

Project Machine Learning

Telco Customer Churn

Framed by: Mr Mohamed Hedi RIAHI

Realised by:

Asma Nouri

2020 – 2021

Contents

General Introduction	1
CHAPTER 0: Project Overview	2
0.1. Introduction.....	2
0.1 Presentation of the project	2
0.2.1 Project framework.....	2
0.2.2 Context and Issue	2
0.2.3 Objectif :	2
0.2 Methodology adopted :.....	2
0.3.1 CRISP-DM.....	3
0.3.2 CRISP-DM phases.....	4
0.3.3 Choice of methodology	4
0.4 Conclusion	4
CHAPTER 1 :Business Understanding	5
1.1. Introduction.....	5
1.2. Business objectives.....	5
1.2.1 Determination of business objectives.....	5
1.2.2 Process Solution.....	5
1.3. Project planning.....	5
1.5. Steps	7
1.6. Conclusion	7
2.1. Introduction.....	8
2.2 Data source.....	8
2.3 Features description.....	8
2.4 Visualizing data for categorical features	10
2.5 Visualizing data for numerical features.....	14
2.6 Conclusion	16
CHAPTER 3 :Data Preparation	17
3.1. Introduction.....	17
3.2 Data selection.....	17
3.3 Data cleaning.....	18
4.5.1. Sparseness Elimination.....	19
4.5.2. Data conversion.....	19
4.5.3. Outliers Detection	19
3.3.3.1 Definition	19

3.3.3.2	Types of outliers	20
3.5	Splitting the dataset.....	21
3.6	Encoding Categorical Data	21
3.7	Data Standardisation	22
CHAPTER 4	:Modeling.....	25
4.1	Introduction.....	25
4.2	Used methods.....	Erreur ! Signet non défini.
4.3	Feature Selection	26
4.4	Gradient Boost Classifier.....	28
4.4.1.	Definition	28
4.4.2.	Searching for the best parameters.....	29
4.4.3.	feature selection.....	30
4.5	XGBoost	31
4.5.1.	Definition :	31
4.5.2.	Searching for the best parameters.....	32
4.5.3.	Feature selection	32
4.6	Ada Boost	33
4.6.1.	Definition	33
4.6.2.	Searching for the best parameters.....	35
4.6.3.	Feature Selection.....	35
4.7	Logistic regression	36
4.7.1.	Definition	36
4.7.2.	Searching for the best parameters.....	37
4.7.3.	Feature Selection.....	37
4.8	Tree Decision	37
4.8.1.	Definition	37
4.8.2.	Searching for the best parameters.....	38
4.8.3.	Feature selection using Select From Model	38
4.9	K-nearest neighbors (KNN).....	40
4.9.1.	Definition	40
4.9.2.	Searching for the best parameters.....	40
4.9.3.	Feature Selection.....	41
4.9.3.1	Correlation Matrix	41
Figure 30:	Feature selection	43
4.10	Random Forest	43
4.10.1	Definition	43

Figure 31: Random Forest	44
4.10.2. Searching for the best parameters.....	45
4.10.3. Feature Selection using RFECV	45
4.11 Naive Baies	46
4.11.1. Definition	46
4.11.2. Searching for the best parameters.....	46
4.11.3 Feature Selection with sbfs	47
4.12 Standard SVM	48
4.12.1. Definition	48
4.12.2. Searching for the best parameters.....	49
4.12.3. Features Selection using Variance Threshold.....	49
Chapter 5 Evaluation	50
5.1 Definitions	50
5.2 Gradient Boost Classifier	52
5.2.1. Train and test scoring:	52
5.2.2. Classification report.....	53
5.2.3. Confusion matrix visualization	54
5.3 XGBoost	54
5.3.1. Train and test scoring:	54
5.3.2. Classification report.....	55
5.3.3. Confusion matrix visualization	55
5.4 Ada Boost	56
5.4.1. Train and test scoring:	56
5.4.2. Classification report.....	56
5.4.3. Confusion matrix visualization	57
5.5 Logitic Regression	58
5.5.1. Train and test scoring:	58
5.5.2. Classification report.....	58
5.5.3. Confusion matrix visualization	59
5.6 K-nearest neighbors (KNN)	59
5.6.1. Train and test scoring:	59
5.6.2. Classification report.....	60
5.6.3. Confusion matrix visualization	61
5.7 Tree decision	61
5.7.1. Train and test scoring:	61
5.7.2. Classification report.....	61

5.7.3. Confusion matrix visualization	62
5.8 Random forest	63
5.8.1. Train and test scoring:	63
5.8.2. Classification report.....	63
Figure 25: Classification report table	Erreur ! Signet non défini.
5.8.3. Confusion matrix visualization	64
Figure 39: Confusion matrix	64
5.9 Naïve baies	64
5.9.1. Train and test scoring:	64
5.9.2. Classification report.....	65
Figure 25: Classification report table	Erreur ! Signet non défini.
5.9.3. Confusion matrix visualization	66
5.10 Comparing the performance of the models.....	68
5.10.1. Comparative table	68
5.10.2 Roc Curves For each model	70
5.10.2.1. Interpretation:	71
5.10.2.2. Precision vs Recall.....	71
Interpretation :	Erreur ! Signet non défini.
5.11 Conclusion :	72
6.1 Introduction.....	73
6.2 Concept and deployment strategy	73
6.3 Deployment environment(tools) :	73
6.4 The web application	73
6.4.1 Web interface.....	74
6.4.1.1 Interface of welcoming.....	74
6.4.1.2 Interface of the customer churn prediction.....	74
6.5 Conclusion	76
Chapitre 7 : Perspectives and contributions	77
7.1 Perspectives.....	Erreur ! Signet non défini.
7.2 Academic and professional contributions.....	78
General conclusion	79

List of Figures

Figure 1: CRISP-DM method.....	3
Figure 2:Used tools.....	6
Figure 3:Django logo.....	7
Figure 4:the dataset	8
Figure 5: percentage of churn in Dataset.....	10
Figure 6: vizualizing country feature	11
Figure 7: vizualizing state feature	11
Figure 8: vizualizing City feature	12
Figure 9: vizualizing Gender Feature.....	12
Figure 10:vizualizing internet Service feature.....	13
Figure 11: vizualizing partner feature	14
Figure 12:vizualizing dependent feature.....	14
Figure 13: vizualizing count feature	14
Figure 14:vizualizing mounthly charges feature	15
Figure 15:vsualizing tenure charges feature	15
Figure 16: the features distrubtion	20
Figure 17: Uniques values per feature	Erreur ! Signet non défini.
Figure 18: Cross validation process	26
Figure 19: Gardient Boost Classifier	29
Figure 20: Features Importance Vizualization.....	30
Figure 21:RFECV method.....	31
Figure 22:XGBoost Process.....	32
Figure 23: Features importance visualization	33
Figure 24: Ada Boost process	35
Figure 25: Logistic regression process.....	36
Figure 26: Features importance visualization	39
Figure 27: K-Nearest Neighbours	40
Figure 28: Correlation Matrix	41
Figure 29: Dataset representation	42
Figure 30: Feature selection	43
Figure 31: Random Forest	44
Figure 32: The Dataframe of the Sequential Selector	47
Figure 33: Confusion matrix	54
Figure 34: Confusion matrix	56
Figure 35: Confusion matrix	58
Figure 36: Confusion matrix	59
Figure 37: Confusion matrix	61
Figure 38: Confusion matrix	62
Figure 39: Confusion matrix	64
Figure 40: Confusion matrix	66
Figure 41: comparative table	69
Figure 42: Train scores	70
Figure 43: Test scores.....	70

Figure 44: Roc curves for each model	71
Figure 45: Roc curves for each model	72
Figure 46: Welcoming interface	74
Figure 47: Interface of the customer chum prediction	75

General Introduction

In recent decades, the telecommunications sector has become one of the main industries in the world and the technical progress and the increasing number of operators raised the level of competition between companies.

Everyone is having a communication medium through mobile phones, hence the market saturation. So, companies are working hard to survive in this fluid competitive market depending on multiple strategies and they are aware of the importance of retaining existing customers because according to research, the cost of recruiting new customers is more expensive than retaining existing ones.

This study describes the research and technical work in the field of machine learning to predict Telco's customer churn.

One of the challenges of Telco company is to maintain the loyalty of the customer. To do so, Telco must improve customer service, improve product quality, and must be able to know in advance which customers have the possibility of leaving the company.

A good accurate customer churn prediction model can effectively help in taking decisions and actions early to prevent churn and increasing profits.

In order to achieve the objective of our project, a machine learning model, and for that the crisp-dm methodology comes, as the best methodology appropriate to a data mining project, to guide us throughout this journey.

On chapter 0 we are going to enunciate the methodology adopted, after that we will explicate our business in the first chapter, determinate our goals and make the project clearer. On the chapter 2, it comes the turn of the data understanding, on this chapter we will try to explore our data and verify it quality, And after that we will prepare our data in the Data preparation chapter, we have to select the data that we will use after that in Modeling. As we mentioned Modeling is our next chapter when we have to build our models, and after building our models we have to evaluate them in the chapter 6 entitled evaluation, finally the last chapter deployment it covers our activities through this whole project in order to organize knowledge gained.

CHAPTER 0: Project Overview

0.1. Introduction

This chapter is devoted to putting the project in its framework. And this, by presenting the problematic which proposed it, the objectives and the methodology of the work adopted.

0.1 Presentation of the project

In this part of the report, we will present the framework of the project.

Subsequently, we will invoke the context and the problematic of the subject.

0.2.1 Project framework

This project is carried out as part of the preparation of the Machine Learning project at ESPRIT: Private High School of Engineering and Technologies. This project was carried out for a period of 2 months: from 02/11/2020 until 02/01/2021.

0.2.2 Context and Issue

Churn prediction consists of detecting which customers are likely to cancel a subscription to a service based on how they use the service. We want to predict the answer to the following question, asked for each current customer: *“Is this customer going to leave us within the next X months?”* There are only two possible answers, yes or no, and it is what we call a *binary classification task*. Here, the *input* of the task is a customer and the *output* is the answer to the question (yes or no).

0.2.3 Objectif:

The purpose of this research is to develop and design an effective and efficient model for customer churn prediction in telecommunication industry

0.2 Methodology adopted :

As is the case in many projects, it is the lack of studies that leads to the non-fulfillment of the necessary requests and therefore to a non-satisfaction of the customers.

Thus, we count on the choice of the work methodology and the good planning to guarantee the level of satisfaction required throughout the process of realization. Considering that this

is a complex process, and in order to ensure the realization of a data science project under the best conditions, it is imperative to make a good management and a good study of the resources at the same time human and material. Therefore, such a need puts us in front of a demand to ensure a good use of the methodologies which correspond to the needs of our project.

0.3.1 CRISP-DM

The CRISP-DM method (Cross-Industry Standard Process for Data Mining) was originally developed by IBM in the 1960s to carry out Data Mining projects. It remains today the only method that can be used effectively for all Data Science projects. CRISP-DM succeeded because it is solidly based on practical experience, the real world of how specialists manage Data Mining projects.



Figure 1: CRISP-DM method

0.3.2 CRISP-DM phases

1. Business Understanding: The first step defines the business goals and the reasons for wanting to achieve those goals.
2. Data Understanding: Associate the data with their signification from a business perspective, to determine precisely the data to be analyzed and to determine its quality.
3. Data Preparation: This phase includes the classification of the data according to selected criteria, the cleaning of the data, and especially to make them compatible with the algorithms that will be used.
4. Modeling: The modeling includes the choice, parameterization and testing of various algorithms and their sequence, which constitutes a model.
5. Evaluation: The evaluation aims to verify the model or knowledge obtained to ensure that they meet the objectives formulated at the beginning of the process. At this stage, the robustness and precision of the models obtained are tested in particular.
6. Deployment: The deployment can thus go, from the simple generation of a report describing the knowledge obtained until the implementation of an application, allowing the use of the model obtained, for the prediction of unknown values of an element of interest.

0.3.3 Choice of methodology

We have chosen to work with the CRISP-DM methodology.

This methodology will provide us with the most adequate solution for the needs of customers with a better workflow and it represents today the most effective methodology for all data science projects since it has the specificity of aligning itself with an approach. Iterative and cyclical.

0.4 Conclusion

This introductory chapter allowed us to detail the overall framework of the system by presenting the problem, the solution that we are going to design and develop, as well as the methodology to follow.

In the next chapter, we will begin the theoretical study necessary to achieve this solution and we start with the first phase of our methodology, which is business understanding.

CHAPTER 1: Business Understanding

1.1. Introduction

The first step in the CRISP-DM process is business understanding, which is understanding what the customer wants to accomplish from a business perspective.

Throughout this chapter, we will focus more on the pre-established goals through which we will establish an action plan that we will follow to ensure the success of the business.

1.2. Business objectives

During this phase, it is essential that a data scientist understands the commercial and technical issues even before approaching the data or the work tools, ... We must therefore define what we want to accomplish and define the reasons that we push to achieve this goal.

1.2.1 Determination of business objectives

A business goal describes goals in business terminology while a data mining goal describes project goals in technical terms. Therefore, these are our main objectives through the process.

- Build a good accurate model able to predict customers churn early to allow the marketing and retention department to try to retain their business by luring the churners with attractive offers
- Implementing different customer churn prediction models using Telco dataset

1.2.2 Process Solution

As a solution the process of modeling their worries is referred to as “churn Detection”, the churn detection models are made to avoid large losses. As a fact, it can be said that churn detection model enables to assess boost the knowledge quickly. Furthermore, the churn detection gives a chance to upgrade customer services.

1.3. Project planning

Planning which consists in properly determining the list of tasks to be carried out in a project is a crucial step that will allow us to monitor the progress of the work throughout its progress and specially to fix the time constraint and control the allocation of resources to carry out the various tasks. Having a plan in place at the start of the project helps meet the data mining goals and business goals set at the start of the project. Ensuring good project management will allow us to ensure that the objectives of the project are well aligned with the strategic objectives of the company.

1.4. Used tools

To predict if a customer will churn or not, we are working with Python and it's amazing open-source libraries. First of all, we use Jupyter Notebook, that is an open-source application for live coding and it allows us to tell a story with the code. Furthermore, we import Pandas, which puts our data in an easy-to-use structure for data analysis and data transformation. To make data exploration more graspable, we use Plotly to visualize some of our insights. Finally, with scikit-learn we will split our dataset and train our predictive model.

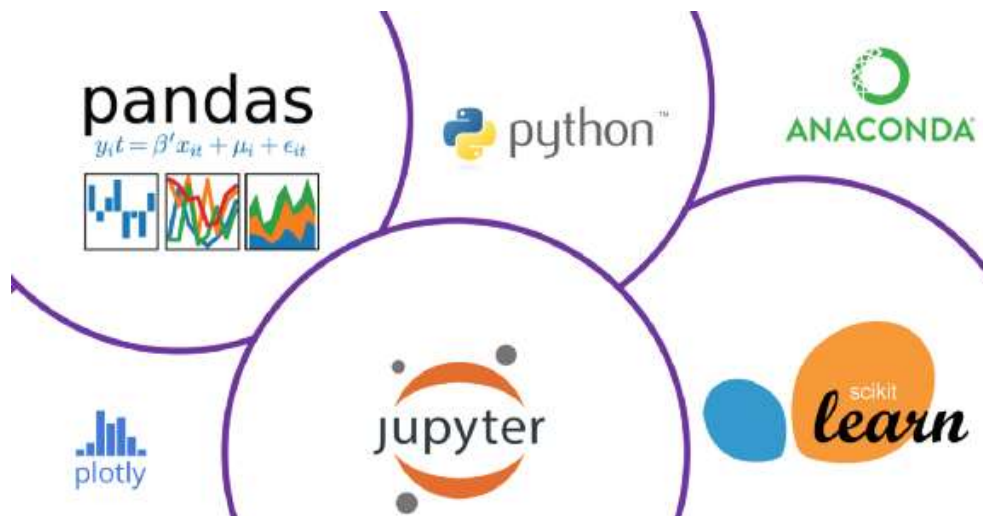


Figure 2:Used tools

Then in the deployment phase we used the framework Django:



Figure 3:Django logo

1.5. Steps

Here's an outline of our step-by-step: First of all, we collect the data as our first step. Then, we have to discover our data, explore it, understand it and verify its quality. Next, the data preparation where our data gets prepared for modeling. Many techniques take place to prepare it such as inclusion, cleaning, exclusion criteria, merging, formatting, etc. Next, Modeling phase, here we have to use several algorithms, build them and assess them. And finally, we have to evaluate our models already built, and fit our achievement into our client demands.

1.6. Conclusion

Throughout this chapter, we have focused on the project objectives from a business point of view then we have clearly defined the steps that we followed for the Business understanding part. The following chapter will cover the second phase of CRISP which is understanding the data.

CHAPTER 2: Data Understanding

2.1. Introduction

The second step of the CRISP-DM process requires the acquisition of data (or access to the data) listed in the project resources. This initial collection includes loading the data necessary to understand the data. A good understanding of the data will allow us to have a good preprocessing phase. In this chapter we will focus on the process of collecting the data needed to better explore it and understand some primary hunches.

2.2 Data source

The dataset consists 7043 rows with 33 attributes. All entries have several features and of course, a column stating if the customer has churned or not. The dataset contains 33 features: 24 features are of type object and 9 features are numeric. The dataset contains features with categorical and numeric types. The dataset consists 7043 rows with 28 attributes. All entries have several features and of course a column stating if the customer has churned or not. The dataset contains features with categorical and numeric types. To better understand the data we will first load it into pandas dataframe and explore it with the help of some very basic commands.

