Team:

- FERCHICHI LOUAY
- HADDAD SKANDER
- HASNAOUI FARES
- MAHFOUDH HAZEM
- TARHOUNI ERIJ

Customer Churn Prediction

PROFESSOR : NCIB LOTFI
YEAR : 2020-2021
COLLEGE : ESPRIT (PRIVATE HIGHER SCHOOL OF ENGINEERING AND TECHNOLOGY)

TABLE OF FIGURES

Abstract

Business nowadays is a very competitive world. Especially, the IT fields like telecommunication that face complex challenges due to the huge number of service providers competitors. In fact, the cost of acquiring new customers is much higher than the cost of retaining the existing customers because of marketing methods that the company has to resort to. Therefore, isn't time for the telecom industries to improve their market value by working on retaining their existing customers?
To achieve this goal, several data mining techniques have been proposed and improved in the last decades to predict churners using heterogeneous customer records. Our project reviews the different categories of customer data available in open datasets, predictive models and performance metrics used for churn prediction in the telecom industry. For this purpose, we will use CRISP-DM methodology as well as different machine learning algorithms such as (KNN (K Nearest Neighbours), Naive Bayes classifier, k means ...)

Key Words: customer churn prediction, predictive models, CRISP-DM, KNN, Naïve Bayes classifier, k means.

INTRODUCTION

In telecommunication paradigm, Churn is defined to be the activity of customers leaving the company and discarding the services offered by it due to dissatisfaction of the services and/or due to better offering from other network providers within the affordable price tag of the customer. This leads to a potential loss of revenue/profit to the company. Also, it has become a challenging task to retain the customers. Therefore, companies are going behind introducing new state of the art applications and technologies to offer their customers as much better services as possible so as to retain them intact. Before doing so, it is necessary to identify those customers who are likely to leave the company in the near future in advance because losing them would result in significant loss of profit for the company. This process is called Churn Prediction thus data science teams use the CRISP-DM methodology.

In this project, we review the existing works on churn prediction in three different perspectives: datasets, methods, and metrics. Firstly, we present the details about the availability of public datasets and what kinds of customer details are available in each dataset for predicting customer churn. Secondly, we compare and contrast the various predictive modelling methods that have been used in the literature for predicting the churners using different categories of customer records, and then quantitatively compare their performances.

Finally, we summarize what kinds of performance metrics have been used to evaluate the existing churn prediction methods. Analysing all these three perspectives is very crucial for developing a more efficient churn prediction system for telecom industries. While there are other churn prediction surveys available in the literature, they primarily focused on different modelling techniques. To the best of our knowledge, none of those surveys reviewed the datasets and metrics for evaluating the churn prediction models. Hence, we believe that this survey can provide a roadmap for both researchers and customer relationship managers to better understand the domain and challenges in detail.

CHAPTER 1: CRISP-DM

INTRODUCTION

CRISP-DM **(Cross-Industry Standard Process for Data Mining)** is a comprehensive data mining methodology and process model that provides data mining experts or even novices with a complete organized blueprint for conducting a data mining project.

CRISP-DM is a process made up of six different phases. These include **Business Understanding**, **Data Understanding**, **Data Preparation**, **Modeling**, **Evaluation** and **Deployment**. These phases are, at a nominal level, approached sequentially, however the process itself is iterative, meaning that any models are designed to be improved by subsequent knowledge gained throughout the process.

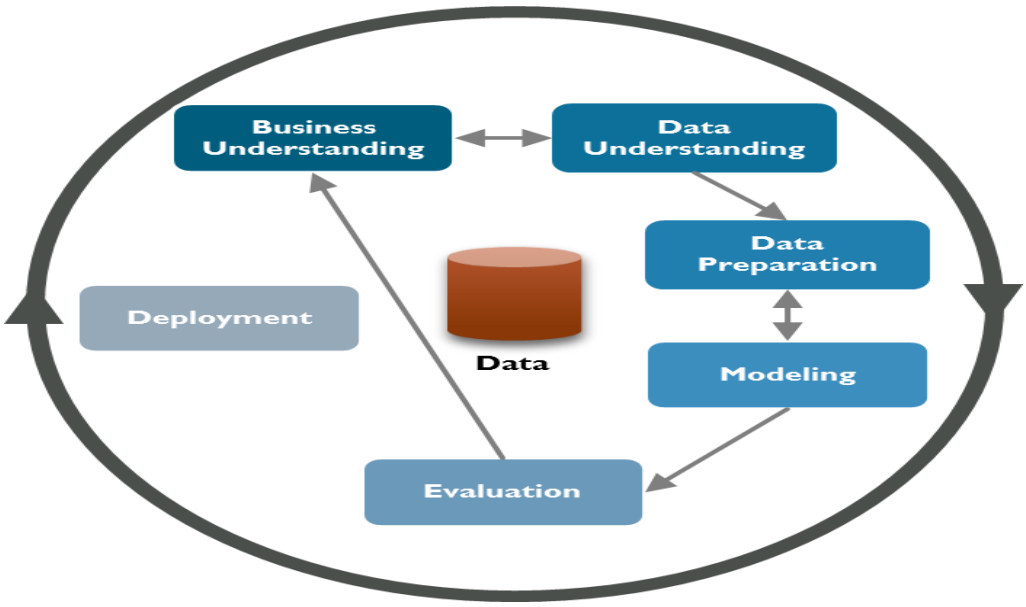

Figure 1: CRISP-DM summary

- **Business Understanding**

Understanding a business involves identifying which problems a business has that they wish to solve.

Understanding project objectives and requirements.

- **Data Understanding**

Initial data collection and familiarization: discover first insights into the data, Identify data quality issues, detect interesting subsets to form hypotheses for hidden information.

- **Data Preparation**

Analyzing data, selecting attributes and values to construct the final dataset from the initial raw data. The data preparation process is extensive and tends to take approximately 80% of the project time.

- **Modeling**

Modeling techniques are selected and applied. Since some techniques have specific requirements regarding the form of the data, there can be a loop back here to data preparation.

- **Evaluation**

Test the models to ensure they generalize against unseen data and that all key business issues have been sufficiently considered

Identify business issues that should have been addressed earlier and select the final models.

- **Deployment**

Deploying a code representation of the model into an operating system to score or categorize new unseen data as it arises and to create a mechanism for the use of that new information in the solution of the original business problem.

