**ClassIFIcation on Bank Marketing Data Set to predict client’s intention of a term subscription**

Contributors:

* Tina Pai
* Yang Zhang
* Balaji Avvaru

# Introduction

Marketing campaigns develop strategies to enhance businesses. Companies use direct marketing by targeting segments of customers and contacting them to meet specific goals. Centralizing customer remote interactions in a contact center eases operational management of campaigns. Such centers allow communicating with customers through channels such as telephone (fixed-line or mobile).

The success of bank marketing campaign is predicted with customer features, campaign information and economic attributes. Here we attempt to analyze the eﬀect of telemarketing on attracting new clients in a ﬁnance industry by looking at the success of telemarketing calls for selling bank long-term deposits recorded by a Portuguese retail bank.

# Data Description

We obtained dataset from UCI [<https://archive.ics.uci.edu/ml/datasets/Bank+Marketing>] containing data originally sourced from the direct marketing campaign performed by a Portuguese bank. The bank collected data from May 2008 to November 2010 and the data consist of 41188 observations and 21 variables. The target response is a binary, categorical variable indicating whether a client subscribed to a term deposit or not.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Variable Name | Description | Min | Max | Median | Mean |
| age | Age | 17 | 98 | 38 | 40.02 |
| job | Type of job:  'admin.',  'blue-collar',  'entrepreneur',  'housemaid',  'management',  'retired',  'self-employed',  'services',  'student',  'technician',  'unemployed',  'unknown' | N/A | N/A | N/A | N/A |
| marital | Marital status  'divorced',  'married',  'single',  'unknown'; | N/A | N/A | N/A | N/A |
| education | Education  'basic.4y',  'basic.6y',  'basic.9y',  'high.school',  'illiterate',  'professional.course',  'university.degree',  'unknown' | N/A | N/A | N/A | N/A |
| default | Has credit in default?  'no',  'yes',  'unknown' | N/A | N/A | N/A | N/A |
| housing | Has housing loan?  'no',  'yes',  'unknown' | N/A | N/A | N/A | N/A |
| loan | Has personal loan?  'no',  'yes',  'unknown' | N/A | N/A | N/A | N/A |
| contact | Contact communication type  'cellular'  'telephone' | N/A | N/A | N/A | N/A |
| month | Last contact month of year  'Jan',  'Feb',  'Mar', ...,  'Nov',  'Dec' | N/A | N/A | N/A | N/A |
| day\_of\_week | Last contact day of the week  'Mon', 'Tue', 'Wed',  'Thu', 'Fri' | N/A | N/A | N/A | N/A |
| duration | Last contact duration, in seconds | 0.00 | 4918.0 | 180.0 | 258.3 |
| campaign | Number of contacts performed during this campaign and for this client | 1.0 | 56.0 | 2.0 | 2.568 |
| pdays | Number of days that passed by after the client was last contacted from a previous campaign | 0.0 | 999.0 | 999.0 | 962.5 |
| previous | Number of contacts performed before this campaign and for this client | 0.0 | 7.0 | 0.0 | 0.173 |
| poutcome | Outcome of the previous marketing campaign  'failure',  'nonexistent',  'success' | N/A | N/A | N/A | N/A |
| emp.var.rate | Employment variation rate - quarterly indicator | -3.4 | 1.4 | 1.1 | 0.0819 |
| cons.price.idx | Consumer price index - monthly indicator | 92.20 | 94.77 | 93.75 | 93.58 |
| cons.conf.idx | Consumer confidence index - monthly indicator | -50.8 | -26.9 | -41.8 | -40.8 |
| euribor3m | Euribor 3-month rate - daily indicator | 0.634 | 5.045 | 4.857 | 3.621 |
| nr.employed | Number of employees - quarterly indicator | 4964 | 5228 | 5191 | 5167 |
| Y | has the client subscribed a term deposit?  'yes', 'no' | N/A | N/A | N/A | N/A |

According to the statistics summary, there seems to be an issue of imbalance in our data with 36548 cases in ’No’ and 4640 cases in ’Yes’. A dataset is imbalanced if the classiﬁcation categories are not approximately equally represented. Imbalance in data, especially in class, always contributes to worse prediction. We used “upsample” technique to construct balanced dataset. The general idea of this method is to randomly sample (with replacement) the minority class to be the same size as the majority class.

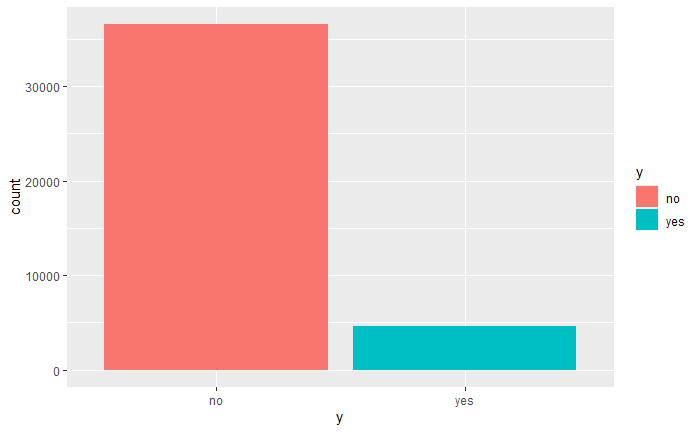


Figure: Count Plot of Response Variable (Y)

# Exploratory Analysis

For our analysis, we have used all variables. We separate variables into categorical and continuous and examine each group separately.

Figure 1-10 shows the spine plots of the categorical variables for each factor level by the response variable. It lets us see if a speciﬁc level or group of a factor has a higher or lower count than its counterparts that might contribute to the likelihood of subscribing a term deposit. The proportion of clients who subscribed to a term deposit seems to vary by job categories even for those with roughly the same sample size. For example, the proportion of subscribing to a term deposit is higher for clients who hold an administrative position and the proportion is lower for individuals who are self-employed. Thus, job possibly has an eﬀect on the likelihood of a client subscribing to a term deposit. We ran chi-square test on all categorical variables with response variable to see whether any of the predictors has dependency relationship with response variables. Reviewing the frequency tables and chi-square p-values for the remaining categorical variables , it appears that the variables “job”, “marital”, “education”, “contact”, “month”, “day\_of\_week” and “poutcome” could contribute, the proportions vary across the factor levels within each variable.

We decided to check whether there exists multicollinearity among continuous variables. Multicollinearity refers to the correlation between variables. To check the multicollinearity, we developed a correlation matrix on the variables. Looking at correlation matrix shows strong positive correlation between euribor3m and nr.employed (0.95), emp.var.rate and euribor3m (0.97) and emp.var.rate and nr.employed (0.91). There are negative correlations between previous and pdays (-0.59), previous and nr.employed (-0.50)

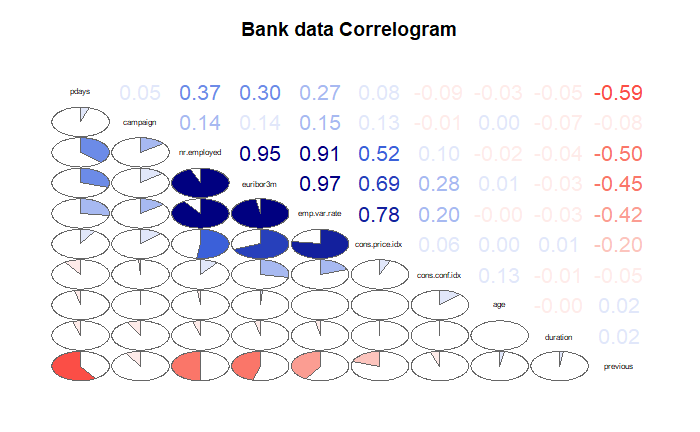


Figure: Correlation plot

We also plot out the distribution of the continuous variables and the factor levels of the categorical variables, as well as some multi-variable relationships. The figure below shows one observation we’ve made on the several continuous variables – they are all appear to be highly right skewed. For the modeling approaches that require normality, some transformation may need to be performed. For detailed EDA plots please reference to the appendix.

## 

Figure: Histogram for continuous variables

## Missing Data

Data has some unknown data, which is not treated as missing, So, considering unknown as one the factor in those variables.

# Objective 1 - Baseline model. Logistic Regression

For our ﬁrst two models, we ﬁt logistic regression to balanced and unbalanced datasets. We estimate the performance of both models on the same test dataset to see if one model has a better predictive power than the other.

Key Assumptions:

Before running logistic regression, we proceed making sure the following three key assumptions are met.

* Logistic regression requires the observations to be independent of each other. For this data set we do not have information if any of the observations recorded belonged to members from the same family. And hence we assume that all the observations are independent of each other.
* Logistic regression requires there to be little or no multicollinearity among the independent variables. This means that the independent variables should not be too highly correlated with each other. During the EDA we identified highly correlated variables and we make sure to remove them when building the models.
* Logistic regression requires linearity of independent variables and log odds. As this is not something we are able to control, we will assume we meet this assumption.

Using Balanced Training Data

We would like to ﬁnd out if we will get a diﬀerent logistic regression model if the training dataset is unbalanced compared to if the training dataset is balanced. We will repeat analyses using the balanced and the unbalanced training dataset.

Figure below shows the ROC curve of the training dataset (orange) and the ROC curve on the test dataset (green). The area under the curve (AUC) is commonly used to assess the prediction performance of the logistics model, the closer it’s to 1, the better the prediction is. The AUC based on the training data is 0.913 and 0.905 for the test data, which indicates that we did not overﬁt the model and the predictability power of the model is quite high.

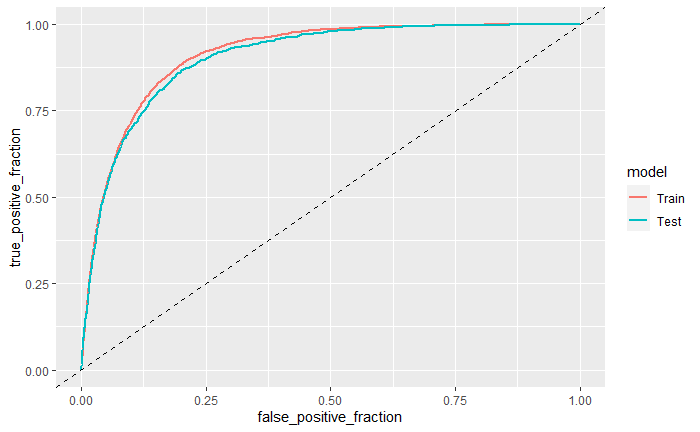


Figure: ROC curves for the balanced training dataset and the test dataset

The classiﬁcation tables (Table 1 and Table 2) can also be used to assess how well the model performs in classifying the dichotomous response variable. The accuracy is measured by its sensitivity (the ability to predict an event correctly) and speciﬁcity (the ability to predict a nonevent correctly). At the probability level of 0.5, the model can correctly classify 85.66% of the event (not subscribed for term deposit) and 81.25% of the non-event (subscribed for term deposit), with an overall rate of 83.45% on the training data. For the test data, the sensitivity is 85.293%, the speciﬁcity is 79.34% and the overall accuracy increase to 84.61.

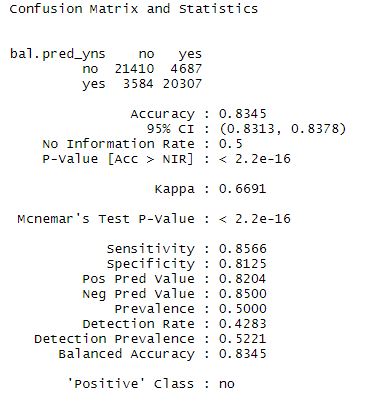
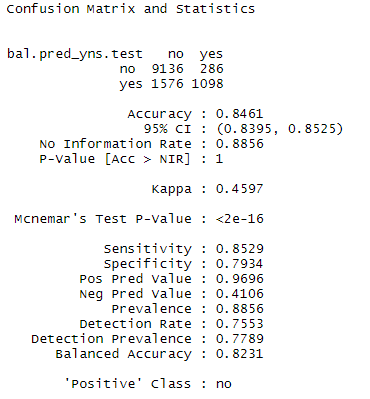


Table 1: Confusion matrix with test dataset Table 2: Confusion matrix with training dataset

Using Imbalanced Dataset

Using the resulting model that is built with the unbalanced dataset, we examine the ROC curve of the training dataset and also on the same test dataset to determine the predictability power of the model.

Figure below illustrates the ROC curve on the training dataset and test dataset. The AUC is 0.9175 for the model based on the training data and 0.9112 for the test data. The values are slightly higher than those that are obtained from the balanced model respectively.