	CustomerID	Count	Country	State	City	Zip Code	Lat Long	Latitude	Longitude	Gender	...	Contract	Paperless Billing	Payment Method	Monthly Charges	Ti Charj
0	3662-QPYBK	1	United States	California	Los Angeles	90003	33.964131, -118.272783	33.964131	-118.272783	Male	—	Month-to-month	Yes	Mailed check	53.85	108
1	9237-HQITU	1	United States	California	Los Angeles	90005	34.059281, -118.30742	34.059281	-118.307420	Female	—	Month-to-month	Yes	Electronic check	70.70	151
2	9305-CDSKC	1	United States	California	Los Angeles	90006	34.048013, -118.293953	34.048013	-118.293953	Female	—	Month-to-month	Yes	Electronic check	99.85	82
3	7892-PQOKP	1	United States	California	Los Angeles	90010	34.062125, -118.315709	34.062125	-118.315709	Female	—	Month-to-month	Yes	Electronic check	104.80	3046
4	0280-XJGEX	1	United States	California	Los Angeles	90015	34.039224, -118.266293	34.039224	-118.266293	Male	—	Month-to-month	Yes	Bank transfer (automatic)	103.70	503

Figure 4:the dataset

2.3 Features description

- customerID - Customer unique identifier
- gender - Customer gender - ['Female' 'Male']
- SeniorCitizen - Elderly or retired person
- Partner - - ['No' 'Yes']
- Dependents - If customer has dependents - ['No' 'Yes']
- Tenure - Customer lifespan (in months)
- PhoneService - - ['No' 'Yes']
- MultipleLines - - ['No' 'No phone service' 'Yes']
- InternetService - - ['No' 'No internet service' 'Yes']
- OnlineSecurity - - ['No' 'No internet service' 'Yes']
- OnlineBackup - - ['No' 'No internet service' 'Yes']
- DeviceProtection - - ['No' 'No internet service' 'Yes']
- TechSupport - - ['No' 'No internet service' 'Yes']
- StreamingTV - - ['No' 'No internet service' 'Yes']
- StreamingMovies - - ['No' 'No internet service' 'Yes']
- PaperlessBilling - - ['No' 'Yes']
- PaymentMethod - payment method - ['Bank transfer (automatic)', 'Credit card']
- Contract - Type of contract - ['Month-to-month' 'One year' 'Two year']
- (automatic)', 'Electronic check', 'Mailed check']
- MonthlyCharges - Monthly Recurring Charges
- TotalCharges - Life time value
- Churn - Churn value, the target vector - ['No' 'Yes']
- Count 1
- Country
- State
- City
- Zip Code
- Lat long
- Latitude
- Longitude
- Churn label

- Churn score
- CLTV
- Churn Reason

In detail we have a look at the target feature, the actual “Churn”. Therefore, we plot it accordingly and see that 26,5% Of the total amount of customer churn. This is important to know, so we have not the same proportion of Churned Customers to Non-Churned Customers in our training data

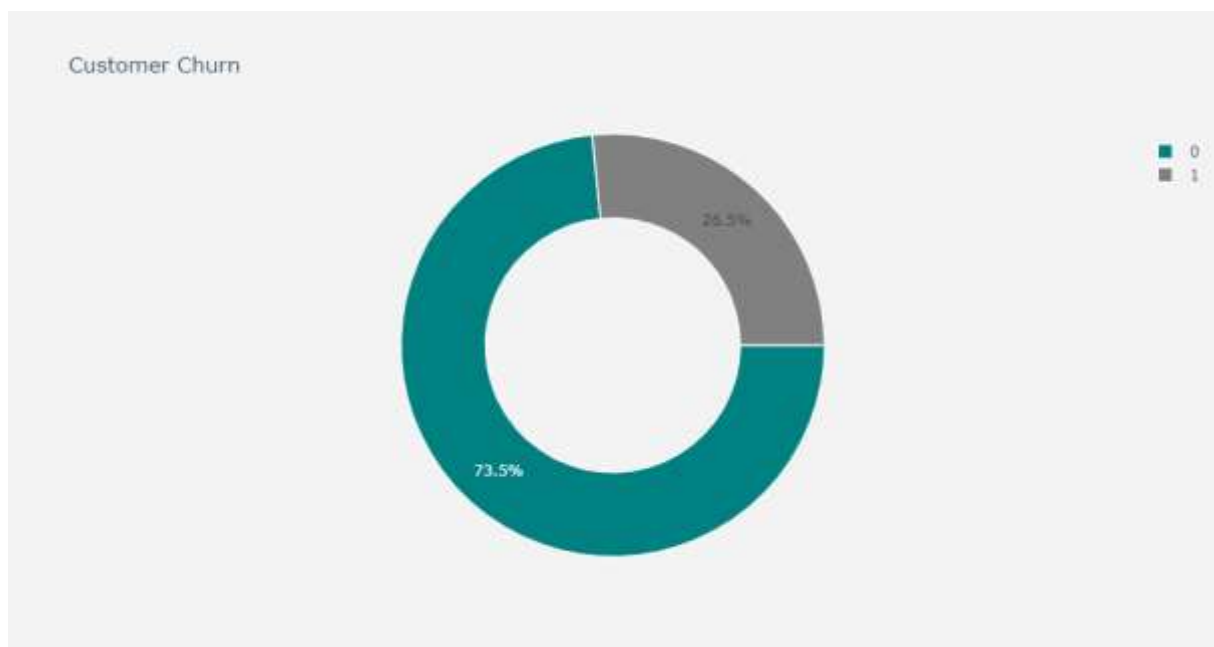


Figure 5: percentage of churn in Dataset

73.4630129206304 % of customers are not churners.

26.536987079369588 % of customers are churners.

Our data is unbalanced that's why we are going to use stratify=y in the split « method »

2.4 Visualizing data for categorical features

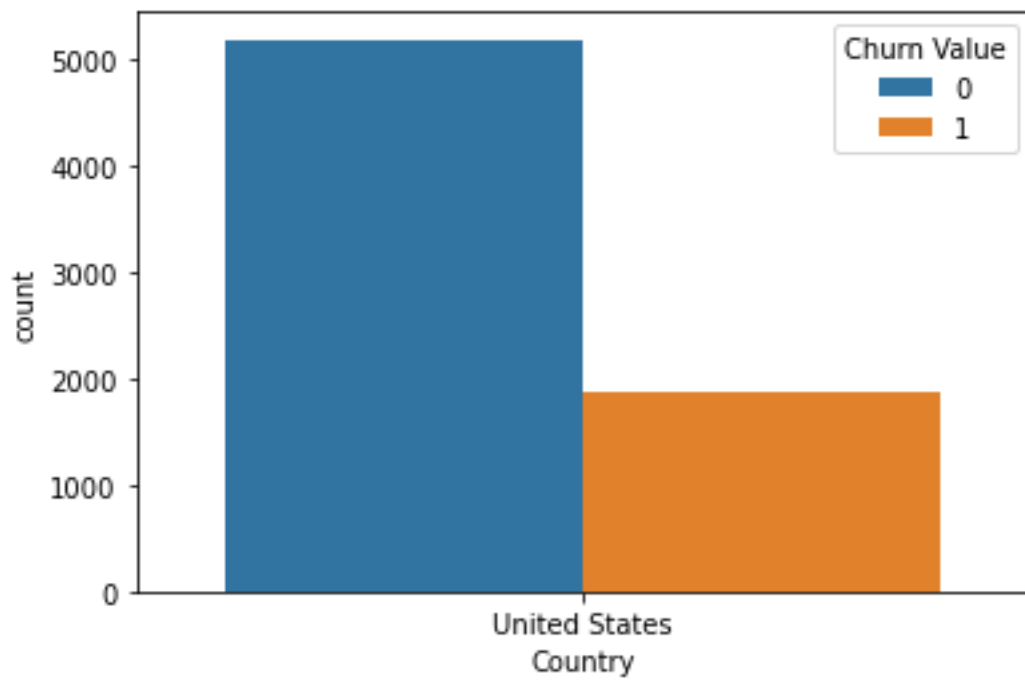


Figure 6: visualizing country feature

We can notice that all customers are from the United States.

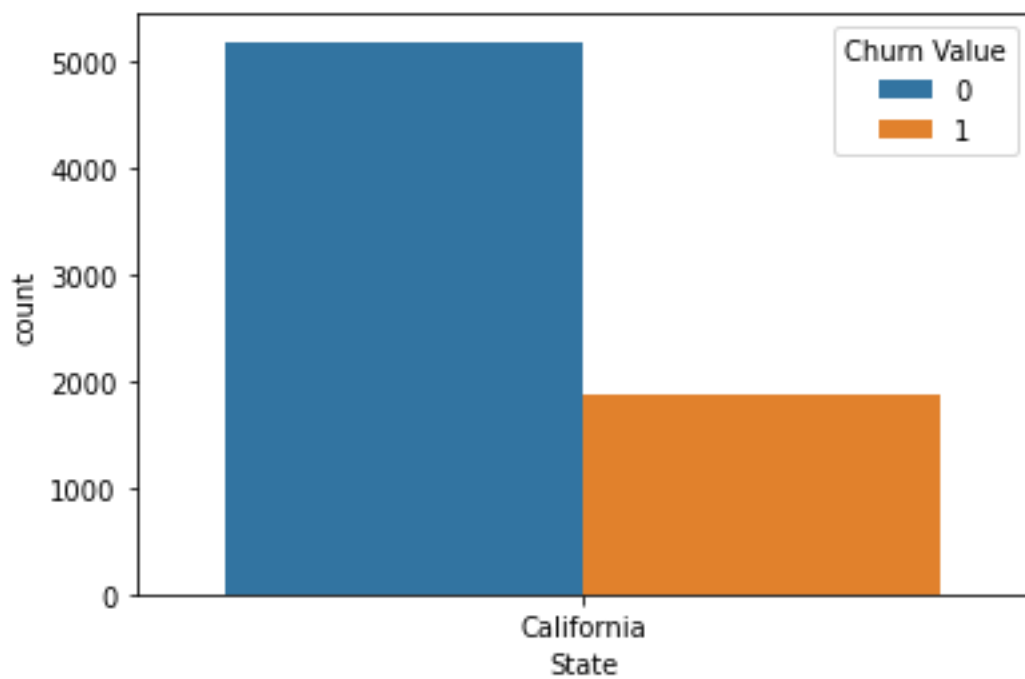


Figure 7: visualizing state feature

All consumers are from the state of California.

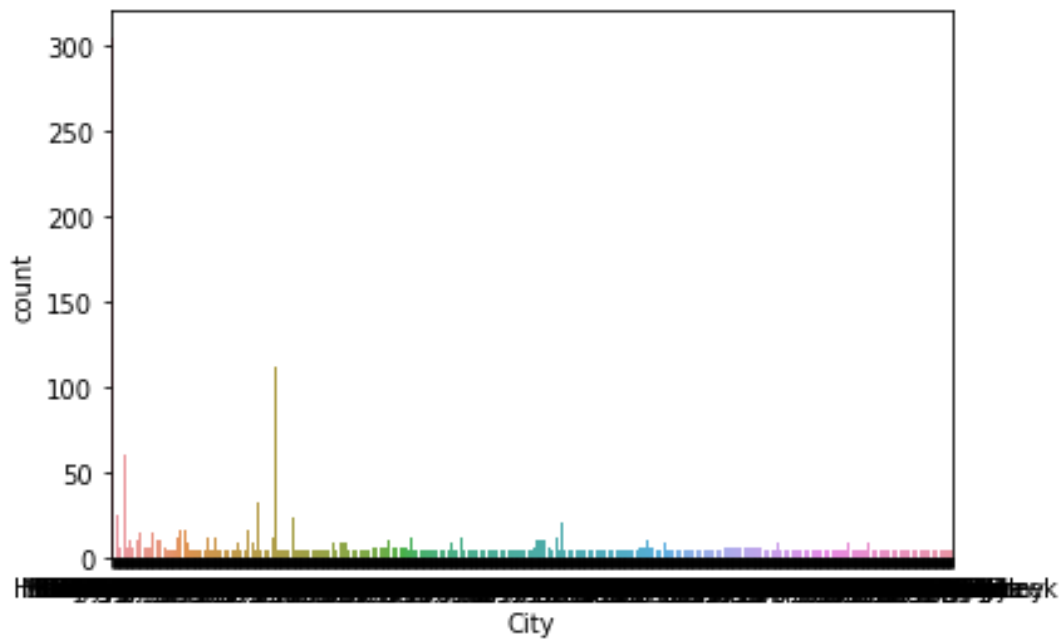


Figure 8: visualizing City feature

For the City and Lat Long features, they have many duplicated values so these features won't be good for classification model.

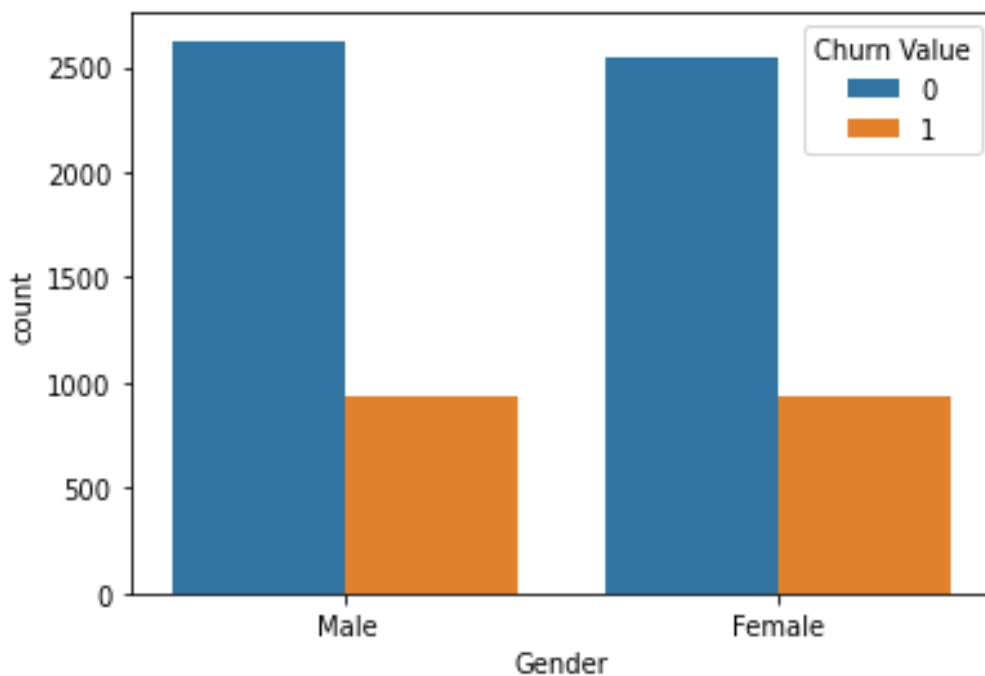


Figure 9: visualizing Gender Feature

There is no difference between male or female, so we can delete this feature too.

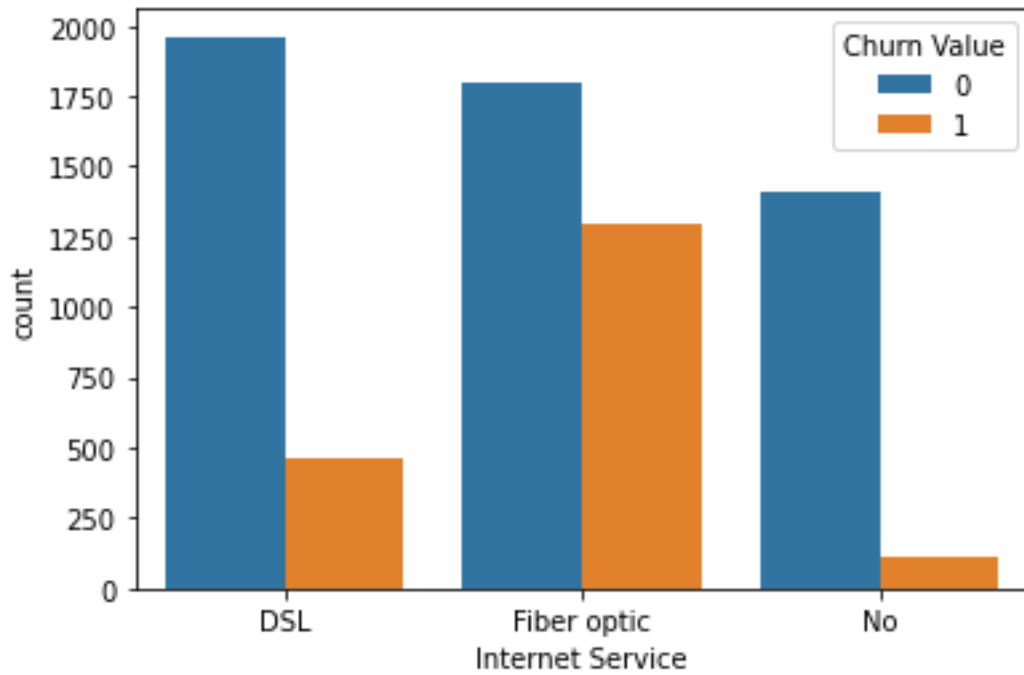


Figure 10: vizualizing internet Service feature

We can conclude that the feature "InternetService" is interesting, because it shows us a difference between whether the customer would churn or not depending on the type of internet service; that most customers that churned had the Fiber optic internet service, and the most customers that were retained had DSL internet service.

