Portuguese Bank Institution
Marketing

(Using k-NN)

Mayur Desai

Introduction

This project applies a machine learning technique that goes beyond standard linear regression. I had the prospect to use a publicly available dataset to resolve the trouble of my alternative. I sifted through the datasets available on Kaggle and chose a finance/bank related dataset.

The project aims to answer the following question::

What type of management do potential consumers show that result in them additional possible to subscribe to a term deposit?

The industry difficulty is to devise an objective marketing strategy for the bank base on the behavioural data collected. The dataset is included in one of the obedience files and can be downloaded from Kaggle https://www.kaggle.com/henriqueyamahata/bank-marketing).

The Dataset: It contains 41,188 customer data on a direct advertising campaign (phone calls) of a Portuguese banking organization.

It has the following variables:

Client: age, job, default status, housing, marital, education, and loan

Campaign: last contact day of the week, last contact type, last contact, last contact duration and month of year

Others: number of contacts performed in current campaign, number of days that passed by after the client was last contacted, number of contacts performed before this campaign, outcome of previous campaign, and whether a client has subscribed a term deposit

Key Steps Performed:

.

I first used Data Classification to examine the set related to direct marketing campaigns of a Portuguese banking institution. The objective of the classification is to predict if the client will subscribe to a Term Deposit. Data Classification is the use of machine learning techniques to organize datasets into related sub-populations, not previously specified in the dataset. This can uncover hidden characteristics within data, and identify hidden categories that new data belongs within. The rest of the key steps that were performed used the data science techniques of Exploratory Data Analysis, Data Classification with the help of K-Nearest Neighbor(k-NN).

1. Data Analysis

Exploratory Analysis

Loading the required packages:

```
rm(list = ls())
options(warn=-1)

if(!require(readr)) install.packages("readr", repos = "")
if(!require(tidyverse)) install.packages("tidyverse", repos = "")
if(!require(GGally)) install.packages("GGally", repos = "")
if(!require(glmnet)) install.packages("glmnet", repos = "")
if(!require(Matrix)) install.packages("Matrix", repos = "")
if(!require(DataExplorer)) install.packages("DataExplorer", repos = "")
if(!require(corrplot)) install.packages("corrplot", repos = "")
if(!require(caret)) install.packages("caret", repos = "")
if(!require(randomForest)) install.packages("randomForest", repos = "")
if(!require(class)) install.packages("class", repos = "")
if(!require(gmodels)) install.packages("gmodels", repos = "")
if(!require(dplyr)) install.packages("dplyr", repos = "")
if(!require(psych)) install.packages("psych", repos =  "")
library(readr)
library(tidyverse)
library(GGally)
library(glmnet)
library(Matrix)
library(ggplot2)
library(DataExplorer)
library(corrplot)
library(caret)
library(randomForest)
library(class)
library(gmodels)
library(dplyr)
library(psych)
set.seed(1)
```

Loading the dataset:

```
#setwd(" E:\mayur\harvard certificate\Bank-Marketing")

 #data.df <- read.csv("bank-additional-full.csv", header=TRUE, sep=";")

#data.df <- read.csv("https:// https://github.com/mayurdesai/Bank-Marketing  ",
header=TRUE)
```

Viewing the column names of the dataset:

```
names(data.df)

##  [1]  "age"             "job"            "marital"       "education"
##  [5] "default"          "housing"        "loan"          "contact"
##  [9] "month"            "day_of_week"    "duration"      "campaign"
## [13] "pdays"            "previous"       "poutcome"      "emp.var.rate"
## [17] "cons.price.idx" "cons.conf.idx"    "euribor3m"     "nr.employed"
## [21] "y"
```

Column details of the dataset:

```
str(data.df)

## 'data.frame':     41188 obs. of  21 variables:
##  $ age            : int  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       : int  261 149 226 151 307 198 139 217 380  50 ...
##  $ campaign       : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ pdays          : int  999 999 999 999 999 999 999 999 999  999 ...
##  $ previous       : int  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 -3
6.4 -36.4 ...
```

```
## $ euribor3m : num 4.86 4.86 4.86 4.86 4.86 ...
## $ nr.employed : num 5191 5191 5191 5191 5191 ...
## $ y : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...
```

Summary analysis of the dataset:

```
summary(data.df)

## age job marital
## Min. :17.00 admin. :10422 divorced: 4612
## 1st Qu.:32.00 blue-collar: 9254 married :24928
## Median :38.00 technician : 6743 single :11568
## Mean :40.02 services : 3969 unknown : 80
## 3rd Qu.:47.00 management : 2924
## Max. :98.00 retired : 1720
## (Other) : 6156
## education default housing
## university.degree :12168 no :32588 no 18622
## high.school : 9515 unknown: 8597 unknown: 990
## basic.9y : 6045 yes : 3 yes 21576
## professional.course: 5243
## basic.4y : 4176
## basic.6y : 2292
## (Other) : 1749
## loan contact month day_of_week
## no :33950 cellular :26144 may :13769 fri:7827
## unknown: 990 telephone:15044 jul : 7174 mon:8514
## yes : 6248 aug : 6178 thu:8623
## jun : 5318 tue:8090
## nov : 4101 wed:8134
## apr : 2632
## (Other): 2016
## duration campaign pdays previous
## Min. : 0.0 Min. : 1.000 Min.: 0.0Min. :0.000
## 1st Qu.: 102.0 1st Qu.: 1.000 1st Qu.:999.0 1st Qu.:0.000
## Median : 180.0 Median : 2.000 Median :999.0 Median :0.000
## Mean : 258.3 Mean : 2.568 Mean :962.5 Mean :0.173
## 3rd Qu.: 319.0 3rd Qu.: 3.000 3rd Qu.:999.0 3rd Qu.:0.000
## Max. :4918.0 Max. :56.000 Max. :999.0 Max. :7.000
##
## poutcome emp.var.rate cons.price.idx cons.conf.idx
## failure : 4252 Min. :-3.40000 Min. :92.20 Min. :-50.8
## nonexistent:35563 1st Qu.:-1.80000 1st Qu.:93.08 1st Qu.:-42.7
## success : 1373 Median : 1.10000 Median :93.75 Median :-41.8
## Mean : 0.08189 Mean :93.58 Mean :-40.5
## 3rd Qu.: 1.40000 3rd Qu.:93.99 3rd Qu.:-36.4
## Max. : 1.40000 Max. :94.77 Max. :-26.9
##
## euribor3m nr.employed y
## Min. :0.634 Min. :4964 no :36548
```

```
##   1st Qu.:1.344     1st  Qu.:5099     yes:  4640
##   Median :4.857     Median  :5191
##   Mean    :3.621     Mean     :5167
##   3rd Qu.:4.961     3rd  Qu.:5228
##   Max.    :5.045     Max.     :5228
##
```

Data Preparation

We check if any missing values exist:

```
sum(is.na(data.df))

## [1] 0
```

There are no missing values in our dataset. In the above investigative analysis, we experimental that there are many variables with class=int; hence, we change them keen on numeric class

Converting quantitative values to numeric class:

```
data.df$age <- as.numeric(data.df$age)
data.df$duration <- as.numeric(data.df$duration)
data.df$campaign <- as.numeric(data.df$campaign)
data.df$pdays <- as.numeric(data.df$pdays)
data.df$previous <- as.numeric(data.df$previous)
```

Ordering the levels of month:

```
data.df$month<- factor(data.df$month, ordered = TRUE, levels = c("mar", "apr"
, "may", "jun", "jul", "aug", "sep", "oct", "nov", "dec"))
```

Since the target variable is a categorical variables with 2 possible values: yes, no; we transform it into a numerical denotation: 1,0 respectively.

