Portuguese Bank Marketing Data

Team ID: Group 25

- Truc Le (trlle@ucdavis.edu (mailto:trlle@ucdavis.edu))- (Data Wrangling, Data Exploration, visualization, Leading the Team, Coding, Primary Question, Stitching report together)
- Mengna Lin (menlin@ucdavis.edu (mailto:menlin@ucdavis.edu))- (Introduction- background, Literature review, Reference citation, Leading the team,Editing final report)
- Lik Xian Lim (lxlim@ucdavis.edu (mailto:lxlim@ucdavis.edu))- (Analysis Plan, Literature Review, Extra Credit Method, Inferential Analysis of Extra Credit method, Leading the Team, Editing the report)
- Mary Vang (marvang@ucdavis.edu (mailto:marvang@ucdavis.edu))- (Exploratory Analysis Interpretation, Main Analysis and Methods Interpretation, Results - Inferential Analysis, Secondary Question, Leading the Team (Meeting Agenda))

---

1 Introduction

1.1 Background

In this project, we analyze empirical data from a 2008 telemarketing campaign from a Portuguese retail bank aimed at subscribing new users to a term deposit. The authors of the original data focus on evaluating different methods that can be used to determine the extent to which bank telemarketing is successful at predicting the likelihood of a client subscribing to a long-term deposit. The dataset is available for download from the UCI Machine Learning Repository at the link (https://archive.ics.uci.edu/ml/datasets/Bank+Marketing#) along with more information in the original paper by Moro, Cortez, and Rita (2014).

For our specific project, we will be analyzing the “bank-additional-full.csv” that includes a variety of predictor variables to address our primary question of interest: build an effective prediction model for whether a client will sign onto a long-term deposit. Our secondary question of interest focuses on identifying the “type” of client data (bank telemarketing, campaign, socio-economic, and individual background) most pivotal in determining whether a client will sign onto a long-term deposit. We focus on three primary statistical tools for analyzing and better visualizing potential relationships between our variables: decision trees, logistic regression, and k-nearest neighbor. According to Decision Trees for Analytics: Using SAS Enterprise Miner, decision trees have been useful in substituting and supporting various forms of multiple variable analysis including traditional forms of multivariate analysis (ie. multiple linear regression), data mining tools and techniques (ie. neutral networks), and multidimensional forms of reporting and analysis. By being able to combine categories that have similar values into ranges that align with some target value, we see improvement in prediction and classification results as less information is being lost when collapsing a set of categorical variables together [1]. Considering the varying units and level of measurements in our dataset, a decision tree is desirable for its relative power, flexibility with a variety of data, and ease of interpretability.

As for logistic regression, it has been one of the most widely used statistical tools for analyzing binary and proportional response data. Basic logistic regression models can be used to examine relationships between binary response variables and independent response variables that come in many forms (ie. binary, continuous, categorical) [2]. In logistic regression, we assess the goodness of fit and predictive utility of the model, and turn to regression coefficients and inferential statistics to determine the significance of individual predictors [4]. Due to its flexibility with a variety of data, ease of implementation in statistical software algorithms, and consistency with assumptions in real-world empirical data, logistic regression models have been used to understand complex relationships from a variety of disciplines emphasizing prediction and causality.

Regarding k-Nearest Neighbor (k-NN), it has been widely used as a non-parametric statistical technique for applied statistical estimation and pattern recognition, while also being known as an instance based learning type, where everything is approximated regionally, and computation is suspended till classification [3]. According to Application of k-NN and Naïve Bayes Algorithm in Banking and Insurance Domain, the k-NN algorithm rule is one of the most useful among machine learning algorithms. By comparing a testing dataset to the existing samples within a training dataset, using a distance measurement technique, we can appropriately classify a dataset without having to create any assumptions regarding the distribution of both our training and testing datasets. Since the k-NN algorithm can be applied for classification or regression, there is more flexibility as to how we want to apply k-NN to best address our statistical questions of interest.

[1] De Ville, Barry, and Padraic Neville. Decision Trees for Analytics: Using SAS Enterprise Miner. Cary, NC: SAS Institute, 2013.

[2] Hilbe, Joseph M. Logistic regression models. CRC press, 2009.

[3] Rahangdale, Gourav, Manish Ahirwar, and Mahesh Motwani. "Application of k-NN and Naive Bayes Algorithm in Banking and Insurance Domain." International Journal of Computer Science Issues (IJCSI) 13, no. 5 (2016): 69.

[4] Salkind, Neil J., ed. Encyclopedia of research design. Vol. 1. Sage, 2010.

1.2 Statistical questions of interest

To answer the primary question of interest, we propose visualizing our data with a decision tree, then using k-nearest neighbor to classify our dataset, and finally using logistic regression to frame our model. In terms of our secondary question, we will use data exploratory analysis with half of our variables to infer which data type is our strongest predictor for client subscription.

- 1. Primary Question: Build a prediction model for whether a client will sign on to a long-term deposit.
- 2. Secondary Question: What type of client data from our dataset most strongly predicts if a client will sign onto a long-term deposit? (e.g. Customer Background Data, Telemarketing Data, Economic Data, Other)

2 Analysis Plan

The bank-additional-full data consists of different client data with variables including age, marital status, gender, education, job, etc.. The purpose of the original data was aimed at determining whether or not there was a significant relationship between different predictor variables, and what directly affects clients in deciding whether or not to subscribe onto a long-term deposit. A prediction model can be built to better understand the different factors that play a role in their decision. We choose the following variables to be analyzed: duration, campaign, emp.var.rate, cons.price.idx, cons.conf.idx, euribor3m, nr.employed, job, marital, education, default, housing, loan, contact, month, day of week, and outcome. In doing so, we were able to identify through correlation testing for the numeric variables the factors that affect the deposit decisions.

We used predictive models such as **classification trees** to visualise the data such that the target's variable values could be predicted on other values, **k- Nearest Neighbor** to approximate the association between the independent variables and the continuous outcomes, and **logistic regression** to frame our binary output model through logistic function. We also used a **Support Vector Matrix (SVM)** which was used to predict the decision which has a higher sensitivity with less computation on the long term deposit based on the factors.

2.1 Population and study design

The data used was directly collected from a telemarketing campaign. The data was collected from May 2008 to November 2010 through a series of phone calls. The "product" they tried to sell was essentially a long-term bank deposit, which typically took more than one phone call for a customer to reach an affirmative decision. The data collected focuses on 17 key attributes, broken up into 4 subcategories - bank client data, data related to last contact with the campaign, social-economic context attributes, and other related data. The output is whether a client had ultimately decided to subscribe to a long-term deposit.

We identified the main key attributes that are significant to the study used. This also includes determining the data type, ie. whether a variable is continuous or categorical. The study did analyze the attributes, and identified the significance of each variable on the outcome.

- Our 20 attributes can be categorized into 4 types of different client data:

1. Client background data: age, job, marital status, education, default, housing, and loan
2. Bank telemarketing data: contact, month, day of the week, and duration
3. Socio-economic data: employment variation rate, consumer price index, consumer confidence index, 3 month Euribor rate, and number of employees 4.Other data: campaign, past days, previous, and past outcome

We will be focusing on this attribute categorization and applying exploratory analysis to answer our secondary question.

Definition of Input Variables

age - Age of the client- (numeric)

job - Client's occupation - (categorical) (admin, bluecollar, entrepreneur, housemaid, management, retired, selfemployed, services, student, technician, unemployed, unknown)

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

education - Client's education level - (categorical) (basic.4y, basic.6y, basic.9y, high.school, illiterate, professional.course, university.degree, unknown)

default - Indicates if the client has credit in default - (categorical) (no, yes, unknown)

housing - Does the client as a housing loan? - (categorical) (no, yes, unknown)

loan - Does the client as a personal loan? - (categorical) (no, yes, unknown')

contact - Type of communication contact - (categorical) (cellular, telephone)

month - Month of last contact with client - (categorical) (January - December)

day_of_week - Day of last contact with client - (categorical) (Monday - Friday)

duration - Duration of last contact with client, in seconds - (numeric) For benchmark purposes only, and not reliable for predictive modeling

campaign - Number of client contacts during this campaign - (numeric) (includes last contact)

pdays - Number of days from last contacted from a previous campaign - (numeric) (999 means client was not previously contacted)

previous - Number of client contacts performed before this campaign - (numeric)

poutcome - Previous marketing campaign outcome - (categorical) (failure, nonexistent , success)

emp.var.rate - Quarterly employment variation rate - (numeric)

cons.price.idx - Monthly consumer price index - (numeric)

cons.conf.idx - Monthly consumer confidence index - (numeric)

euribor3m - Daily euribor 3 month rate - (numeric)

nr.employed - Quarterly number of employees - (numeric)

Output variable (desired target) - **Term Deposit** - subscription verified (binary: 'yes','no')

2.2 Statistical Analysis

2.2.1 Descriptive Analysis

These dataset attributes denote client background data, socio-economic data, telemarketing data, and other data. Some attributes are numerical, and some are categorical. For data validation, we first checked our dataset for duplicate rows, missing data, and missing values by variable. Next for data cleaning, we removed rows with columns that had missing values, saved duplicated rows, and filled in missing values. Since certain attributes needed to be transformed into a numeric variable, we used the as.numeric function to be able to better fit our model. After the final removal of duplicate rows, we verified our new, compact "bank" dataset is fully cleaned. Finally, we coded a binary response (yes/no) to represent our response variable (y).

2.2.1.1 Set up and Data Wrangling

Transform all of the quantitative values into numeric class for easier manipulation later on. See how many rows have missing data

```
# this allows for the data to be separated rather than being mushed together.
# to see how many rows have missing data
sum(!complete.cases(bank_additional_full))
```

```
## [1] 0
```

Show the missing values within the dataset for each categories

```
##             age             job         marital       education         default
##               0               0               0               0               0
##         housing            loan         contact           month     day_of_week
##               0               0               0               0               0
##        duration        campaign           pdays        previous        poutcome
##               0               0               0               0               0
##    emp.var.rate  cons.price.idx   cons.conf.idx       euribor3m     nr.employed
##               0               0               0               0               0
##               y
##               0
```

Write the clean data into a separate csv that we can work with, and get rid of a column in our new dataset when we go to read it.

Transform all of the quantitative values into numeric class for easier manipulation later on.

```
bank$age <- as.numeric(bank$age)
bank$duration <- as.numeric(bank$duration)
bank$campaign <- as.numeric(bank$campaign)
bank$pdays <- as.numeric(bank$pdays)
bank$previous <- as.numeric(bank$previous)
bank$emp.var.rate <- as.numeric(bank$emp.var.rate)
bank$cons.price.idx <- as.numeric(bank$cons.price.idx)
bank$cons.conf.idx <- as.numeric(bank$cons.conf.idx)
bank$nr.employed <- as.numeric(bank$nr.employed)
```

Checking if there are any missing data within each categories and push them into a category of there own. Change the subscription category into binary as well.

```
bank$job = fct_explicit_na(bank$job, "missing")
bank$marital = fct_explicit_na(bank$marital, "missing")
bank$education = fct_explicit_na(bank$education, "missing")
bank$default = fct_explicit_na(bank$default, "missing")
bank$loan = fct_explicit_na(bank$loan, "missing")
bank$contact = fct_explicit_na(bank$contact, "missing")
bank$poutcome = fct_explicit_na(bank$poutcome, "missing")
bank$day_of_week = fct_explicit_na(bank$day_of_week, "missing")
bank$housing = fct_explicit_na(bank$housing, "missing")
bank$month = fct_explicit_na(bank$month, "missing")
bank$y =ifelse(bank$y =='yes',1,0) # transforming 'yes' category into a binary 1=yes 0=no
```

