Nova University of Lisbon
NOVA IMS Information Management School
Postgraduate Program in Enterprise Data Science & Analytics

Group Project - Unsupervised Learning

Book Me

Data Science and Machine Learning

Ana Lúcia Barriga, 20211812
Ana Rita Coelho, 20211335
João Frederico Monteiro, 20211786
Manuel Félix, 20211333
Manuel Fernandes, M20180396

Professors: Carina Albuquerque and Roberto Henriques

Spring Semester 2022

Contents

1 Background

It was another morning at Book.me headquarters when our management team summoned us to the large meeting room. They had received a phone call shortly after our yearly presentation and IPO announcement that Booking.com was interesting in acquiring us. Even though we had been named Startup of the year 2021 and have been performing well for the last three years, this came as a shock based on our current forecast, and both management and marketing were scratching their heads on how we were able to draw Booking.com attention. This meeting was to propose a new challenge to our data analysis team: in a nutshell, figure out what our team had done right last year to try and bargain with Booking.com or try to uncover more about their idea for acquiring us. Also, this would help the marketing team focus their energy and resources in demographics that would make a difference. As suggested by both management and the marketing team, the best place to start would be with our customers: trying to understand how they are clustered and how to best tap into what drives them.

2 Data Exploration

Based on the initial perspectives showcased by management, the focus was clear: understand our customer demographic, sort it by groups and then target them with different strategies alongside the Marketing team. We set out to uncover more information about our data set.

Our starting variables were mainly connected to the customer profiles and historical data that were extracted from our database, containing both numerical variables, such as the customer's year of birth, number of reward points and ratings for each of the different aspects assessed in each hotel; and categorical variables, such as room type selected, type of travel, whether that customer has left the platform or not (churn) and whether that customer had been in our platform for over a year.

2.1 *Variable Exploration*

Regarding customer characteristics, we were able to identify that most of our customers were born between 1970/1980, followed by those born between 1980/1990 and 1990/2000, respectively (Figure 1). This depicts our customer demographic between working age, from early 20's until mid 50's. This does not come as a surprise when we assess the type of travel per customer, as 69% of them use the platform for booking business trips. Most of our customers are also considered long time users, as 81.5% have been using the platform for over a year. In terms of room types preferred, both single (47.7%) and double room (45%) have an advantage over suites (7.3%). Regarding reward points, by far the vaguest of all the variables connected, we can see that the majority of our customers have 3000 reward points or more. It is uncertain to us how they can capitalize these points or whether they can use them in discounted bookings, upgrades towards suites or better amenities. This might pose an interesting perspective if included in our strategy (elaborated further in section 5).

Regarding satisfaction, our current survey includes 14 different topics: comfort experienced by the customer during his/her stay; reception schedule for check in, check out and general enquiries; food and drinks served at the restaurants or bars, should the hotel have them; hotel location; quality of WiFi experienced in different hotel sections; available amenities experienced in the hotel, such as gym, sauna and pool, leisure rooms, etc; staff's availability, conduct and attitude; online booking functionality on our website and app; price/quality ratio, that is, paid price per perceived hotel quality; available room space; check-in and

check-out facilities (such as express check-in/out, digital invoicing for company expenses, etc.); general cleanliness of the hotel; and bar service (quality of space and service provided). At first glance, *Comfort*, *FoodDrink*, *ReceptionSchedule* and *Location* should be further analysed as their mean value is below 3.0. Conversely, *CheckOut*, *Cleanliness* and *Staff* represent the best average values (all above 3.5).

A skewness and kurtosis evaluation was carried out to understand the above-mentioned variables a bit better and potentially pin down some outliers. As expected, *Comfort* and *FoodDrink* are the only positively skewed variables, while *Staff*, *CheckOut* and *Cleanliness* are the most negatively skewed variables (with relevant skewness values over $|0.5|$). However, after running a kurtosis analysis, none of them represent potential outliers. Only *RewardPoints* has a positive kurtosis value, to which we have identified some outliers by conducting an in-depth analysis using a histogram and boxplot.

The initial challenge presented by the management board to our data analysis team was to focus on customer characteristics and customer satisfaction. However, as it is standard for data analysts, we would prefer to trim down the number of variables as much as possible, since that represents a better visual representation and interpretation, as well as reducing the amount of work needed. As such, we decided to run an average rating to understand if this would be a good representation of the 14 variables.