Transforming the target variable as Yes=1 and No=0:

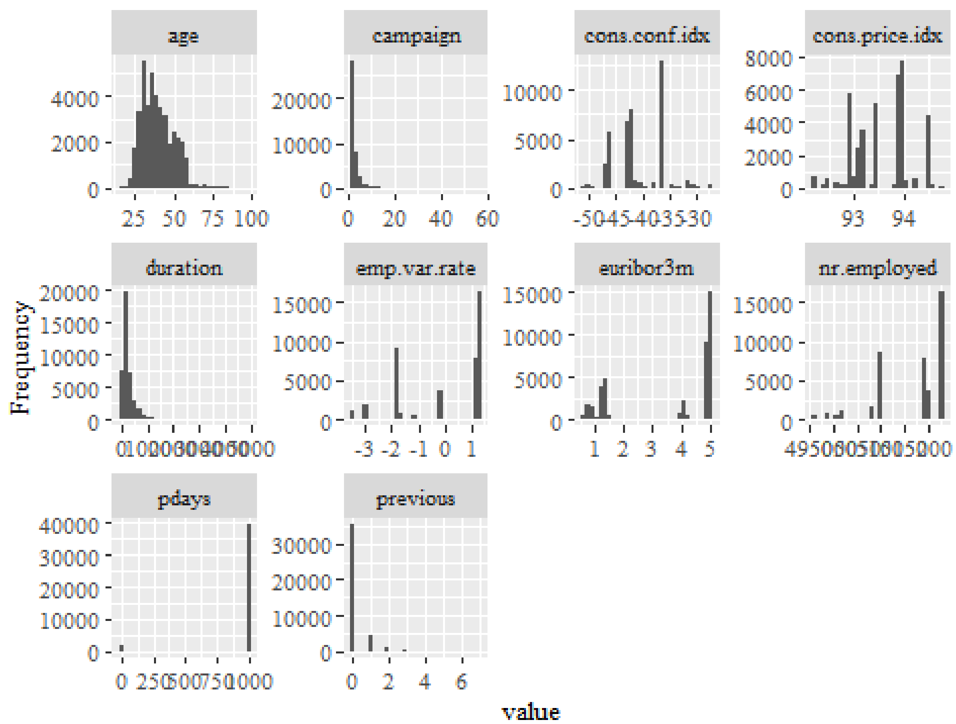
```
table(data.df$y)

##
##      no     yes
## 36548    4640

data.df <- data.df %>%
  mutate(y = ifelse(y=="yes", 1, 0))

data.df$y <- as.factor(data.df$y)
table(data.df$y)
```

```
##
##      0     1
## 36548  4640
```

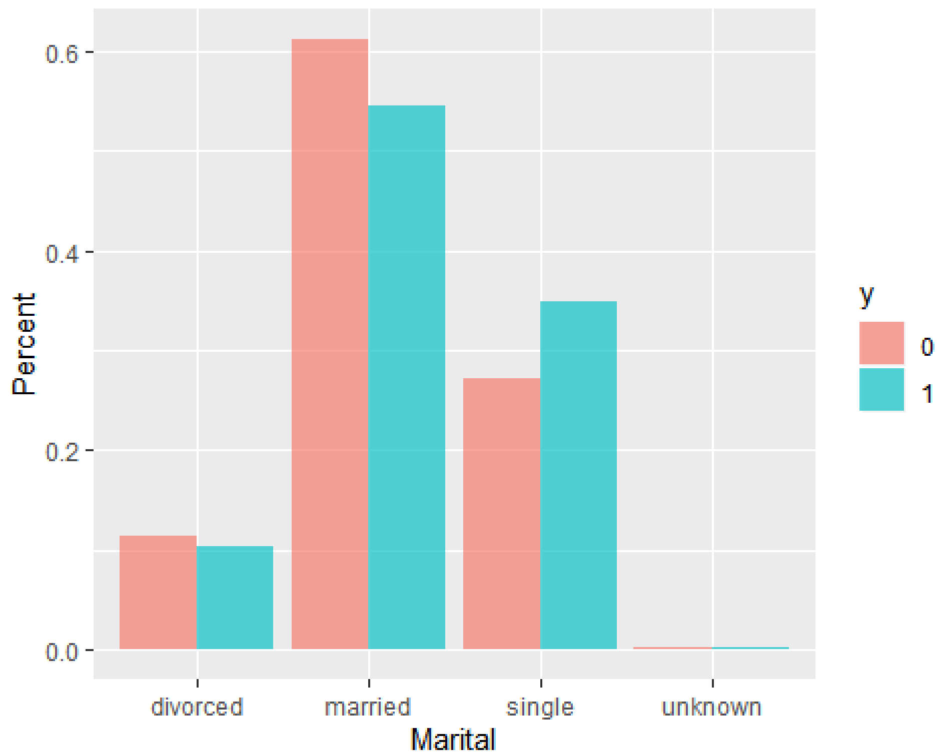
Descriptive Analysis

Let us look at the histogram of the input variables:

```
plot_histogram(data.df[,-21],ggtheme = theme_gray(base_size = 10, base_family
= "serif"))
```

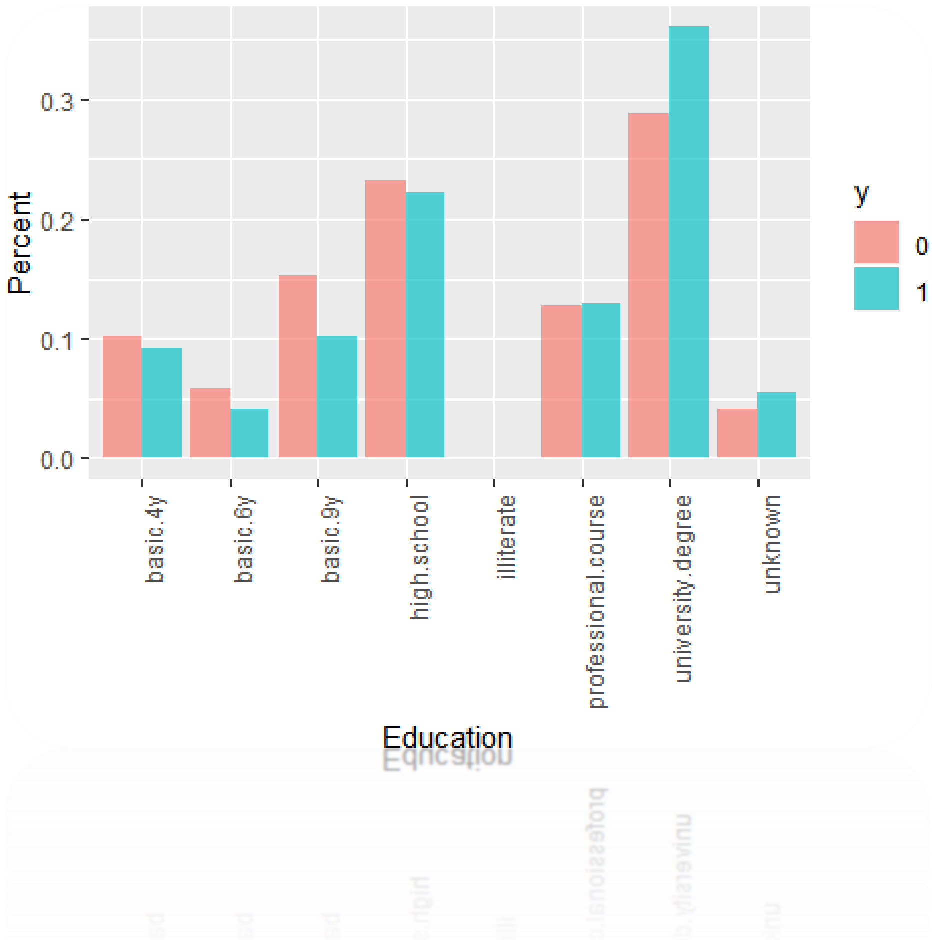



```
mytable <- table(data.df$marital, data.df$y)
tab <- as.data.frame(prop.table(mytable, 2))
colnames(tab) <-  c("marital", "y", "perc")
ggplot(data = tab, aes(x = marital, y = perc, fill = y)) +
  geom_bar(stat = 'identity', position = 'dodge', alpha = 2/3) +
  xlab("Marital")+ylab("Percent")
```