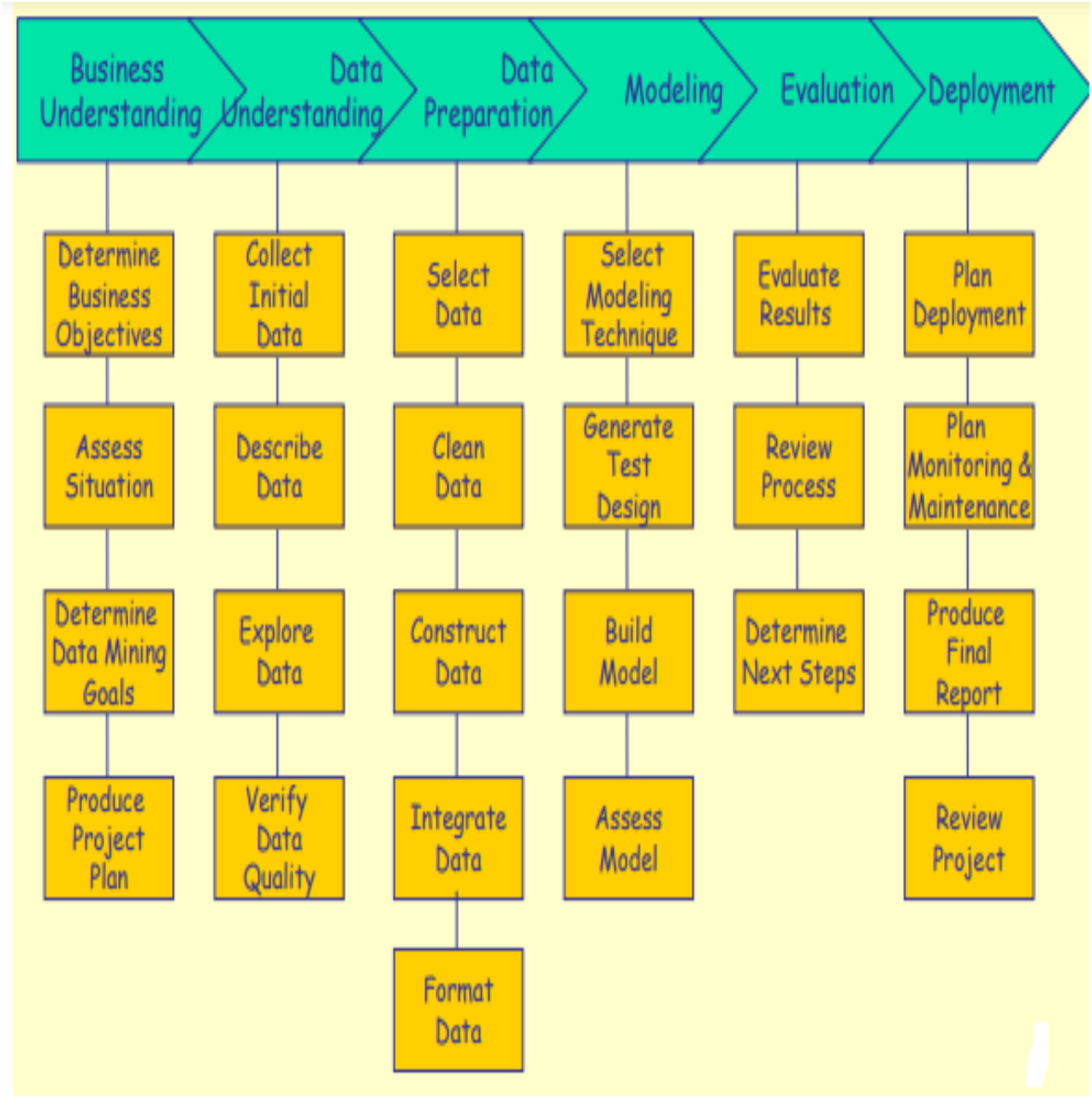

Figure 2: CRISP-DM steps details

CHAPTER 2: BUSINESS UNDERSTANDING

INTRODUCTION

Arguably the most important part of any project is establishing what the goals of the project and success criteria are, and how to measure them. In our case, we are studying customer churn and its impacts on companies business like telecommunication companies, so the most important aspect at this stage is to precisely define what churn is in the given context.

The dataset which used for experiment is Telco Customer Churn dataset that obtained from https://www.kaggle.com/ylchang/telco-customer-churn-1113. The dataset contains 33 features, one label and 7043 records. The dataset contains features with categorical and numeric types. The proposed model can only process numeric data, so there are some steps we should respected them based on CRISP-DM methodology.

What is customer churn?

Customer churn is the percentage of customers that stopped using a company's product or service during a given time period and one of the most important metrics for businesses to evaluate.

Why is it important to care about customer churn?

Departing customers obviously take their revenue with them, which will negatively affect the commercial profitability of the company, in other ways, all companies seek to reduce churn because it drags down their growth.

In fact, acquiring new customers costs much more than keeping them, it needs many sales and marketing efforts and so huge amounts of money and companies sacrifice such costs on new customers in the hope that these investments will be paid back several times over during the customer's lifetime. But what if customers leave earlier than expected? We can imagine the huge amount of money the company may lose. The longer the brand can hold onto their customers, the greater value each customer is worth over their lifespan.

Business objectives

The main objectives presented by our project are to:

- Identify the possible churners in advance before they leave the network.

- Improve customer satisfaction by taking the required retention policies to attract the likely churners and to retain them.

- Select the model with the highest accuracy in order to find the best solution to decrease the number of churners.

Business success criteria:

- Attracting the possible churners and maintain the loyal customers.

- A minimum increase in churners.

- Remaining competitive in telecommunication industry.

CONCLUSION

Ultimately, the goal for businesses is twofold: identify customers who are at risk of leaving and put more resources into keeping existing customers. Doing both of these helps reduce churn and save money. For such reasons, it was important for companies to resort to data science solutions in order to predict leaving customers.

CHAPTER 3: DATA UNDERSTANDING

INTRODUCTION:

During this chapter, we'll be going through the data comprehension by describing its content and exploring it, in order to gain some insights through primary intuition.
In the Telco_customer_churn.csv file we have observations that contain the input features as well as target variable: Churn Value. We will analyse it to identify all features helpful to predict if a customer is churning by applying different analysing techniques and methods.

DATASET EXPLANATION

The dataset includes information about:

* Customers who left within the last month — the column is called Churn

* 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

* Customer account information — how long they've been a customer, contract, payment method, paperless billing, monthly charges, and total charges

* Demographic info about customers — gender, age range, and if they have partners and dependents

The dataset consists of 7,044 customers and 33 variables:

* **CustomerID:** A unique ID that identifies each customer.

* **Count:** A value used in reporting/dashboarding to sum up the number of customers in a filtered set.

* **Country:** The country of the customer's primary residence.

* **State:** The state of the customer's primary residence.

* **City:** The city of the customer's primary residence.

* **Zip code:** The Postal code of the customer's primary residence.

* **Lat Long:** The combined latitude and longitude of the customer's primary residence.

* **Latitude:** The latitude of the customer's primary residence.

* **Longitude:** The longitude of the customer's primary residence.

* **Gender:** Whether the customer is a male or a female

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

* **Partner:** Whether the customer has a partner or not (Yes, No)

* **Dependents:** Whether the customer has dependents or not (Yes, No)

* **Tenure:** Number of months the customer has stayed with the company

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

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

* **InternetService:** Customer's internet service provider (DSL, Fiber optic, No)

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

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

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

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

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

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

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

* **PaperlessBilling:** Whether the customer has paperless billing or not (Yes, No)

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

* **MonthlyCharges:** The amount charged to the customer monthly

* **TotalCharges:** The total amount charged to the customer

*** CLTV (Customer LifeTime Value):** The total worth to a business of a customer over the whole period of their relationship. The higher the value, the more valuable the customer. High value customers should be monitored for churn.

*** ChurnLabel:** Yes = the customer left the company this quarter. No = the customer remained with the company.

*** ChurnValue:** 1 = the customer left the company this quarter. 0 = the customer remained with the company.

*** ChurnReason:** A customer's specific reason for leaving the company.

FEATURES ANALYSIS:

- **churn value:**

  Whether the customer churned or not (0 or 1), the data in this column is unbalanced seeing that churn rate(yes) is far less than the non-churners'.

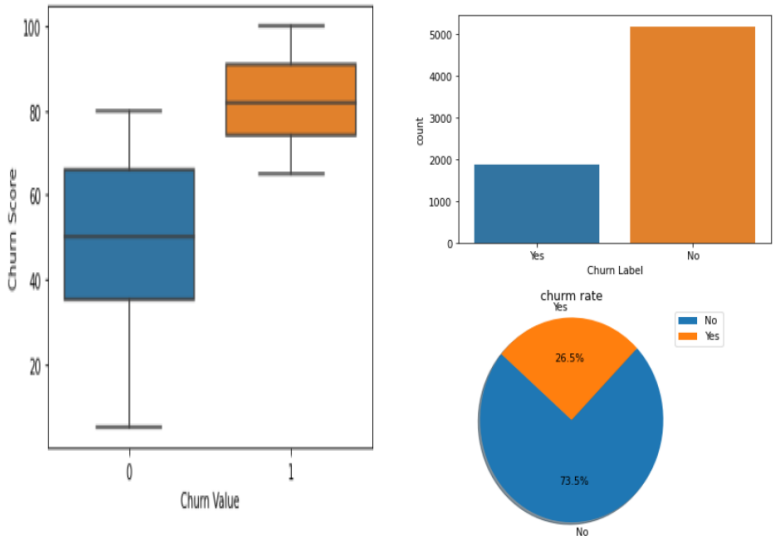

Figure 3: churn score feature

- **internet service:**

  we notice that the percentage of customers using "Fiber Optic" type (43.9) is almost the double of the percentage of customers using the DSL type (34.3) and the customers not using any type (21.6).

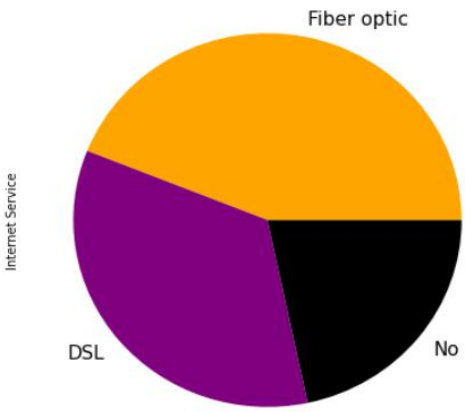

Figure 4: Internet Service feature

- **Gender distribution:**

  We can notice that our data is homogeneous because according to the histogram the proportions of churns are almost the same between males and females. Hence this variable does not have an impact on our target and we will prove it by correlation.

