Telcom Customer Churn dataset analysis

Marco Cardia 530567, Luca Cosseddu 590676, Francesco Sabiu 533231

14 February 2019

Abstract

The aim of the project is to analyze demographic info about customers, services that each customer has signed up for and customer account information in order to predict customer attrition. Customer attrition refers to the loss of clients. Moreover, in this report, we are going to compare different clustering algorithm in order to find the best clustering approach. Association rules mining will be used to replace missing values and predict the target variable.

Contents

Chapter 1

Data Understanding

1.1 Data semantics

The provided dataset contains 7043 records, where each record represents a customer. Each record is composed of values of twenty possible features. It is possible to classify these features into three groups: demographic info about customers, services that each customer has signed up for and customer account information. For first we are going to describe all the features the dataset is composed of. Moreover, we will indicate, for each feature, its domain. Provided demographic info about customer are:

1. **gender:** (categorical) Whether the customer is a male or a female.

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

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

4. **SeniorCitizen:** (categorical) The attribute assumes values in the domain 0,1, but it is considered categorical since it indicates whether the customer is ancient (value 1) or not (value 0).

Customer account information includes:

1. **tenure:** (numerical) Number of months the customer has stayed with the company.

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

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

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

5. **MonthlyCharges:** (numerical) The amount charged to the customer monthly.

6. **TotalCharges:** (numerical) The total amount charged to the customer.

Services that each customer could sign up for are:

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

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

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

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

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

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

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

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

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

Finally, the target attribute is:

1. **Churn:** (categorical) Whether the customer churned or not (Yes or No).

1.2 Distribution of the variables and statistics

In this section, we are going to analyse the distribution of the variables.

The distribution of the class attribute is strongly imbalanced: most of the records have “not churn” value as class label. The proportion of the dataset is 73:27. Figure 1.1 represents the distribution of the target attribute.

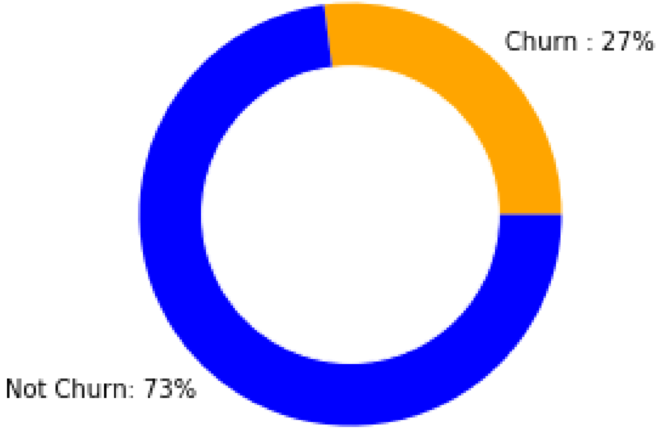


Figure 1.1: Percentage of the Churn and No Churn customers

Before building our models we will plot some variable in order to see its distribution and to get an idea about its trend. Our dataset is composed of three numerical attributes, we will start describing their distribution. As mentioned, **‘tenure’** represents the number of months a customer stayed with the company, its domain is represented by the integers between 0 and 72. It doesn’t contain any missing values or outliers. Its distribution is represented in 1.2. In the x-axis, we have bins constructed using Sturges rule. As it is possible to see it follows a bimodal distribution. Most of the customer are in in the range [0,5) and [66,73), i.e. most of the customers are new in the company or are in the company for a long time. We suppose that customers which stay in the company for a long time are less likely to leave the company. We will go deeper in the next section.

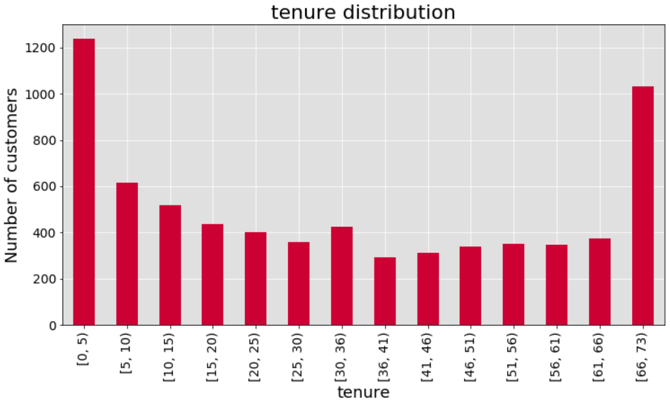


Figure 1.2: Tenure distribution

MonthyCharges attribute represents the amount charged to the customer monthly. It contains neither any outliers nor any missing values. Its distribution is represented in 1.3. It follows a bimodal distribution. In the x-axis, we have bins constructed using Sturges rule. Most of the records are in the [18,25) range, which means that a great part of the customers prefers to pay less instead of having more services.

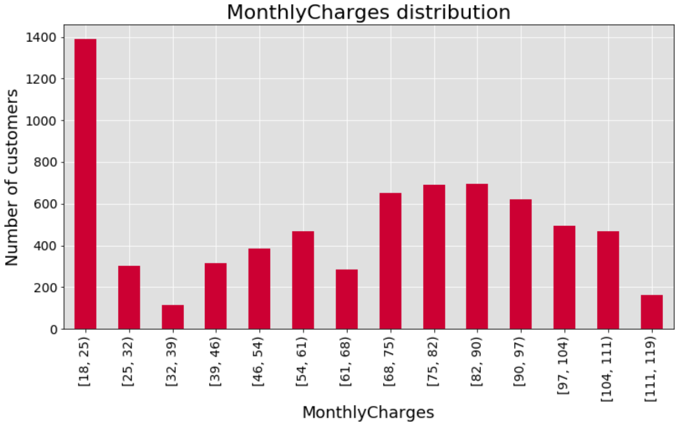


Figure 1.3: MonthlyCharges distribution

TotalCharges attribute represents the total amount charged to the customer. Initially, its data type was a string so we had to convert it into numerical. In some cases, this value corresponds to the product of tenure with MonthlyCharges. In other cases, this is not true, since a customer could change services during its subscription. We found 11 missing values. We replace these values with the product of MonthlyCharges with tenure since its correlation, using the Pearson coefficient, is 0.9995, as its possible to see in table 1.1. We marked this product as ’newTotal’. It doesn’t contain any outliers. Its distribution is represented in 1.4. It follows a negative exponential distribution. Most of the records are in the [18,637) range, values that reflect values we found in MonthlyCharges.

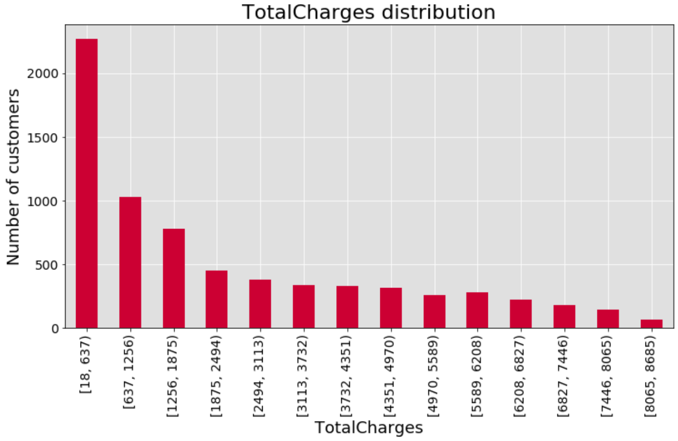


Figure 1.4: TotalCharges distribution

Now we have a look at some categorical attribute. A customer can subscribe to either PhoneService, InternetService or both of them. Figure 1.5 and 1.6 shows the distribution of respectively PhoneService and InternetService. The first one includes 90% of the customers, the second one includes 78% subscribers. There are no customers with neither PhoneService nor InternetService.