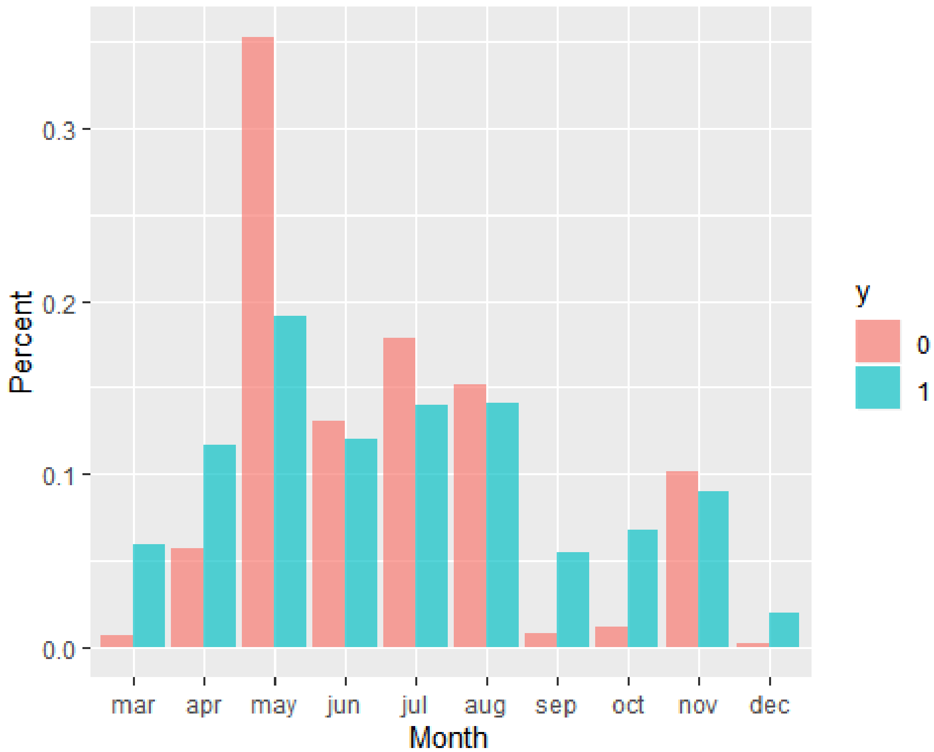
With deference to Marital Status, there is not a significant experiential difference in the number of people subscribe to term deposit and people without term deposits.

```
mytable <- table(data.df$education, data.df$y)
tab <- as.data.frame(prop.table(mytable, 2))
colnames(tab) <-  c("education", "y", "perc")
ggplot(data = tab, aes(x = education, y = perc, fill = y)) +
  geom_bar(stat = 'identity', position = 'dodge', alpha = 2/3) +
theme(axis.text.x=element_text(angle=90,hjust=1)) +
  xlab("Education")+ylab("Percent")
```



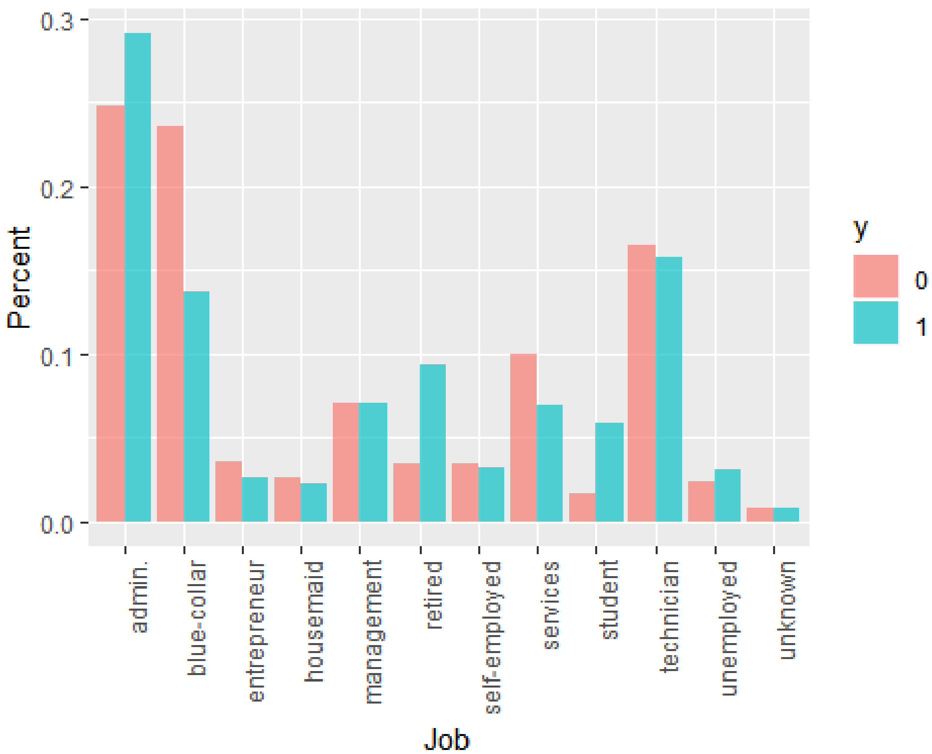
We can recognise that customers who sign up for bank deposits, proportionally, have reached a higher level of education, than those who didn't sign up.

```
mytable <- table(data.df$month, data.df$y)
tab <- as.data.frame(prop.table(mytable, 2))
colnames(tab) <-  c("month", "y", "perc")
ggplot(data = tab, aes(x = month, y = perc, fill = y)) +
  geom_bar(stat = 'identity', position = 'dodge', alpha = 2/3) +
  xlab("Month")+ylab("Percent")
```



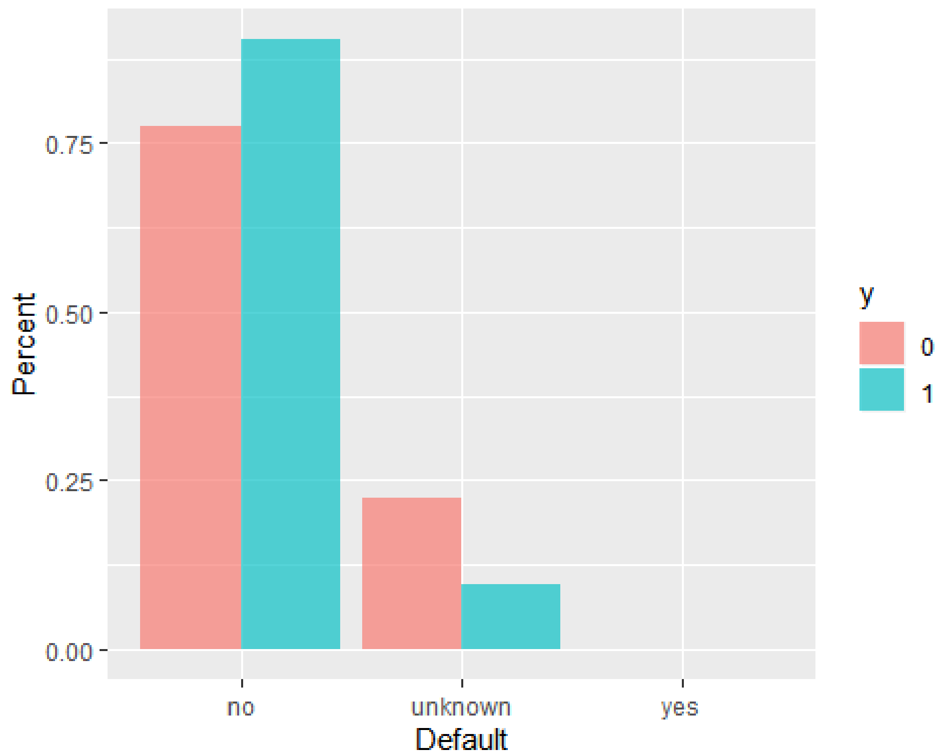
The month of May is when the highest quantity of calls was located for marketing deposits. The periods of April, September, October, and December is the time when a higher proportion of populace subscribed for expression deposits.

```
mytable <- table(data.df$job,  data.df$y)
tab <- as.data.frame(prop.table(mytable, 2))
colnames(tab) <-   c("job", "y", "perc")
ggplot(data = tab, aes(x = job, y = perc, fill  = y)) + geom_bar(stat
  = 'identity', position = 'dodge', alpha = 2/3) +
theme(axis.text.x=element_text(angle=90,hjust=1)) +
  xlab("Job")+ylab("Percent")
```



