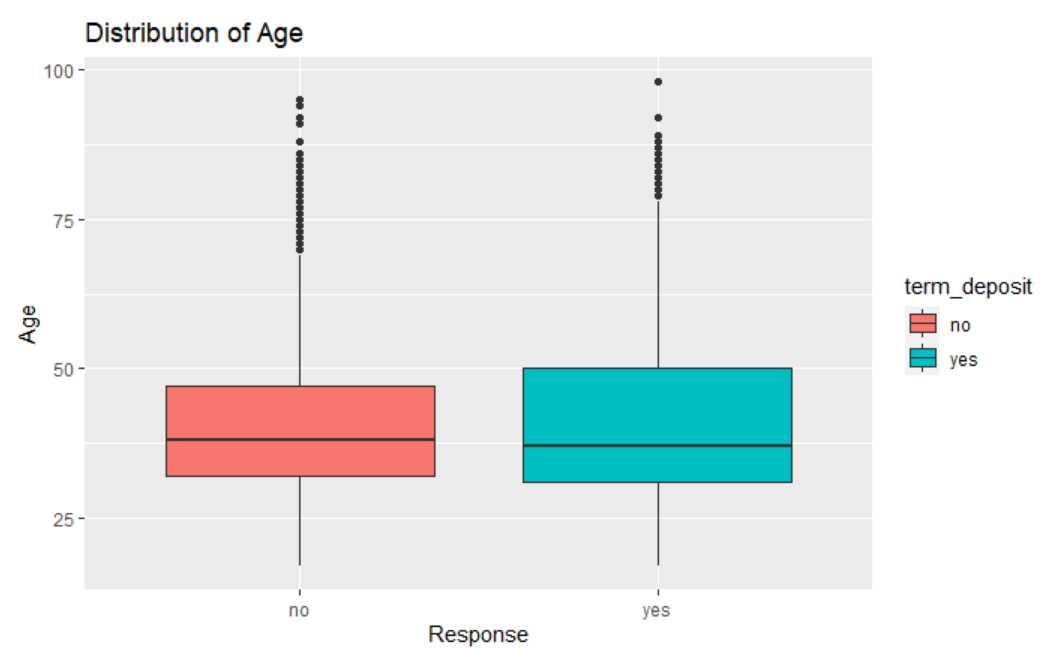
In terms of accuracy, KNN model performed well but specificity is not as good as LDA. It is important to note that KNN and LDA were run with continuous predictors only to assist with classification. The sensitivity and specificity values are all above 70% which helps true positive and true negative scores. These were alternative models that were created to see how well they perform with the logistic regression models. We used a KNN tuning parameter to check for the importance of predictors for all variables. (Figure 3 KNN, Appendix)

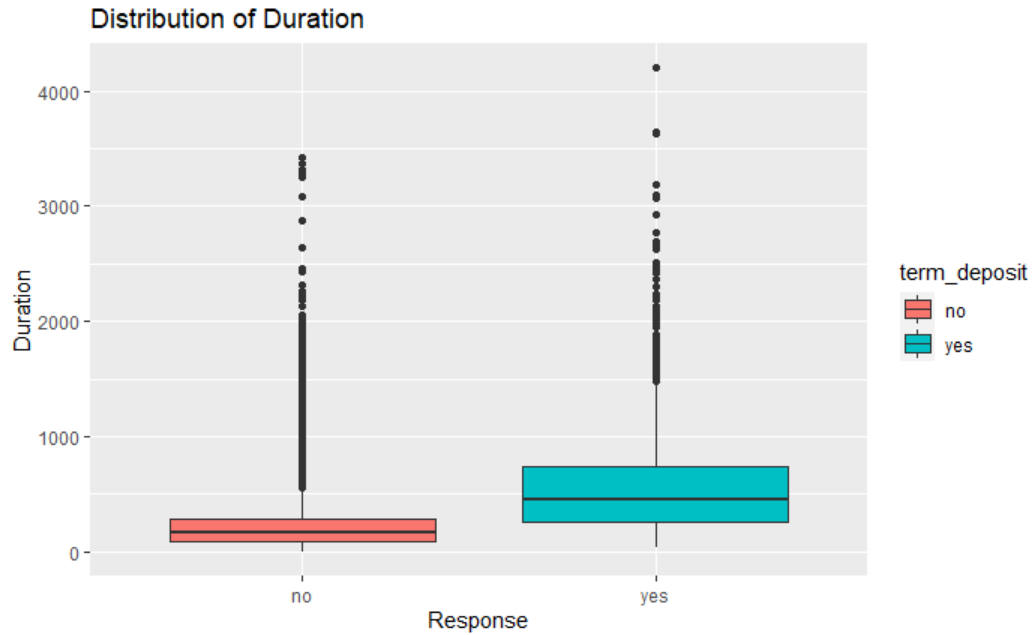
#### Conclusion/Discussion

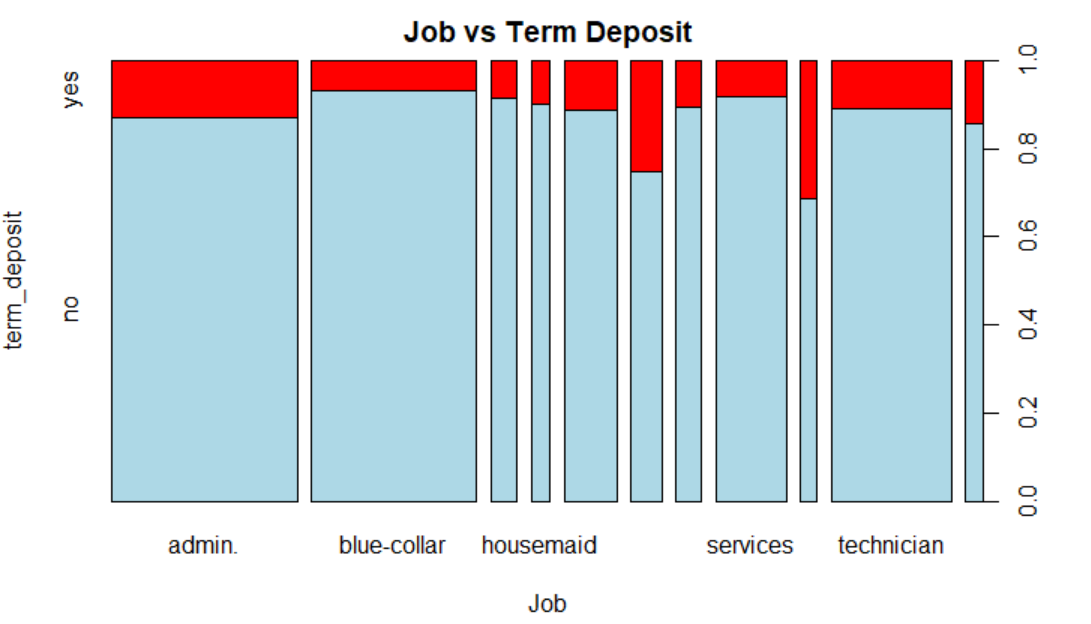
In conclusion, all models were compared and from the table provided the Logistic Regression Model with LASSO selection is our pick for the final model. We believe the reason why LASSO predicted well is because it performs CV by default. The LASSO call did the tough work of determining the optimal number of variables to include. Especially with this dataset, there were levels to the categorical variables which added more predictors. We also tried removing ***pdays****,* ***duration****,* ***previous****, and* ***poutcome*** to improve performance. It had the opposite effect and hurt the model in specificity, although, accuracy was still above 80%. The KNN model was also a surprise, we did not expect KNN to perform as well as it did on the accuracy front. This data is suited for non-parametric models such as KNN because it is used for classification, so it made sense at the end of the day. Something we could have done differently is tried some down sampling techniques on the training set to build up alternate models to check predictive performance. We also performed random forest as an unsupervised tool, but we did not interpret the output. It is included in the overall code but was not utilized to make any improvements.