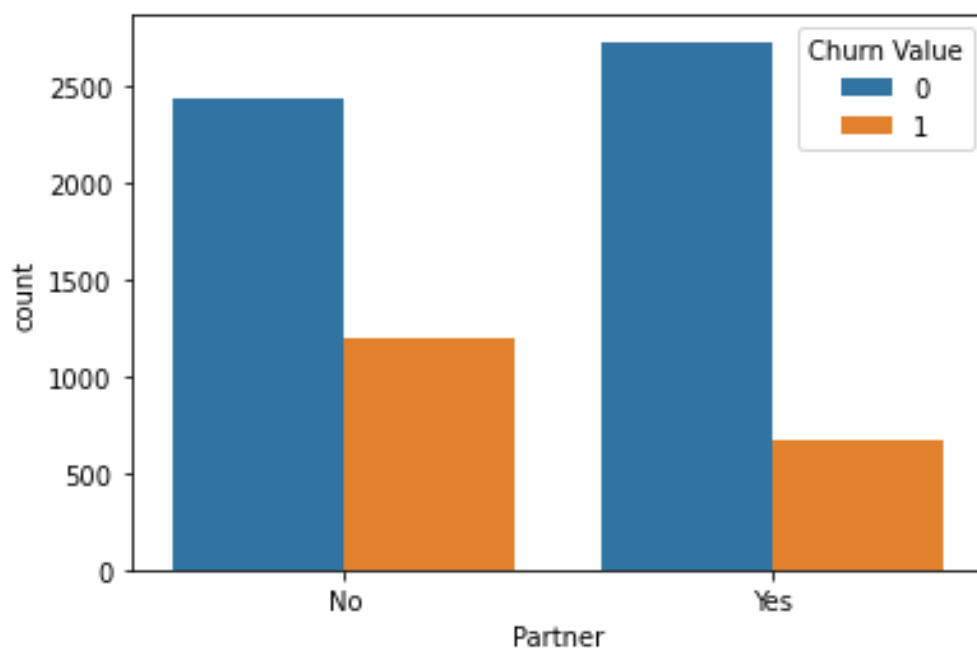


Figure 11: visualizing partner feature

We can notice that customers with no partner are most likely to leave

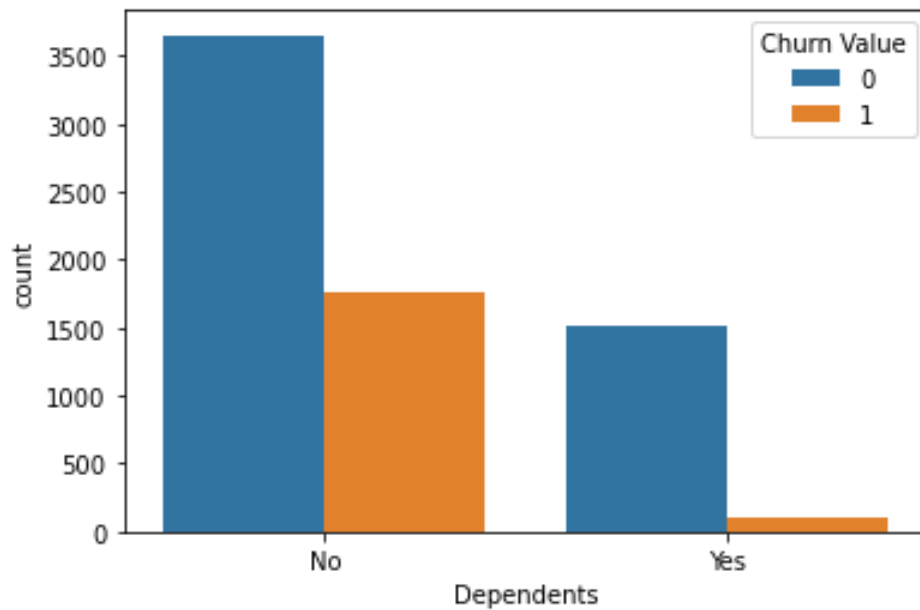
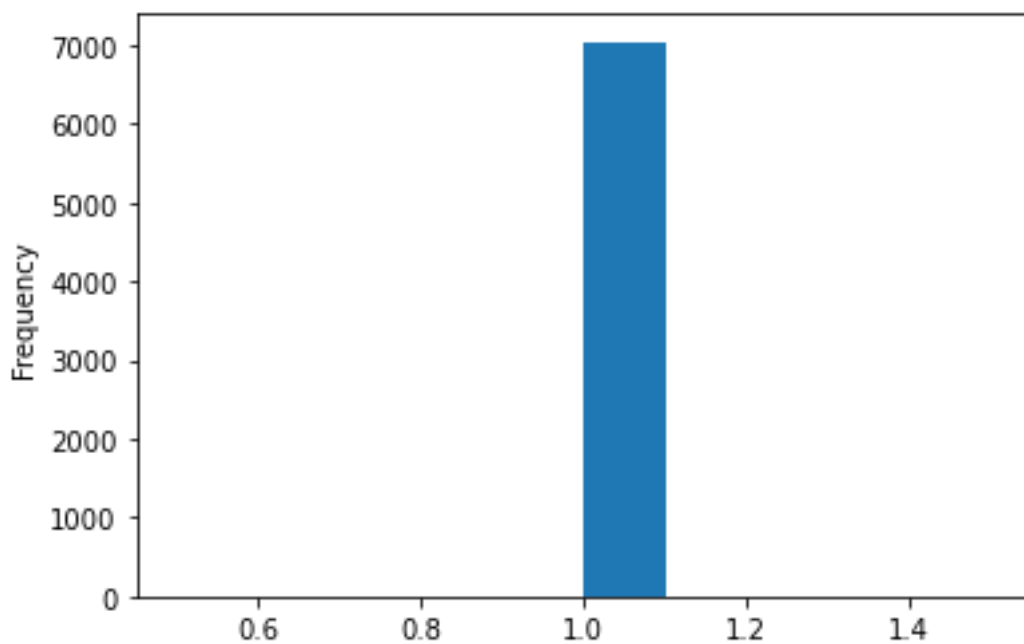


Figure 12:visualizing dependent feature

Customers with dependents are not leaving unlike customers without dependents.

2.5 Visualizing data for numerical features



.Figure 13: visualizing count feature

For the Count feature, we have only one value equal to 1.

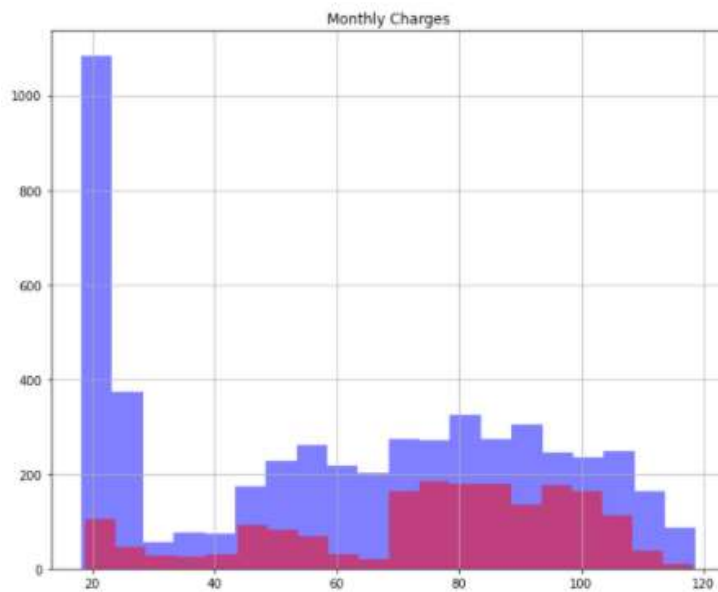


Figure 14:visualizing monthly charges feature

The monthly charges feature shows that most of the loyal customers that stayed with the company had a monthly charge between 20 and 30. Most of the customers that churned had a monthly charge of 70 to 105.

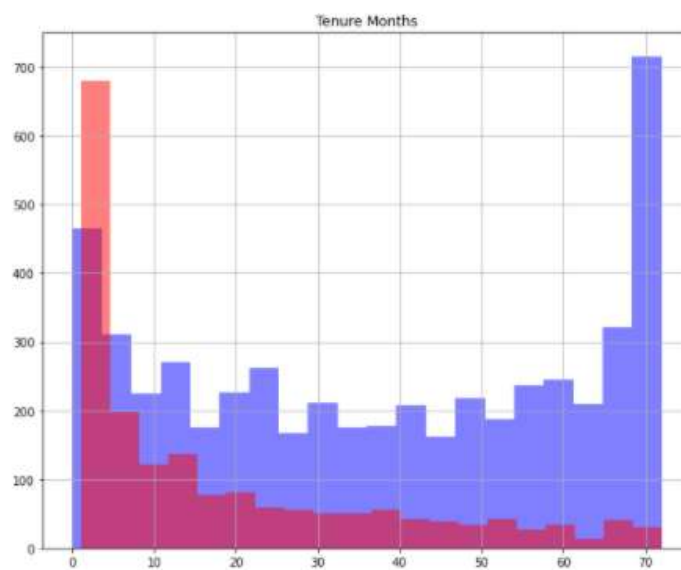


Figure 15:visualizing tenure charges feature

The tenure features shows that most of the customers that churned had between 1 and 8 months with the company, while most of the retained customers had a tenure between 50 and 72 months

2.6 Conclusion

In this chapter, we established an explanation of the process we followed to ensure the data comprehension stage, dealt with missing data and outliers and held a data overview. This data will be transformed and corrected during the data preparation phase which will be the subject of the next chapter.

CHAPTER 3: Data Preparation

3.1. Introduction

This phase is where the final dataset gets prepared to be used for modeling. It's based on human creativity, common sense, and business knowledge. It's one of the most important and time-consuming phases in the project. Preparing the data could spend 70% of a project's time and effort. Devoting adequate energy to the earlier phases of business understanding and data understanding obviously plays an important role in minimizing this additional cost. We will see how the data will be transformed and prepared for the next chapter "Modeling".

3.2 Feature selection

In this step we will decide which part of the data we are actually going to use in the modeling part. We will explain the rationale for inclusion and exclusion, which data will be used and which will not be used as part of the project.

Based on the interpretations of data Visualization and the unique values we are going to remove the following features: Country, State, Count, Gender, Lat Long, City, Zip Code, Lat Long, Latitude and Longitude.


```

Unique values (per feature):
Count                1
Country              1
State                1
City                1129
Zip Code            1652
Lat Long            1652
Latitude            1652
Longitude           1651
Gender               2
Senior Citizen       2
Partner              2
Dependents           2
Tenure Months        72
Phone Service         2
Multiple Lines        3
Internet Service      3
Online Security       3
Online Backup         3
Device Protection     3
Tech Support          3
Streaming TV          3
Streaming Movies      3
Contract              3
Paperless Billing      2
Payment Method        4
Monthly Charges      1584
Total Charges         6530
Churn Value           2
dtype: int64

```

Figure 16: Unique values per feature

Features to drop: 'CustomerID', 'Churn Label', 'Churn Score', 'CLTV', 'Churn Reason'

These features are not relevant for our project and more than 80% of the data in these columns is empty or even a duplication of another column under another name that we will detail in the data cleaning part

3.3 Data cleaning

In every data science project, it is practically impossible to have a perfect case and a database as clean, homogeneous as it is complete. Thus, it is absolutely necessary to correct several ambiguities. These problems must be fixed manually, either in the csv / excel file or with a

python script during the preprocessing phase. We must choose the necessary transformations according to the business objective which helps to increase the results of the model. In this case, we have offered the client some necessary changes regarding missing data and removing some inappropriate columns that will not affect the consistency of the database.

4.5.1. Sparseness Elimination

First step in Data preparation is Sparseness Elimination which is Searching for the columns having missing values and, in this step, we have many methods to resolve that. There are various ways to handle this issue: Drop rows with missing values, fill in the missing value with one of the following strategies: Most frequent values, zero, Mean of the values, Random value, etc...). In our case, we see that the Total Charges column has 11 missing values. So we will remove the rows with the missing values because it's only 11 missing values it won't affect the dataset (and our dataset contains 7034 rows). During this step, the necessary task to be done is to check the number of all the columns and to eliminate the empty and missing columns. The application of the predefined functions "isna ()" and "sum ()" is necessary to observe the number of missing values in each column.

4.5.2. Data conversion

From our data exploration (in this case "data.dtypes()") we can see that the data. Types columns MonthlyCharges and TotalCharges are numbers, but actually in the object format. Our machine learning model can only work with actual numeric data. Therefore, with the "to_numeric" function we can change the format and prepare the data for our machine learning model.

4.5.3. Outliers Detection

3.3.3.1 Definition

Outliers are extreme values that deviate from other observations on data, they may indicate a variability in a measurement, experimental errors or a novelty. In other words, an outlier is an observation that diverges from an overall pattern on a sample.

3.3.3.2 Types of outliers

Outliers can be of two kinds: univariate and multivariate:

- Univariate outliers can be found when looking at a distribution of values in a single feature space.
- Multivariate outliers can be found in a n-dimensional space (of n-features). Looking at distributions in n-dimensional spaces can be very difficult for the human brain, that is why we need to train a model to do it for us.

The second step in data preparation is Outliers Detection. The presence of outliers in the dataset can result in a poor fit and lower the predictive modeling performance. We created a function to visualize the features distribution which takes as parameters the dataset and the feature to plot

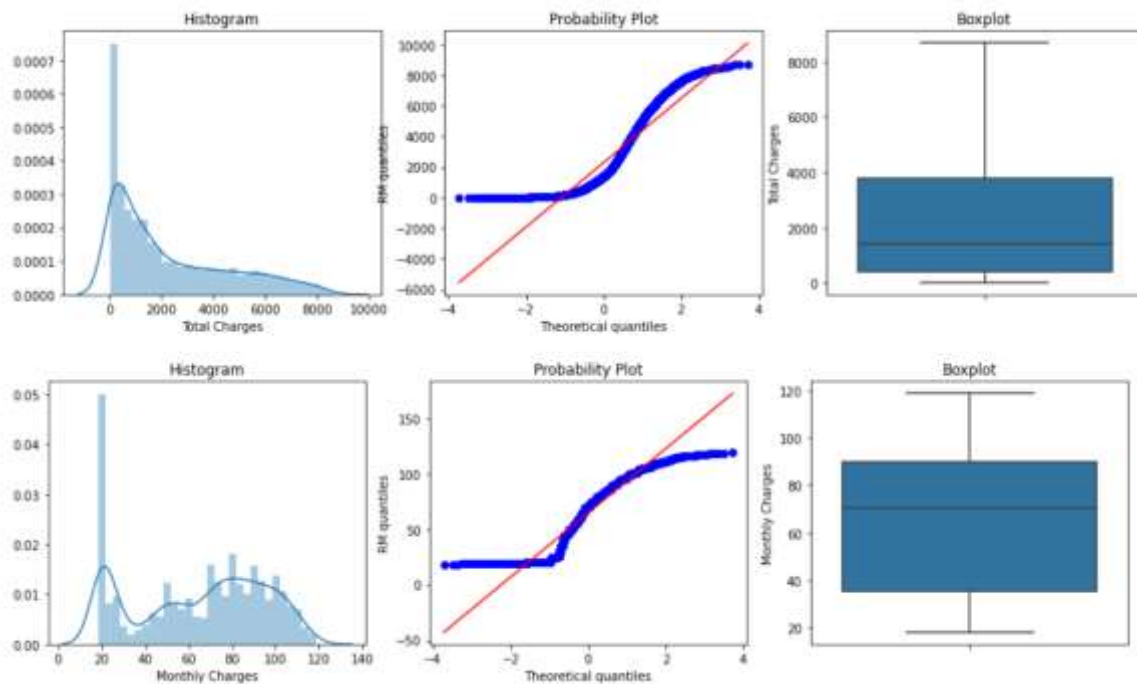


Figure 17: the features distribution

So the graphics showed us that both the Total Charges feature and Monthly Charges features haven't any outliers values also we can conclude that they doesn't have a normal (Gaussian)

distribution. In fact, for both Boxplot: Mounity charges and Total Charges the values didn't across the maximum or the minimum value line. So, we can affirm that we don't have outliers for these two features.

We also used quartile method to confirm that we don't have outliers in our dataset and we obtained to empty arrays for testing with Total Charges and Monthly Charges so it's mean we don't have outliers in these 2 features

3.5 Splitting the dataset

First our model needs to be trained, second our model needs to be tested. Therefore, it is best to have two different datasets. As for now we only have one, it is very common to split the data accordingly. X is the data with the independent variables, Y is the data with the dependent variable. The test size variable determines in which ratio the data will be split. It is quite common to do this in a 80 Training / 20 Test ratio.

Poor training and testing sets can lead to unpredictable effects on the output of the model. It may lead to overfitting or underfitting of the data and our model may end up giving biased results. That's why we use the `train_test_split` method (with the parameter `stratify` to split the data into training and test subsets that have the same proportions.

Train Set:

The train set would contain the data which will be fed into the model. In simple terms, our model would learn from this data.

Test Set:

The test set contains the data on which we test the trained and validated model. It tells us how efficient our overall model is and how likely is it going to predict something which does not make sense.

3.6 Encoding Categorical Data

There are three common approaches for converting ordinal and categorical variables to numerical values. They are:

- Ordinal Encoding
- One-Hot Encoding
- Dummy Variable Encoding

For our case, we have used the Ordinal Encoding, In ordinal encoding, each unique category value is assigned an integer value. For example, “red” is 1, “green” is 2, and “blue” is 3. This is called an ordinal encoding or an integer encoding and is easily reversible. Often, integer values starting at zero are used. For some variables, an ordinal encoding may be enough. The integer values have a natural ordered relationship between each other and machine learning algorithms may be able to understand and harness this relationship. It is a natural encoding for ordinal variables. For categorical variables, it imposes an ordinal relationship where no such relationship may exist. This can cause problems and a one-hot encoding may be used instead. This ordinal encoding transform is available in the scikit-learn Python machine learning library via the Ordinal Encoder class. By default, it will assign integers to labels in the order that is observed in the data. If a specific order is desired, it can be specified via the “categories” argument as a list with the rank order of all expected labels. Machine learning models require all input and output variables to be numeric. This means that if your data contains categorical data, you must encode it to numbers before you can fit and evaluate a model. The two most popular techniques are an Ordinal Encoding and a One-Hot Encoding. Through our work, we have used the Ordinal Encoding: Encode categorical features as an integer array. The input to this transformer should be an array-like of integers or strings, denoting the values taken on by categorical (discrete) features. The features are converted to ordinal integers. This results in a single column of integers (0 to $n_{\text{categories}} - 1$) per feature.

3.7 Data Standardization:

This transformation standardizes a function by scaling the unit variance after subtracting the mean. The unit variance is the result of dividing all the values by the standard deviation to result in a distribution which has a standard deviation of 1, the variance in this case is also equivalent to 1 (since variance = deviation- type squared).

Normalization is a technique often applied as part of data preparation for machine learning. The goal of normalization is to change the values of numeric columns in the dataset to a

common scale, without distorting differences in the ranges of values, to avoid building incorrect ML models while training and/or executing data analysis. For machine learning, every dataset does not require normalization. It is required only when features have different ranges. (Having features on a similar scale can help the gradient descent converge more quickly towards the minima.)

MinMaxScaler:

An alternative approach to Z-score normalization (or standardization) is the so-called Min-Max scaling (often also simply called "normalization" - a common cause for ambiguities). In this approach, the data is scaled to a fixed range - usually 0 to 1. The cost of having this bounded range - in contrast to standardization - is that we will end up with smaller standard deviations, which can suppress the effect of outliers.

A Min-Max scaling is typically done via the following equation:

$$X_{sc} = \frac{X - X_{min}}{X_{max} - X_{min}}.$$

Robust Scaler:

The Robust Scaler uses a similar method to the Min-Max scaler but it instead uses the interquartile range, rather than the min-max, so that it is robust to outliers. Therefore, it follows the formula:

$$x'_i = \frac{x_i - Q1}{Q3 - Q1}$$

For each feature. Of course, this means it is using the less of the data for scaling so it's more suitable for when there are outliers in the data.

StandardScaler:

The standard scaler assumes our data is normally distributed within each feature and will scale them such that the distribution is now centered around 0, with a standard deviation of 1. The mean and standard deviation are calculated for the feature and then the feature is scaled based on below formula :

$$z = \frac{x - \mu}{\sigma}$$

μ = Mean

σ = Standard Deviation

If data is not normally distributed, this is not the best scaler to use. In our notebook, we have created a list of the numerical features called Lnum with Lnum = ['Tenure Months', 'Monthly Charges', 'Total Charges', 'Churn Value']. Then we deleted Churn Value the target from the list and finally apply the Standardization with StandardScaler. StandardScaler standardize the features by removing the mean and scaling to unit variance.