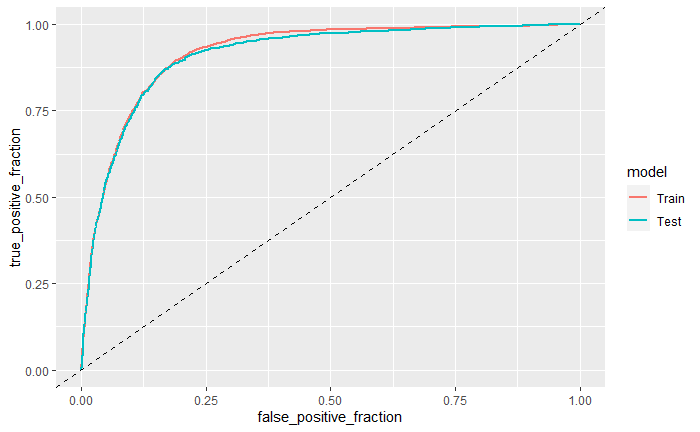


Figure: ROC curves for the unbalanced training dataset and the test dataset

The major difference between training on balanced and training on imbalanced data is the probability threshold at which we can achieve a reasonable specificity. The classiﬁcation table in Figure 27 (top) displays the sensitivity and the speciﬁcity of the model. At the probability level of 0.5, the model can correctly classify 97.56% of the event (not subscribed for term deposit) and 37.29% of the non-event (subscribed for term deposit), with an overall rate of 90.66% on the training data. For the test data, the sensitivity is 97.53%, the speciﬁcity is 37.37% and the overall accuracy increase to 90.76.

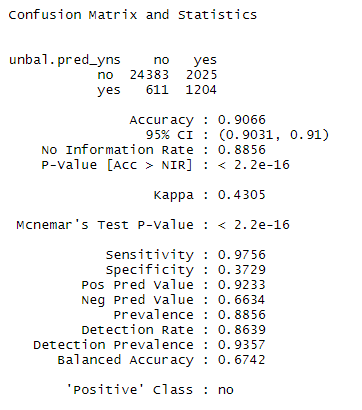
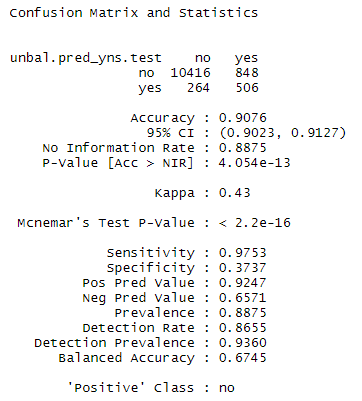


Table 1: Confusion matrix with test dataset Table 2: Confusion matrix with training dataset

Compared to the prior model with the balanced training data, the sensitivity is almost higher and the speciﬁcity is lower, which makes sense since the latter model is built based on the disproportionate ratio of ‘no’ and ‘yes’ responses, having many more observations of ‘no’ than ‘yes’. Thus, the model can more accurately classify the events resulting in higher sensitivity. On the other hand, the speciﬁcity is low due to the small number of ‘yes’ records in the training dataset. Thus, there is not enough information for the model to correctly classify the event.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Training Dataset | | | | Test Dataset | | | |
| Model | Accuracy | Sensitivity | Specificity | AUC | Accuracy | Sensitivity | Specificity | AUC |
| LR (balanced dataset) | 83.45 | 85.66 | 81.25 | 0.913 | 84.61 | 85.29 | 79.34 | 0.905 |
| LR (unbalanced dataset) | 90.66 | 97.56 | 37.29 | 0.9175 | 90.76 | 97.53 | 37.37 | 0.9175 |

Variable Selection

We ran a manual selection by starting with all the explanatory variables (excluding three variables which are independent of response variable based on Chi-square test) and having “y”(customer subscribed for term deposit or not) as the outcome in a logistic regression model, we took off the nonstatistical significant variables and then we adjusted for multicollinearity. The manual model was the one used in the analysis above, and as mentioned, the test sensitivity is 85.29%, speciﬁcity is 79.34% and the overall accuracy 84.61%. The model formula is:

y ~ age + job + marital + education + month + day\_of\_week + duration + campaign + pdays + previous + poutcome + cons.conf.idx

We also ran a forward selection which gave us the following model with test accuracy 85.8%, sensitivity 85.4%, specificity 88.6%:

y ~ duration + nr.employed + month + poutcome + emp.var.rate +

cons.price.idx + job + contact + euribor3m + default + day\_of\_week +

pdays + campaign + cons.conf.idx

As the metrics for the model given by the forward selection were better, we decided to proceed with the forward selection model.

Lack of fit test

We use the Hosmer and Lemeshow Goodness-of-Fit test with the null hypothesis that the ﬁtted model is a good fit. The output p-value is a number between 0 and 1 with higher values indicating a better ﬁt. The p-value we obtain from the test is <0.0001, which is statistically signiﬁcant and implies that the null hypothesis should be rejected.

Since we see that the test indicates that the model is not a good fit, we will add additional complexity and try other classification algorithms in the next section.

Hosmer and Lemeshow goodness of fit (GOF) test

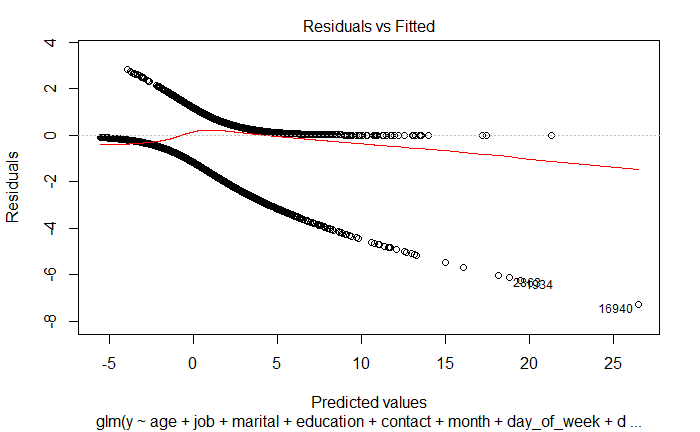
data: model.main1$y, fitted(model.main1)

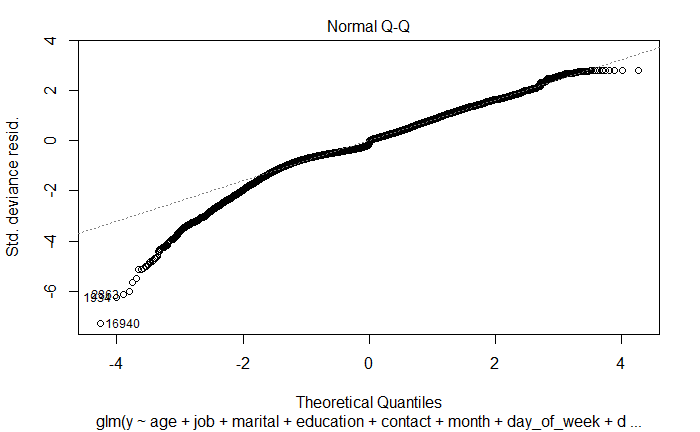
X-squared = 3501.9, df = 8, p-value < 2.2e-16

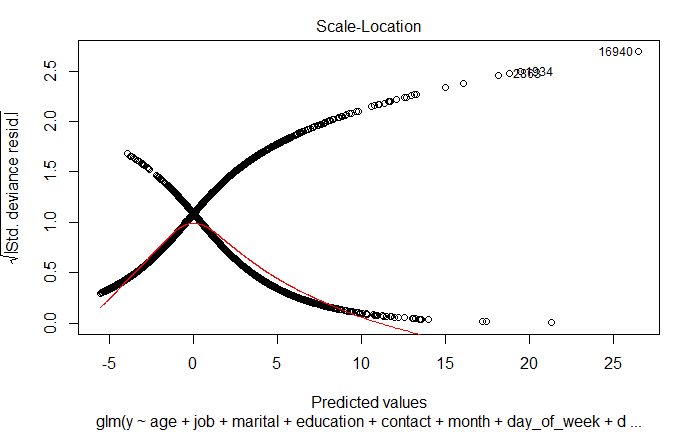
Figure: Hosmer and Lemeshow Goodness-of-Fit Test Result

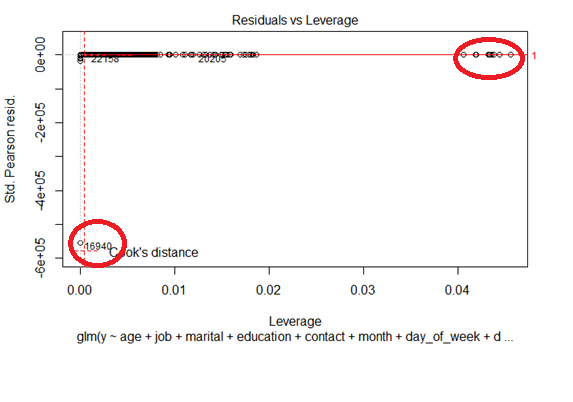
Influential Points and Residual diagnostics

Plots here help us examine Cook’s D graph:









When checking Cook’s D plot, if observations are outside the Cook’s distance (meaning they have a high Cook’s distance score) the observations are influential to the regression results. In this case, from the Cook’s D plot above, we see

* There are a few high leverage points, but the Cook’s distance for these points is not high. These points are likely not cause for concern.
* Observation 16940 (observation with max duration of 4918 seconds) looks like an outlier, but it is a low leverage point. This point should not cause influence on the fit. Refit the model without this observation which slightly increases the accuracy of the model.

Parameter Interpretation and Confidence Intervals

Figure 12 displays the coeﬃcient estimates for each factor level and Figure 13 displays the odd ratio estimates and the conﬁdence intervals for each level. Here is our interpretation of a subset of most interesting estimates.

# age: Holding all other explanatory variables fixed, odds of a client subscribing a term-deposit is 1.0024 times higher than a client 1 year younger. The 95% confidence interval is [0.9994, 1.00549].

# job (admin vs blue-collar): Holding all other explanatory variables fixed, the odds ratio of subscribing to a term-deposit for clients with admin job title relative to clients who are blue collar job tittle is 0.662. The 95% confidence interval is [0.601, 0.731].

# Marital (divorced vs single): Holding all other explanatory variables fixed, the odds ratio of subscribing to a term-deposit for divorced clients relative to single clients is 1.355. The 95% confidence interval is [1.231, 1.491].

# Education (basic 4-year vs University degree): Holding all other explanatory variables fixed, the odds ratio of subscribing to a term-deposit for clients with basic 4-year education relative to clients with university degree is 1.475. The 95% confidence interval is [1.312, 1.658].

# Contact (Cellular vs Telephone): Holding all other explanatory variables fixed, the odds ratio of subscribing to a term-deposit for clients using cellular phone relative to clients using telephone is 0.183. The 95% confidence interval is [0.169, 0.197].

# Month (April vs August): Holding all other explanatory variables fixed, the odds ratio of subscribing to a term-deposit for clients who last contacted in April relative to clients who last contacted in August is 0.130. The 95% confidence interval is [0.115, 0.148].

# Duration: Holding all other explanatory variables fixed, odds of a client subscribing a term-deposit is 1.0062 times higher than a client whose last contact duration is 1 second less. The 95% confidence interval is [1.0061, 1.0063].

# Campaign: Holding all other explanatory variables fixed, odds of a client subscribing a term-deposit is 0.9066 times higher than a client who contacted 1 time less during current campaign. The 95% confidence interval is [0.891, 0.922].

# pdays: Holding all other explanatory variables fixed, odds of a client subscribing a term-deposit is 0.99841 times higher than a client whose previous contact 1 day less. The 95% confidence interval is [0.9980, 0.9987].

# previous: Holding all other explanatory variables fixed, odds of a client subscribing a term-deposit is 1.408 times higher than a client who contacted 1 time less during previous campaign. The 95% confidence interval is [1.266, 1.567].

# poutcome (Failure vs Success): Holding all other explanatory variables fixed, the odds ratio of subscribing to a term-deposit for clients who did not subscribe in the previous campaign relative to clients who subscribed in the previous campaign is 2.053. The 95% confidence interval is [1.425, 2.957].

# Consumer Confidence index: Holding all other explanatory variables fixed, odds of a client subscribing a term-deposit is 1.0936 times higher than a client whose confidence index is 1 less. The 95% confidence interval is [1.086, 1.101].

Conclusion of Objective 1

Our simple logistic regression when trained on the balanced training set worked to give us decent predictions, but because it was not meeting the test of good fit, we will try adding complexity and other classification algorithms.

# Objective 2 - Additional Models

Logistic Regression model (LRM) with interaction

To add complexity to our model, we investigated what interactions may be significant. To do this, we created a list of all the possible pair combinations of the variables and used the Wald’s Z test on whether the interaction term’s coefficient was significantly different from 0.

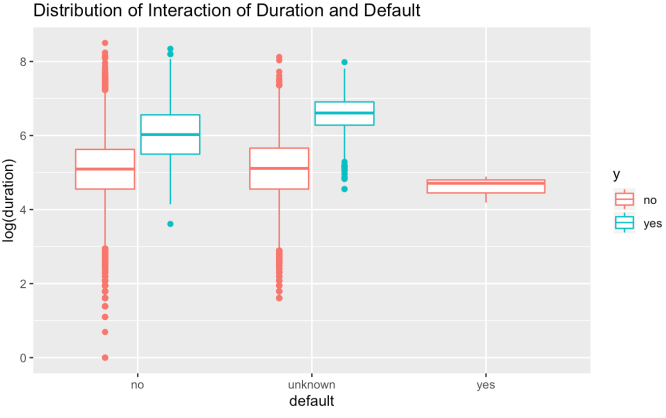
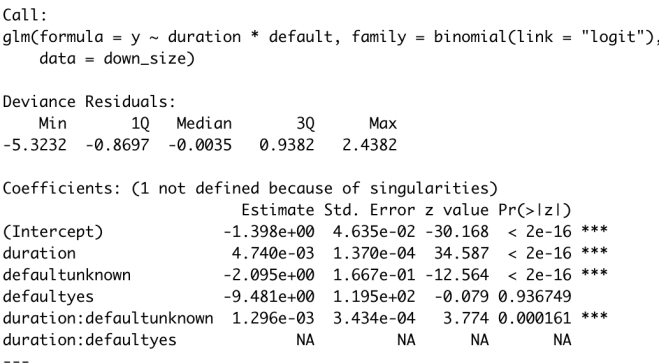


Figure: The interaction between duration and default is significant (p =.00016).