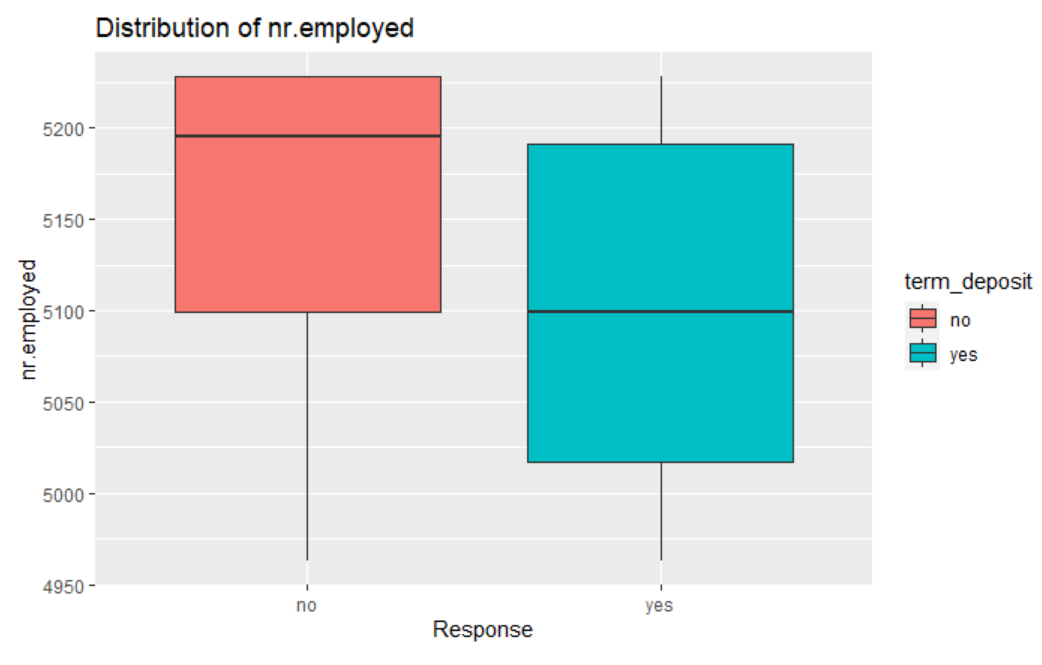
#### Appendix

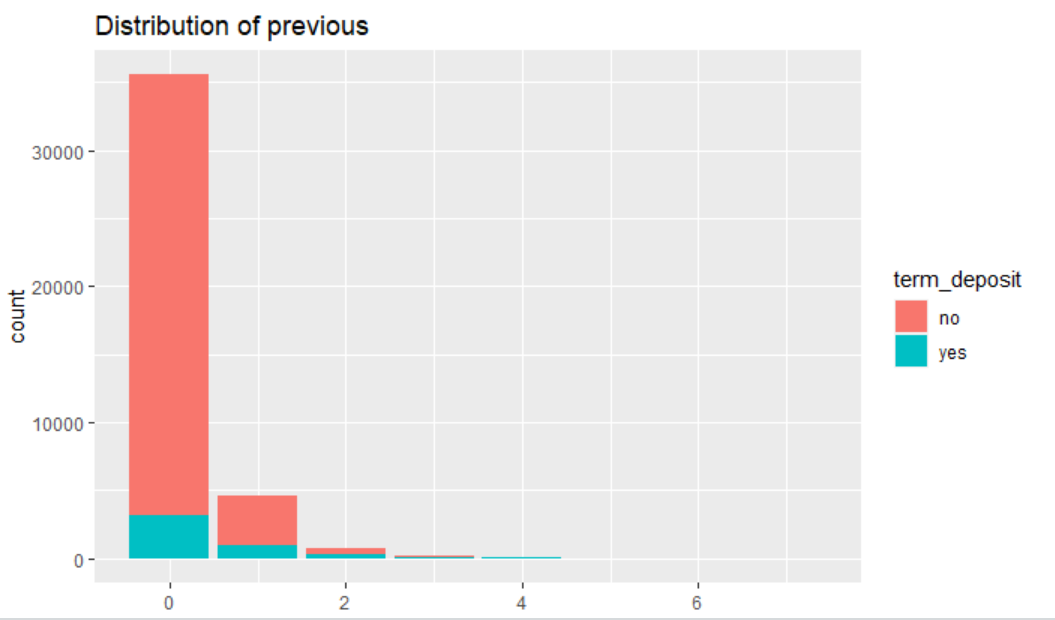
#### EDA Graphs

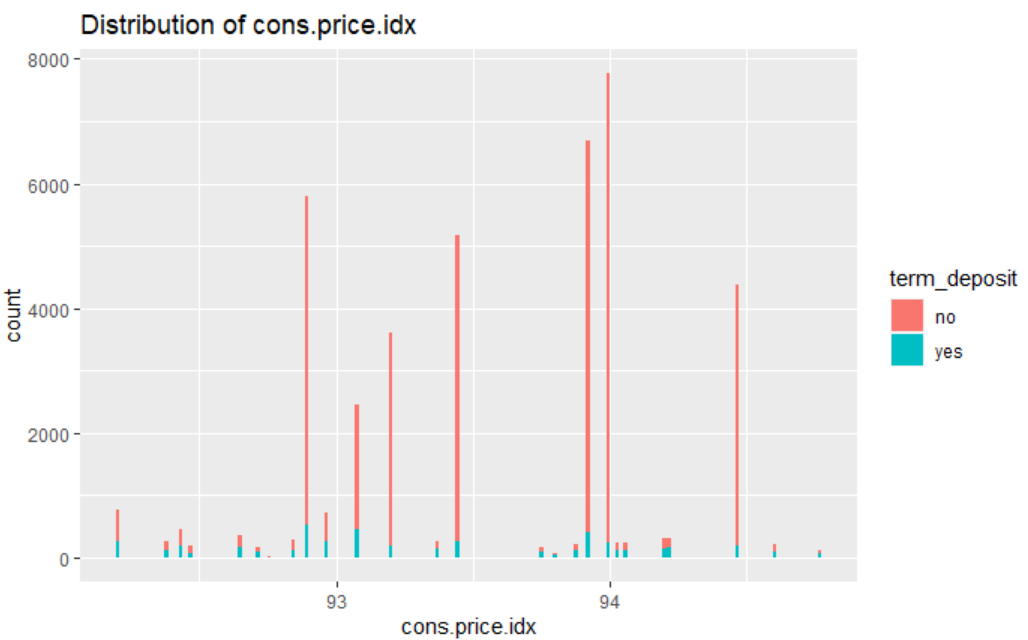


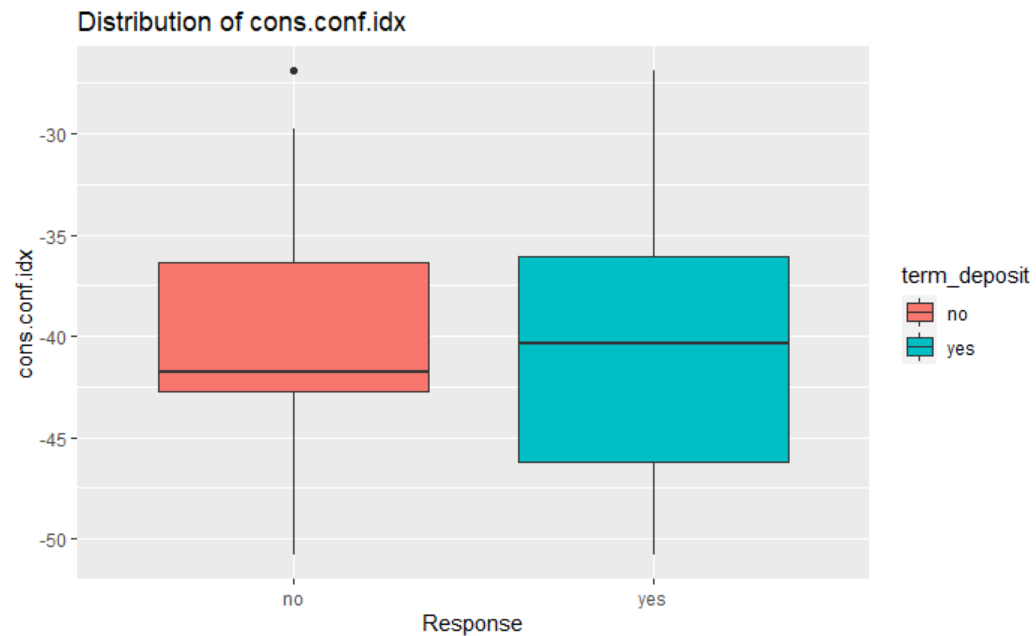






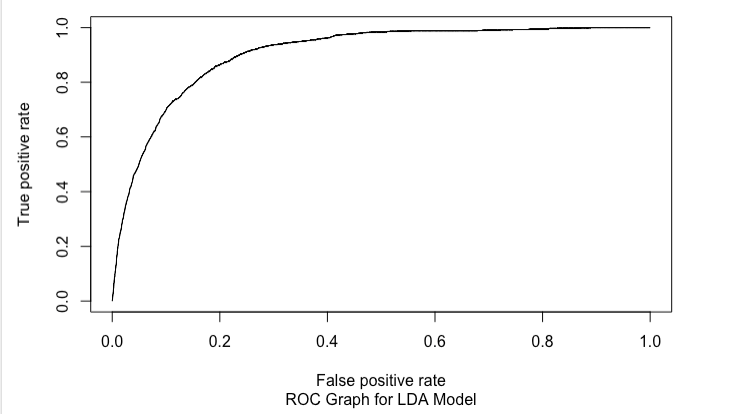






