Churn

Alejandro Greco

10/10/2019

Contents

1 Introduction

This project has the objective to make a recommendation system to reduce the effort of loyalty actions beeing more efficient by increasing the precision of the actions based on churn data of a telecommunication company provided by kaggle.

The objective is to obtain the lower possible Root Mean Squared Error (RMSE), lower than 0.40, using the data provided divided into a training set and validation set. The original data content is about 7 thousand registers with 21 variables but after the cleaning process removing the null values is about 6.5 thousand and 2 new variables were added to transform continues values of money to categorical. This data was divided into 2 data set, one with 90% of the data named "churn_data" with about 5.9 thousand observations and other named "validation" with about 0.6 thousand registers both with 23 columns with not null values.

2 Methods

In this project was evaluated 5 models to make the best recommendation possible with the lower value in loss function Root mean squared error (RMSE):

1 - Naive: A model using only the average churn;

2 - Contracts: A model using model 1 adding Contracts type of is month to month, one year or two years. because is impacting the churn;

3 - Total Charges: A model using model 2 adding Total Charges because some users could spend extra of the monthly charges and we could guess that a user that spend more on the company's products could reflect their preferences, loyalty and reduce the probability of churn;

4 - Monthly Charges: A model using model 3 adding the monthly charges of users. This is a reflection of the users' preferences, loyalty, and the market because is the amount of money the users are willing to pay for the company's services;

5 - Tenure: A model using model 4 adding the tenure of the user. I considered this variable super important because could be used as a trigger to take loyalty actions.

Using Root mean squared error (RMSE) will would compare the efficiency of the models.

2.1 Data

The data is from kaggle, please log in with your account and download it with this link:

https://www.kaggle.com/blastchar/telco-customer-churn/download.

This file must be unzipped and move it into your working directory that could be found with "getwd()".

```
#getwd()
```

```
################################
# Create churn
################################

# Note: this process could take a couple of minutes

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

```
## Loading required package: tidyverse
```

```
##   Attaching packages                                           tidyverse 1.3.0
```

```
##   ggplot2 3.2.1      purrr   0.3.3
##   tibble  2.1.3      dplyr   0.8.3
```

```
##   tidyr   1.0.0       stringr 1.4.0
##   readr   1.3.1       forcats 0.4.0
```

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

```
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
```

```
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
```

```
# you must download the file from https://www.kaggle.com/blastchar/telco-customer-churn/download unzipp

churn <- read.csv("WA_Fn-UseC_-Telco-Customer-Churn.csv")
```

2.2 Requirements

Installation of these packages are required for this project:

```
# Packages

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

```
## Loading required package: bazar
```

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

```
## Loading required package: car
```

```
## Loading required package: carData
```

```
##
## Attaching package: 'car'
```

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

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

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

```
## Loading required package: corrplot
```

```
## corrplot 0.84 loaded
```

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

```
## Loading required package: ggthemes
```

```
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
```

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

```
## Loading required package: Hmisc
```

```
## Loading required package: survival
```

```
##
## Attaching package: 'survival'
```

```
## The following object is masked from 'package:caret':
##
##     cluster
```

```
## Loading required package: Formula
```

```
##
## Attaching package: 'Hmisc'
```

```
## The following object is masked from 'package:bazar':
##
##     %nin%
```

```
## The following objects are masked from 'package:dplyr':
##
##     src, summarize
```

```
## The following objects are masked from 'package:base':
##
##     format.pval, units
```

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

```
## Loading required package: kableExtra
```

```
##
## Attaching package: 'kableExtra'
```

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

```
##     group_rows
```

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

```
## Loading required package: rapportools
```

```
## Loading required package: reshape
```

```
##
## Attaching package: 'reshape'
```

```
## The following object is masked from 'package:data.table':
##
##     melt
```

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

```
## The following objects are masked from 'package:tidyr':
##
##     expand, smiths
```

```
##
## Attaching package: 'rapportools'
```

```
## The following objects are masked from 'package:Hmisc':
##
##     label, label<-
```

```
## The following object is masked from 'package:bazar':
##
##     is.empty
```

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

```
## The following objects are masked from 'package:stats':
##
##     IQR, median, sd, var
```

```
## The following objects are masked from 'package:base':
##
##     max, mean, min, range, sum
```

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

```
## Loading required package: rmarkdown
```

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

```
## Loading required package: tinytex
```

2.3 Data cleaning

In the table below you can observe the head of the data set.

```
kable(head(churn))
```

customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetS
7590-VHVEG	Female	0	Yes	No	1	No	No phone service	DSL
5575-GNVDE	Male	0	No	No	34	Yes	No	DSL
3668-QPYBK	Male	0	No	No	2	Yes	No	DSL
7795-CFOCW	Male	0	No	No	45	No	No phone service	DSL
9237-HQITU	Female	0	No	No	2	Yes	No	Fiber opt
9305-CDSKC	Female	0	No	No	8	Yes	Yes	Fiber opt

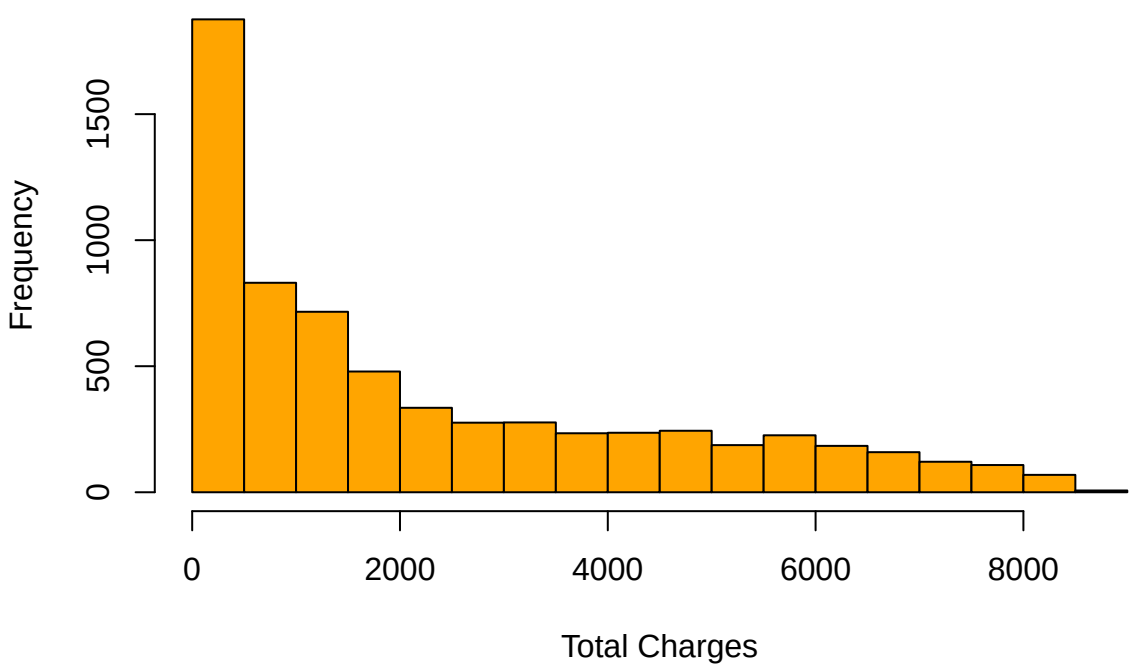
First, it is suggested to clean the data before creating the training and validation set to save time and because is more efficient. The changes required are:

1 - The objective variable is Churn, and it will be transformed into numeric. Yes for 1 and No made churn for 0.

2 - All the "no ..." will be transformed to "no" because for this analysis if the person does not have phone service that means that it does not have multiple lines. The same occurred with all the other variables bellow.