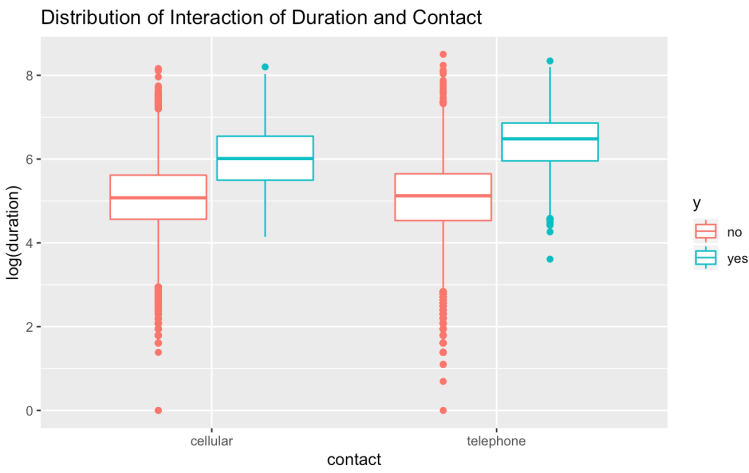
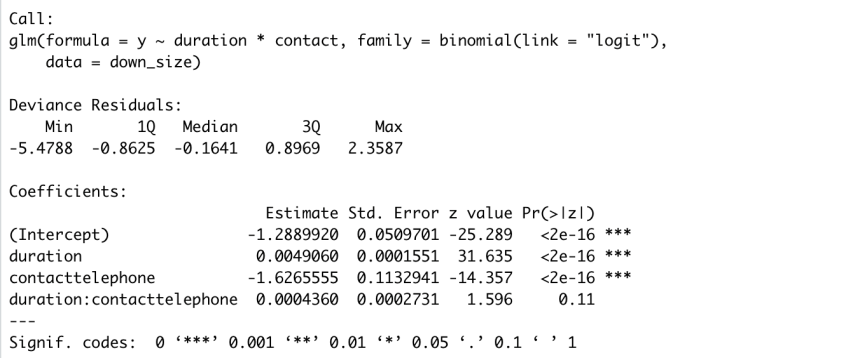


Figure: The interaction between duration and contact is not significant (p = .11)

Taking into consideration the multiple tests, we chose .0001 as the p value threshold for significance in the interaction. We kept the interactions with p<.0001 and included them into the model.



Figure: The list of significant interaction terms

The complex model has some manually selected variables plus these selected interactions. The model is:

model.interaction1 <- glm(y ~ duration \* nr.employed + month + poutcome + emp.var.rate + cons.price.idx + job + contact + euribor3m + default + day\_of\_week + pdays + campaign + cons.conf.idx + duration\*nr.employed + duration\*poutcome + duration \* emp.var.rate + duration \* cons.price.idx + duration \* job + duration \* euribor3m + duration \* cons.conf.idx + nr.employed \* emp.var.rate + nr.employed \* euribor3m + nr.employed \* campaign + nr.employed \* cons.conf.idx + month \* cons.price.idx + month \* job + month \* contact + month \* default + month \* campaign + poutcome \* emp.var.rate + poutcome \* job + poutcome \* euribor3m + poutcome \* pdays + poutcome \* cons.conf.idx + emp.var.rate \* euribor3m + emp.var.rate \* campaign + emp.var.rate \* cons.conf.idx + cons.price.idx \* contact + cons.price.idx \* pdays + cons.price.idx \* cons.conf.idx + euribor3m \* campaign + euribor3m \* cons.conf.idx + default \* pdays + default \* campaign + default \* cons.conf.idx, data=bank, family="binomial")

The AUC of this model is .937. We picked the threshold of .5 to get a sensitivity of 85.7%, specificity of 89.7% and overall accuracy of 86.2%.

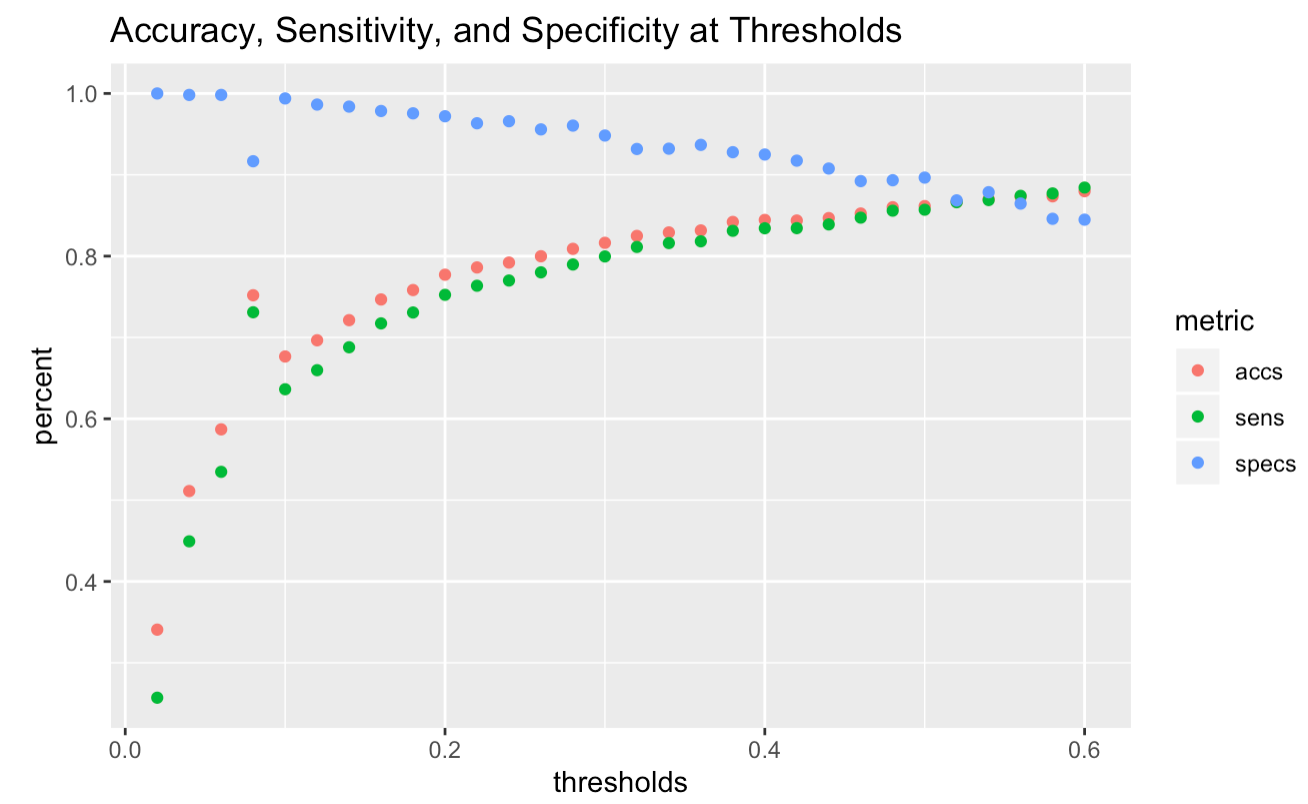


Figure: Accuracy, sensitivity, and specificity across various thresholds

Linear Discriminant Analysis model (LDA)

Because the response variable is categorical, we can also use the series of discriminant analysis methods (e.g., LDA, QDA) to construct a competing model.

All the continuous variables are included as explanatory variables for LDA method. We also visited the assumptions required and normality is one of them. Based on the previous exploration data analysis results, we saw that a few variables (e.g., age, duration, campaign) are plotted right skewed in histograms. We formed a model with original variables and another model with log transformation performed.

The below table and figure show the performance metrics and ROC curve comparisons for the two models: LDA with no log transformation has an accuracy of 85.6% but AUC of 0.912, LDA with log transformation has an accuracy of 81% which is lower than the previous one, but higher AUC (0.914). There’s no firm winning model but the log transformed one is preferred for high AUC and specificity.

Another experiment is conducted using QDA modeling on all continuous variables. The performance matric is less preferred in both the accuracy/specificity measurements and the AUC. The preferred discriminant analysis model is still LDA.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Test Dataset | | | |
| Model | Accuracy | Sensitivity | Specificity | AUC |
| LDA | 85.6% | 86.9% | 84.1% | 0.912 |
| LDA log transformed | 81% | 80% | 89% | 0.914 |
| QDA | 82% | 88.3% | 75.6% | 0.899 |

|  |  |
| --- | --- |
|  |  |

Figure: (1) ROC curve comparisons of LDA before and after log transformation (2) ROC curve comparisons of LDA and QDA method.

Non-parametric model. KNN (K nearest Neighbor)

We are constructing another competing model using nonparametric model approach. We start with K nearest neighbor approach with a fixed k selection.

There’s no formal distribution assumption here. However parameter of k is usually decided ahead of time. We start with using k=3 and get the performance metric of an accuracy of 81.5%, with sensitivity of 86.7% and specificity of 76.3%. A later iteration is done to search for the best k number choice. From the figure below, we can see that the sensitivity is highest with small k number, while the specificity is lowest. To balance the sensitivity and specificity while keeping accuracy high, we pick k=5 as a benchmark model here.

For k=5, we get an accuracy measured to be 83.6%, which is higher than the previous model. Both models have a much lower AUC compared with previous models.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Test Dataset | | | |
| Model | Accuracy | Sensitivity | Specificity | AUC |
| KNN (k=3) | 81.5% | 86.7% | 76.3% | 0.616 |
| KNN (k=5) | 83.6% | 83.6% | 82.9% | 0.647 |

|  |  |  |
| --- | --- | --- |
| k | k | k |

Figure: Iteration on value changes of (1) accuracy (2) Sensitivity (3) Specificity with number k selection

Non-parametric model. Random Forest (RF)

Another category of non-parametric model is random forest. Random forest allows both continuous and categorical variables to be included into modeling. We expect to boost performance of the model by using bagging criteria and be able of use all available information.

First we use the original dataset without balancing and the resulted accuracy and AUC appear to be high (91.2% and 0.94) shown as below. However if we take a look at the specificity is only 52%. This is consistent with our Objective 1 modeling, indicating that balancing action is needed to satisfy all performance metric standards.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Test Dataset | | | |
| Model | Accuracy | Sensitivity | Specificity | AUC |
| RF (balanced dataset)  All cons | 89.2% | 90.2% | 81.5% | 0.939 |
| RF (unbalanced dataset) | 91.2% | 96% | 52% | 0.94 |

To include also all categorical variables into random forest modeling, we have to convert them to factor levels first. The following table shows the comparison between using only continuous variables and using all variables. Reading from the ROC curves the sensitivity of using all variables is higher than only using continuous variable. It also has a higher AUC (0.947), with a lower specificity. A benchmark model with only the selected variables from Objective 1 is also provided. The performance is very similar to the RF model with only continuous variable.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Test Dataset | | | |
| Model | Accuracy | Sensitivity | Specificity | AUC |
| RF  All continuous | 89.2% | 90.2% | 81.5% | 0.939 |
| RF  All variables | 91% | 93% | 72% | 0.947 |
| RF  Manual | 89.1% | 88.1% | 86% | 0.941 |

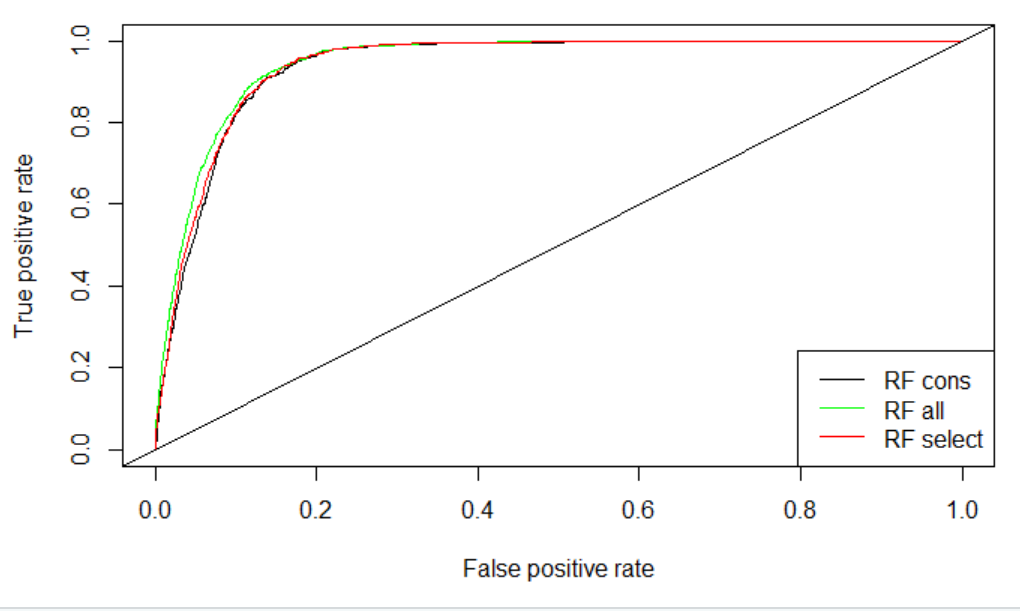


Figure: ROC curve comparisons of Random forest with (1) all continuous variables (2) all variables (3) selected variables from objective 1

Comparison of all models and Conclusion

In the table and figure below, we list out the related performance metric for the optimal models we’ve construct with each approach. We can do a cross-comparison on their performance on the same test dataset, with measurements on accuracy/sensitivity/specificity and also ROC curves/AUC.

