FITCHBURG STATE UNIVERSITY

Predicting Subscription to Bank Term Deposit for Clients

AS a request for Machine Learning Data Science Course Requirements

Yasaswini Nallapula St#: Spring 2023 Machine Learning Data Science

INTRODUCTION:

MOTIVATIONS BEHIND THE PROJECT:

The finance industry is among the top industries exploiting the value of big data. As with any business, the effective use of digital marketing has become increasingly important for community banks focused on expanding their core customer base. According to 35+ years of research, the average US adult has had the same checking account for over 10 years. People like to stay in their bank for the long term.

Marketing new potential customers and retaining them over the long term is a constant challenge for banks. To reach profitable customers, banks often use media such as social and digital media, customer service and strategic partnerships. The primary function of a bank is to accept deposits for the purpose of lending. Banks normally promote their products (Services) through direct marketing campaigns. For this project, I will be using Portuguese Bank dataset.

GOALS OF THE PROJECT:

The goal of this project is to:

(1) To explain how Portugal banking institution can use its client data and machine learning techniques to predict which customers would subscribe for bank term deposits.

(2) To identify the variable/feature importance in the accuracy of predictions.

WHY IT IS INTERESTING OR IMPORTANT:

Is it possible for banks to market to specific locations, communities, and groups of people? Fortunately, with the advent of machine learning technology, banking institutions are leveraging data and analytics solutions to target specific target customers. and to predict which customers accurately and intelligently are likely to purchase financial products and services. This project is related to one of the financial products in the Banking industry called Term Deposit.

What is Term Deposit and how is it useful to banks and to customers?

A term deposit is a fixed term deposit of money in an account of a financial institution. If a client or investor decides to deposit or invest in any of these accounts, they agree not to withdraw money for a period (from 1 month to 30 years) in exchange for a higher interest rate on that account. Banks can use this money to invest elsewhere or lend it to someone else for an agreed period. In other words, a term deposit guarantees that the client will receive money at a fixed interest rate for a fixed time. As term deposits are an important source of income for banks, banks invest large sums of money and focus on marketing campaigns to attract more customers to commit to a term deposit. However, not everyone can afford to put their money away for a while or even want to, therefore it would be a waste of resources to include them in the marketing campaign. Identifying the target market, the group of customers who are likely to buy term deposits, is a key task that allows banks to focus resources only on those customers with high potential for sale. Targeting is the most time efficient and highest ROI marketing technique.

WHY MACHINE LEARNING IS A REASONABLE APPROACH:

Machine Learning (ML) has gained a lot of traction in recent years due to its use across a wide variety of industries. ML algorithms are used to perform a specific task without being explicitly programmed - instead, they recognize patterns in the data and make predictions based on their learnings once new data arrives. ML is a great way of automating complex tasks that go further than rule-based automation. All businesses, large and small, strive to gain insight from the vast amounts of data they store and process on a regular basis. The desire to predict the future drives the work of analysts and data scientists in a variety of business areas such as healthcare, fraud

detection, marketing etc., Here, I have used R language to create a machine learning classification model.
R is a popular open-source data science programming language. It has the powerful visualization capabilities to explore the data before applying machine learning algorithms and evaluating their output. Many R machine learning packages are commercially available, and many modern statistical learning techniques are implemented in R.

BACKGROUND:

BACKGROUND INFORMATION REGARDING THE DATASET AND THE OVERALL MESSAGE/INFORMATION TO CONVEY:

The Bank of Portugal has collected a huge amount of data via phone calls. It was developed to support the marketing team of a Portuguese banking institution to be used for marketing campaigns. The dataset includes customers profiles of those who have subscribed to term deposits and the ones who did not subscribe to a term deposit.
The success criteria are defined as the "Accuracy with which the potential customers are identified for the term deposit subscription without the corresponding risk of incorrectly tagging the non-subscribers as potential subscribers and spending the time and money on them".
The dataset contains 45211 observations. Data is a mix of numeric and categorical variables including demographic details. Examples of numeric variables are customer's age, balance loan amount, last call duration etc., Examples of categorical variables are marital status of customer, whether customer has taken housing/personal loan etc.,
Duration is one of the numeric variables which is about last contact duration, measured in seconds and it highly affects the target variable (e.g., if duration=0 then term deposit='no').

RESEARCH QUESTION

Research question is – To examine existing records of banking customers in the training set and use this knowledge to predict whether customers in the evaluation set are likely to subscribe to a Term deposit or not. Here, I am planning to use Random Forest Machine Learning algorithm to build a predictive model.

IS THE AVAILABLE DATA SUFFICIENT TO DELIVER MESSAGE?

The dataset is about the customers of Bank of Portugal, and it contains approximately 45211 observations, including 5289 customers who have subscribed to term deposit. From this, we can conclude that the data landscape is highly imbalanced, with positive subscriptions accounting for 11.7% of the total observations.

DATASET DESCRIPTION

SOURCES OF THE DATA

The bank marketing dataset is from the UCI Repository, it contains 45211 observations/records and 17 variables/attributes.
Source of the dataset:
https://archive.ics.uci.edu/ml/datasets/Bank+Marketing#
The Bank of Portugal has collected a huge amount of data via phone calls that includes customers profiles of those who have subscribed to term deposits and the ones who did not subscribe to a

term deposit. The data contains detailed information of customers such as job, education, marital status, housing loan status, loan balance, term deposit status etc.,

WHAT DATASET IS PRESENT AND WHAT DOES IT MEANS:

The dataset has 16 independent variables and 1 dependent/target variable. Often, more than one contact with the same customer is required, to access if the product (bank term deposit) would be (or not) subscribed. The dependent variable is 'y- it describes the term deposit subscription status which is recoded as - yes and no.

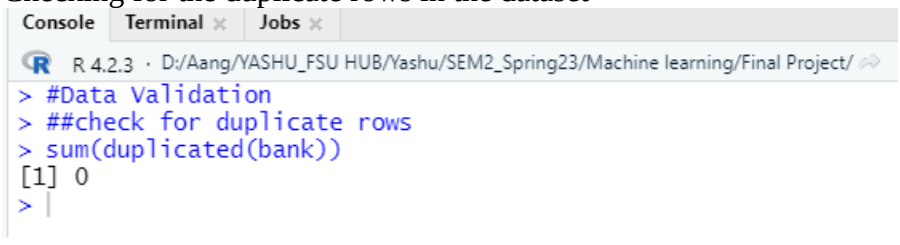
The classification goal is to predict if the client will subscribe (yes/no) to a term deposit, this is denoted by variable 'y'.

Sl. No	Variable	Description
1	age	numeric
2	job	type of job (categorical: 'admin', 'blue-collar', 'entrepreneur', 'housemaid', 'management', 'retired', 'self-employed', 'services', 'student', 'technician', 'unemployed', 'unknown')
3	marital	marital status (categorical: 'divorced','married','single'; note: 'divorced' means divorced or widowed)
4	education	(categorical: 'unknown','secondary','primary','tertiary')
5	default	has credit in default? (categorical: 'no','yes')
6	balance	average yearly loan balance, in euros (numeric)
7	housing	has housing loan? (categorical: 'no','yes')
8	loan	has personal loan? (categorical: 'no','yes')
9	contact	contact communication type (categorical: 'cellular','telephone', 'unknown')
10	day	last contact day of the month (numeric)
11	month	last contact month of year (categorical: 'jan', 'feb', 'mar', ..., 'nov', 'dec')
12	duration	last contact duration, in seconds (numeric)
13	campaign	number of contacts performed during this campaign and for this client (numeric, includes last contact)
14	pdays	number of days that passed by after the client was last contacted from a previous campaign (numeric; 999 means client was not previously contacted)
15	previous	number of contacts performed before this campaign and for this client (numeric)
16	poutcome	outcome of the previous marketing campaign (categorical: 'failure',unknown,'success', 'other')
17	y	has the client subscribed to a term deposit? (Categorical: yes, no)

CLEANING OR CUSTOMIZATION OF THE DATASET:

Data Validation and Data Cleaning:

Checking for the duplicate rows in the dataset



Checking for the rows that contain missing data. Checking for the missing values by each variable:

```
R 4.2.3 · D:/Aang/YASHU_FSU HUB/Yashu/SEM2_Spring23/Machine learning/Final Project/
> ##check for rows with Missing values
> sum(!complete.cases(bank))
[1] 0
> ##check for Missing values by variable
> sapply(bank, function(x) sum(is.na(x)))
      age       job   marital education   default   balance   housing      loan   contact       day     month
        0         0         0         0         0         0         0         0         0         0         0
 duration  campaign     pdays  previous  poutcome         y
        0         0         0         0         0         0
>
```

Data validation is an important step, because if the data has duplicates/missing or null values, it gives wrong results while building the models. Here, the bank dataset has no duplicates rows, and it has no null values (no missing values). So, the data is clean.