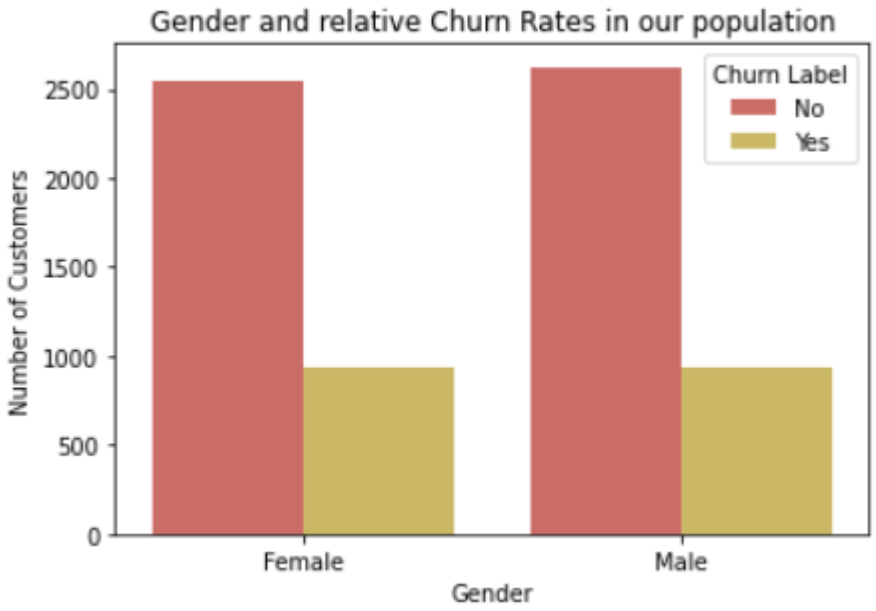

Figure 5: Gender feature

- **Multiple lines:**

  Customers with multiple lines, without multiple lines and without phones have the same Churn percentages which are 0.75, 0.749 and 0.713 successively.

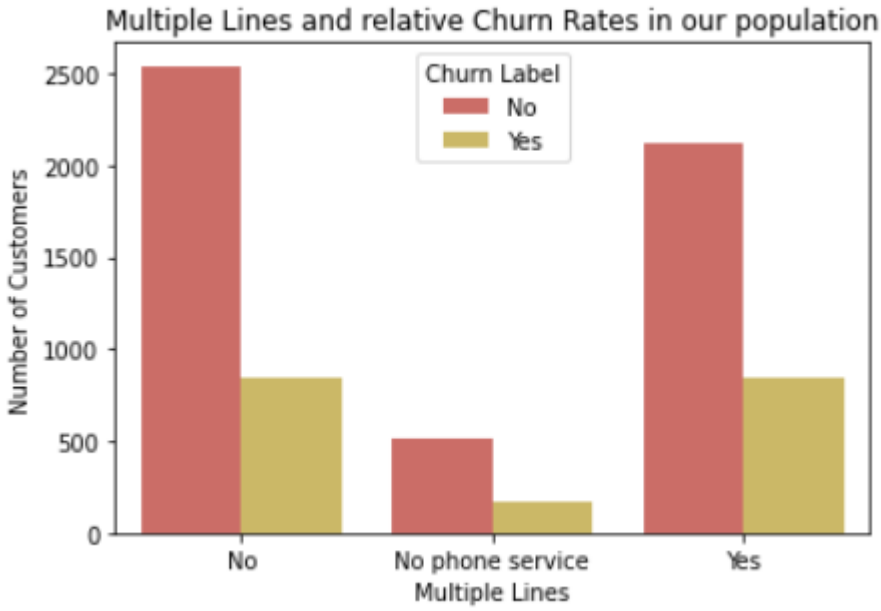

Figure 6: Multiple lines feature distribution

```
pourcentage of No: 0.7495575221238938 %
pourcentage of Yes : 0.750733137829912 %
pourcentage of No Phone : 0.713901043419724 %
```

Figure 7: Gender feature percentages

- **Age:**

  By looking at the histogram we notice that seniors are more likely to churn according to the churn rate so the age variable may have an impact on the target we will verify it later using correlation.

	Senior Citizen	Churn Label	Number of Customers
0	Young	No	4508
1	Young	Yes	1393
2	Senior	No	666
3	Senior	Yes	476

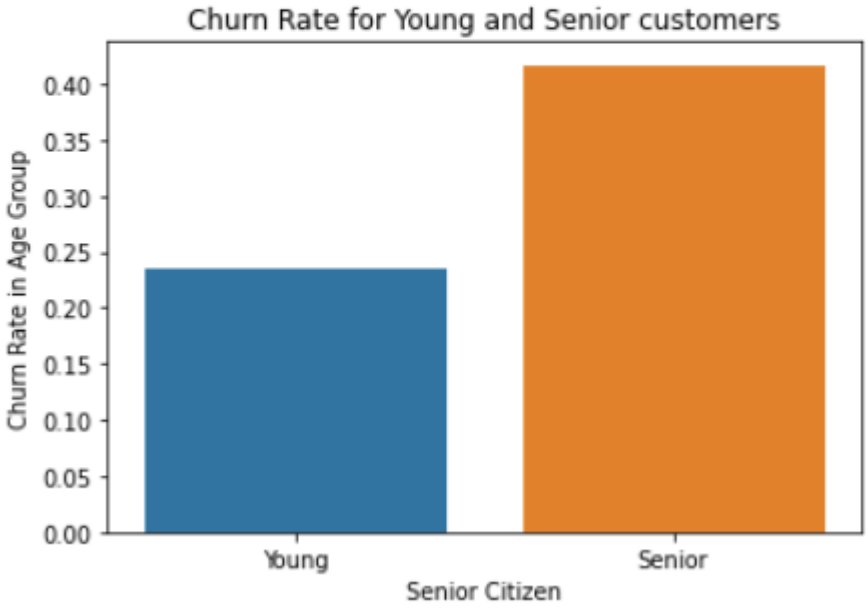

Figure 8: Age feature distribution

- **Phone Service:**

We can notice that the majority of customers are using the phone service.

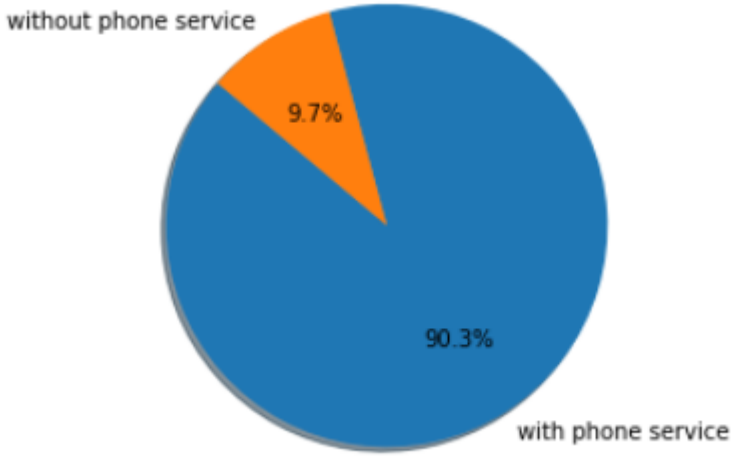

Figure 9: Phone service feature distribution

- **Contract:**

We can notice that the majority of customers who are going to churn have more likely a “Month-to-month contract” type and our data are heterogeneous.

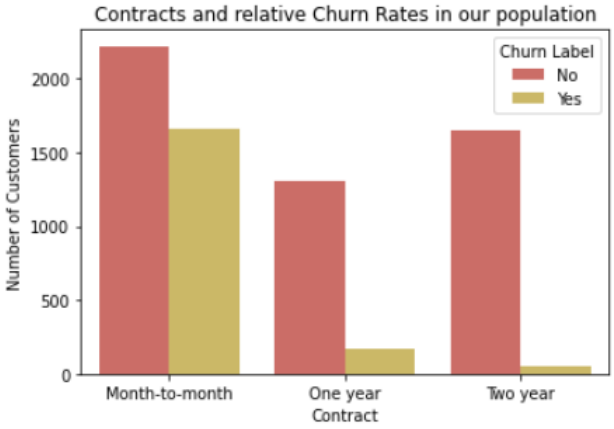

Figure 10: Contract feature distribution

- **Tenure months:**

In this sample we notice that the majority of the clients are new customers and a big percentage are clients for more than 70 months.
By looking at the data we can say that there was probably a competitive launch offer, which could explain the high number through efficient retention rates that would lead to fast market saturation that should explain such a sharp kickstart in the number of subscriptions and their sudden drop.