2.2 *Insights*

After running a *spearman* correlation heatmap with correlation coefficients (add figure), our team was able to extract some insights. Firstly, **Customers with higher average rating (or more specifically in Amenities, Staff and Online Booking), are less likely to churn**. Also, **Customers who travel for business trips tend to book single rooms**. Besides, **Customers who book either doubles and suites are more prone to churning**. **Recent customers (with Longevity = 0) are more prone to churning**. And, lastly, **Long-time customers (with Longevity = 1) are likely to be traveling for business**.

The need to understand and potentially remove duplicates is paramount. In our data set, this task proved not to be as straightforward, since both the same customer can use our platform to perform two different bookings and we would miss out on identifying potential patterns. The same can be said for different customers with the same name, that could mistakenly be assigned under the same profile. Likewise, no information was provided about reward points and how they tie in with the number or type of trips.

As part of our exploration, we tried to understand a bit more about customer recurrence in order to understand if the same customers are tracked and whether we are able to extract any useful information. We initially filtered by name and year of birth, which provided 39 customers that could potentially be recurring. We then considered their longevity, with the intention of cutting any ties between new subscriptions and existing customers, which led to 26 potential recurring customers. Lastly, we narrowed it down to recurring customers by room type, more specifically, 13 recurring business customers. Out of these 13, we were able to identify a pattern of around 700 reward points per recurrence on average. This will be helpful when defining the Marketing strategy and will be explored further in section 5.

We also identified 1051 customers whose year of birth was greater than 2004, then underage customers. However, they can be indeed travelling themselves, and, based on the prior explorations, they can be considered as travel companions. So no alterations were performed on these customers.

Regarding the missing values that will need to be filled, we were also able to find a pattern: 81.5% of missing values in *Year_Birth* turned out to be female customers. This means that female customers are less likely to fully fill satisfaction surveys with their age (missing at random). Another interesting aspect that will need to be addressed during pre-processing is the different scaling for each of the satisfaction ratings.

Each one suggests that customers can fill their satisfaction rating from 0-5, however, some of them have 1 as minimum value, with *WiFi* being ranked as high as 6 (showcasing a proper 5Ghz WiFi router investment in some hotels). In order to normalise this, we set every scale as 1-5, with 1 being our new minimum (changing every 0 to 1 as a result) and 5 being our new maximum (changing every 6 to 5). This is also necessary if we start bundling variables together.

On a side note, our team was left puzzled with some aspects regarding database administration: how come most of these missing values and non-normalized values are happening? Perhaps the survey prompts the customer to fill in the numbers as opposed to selecting from a standardised list? Also, regarding reward points and recurrence, surely creating specific customer IDs and date of booking to better understand the time-frame associated to this database would increase our understanding of the customer behaviour. In hindsight, we can now understand the impact of budget cuts in the Database Administration team.

We also understood the sudden interest in Booking.com in our platform: we are clearly outperforming them in a niche demographic that they do not control - the business market. We should be able to maintain this demographic as much as possible, while trying to attract more customers in different ones.

3 Data Pre-Processing

The next step for our team was to clean up the data set, removing any inconsistencies regarding missing values or non-normalised data. The first step is eliminating duplicates and, with the knowledge that we now have from the exploration regarding recurrence, we are able to only identify and successfully eliminate 3 customers that were considered duplicates, since they were identical matches on every variable. Our data set was left with a 15586 customers to run our analysis. We also discarded the existence of outliers, as the only potential one were customers that had rated everything as all 0s, all 1s and all 5s (which did not happen often and are perfectly within reasonable intervals); or instead some customers having a lower number of reward points (which is also acceptable, given our interpretation of this variable). We also had to change some of the values in Longevity, as some customers wrote 'y' instead of 'yes' as an answer.

We had to solve the missing values in customer's year of birth to try to understand if there are patterns associated to it. For that, we considered filling the gaps with mean or median values, but ultimately went with a KNN Imputer method (subsection 6.2) that computes a local mean. In order to fill the 195 missing values in age, we used the nearest 125 neighbours (k = 125), with uniformed weights (as we are unsure whether any particular characteristic is more important than others) and euclidean distance metric between neighbouring points (which meant assuming that similar characteristics will fall under a similar age profile). After filling the missing values, we turned the variable *Year_Birth* in *Age*, which is much easier to identify.

