TelcoChurn

Carlo Cadei

2022-03-01

Loading libraries

```r
#Loading libraries
if(!require(tidyverse)) install.packages("tidyverse", repos = "http://cran.us.r-project.org")
```

```
## Loading required package: tidyverse

## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.8
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
if(!require(caret)) install.packages("caret", repos = "http://cran.us.r-project.org")
```

```
## Loading required package: caret

## Loading required package: lattice

##
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':
##
##     lift
```

```r
if(!require(data.table)) install.packages("data.table", repos = "http://cran.us.r-project.org")
```

```
## Loading required package: data.table

##
## Attaching package: 'data.table'
```

```
## The following objects are masked from 'package:dplyr':
##
##     between, first, last
```

```
## The following object is masked from 'package:purrr':
##
##     transpose
```

```r
if(!require(gridExtra)) install.packages("gridExtra", repos = "http://cran.us.r-project.org")
```

```
## Loading required package: gridExtra
```

```
##
## Attaching package: 'gridExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```r
if(!require(plyr)) install.packages("plyr", repos = "http://cran.us.r-project.org")
```

```
## Loading required package: plyr
```

```
## ------------------------------------------------------------------------------
```

```
## You have loaded plyr after dplyr - this is likely to cause problems.
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:
## library(plyr); library(dplyr)
```

```
## ------------------------------------------------------------------------------
```

```
##
## Attaching package: 'plyr'
```

```
## The following objects are masked from 'package:dplyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
```

```
## The following object is masked from 'package:purrr':
##
##     compact
```

```r
if(!require(rpart)) install.packages("rpart", repos = "http://cran.us.r-project.org")
```

```
## Loading required package: rpart
```

```
if(!require(rpart.plot)) install.packages("rpart.plot", repos = "http://cran.us.r-project.org")
```

```
## Loading required package: rpart.plot
```

```
if(!require(randomForest)) install.packages("randomForest", repos = "http://cran.us.r-project.org")
```

```
## Loading required package: randomForest

## randomForest 4.7-1

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:gridExtra':
##
##     combine

## The following object is masked from 'package:dplyr':
##
##     combine

## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
library(tidyverse)
library(caret)
library(data.table)
library(gridExtra)
library(plyr)
library(rpart)
library(rpart.plot)
library(randomForest)
```

if it does not create pdf: tinytex::install_tinytex()

TelcoChurn Report by Carlo Cadei

Introduction

Our client is a telecommunication company, we work with a commercial retention team that needs to find the reasons why a good portion of customers leaves for competitors and has to suggest particular offers for small groups of clients to minimize the risk of churn. We will work on two data files provided by the company:

- churn-customers.csv with customer personal details:

- customerID: internal ID
- genderCustomer: gender (female, male)
- SeniorCitizen: whether the customer is a senior citizen or not (1, 0)
- PartnerWhether: whether the customer has a partner or not (Yes, No)
- Dependents: whether the customer has dependents or not (Yes, No)

- churn-billing.csv with historical and contract data:
  - customerID: internal ID
  - tenure: number of months the customer has stayed with the company (number of months)
  - PhoneService: whether the customer has a phone service or not (Yes, No)
  - MultipleLines: whether the customer has multiple lines or not (Yes, No, No phone service)
  - InternetService: type of internet service subscribed (DSL, Fiber optic, No)
  - OnlineSecurity: whether the customer has online security or not (Yes, No, No internet service)
  - OnlineBackup: whether the customer has online backup or not (Yes, No, No internet service)
  - DeviceProtection: whether the customer has device protection or not (Yes, No, No internet service)
  - TechSupport: whether the customer has tech support or not (Yes, No, No internet service)
  - StreamingTV: whether the customer has streaming TV or not (Yes, No, No internet service)
  - StreamingMovies: whether the customer has streaming movies or not (Yes, No, No internet service)
  - Contract: customer contract term (Month-to-month, One year, Two year)
  - PaperlessBilling: whether the customer has paperless billing or not (Yes, No)
  - PaymentMethod: type of payment method subscribed (Electronic check, Mailed check, Bank transfer, Credit card)
  - MonthlyCharges: the amount charged to the customer monthly (amount of money)
  - TotalCharges: the total amount charged to the customer (amount of money)
  - Churn: whether the customer churned or not (Yes, No)

We need to build a machine to recognize whether a customer is going to leave the operator (Churn = Yes) or not (Churn = No).

Data loading and wrangling

After loading all necessary libraries we are now loading data, having two different files we need to join them in one cleaned file for analytics.

```
#Loading data
urlcb <- "https://raw.githubusercontent.com/ccadei/HarvardX/main/churn-billing.csv"
cb <- read.csv(urlcb)

urlcc <- "https://raw.githubusercontent.com/ccadei/HarvardX/main/churn-customer.csv"
cc <- read.csv(urlcc)

#Checking data
str(cb)
```

```
## 'data.frame':    7043 obs. of  17 variables:
##  $ customerID      : chr  "7590-VHVEG" "5575-GNVDE" "3668-QPYBK" "7795-CFOCW" ...
##  $ tenure          : int  1 34 2 45 2 8 22 10 28 62 ...
##  $ PhoneService    : chr  "No" "Yes" "Yes" "No" ...
##  $ MultipleLines   : chr  "No phone service" "No" "No" "No phone service" ...
##  $ InternetService : chr  "DSL" "DSL" "DSL" "DSL" ...
##  $ OnlineSecurity  : chr  "No" "Yes" "Yes" "Yes" ...
##  $ OnlineBackup    : chr  "Yes" "No" "Yes" "No" ...
##  $ DeviceProtection: chr  "No" "Yes" "No" "Yes" ...
##  $ TechSupport     : chr  "No" "No" "No" "Yes" ...
##  $ StreamingTV     : chr  "No" "No" "No" "No" ...
##  $ StreamingMovies : chr  "No" "No" "No" "No" ...
##  $ Contract        : chr  "Month-to-month" "One year" "Month-to-month" "One year" ...
##  $ PaperlessBilling: chr  "Yes" "No" "Yes" "No" ...
##  $ PaymentMethod   : chr  "Electronic check" "Mailed check" "Mailed check" "Bank transfer (automatic)
##  $ MonthlyCharges  : num  29.9 57 53.9 42.3 70.7 ...
##  $ TotalCharges    : num  29.9 1889.5 108.2 1840.8 151.7 ...
##  $ Churn           : chr  "No" "No" "Yes" "No" ...
```

```r
str(cc)
```

```
## 'data.frame':    7043 obs. of  5 variables:
##  $ customerID   : chr  "3999-WRNGR" "1965-DDBWU" "6734-CKRSM" "1761-AEZZR" ...
##  $ gender       : chr  "Female" "Male" "Female" "Male" ...
##  $ SeniorCitizen: int  0 0 0 0 0 1 0 0 0 0 ...
##  $ Partner      : chr  "Yes" "No" "No" "No" ...
##  $ Dependents   : chr  "Yes" "No" "No" "No" ...
```

```r
#Join data by customerID
telco <- inner_join(cc, cb, by = "customerID")

#Delete rows with missing data
sum(is.na(telco))
```

```
## [1] 11
```

```r
telco <- drop_na(telco)

#Convert character columns as factors
telco$SeniorCitizen <- as.factor(mapvalues(telco$SeniorCitizen,
                                          from = c("0", "1"),
                                          to = c("No", "Yes")))

telco$MultipleLines <- as.factor(mapvalues(telco$MultipleLines,
                                          from = c("No phone service"),
                                          to = c("No")))

for(i in 10:15){
  telco[,i] <- as.factor(mapvalues(telco[,i],
                                  from = c("No internet service"), to = c("No")))
}
```

```r
telco <- as.data.frame(unclass(telco),
                       stringsAsFactors = TRUE)

#Checking data
head(telco)
```

```
##   customerID gender SeniorCitizen Partner Dependents tenure PhoneService
## 1 3999-WRNGR Female            No     Yes        Yes     60           No
## 2 1965-DDBWU   Male            No      No         No     16          Yes
## 3 6734-CKRSM Female            No      No         No      3          Yes
## 4 1761-AEZZR   Male            No      No         No      1          Yes
## 5 4138-NAXED   Male            No      No         No     51          Yes
## 6 9355-NPPFS Female           Yes      No         No     26          Yes
##   MultipleLines InternetService OnlineSecurity OnlineBackup DeviceProtection
## 1            No             DSL             No          Yes               No
## 2           Yes     Fiber optic             No           No               No
## 3            No              No             No           No               No
## 4            No     Fiber optic             No           No               No
## 5           Yes     Fiber optic             No          Yes               No
## 6            No     Fiber optic             No           No               No
##   TechSupport StreamingTV StreamingMovies       Contract PaperlessBilling
## 1          No         Yes             Yes Month-to-month              Yes
## 2         Yes          No             Yes Month-to-month              Yes
## 3          No          No              No Month-to-month               No
## 4          No         Yes              No Month-to-month              Yes
## 5          No          No              No Month-to-month               No
## 6          No          No             Yes Month-to-month              Yes
##               PaymentMethod MonthlyCharges TotalCharges Churn
## 1          Electronic check          49.75      3069.45    No
## 2   Credit card (automatic)          89.05      1448.60   Yes
## 3              Mailed check          20.00        63.60    No
## 4          Electronic check          79.55        79.55   Yes
## 5 Bank transfer (automatic)          81.00      4085.75    No
## 6          Electronic check          78.80      2006.10    No
```

```r
str(telco)
```

```
## 'data.frame':    7032 obs. of  21 variables:
##  $ customerID      : Factor w/ 7032 levels "0002-ORFBO","0003-MKNFE",..: 2805 1323 4781 1195 2900 65
##  $ gender          : Factor w/ 2 levels "Female","Male": 1 2 1 2 2 1 1 1 1 2 ...
##  $ SeniorCitizen   : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 2 1 1 1 1 ...
##  $ Partner         : Factor w/ 2 levels "No","Yes": 2 1 1 1 1 1 1 1 2 1 ...
##  $ Dependents      : Factor w/ 2 levels "No","Yes": 2 1 1 1 1 1 1 1 1 1 ...
##  $ tenure          : int  60 16 3 1 51 26 3 28 16 1 ...
##  $ PhoneService    : Factor w/ 2 levels "No","Yes": 1 2 2 2 2 2 1 2 2 2 ...
##  $ MultipleLines   : Factor w/ 2 levels "No","Yes": 1 2 1 1 2 1 1 1 2 2 ...
##  $ InternetService : Factor w/ 3 levels "DSL","Fiber optic",..: 1 2 3 2 2 2 1 1 2 1 ...
##  $ OnlineSecurity  : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 2 2 1 ...
##  $ OnlineBackup    : Factor w/ 2 levels "No","Yes": 2 1 1 1 2 1 2 2 2 2 ...
##  $ DeviceProtection: Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 2 1 ...
##  $ TechSupport     : Factor w/ 2 levels "No","Yes": 1 2 1 1 1 1 1 1 1 1 ...
##  $ StreamingTV     : Factor w/ 2 levels "No","Yes": 2 1 1 2 1 1 2 1 1 1 ...
##  $ StreamingMovies : Factor w/ 2 levels "No","Yes": 2 2 1 1 1 2 2 1 1 1 ...
```

```
##  $ Contract        : Factor w/ 3 levels "Month-to-month",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ PaperlessBilling: Factor w/ 2 levels "No","Yes": 2 2 1 2 1 2 2 1 1 1 ...
##  $ PaymentMethod   : Factor w/ 4 levels "Bank transfer (automatic)",..: 3 2 4 3 1 3 3 3 3 3 ...
##  $ MonthlyCharges  : num  49.8 89 20 79.5 81 ...
##  $ TotalCharges    : num  3069.4 1448.6 63.6 79.5 4085.8 ...
##  $ Churn           : Factor w/ 2 levels "No","Yes": 1 2 1 2 1 1 2 1 1 1 ...
```