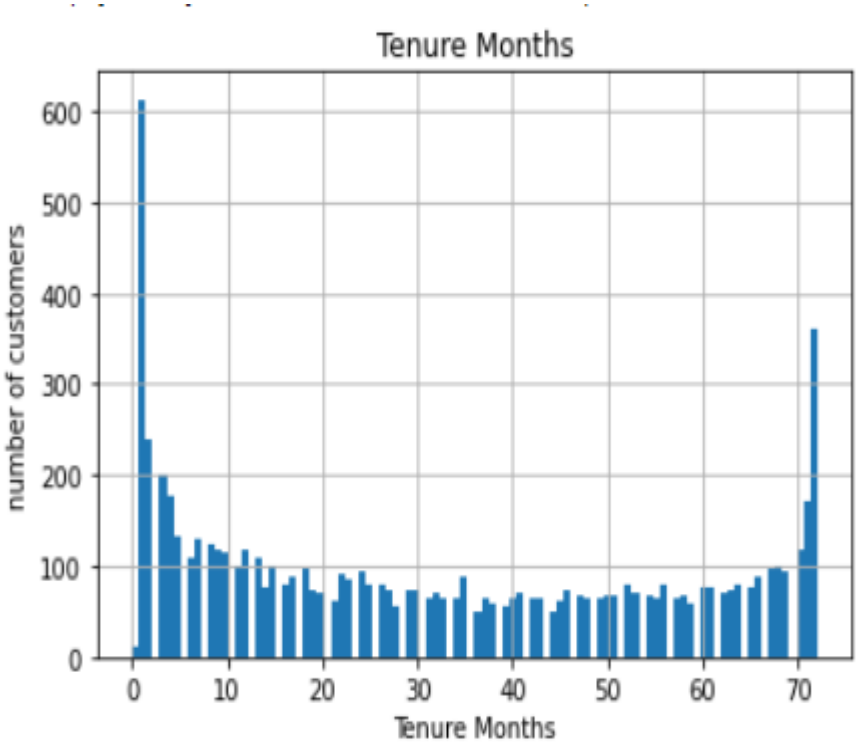

Figure 11: Tenure feature distribution

- **Summary:**

All of these histograms summarize what was previously mentioned.

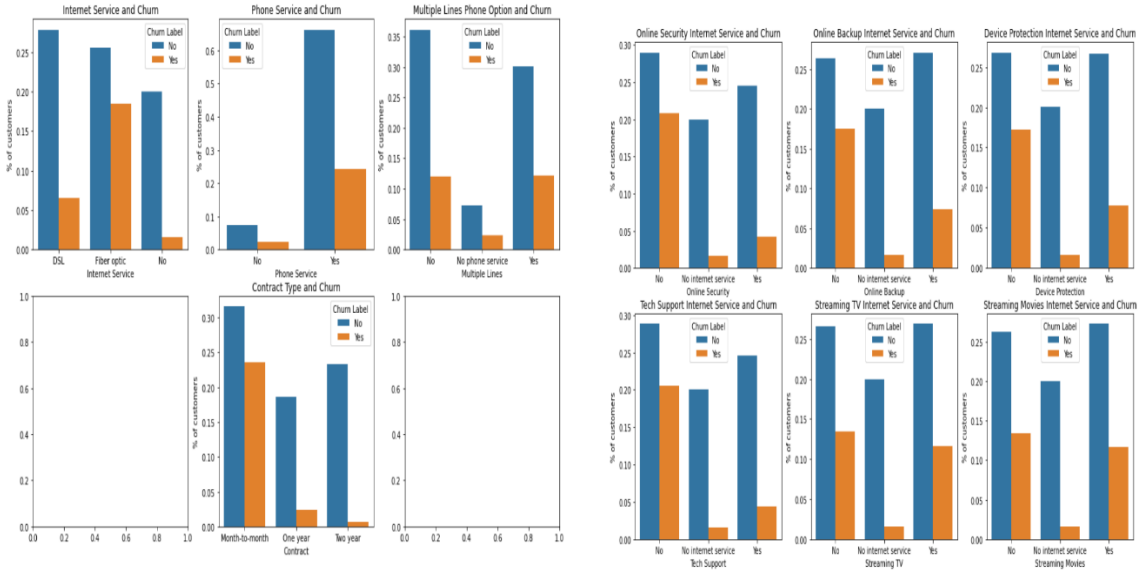

Figure 12: Summary of all previous features' distributions

CHAPTER 4: DATA PREPARATION

INTRODUCTION:

A well-prepared dataset allows for precise analysis, limits errors and inaccuracies.
It's gotten easier with the new methods and functions developed to cleanse and qualify data.

In this chapter we will detail the process of our data preparation during which we cleaned and transformed raw data. We reformatted the whole dataset and made some corrections. For example, we standardized some data formats and removed outliers.

DATA CLEANING:

We deleted the columns containing a unique value in each row.

```
CustomerID          7043
Count                  1
Country                1
State                  1
City                1129
Zip Code            1652
```

```
1 To_drop=['CustomerID']
2 data=data.drop(To_drop,axis=1)
```

Figure 13: Code for dropping customerID

```
1 To_drop=['Count','Country','State']
2 data=data.drop(To_drop,axis=1)
```

Figure 14: Code for dropping count, country, state

We deleted the column “churn reason” because it contains a lot of missing values (5174) and “churn label” because it’s basically the same column as “churn value” but with a ‘yes’ and ‘no’ values.

```
[86]  1 To_drop=['Churn Reason']
      2 data=data.drop(To_drop,axis=1)
```

```
[87]  1 To_drop=['Churn Label']
      2 data=data.drop(To_drop,axis=1)
```

Figure 15: Code for dropping churn reason and churn label

We erased 11 rows containing “space” value in Total Charges feature.

```
[63]  1 print(data["Total Charges"].value_counts().head(n=8))

           11
20.2       11
19.75       9
20.05       8
19.9        8
19.65       8
19.55       7
45.3        7
Name: Total Charges, dtype: int64
```

Figure 16: Code for erasing the space value in total charges

The churn score value and CLTV has no precise formula to calculate it so we tried to predict its values using the regression methods but we got very low prediction scores so we dropped them along with churn value that we copied to use as a target.

```
5]  1 X_samp=df_upsampled.copy()
    2 Y_samp=df_upsampled['Churn Value']
    3 To_drop=['Churn Value','CLTV','Churn Score']
    4 X_samp=X_samp.drop(To_drop,axis=1)
```

Figure 17: Code unsampling churn score

FEATURES SELECTION:

- **Encoding qualitative variables:**

For the qualitative variables with 2 and more modalities, we used “**get_dummies”** from Pandas to encode them.

```
1 df.columns
```

```
Index(['Zip Code', 'Latitude', 'Longitude', 'Gender', 'Senior Citizen',
       'Partner', 'Dependents', 'Tenure Months', 'Phone Service',
       'Multiple Lines', 'Internet Service', 'Online Security',
       'Online Backup', 'Device Protection', 'Tech Support', 'Streaming TV',
       'Streaming Movies', 'Contract', 'Paperless Billing', 'Payment Method',
       'Monthly Charges', 'Total Charges', 'Churn Value', 'Churn Score',
       'CLTV'],
      dtype='object')
```

```
[102] 1 df = pd.get_dummies(df.iloc[:, 1 :])
      2 df.head()
```

	Latitude	Longitude	Partner	Dependents	Tenure Months	Phone Service	Paperless Billing	Monthly Charges	Total Charges	Churn Value	Churn Score	CLTV	Gender_Female	Gender_Male	Senior Citizen_No	Se Citizen
0	33.964131	-118.272783	0	0	2	1	1	53.85	108.15	1	86	3239	0	1	1	
1	34.059281	-118.307420	0	1	2	1	1	70.70	151.65	1	67	2701	1	0	1	
2	34.048013	-118.293953	0	1	8	1	1	99.65	820.50	1	86	5372	1	0	1	
3	34.062125	-118.315709	1	1	28	1	1	104.80	3046.05	1	84	5003	1	0	1	
4	34.039224	-118.266293	0	1	49	1	1	103.70	5036.30	1	89	5340	0	1	1	

Figure 18: Code for encoding the qualitative values

- **normality test:**

We tested the normality of our distributions to adjust the correlation parameters (spearman or Pearson).