Based on greater correlations identified among the satisfaction variables, we also generated three buckets of variables, *Rating1*, *Rating2* and *Rating3*, that allow an overall look on different satisfaction perspectives. They will be later discussed in subsection 4.2.

From the prefix "Mr." or "Ms." in the variable *Name*, we are also able to consider a new variable *Gender*.

The next step of the process was to turn categorical variables into numerical ones (dummy variables). The easiest way of achieving this was to turn them into 1 or 0 value depending on the outcome. This was possible for *Gender* (female as 0, male as 1), *Longevity* (under 1 year as 0 and over 1 year as 1, after normalizing everyone as 'yes' or 'no'); *TypeTravel* (business as 1 and leisure as 0) and*Churn* (0 for no churn and 1 for churn). *Room Type* required an explicit creation of dummy variables that acted as flags for whether the room type was single, double or suite.

With every variable normalised and processed, we revisited the *spearman*-correlated heatmaps (Figure 2) to check whether our initial insights still stood. One of the most interesting insights that we were not able to gather before was the relationship between our bundled *Ratings2* and *Rating3* variables with *Churn*, which meant that two of our envisioned rating bundles create more impact to customers than the third one. Still, this is a business-led idea that needs to be validated with clustering before proceeding.

4 Clustering

4.1 *Customer characteristics*

In a first instance, clustering algorithms were applied to a ”customer characteristics” perspective that includes the variables of the data set that allow us to understanding which types of customers are using our services. Since the not all variables considered range between the same values, we proceeded to normalize the data using the MinMaxScaler, which scaled each feature individually between 0 and 1.

Before moving forward to the clustering algorithms, used Principal Component Analysis (PCA) (subsection 6.3) in order to perceive if any other feature reduction could be performed and to infer and confirm correlations among this data set. By analysing PCA results, we observe that for the first three principal components (PCs), which explain 85% of the variance of the data set (Figure 3), the variables on the room type (single or double), gender, and type travel and longevity, are sequentially moderate or highly correlated with them (Figure 4). We suppose that the type of room our customers book can be relevant to help us segment them into different clusters. In contrast, age does not seem to be correlated to any of the PCs, thus probably our customers will not be clustered according to this characteristic. Despite the meaningful insights gathered from the PCA, we decided not to move forward with it since the number of features in the original data set dimensions is adequate for a clustering algorithm. In addition, the gain in computational time is not balanced by the loss in the data set variance, since the number of rows is relative small.

Regarding the clustering algorithms, we started by applying KMeans to the data on customers' characteristics. The elbow method was useful to help us define an adequate number of clusters to start our analysis, by plotting the inertia against different numbers of clusters (k). By looking at Figure 5, a greater decrease on the curve's slope is attained when k equals 2, then suggesting this might be a good number of clusters. In parallel, the dendrogram from the agglomerative bottom-up approach hierarchical clustering algorithm applied to our data, in Figure 6, indicates that 3 can be a good number of clusters. The agglomerations below this level lead to a large number of clusters that is not compatible with the aim of defining customer segments. Therefore, KMeans with 2 and 3 clusters was applied to this data set.

A summary of the the clusters' profiles for both k = 2 and k = 3 can be observed in Figure 7. Note that when applying KMeans using or not the *kmeans++* initialization, KMeans converged to very similar solutions. This could be possibly explained by the fact that the data set under analysis is relative small, thus not requiring a greater need of initializing the centroids (seeds) not randomly.

- Using k = 2, the clusters are primarily differentiated by the type or room - single or double - our customers book. Broadly, customers in the first cluster travel by business reasons and stay on single rooms, while the ones in the second cluster travel mainly by leisure and stay in double rooms. Customers are distributed approximately evenly between the two clusters.

- Using k = 3, we get a clearer distinction between customers who travel by leisure and those who travel by business reasons. Moreover, we are able to distinguish between two types of customers between

those who travel in business trips - the ones who stay in single and those who stay in double rooms. About half of the total number of customers travel by business reasons and reserve single rooms, while the remaining are distributed by the other two clusters. Besides, the longevity of customers who travel in business and stay in double rooms is smaller, suggesting these are ”new customers” compared to the ones in the remaining clusters, for whose longevity mean values are closer to 1. The remaining variables, such as age, gender or the number of reward points, do not appear to be distinctive between the three different clusters.