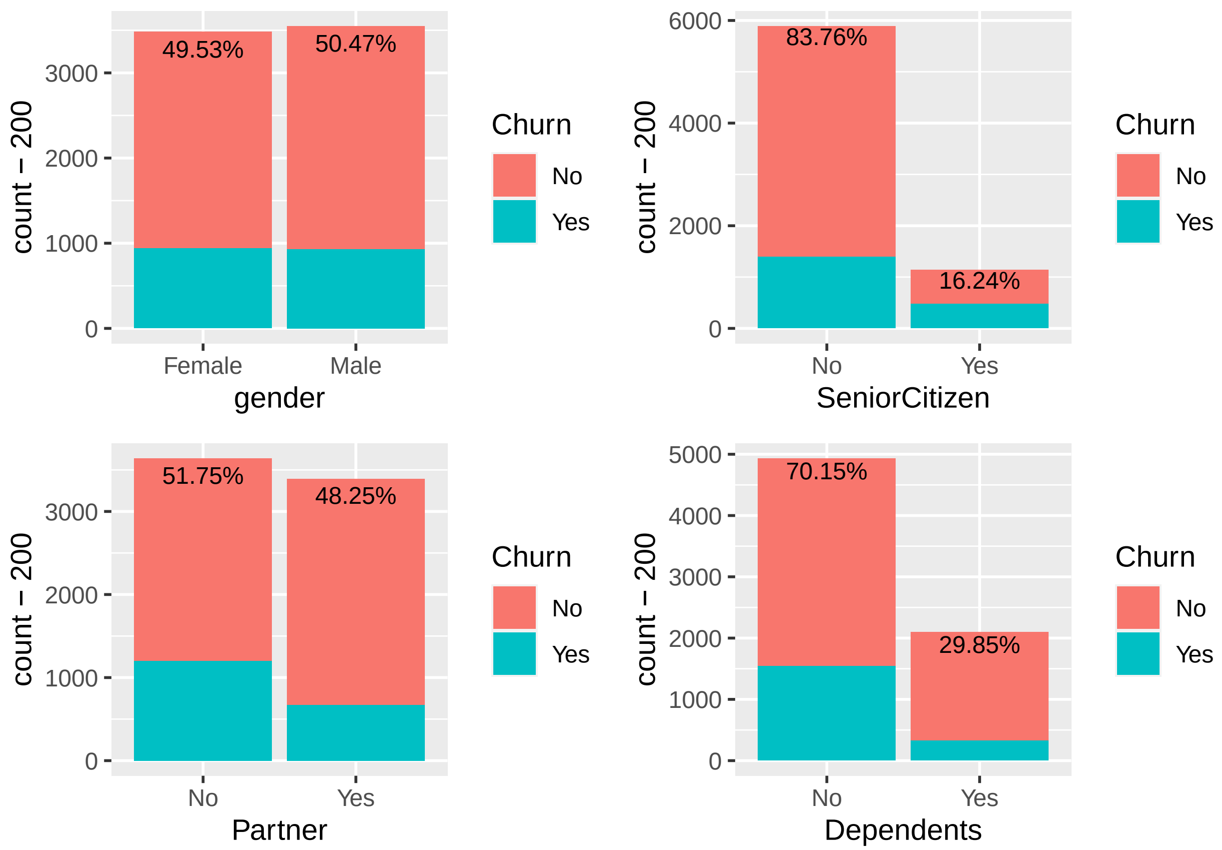
Having two files with custumerID as common variable, we (inner) join them via customerID to have a unique and complete information table. We checked for rows with missing data (NA). We also checked the type of information by row. We found 11 rows with missing data (NA); with more than 7000 rows we decide to delete rows with (NA). Some variables of services are dependent on other variables so we changed the responses from 'No phone service / No internet service' to 'No' for these variables. We also updated character rows as factors. We now have a new cleaned file ready for analytics.

Data visualisation

We can plot and examine our variables using bar charts for categorical variables and histograms for quantitative data.

```r
#Plotting variables
#Gender plot
p1 <- ggplot(telco, aes(x = gender)) +
  geom_bar(aes(fill = Churn)) +
  geom_text(aes(y = ..count..-200,
                label = paste0(round(prop.table(..count..),4) * 100, '%')),
            stat = 'count',
            position = position_dodge(.1),
            size = 3)

#Senior citizen plot
p2 <- ggplot(telco, aes(x = SeniorCitizen)) +
  geom_bar(aes(fill = Churn)) +
  geom_text(aes(y = ..count.. -200,
                label = paste0(round(prop.table(..count..),4) * 100, '%')),
            stat = 'count',
            position = position_dodge(.1),
            size = 3)

#Partner plot
p3 <- ggplot(telco, aes(x = Partner)) +
  geom_bar(aes(fill = Churn)) +
  geom_text(aes(y = ..count.. -200,
                label = paste0(round(prop.table(..count..),4) * 100, '%')),
            stat = 'count',
            position = position_dodge(.1),
            size = 3)

#Dependents plot
p4 <- ggplot(telco, aes(x = Dependents)) +
  geom_bar(aes(fill = Churn)) +
  geom_text(aes(y = ..count.. -200,
                label = paste0(round(prop.table(..count..),4) * 100, '%')),
            stat = 'count',
```