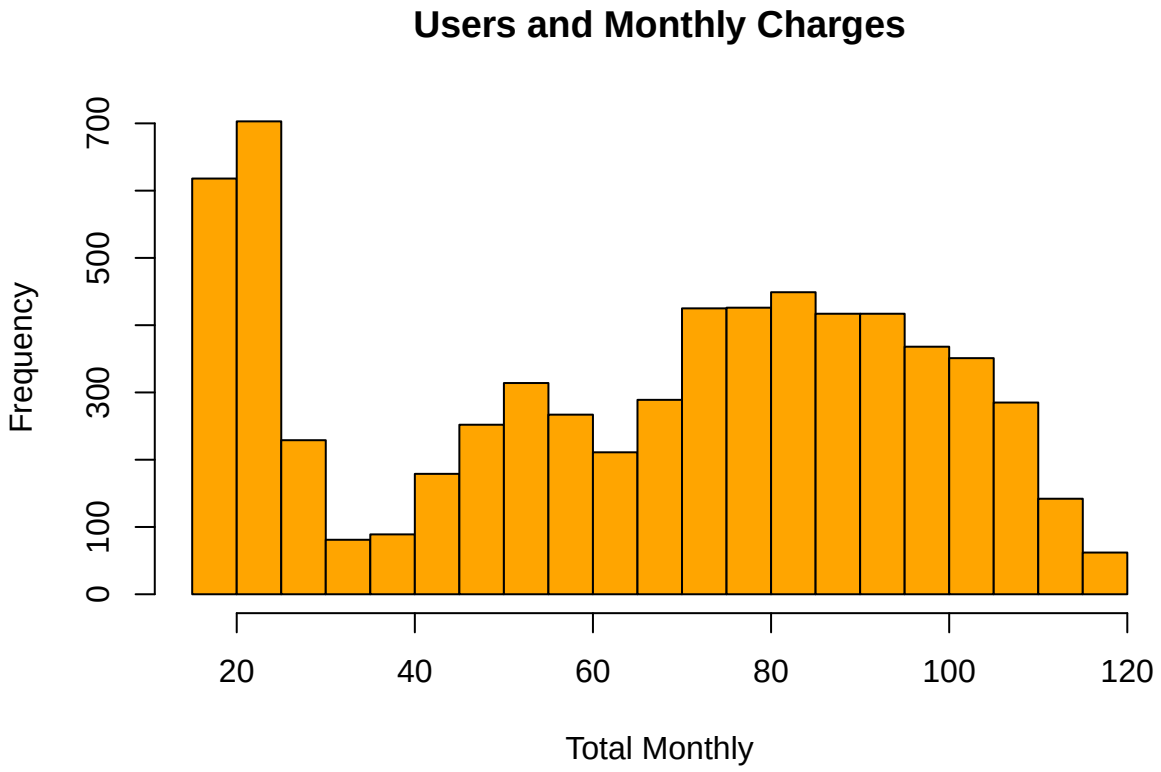
Two columns will be added to categorize the monetary values using Scott's breaks.

```
#The objective variable is Churn, and it will be transformed into numeric.

churn$Churn <- as.character(churn$Churn)
churn$Churn[churn$Churn == "Yes"] <- 1
churn$Churn[churn$Churn == "No"] <- 0
churn$Churn <- as.numeric(churn$Churn)

# This is to correct some text data but first, we need to change the columns to act as characters.

churn$MultipleLines <- as.character(churn$MultipleLines)
churn$OnlineSecurity <- as.character(churn$OnlineSecurity)
churn$OnlineBackup <- as.character(churn$OnlineBackup)
churn$DeviceProtection <- as.character(churn$DeviceProtection)
churn$TechSupport <- as.character(churn$TechSupport)
churn$StreamingTV <- as.character(churn$StreamingTV)

#All the "no ..." will be transformed into "no" because for this analysis if the person does not have p

churn$MultipleLines[churn$MultipleLines == "No phone service"] <- "No"
churn$OnlineSecurity[churn$OnlineSecurity == "No internet service"] <- "No"
churn$OnlineBackup[churn$OnlineBackup == "No internet service"] <- "No"
churn$DeviceProtection[churn$DeviceProtection == "No internet service"] <- "No"
churn$TechSupport[churn$TechSupport == "No internet service"] <- "No"
churn$StreamingTV[churn$StreamingTV == "No internet service"] <- "No"

#There are two columns that will be added based on continues values (monetary) into categorical using S

#To categorized total charges, the breaks used are 19 as you would observe in the next table.
kable(summary(
              hist(churn$TotalCharges, breaks = "Scott", col='Orange', main="Users and Total Charges", 
      )      )
```



	Length	Class	Mode
breaks	19	-none-	numeric
counts	18	-none-	numeric
density	18	-none-	numeric
mids	18	-none-	numeric
xname	1	-none-	character
equidist	1	-none-	logical

```
#This is to create class size

lower_total<-min(churn$TotalCharges)
upper_total<-max(churn$TotalCharges)

classsize<-(upper_total-lower_total)/19

#This is the sequences using the class size
break_seq<-seq(lower_total, upper_total, classsize)

#Finally this adds a new variable with the category to the data set.

churn$TotalChargesCategory <- cut(churn$TotalCharges, breaks = break_seq , labels=1:19)

#The other variable to classify is Monthly Charges. In this case, we need 22 Scott breaks as you could

kable(summary(
              hist(churn$MonthlyCharges, breaks = "Scott", col='Orange', main="Users and Monthly Charges
              ))
```



	Length	Class	Mode
breaks	22	-none-	numeric
counts	21	-none-	numeric
density	21	-none-	numeric
mids	21	-none-	numeric
xname	1	-none-	character
equidist	1	-none-	logical

```
#This is to create class size

lower_monthly<-min(churn$MonthlyCharges)
upper_monthly<-max(churn$MonthlyCharges)

classsize_monthly <- (upper_monthly-lower_monthly)/22

#This is the sequences using the class size

break_seq_monthly <- seq(lower_monthly, upper_monthly, classsize_monthly)

#Finally this adds a new variable with the category to the data set.

churn$MonthlyChargesCategory <- cut(churn$MonthlyCharges, breaks = break_seq_monthly , labels=1:22)

#This is to remove na or null values.

churn <- churn[complete.cases(churn), ]

# This is to split the data into churn_data and validation set 90% and 10% respectively.

set.seed(1)
```

```
test_index <- createDataPartition(y = churn$Churn, times = 1, p = 0.1, list = FALSE)
churn_data <- churn[-test_index,]
validation <- churn[test_index,]

rm( test_index)
```

2.4 Data exploration and visualization

This data has 7,043 rows originally and after remove the null values was 6,563 that was divided it in 90% (5,906 registers) for training the model and 10% (657 registers) to validate it. In the training set there are 2,913 female and 2,993 male; partner No 3,059 and Yes 2,847; dependents No 4,132 and Yes 1,774; tenure minimum 1, maximum 72, mean 32.23, and median 29; Contract Month-to-month 3,262 , One year 1,250 and Two year 1,394; MonthlyCharges minimum 18.70, median 70.40, mean 64.68 and maximum 118.75; TotalCharges minimum 18.85 Median 1,389.72 Mean 2,270.84 and maximum 8,684.80; and churn with a mean of 26.58%. You can observe this information in the table below.

```
original_users <- count(churn)
original_users
```

```
## # A tibble: 1 x 1
##       n
##   <int>
## 1  6563
```

```
after_clean_useres <- count(churn_data)
after_clean_useres
```

```
## # A tibble: 1 x 1
##       n
##   <int>
## 1  5906
```

```
users_in_validateion <- count(validation)
users_in_validateion
```

```
## # A tibble: 1 x 1
##       n
##   <int>
## 1   657
```

```
summary(churn_data)
```

```
##       customerID      gender     SeniorCitizen    Partner    Dependents
##  0002-ORFBO:   1   Female:2913   Min.   :0.0000   No :3059   No :4132
##  0003-MKNFE:   1   Male  :2993   1st Qu.:0.0000   Yes:2847   Yes:1774
##  0004-TLHLJ:   1                 Median :0.0000
##  0011-IGKFF:   1                 Mean   :0.1637
##  0013-EXCHZ:   1                 3rd Qu.:0.0000
##  0013-MHZWF:   1                 Max.   :1.0000
##  (Other)   :5900
##      tenure      PhoneService MultipleLines         InternetService
##  Min.   : 1.00   No : 563     Length:5906        DSL        :2016
##  1st Qu.: 9.00   Yes:5343     Class :character   Fiber optic:2602
##  Median :29.00                Mode  :character   No         :1288
##  Mean   :32.25
##  3rd Qu.:55.00
```

```
##  Max.   :72.00
##
##  OnlineSecurity     OnlineBackup       DeviceProtection   TechSupport
##  Length:5906        Length:5906        Length:5906        Length:5906
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##  StreamingTV                   StreamingMovies           Contract
##  Length:5906        No                 :2319   Month-to-month:3262
##  Class :character   No internet service:1288   One year      :1250
##  Mode  :character   Yes                :2299   Two year      :1394
##
##
##
##
##  PaperlessBilling                   PaymentMethod  MonthlyCharges
##  No :2401         Bank transfer (automatic):1286   Min.   : 18.70
##  Yes:3505         Credit card (automatic)  :1272   1st Qu.: 35.40
##                   El                       :   0   Median : 70.40
##                   Electronic check         :1996   Mean   : 64.68
##                   Mailed check             :1352   3rd Qu.: 89.85
##                                                    Max.   :118.75
##
##   TotalCharges         Churn        TotalChargesCategory MonthlyChargesCategory
##  Min.   :  18.85   Min.   :0.0000   1      :1647         1      :1000
##  1st Qu.: 389.84   1st Qu.:0.0000   2      : 706         14     : 409
##  Median :1389.72   Median :0.0000   3      : 592         13     : 371
##  Mean   :2270.84   Mean   :0.2658   4      : 459         15     : 371
##  3rd Qu.:3765.15   3rd Qu.:1.0000   5      : 296         2      : 354
##  Max.   :8684.80   Max.   :1.0000   6      : 246         12     : 350
##                                     (Other):1960         (Other):3051
```

```
kable(head(churn_data))
```

customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetServi
7590-VHVEG	Female	0	Yes	No	1	No	No	DSL
5575-GNVDE	Male	0	No	No	34	Yes	No	DSL
3668-QPYBK	Male	0	No	No	2	Yes	No	DSL
7795-CFOCW	Male	0	No	No	45	No	No	DSL
9237-HQITU	Female	0	No	No	2	Yes	No	Fiber optic
9305-CDSKC	Female	0	No	No	8	Yes	Yes	Fiber optic

```
which(is.na(churn_data$TotalCharges))
```

```
## integer(0)
```

In the data, each row represents a customer, each column contains customer's attributes includes information about:

1 - Customers who left within the last month – the column is called Churn.

2 - Services that each customer has signed up for – phone, multiple lines, internet, online security, online backup, device protection, tech support, and streaming TV and movies.

3 - Customer account information – how long they have been a customer, contract, payment method, paperless billing, monthly charges, and total charges.

4 - Demographic info about customers – gender, age range, and if they have partners and dependents.

The columns information form the original data:

1 - customerID : Customer ID;

2 - gender : Whether the customer is a male or a female;

3 - SeniorCitizen : Whether the customer is a senior citizen or not (1, 0);

4 - Partner : Whether the customer has a partner or not (Yes, No);

5 - Dependents : Whether the customer has dependents or not (Yes, No);

6 - tenure : Number of months the customer has stayed with the company;

7 - PhoneService : Whether the customer has a phone service or not (Yes, No);

8 - MultipleLines : Whether the customer has multiple lines or not (Yes, No, No phone service);

9 - InternetService : Customer's internet service provider (DSL, Fiber optic, No);

10 - OnlineSecurity : Whether the customer has online security or not (Yes, No, No internet service);

11 - OnlineBackup : Whether the customer has online backup or not (Yes, No, No internet service);

12 - DeviceProtection : Whether the customer has device protection or not (Yes, No, No internet service);

13 - TechSupport : Whether the customer has tech support or not (Yes, No, No internet service);

14 - StreamingTV : Whether the customer has streaming TV or not (Yes, No, No internet service);

15 - StreamingMovies : Whether the customer has streaming movies or not (Yes, No, No internet service);

16 - Contract : The contract term of the customer (Month-to-month, One year, Two years);

17 - PaperlessBilling : Whether the customer has paperless billing or not (Yes, No);

18 - PaymentMethod : The customer's payment method (Electronic check, Mailed check, Bank transfer (automatic), Credit card (automatic));

19 - MonthlyCharges : The amount charged to the customer monthly;

20 - TotalCharges : The total amount charged to the customer;

21 - Churn : Whether the customer churned or not (Yes or No);

2.4.1 Demographic

Exploring the data we could observe that some variables have an impact on the churn. For example, looks like gender, senior citizen, partner, dependents have influence in the churn. In this table, reflect that a female senior with no partner and no dependents has an average churn of about 50% vs no senior female with partner and dependents that are about 15%.

```
churn_gender<-churn_data %>%
              group_by(gender,SeniorCitizen,Partner,Dependents) %>%
              summarise(churn=sum(Churn),
                        avg_churn=mean(Churn),
                        sd_churn=sd(Churn)
                       )

kable(churn_gender)
```

gender	SeniorCitizen	Partner	Dependents	churn	avg_churn	sd_churn
Female	0	No	No	363	0.3229537	0.4678133
Female	0	No	Yes	28	0.2258065	0.4198085
Female	0	Yes	No	89	0.1823770	0.3865512
Female	0	Yes	Yes	105	0.1502146	0.3575375
Female	1	No	No	132	0.4907063	0.5008454
Female	1	No	Yes	1	0.3333333	0.5773503
Female	1	Yes	No	59	0.3470588	0.4774410
Female	1	Yes	Yes	6	0.1666667	0.3779645
Male	0	No	No	350	0.3030303	0.4597673
Male	0	No	Yes	35	0.2000000	0.4011478
Male	0	Yes	No	112	0.2343096	0.4240103
Male	0	Yes	Yes	92	0.1321839	0.3389342
Male	1	No	No	97	0.4754902	0.5006274
Male	1	No	Yes	1	0.2000000	0.4472136
Male	1	Yes	No	89	0.3647541	0.4823506
Male	1	Yes	Yes	11	0.3055556	0.4671766

2.4.2 Tenure

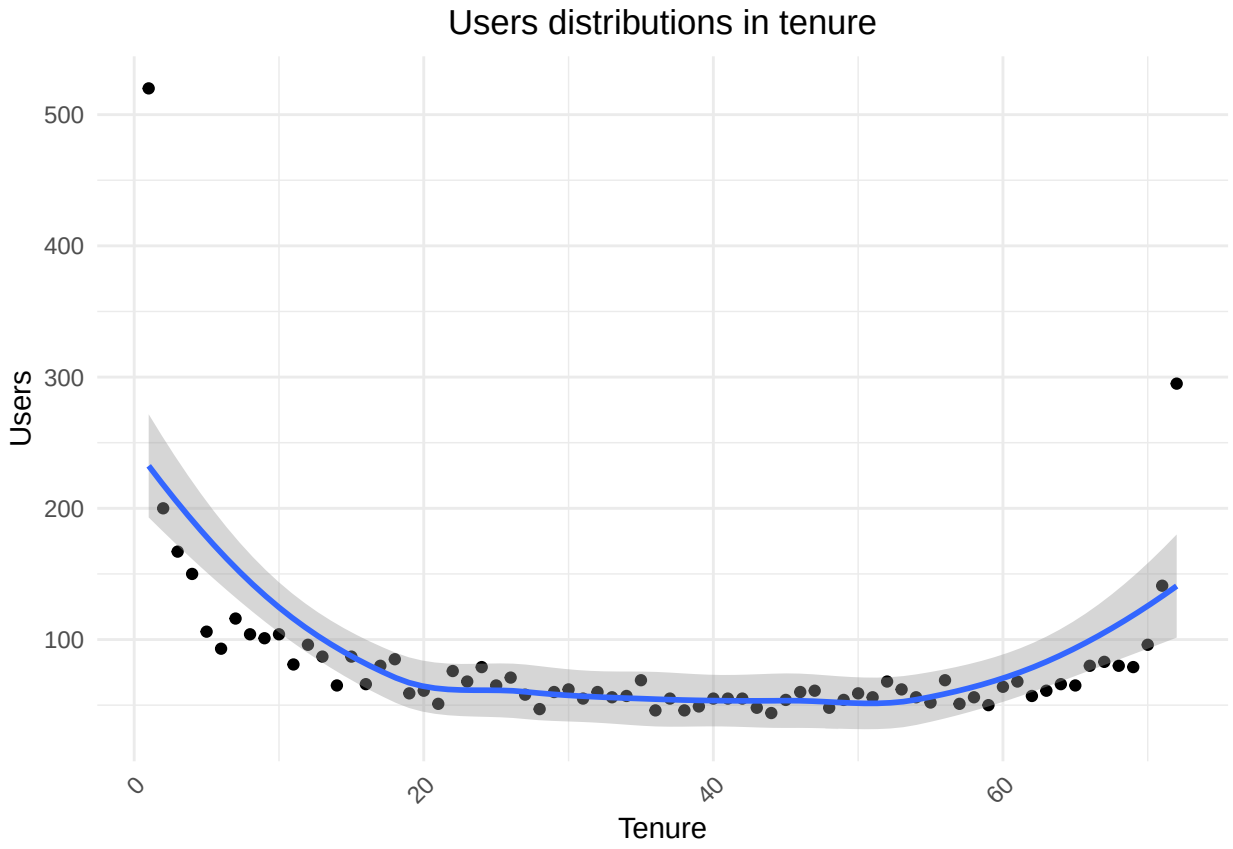
In the first month, the average churn is above 61%, but when the tenure is in the tenth month, the average churn is close to 39%. In these graphics illustrated the tendency behavior affected by the tenure. This information is represented in the table below. Overall, churn tended to decrease against increase the tenure as you could observe in the graphic below.

```
# Here is the churn by tenure.
churn_tenure <- churn_data %>%
              group_by(tenure) %>%
              summarise(count_customers=n(customerID),
                        churn=sum(Churn),
                        avg_churn=mean(Churn),
                        sd_churn=sd(Churn),
                        avg_monthly=mean(MonthlyCharges),
                        min_monthly=min(MonthlyCharges),
                        max_monthly=max(MonthlyCharges),
              )

kable(churn_tenure)
```