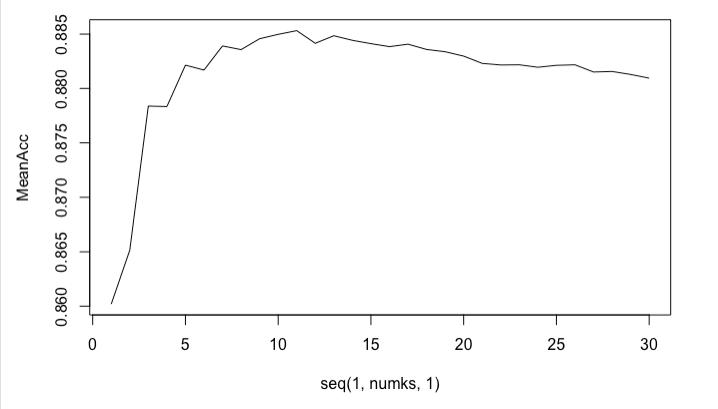
#### LDA Graphics

##### (Figure 1, LDA)



#### KNN Graphics

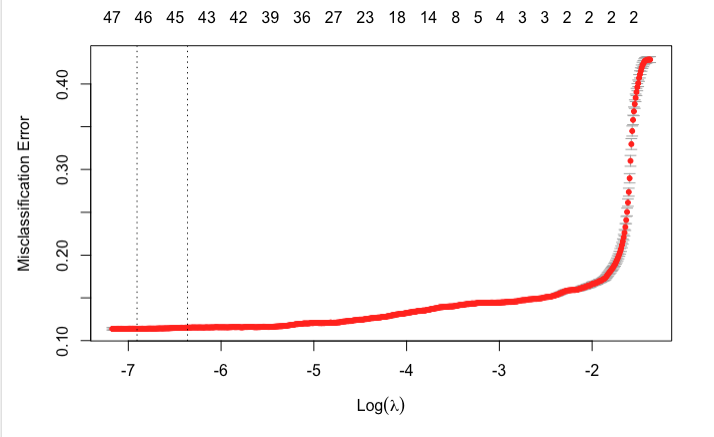
##### (Figure 2, KNN)



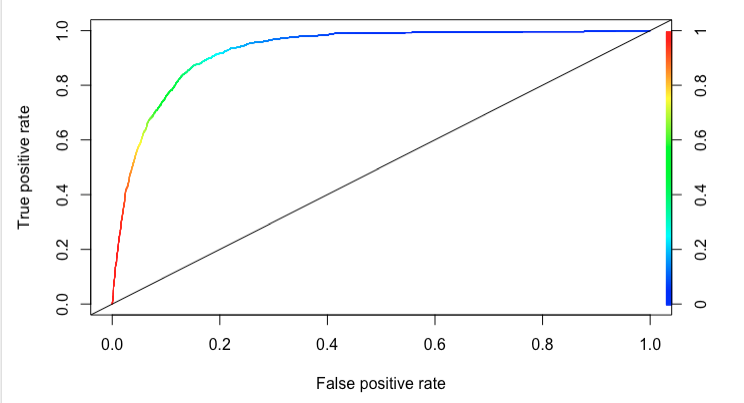
##### (Figure 3, KNN)