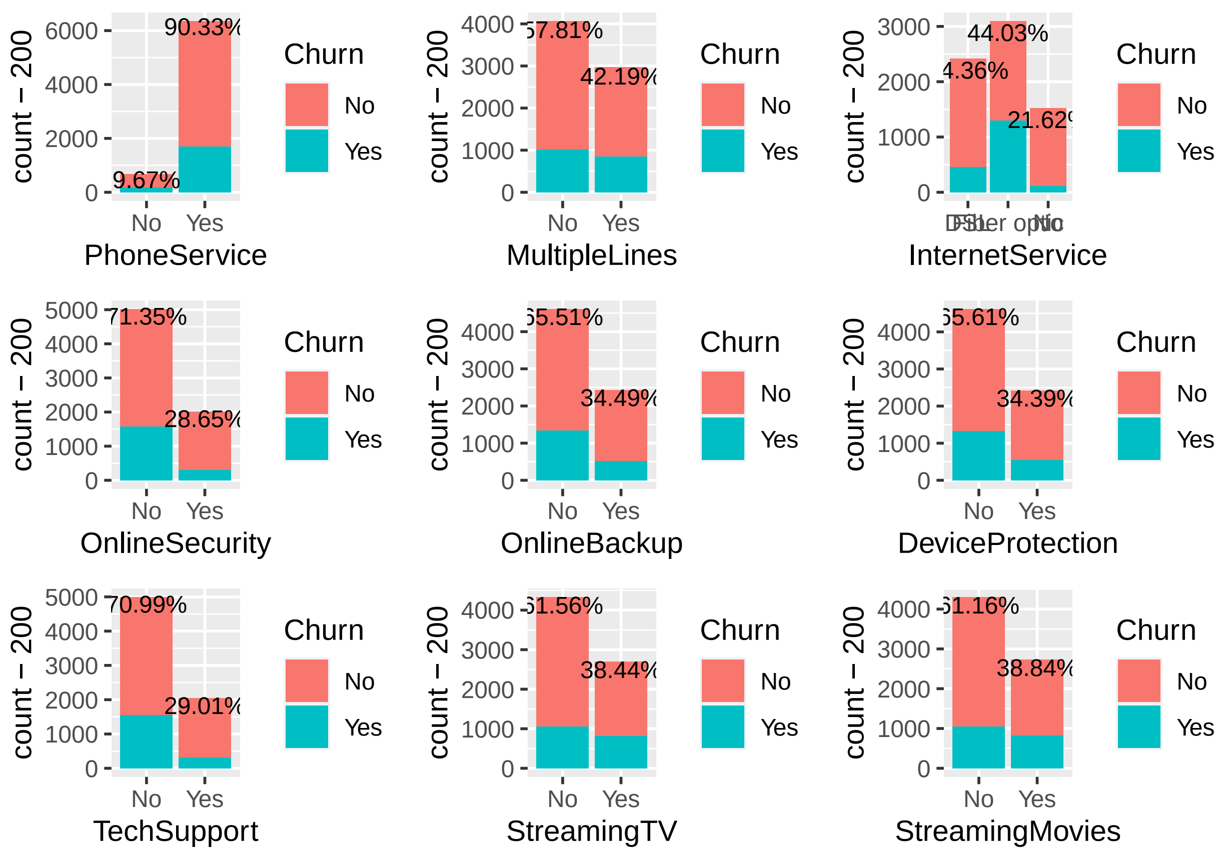
```
            position = position_dodge(.1),
            size = 3)

#Plot demographic data within a grid
grid.arrange(p1, p2, p3, p4, ncol = 2)
```




```
#Phone service plot
p5 <- ggplot(telco, aes(x = PhoneService)) +
  geom_bar(aes(fill = Churn)) +
  geom_text(aes(y = ..count.. -200,
                label = paste0(round(prop.table(..count..),4) * 100, '%')),
            stat = 'count',
            position = position_dodge(.1),
            size = 3)

#Multiple phone lines plot
p6 <- ggplot(telco, aes(x = MultipleLines)) +
  geom_bar(aes(fill = Churn)) +
  geom_text(aes(y = ..count.. -200,
                label = paste0(round(prop.table(..count..),4) * 100, '%')),
            stat = 'count',
            position = position_dodge(.1),
            size = 3)

#Internet service plot
```

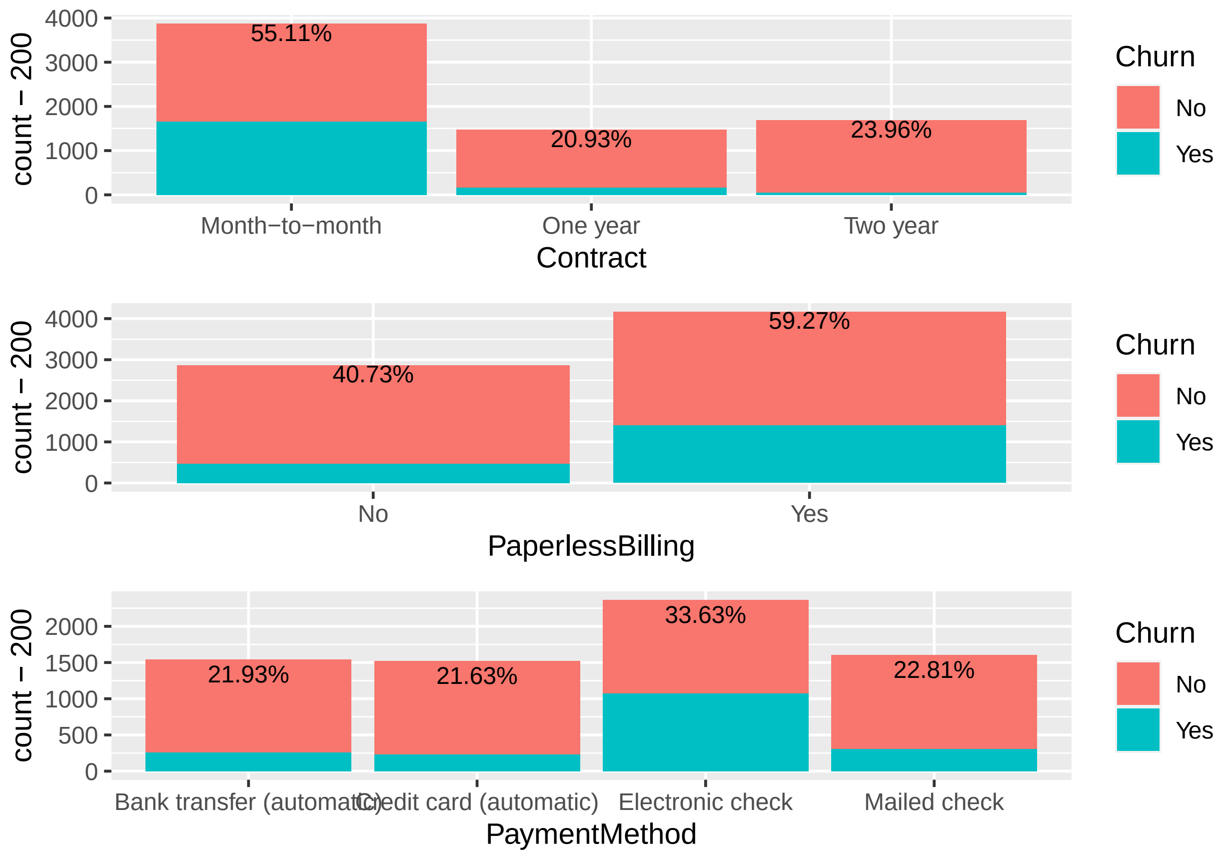
```r
p7 <- ggplot(telco, aes(x = InternetService)) +
  geom_bar(aes(fill = Churn)) +
  geom_text(aes(y = ..count.. -200,
                label = paste0(round(prop.table(..count..),4) * 100, '%')),
            stat = 'count',
            position = position_dodge(.1),
            size = 3)

#Online security service plot
p8 <- ggplot(telco, aes(x = OnlineSecurity)) +
  geom_bar(aes(fill = Churn)) +
  geom_text(aes(y = ..count.. -200,
                label = paste0(round(prop.table(..count..),4) * 100, '%')),
            stat = 'count',
            position = position_dodge(.1),
            size = 3)

#Online backup service plot
p9 <- ggplot(telco, aes(x = OnlineBackup)) +
  geom_bar(aes(fill = Churn)) +
  geom_text(aes(y = ..count.. -200,
                label = paste0(round(prop.table(..count..),4) * 100, '%')),
            stat = 'count',
            position = position_dodge(.1),
            size = 3)

#Device Protection service plot
p10 <- ggplot(telco, aes(x = DeviceProtection)) +
  geom_bar(aes(fill = Churn)) +
  geom_text(aes(y = ..count.. -200,
                label = paste0(round(prop.table(..count..),4) * 100, '%')),
            stat = 'count',
            position = position_dodge(.1),
            size = 3)

#Tech Support service plot
p11 <- ggplot(telco, aes(x = TechSupport)) +
  geom_bar(aes(fill = Churn)) +
  geom_text(aes(y = ..count.. -200,
                label = paste0(round(prop.table(..count..),4) * 100, '%')),
            stat = 'count',
            position = position_dodge(.1),
            size = 3)

#Streaming TV service plot
p12 <- ggplot(telco, aes(x = StreamingTV)) +
  geom_bar(aes(fill = Churn)) +
  geom_text(aes(y = ..count.. -200,
                label = paste0(round(prop.table(..count..),4) * 100, '%')),
            stat = 'count',
            position = position_dodge(.1),
            size = 3)
```

```
#Streaming Movies service plot
p13 <- ggplot(telco, aes(x = StreamingMovies)) +
  geom_bar(aes(fill = Churn)) +
  geom_text(aes(y = ..count.. -200,
                label = paste0(round(prop.table(..count..),4) * 100, '%')),
            stat = 'count',
            position = position_dodge(.1),
            size = 3)

#Plot service data within a grid
grid.arrange(p5, p6, p7,p8, p9, p10,
             p11, p12, p13, ncol = 3)
```



