CCT College Dublin

Assessment Cover Page

Module Title:	Strategic Thinking
Assessment Title:	CA 2 Presentation
Lecturer Name:	James Garza
Student Full Name:	Arthur Claudino Gomes de Assis
Student Number:	2023146
Assessment Due Date:	15/11/2023
Date of Submission:	20/11/2023

Declaration

By submitting this assessment, I confirm that I have read the CCT policy on Academic Misconduct and understand the implications of submitting work that is not my own or does not appropriately reference material taken from a third party or other source. I declare it to be my own work and that all material from third parties has been appropriately referenced. I further confirm that this work has not previously been submitted for assessment by myself or someone else in CCT College Dublin or any other higher education institution.

1 Table of Contents

2 Introduction

Before analysing our database, we will briefly introduce what is churn and its importance.

Churn is a common problem in the telecommunications business and refers to customers who cancel or do not renew their contract with a telecommunications company in a given period. Churn is a very important indicator for telecommunications companies since it is much more expensive to attract new customers than to retain existing ones.

What we aim with our analysis is identifying according to our dataset how to predict and prevent churn.

2.1 Causes of Customer Churn

1- Price: If customers find a more cost-effective solution to themselves, they may churn. It is essential to present the added value, so customers feel that the purchase is worth the cost.

2- Product/Market Fit: When the client feel that they cannot achieve their goals with our solution.

3- User Experience: If the user experience with the product or application is buggy, and glitchy, for them, they will be less likely to use it on a regular basis and build expertise with it.

4- Customer experience: If a customer's experience connecting with other aspects of the company, such as customer service, executives, technical support, and installation service, is not positive, the likelihood of churn could increase.

5- Other causes: It is also considered churn when a customer moves to another address in which the company cannot service it, when customer deceases, when the demand for the service no longer exist.

2.2 Types of Customer Churn

1- Revenue Churn: This happens when customers downgrade to a cheaper version of our product.

2- Competitor Intervention: It is very important to focus on the reason why customers leave the company for our competitors. Are we a bad option for your business? Or is it something we are doing that is driving them away?

3- Unsuccessful Onboarding: This happens when executives focus only on the sale and not on the right solution for the client as a technology partner.

4- Desired Feature or Functionality: This happens when we offer all customers the same product, and we do not understand that the product must be adapted to the customer and not the customer to the product.

In this project we will analyse the data we have about Churn, we will study its possible causes and will provide some solutions to reduce Churn and increasing revenue.

3 Loading Libraries and Dataset

3.1 Loading libraries

```
# We import all the libraries that we need in our analysis and we will import others as we need them.

import pandas as pd
import math
import numpy as np
import scipy.stats as stats
from scipy.stats import binom
import seaborn as sns
import matplotlib.pyplot as plt
from scipy.stats import norm
from sklearn.model_selection import GridSearchCV
import statsmodels as sm
from scipy.stats import chisquare
from collections import Counter

#Models for Modeling Machine Learning Models
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import StratifiedKFold
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC

import warnings
warnings.filterwarnings('ignore')
```

3.2 Loading Dataset

```
# Importing the dataset.

df_churn = pd.read_csv('Telco_Churn.csv')
```

The first step was reading the dataset using the function read_csv. As it follows we will print the first 5 rows of the dataset to have a quick look, and also get the most basic information about the dataset using the function info().

```
1 df_churn.head()
```

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...	DeviceProtection	TechSu
0	7590-VHVEG	Female	0	Yes	No	1	No	No phone service	DSL	No	...	No	
1	5575-GNVDE	Male	0	No	No	34	Yes	No	DSL	Yes	...	Yes	
2	3668-QPYBK	Male	0	No	No	2	Yes	No	DSL	Yes	...	No	
3	7795-CFOCW	Male	0	No	No	45	No	No phone service	DSL	Yes	...	Yes	
4	9237-HQITU	Female	0	No	No	2	Yes	No	Fiber optic	No	...	No	

5 rows × 21 columns

```
1 df_churn.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 21 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   customerID        7043 non-null   object 
 1   gender            7043 non-null   object 
 2   SeniorCitizen     7043 non-null   int64  
 3   Partner           7043 non-null   object 
 4   Dependents        7043 non-null   object 
 5   tenure            7043 non-null   int64  
 6   PhoneService      7043 non-null   object 
 7   MultipleLines     7043 non-null   object 
 8   InternetService   7043 non-null   object 
 9   OnlineSecurity    7043 non-null   object 
 10  OnlineBackup      7043 non-null   object 
 11  DeviceProtection  7043 non-null   object 
 12  TechSupport       7043 non-null   object 
 13  StreamingTV       7043 non-null   object 
 14  StreamingMovies   7043 non-null   object 
 15  Contract          7043 non-null   object 
 16  PaperlessBilling  7043 non-null   object 
 17  PaymentMethod     7043 non-null   object 
 18  MonthlyCharges    7043 non-null   float64
 19  TotalCharges      7043 non-null   object 
 20  Churn             7043 non-null   object 
dtypes: float64(1), int64(2), object(18)
memory usage: 1.1+ MB
```

We can verify that the dataset has 7043 rows and 21 features, divided among categorical and numerical variables. Most of the variables are classified as objects but we also have integers and continuous variables. Initially we cannot identify any variable with null values.

Something that catches our intention in this initial glance on the dataset is that the 19th feature, TotalCharges, is classified as object, although we can verify numerical values on the five first rows above. Therefore this variable will require some kind of preparation before any analysis.

4 Data Cleaning and Engineering

Let us first verify if any of the features contain null values that have not been shown in the function info().

```
#Calculating the number of null values in each feature of our dataset.

df_churn.isnull().sum()
```

```
customerID          0
gender              0
SeniorCitizen       0
Partner             0
Dependents          0
tenure              0
PhoneService        0
MultipleLines       0
InternetService     0
OnlineSecurity      0
OnlineBackup        0
DeviceProtection    0
TechSupport         0
StreamingTV         0
StreamingMovies     0
Contract            0
PaperlessBilling    0
PaymentMethod       0
MonthlyCharges      0
TotalCharges        0
Churn               0
dtype: int64
```

As we can verify, none of the columns have null values, and therefore we will try to catch any undesired on unformatted data among the numbers, and verify why the total TotalCharges is read as object.

The following syntax works to replace all values in our "df_churn" Data Frame that are in the list ["n.a.","?","NA","n/a", "na", "--" , " "] to np.nan, which is the value typically used to represent a missing value in Python.

This is useful in data preprocessing as it is common for data to have missing values or missing data that is represented in different ways. By replacing them with a single value (np.nan), methods can be applied to handle these missing values uniformly.

To understand what to replace in this case, considering that there are only 11 values missing. The dataframe will be filtered to verify other values of these same rows.

```
#Applying the list suggested to replace some common mistaken data on the dataset by NaN, so
#that they may be undercovered.

df_churn.replace(["n.a.","?","NA","n/a", "na", "--", " "], np.nan, inplace = True)
```

```
#Filtering the null values on the feature TotalCharges
df_churn[df_churn['TotalCharges'].isnull()]
```

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...	DeviceProtection	TechS
488	4472-LVYGI	Female	0	Yes	Yes	0	No	No phone service	DSL	Yes	...	Yes	
753	3115-CZMZD	Male	0	No	Yes	0	Yes	No	No	No internet service	...	No internet service	No
936	5709-LVOEQ	Female	0	Yes	Yes	0	Yes	No	DSL	Yes	...	Yes	
1082	4367-NUYAO	Male	0	Yes	Yes	0	Yes	Yes	No	No internet service	...	No internet service	No
1340	1371-DWPAZ	Female	0	Yes	Yes	0	No	No phone service	DSL	Yes	...	Yes	
3331	7644-OMVMY	Male	0	Yes	Yes	0	Yes	No	No	No internet service	...	No internet service	No
3826	3213-VVOLG	Male	0	Yes	Yes	0	Yes	Yes	No	No internet service	...	No internet service	No
	2520-									No internet		No internet	No

It is possible to identify that all the rows missing values for TotalCharges have tenure equals to 0. Tenure equals to 0 means that this customer has onboarded under a month ago, and there it has not paid any monthly charge yet. Therefore, we identify that missing values on TotalCharges are mistypings that should have been filled with 0. Thus, they will be replace by 0. After executing this procedure, the column will be converted to a numeric column.

```
#Verifying information about the dataset again.

df_churn.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 21 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   customerID        7043 non-null   object 
 1   gender            7043 non-null   object 
 2   SeniorCitizen     7043 non-null   int64  
 3   Partner           7043 non-null   object 
 4   Dependents        7043 non-null   object 
 5   tenure            7043 non-null   int64  
 6   PhoneService      7043 non-null   object 
 7   MultipleLines     7043 non-null   object 
 8   InternetService   7043 non-null   object 
 9   OnlineSecurity    7043 non-null   object 
 10  OnlineBackup      7043 non-null   object 
 11  DeviceProtection  7043 non-null   object 
 12  TechSupport       7043 non-null   object 
 13  StreamingTV       7043 non-null   object 
 14  StreamingMovies   7043 non-null   object 
 15  Contract          7043 non-null   object 
 16  PaperlessBilling  7043 non-null   object 
 17  PaymentMethod     7043 non-null   object 
 18  MonthlyCharges    7043 non-null   float64
 19  TotalCharges      7043 non-null   float64
 20  Churn             7043 non-null   object 
dtypes: float64(2), int64(2), object(17)
memory usage: 1.1+ MB
```

Now we can call a descriptive analysis of our dataset to verify some main features of the numerical variables.

	SeniorCitizen	tenure	MonthlyCharges	TotalCharges
count	7043.000000	7043.000000	7043.000000	7043.000000
mean	0.162147	32.371149	64.761692	2279.734304
std	0.368612	24.559481	30.090047	2266.794470
min	0.000000	0.000000	18.250000	0.000000
25%	0.000000	9.000000	35.500000	398.550000
50%	0.000000	29.000000	70.350000	1394.550000
75%	0.000000	55.000000	89.850000	3786.600000
max	1.000000	72.000000	118.750000	8684.800000

This function returns a summary of the statistics of numeric columns on the dataset.

The summary includes the number of non-null values, the mean, the standard deviation, the minimum and maximum values, and the 25%, 50%, and 75% percentiles.

The descriptive analysis shows us something we have not noticed before, although the feature SeniorCitizen is read as an integer, it is actually a binary variable that represents categories, and it will need to be converted to such.

We can verify from our descriptive analysis:

1- Contracts of the company are as long as 32 months in average. Half of the contracts concentrate up to 29 months, and the longer contract has 72 months.

2- The smallest month charge in our dataset is 18.25 and the highest 118.75. The average monthly charge of the company is 64.76.

3- Our maximum values in TotalCharges is 8684.8, however 75% of the total amounts concentrate until 3786.60 and 50% concentrate until 1394.55. This difference is a way higher than what we can observe in tenure and Monthly Charges. Considering that TotalCharges should be approximately the product of tenure by Monthly Charges, that might mean that customers either have high priced contracts, or stay long time as customers, but both is unlikely.

After the process that has been performed, we could verify that the feature customerID has only is a categorical column that has only unique values, and they represent an identification of the customer, therefore it does not help us with any prediction power, and it will be dropped.

Once our dataset is clean and prepared, we can follow to our exploratory data analysis.

5 Exploratory Data Analysis

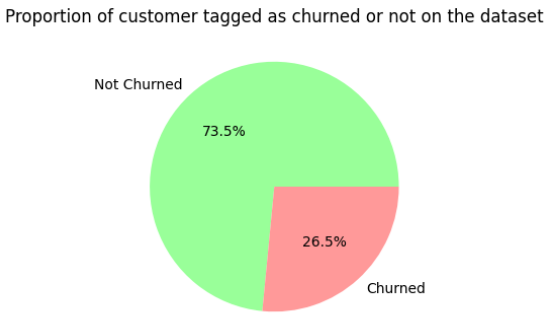
We will explore our data feature by feature, to verify what can we find out about our dataset, and how variables may influence our variable churn.

5.1 Analysing our target variable

As it is possible to see, our dataset is unbalanced. We have much more negative categories, customer that has not churned, that positive categories, customers that has churned. All the features that will be analysed onwards, must be analysed by comparison between percentages in category churn and not churned. That is because even if a feature has high percentage within customers that have churned, it might have an even higher category among customers that have not churned. So to reach proper conclusions, comparisons are necessary.