For all dimensions we could observe that Random Forest and Logistic Regression models lead the performance metrics. There are no obvious differences between the two, but random forest has a slightly higher overall accuracy and AUC. Relatively, both Specificity measurements and ROC curve show that final logistic regression model has a higher specificity.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Test Dataset | | | |
| Model | Accuracy | Sensitivity | Specificity | AUC |
| LDA | 85.6% | 86.9% | 84.1% | 0.912 |
| KNN | 83.6% | 83.6% | 82.9% | 0.647 |
| RF | 87.8% | 88.1% | 86% | 0.941 |
| Logistic Regression  (Final) | 86.3% | 86.2% | 87.1% | 0.937 |

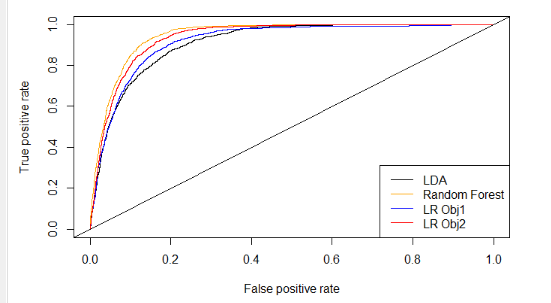


Figure: ROC curve comparisons of different approaches (1) LDA (2) Random Forest (3) Logistic regression model from objective 1 (4) Final logistic regression model

There’s no universal standard of choosing the “winning” model. We put our preference on the logistic regression model for the following reasons:

* Logistic regression has linearity assumption. We did not find obvious higher order relationship in our EDA analysis. It is consistent with the comparable performance of random forest and logistic regression. Under this condition, logistic regression can provide more interpretable results.
* The response variable in this problem is whether the client subscribed a term deposit. What makes this model meaningful is to find out the client so they say “yes” to subscribe. Therefore it is important to make the right call to find potential customers – do not miss who potentially will say “yes”. So we must treat the specificity measurement with some priority. In that aspect logistic regression may have a merit.

As a conclusion, we choose a logistic regression model as our solution to this problem to help with both interpretation and prediction of the bank marketing problem.

**APPENDIX**

Spine plots of Categorical Variables

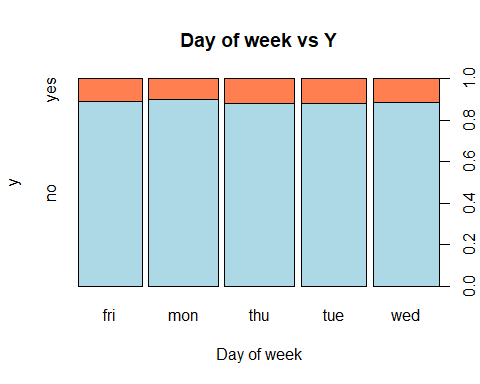


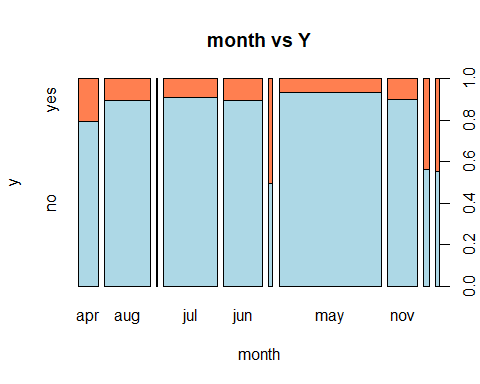
Figure 1: Spine plot of day of Week with response variable

Figure 2: Spine plot of month with response variable

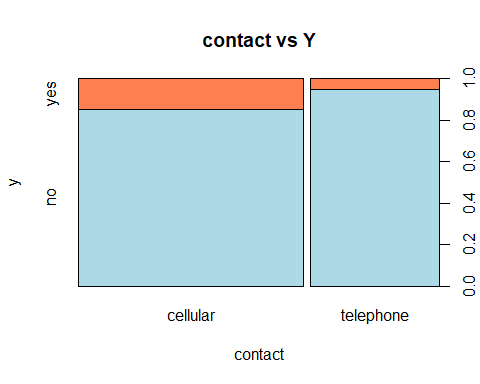


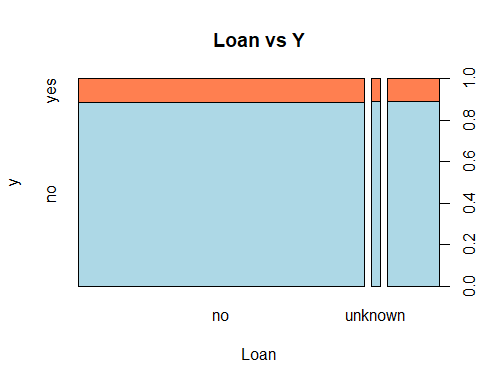
Figure 3: Spine plot of contact with response variable

Figure 4: Spine plot of Loan with response variable

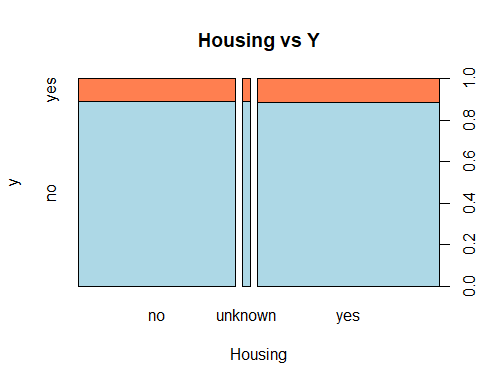


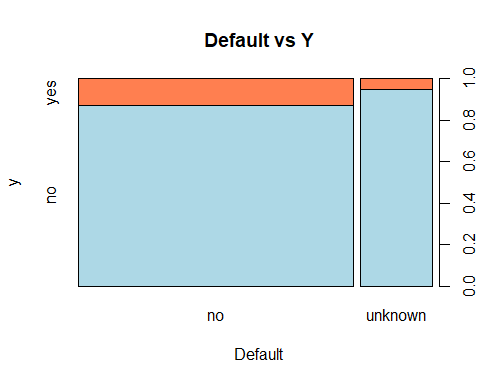
Figure 5: Spine plot of Housing with response variable

Figure 6: Spine plot of Default with response variable

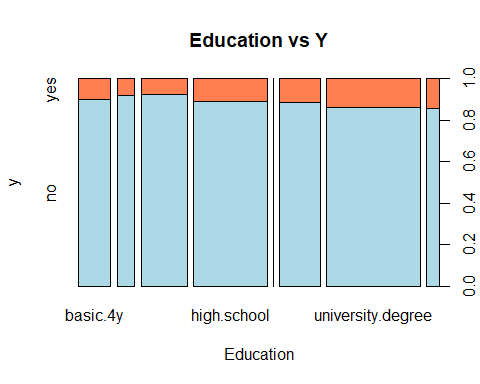


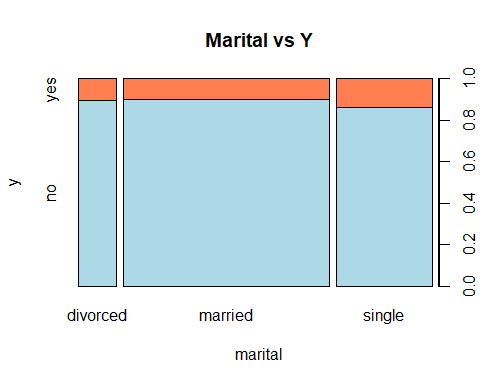
Figure 7: Spine plot of Education with response variable