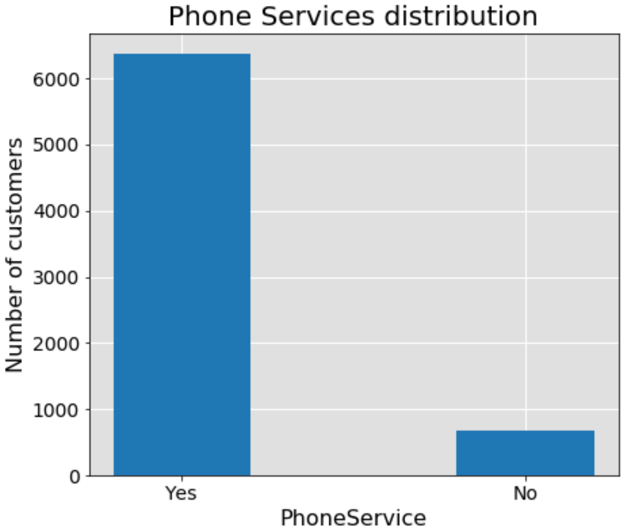


Figure 1.5: PhoneService distribution

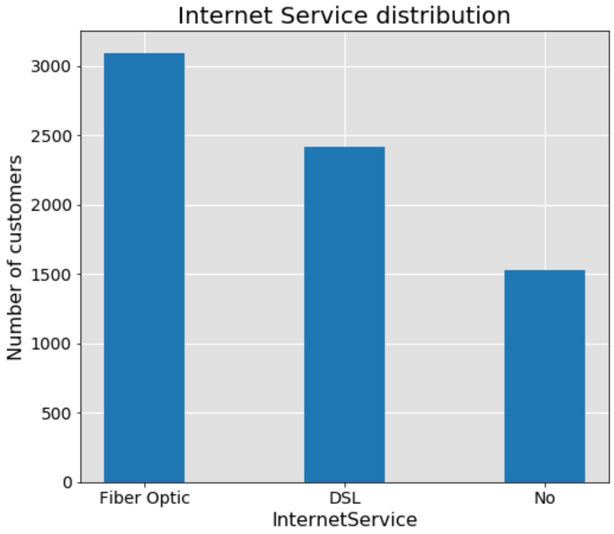


Figure 1.6: InternetService distribution

There is one aid related to Phone Service, i.e. MultipleLines. Its distribution is shown in Figure 1.7. On the other hand, internet service has 6 aids related to its attribute. These six treatments are displayed in Figure 1.8. Most of InternetService customers have, as aids, StreamingTv and StreamingMovies.

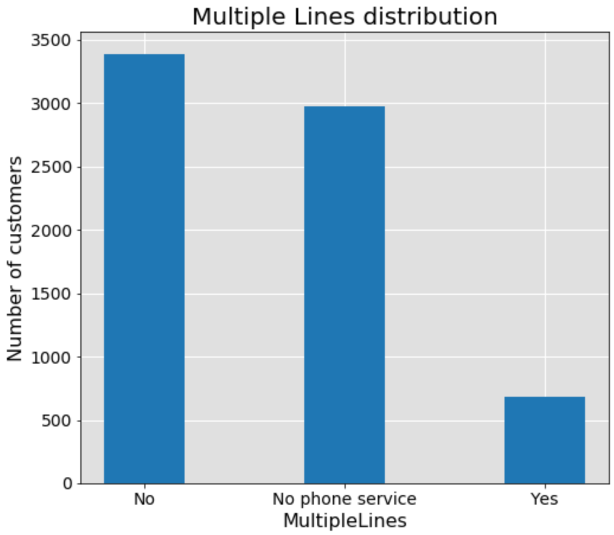


Figure 1.7: MultipleLines distribution

	Tenure	Monthly Charges	Total Charges	Service Count	weightedTotal
Tenure	1	0.2479	0.8258	0.4942	0.8265
Monthly Charges	0.2479	1	0.6510	0.7247	0.6515
Total Charges	0.8258	0.6510	1	0.7461	0.9995
Service Count	0.4942	0.7247	0.7461	1	0.7449
weightedTotal	0.8265	0.6515	0.9995	0.7449	1

Table 1.1: Pearson correlation matrix

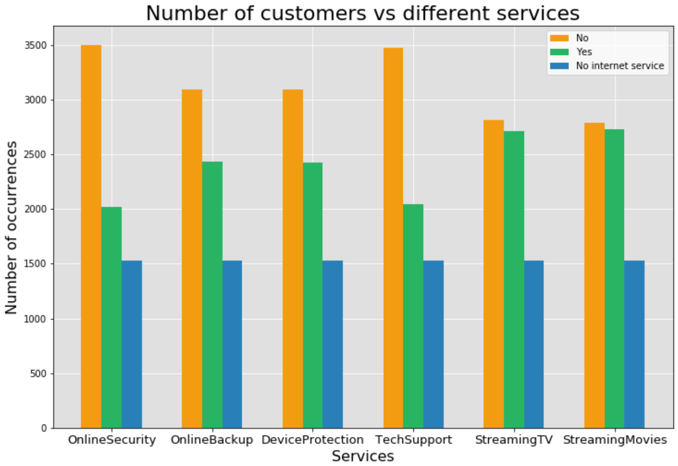


Figure 1.8: Services distribution

The last attribute we are interested in is Contracts, that shows which kind of contract a customer has subscribed for. Its distribution is shown in figure 1.9.

1.3 Data quality and missing values

In this section, we are going to discuss data quality and missing values.

First, we studied syntactic accuracy, we found some entries that were not in the domain. In particular, we found Senior Citizen column with numerical values even if its values are categorical (Yes and No). Moreover, TotalCharges were found as string values, even if its values where numerical. We replace its type as float value.

There are 11 missing values on Total Charges attribute. These values were replaced with the product of MonthlyCharges with tenure since its correlation, using the Pearson coefficient, is 0.99, as its possible to observe in table 1.1. No other missing values were found. Concerning outliers, boxplot of numerical attributes is shown in figure 1.10. No outliers were found in these numerical attributes.

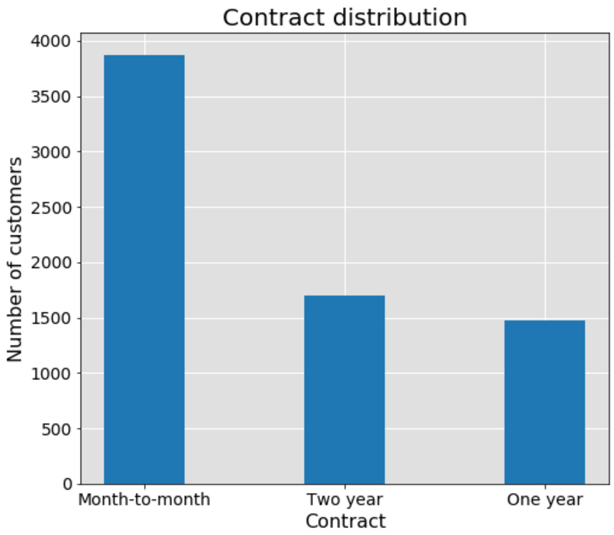


Figure 1.9: Contract distribution

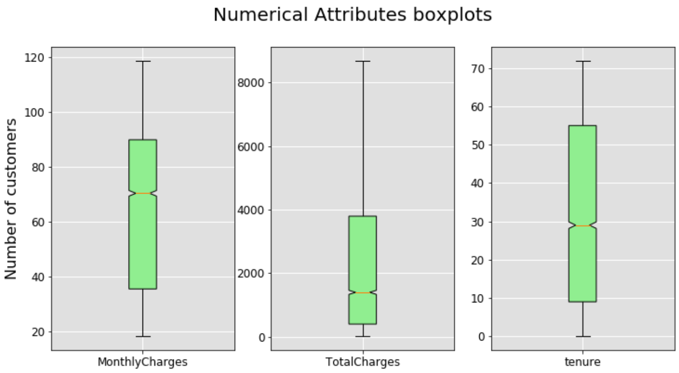


Figure 1.10: Boxplot with numerical attributes

1.4 Variable Transformation

Phone Service and Multiple lines were semantically redundant: we dropped Multiple Lines and changed Phone Service values into 'Single Line', 'Multiple Lines' and 'No phone service'.
We created a new attribute, 'ServiceCount', that refers to

the number of Internet Service facilities a customer subscribed for. This attribute summarize all the services related to InternetService, so we dropped all the categorical fields related to InternetService, i.e. OnlineSecurity, OnlineBackup, DeviceProtection, TechSupport, StreamingTv, StreamingMovies. Distribution of this feature can be found in figure 1.11. Furthermore, we found some useful information by correlating this attribute with other features. These explanations will be discussed in section AttributeCorrelation.

We also dropped gender and partner attributes since their distributions were balanced, hence they don't give relevant information.

TotalCharges attribute is redundant since its correlation with the product of MonthlyCharges and tenure is higher than 0.99; hence we dropped also this feature.