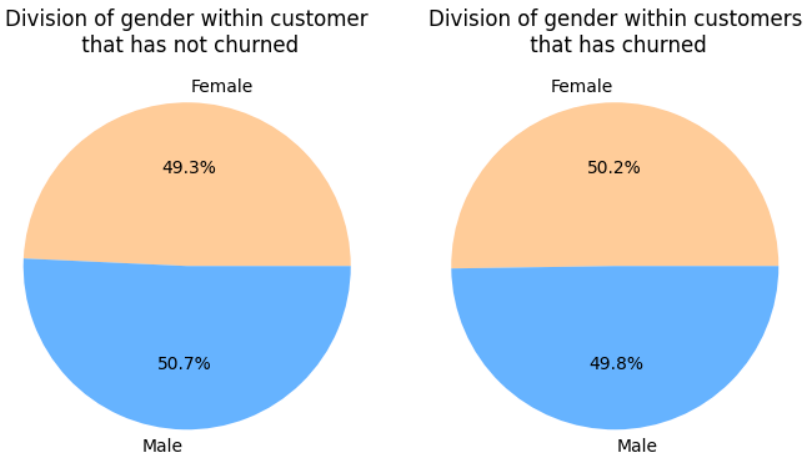
5.2 Analysis Customer's Features vs Churn

5.2.1 Analysis Gender vs Churn



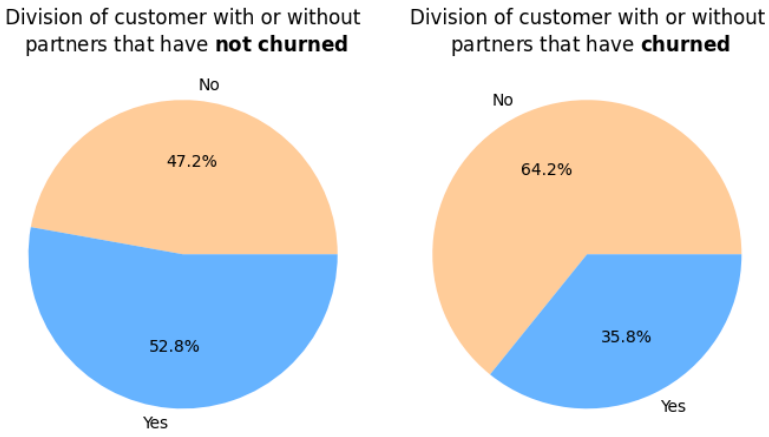
As we can verify, the proportion of male and female customers among customers that have and have not churned is very similar. Thus we understand this variable does not influence on churning rates.

5.2.2 Analysis Senior Citizens vs Churn



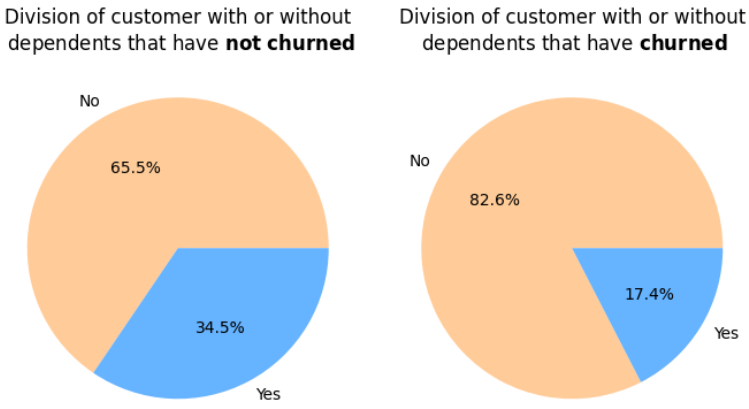
It is possible to verify that the percentage of Senior Citizens among customers that have churned is higher than those that have not churned, therefore, we expect this variable to have some influence over churn.

5.2.3 Analysis Partner vs Churn



It is possible to verify that a higher percentage of people who do not partner have churned compared to those that have a partner. It means that the feature partner might contribute negatively for churn.

5.2.4 Analysis Dependents vs Churn



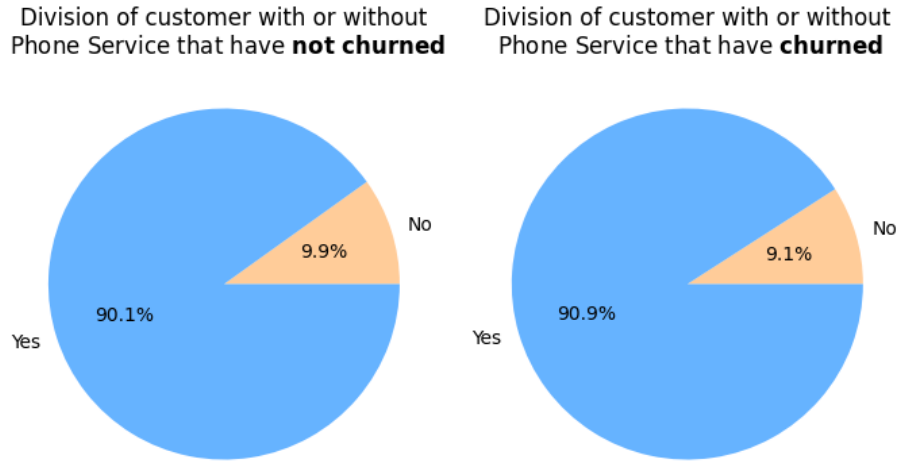
Again, as it happened with partners, we can verify that the percentage of customers without dependents is higher among those who have churned than those who have not churned, that means that having a dependent might reduce the chances of a customers to leave the company.

It is important to understand, that as those features are related to customer, they are directly connected with the marketing strategy for specific targets, rather than considering that some target should be favourited by the company. We have seen that young people, without a partner and without dependents are more likely to churn, therefore, the company must keep

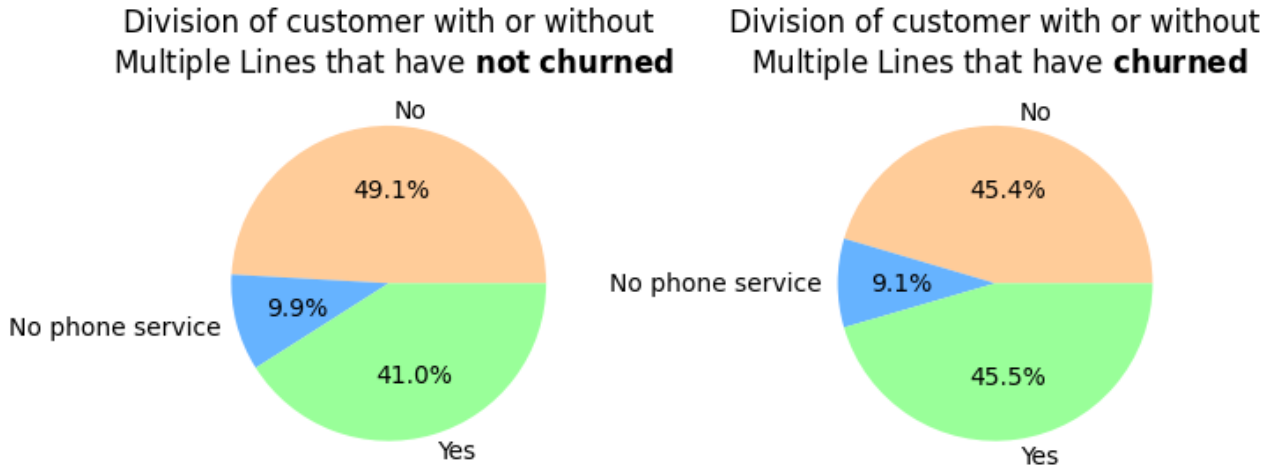
a closer contact with this target, offering new products, attempting renewing contracts, collecting feedbacks, and so on.

5.3 Analysis Product Features vs Churn

5.3.1 Analysis Phone Service and MultipleLines vs Churn

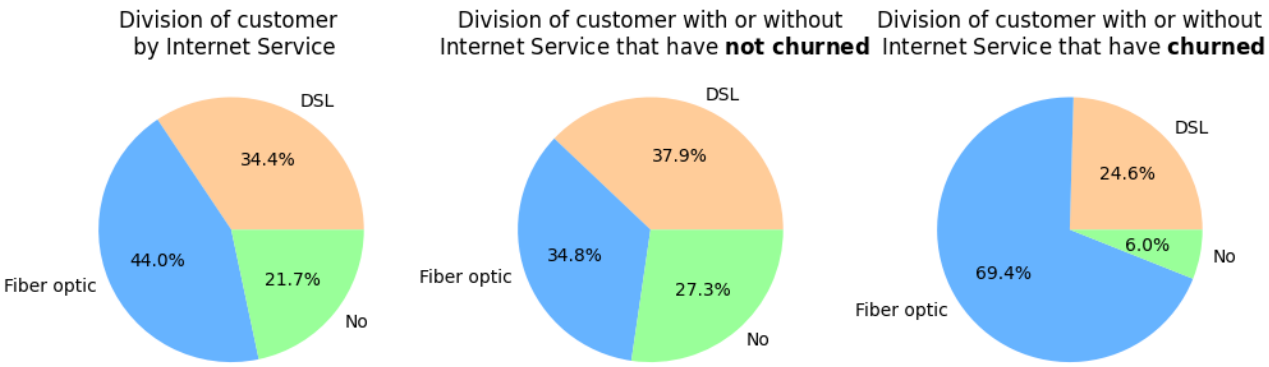


We can verify that whether a customer has phone service or not does not strongly influence churn, because the percentages in both groups that have and have not churned is very similar



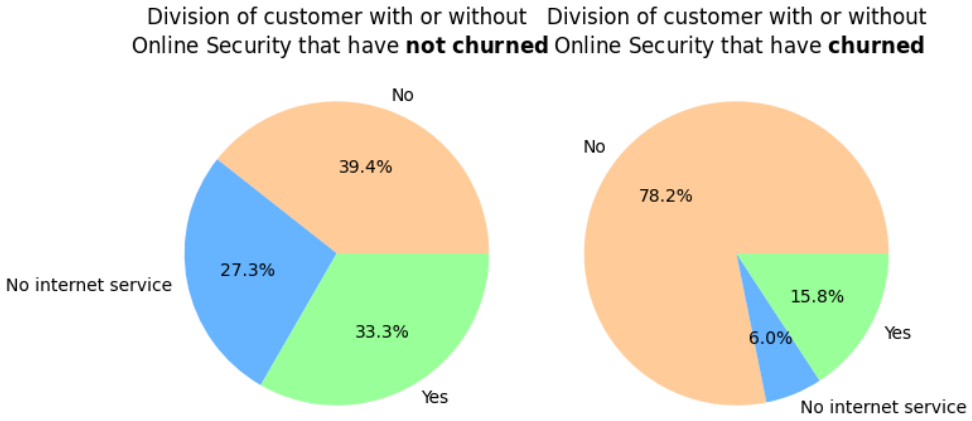
Although there is a small difference in the percentages in within the groups that have and have not churn, the difference is small, and the variable probably does not influence churn enough to be considered. A statistic test will make clear whether or not the variable should be brought to machine learning models.

5.3.2 Analysis Internet Service vs Churn

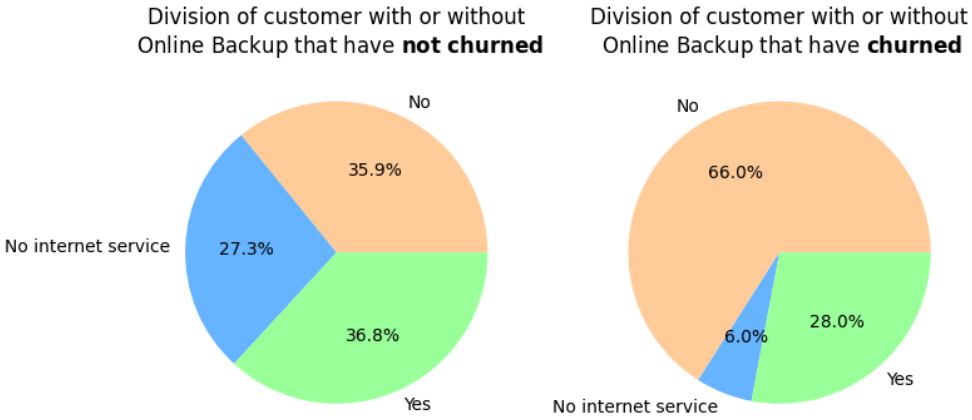


Reviewing the 'InternetService' column, we can identify that on the group that have churned, Fiber optic represents nearly the double of the percentage of customers that have not churned and have the same service. It is possible to see also that customer that do not have internet are less likely to churn. That means that either exists a problem on internet service of the company, that do not attend the demand properly, or just exposes the volatility and competitiveness of the internet sector, over which customers are more exigent and competitors are more aggressive.