tenure	count_customers	churn	avg_churn	sd_churn	avg_monthly	min_monthly	max_monthly
1	520	318	0.6115385	0.4878698	50.51817	18.85	102.45
2	200	101	0.5050000	0.5012296	57.05150	18.75	104.40
3	167	76	0.4550898	0.4994767	57.61587	18.80	107.95
4	150	70	0.4666667	0.5005590	57.16800	18.85	105.65
5	106	51	0.4811321	0.5020175	59.19670	19.25	105.30
6	93	34	0.3655914	0.4842059	56.50699	18.95	109.90
7	116	45	0.3879310	0.4893927	59.48405	18.95	101.95
8	104	37	0.3557692	0.4810641	58.19760	19.20	105.50
9	101	39	0.3861386	0.4892913	62.15644	18.95	103.10
10	104	41	0.3942308	0.4910514	57.51298	18.85	110.10
11	81	24	0.2962963	0.4594683	55.92407	19.25	111.40
12	96	31	0.3229167	0.4700457	57.29948	19.00	112.95
13	87	31	0.3563218	0.4816882	58.44885	18.80	106.90
14	65	18	0.2769231	0.4509605	62.43308	18.80	104.85
15	87	32	0.3678161	0.4850064	62.20115	19.40	105.35
16	66	24	0.3636364	0.4847319	62.65758	18.95	100.70
17	80	26	0.3250000	0.4713299	63.37500	19.15	104.20
18	85	20	0.2352941	0.4266999	57.35353	19.00	101.30
19	59	16	0.2711864	0.4483882	57.92627	19.65	106.60
20	61	16	0.2622951	0.4435328	60.17459	18.90	108.20
21	51	14	0.2745098	0.4507075	65.50784	19.60	111.20
22	76	24	0.3157895	0.4679181	63.98553	19.60	104.60
23	68	9	0.1323529	0.3413936	60.86691	19.60	106.40
24	79	21	0.2658228	0.4445932	60.91329	19.70	104.65
25	65	18	0.2769231	0.4509605	58.61462	19.05	108.90
26	71	13	0.1830986	0.3895000	64.16408	19.15	105.75
27	58	9	0.1551724	0.3652312	61.01293	19.15	110.50
28	47	12	0.2553191	0.4407545	70.64894	19.55	110.85
29	60	13	0.2166667	0.4154502	60.42500	19.40	103.95
30	62	12	0.1935484	0.3983042	68.10081	19.05	110.45
31	55	13	0.2363636	0.4287638	73.54091	19.55	104.35
32	60	17	0.2833333	0.4544196	73.40083	18.95	109.55
33	56	11	0.1964286	0.4008919	64.64464	19.45	110.45
34	57	11	0.1929825	0.3981473	68.25175	19.60	116.15
35	69	13	0.1884058	0.3939006	60.79348	19.15	113.20
36	46	10	0.2173913	0.4170288	64.90109	19.20	106.05
37	55	13	0.2363636	0.4287638	65.78818	19.35	106.75
38	46	11	0.2391304	0.4312660	64.07500	19.30	110.70
39	49	13	0.2653061	0.4460713	60.89082	19.35	106.40
40	55	10	0.1818182	0.3892495	64.79273	19.60	110.10
41	55	9	0.1636364	0.3733550	68.10000	19.30	114.50
42	55	13	0.2363636	0.4287638	65.50455	19.05	108.30
43	48	10	0.2083333	0.4104141	72.40625	19.20	110.75
44	44	6	0.1363636	0.3471418	63.84432	19.35	111.50
45	54	4	0.0740741	0.2643505	73.81667	18.85	115.65
46	60	10	0.1666667	0.3758230	68.12917	19.25	110.00
47	61	14	0.2295082	0.4240064	67.46639	19.30	113.45
48	48	6	0.1250000	0.3342187	63.94687	19.25	117.45
49	54	15	0.2777778	0.4521090	76.25093	19.00	109.20
50	59	9	0.1525424	0.3626321	72.25254	19.35	114.35
51	56	7	0.1250000	0.3337119	65.52232	19.10	111.55
52	68	8	0.1176471	0.3245852	69.49632	19.20	111.25
53	62	14	0.2258065	0.4215255	70.23145	18.70	110.50
54	56	11	0.1964286	0.4008919	75.89107	18.95	115.60
55	52	7	0.1346154	0.3446423	68.25577	19.10	116.50
56	69	6	0.0869565	0.2838356	75.19493	19.55	115.85
57	51	7	0.1372549	0.3475404	68.08235	18.80	112.95

```
#Scatter plot exploring tenure and number of users
sp_user_tenure_temp <- qplot(churn_tenure$tenure,
                             churn_tenure$count_customers,
                             data = churn_tenure) +
                       geom_smooth() +
                       labs(x="Tenure", y="Users")

sp_user_tenure <- sp_user_tenure_temp +
                      theme_minimal() +
                      ggtitle("Users distributions in tenure") +
                      theme(axis.text.x = element_text(angle = 45, hjust = 1),
                            plot.title=element_text(hjust=0.5))

sp_user_tenure
```

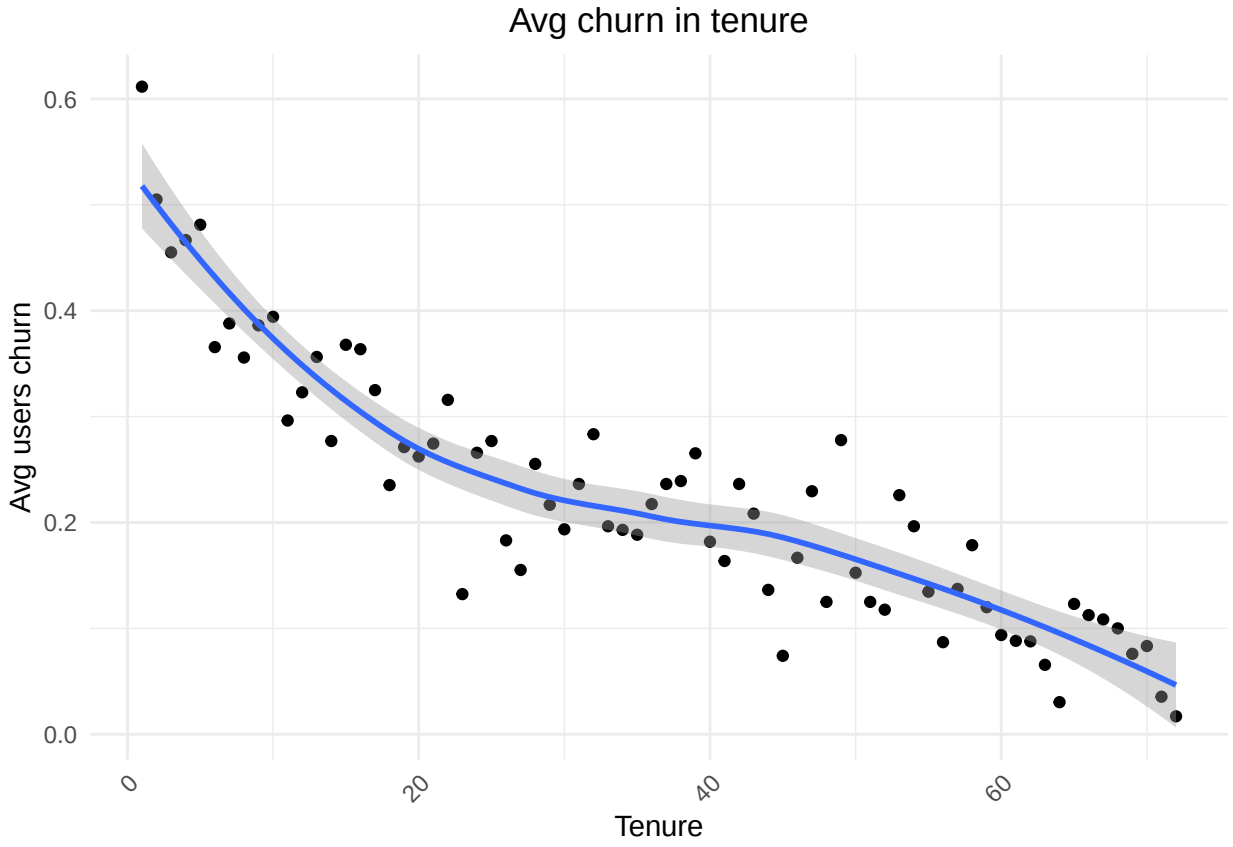
```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```