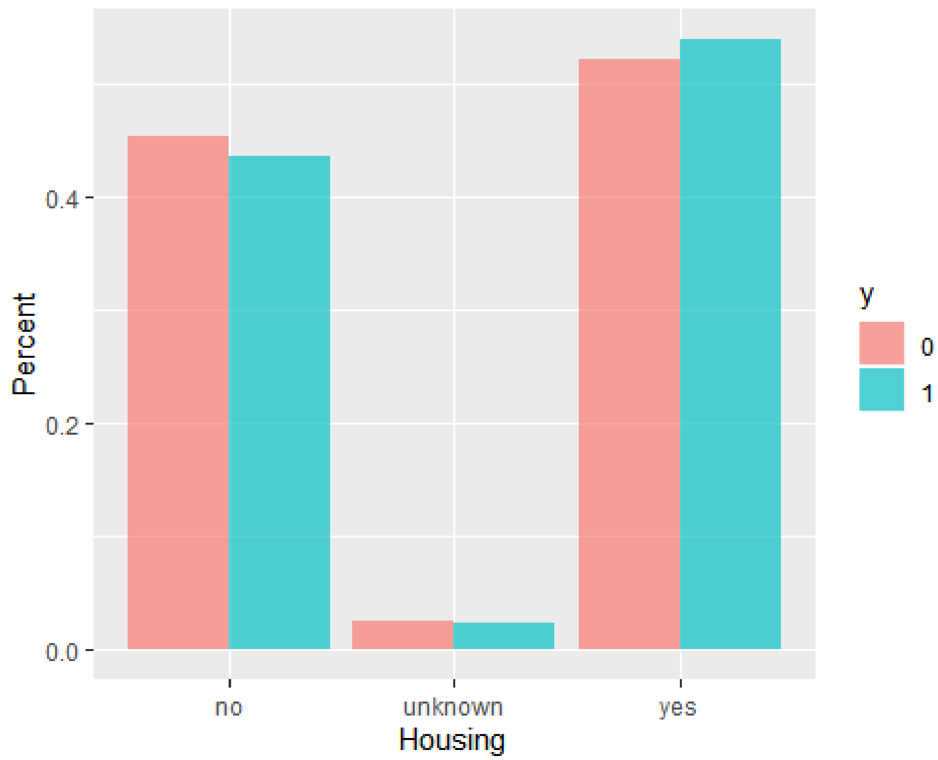
We understand there is a larger size for customers sign up for the term deposits which have the job of admin, retired, and students.

```
mytable <- table(data.df$default, data.df$y)
tab <- as.data.frame(prop.table(mytable, 2))
colnames(tab) <-  c("default", "y", "perc")
ggplot(data = tab, aes(x = default, y = perc, fill = y)) +
  geom_bar(stat = 'identity', position = 'dodge', alpha = 2/3) +
  xlab("Default")+ylab("Percent")
```



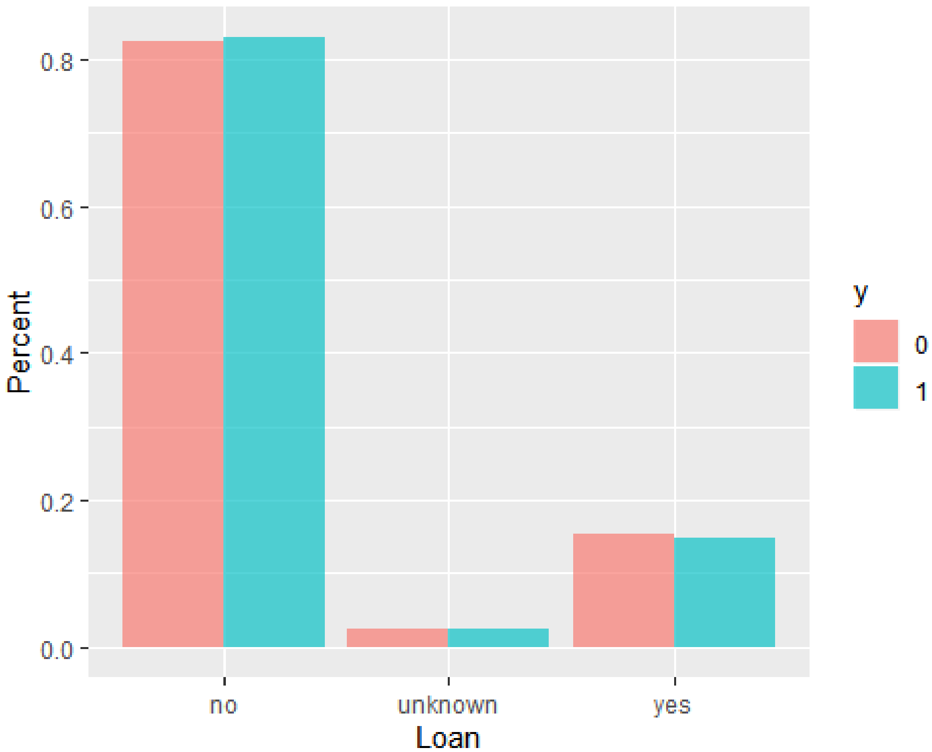
The data explain that people who aren't in want are a higher proportion of people who have contributed to bank deposits.

```
mytable <- table(data.df$housing, data.df$y)
tab <- as.data.frame(prop.table(mytable, 2))
colnames(tab) <-  c("housing", "y", "perc")
ggplot(data = tab, aes(x = housing, y = perc, fill = y)) +
  geom_bar(stat = 'identity', position = 'dodge', alpha = 2/3) +
  xlab("Housing")+ylab("Percent")
```