CHAPTER 4 :Modeling

4.1 Introduction

The outputs of prediction and feature engineering are a set of label times, historical examples of what we want to predict, and features, predictor variables used to train a model to predict the label. The process of modeling means training a machine learning algorithm to predict the labels from the features, tuning it for the business need, and validating it on holdout data. Similar to feature engineering, modeling is independent of the previous steps in the machine learning process and has standardized inputs which means we can alter the prediction problem without needing to rewrite all our code. If the business requirements change, we can generate new label times, build corresponding features, and input them into the model. In this phase we are going to use supervised learning algorithms for classification. And we did use a remarkable number of classification algorithms starting by :GradientBoostingClassifier , XGBoost , AdaBoost, KNN, Random Forest, Naive Bayes, SVM. We start by searching the best parameters for each classifier using Grid Search. Then we evaluate the model using scoring, confusion matrix, etc and based on the evaluation, we try to improve the model using feature selection methods.

4.2 Used methods:

Here some definitions of some concepts and methods we have used.

Grid Search: Grid Search is an effective method for adjusting the parameters in supervised learning and improve the generalization performance of a model. In our work ,we have used The GridSearchCV instance witch implements the usual estimator API: when “fitting” it on a dataset all the possible combinations of parameter values are evaluated and the best combination is retained.

Cross validation: Cross-validation is a technique for evaluating models by training several models on subsets of the available input data and evaluating them on the complementary subset of the data. sometimes called rotation estimation or out-of-sample testing, is any of various similar model validation techniques for assessing how the results of

a statistical analysis will generalize to an independent data set. It is mainly used in settings where the goal is prediction, and one wants to estimate how accurately a predictive model will perform in practice. In a prediction problem, a model is usually given a dataset of known data on which training is run (training dataset), and a dataset of unknown data (or first seen data) against which the model is tested (called the validation dataset or testing set). The goal of cross-validation is to test the model's ability to predict new data that was not used in estimating it, in order to flag problems like overfitting or selection bias and to give an insight on how the model will generalize to an independent dataset.

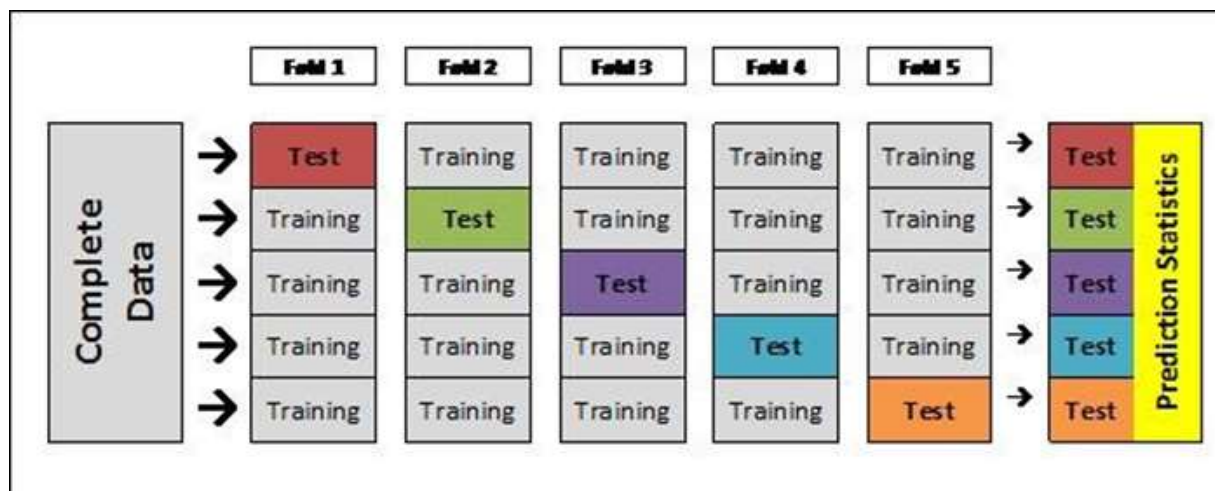


Figure 18: Cross validation process

4.3 Feature Selection

Feature selection is the process of reducing the number of input variables when developing a predictive model. It is desirable to reduce the number of input variables to both reduce the computational cost of modeling and, in some cases, to improve the performance of the model. Statistical-based feature selection methods involve evaluating the relationship between each input variable and the target variable using statistics and selecting those input variables that have the strongest relationship with the target variable. These methods can be fast and effective, although the choice of statistical measures depends on the data type of both the input and output variables.

Here a list of the methods that we used:

- **Correlation:** Features with high correlation are more linearly dependent and hence have almost the same effect on the dependent variable. So, when two features have high correlation, we can drop one of the two features.

We can compare all features with the target; features with lower correlation with the target aren't important

- **RFECV:** (Recursive Feature Elimination and Cross-Validation Selection) : an algorithm that eliminates irrelevant features based on validation scores. In fact, when the array gave us the ranking of each features, the feature having rank 1 have the highest importance and the feature having rank 5 have the lowest importance and must be deleted. The visualization plots the score relative to each subset and shows trends in feature elimination. If the feature elimination CV score is flat, then potentially there are not enough features in the model. An ideal curve is when the score jumps from low to high as the number of features removed increases, then slowly decreases again from the optimal number of features.
- **Select From Model:** Meta-transformer for selecting features based on importance weights.
- **SBFS:** (Sequential Backward Floating Selection) have an additional exclusion or inclusion step to remove features once they were included (or excluded), so that a larger number of feature subset combinations can be sampled
- **Variance Threshold:** A simple baseline approach to feature selection. It removes all features whose variance doesn't meet some threshold. By default, it removes all zero-variance features, i.e. features that have the same value in all samples.
- **Features importance visualization** Feature importance scores can be used to help interpret the data, but they can also be used directly to help rank and select features that are most useful to a predictive model.

NB: KNN does not provide logic to do feature selection. We cannot use sklearn's implementation to achieve such goal, unless we define our own measure of feature importance for KNN.

There is no such general object, and so scikit-learn does not implement it.

SVM on the other hand, like every linear model, provides such information.

4.4 Gradient Boost Classifier

4.4.1. Definition

Boosting is a special type of Ensemble Learning technique that works by combining several weak learners (predictors with poor accuracy) into a strong learner (a model with strong accuracy). This works by each model paying attention to its predecessor's mistakes.

The two most popular boosting methods are:

- Adaptive Boosting (Ada Boosting)
- Gradient Boosting

We will be discussing Gradient Boosting which involves three elements:

- A loss function to be optimized.
- A weak learner to make predictions.
- An additive model to add weak learners to minimize the loss function.

The term gradient boosting consists of two sub-terms, gradient and boosting. Gradient boosting re-defines boosting as a numerical optimization problem where the objective is to minimize the loss function of the model by adding weak learners using gradient descent. Gradient descent is a first-order iterative optimization algorithm for finding a local minimum of a differentiable function. As gradient boosting is based on minimizing a loss function, different types of loss functions can be used resulting in a flexible technique that can be applied to regression, multi-class classification, etc.

Intuitively, gradient boosting is a stage-wise additive model that generates learners during the learning process (i.e., trees are added one at a time, and existing trees in the model are not changed). The contribution of the weak learner to the ensemble is based on the gradient descent optimization process. The calculated contribution of each tree is based on minimizing the overall error of the strong learner. Gradient boosting does not modify the sample distribution as weak learners train on the remaining

residual errors of a strong learner). By training on the residuals of the model, this is an alternative means

to give more importance to misclassified observations. Intuitively, new weak learners are being added to

concentrate on the areas where the existing learners are performing poorly. The contribution of each

weak learner to the final prediction is based on a gradient optimization process to minimize the overall error of the strong learner.

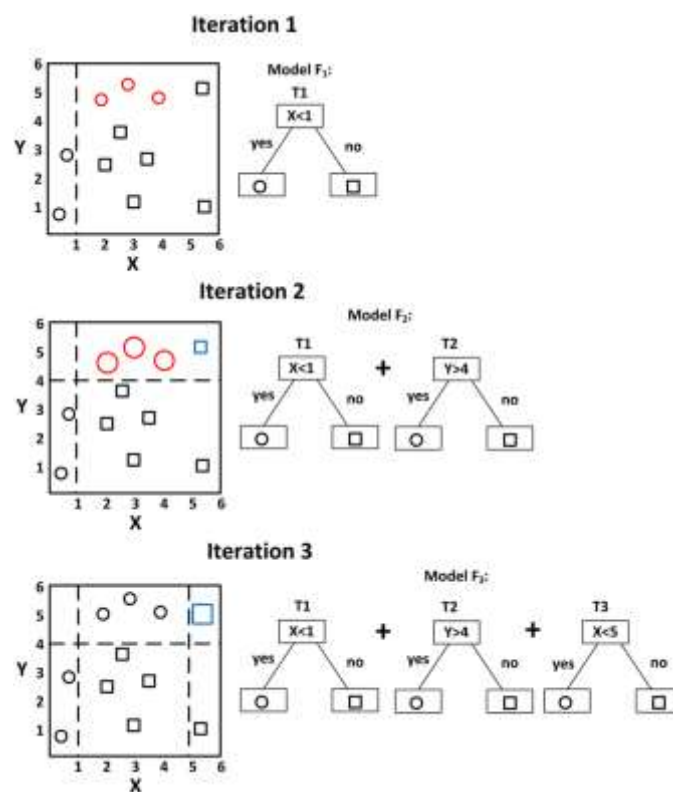


Figure 19: Gradient Boost Classifier

We start by searching the best parameters for each classifier using Grid Search. Then we tried to improve the model using feature selection methods.

4.4.2. Searching for the best parameters

We start by searching the best parameters for each classifier using Grid Search

In our case, the best parameters are:

criterion: friedman_mse

loss: deviance

max_features': sqrt

4.4.3. feature selection

We try to improve the model using feature selection methods.

First, we used the Features importance visualization

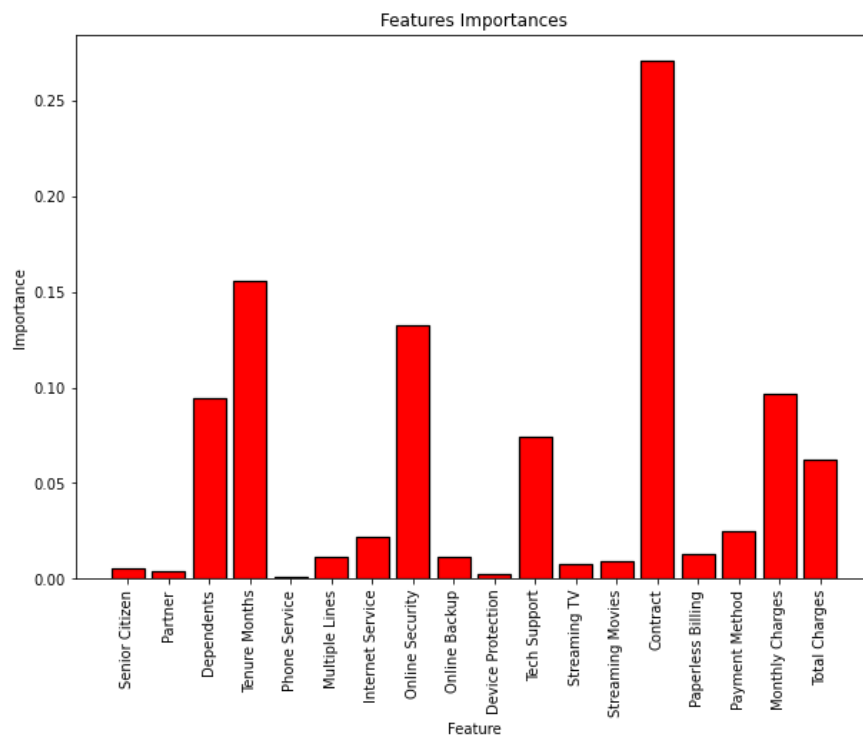


Figure 20: Features Importance Visualization

Then we used the RFECV method

```

Entrée [571]: selector = RFECV(final_Gbc,
                             step=1,
                             min_features_to_select=5,
                             cv=5)
selector.fit(X_train_Gboost, y_train_Gboost)
selector.grid_scores_

Out[571]: array([0.79893333, 0.80195556, 0.80497778, 0.80373333, 0.80462222,
                0.80533333, 0.80711111, 0.80764444, 0.80835556, 0.80924444,
                0.81048889, 0.80871111, 0.80995556, 0.80924444])

This array indicates the score of Gradient Boosting classifier at each iteration

Entrée [572]: selector.ranking_

Out[572]: array([2, 1, 1, 1, 4, 1, 1, 1, 1, 3, 1, 1, 1, 1, 1, 1])

```

Figure 21:RFECV method

This array gave us the ranking of each features. The feature having rank 1 have high importance

- Feature number 5 (Phone services) has a rank 4 so this feature has the lowest importance
- Feature number 10 (Device protection) has a rank 3 it's not an important feature

Based one the feature importance and features ranking we are going to remove the feature Phone services and the Device Protection features. Then, we have changed the X_train and X_test set according to the new features after selection.

4.5 XGBoost :

4.5.1. Definition :

XGBoost is an optimized distributed gradient boosting library designed to be highly efficient, flexible and portable. It implements machine learning algorithms under the Gradient Boosting framework. XGBoost provides a parallel tree boosting (also known as GBDT, GBM) that solve many data science problems in a fast and accurate way.

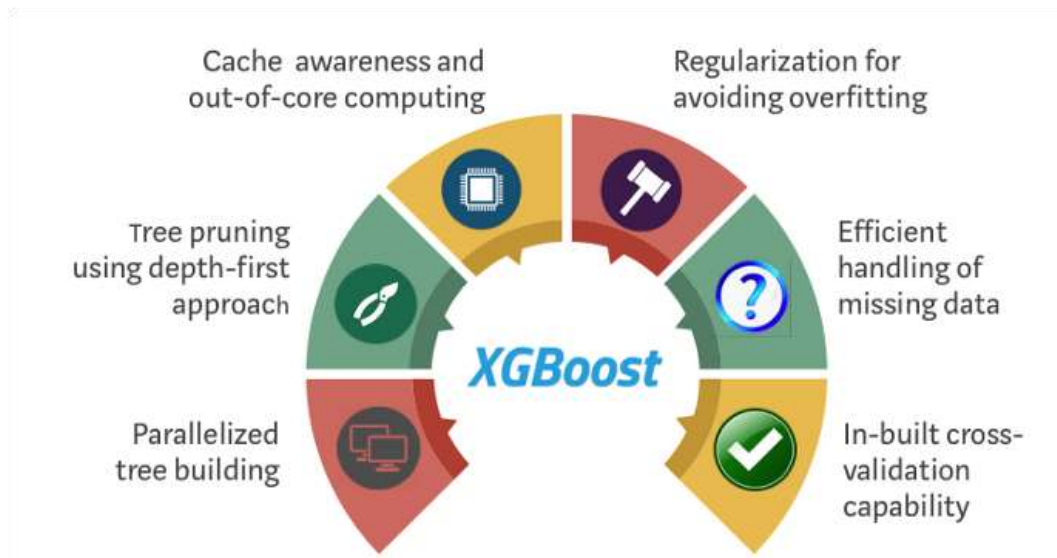


Figure 22:XGBoost Process

4.5.2. Searching for the best parameters

So after creating a copy of the train set and test set we have Searched for the best parameters with the method Grid Search.

The best parameters are:

learning_rate: 0.1

max_depth: 2

n_estimators: 140

4.5.3. Feature selection

we try to improve the model using feature selection methods.

First, we used also the Features importance visualization

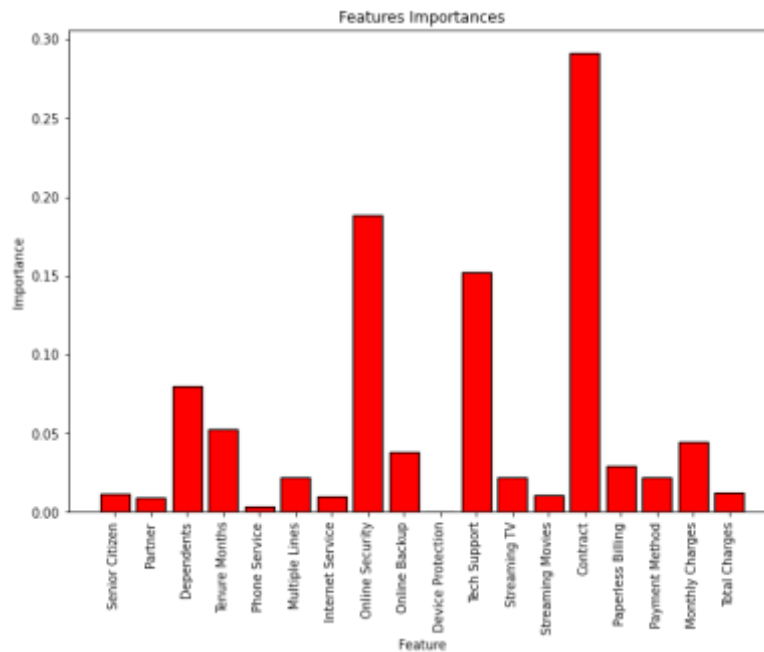


Figure 23: Features importance visualization

This figure shows that Device Protection has the lowest importance almost equal to 0 so we are going to remove this feature

Then we used the RFECV method

```
rfecv.ranking_
array([1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1])
```

This array gave us the ranking of each features. All the features as we can see are having rank 1 witch worth a high importance. We started by 18 feature and after the feature selection using RFECV we obtained 18 features this mean that there isn't any feature that we can remove which help us to improve the score of our model.

4.6 Ada Boost

4.6.1. Definition

AdaBoost (Adaptive Boosting) is a very popular boosting technique that aims at combining multiple weak classifiers to build one strong classifier.

A single classifier may not be able to accurately predict the class of an object, but when we group multiple weak classifiers with each one progressively learning from the others' wrongly classified objects, we can build one such strong model. The classifier mentioned here could be any of your basic classifiers, from Decision Trees (often the default) to Logistic Regression, etc.

A weak classifier is one that performs better than random guessing, but still performs poorly at designating classes to objects. For example, a weak classifier may predict that everyone above the age of 40 could not run a marathon but people falling below that age could. Now, we might get above 60% accuracy, but we would still be misclassifying a lot of data points!

Rather than being a model in itself, AdaBoost can be applied on top of any classifier to learn from its shortcomings and propose a more accurate model. It is usually called the “best out-of-the-box classifier” for this reason. Decision Stumps are like trees in a Random Forest, but not "fully grown." They have one node and two leaves. AdaBoost uses a forest of such stumps rather than trees. Stumps alone are not a good way to make decisions. A full-grown tree combines the decisions from all variables to predict the target value. A stump, on the other hand, can only use one variable to make a decision.