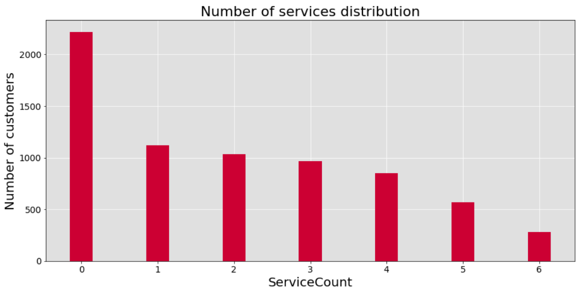


Figure 1.11: Service count distribution

1.5 Attributes Correlation

In this section, we are going to analyze the pairwise correlation of attributes. We found very interesting information using 'ServiceCount' feature, so we are going to examine the correlation between the number of services with other attributes. Figure 1.12 shows the number of services with respect to the number of churns. In this plot, we can find that customers who don't avail any internet services are churning least in proportion. While customers who are availing just one Internet treatment are churning highest. As the number of online services increases beyond one service, the less is the proportion of churn.

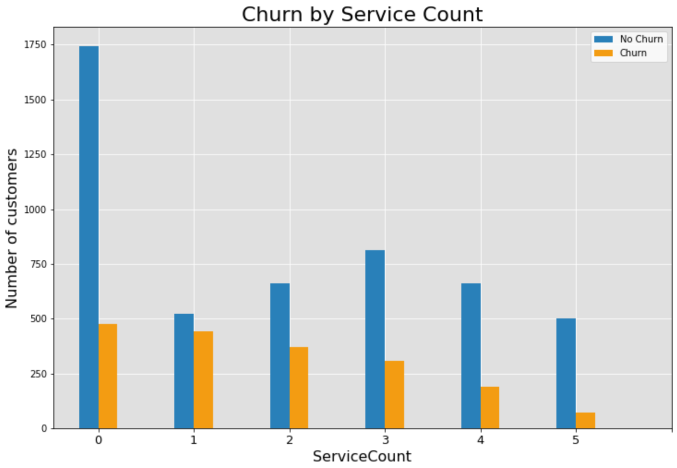


Figure 1.12: Comparison between churns and service count

Going deeper into the analysis of 'ServiceCount' feature we can find that customers who do not avail any internet service are paying just 33€ in average, while those with one service are paying double, i.e. 66€ in average. As the number of services availed increases, the Average Monthly Charges are increasing linearly.These information are displayed in figure 1.13

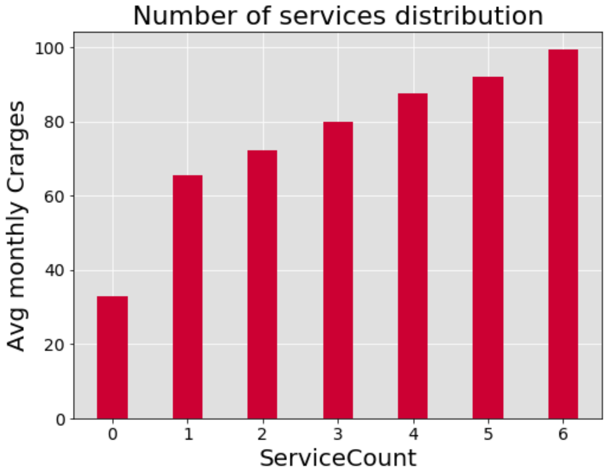


Figure 1.13: Comparison between the average of Monthly charge and service count

Higher the 'MonthlyCharges', more is the possibility of Churn, non-churners are paying just over 60€, while churners are paying nearly 80€. Boxplot in figure 1.14 shows the distribution of 'MonthyCharges' for churners and non-churners.

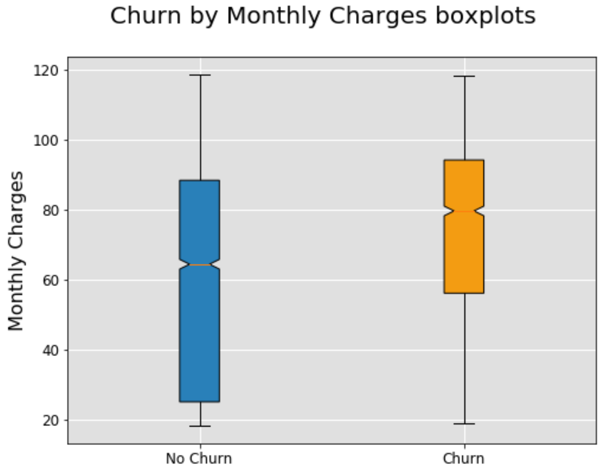


Figure 1.14: MonthyCharges distribution by churn

Customers with Month-to-Month contract are churning more, while two-year contract customers are churning least. One possible reason could refer to the contractual link between the two parts. We can observe this fact in figure 1.15.

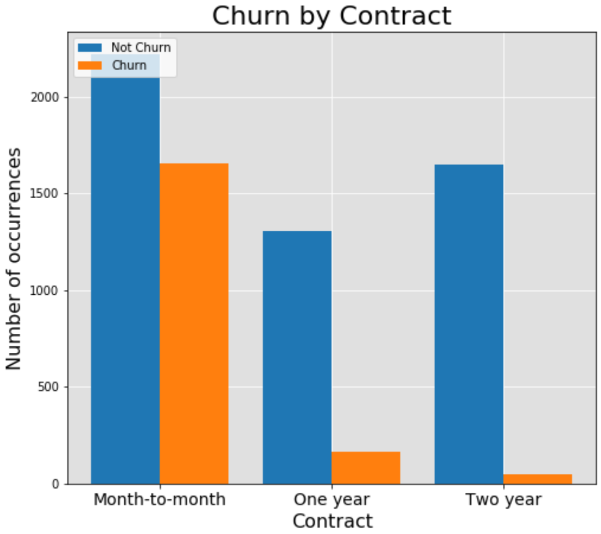


Figure 1.15: Churn by Contract

Customers with Electronic check as payment method are churning more, while others payment methods are churning less, as it is possible to see in 1.16

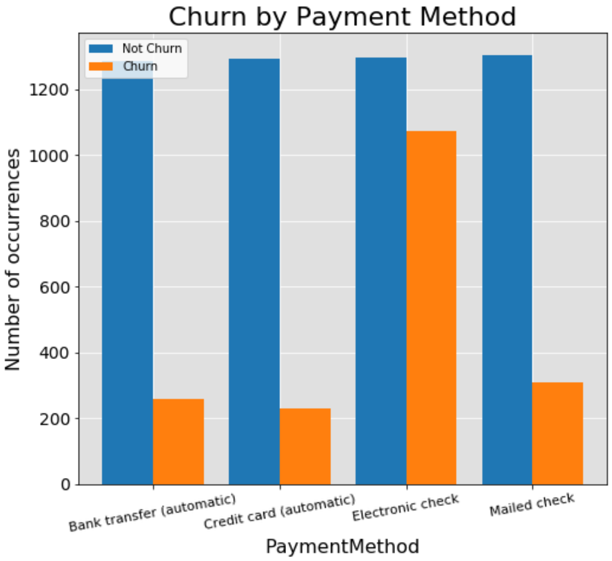


Figure 1.16: Churn by PaymentMethod

Surprisingly, customers with MultipleLines are churning in higher proportion.

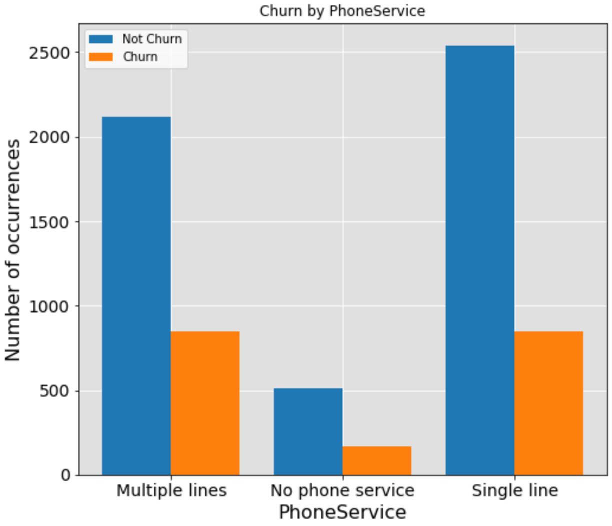


Figure 1.17: Churn by PhoneService

We found very interesting the relationship between tenure and churn. As we expect the number of churn clients decreased as the value of tenure increased. Most of the customers with a tenure value of 71 and 72 don't churn. Figure 1.18 shows the distribution of tenure with respect to churner customers and non-churn customers.