2.2.1.2 Exploratory Analysis

Summary of the Bank Additional Full Data set

Data Structure of Dataset

```
## 'data.frame':    41188 obs. of  21 variables:
##  $ age           : num  56 57 37 40 56 45 59 41 24 25 ...
##  $ job           : Factor w/ 12 levels "admin.","blue-collar",..: 4 8 8 1 8 8 1 2 10 8 ...
##  $ marital       : Factor w/ 4 levels "divorced","married",..: 2 2 2 2 2 2 2 2 3 3 ...
##  $ education     : Factor w/ 8 levels "basic.4y","basic.6y",..: 1 4 4 2 4 3 6 8 6 4 ...
##  $ default       : Factor w/ 3 levels "no","unknown",..: 1 2 1 1 1 2 1 2 1 1 ...
##  $ housing       : Factor w/ 3 levels "no","unknown",..: 1 1 3 1 1 1 1 1 3 3 ...
##  $ loan          : Factor w/ 3 levels "no","unknown",..: 1 1 1 1 3 1 1 1 1 1 ...
##  $ contact       : Factor w/ 2 levels "cellular","telephone": 2 2 2 2 2 2 2 2 2 2 ...
##  $ month         : Factor w/ 10 levels "apr","aug","dec",..: 7 7 7 7 7 7 7 7 7 7 ...
##  $ day_of_week   : Factor w/ 5 levels "fri","mon","thu",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ duration      : num  261 149 226 151 307 198 139 217 380 50 ...
##  $ campaign      : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ pdays         : num  999 999 999 999 999 999 999 999 999 999 ...
##  $ previous      : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ poutcome      : Factor w/ 3 levels "failure","nonexistent",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ emp.var.rate  : num  1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1 ...
##  $ cons.price.idx: num  94 94 94 94 94 ...
##  $ cons.conf.idx : num  -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 ...
##  $ euribor3m     : num  4.86 4.86 4.86 4.86 4.86 ...
##  $ nr.employed   : num  5191 5191 5191 5191 5191 ...
##  $ y             : num  0 0 0 0 0 0 0 0 0 0 ...
```

Summary of the Data set

age	job	marital	education	default	housing	loan	contact	month	day_of_week	duration	campaign	pdays	previous	poutcon
Min. :17.00	admin. :10422	divorced: 4612	university.degree :12168	no :32588	no :18622	no :33950	cellular :26144	may :13769	fri:7827	Min. : 0.0	Min. : 1.000	Min. : 0.0	Min. :0.000	failure : 4
1st Qu.:32.00	blue-collar: 9254	married :24928	high.school : 9515	unknown: 8597	unknown: 990	unknown: 990	telephone:15044	jul : 7174	mon:8514	1st Qu.: 102.0	1st Qu.: 1.000	1st Qu.:999.0	1st Qu.:0.000	nonexiste
Median :38.00	technician : 6743	single :11568	basic.9y : 6045	yes : 3	yes :21576	yes : 6248	NA	aug : 6178	thu:8623	Median : 180.0	Median : 2.000	Median :999.0	Median :0.000	success :
Mean :40.02	services : 3969	unknown : 80	professional.course: 5243	NA	NA	NA	NA	jun : 5318	tue:8090	Mean : 258.3	Mean : 2.568	Mean :962.5	Mean :0.173	NA
3rd Qu.:47.00	management : 2924	NA	basic.4y : 4176	NA	NA	NA	NA	nov : 4101	wed:8134	3rd Qu.: 319.0	3rd Qu.: 3.000	3rd Qu.:999.0	3rd Qu.:0.000	NA
Max. :98.00	retired : 1720	NA	basic.6y : 2292	NA	NA	NA	NA	apr : 2632	NA	Max. :4918.0	Max. :56.000	Max. :999.0	Max. :7.000	NA
NA	(Other) : 6156	NA	(Other) : 1749	NA	NA	NA	NA	(Other): 2016	NA	NA	NA	NA	NA	NA

Frequency of Response Variable

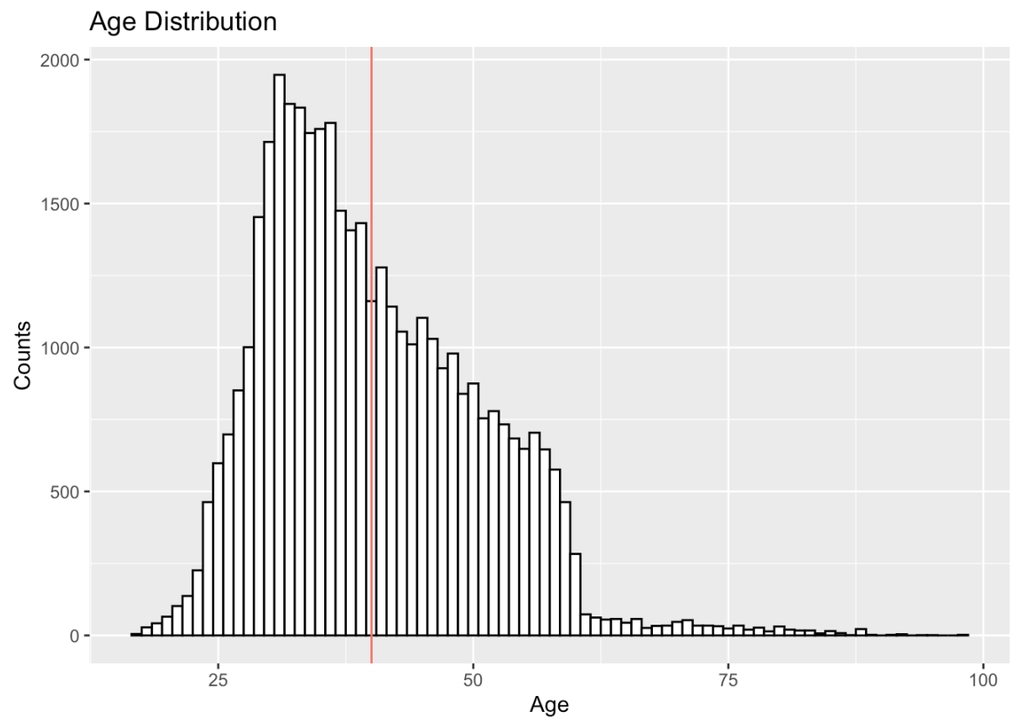
Var1	Freq
0	0.8873458
1	0.1126542

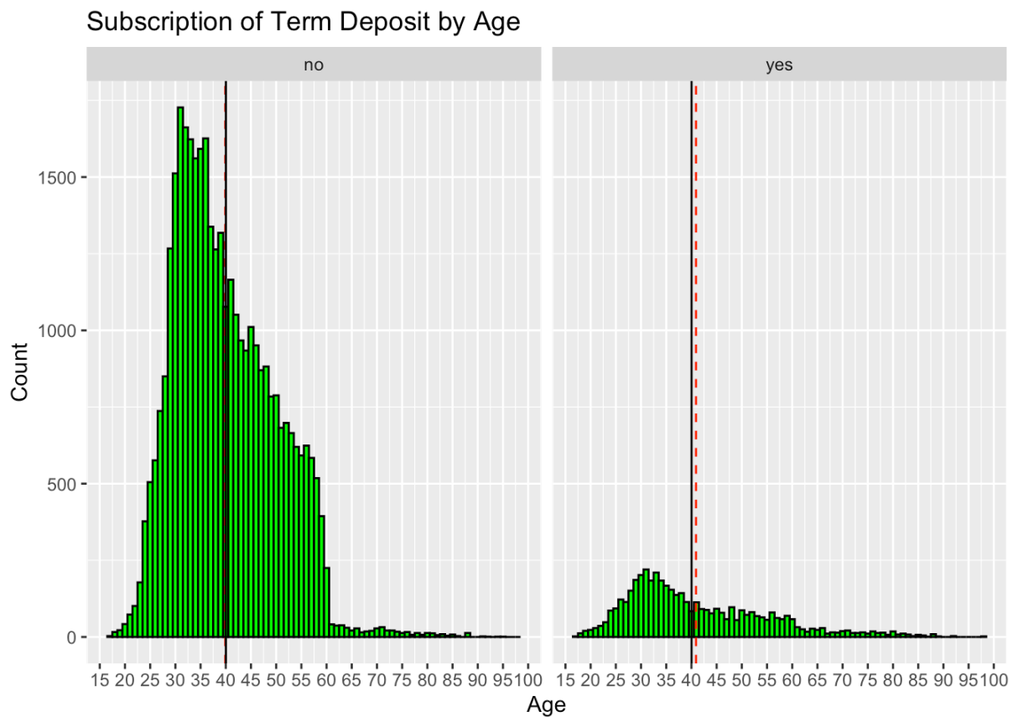
After observing the frequency of the response variable, it is evident that the predicted outcome (y) is skewed to "no"" by 88.8%

Age

The majority of clients are between the ages of 32 (1st Quartile) and 47 (3rd Quartile) with a mean of about 40 as indicated by the red vertical line on the histogram.

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
17	32	38	40.02	47	98



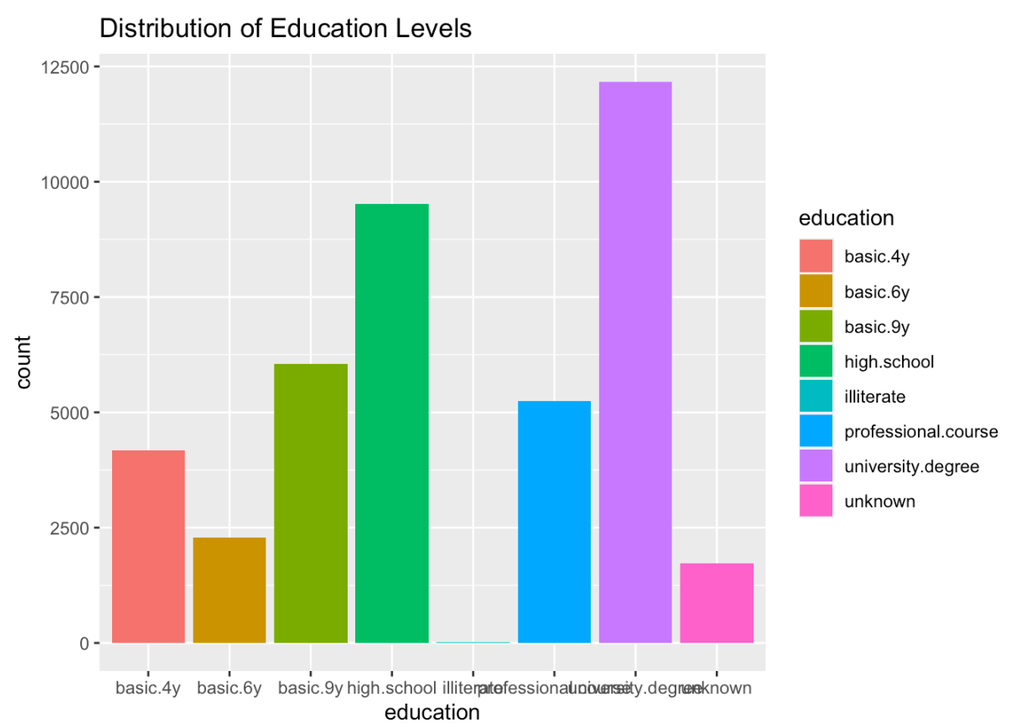


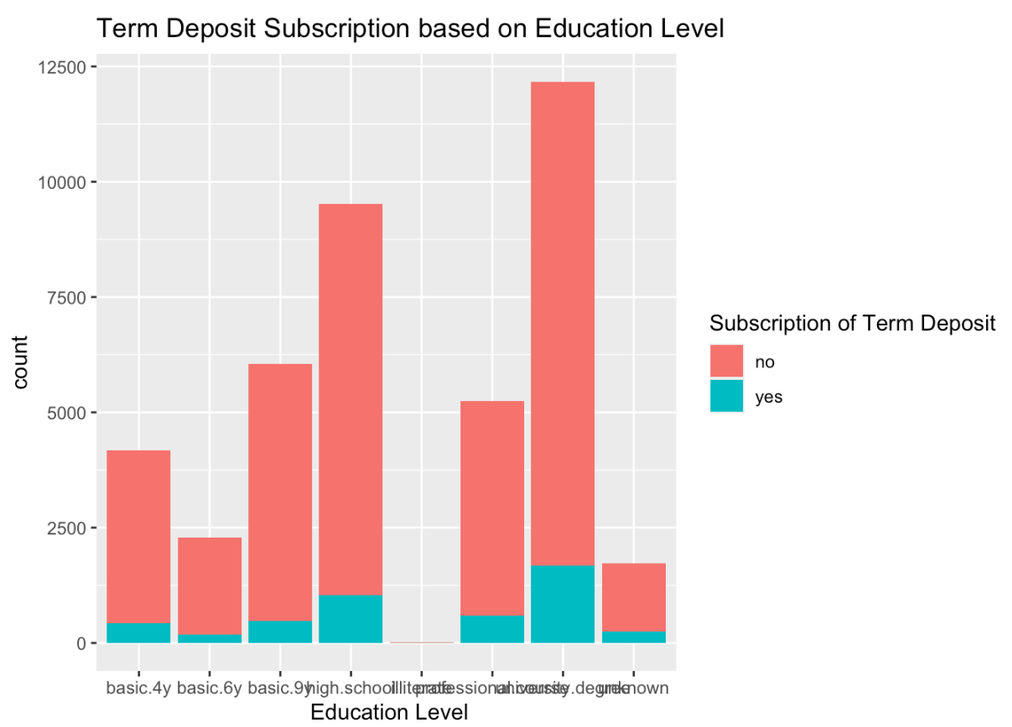
Age Percentage to Subscribe

age	age.cnt	pct.con.yes
31	1947	11.299435
32	1846	9.967497
33	1833	11.456629
36	1780	8.651685
35	1759	9.494031
34	1745	10.544413

Education

Having higher education is seen to contribute to higher subscription of a term deposit. Most clients who subscribe have received secondary education or above. In addition, clients with university degrees seem to have a higher rate of subscription compared to other clients.