```
#scatter plot exploring tenure and avg churn
sp_avg_churn_tenure_temp <- qplot(churn_tenure$tenure,
                             churn_tenure$avg_churn,
                             data = churn_tenure) +
                       geom_smooth() +
                       labs(x="Tenure", y="Avg users churn")

sp_avg_churn_tenure <- sp_avg_churn_tenure_temp +
                    theme_minimal() +
                    ggtitle("Avg churn in tenure") +
```

```
                    theme(axis.text.x = element_text(angle = 45, hjust = 1),
                          plot.title=element_text(hjust=0.5))

sp_avg_churn_tenure
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



2.4.3 Charges

The products and services prices, annually or monthly customer spending have a strong influence in the churn. In these graphics and tables bellow, you could observe customers with their tenure and on average how much they spend monthly and annually. These variables affect churn because in almost every tenure interval, the users that spent more, are more susceptible to make churn. In the graphic below illustrated the correlation between spending and churn.

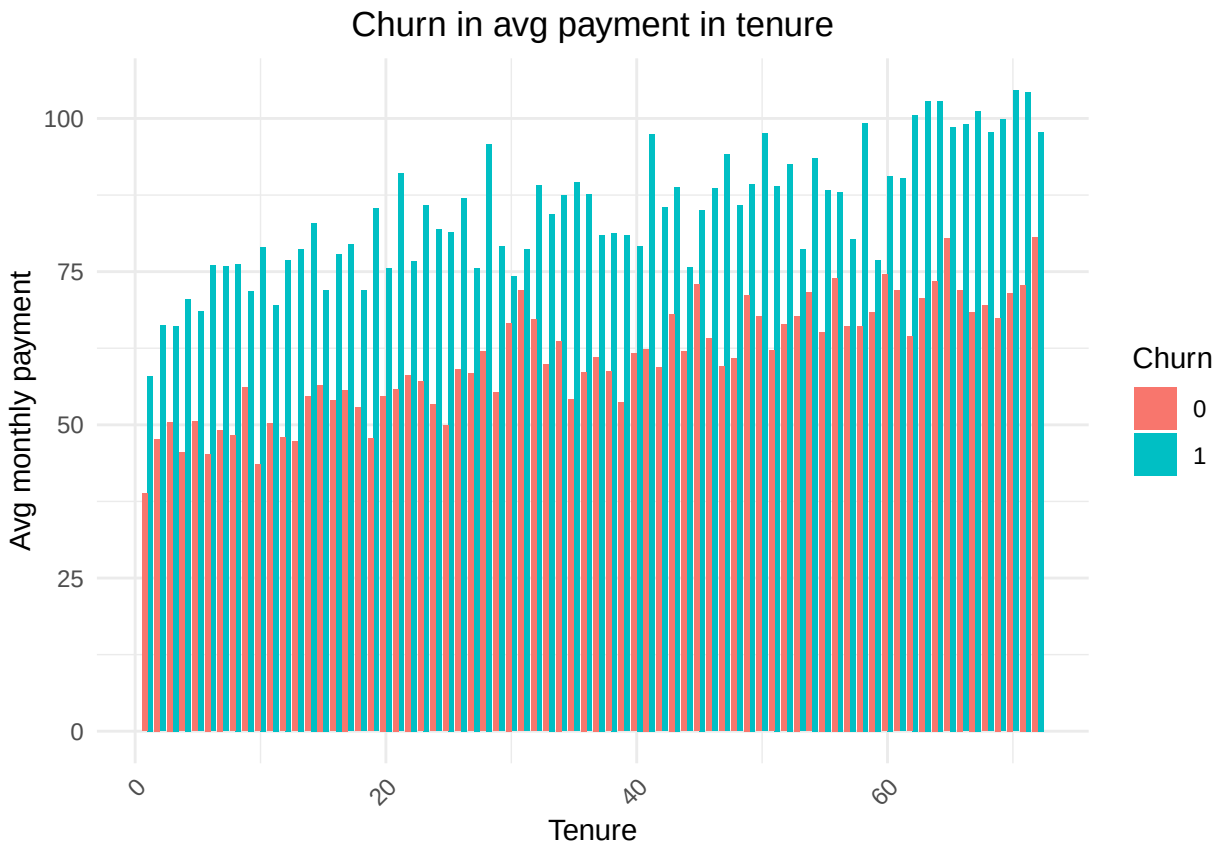
```
#This table shows the total charges category and churn
churn_total_charges <- churn_data %>%
                    group_by(TotalChargesCategory) %>%
                    summarise(count_customers=n(customerID),
                              churn=sum(Churn),
                              avg_chrun=mean(Churn),
                              sd_chrun=sd(Churn),
                              avg_totalCharges=mean(TotalCharges),
                              min_totalCharges=min(TotalCharges),
                              max_totalCharges=max(TotalCharges),
                    )

churn_total_charges
```

```
## # A tibble: 19 x 8
##    TotalChargesCat… count_customers churn avg_chrun sd_chrun avg_totalCharges
##    <fct>                     <int> <dbl>     <dbl>    <dbl>            <dbl>
##  1 1                          1647   681    0.413     0.493             179.
##  2 2                           706   193    0.273     0.446             691.
##  3 3                           592   134    0.226     0.419            1157.
##  4 4                           459    91    0.198     0.399            1598.
##  5 5                           296    71    0.240     0.428            2060.
##  6 6                           246    70    0.285     0.452            2522.
##  7 7                           235    62    0.264     0.442            3000.
##  8 8                           215    40    0.186     0.390            3445.
##  9 9                           192    33    0.172     0.378            3910.
## 10 10                          198    33    0.167     0.374            4345.
## 11 11                          194    30    0.155     0.362            4821.
## 12 12                          152    29    0.191     0.394            5273.
## 13 13                          175    28    0.16      0.368            5717.
## 14 14                          160    22    0.138     0.345            6166.
## 15 15                          131    13    0.0992    0.300            6629.
## 16 16                          110    19    0.173     0.380            7070.
## 17 17                           90    15    0.167     0.375            7539.
## 18 18                           77     5    0.0649    0.248            7980.
## 19 19                           31     1    0.0323    0.180            8414.
## # … with 2 more variables: min_totalCharges <dbl>, max_totalCharges <dbl>
```

```
#This table shows the tenure and churn
churn_tenure_charges <- churn_data %>%
                        group_by(tenure,Churn) %>%
                        summarise(count_customers=n(customerID),
                                  avg_monthly=mean(MonthlyCharges),
                                  min_monthly=min(MonthlyCharges),
                                  max_monthly=max(MonthlyCharges),
                                  avg_total=mean(TotalCharges),
                                  min_total=min(TotalCharges),
                                  max_total=max(TotalCharges),
                                  )

churn_tenure_charges
```