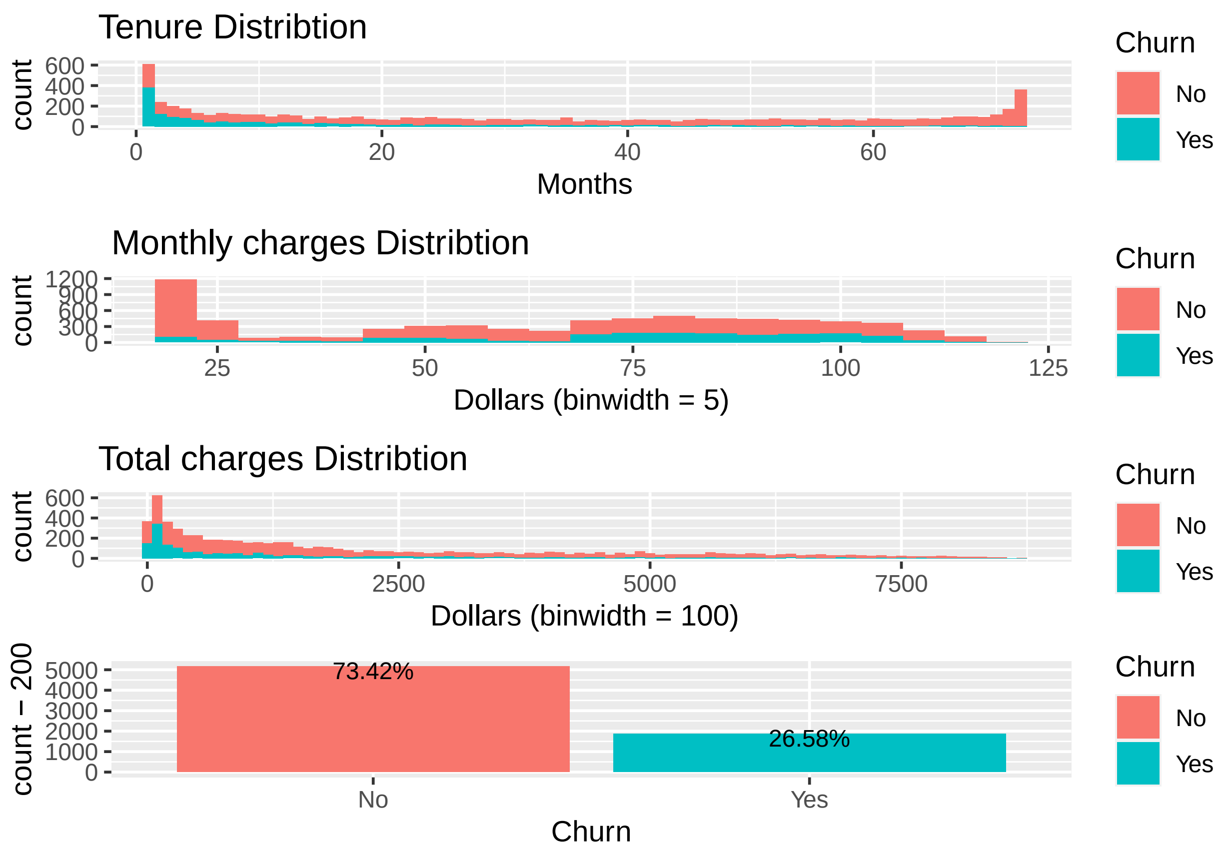

```
#Contract status plot
p14 <- ggplot(telco, aes(x = Contract)) +
  geom_bar(aes(fill = Churn)) +
  geom_text(aes(y = ..count.. -200,
                label = paste0(round(prop.table(..count..),4) * 100, '%')),
            stat = 'count',
            position = position_dodge(.1),
            size = 3)

#Paperless billing plot
p15 <- ggplot(telco, aes(x = PaperlessBilling)) +
  geom_bar(aes(fill = Churn)) +
```

```
  geom_text(aes(y = ..count.. -200,
                label = paste0(round(prop.table(..count..),4) * 100, '%')),
            stat = 'count',
            position = position_dodge(.1),
            size = 3)

#Payment method plot
p16 <- ggplot(telco, aes(x = PaymentMethod)) +
  geom_bar(aes(fill = Churn)) +
  geom_text(aes(y = ..count.. -200,
                label = paste0(round(prop.table(..count..),4) * 100, '%')),
            stat = 'count',
            position = position_dodge(.1),
            size = 3)

#Plot contract data within a grid
grid.arrange(p14, p15, p16, ncol = 1)
```




```
#Tenure histogram
p17 <- ggplot(telco, aes(x = tenure, fill = Churn)) +
  geom_histogram(binwidth = 1) +
  labs(x = "Months",
       title = "Tenure Distribtion")

#Monthly charges histogram
```

```r
p18 <- ggplot(telco, aes(x = MonthlyCharges, fill = Churn)) +
  geom_histogram(binwidth = 5) +
  labs(x = "Dollars (binwidth = 5)",
       title = "Monthly charges Distribtion")

#Total charges histogram
p19 <- ggplot(telco, aes(x = TotalCharges, fill = Churn)) +
  geom_histogram(binwidth = 100) +
  labs(x = "Dollars (binwidth = 100)",
       title = "Total charges Distribtion")

#Churn plot
p20 <- ggplot(telco, aes(x = Churn, fill = Churn)) +
  geom_bar() +
  geom_text(aes(y = ..count.. -200,
                label = paste0(round(prop.table(..count..),4) * 100, '%')),
            stat = 'count',
            position = position_dodge(.1),
            size = 3)

#Plot quantitative and churn data within a grid
grid.arrange(p17, p18, p19, p20, ncol = 1)
```



From the first block of demographic bar chart plots we notice that the sample is evenly split across gender and gender seems to have no influence on churn rate.

From the second block of services bar chart plots we can see two pairs of variables that seem to have the same consistence: OnlineBackup with DeviceProtection and StreamingTV with StreamingMovies.

From the third block of contract bar chart plots we find out that roughly half of the sample is on month-to-month contract with a very high rate of churn, with the remaining split between one and two year contracts with low rate of churn.

From the fourth block of numerical variables, the tenure variable is stacked at the tails, therefore a large proportion of customers has either had the shortest (1 month) with high rate of churn or the longest (72 months) tenure. It is possible that the beginning of the collection of our data started 72 months ago. Further investigation on the 72 month tenure would be necessary if the rate of churn were not as low as it is in that area. The TotalCharges variable is the mathematical product of tenure and monthly charges.

Considering that we have to present our work to business people and that we could have much bigger database to test in the future, it seems reasonable to try and reduce the number of variables eliminating the one that, based on previous observations, are little significant.