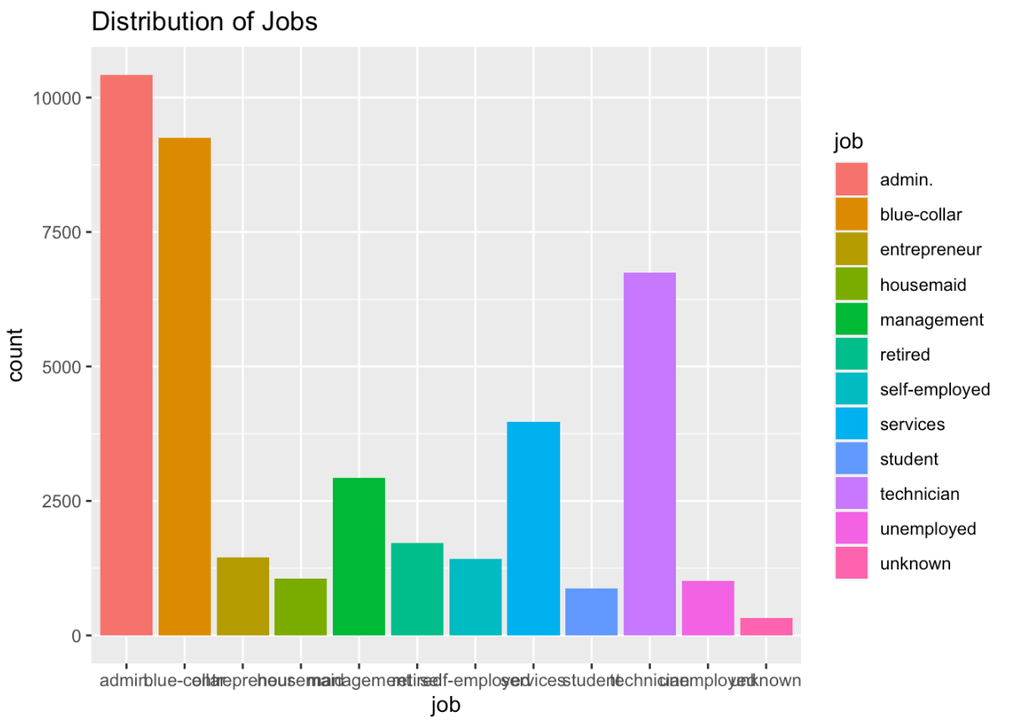


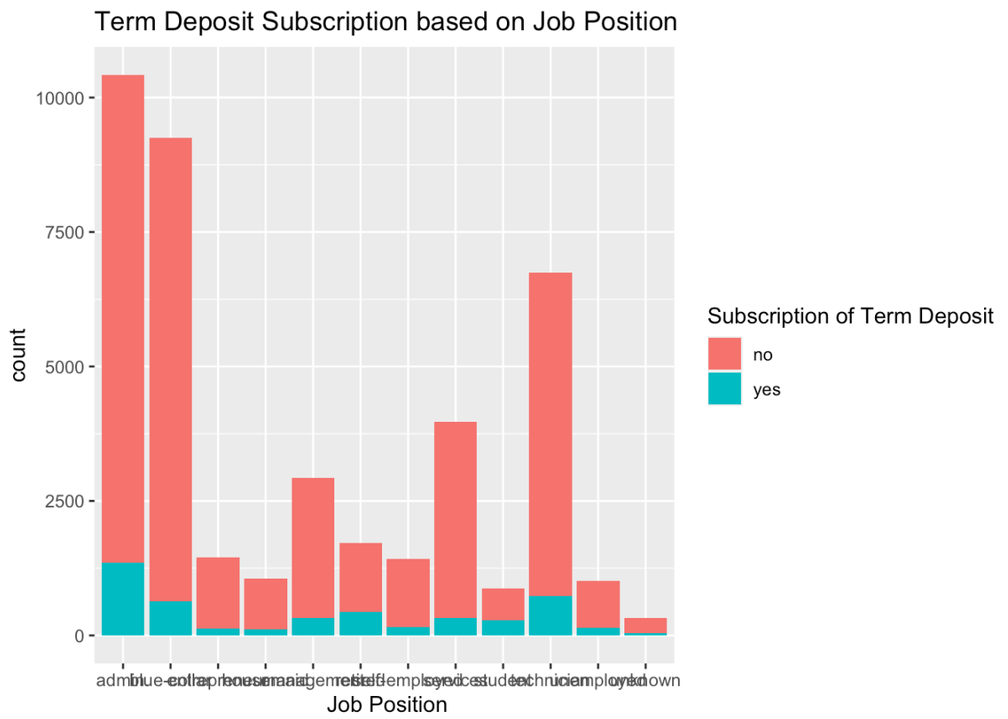
Education Percentage to Subscribe

education	edu.cnt	pct.con.yes
university.degree	12168	13.724523
high.school	9515	10.835523
basic.9y	6045	7.824649
professional.course	5243	11.348465
basic.4y	4176	10.249042
basic.6y	2292	8.202443

Job

The majority of clients have administrative, blue-collar, or technician jobs. However, the students and those retired have the highest job percentage (31.4% and 25.3% respectively) known to subscribe.





Job Percentage to Subscribe

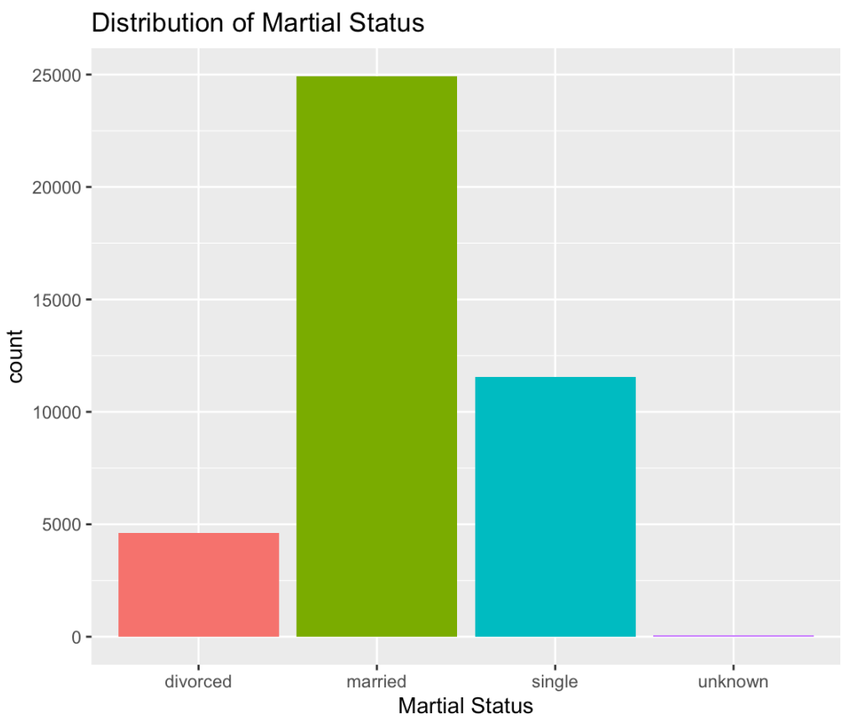
job	job.cnt	pct.con.yes
admin.	10422	12.972558
blue-collar	9254	6.894316
technician	6743	10.826042
services	3969	8.138070
management	2924	11.217510
retired	1720	25.232558

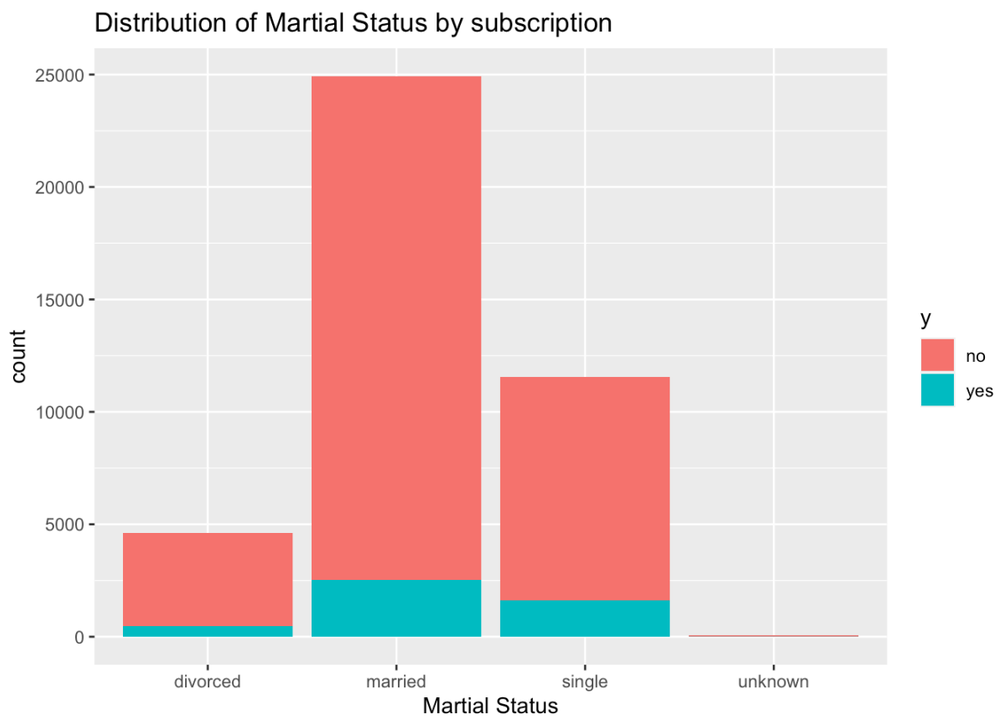
Marital Status

The majority of clients are married. However, the percentage of those who are married, divorced or single are roughly similar in terms of likelihood of subscription.

marital Percentage to Subscribe

marital	marital.cnt	pct.con.yes
married	24928	10.15725
single	11568	14.00415
divorced	4612	10.32090
unknown	80	15.00000