Figure 8: Spine plot of Marital with response variable

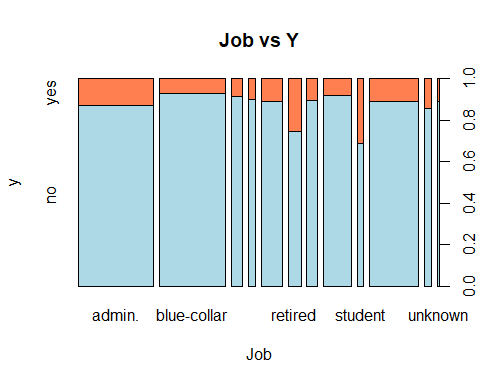


Figure 9: Spine plot of job with response variable

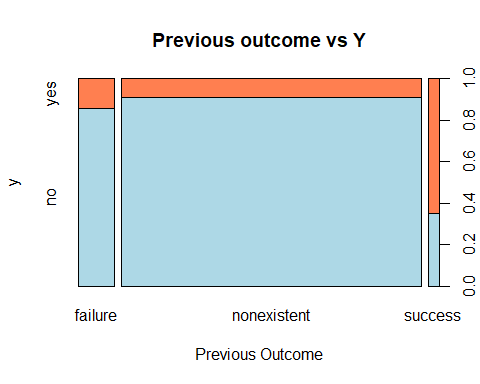


Figure 10: Spine plot of previous with response variable

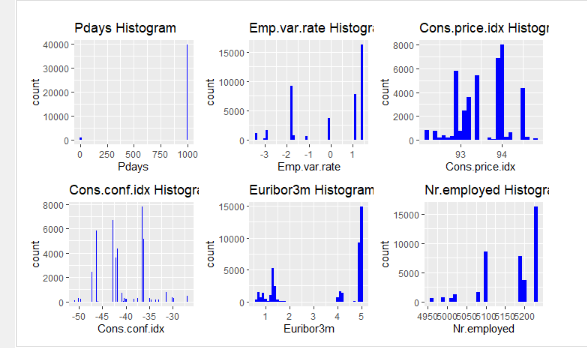


Figure 11A: Histogram plots of rest of explanatory variables

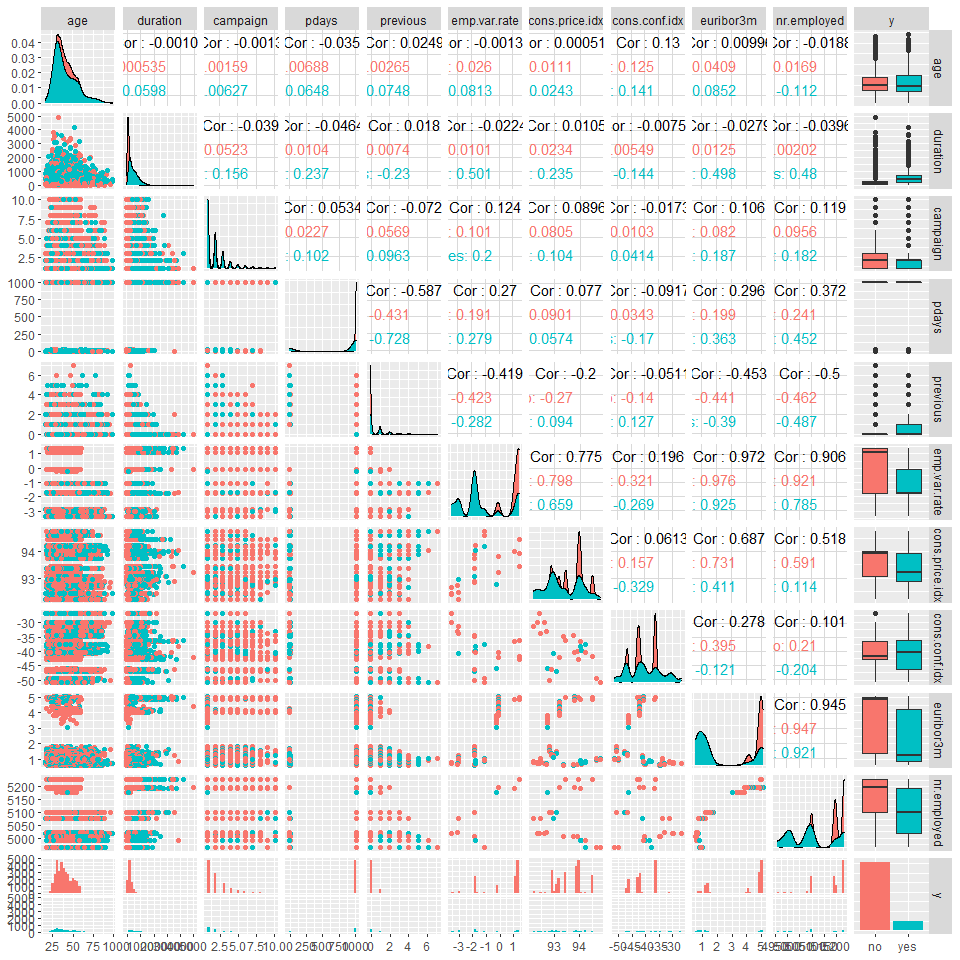


Figure 11B: Pair wise plots for continuous variables with response variable

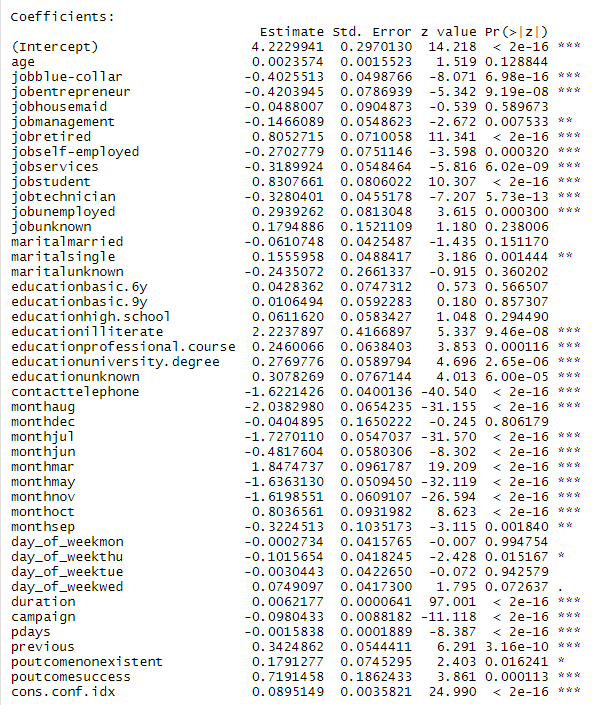


Figure 12: Coeﬃcient estimates

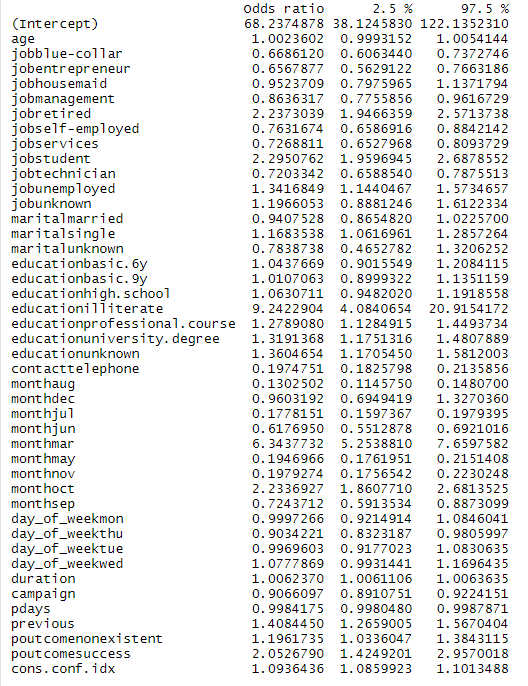


Figure 13: Odds Ratio estimates and conﬁdence intervals

Code:

# Libraries

```{r }

library(plotROC)

library(GGally)

library(mlr)

library(gmodels)

library(mosaic)

library(ggmosaic)

library(dplyr)

library(ggplot2)

library(tidyr)

#library(SDMTools)

library(readr)

library(digest)

library(ISLR)

library(leaps)

library( Matrix)

library(foreach)

library(glmnet)

library(VIM)

library(mice)

library(corrgram)

library(tidyverse)

#library(limma)

library(gridExtra)

library(MASS)

library(mvtnorm)

library(class)

library(caret)

library(e1071)

library(ResourceSelection)

library(car)

```

## Load the data

```{r }

setwd("C:/Users/BelajiAvvaru/Desktop/MSDS 6372/Project 2/")

#bankdata<- read.csv(file.choose())

bankdata<-read.csv("bank-additional-full.csv",header=T, sep=";")

head(bankdata)

summary(bankdata$y)

```

## number of rows, number of columns, summary and structure of bank data

```{r }

nrow(bankdata)

ncol(bankdata)

summary(bankdata)

str(bankdata)

```

# missing data

```{r }

#Display missing-data patterns

md.pattern(bankdata, plot=TRUE, rotate.names = TRUE)

#Display missing-data in a bar-plot

mice\_plot <- aggr(bankdata, col=c('navyblue','yellow'),

numbers=TRUE, sortVars=TRUE,

labels=names(bankdata), cex.axis=.7,

gap=3, ylab=c("Missing data","Pattern"))

```

## Summary of response variable

```{r }

summary(bankdata$y)

ggplot(bankdata, aes(x = factor(y), fill=y)) +

geom\_bar() + xlab("y")

```

## EDA on categorical variables

```{r }

catnames = names(bankdata)[sapply(bankdata, class) == "factor"]

```

```{r }

## Summary on Job variable, customers job status

summary(bankdata$job)

spineplot(x = bankdata$job, y = bankdata$y, xlab = "Job", ylab = "y",

main = "Job vs Y", col = c("lightblue", "coral"), xaxlabels = levels(bankdata$job))

CrossTable(bankdata$job, bankdata$y)

chisq.test(bankdata$job, bankdata$y)

```

## Overall, job has some difference in "yes" and "no" among its categories

## Very low p-value of Chi-Square Test suggests that the variable "job" has a relationship with response variable. We can add keep this variable for final analysis

```{r }

## Summary on marital variable, customers marital status

summary(bankdata$marital)

spineplot(x = bankdata$marital, y = bankdata$y, xlab = "marital", ylab = "y",

main = "Marital vs Y", col = c("lightblue", "coral"), xaxlabels = levels(bankdata$marital))

chisq.test(bankdata$marital, bankdata$y)

CrossTable(bankdata$marital, bankdata$y)

```

## Singles (14.0%) slightly more than other levels divorced (10.3%) or married customers (10.2%) in regards to response "yes".

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

```{r }

## Summary on education variable, customers education level

summary(bankdata$education)

spineplot(x = bankdata$education, y = bankdata$y, xlab = "Education", ylab = "y",

main = "Education vs Y", col = c("lightblue", "coral"), xaxlabels = levels(bankdata$education))

CrossTable(bankdata$education, bankdata$y)

chisq.test(bankdata$education, bankdata$y)

```

## Education has some difference in "yes" and "no" among its categories

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

```{r }

## Summary on default variable, whether the customer is in default list or not

summary(bankdata$default)

spineplot(x = bankdata$default, y = bankdata$y, xlab = "Default", ylab = "y",

main = "Default vs Y", col = c("lightblue", "coral"), xaxlabels = levels(bankdata$default))

CrossTable(bankdata$default, bankdata$y)

chisq.test(bankdata$default, bankdata$y)

```

## Only 3 observations with response “yes” and other observations either "no" or "unknown".

## This variable can be revmoed from final analysis even though Chi-Square Test suggests that the variable "default" has a relationship with response variable

```{r }

## Summary on housing variable, whether the customers has housing loan with the bank or not

summary(bankdata$housing)

spineplot(x = bankdata$housing, y = bankdata$y, xlab = "Housing", ylab = "y",

main = "Housing vs Y", col = c("lightblue", "coral"), xaxlabels = levels(bankdata$housing))

CrossTable(bankdata$housing, bankdata$y)

chisq.test(bankdata$housing, bankdata$y)

```

## P-value of Chi-Square Test suggests that the variable "housing" has a no relationship with response variable. We can remove this variable for final analysis

```{r }

## Summary on Loan variable, whether the customers has loan with bank or not

summary(bankdata$loan)

spineplot(x = bankdata$loan, y = bankdata$y, xlab = "Loan", ylab = "y",

main = "Loan vs Y", col = c("lightblue", "coral"), xaxlabels = levels(bankdata$loan))

CrossTable(bankdata$loan, bankdata$y)

chisq.test(bankdata$loan, bankdata$y)

```

## P-value of Chi-Square Test suggests that the variable "loan" has a no relationship with response variable. We can remove this variable for final analysis

```{r }

## Summary on Contact variable, how customers where contacted during campaign

summary(bankdata$contact)

spineplot(x = bankdata$contact, y = bankdata$y, xlab = "contact", ylab = "y",

main = "contact vs Y", col = c("lightblue", "coral"), xaxlabels = levels(bankdata$contact))

CrossTable(bankdata$contact, bankdata$y)

chisq.test(bankdata$contact, bankdata$y)

```

## 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

```{r }

## Summary on Month variable

summary(bankdata$month)

spineplot(x = bankdata$month, y = bankdata$y, xlab = "month", ylab = "y",

main = "month vs Y", col = c("lightblue", "coral"), xaxlabels = levels(bankdata$month))

CrossTable(bankdata$month, bankdata$y)

chisq.test(bankdata$month, bankdata$y)

```

## month has some difference in "yes" and "no" among its categories. Most of the calls were in May but there is higher coversion in March, September, October, and in December. We also notice that no contact has been made during January and February.

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

```{r }

## Summary on day\_of\_week variable

summary(bankdata$day\_of\_week)

spineplot(x = bankdata$day\_of\_week, y = bankdata$y, xlab = "Day of week", ylab = "y",

main = "Day of week vs Y", col = c("lightblue", "coral"), xaxlabels = levels(bankdata$day\_of\_week))

CrossTable(bankdata$day\_of\_week, bankdata$y)

chisq.test(bankdata$day\_of\_week, bankdata$y)

```

## 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

```{r }

## Summary on previous outcome variable

summary(bankdata$poutcome)

spineplot(x = bankdata$poutcome, y = bankdata$y, xlab = "Previous Outcome", ylab = "y",

main = "Previous outcome vs Y", col = c("lightblue", "coral"), xaxlabels = levels(bankdata$poutcome))

CrossTable(bankdata$poutcome, bankdata$y)

chisq.test(bankdata$poutcome, bankdata$y)

```

## 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

## EDA on continuous variables

```{r }

contnames = names(bankdata)[sapply(bankdata, class) == "integer" | sapply(bankdata, class) == "numeric"]

```

```{r}

#correlation plot for all continuous variables

corrgram(bankdata, 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)

```{r }

## Summary on age variable

summary(bankdata$age)

bankdata %>% ggplot(aes(x = age, fill = y, color=y)) + geom\_bar() + ggtitle("Distribution of Age") + xlab("Age") +

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

ggplot(bankdata, aes(x = y, y=age, fill=y)) + 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

```{r }

## Summary on duration variable

summary(bankdata$duration)

bankdata %>% ggplot(aes(x = duration, fill = y)) + geom\_bar() + ggtitle("Distribution of Duration") + xlab("Duration") +

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

ggplot(bankdata, aes(x = y, y=duration, fill=y)) + 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 "y"are pretty strongly associated. The longer duration is, the bigger prportion of people subscibe a term deposit.

```{r }

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

summary(bankdata$campaign)

bankdata %>% ggplot(aes(x = campaign, fill = y)) + geom\_bar() + ggtitle("Distribution of Campaign") + xlab("Campaign")+

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

ggplot(bankdata, aes(x = y, y=campaign, fill=y)) + geom\_boxplot() + ggtitle("Distribution of campaign") + xlab("Response") + ylab ("campaign")

aggregate(data.frame(count = bankdata$campaign), list(value = bankdata$campaign), length)

bankdata <- bankdata %>%

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.

```{r }

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

summary(bankdata$pdays)

bankdata %>% ggplot(aes(x = pdays, fill = y)) + geom\_bar() + ggtitle("Distribution of pdays") + xlab("pdays")

ggplot(bankdata, aes(x = y, y=pdays, fill=y)) + geom\_boxplot() + ggtitle("Distribution of pdays") + xlab("Response") + ylab ("pdays")

aggregate(data.frame(count = bankdata$pdays), list(value = bankdata$pdays), length)

```

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

## we can create two categories for this variable, first with 999 (customer never contacted) and second with other (customer contacted at least once). We can get difference among yes and no proportions between categories

```{r }

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

summary(bankdata$previous)

bankdata %>% ggplot(aes(x = previous, fill = y)) + geom\_bar() + ggtitle("Distribution of previous") + xlab("previous")

ggplot(bankdata, aes(x = y, y=previous, fill=y)) + geom\_boxplot() + ggtitle("Distribution of previous") + xlab("Response") + ylab ("previous")

aggregate(data.frame(count = bankdata$previous), list(value = bankdata$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.

## we can create three categories for this variable, first with 0 (customer never contacted), second with 1 (customer contacted one time) and third with more than one. We can get difference among yes and no proportions between categories

```{r }

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

summary(bankdata$emp.var.rate)

bankdata %>% ggplot(aes(x = emp.var.rate, fill = y)) + geom\_bar() + ggtitle("Distribution of emp.var.rate") + xlab("emp.var.rate")

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

```

```{r }

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

summary(bankdata$cons.price.idx)

bankdata %>% ggplot(aes(x = cons.price.idx, fill = y)) + geom\_bar() + ggtitle("Distribution of cons.price.idx") + xlab("cons.price.idx")

ggplot(bankdata, aes(x = y, y=cons.price.idx, fill=y)) + 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

```{r }

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

summary(bankdata$cons.conf.idx)

bankdata %>% ggplot(aes(x = cons.conf.idx, fill = y)) + geom\_bar() + ggtitle("Distribution of cons.conf.idx") + xlab("cons.conf.idx")

ggplot(bankdata, aes(x = y, y=cons.conf.idx, fill=y)) + 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