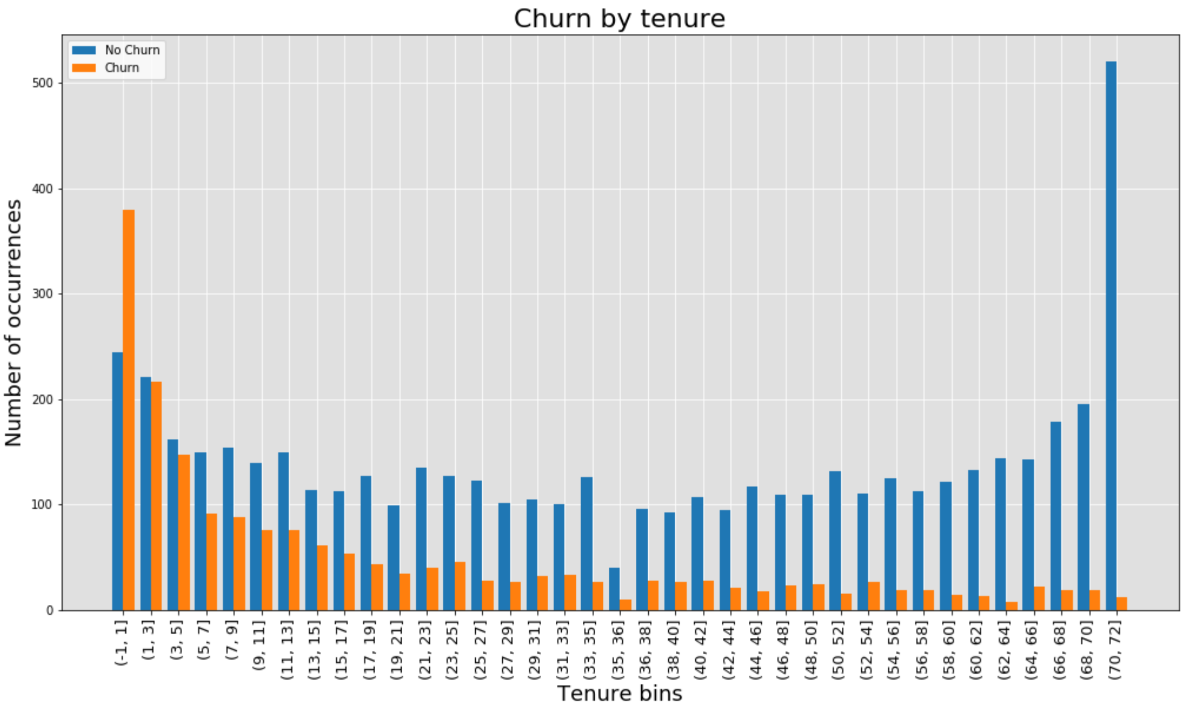


Figure 1.18: Churn By Tenure

	Senior Citizen	Dependents	Internet Service	Contract	Paperless Billing	Payment Method	Phone Service	Tenure	Monthly Charges	Service Count
0	No	No	Fiber optic	Month-to-month	Yes	Electronic check	Multiple lines	24	88	2
1	No	No	No	Two year	No	Credit card	Single line	54	27	0
2	No	No	No	Month-to-month	No	Mailed check	Single line	11	25	0
3	No	No	DSL	Month-to-month	Yes	Electronic check	Single line	15	57	1
4	No	No	Fiber optic	Two year	Yes	Credit card	Multiple lines	61	90	4

Table 1.2: Clustering centroids

Chapter 2

Clustering

Clustering has been used to group sets of related customers sharing common characteristics in order to find unexpected correlations and to discover interesting features that have gone unnoticed in data understanding. We mainly performed K-Prototypes, i.e. an algorithm that integrates the k-means and k-modes algorithms to allow for clustering objects described by mixed numeric and categorical attributes; DBSCAN and Hierarchical clustering.

2.1 K-prototypes

In our project, we decided to perform K-prototypes algorithm instead of k-means. This is due since our dataset is composed of categorical and numerical attributes. We also applied k-modes to our dataset, after discretization of numerical attributes, but we got worse performance in terms of cost and silhouette as shown in table 2.1.

	K-Modes	K-Prototypes
Intracluster distance	28300	2900
Silhouette	0.08	0.26

Table 2.1: Silhouette and cost comparing

In order to perform clustering, we chose to use 10 attributes: SeniorCitizen, Dependents, InternetService, Contract, PaperlessBilling, PaymentMethod, PhoneService, tenure, MonthlyCharges, ServiceCount. In order to compute the distance among numerical attributes, K-prototypes performs euclidean distance. On the other hand to perform distances among categorical attributes, it will perform matching similarity. This distance is defined as:

$$D_1(X,Y) = \sum_{j=1}^{m} \delta(x_j, y_j)$$

where

$$\delta(x_j, y_j) = \begin{cases} 0 & \text{if } \mathrm{x_j} = \mathrm{y_j} \\ 1 & \text{if } \mathrm{x_j} \neq \mathrm{y_j} \end{cases}$$

The init value is the one defined by Cao in citeCAO. We get good results by using a value of $n\_init = 10$ The number of iteration the algorithm takes to converge is lower than 20 iterations, so even a value of $max\_iter = 20$ would be enough. We observed this fact by using the 'verbose' parameter.

We ran the algorithm for values of k from 2 to 30. Then we plot a line graph with the relationship between the k value and the cost, defined as the sum distance of all points to their respective cluster centroids. This plot is shown in figure 2.1. In order to choose the best value of K, we wanted to minimize intra-cluster and maximize inter-cluster distances respectively with cost and silhouettes scores (Its line graph is shown in figure 2.2). So we consider either the location of a bend (knee) in the plot in figure 2.1 and the highest silhouettes score (shown in figure 2.2) close to the 'knee'.

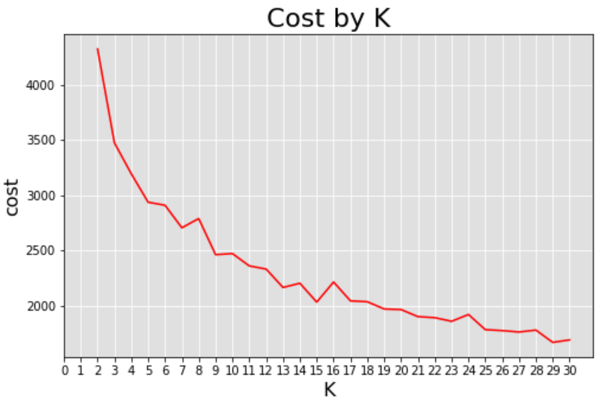


Figure 2.1: Cost by k

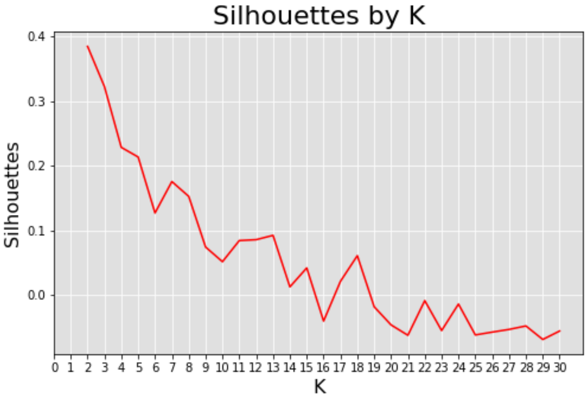


Figure 2.2: Silhouette by k

The line graph in figure 2.1 describes the variation of the cost with respect to K: the cost decreases as k increases, it is possible to see a bend (or a "knee") at k=5. This bend indicates that addition clusters beyond the 5th have a low value, i.e. its improvement will decline. In that point, we have also good value of silhouette, i.e. 0.21.

Table 1.2 shows the clustering centroids. Moreover 2.3 and 2.4 shows distributions of centroids respectively in numerical attribute and categorical. By observing both table and figures we can notice that centroids are not influenced by attributes like Senior Citizen and dependents. While clusters are strongly influenced by attributes like Monthly Charges.