#### Model Graphics/Note

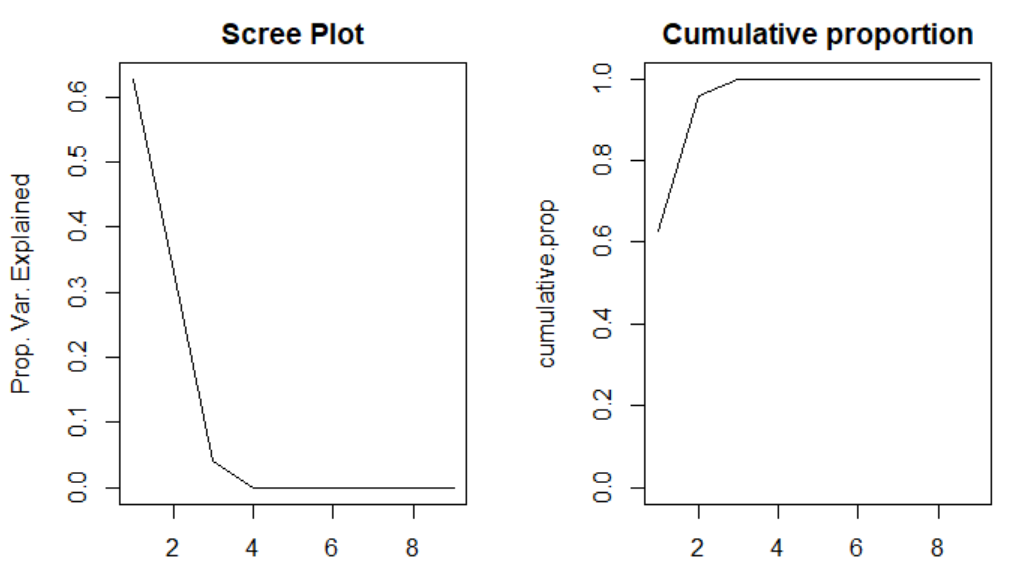
##### Figure 4, Models



##### Figure 5, Models



PCA Scree plot



### **Code**

---

title: "Stats2Project2"

author: "Aniketh V, Vijay Kaniti"

date: "7/22/2020"

output: html\_document

---

```{r setup, include=FALSE}

knitr::opts\_chunk$set(echo = TRUE)

```

Load the libraries

```{r}

library(dplyr)

library(ggplot2)

library(tidyr)

library(magrittr)

library(stringr)

library(corrplot)

library(caret)

library(glmnet)

library(Lahman)

library(mice)

library(car)

library(MASS)

library(ROCR)

library(caret)

library(dplyr)

library(ggplot2)

library(tidyr)

library(readr)

library(digest)

library(ISLR)

library(car)

library(leaps)

library(Matrix)

library(foreach)

library(glmnet)

library(VIM)

library(mice)

library(corrgram)

library(car)

library(gridExtra)

library(MASS)

library(mvtnorm)

library(class)

library(caret)

library(e1071)

library(class)

library(generalhoslem)

```

Loading dataset

```{r}

#imported the main data

Bank\_Fix <- read.csv("bank-additional-full.csv",sep=";",header=TRUE, strip.white = TRUE, na.strings = c("unknown"))

#Separted the header string into separate columns

#Bank\_Fix = mainBank %>% separate(age.job.marital.education.default.housing.loan.contact.month.day\_of\_week.duration.campaign.pdays.previous.poutcome.emp.var.rate.cons.price.idx.cons.conf.idx.euribor3m.nr.employed.y, c("age", "job", "marital", "education", "default", "housing", "loan", "contact", "month", "day\_of\_week", "duration", "campaign", "pdays", "previous", "poutcome", "emp.var.rate", "cons.price.idx", "cons.conf.idx", "euribor3m", "nr.employed", "y"), ";", extra = "merge")

Bank\_Fix$age = as.integer(Bank\_Fix$age)

Bank\_Fix$nr.employed = as.integer(Bank\_Fix$nr.employed)

Bank\_Fix$euribor3m = as.double(Bank\_Fix$euribor3m)

Bank\_Fix$cons.conf.idx = as.double(Bank\_Fix$cons.conf.idx)

Bank\_Fix$cons.price.idx = as.double(Bank\_Fix$cons.price.idx)

Bank\_Fix$emp.var.rate = as.double(Bank\_Fix$emp.var.rate)

Bank\_Fix$previous = as.integer(Bank\_Fix$previous)

Bank\_Fix$pdays = as.integer(Bank\_Fix$pdays)

Bank\_Fix$campaign = as.integer(Bank\_Fix$campaign)

Bank\_Fix$duration = as.integer(Bank\_Fix$duration)

Bank\_Fix$y = as.factor(Bank\_Fix$y)

Bank\_Fix$poutcome = as.factor(Bank\_Fix$poutcome)

Bank\_Fix$term\_deposit = Bank\_Fix$y

Bank\_Fix$default = as.factor(Bank\_Fix$default)

Bank\_Fix$housing = as.factor(Bank\_Fix$housing)

Bank\_Fix$marital = as.factor(Bank\_Fix$marital)

Bank\_Fix$loan = as.factor(Bank\_Fix$loan)

Bank\_Fix$job = as.factor(Bank\_Fix$job)

for(i in 1 : nrow(Bank\_Fix)){

if (Bank\_Fix$age[i] < 20){

Bank\_Fix$Agegroup[i] = 'Teenagers'

} else if (Bank\_Fix$age[i] < 35 & Bank\_Fix$age[i] > 19){

Bank\_Fix$Agegroup[i] = 'Young Adults'

} else if (Bank\_Fix$age[i] < 60 & Bank\_Fix$age[i] > 34){

Bank\_Fix$Agegroup[i] = 'Adults'

} else if (Bank\_Fix$age[i] > 59){

Bank\_Fix$Agegroup[i] = 'Senior Citizens'

}

}

Bank\_Fix$Agegroup<-as.factor(Bank\_Fix$Agegroup)

Bank\_Fix$y = NULL

#Bank\_Fix[Bank\_Fix=="unknown"]<-NA

#Bank\_Fix$default = str\_replace(Bank\_Fix$default, "unknown", "NA")

# Bank\_Fix$housing = str\_replace(Bank\_Fix$housing, "unknown", "NA")

# Bank\_Fix$marital = str\_replace(Bank\_Fix$marital, "unknown", "NA")

# Bank\_Fix$loan = str\_replace(Bank\_Fix$loan, "unknown", "NA")

#find out the data types

dplyr::glimpse(Bank\_Fix)

#Imputed NA values

tempData <- mice(Bank\_Fix,m=1,maxit=0,method ='logreg',seed=500)

Bank\_Fix\_Imp <- complete(tempData,1)

#Removing Outliers

Bank\_Fix\_Imp = Bank\_Fix\_Imp[-c(36044,40538,24092), ]

summary(Bank\_Fix\_Imp)