Lastly, we performed clustering on our data set using Density Based Spatial Clustering of Applications with Noise (DBSCAN) (subsection 6.4), since it is a density-based method unlike KMeans or hierarchical clustering. This way, we are able to compare results provided, unveiling new properties of clusters, and cross-validating the results obtained by KMeans. We performed a similar analysis to the ”elbow method” in order to determine which could be the best parameters for this algorithm, which can be tough to tune. However, it seemed like 20 clusters would be generated. Looking at the dendogram previously drawn for the agglomerative hierarchical clustering algorithm, this number is a resonable possibility, but it does not suit out initial problem, once we want to segregate customers in distinct, comprehensible, clusters. We managed to get only 3 clusters (Figure 8) by using *eps* around 1 and *min_samples* equal to 15, which segregated the data based on the room type. However, we can still identify similarities the clusters.

Based on the conclusions drawn from the previous analyses, and minding our main goal, the results obtained from KMeans with k = 3 were used to define the customer segments.

4.2 *Customer satisfaction*

Since our first approach when delving through the data - especially the correlation matrix - one thing was clear: we needed to get a better understanding on our customers satisfaction, a difficult task since we had this perspective spread throughout 14 different variables, from Comfort to Bar Service. We needed a better approach on these variables to take conclusions out of them, so we thought PCA, again could help us. Truth be told that reducing the number of variables leads to a reduction of accuracy, but when done accurately we can trade some accuracy for simplicity, an aspect we definitely need.

The first step before applying PCA would be, again, to normalize our data, but since all the 14 variables vary between 1 and 5, there was no need to do this, so we went forward to calculate the PCs. We looked at the amount of variance explained by each PC in Figure 9, to understand if reduction we could be performed.

Knowing that in order to reduce the number of variables we need to reach a threshold of around 85% of the explained variance, and although our first 3 PCs explain 61.55% of it, we still needed 8 PCs, reaching 86.74%. Well, 8 components still sounded to be too many, so we went through to understand what conclusions we could get out of it. By observing the leading scores of each PC, we noticed that only the 5^{th} and 6^{th} PCs showed strong correlations with one variable, each one, of our data set (Figure 10). Regarding the first four PCs: the first PC, representing 27.43% of the variance, does not show even a moderate correlation with any variable, nor does the third PC; the second PC, representing 19.84% of the variance, exhibit a moderate correlation with Reception Schedule and Food Drink; lastly, the fourth PC, representing 7.37% of the variance, demonstrates a moderate correlation with Amenities.

Thus, we concluded that implementing a PCA on the customer satisfaction data set may not present great advantages. However, from the correlation matrix (add reference), we found correlations among these 14 variables, so we knew we could encompass them into different buckets. So, proceeded to apply

clustering method on this data set to help us define these buckets.

From our previous section, we know that we can obtain good information from the KMeans method. Again, we started by defining the number of clusters to use through the elbow method. We reached the value of 3 clusters (add reference) and decided to analyse our data for a couple of values of k around 3: k = 3, k = 4 and k = 5 clusters (Figure 11). We could gather meaningful results:

- Clustering with k = 3, k = 4 and k = 5 indicated two clusters that did not change (clusters with labels 1 and 2 for k = 3). One of them contains customers that are satisfied with all the variables analysed; the other includes customers for whose there is a clear distinction between the first four variables (with a lower score) and the remaining ones. This suggests we can aggregate the variables into two possible buckets: one with *Comfort*, *Reception Schedule*, *Food Drink* and *Location*; and another with all the remaining variables. The cluster values can be seen on the following figures: Figure 12, Figure 13, Figure 14
- When looking at the result for k = 4, we start to observe two new buckets taking form according the same reasoning: one that encompasses *Wifi*, *Amenities*, *Staff* and *Bar Service* (*Rating 1*); and another with *Price Quality*, *Room Space*, *Check Out*, *Check In* and *Cleanliness* (*Rating 2*).
- Finally, with the results from k = 5, we get the sense we found our buckets: *Rating 1* and *Rating 2*, already mentioned, and a third bucket (*Rating 3*) with the variables *Comfort*, *Reception Schedule*, *Food Drink* and *Location*. Such findings are supported not only by the results for k = 3 (the cluster that was associated to a lower score to this set of variables), but also from the results for k = 3 (the cluster with label 2 that shows good rating among these variables).