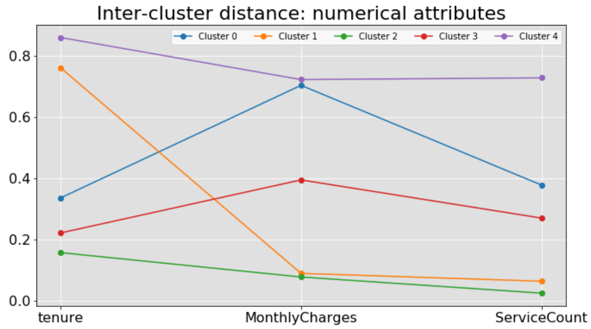


Figure 2.3: Centroids numerical values plot

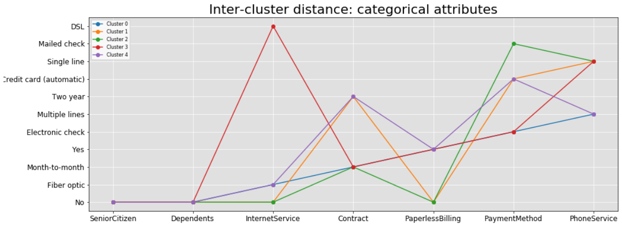


Figure 2.4: Centroids categorical values plot

Clustering divided the dataset into 5 parts:

- Centroid 0 suggests that cluster 0 is composed of customers with a high value of Monthly Charges, a value of tenure of 24 on average, and a value of service count of 2 on average (customers who stay in the company for a discrete amount of time and have a very high monthly bill with a small number of services). Concerning categorical attribute, are mainly composed of customers with a Fiber Optic as internet service and multiple lines as phone service, this probably influences the high value of Monthly Charges. Other characteristics of this cluster are, contract month-to-month, paperless billing and electronic check as payment method.

- Centroid 1 hints that cluster 1 is composed of a low value of Monthly Charges, a high value of tenure and without additional internet services (customers who stay in the company for a long time, who have a small number of monthly charges and without internet services). These customers have a single line as phone service, a contract of two years and they pay by credit card.

- Centroid 2 suggest that cluster 2 is composed of new customers (low tenure), small monthly bill and without internet service. These customers have a Month-to-month contract and as payment method, they have a mailed check.

- Centroid 3 hints that cluster 3 is composed of new customers (low tenure), a discrete monthly bill and a small number of service count. These customers have a DSL as internet service, single line phone service, Month-to-month contract, paperless billing and electronic check as payment method.

- Centroid 4 suggest that cluster 4 is composed by of senior customers (high tenure), a high number of monthly charges and a high number of additional services. These clients have fiber optic as internet service multiple lines as phone service, a two years contract, paperless billing and credit card as payment method.

Table 1.2 gives us some useful information like clients who have a high value of monthly charges have fiber optic as internet service, multiple lines as phone service and, often, a high number of additional internet service. Senior clients have two years contract and pay by credit card.

Before analyzing the distribution of the attributes in the different clusters, we first have a look at the distribution of the entries in the clusters with respect to the target, churn. This distribution is shown in table 2.2

Churn\Cluster	0	1	2	3	4
No	910	797	926	1080	1461
Yes	890	19	192	603	165

Table 2.2: Clustering centroids with respect to Churn class

Considering the distribution of target attribute and considerations we did by describing centroids, we will report most interesting clusters, that are cluster 0, cluster 3 and cluster 4.

In the first scatter plot, figure 2.5, we represent in the x-axis Monthly Charges, in the y-axis the Total Charges. The target value for each cluster is included. We marked with 'x' all the churning customers. We can easily observe that the churning elements are denser in the bottom right side of the image. It means that there are more churners for higher Monthly charges values and a medium-low value of TotalCharges. This area is represented by Cluster 0, that is the cluster containing the highest number of churners.

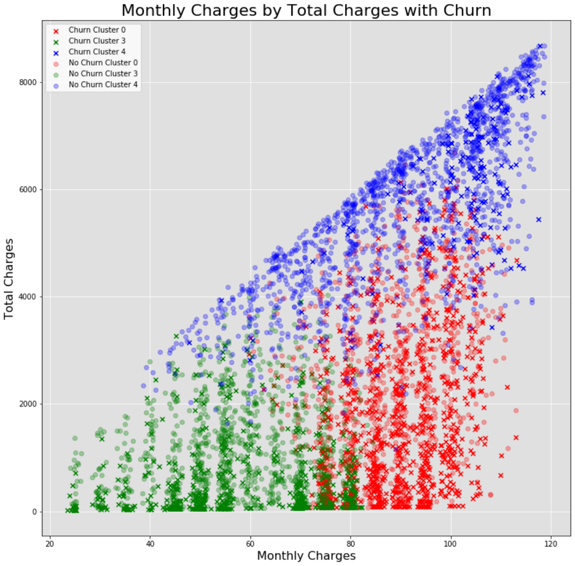


Figure 2.5: Distribution of Monthly Charges and Total Charges in Clusters 0, 3 and 4

Another view of the cluster can be represented as follows: in the Cartesian axis tenure and monthly charges are represented (Figure 2.6). Here, churning users are concentrated on the top-left area of the plane. This means that, for the selected clusters, a small tenure value is more frequent for churning users. Again, the top left corner is composed by customers belonging to clusters 0.

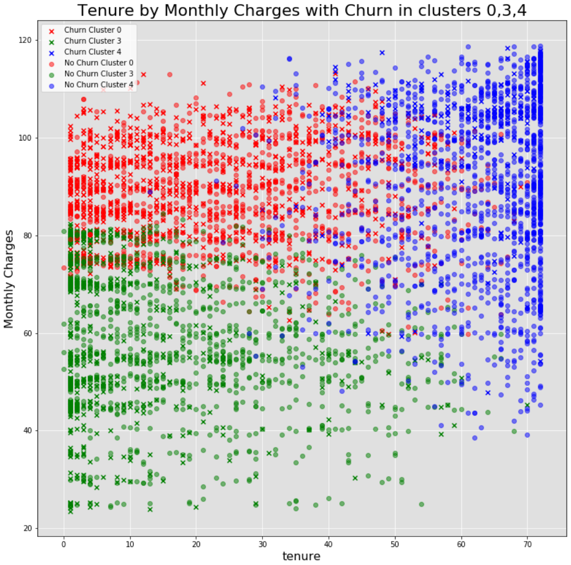


Figure 2.6: Distribution of Monthly Charges and Total Charges in Clusters 0, 3 and 4

2.2 DBscan

We applied DBscan in order to find some interesting pattern based on the density areas. We used the same attributes as K-Prototypes. Firstly, as preprocessing phase, we normalized numerical attributes. Secondly, we needed to define a distance function in order to compute distances between records having both categorical and numerical attribute. We used a custom function defined as:

$$dist(X,Y) = \frac{n_{num}}{n_{num}+n_{cat}} euclidean(X[num],Y[num]) +$$

$$\frac{n_{cat}}{n_{num}+n_{cat}} hamming(X[cat],Y[cat])$$

where n_{num} is the number or numerical attributes and n_{cat} is the number of categorical attributes.

In order to obtain the best clustering, we need to tune some values of MinPoints and *epsilon*. We will start with a consideration. Since we applied DBscan to a dataset composed of 10 attributes, i.e. the dimension of our space is 10, we need to run it with a number of MinPoints value of at least 11. In order to choose the epsilon value, we plotted some line graph on different value of MinPoints. We take into account values of MinPoints of 11, 13 15 and 18.

In order to choose the best value of *epsilon* we plotted in figure 2.7 the sorted distances from the MinPoints-th point.

From the plot we can see that the curves have an elbow for values of sorted distances close to 6000, that correspond to an epsilon value of 0.15.

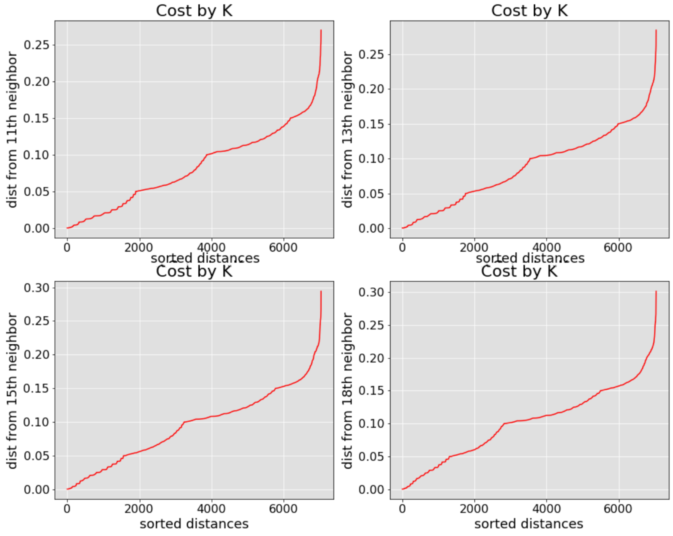


Figure 2.7: Sorted distances from the MinPointh-th point fro values of minpoints of 11, 13, 15, 18

We obtained good results considering k=13 and epsilon = 0.15. In particular, as shown in table 2.3, we got 3 clusters plus the noise points (labelled as -1). Cluster 0 is composed of 2177 customers, cluster 1 has 2992 customers, cluster 2 has 1521 customers in total. Distribution of churners over these clusters is shown in table 2.3.