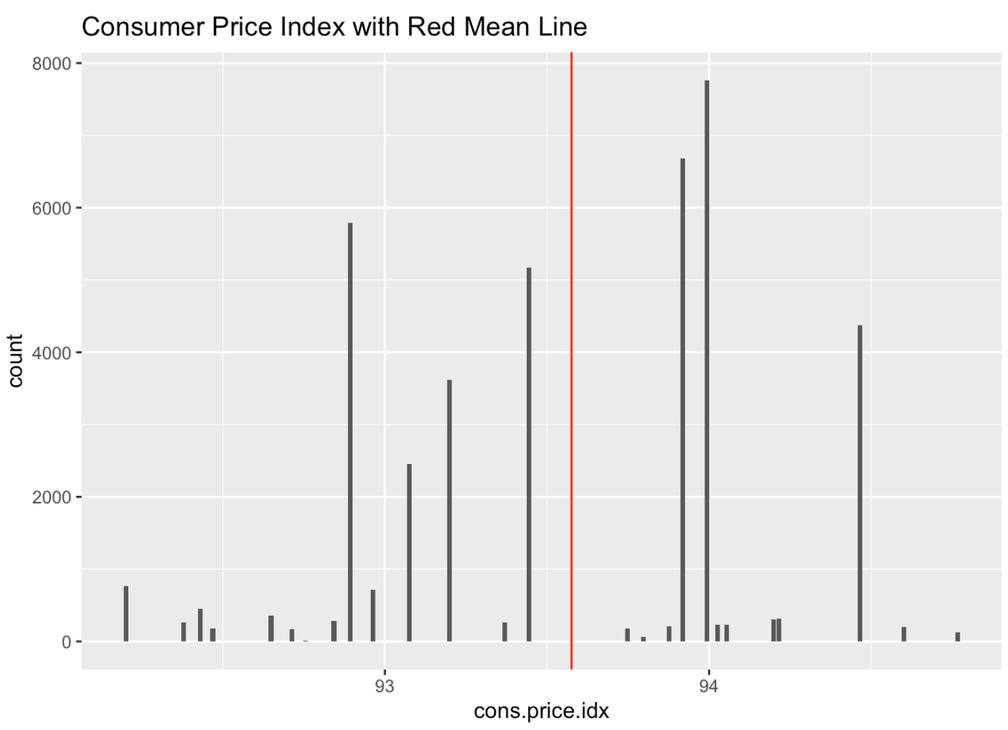
Consumer Price Index

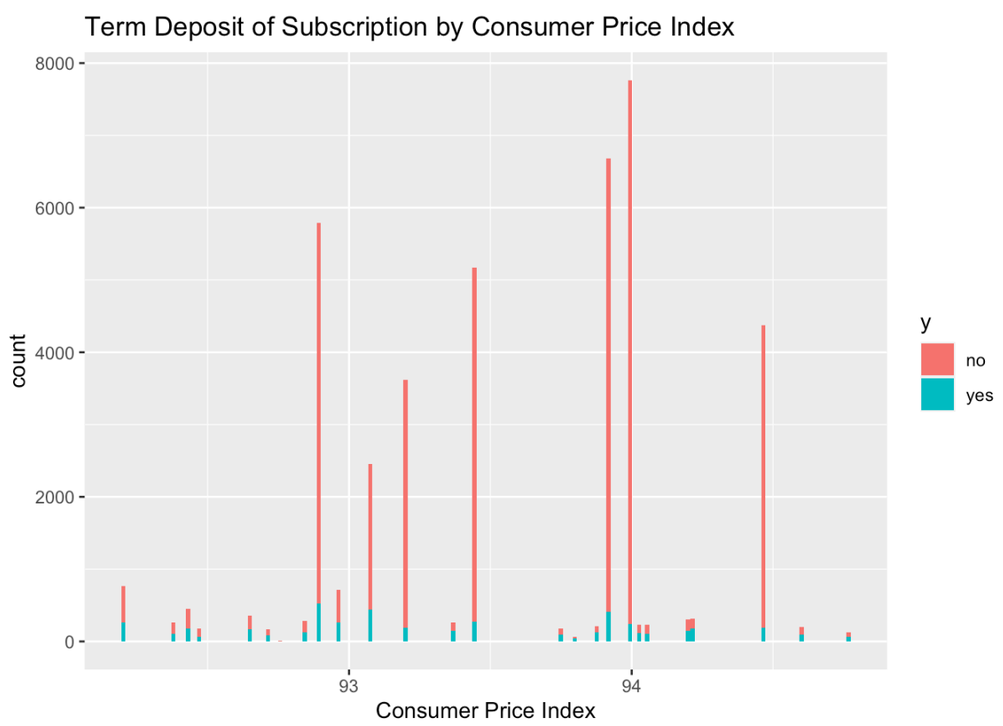
The monthly consumer price index either does not display enough, or displays excess information for us to draw conclusions from. Therefore, we can go ahead and disregard this variable when considering our secondary question.

Consumer Price Index Percentage to Subscribe

cons.price.idx	cpi.cnt	pct.con.yes
93.994	7763	3.091588
93.918	6685	6.088257
92.893	5794	9.043838

93.444	5175	5.236715
94.465	4374	4.298125
93.200	3616	5.254425



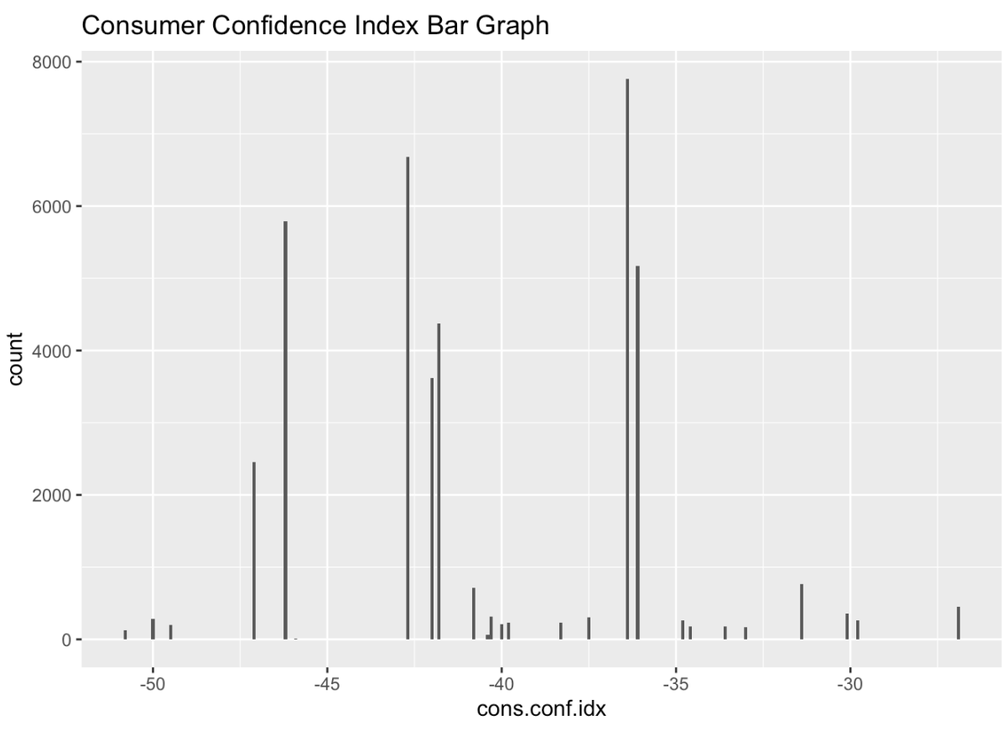


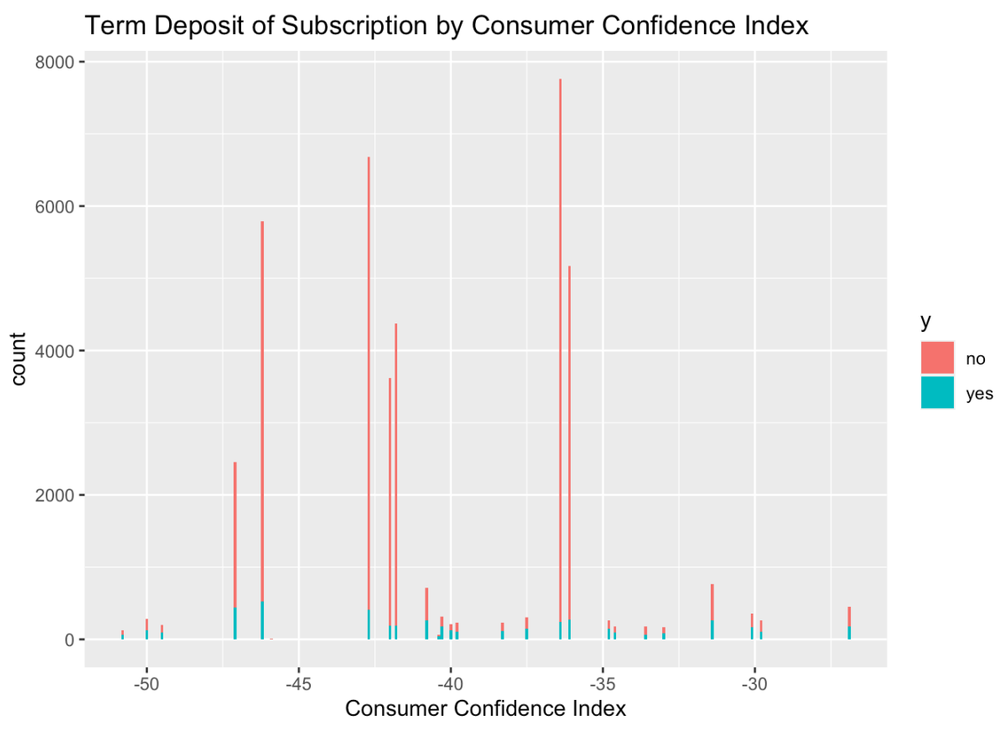
Consumer Confidence Index

The majority of clients have a consumer confidence index between -46 to -36. However, from the red bar graph, it is evident the clients' consumer confidence index percentage of subscribing is very low (which we can consider irrelevant).

Consumer Confidence Index Percentage to Subscribe

cons.conf.idx	cci.cnt	pct.con.yes
-36.4	7763	3.091588
-42.7	6685	6.088257
-46.2	5794	9.043838
-36.1	5175	5.236715
-41.8	4374	4.298125
-42.0	3616	5.254425



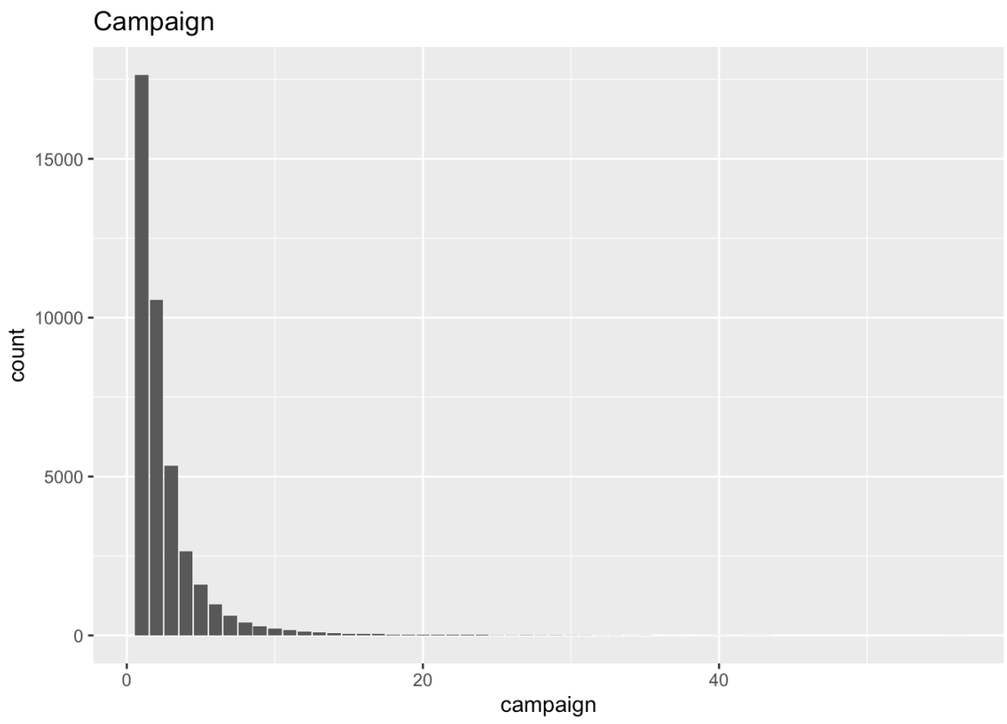


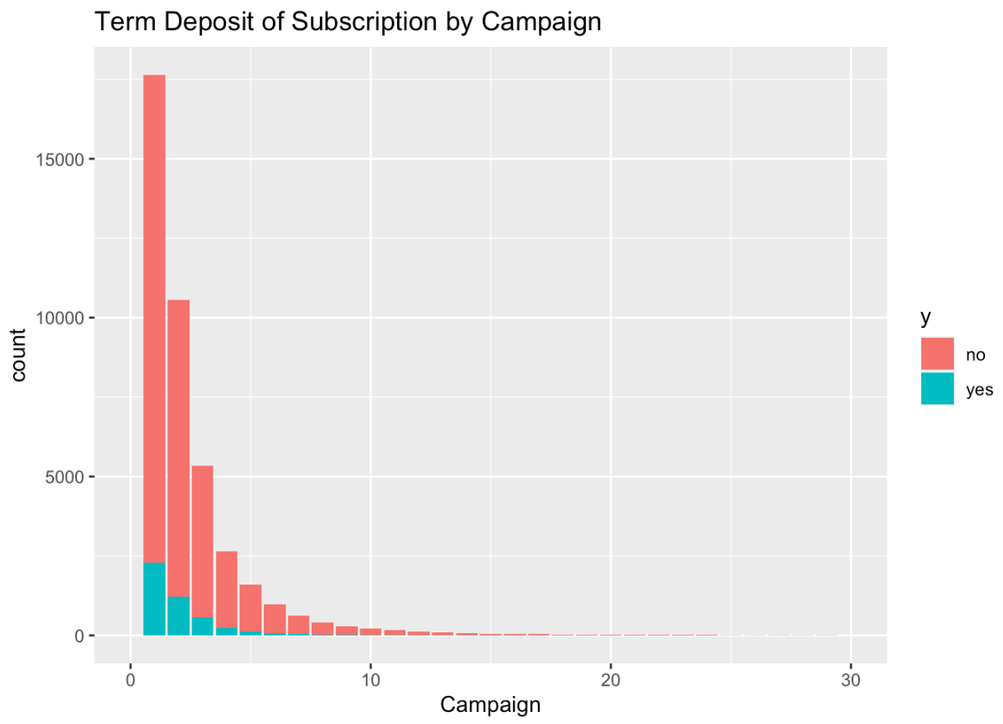
Campaign

From the barchart, there will be no subscription beyond 7 contacts during the campaign. Future campaigns could improve resource utilization by setting limits to contacts during a campaign. For instance, future campaigns should focus on first 3 contacts as it will have a higher subscription rate.