An Example of How AdaBoost Works (determine whether a person is "fit" (in good health) or not)

- Step 1: A weak classifier (e.g. a decision stump) is made on top of the training data based on the weighted samples. Here, the weights of each sample indicate how important it is to be correctly classified. Initially, for the first stump, we give all the samples equal weights.
- Step 2: We create a decision stump for each variable and see how well each stump classifies samples to their target classes. For example, in the diagram below we check for Age, Eating Junk Food, and Exercise. We'd look at how many samples are correctly or incorrectly classified as Fit or Unfit for each individual stump.
- Step 3: More weight is assigned to the incorrectly classified samples so that they're classified correctly in the next decision stump. Weight is also assigned to each classifier based on the accuracy of the classifier, which means high accuracy = high weight!
- Step 4: Reiterate from Step 2 until all the data points have been correctly classified, or the maximum iteration level has been reached.

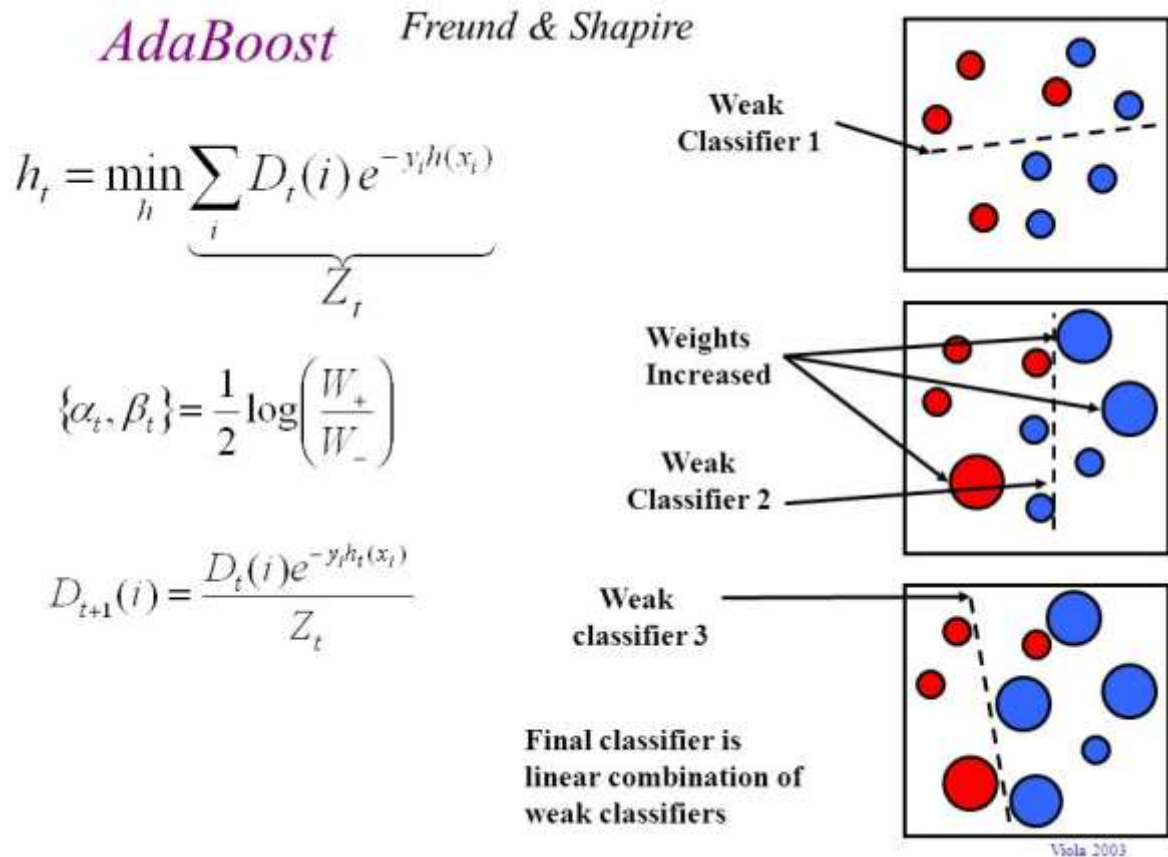


Figure 24: Ada Boost process

4.6.2. Searching for the best parameters

We start by searching for the best parameters with Grid search method and the parameters below are the best

learning_rate: 0.1

n_estimators: 1000

4.6.3. Feature Selection

With this model we are going to use RFECV for feature selection step

The results shows the array below ([1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1])

As we can see that all features have the same ranking a high raking. According to the result obtained by RFECV any feature selection would decrease the score of the model. Our model is already using the best 18 features

4.7 Logistic regression

4.7.1. Definition

A commonly used model is a sigmoid function. In the sigmoid function, also known as a squashing function, outputs are contained between the boundaries of 0 and 1. Here, we can use the model:

$$y = e^{(b_0 + b_1 \cdot x)} / (1 + e^{(b_0 + b_1 \cdot x)})$$

Note that in the function above, there are variables b_0 , and b_1 . These are called the weights, or coefficient values. b_0 represents the bias, or intercept, and b_1 is the coefficient. These weights are learned and trained from the existing data set. The product of this formula will produce a percentage, or probability, that will be mapped over discrete classes. The defined separation between two classes is known as the decision boundary. For example, if a probability is over, or under, a certain threshold it then falls into one or the other category.

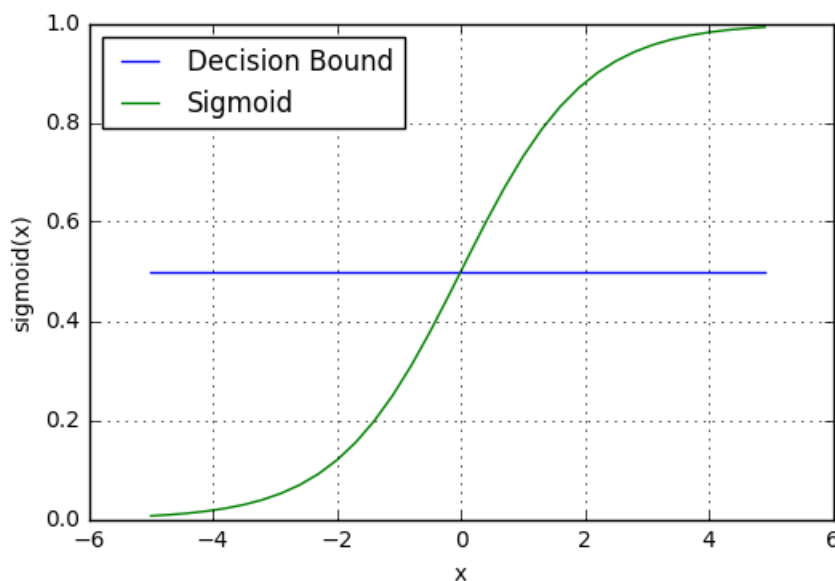


Figure 25: Logistic regression process

As logistic regression analysis is a great tool for understanding probability. A machine learning algorithm can take a given data set, analyze for weights and biases, and based upon

a defined decision boundary, can make predictions about a variable within the context of the function.

4.7.2. Searching for the best parameters

We start by searching for the best parameters with Grid search method and down below are the best ones:

penalty: l1

solver: saga

4.7.3. Feature Selection

With this model we are going to use RFECV for feature selection step

The results shows the array below ([1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1])

As we can see that all features have the same ranking a high raking. According to the result obtained by RFECV any feature selection would decrease the score of the model. Our model is already using the best 18 features

According to the result obtained by RFECV any feature selection would decrease the score of the model. Our model is already using the best 18 features

4.8 Tree Decision

4.8.1. Definition

Decision tree learning refers to a method based on the use of a decision tree as a predictive model. It is used in particular in data mining and machine learning.

In these tree structures, the leaves represent the values of the target variable and the branches correspond to combinations of input variables that lead to these values. In decision analysis, a decision tree can be used to explicitly represent the decisions made and the processes that lead to them. In learning and in data mining, a decision tree describes the data but not the decisions themselves, the tree would be used as a starting point for the decision process.

It is a supervised learning technique: we use a set of data for which we know the value of the target variable in order to build the tree (so-called labeled data), then we extrapolate the results to the set of data test. Decision trees are among the most popular algorithms in machine learning

4.8.2. Searching for the best parameters

We start by searching the best parameters for each classifier using Grid Search. {'criterion': 'gini', 'max_depth': 5}

Then we evaluate the model using scoring, confusion matrix and based on the evaluation, we will try to improve the model using feature selection methods. After creating the model, we displayed the train and test score and we obtained:

train score= 0.808

test score= 0.77114427860

4.8.3. Feature selection using Select From Model

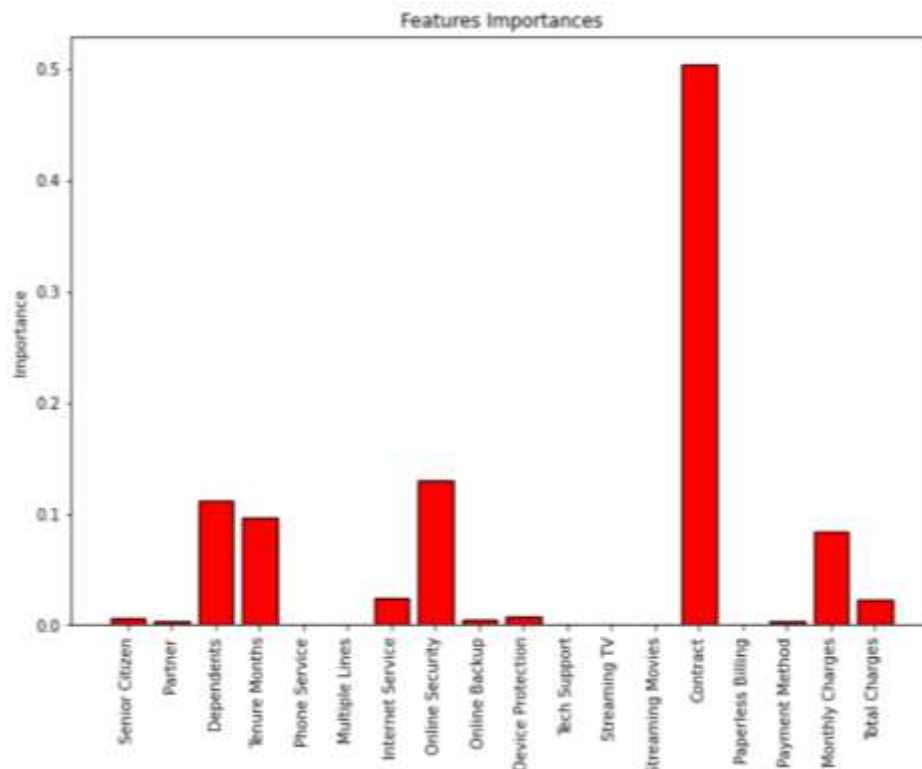


Figure 26: Features importance visualization

As we can see:

- Contract has the highest importance
- Dependents, Tenure Months, Online Security, Monthly Charges have an average importance
- Senior Citizen, Partner, Online Backup, Internet Service, Device Protection, Payment Method, Total Charges have a very low importance
- the rest of features are not important we can remove them

We tried to improve our model using the Select From Model selector and we obtained the following result:

Selected Features: Senior Citizen, Dependents, Tenure Months, Internet Service, Online Security, Device Protection, Contract, Monthly Charges, Total Charges

So, we started with 18 features but retained only 9 of them!

4.9 K-nearest neighbors (KNN)

4.9.1. Definition

In pattern recognition, the k nearest neighbors (k-NN) algorithm is a nonparametric method used for classification and regression. In both cases, it is a matter of classifying the entry in the category to which the k nearest neighbors belongs in the space of characteristics identified by learning. The result depends on whether the algorithm is used for classification or regression purposes: in k-NN classification, the result is a membership class. An input object is classified according to the majority result of the membership class statistics of its k nearest neighbors, (k is a generally small positive integer). If $k = 1$, then the object is assigned to the membership class of its close neighbor. in k-NN regression, the result is the value for that object. This value is the average of the values of the k nearest neighbors. The k-NN method is based on prior learning, or weak learning, where the function is evaluated locally, the final calculation being performed at the end of the classification. The k-NN algorithm is among the simplest of the machine learning algorithms

4.7.2- Searching for the best parameters

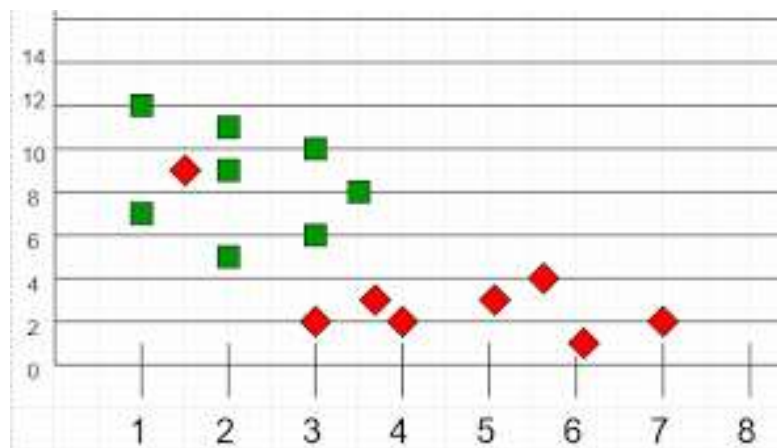


Figure 27: K-Nearest Neighbors

4.9.2. Searching for the best parameters

Like every algorithm we started by searching for the best parameters with GridSearch and we obtained the parameters below

'metric': 'manhattan'

'n_neighbors': 18,

'weights': 'uniform'

4.9.3. Feature Selection

4.9.3.1 Correlation Matrix

A correlation matrix is a tabular data representing the 'correlations' between pairs of variables in a given data. Each row and column represent a variable, and each value in this matrix is the correlation coefficient between the variables represented by the corresponding row and column. The Correlation matrix is an important metric that is computed to summarize data to understand the relationship between various variables and make decisions accordingly.

And we are going to use it to reduce the dimension of our dataset (features selection)

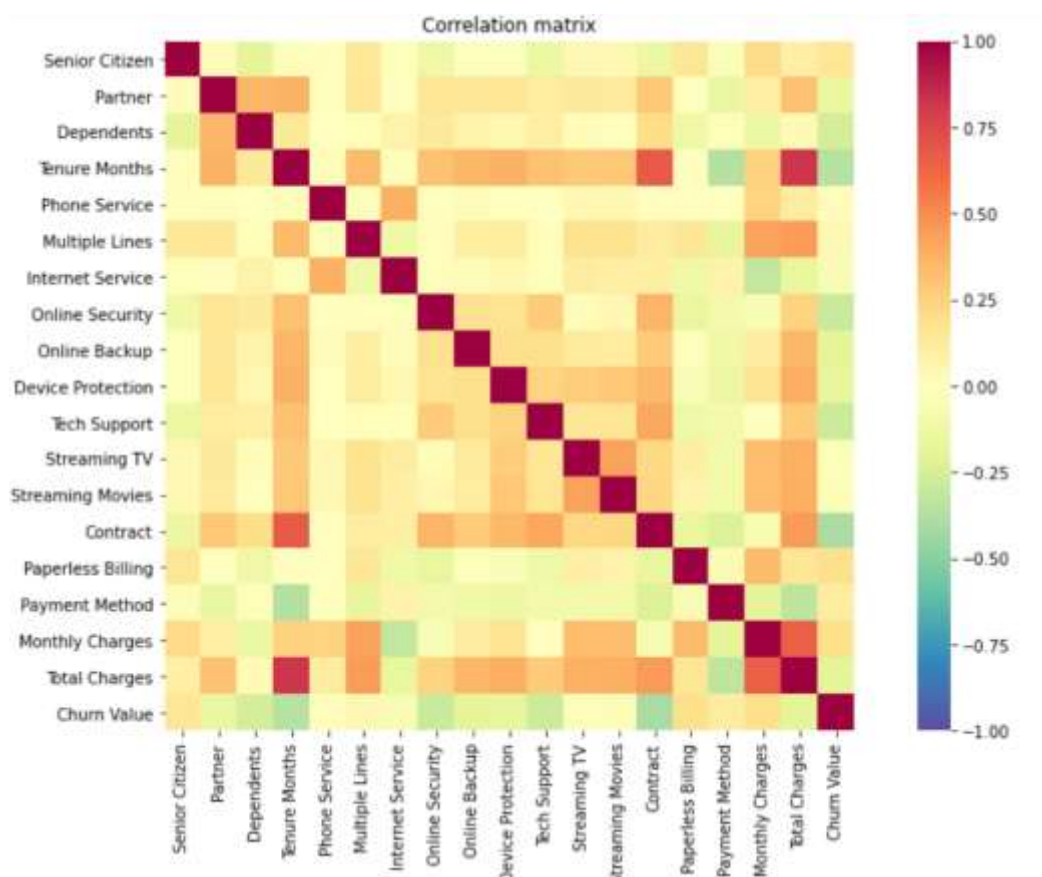


Figure 28: Correlation Matrix

	Senior Citizen	Partner	Dependents	Tenure Months	Phone Service	Multiple Lines	Internet Service	Online Security	Online Backup	Device Protection	Tech Support	Streaming TV	Streaming Movies	Churn Value
Senior Citizen	1.000000	0.019028	-0.180432	0.014013	0.003540	0.141540	-0.029871	-0.123781	-0.023071	-0.022693	-0.144611	0.046654	0.046509	-0.150970
Partner	0.019028	1.000000	0.363622	0.381717	0.009492	0.149845	0.001889	0.157440	0.149950	0.163133	0.121878	0.137855	0.126893	0.154306
Dependents	-0.180432	0.363622	1.000000	0.134950	-0.004062	-0.027210	0.079041	0.133821	0.082163	0.052833	0.103806	0.022590	0.004875	-0.254806
Tenure Months	0.014013	0.381717	0.134950	1.000000	0.009905	0.346069	-0.024787	0.318169	0.364522	0.380285	0.316714	0.286762	0.293167	-0.351710
Phone Service	0.003540	0.009492	-0.004062	0.009905	1.000000	-0.019027	0.387511	-0.009818	0.019743	-0.000173	-0.003948	0.053599	0.039597	0.009318
Multiple Lines	0.141540	0.149845	-0.027210	0.346069	-0.019027	1.000000	-0.111241	0.008813	0.107877	0.124802	0.008960	0.176312	0.179009	0.043010
Internet Service	-0.029871	0.001889	0.079041	-0.024787	0.387511	-0.111241	1.000000	-0.027911	0.029210	0.047065	-0.024059	0.110842	0.098251	-0.047040
Online Security	-0.123781	0.157440	0.133821	0.318169	-0.009818	0.008813	-0.027911	1.000000	0.180468	0.170020	0.276386	0.030852	0.053455	-0.169673
Online Backup	-0.023071	0.149950	0.082163	0.364522	0.019743	0.107877	0.029210	0.180468	1.000000	0.188067	0.188892	0.129081	0.118886	-0.292748
Device Protection	-0.022693	0.163133	0.052833	0.380285	-0.000173	0.124802	0.047065	0.170020	0.188067	1.000000	0.248731	0.271669	0.293366	-0.198804
Tech Support	-0.144611	0.121878	0.103806	0.316714	-0.003948	0.008960	-0.024059	0.276386	0.188892	0.248731	1.000000	0.153438	0.158250	-0.175113
Streaming TV	0.046654	0.137855	0.022590	0.286762	0.053599	0.176312	0.110842	0.030852	0.129081	0.271669	0.153438	1.000000	0.434434	-0.281997
Streaming Movies	0.046509	0.126893	0.004875	0.293167	0.039597	0.179009	0.098251	0.053455	0.118886	0.293366	0.158250	0.434434	1.000000	-0.025026
Contract	-0.134065	0.290174	0.202529	0.675395	0.006970	0.115689	0.102242	0.371281	0.276963	0.355023	0.418779	0.220204	0.231337	0.003074
Paperless Billing	0.154826	-0.010699	-0.121244	0.007407	0.003038	0.151021	-0.131251	-0.164190	-0.016916	-0.038070	-0.118391	0.103175	0.083416	-0.043070
Payment Method	-0.033978	-0.162671	-0.030284	-0.366558	-0.005743	-0.178739	0.081388	-0.090730	-0.126197	-0.135275	-0.098101	-0.098041	-0.109335	-0.047040
Monthly Charges	0.213735	0.101056	-0.141556	0.250410	0.248752	0.434846	-0.319196	-0.055029	0.115365	0.168645	0.001992	0.340767	0.334962	-0.047040
Total Charges	0.066939	0.318680	0.032178	0.827181	0.115183	0.453047	-0.169673	0.245858	0.364988	0.395603	0.274815	0.389699	0.391964	-0.047040
Churn Value	0.150970	-0.154306	-0.254806	-0.351710	0.009318	0.043010	-0.047040	-0.292748	-0.198804	-0.175113	-0.281997	-0.025026	-0.033074	-0.150970

Figure 29: Dataset representation

We are going to use the last row of our dataset which represent the target and we will try to filter out those feature whose correlation coefficient value with the target are greater than a threshold that we will determine later.