Churn\Label	-1	0	1	2
No	313	1742	1711	1408
Yes	40	435	1281	113

Table 2.3: Distribution of churnes over clusters

As it is possible to see in figure 2.8 and figure 2.9 clusters are strongly influenced by MonthlyCharges attribute even if we used as distance measure a function that takes into account both numerical and categorical attribute.

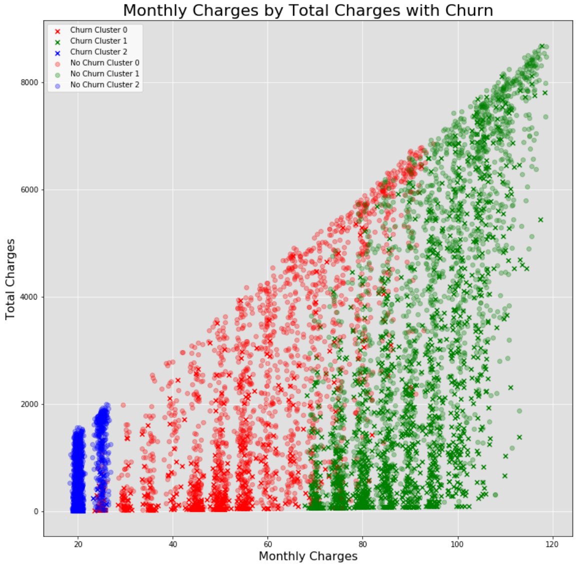


Figure 2.8: Distribution of Monthly Charges and Total Charges in Clusters 0, 1 and 2 generated by DBscan

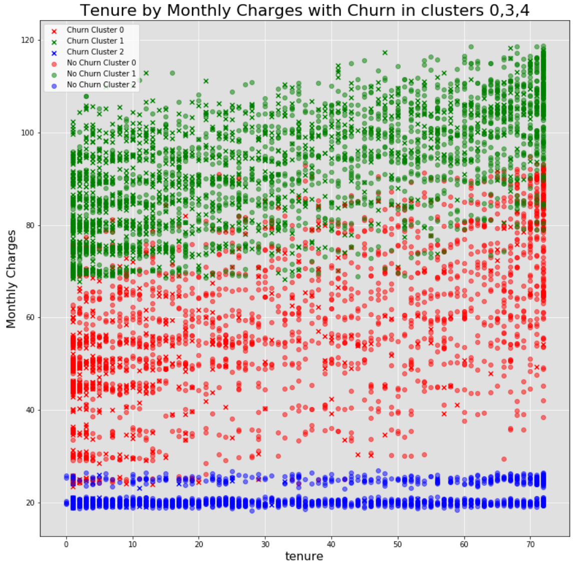
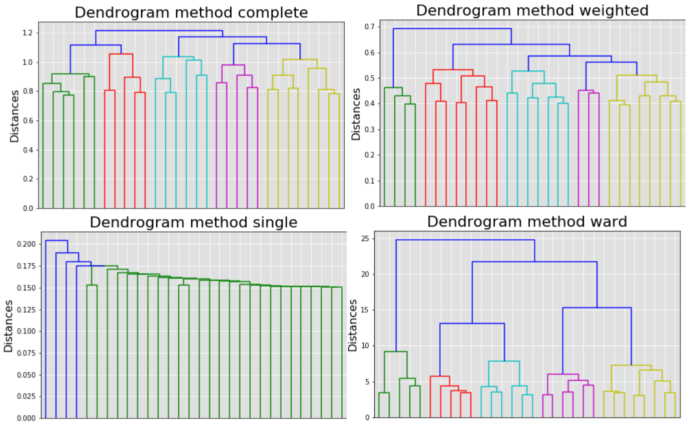


Figure 2.9: Distribution of Monthly Charges and Total Charges in Clusters 0, 1 and 2 generated by DBscan

2.3 Hierarchical clustering

Concerning hierarchical clustering, we used the same ten attributes as we did in K-prototypes algorithm and DB-scan. As metric we used the same as DB-scan for the same reason we explained above.

We tune different values of connection criteria parameter, i.e. complete, single, ward, average and weighted, in order to perform hierarchical clustering. In figure 2.10 and 2.11 are shown results of hierarchical clustering, where the cutting heights were applied in order to obtain 5 clusters.


Dendrogram method complete

Dendrogram method weighted

Dendrogram method single

Dendrogram method ward

Figure 2.10: Dendrogram for the hierarchical clustering with different linkage methods

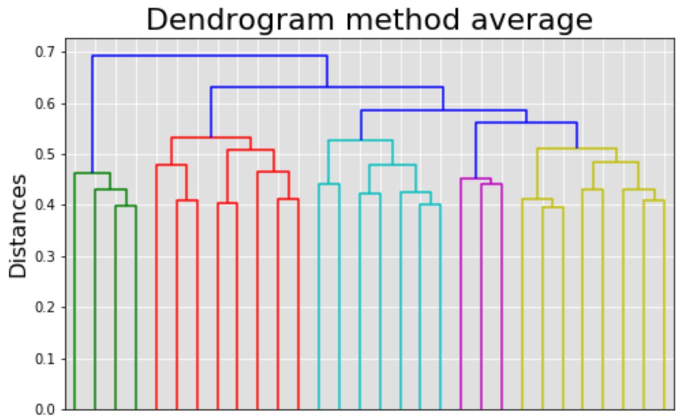

Dendrogram method average

Figure 2.11: Dendrogram for the hierarchical clustering with average linkage method

Table 2.4 shows the distribution of customers in the different clusters with the different connection criteria.

Method	Composition of clusters
Complete	[1959 1544 1495 1247 798]
Single	[7039 1 1 1 1]
Weighted	[3087 1526 1083 757 590]
Average	[3335 1526 1157 940 85]
Ward	[2124 1526 1269 1152 972]

Table 2.4: Composition of cluster for hierarchical clustering algorithm

As it is possible to see in the table, in the clustering generated by the single method approach, the first cluster contains almost all of the customers, while the others contain only one customer each. The others are pretty good clusters, in terms of distribution of customers.

Figures 2.10 and 2.11 show the different dendrogram generated with different linkage methods. As it is possible to observe, clusters using single linkage joined together at lower distances with respect to complete, weighted and average linkage. This is due to the fact that single linkage uses the minimum distance between each couple of points of two clusters.

Single link method produces the worst clusterization. According to dendrograms other methods produces a balanced distribution over the customers.

2.4 Final evaluation

From the descriptions of different clustering algorithms, we can conclude that we get pretty good clusters by using K-Prototype algorithm and Hierarchical with weighted linkage. Table 2.5 contains the silhouette score of the different clustering algorithms. K-Prototype creates clustering balanced in terms of the number of customers per cluster. DBscan generated three balanced clusters, but a low silhouette score. Hierarchical with complete linkage creates balanced clusters in terms of customers, but with a low silhouette score. Hierarchical with single linkage generates unbalanced clusters and a negative silhouette score. Hierarchical with average and ward linkage creates balanced clusters, but have a small value of silhouette. Best clustering algorithm in terms of silhouette is hierarchical with weighted linkage.

Clustering	Silhouette
K-Prototrypes	0.2633
DBscan	0.1199
Hierarchical with complete linkage	0.1689
Hierarchical with single linkage	-0.2070
Hierarchical with weighted linkage	0.3039
Hierarchical with average linkage	0.1720
Hierarchical with ward linkage	0.2091

Table 2.5: Comparison among different clustering algorithms

We can conclude that the best clustering algorithms for this dataset are K means and Hierarchical with weighted linkage.

Chapter 3

Association Rules

For first in order to extract association rules, we have to transform our dataset in a transactional dataset. Numerical attributes were discretized. In a transactional dataset every item is an attribute value of the original dataset.