```

```{r}

#Some initial EDAs

summary(Bank\_Fix\_Imp)

#Split catergorical and continuous data

Bank\_Conti <- Bank\_Fix\_Imp[, !sapply(Bank\_Fix, is.factor)]

Bank\_categ <- Bank\_Fix\_Imp[, sapply(Bank\_Fix, is.factor)]

Bank\_Conti <- Bank\_Fix\_Imp[, !sapply(Bank\_Fix\_Imp, is.factor)]

Bank\_categ <- Bank\_Fix\_Imp[, sapply(Bank\_Fix\_Imp, is.factor)]

#Boxplots

boxplot(Bank\_Conti$age)

boxplot(Bank\_Conti$nr.employed)

boxplot(Bank\_Conti[,2,3])

#ScatterPlot

pairs(Bank\_Conti[,1:10], pch = 19)

```

```{r,EDA}

## Summary on Job variable, customers job status

summary(Bank\_Fix\_Imp$job)

catnames = names(Bank\_Fix\_Imp)[sapply(Bank\_Fix\_Imp, class) == "factor"]

Bank\_Fix\_Imp$term\_deposit = as.factor(Bank\_Fix\_Imp$term\_deposit)

spineplot(x = Bank\_Fix\_Imp$job, y = Bank\_Fix\_Imp$term\_deposit, xlab = "Job", ylab = "y",

main = "Job vs Y", col = c("lightblue", "coral"), xaxlabels = levels(Bank\_Fix\_Imp$job))

chisq.test(Bank\_Fix\_Imp$job, Bank\_Fix\_Imp$term\_deposit)

#CrossTable(Bank\_Fix\_Imp$job, Bank\_Fix\_Imp$term\_deposit)

##job is dependent on term\_deposit

##marital is dependent on term\_deposit

summary(Bank\_Fix\_Imp$marital)

chisq.test(Bank\_Fix\_Imp$marital, Bank\_Fix\_Imp$term\_deposit)

#CrossTable(Bank\_Fix\_Imp$marital, Bank\_Fix\_Imp$term\_deposit)

chisq.test(Bank\_Fix\_Imp$education, Bank\_Fix\_Imp$term\_deposit)

chisq.test(Bank\_Fix\_Imp$default, Bank\_Fix\_Imp$term\_deposit)

chisq.test(Bank\_Fix\_Imp$housing, Bank\_Fix\_Imp$term\_deposit)

chisq.test(Bank\_Fix\_Imp$loan, Bank\_Fix\_Imp$term\_deposit)

chisq.test(Bank\_Fix\_Imp$contact, Bank\_Fix\_Imp$term\_deposit)

chisq.test(Bank\_Fix\_Imp$month, Bank\_Fix\_Imp$term\_deposit)

chisq.test(Bank\_Fix\_Imp$day\_of\_week, Bank\_Fix\_Imp$term\_deposit)

chisq.test(Bank\_Fix\_Imp$poutcome, Bank\_Fix\_Imp$term\_deposit)

#CrossTable(Bank\_Fix\_Imp$poutcome, Bank\_Fix\_Imp$term\_deposit)

##marital is dependent on term\_deposit

## contact has some difference in "yes" and "no" among its categories (cellular and telephone). cellular with 14.7% and 5.2% for "yes" rsponse

## P-value of Chi-Square Test suggests that the variable "contact" has a relationship with response variable. We can keep this variable for final analysis

## Day of the week has some difference in "yes" and "no" among its categories. Most of the calls were on Thursday (12.1%) and other days are close to 10%

## P-value of Chi-Square Test suggests that the variable "day\_of\_week" has a relationship with response variable. We can keep this variable for final analysis

## 65.1% of customers where previous outcome was "Success" has a response of "yes"

## 14.2% of customers where previous outcome was "failure" has a response of "yes"

## 8.8% of customers who were not contacted has a response of "yes"

## P-value of Chi-Square Test suggests that the variable "poutcome" has a relationship with response variable. We can keep this variable for final analysis

##### We need to keep below variables in the predictive model

## job marital education contact month day\_of\_week poutcome

##### Below variables will not be included in the predictive model as there is no significance with response variable

## default housing loan

#multicolliniarity check

bank.model<-lm(age~duration+campaign+pdays+previous+emp.var.rate+cons.price.idx+cons.conf.idx+euribor3m+nr.employed, data=Bank\_Fix\_Imp)

summary(bank.model)

vif(bank.model)

# removed varaible euribor3m which has VIF 63.51

bank.model1<-lm(age~duration+campaign+pdays+previous+emp.var.rate+cons.price.idx+cons.conf.idx+nr.employed, data=Bank\_Fix\_Imp)

summary(bank.model1)

vif(bank.model1)

#removed varaible emp.var.rate which has VIF 24.12

bank.model2<-lm(age~duration+campaign+pdays+previous+cons.price.idx+cons.conf.idx+nr.employed, data=Bank\_Fix\_Imp)

summary(bank.model2)

vif(bank.model2)

#### We can remove variables emp.var.rate and euribor3m as these variables are highly correlated with nr.employed

corrgram(Bank\_Fix\_Imp, order=TRUE,

upper.panel=panel.cor, lower.panel=panel.pie, main="Bank data Correlogram")

## euribor3m and nr.employed are highly correlated (0.95)

## emp.var.rate and euribor3m are highly correlated (0.97)

## emp.var.rate and nr.employed are highly correlated (0.91)

summary(Bank\_Fix\_Imp$age)

Bank\_Fix\_Imp %>% ggplot(aes(x = age, fill = term\_deposit, color=term\_deposit)) + geom\_bar() + ggtitle("Distribution of Age") + xlab("Age") +

scale\_x\_continuous(breaks = seq(0, 100, 5))

ggplot(Bank\_Fix\_Imp, aes(x = term\_deposit, y=age, fill=term\_deposit)) + geom\_boxplot() + ggtitle("Distribution of Age") + xlab("Response") + ylab ("Age")

## The minimum and maximum values are 17 and 98 and distribution of age is slightly right screwed

## Highest concentration of values between 22 and 60 and distribution of values between 22 and 60 is normal

## Summary on duration variable

summary(Bank\_Fix\_Imp$duration)

Bank\_Fix\_Imp %>% ggplot(aes(x = duration, fill = term\_deposit)) + geom\_bar() + ggtitle("Distribution of Duration") + xlab("Duration") +

scale\_x\_continuous(breaks = seq(0, 5000, 300))

ggplot(Bank\_Fix\_Imp, aes(x = term\_deposit, y=duration, fill=term\_deposit)) + geom\_boxplot() + ggtitle("Distribution of Duration") + xlab("Response") + ylab ("Duration")