Tests for normal distributions

```
105] 1 liste_columns=df.columns
     2 k=0
     3 for i in liste_columns:
     4   test=stats.shapiro(df[i])
     5   if (test[1]>0.05):
     6     k=k+1
     7     print('the p_value of ',i,' is ',test[1])
     8 if k==0:
     9   print('all variables has not a normal distribution because all p_values are smaller than 0.05')
    10
```

```
all variables has not a normal distribution because all p_values are smaller than 0.05
```

Figure 19: Code normal distributions test

- **correlation:**

The spearman parameter was used to determine the correlation between features and target.

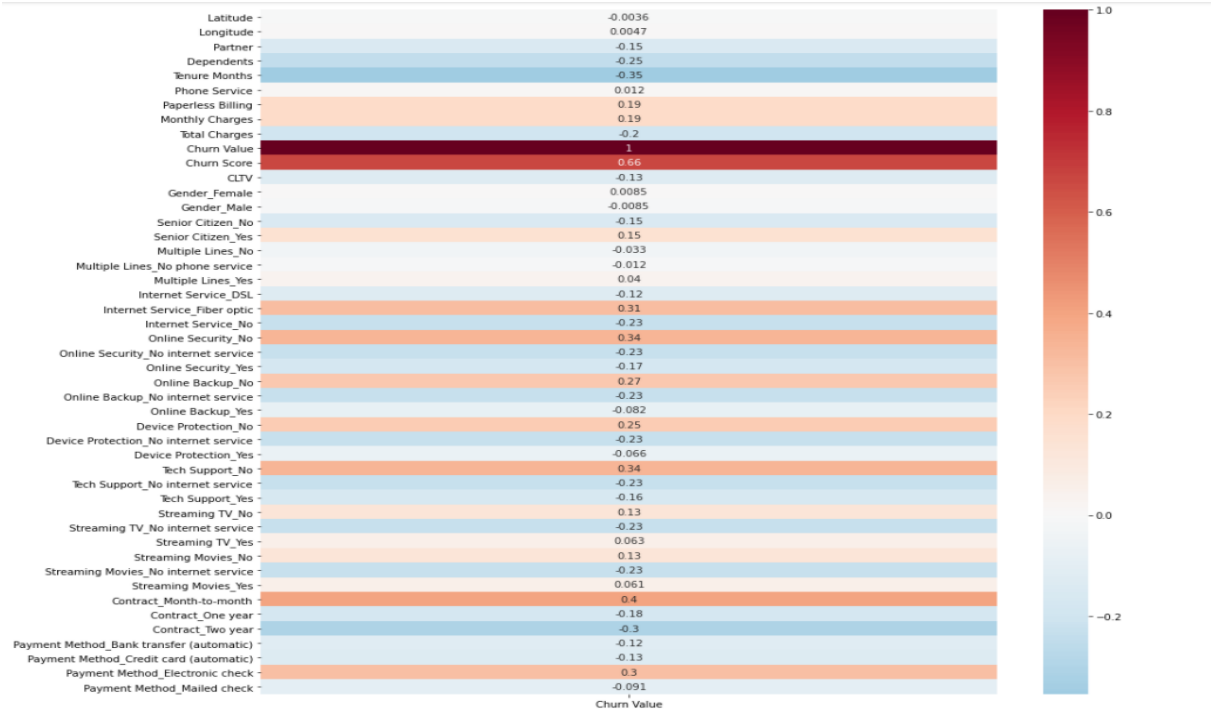

Figure 20: Correlation results

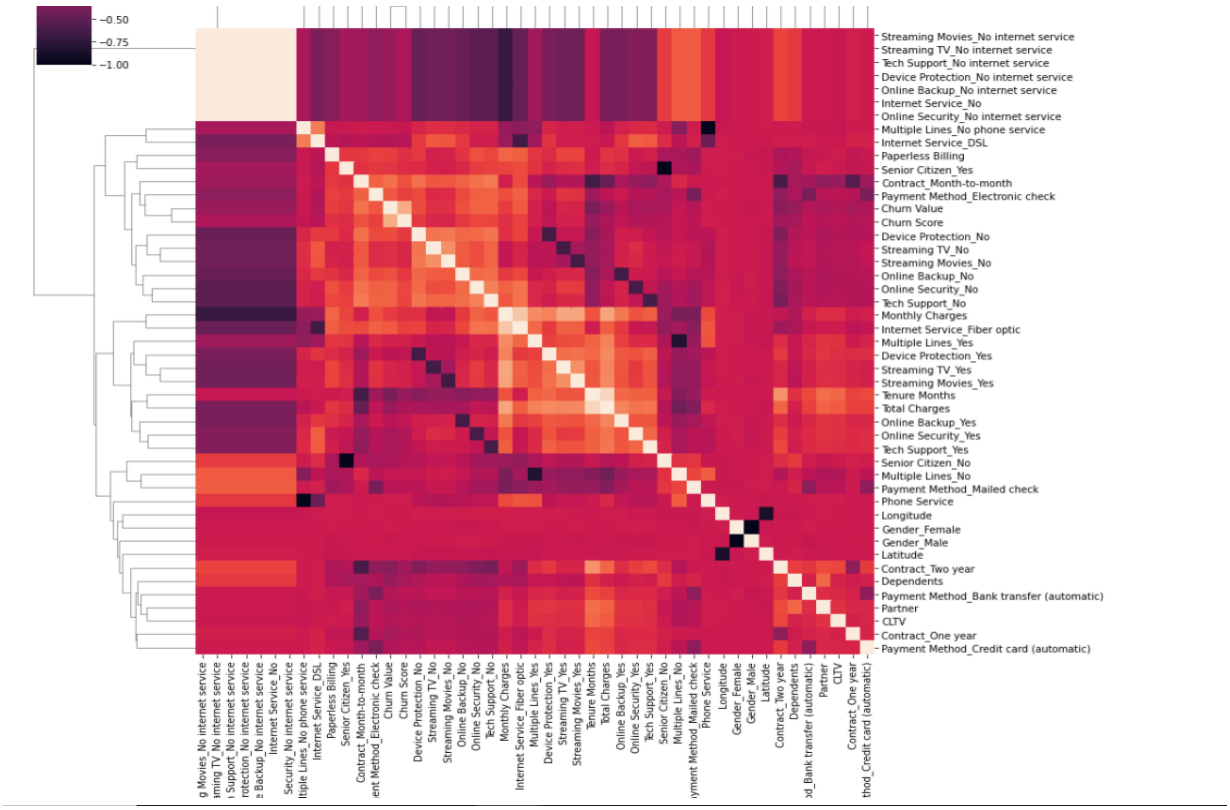

Figure 21: Correlation result_2

OVERSAMPLING:

After noticing the unbalance in the churn value column (a huge difference between the number of 0: non-churners and 1: churners) we used this method to add more rows for the churners (1) to get a balanced dataset

```
[ ]  1 var_gender='Churn Value'
     2 print(df[var_gender].unique())
     3 y=df[var_gender]
     4 no, yes = y.value_counts()
     5 print('Number of yes: ',yes)
     6 print('Number of no : ',no)
```

```
[1 0]
Number of yes:  1869
Number of no :  5163
```

```
[ ]  1 df_majority = df[df['Churn Value']==0]
     2 df_minority = df[df['Churn Value']==1]
     3 df_minority_upsampled = resample(df_minority,
     4                                  replace=True,
     5                                  n_samples=len(df_majority),
     6                                  random_state=123)
     7 df_upsampled = pd.concat([df_majority, df_minority_upsampled])
     8
```

Figure 22: Oversampling

CHAPTER 4: MODELING

INTRODUCTION:

After cleaning and preparing our final dataset we can start the modelling stage during which we developed a number of different models using a variety of techniques and metrics.

For each model we assessed its results, interpreted them and compared them to other models by their confusion matrix, scores ...etc to finally choose the best one of them.
In this chapter we will expose some results of the different models we used such as KNN, Random Forest, XGBoost ...