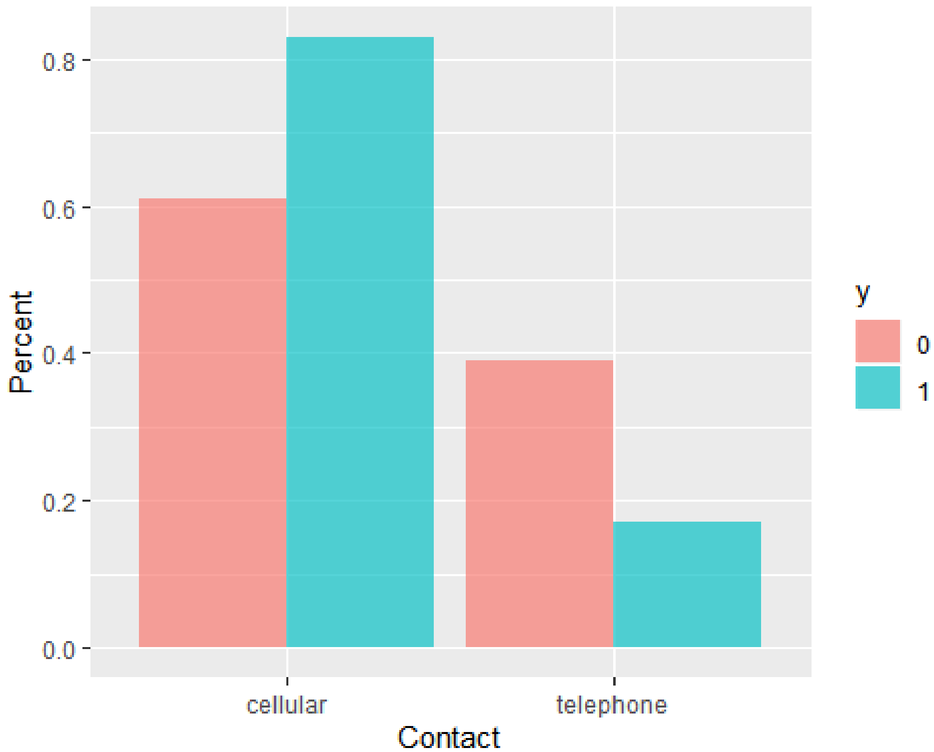
We notice that a large number of people who have contributed for bank deposit are homeowner versus ones that don't hold their own houses.

```
mytable <- table(data.df$loan, data.df$y)
tab <- as.data.frame(prop.table(mytable, 2))
colnames(tab) <-  c("loan", "y", "perc")
ggplot(data = tab, aes(x = loan, y = perc, fill = y)) +
  geom_bar(stat = 'identity', position = 'dodge', alpha = 2/3) +
  xlab("Loan")+ylab("Percent")
```



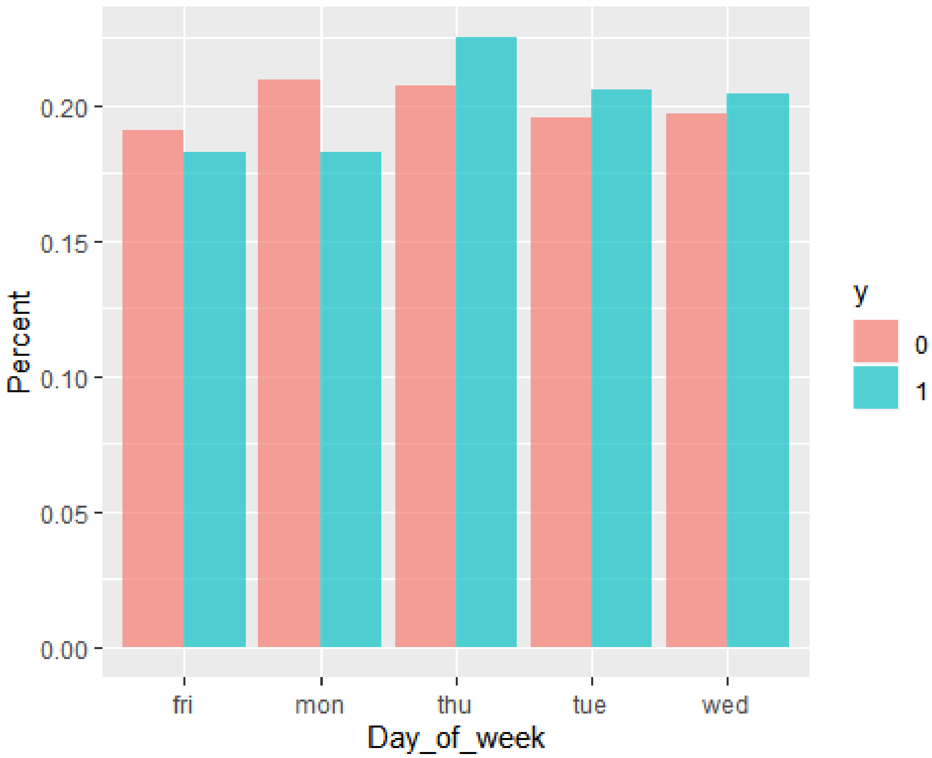
We see the share of people who have to support and not contribute to a term deposit is identical for categories of the Loan..

```
mytable <- table(data.df$contact, data.df$y)
tab <- as.data.frame(prop.table(mytable, 2))
colnames(tab) <-  c("contact", "y", "perc")
ggplot(data = tab, aes(x = contact, y = perc, fill = y)) +
  geom_bar(stat = 'identity', position = 'dodge', alpha = 2/3) +
  xlab("Contact")+ylab("Percent")
```



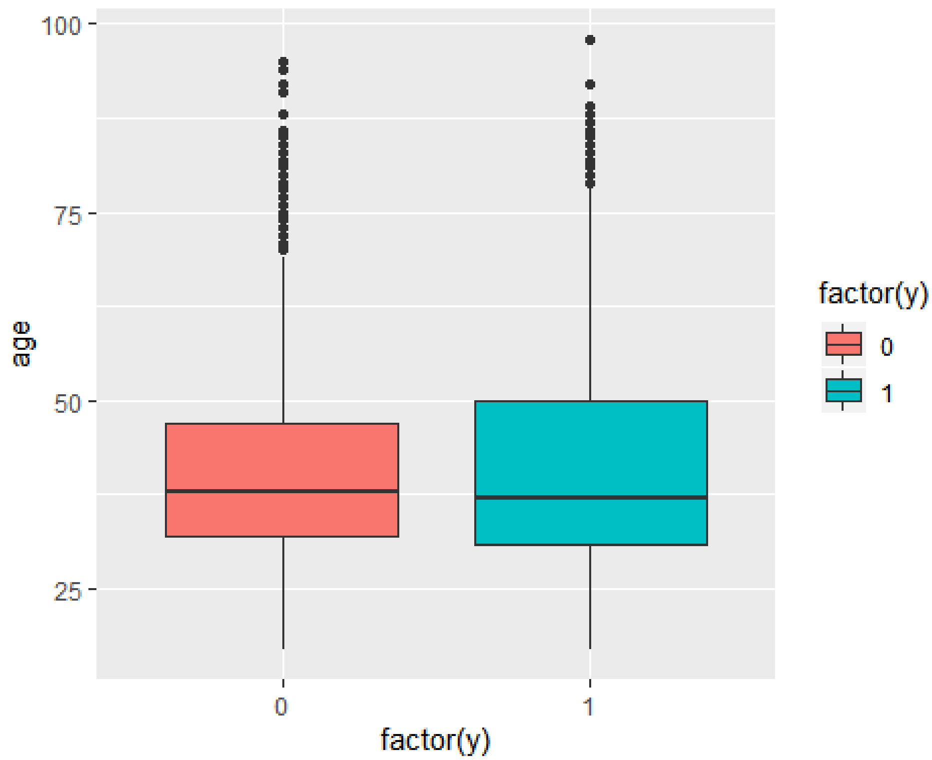
Customers who own cell phones, and hence a more immediate way of communicating, signed up for period deposits longer than those who just had a landline telephone..

```
mytable <- table(data.df$day_of_week,data.df$y)
tab <- as.data.frame(prop.table(mytable, 2))
colnames(tab) <-  c("day_of_week", "y", "perc")
ggplot(data = tab, aes(x = day_of_week, y = perc, fill = y)) +
  geom_bar(stat = 'identity', position = 'dodge', alpha = 2/3) +
  xlab("Day_of_week")+ylab("Percent")
```