Customization of the Dataset:

String (character) type variables and the target variable ('y') are converted to factor type for the purpose of data analysis, visualizations, and Random Forest ML model building.

```
R 4.2.3 · D:/Aang/YASHU_FSU HUB/Yashu/SEM2_Spring23/Machine learning/Final Project/
> #convert character to factor type
> bank <- as.data.frame(unclass(bank), stringsAsFactors = TRUE)
>
> #print all the levels of factor variables
> levels(bank$job)
 [1] "admin."        "blue-collar"   "entrepreneur"  "housemaid"     "management"    "retired"
 [7] "self-employed" "services"      "student"       "technician"    "unemployed"    "unknown"
> levels(bank$marital)
[1] "divorced" "married"  "single"
> levels(bank$education)
[1] "primary"   "secondary" "tertiary"  "unknown"
> levels(bank$default)
[1] "no"  "yes"
> levels(bank$housing)
[1] "no"  "yes"
> levels(bank$loan)
[1] "no"  "yes"
> levels(bank$contact)
[1] "cellular"  "telephone" "unknown"
> levels(bank$month)
 [1] "apr" "aug" "dec" "feb" "jan" "jul" "jun" "mar" "may" "nov" "oct" "sep"
> levels(bank$poutcome)
[1] "failure" "other"   "success" "unknown"
> levels(bank$y)
[1] "no"  "yes"
>
```

From the above code, variables do not have missing values and data is pre-processed. Hence, the data is clean and clear to perform further exploratory data analysis, and for ML model building.

EXPLORE YOUR DATA (EXPLORATORY DATA ANALYSIS):

In the data, the number of persons who subscribed to Term Deposit are 5289 and it is 11.7%. The number of people who are not subscribed is 39922 and it is 88.3%.

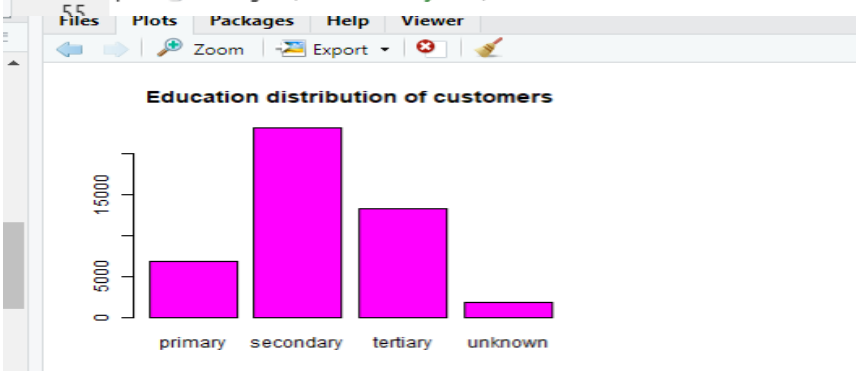
```
Console  Terminal  Jobs
R 4.2.3 · D:/Aang/YASHU_FSU HUB/Yashu/SEM2_Spring23/Machine learning/Final Project/
> #data analysis
> table(bank$y)

   no   yes
39922  5289
> round(prop.table(table(bank$y)) * 100, digits = 1)

  no  yes
88.3 11.7
>
```

Required Libraries to be installed and imported are:

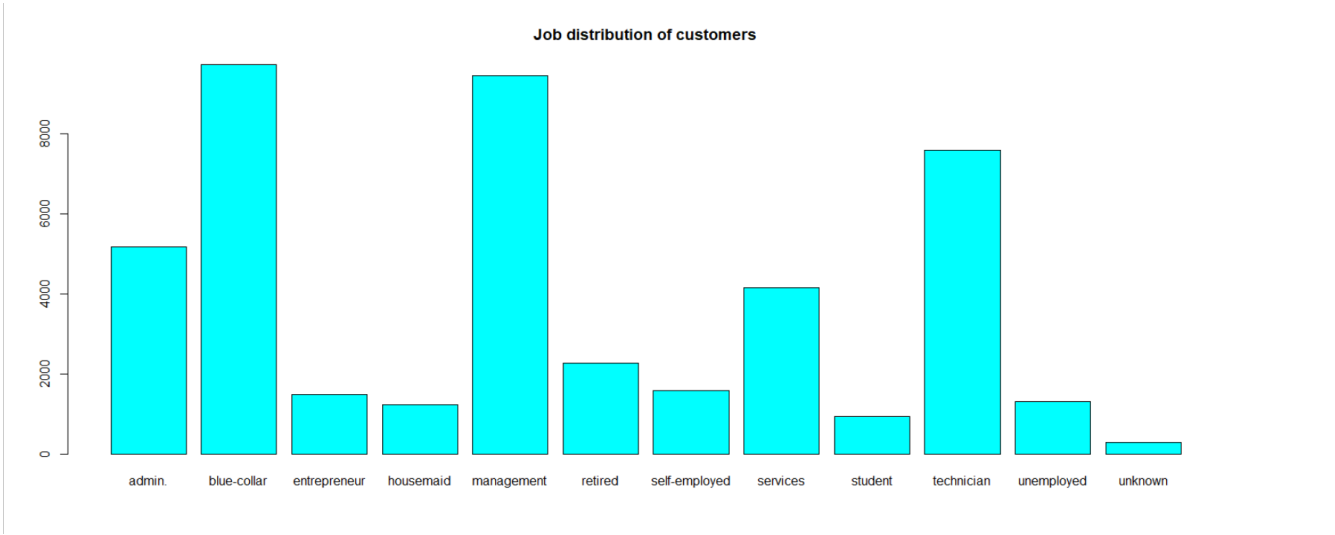
```r
#installing libraries
library(caret)
install.packages("DataExplorer")
library(DataExplorer)
library(dplyr)
library(glue)
```

Univariate Analysis: Distribution of each variable separately in the dataset

```r
#univariate analysis

plot(bank$education, col = "magenta", main = "Education distribution of customers")
plot(bank$job, col = "cyan", main = "Job distribution of customers")
```