```{r }

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

summary(bankdata$euribor3m)

bankdata %>% ggplot(aes(x = euribor3m, fill = y)) + geom\_bar() + ggtitle("Distribution of euribor3m") + xlab("euribor3m")

ggplot(bankdata, aes(x = y, y=euribor3m, fill=y)) + geom\_boxplot() + ggtitle("Distribution of euribor3m") + xlab("Response") + ylab ("euribor3m")

```

## Minimum and maximum values are 0.634 and 5.045 respectively

```{r }

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

summary(bankdata$nr.employed)

bankdata %>% ggplot(aes(x = nr.employed, fill = y)) + geom\_bar() + ggtitle("Distribution of nr.employed") + xlab("nr.employed")

ggplot(bankdata, aes(x = y, y=nr.employed, fill=y)) + geom\_boxplot() + ggtitle("Distribution of nr.employed") + xlab("Response") + ylab ("nr.employed")

```

# Performing pairwise plots with color coding by which will help in identify any relationships

[1] "age" "duration" "campaign" "pdays" "previous" "emp.var.rate"

[7] "cons.price.idx" "cons.conf.idx" "euribor3m" "nr.employed"

```{r fig.height = 10, fig.width = 10}

bankdata %>% dplyr::select(age, duration, campaign, pdays, previous, emp.var.rate, cons.price.idx, cons.conf.idx, euribor3m, nr.employed, y) %>% ggpairs(aes(color = y)) %>% print(progress=F)

```

#### Creating a balanced dataset

### First of all let's create a new dataset with our class balanced with oversampling

## Splitting the data into training and test datasets (70-30 split)

```{r}

set.seed(1100)

bankdata\_new = bankdata %>% dplyr::select(-c('default', 'housing', 'loan'))

# dataset with "no"

bankdata\_no = bankdata\_new[which(bankdata\_new$y=="no"),]

# dataset with "yes"

bankdata\_yes = bankdata\_new[which(bankdata\_new$y=="yes"),]

splitPerc = .7 #Training / Test split Percentage

# Training / Test split of "no" dataset

noIndices = sample(1:dim(bankdata\_no)[1],round(splitPerc \* dim(bankdata\_no)[1]))

train\_no = bankdata\_no[noIndices,]

test\_no = bankdata\_no[-noIndices,]

# Training / Test split of "yes" dataset

yesIndices = sample(1:dim(bankdata\_yes)[1],round(splitPerc \* dim(bankdata\_yes)[1]))

train\_yes = bankdata\_yes[yesIndices,]

test\_yes = bankdata\_yes[-yesIndices,]

# Combine training "no" and "yes"

bank\_train = rbind(train\_no, train\_yes)

# Combine test "no" and "yes"

bank\_test = rbind(test\_no, test\_yes)

# upsampling training dataset

bank\_train\_upsample <- upSample(x = bank\_train[, -ncol(bank\_train)], y = bank\_train$y)

colnames(bank\_train\_upsample)[18] <- "y"

summary(bank\_train\_upsample$month)

# remove bservations with "unknown" in the "Loan" variable

#bank\_train\_upsample = bank\_train\_upsample[-(which(bank\_train\_upsample$loan=="unknown")), ]

model.main<-glm(y ~ . , data=bank\_train\_upsample,family = binomial(link="logit"))

(vif(model.main)[,3])^2

## Below variables has VIF more than 10

## nr.employed - 145.027646

## euribor3m - 144.159591

## emp.var.rate - 132.439592

## cons.price.idx - 55.437501

# remove observation with duration of

bank\_train\_upsample = bank\_train\_upsample[!bank\_train\_upsample$duration == 4918,]

model.main1<-glm(y ~ age+job+marital+education+contact+month+day\_of\_week+duration+campaign+pdays+previous+poutcome+ cons.conf.idx, data=bank\_train\_upsample,family = binomial(link="logit"))

(vif(model.main1)[,3])^2

summary(model.main1)

exp(cbind("Odds ratio" = coef(model.main1), confint.default(model.main1, level = 0.95)))

```

## parameter interpretation

# age : Holding all other explanatory variables fixed, odds of a client subscribing a term-deposit is 1.003 times higher than a client 1 year younger. The 95% confidence interval is [1.0001,1.006].

# job (admin vs blue-collar) : Holding all other explanatory variables fixed, The odds ratio of subscribing to a term-deposit for clients with admin job title relative to clients who are blue collar job tittle is 0.636. The 95% confidence interval is [0.575, 0.702].

# Marital (divorced vs single) : Holding all other explanatory variables fixed, The odds ratio of subscribing to a term-deposit for divorced clients relative to single clients is 1.323. The 95% confidence interval is [1.201, 1.458].

# Residual diagnostics - Plots here help us examine Cook's D graph:

```{r, echo = FALSE}

plot(model.main1)

bank\_train\_upsample[15056,]

```

# Lack of fit test - Hosmer Lemeshow test to check the lack of fit.

```{r, echo = FALSE}

hoslem.test(model.main1$y, fitted(model.main1), g=10) # it is ok

```

# ROC plot with balanced dataset

```{r }

# remove bservations with "unknown" in the "housing" variable

#bank\_test1 = bank\_test[-(which(bank\_test$housing=="unknown")), ]

model.main1<-glm(y ~ age+job+marital+education+month+day\_of\_week+duration+campaign+pdays+previous+poutcome+ cons.conf.idx, data=bank\_train\_upsample,family = binomial(link="logit"))

summary(model.main1)

bal.pred\_probs <- predict(model.main1, bank\_train\_upsample, type="response")

bal.pred\_yns <- factor(ifelse(bal.pred\_probs>0.5, "yes", "no"))

bal.cm <- confusionMatrix(table(bal.pred\_yns, bank\_train\_upsample$y))

bal.cm

bal.pred\_probs.test <- predict(model.main1, bank\_test, type="response")

bal.pred\_yns.test <- factor(ifelse(bal.pred\_probs.test>0.5, "yes", "no"))

bal.cm.test <- confusionMatrix(table(bal.pred\_yns.test, bank\_test$y))

bal.cm.test

df <- rbind(data.frame(predictor = predict(model.main1, bank\_train\_upsample),

known.truth = bank\_train\_upsample$y,

model = "Train"),

data.frame(predictor = predict(model.main1, bank\_test),

known.truth = bank\_test$y,

model = "Test"))

summary(bank\_test$y)

# the aesthetic names are not the most intuitive

# `d` holds the known truth

# `m` holds the predictor values

ggroc <- ggplot(df, aes(d = known.truth, m = predictor, color = model)) +

geom\_roc(n.cuts = 0) + geom\_abline(intercept = 0, slope = 1, linetype='dashed')

calc\_auc(ggroc)

ggroc

```

# ROC plot with unbalanced dataset

```{r }

# Unbalanced training dataset

Unbal\_bank\_train = bank\_train

# remove bservations with "unknown" in the "housing" variable

#Unbal\_bank\_test = bank\_test[-(which(bank\_test$housing=="unknown")), ]

unbal.model.main<-glm(y ~ age+job+marital+education+contact+month+day\_of\_week+duration+campaign+pdays+previous+poutcome+ cons.conf.idx, data=Unbal\_bank\_train,family = binomial(link="logit"))

summary(unbal.model.main)

# Removed age, day\_of\_Week, marital and Education

unbal.model.main1<-glm(y ~ job+contact+month+duration+campaign+pdays+previous+poutcome+ cons.conf.idx, data=Unbal\_bank\_train,family = binomial(link="logit"))

summary(unbal.model.main1)

unbal.pred\_probs <- predict(unbal.model.main1, Unbal\_bank\_train, type="response")

unbal.pred\_yns <- factor(ifelse(unbal.pred\_probs>0.5, "yes", "no"))

unbal.cm <- confusionMatrix(table(unbal.pred\_yns, Unbal\_bank\_train$y))

unbal.cm

unbal.pred\_probs.test <- predict(unbal.model.main1, Unbal\_bank\_test, type="response")

unbal.pred\_yns.test <- factor(ifelse(unbal.pred\_probs.test>0.5, "yes", "no"))

unbal.cm.test <- confusionMatrix(table(unbal.pred\_yns.test, Unbal\_bank\_test$y))

unbal.cm.test

unbal.df <- rbind(data.frame(predictor1 = predict(unbal.model.main1, Unbal\_bank\_train),

known.truth = Unbal\_bank\_train$y,

model = "Train"),

data.frame(predictor1 = predict(unbal.model.main1, Unbal\_bank\_test),

known.truth = Unbal\_bank\_test$y,

model = "Test"))

# the aesthetic names are not the most intuitive

# `d` holds the known truth

# `m` holds the predictor values

unbal.ggroc <- ggplot(unbal.df, aes(d = known.truth, m = predictor1, color = model)) +

geom\_roc(n.cuts = 0) + geom\_abline(intercept = 0, slope = 1, linetype='dashed')

calc\_auc(unbal.ggroc)

unbal.ggroc

```

######## Logistic Regression with interaction ######################

```{r datawrangle}

#downsample

down\_size <- downSample(x = bankdata[, -ncol(bankdata)], y = bankdata$y)

colnames(down\_size)[21] <- "y"

table(down\_size$y)

combos <- combn(c("duration","nr.employed" , "month", "poutcome", "emp.var.rate",

"cons.price.idx", "job", "contact", "euribor3m", "default", "day\_of\_week",

"pdays", "campaign", "cons.conf.idx"), 2, FUN = NULL, simplify = TRUE)

combos[,1]

n = length(combos[1,])

var\_1 = c()

var\_2 = c()

interaction\_ps = c()

for(i in 1:n){

model.main<-glm(y ~ eval(parse(text = combos[1,i])) \* eval(parse(text = combos[2,i])), data=down\_size,family = binomial(link="logit"))

s <- summary(model.main)

var\_1 = c(var\_1, combos[1,i])

var\_2 = c(var\_2, combos[2,i])

interaction\_ps = c(interaction\_ps, s$coefficients[4,4])

}

interaction\_df = data.frame(var\_1, var\_2, interaction\_ps)

head(interaction\_df)

interactions <- interaction\_df %>%

filter(interaction\_ps < .0001) %>%

mutate(var\_1 = as.character(var\_1)) %>%

mutate(var\_2 = as.character(var\_2))

```

```{r}

print(interactions)

```

Interaction Plots

```{r interactionplots}

for(i in 1:dim(interactions)[1]) {

#barplot of two variables blocked on y/n

print(bankdata %>%

ggplot(aes(y=eval(parse(text = interactions$var\_1[i])), x=eval(parse(text = interactions$var\_2[i])), color = y)) +

geom\_boxplot()+

xlab(interactions$var\_2[i]) +

ylab(interactions$var\_1[i]) +

ggtitle("Distribution of ")

)

#summary of p value of interaction term

model<-glm(y ~ eval(parse(text=interactions$var\_1[i])) \* eval(parse(text=interactions$var\_2[i])), data=down\_size,family = binomial(link="logit"))

print(summary(model))

}

```

Make model with interaction terms

```{r}

model.interaction1 <- glm(y ~ duration \* nr.employed + month + poutcome + emp.var.rate +

cons.price.idx + job + contact + euribor3m + default + day\_of\_week +

pdays + campaign + cons.conf.idx +

duration\*nr.employed +

duration\*poutcome +

duration \* emp.var.rate +

duration \* cons.price.idx +

duration \* job +

duration \* euribor3m +

duration \* cons.conf.idx +

nr.employed \* emp.var.rate +

nr.employed \* euribor3m +

nr.employed \* campaign +

nr.employed \* cons.conf.idx +

month \* cons.price.idx +

month \* job +

month \* contact +

month \* default +

month \* campaign +

poutcome \* emp.var.rate +

poutcome \* job +

poutcome \* euribor3m +

poutcome \* pdays +

poutcome \* cons.conf.idx +

emp.var.rate \* euribor3m +

emp.var.rate \* campaign +

emp.var.rate \* cons.conf.idx +

cons.price.idx \* contact +

cons.price.idx \* pdays +

cons.price.idx \* cons.conf.idx +

euribor3m \* campaign +

euribor3m \* cons.conf.idx +

default \* pdays +

default \* campaign +

default \* cons.conf.idx, data=bankdata, family="binomial")

```

Make functions for cross validation

```{r cv, echo=FALSE}

#--- CROSS VALIDATION SETUP ---

#creates a train test split with upsampling train data for balance

split\_train\_test <- function(bankdata) {

# dataset with "no"

bankdata\_no = bankdata[which(bankdata$y=="no"),]

# dataset with "yes"

bankdata\_yes = bankdata[which(bankdata$y=="yes"),]

splitPerc = .7 #Training / Test split Percentage

# Training / Test split of "no" dataset

noIndices = sample(1:dim(bankdata\_no)[1],round(splitPerc \* dim(bankdata\_no)[1]))

train\_no = bankdata\_no[noIndices,]

test\_no = bankdata\_no[-noIndices,]

# Training / Test split of "yes" dataset

yesIndices = sample(1:dim(bankdata\_yes)[1],round(splitPerc \* dim(bankdata\_yes)[1]))

train\_yes = bankdata\_yes[yesIndices,]

test\_yes = bankdata\_yes[-yesIndices,]

# Combine training "no" and "yes"

bank\_train = rbind(train\_no, train\_yes)

# Combine test "no" and "yes"

bank\_test = rbind(test\_no, test\_yes)