Here some results:

```

train score= 0.8133333333333334
test score= 0.7725657427149965
seuil 0.0

L ['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']
18
train score= 0.8140444444444445
test score= 0.7697228144989339
seuil 0.01

L ['Senior Citizen', 'Partner', 'Dependents', 'Tenure Months', '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']
17
train score= 0.8140444444444445
test score= 0.7697228144989339
seuil 0.02

```

Figure 30: Feature selection

We will choose the best threshold that gave us the best testing score which is equal to 0.04.

The new 16 features are:

Senior Citizen, Partner, Dependents, Tenure Months, Multiple Lines, Internet Service, Online Security, Online Backup, Device Protection, Tech Support, Streaming Movies, Contract, Paperless Billing, Payment Method, Monthly Charges, Total Charges

4.10 Random Forest

4.10.1 Definition

The Random forest is a classification algorithm consisting of many decisions trees. It uses bagging and feature randomness when building each individual tree to try to create an uncorrelated forest of trees whose prediction by committee is more accurate than that of any individual tree.

Random forest, like its name implies, consists of a large number of individual decision trees that operate as an ensemble. Each individual tree in the random forest spits out a class prediction and the class with the most votes becomes our model's prediction (see figure below).

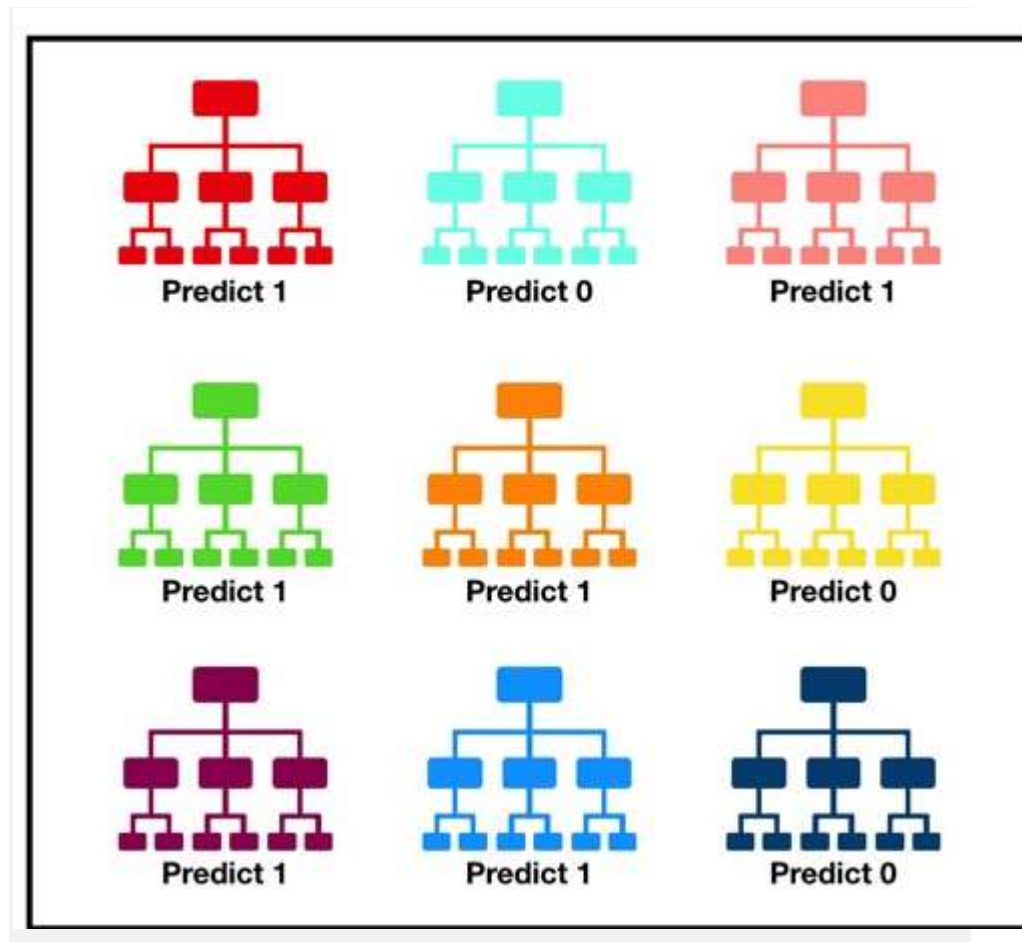


Figure 31: Random Forest

Visualization of a Random Forest Model Making a Prediction

The fundamental concept behind random forest is a simple but powerful one — the wisdom of crowds. In data science speak, the reason that the random forest model works so well is:

A large number of relatively uncorrelated models (trees) operating as a committee will outperform any of the individual constituent models.

The low correlation between models is the key. Just like how investments with low correlations (like stocks and bonds) come together to form a portfolio that is greater than the sum of its parts, uncorrelated models can produce ensemble predictions that are more accurate than any of the individual predictions. The reason for this wonderful effect is that the trees protect each other from their individual errors (as long as they don't constantly all err in the same direction). While some trees may be wrong, many other trees will be right, so as a

group the trees are able to move in the correct direction. So the prerequisites for random forest to perform well are:

1. The need to have some actual signal in our features so that models built using those features do better than random guessing.
2. The predictions (and therefore the errors) made by the individual trees need to have low correlations with each other.

4.10.2. Searching for the best parameters

We start by searching the best parameters for each classifier using Grid Search. Then we evaluate the model using scoring, confusion matrix, etc. and based on the evaluation, we try to improve the model using feature selection methods.

```
{'criterion': 'gini', 'max_depth': 5}
```

4.10.3. Feature Selection using RFECV

The results shows the array below ([1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1])

As we can see that all features have the same ranking a high raking. According to the result obtained by RFECV any feature selection would decrease the score of the model. Our model is already using the best 18 features. Based on the result obtained by RFECV we can conclude that any feature selection would decrease the performance of the model. This model is already using the best features

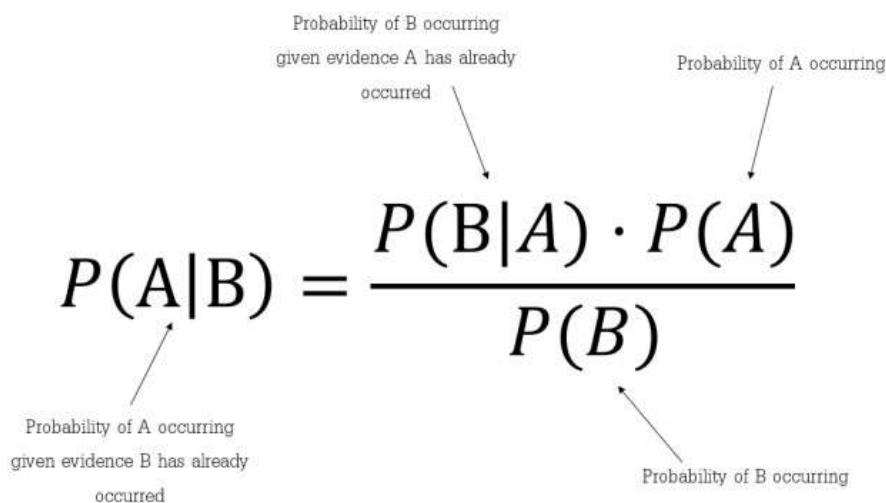
4.11 Naive Bayes

4.11.1. Definition

It is a classification technique based on Bayes' Theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.

Naive Bayes Classifier is a popular machine learning algorithm. It is a Supervised Learning algorithm used for classification. It is particularly useful for text classification issues. An example of the use of Naive Bayes is that of the spam filter.

Conditional probabilities: The naive Bayes classifier is based on Bayes' theorem. The latter is a classic of probability theory. This theorem is based on conditional probabilities



The diagram shows the formula for Bayes' Theorem:
$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$$
 Annotations with arrows pointing to the terms in the formula:

- An arrow from the text "Probability of B occurring given evidence A has already occurred" points to $P(B|A)$.
- An arrow from the text "Probability of A occurring" points to $P(A)$.
- An arrow from the text "Probability of A occurring given evidence B has already occurred" points to $P(A|B)$.
- An arrow from the text "Probability of B occurring" points to $P(B)$.

4.11.2. Searching for the best parameters

We start by searching the best types for this classifier: The best type is Gaussian that we will use after as our classifier.

4.11.3 Feature Selection with SBFS

The floating variants, SBFS, can be considered as extensions to the simpler SBS algorithms. The floating algorithms have an additional exclusion or inclusion step to remove features once they were included (or excluded), so that a larger number of feature subset combinations can be sampled.

For our convenience, we can visualize the output from the feature selection in a pandas DataFrame format using the `get_metric_dict` method of the Sequential Feature Selector object. The columns `std_dev` and `std_err` represent the standard deviation and standard errors of the cross-validation scores, respectively. Below, we see the DataFrame of the Sequential Forward Selector:

Case 1: So here we started by `k_features = 1`

	feature_idx	cv_scores	avg_score	feature_names	ci_bound	std_dev	std_err
18	(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, ...)	[0.7888888888888888, 0.784, 0.7724444444444445, ...]	0.769244	(Senior Citizen, Partner, Dependents, Tenure M...	0.0126162	0.00997142	0.00498571
17	(0, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, ...)	[0.7706666666666667, 0.7893333333333333, 0.783, ...]	0.7728	(Senior Citizen, Dependents, Tenure Months, Ph...	0.0162955	0.0126784	0.00633622
16	(2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 1...	[0.7733333333333333, 0.7886666666666666, 0.785, ...]	0.774933	(Dependents, Tenure Months, Phone Service, Mul...	0.0127508	0.00992057	0.00496029
15	(2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 14, 15, 1...	[0.7733333333333333, 0.7875555555555556, 0.787, ...]	0.775289	(Dependents, Tenure Months, Phone Service, Mul...	0.0139813	0.0109779	0.00543895
14	(2, 3, 4, 6, 7, 8, 9, 10, 11, 13, 14, 15, 16, 17)	[0.7724444444444445, 0.7902222222222223, 0.788, ...]	0.776178	(Dependents, Tenure Months, Phone Service, Int...	0.0148153	0.0115288	0.0057634
13	(0, 2, 3, 4, 5, 6, 7, 9, 10, 14, 15, 16, 17)	[0.7786666666666666, 0.7902222222222223, 0.787, ...]	0.780622	(Senior Citizen, Dependents, Tenure Months, Ph...	0.01118	0.0086984	0.0043492
12	(0, 2, 3, 4, 5, 7, 9, 10, 14, 15, 16, 17)	[0.7786666666666666, 0.7955555555555556, 0.784, ...]	0.780444	(Senior Citizen, Dependents, Tenure Months, Ph...	0.0140854	0.010959	0.00547948
11	(0, 2, 3, 4, 7, 9, 10, 14, 15, 16, 17)	[0.7804444444444445, 0.7928888888888889, 0.784, ...]	0.7792	(Senior Citizen, Dependents, Tenure Months, Ph...	0.0138537	0.0107787	0.00538934
10	(0, 2, 3, 7, 9, 10, 14, 15, 16, 17)	[0.7795555555555556, 0.7928888888888889, 0.784, ...]	0.778844	(Senior Citizen, Dependents, Tenure Months, On...	0.0134871	0.0104934	0.0052467
9	(2, 3, 5, 7, 9, 10, 14, 15, 16)	[0.7804444444444445, 0.7911111111111111, 0.776, ...]	0.779378	(Dependents, Tenure Months, Multiple Lines, On...	0.00886141	0.00689448	0.00344724
8	(2, 3, 7, 10, 12, 14, 15, 16)	[0.7822222222222223, 0.7884444444444444, 0.776, ...]	0.779911	(Dependents, Tenure Months, Online Security, T...	0.00628261	0.00488808	0.00244404
7	(3, 4, 7, 10, 14, 15, 16)	[0.7733333333333333, 0.8106666666666666, 0.768, ...]	0.781689	(Tenure Months, Phone Service, Online Security...	0.0201791	0.0156969	0.00784844
6	(3, 7, 10, 14, 15, 16)	[0.7751111111111111, 0.8088888888888889, 0.770, ...]	0.780989	(Tenure Months, Online Security, Tech Support...	0.0200843	0.0156263	0.00781313
5	(3, 7, 14, 15, 16)	[0.7687777777777778, 0.8106666666666666, 0.774, ...]	0.778133	(Tenure Months, Online Security, Paperless Bi...	0.0214178	0.0168638	0.00833188
4	(3, 4, 5, 16)	[0.7786666666666666, 0.8026666666666666, 0.800, ...]	0.790044	(Tenure Months, Phone Service, Multiple Lines...	0.0126067	0.00980844	0.00480422
3	(3, 5, 16)	[0.776, 0.8035555555555556, 0.7946666666666666, ...]	0.788978	(Tenure Months, Multiple Lines, Monthly Charges)	0.0123178	0.00958352	0.00479178
2	(3, 16)	[0.7742222222222223, 0.7928888888888889, 0.799, ...]	0.7856	(Tenure Months, Monthly Charges)	0.0116376	0.00905446	0.00452723
1	(16,)	[0.7342222222222222, 0.7342222222222222, 0.734, ...]	0.734222	(Monthly Charges,)	0	0	0

Figure 32: The Dataframe of the Sequential Selector

The `ci_bound` column in the DataFrames above represents the confidence interval around the computed cross-validation scores. By default, a confidence interval of 95% is used, but we can

use different confidence bounds via the `confidence_interval` parameter. Now, we tried to add every time one more feature and compare the scores. We got as a result:

k_features	Score	Feature names
1	0.7342222	Monthly Charges
2	0.7856000	Tenure Months Monthly Charges
3	0.7889777	Tenure Months Multiple Lines Monthly Charges
4	0.7713777	Tenure Months Online Security Paperless Billing Monthly Charges

The table shows that in the 3 first rows the score is increasing yet starting row 4 the score decreases which means that the best score is obtained using the following 3 features: Tenure Months, Multiple Lines and Monthly

4.12 Standard SVM

4.12.1. Definition

The standard SVM algorithm is formulated for binary classification problems, and multiclass problems are typically reduced to a series of binary ones. SVMs are a family of machine learning algorithms that solve both classification, regression and anomaly detection problems. They are known for their strong theoretical guarantees, their great flexibility and their ease of use even without a great knowledge of data mining

4.12.2. Searching for the best parameters

We start by searching the best parameters for each classifier using Grid Search.

```
'C': 100,  
'gamma': 0.01,  
'kernel': 'rbf'
```

4.12.3. Features Selection using Variance Threshold

According to the results, we found that our model is already using the best features. The feature selection won't help improving the score of the model

Conclusion

In this chapter, We have defined all the algorithms we worked with ,in fact The process of modeling means training a machine learning algorithm to predict the labels from the features, tuning it for the business need, and validating it on holdout data.and the next chapter we will evaluate each algorithm.

Chapter 5 Evaluation

5.1 Definitions

Here some definitions of some concepts that we have used:

Confusion Matrix

Confusion matrix is a clear way to evaluate a classification model as it erases ambiguity about the performance of the model on each class. Let's stay in the example of a binary classifier, which predicts 2 classes: class 0 and class 1.

To measure the performance of this classifier, we need to distinguish 4 types of elements:

- **True Positive (TP):** It refers to the number of predictions where the classifier correctly predicts the positive class as positive (elements of the class 1 correctly predicted).
- **True Negative (TN):** It refers to the number of predictions where the classifier correctly predicts the negative class as negative (elements of class 0 correctly predicted).
- **False Positive (FP):** It refers to the number of predictions where the classifier incorrectly predicts the negative class as positive (elements of class 1 wrongly predicted).
- **False Negative (FN):** It refers to the number of predictions where the classifier incorrectly predicts the positive class as negative (elements of class 0 wrongly predicted).

This information can be gathered and displayed in tabular form in a confusion matrix. In the case of a binary classifier, we obtain:

		True Class	
		Positive	Negative
Predicted Class	Positive	TP	FP
	Negative	FN	TN

Classification report

The classification report visualizer displays the precision, recall, F1, and support scores for the model.