A campaign that was shown midweek, on Tuesdays, Wednesdays, and Thursdays had a slightly higher section of people who subscribe for bank deposit.

```
mytable <- table(data.df$poutcome, data.df$y)
tab <- as.data.frame(prop.table(mytable, 2))
colnames(tab) <-  c("poutcome", "y", "perc")
ggplot(data = tab, aes(x = poutcome, y = perc, fill = y)) +
  geom_bar(stat = 'identity', position = 'dodge', alpha = 2/3) +
  xlab("Outcome of previous marketing  campaign")+ylab("Percent")
```



The age difference for vigorous exchange has a somewhat lower median, but higher quartile ranges.

```
df_cor <- select_if(data.df, is.numeric) %>% cor()
corrplot(df_cor, method = "number")
```

2. Results ()

Data Preparation

Includes Data Modeling and performance missing values for the period were clean out because if duration=0 then y="no" (no call was made). Thus, it doesn't make logic to have a 0-second duration. I also filtered out education uneducated, and default yes since they only have one surveillance each. We can't predict these circumstances if they happen to be in the examination data but not the train data.

```
data.df <- data.df %>%
  filter(duration != 0, education != "illiterate", default != "yes") %>%
  mutate(y = ifelse(y==1, 1, 0))
```

Split the data into training and test datasets:

```
set.seed(123)
trainIndex <- createDataPartition(data.df$y,
                                  p = 0.8, # training contains 80% of data
                                  list = FALSE)
dfTrain <- data.df[ trainIndex,]
dfTest  <-  data.df[-trainIndex,]

dim(dfTrain)

## [1] 32931      21

dim(dfTest)

## [1] 8232    21
```

The code and output over the note that the train Data dataset has 8929 rows and 17 columns and the examination Data dataset gas 2233 rows and 17 columns. The number of columns prevails the equivalent because the dataset was split up and down.

Data Modeling using KNN

We will make a copy of our data set so that we can prepare it for our K-NN classification.

```
data_knn <- data.df

str(data_knn)

## 'data.frame': 41163 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 : Ord.factor w/ 10 levels "mar"<"apr"<"may"<..: 3 3 3 3 3 3 3 3 3 3 ...
## $ 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 ...
```

Because the K-NN algorithm involves influential distances between data points, we have to use numeric variables only. This is relevant only to independent variables. The objective variable fork-NN categorization should remain a factor changeable. First, we scale the data just in case our features are on different metrics. For example, if we had "duration" as a variable, it would be on a much better scale than "age", which might be problematic given the k-NN relies on distance. Note that we are using the 'scale' function here, which income we are scaling to a z- score metric.

We see that the variables "age", "duration", "campaign", "pdays", "previous", "emp.var.rate", "cons.price.idx", "cons.conf.idx", "euribor3m" and "nr.employed" are interger variables, which means they can be scaled.

```
data_knn[, c("age", "duration", "campaign", "pdays", "previous", "emp.var.rate", "cons.price.idx", "cons.conf.idx", "euribor3m","nr.employed")] <- scale(data_knn[, c("age", "duration", "campaign", "pdays", "previous", "emp.var.rate", "cons.price.idx", "cons.conf.idx", "euribor3m","nr.employed")])

head(data_knn)

##            age       job marital   education default housing loan
## 1  1.533684728 housemaid married    basic.4y      no      no   no
## 2  1.629657912  services married high.school unknown      no   no
## 3 -0.289805768  services married high.school      no     yes   no
## 4 -0.001886216    admin. married    basic.6y      no      no   no
## 5  1.533684728  services married high.school      no      no  yes
## 6  0.477979704  services married    basic.9y unknown      no   no
##     contact month day_of_week    duration   campaign     pdays   previous
## 1 telephone   may         mon  0.01036255 -0.5658418 0.1954061 -0.3494959
## 2 telephone   may         mon -0.42162181 -0.5658418 0.1954061 -0.3494959
## 3 telephone   may         mon -0.12463256 -0.5658418 0.1954061 -0.3494959
## 4 telephone   may         mon -0.41390781 -0.5658418 0.1954061 -0.3494959
## 5 telephone   may         mon  0.18778470 -0.5658418 0.1954061 -0.3494959
## 6 telephone   may         mon -0.23262865 -0.5658418 0.1954061 -0.3494959
##      poutcome emp.var.rate cons.price.idx cons.conf.idx euribor3m
## 1 nonexistent    0.6480509      0.7224735     0.8865513 0.7124339
## 2 nonexistent    0.6480509      0.7224735     0.8865513 0.7124339
## 3 nonexistent    0.6480509      0.7224735     0.8865513 0.7124339
## 4 nonexistent    0.6480509      0.7224735     0.8865513 0.7124339
## 5 nonexistent    0.6480509      0.7224735     0.8865513 0.7124339
## 6 nonexistent    0.6480509      0.7224735     0.8865513 0.7124339
##   nr.employed y
## 1   0.3317071 0
## 2   0.3317071 0
## 3   0.3317071 0
## 4   0.3317071 0
## 5   0.3317071 0
## 6   0.3317071 0

str(data_knn)

## 'data.frame':    41163 obs. of  21 variables:
##  $ age          : num  1.53368 1.62966 -0.28981 -0.00189 1.53368 ...
##  $ 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         : Ord.factor w/ 10 levels "mar"<"apr"<"may"<..: 3 3 3 3 3 3 3 3 3 3 ...
## $ day_of_week   : Factor w/ 5 levels "fri","mon","thu",..: 2 2 2 2 2 2 2 2 2 2 ...
## $ duration      : num  0.0104 -0.4216 -0.1246 -0.4139 0.1878 ...
## $ campaign      : num  -0.566 -0.566 -0.566 -0.566 -0.566 ...
## $ emp.var.rate  : num  0.648 0.648 0.648 0.648 0.648 ...
## $ cons.price.idx: num  0.722 0.722 0.722 0.722 0.722 ...
## $ cons.conf.idx : num  0.887 0.887 0.887 0.887 0.887 ...
## $ euribor3m     : num  0.712 0.712 0.712 0.712 0.712 ...
## $ nr.employed   : num  0.332 0.332 0.332 0.332 0.332 ...
## $ y             : num  0 0 0 0 0 0 0 0 0 0 ...
```

We can observe that the variables “job”, “marital”, “education”, “default”, “housing”, “loan”, “contact”, “month”, “day_of_week” and “poutcome” are thing variables that have two or more levels.

** Then dummy code variables that have two levels, but are not numeric. **

```
data_knn$contact <- dummy.code(data_knn$contact)
```

Next we dummy code variables that have three or more levels.

```
job <- as.data.frame(dummy.code(data_knn$job))
marital <- as.data.frame(dummy.code(data_knn$marital))
education <- as.data.frame(dummy.code(data_knn$education))
default <- as.data.frame(dummy.code(data_knn$default))
housing <- as.data.frame(dummy.code(data_knn$housing))
loan <- as.data.frame(dummy.code(data_knn$loan))
month <- as.data.frame(dummy.code(data_knn$month))
day_of_week <- as.data.frame(dummy.code(data_knn$day_of_week))
poutcome <- as.data.frame(dummy.code(data_knn$poutcome))
```

Rename “unknown” columns, so we don’t have duplicate columns later).

```
## $ pdays         : num  0.195 0.195 0.195 0.195 0.195 ...
## $ previous      : num  -0.349 -0.349 -0.349 -0.349 -0.349 ...
## $ poutcome      : Factor w/ 3 levels "failure","nonexistent",..: 2 2 2 2 2 2 2 2 2 2 ...
```