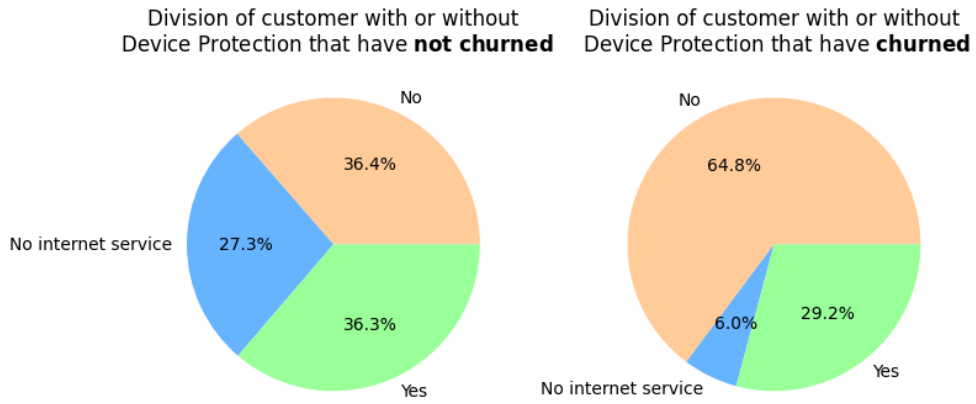
5.3.3 Analysis Online Security vs Churn



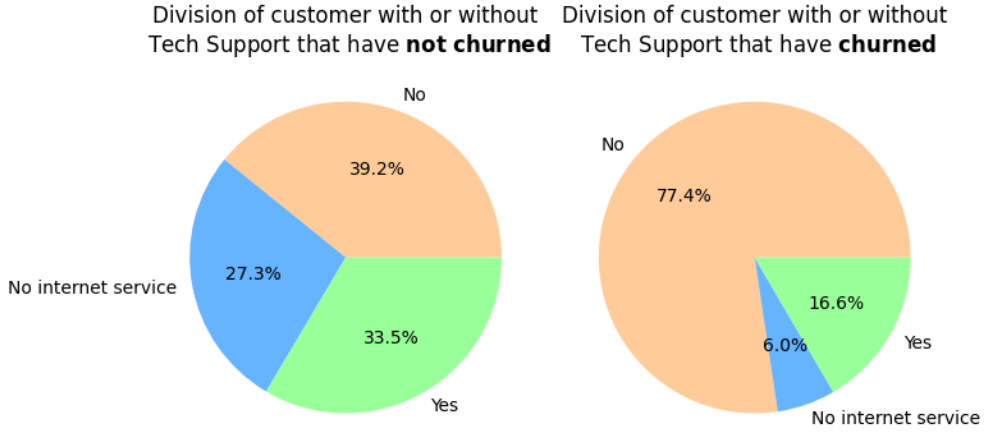
5.3.4 Analysis Online Backup vs Churn



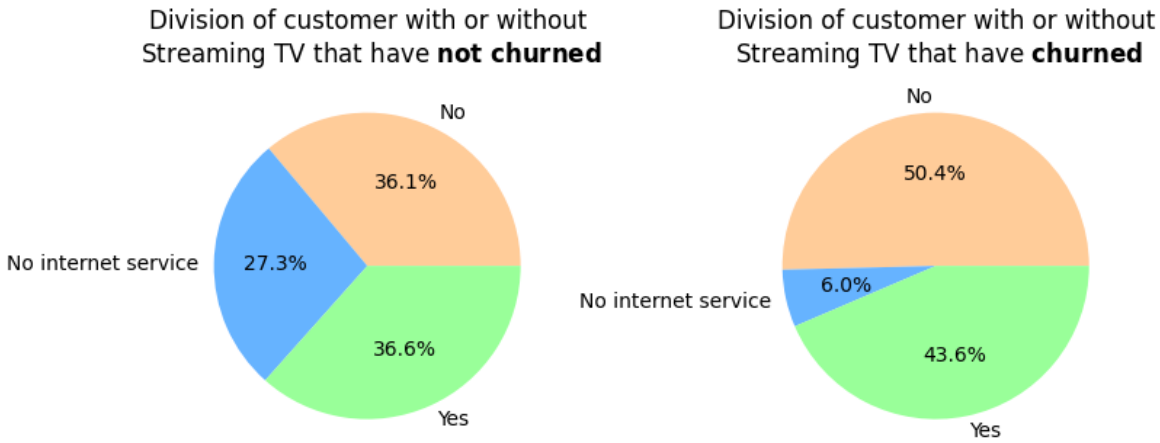
5.3.5 Analysis Device Protection vs Churn

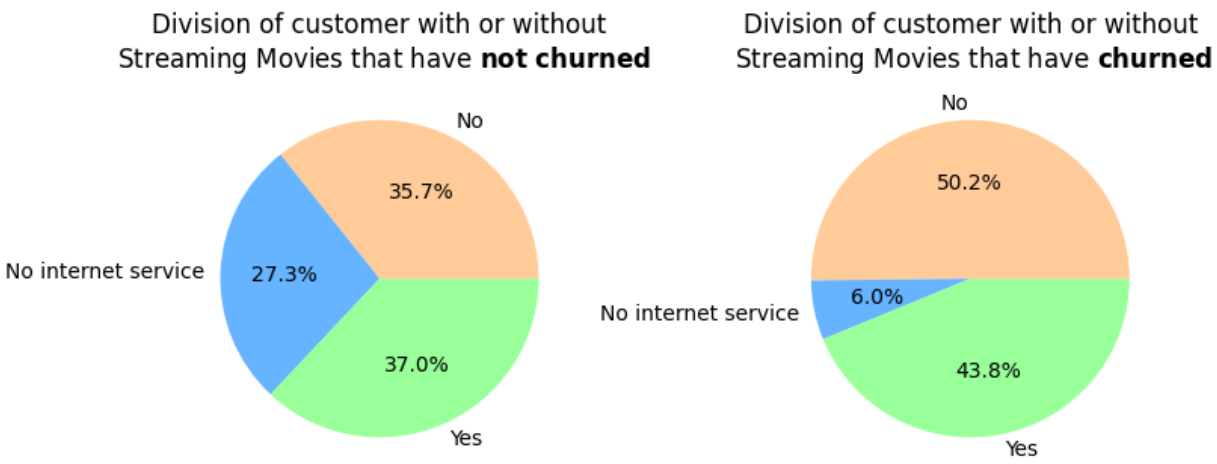


5.3.6 Analysis Tech Support vs Churn



5.3.7 Analysis Streaming TV and Streaming Movie vs Churn





5.3.8 Summary Product Features vs Churn

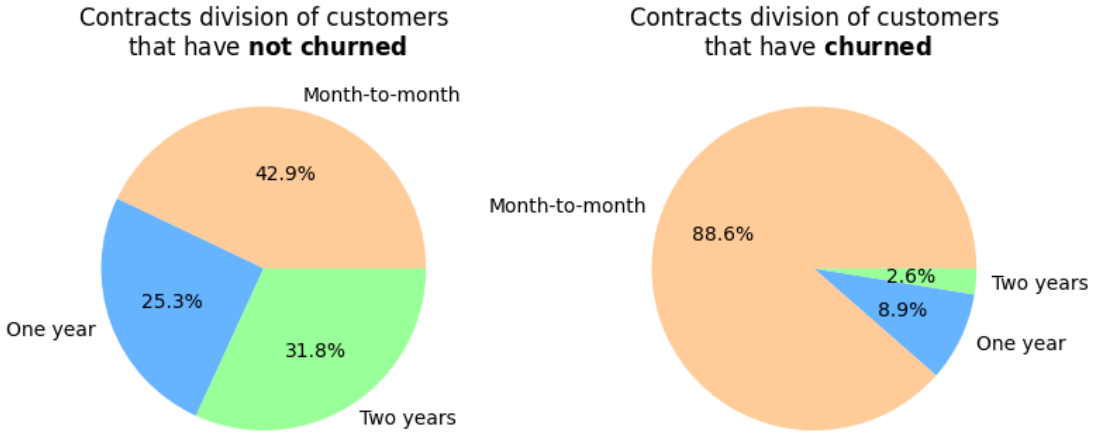
When we review all the product's features, we verify that, overall, each one by a certain extent, all the extra service that can be provided to customers reduce their chance to churn, what includes, Online Security, Online Back Up, Device Protection, Tech Support, and Streaming both for TV and Movies. Usually, telecommunication companies use offering extra service to keep customers engaged, and also to renew contracts, creating a new boundary between customers and company. Moreover, it was also possible to verify that customer over 2/3 of customers that have churned had fiber optic internet, whereas less than 1/3 of customer that remain on the company use fiber optic internet. That may indicate a problem on related to this service.

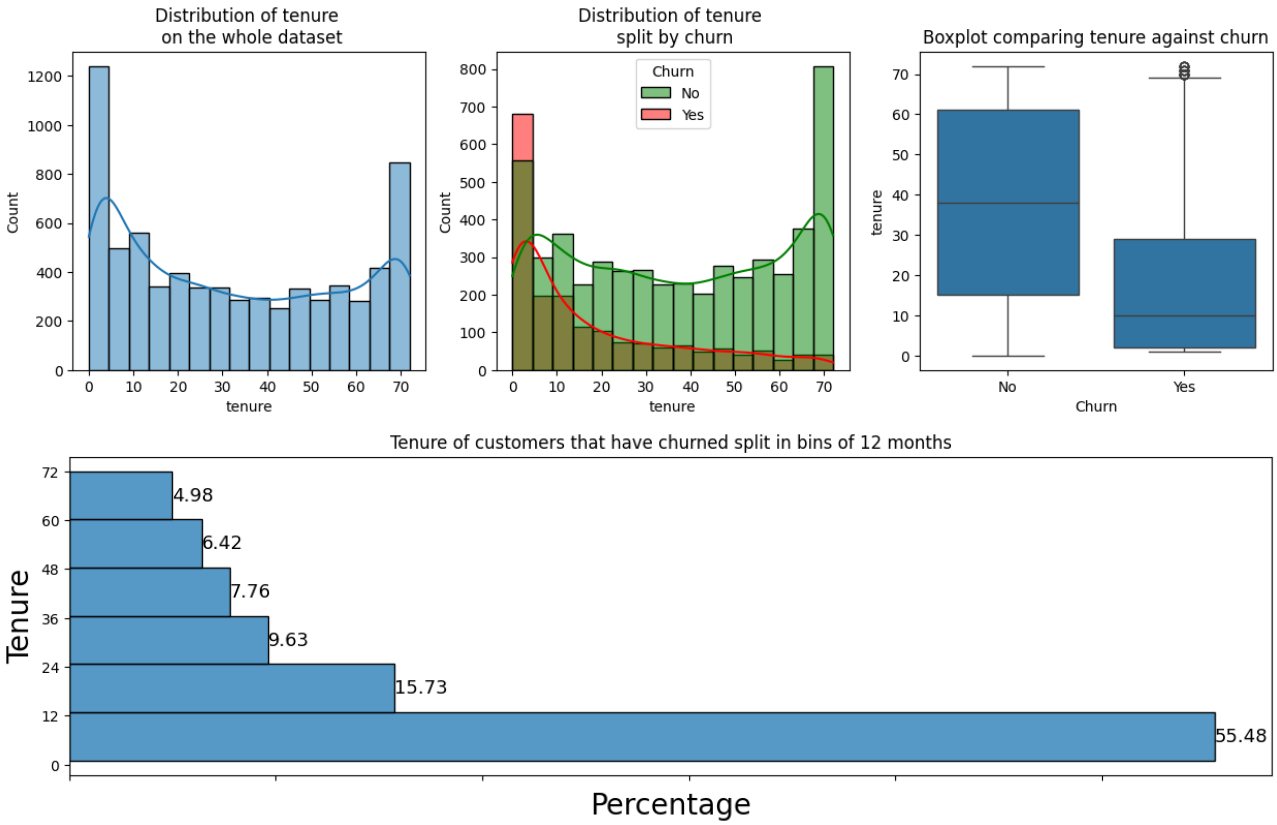
Online Security, Tech Support and Device protection seem to be the features that most influence churn so far, although other variables also do it but not so significantly.

The next step will be analysing contract features against churn.

5.4 Analysis Contract Features vs Churn

5.4.1 Analysis Contract and Tenure vs Churn





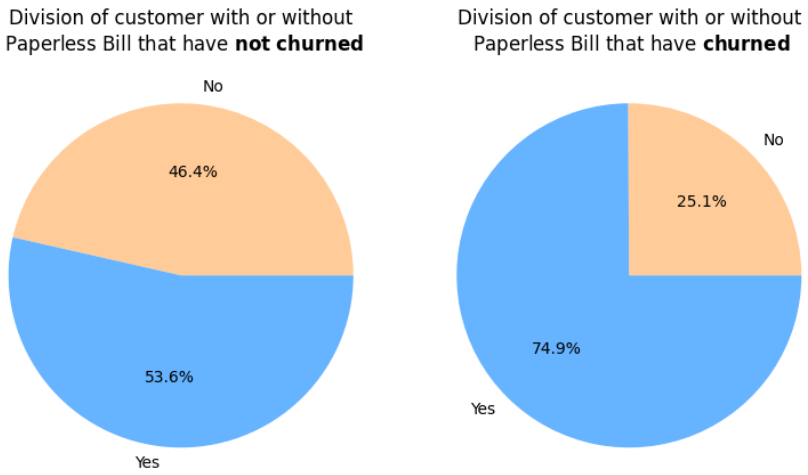
In our analysis, we compare the 'Contract' column with the clients that are within Churn, where we can identify that the clients that have a "Month-to-month" contract have the highest percentage within the clients with Churn with 88.55%. We can also verify that in proportion this is nearly the double of the same group among customers that have not churned. Moreover, the longer is the contract, the lesser is the percentage of customers within churn. Therefore, the company should promote longer contracts as an strategy to avoid churn.