Get ready for modelling

To get ready for modelling we are going to cut unnecessary variables and divide our database between train and test set.

```
#Reduce variables and split data
#Simplify data cutting columns
newtelco <- telco %>%
  select(-customerID, -gender, -DeviceProtection, -StreamingMovies, -TotalCharges)

#Checking data
str(newtelco)
```

```
## 'data.frame':    7032 obs. of  16 variables:
##  $ SeniorCitizen   : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 2 1 1 1 1 ...
##  $ Partner         : Factor w/ 2 levels "No","Yes": 2 1 1 1 1 1 1 1 2 1 ...
##  $ Dependents      : Factor w/ 2 levels "No","Yes": 2 1 1 1 1 1 1 1 1 1 ...
##  $ tenure          : int  60 16 3 1 51 26 3 28 16 1 ...
##  $ PhoneService    : Factor w/ 2 levels "No","Yes": 1 2 2 2 2 2 1 2 2 2 ...
##  $ MultipleLines   : Factor w/ 2 levels "No","Yes": 1 2 1 1 2 1 1 1 2 2 ...
##  $ InternetService : Factor w/ 3 levels "DSL","Fiber optic",..: 1 2 3 2 2 2 1 1 2 1 ...
##  $ OnlineSecurity  : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 2 2 1 ...
##  $ OnlineBackup    : Factor w/ 2 levels "No","Yes": 2 1 1 1 2 1 2 2 2 2 ...
##  $ TechSupport     : Factor w/ 2 levels "No","Yes": 1 2 1 1 1 1 1 1 1 1 ...
##  $ StreamingTV     : Factor w/ 2 levels "No","Yes": 2 1 1 2 1 1 2 1 1 1 ...
##  $ Contract        : Factor w/ 3 levels "Month-to-month",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ PaperlessBilling: Factor w/ 2 levels "No","Yes": 2 2 1 2 1 2 2 1 1 1 ...
##  $ PaymentMethod   : Factor w/ 4 levels "Bank transfer (automatic)",..: 3 2 4 3 1 3 3 3 3 3 ...
##  $ MonthlyCharges  : num  49.8 89 20 79.5 81 ...
##  $ Churn           : Factor w/ 2 levels "No","Yes": 1 2 1 2 1 1 2 1 1 1 ...
```

```
#Splitting in train and test set
set.seed(2022, sample.kind = "Rounding")
```

```
## Warning in set.seed(2022, sample.kind = "Rounding"): non-uniform 'Rounding'
## sampler used
```

```r
test_index <- createDataPartition(newtelco$Churn, times = 1, p = 0.7, list = FALSE)
train <- newtelco[test_index,]
test <- newtelco[-test_index,]

#Checking data
str(train)
```

```
## 'data.frame':    4924 obs. of  16 variables:
##  $ SeniorCitizen   : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 2 1 1 ...
##  $ Partner         : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 2 1 1 2 ...
##  $ Dependents      : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 2 1 1 2 ...
##  $ tenure          : int  16 3 1 51 28 1 37 69 2 61 ...
##  $ PhoneService    : Factor w/ 2 levels "No","Yes": 2 2 2 2 2 2 2 2 2 1 ...
##  $ MultipleLines   : Factor w/ 2 levels "No","Yes": 2 1 1 2 1 2 1 2 1 1 ...
##  $ InternetService : Factor w/ 3 levels "DSL","Fiber optic",..: 2 3 2 2 1 1 2 2 3 1 ...
##  $ OnlineSecurity  : Factor w/ 2 levels "No","Yes": 1 1 1 1 2 1 1 2 1 2 ...
##  $ OnlineBackup    : Factor w/ 2 levels "No","Yes": 1 1 1 2 2 2 1 1 1 1 ...
##  $ TechSupport     : Factor w/ 2 levels "No","Yes": 2 1 1 1 1 1 1 2 1 1 ...
##  $ StreamingTV     : Factor w/ 2 levels "No","Yes": 1 1 2 1 1 1 2 2 1 1 ...
##  $ Contract        : Factor w/ 3 levels "Month-to-month",..: 1 1 1 1 1 1 2 3 1 1 ...
##  $ PaperlessBilling: Factor w/ 2 levels "No","Yes": 2 1 2 1 1 1 1 1 1 1 ...
##  $ PaymentMethod   : Factor w/ 4 levels "Bank transfer (automatic)",..: 2 4 3 1 3 3 2 2 4 1 ...
##  $ MonthlyCharges  : num  89 20 79.5 81 56 ...
##  $ Churn           : Factor w/ 2 levels "No","Yes": 2 1 2 1 1 1 1 1 1 1 ...
```

```r
str(test)
```

```
## 'data.frame':    2108 obs. of  16 variables:
##  $ SeniorCitizen   : Factor w/ 2 levels "No","Yes": 1 2 1 1 1 1 1 1 1 1 ...
##  $ Partner         : Factor w/ 2 levels "No","Yes": 2 1 1 2 1 1 2 2 2 1 ...
##  $ Dependents      : Factor w/ 2 levels "No","Yes": 2 1 1 1 1 1 1 2 1 1 ...
##  $ tenure          : int  60 26 3 16 2 4 72 72 31 6 ...
##  $ PhoneService    : Factor w/ 2 levels "No","Yes": 1 2 1 2 2 2 2 2 2 2 ...
##  $ MultipleLines   : Factor w/ 2 levels "No","Yes": 1 1 1 2 1 2 2 2 1 1 ...
##  $ InternetService : Factor w/ 3 levels "DSL","Fiber optic",..: 1 2 1 2 1 2 2 2 1 2 ...
##  $ OnlineSecurity  : Factor w/ 2 levels "No","Yes": 1 1 1 2 1 1 2 2 1 1 ...
##  $ OnlineBackup    : Factor w/ 2 levels "No","Yes": 2 1 2 2 1 1 1 2 1 2 ...
##  $ TechSupport     : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 2 1 2 1 ...
##  $ StreamingTV     : Factor w/ 2 levels "No","Yes": 2 1 2 1 1 1 2 2 1 2 ...
##  $ Contract        : Factor w/ 3 levels "Month-to-month",..: 1 1 1 1 1 1 3 3 1 1 ...
##  $ PaperlessBilling: Factor w/ 2 levels "No","Yes": 2 2 2 1 2 2 2 1 2 2 ...
##  $ PaymentMethod   : Factor w/ 4 levels "Bank transfer (automatic)",..: 3 3 3 3 4 3 2 2 4 3 ...
##  $ MonthlyCharges  : num  49.8 78.8 50.6 90.7 45.9 ...
##  $ Churn           : Factor w/ 2 levels "No","Yes": 1 1 2 1 1 2 1 1 1 2 ...
```

After splitting the data set in train and test we are now ready for data analysis and prediction algorithms. We are going to apply three different methods of analysis:

- Logistic regression
- Decision tree
- Random forests

We are going to compare results using the CONFUSION MATRIX:

```
PREDICTED VALUES    ACTUAL VALUES
```

………………. |	Positive |	Negative
Positive	TP	FP
Negative	FN	TN

- True Positive (TP): number of predictions where the classifier correctly predicts the positive class as positive.
- True Negative (TN): number of predictions where the classifier correctly predicts the negative class as negative.
- False Positive (FP): number of predictions where the classifier incorrectly predicts the negative class as positive.
- False Negative (FN): number of predictions where the classifier incorrectly predicts the positive class as negative.
- Accuracy: overall accuracy of the model, meaning the fraction of the total samples that were correctly classified by the classifier. Formula: (TP+TN)/(TP+TN+FP+FN).
- Sensitivity: fraction of all positive samples that were correctly predicted as positive by the classifier. Formula: TP/(TP+FN).
- Specificity: fraction of all negative samples that were correctly predicted as negative by the classifier. Formula: TN/(TN+FP).