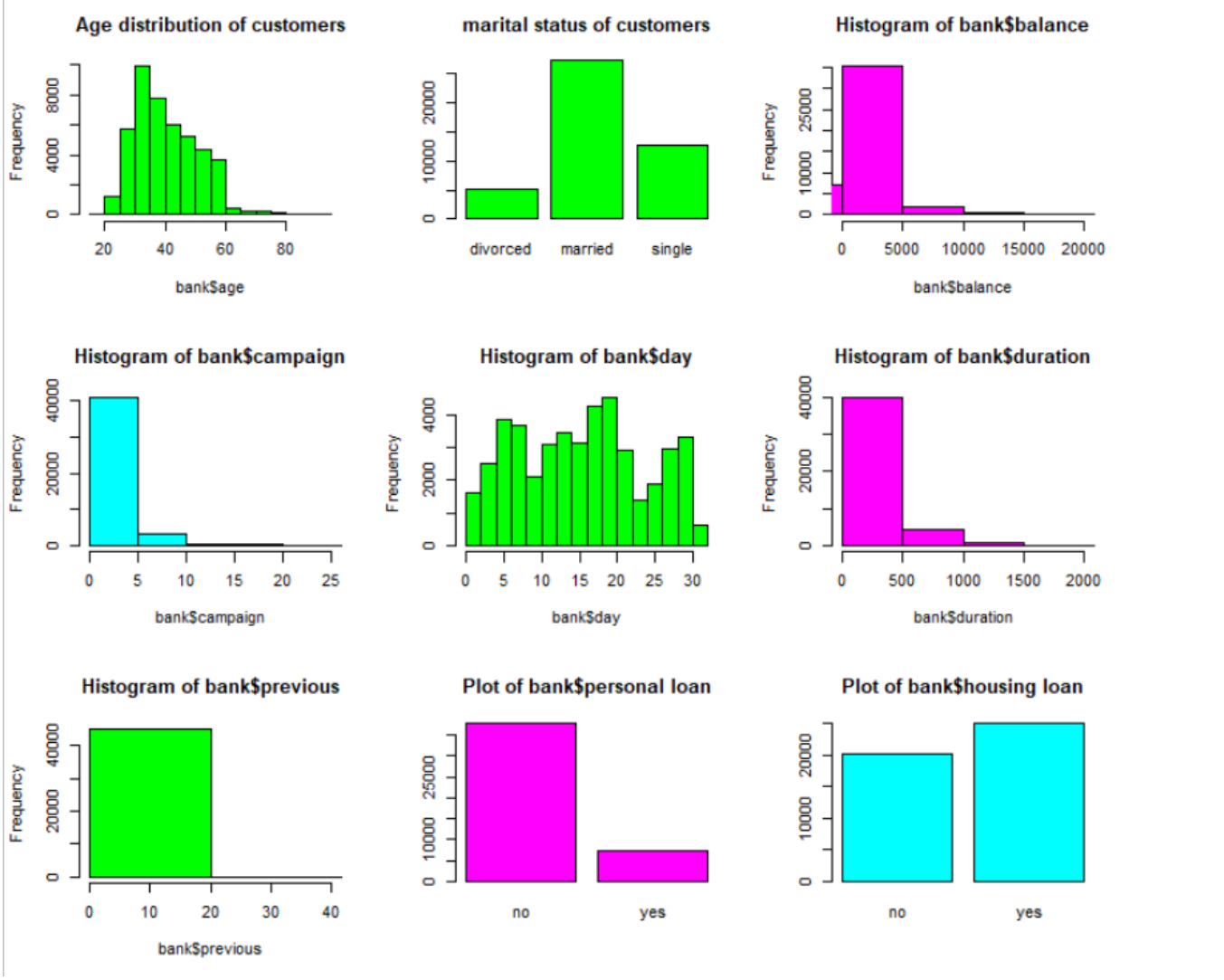


In the above plot, most of the customers are under the category of secondary and tertiary in their education levels. It means, most of them are well educated. There are a few customers (around 2000) who did not mention their education level.



In the above bar plot, most of the customers are under the job categories of blue-collar and management. There is minimal difference between the number of customers who are self-employed, entrepreneur, unemployed, housemaid and student. There are few customers who did not mention their job in the data.

```r
#univariate analysis
par(mfrow = c(3, 3))

p1 <- hist(bank$age, col = "green", main = "Age distribution of customers")
p4 <- plot(bank$marital, col = "green", main = "marital status of customers")
p5 <- hist(bank$balance, col = "magenta", xlim = c(0,20000), main = "Histogram of bank$balance")
p6 <- hist(bank$campaign, col = "cyan", xlim = c(0,25), main = "Histogram of bank$campaign")
p7 <- hist(bank$day, col = "green", main = "Histogram of bank$day")
p8 <- hist(bank$duration, col = "magenta", xlim = c(0,2000), main = "Histogram of bank$duration")
p10 <- hist(bank$previous, col = "green", xlim = c(0,40), main = "Histogram of bank$previous")
p11 <- plot(bank$loan, col = "magenta", main = "Plot of bank$personal loan")
p12 <- plot(bank$housing, col = "cyan", main = "Plot of bank$housing loan")
```