Campaign Percentage to Subscribe

campaign	contact.cnt	pct.con.yes
1	17642	13.037071
2	10570	11.456954
3	5341	10.747051
4	2651	9.392682
5	1599	7.504690
6	979	7.660878



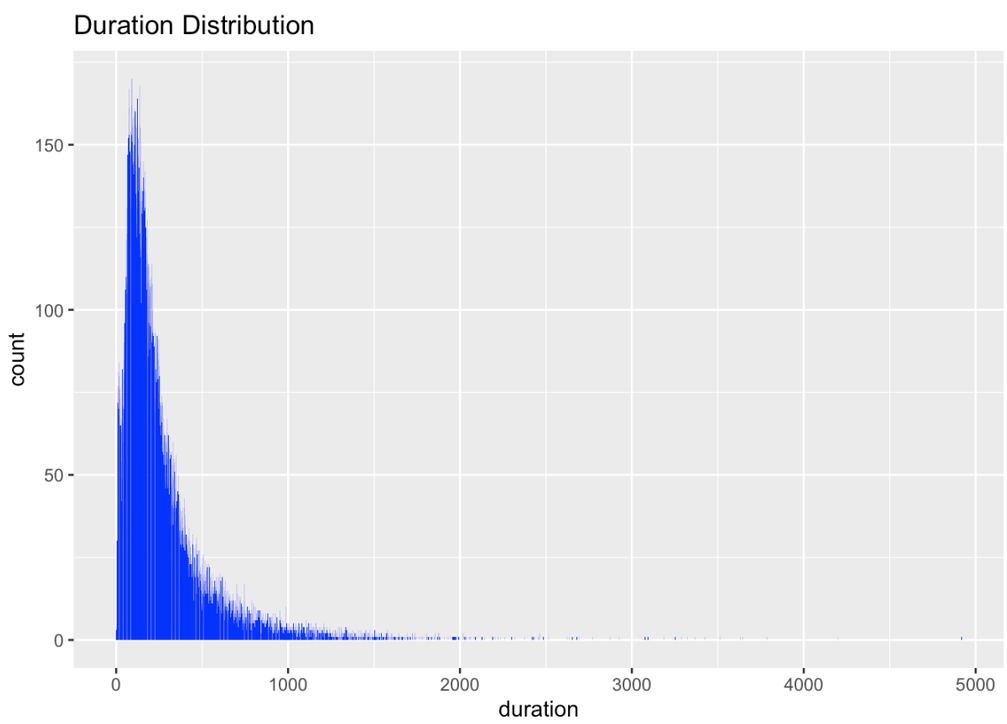


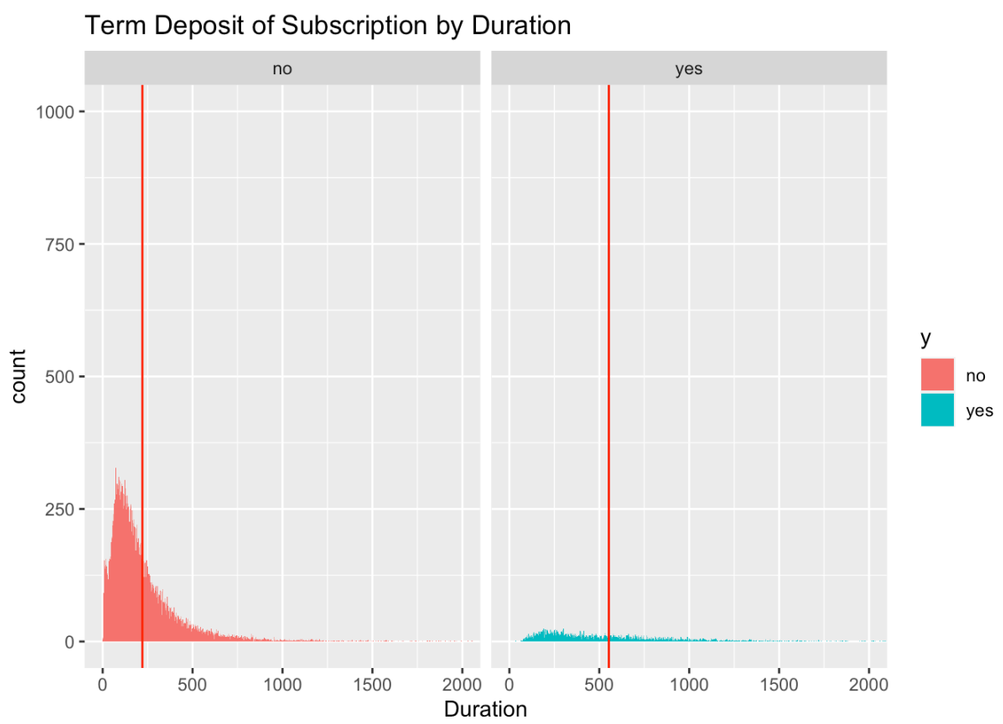
Duration

Given the graph, duration does not say much about if a client will sign onto a long-term deposit or not. We can go ahead and disregard this variable to answer our secondary question.

Duration Percentage to Subscribe

duration	contact.cnt	pct.con.yes
85	170	1.1764706
90	170	1.1764706
136	168	4.7619048
73	167	0.5988024
124	164	2.4390244
87	162	1.8518519



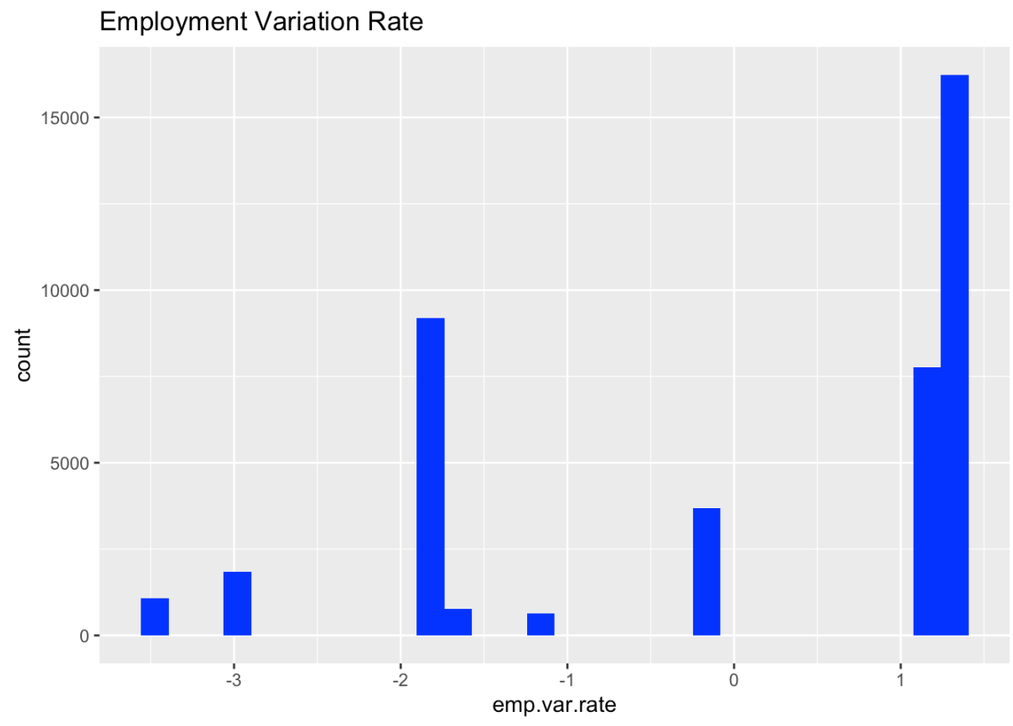


Employment Variation Rate

Clients with quarterly employee variation rate between -1 to -3 have a higher likelihood of subscribing on to a deposit.

Employment Variation Rate Percentage to Subscribe

emp.var.rate	contact.cnt	pct.con.yes
1.4	16234	5.334483
-1.8	9184	15.908101
1.1	7763	3.091588
-0.1	3683	6.299213
-2.9	1663	35.718581
-3.4	1071	42.390289



2.2.2 Main Analysis

Moving forward, we plan to dive into our different types of algorithms to determine the best-suited predictive model for our dataset. The classification goal is to predict if the client will subscribe (yes/no) to a term deposit (variable y). The list below will further clarify why we chose each method and how we plan to build the best-suited predictive model to determine our primary question.

Decision Tree

- A decision tree is a classification algorithm that supports non linearity. Decision trees can provide understandable explanations over the prediction and make it easier to interpret comparison for categorical independent variables.. It is one way to display an algorithm that only contains conditional control statements. Use 70% of our dataset as training data and 30% as our test data.

General Logistic Regression

- In contrast to linear regression where the dependent variable is continuous, the logistic regression binary classification model uses a different method of estimating the parameters (logistic function) to give results with lower variances. The output of the logistic regression will be a probability ($0 \leq x \leq 1$), and can be used to predict the binary 0 or 1 as the output.

k-Nearest Neighbors (k-NN)

- In k-NN, we explore the features' neighborhood and assume the test data point to be similar to them and derive the output. When k-NN is used for classification, the output can be calculated as the class with the highest frequency from the K-most similar instances.
- Use 70% of our dataset as training data and 30% as our test data.

We will fit each model to implement a predictive model. Additionally, we will use Confusion Matrix to return accuracy rates for each method. Finally, we will calculate and plot the ROC curve to return AUC rates for each of the three methods.

3 Extra-Credit Method: Support Vector Machine (SVM)

Support vector machine (SVM) was used because it is a method of supervised learning that can be used for both classification and regression problems. SVM supports both linear and non-linear solutions using the kernel trick. It is good at pattern recognition which is fitting for the dataset that we have [5]. SVM constructs a high dimensional space, the hyperplane that separates the data into 2 categories with its particular linear classifiers based on the margin maximization principle. More importantly, it has a higher accuracy with less computational power. In using a hyperplane that provides the best generalization for the margin between the 2 data categories that we chose to be factors affecting the decision for the long term deposit [6].

[5] Adankon M., Cheriet M. (2009) Support Vector Machine. In: Li S.Z., Jain A. (eds) Encyclopedia of Biometrics. Springer, Boston, MA.