BEST METRICS SELECTION:

1. This loop allows us to go through all the features and to change the train size in a very precise interval to make the function called Best_RF (Best_RandomForest).

2. each time we add a column and we test it with different train sizes with a 0.01 step until we reach the totality of columns.

```
start = timeit.default_timer()
Liste_fin=[]
for j in np.arange(len(All_features)-1,-1,-1):
  print(j)
  Features_model_RF=X.copy()
  L=[]
  To_drop=All_features[:j]
  Features_model_RF=Features_model_RF.drop(To_drop,axis=1)
  for k in np.arange (0.75,0.9,0.01):
    var=Best_RF(Features_model_RF,k)
    L.append(var)
    print('-------')
  print(L)
  print('when we have '+str(j)+' features')
  print(idx_max_RF(L))
  Liste_fin.append(idx_max_RF(L))
print(Liste_fin)
stop = timeit.default_timer()
print('Time: ', stop - start)
```

Figure 23: Best metrics selection

Best_RF:

- splitting the data
- normalization
- Application of RandomSearchCV
- Training the final model with the best parameters.
- Getting the output [AUC, best_params].

```
def Best_RF(liste,n):
  X_train, X_test, Y_train, Y_test= train_test_split(liste,Y, train_size=n, random_state=6,stratify=Y)
  robust = RobustScaler()
  X_train=robust.fit_transform(X_train)
  X_test=robust.transform(X_test)
  RF=RandomForestClassifier(random_state=3)
  params_distributions={'criterion':['gini','entropy'],
                        'n_estimators': [100,200,300],
                        'n_jobs':[-1]}
  RF_random=RandomizedSearchCV(RF,param_distributions=params_distributions,n_iter=15,random_state=3,cv=5,n_jobs=-1,verbose=2,scoring='roc_auc')
  start = timeit.default_timer()
  RF_random.fit(X_train,Y_train)
  stop = timeit.default_timer()
  print('Time: ', stop - start)
  print("Tuned RandomForest Parameters (RandomizedSearchCV): {}".format(RF_random.best_params_))
  print("Best RandomForest Training Score (RandomizedSearchCV) :{}".format(RF_random.best_score_))
  dictio=RF_random.best_params_
  final_model = RandomForestClassifier(criterion=dictio['criterion'],n_estimators=dictio['n_estimators'],random_state=3,n_jobs=-1)
  final_model.fit(X_train, Y_train)
  print(final_model.score(X_test, Y_test))
  print(final_model.score(X_train, Y_train))
  RF_pred_prob =final_model.predict_proba(X_test)[:,1]
  RF_auroc = roc_auc_score(Y_test, RF_pred_prob)
  print("RandomForest AUROC: {}".format(RF_auroc))
  return[RF_auroc,RF_random.best_params_]
```

Figure 24: Best RF

⇨ **We will repeat the same process for all the remaining models (KNN, Decision Tree, XGBoost, SVM).**

RANDOM FOREST:

In this model we used a train size equal to 0.75 and 43 features.

Hyperparameters: {'n_jobs': -1, 'n_estimators': 300, 'criterion': 'gini'}

Confusion matrix:

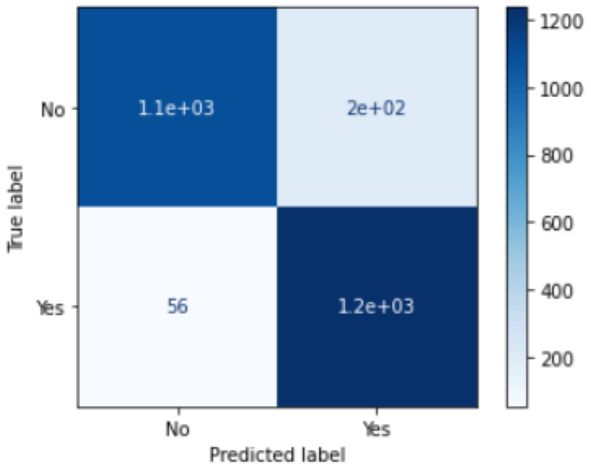

Figure 25: Random Forest Confusion Matrix

Classification report:

RandomForest AUROC: 0.9718434421463975

	precision	recall	f1-score	support
0	0.951347	0.848180	0.896806	1291
1	0.863033	0.956623	0.907421	1291
accuracy			0.902401	2582
macro avg	0.907190	0.902401	0.902113	2582
weighted avg	0.907190	0.902401	0.902113	2582

Figure 26: Random Forest Classification report

DECISION TREE:

In this model we used a train size equal to 0.86 and 27 features.

Hyperparameters: {'splitter': 'random', 'max_depth': 27, 'criterion': 'gini'}

Confusion matrix:

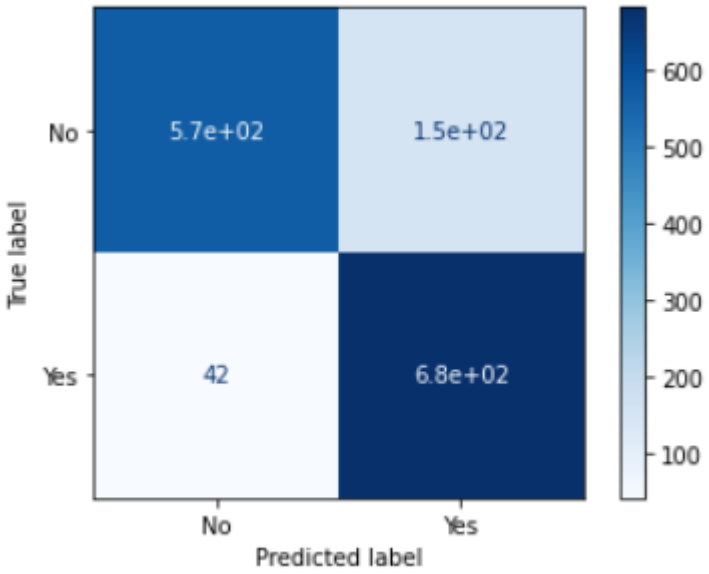

Figure 27: Decision Tree confusion matrix

Classification report:

DecisionTree AUROC: 0.8686001733211665

	precision	recall	f1-score	support
0	0.931485	0.789765	0.854790	723
1	0.817527	0.941909	0.875321	723
accuracy			0.865837	1446
macro avg	0.874506	0.865837	0.865056	1446
weighted avg	0.874506	0.865837	0.865056	1446

Figure 28: Decision Tree classification report

K-NEAREST NEIGHBORS (KNN):

In this model we used a train size equal to 0.86 and 27 features.

Hyperparameters: {'weights': 'distance', 'n_neighbors': 29, 'metric': 'manhattan', 'algorithm': 'brute'}

Confusion matrix:

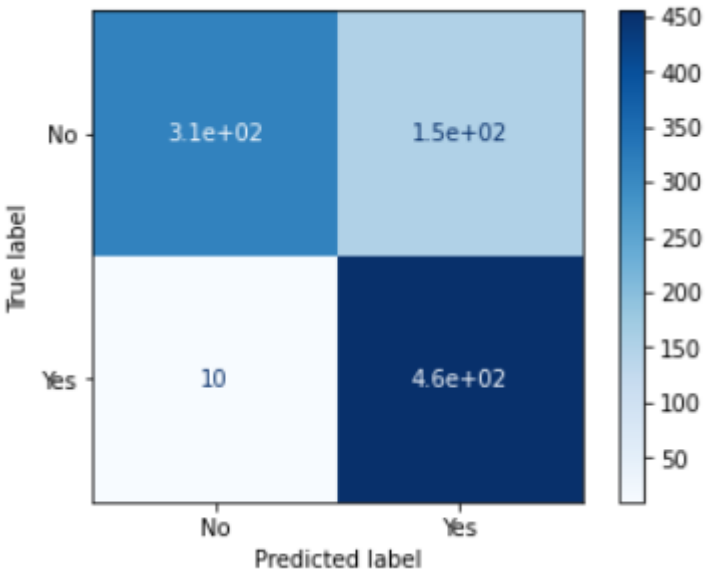

Figure 29: KNN confusion matrix

Classification report:

```
KNN AUROC: 0.9806359116660888
              precision    recall  f1-score   support

           0   0.969136  0.675269  0.795944       465
           1   0.750825  0.978495  0.849673       465

    accuracy                       0.826882       930
   macro avg   0.859980  0.826882  0.822809       930
weighted avg   0.859980  0.826882  0.822809       930
```

Figure 30: KNN classification report

XGBOOST:

In this model we used a train size equal to 0.81 and 44 features.