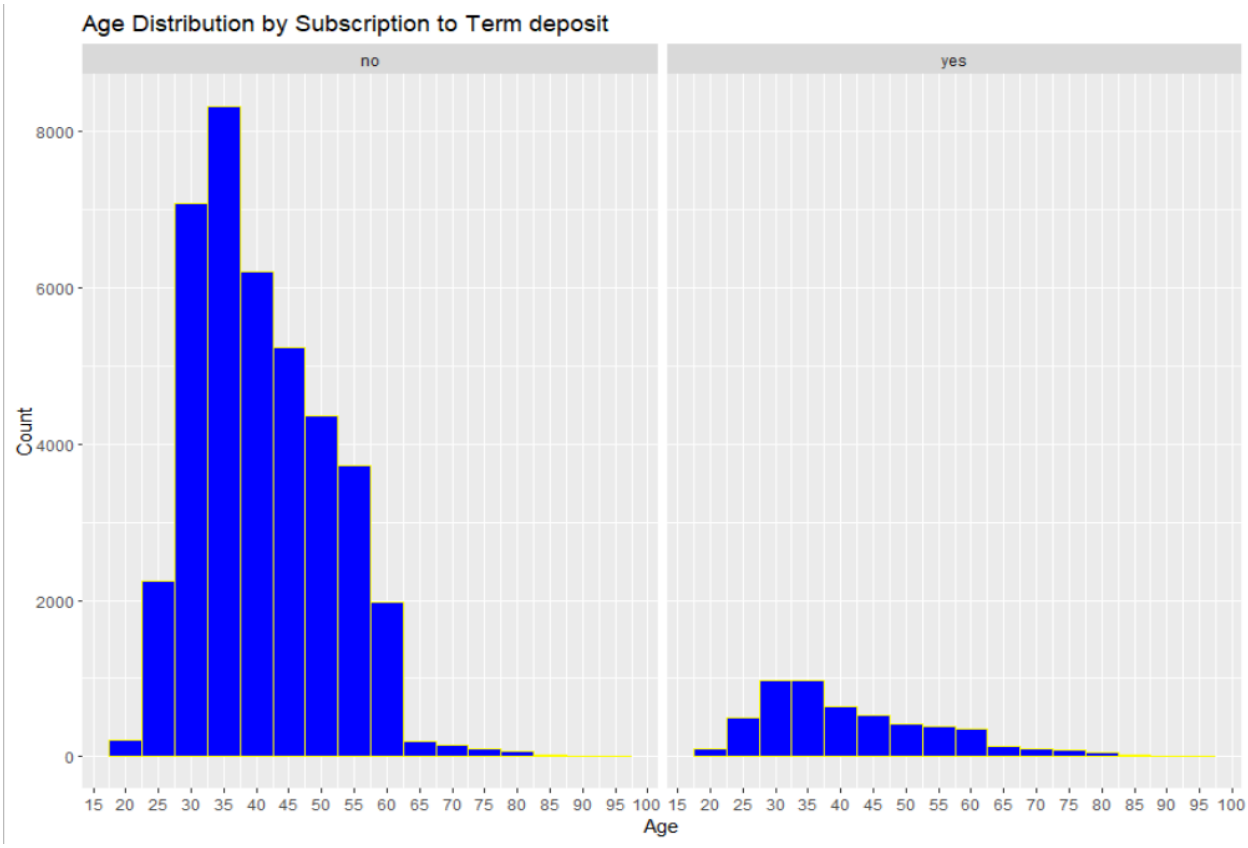


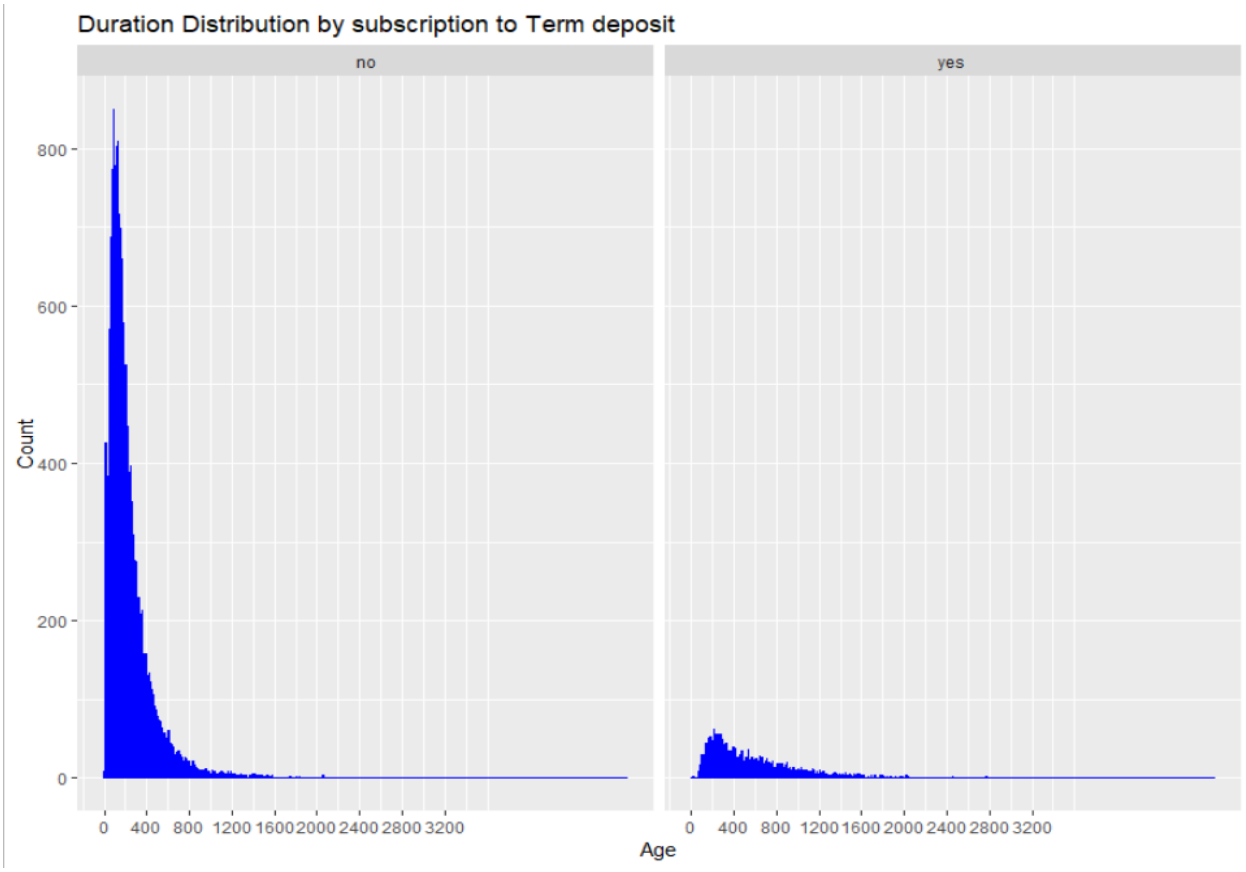
From the above plots, most of the customers in the dataset are under the age group of 25-45 and with the marital status of married or single. Most of the balance loan amount in under 5000 euros. The maximum last call duration was between 10-500 seconds. Personal loan was not taken by most of the customers in the dataset, whereas housing loan was taken by approximately 60% of the customers. Often, there are more contact for a customer which is denoted by campaign and the most percentage of contacts comes under 2-5.

Multivariate Analysis: Analyzing the distribution of independent variables based on the target variable ('y') i.e., term deposit subscription:

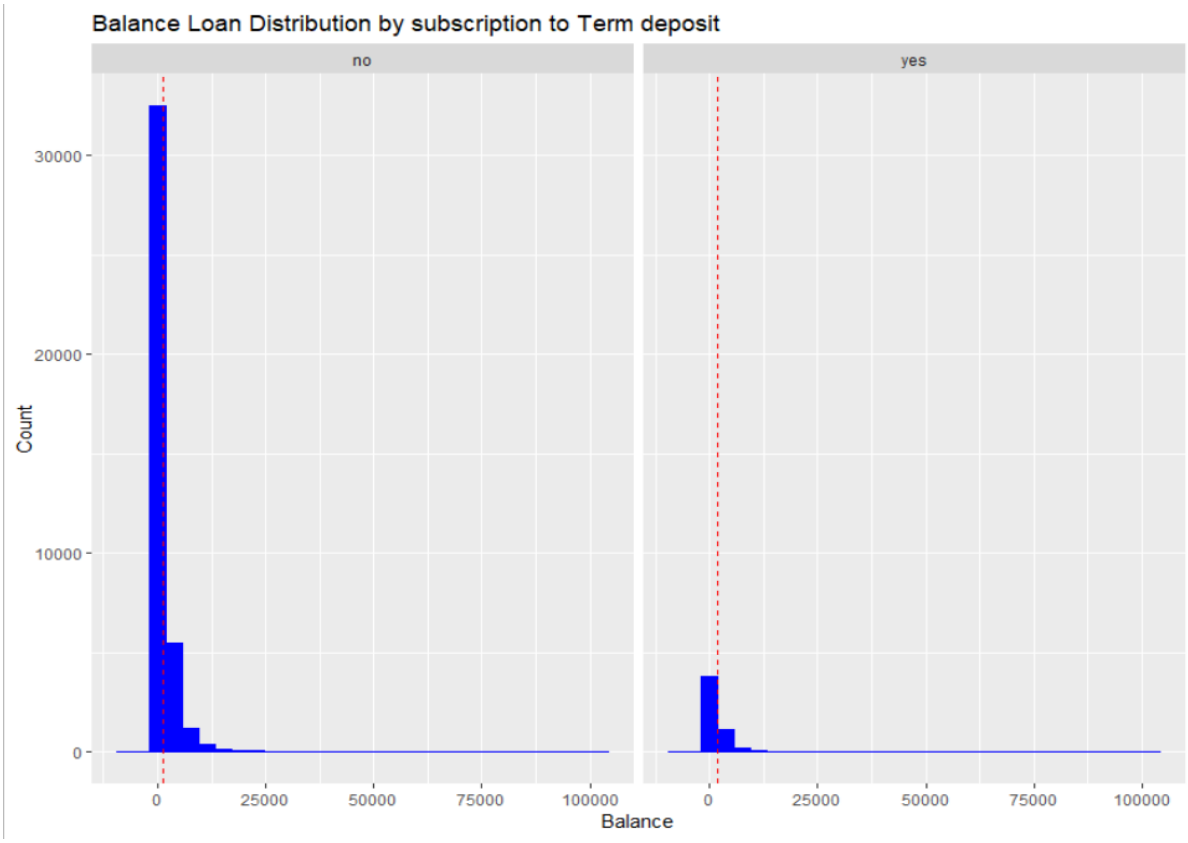
```r
# Multi variate analysis

age_group <- bank %>% group_by(y) %>% summarize(grp.mean=mean(age))
ggplot (bank, aes(x=age)) +
  geom_histogram(color = "yellow", fill = "blue", binwidth = 5) +
  facet_grid(cols=vars(y)) +
  ggtitle('Age Distribution by Subscription to Term deposit') + ylab('Count') + xlab('Age') +
  scale_x_continuous(breaks = seq(0,100,5))

duration_group <- bank %>% group_by(y) %>% summarize(grp.mean=mean(duration))
ggplot (bank, aes(x=duration)) +
  geom_histogram(color = "blue", fill = "blue", binwidth = 5) +
  facet_grid(cols=vars(y)) +
  ggtitle('Duration Distribution by subscription to Term deposit') + ylab('Count') + xlab('Age') +
  scale_x_continuous(breaks = seq(0,3200,400))

balance_group <- bank %>% group_by(y) %>% summarize(grp.mean=mean(balance))
ggplot (bank, aes(x=balance)) +
  geom_histogram(color = "blue", fill = "blue") +
  facet_grid(cols=vars(y)) +
  ggtitle('Balance Loan Distribution by subscription to Term deposit') + ylab('Count') + xlab('Balance') +
  geom_vline(data=balance_group, aes(xintercept=grp.mean), color="red", linetype="dashed")

ggplot(data=bank, aes(x=campaign, fill=y))+
  geom_histogram()+
  ggtitle("Subscription based on Number of Contact during the Campaign")+
  xlab("Number of Contact during the Campaign")+
  xlim(c(min=1,max=30)) +
  guides(fill=guide_legend(title="Subscription of Term Deposit"))

barplot(table(bank$y, bank$education), main="Distribution of Education vs Term Deposit",
        xlab="Education", col=c("darkblue","red"),
        legend = rownames(table(bank$y, bank$education)), beside=TRUE)

barplot(table(bank$y, bank$job), main="Distribution of Job vs Term Deposit",
        xlab="Job", col=c("darkblue","red"),
        legend = rownames(table(bank$y, bank$job)), beside=TRUE)

barplot(table(bank$y, bank$marital), main="Distribution of Marital status vs Deposit",
        xlab="Marital status", col=c("darkblue","red"),
        legend = rownames(table(bank$y, bank$marital)), beside=TRUE)

barplot(table(bank$y, bank$month), main="Distribution of Last contact month vs Deposit",
        xlab="Month", col=c("darkblue","red"),
        legend = rownames(table(bank$y, bank$month)), beside=TRUE)

```

The below bar plot divides the customers into two groups who are subscribed and who are not subscribed to term deposit based on their age. In both plots, most of the people under the age of 30-35 are the ones who are subscribed and who are not.