# upsampling training dataset

bank\_train\_upsample <- upSample(x = bank\_train[, -ncol(bank\_train)], y = bank\_train$y)

colnames(bank\_train\_upsample)[21] <- "y"

summary(bank\_train\_upsample$y)

return(list(bank\_train\_upsample, bank\_test))

}

#gives the average accuracy, sensitivity, and specificity over 5 train-test splits

#for logistic regression model

get\_test\_metrics <- function(data, formula, thresh){

accuracies <- c()

sensitivities <- c()

specificities <- c()

for(i in 1:2){

split\_data <- split\_train\_test(data)

train <- split\_data[[1]]

test <- split\_data[[2]]

model <- glm(formula, data=train,family = binomial(link="logit"))

pred\_probs <- predict(model, test, type="response")

pred\_yns <- factor(ifelse(pred\_probs>thresh, "yes", "no"))

cm <- confusionMatrix(table(pred\_yns, test$y))

accuracies <- c(accuracies, cm$overall[1])

sensitivities <- c(sensitivities, cm$byClass[1])

specificities <- c(specificities, cm$byClass[2])

}

acc = mean(accuracies)

sens = mean(sensitivities)

specs = mean(specificities)

result = list(acc, sens, specs)

names(result) = c("Accuracy", "Sensitivity", "Specificity")

return(result)

}

```

see thresholds for the model

```{r}

#finding the best threshold for yes/no

thresholds <- seq(0.02, .6, by=.02)

accs <- c()

sens <- c()

specs <- c()

for(i in thresholds){

print("i", i)

print(i)

m <- get\_test\_metrics(bankdata, y~duration + nr.employed + month + poutcome + emp.var.rate +

cons.price.idx + job + contact + euribor3m + default + day\_of\_week +

pdays + campaign + cons.conf.idx, i)

accs <- c(accs, m$Accuracy)

sens <- c(sens, m$Sensitivity)

specs <- c(specs, m$Specificity)

}

accs

sens

specs

metrics <- data.frame(thresholds,accs,sens,specs)

metrics\_long <- metrics %>%

gather(key=metric, percent, accs:specs)

#plot the accuracy, sensitivity, and specificity over the thresholds; between .1-.2 seems decent to me

metrics\_long %>%

ggplot(aes(x=thresholds, y=percent, color = metric)) +

geom\_point() +

ggtitle("Accuracy, Sensitivity, and Specificity at Thresholds")

```

a simple model

```{r}

model.main<-glm(y ~ duration \* nr.employed + month + poutcome + emp.var.rate +

cons.price.idx + job + contact + euribor3m + default + day\_of\_week +

pdays + campaign + cons.conf.idx, data=bankdata,family = binomial(link="logit"))

```

Compare ROC between complex model vs simple model

```{r}

df <- rbind(data.frame(predictor = predict(model.interaction1, bankdata),

known.truth = bankdata$y,

model = "Complex Model"),

data.frame(predictor = predict(model.main, bankdata),

known.truth = bankdata$y,

model = "Simple Model"))

ggroc <- ggplot(df, aes(d = known.truth, m = predictor, color = model)) +

geom\_roc(n.cuts = 0) + geom\_abline(intercept = 0, slope = 1, linetype='dashed')

ggroc

calc\_auc(ggroc)

```

######## LDA/QDA/KNN/RF Models ######################

#### Unbalance Data: LDA with all exist continuous variables can get an accuracy of 90.7%.

````{r LDA1, echo=TRUE}

### LDA with all continuous variables

library(MASS)

bank\_con <- bank %>% dplyr::select(age, duration,campaign,pdays,previous,emp.var.rate,cons.price.idx,cons.conf.idx,euribor3m,nr.employed)

mylda1 <- lda(bank$y ~ . , data = bank\_con)

x1=table(predict(mylda1, type="class")$class, bank$y)

x1\_con <- confusionMatrix(x1)

x1\_con

x1\_con$overall[1]

x1\_con$byClass[1]

x1\_con$byClass[2]

```

#### Unbalance Data: QDA with all continuous variables give an accuracy of 87.7%.

```{r QDA4, echo=TRUE}

myqda4 <- qda(bank$y ~ . , data = bank\_con)

x1=table(predict(myqda4, type="class")$class, bank$y)

x1\_con <- confusionMatrix(x1)

x1\_con

x1\_con$overall[1]

x1\_con$byClass[1]

x1\_con$byClass[2]

```

#### Creating a balanced dataset

### First of all let's create a new dataset with our class balanced with oversampling and also with undersampling

## Splitting the data into training and test datasets (80-20 split)

```{r}

bank$y <- as.factor(bank$y)

bankdata <- bank

# dataset with "no"

bankdata\_no = bankdata[which(bankdata$y=="no"),]

# dataset with "yes"

bankdata\_yes = bankdata[which(bankdata$y=="yes"),]

splitPerc = .7 #Training / Test split Percentage

# Training / Test split of "no" dataset

noIndices = sample(1:dim(bankdata\_no)[1],round(splitPerc \* dim(bankdata\_no)[1]))

train\_no = bankdata\_no[noIndices,]

test\_no = bankdata\_no[-noIndices,]

# Training / Test split of "yes" dataset

yesIndices = sample(1:dim(bankdata\_yes)[1],round(splitPerc \* dim(bankdata\_yes)[1]))

train\_yes = bankdata\_yes[yesIndices,]

test\_yes = bankdata\_yes[-yesIndices,]

# Combine training "no" and "yes"

bank\_train = rbind(train\_no, train\_yes)

# Combine test "no" and "yes"

bank\_test = rbind(test\_no, test\_yes)

# remove bservations with "unknown" in the "Loan" variable

bank\_train = bank\_train[-(which(bank\_train$loan=="unknown")), ]

bank\_train = bank\_train[-(which(bank\_train$default=="yes")), ]

# upsampling training dataset

bank\_train\_upsample <- upSample(x = bank\_train[, -ncol(bank\_train)], y = as.factor(bank\_train$y))

colnames(bank\_train\_upsample)[21] <- "y"

summary(bank\_train\_upsample$y)

# remove bservations with "unknown" in the "Loan" variable

#bank\_train\_upsample = bank\_train\_upsample[-(which(bank\_train\_upsample$loan=="unknown")), ]

# remove bservations with "unknown" in the "Loan" variable

bank\_test = bank\_test[-(which(bank\_test$loan=="unknown")), ]

bank\_test = bank\_test[-(which(bank\_test$default=="yes")), ]

model.main<-glm(y ~ . , data=bank\_train\_upsample,family = binomial(link="logit"))

(vif(model.main)[,3])^2

## Below variables has VIF more than 10

## nr.employed - 145.027646

## euribor3m - 144.159591

## emp.var.rate - 132.439592

## cons.price.idx - 55.437501

model.main1<-glm(y ~ age+job+marital+education+default+housing+loan+contact+month+day\_of\_week+duration+campaign+pdays+previous+poutcome+ cons.conf.idx, data=bank\_train\_upsample,family = binomial(link="logit"))

(vif(model.main1)[,3])^2

summary(model.main1)

exp(cbind("Odds ratio" = coef(model.main1), confint.default(model.main1, level = 0.95)))

```

### Observations:

#### Balanced Dataset: LDA with all variables for upsampled dataset can get an accuracy of 85.6%.

```{r LDA\_new, echo=TRUE}

### LDA with all continous variables

bank\_upsample\_con <- bank\_train\_upsample %>% dplyr::select(age, duration,campaign,previous,emp.var.rate,cons.price.idx,cons.conf.idx,euribor3m,nr.employed,y)

bank\_test\_con <- bank\_test %>% dplyr::select(age, duration,campaign,previous,emp.var.rate,cons.price.idx,cons.conf.idx,euribor3m,nr.employed,y)

mylda\_new <- lda(bank\_upsample\_con$y ~ . , data = bank\_upsample\_con)

x1=table(predict(mylda\_new, type="class")$class, bank\_upsample\_con$y)

x1\_con <- confusionMatrix(x1)

x1\_con

x1\_con$overall[1]

x1\_con$byClass[1]

x1\_con$byClass[2]

pred\_lda <- predict(mylda\_new, bank\_upsample\_con, type = "response")$posterior

pred\_lda <- as.data.frame(pred\_lda)

pred\_lda\_train <- prediction(pred\_lda[,2],bank\_upsample\_con$y)

roc.perf2 = performance(pred\_lda\_train, measure = "tpr", x.measure = "fpr")

auc.train <- performance(pred\_lda\_train, measure = "auc")

auc.train <- auc.train@y.values

# Test group confusion metric statistics

x2=table(predict(mylda\_new, bank\_test\_con, type="class")$class, bank\_test$y)

x2\_con <- confusionMatrix(x2)

x2\_con

x2\_con$overall[1]

x2\_con$byClass[1]

x2\_con$byClass[2]

pred\_ldanew <- predict(mylda\_new, bank\_test\_con, type = "response")$posterior

pred\_ldanew <- as.data.frame(pred\_ldanew)

pred\_ldanew\_test <- prediction(pred\_ldanew[,2],bank\_test\_con$y)

roc.perf3 = performance(pred\_ldanew\_test, measure = "tpr", x.measure = "fpr")

auc.test <- performance(pred\_ldanew\_test, measure = "auc")

auc.test <- auc.test@y.values

plot(roc.perf2)

plot(roc.perf3,col="orange", add = TRUE)

legend("bottomright",legend=c("Train","Test"),col=c("black","orange"),lty=1,lwd=1)

abline(a=0, b= 1)

#text(x = .40, y = .6,paste("AUC = ", round(auc.train[[1]],3), sep = ""))

text(x = .40, y = .6,paste("AUC = ", round(auc.test[[1]],3), sep = ""))

```

```{r LDA\_new log transform, echo=TRUE}

### LDA with all variables

bank\_train\_upsample\_clean = bank\_upsample\_con[!is.infinite(log(bank\_upsample\_con$duration)),]

mylda\_new2\_log <- lda(bank\_train\_upsample\_clean$y ~ log(age) + log(duration) + log(campaign) + previous + emp.var.rate + cons.price.idx+ cons.conf.idx+ euribor3m+ nr.employed, data = bank\_train\_upsample\_clean)

x1=table(predict(mylda\_new2\_log, type="class")$class, bank\_train\_upsample\_clean$y)

x1\_con <- confusionMatrix(x1)

x1\_con

x1\_con$overall[1]

x1\_con$byClass[1]

x1\_con$byClass[2]

pred\_lda\_log <- predict(mylda\_new2\_log, bank\_train\_upsample\_clean, type = "response")$posterior

pred\_lda\_log <- as.data.frame(pred\_lda\_log)

pred\_lda\_log\_train <- prediction(pred\_lda\_log[,2],bank\_train\_upsample\_clean$y)

roc.perflog = performance(pred\_lda\_log\_train, measure = "tpr", x.measure = "fpr")

auc.train <- performance(pred\_lda\_log\_train, measure = "auc")

auc.train <- auc.train@y.values

# Test group confusion metric statistics

x2=table(predict(mylda\_new2\_log, bank\_test, type="class")$class, bank\_test$y)

x2\_con <- confusionMatrix(x2)

x2\_con

x2\_con$overall[1]

x2\_con$byClass[1]

x2\_con$byClass[2]

pred\_lda\_log2 <- predict(mylda\_new2\_log, bank\_test, type = "response")$posterior

pred\_lda\_log2 <- as.data.frame(pred\_lda\_log2)

pred\_lda\_log\_test <- prediction(pred\_lda\_log2[,2],bank\_test$y)

roc.perflog\_test = performance(pred\_lda\_log\_test, measure = "tpr", x.measure = "fpr")

#auc.train <- performance(roc.perflog\_test, measure = "auc")

#auc.train <- auc.train@y.values

plot(roc.perflog)

plot(roc.perflog\_test,col="orange", add = TRUE)

legend("bottomright",legend=c("Train","Test"),col=c("black","orange"),lty=1,lwd=1)

abline(a=0, b= 1)

#text(x = .40, y = .6,paste("AUC = ", round(auc.train[[1]],3), sep = ""))

text(x = .40, y = .6,paste("AUC = ", round(auc.test[[1]],3), sep = ""))

plot(roc.perf\_ldanew2\_test)

plot(roc.perflog\_test,col="orange", add = TRUE)

legend("bottomright",legend=c("LDA Before Log","LDA After Log"),col=c("black","orange"),lty=1,lwd=1)

abline(a=0, b= 1)

```

### QDA with selected continuous variables from upsampled dataset give an accuracy of 82.2%.

```{r QDA\_new, echo=TRUE}

myqda\_new <- qda(bank\_upsample\_con$y ~ . , data = bank\_upsample\_con)

x1=table(predict(myqda\_new, type="class")$class, bank\_upsample\_con$y)

x1\_con <- confusionMatrix(x1)

x1\_con

x1\_con$overall[1]

x1\_con$byClass[1]

x1\_con$byClass[2]

# Test group confusion metric statistics

x2=table(predict(myqda\_new, bank\_test, type="class")$class, bank\_test$y)

x2\_con <- confusionMatrix(x2)

x2\_con

x2\_con$overall[1]

x2\_con$byClass[1]

x2\_con$byClass[2]

pred\_qdanew <- predict(myqda\_new, bank\_test, type = "response")$posterior

pred\_qdanew <- as.data.frame(pred\_qdanew)

pred\_qdanew\_test <- prediction(pred\_qdanew[,2],bank\_test$y)

roc.perf\_qdanew\_test = performance(pred\_qdanew\_test, measure = "tpr", x.measure = "fpr")

auc.train <- performance(pred\_qdanew\_test, measure = "auc")

auc.train <- auc.train@y.values

plot(roc.perf\_qdanew\_test)

abline(a=0, b= 1)

text(x = .40, y = .6,paste("AUC = ", round(auc.train[[1]],3), sep = ""))

```

### QDA with selected continuous variables from upsampled dataset give an accuracy of 86.7%.

### QDA not perform better than LDA.