On its turn, DBSCAN did not provide a meaningful segmentation of customers in this perspective.

The findings from the previous analyses are consistent with the findings from the data exploration and pre-processing steps of our project, in which we identified the same three possible buckets of variables according to the correlations between the initial 14 satisfaction variables. We then considered these new three variables, *Rating 1*, *Rating 2* and *3*, to define a third perspective and obtain a more clear, objective, vision of our data.

4.3 *Quality of service*

We moved the perspective of out data regarding the quality of the service, with the aim of taking a wider look on our customers' satisfaction given the services we are providing them, and understanding what is causing them to abandon our platform. This vision includes the rating variables *Rating 1*, *Rating 2* and *Rating 3*, as well as the *Churn* variable. Given that we are only applying clustering techniques to a data set with 4 variables, PCA was not considered in this analysis.

Proceeding to the clustering step, the elbow method (Figure 15) indicated that k = 3 would be a suitable number of clusters; the dendogram provided by agglomerative hierarchical clustering algorithm (Figure 16) suggests a k = 2 clusters. Based on these results, we decided to conduct several KMeans analyses for k = 2, k = 3 and k = 4 . Again, the results obtained using the *kmeans++* initialization were very similar to the ones obtained when assigning the seeds randomly.

The results obtained with the *KMeans* for different k values in Figure 17 are summarized below:

- Using k = 2, the clusters are differentiated by customers that churn and give a lower rating overall and by customers that do not churn and give a good feedback in every rating.

- Using k = 3, one of the clusters aggregates the customers who do not churn and give a high review in the three rating variables. Another cluster groups customers that also do not churn and give a good review in *Ratings* 2 and 3 despite a lower classification in *Rating 1*. Lastly, the last cluster contains mainly customers who churn and are overall less happy with the service hence the lower classifications in every rating variable.
- Using k = 4 (Figure 18 and Figure 19), we get the same segmentation for customers who do not churn as in k = 3, but there is a distinction for customers who do churn. One of the clusters contains customers who churn and provide a lower classification in *Rating 1* and *Rating 2* whilst the second cluster aggregates customers that churn and give lower ratings in *Rating 1* and *Rating 3*. From this analysis, *Rating 1* classification does not appear to be determinant for a customer to churn.

Customers are distributed quite evenly between clusters for each one of the cases detailed above.

Again, for a matter of comparison, we also generated clusters using the DBSCAN. For the default values of the DBSCAN parameters, it was possible to get 3 clusters. By tuning the *eps* for eps = 0,4 while keeping *min_samples* = 5 (Figure 20), we were able to get 4 clusters. Although the number of clusters is quite similar to the one suggested by the elbow method in *KMeans++*, DBSCAN did not deliver good results for this data set. For k = 4 (Figure 21), two of the clusters aggregate the majority of the points: cluster of customers who do not churn give good reviews overall and the cluster which has the majority of customers that churn and give lower reviews overall. The size of the remaining clusters is reduced and only differentiate between customers who do not churn, which might not be relevant from a business point of view.

Minding our goal, we decided to establish the four clusters from *KMeans* since they provide a meaningful, distinct customer segmentation.

4.4 *Customer segmentation*

In order to maximize the potential of our marketing plan, and in an effort to grow our business, we segmented our customers based on the clustering analysis. Therefore, we concatenated the clusters from both perspectives to combine all meaningful insights and draft a more incisive, targeted marketing plan. On total, we reached 12 customer segments (Table 1).

Recall the three clusters in ”customer characteristics”: customers that predominantly traveled for business, booked single Rooms and are loyal to BookMe; customers that travel almost exclusively for leisure, book mostly double rooms and are also loyal to our business; and lastly, customers that only travel for business and book mainly double rooms, this last type of customer stands out for the lack of loyalty since only nearly half has been a customer for over a year.

In the ”quality of service” perspective, we are able to separate customers into four distinct groups: the ones who do not churn and are overall satisfied; the ones who do not churn despite the lower review in *Rating 1*; the ones who churn with despite the good reviews in *Rating 3*; and the ones who churn despite the good reviews in *Rating 2*.