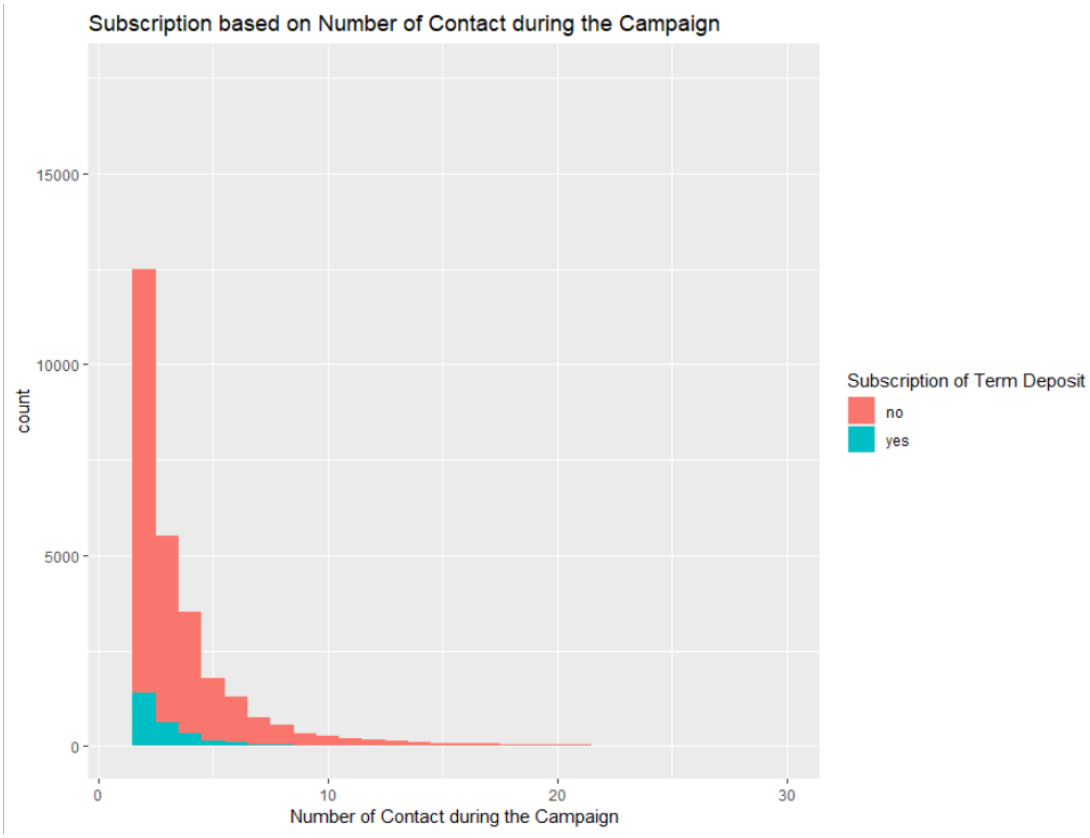




From the above plot visualization, most clients that subscribe had call duration between 150-1000 seconds.

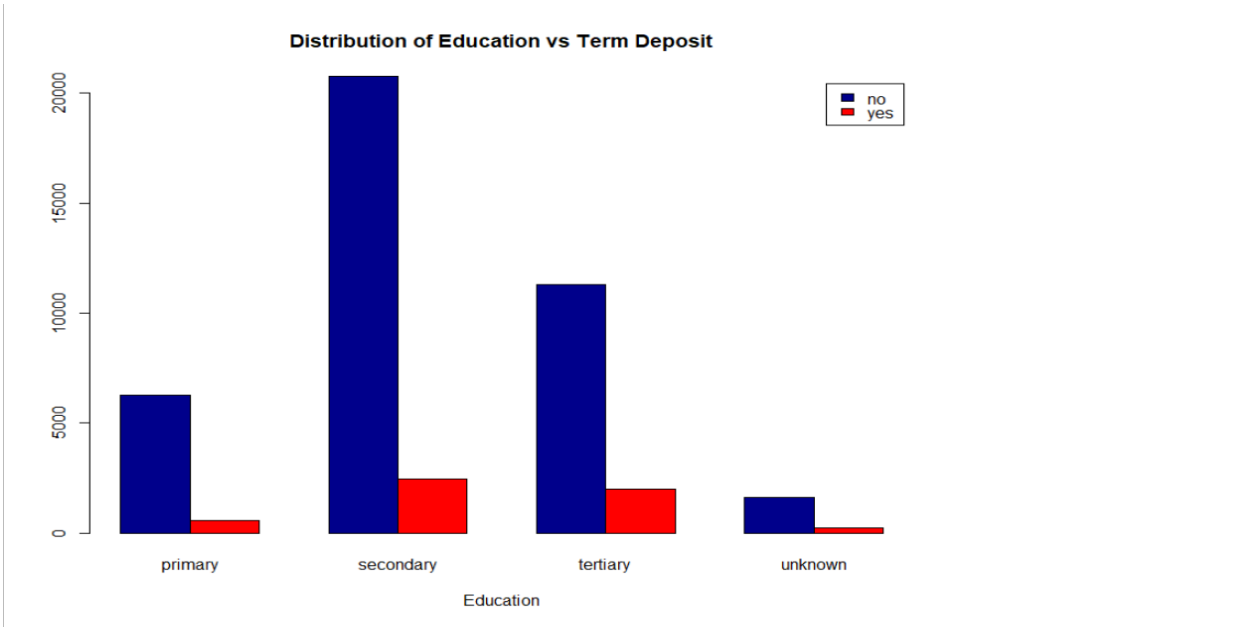


From the above Bar plot, customers who subscribe to term deposits had lower loan balance amounts.

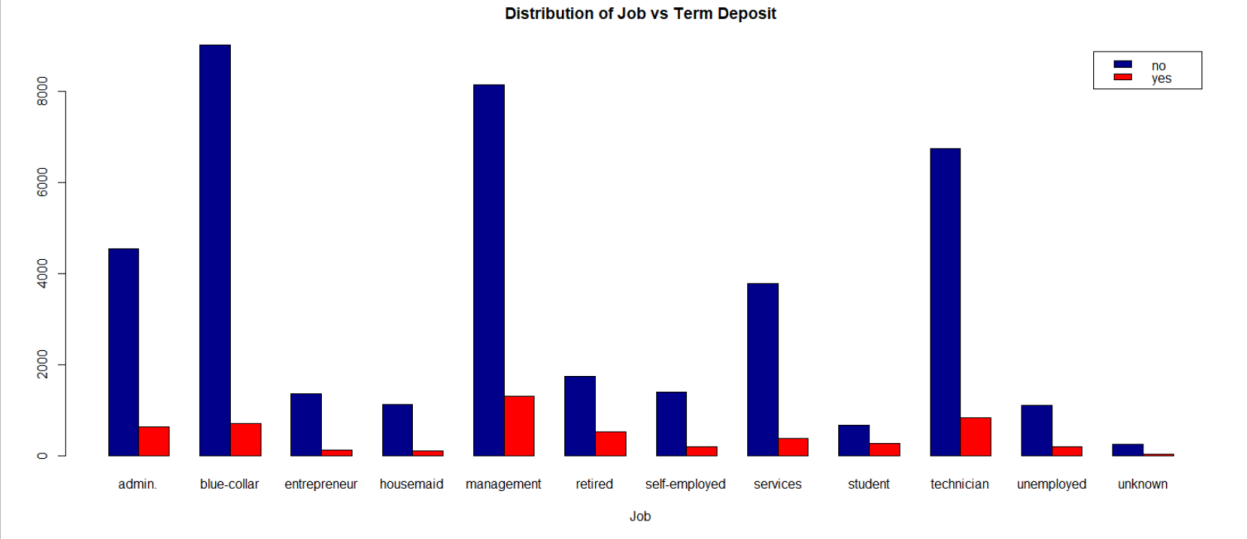


The above bar chart shows that there are no subscriptions beyond the number of contacts as 7 during the campaign. This could save resource utilization by setting limits on contacts during a future campaign. The first 4-5 contacts can be focused more as it will have a higher subscription rate.