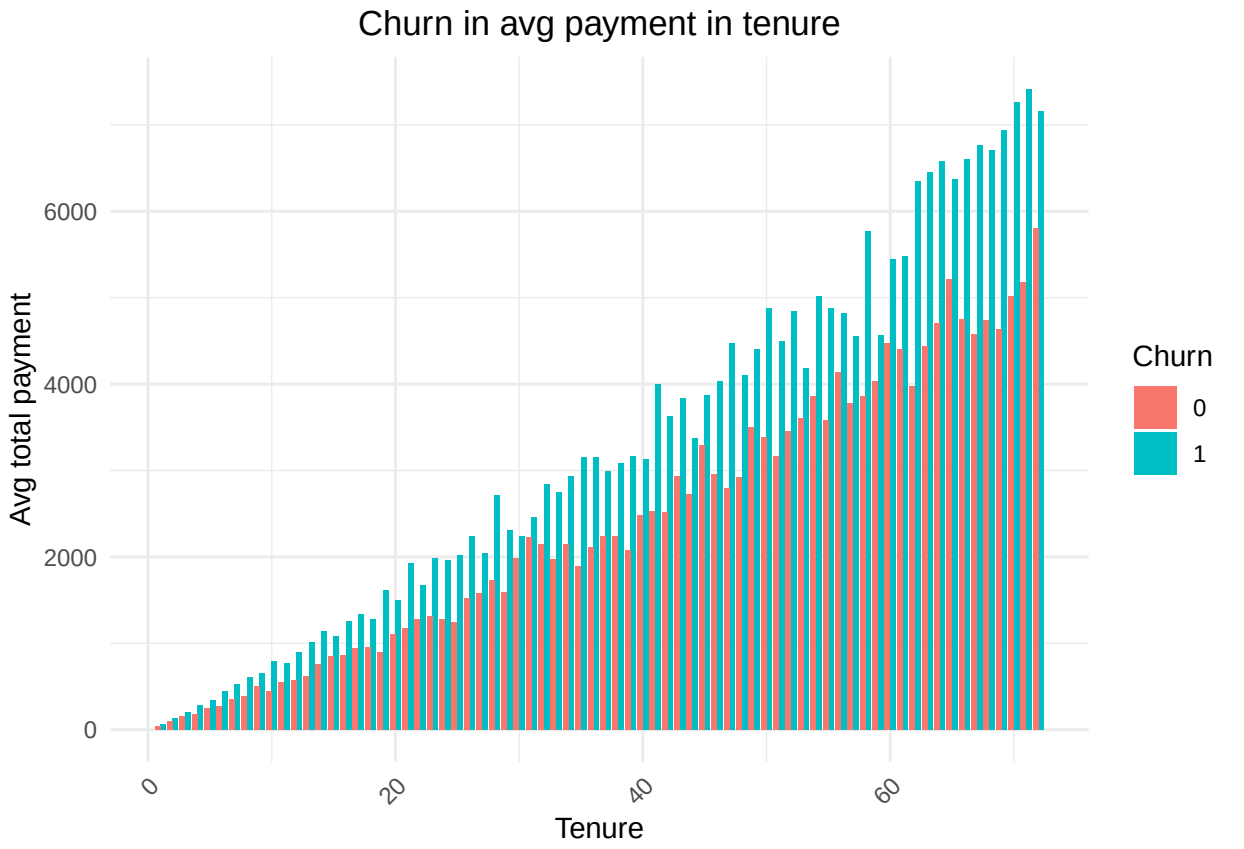
```
## # A tibble: 144 x 9
## # Groups:   tenure [72]
##    tenure Churn count_customers avg_monthly min_monthly max_monthly avg_total
##     <int> <dbl>           <int>       <dbl>       <dbl>       <dbl>     <dbl>
##  1      1     0             202        38.8        18.9        95.8      38.8
##  2      1     1             318        58.0        18.8       102.       58.0
##  3      2     0              99        47.7        18.8       100.       95.4
##  4      2     1             101        66.2        19.3       104.      132.
##  5      3     0              91        50.5        18.8       108.      151.
##  6      3     1              76        66.2        19.6       105.      201.
##  7      4     0              80        45.5        18.8       101.      181.
##  8      4     1              70        70.5        20.4       106.      284.
##  9      5     0              55        50.6        19.2       105.      252.
## 10      5     1              51        68.5        19.4       100.      341.
## # … with 134 more rows, and 2 more variables: min_total <dbl>, max_total <dbl>
```

```
#In these graphics bellow, you can observe customers with their tenure and on average how much they spe

churn_tenure_Monthlypayment <- ggplot(churn_tenure_charges,
                                aes(fill=as.factor(churn_tenure_charges$Churn),
                                    y=churn_tenure_charges$avg_monthly,
                                    x=churn_tenure_charges$tenure)) +
                               geom_bar(position="dodge", stat="identity") +
                                 theme_minimal() +
                                 ggtitle("Churn in avg payment in tenure") +
                                 labs(x="Tenure", y="Avg monthly payment", fill="Churn") +
                                 theme(axis.text.x = element_text(angle = 45, hjust = 1),
                                       plot.title=element_text(hjust=0.5))


churn_tenure_Monthlypayment
```




```
churn_tenure_TotalPayment <- ggplot(churn_tenure_charges,
                                aes(fill=as.factor(churn_tenure_charges$Churn),
                                    y=churn_tenure_charges$avg_total,
                                    x=churn_tenure_charges$tenure)) +
                               geom_bar(position="dodge", stat="identity") +
                               theme_minimal() +
                               ggtitle("Churn in avg payment in tenure") +
                               labs(x="Tenure", y="Avg total payment", fill="Churn") +
                               theme(axis.text.x = element_text(angle = 45, hjust = 1),
                                     plot.title=element_text(hjust=0.5))
```

```
churn_tenure_TotalPayment
```



2.4.4 Linear regressions

Using linear regression is possible to obtain the more relevant variables, between the 21 that are in the data, to make the best prediction. The more relevant variables that have a significant impact on the churn are:

```
lr_churn<-lm(Churn ~ gender + SeniorCitizen + Partner + Dependents + tenure + PhoneService + MultipleLi
               InternetService + OnlineSecurity + OnlineBackup + DeviceProtection + TechSupport + Strea
               StreamingTV + Contract + PaperlessBilling + PaymentMethod + MonthlyCharges + TotalCharge
             , data=churn_data)
summary(lr_churn)
```

```
##
## Call:
## lm(formula = Churn ~ gender + SeniorCitizen + Partner + Dependents +
##     tenure + PhoneService + MultipleLines + InternetService +
##     OnlineSecurity + OnlineBackup + DeviceProtection + TechSupport +
##     StreamingTV + StreamingTV + Contract + PaperlessBilling +
##     PaymentMethod + MonthlyCharges + TotalCharges, data = churn_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.80589 -0.25895 -0.06022  0.28505  1.13588
##
## Coefficients:
##                                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)                         2.389e-01  4.403e-02   5.427 5.96e-08 ***
```

```
## genderMale                                2.469e-03  9.769e-03   0.253 0.800514
## SeniorCitizen                             3.171e-02  1.416e-02   2.239 0.025185 *
## PartnerYes                                1.859e-03  1.183e-02   0.157 0.875119
## DependentsYes                            -1.863e-02  1.253e-02  -1.487 0.137004
## tenure                                   -1.960e-03  5.460e-04  -3.589 0.000335 ***
## PhoneServiceYes                          -1.178e-01  3.105e-02  -3.793 0.000151 ***
## MultipleLinesYes                          2.959e-02  1.355e-02   2.183 0.029043 *
## InternetServiceFiber optic                6.170e-02  3.519e-02   1.754 0.079560 .
## InternetServiceNo                        -1.944e-02  3.853e-02  -0.505 0.613818
## OnlineSecurityYes                        -7.311e-02  1.436e-02  -5.091 3.67e-07 ***
## OnlineBackupYes                          -4.237e-02  1.374e-02  -3.085 0.002046 **
## DeviceProtectionYes                      -1.817e-02  1.475e-02  -1.231 0.218279
## TechSupportYes                           -7.610e-02  1.499e-02  -5.077 3.95e-07 ***
## StreamingTVYes                            1.215e-02  2.040e-02   0.596 0.551409
## ContractOne year                         -1.042e-01  1.525e-02  -6.829 9.40e-12 ***
## ContractTwo year                         -7.675e-02  1.864e-02  -4.117 3.89e-05 ***
## PaperlessBillingYes                       5.273e-02  1.094e-02   4.821 1.46e-06 ***
## PaymentMethodCredit card (automatic)     -8.315e-03  1.485e-02  -0.560 0.575576
## PaymentMethodElectronic check             6.495e-02  1.461e-02   4.447 8.87e-06 ***
## PaymentMethodMailed check                -1.324e-03  1.589e-02  -0.083 0.933582
## MonthlyCharges                            4.745e-03  1.284e-03   3.697 0.000220 ***
## TotalCharges                             -4.385e-05  7.105e-06  -6.171 7.22e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.375 on 5883 degrees of freedom
## Multiple R-squared:  0.2821, Adjusted R-squared:  0.2794
## F-statistic: 105.1 on 22 and 5883 DF,  p-value: < 2.2e-16
```