Combine new dummy variables with original data set.

```
## 'data.frame':   41163 obs. of  72 variables:
## $ age              : num  1.53368 1.62966 -0.28981 -0.00189 1.53368 ...
## $ 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          : num [1:41163, 1:2] 0 0 0 0 0 0 0 0 0 0 ...
##  ..- attr(*, "dimnames")=List of 2
##  .. ..$ : NULL
##  .. ..$ : chr  "cellular" "telephone"
## $ month            : Ord.factor w/ 10 levels "mar"<"apr"<"may"<..: 3 3
3 3 3 3 3 3 3 3 ...
## $ day_of_week      : Factor w/ 5 levels "fri","mon","thu",..: 2 2 2     2 2
2 2 2 2 2 ...
## $ duration         : num  0.0104 -0.4216 -0.1246 -0.4139 0.1878 ...
## $ campaign         : num  -0.566 -0.566 -0.566 -0.566 -0.566 ...
## $ pdays            : num  0.195 0.195 0.195 0.195 0.195 ...
## $ previous         : num  -0.349 -0.349 -0.349 -0.349 -0.349 ...
2 2 2 2 2 2 2 2  ...
## $ emp.var.rate     : num  0.648 0.648 0.648 0.648 0.648 ...
## $ cons.price.idx   : num  0.722 0.722 0.722 0.722 0.722 ...
## $ cons.conf.idx    : num  0.887 0.887 0.887 0.887 0.887 ...
## $ euribor3m        : num  0.712 0.712 0.712 0.712 0.712 ...
## $ nr.employed      : num  0.332 0.332 0.332 0.332 0.332 ...
## $ y                : num  0 0 0 0 0 0 0 0 0 0 ...
## $ admin.           : num  0 0 0 1 0 0 1 0 0 0 ...
## $ blue-collar      : num  0 0 0 0 0 0 0 1 0 0 ...
## $ entrepreneur     : num  0 0 0 0 0 0 0 0 0 0 ...
## $ housemaid        : num  1 0 0 0 0 0 0 0 0 0 ...
## $ management       : num  0 0 0 0 0 0 0 0 0 0 ...
## $ retired          : num  0 0 0 0 0 0 0 0 0 0 ...
## $ self-employed    : num  0 0 0 0 0 0 0 0 0 0 ...
## $ services         : num  0 1 1 0 1 1 0 0 0 1 ...
## $ student          : num  0 0 0 0 0 0 0 0 0 0 ...
## $ technician       : num  0 0 0 0 0 0 0 0 1 0 ...
## $ unemployed       : num  0 0 0 0 0 0 0 0 0 0 ...
## $ unknown_job      : num  0 0 0 0 0 0 0 0 0 0 ...
## $ divorced         : num  0 0 0 0 0 0 0 0 0 0 ...
## $ married          : num  1 1 1 1 1 1 1 1 0 0 ...
## $ single           : num  0 0 0 0 0 0 0 0 1 1 ...
## $ unknown_marital  : num  0 0 0 0 0 0 0 0 0 0 ...
```

```
##  $ basic.4y           : num  1 0 0 0 0 0 0 0 0 0 ...
##  $ basic.6y           : num  0 0 0 1 0 0 0 0 0 0 ...
##  $ basic.9y           : num  0 0 0 0 0 1 0 0 0 0 ...
##  $ high.school        : num  0 1 1 0 1 0 0 0 0 1 ...
##  $ illiterate         : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ professional.course: num  0 0 0 0 0 0 1 0 1 0 ...
##  $ university.degree  : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ unknown_education  : num  0 0 0 0 0 0 0 1 0 0 ...
##  $ no_default         : num  1 0 1 1 1 0 1 0 1 1 ...
##  $ unknown_default    : num  0 1 0 0 0 1 0 1 0 0 ...
##  $ yes_default        : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ no_housing         : num  1 1 0 1 1 1 1 1 0 0 ...
##  $ unknown_housing    : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ yes_housing        : num  0 0 1 0 0 0 0 0 1 1 ...
##  $ no_loan            : num  1 1 1 1 0 1 1 1 1 1 ...
##  $ unknown_loan       : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ yes_loan           : num  0 0 0 0 1 0 0 0 0 0 ...
##  $ mar                : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ apr                : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ may                : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ jun                : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ jul                : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ aug                : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ sep                : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ oct                : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ nov                : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ dec                : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ fri                : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ mon                : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ thu                : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ tue                : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ wed                : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ failure            : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ nonexistent        : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ success            : num  0 0 0 0 0 0 0 0 0 0 ...
```

Remove original variables that had to be dummy coded.

```
data_knn <- data_knn %>% select(-one_of(c("job", "marital", "education", "def
ault", "housing", "loan", "month", "day_of_week", "poutcome")))

head(data_knn)
```

```
##            age contact.cellular contact.telephone    duration  campaign
## 1  1.533684728                0                 1  0.01036255 -0.5658418
## 2  1.629657912                0                 1 -0.42162181 -0.5658418
## 3 -0.289805768                0                 1 -0.12463256 -0.5658418
## 4 -0.001886216                0                 1 -0.41390781 -0.5658418
## 5  1.533684728                0                 1  0.18778470 -0.5658418
## 6  0.477979704                0                 1 -0.23262865 -0.5658418
```

```
##        pdays   previous emp.var.rate cons.price.idx cons.conf.idx euribor3m
## 1 0.1954061 -0.3494959    0.6480509      0.7224735     0.8865513 0.7124339
## 2 0.1954061 -0.3494959    0.6480509      0.7224735     0.8865513 0.7124339
## 3 0.1954061 -0.3494959    0.6480509      0.7224735     0.8865513 0.7124339
## 4 0.1954061 -0.3494959    0.6480509      0.7224735     0.8865513 0.7124339
## 5 0.1954061 -0.3494959    0.6480509      0.7224735     0.8865513 0.7124339
## 6 0.1954061 -0.3494959    0.6480509      0.7224735     0.8865513 0.7124339
##   nr.employed y admin. blue-collar entrepreneur housemaid management
## 1   0.3317071 0      0           0            0         1          0
## 2   0.3317071 0      0           0            0         0          0
## 3   0.3317071 0      0           0            0         0          0
## 4   0.3317071 0      1           0            0         0          0
## 5   0.3317071 0      0           0            0         0          0
## 6   0.3317071 0      0           0            0         0          0
##   Retiredself-employed services student technician unemployed unknown_job
## 1       0             0        0       0          0          0           0
## 2       0             0        1       0          0          0           0
## 3       0             0        1       0          0          0           0
## 4       0             0        0       0          0          0           0
## 5       0             0        1       0          0          0           0
## 6       0             0        1       0          0          0           0
##   divorced married single unknown_marital basic.4y basic.6y basic.9y
## 1        0       1      0               0        1        0        0
## 2        0       1      0               0        0        0        0
## 3        0       1      0               0        0        0        0
## 4        0       1      0               0        0        1        0
## 5        0       1      0               0        0        0        0
## 6        0       1      0               0        0        0        1
##   high.school illiterate professional.course university.degree
## 1           0          0                   0                 0
## 2           1          0                   0                 0
## 3           1          0                   0                 0
## 4           0          0                   0                 0
## 5           1          0                   0                 0
## 6           0          0                   0                 0
##   unknown_education no_default unknown_default yes_default no_housing
## 1                 0          1               0           0          1
## 2                 0          0               1           0          1
## 3                 0          1               0           0          0
## 4                 0          1               0           0          1
## 5                 0          1               0           0          1
## 6                 0          0               1           0          1
##   unknown_housing yes_housing no_loan unknown_loan yes_loan mar apr may
## 1               0           0       1            0        0   0   0   1
## 2               0           0       1            0        0   0   0   1
## 3               0           1       1            0        0   0   0   1
## 4               0           0       1            0        0   0   0   1
## 5               0           0       0            0        1   0   0   1
## 6               0           0       1            0        0   0   0   1
##   jun jul aug sep oct nov dec fri mon thu tue wed failure nonexistent
```

```
## 1    0    0    0    0    0    0    0    0    1    0    0    0          0              1
## 2    0    0    0    0    0    0    0    0    1    0    0    0          0              1
## 3    0    0    0    0    0    0    0    0    1    0    0    0          0              1
## 4    0    0    0    0    0    0    0    0    1    0    0    0          0              1
## 5    0    0    0    0    0    0    0    0    1    0    0    0          0              1
## 6    0    0    0    0    0    0    0    0    1    0    0    0          0              1
##     success
## 1         0
## 2         0
## 3         0
## 4         0
## 5         0
## 6         0
```

We are now ready for k-NN classification. We split the data into training and test sets. We partition 80% of the data into the training set and the remaining 20% into the test set.