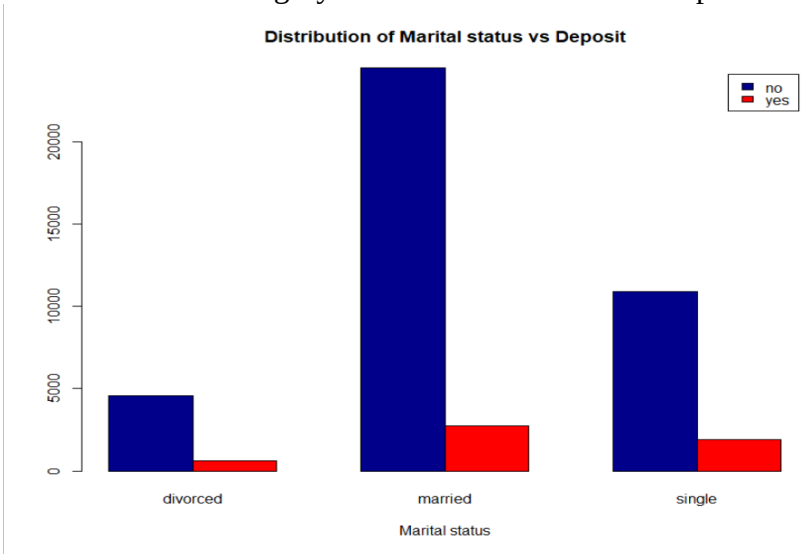
The below plot depicts most of the clients who subscribe are from ‘secondary’ and ‘tertiary’ education levels. Customers under Tertiary category have a higher rate of subscription compared to others.



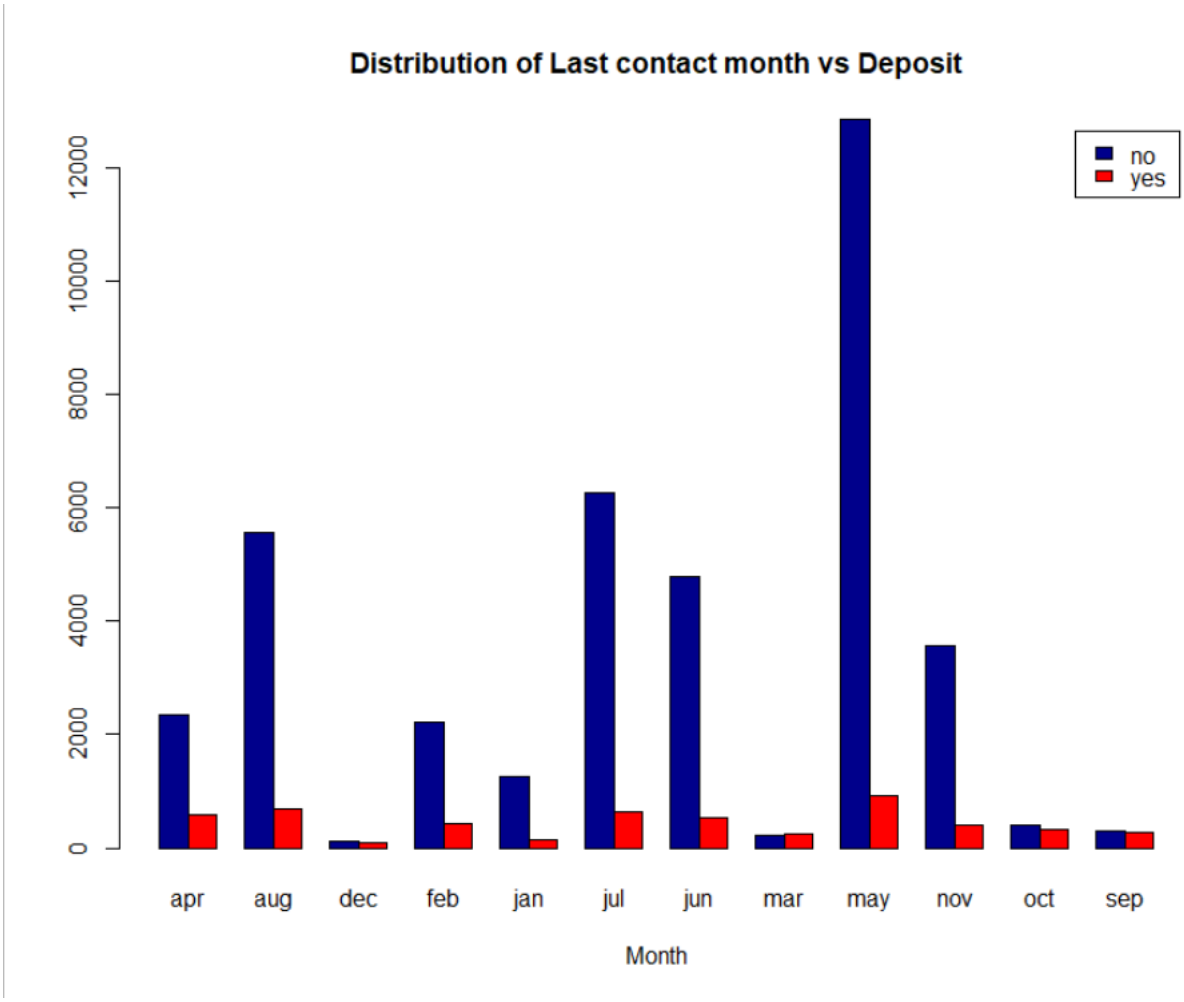
The plot described below shows most of the clients who subscribe are under the categories of management, technician, admin, blue-collar and retired. Customers under the retired category have a higher rate of subscription compared to others.



In the below plot, customers with all three levels of marital status have subscribed to term deposit. Clients under the single category have a higher rate of subscription compared to others, whereas clients under the category of divorced had less subscriptions.



The below plot depicts most of the customers who are contacted during April-August and these are the ones who have more subscriptions to term deposit. Very minimal contacts are made during December and March months.



MODEL'S BASELINE:

DESCRIPTION OF THE MACHINE LEARNING ALGORITHM THAT WE USE TO BUILD THIS MODEL:

Random forest is a popular supervised machine learning algorithm used for both classification and regression problems. It is an ensemble learning technique that combines multiple decision trees to make a final prediction. In a Random Forest, a set of decision trees is generated using a random subset of features and training data. Each decision tree is trained on a randomly selected subset of the training data and a random subset of features. The output of the Random Forest is the majority vote of all the decision trees. Random Forest can be used for a wide range of applications, such as fraud detection, image classification, stock price prediction, and customer churn prediction, among others.

WHY WE PREFERRED THIS ALGORITHM?

Why use random forests when decision trees can solve the same problem?
Decision trees are practical and easy to implement, but they lack accuracy. Decision trees work well with the training data that created them, but they are inflexible when it comes to classifying new samples. This means that the accuracy during the testing phase is very low. This happens due to a process called overfitting. Overfitting occurs when the model examines the training data enough to adversely affect the model`s performance on new data.
This means that perturbations in the training data are captured and learned as concepts by the model. The problem here, however, is that these concepts are not applied to the test data, adversely affecting the model`s ability to classify new data and reducing the accuracy of the test data. This is where Random Forest comes into play. It is based on the idea of bagging, which is used to reduce prediction variability by combining multiple decision tree results for different samples of the dataset.
Overall, Random Forest is a powerful and versatile algorithm that is widely used in the field of machine learning and data science. This model could handle noisy and high-dimensional data, avoid overfitting, and provide variable importance measures.

THE PROS AND CONS OF CHOSEN ALGORITHM:

Pros:

- Flexible for classification and regression problems.
- Works well with both categorical and continuous values.
- Helps improve accuracy by reducing overfitting of decision trees.
- Ability to efficiently handle missing data and save the generated forest with other data for future use.
- No pre-processing required.
- No data normalization is required as a rule-based approach is used.
- Estimate which variables are important in classification (variable importance).
- Built-in validation set: no need to sacrifice data for additional validation.
- Typically, very good performance.
- Calculate the similarity between pairs of cases. This can be used for clustering, outlier detection, or providing an interesting view of your data.

Cons:

- It might be slow for large datasets.

• It is hard to interpret.
• Requires a lot of computing power and resources to build many trees to combine the outputs.
• Also, many decision trees are combined to determine the class, so it takes a lot of time to learn.
• Although accurate, it often lags advanced boosting algorithms.
• Due to the ensemble of decision trees, there are also interpretability problems, and the significance of each variable cannot be determined.