We tried to extract frequent itemsets for different values of min support.

- min_support = 0.15 produced 257 itemsets, whose a half has length ≤ 2 and support < 0.2. While the 75% has length ≤ 3 and support < 0.25. Highest support is the itemset composed of the only PhoneService=yes with support 0.9.
- min_support = 0.2 produced 114 itemsets, whose a half has length ≤ 2 and support ≤ 0.25, the 75% has length ≤ 3 and support ≤ 0.3.
- min_support = 0.25 produced 56 itemset, whose a half has length ≤ 2 and support ≤ 0.3. The 75% has length ≤ 3 and support ≤ 0.42.

Table 3.1 shows the firsts frequent itemsets with length > 1.

Support	Itemsets
0.661934	(No Churn, PhoneService)
0.537271	(PaperlessBilling, PhoneService)
0.496805	(Month-to-month, PhoneService)
0.454920	(Male, PhoneService)
0.448246	(Female, PhoneService)
0.439585	(Fiber optic, PhoneService)
0.438875	(PhoneService, Partner)
0.421837	(MultipleLines, PhoneService)
0.393440	(PaperlessBilling, No Churn)
0.388045	(No Churn, Partner)

Table 3.1: Frequent itemsets with length > 1

In firsts ten itemsets, three of them contains No Churn. In the firsts 70 itemsets with a min_support = 0.2 we have about one half of them containing NoChurn. The first frequent itemset that contains Churn is (PhoneService, Churn) with support of 0.24, but this is a trivial-one. While one more interesting frequent itemset could be (Month-to-Month, Churn) with support of 0.23.

3.1 Association rules extraction with different values of confidence

We made some tests for different values of min_support and min_confidence. Rules with the highest confidence involved PhoneService, but this value is trivial since his support is too high and it doesn't distinguish a customer category from one other. We are more interested in association rules, involving Churn and NoChurn. Rules involving NoChurn are shown in table 3.4. In order to obtain these rules we used a min_support of 0.2 and a min_confidence of 0.7.

On the other hand, to obtain association rules which include Churn=Yes, we have to reduce min_support to 0.1 and the confidence to 0.5. In this way we obtain values in table 3.3. Significant rules that don't involve Churn are shown in table 3.2.

Antecedents	Consequents	Support	Confidence
Dependents	Partner	24.83%	82.89%
StreamingMovies	StreamingTV	27.55%	71.01%
MultipleLines	Fiber Optic	27.51%	65.23%

Table 3.2: Most significant rules not involving churn

3.2 Discussion of the most interesting rules

Most interesting attributes in association rules are Contract, with values Month-to-Month and TwoYears, and InternetService, with values NoInternetService and Fiber optic. Table 3.3 reports the most interesting rules involving

Churn=yes item. Unfortunately, accuracy for these values is lower than 60%.

Antecedents	Consequents	Accuracy	Support
(Fiber optic, Month-to-month, PaperlessBilling)	Churn	58%	14%
(Fiber optic, Month-to-month)	Churn	53%	16%
(Fiber optic, PhoneService, Month-to-month)	Churn	53%	16%

Table 3.3: Most interesting rules churn label

We obtained better results for No Churn prediction. Table 3.4 shows most interesting rules involving Churn=No.

3.3 Association rules to replace missing values

The dataset doesn't contain missing values related to the most interesting rules, we only have missing values in Total Charges attribute, so we decide to simulate the substitution, in a way so that we can evaluate the accuracy of some rules. Table 3.5 reports some rules that can be used to replace missing values.

Accuracy for predicting low values of TotalCharges and Contract=month-to-month is pretty good.

3.4 Association rules to predict the target variable

We already showed some rules to predict the target variable Churn=Yes in table 3.3. These values have an accuracy greater than 50% but a small value of support. Some other rules will be shown in table 3.6. These results have smaller values of both support and accuracy.

As well we showed some rules to predict NoChurn customers in table 3.4. These values have excellent accuracy, but small support. However, these are good rules to predict No Churners.

Antecedents	Consequents	Accuracy	Support
Two Years	No Churn	97.13%	23.38%
PhoneService, TwoYears	No Churn	97.06%	21.14%
No InternetService, PhoneService	No Churn	92.57%	20.06%
Dependents, Partner	No Churn	85.78%	21.3%
OnlineSecurity	No Churn	85.54%	24.48%

Table 3.4: Most interesting rules with no churn as consequence

Antecedents	Consequents	Support	Accuracy	Lift
(Month-to-month, PhoneService, tenure=[0, 5))	(TotalCharges=[18, 638))	15.13%	100%	3.098
(Month-to-month, PaperlessBilling, MultipleLines)	(Fiber optic)	14.41%	87.57%	1.992
(PaperlessBilling, PhoneService, StreamingMovies,StreamingTV)	(Fiber optic, MultipleLines)	11.10%	59.69%	2.169
(tenure=[0, 5), TotalCharges=[18, 638))	(Month-to-month)	17.48%	96.66%	1.757

Table 3.5: Meaningful rules to replace missing values

Antecedents	Consequents	Support	Accuracy	Lift
(Month-to-month, TotalCharges=[18, 638))	Churn	12.63%	47.39%	1.785
(Fiber optic, PaperlessBilling, PhoneService)	Churn	15.16%	44.59%	1.68

Table 3.6: More rules with churning customers

Chapter 4

Classification

The purpose of the classification we are going to illustrate is to predict whether a customer will churn or not. The adopted classification model is the Decision Tree (DT). The dataset is pretty unbalanced towards 'no churn' values, which represent the 73% of the total. For this reason, after the choice of the prediction model, we trained it with different dataset balances in order to compare the results of the model applications.
First, we split the given dataset into two subsets: one dedicated to the training, composed of the 80% of the starting dataset, and one with the rest of the entries (20%).
In order to get better performances in the prediction, we decided to undersample those records with a "no churn" label, until reaching a 50:50 proportion between the labels. "Random sampling" technique has been used to balance data. After that, the training set contained 2990 entries and it has been split one more time into train and validation sets: the validation one has been used to ensure that the model didn't overfit.
Then, the results of both undersampled and original datasets have been compared. In the following sections the parameter choice and the results comparison will be described and illustrated with respect to the above mentioned training sets.

4.1 Hyperparameter optimization

In order to find the best parameter configuration and optimise DT performances, we compared the results of both Grid and Random Searches. Although the last one is an approximation of the first, it returned models with similar performance, with significantly lower execution time.
The parameters we chose to optimise (for both the 50:50 and the 73:27 balanced datasets) are:

- Split criterion: between entropy and gini;
- max_depth: integers values between 4 and 10;
- min_samples_split: with values between 2 and 300;
- min_sample_leaf: with values between 1 and 150

Running Random Search algorithm with the 50:50 balanced dataset, the best parameters we got for the decision tree were the following. Other resulting models differed in depth and min_sample_split but not in mean validation score value.

Model (with rank: 1)
Mean validation score: 0.752 (std: 0.017)
Parameters: {'criterion': 'gini', 'max_depth': None, 'min_samples_leaf': 50, 'min_samples_split': 200}

4.2 Decision tree validation and performances

In order to assert whether the model is affected by overfitting we can measure its performance on both training and validation sets. The results can be summarized as in the following table.

Target	Precision	Recall	F1-score
No Churn	0.81	0.72	0.77
Churn	0.75	0.83	0.79
Average	0.78	0.78	0.78

Table 4.1: Performances measured on training set. The average accuracy is 0.78

		Actual	
		YES	NO
Predictions	YES	866	330
	NO	200	996

Target	Precision	Recall	F1-score
No Churn	0.79	0.68	0.73
Churn	0.72	0.82	0.76
Average	0.75	0.75	0.75

Table 4.2: Performances measured on validation set. Average accuracy is 0.75

		Actual	
		YES	NO
Predictions	YES	203	96
	NO	55	44

Validation performances are slightly lower than those obtained from the training set. Since we got two similar performances from the compared tests, we can assume that the model isn't overfitted.

The just obtained results of the prediction model could be compared with those obtained by using the original unbalanced dataset.
By running the Random Search algorithm with the same parameters as before, we can get the following results.

Model (with rank: 1)
Mean validation score: 0.791 (std: 0.005)
Parameters: {'min_samples_split': 60, 'min_samples_leaf': 40, 'max_depth': 9, 'criterion': 'entropy'}

Model (with rank: 2)
Mean validation score: 0.791 (std: 0.003)
Parameters: {'min_samples_split': 90, 'min_samples_leaf': 30, 'max_depth': 7, 'criterion': 'entropy'}

Fitting the Decision Tree with the above obtained parameters on the initial unbalanced dataset we got the following performance.