Splitting the dataset into Test and Train:

```
set.seed(1234) # set the seed to make the partition  reproducible

# 80% of the sample size
sample_size <- floor(0.8 * nrow(data_knn))


train_index <- sample(seq_len(nrow(data_knn)), size = sample_size)

# put outcome in its own object
knn_outcome <- data_knn %>% select(y)

# remove original variable from the data set
data_knn <- data_knn %>% select(-y)

# creating test and training sets that contain all of the predictors
knn_data_train <- data_knn[train_index,]
knn_data_test <- data_knn[-train_index,]

# Split outcome  variable into training and  test sets using the same  partition
as  above.
knn_outcome_train <- knn_outcome[train_index, ]
knn_outcome_test <- knn_outcome[-train_index, ]
```

Using ‘class’ package, we run k-NN categorization on our data. We have to make a decision on the number of neighbors (k).This is an iterative exercise as we need to keep changing the value of k to dtermine the optimum performance. In our case, we started with k=10 till k=20, and finally got an optimum performance at k=17.

```r
model_knn <- knn(train = knn_data_train, test = knn_data_test, cl = knn_outco
me_train, k=17)
```

Model evaluation:

```r
# put "knn_outcome_test" in a data frame
knn_outcome_test <- data.frame(knn_outcome_test)

# merge "model_knn" and "knn_outcome_test"
knn_comparison_df <- data.frame(model_knn, knn_outcome_test)

# specify column names for "knn_comparison_df"
names(knn_comparison_df) <- c("KNN_Predicted_y", "KNN_Observed_y")

knn_comparison_df$KNN_Predicted_y <- as.factor(knn_comparison_df$KNN_Predicte
d_y)
knn_comparison_df$KNN_Observed_y <- as.factor(knn_comparison_df$KNN_Observed_
y)

# inspect "knn_comparison_df"
head(knn_comparison_df)

##    KNN_Predicted_y  KNN_Observed_y
## 1                0               0
## 2                0               0
## 3                0               0
## 4                0               0
## 5                0               0
## 6                0               0
```

Next, we evaluate our predict values of deposit to our actual values. The confusion matrix gives an indication of how well our model predicted the actual values. The confusion matrix output also shows overall model statistics and statistics by class

```r
confusionMatrix(knn_comparison_df$KNN_Observed_y,knn_comparison_df$KNN_Predic
ted_y)

## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 7071  179
##          1  586  397
##
##                Accuracy : 0.9071
##                  95% CI : (0.9006, 0.9133)
##     No Information Rate : 0.93
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.4618
```

```
##
##   Mcnemar's Test P-Value : <2e-16
##
##              Sensitivity : 0.9235
##              Specificity : 0.6892
##           Pos Pred Value : 0.9753
##           Neg Pred Value : 0.4039
##               Prevalence : 0.9300
##           Detection Rate : 0.8589
##     Detection Prevalence : 0.8806
##        Balanced Accuracy : 0.8064
##
##         'Positive' Class : 0
##
```

The K-NN test data consisted of 8238 interpretation. Out of which 7128 cases have been accurately predicted (TN->True Negatives) as the negative class (0) which constitutes 87%. Also, 367 out of 8238 observations were correctly predicted (TP-> True Positives) as the positive class

1) which represents 4%. Thus a total of 367 out of 8238 predictions where TP, i.e., True Positive in nature.

There were 544 cases of False Positives (FP) meaning 544 cases out of 8238 were negative but got predict as positive.

There were 199 cases of False Negatives (FN) meaning 199 cases were positive but got predict as unfavourable.

Accuracy of the model is the correctly classified positive and negative cases divided by all the circumstances. The total efficiency of the model is 91.13%, which means the model prediction is very accurate.

3. Conclusion

Model Comparison

K Nearest Neighbor is generating high accuracy when trained with the bank marketing dataset. The parameter K-NN model is:

Parameter	K-NN Model
Accuracy	91.13%
Sensitivity	93.03%
Specificity	65.68%
Pos Pred Value	97.31%
Neg Pred Value	41.38%

Analysis Summary

The critical insight derived from the general analysis are:

- People who aren't in default are a higher amount of people who have subscribed for bank deposits.

-A higher amount of people who have contributed for bank deposit is homeowners versus people that don't own their own houses.

- To Marital Status, there is not a sizeable experiential variation in the relationship of people contributed to term deposits and people outdoors term deposits.

- The relationship of people who have supported and not contributed to a term deposit is very for categories of the Loan.

- Consumers who sign up for bank deposits, proportionally, have reached a higher level of education, than those who didn't sign up.

- The months of April, September, October, and December is the occasion when a higher relationship of people subscribed for term deposits.

- There are higher balances for consumers signing up for the term deposits which become the jobs of admin, retired, and students.

- Consumers who have cell phones, and therefore a more direct way of communicating, signed up for term deposits more than those who only had a landline telephone.

- The age limit for prosperous growth has a slightly lower average, but higher quartile ranges.

- Operations that were played midweek, on Tuesdays, Wednesdays, and Thursdays had a somewhat higher relationship of people who subscribed for bank deposit.

- Possible consumers who successfully attached and responded in previous campaigns had a higher balance of signing up for the term deposit.

- The more extended the telephone conversation, the higher the exchange valuation is for the potential customer to sign up for the term deposit.

Contributing to term deposit has a high positive association with the term and if the customer was involved and connected in a previous campaign. At the same time, there's a negative correlation with Nr.employed (number of employees), pdays (number of days from the last contact), Euribor3m (Euribor 3 month rate) and emp.var. Rate (employee variation rate)

Target Market Strategy

The business problem described in the introduction is to devise a target marketing policy for the bank based on the behavioural data obtained. We found the varieties of perceptions and behaviours of potential consumers that result in them also likely to subscribe to a term deposit.

Using the high penetrations, the bank should devise a purpose marketing plan that is customized towards possible customers with those who now have an existing account with the bank and have higher study. Those consumers who generally are either employed in admin related jobs, or are students, or retired are those the bank can further pull.

Those consumers who are readily available with a mobile number will be the ones to return the call; the key is to have a pleasant and personable communication with the consumer and install a relationship where they feel comfortable signing up for a term deposit with the bank. Those customers who were part of the unwarranted operations should be contacted by the bank again because we discussed following up and having a continued discussion and contact results in a higher number of those who sign up. Also, to improve the probability of success, the campaigns should be started in the last third of the calendar year when people are thinking of saving for the future and preparing for year-end taxes.

Future Work

Data analytics is usually used to contain and work with big data such as the one we presented in the project here. It promotes the cross-examination of the data and the purposes of finding connections within the data, so it becomes more comfortable. There are several things that we can do in future upon the current model such as restricting the right day of the week and time for each of the target public or create custom models for different groups to improve the forecast rate further and reduce the error rate.

The work that has done on this modelling and interpretation is an excellent source for the bank to acquire more consumers efficiently. There can always be developments and tweaking as further data comes in, as well as investigating corners and crowds within the data. What a great start to improving the target market strategy!