Logistic regression

Logistic regression is a method for fitting a regression sigmoid curve, y = f(x), when y is a categorical variable. The typical use of this model predicts y given a set of predictors x. The predictors can be continuous, categorical or a mix of both. In our model the categorical variable Churn is binary meaning that it can assume either the value 1 or 0 (yes or no). Our predictors are a mix of categorical (all the rest) and continuous (tenure and monthly payment) variables.

```
#Logistic regression
fit_glm <- glm(Churn ~., data = train, family = "binomial")
summary(fit_glm)
```

```
##
## Call:
## glm(formula = Churn ~ ., family = "binomial", data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9119  -0.6751  -0.2857   0.7156   3.2023
##
## Coefficients:
##                                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)                         -0.906652   0.278961  -3.250 0.001154 **
## SeniorCitizenYes                     0.142406   0.102021   1.396 0.162759
## PartnerYes                          -0.007866   0.093927  -0.084 0.933259
## DependentsYes                       -0.109354   0.107878  -1.014 0.310733
## tenure                              -0.033944   0.002898 -11.712  < 2e-16 ***
## PhoneServiceYes                     -0.944528   0.216420  -4.364 1.28e-05 ***
```

```
## MultipleLinesYes                       0.138362   0.103608   1.335 0.181735
## InternetServiceFiber optic             0.529607   0.235610   2.248 0.024588 *
## InternetServiceNo                     -0.189463   0.271885  -0.697 0.485896
## OnlineSecurityYes                     -0.328464   0.107838  -3.046 0.002320 **
## OnlineBackupYes                       -0.241807   0.100556  -2.405 0.016186 *
## TechSupportYes                        -0.392658   0.113748  -3.452 0.000556 ***
## StreamingTVYes                         0.054829   0.141488   0.388 0.698372
## ContractOne year                      -0.716851   0.127938  -5.603 2.11e-08 ***
## ContractTwo year                      -1.490491   0.217533  -6.852 7.29e-12 ***
## PaperlessBillingYes                    0.385031   0.087827   4.384 1.17e-05 ***
## PaymentMethodCredit card (automatic)  0.042144   0.138450   0.304 0.760826
## PaymentMethodElectronic check          0.431274   0.113845   3.788 0.000152 ***
## PaymentMethodMailed check              0.136750   0.136868   0.999 0.317727
## MonthlyCharges                         0.020283   0.007971   2.545 0.010939 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 5702.8  on 4923  degrees of freedom
## Residual deviance: 4088.7  on 4904  degrees of freedom
## AIC: 4128.7
##
## Number of Fisher Scoring iterations: 6
```

```
p_hat_glm <- predict(fit_glm, test)
test_hat_glm <- factor(ifelse(p_hat_glm > 0.5, "Yes", "No"))
confusionMatrix(test_hat_glm, test$Churn)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   No  Yes
##        No  1477  363
##        Yes   71  197
##
##                Accuracy : 0.7941
##                  95% CI : (0.7762, 0.8112)
##     No Information Rate : 0.7343
##     P-Value [Acc > NIR] : 1.058e-10
##
##                   Kappa : 0.367
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9541
##             Specificity : 0.3518
##          Pos Pred Value : 0.8027
##          Neg Pred Value : 0.7351
##              Prevalence : 0.7343
##          Detection Rate : 0.7007
##    Detection Prevalence : 0.8729
##       Balanced Accuracy : 0.6530
##
```

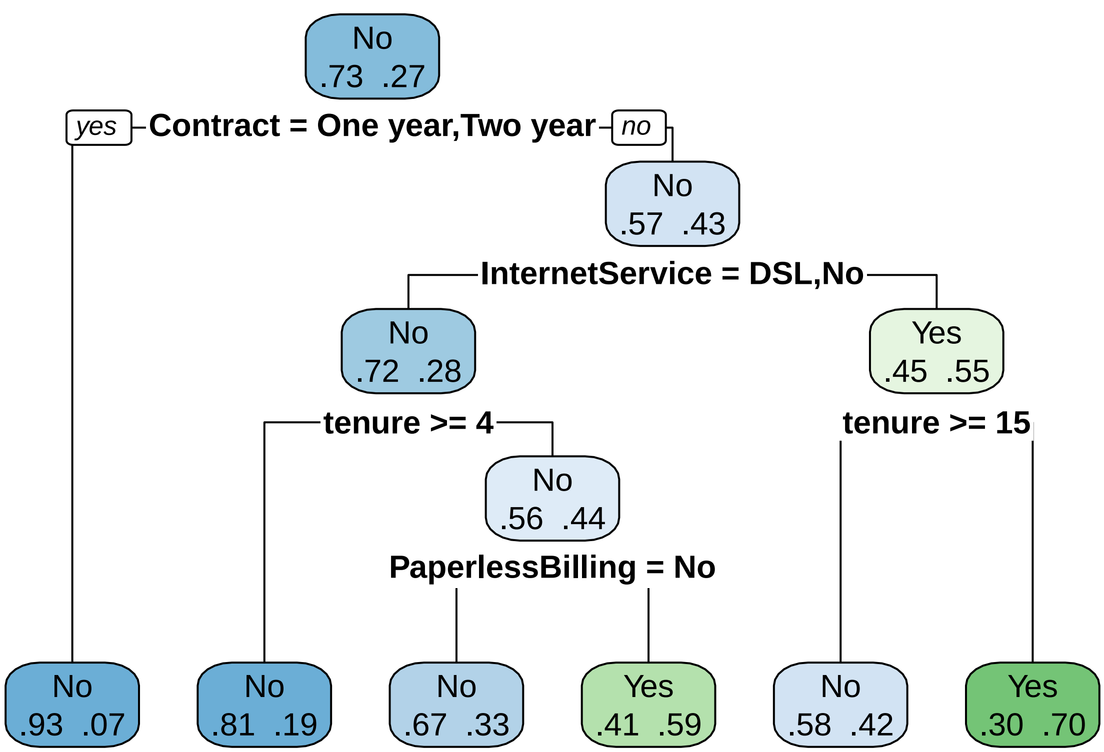
```
##        'Positive' Class : No
##
```

Examining the most significant p-values, we can identify the best predictors of churn based on this algorithm: tenure length, PhoneService yes, TechSupport yes, Contract one and two years yes, PaperlessBilling yes and PaymentMethodElectronic check. The confusion matrix returned an overall accuracy of 0.7941, a sensitivity of 0.9541 and a specificity of 0.3518. The machine predicted 197 customers leaving the company correctly and 363 incorrectly.

Decision tree

Decision tree is a technique for fitting non-linear models, it works performing binary splits on the recursive predictors mapping the possible outcomes of a series of related choices. In our model, the possible outcomes are Churn (yes or no) based on the client choices of different types of contract duration, payment, services, prices,etc..

```
#Decision tree
tr_fit <- rpart(Churn ~., data = train, method="class")
rpart.plot(tr_fit, extra = 4)
```




```
p_hat_tr <- predict(tr_fit, test)
test_hat_tr <- factor(ifelse(p_hat_tr[,2] > 0.5, "Yes", "No"))
confusionMatrix(test_hat_tr, test$Churn)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   No  Yes
##        No  1418  329
##        Yes  130  231
##
##                Accuracy : 0.7823
##                  95% CI : (0.764, 0.7997)
##     No Information Rate : 0.7343
##     P-Value [Acc > NIR] : 2.133e-07
##
##                   Kappa : 0.3705
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9160
##             Specificity : 0.4125
##          Pos Pred Value : 0.8117
##          Neg Pred Value : 0.6399
##              Prevalence : 0.7343
##          Detection Rate : 0.6727
##    Detection Prevalence : 0.8287
##       Balanced Accuracy : 0.6643
##
##        'Positive' Class : No
##
```

Examining the tree, we can identify the best predictors of churn based on this algorithm: Contract = one and two years yes, InternetService = DSL no, tenure longer than 4 months and PaperlessBilling = no. The confusion matrix returned an overall accuracy of 0.7823, a sensitivity of 0.9160 and a specificity of 0.4125. The machine predicted 231 customers leaving the company correctly and 329 incorrectly.

Random forests

Random forests are a type of ensemble method, a process in which numerous decision trees are randomly fitted and the results are combined for stronger prediction. Unfortunately, inference and explainability are limited with this algorithm.

```r
#Random forest
rf_fit <- randomForest(Churn ~., data = train)
varImp(rf_fit)
```

```
##                    Overall
## SeniorCitizen     30.11063
## Partner           32.77758
## Dependents        29.48514
## tenure           327.23600
## PhoneService      12.72266
## MultipleLines     29.48577
## InternetService  102.09039
## OnlineSecurity    35.54993
## OnlineBackup      35.99980
```

```
## TechSupport       39.25810
## StreamingTV       28.51440
## Contract         164.96381
## PaperlessBilling  44.47389
## PaymentMethod    113.03194
## MonthlyCharges   274.66483
```

```
confusionMatrix(predict(rf_fit, test), test$Churn)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   No  Yes
##        No  1398  275
##        Yes  150  285
##
##                Accuracy : 0.7984
##                  95% CI : (0.7806, 0.8153)
##     No Information Rate : 0.7343
##     P-Value [Acc > NIR] : 4.447e-12
##
##                   Kappa : 0.4436
##
##  Mcnemar's Test P-Value : 1.800e-09
##
##             Sensitivity : 0.9031
##             Specificity : 0.5089
##          Pos Pred Value : 0.8356
##          Neg Pred Value : 0.6552
##              Prevalence : 0.7343
##          Detection Rate : 0.6632
##    Detection Prevalence : 0.7936
##       Balanced Accuracy : 0.7060
##
##        'Positive' Class : No
##
```

Examining the overall most important variables we can identify: tenure, MonthlyCharges, Contract, InternetService, PayementMethod. The confusion matrix returned an overall accuracy of 0.7984, a sensitivity of 0.9031 and a specificity of 0.5089. The machine predicted 285 customers leaving the company correctly and 275 incorrectly.