BUILDING MACHINE LEARNING MODEL

STEP 1: IMPORT REQUIRED LIBRARIES

First, let's load the necessary libraries.
In R, we use the install() command to install the packages that are not pre-existing. We use the library() command to load/import the library.

```
#installing and importing libraries
library(caret)
library(randomForest)
library(gmodels)
```

STEP 2: LOAD THE DATA SET

```
#Load the dataset into R
setwd("D:/Aang/YASHU_FSU HUB/Yashu/SEM2_Spring23/Machine learning/Final Project")
getwd()
bank <- read.csv("bank_marketing.csv", header=TRUE, sep=",")
```

STEP 3: CHECK THE STRUCTURE OF THE DATASET

```
Console  Terminal  Jobs
R 4.2.3 · D:/Aang/YASHU_FSU HUB/Yashu/SEM2_Spring23/Machine learning/Final Project/
> #structure of dataset
> str(bank)
'data.frame':   45211 obs. of  17 variables:
 $ age      : int  58 44 33 47 33 35 28 42 58 43 ...
 $ job      : Factor w/ 12 levels "admin.","blue-collar",..: 5 10 3 2 12 5 5 3 6 10 ...
 $ marital  : Factor w/ 3 levels "divorced","married",..: 2 3 2 2 3 2 2 3 1 2 3 ...
 $ education: Factor w/ 4 levels "primary","secondary",..: 3 2 2 4 4 3 3 3 1 2 ...
 $ default  : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 2 1 1 ...
 $ balance  : int  2143 29 2 1506 1 231 447 2 121 593 ...
 $ housing  : Factor w/ 2 levels "no","yes": 2 2 2 2 1 2 2 2 2 2 ...
 $ loan     : Factor w/ 2 levels "no","yes": 1 1 2 1 1 1 2 1 1 1 ...
 $ contact  : Factor w/ 3 levels "cellular","telephone",..: 3 3 3 3 3 3 3 3 3 3 ...
 $ day      : int  5 5 5 5 5 5 5 5 5 5 ...
 $ month    : Factor w/ 12 levels "apr","aug","dec",..: 9 9 9 9 9 9 9 9 9 9 ...
 $ duration : int  261 151 76 92 198 139 217 380 50 55 ...
 $ campaign : int  1 1 1 1 1 1 1 1 1 1 ...
 $ pdays    : int  -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ...
 $ previous : int  0 0 0 0 0 0 0 0 0 0 ...
 $ poutcome : Factor w/ 4 levels "failure","other",..: 4 4 4 4 4 4 4 4 4 4 ...
 $ y        : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...
> dim(bank)
[1] 45211    17
>
```

The dataset has 45211 observations and 17 variables.

STEP 4: CHECKING THE SUMMARY

This gives an overview of the statistical distribution of the data. For example, in the data the median and mean ages of the group of clients are 39 and 41 respectively.

```
R 4.2.3 · D:/Aang/YASHU_FSU HUB/Yashu/SEM2_Spring23/Machine learning/Final Project/
> #summary of dataset
> summary(bank)
      age                job          marital          education     default        balance       housing      loan
 Min.   :18.00   blue-collar:9732   divorced: 5207   primary  : 6851   no :44396   Min.   : -8019   no :20081   no :37967
 1st Qu.:33.00   management :9458   married :27214   secondary:23202   yes:  815   1st Qu.:    72   yes:25130   yes: 7244
 Median :39.00   technician :7597   single  :12790   tertiary :13301               Median :   448
 Mean   :40.94   admin.     :5171                    unknown  : 1857               Mean   :  1362
 3rd Qu.:48.00   services   :4154                                                  3rd Qu.:  1428
 Max.   :95.00   retired    :2264                                                  Max.   :102127
                 (Other)    :6835
     contact           day            month         duration        campaign          pdays          previous          poutcome
 cellular :29285   Min.   : 1.00   may    :13766   Min.   :   0.0   Min.   : 1.000   Min.   : -1.0   Min.   :  0.0000   failure: 4901
 telephone: 2906   1st Qu.: 8.00   jul    : 6895   1st Qu.: 103.0   1st Qu.: 1.000   1st Qu.: -1.0   1st Qu.:  0.0000   other  : 1840
 unknown  :13020   Median :16.00   aug    : 6247   Median : 180.0   Median : 2.000   Median : -1.0   Median :  0.0000   success: 1511
                   Mean   :15.81   jun    : 5341   Mean   : 258.2   Mean   : 2.764   Mean   : 40.2   Mean   :  0.5803   unknown:36959
                   3rd Qu.:21.00   nov    : 3970   3rd Qu.: 319.0   3rd Qu.: 3.000   3rd Qu.: -1.0   3rd Qu.:  0.0000
                   Max.   :31.00   apr    : 2932   Max.   :4918.0   Max.   :63.000   Max.   :871.0   Max.   :275.0000
                                   (Other): 6060
  y
 no :39922
 yes: 5289
```

STEP 5: TRAIN - TEST SPLIT

First select a random seed using 'set.seed()' to make the model reproducible. Splitting the data into training and testing sets using createDataPartition() function with 85% for training data and 15% for testing data.

```
Console  Terminal  Jobs
R 4.2.3 · D:/Aang/YASHU_FSU HUB/Yashu/SEM2_Spring23/Machine learning/Final Project/
> #split the data into training and testing for the Random Forest
> bank_rf <- bank
> set.seed(126) # Set random seed for reproducibility
> train_indices_rf <- createDataPartition(bank_rf$y, p = 0.85, list = FALSE)
> bank_rf_train <- bank_rf[train_indices_rf, ]
> bank_rf_test <- bank_rf[-train_indices_rf, ]
>
```

STEP 6: SEPARATE THE TEST LABELS FROM THE TEST DATA

```
R 4.2.3 · D:/Aang/YASHU_FSU HUB/Yashu/SEM2_Spring23/Machine learning/Final Project/
> # separate the test labels from the test data
> bank_rf_test_data <- bank_rf_test[1:16]
> bank_rf_test_label <- bank_rf_test[,17]
> str(bank_rf_test_data)
'data.frame':   6781 obs. of  16 variables:
 $ age       : int  44 28 44 52 60 58 54 48 53 51 ...
 $ job       : Factor w/ 12 levels "admin.","blue-collar",..: 10 2 1 3 1 6 2 5 10 5 ...
 $ marital   : Factor w/ 3 levels "divorced","married",..: 3 2 2 2 2 2 2 1 1 2 ...
 $ education: Factor w/ 4 levels "primary","secondary",..: 2 2 2 2 2 4 2 3 2 3 ...
 $ default   : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...
 $ balance   : int  29 723 -372 113 39 96 1291 -244 989 6530 ...
 $ housing   : Factor w/ 2 levels "no","yes": 2 2 2 2 2 2 2 2 2 2 ...
 $ loan      : Factor w/ 2 levels "no","yes": 1 2 1 2 2 1 1 1 1 1 ...
 $ contact   : Factor w/ 3 levels "cellular","telephone",..: 3 3 3 3 3 3 3 3 3 3 ...
 $ day       : int  5 5 5 5 5 5 5 5 5 5 ...
 $ month     : Factor w/ 12 levels "apr","aug","dec",..: 9 9 9 9 9 9 9 9 9 9 ...
 $ duration  : int  151 262 172 127 208 616 266 253 812 91 ...
 $ campaign  : int  1 1 1 1 1 1 1 1 1 1 ...
 $ pdays     : int  -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ...
 $ previous  : int  0 0 0 0 0 0 0 0 0 0 ...
 $ poutcome  : Factor w/ 4 levels "failure","other",..: 4 4 4 4 4 4 4 4 4 4 ...
> str(bank_rf_test_label)
 Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...
>
```

Target variable is ‘y’, and it is separated as test label from the test data.

STEP 7: TRAIN THE MODEL

Using randomForest classifier algorithm, y is the outcome in the data to be modeled and all other variables are used as input references and training set is considered as input data for model building.

```
>
> # Creating a random forest classifier
> rf_model <- randomForest(formula = y ~ ., data = bank_rf_train)
> rf_model

Call:
 randomForest(formula = y ~ ., data = bank_rf_train)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 4

        OOB estimate of  error rate: 9.13%
Confusion matrix:
       no  yes class.error
no  32723 1211  0.03568692
yes  2296 2200  0.51067616
>
```

From the trained random forest model, the number of trees is 500 and each split is created after examining every variable and picking the best split from all the variables, so here it is 4. For each tree, using the leftover data, misclassification rate is calculated called as out of bag (OOB) error rate. Aggregate error from all trees is to determine overall OOB error rate for the classification. Here it is 9.13%.