5 Marketing plan

Upon identifying the differences within customer characteristics and customer satisfaction by clustering them and successfully identifying the different customer segments, we gathered with our Marketing team to try and bring innovative ideas to management. After our insights and clustering process, we were sure we

had pinpointed the interest behind Booking.com, and we were set out to demonstrate how we could further expand our business:

5.1 *Book.me Bargain - New booking option*

In this option, the goal is to **improve Rating 3 to prevent churned customers**. Churned customers with bad average scores on Rating 3 are the main target for this bundle, however, we believe that extending it to everyone would be beneficial. At a cost of claiming half reward points, customers are now able to book ”Budget Accommodations”, ideally impacting in the *PriceQuality* ratio. As the largest slice of these customers are located in the Business demographic for type of travel, we also included earlier check-ins and late check-out options if needed.

Since new customers have been booking double rooms for Business purposes, we figured we should include first time companion discount to help promote this, so on top of the previous perks, they also get a 50% discount voucher when travelling with a buddy (we call it the buddy system).

5.2 *Book.me Travelers Program - New subscription option*

In this option, the goal is to **improve Rating 1 on non-churned customers and help prevent data gaps**. Fortunately, a large part of the customer base is still happy to be partnering with Book.Me. To help celebrate these customers, we suggested a Book.me Traveler Program; a subscription option that seeks to reward customers that have been with us for a long time. The Book.me Travelers Program has 3 different levels of loyalty based on the amount of bookings per year (Local Hero, Intercontinental Traveler and Worldwide Explorer) and offers discounts in local restaurants, room upgrades and early check-in/late check-out options. Additionally, customers would immediately enroll in the Loyalty Package Program which would enable a gift voucher on each anniversary with Book.me, as well as a referral program to earn additional reward points.

5.3 *Book.me Premium and Book.me Nomad - New upgrade features*

In this option, the goal is to **improve rating 2 on churned leisure (Book.me Premium) and business (Book.me Nomad) customers**. Customers who enroll in the Travelers Program are now able to choose between two upgrade features, one focused on Leisure and another one for Business. Both upgrades can be used by cashing in accumulated reward points (RP) from previous trips.

Available for 2100 RP, the Book.me Premium is our answer to churned customers traveling for leisure. To improve *BarService*, *Amenities* and *Staff* ratings, only the best accommodation options will be available for this upgrade, and it will include a premium bar access, superior amenity options such as gym, pool and spa access and a ”no cancellation fee” option. Selected hotels can also include room upgrades and complementary welcome drinks.

Available for 1400RP and specifically tailored for the customers that like to work on the road or at the bar, the Book.me Nomad upgrade was designed to improve *BarService* and *WiFi* ratings. The available accommodations in this package will be fully equipped with chill-out lounges, with that rank-6 5GHz WiFi and extended happy hour. Customers will be greeted with a welcome drink (after working hours, of course) at the exchange of hard-earned reward points from previous business trips.

6 Annex

6.1 *Figures and tables*

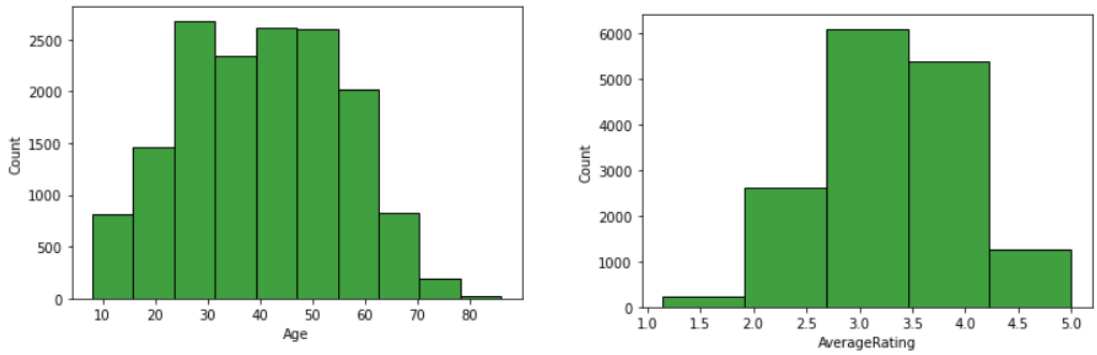

Figure 1: Data Exploration: *Age* distribution and Average Rating.