## The minimum and maximum values are 0 and 4918 sec and distribution of duration is highly right screwed

## "duration" and "term\_deposit"are pretty strongly associated. The longer duration is, the bigger prportion of people subscibe a term deposit.

## Summary on campaign variable. Number of contacts performed during this campaign and for this client

# summary(Bank\_Fix\_Imp$campaign)

# Bank\_Fix\_Imp %>% ggplot(aes(x = campaign, fill = term\_deposit)) + geom\_bar() + ggtitle("Distribution of Campaign") + xlab("Campaign")+

# scale\_x\_continuous(breaks = seq(0, 50, 1))

# ggplot(Bank\_Fix\_Imp, aes(x = term\_deposit, y=campaign, fill=term\_deposit)) + geom\_boxplot() + ggtitle("Distribution of campaign") + xlab("Response") + ylab ("campaign")

# aggregate(data.frame(count = Bank\_Fix\_Imp$campaign), list(value = Bank\_Fix\_Imp$campaign), length)

# Bank\_Fix\_Imp <- Bank\_Fix\_Imp %>%

# filter(campaign <= 10)

## The minimum and maximum values are 1 and 56 and distribution of campaign is right screwed

## looks like outlier in capaign varaible, after 8, the outcome is "no" for all observations. we can limit our study to 8

## Most of the campaign is on 1 and 2.

## There is a trend that the more number of campaign, the less percentage of clients substribe a term deposit, Expecially for campaign more than 3.

## Summary on pdays variable. Number of days that passed by after the client was last contacted from a previous campaign

summary(Bank\_Fix\_Imp$pdays)

Bank\_Fix\_Imp %>% ggplot(aes(x = pdays, fill = term\_deposit)) + geom\_bar() + ggtitle("Distribution of pdays") + xlab("pdays")

ggplot(Bank\_Fix\_Imp, aes(x = term\_deposit, y=pdays, fill=term\_deposit)) + geom\_boxplot() + ggtitle("Distribution of pdays") + xlab("Response") + ylab ("pdays")

aggregate(data.frame(count = Bank\_Fix\_Imp$pdays), list(value = Bank\_Fix\_Imp$pdays), length)

## most of the observations has value of 999 which mean these customers never contacted in the past.

## Summary on previous variable. How many number of contacts performed before this campaign

summary(Bank\_Fix\_Imp$previous)

Bank\_Fix\_Imp %>% ggplot(aes(x = previous, fill = term\_deposit)) + geom\_bar() + ggtitle("Distribution of previous") + xlab("previous")

ggplot(Bank\_Fix\_Imp, aes(x = term\_deposit, y=previous, fill=term\_deposit)) + geom\_boxplot() + ggtitle("Distribution of previous") + xlab("Response") + ylab ("previous")

aggregate(data.frame(count = Bank\_Fix\_Imp$previous), list(value = Bank\_Fix\_Imp$previous), length)

## The minimum and maximum values are 0 and 7. Most of the obserations with 0 value mean the customers never contacted in the past.

## Summary on emp.var.rate variable. We can remove this variable from our analysis because of multicolliniarity

summary(Bank\_Fix\_Imp$emp.var.rate)

Bank\_Fix\_Imp %>% ggplot(aes(x = emp.var.rate, fill = term\_deposit)) + geom\_bar() + ggtitle("Distribution of emp.var.rate") + xlab("emp.var.rate")

ggplot(Bank\_Fix\_Imp, aes(x = term\_deposit, y=emp.var.rate, fill=term\_deposit)) + geom\_boxplot() + ggtitle("Distribution of emp.var.rate") + xlab("Response") + ylab ("emp.var.rate")

## Summary on cons.price.idx variable. consumer price index - monthly indicator

summary(Bank\_Fix\_Imp$cons.price.idx)

Bank\_Fix\_Imp %>% ggplot(aes(x = cons.price.idx, fill = term\_deposit)) + geom\_bar() + ggtitle("Distribution of cons.price.idx") + xlab("cons.price.idx")

ggplot(Bank\_Fix\_Imp, aes(x = term\_deposit, y=cons.price.idx, fill=term\_deposit)) + geom\_boxplot() + ggtitle("Distribution of cons.price.idx") + xlab("Response") + ylab ("cons.price.idx")

## Overall, comsumer price index has some difference in "yes" and "no" among different values

## Minimum and maximum values are 92.20 and 94.77 respectively

## Summary on cons.conf.idx variable. consumer confidence index - monthly indicator

summary(Bank\_Fix\_Imp$cons.conf.idx)

Bank\_Fix\_Imp %>% ggplot(aes(x = cons.conf.idx, fill = term\_deposit)) + geom\_bar() + ggtitle("Distribution of cons.conf.idx") + xlab("cons.conf.idx")

ggplot(Bank\_Fix\_Imp, aes(x = term\_deposit, y=cons.conf.idx, fill=term\_deposit)) + geom\_boxplot() + ggtitle("Distribution of cons.conf.idx") + xlab("Response") + ylab ("cons.conf.idx")

## Overall, comsumer confidence index has some difference in "yes" and "no" among different values

## Minimum and maximum values are -50.8 and -26.9 respectively

## Summary on euribor3m variable. euribor 3 month rate - daily indicator

summary(Bank\_Fix\_Imp$euribor3m)

Bank\_Fix\_Imp %>% ggplot(aes(x = euribor3m, fill = term\_deposit)) + geom\_bar() + ggtitle("Distribution of euribor3m") + xlab("euribor3m")

ggplot(Bank\_Fix\_Imp, aes(x = term\_deposit, y=euribor3m, fill=term\_deposit)) + geom\_boxplot() + ggtitle("Distribution of euribor3m") + xlab("Response") + ylab ("euribor3m")

## Minimum and maximum values are 0.634 and 5.045 respectively

## Summary on nr.employed variable. We can remove this variable from our analysis because of multicolliniarity

summary(Bank\_Fix\_Imp$nr.employed)

Bank\_Fix\_Imp %>% ggplot(aes(x = nr.employed, fill = term\_deposit)) + geom\_bar() + ggtitle("Distribution of nr.employed") + xlab("nr.employed")

ggplot(Bank\_Fix\_Imp, aes(x = term\_deposit, y=nr.employed, fill=term\_deposit)) + geom\_boxplot() + ggtitle("Distribution of nr.employed") + xlab("Response") + ylab ("nr.employed")

### Delete variables which are multicollinear and correlated

##### Below variables will not be included in the predictive model as there is no significance with response variable

## default housing loan

###Remove emp.var.rate and nr.employed -multicollinear

View (Bank\_Fix\_Imp)

Bank\_Fix\_Imp$default <- Bank\_Fix\_Imp$housing <- Bank\_Fix\_Imp$loan <- NULL

Bank\_Fix\_Imp$emp.var.rate <- Bank\_Fix\_Imp$nr.employed <- NULL