STEP 8: MAKE PREDICTIONS

The function predict() is used to predict the values based on the input/independent variables. Arguments to be passed are: (1) the model instance for which predictions are required, here it is rf_model and (2) the second argument is the input test data to predict the values.

```
>
> # Making predictions on the testing data
> deposit_pred <- predict(rf_model, bank_rf_test_data)
>
>
> # Evaluating the accuracy of the model
> accuracy <- confusionMatrix(deposit_pred, bank_rf_test_label)$overall["Accuracy"]
> print(paste("Accuracy:", round(accuracy, 4)))
[1] "Accuracy: 0.9086"
>
```

The accuracy of predicted model is 90.86%. Here, the training data is 85% and test data is 15%.

STEP 9: COMPARE THE PREDICTED AND ACTUAL VALUES

```
R 4.2.3 · D:/Aang/YASHU_FSU HUB/Yashu/SEM2_Spring23/Machine learning/Final Project/
> CrossTable(x=bank_rf_test_label, y=deposit_pred, prop.chisq=FALSE)


   Cell Contents
|-------------------------|
|                       N |
|           N / Row Total |
|           N / Col Total |
|         N / Table Total |
|-------------------------|


Total Observations in Table:  6781


                 | deposit_pred
bank_rf_test_label |        no |       yes | Row Total |
-------------------|-----------|-----------|-----------|
                no |      5775 |       213 |      5988 |
                   |     0.964 |     0.036 |     0.883 |
                   |     0.934 |     0.356 |           |
                   |     0.852 |     0.031 |           |
-------------------|-----------|-----------|-----------|
               yes |       407 |       386 |       793 |
                   |     0.513 |     0.487 |     0.117 |
                   |     0.066 |     0.644 |           |
                   |     0.060 |     0.057 |           |
-------------------|-----------|-----------|-----------|
      Column Total |      6182 |       599 |      6781 |
                   |     0.912 |     0.088 |           |
-------------------|-----------|-----------|-----------|
```

To compare the predicted and actual values, crossTable() function is used ('gmodels' package is required to use it). This function takes two arguments: (1) x is the actual values and (2) y is the predicted values.
Here, total observations considered for the prediction are 6781 in count. Row represents actual values and Column denotes predicted values. Among these, a few observations are wrongly predicted (as per the actual values).
To be specific, 213 customers are predicted to be subscribed to term deposit but in actual they are not subscribed. And 407 customers are predicted to be not subscribed but in actual they are subscribed. Totally, 6161 out of 6781 observations are predicted correctly. Hence, accuracy rate is (6161*100)/6781 = 90.86 %.

```
R 4.2.3 · D:/Aang/YASHU_FSU HUB/Yashu/SEM2_Spring23/Machine learning/Final Project/
> #comparing the predicted and actual values
> confusionMatrix(deposit_pred, bank_rf_test_label)
Confusion Matrix and Statistics

          Reference
Prediction   no  yes
       no  5775  407
       yes  213  386

               Accuracy : 0.9086
                 95% CI : (0.9015, 0.9153)
    No Information Rate : 0.8831
    P-Value [Acc > NIR] : 7.872e-12

                  Kappa : 0.5048

 Mcnemar's Test P-Value : 9.112e-15

            Sensitivity : 0.9644
            Specificity : 0.4868
         Pos Pred Value : 0.9342
         Neg Pred Value : 0.6444
             Prevalence : 0.8831
         Detection Rate : 0.8516
   Detection Prevalence : 0.9117
      Balanced Accuracy : 0.7256

       'Positive' Class : no

>
```

confusionMatrix() is function which can be also be used to compare the predicted and actual values ('caret' package is required to use it). Additionally, it provides an accuracy rate which is the same as the one I have calculated above i.e., 90.86%.

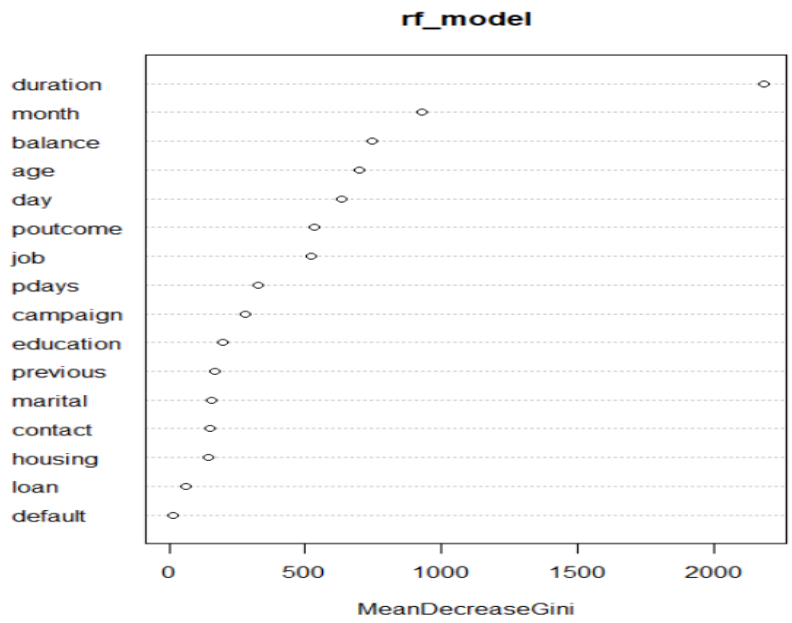
CONCLUSION:

ADD TO YOUR CONCLUSION

With real-world impact in mind, we should decide which metrics are most important. In this case, false negatives have a high negative impact because with this the bank might lose potential customers who are willing to subscribe to term deposit.

Feature/Variable importance based on the built random forest model is:

```
Console  Terminal  Jobs
R 4.2.3 · D:/Aang/YASHU_FSU HUB/Yashu/SEM2_Spring23/Machine learning/Final Project/
> # variable/feature importance
> varImpPlot(rf_model)
>
```



The higher the value of MeanDecreaseGini score, the higher the importance of the variable in the model. In the plot shown above, duration is the most important variable. We conclude that duration, month, balance, age, day, job influence the subscription to term deposit.
An oversampling technique was used because the original dataset has a very high-class imbalance. It would be interesting to create a new model using the balanced dataset.

STRUGGLES AND DIFFICULTIES USING R-STUDIO TO BUILD THE ML MODEL

Using RStudio is not so difficulty for the following reasons:

- The R language is an open-source tool as powerful and popular as Python.
- R's syntax is different from Python's, but the code is easy to understand for beginners.
- I was able to quickly build a random forest classification machine learning model in R.

In the initial dataset variables are of string and number types, I had to convert them to factor type (as factor is categorical variable that stores both integer and string data values as levels) to plot the data visualization and for Random Forest model. Initially, I tried converting each variable and it is a repetitive task. Later, I found one source code where all variables of string type can be converted to factor type in one line of code.
bank <- as.data.frame(unclass(bank), stringsAsFactors = TRUE)

SUGGESTION OF OTHER SOFTWARE(S) WORKS PROPERLY WITH OUR ML MODEL? WHY?

As a technology, ML uses different languages and tools to increase productivity. Along with R, one of the best languages for Machine Learning is Python.
The origins of Python can be traced back to the 1980s when Guido van Rossum started working on it. But it wasn't until the 2000s that Python became what it is today. And it's an open-source programming language. Python is versatile, code is easy to read and highly interpretable. As it is object-oriented also, we can write clear, concise code and use it in small and enterprise projects.
There are no type declarations (dynamic typing) for variables, parameters, functions, or methods in the source code. This keeps our code flexible, simple, and shorter in lines.

REFERENCES

1. https://www.kdnuggets.com/2019/02/caret-r-classify-term-deposit-subscriptions-bank.html
2. https://medium.com/analytics-vidhya/a-machine-learning-approach-to-identifying-customers-of-bank-of-portugal-who-would-subscribe-to-a-8bd04387aac2
3. https://rpubs.com/shienlong/wqd7004_RRookie
4. https://rstudio-pubs-static.s3.amazonaws.com/463779_7be86938710149cbb44633b2466cef7a.html
5. https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/