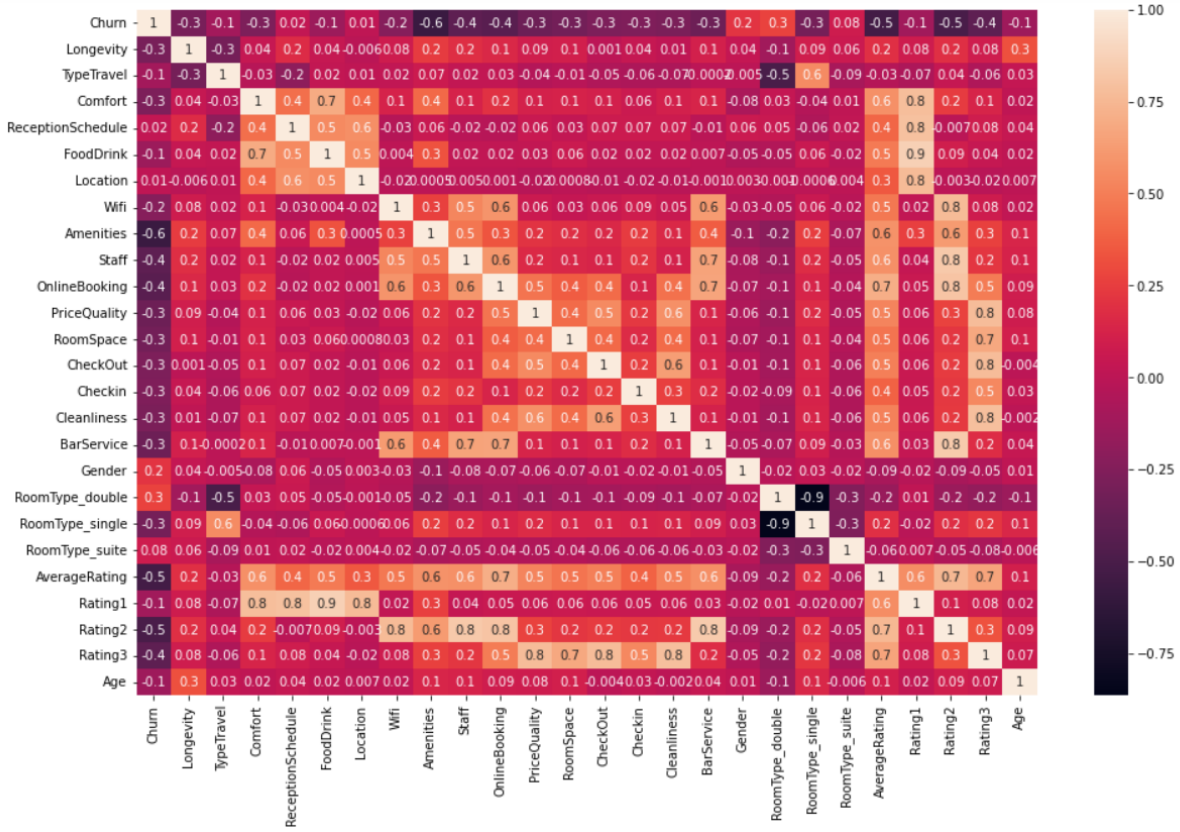

Figure 2: *Spearman*-correlated heatmap (post-boolean and normalisation processes.

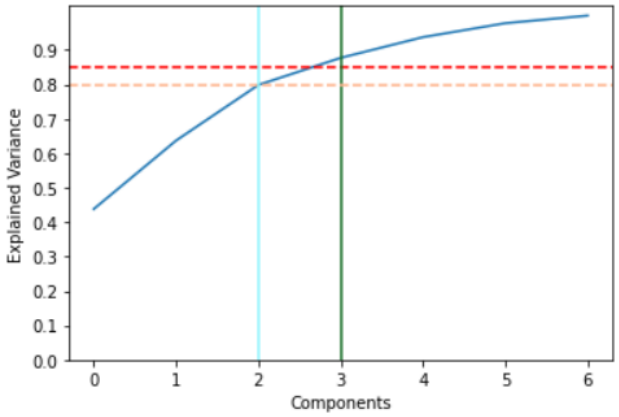

Figure 3: Customer Characteristics PCA: cumulative explained variance according to the number of PCs.