Target	Precision	Recall	F1-score
No Churn	0.78	0.77	0.78
Churn	0.77	0.78	0.78
Average	0.78	0.78	0.78

Table 4.3: Performances measured on training set. Average accuracy is 0.78

Target	Precision	Recall	F1-score
No Churn	0.75	0.73	0.74
Churn	0.74	0.76	0.75
Average	0.74	0.74	0.74

Table 4.4: Performances measured on training set. Average accuracy is 0.74

The above illustrated performance results show that the original dataset actually brings to a better prediction model. Therefore, we decided to use it.

4.2.1 Cross validation

Cross validation has been used to get a better estimation of the model performances.

The average evaluation of its application (with 10 folds) is the following (50:50 balanced set):
Accuracy: 0.7535 (std: 0.06)
F1-score: 0.7531 (std: 0.06)

The average evaluation of its application (with 10 folds) is the following (73:27 unbalanced set):
Accuracy: 0.7946 (std: 0.03)
F1-score: 0.7114 (std: 0.04)

4.3 Best prediction model

Cross validation provided similar results for both datasets, but with the balanced one, training set F1-score resulted a bit better (about +4%). For this reason we'll use this one to test data. The resulting decision tree is illustrated in figure 4.1. The first node spits customers in an unbalanced way, about the 75% follow contract $\leq$ than 0.5, i.e. as a contract type ”Month-to-month”. Red nodes represents a majority class of No Churners, while blue ones have a majority class of Churners. Customers following ”Month-to-month” customers are more likely to churn.

Applying the model to the given test set, we got the following performances.

Target	Precision	Recall	F1-score
No Churn	0.90	0.70	0.79
Churn	0.48	0.79	0.60
Average	0.79	0.72	0.74

From the mentioned decision tree we can extract some information such as the feature importance of the given dataset. It can be summarised by the bar chart in figure 4.2, in which we can observe which attributes (and in which measure) make their contribution for the target prediction.

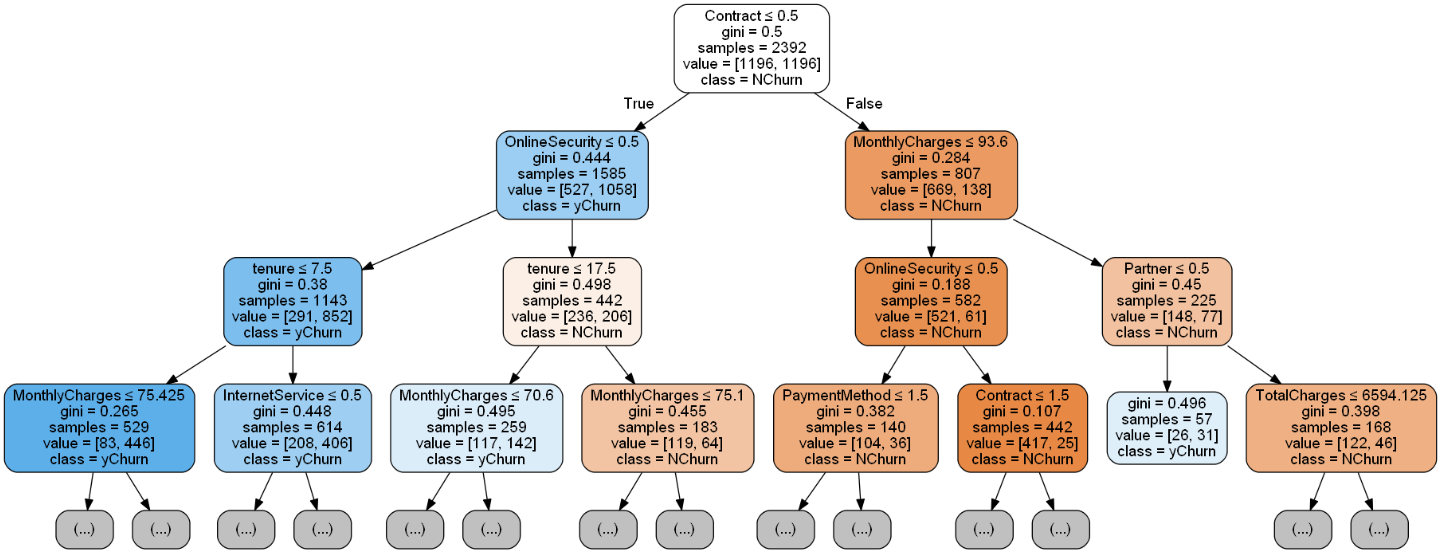


Figure 4.1: Decision Tree generated by random search over balanced data

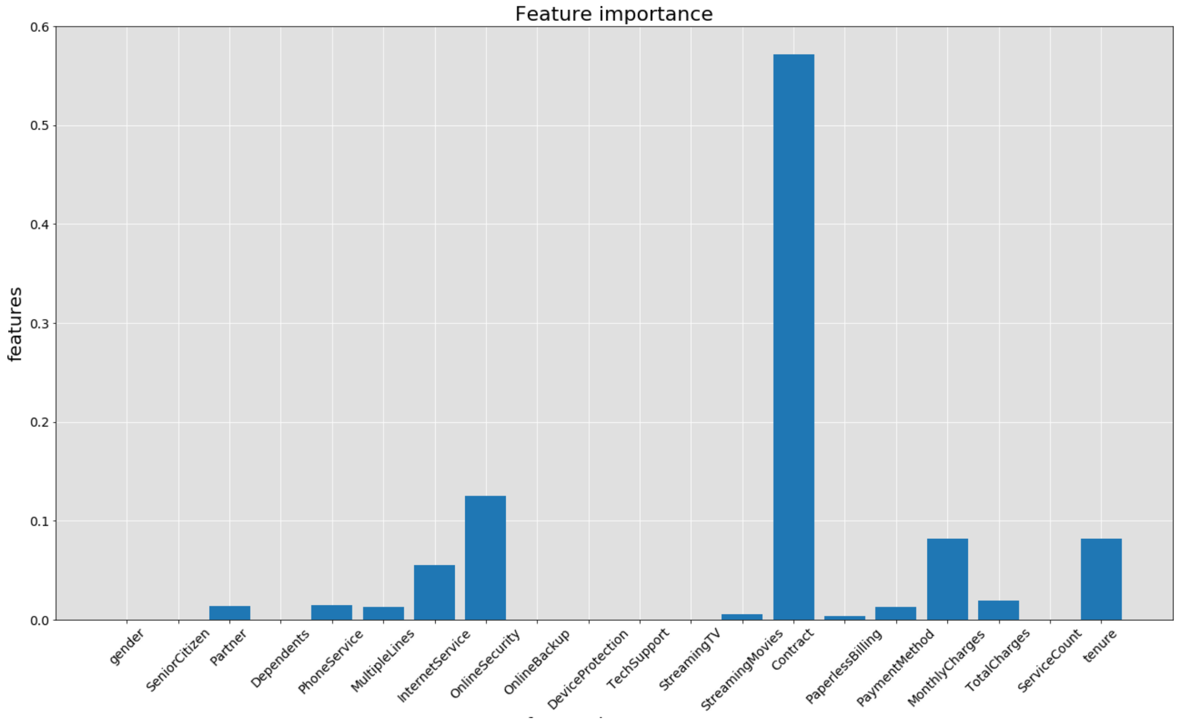


Figure 4.2: Feature importance Decision Tree model

Chapter 5

Conclusion

The provided dataset is mostly composed of categorical attributes, mainly concerning customers personal information and services information. As we observed from the feature importance of the variables used to determine the final decision tree, the most relevant ones are those concerning the relationship between users and the company. Indeed, attributes regarding private user information like age range, partner or gender resulted pretty useless for our goal.

Clusters found in Chapter 2 denotes some interesting groups of users having common features, that can also be helpful to determine the target: significant examples are clusters 1 and 4, which represent groups of non churning users with a pretty high accuracy.

Finally, association rules found in Chapter 3, don't give significant support to confirm or refute any common bias with regard to the context.

Through the ran tests, we also observed a non insignificant aspect of some prediction models like DTs: the bias towards the more frequent class. Although the performance of the tested models were pretty similar, the one performed with a balanced training has proven to be better than the first one during the test step.