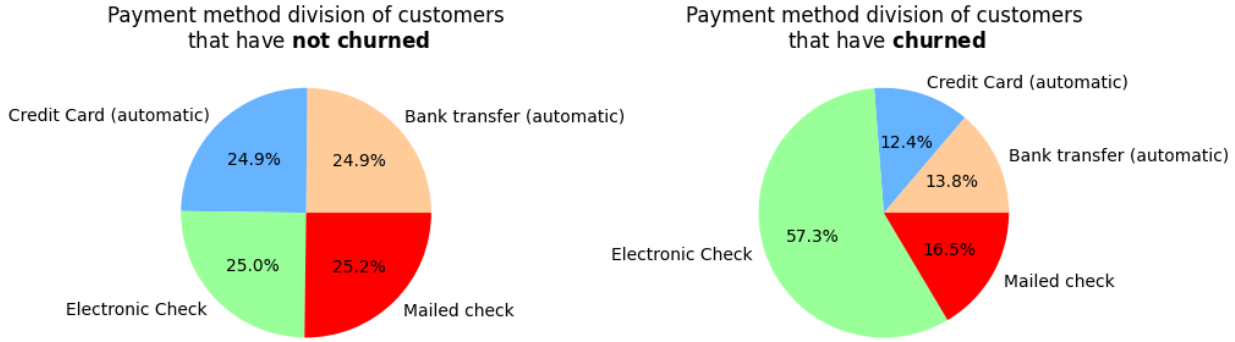
When we analyse this variable together with churn, we can verify that our variable tenure is not normal, and the values that have more volume are the farthest from the mean and median. That means that although we have discussed before that tenure average in our dataset is around 32, that number do not quite represent the data. As we can see, we have a high parcel of customers that have just onboarded on the company. The volume decreases within the next periods of time, reaching a minimum in around 40 months, and it starts increasing again, to reach another peak in around 72 months. When we verify the graph in red on the middle, we can understand that customers that churned are in majority recent on the company, and the long customers stay loyal to the company the less they tend to churn.

We can verify in the next graph that over 50% of the customers that have churned, did it in the first 12 months of contracts, and then figures steeply reduce until less than 5% of churn by customers that are with the company for over 5 years.

This indicates that the company does a good job when it comes to permanence of customers, once they are settled on the company, the chances of churning reduces drastically. However, it seems that the onboarding process needs to be watched more closely and receive improvements.

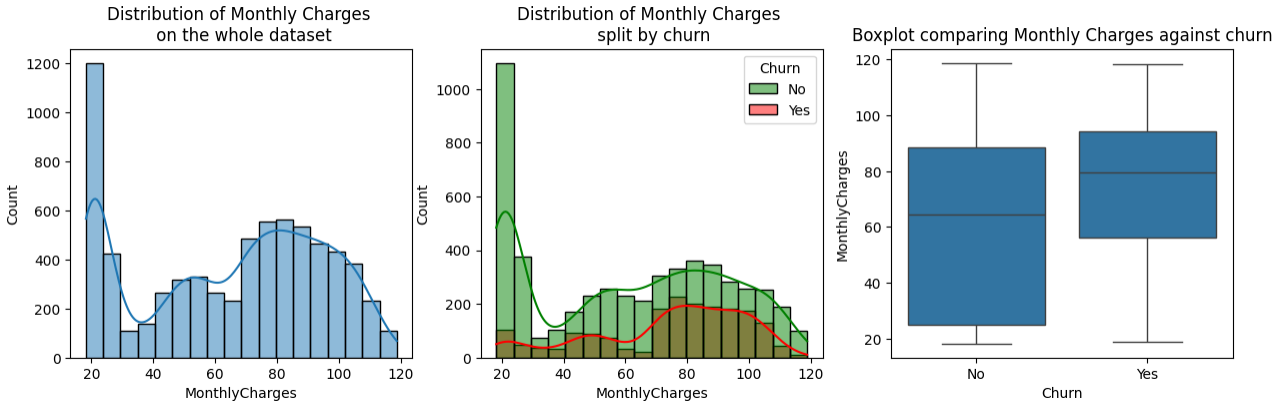
5.4.2 Analysis Paperless Billing and Payment Method vs Churn



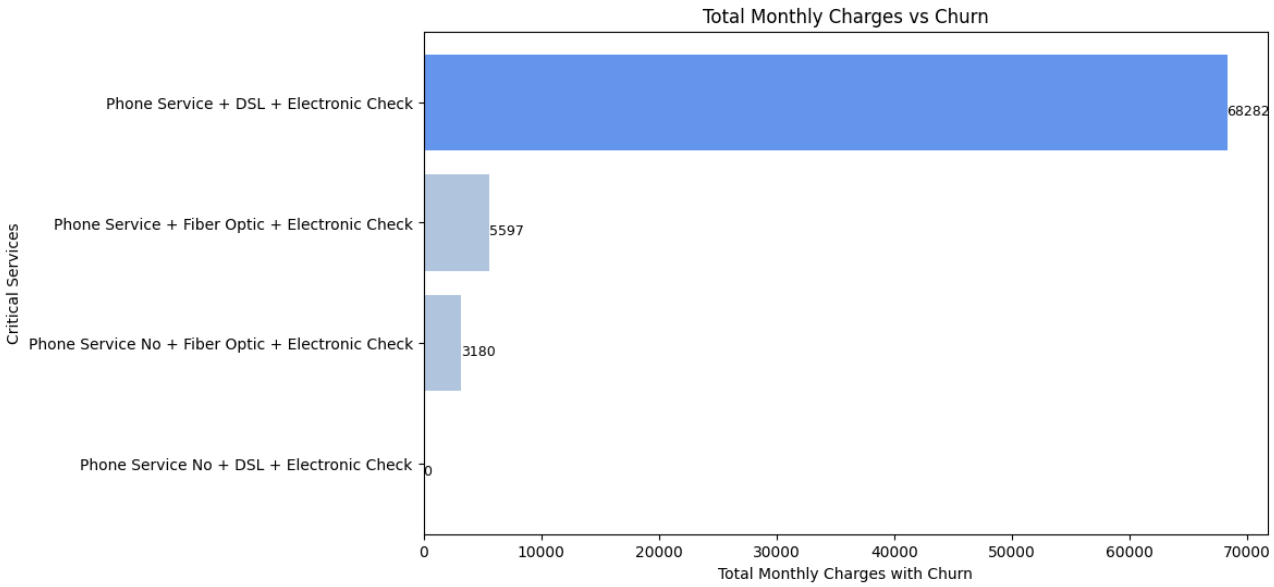


We can verify customers who have paperless billing are more frequent among those who have churned than those who have not. Thus the variable may influence in the variable churn. It happens the same with electronic check. Payment methods among those who have not churned is well balanced, basically a quarter for each category, and among those who have churned, Electronic check reaches nearly 60%.

5.4.3 Analysis Monthly Charges vs Churn



Total amount monthly lost due to churn: 139130.85



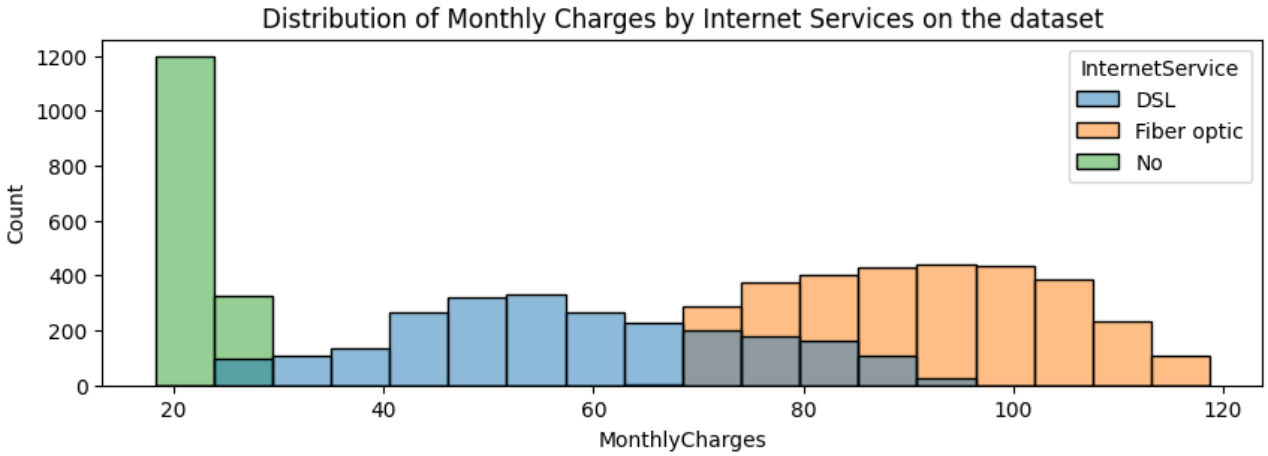
When analysing the 'MonthlyCharges' column, we can realize something important that we had not seen before, and this is that the Phone Service is associated with the Fiber Optic Internet Service.

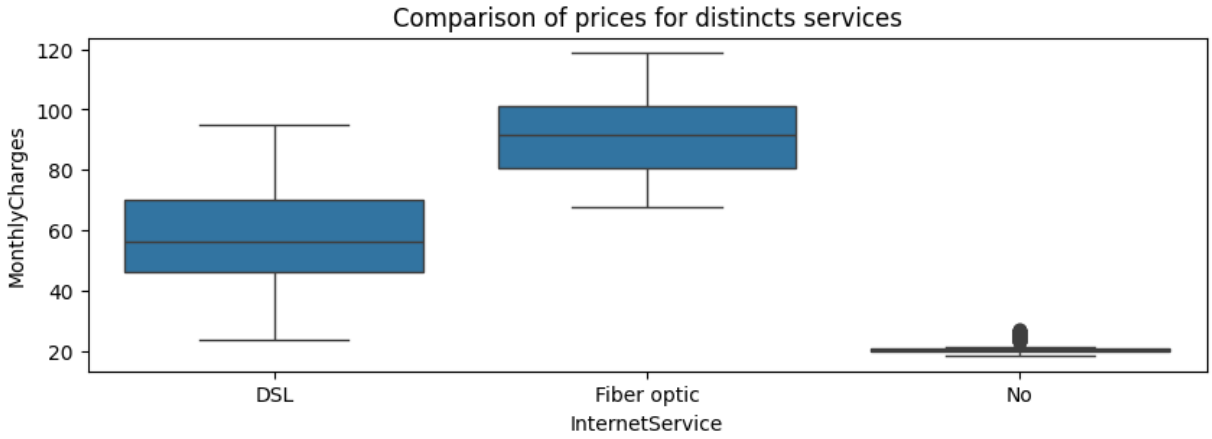
This means that if the customers does not have Phone Service and wants Internet Service, the only possible option is DSL Internet Service. On the other hand, if the customer has Phone Service, they can choose between DSL or Fiber Optic.

We can also see that the total monthly amount that the company loses due to Phone Service, DSL and Electronic check is €68,282, this is equivalent to 49% of the total monthly loss due to Churn.

We can also verify that our dataset contain a big amount of people that pay less than 30 per month, in which the majority pays a low price because they do not have internet and some of them have just DSL internet. It is also possible to see that price does not seem to influence strongly churn, once the distribution for both customer that have and have not churned is very similar. The box plot shows a higher mean for monthly prices for customer within churn, nevertheless, that happens because the high amount of customer without internet service among customer that have not churned, paying a very low price, drags the mean down.

We are also able to see, that the graph has two peaks, one around 50 and one around 80. Let us investigate this through another graphs.

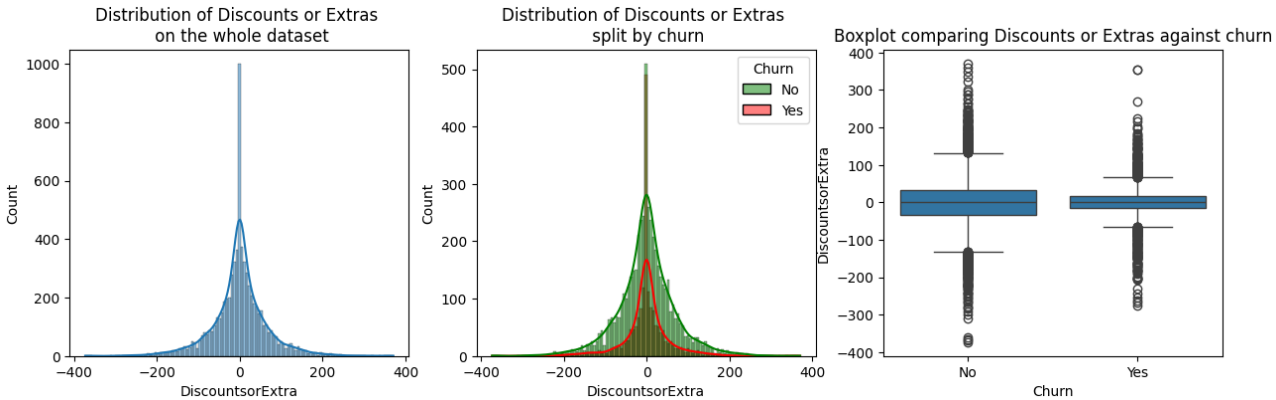




As we can verify, the graph observed before was a composition of normal graphs when we overlap the services all together. We can verify that Monthly Charges follows an approximately normal distribution for DSL service and for Fiber Optic separately, nevertheless not for both of them together. That happens because the minimum and maximum price for both service are very different one each other, as well as the mean. We can see from the boxplot the Fiber Optic services are more expensive. The fact that DSL distribution is slightly skewed to the left whereas Fiber Optic seems to be more cantered may mean that customers with DSL service contract less extras services.