Hyperparameters: {'silent': False, 'n_jobs': -1, 'n_estimators': 200, 'gamma': 0.25}

Confusion matrix:

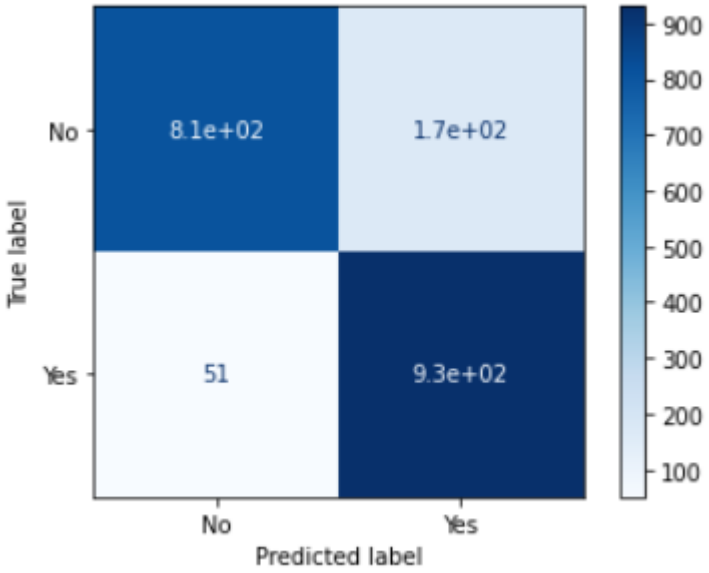

Figure 31: XGBoost confusion matrix

Classification report:

```
xgboost AUROC: 0.9519400723844795
              precision    recall  f1-score   support

           0   0.940904  0.827727  0.880694       981
           1   0.846224  0.948012  0.894231       981

    accuracy                       0.887870      1962
   macro avg   0.893564  0.887870  0.887462      1962
weighted avg   0.893564  0.887870  0.887462      1962
```

Figure 32: XGBoost classification report

CHAPTER 5: EVALUATION

INTRODUCTION:

This phase has many fruitful aims to build the optimal model. According to all the models results we decide which model to use, identify areas for further optimisation and finally deploy it.

For our models' evaluation we made this comparison table.

Model name	accuracy	f1_score (of class 1)	AUC	False Negative (Confusion Matrix)	Train_size
Random Forest	0.902	0.907	0.971	56	0.75
XGBoost	0.887	0.894	0.951	51	0.81
Decision Tree	0.865	0.875	0.868	42	0.86
KNN	0.826	0.849	0.980	10	0.91

❖ FINAL MODEL:

At the end of this phase we decided to go for the Random Forest because it has the highest accuracy and f1_score (of class 1: churners) so it detected more churners compared to other models. It also has the highest AUC value.

Its confusion matrix has 56 False Negatives from 10 326 rows in the whole dataset which is a very satisfying result.

From those results we can assume that using the **Random Forest** model can be the wisest and safest decision.

Limitations:

The main limitation of random forest is that a large number of trees can make the algorithm too slow and ineffective for real-time predictions.
In general, these algorithms are fast to train, but quite slow to create predictions once they are trained. A more accurate prediction requires more trees, which results in a slower model.

CHAPTER 6: DEPLOYMENT

INTRODUCTION:

In the previous chapter we explained our models building process and interpretations.

Finally, we choose the model with the best prediction scores (Random Forest).

As a final step we have the deployment phase where we used our model with a clean captivating and easy-to-understand User Interface.

In fact, after the models have been used for some time, they should be refreshed. This work should also be carried out using CRISP-DM but it may not require the first three stages (Business Understanding, Data Understanding and Data Preparation) to the same extents as for the first model.

OUR USER INTERFACE:

Home page

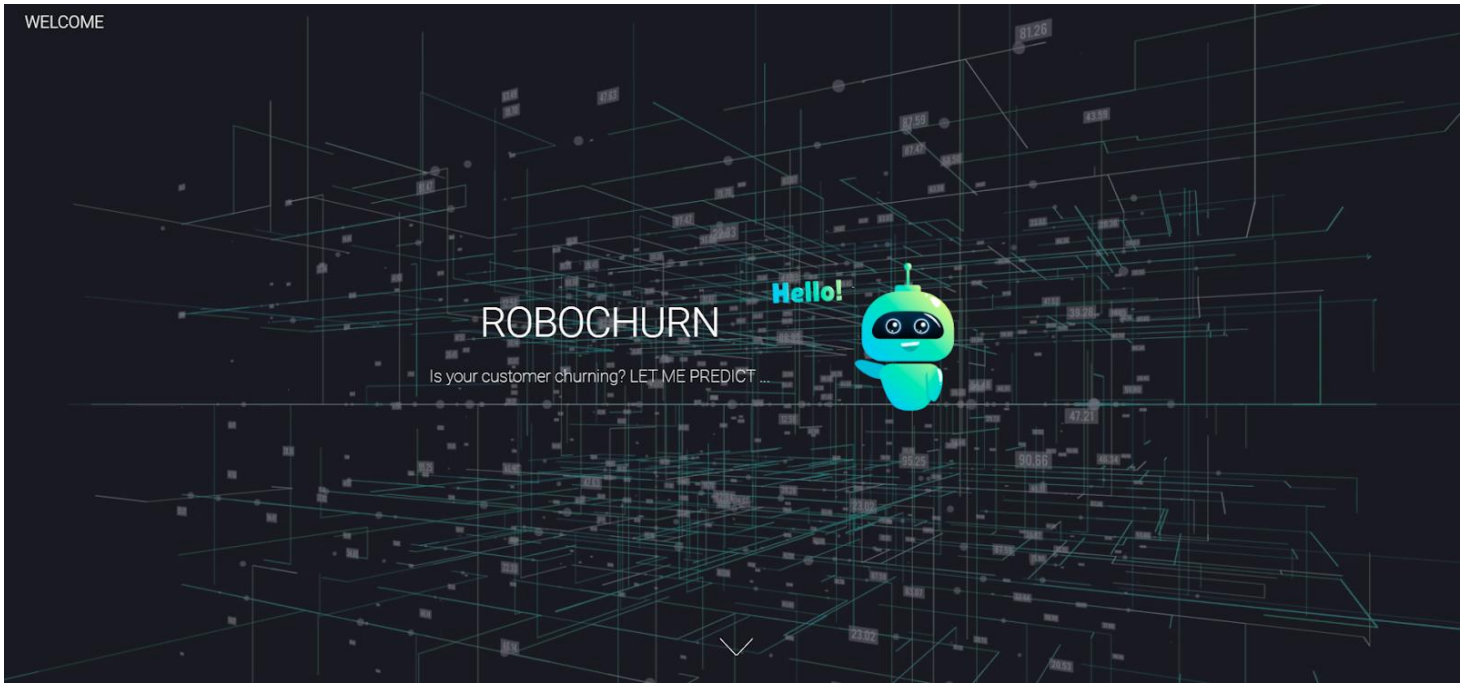

Figure 33: Application homepage

Then we find the section where we should fill up all the required data for prediction.