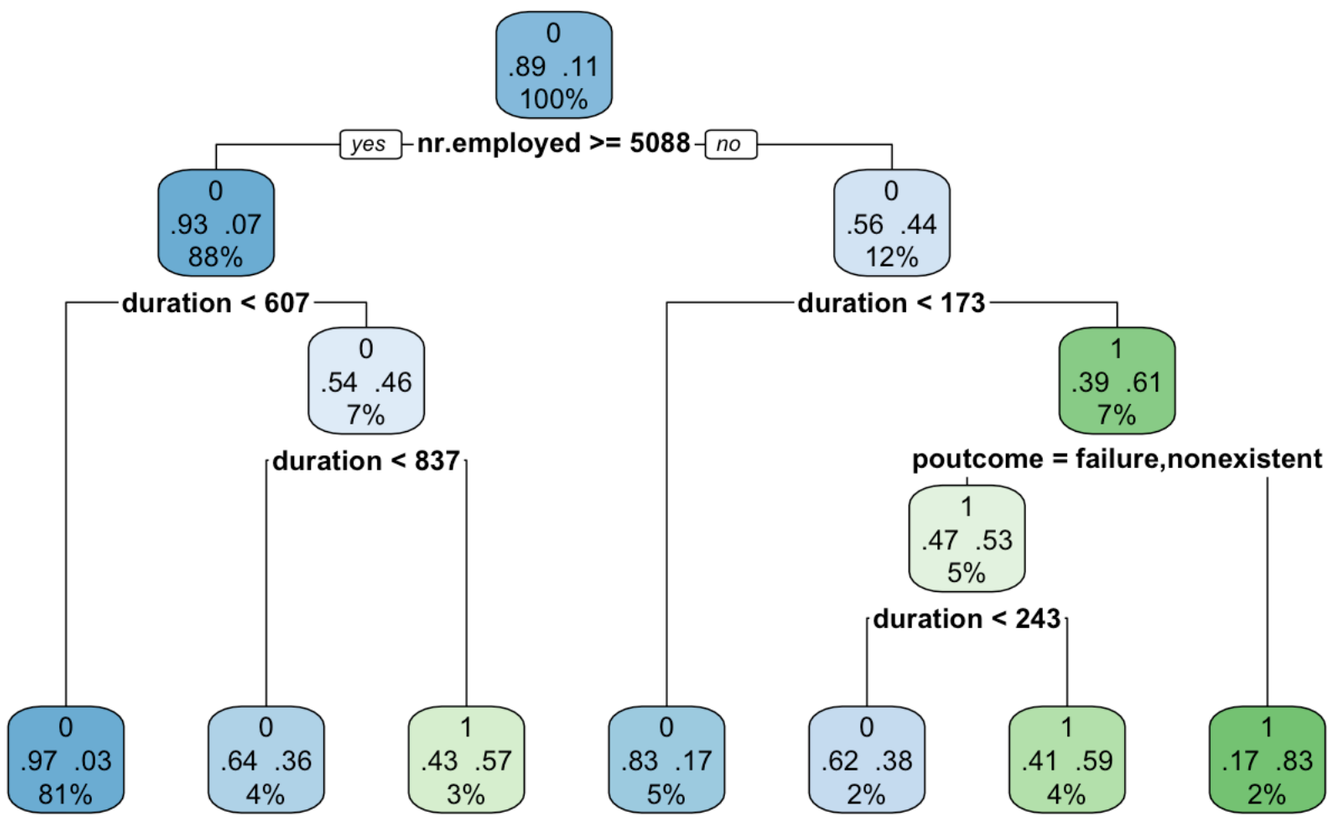
[6] Ashanira Mat Deris, Azlan Mohd Zain, Roselina Sallehuddin, Overview of Support Vector Machine in Modeling Machining Performances, Procedia Engineering, Volume 24, 2011, Pages 308-312, ISSN 1877-7058

4 Result

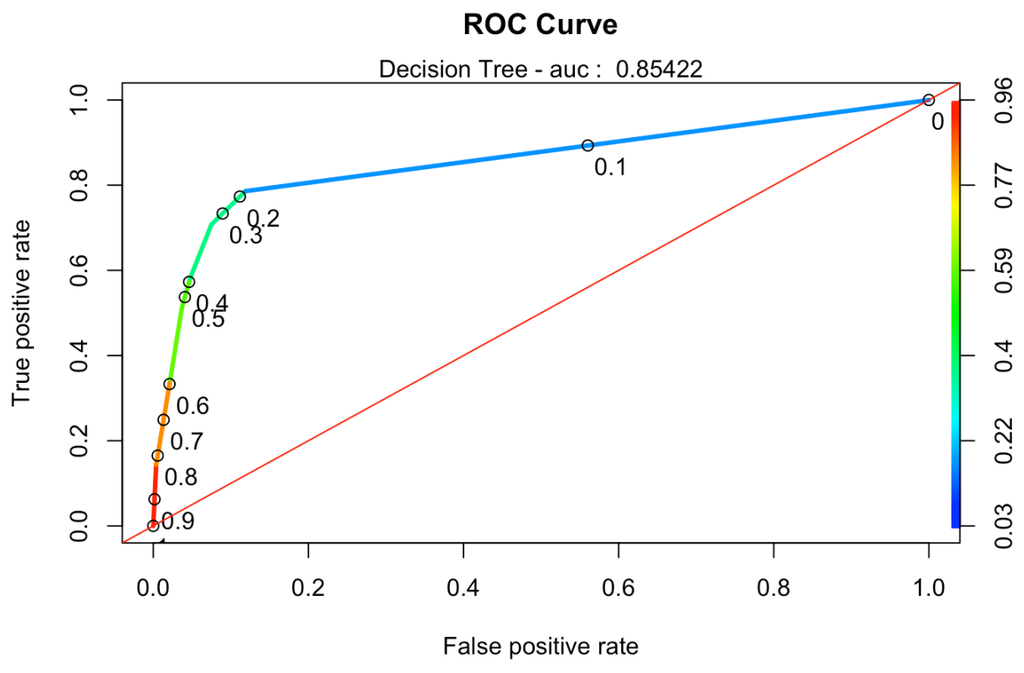
There are many metrics to evaluate our methods, however, we will be using Confusion matrix/Accuracy and ROC/AUC to determine which algorithm performed the best.

4.1 Descriptive Analysis

Decision Tree




```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     0     1
##          0 10560   404
##          1   680   712
##
##                Accuracy : 0.9123
##                  95% CI : (0.9071, 0.9172)
##     No Information Rate : 0.9097
##     P-Value [Acc > NIR] : 0.1614
##
##                   Kappa : 0.5196
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.9395
##             Specificity : 0.6380
##          Pos Pred Value : 0.9632
##          Neg Pred Value : 0.5115
##              Prevalence : 0.9097
##          Detection Rate : 0.8546
##    Detection Prevalence : 0.8873
##       Balanced Accuracy : 0.7887
##
##        'Positive' Class : 0
##
```

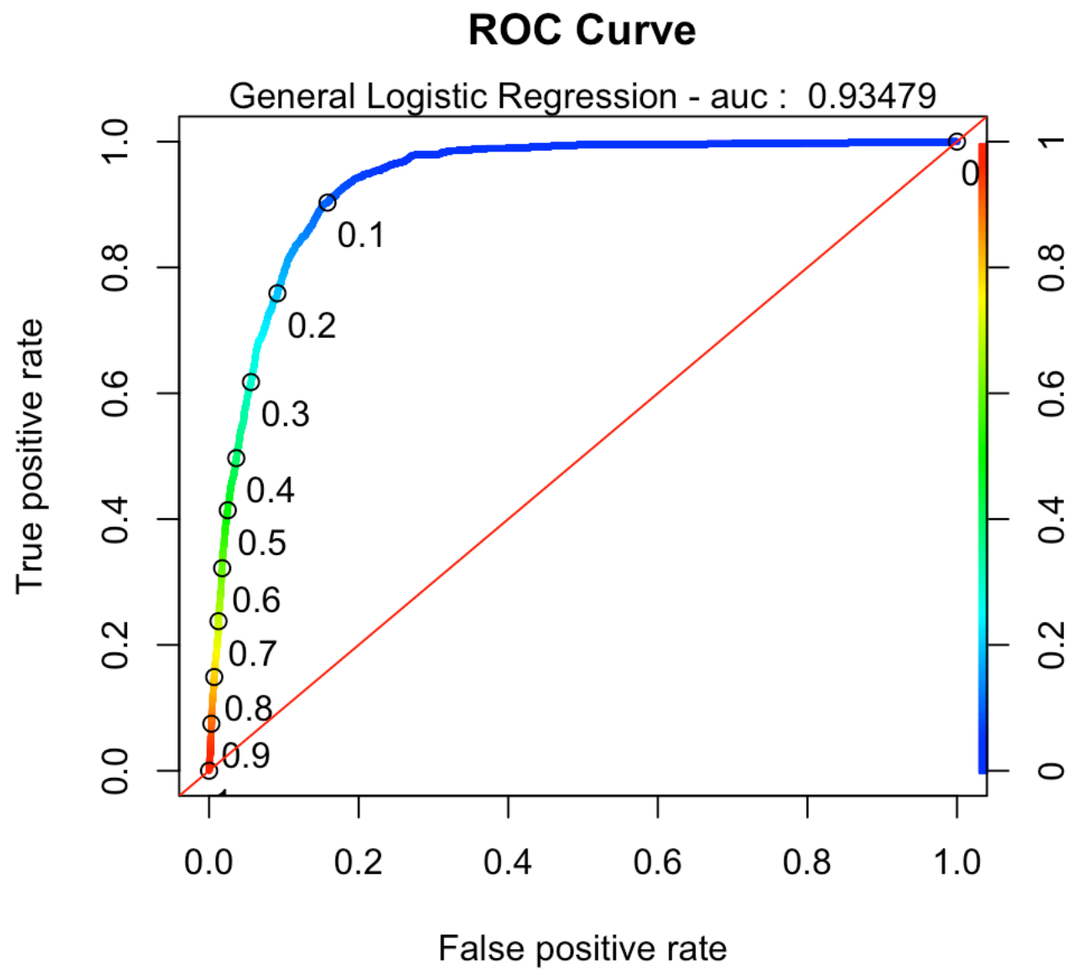



```
## NULL
```

General Logistic Regression

```
##
## Call:
## glm(formula = y ~ age + job + marital + education + default +
##     housing + loan + contact + month + duration + day_of_week +
##     campaign + pdays + previous + poutcome + emp.var.rate + cons.price.idx +
##     cons.conf.idx + euribor3m + nr.employed, family = "binomial",
##     data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -5.9514  -0.3035  -0.1876  -0.1335   3.1540
##
## Coefficients: (1 not defined because of singularities)
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                 -2.365e+02  4.589e+01  -5.154 2.55e-07 ***
## age                          1.706e-03  2.914e-03   0.586 0.558120
## jobblue-collar              -1.423e-01  9.450e-02  -1.505 0.132227
## jobentrepreneur             -6.015e-02  1.465e-01  -0.411 0.681331
## jobhousemaid                -3.979e-02  1.743e-01  -0.228 0.819436
## jobmanagement               -8.341e-02  1.035e-01  -0.806 0.420091
## jobretired                   2.946e-01  1.271e-01   2.319 0.020400 *
## jobself-employed            -2.445e-01  1.443e-01  -1.695 0.090079 .
## jobservices                 -1.056e-01  1.032e-01  -1.023 0.306128
## jobstudent                   3.047e-01  1.333e-01   2.286 0.022275 *
##  [ reached getOption("max.print") -- omitted 44 rows ]
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 20299  on 28831  degrees of freedom
## Residual deviance: 12030  on 28779  degrees of freedom
## AIC: 12136
##
## Number of Fisher Scoring iterations: 10
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     0     1
##          0 10350   533
##          1   614   859
##
##                Accuracy : 0.9072
##                  95% CI : (0.9019, 0.9122)
##     No Information Rate : 0.8873
##     P-Value [Acc > NIR] : 4.468e-13
##
##                   Kappa : 0.5472
##
##  Mcnemar's Test P-Value : 0.01817
##
##             Sensitivity : 0.61710
##             Specificity : 0.94400
##          Pos Pred Value : 0.58316
##          Neg Pred Value : 0.95102
##              Prevalence : 0.11266
##          Detection Rate : 0.06952
##    Detection Prevalence : 0.11921
##       Balanced Accuracy : 0.78055
##
##        'Positive' Class : 1
##
```