- The **recall** It tells you what fraction of all positive samples were correctly predicted as positive by the classifier. It is also known as True Positive Rate (TPR), Sensitivity, Probability of Detection. To calculate Recall, use the following formula:

$$TP/(TP+FN)$$

- The **precision** It tells you what fraction of predictions as a positive class were actually positive. To calculate precision, use the following formula:

$$TP/(TP+FP)$$

- The **support** is the number of occurrences of the given class in your dataset
- The **accuracy**: It gives you the overall accuracy of the model, meaning the fraction of the total samples that were correctly classified by the classifier. To calculate accuracy, use the following formula:

$$(TP+TN)/(TP+TN+FP+FN).$$

- The **f1-score**: It combines precision and recall into a single measure. Mathematically it's the harmonic mean of precision and recall. It can be calculated as follows:

$$F_1\text{-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2TP}{2TP + FP + FN}$$

- **Specificity**: It tells you what fraction of all negative samples are correctly predicted as negative by the classifier. It is also known as True Negative Rate (TNR). To calculate specificity, use the following formula:

$$TN/(TN+FP)$$

- **Misclassification Rate**: It tells you what fraction of predictions were incorrect. It is also known as Classification Error. You can calculate it using :

$$(FP+FN)/(TP+TN+FP+FN) \text{ or } (1-\text{Accuracy}).$$

5.2 Gradient Boost Classifier

5.2.1. Train and test scoring:

Now we display the train score and the test score before and after features selection:

	Train score	Test score
Before feature selection	0.827377777	0.798862828713575
After feature selection	0.8252444444444444	0.808813077469793

So as we can constate that after features selection, test score have been actually increased.

5.2.2. Classification report

a. before feature selection:

	precision	recall	f1-score	support
0	0.85	0.89	0.87	1033
1	0.64	0.56	0.60	374
accuracy			0.80	1407
macro avg	0.74	0.72	0.73	1407
weighted avg	0.79	0.80	0.79	1407

As it shows, the accuracy is 0.80 and the recall for non-churners customer is 0.89 and for those churner customer is 0.56. Precision and recall is highly used for imbalanced dataset because in a highly imbalanced dataset, a 99% accuracy can be meaningless in our case is 0.80.

b. After feature selection:

	precision	recall	f1-score	support
0	0.85	0.89	0.87	1033
1	0.66	0.57	0.62	374
accuracy			0.81	1407
macro avg	0.76	0.73	0.74	1407
weighted avg	0.80	0.81	0.80	1407

After the feature selection using RFECV the train score has decreased a little bit but the test score, the accuracy, and we constate the recall and the precision have improved

5.2.3. Confusion matrix visualization before feature selection

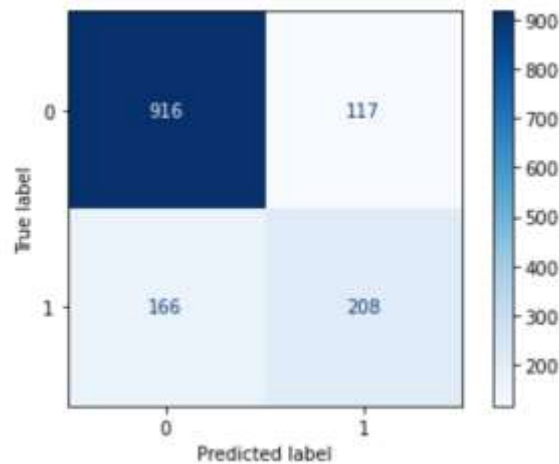


Figure 33: Confusion matrix

It can be seen in confusion matrix, the true predicted label for non-churner customer is 916, and for those who will not quit but it shows that they will quit is 117 so we can admit that it is acceptable. Now for the churned customers who said that they will quit and they actually quit are 208 and the rest 166 says that they will not churn but they did.

5.3 XGBoost

5.3.1. Train and test scoring:

Now we display the train score and the test score before and after features selection:

	Train score	Test score
Before feature selection	0.8250666666666666	0.7974413646055437
After feature selection	0.8250666666666666	0.7974413646055437

So as we can conclude that after features selection, test score have been actually increased

5.3.2. Classification report

a. before feature selection:

	precision	recall	f1-score	support
0	0.84	0.89	0.87	1033
1	0.64	0.55	0.59	374
accuracy			0.80	1407
macro avg	0.74	0.72	0.73	1407
weighted avg	0.79	0.80	0.79	1407

As it shows, the accuracy is 0.80 and the recall for non-churner customer is 0.89 and for the churner customer is 0.55.

b. after feature selection:

	precision	recall	f1-score	support
0	0.84	0.89	0.87	1033
1	0.64	0.55	0.59	374
accuracy			0.80	1407
macro avg	0.74	0.72	0.73	1407
weighted avg	0.79	0.80	0.79	1407

the train score, test score, accuracy, recall and the precision remain the same but it helps reducing the training Time. Less data means the algorithm train faster.

5.3.3. Confusion matrix visualization before feature selection

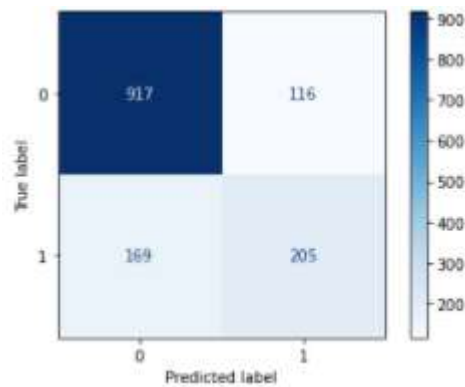


Figure 34: Confusion matrix

It can be seen in confusion matrix, the true predicted label for non-churners customer is 917, and for those who will not quit but it shows that they will quit is 116 so we can admit that it is acceptable. Now for the churned customers who said that they will quit and they actually quit are 205 and the rest 169 says that they will not churn but they did.

5.4 Ada Boost

5.4.1. Train and test scoring:

Now we display the train score and the test score before and after features selection:

	Train score	Test score
Before feature selection	0.8174222222222223	0.7995735607675906
After feature selection	0.8174222222222223	0.7995735607675906

Both train and test score remain the same.

5.4.2. Classification report

- a. before feature selection:

	precision	recall	f1-score	support
0	0.85258216	0.87899322	0.86558627	1033
1	0.63450292	0.58021390	0.60614525	374
accuracy			0.79957356	1407
macro avg	0.74354254	0.72960356	0.73586576	1407
weighted avg	0.79461369	0.79957356	0.79662327	1407

As it shows, the accuracy is 0.799 and the recall for non-churner customer is 0.87 and for those churner customer is 0.58.

b. after feature selection:

	precision	recall	f1-score	support
0	0.85258216	0.87899322	0.86558627	1033
1	0.63450292	0.58021390	0.60614525	374
accuracy			0.79957356	1407
macro avg	0.74354254	0.72960356	0.73586576	1407
weighted avg	0.79461369	0.79957356	0.79662327	1407

Removing feature doesn't affect the performance of the accuracy, the recall and the precision. But it reduces the execution time for training.

5.4.3. Confusion matrix visualization before feature selection

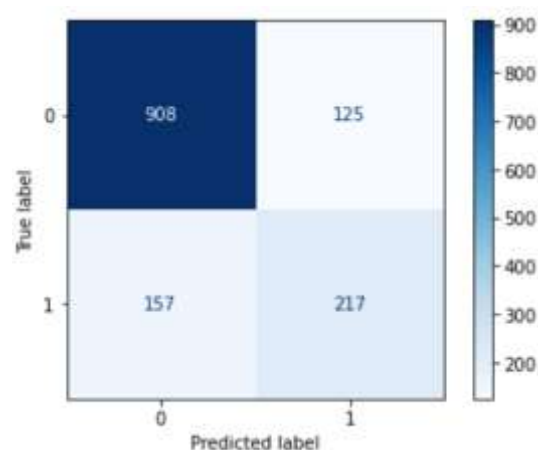


Figure 35: Confusion matrix

From the confusion matrix, the true predicted label for non-churner is 908, and for those who will not quit but it shows that they will quit is 125 so we can admit that it is acceptable.

And the number of the churner customers that are predicted to quit and they actually quit is 217 and 157 for those predicted to stay but they quit.

5.5 Logistic Regression

5.5.1. Train and test scoring:

Now we display the train score and the test score before and after features selection:

	Train score	Test score
Before feature selection	0.8103111111111111	0.8052594171997157
After feature selection	–	–

According to the result obtained by RFECV any feature selection would decrease the score of the model. So, we didn't reduce the number of features.

5.5.2. Classification report

a. before feature selection:

	precision	recall	f1-score	support
0	0.86	0.88	0.87	1033
1	0.64	0.61	0.62	374
accuracy			0.81	1407
macro avg	0.75	0.74	0.75	1407
weighted avg	0.80	0.81	0.80	1407

As it shows, the accuracy is 0.81 and the recall for non-churner customer is 0.88 and for the churner customer is 0.61.

5.5.3. Confusion matrix visualization before feature selection

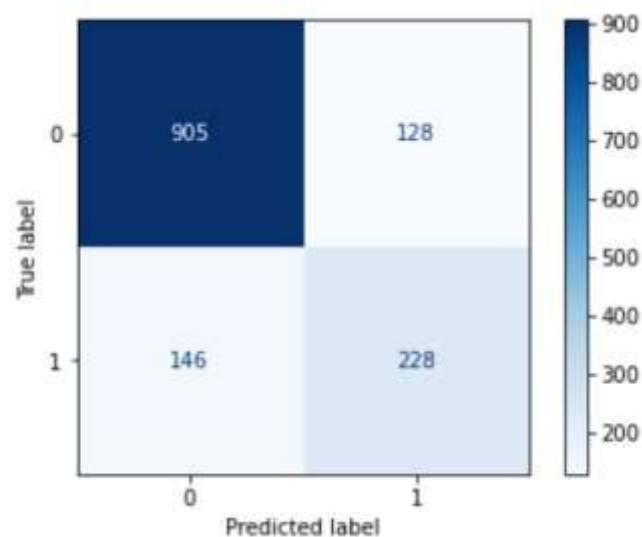


Figure 36: Confusion matrix

From the confusion matrix, the true predicted label for non-churner is 905, and for those who will not quit but it shows that they will quit is 128 so we can admit that it is acceptable.

And the number of the churner customers that are predicted to quit and they actually quit is 228 and 146 for those predicted to stay but they quit.

5.6 K-nearest neighbors (KNN)

5.6.1. Train and test scoring:

Now we display the train score and the test score before and after features selection:

	Train score	Test score
Before feature selection	0.7889777777777777	0.7725657427149965
After feature selection	0.8167111111111112	0.7810945273631841

So as we can conclude that after features selection, test score have been actually increased.

5.6.2. Classification report

a. before feature selection:

```

              precision    recall  f1-score   support

     0       0.84         0.86         0.85         1033
     1       0.58         0.53         0.56          374

 accuracy          0.77         1407
 macro avg         0.71         0.70         0.70         1407
 weighted avg      0.77         0.77         0.77         1407

```

As it shows, the accuracy is 0.77 and the recall for non-churner customer is 0.86 and for those churned customer is 0.53.

b. after feature selection:

```

              precision    recall  f1-score   support

     0       0.84         0.89         0.87         1033
     1       0.64         0.55         0.59          374

 accuracy          0.80         1407
 macro avg         0.74         0.72         0.73         1407
 weighted avg      0.79         0.80         0.79         1407

```

After the feature selection, using correlation the performance of our model has improved. We have a better train score, test score, accuracy and recall

5.6.3. Confusion matrix visualization before feature selection

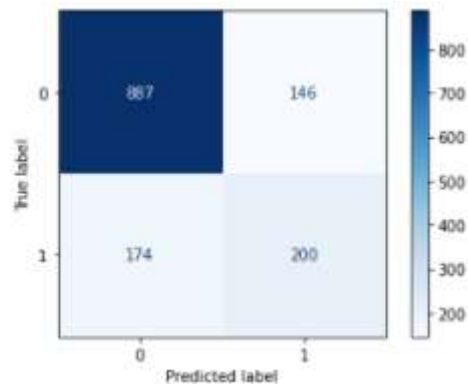


Figure 37: Confusion matrix

It can be seen in confusion matrix, the true predicted label for non-churner customer is 887, and for those who will not churn but it shows that they will quit is 146 so we can admit that it is acceptable. Now for the churner customers who said that they will quit and they actually quit are 200 and the rest 174 says that they will not churn but they did.

5.7 Tree decision

5.7.1. Train and test scoring:

Now we display the train score and the test score before and after features selection:

	Train score	Test score
Before feature selection	0.808	0.7711442786069652
After feature selection	0.8083555555555556	0.7711442786069652

So as we can constate that after features selection, test score have been actually increased.

5.7.2. Classification report

a. before feature selection:

	precision	recall	f1-score	support
0	0.86	0.83	0.84	1033
1	0.56	0.62	0.59	374
accuracy			0.77	1407
macro avg	0.71	0.72	0.72	1407
weighted avg	0.78	0.77	0.77	1407

As it shows, the accuracy is 0.77 and the recall for non-churner customer is 0.83 and for the churner customer is 0.62.

b. After feature selection:

	precision	recall	f1-score	support
0	0.86	0.82	0.84	1033
1	0.56	0.63	0.59	374
accuracy			0.77	1407
macro avg	0.71	0.72	0.72	1407
weighted avg	0.78	0.77	0.77	1407

Using the Select from Model selector we were able to reduce the number of features from 18 to 9 and the recall has a bit improved.

5.7.3. Confusion matrix visualization before feature selection

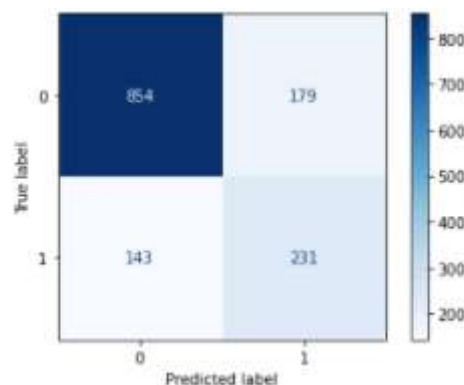


Figure 38: Confusion matrix

It can be seen in confusion matrix, the true predicted label for non-churner customer is 854, and for those who will not quit but it shows that they will quit is 179 so we can admit that it is acceptable. Now for the churner customers who said that they will quit and they actually quit are 231 and the rest 143 says that they will not churn but they did.

5.8 Random forest

5.8.1. Train and test scoring:

Now we display the train score and the test score before and after features selection:

	Train score	Test score
Before feature selection	0.8456888888888889	0.7995735607675906
After feature selection	–	–

So, as we can conclude that after features selection, test score have been actually increased.

5.8.2. Classification report

a. before feature selection:

```

precision    recall  f1-score   support

0   0.84481175  0.89060987  0.86710650    1033
1   0.64465409  0.54812834  0.59248555     374

accuracy          0.79957356    1407
macro avg  0.74473292  0.71936911  0.72979603    1407
weighted avg  0.79160709  0.79957356  0.79410847    1407

```

As it shows, the accuracy is 0.799 and the recall for non-churner customer is 0.89 and for those churned customer is 0.54.

b. After feature selection:

Based on the result obtained by RFECV we can conclude that any feature selection would decrease the performance of the model. This model is already using the best features

5.8.3. Confusion matrix visualization before feature selection

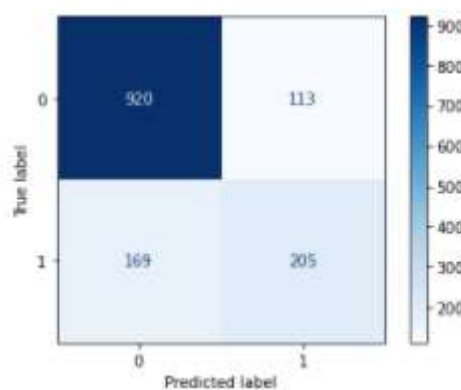


Figure 39: Confusion matrix

It can be seen in confusion matrix, the true predicted label for non-churner customer is 920, and for those who will not quit but it shows that they will quit is 113 so we can admit that it is acceptable. Now for the churner customers who said that they will quit and they actually quit are 205 and the rest 169 says that they will not churn but they did.

5.9 Naïve Bayes

5.9.1. Train and test scoring:

Now we display the train score and the test score before and after features selection:

	Train score	Test score
Before feature selection	0.7708444444444444	0.7427149964463398
After feature selection	0.7904	0.7768301350

So, as we can conclude that after features selection, train and test score have been actually increased.

5.9.2. Classification report

a. before feature selection:

```

              precision    recall  f1-score   support

     0       0.89         0.75         0.81         1033
     1       0.51         0.74         0.60          374

 accuracy          0.74         1407
 macro avg         0.70         0.74         0.71         1407
 weighted avg         0.79         0.74         0.75         1407

```

As it shows, the accuracy is 0.74 and the recall for non-churner customer is 0.75 and for churner customer is 0.74.

b. After feature selection:

	precision	recall	f1-score	support
0	0.81	0.92	0.86	1033
1	0.63	0.39	0.48	374
accuracy			0.78	1407
macro avg	0.72	0.65	0.67	1407
weighted avg	0.76	0.78	0.76	1407

After the feature selection the train score and test score, the accuracy and the recall for value 0 has improved yet the precision and the recall for value 1 has decreased too much which is problematic and affect the performance of our model so we will not use this selection of features

5.9.3. Confusion matrix visualization before feature selection

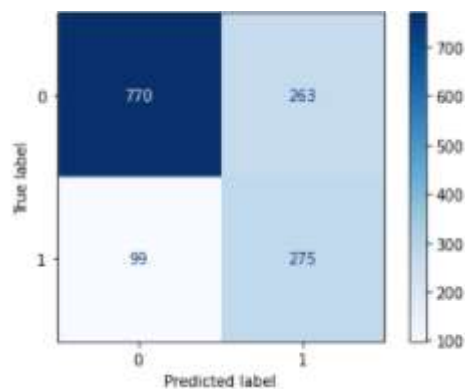


Figure 40: Confusion matrix

It can be seen in confusion matrix, the true predicted label for non-churner customer is 770, and for those who will not quit but it shows that they will quit is 263 so we can admit that it is acceptable. Now, for the churner customers who said that they will quit and they actually quit are 275 and the rest 99 says that they will not churn but they did.