The graphs gives us a hint that when modelling our machine learning models, we may perform better treating the services isolated.

5.4.4 Analysis Discounts or Extra vs Churn



5.5 Resuming our EDA

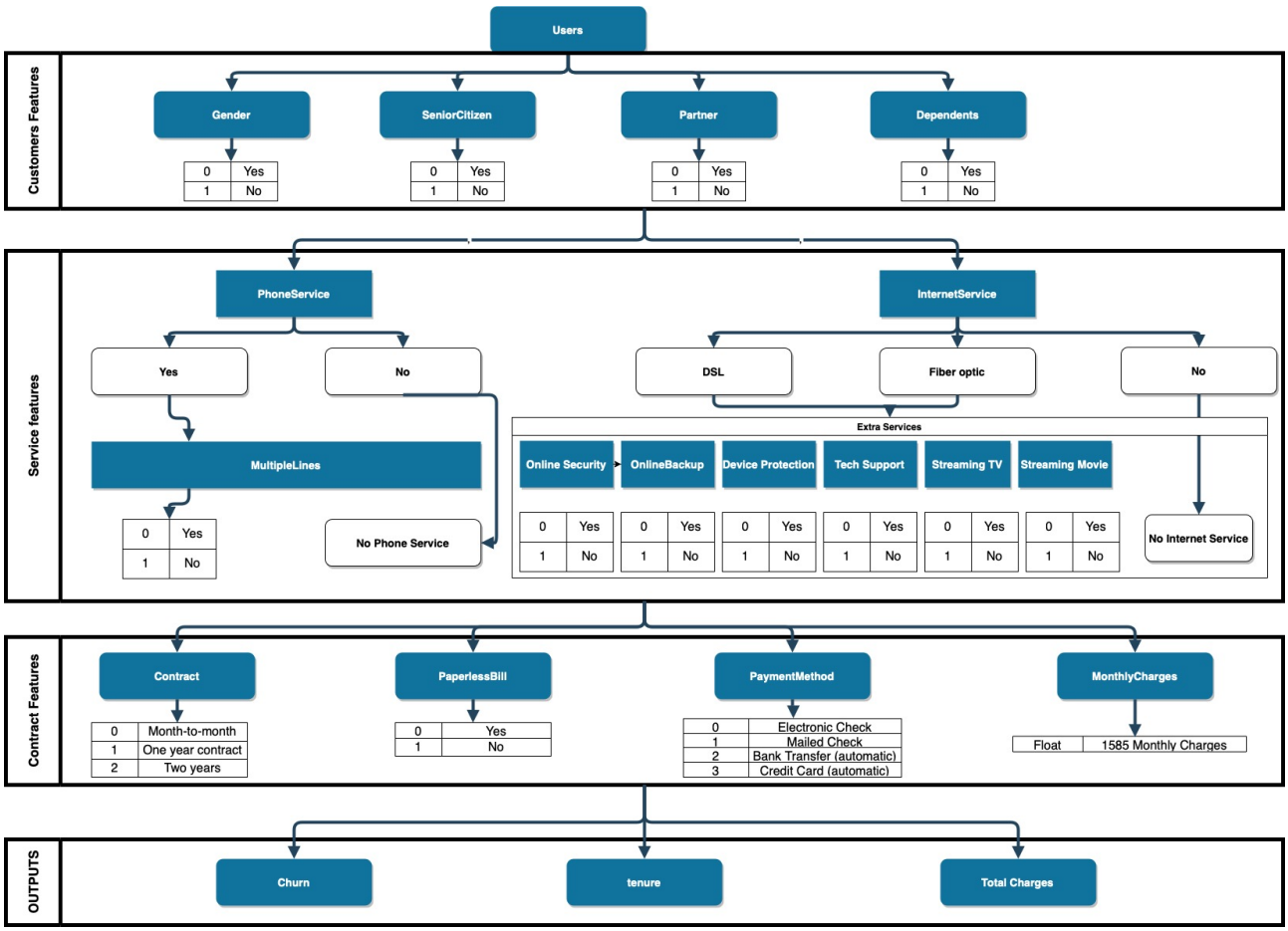
After the initial analysis, we were able to observe the structure of the data. The diagram below represents the structure of the data observed. It is important to observe, that although tenure and total charges are considered outputs on the diagram, they are being considered as variables with influence over churn so far. In addition, we also understood that a customer can only access fiber optic access if they have Phone Service, otherwise only DSL will be available for them. To make the representation more visual, this information was omitted.

We could verify through our exploration on the data that Customer features, a part from gender, has a week influence over churn. When it comes to Service Features, the features related exclusively to phone service do not influence considerably on churn, and after analysing them through statistical methods, they may dropped from our analysis. The features related to Internet Service have strong influence over churn, generally speaking, customers that have additional services tend to churn less than those that have only the most basic.

In regards to contract features, we verified that all the variables influence, except Monthly Charges, that seems to be the least reason for churning in this kind of feature. Overall, we verified that longer contracts help to avoid churn and that Electronic Check seems to be a problem and to influence strongly churn. In our opinion, as a Business Data Analytics, the analysis and previous observation of the data are essential to understand the data, understand the business and identify the objective we seek.

In regards to the outputs, we verified that customers tend to churn more when tenure is small. In other words, customer churn more in the beginning of the contract. The variable engineered to verify its influence over Churn has shown itself non related to it, and therefore it will be dropped.

The next step will be verifying the correlations on the dataset and performing statistics tests for feature selection.



5.6 Correlation Matrix

Now we will see correlation that the variables have between them. Our target variable and most of the variables in our dataset are categorical, in which some of them are not even just binary. Therefore to verify correlation between the categorical variables, it is necessary to encode them, what means that we need to transform them into numerical variables, so that algorithms can understand relationships between them. The method that will be used in this project is the One Hot Encoding, in which each category is turned into a column, and each row will show through 1's and 0's whether or not that variable is present there. It is important to observe that binary variables do not require One Hot Encoding, they will be transformed into 1's and 0's arbitrarily.

```
#Converting all binary variables into numerical

df_churn['Churn'] = df_churn['Churn'].map({'Yes': 1, 'No': 0})
df_churn['gender'] = df_churn['gender'].map({'Male': 1, 'Female': 0})
df_churn['Partner'] = df_churn['Partner'].map({'Yes': 1, 'No': 0})
df_churn['Dependents'] = df_churn['Dependents'].map({'Yes': 1, 'No': 0})
df_churn['PhoneService'] = df_churn['PhoneService'].map({'Yes': 1, 'No': 0})
df_churn['PaperlessBilling'] = df_churn['PaperlessBilling'].map({'Yes': 1, 'No': 0})
```

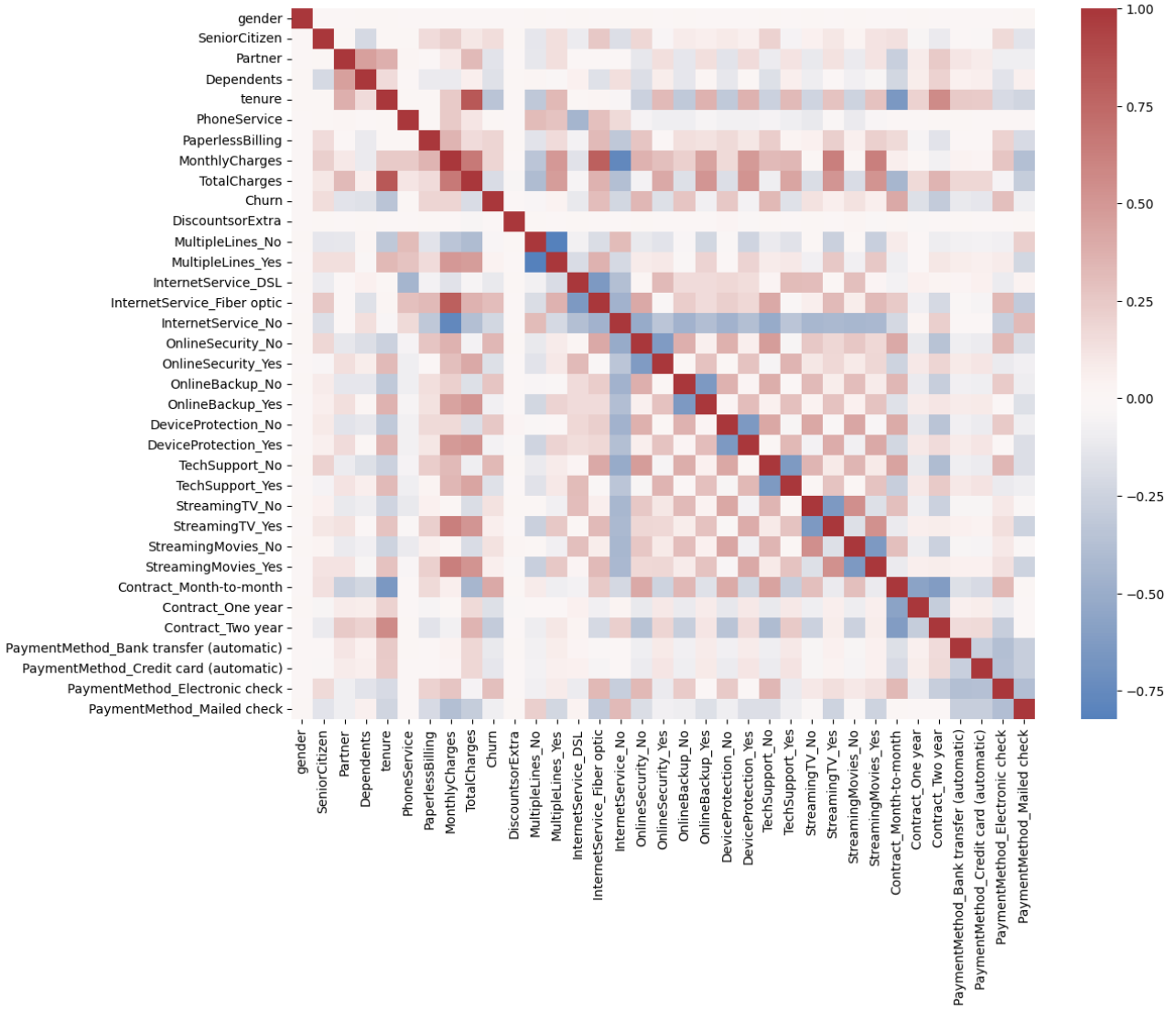
```
#Perfoming one hot encodding through the function get_dummies on the dataset

df_churn_dummies = pd.get_dummies(data=df_churn, columns = ['MultipleLines', 'InternetService', 'OnlineSecurity'
       'OnlineBackup', 'DeviceProtection', 'TechSupport', 'StreamingTV',
       'StreamingMovies', 'Contract', 'PaymentMethod'])
```

Once dummies were got, some of the features created have no more meaning, because they only carry the same information repeatedly, it is the same for all service followed by No internet service, and for multiple lines followed by no phone service, therefore this columns will be dropped.

```
df_churn_dummies.drop(columns = ['MultipleLines_No phone service','OnlineSecurity_No internet service',
                                 'OnlineBackup_No internet service', 'DeviceProtection_No internet service',
                                 'TechSupport_No internet service','StreamingTV_No internet service',
                                 'StreamingMovies_No internet service'], inplace = True)
```

All the columns were converted into numeric, and now we can perform our analysis of correlation through a correlation matrix.



We can verify from our correlation matrix that in regards to churn, the columns gender, PhoneService, DiscountsorExtras, and Multiples Lines, present a nearly white color, and therefore nearly 0 correlation. This had been point before, and the correlation matrix confirms the previous analysis. With the remaining columns, it is possible to distinguish some colors, what makes them potential for the next analysis.