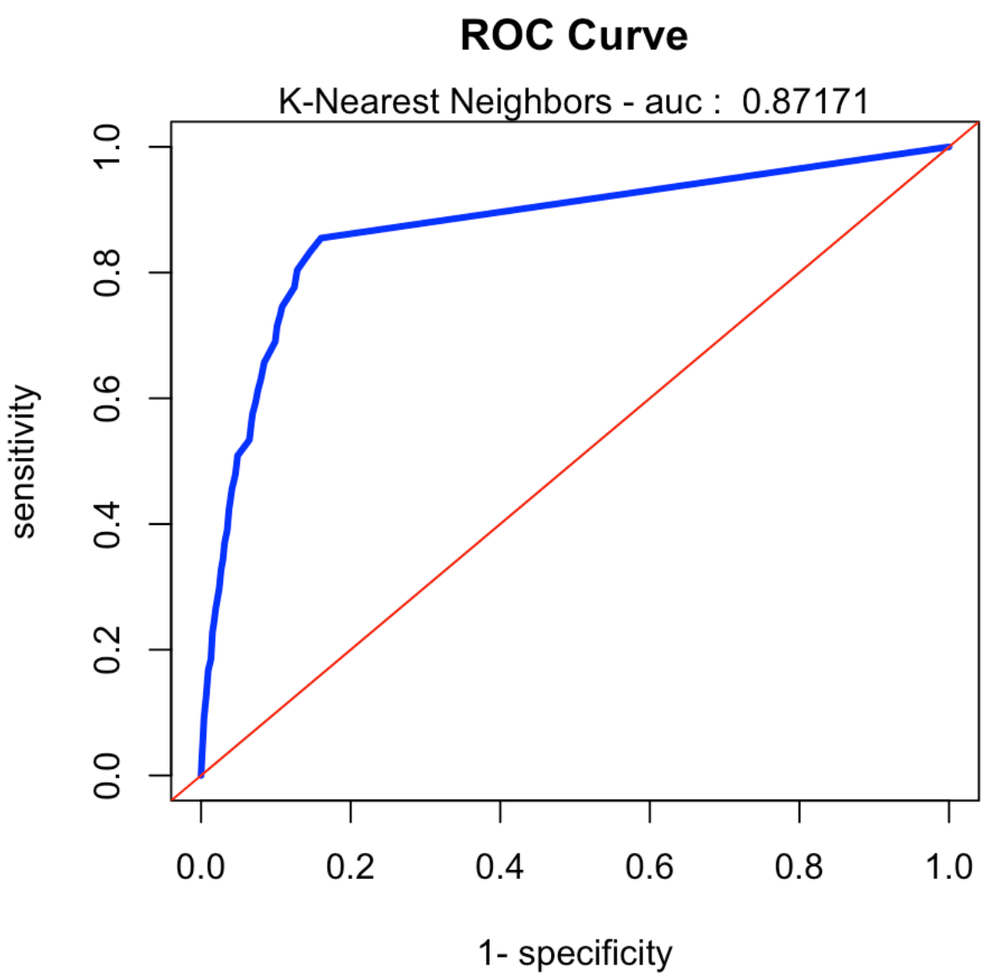



```
## NULL
```

K-Nearest Neighbor

```
##
## Call:
## kknn(formula = y ~ age + job + marital + education + default +     housing + loan + contact + month + duration
+ day_of_week +     campaign + pdays + previous + poutcome + emp.var.rate + cons.price.idx +     cons.conf.idx +
euribor3m + nr.employed, train = train, test = test,     na.action = getOption(max.print = 20), k = 5, distance =
2,     scale = FALSE)
##
## Response: "nominal"
##    fit prob.0 prob.1
## 1    0      1      0
## 2    0      1      0
## 3    0      1      0
## 4    0      1      0
## 5    0      1      0
## 6    0      1      0
## 7    0      1      0
## 8    0      1      0
## 9    0      1      0
## 10   0      1      0
## 11   0      1      0
## 12   0      1      0
## 13   0      1      0
## 14   0      1      0
## 15   0      1      0
## 16   0      1      0
##  [ reached 'max' / getOption("max.print") -- omitted 12340 rows ]
```

```
## Confusion Matrix and Statistics
##
##
## bank_knn_pred     0     1
##             0 10429   684
##             1   535   708
##
##                Accuracy : 0.9013
##                  95% CI : (0.896, 0.9065)
##     No Information Rate : 0.8873
##     P-Value [Acc > NIR] : 2.895e-07
##
##                   Kappa : 0.4824
##
##  Mcnemar's Test P-Value : 2.246e-05
##
##             Sensitivity : 0.5086
##             Specificity : 0.9512
##          Pos Pred Value : 0.5696
##          Neg Pred Value : 0.9385
##              Prevalence : 0.1127
##          Detection Rate : 0.0573
##    Detection Prevalence : 0.1006
##       Balanced Accuracy : 0.7299
##
##        'Positive' Class : 1
##
```



4.2 Inferential Analysis

For our confusion matrices, we will look at the accuracy, sensitivity, and specificity of each method. Accuracy measures all our correct responses compared to the overall responses. Sensitivity measures the value of true positives over all actual positives. Specificity measures the value of true negatives over all actual negatives. The higher the accuracy, sensitivity, and specificity rates, the better performing our algorithms. We must look at all the factors to determine the overall performance.

The Receiver Operating Characteristics curve (ROC curve) will be the main measure for comparison. The ROC curve is a graphical plot that shows the diagnostic ability of a binary classifier system as its discrimination threshold is varied. It plots the specificity, true negative rate, against the sensitivity, true positive rate, at various threshold settings. The main measure returned from the ROC curve is the Area Under the Curve (AUC), which is equal to the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative one.

Decision Tree	**Results**
Accuracy	0.9123
Sensitivity	0.9395
Specificty	0.6380
AUC	0.85422

Logistic Regression	**Results**

Accuracy	0.9072
Sensitivity	0.61710
Specificty	0.94400
AUC	0.93479

KNN	**Results**
Accuracy	0.9013
Sensitivity	0.5086
Specificty	0.9512
AUC	0.87171

All of our algorithms return closely similar accuracies/misclassification rates. Decision tree returned the highest accuracy (0.9123) and sensitivity rate (0.9395). k-NN returned the highest specificity rate. However, from our ROC plots, logistic regression returns the highest AUC value (0.93479). The accuracy and specificity rates of our logistic regression model did perform high as well. For this reasoning, we will determine logistic regression as the main predictive model for whether a client will sign on to a long-term deposit.

Furthermore, from our exploratory analysis, client background data (age, job, marital status, education, default, housing and loan) returned a higher percentage of subscription rates compared to the other attribute categories. Compared to client background data, telemarketing and socio-economic data did not clearly indicate whether a client would subscribe to a long-term deposit. For example, variables such as “consumer price index” (socio-economic) and “duration” (telemarketing) displayed a wide range of different client information, but returned low subscription percentages. We can conclude that variables from client background data will be the strongest predictors of whether a client will subscribe to a long-term bank deposit or not.

5 Extra-Credit Method Results

We used the SVM matrix as a submethod to the other methods mentioned above. It is a higher accuracy model that uses less computational power. This method outputs the agreement between the factors on whether it is able to agree to a long term loan. We used a linear kernel model that predicted the results.

SVM (Kernel Fit)	**Results**
Accuracy	0.9030
Sensitivity	0.9781
Specificty	0.3118
Pos Pred Value	0.9180
Neg Pred Value	0.6439
Agreement True	0.90304
Agreement False	0.09605

SVM (Gaussian Fit)	**Results**
Accuracy	0.9113
Sensitivity	0.9743
Specificty	0.4152
Pos Pred Value	0.9292
Neg Pred Value	0.6721
Agreement True	0.9113
Agreement False	0.0887

From the SVM matrix, it returns a lower accuracy (0.9030) which shows that in fact this is a slightly less accurate model as compared to the above models. However, this model is more sensitive (0.9781) as compared to logistic regression (0.6171). When using the Gaussian Fit for the SVM, it also has a lower accuracy at (0.9113), but overall a higher sensitivity at 0.9743.

As this is a marginalized model, it is only good as a model to further enhance and aid the current models that we currently have since it does not take into account single factors, but the factors overall. It will be good to use SVM over a cluster of factors against the decision. Hence in this study, it may not be the best fit model for the given dataset, but it gives an idea of the overall relationship between the factors and the result. SVM can be considered a model for predicting whether a client will sign onto a long-term deposit when used together with other models that specify the factor.