```

```{r, split}

#Training and Test from imputed data

set.seed(100)

split\_percent = .70

trainIndices = sample(1:dim(Bank\_Fix\_Imp)[1],round(split\_percent \* dim(Bank\_Fix\_Imp)[1]))

train = Bank\_Fix\_Imp[trainIndices,]

test = Bank\_Fix\_Imp[-trainIndices,]

summary(train)

table(train$term\_deposit)

#Downsampling to make a better distribution of yes and no. Used the smote function for the downsampling to make it better

#set.seed(1000)

#down\_train <- downSample(x = train[, -ncol(train)],

# y = train$term\_deposit)

#table(down\_train$Class)

library(DMwR)

set.seed(9560)

smote\_train <- SMOTE(term\_deposit ~ ., data = train)

table(smote\_train$term\_deposit)

```

####PCA and LDA

```{r}

Bank\_Fix\_Imp\_Numeric <- cbind(Bank\_Fix\_Imp[,8:11],Bank\_Fix\_Imp[,13:15])

pc.result<-prcomp(Bank\_Fix\_Imp\_Numeric,scale=FALSE)

pc.scores<-pc.result$x

cor(pc.scores)

par(mfrow=c(1,2))

eigenvals<-(pc.result$sdev)^2

plot(1:7,eigenvals/sum(eigenvals),type="l",main="Scree Plot",ylab="Prop. Var. Explained")

cumulative.prop<-cumsum(eigenvals/sum(eigenvals))

plot(1:7,cumulative.prop,type="l",main="Cumulative proportion",ylim=c(0,1))

par(mfrow=c(1,1))

summary(pc.result)

##PC3 shows 99.99 cummulative proportion

#Adding the response column to the PC's data frame

pc.scores<-data.frame(pc.scores)

pc.scores$y<-Bank\_Fix\_Imp$term\_deposit

#Use ggplot2 to plot the first few pc's

library(ggplot2)

ggplot(data = pc.scores, aes(x = PC1, y = PC2)) +

geom\_point(aes(col=y), size=1)+ ggtitle("PCA of Responses")

loadinscores<-pc.result$rotation[,1]

var\_scores<-abs(loadinscores)

var\_scores\_ranked<-sort(var\_scores,decreasing = TRUE)

var\_scores\_ranked

library(ROCR)

library(MASS)

mylda<- lda(term\_deposit ~ duration+campaign+euribor3m+nr.employed+cons.price.idx+cons.conf.idx, data = smote\_train)

##myqda<- qda(term\_deposit ~ duration+pdays+campaign+euribor3m, data = Bank\_Fix\_Imp)

#confusion matrix and accuracy

prd<-predict(mylda, newdata = test)$class

table(prd,test$term\_deposit)

mean(prd==test$term\_deposit)

#For ROC curves

my.prd = predict(mylda, newdata = test)

#ROC Curve

prediction(my.prd$posterior[,2], test$term\_deposit) %>%

performance(measure = "tpr", x.measure = "fpr") %>%

plot() %>% title("ROC Graph for LDA Model")

#AUC

prediction(my.prd$posterior[,2], test$term\_deposit) %>%

performance(measure = "auc") %>%

.@y.values

#Overall Misclassification Error rate on the test is

##Accuracy (all correct / all) = TP + TN / TP + TN + FP + FN

##Misclassification (all incorrect / all) = FP + FN / TP + TN + FP + FN

# prd no yes

##no 35306 2710

##yes 1242 1930

##37236/41188=90.4%

```

All Models and Prediction

```{r}

#Simple Model

simple.log = glm(term\_deposit ~ Agegroup + job + marital + education + contact + month + day\_of\_week + poutcome + campaign + duration + pdays + previous + cons.price.idx + cons.conf.idx, family = "binomial", data = smote\_train)

summary(simple.log)

exp(cbind("Odds ratio" = coef(simple.log), confint.default(simple.log, level = 0.95)))

#pdays

#Testing the complex logistic model with different predictors as achieved from the EDA

simple.final.log = glm(term\_deposit ~ job + Agegroup + marital + education + contact + month + day\_of\_week + poutcome + duration\*poutcome + cons.price.idx\*cons.conf.idx + Agegroup\*marital, data = smote\_train, family = "binomial")

summary(simple.final.log)

#nr.employed + cons.price.idx\*cons.conf.idx

#smote Train data step wise

full.log = glm(term\_deposit ~ ., family = "binomial", data = smote\_train)

step.log = full.log %>% stepAIC(trace=FALSE)

summary(step.log)

exp(cbind("Odds ratio" = coef(step.log), confint.default(step.log, level = 0.95)))

vif(step.log)

#smote Train data lasso

dat.train.x <- model.matrix(term\_deposit ~ ., smote\_train)[,-1]

dat.train.y<-smote\_train$term\_deposit

cvfit <- cv.glmnet(dat.train.x, dat.train.y, family = "binomial", type.measure = "class", nlambda = 1000)

plot(cvfit)

coef(cvfit, s = "lambda.min")

#final lasso model

finalLassoModel<-glmnet(dat.train.x, dat.train.y, family = "binomial",lambda=cvfit$lambda.min)

cvfit$lambda.min

tLL = finalLassoModel$nulldev - deviance(finalLassoModel)

k = finalLassoModel$df

n = finalLassoModel$nobs

AICc = -tLL+2\*k\*2\*k\*(k+1)/(n-k-1)

##performing prediction using both models

dat.test.x<-model.matrix(term\_deposit ~ ., test)[,-1]

fit.pred.lasso <- predict(finalLassoModel, newx = dat.test.x, type = "response")

fit.pred.step<-predict(step.log,newdata=test,type="response")

fit.pred.final <- predict(simple.final.log, newdata = test, type = "response")

fit.pred.simple <- predict(simple.log, newdata = test, type = "response")

##Error rate simple model

mean(ifelse(fit.pred.simple > 0.5, "yes", "no") != test$term\_deposit)

#Error Rate for complex model

mean(ifelse(fit.pred.final > 0.5, "yes", "no") != test$term\_deposit)

#Error Rate for stepwise model

mean(ifelse(fit.pred.step > 0.5, "yes", "no") != test$term\_deposit)

#Error rate for lasso model

mean(ifelse(fit.pred.lasso > 0.5, "yes", "no") != test$term\_deposit)

cutoff<-0.5

class.lasso<-factor(ifelse(fit.pred.lasso>cutoff,"yes","no"),levels=c("no","yes"))

class.step<-factor(ifelse(fit.pred.step>cutoff,"yes","no"),levels=c("no","yes"))

class.simple<-factor(ifelse(fit.pred.simple>cutoff,"yes","no"),levels=c("no","yes"))

class.simpleFinal<-factor(ifelse(fit.pred.final>cutoff,"yes","no"),levels=c("no","yes"))

#Confusion Matrix for Lasso

conf.lasso<-table(class.lasso,test$term\_deposit)

print("Confusion matrix for LASSO")

## [1] "Confusion matrix for LASSO"

conf.lasso

#Confusion Matrix for Stepwise

conf.step<-table(class.step,test$term\_deposit)

print("Confusion matrix for Stepwise")

## [1] "Confusion matrix for Stepwise"

conf.step

#Confusion Matrix for Simple model

conf.simple = table(class.simple, test$term\_deposit)

print("Confusion Matrix for Simple")

conf.simple

#Confusion Matrix for Simple model

conf.final = table(class.simpleFinal, test$term\_deposit)

print("Confusion Matrix for Simple Final")

conf.final

#Confusion Matrix for Simple complex

sum(diag(conf.lasso))/sum(conf.lasso)

sum(diag(conf.step))/sum(conf.step)

sum(diag(conf.simple))/sum(conf.simple)

sum(diag(conf.final))/sum(conf.final)

#Lasso ROC Curve

results.lasso<-prediction(fit.pred.lasso, test$term\_deposit)

roc.lasso = performance(results.lasso, measure = "tpr", x.measure = "fpr")

plot(roc.lasso,colorize = TRUE)

abline(a=0, b= 1)

#stepwise ROC Curve

results.step = prediction(fit.pred.step, test$term\_deposit, label.ordering = c("no","yes"))

roc.step = performance(results.step, measure = "tpr", x.measure = "fpr")

#simple model

results.simple = prediction(fit.pred.simple, test$term\_deposit, label.ordering = c("no", "yes"))

roc.simple = performance(results.simple, measure = "tpr", x.measure = "fpr")

#Complex model

result.final = prediction(fit.pred.final, test$term\_deposit, label.ordering = c("no", "yes"))

roc.final = performance(result.final, measure = "tpr", x.measure = "fpr")

#Plot stepwise, Lasso, and gen simple model

plot(roc.lasso)

plot(roc.step, col = "orange", add = TRUE)

plot(roc.simple, col = "blue", add = TRUE)

plot(roc.final, col = "red", add = TRUE)

legend("bottomright",legend=c("Lasso","Stepwise","Simple Model Only", "Complex Model"),col=c("black","orange","blue", "red"),lty=1,lwd=1)

abline(a=0, b= 1)

#AUC Stepwise

auc.tmp.step = performance(results.step, "auc")

auc.step = as.numeric(auc.tmp.step@y.values)

auc.step

#AUC Lasso

auc.tmp.lasso = performance(results.lasso, "auc")

auc.lasso = as.numeric(auc.tmp.lasso@y.values)

auc.lasso

#AUC Simple

auc.tmp.simple = performance(results.simple, "auc")

auc.simple = as.numeric(auc.tmp.simple@y.values)

auc.simple

#AUC for Complex

auc.tmp.compex = performance(result.final, "auc")

auc.complex = as.numeric(auc.tmp.compex@y.values)

auc.complex

#AUC for the LDA

auc.tmp.lda = performance(prd, "auc")

auc.lda = as.numeric(auc.tmp.lda)

auc.lda