The stronger correlations that we observe in the whole graph are Total Charges x tenure, Fiber Optic internet x Monthly Charges and No internet x Monthly Charges.

```
Churn                                     1.000000
Contract_Month-to-month                   0.405103
OnlineSecurity_No                         0.342637
TechSupport_No                            0.337281
InternetService_Fiber optic               0.308020
PaymentMethod_Electronic check            0.301919
OnlineBackup_No                           0.268005
DeviceProtection_No                       0.252481
MonthlyCharges                            0.193356
PaperlessBilling                          0.191825
SeniorCitizen                             0.150889
StreamingMovies_No                        0.130845
StreamingTV_No                            0.128916
StreamingTV_Yes                           0.063228
StreamingMovies_Yes                       0.061382
MultipleLines_Yes                         0.040102
PhoneService                              0.011942
DiscountsorExtra                         -0.000307
gender                                   -0.008612
MultipleLines_No                         -0.032569
DeviceProtection_Yes                     -0.066160
OnlineBackup_Yes                         -0.082255
PaymentMethod_Mailed check               -0.091683
PaymentMethod_Bank transfer (automatic)  -0.117937
InternetService_DSL                      -0.124214
PaymentMethod_Credit card (automatic)    -0.134302
Partner                                  -0.150448
Dependents                               -0.164221
TechSupport_Yes                          -0.164674
OnlineSecurity_Yes                       -0.171226
Contract_One year                        -0.177820
TotalCharges                             -0.198324
InternetService_No                       -0.227890
Contract_Two year                        -0.302253
tenure                                   -0.352229
Name: Churn, dtype: float64
```

```
1 #Verifying the values of correlation with Total Charges in descending order.
2
3 corr_matrix['TotalCharges'].sort_values(ascending=False)
```

```
TotalCharges                              1.000000
tenure                                    0.826178
MonthlyCharges                            0.651174
DeviceProtection_Yes                      0.521983
StreamingMovies_Yes                       0.520122
StreamingTV_Yes                           0.514973
OnlineBackup_Yes                          0.509226
MultipleLines_Yes                         0.468504
TechSupport_Yes                           0.431883
OnlineSecurity_Yes                        0.411651
InternetService_Fiber optic               0.361655
Contract_Two year                         0.354481
Partner                                   0.317504
PaymentMethod_Bank transfer (automatic)   0.185987
PaymentMethod_Credit card (automatic)     0.182915
Contract_One year                         0.170814
```

As we can verify, the column tenure is very correlated to the column TotalCharges, therefore, the column TotalCharges will be dropped, and just tenure will be kept.

5.7 Observations

So far, this project has been following the same structure of Part 1, however changing some concepts that were understood to have a better way to be treated, visualizations, and enhancing the discussion in regards to the business context. However, the next section 'Modelling' will not be changed in regards to technics applied, and it will be only reformatted in regards to texts, so that it can follow the same structure of the whole project. Results may slightly change due to different procedures in Data Cleaning and Engineering, but the technics performed will be exactly the same.

Afterwards, a new step of preparation will be performed according with new concepts learner throughout this project, and a new modelling will be performed. Below, the only feature that will be dropped, that was already analysed and we know it has no predictive power over churn, is DiscountsofExtra, so that we can have the same data structure previously explored.

6 Modelling

```
#Creating lis with all the model that will be applied

models = []
models.append(('NB', GaussianNB()))
models.append(('SVM', SVC(gamma='auto')))
models.append(('KNN', KNeighborsClassifier()))
models.append(('RFC', RandomForestClassifier()))
models.append(('CART', DecisionTreeClassifier()))
models.append(('LR', LogisticRegression(solver='liblinear', multi_class='ovr')))
```

In the scope of this project we will apply the 6 models listed above. All the models will be applied in different percentages of split of our dataset, 10% test and 90% training, 20% test and 80% training, and 30% test and 70% training, searching to sort the better performance and the best model. Afterwards we will try to improve this model through different technics and reach a final result.

The metric used for this part of the project was accuracy, that represents the total amount of correct predictions out of all predictions. Its use will be discussed later.

6.1 Methodology performed in first part of the project

6.1.1 Applying the models with 10% test and 90% training

```
#Dividing the dataset into dependent and independent variables
X = df_churn_dummies.drop('Churn', axis=1) #independent variables
y = df_churn_dummies['Churn'] #dependent variable (Target)

#Splitting the dataset using 10% for the test size
X_train, X_validation, y_train, y_validation = train_test_split(X, y, test_size=0.1, random_state=1)

#Printing the shapes of the parts of the dataset created
print(X_train.shape, X_validation.shape, y_train.shape, y_validation.shape)
```

```
(6338, 33) (705, 33) (6338,) (705,)
```

```python
#Testing all models listed before inside of a loop. The test includes cross validation using StratifiedKFold
#that assures every fold contains the same amount of Trues and Falses, once our dataset is unbalanced,
#and it is used 20 Splits of the dataset.

results = []
names = []
for name, model in models:
    kfold = StratifiedKFold(n_splits=20, random_state=1, shuffle=True)
    cv_results = cross_val_score(model, X_train, y_train, cv=kfold, scoring='accuracy')
    results.append(cv_results)
    names.append(name)
    print('%s: %f (%f)' % (name, cv_results.mean(), cv_results.std()))
```

```
NB: 0.736819 (0.025061)
SVM: 0.771859 (0.014836)
KNN: 0.765698 (0.015271)
RFC: 0.785578 (0.016900)
CART: 0.734623 (0.019100)
LR: 0.800719 (0.018937)
```

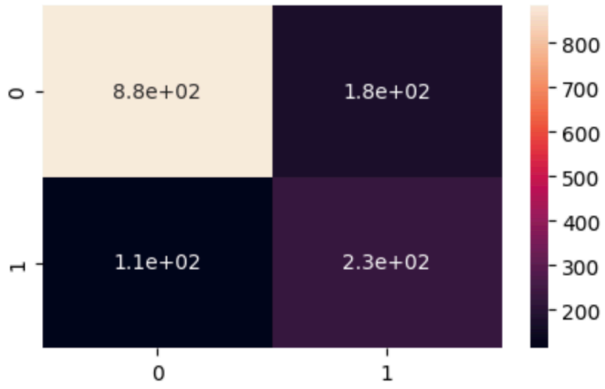
```
Accuracy Score:  0.7934705464868701

Classification Report:
              precision    recall  f1-score   support

           0       0.89      0.83      0.86      1061
           1       0.57      0.67      0.62       348

    accuracy                           0.79      1409
   macro avg       0.73      0.75      0.74      1409
weighted avg       0.81      0.79      0.80      1409


<Axes: >
```



6.1.2 Applying the models with 20% test and 80% training

```
NB: 0.735014 (0.030762)
SVM: 0.769793 (0.018029)
KNN: 0.767488 (0.021144)
RFC: 0.784870 (0.021056)
CART: 0.727896 (0.026428)
LR: 0.802631 (0.023340)
```

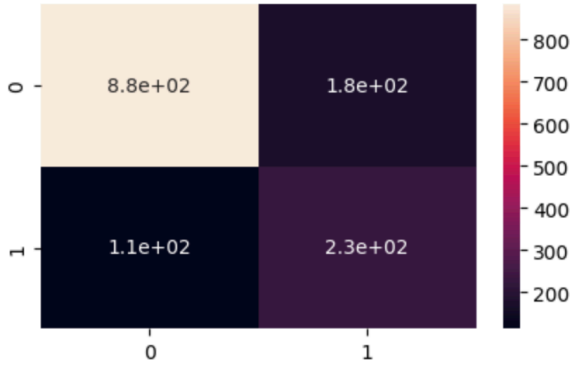
```
Accuracy Score:  0.7934705464868701

Classification Report:
              precision    recall  f1-score   support

           0       0.89      0.83      0.86      1061
           1       0.57      0.67      0.62       348

    accuracy                           0.79      1409
   macro avg       0.73      0.75      0.74      1409
weighted avg       0.81      0.79      0.80      1409

<Axes: >
```



6.1.3 Applying the models with 30% test and 70% training

```
NB: 0.731868 (0.023423)
SVM: 0.765740 (0.017405)
KNN: 0.754378 (0.024537)
RFC: 0.783391 (0.025494)
CART: 0.709527 (0.031993)
LR: 0.803057 (0.029875)
```

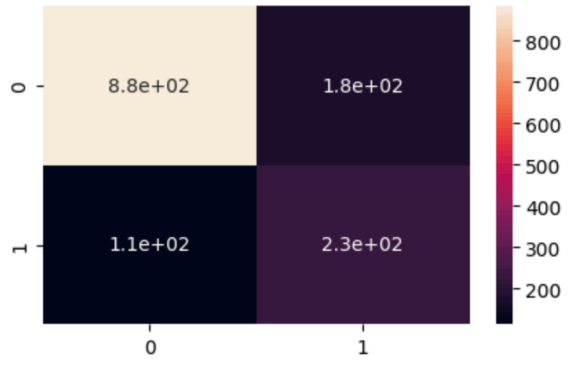
```
Accuracy Score:  0.7934705464868701

Classification Report:
              precision    recall  f1-score   support

           0       0.89      0.83      0.86      1061
           1       0.57      0.67      0.62       348

    accuracy                           0.79      1409
   macro avg       0.73      0.75      0.74      1409
weighted avg       0.81      0.79      0.80      1409

<Axes: >
```



6.1.4 Applying the Logistic Regression model with 20% testing and 80% training using the SMOTE technique

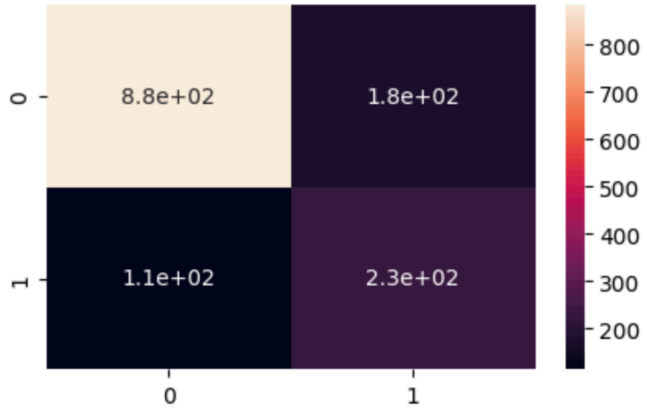
```
Accuracy Score:  0.7934705464868701

Classification Report:
              precision    recall  f1-score   support

           0       0.89      0.83      0.86      1061
           1       0.57      0.67      0.62       348

    accuracy                           0.79      1409
   macro avg       0.73      0.75      0.74      1409
weighted avg       0.81      0.79      0.80      1409

<Axes: >
```



6.1.5 Observation

After having carried out the different models with different percentages in testing and training, we can conclude the following:

For the data that we are analysing, the best resulting model is the Logistic Regression with 20% testing and 80% training, which gives us the following information:

The Accuracy Score of the model is 0.8105, which indicates that the model is capable of correctly predicting 81.05% of the cases.

The Classification Report shows us the metrics of precision (precision), recall (recall) and F1-score (f1-score) for each Churn class (0 and 1).

. Precision tells us how accurate the model is when predicting a given class.

. Recall indicates how well cases of the given class recover.

. The F1-score is a combined measure of accuracy and recovery.

After having clarified these points we can indicate that:

The report shows us that the model has a good precision for class 0 (87%) and a moderate precision for class 1 (63%). Recall is high for class 0 (89%) and low for class 1 (58%). The F1 score also reflects these trends, being higher for class 0 (88%) and lower for class 1 (61%).

The confusion matrix shows us the number of true positives (TP), false positives (FP), true negatives (TN) and false negatives (FN) in the model. In this case, the model correctly predicted 941 Churn equals 0 (TN) and 201 Churn equals 1 (TP).

However, it also incorrectly predicted 120 instances of Churn equal to 0 as Churn equal to 1 (FP) and 147 instances of Churn equal to 1 as Churn equal to 0 (FN).

In general, the Logistic Regression model has good accuracy in predicting the majority class of Churn equal to 0 but has difficulties correctly predicting the minority class of Churn equal to 1. This is because our data is unbalanced, which mostly has Churn data equal to 0 at 73.46% and Churn data equal to 1 at 26.54%.

In order to try to perform better, we apply the Logistic Regression model with 20% testing and 80% training using the SMOTE technique to add synthetic data to our minority variable equivalent to Churn equal to 1

In terms of Accuracy Score, the Logistic Regression model without applying the SMOTE technique (0.8105) performs better than the model applying the SMOTE technique (0.7956).

However, when the Classification Report metrics are analysed, it is observed that the model applying the SMOTE technique presents better performance in terms of recall for the minority class (Churn = 1).

We can see this reflected in the confusion matrix, where it is observed that the model applying the SMOTE technique has fewer false negatives (116) than the model without the technique (147).

In general, the choice of the model depends on the business objective, for this reason, an initial analysis of the business and our data is of the utmost importance. This way we will be able to detect the importance that is given to each of the metrics that we are evaluating.

As we already know in the previous analysis, we are looking for a model that has a better performance for the detection of the minority class, in this case, the Clients with Churn (1), knowing this we can opt for the model that uses the SMOTE technique.