Conclusion

Comparing models:

Type |	Logistic reg. |	Decision tree |	Random forests
accuracy	0.7941	0.7823	0.7984
sensitivity	0.9541	0.9160	0.9031
specificity	0.3518	0.4125	0.5089

After running three different models we can appreciate how the accuracy is very similar while sensitivity is better for Logistic regression, specificity is better for Random forests and the Decision tree is in the middle. We can advise the company that clients on month to month contract are the most likely to churn, they should try to offer better value for money in one or two year contracts to increase the number of long tenure contracts. The company may have some problems with fiber optic clients and with the ones without technical support. The company should work to provide more technical support and a better fiber optic service. The longer a client stays with the company the less is likely to leave.

Further investigation

We are interested to know if, by keeping all the variables in, we could improve our models and how much. Hence, we decide to repeat all the analysis and compare results.

```
#All variables models
#Simplify data cutting columns
newtelco <- telco %>%
  select(-customerID)

#Checking data
str(newtelco)
```

```
## 'data.frame':    7032 obs. of  20 variables:
##  $ gender          : Factor w/ 2 levels "Female","Male": 1 2 1 2 2 1 1 1 1 2 ...
##  $ SeniorCitizen   : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 2 1 1 1 1 ...
##  $ Partner         : Factor w/ 2 levels "No","Yes": 2 1 1 1 1 1 1 1 2 1 ...
##  $ Dependents      : Factor w/ 2 levels "No","Yes": 2 1 1 1 1 1 1 1 1 1 ...
##  $ tenure          : int  60 16 3 1 51 26 3 28 16 1 ...
##  $ PhoneService    : Factor w/ 2 levels "No","Yes": 1 2 2 2 2 2 1 2 2 2 ...
##  $ MultipleLines   : Factor w/ 2 levels "No","Yes": 1 2 1 1 2 1 1 1 2 2 ...
##  $ InternetService : Factor w/ 3 levels "DSL","Fiber optic",..: 1 2 3 2 2 2 1 1 2 1 ...
##  $ OnlineSecurity  : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 2 2 1 ...
##  $ OnlineBackup    : Factor w/ 2 levels "No","Yes": 2 1 1 1 2 1 2 2 2 2 ...
##  $ DeviceProtection: Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 2 1 ...
##  $ TechSupport     : Factor w/ 2 levels "No","Yes": 1 2 1 1 1 1 1 1 1 1 ...
##  $ StreamingTV     : Factor w/ 2 levels "No","Yes": 2 1 1 2 1 1 2 1 1 1 ...
##  $ StreamingMovies : Factor w/ 2 levels "No","Yes": 2 2 1 1 1 2 2 1 1 1 ...
##  $ Contract        : Factor w/ 3 levels "Month-to-month",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ PaperlessBilling: Factor w/ 2 levels "No","Yes": 2 2 1 2 1 2 2 1 1 1 ...
##  $ PaymentMethod   : Factor w/ 4 levels "Bank transfer (automatic)",..: 3 2 4 3 1 3 3 3 3 3 ...
##  $ MonthlyCharges  : num  49.8 89 20 79.5 81 ...
##  $ TotalCharges    : num  3069.4 1448.6 63.6 79.5 4085.8 ...
##  $ Churn           : Factor w/ 2 levels "No","Yes": 1 2 1 2 1 1 2 1 1 1 ...
```

```
#Splitting in train and test set
set.seed(2022, sample.kind = "Rounding")
```

```
## Warning in set.seed(2022, sample.kind = "Rounding"): non-uniform 'Rounding'
## sampler used
```

```
test_index <- createDataPartition(newtelco$Churn, times = 1, p = 0.7, list = FALSE)
train <- newtelco[test_index,]
test <- newtelco[-test_index,]
```

```r
#Checking data
str(train)
```

```
## 'data.frame':    4924 obs. of  20 variables:
##  $ gender          : Factor w/ 2 levels "Female","Male": 2 1 2 2 1 2 2 1 2 2 ...
##  $ SeniorCitizen   : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 2 1 1 ...
##  $ Partner         : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 2 1 1 2 ...
##  $ Dependents      : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 2 1 1 2 ...
##  $ tenure          : int  16 3 1 51 28 1 37 69 2 61 ...
##  $ PhoneService    : Factor w/ 2 levels "No","Yes": 2 2 2 2 2 2 2 2 2 1 ...
##  $ MultipleLines   : Factor w/ 2 levels "No","Yes": 2 1 1 2 1 2 1 2 1 1 ...
##  $ InternetService : Factor w/ 3 levels "DSL","Fiber optic",..: 2 3 2 2 1 1 2 2 3 1 ...
##  $ OnlineSecurity  : Factor w/ 2 levels "No","Yes": 1 1 1 1 2 1 1 2 1 2 ...
##  $ OnlineBackup    : Factor w/ 2 levels "No","Yes": 1 1 1 2 2 2 1 1 1 1 ...
##  $ DeviceProtection: Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 2 1 2 ...
##  $ TechSupport     : Factor w/ 2 levels "No","Yes": 2 1 1 1 1 1 1 2 1 1 ...
##  $ StreamingTV     : Factor w/ 2 levels "No","Yes": 1 1 2 1 1 1 2 2 1 1 ...
##  $ StreamingMovies : Factor w/ 2 levels "No","Yes": 2 1 1 1 1 1 2 2 1 1 ...
##  $ Contract        : Factor w/ 3 levels "Month-to-month",..: 1 1 1 1 1 1 2 3 1 1 ...
##  $ PaperlessBilling: Factor w/ 2 levels "No","Yes": 2 1 2 1 1 1 1 1 1 1 ...
##  $ PaymentMethod   : Factor w/ 4 levels "Bank transfer (automatic)",..: 2 4 3 1 3 3 2 2 4 1 ...
##  $ MonthlyCharges  : num  89 20 79.5 81 56 ...
##  $ TotalCharges    : num  1448.6 63.6 79.5 4085.8 1522.7 ...
##  $ Churn           : Factor w/ 2 levels "No","Yes": 2 1 2 1 1 1 1 1 1 1 ...
```

```r
str(test)
```

```
## 'data.frame':    2108 obs. of  20 variables:
##  $ gender          : Factor w/ 2 levels "Female","Male": 1 1 1 1 1 2 1 1 1 2 ...
##  $ SeniorCitizen   : Factor w/ 2 levels "No","Yes": 1 2 1 1 1 1 1 1 1 1 ...
##  $ Partner         : Factor w/ 2 levels "No","Yes": 2 1 1 2 1 1 2 2 2 1 ...
##  $ Dependents      : Factor w/ 2 levels "No","Yes": 2 1 1 1 1 1 1 2 1 1 ...
##  $ tenure          : int  60 26 3 16 2 4 72 72 31 6 ...
##  $ PhoneService    : Factor w/ 2 levels "No","Yes": 1 2 1 2 2 2 2 2 2 2 ...
##  $ MultipleLines   : Factor w/ 2 levels "No","Yes": 1 1 1 2 1 2 2 2 1 1 ...
##  $ InternetService : Factor w/ 3 levels "DSL","Fiber optic",..: 1 2 1 2 1 2 2 2 1 2 ...
##  $ OnlineSecurity  : Factor w/ 2 levels "No","Yes": 1 1 1 2 1 1 2 2 1 1 ...
##  $ OnlineBackup    : Factor w/ 2 levels "No","Yes": 2 1 2 2 1 1 1 2 1 2 ...
##  $ DeviceProtection: Factor w/ 2 levels "No","Yes": 1 1 1 2 1 1 2 2 1 1 ...
##  $ TechSupport     : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 2 1 2 1 ...
##  $ StreamingTV     : Factor w/ 2 levels "No","Yes": 2 1 2 1 1 1 2 2 1 2 ...
##  $ StreamingMovies : Factor w/ 2 levels "No","Yes": 2 2 2 1 1 2 2 2 1 1 ...
##  $ Contract        : Factor w/ 3 levels "Month-to-month",..: 1 1 1 1 1 1 3 3 1 1 ...
##  $ PaperlessBilling: Factor w/ 2 levels "No","Yes": 2 2 2 1 2 2 2 1 2 2 ...
##  $ PaymentMethod   : Factor w/ 4 levels "Bank transfer (automatic)",..: 3 3 3 3 4 3 2 2 4 3 ...
##  $ MonthlyCharges  : num  49.8 78.8 50.6 90.7 45.9 ...
##  $ TotalCharges    : num  3069 2006 155 1375 106 ...
##  $ Churn           : Factor w/ 2 levels "No","Yes": 1 1 2 1 1 2 1 1 1 2 ...
```

```r
#Logistic regression
fit_glm <- glm(Churn ~., data = train, family = "binomial")
summary(fit_glm)
```