There are several variables that have significance into the churn. The top are:

1 - Contract,

2 - TotalCharges,

3 - OnlineSecurity,

4 - TechSupport,

5 - PaperlessBilling,

6 - PaymentMethod,

7 - PhoneService,

8 - MonthlyCharges,

9 - tenure,

10 - OnlineBackup,

11 - SeniorCitizen

If we select this top, the interception p-value improves from 5.96e-08 to 3.21e-12. But, this selection could be better with fewer variables. The payment method and senior citizen has no significance compared with the other selected variables.

```
lr_churn<-lm(Churn ~ Contract+
                     TotalCharges +
                     OnlineSecurity +
                     TechSupport +
```

```
                    PaperlessBilling +
                    PaymentMethod +
                    PhoneService +
                    MonthlyCharges +
                    tenure +
                    OnlineBackup +
                    SeniorCitizen
             , data=churn_data)
summary(lr_churn)
```

```
##
## Call:
## lm(formula = Churn ~ Contract + TotalCharges + OnlineSecurity +
##     TechSupport + PaperlessBilling + PaymentMethod + PhoneService +
##     MonthlyCharges + tenure + OnlineBackup + SeniorCitizen, data = churn_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.79625 -0.26055 -0.05782  0.28756  1.14255
##
## Coefficients:
##                                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)                            1.878e-01  2.689e-02   6.983 3.21e-12 ***
## ContractOne year                      -1.121e-01  1.491e-02  -7.517 6.46e-14 ***
## ContractTwo year                      -8.520e-02  1.796e-02  -4.744 2.14e-06 ***
## TotalCharges                          -4.471e-05  6.934e-06  -6.448 1.22e-10 ***
## OnlineSecurityYes                     -8.739e-02  1.256e-02  -6.959 3.81e-12 ***
## TechSupportYes                        -9.431e-02  1.295e-02  -7.285 3.64e-13 ***
## PaperlessBillingYes                    5.438e-02  1.092e-02   4.978 6.61e-07 ***
## PaymentMethodCredit card (automatic) -9.307e-03  1.485e-02  -0.627 0.530999
## PaymentMethodElectronic check          6.620e-02  1.460e-02   4.533 5.92e-06 ***
## PaymentMethodMailed check             -2.151e-03  1.588e-02  -0.135 0.892286
## PhoneServiceYes                       -1.192e-01  1.833e-02  -6.501 8.65e-11 ***
## MonthlyCharges                         6.211e-03  3.464e-04  17.931  < 2e-16 ***
## tenure                                -1.824e-03  5.388e-04  -3.386 0.000714 ***
## OnlineBackupYes                       -5.322e-02  1.238e-02  -4.300 1.74e-05 ***
## SeniorCitizen                          3.844e-02  1.389e-02   2.768 0.005666 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3753 on 5891 degrees of freedom
## Multiple R-squared:   0.28,  Adjusted R-squared:  0.2783
## F-statistic: 163.7 on 14 and 5891 DF,  p-value: < 2.2e-16
```

Removing payment method and SeniorCitizen, we have the strongest linear regression.

```
lr_churn<-lm(Churn ~ Contract+
                    TotalCharges +
                    OnlineSecurity +
                    TechSupport +
                    PaperlessBilling  +
                    PhoneService +
                    MonthlyCharges +
                    tenure +
                    OnlineBackup
```

```
             , data=churn_data)
summary(lr_churn)
```

```
##
## Call:
## lm(formula = Churn ~ Contract + TotalCharges + OnlineSecurity +
##     TechSupport + PaperlessBilling + PhoneService + MonthlyCharges +
##     tenure + OnlineBackup, data = churn_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.7696 -0.2606 -0.0569  0.2960  1.1414
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          1.977e-01  2.358e-02   8.383  < 2e-16 ***
## ContractOne year    -1.234e-01  1.486e-02  -8.302  < 2e-16 ***
## ContractTwo year    -1.029e-01  1.782e-02  -5.779 7.91e-09 ***
## TotalCharges        -4.908e-05  6.872e-06  -7.142 1.03e-12 ***
## OnlineSecurityYes   -9.710e-02  1.251e-02  -7.760 9.97e-15 ***
## TechSupportYes      -1.068e-01  1.284e-02  -8.317  < 2e-16 ***
## PaperlessBillingYes  5.857e-02  1.093e-02   5.359 8.69e-08 ***
## PhoneServiceYes     -1.343e-01  1.820e-02  -7.379 1.81e-13 ***
## MonthlyCharges       6.909e-03  3.230e-04  21.391  < 2e-16 ***
## tenure              -1.632e-03  5.256e-04  -3.105  0.00191 **
## OnlineBackupYes     -5.627e-02  1.241e-02  -4.533 5.92e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3766 on 5895 degrees of freedom
## Multiple R-squared:  0.2745, Adjusted R-squared:  0.2733
## F-statistic: 223.1 on 10 and 5895 DF,  p-value: < 2.2e-16
```

These are the strongest significant variables in a linear regression:

1 - Contract,

2 - TotalCharges,

3 - MonthlyCharges,

4 - TechSupport,

5 - OnlineSecurity,

6 - PhoneService,

7 - PaperlessBilling,

8 - OnlineBackup

But, based on the objective of taking loyalty action over the user that could cancel the company services, and avoid overfitting, I selected the variables contract, total charges, monthly charges and tenure to be evaluated them. I considered these variables as customer-centric to evaluate customer behavior to predict their churn. TechSupport, OnlineSecurity, PhoneService, PaperlessBilling and OnlineBackup are added value to the company's products but are not intrinsic related to uses of the services. Although, it is suggested to be used them as tools for loyalty actions. This linear regression obtained 2.5e-09 for the p-value interception that is a really good one and not overfitted.

```r
lr_churn <- lm(Churn ~ Contract+
                    TotalCharges +
                    MonthlyCharges +
                    tenure
            , data=churn_data)
summary(lr_churn)
```

```
##
## Call:
## lm(formula = Churn ~ Contract + TotalCharges + MonthlyCharges +
##     tenure, data = churn_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.76863 -0.26557 -0.05049  0.33694  1.05458
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        1.177e-01  1.972e-02   5.971  2.5e-09 ***
## ContractOne year -1.694e-01  1.483e-02 -11.423  < 2e-16 ***
## ContractTwo year -1.753e-01  1.742e-02 -10.065  < 2e-16 ***
## TotalCharges     -6.850e-05  6.836e-06 -10.021  < 2e-16 ***
## MonthlyCharges    6.249e-03  3.068e-04  20.367  < 2e-16 ***
## tenure           -7.222e-04  5.322e-04  -1.357    0.175
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3842 on 5900 degrees of freedom
## Multiple R-squared:  0.2446, Adjusted R-squared:  0.2439
## F-statistic:   382 on 5 and 5900 DF,  p-value: < 2.2e-16
```

2.5 Insights gained

This is a small data but is very interesting because this approach could be used for any company's services. Considering this data as a picture in a certain time, immersed into an open complex system (CLIOS Process), where some users made churn to another company and others no yet for the moment. It is uncertain if they will keep been customers for 1, 2 or 60 months more because it depends on internal and external factors. This means they are making churn to other companies could be because the technology is constantly evolving as the same as the market growth in the offer and the demands are variating but the only constant is the communication needs will always exist as an intrinsic aspect of human beings and universal rights; or could be for opportunities that better fit their needs based on cost, quality and quantity services; or other factors like the prestige of a mark or because the person just wants a change without any other reason.

The demographics suggested that some users are more characterize with the company than others but in my opinion, are not actionable. I can not recommend not to sell to senior citizens females with no partner and not dependents because more of the half will make churn. Not only because is illegal but more important, unethical. Everyone has the right to communicate and is a customer so is not possible to be selective.

2.6 Modeling approach

This study considered 5 different approaches:

1 - The simple average of the rating;

2 - The previous model adding contracts effect;

3 - The previous model adding total charges effect;

4 - The previous model adding monthly charges effect;

5 - The previous model adding tenure effect.