6 Session Information (R-Code)

```
knitr::opts_chunk$set(echo=FALSE,message=FALSE,warning=FALSE)
```

```
set.seed(1)
library(ggplot2)
library(knitr)
library(magrittr)
library(dplyr)
library(tidyverse)
library(ggcorrplot)
library(rpart)
library(rpart.plot)
library(ISLR)
library(caret)
library(gbm)
library(ROCR)
library(corrplot)
library(MASS)
library(caTools)
library(rsample)
library(class)
library(kknn)
library(AUC)
library(rmarkdown)
library(e1071)
library(xtable)
library(kableExtra)
library(devtools)
library(pander)
library(kernlab)
setwd("/Users/trucle/Desktop/STA\ 138/Bank/bank-additional")
# for this project we will be using the bank-additional-full.csv to do our analysis on.
bank_additional_full <- read.csv("bank-additional-full.csv", sep=";", stringsAsFactors = F, header = T)

# this allows for the data to be separated rather than being mushed together.
# to see how many rows have missing data
sum(!complete.cases(bank_additional_full))
sapply(bank_additional_full, function(x) sum(is.na(x)))
write.csv(bank_additional_full, "cleaned_bank_additional_full.csv")
# this write the clean data into a separate csv that we can use to work with
bank <- read.csv("cleaned_bank_additional_full.csv")[-1]
# -1 to get rid of them counting the # of rows


bank$age <- as.numeric(bank$age)
bank$duration <- as.numeric(bank$duration)
bank$campaign <- as.numeric(bank$campaign)
bank$pdays <- as.numeric(bank$pdays)
bank$previous <- as.numeric(bank$previous)
bank$emp.var.rate <- as.numeric(bank$emp.var.rate)
bank$cons.price.idx <- as.numeric(bank$cons.price.idx)
bank$cons.conf.idx <- as.numeric(bank$cons.conf.idx)
bank$nr.employed <- as.numeric(bank$nr.employed)

bank$job = fct_explicit_na(bank$job, "missing")
bank$marital = fct_explicit_na(bank$marital, "missing")
bank$education = fct_explicit_na(bank$education, "missing")
bank$default = fct_explicit_na(bank$default, "missing")
bank$loan = fct_explicit_na(bank$loan, "missing")
bank$contact = fct_explicit_na(bank$contact, "missing")
bank$poutcome = fct_explicit_na(bank$poutcome, "missing")
bank$day_of_week = fct_explicit_na(bank$day_of_week, "missing")
bank$housing = fct_explicit_na(bank$housing, "missing")
bank$month = fct_explicit_na(bank$month, "missing")
bank$y =ifelse(bank$y =='yes',1,0) # transforming 'yes' category into a binary 1=yes 0=no
str(bank)
kable(summary(bank), format = "html",col.names = colnames(bank)) %>%
  kable_styling(font_size = 10, full_width = FALSE)
kbl(prop.table(table(bank$y)), booktabs = TRUE )
pandoc.table(summary(bank$age), style="rmarkdown")
##### Category: Age
gg <- ggplot(bank)
# histogram plot for the distribution of age
his_age = gg + geom_histogram(aes(x=age), color = "black", fill ="white", binwidth = 1) +
  ggtitle("Age Distribution")+
  xlab("Age") + ylab("Counts")+
  geom_vline(aes(xintercept=mean(age), color="red"))+
  theme(legend.position = "none")

# made outliers data points red
mean_age <- bank_additional_full %>% group_by(y) %>% summarize(grp.mean=mean(age))
his_age_sub = ggplot (bank_additional_full, aes(x=age)) +
  geom_histogram(color = "black", fill = "green", binwidth = 1) +
  ggtitle('Subscription of Term Deposit by Age') + ylab('Count') + xlab('Age') +
```

```
  facet_grid(cols=vars(y))+
  scale_x_continuous(breaks = seq(0,100,5)) +
  geom_vline(data=mean_age, aes(xintercept=grp.mean), color="red", linetype="dashed") + geom_vline(data=bank_addi
tional_full, aes(xintercept=mean(age)), color="black")
# this output two histograms for comparison, left histogram is for Age Distribution and the right histogram is fo
r Subscription by Age. The black line shows the mean of the age superimposed onto the histograms and the dashed r
ed line shows the mean of the age that subscriped onto the term deposit.
his_age
his_age_sub
age_<-bank_additional_full %>%
  group_by(age) %>%
  summarize(age.cnt = n(), pct.con.yes = mean(y=="yes")*100) %>%
  arrange(desc(age.cnt)) %>%
  head()
kbl(age_, booktabs = TRUE, caption = "Age Percentage to Subscribe",format = "html")%>%
  kable_styling(full_width = TRUE, position = "left",font_size = 10)
# Category: Education
#Bar graph for education
gg_edu=ggplot(bank, aes(x = education, fill=education))+geom_bar()+ggtitle("Distribution of Education Levels")
# Subscription by Education
edu_sub = ggplot(data = bank_additional_full, aes(x=education, fill=y)) +
         geom_bar() +
         ggtitle("Term Deposit Subscription based on Education Level") +
         xlab(" Education Level") +
         guides(fill=guide_legend(title="Subscription of Term Deposit"))
# need to use the bank_additional_full dataset to get the two layers bar graphs.

edu_<-bank_additional_full %>%
  group_by(education) %>%
  summarize(edu.cnt = n(), pct.con.yes = mean(y=="yes")*100) %>%
  arrange(desc(edu.cnt)) %>%
  head()
# Percentage of Yes within each group based on their education level
t2<-kbl(edu_, booktabs = TRUE, caption = "Education Percentage to Subscribe", format="html")


gg_edu
edu_sub
t2 %>%
  kable_styling(full_width = TRUE, font_size = 10)

# Category: Job
# bar graph for job
gg_job = ggplot(bank, aes(x = job, fill=job))+geom_bar()+ggtitle("Distribution of Jobs")
# subscription by job title
job_sub = ggplot(data = bank_additional_full, aes(x=job, fill=y)) +
         geom_bar() +
         ggtitle("Term Deposit Subscription based on Job Position") +
         xlab(" Job Position") +
         guides(fill=guide_legend(title="Subscription of Term Deposit"))
gg_job
job_sub
# Percentage of Yes in each group based on their job title
job_<-bank_additional_full %>%
  group_by(job) %>%
  summarize(job.cnt = n(), pct.con.yes = mean(y=="yes")*100) %>%
  arrange(desc(job.cnt)) %>%
  head()

kbl(job_, booktabs = TRUE, caption = "Job Percentage to Subscribe")%>%
  kable_styling(full_width = TRUE, font_size = 10)
# Category: Marital Statues
# bar chart for marital status
gg_marital = ggplot(bank, aes(x = marital, fill=marital)) + geom_bar() +
  ggtitle("Distribution of Martial Status") + xlab("Martial Status")
# Subscription by marital status
marital_sub = ggplot(bank_additional_full, aes(x = marital, fill=y)) + geom_bar() +
  ggtitle("Distribution of Martial Status by subscription") + xlab("Martial Status")

mar_<-bank_additional_full %>%
  group_by(marital) %>%
  summarize(marital.cnt = n(), pct.con.yes = mean(y=="yes")*100) %>%
  arrange(desc(marital.cnt)) %>%
  head()

kbl(mar_, booktabs = TRUE, caption = "marital Percentage to Subscribe")%>%
  kable_styling(full_width = TRUE, font_size = 10)
# Percentage of Yes in each group based on their Marital status
gg_marital
marital_sub
# Consumer Price Index Distribution Graphs
```

```
# Using GGplot, I constructed a bar graph to show the distribution of the consumer price index
bar_price.idx=ggplot(bank, aes(x = cons.price.idx, fill=cons.price.idx)) + geom_bar()+
  geom_vline(aes(xintercept=mean(cons.price.idx)),color="red")+ggtitle("Consumer Price Index with Red Mean Line")

bar_cpi_sub = ggplot(bank_additional_full, aes(x = cons.price.idx, fill=y)) + geom_bar() +
  ggtitle("Term Deposit of Subscription by Consumer Price Index") + xlab("Consumer Price Index")

cpi_<-bank_additional_full %>%
  group_by(cons.price.idx) %>%
  summarize(cpi.cnt = n(), pct.con.yes = mean(y=="yes")*100) %>%
  arrange(desc(cpi.cnt)) %>%
  head()

kbl(cpi_, booktabs = TRUE, caption = "Consumer Price Index Percentage to Subscribe")%>%
  kable_styling(full_width = TRUE, font_size = 10)
bar_price.idx
bar_cpi_sub
# Category: Consumer Confidence Index Graphs
bar_conf.idx=ggplot(bank, aes(x = cons.conf.idx, fill=cons.conf.idx)) + geom_bar()+
  ggtitle("Consumer Confidence Index Bar Graph")
bar_cci_sub = ggplot(bank_additional_full, aes(x = cons.conf.idx, fill=y)) + geom_bar() +
  ggtitle("Term Deposit of Subscription by Consumer Confidence Index") + xlab("Consumer Confidence Index")

cci_<-bank_additional_full %>%
  group_by(cons.conf.idx) %>%
  summarize(cci.cnt = n(), pct.con.yes = mean(y=="yes")*100) %>%
  arrange(desc(cci.cnt)) %>%
  head()

kbl(cci_, booktabs = TRUE, caption = "Consumer Confidence Index Percentage to Subscribe")%>%
  kable_styling(full_width = TRUE, font_size = 10 )

bar_conf.idx
bar_cci_sub

# Category: Campaign
bar_campaign=ggplot(bank, aes(x = campaign, fill=campaign)) + geom_bar()+
  ggtitle("Campaign")
bar_campaign_sub = ggplot(bank_additional_full, aes(x = campaign, fill=y)) + geom_bar() +
  ggtitle("Term Deposit of Subscription by Campaign") + xlab("Campaign")+xlim(c(min=0, max=30))

cam_<-bank_additional_full %>%
  group_by(campaign) %>%
  summarize(contact.cnt = n(), pct.con.yes = mean(y=="yes")*100) %>%
  arrange(desc(contact.cnt)) %>%
  head()
kbl(cam_, booktabs = TRUE, caption = "Campaign Percentage to Subscribe")%>%
  kable_styling(full_width = TRUE, font_size = 10)
bar_campaign
bar_campaign_sub
# Category : Duration
bar_duration=ggplot(bank, aes(x = duration, fill=duration)) + geom_bar(fill="blue")+
  ggtitle("Duration Distribution")
mean_duration <- bank_additional_full %>% group_by(y) %>% summarize(grp.mean=mean(duration))
his_dur_sub = ggplot(bank_additional_full, aes(x = duration, fill=y)) + geom_histogram(binwidth = 2) +
  facet_grid(cols=vars(y))+
  ggtitle("Term Deposit of Subscription by Duration") + xlab("Duration")+coord_cartesian(xlim = c(0,2000), ylim =
c(0,1000))+geom_vline(data=mean_duration, aes(xintercept = grp.mean), color="red")

dur_<-bank_additional_full %>%
  group_by(duration) %>%
  summarize(contact.cnt = n(), pct.con.yes = mean(y=="yes")*100) %>%
  arrange(desc(contact.cnt)) %>%
  head()
kbl(dur_, booktabs = TRUE, caption = "Duration Percentage to Subscribe")%>%
  kable_styling(full_width = TRUE, font_size = 10)
bar_duration
his_dur_sub
# Employment Variation Rate
his_evr = ggplot(bank, aes(x = emp.var.rate, fill=emp.var.rate)) + geom_histogram(fill="blue")+
  ggtitle("Employment Variation Rate")
mean_evr <- bank_additional_full %>% group_by(y) %>% summarize(grp.mean=mean(emp.var.rate))
his_evr_sub = ggplot(bank_additional_full, aes(x = emp.var.rate, fill=y)) + geom_histogram(binwidth = 2) +facet_g
rid(cols=vars(y))+ggtitle("Term Deposit of Subscription by Employment Variation Rate") + xlab("Employment Variati
on Rate")+geom_vline(data=mean_evr, aes(xintercept = grp.mean), color="red")

evr_<-bank_additional_full %>%
  group_by(emp.var.rate) %>%
  summarize(contact.cnt = n(), pct.con.yes = mean(y=="yes")*100) %>%
  arrange(desc(contact.cnt)) %>%
  head()
```

```
kbl(evr_, booktabs = TRUE, caption = "Employment Variation Rate Percentage to Subscribe")%>%
  kable_styling(full_width = TRUE, font_size = 10)
his_evr
his_evr_sub
#Split the Training / Testing data
set.seed(1)
split = sample.split(bank$y,SplitRatio = 0.70)
train = subset(bank, split == TRUE)
test = subset(bank, split == FALSE)
# Model 1: Decision Tree with training set
tree_model <-rpart(y ~ .,
                   data=train,
                   method ="class")
rpart.plot(tree_model,
           main="Decision Tree",
           type=2,
           extra = 104,
           fallen.leaves = TRUE)
# type= 2 : draw the split labels below the node labels
# extra = 104 : class model with a response having more than two levels
#Evaluating the Decision Tree Model
#prediction model
pred<-predict(tree_model,test,type = "class")
#confusion matrix
confusionMatrix(as.factor(test$y),as.factor(pred))
# calculate ROC curve
pred.DT = predict(tree_model, newdata = test, type = 'prob')
rocr.pred = prediction(predictions=pred.DT[,2], labels = test$y)
rocr.perf = performance(rocr.pred, measure = "tpr", x.measure = "fpr")
rocr.auc = as.numeric(performance(rocr.pred, "auc")@y.values)
# print ROC AUC
invisible(rocr.auc)
#plot ROC curve
roc_curve<-{plot(rocr.perf,
     lwd = 3, colorize = TRUE,
     print.cutoffs.at = seq(0, 1, by = 0.1),
     text.adj = c(-0.2, 1.7),
     main = 'ROC Curve')
mtext(paste('Decision Tree - auc : ', round(rocr.auc, 5)))
abline(0, 1, col = "red", lty = 1)}
roc_curve
# Model 2: General Logistic Regression
invisible(table(train$y))
model_glm <- glm(formula = y ~ age+ job + marital + education + default + housing +
    loan + contact + month + duration+ day_of_week + campaign + pdays +
    previous + poutcome + emp.var.rate + cons.price.idx + cons.conf.idx +
    euribor3m + nr.employed, family = "binomial", data = train)
options(max.print = 50)
summary(model_glm)
bank_glm_pred <- predict(model_glm, test, type = "response")
bank_glm_pred_label <- as.factor(ifelse(bank_glm_pred>.3, "1", "0"))
confusionMatrix(factor(bank_glm_pred_label), factor(test$y), positive = "1")
bank_glm_roc <- prediction(bank_glm_pred, test$y)
rocr.perf_glm=performance(bank_glm_roc, "tpr", "fpr")
bank_glm_auc <- performance(bank_glm_roc, "auc")
auc_glm<-bank_glm_auc@y.values[[1]]
# print ROC AUC
invisible(auc_glm)
#plot ROC curve
roc_curve<-{plot(rocr.perf_glm,
     lwd = 3, colorize = TRUE,
     print.cutoffs.at = seq(0, 1, by = 0.1),
     text.adj = c(-0.2, 1.7),
     main = 'ROC Curve')
mtext(paste('General Logistic Regression - auc : ', round(auc_glm, 5)))
abline(0, 1, col = "red", lty = 1)}
roc_curve
# Model 3: K-Nearest Neighbor
# have to make the y into a factor in order to run the knn code
set.seed(1)
split = sample.split(bank$y,SplitRatio = 0.70)
train = subset(bank, split == TRUE)
test = subset(bank, split == FALSE)
train$y <- as.factor(train$y)
test$y <- as.factor(test$y)
options(max.print = 20)
model.KNN <- kknn( y ~ age+ job + marital + education + default + housing +
    loan + contact + month + duration+ day_of_week + campaign + pdays +
    previous + poutcome + emp.var.rate + cons.price.idx + cons.conf.idx +
    euribor3m + nr.employed, train, test, k=5, distance = 2,scale=FALSE, getOption(max.print=20))
options(max.print = 50)
```

```
summary(model.KNN)
bank_knn_pred <-NULL
bank_knn_pred <- predict(model.KNN, test, type="raw")
bank_knn_pred_label <- table(bank_knn_pred, test$y)
confusionMatrix(bank_knn_pred_label, positive = "1")
pb_bank <- NULL
pb_bank <- predict(model.KNN, test, type="prob")
pb_bank <- as.data.frame(pb_bank)
pred.KNN <- data.frame(test$y, pb_bank$"1")
labels <- as.factor(ifelse(pred.KNN$test.y=="1", 1, 0))
predictions <- pred.KNN$pb_bank..1.
auc_knn<-auc(roc(predictions, labels), min = 0, max = 1)

#plot ROC curve
{plot(roc(predictions, labels), col="blue",
      lwd=3,
      main= "ROC Curve")
  mtext(paste('K-Nearest Neighbors - auc : ', round(auc_knn, 5)))
  abline(0, 1, col = "red", lty = 1)}
```