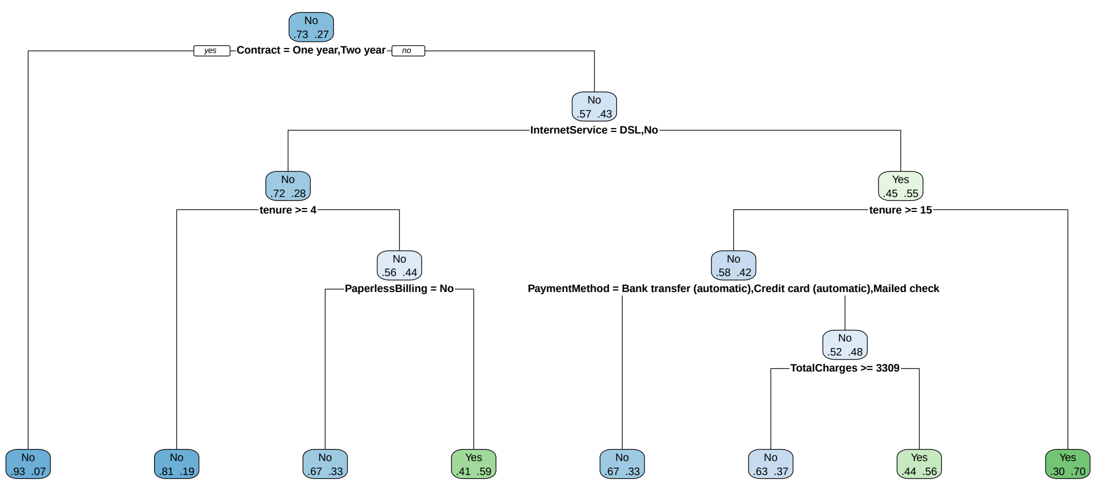
```
##
## Call:
## glm(formula = Churn ~ ., family = "binomial", data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9053  -0.6821  -0.2740   0.7355   3.4635
##
## Coefficients:
##                                        Estimate Std. Error z value Pr(>|z|)
## (Intercept)                           2.079e+00  9.854e-01   2.110 0.034861 *
## genderMale                            2.162e-02  7.767e-02   0.278 0.780737
## SeniorCitizenYes                      1.365e-01  1.016e-01   1.344 0.179039
## PartnerYes                           -9.970e-03  9.427e-02  -0.106 0.915769
## DependentsYes                        -9.438e-02  1.080e-01  -0.874 0.382328
## tenure                               -6.161e-02  7.722e-03  -7.978 1.48e-15 ***
## PhoneServiceYes                       1.008e+00  7.813e-01   1.290 0.196911
## MultipleLinesYes                      5.983e-01  2.129e-01   2.811 0.004941 **
## InternetServiceFiber optic            2.982e+00  9.658e-01   3.087 0.002020 **
## InternetServiceNo                    -2.808e+00  9.740e-01  -2.883 0.003944 **
## OnlineSecurityYes                     1.448e-01  2.151e-01   0.673 0.500742
## OnlineBackupYes                       2.185e-01  2.129e-01   1.026 0.304709
## DeviceProtectionYes                   2.985e-01  2.137e-01   1.397 0.162480
## TechSupportYes                        8.265e-02  2.185e-01   0.378 0.705231
## StreamingTVYes                        9.923e-01  3.933e-01   2.523 0.011639 *
## StreamingMoviesYes                    1.112e+00  3.947e-01   2.818 0.004837 **
## ContractOne year                     -7.119e-01  1.293e-01  -5.505 3.69e-08 ***
## ContractTwo year                     -1.496e+00  2.200e-01  -6.798 1.06e-11 ***
## PaperlessBillingYes                   3.799e-01  8.825e-02   4.305 1.67e-05 ***
## PaymentMethodCredit card (automatic)  4.285e-02  1.387e-01   0.309 0.757433
## PaymentMethodElectronic check         4.141e-01  1.140e-01   3.632 0.000281 ***
## PaymentMethodMailed check             8.809e-02  1.384e-01   0.637 0.524363
## MonthlyCharges                       -8.454e-02  3.837e-02  -2.203 0.027566 *
## TotalCharges                          3.471e-04  8.709e-05   3.985 6.74e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 5702.8  on 4923  degrees of freedom
## Residual deviance: 4060.7  on 4900  degrees of freedom
## AIC: 4108.7
##
## Number of Fisher Scoring iterations: 6
```

```r
p_hat_glm <- predict(fit_glm, test)
test_hat_glm <- factor(ifelse(p_hat_glm > 0.5, "Yes", "No"))
confusionMatrix(test_hat_glm, test$Churn)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   No  Yes
##        No  1475  368
```

```
##        Yes   73  192
##
##                Accuracy : 0.7908
##                  95% CI : (0.7728, 0.808)
##     No Information Rate : 0.7343
##     P-Value [Acc > NIR] : 1.062e-09
##
##                   Kappa : 0.3555
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9528
##             Specificity : 0.3429
##          Pos Pred Value : 0.8003
##          Neg Pred Value : 0.7245
##              Prevalence : 0.7343
##          Detection Rate : 0.6997
##    Detection Prevalence : 0.8743
##       Balanced Accuracy : 0.6478
##
##        'Positive' Class : No
##
```

```r
#Decision tree
tr_fit <- rpart(Churn ~., data = train, method="class")
rpart.plot(tr_fit, extra = 4)
```



```r
p_hat_tr <- predict(tr_fit, test)
test_hat_tr <- factor(ifelse(p_hat_tr[,2] > 0.5, "Yes", "No"))
confusionMatrix(test_hat_tr, test$Churn)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   No  Yes
##        No  1368  271
##        Yes  180  289
##
##                Accuracy : 0.7861
##                  95% CI : (0.7679, 0.8034)
##     No Information Rate : 0.7343
##     P-Value [Acc > NIR] : 2.256e-08
##
##                   Kappa : 0.4217
##
##  Mcnemar's Test P-Value : 2.256e-05
##
##             Sensitivity : 0.8837
##             Specificity : 0.5161
##          Pos Pred Value : 0.8347
##          Neg Pred Value : 0.6162
##              Prevalence : 0.7343
##          Detection Rate : 0.6490
##    Detection Prevalence : 0.7775
##       Balanced Accuracy : 0.6999
##
##        'Positive' Class : No
##
```

```r
#Random forest
rf_fit <- randomForest(Churn ~., data = train)
varImp(rf_fit)
```

```
##                    Overall
## gender            40.99121
## SeniorCitizen     34.13565
## Partner           36.26765
## Dependents        30.60441
## tenure           288.57386
## PhoneService      12.24272
## MultipleLines     32.16322
## InternetService   98.63323
## OnlineSecurity    35.51974
## OnlineBackup      36.32890
## DeviceProtection  31.80210
## TechSupport       35.54769
## StreamingTV       28.97954
## StreamingMovies   31.02616
## Contract         162.26664
## PaperlessBilling  48.42953
```

```
## PaymentMethod    114.93050
## MonthlyCharges   302.58209
## TotalCharges     315.81630
```

```
confusionMatrix(predict(rf_fit, test), test$Churn)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   No  Yes
##        No  1403  280
##        Yes  145  280
##
##                Accuracy : 0.7984
##                  95% CI : (0.7806, 0.8153)
##     No Information Rate : 0.7343
##     P-Value [Acc > NIR] : 4.447e-12
##
##                   Kappa : 0.4402
##
##  Mcnemar's Test P-Value : 8.034e-11
##
##             Sensitivity : 0.9063
##             Specificity : 0.5000
##          Pos Pred Value : 0.8336
##          Neg Pred Value : 0.6588
##              Prevalence : 0.7343
##          Detection Rate : 0.6656
##    Detection Prevalence : 0.7984
##       Balanced Accuracy : 0.7032
##
##        'Positive' Class : No
##
```

Comparing models:

Type |	Logistic reg. |	Decision tree |	Random forests
accuracy	0.7941	0.7823	0.7984
sensitivity	0.9541	0.9160	0.9031
specificity	0.3518	0.4125	0.5089
accuracy	0.7908	0.7861	0.7984 all variables
sensitivity	0.9528	0.8837	0.9063 all variables
specificity	0.3429	0.5161	0.5000 all variables

Including all variables has not improved our results much, it slightly improved Decision tree specificity thanks to a more complex render of the tree against less sensitivity. It would be interesting to try again reducing the number of variables, keeping only 5 to 7 of them, and see the effect.

Reference

- Rafael A. Irizarry - Introduction to data science

- Andrea De Mauro - Big data analytics
- Jared P. Lander - R for everyone