Every step is increasing the number of variables considered, increase the complexity and reduce the error in our predictions of churn. All of them were evaluated using RMSE and the best option was the 5Th as you could observe in the results bellow.

```
RMSE <- function(true_ratings, predicted_ratings)
{ sqrt(mean((true_ratings - predicted_ratings)^2))}
```

2.6.1 Method 1, churn average

First, considering that the average in churn could predict future churn and the recommendation is using just the churn.

```
mu_hat <- mean(churn_data$Churn)
mu_hat
```

```
## [1] 0.2658314
```

```
rmse_1_mu <- RMSE(churn_data$Churn, mu_hat)
rmse_1_mu
```

```
## [1] 0.4417749
```

```
#Add the rusult to the result table
methods_rmse_results <- data.frame(model="General churn average only", RMSE=rmse_1_mu)

kable(methods_rmse_results)
```

model	RMSE
General churn average only	0.4417749

2.6.2 Method 2, contracts effect

Secondly, based on the previous method, increase complexity by adding contracts to reduce the error and improve the recommendation using churn and contracts types effect.

```
mu <- mean(churn_data$Churn)
mu
```

```
## [1] 0.2658314
```

```
contract_mu <- churn_data %>%
               group_by(Contract) %>%
               summarise(b_c = mean(Churn - mu))
contract_mu
```

```
## # A tibble: 3 x 2
##   Contract          b_c
##   <fct>           <dbl>
## 1 Month-to-month  0.160
## 2 One year       -0.151
## 3 Two year       -0.239
```

```
predicted_churn_2 <- mu + validation %>% left_join(contract_mu, by='Contract') %>% pull(b_c)
```

```
model_2_rmse <- RMSE(predicted_churn_2, validation$Churn)
model_2_rmse
```

```
## [1] 0.3965283
```

```
#Add the rusult to the result table
methods_rmse_results <- methods_rmse_results %>% add_row(model="Contract effect", RMSE=model_2_rmse)

kable(methods_rmse_results)
```

model	RMSE
General churn average only	0.4417749
Contract effect	0.3965283

2.6.3 Method 3, total charges effect

Thirdly, based on the previous method, increase complexity by adding total charges to reduce the error and improve the recommendation using churn, contracts types and total charges effect.

```
totalchargescategory_mu <- churn_data %>%
                  left_join(contract_mu, by='Contract') %>%
                  group_by(TotalChargesCategory) %>%
                  summarise(b_t = mean(Churn - mu - b_c))


predicted_Churns_3 <- validation %>%
                      left_join(contract_mu, by='Contract') %>%
                      left_join(totalchargescategory_mu, by='TotalChargesCategory') %>%
                      mutate(pred = mu + b_c + b_t) %>%
                      pull(pred)


model_3_rmse <- RMSE(predicted_Churns_3, validation$Churn)
model_3_rmse
```

```
## [1] 0.3951132
```

```
#Add the rusult to the result table
methods_rmse_results <- methods_rmse_results %>% add_row(model="Total charges category effect", RMSE=mo
kable(methods_rmse_results)
```

model	RMSE
General churn average only	0.4417749
Contract effect	0.3965283
Total charges category effect	0.3951132

2.6.4 Method 4, monthly charges effect

Fourthly, based on the previous method, increase complexity by adding monthly charges to reduce the error and improve the recommendation using churn, contracts types, total and monthly charges effect.

```
monthly_mu <- churn_data %>%
              left_join(contract_mu, by='Contract') %>%
              left_join(totalchargescategory_mu, by='TotalChargesCategory') %>%
              group_by(MonthlyChargesCategory) %>%
              summarise(b_mo = mean(Churn - mu - b_c - b_t))
monthly_mu
```

```
## # A tibble: 22 x 2
##    MonthlyChargesCategory     b_mo
##    <fct>                     <dbl>
##  1 1                       -0.134
##  2 2                       -0.0403
##  3 3                       -0.0945
##  4 4                       -0.0490
##  5 5                       -0.00461
##  6 6                       -0.0264
##  7 7                       -0.0901
##  8 8                       -0.0547
##  9 9                       -0.0907
## 10 10                      -0.0729
## # … with 12 more rows
```

```
predicted_Churns_4 <- validation %>%
                      left_join(contract_mu, by='Contract') %>%
                      left_join(totalchargescategory_mu, by='TotalChargesCategory') %>%
                      left_join(monthly_mu, by='MonthlyChargesCategory') %>%
                      mutate(pred = mu + b_c + b_t + b_mo) %>%
                      pull(pred)

model_4_rmse <- RMSE(predicted_Churns_4, validation$Churn)
model_4_rmse
```

```
## [1] 0.3872303
```

```
#Add the rusult to the result table

methods_rmse_results <- methods_rmse_results %>%  add_row(model="Monthly charges category effect", RMSE=
kable(methods_rmse_results)
```

model	RMSE
General churn average only	0.4417749
Contract effect	0.3965283
Total charges category effect	0.3951132
Monthly charges category effect	0.3872303

2.6.5 Method 5, tenure effect

Finally, based on the previous method, increase complexity by adding tenure to reduce the error and improve the recommendation using churn, contracts types, total and monthly charges and tenure effect.

```
tenure_mu <- churn_data %>%
              left_join(contract_mu, by='Contract') %>%
              left_join(totalchargescategory_mu, by='TotalChargesCategory') %>%
              left_join(monthly_mu, by='MonthlyChargesCategory') %>%
              group_by(tenure) %>%
              summarise(b_te = mean(Churn - mu - b_c - b_t - b_mo))
tenure_mu
```

```
## # A tibble: 72 x 2
##    tenure     b_te
##     <int>    <dbl>
##  1      1  0.182
##  2      2  0.0670
##  3      3  0.0179
```

```
##  4      4  0.0356
##  5      5  0.0403
##  6      6 -0.0161
##  7      7  0.00680
##  8      8 -0.00218
##  9      9  0.00905
## 10     10  0.0357
## # … with 62 more rows
```

```
predicted_Churns_5 <- validation %>%
                      left_join(contract_mu, by='Contract') %>%
                      left_join(totalchargescategory_mu, by='TotalChargesCategory') %>%
                      left_join(monthly_mu, by='MonthlyChargesCategory') %>%
                      left_join(tenure_mu, by='tenure') %>%
                      mutate(pred = mu + b_c + b_t  + b_mo + b_te) %>%
                      pull(pred)

model_5_rmse <- RMSE(predicted_Churns_5, validation$Churn)
model_5_rmse
```

```
## [1] 0.3818975
```

```
#Add the rusult to the result table

methods_rmse_results <- methods_rmse_results %>% add_row(model="Tenure effect", RMSE=model_5_rmse)
kable(methods_rmse_results)
```

model	RMSE
General churn average only	0.4417749
Contract effect	0.3965283
Total charges category effect	0.3951132
Monthly charges category effect	0.3872303
Tenure effect	0.3818975

3 Results

In conclusion, the best model is the 5th with an excellent value of 0.3818975. This adds more complexity to the model giving better predictions but it is no infinity. Notice that every time a new dimension is added to the model improves the RMSE but in this study, after the 5Th the improvement was very low because it is harder to predict better behavior only considering one company of the market without knowing the quality of the services, prices of the products, experience of use of the users, problem resolutions, between others variables in the industry.

```
kable(methods_rmse_results)
```

model	RMSE
General churn average only	0.4417749
Contract effect	0.3965283
Total charges category effect	0.3951132
Monthly charges category effect	0.3872303
Tenure effect	0.3818975

4 Conclusion

4.1 Summary of the report

In this project, we could evaluate more than 6.5 thousand users from a telco company with the objective to take loyalty actions to retain users that could make churn. We could verify that the customers of this company have a wide variate in gender, ages, partner and dependents. Independent of the demographics of the users we could make a recommendation with an RMSE of 0.3818975 considering the averages of churn, contracts type, total charges, monthly charges and tenure, which is an excellent result based only on commercial results index.

4.2 Limitations

This data is very small with only about 7 thousand users with commercial result indicators without knowing the universe of the populations (total clients) and we don't have performance indicators to evaluate the services neither the performance and result of the customer support. In this data there are some characteristics associated with the products like multiple lines or phone services but there are not the characteristic of services, for example, are unlimited in minutes and/or in data? how could these products compete with the market based on price, characteristic and quality? are users from one country? all these questions remain unanswered that could help to improve the model prediction.

4.3 Future work

This study riched great results but is still under-developed its full potential. The above analysis is also suitable for other services companies but keeping the focus on telecom companies, here we want to extend the discussion and make a few further recommendations. The performance indicators of services, customer support, characteristics of the product and the market competition products not only could improve this prediction of churn to make loyalty actions it also could help to developed better products to sell. If the company has centered on customer satisfaction, the churn could be a great opportunity to know better the reasons for churn with a poll looking to gap to improve commercially and technically the processes and the products offered.