```

```{r}

#Random Forest attempt

bag.adv<-randomForest(term\_deposit ~ job + marital + education + contact + month + day\_of\_week + poutcome + cons.price.idx:cons.conf.idx,data=smote\_train,

mtry=2,importance =TRUE,ntree=100)

yhat.bag = predict(bag.adv, newdata=test)

plot(yhat.bag, test$term\_deposit,main="Bagged Model",xlab="Predicted",ylab="Test Set term")

abline (0,1)

library(tree)

mytree<-tree(term\_deposit ~ job + marital + education + contact + month + day\_of\_week + poutcome + cons.price.idx,smote\_train)

yhat.tree<-predict(mytree,newdata=test)

plot(yhat.tree,test$sales,main="Single Tree with 8 splits",xlab="Predicted",ylab="Test Set Term")

abline(0,1)

mytree<-tree(term\_deposit ~ job + marital + education + contact + month + day\_of\_week + poutcome + cons.price.idx,smote\_train,minsize=8,mindev=.0001)

yhat.tree<-predict(mytree,newdata=test)

plot(yhat.tree,test$sales,main="Single Tree with Deep Splits",xlab="Predicted",ylab="Test Set Term")

abline(0,1)

#Lets take a look at the predicted surface of our bagged model

predictors<-data.frame(TV=rep(0:300,51),radio=rep(0:50,each=301))

bag.full<-randomForest( sales ~ TV+radio,data=Adver , subset=index ,

mtry=2,importance =TRUE,ntree=100)

pred.surface<-matrix(predict(bag.full,predictors),301,51)

plot3d(TV,radio,sales)

surface3d(0:300,0:50,pred.surface,alpha=.4)

# Train model with knn and get importance of predictors

control <- trainControl(method="repeatedcv", number=10, repeats=3)

set.seed(100)

model.knn <- train(term\_deposit ~ ., data=Bank\_Fix\_Imp[,-c(2)], method="knn", trControl=control)

#Top 10 predictor ranking

importance.knn <- varImp(model.knn, scale=FALSE)

rank.knn <- importance.knn$importance

write.csv(rank.knn, "rank.knn.csv")

rank.knn <- read.csv("rank.knn.csv", header=TRUE)

colnames(rank.knn) <- c("Predictors", "Importance")

rank.knn <- rank.knn[order(rank.knn$Importance, decreasing = TRUE),]

ggplot(rank.knn[1:20,], aes(x=reorder(Predictors, Importance),y=Importance)) + geom\_bar(stat = "identity") + coord\_flip() + labs(title="Importance of Predictors", x="Predictors", y="Importance") +theme(axis.text.x=element\_text(hjust=0.5, vjust=0.5, size = 12))+theme(axis.text.y=element\_text(size = 12))

#KNN including all continuous predictors for model performance

set.seed(100)

iterations = 5

numks = 30

splitPerc = .70

kkk = c()

Sens = c()

Spec = c()

masterAcc = matrix(nrow = iterations, ncol = numks)

for(j in 1:iterations)

{

trainIndices = sample(1:dim(Bank\_Fix\_Imp)[1],round(splitPerc \* dim(Bank\_Fix\_Imp)[1]))

train = Bank\_Fix\_Imp[trainIndices,]

test = Bank\_Fix\_Imp[-trainIndices,]

smote\_train <- SMOTE(term\_deposit ~ ., data = train)

for(i in 1:numks)

{

classifications = knn(smote\_train[,c(1,11,12,13,14,16,17,18,19,20)],test[,c(1,11,12,13,14,16,17,18,19,20)],smote\_train$term\_deposit, prob = TRUE, k = i)

u <- union(classifications,test$term\_deposit)

t <- table(factor(classifications, u), factor(test$term\_deposit, u))

CM = confusionMatrix(t)

masterAcc[j,i] = CM$overall[1]

kkk[i] = CM$overall[1]

Sens[i] = CM$byClass[1]

Spec[i] = CM$byClass[2]

}

}

MeanAcc = colMeans(masterAcc)

plot(seq(1,numks,1),MeanAcc, type = "l")

combo = data.frame(k = 1:30, Sensitivity = Sens, Specificity = Spec, MeanAcc)

which.max(MeanAcc)

max(MeanAcc)

mean(combo$Sensitivity)

mean(combo$Specificity)

```