6.2 New methodology explored in the second part of the project

6.2.1 Changing the evaluation metric used on the models

Although we have explained concepts regarding to the metrics that have been used to reach the machine learning model that has been built so far, we shall go back to our business context to analyse what these metrics represent in it. Our dataset brings information about customer characteristics, services contracted by them, and contract assets, and whether or not these customers have churned. In the context of the company and of retention of customers, both customers that have churned and customers that have not churned are important, because both of them should receive different marketing approaches to either recover a lost customer or avoid that a customer churn on future.

We have been focusing on a model that can predict accurately whether or not a customer has churned. However, the another interpretation to our model could be undercovering customers that should have churned according to our model, but have not. Because these are the customers that sectors related to retention, customer success, quality of service, in telecommunication companies must focus to avoid future churn.

Said that, we understand that we have been exploring the wrong metric to evaluate how well and useful our machine learning model is. The focus should be finding the model that reduces the number of false negatives, even though this model might identify much more false positives. The metric that better represents this aim is recall, in which we aim to increase the customer correctly identified as positive, independent of finding false positives.

The reason is because the false positive is exactly understood as customers identified by the model as positive to churn, but that have not churned, therefore customers that are likely unsatisfied and may churn soon. On the other hand, false negatives is either a breach in our system to accurately identify reasons why customers churned, or customers that have churned for reasons that are out of the context of our dataset, for instance, customers that churned because they moved to an area uncovered by the company. The first reason can be improved through tunning the model and testing different models, though, the second is unpredictable.

6.2.2 Testing our features statistically for feature selection

The Chi2 is used when we want to verify whether a categorical variables is correlated with another categorical variable, and it takes as hypothesis.

H0: The features are independent one each other.

HA: The features are not independent one each other.

If p-value < 0.05 , we reject the null hypothesis in favour of the alternative hypothesis if p-value > 0.05, we fail to reject our null hypothesis.

We performed Chi Squared test on every categorical features, and you got the results below.

	feature	Chi2 pvalue
0	gender	4.865787e-01
1	SeniorCitizen	1.510067e-36
2	Partner	2.139911e-36
3	Dependents	4.924922e-43
4	PhoneService	3.387825e-01
5	MultipleLines	3.464383e-03
6	InternetService	9.571788e-160
7	OnlineSecurity	2.661150e-185
8	OnlineBackup	2.079759e-131
9	DeviceProtection	5.505219e-122
10	TechSupport	1.443084e-180
11	StreamingTV	5.528994e-82
12	StreamingMovies	2.667757e-82
13	Contract	5.863038e-258
14	PaperlessBilling	4.073355e-58
15	PaymentMethod	3.682355e-140

As we can verify, p-value > 0.05 for gender and for PhoneService, therefore, these features will be dropped from our analysis.

```
#Dropping columns that were identified as without any correlation with our dataset.
#Total Charges is being dropped because it is almost 90% correlated with tenure, therefore we understand

df_churn.drop(columns = ['gender', 'PhoneService', 'DiscountsorExtra', 'TotalCharges'], inplace = True)
```

6.2.3 Performing One Hot Encoding

```
#Perfoming one hot encodding through the function get_dummies on the dataset
df_churn_dummies = pd.get_dummies(data=df_churn, columns = ['MultipleLines', 'InternetService', 'OnlineSecurity'
       'OnlineBackup', 'DeviceProtection', 'TechSupport', 'StreamingTV',
       'StreamingMovies', 'Contract', 'PaymentMethod'])

#Dropping the columns that would be repeated in our dummies dataset
df_churn_dummies.drop(columns = ['MultipleLines_No phone service','OnlineSecurity_No internet service',
                                 'OnlineBackup_No internet service', 'DeviceProtection_No internet service',
                                 'TechSupport_No internet service','StreamingTV_No internet service',
                                 'StreamingMovies_No internet service'], inplace = True)
```

6.2.4 Applying the models with 20% test and 80% training

Different splits on the dataset were tried, however they did not present significant changes, therefore it was decided to keep only the models for 20/80, following that this was the best split found on part 1.

```python
#Testing all models listed before inside of a loop. The test includes cross validation using StratifiedKFold
#that assures every fold contains the same amount of Trues and Falses, once our dataset is unbalanced,
#and it is used 20 Splits of the dataset.

results = []
names = []
for name, model in models:
    kfold = StratifiedKFold(n_splits=20, random_state=1, shuffle=True)
    cv_results = cross_val_score(model, X_train, y_train, cv=kfold, scoring=scoring)
    results.append(cv_results)
    names.append(name)
    print('%s: %f (%f)' % (name, cv_results.mean(), cv_results.std()))
```

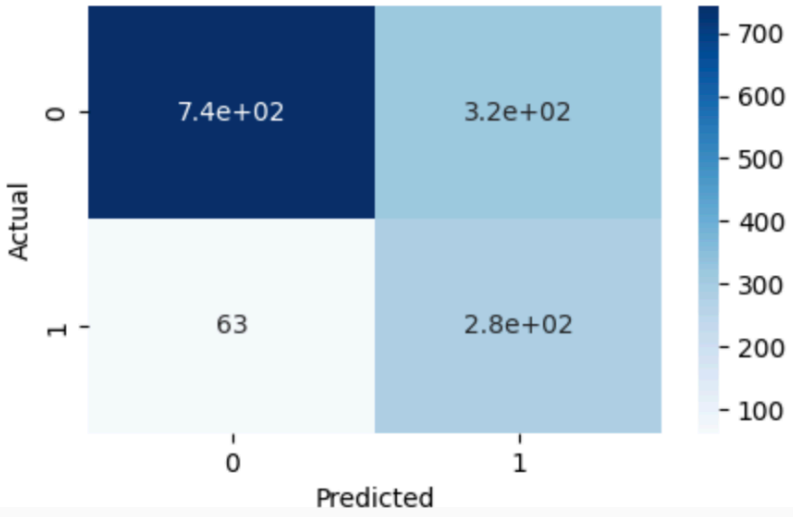
```
NB: 0.783100 (0.059021)
SVM: 0.479306 (0.062168)
KNN: 0.503648 (0.066144)
RFC: 0.486586 (0.068218)
CART: 0.512833 (0.059372)
LR: 0.541755 (0.065373)
```

```
Accuracy Score:  0.730305180979418

Classification Report:
              precision    recall  f1-score   support

           0       0.92      0.70      0.80      1061
           1       0.47      0.82      0.60       348

    accuracy                           0.73      1409
   macro avg       0.70      0.76      0.70      1409
weighted avg       0.81      0.73      0.75      1409
```



Gaussian Naïve Bayes model results.

6.2.5 Testing models applying SMOTE technic to balance the data through upsampling

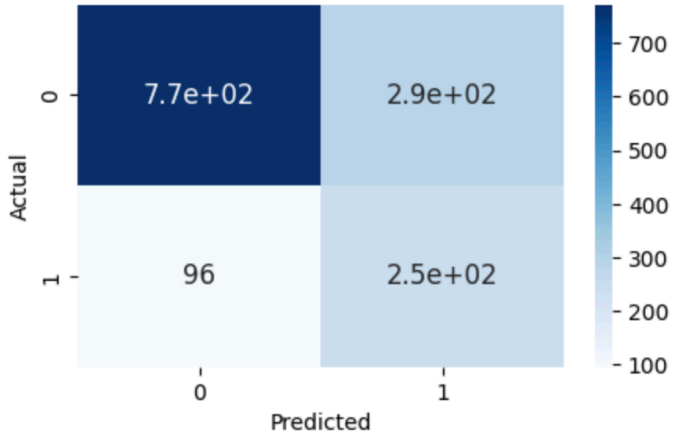
```
NB: 0.822747 (0.029680)
SVM: 0.874520 (0.025064)
KNN: 0.913183 (0.023306)
RFC: 0.847306 (0.024772)
CART: 0.811311 (0.027587)
LR: 0.828586 (0.027218)
```

```
Accuracy Score:  0.7267565649396736

Classification Report:
              precision    recall  f1-score   support

           0       0.89      0.73      0.80      1061
           1       0.47      0.72      0.57       348

    accuracy                           0.73      1409
   macro avg       0.68      0.73      0.68      1409
weighted avg       0.78      0.73      0.74      1409
```



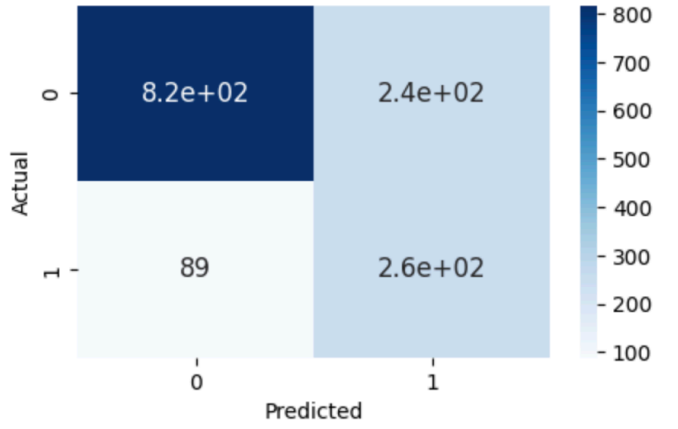
KNN Results

```
Accuracy Score:  0.7636621717530163

Classification Report:
              precision    recall  f1-score   support

           0       0.90      0.77      0.83      1061
           1       0.51      0.74      0.61       348

    accuracy                           0.76      1409
   macro avg       0.71      0.76      0.72      1409
weighted avg       0.81      0.76      0.78      1409
```



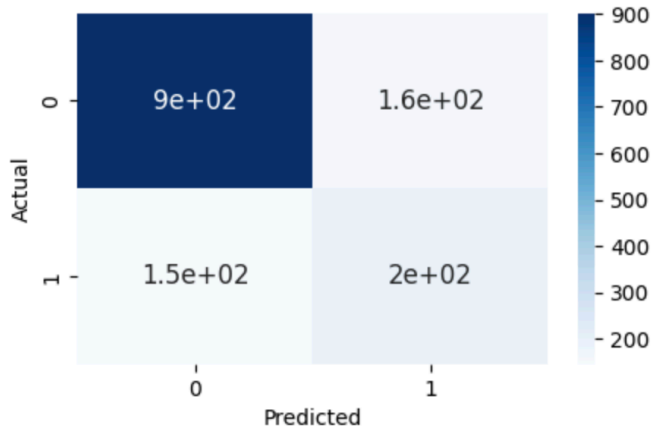
SVC Results

```
Accuracy Score:  0.7835344215755855

Classification Report:
              precision    recall  f1-score   support

           0       0.86      0.85      0.86      1061
           1       0.56      0.58      0.57       348

    accuracy                           0.78      1409
   macro avg       0.71      0.72      0.71      1409
weighted avg       0.79      0.78      0.78      1409
```



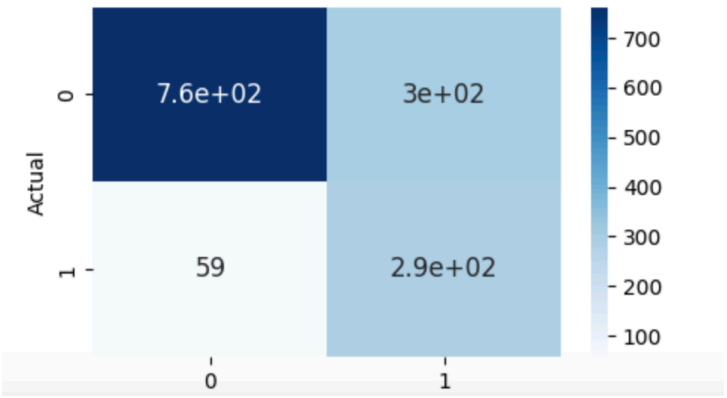
Random Forest Classifier Results

```
Accuracy Score:  0.7452093683463449

Classification Report:
              precision    recall  f1-score   support

           0       0.93      0.72      0.81      1061
           1       0.49      0.83      0.62       348

    accuracy                           0.75      1409
   macro avg       0.71      0.77      0.71      1409
weighted avg       0.82      0.75      0.76      1409
```



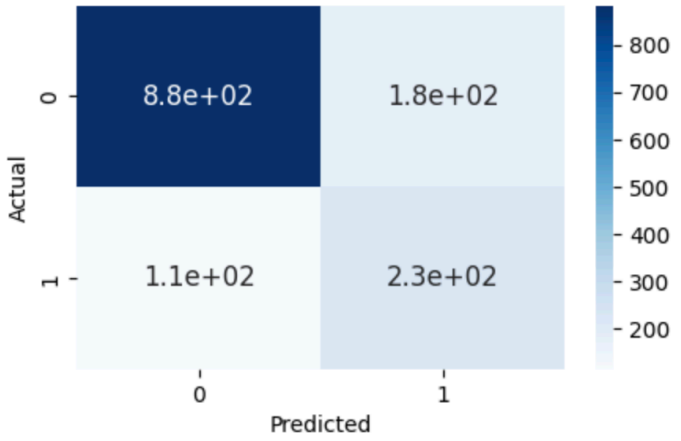
Gaussian NB Results

```
Accuracy Score:  0.7927608232789212

Classification Report:
              precision    recall  f1-score   support

           0       0.89      0.83      0.86      1061
           1       0.57      0.67      0.62       348

    accuracy                           0.79      1409
   macro avg       0.73      0.75      0.74      1409
weighted avg       0.81      0.79      0.80      1409
```