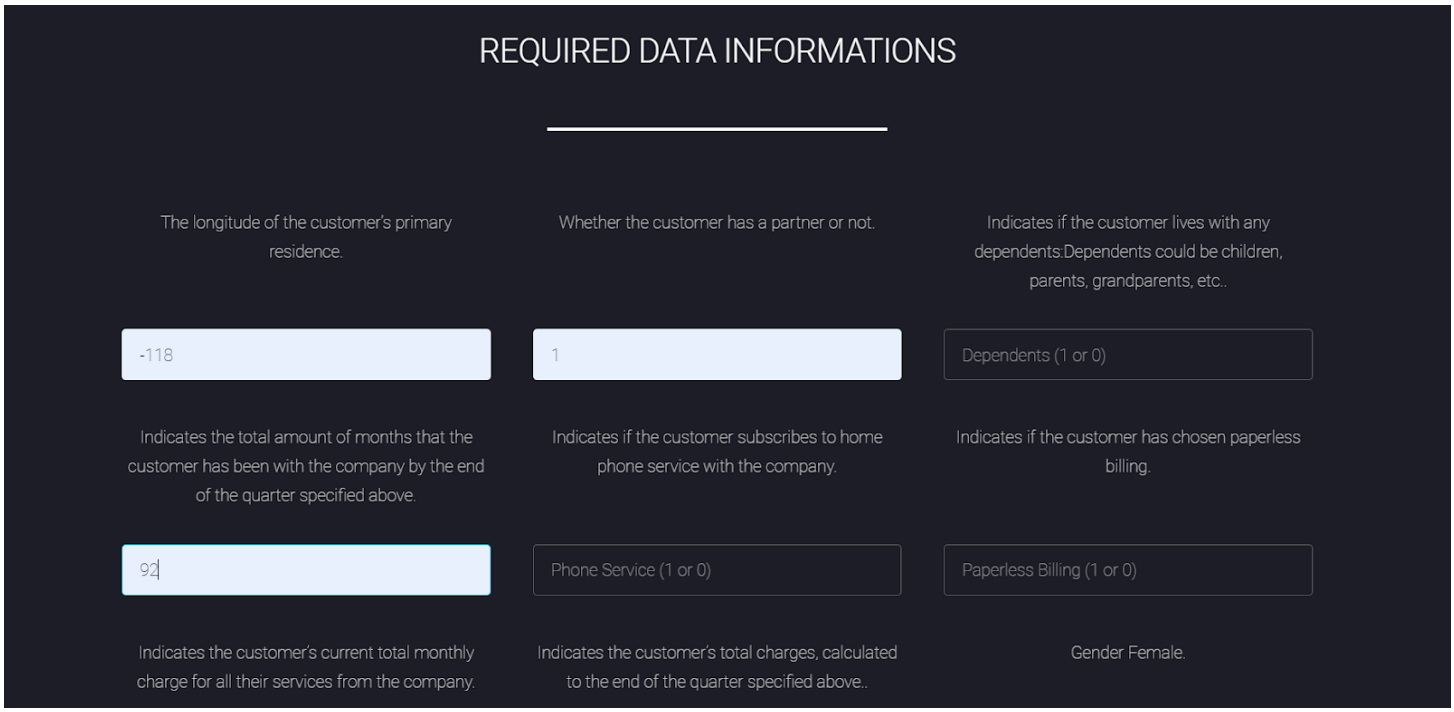

Figure 34: Application Inputs interface

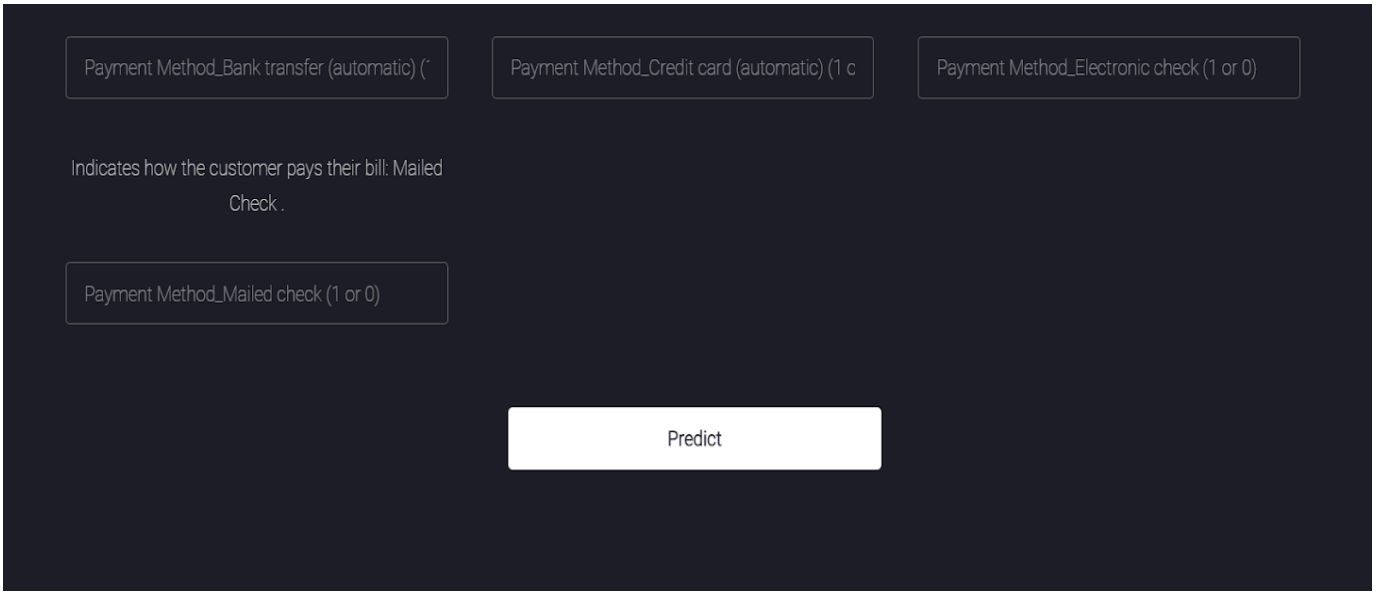

Figure 35: Application Inputs interface and prediction button

Here is the output message for the non-churning customers.

Figure 36: Application message for non-churning customers

Finally, here is the output message for the churning customers.

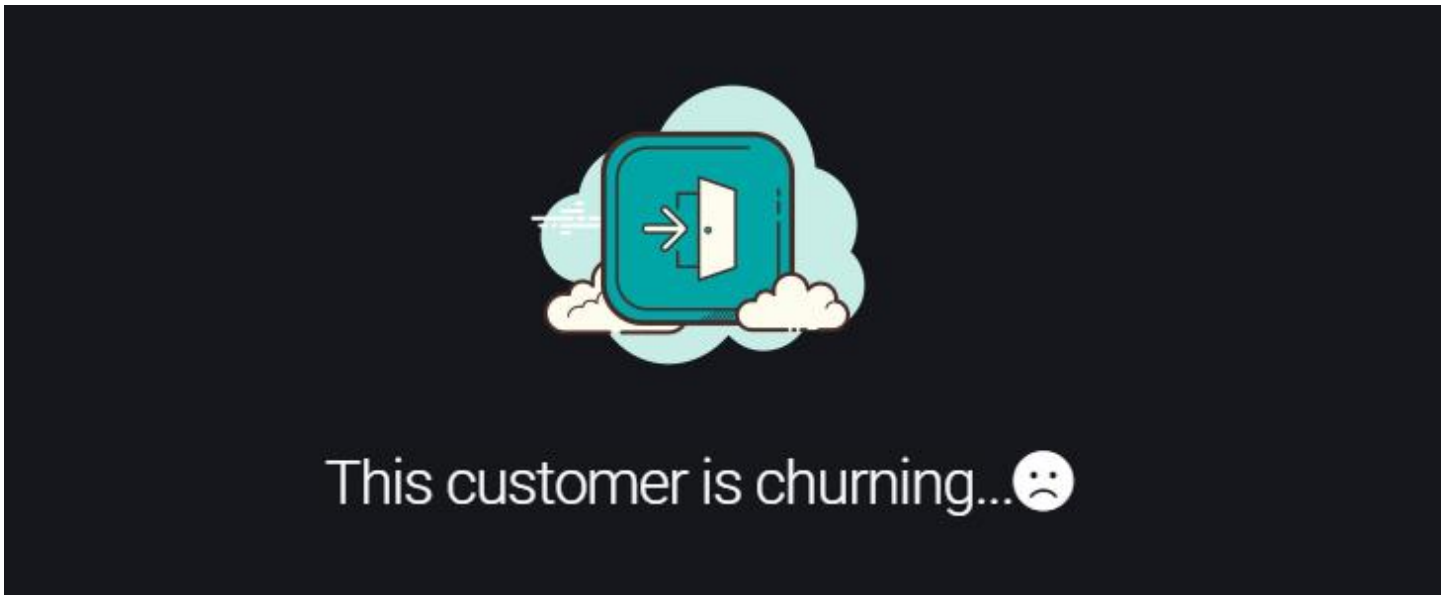

Figure 37: Application message for churning customers

FINAL CONCLUSION

In conclusion, we have seen that predicting customer churn is a very challenging task due to its temporal characteristic, which increases the overall data analysis complexity.

In fact, a standard procedure to build a good classification model does not exists yet, it's a task that is still widely discussed and studied. However, we have proved that machine learning tools can help companies to understand their customers' behavior and deal with churning problems.