5.9 SVM

5.9.1. Train and test scoring:

Now we display the train score and the test score before and after features selection:

	Train score	Test score
Before feature selection	0.7976888888888889	0.8251599147121536
After feature selection	–	–

Our model is already using the best features the feature selection won't help improving the score of the model

5.9.2. Classification report

c. before feature selection:

```
precision    recall  f1-score   support

0   0.86267281  0.90609874  0.88385269     1033
1   0.69875776  0.60160428  0.64655172      374

accuracy                0.82515991     1407
macro avg   0.78071529  0.75385151  0.76520221     1407
weighted avg 0.81910193  0.82515991  0.82077482     1407
```

As it shows, the accuracy is 0.82, the recall for non-churner customer is 0.90, and for the churner customer is 0.60. This algorithm has the best accuracy and recall comparing to others algorithms

5.9.3. Confusion matrix visualization before feature selection

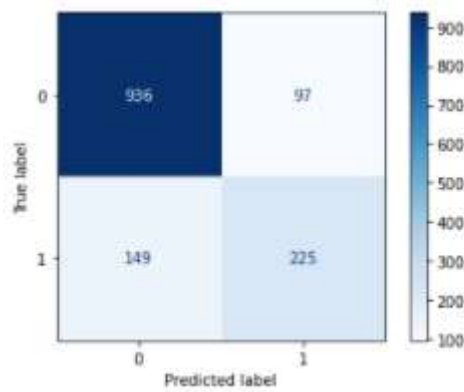


Figure 41: Confusion matrix

It can be seen in confusion matrix, the true predicted label for non-churner customer is 936, and for those who will not churn but it shows that they will quit is 97 so we can admit that it is acceptable!. Now for the churner customers who said that they will quit and they actually quit are 225 and the rest 149 says that they will not churn but they did.

5.10 Comparing the performance of the models

Two diagnostic tools that help in the interpretation of probabilistic forecast for binary (two-class) classification predictive modeling problems are ROC Curves and Precision-Recall curves. In this phase, we will use ROC Curves, Precision-Recall Curves.

- ROC Curves summarize the trade-off between the true positive rate and false positive rate for a predictive model using different probability thresholds.
- Precision-Recall curves summarize the trade-off between the true positive rate and the positive predictive value for a predictive model using different probability thresholds.
- ROC curves are appropriate when the observations are balanced between each class, whereas precision-recall curves are appropriate for imbalanced datasets.

5.10.1. Comparative table

	Algorithms	Train Score	Test Score	Recall	Precision	F1_score
0	Gradient Boost	0.825244	0.808813	0.574866	0.661538	0.615165
1	xgBoost	0.825067	0.797441	0.548128	0.638629	0.589928
2	Ada Boost	0.817422	0.799574	0.580214	0.634503	0.606145
3	Logistic Regression	0.810133	0.805259	0.609626	0.640449	0.624658
4	KNN	0.815111	0.780384	0.534759	0.597015	0.564175
5	Decision Tree	0.808356	0.771144	0.625668	0.562500	0.592405
6	Random Forest	0.845689	0.799574	0.548128	0.644654	0.592486
7	Gaussian NB	0.770844	0.742715	0.735294	0.511152	0.603070
8	SVM	0.797689	0.825160	0.601604	0.698758	0.646552

Figure 42: comparative table

Here we use the Recall, Precision and F1_score which will give us more details.

- SVM has the best test score, precision and F1_score
- Decision Tree has the best recall
- Random Forest has the best train score

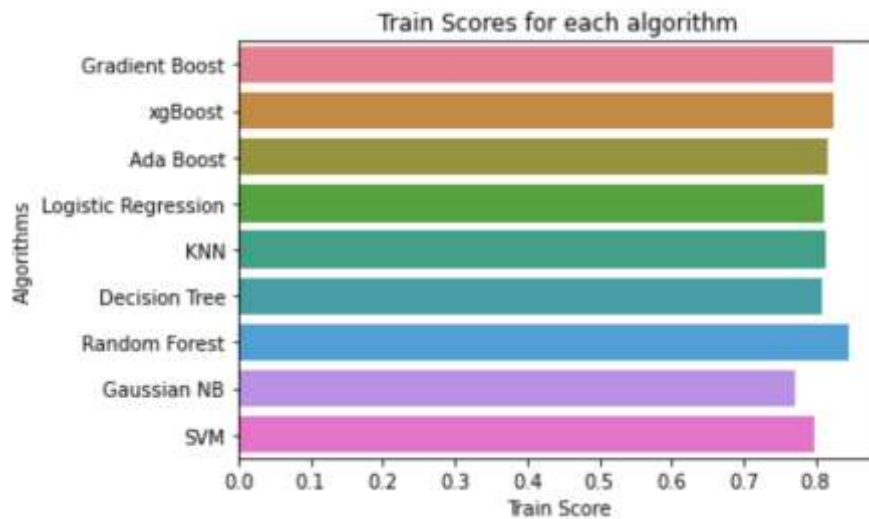


Figure 43: Train scores

We can see that the best train score is obtained using the Random Forest Algorithm and the worst train score is obtained using Naive Bayes algorithm

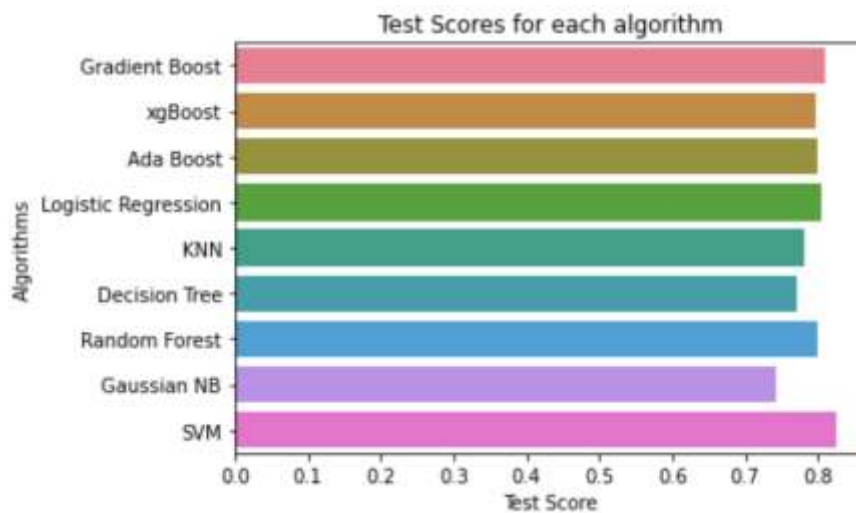


Figure 44: Test scores

We can see that the best test score is obtained using the SVM Algorithm and the worst test score is obtained using Naive Bayes algorithm

5.10.2 Roc Curves For each model

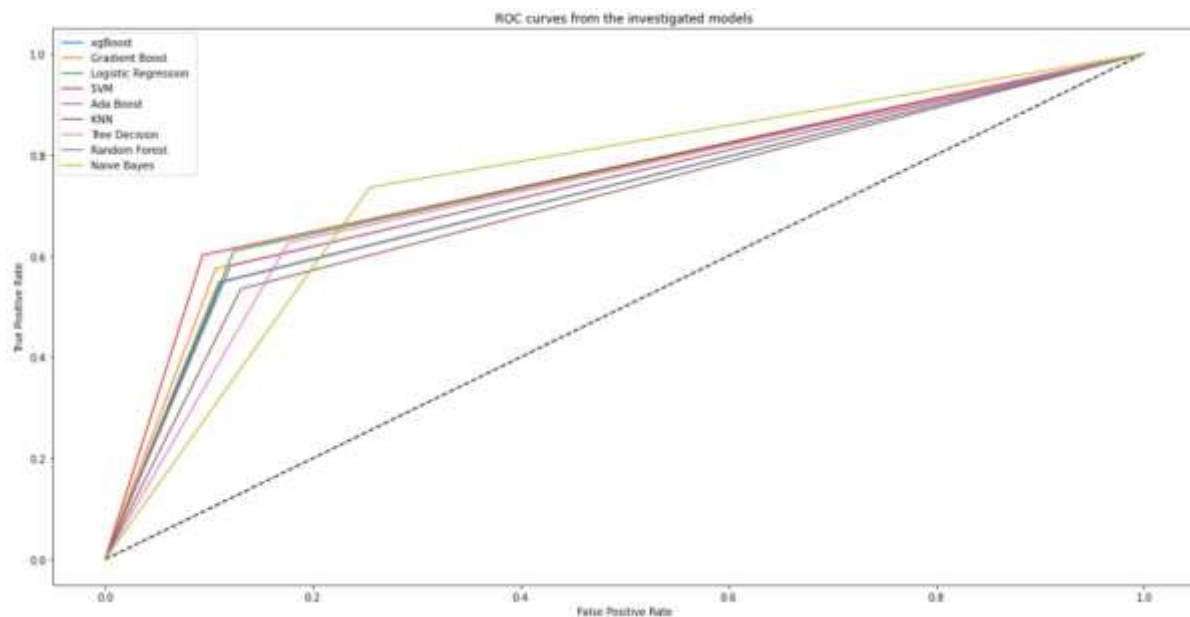


Figure 45: Roc curves for each model

SVM Area Under Curve= 0.753851509802196

Naive Bayes= 0.7403479300723192

5.10.2.1. Interpretation:

The black line represent the situation where True Positive Rate is equal to False Positive Rate . All curve above this line correspond to the situation where the proportion of correctly classified points belonging to the Positive class is greater than the proportion of incorrectly classified points belonging to the Negative class. It is evident from the plot that the Area Under Curve (AUC) for the SVM ROC curve is bigger than the rest. So, we can say that SVM did a better job of classifying the positive class in the dataset.

5.10.2.2. Precision vs Recall

Recall represents the percentage of actual churns in the data

Precision measures the percentage of churns predicted by our model that actually were churns

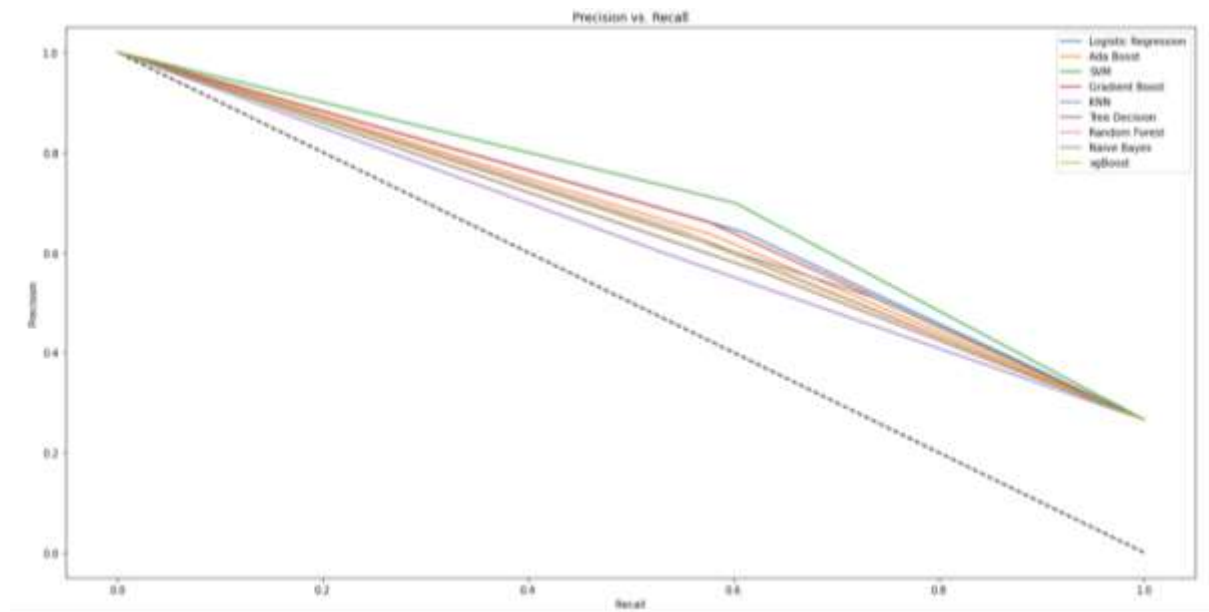


Figure 46: Roc curves for each model

Interpretation:

We can see that SVM and Naive Bayes have the same curve which represent the highest balanced recall and precision.

5.11 Conclusion:

As we exposed in the data understanding section, our target variable has 26.5% churners and 73,5% no churners. So, we are facing an imbalanced classification problem where one class is represented more than another. In such case, accuracy is not an adequate metric. Instead, we need to compare our models using recall and precision and f1_score. So, our best model is SVM because it has the best-balanced recall and precision value. More than that, it has a good accuracy and f1_score

Chapter 6 Deployment

6.1 Introduction

The last phase of a Data Science project is the deployment phase. In this chapter, we will discuss the subject of project deployment and its realization in the form of results given just after having evaluated the various stages as well as the predefined processes.

6.2 Concept and deployment strategy

Deploying machine learning models is the process of making models available in production environments, where they can provide predictions to other software systems. It is only after the model is deployed in production that they begin to add value to the customer, which makes deployment a critical step.

In our case the deployment plan will be a web application that offers the following functionalities:

- An interface that contains a form for entering customer information.
- An interface that contains the result of the model prediction

6.3 Deployment environment(tools):

The main deployment tools used in this part are:

Django: a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel. It's free and open source.

MongoDB: a distributed, universal, document-based database designed for modern application developers and for the cloud age. No other database offers such productivity.

6.4 The web application

The model will allow users to insert all the information about the customer for who they want to predict his churn. Among this information we can cite the Senior citizen, partner, dependents, tenure months, phone service, multiple lines, internet service,

online security, online backup, device protection, tech support, streaming TV, Contract, monthly charges, total charges, etc...

6.4.1 Web interface

The solution was to develop in a first version locally and it has different interfaces. During this section we will test different scenarios.

6.4.1.1 Interface of welcoming

This is the welcoming interface of our web application.



Figure 47: Welcoming interface

6.4.1.2 Interface of the customer churn prediction

The following figure shows the interface of the form to be completed with the various customer information. After submitting the form, our model created in the modeling phase will be loaded in the background and will do the necessary calculation to output as a result, a customer churn prediction value.

The figure displays two screenshots of a web application interface for customer churn prediction.

Top Screenshot: The interface has a header with "SMS Prediction" and "Home" on the left, and "View Database" on the right. Below the header, it says "Please enter details:". There are two columns of input fields. The left column includes: Senior Citizen, Partner, Dependents, Tenure Months, Phone Service, Multiple Lines, Internet Service, Online Security, Online Backup, and Device Protection. The right column includes: Tech Support, Streaming TV, Streaming Movies, Contract, Paperless Billing, Payment Method, Monthly Charges, and Total Charges. A green "Submit" button is located at the bottom right. Below the form, it says "The churn score is" followed by a blank space.

Bottom Screenshot: The interface is similar, but the input fields are now dropdown menus. The left column includes: Senior Citizen (yes), Partner (yes), Dependents (yes), Tenure Months (text box), Phone Service (yes), Multiple Lines (yes), Internet Service (DSL), Online Security (yes), Online Backup (yes), and Device Protection (yes). The right column includes: Tech Support (yes), Streaming TV (yes), Streaming Movies (yes), Contract (one year), Paperless Billing (yes), Payment Method (bank), Monthly Charges (text box), and Total Charges (text box). A green "Submit" button is located at the bottom right. Below the form, it says "This customer is" followed by a blank space.

Figure 48: Interface of the customer chum prediction

6.5 Conclusion

Deployment is the last phase of CRISP-DM, we have highlighted the models developed and specified the technologies used to put our results into production. Then we specified the technologies used and the interactions with the system in order to exploit the results of the models.

Chapter 7: Perspectives and contributions

7.1 Perspectives

Customer churn prediction models aim to indicate the customers with the highest propensity to attrite, allowing to improve the efficiency of customer retention campaigns and to reduce the costs associated with churn. Although cost reduction is their prime objective, churn prediction models are typically evaluated using statistically based performance measures, resulting in suboptimal model selection. So here some perspectives Here some perspectives:

- Having a balanced dataset: having an equal number of churning and non-churning customer.
- Adding more columns and features that are more reliable after in the process.
- Thinking about inventory ideas to engage users and new customers.
- Increase the number of subscribers on the platform.
- Marketing Action Optimization is a methodology of identifying and running the most effective marketing action for each customer. (giving extra free services...)

7.2 Academic and professional contributions

On an academic level, the project we did allowed us, among other things, to take part in the implementation of a machine learning project using an example of an existing and real database. This project allowed us to concretely use what we learned theoretically and this by applying all the learning outcomes seen in class and from our own researches. Thanks to this experience we had the chance to strengthen our knowledge in the sectors affected by the project and above all to better understand the system and the prerequisites of the work through exploration Data. It gave us the opportunity to learn new technologies and be more efficient in the field of machine learning and development Web, with the possibility of seeking and finding better solutions, always more optimal for the ambiguities encountered. This allows us to better appreciate the versatility and the interest of the engineer-manager training that we had at ESPRIT.

Professionally, the project allowed us to understand in depth the role of a data scientist and discover this huge world of data and modeling. This gave us the opportunity to gain practical experience both in the elements of business and data science.

- Reward these customers for their loyalty and reduce their dissatisfaction through the use of an automated giveaway.
- A personalized and user-friendly messages could also be sent to the customer, acknowledging the issues the customer has had while notifying them of a reward or discount on their next bill.
- Once customers have made more than 20 customer service calls during the term of their contract, a discount could be offered on their next bill to allay customer dissatisfaction and demonstrate empathy and recognition of the customer experience.

General conclusion

Today, data science is the area in which there is the most investment in the technology sector information and above all, in terms of research and development. This is why that more and more large companies are turning to this science and working in- integrate into the different departments of their areas of work, in order to provide more and more likely to be the leaders in their markets. This is due to their conscience the fact that whoever holds the information becomes the most powerful and the fact of knowing how to decipher data is an opportunity. With the current market volatility, the integration of machine learning in companies is mandatory to understand and predict the customer's opinion in order to give them what they really need. In our project show the ability to predict that a particular customer is at a high risk of churning, while there is still time to do something about it, represents a huge additional potential revenue source for every online business. Besides the direct loss of revenue, that the customer may not have already covered results from a customer abandoning the business, the costs of initially acquiring that customer is spending to date. (In other words, acquiring that customer may have actually been a losing investment.) Furthermore, it is always more difficult and expensive to acquire a new customer than it is to retain a current paying customer.

Bibliography

<https://www.optimove.com/resources/learning-center/customer-churn-prediction-and-prevention>

<https://towardsdatascience.com/predict-customer-churn-in-python-e8cd6d3aaa7>

<https://www.qualtrics.com/experience-management/customer/customer-churn/>