Logistic Regression Results

6.2.6 Tunning hyperparameters

Parameters to be tunned on the Gaussian NB models were taken from (Jain, 2021).

```
#Setting parameters to be tunned and fold for cross validation
parameters = {
    'var_smoothing': np.logspace(0,-9, num=100)
}
k_folds = StratifiedKFold(n_splits=20, random_state=1, shuffle=True)

# Instantiate the grid search model
grid_search_rf = GridSearchCV(estimator = GaussianNB(), param_grid = parameters,
                          cv = k_folds, n_jobs = -1,verbose = 1, scoring=scoring)
```

```
# printing the optimal Coefficient of Determination and hyperparameters
print('We can get r2 of',grid_search_rf.best_score_,'using',grid_search_rf.best_params_)
```

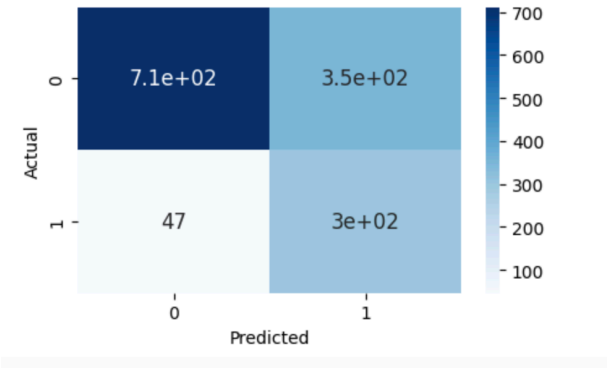
```
We can get r2 of 0.8660395453469096 using {'var_smoothing': 0.0003511191734215131}
```

```
Accuracy Score:  0.7182398864442867

Classification Report:
              precision    recall  f1-score   support

           0       0.94      0.67      0.78      1061
           1       0.46      0.86      0.60       348

    accuracy                           0.72      1409
   macro avg       0.70      0.77      0.69      1409
weighted avg       0.82      0.72      0.74      1409
```



Optimal result for Gaussian NB Model

```
#Tunning hyperparameters for Logistic Regression Model
parameters = {'solver' : ['lbfgs', 'liblinear', 'newton-cg', 'newton-cholesky'],
              'penalty' : ['l1', 'l2', 'elasticnet', None],
               'C' : [100, 10, 1.0, 0.1, 0.01]}
k_folds = StratifiedKFold(n_splits=20, random_state=1, shuffle=True)
# Instantiate the grid search model
grid_search_rf = GridSearchCV(estimator = LogisticRegression(multi_class='ovr'), param_grid = parameters,
                          cv = k_folds, n_jobs = -1,verbose = 1, scoring=scoring)
```

```
# printing the optimal Coefficient of Determination and hyperparameters
print('We can get r2 of',grid_search_rf.best_score_,'using',grid_search_rf.best_params_)
```

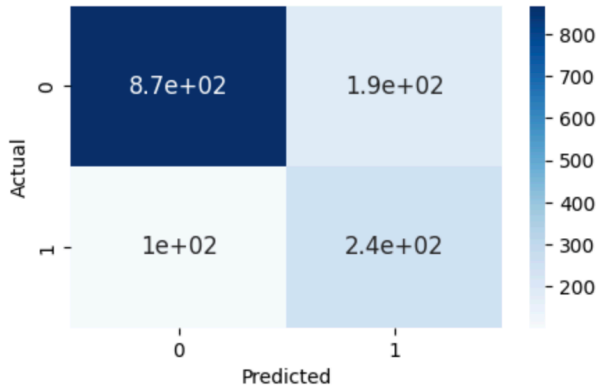
```
We can get r2 of 0.8429327492304048 using {'C': 0.1, 'penalty': 'l2', 'solver': 'newton-cg'}
```

```
Accuracy Score:  0.7892122072391767

Classification Report:
              precision    recall  f1-score   support

           0       0.89      0.82      0.85      1061
           1       0.56      0.70      0.62       348

    accuracy                           0.79      1409
   macro avg       0.73      0.76      0.74      1409
weighted avg       0.81      0.79      0.80      1409
```



Optimal result for Logistic Regression Model

```
#Setting parameters to be tunned and fold for cross validation
parameters = {'C': [0.1, 1, 10, 100, 1000],
              'gamma': [1, 0.1, 0.01, 0.001, 0.0001],
              'kernel': ['rbf']}
k_folds = StratifiedKFold(n_splits=20, random_state=1, shuffle=True)

# Instantiate the grid search model
grid_search_rf = GridSearchCV(estimator = SVC(), param_grid = parameters,
                           cv = k_folds, n_jobs = -1,verbose = 1, scoring=scoring)
```

```
# printing the optimal Coefficient of Determination and hyperparameters
print('We can get r2 of',grid_search_rf.best_score_,'using',grid_search_rf.best_params_)
```

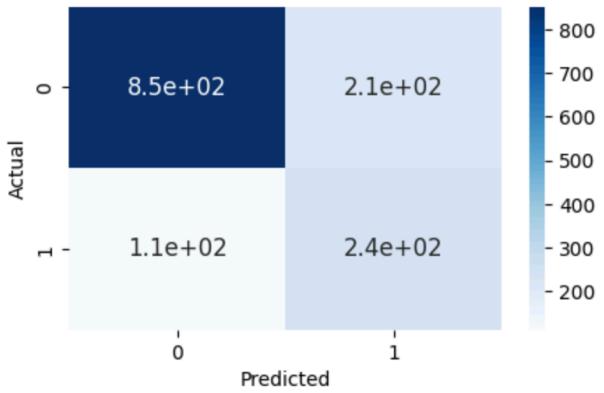
```
We can get r2 of 0.891560502012787 using {'C': 1, 'gamma': 0.1, 'kernel': 'rbf'}
```

```
Accuracy Score:  0.7735982966643009

Classification Report:
              precision    recall  f1-score   support

           0       0.88      0.80      0.84      1061
           1       0.53      0.68      0.60       348

    accuracy                           0.77      1409
   macro avg       0.71      0.74      0.72      1409
weighted avg       0.80      0.77      0.78      1409
```



Optimal result for Support Vector Machine Classifier

6.2.7 Conclusions about modelling

It is possible to verify that the algorithm that better fit our data was Guassian Naive Bayes. After applying SMOTE technic the model seemed to perform better in our training dataset, reaching recalls over 0.84, however when applying the models in the the validation dataset, the results are very similar to what we had without applying SMOTE.

Nevertheless, tunning hyperparameters was tried for both including SMOTE and not including it, and it was discovered that when it is not included, the best hyperparameters do not improve the model significantly, whereas after applying it, the model performed 3% better in the metric we are considering the most important, and we could further reduce the number of false negatives.

Therefore, the best model is the Gaussian Naive Bayes, applying SMOTE technic, and setting to it best parameter tunned. The model performs with overall accurary of 0.733, and it predicts customers that has churned correctly 85% of attempts. The model has low precision when identifying who churned, nearly 50% of its predictions are customers that

have not churned, however, this might be an indication that this customers may churn soon or later.

6.3 Feature Importance

Our best model to predict churn does not have a direct way to extract how each feature has influenced in its decisions, therefore, we will observe feature importance to the Random Forest Classifier model.

The Random Forest Classifier (RFC) performed over the balanced dataset using smote presented the best accuracy among all the models trained in this project, its predictions are 79% accurate. However, it has not performed quite well reducing the number of false negatives, so that in our business context, Gaussian Naive Bayes would be a better choice.

As we can verify and goes towards the same patterns we had seen in our EDA, is that Contract-Month-to-Month, lack of online security and customers that pay with electronic check are the features that contributes the most, according to this model, to customers to churn.

7 CONCLUSION

After analysing the dataset and build our models, we could get to a better understanding of our business and how to predict and prevent churn. During the confection of the second part of this project, the part 1 was entirely reviewed, different procedures were taken to clean and engineer the data, the visualizations were changed in the EDA, and the way conclusion were drawn from the EDA was also changed. EDA analyse was made more complete and reached better results.

A part from reviewing the first part, the business understanding was enhanced, and it was discussed that changing of the metric being used to evaluate the models. We found that Recall is a better metric for our business context, once for us it is very important to classify correctly customer that have churned, even if we get false positives.

New models were trained based in this new metric and we achieved 86% of success identifying customers that have churned, using a Gaussian Naive Bayes model trained using 80% of the dataset. Cross validation was performed, seeking to avoid bias in the analysis, as well as hyperparameter tunning. The tunning improved our desired result in 3%.

It was found that Contract-month-to-month, Electronic Check and Online Security are features that influence the chances of customers churn, and we also found out that Fiber Optic internet service needs improvement. Overall, every extra service that customers acquire seems to influence positively its experience, because the lack of them increases chances of churning.

As next steps, it would be interesting to split the dataset into 2 different analysis, one only for Fiber Optic Service, and one only for DSL. They seem to have very different publics, what might lead to more specific results for each one of them. Another point to improve is

analysing correlation among variables, apart from only the target variable. Other models that have not been tested in this project could perform better and bring different results.

8 Bibliography

Dino, L., 2022. *Medium.* [Online] Available at: https://medium.com/@24littledino/two-sample-chi-square-test-in-python-b9f2db89dc2b
[Accessed 16 11 2023].

Jain, K., 2021. *Analytics Vidhya.* [Online] Available at: https://medium.com/analytics-vidhya/how-to-improve-naive-bayes-9fa698e14cba
[Accessed 18 11 2023].

Jain, R., 2020. *Medium.* [Online] Available at: https://medium.com/@ritesh.110587/correlation-between-categorical-variables-63f6bd9bf2f7
[Accessed 15 11 2023].

Kübler, R., 2021. *Towards Data Science.* [Online] Available at: https://towardsdatascience.com/learning-by-implementing-gaussian-naive-bayes-3f0e3d2c01b2
[Accessed 17 11 2023].

Sharma, A., 2021. *Analytics Vidhya.* [Online] Available at: https://www.analyticsvidhya.com/blog/2021/01/gaussian-naive-bayes-with-hyperpameter-tuning/
[Accessed 15 11 2023].

Stack Overflow [online] (May 10, 2021) Available at: https://stackoverflow.com/questions/67474348/how-do-i-create-a-bar-chart-with-percentage-values-in-python-plotly-express

GeeksforGeeks [online] (16 Jul, 2020) Available at: https://www.geeksforgeeks.org/python-binomial-distribution/?ref=gcse

Scribbr [online] Published on November 5, 2020 by Pritha Bhandari. Available at: https://www.scribbr.com/statistics/standard-normal-distribution/

Kaggle, Updated 5 Years Ago by Blastchar [online] Available at: https://www.kaggle.com/datasets/blastchar/telco-customer-churn

W3Schools, Copyright 1999-2023 by Refsnes Data [online] Available at: https://www.w3schools.com/statistics/statistics_standard_normal_distribution.php

Copyright © the Python Graph Gallery 2018 [online] Available at: https://www.python-graph-gallery.com/barplot/

Scribbr [online] Published on May 13, 2022 by Shaun Turney. Available at: https://www.scribbr.com/statistics/poisson-distribution/

By Abhishek Wasnik / October 26, 2020. Available at: https://www.askpython.com/python/normal-distribution

By GreekDataGuy / Jan 2, 2020. Available at: https://towardsdatascience.com/conditional-probability-with-a-python-example-fd6f5937cd2

GeeksforGeeks., 2021. Exploratory Data Analysis by KattamuriMeghna. Available at: https://www.geeksforgeeks.org/exploratory-data-analysis-in-python/

Scribbr, Published on January 28, 2020 by Rebecca Bevans. Available at: https://www.scribbr.com/statistics/statistical-tests/

Scribbr, Published on December 8, 2021 by Pritha Bhandari. Available at: https://www.scribbr.com/statistics/missing-data/

W3schools, Pandas DataFrame astype() Method. Available at: https://www.w3schools.com/python/pandas/ref_df_astype.asp

Microsoft, Normalize Data component. Article 11/04/2021. Available at: https://learn.microsoft.com/en-us/azure/machine-learning/component-reference/normalize-data?view=azureml-api-2

Analytics Vidhya by Aniruddha Bhandari — Published On April 3, 2020. Available at: https://www.analyticsvidhya.com/blog/2020/04/feature-scaling-machine-learning-normalization-standardization/

GeeksforGeeks. ML | Principal Component Analysis(PCA), by aishwarya.27. Available at: https://www.geeksforgeeks.org/ml-principal-component-analysispca/

Medium Dec 25, 2019, by Aayush Bajaj. Available at: https://towardsdatascience.com/what-does-your-classification-metric-tell-about-your-data-4a8f35408a8b#:~:text=A%20low%20recall%20score%20