```{r}

myqda\_new\_log <- qda(bank\_train\_upsample\_clean$y ~ log(age) + log(duration) + log(campaign) + previous + emp.var.rate + cons.price.idx+ cons.conf.idx+ euribor3m+ nr.employed, data = bank\_train\_upsample\_clean)

# Test group confusion metric statistics

x2=table(predict(myqda\_new\_log, bank\_test, type="class")$class, bank\_test$y)

x2\_con <- confusionMatrix(x2)

x2\_con

x2\_con$overall[1]

x2\_con$byClass[1]

x2\_con$byClass[2]

pred\_qdanew\_log <- predict(myqda\_new\_log, bank\_test, type = "response")$posterior

pred\_qdanew\_log <- as.data.frame(pred\_qdanew\_log)

pred\_qdanew\_log\_test <- prediction(pred\_qdanew\_log[,2],bank\_test$y)

roc.perf\_qdanew\_log\_test = performance(pred\_qdanew\_log\_test, measure = "tpr", x.measure = "fpr")

auc.train <- performance(pred\_qdanew\_log\_test, measure = "auc")

auc.train <- auc.train@y.values

plot(roc.perf\_qdanew\_test)

plot(roc.perf\_qdanew\_log\_test,col="orange", add = TRUE)

plot(roc.perf\_ldanew2\_test,col="blue", add = TRUE)

legend("bottomright",legend=c("QDA Before Log","QDA After Log","LDA Before Log"),col=c("black","orange","blue"),lty=1,lwd=1)

abline(a=0, b= 1)

```

#### KNN model

```{r}

library(class)

bank\_train\_upsample\_cons <- bank\_train\_upsample %>% dplyr::select(age, duration,campaign,previous,emp.var.rate,cons.price.idx,cons.conf.idx,euribor3m,nr.employed)

bank\_test\_cons <- bank\_test %>% dplyr::select(age, duration,campaign,previous,emp.var.rate,cons.price.idx,cons.conf.idx,euribor3m,nr.employed)

model\_knn = class::knn(bank\_train\_upsample\_cons,bank\_test\_cons,bank\_train\_upsample$y, prob = TRUE, k = 3)

table(model\_knn,bank\_test$y)

CM\_knn = confusionMatrix(table(model\_knn,bank\_test$y))

CM\_knn

```

#### The chosen best KNN model is with k=5

```{r}

library(caret)

iterations = 10

numks = 20

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

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

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

for(j in 1:iterations)

{

accs = data.frame(accuracy = numeric(10), k = numeric(10))

for(i in 1:numks)

{

classifications = knn(bank\_train\_upsample\_con,bank\_test\_con,bank\_train\_upsample$y, prob = TRUE, k = i)

table(classifications,bank\_test$y)

CM = confusionMatrix(table(classifications,bank\_test$y))

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

masterSen[j,i] = CM$byClass[1]

masterSpe[j,i] = CM$byClass[2]

}

}

MeanAcc = colMeans(masterAcc)

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

MeanSen = colMeans(masterSen)

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

MeanSpe = colMeans(masterSpe)

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

```

```{r}

model\_knn5 = class::knn(bank\_train\_upsample\_cons,bank\_test\_cons,bank\_train\_upsample$y, prob = FALSE, k = 5)

table(model\_knn5,bank\_test$y)

CM\_knn5 = confusionMatrix(table(model\_knn5,bank\_test$y))

CM\_knn5

```

#### Random Forest model

#### With Unbalanced Data (All continous)

```{r}

library(randomForest)

bank\_train\_con <- bank\_train %>% dplyr::select(age, duration,campaign,previous,emp.var.rate,cons.price.idx,cons.conf.idx,euribor3m,nr.employed)

## Classification Method: RandomForest

model\_rf = randomForest(bank\_train\_con,as.factor(bank\_train$y),ntree=500)

CM\_rf = confusionMatrix(table(predict(model\_rf,bank\_test\_con),bank\_test$y))

CM\_rf

## But random forest can provide with feature importance

varImpPlot(model\_rf)

```

#### With balanced Data (All continous)

```{r}

## Classification Method: RandomForest

model\_rf2 = randomForest(bank\_train\_upsample\_con,as.factor(bank\_train\_upsample$y),ntree=500)

CM\_rf2 = confusionMatrix(table(predict(model\_rf2,bank\_test\_con),bank\_test$y))

CM\_rf2

## But random forest can provide with feature importance

varImpPlot(model\_rf2)

fit.pred<-predict(model\_rf2,newdata=bank\_test\_con,type="prob")

pred <- prediction(fit.pred[,2], bank\_test$y)

roc.perf\_rf2 = performance(pred, measure = "tpr", x.measure = "fpr")

auc.train <- performance(pred, measure = "auc")

auc.train <- auc.train@y.values

plot(roc.perf\_rf2)

abline(a=0, b= 1)

text(x = .40, y = .6,paste("AUC = ", round(auc.train[[1]],3), sep = ""))

```

#### With balanced Data (All variables)

```{r}

bank\_train\_upsample\_rf <- bank\_train\_upsample

# Make dependent variable as a factor (categorical)

bank\_train\_upsample\_rf$job = as.factor(bank\_train\_upsample\_rf$job)

bank\_train\_upsample\_rf$marital = as.factor(bank\_train\_upsample\_rf$marital)

bank\_train\_upsample\_rf$education = as.factor(bank\_train\_upsample\_rf$education)

bank\_train\_upsample\_rf$default = as.factor(bank\_train\_upsample\_rf$default)

bank\_train\_upsample\_rf$housing = as.factor(bank\_train\_upsample\_rf$housing)

bank\_train\_upsample\_rf$loan = as.factor(bank\_train\_upsample\_rf$loan)

bank\_train\_upsample\_rf$contact = as.factor(bank\_train\_upsample\_rf$contact)

bank\_train\_upsample\_rf$month = as.factor(bank\_train\_upsample\_rf$month)

bank\_train\_upsample\_rf$day\_of\_week = as.factor(bank\_train\_upsample\_rf$day\_of\_week)

bank\_train\_upsample\_rf$poutcome = as.factor(bank\_train\_upsample\_rf$poutcome)

bank\_test\_rf <- bank\_test

# Make dependent variable as a factor (categorical)

bank\_test\_rf$job = as.factor(bank\_test$job)

bank\_test\_rf$marital = as.factor(bank\_test$marital)

bank\_test\_rf$education = as.factor(bank\_test$education)

bank\_test\_rf$default = as.factor(bank\_test$default)

bank\_test\_rf$housing = as.factor(bank\_test$housing)

bank\_test\_rf$loan = as.factor(bank\_test$loan)

bank\_test\_rf$contact = as.factor(bank\_test$contact)

bank\_test\_rf$month = as.factor(bank\_test$month)

bank\_test\_rf$day\_of\_week = as.factor(bank\_test$day\_of\_week)

bank\_test\_rf$poutcome = as.factor(bank\_test$poutcome)

## Classification Method: RandomForest

model\_rf\_all = randomForest(bank\_train\_upsample\_rf[,1:20],as.factor(bank\_train\_upsample$y),ntree=500)

CM\_rf\_all = confusionMatrix(table(predict(model\_rf\_all,bank\_test\_rf[,1:20]),bank\_test$y))

CM\_rf\_all

## But random forest can provide with feature importance

varImpPlot(model\_rf\_all)

fit.pred<-predict(model\_rf\_all,newdata=bank\_test\_rf[,1:20],type="prob")

pred <- prediction(fit.pred[,2], bank\_test$y)

roc.perf\_rf3 = performance(pred, measure = "tpr", x.measure = "fpr")

auc.train <- performance(pred, measure = "auc")

auc.train <- auc.train@y.values

plot(roc.perf\_rf3)

abline(a=0, b= 1)

text(x = .40, y = .6,paste("AUC = ", round(auc.train[[1]],3), sep = ""))

```

#### With balanced Data (Variables chosen)

```{r}

bank\_upsample\_select <- bank\_train\_upsample\_rf %>% dplyr::select(c('age', 'duration', 'campaign', 'pdays', 'previous', 'cons.price.idx', 'cons.conf.idx','euribor3m','y'))

bank\_test\_select <- bank\_test\_rf %>% dplyr::select(c('age', 'duration', 'campaign', 'pdays', 'previous', 'cons.price.idx', 'cons.conf.idx','euribor3m','y'))

## Classification Method: RandomForest

model\_rf\_sel = randomForest(bank\_upsample\_select[,1:8],as.factor(bank\_upsample\_select$y),ntree=500)

CM\_rf\_sel = confusionMatrix(table(predict(model\_rf\_sel,bank\_test\_select[,1:8]),bank\_test\_select$y))

CM\_rf\_sel

## But random forest can provide with feature importance

varImpPlot(model\_rf\_sel)

fit.pred<-predict(model\_rf\_sel,newdata=bank\_test\_select[,1:8],type="prob")

pred <- prediction(fit.pred[,2], bank\_test\_select$y)

roc.perf\_rf4 = performance(pred, measure = "tpr", x.measure = "fpr")

auc.train <- performance(pred, measure = "auc")

auc.train <- auc.train@y.values

plot(roc.perf\_rf4)

abline(a=0, b= 1)

text(x = .40, y = .6,paste("AUC = ", round(auc.train[[1]],3), sep = ""))

```

#### ROCs comparisons across different random forest results

```{r}

plot(roc.perf\_rf2)

plot(roc.perf\_rf3,col="orange", add = TRUE)

plot(roc.perf\_rf4,col="blue", add = TRUE)

legend("bottomright",legend=c("RF1","RF2","RF3"),col=c("black","orange","blue"),lty=1,lwd=1)

abline(a=0, b= 1)

```

###Logistic regression

#### Objective 1 Model

```{r}

model.main1<-glm(y ~ age+job+marital+education+contact+month+housing+loan+day\_of\_week+duration+campaign+pdays+previous+poutcome+ cons.conf.idx, data=bank\_train\_upsample,family = binomial(link="logit"))

bal.pred\_probs <- predict(model.main1, bank\_test, type="response")

bal.pred\_yns <- factor(ifelse(bal.pred\_probs>0.5, "yes", "no"))

bal.cm <- confusionMatrix(table(bal.pred\_yns, bank\_test$y))

bal.cm

#fit.pred<-predict(model.main1,newdata=bank\_test,type="prob")

pred <- prediction(bal.pred\_probs, bank\_test$y)

roc.perf\_lr = performance(pred, measure = "tpr", x.measure = "fpr")

auc.train <- performance(pred, measure = "auc")

auc.train <- auc.train@y.values

plot(roc.perf\_lr)

abline(a=0, b= 1)

text(x = .40, y = .6,paste("AUC = ", round(auc.train[[1]],3), sep = ""))

```

#### Tina's model with interaction terms

```{r}

model.test<-glm(y ~ duration \* nr.employed + month + poutcome + emp.var.rate +

cons.price.idx + job + contact + euribor3m + default + day\_of\_week +

pdays + campaign + cons.conf.idx +

duration\*nr.employed +

duration\*poutcome +

duration \* emp.var.rate +

duration \* cons.price.idx +

duration \* job +

duration \* euribor3m +

duration \* cons.conf.idx +

nr.employed \* emp.var.rate +

nr.employed \* euribor3m +

nr.employed \* campaign +

nr.employed \* cons.conf.idx +

month \* cons.price.idx +

month \* job +

month \* contact +

month \* default +

month \* campaign +

poutcome \* emp.var.rate +

poutcome \* job +

poutcome \* euribor3m +

poutcome \* pdays +

poutcome \* cons.conf.idx +

emp.var.rate \* euribor3m +

emp.var.rate \* campaign +

emp.var.rate \* cons.conf.idx +

cons.price.idx \* contact +

cons.price.idx \* pdays +

cons.price.idx \* cons.conf.idx +

euribor3m \* campaign +

euribor3m \* cons.conf.idx +

default \* pdays +

default \* campaign +

default \* cons.conf.idx, data=bank\_train\_upsample, family="binomial")

bal.pred\_probs <- predict(model.test, bank\_test, type="response")

pred <- prediction(bal.pred\_probs, bank\_test$y)

roc.perf\_lr2 = performance(pred, measure = "tpr", x.measure = "fpr")

```

#### ROCs comparisons across different methods

```{r}

plot(roc.perf1)

plot(roc.perf\_rf3,col="orange", add = TRUE)

plot(roc.perf\_lr,col="blue", add = TRUE)

plot(roc.perf\_lr2,col="red", add = TRUE)

legend("bottomright",legend=c("LDA","Random Forest","Logistic Regress","lr2"),col=c("black","orange","blue","red"),lty=1,lwd=1)

abline(a=0, b= 1)

```