	0	1	2	3	4	5
Gender	-0.02	-0.98	0.17	0.03	-0.00	-0.04
Age	-0.03	-0.02	-0.12	-0.05	-0.24	-0.39
TypeTravel	-0.45	0.11	0.54	-0.03	-0.69	0.08
RoomType_double	0.62	0.02	0.25	-0.43	-0.19	0.03
RoomType_single	-0.64	-0.02	-0.15	-0.37	0.31	-0.08
RoomType_suite	0.01	0.00	-0.10	0.80	-0.12	0.05
Longevity	-0.00	-0.14	-0.75	-0.18	-0.56	0.16
RewardPoints_Bins	0.06	0.04	-0.02	0.05	-0.10	-0.90

Moderate Correlation: >=0.5

Strong Correlation: >=0.75

Figure 4: Customer Characteristics PCA: component/correlation matrix (loading scores) for all the estimated PCs. Greater correlations between variables and PCs are highlighted in light blue (moderate correlations) and dark blue (strong correlations).

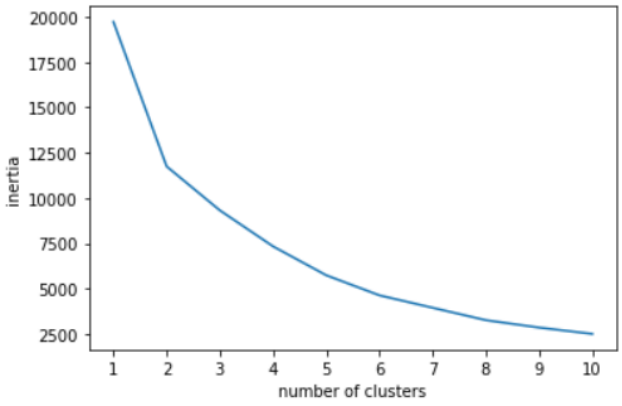

Figure 5: Customer Characteristics KMeans: elbow method.

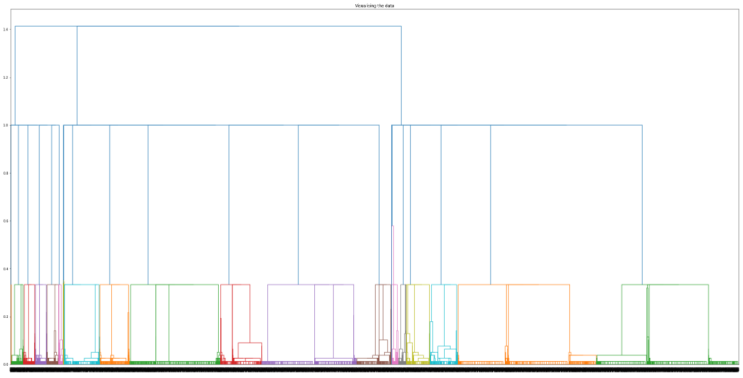

Figure 6: Customer Characteristics Agglomerative Hierarchical Clustering results: dendogram.

label	0	1
Gender	0.496028	0.479150
Age	0.439639	0.386715
TypeTravel	0.959906	0.401195
RoomType_double	0.000000	0.932138
RoomType_single	0.923784	0.000000
RoomType_suite	0.076216	0.067862
Longevity	0.849553	0.779283
RewardPoints_Bins	0.721119	0.792032

label	0	1	2
Gender	0.501196	0.489244	0.452107
Age	0.441310	0.402633	0.361434
TypeTravel	0.959346	0.000000	1.000000
RoomType_double	0.000000	0.886671	0.964559
RoomType_single	0.936690	0.000000	0.000000
RoomType_suite	0.063310	0.113329	0.035441
Longevity	0.861422	0.994899	0.441252
RewardPoints_Bins	0.720621	0.793894	0.788101

Figure 7: Customer Characteristics KMeans results: mean values for each cluster obtained for the analyses with k = 2 and k = 3, from left to right respectively. Note that these analyses were performed on the normalized data set.

label	0	1	2
Gender	0.501881	0.477418	0.460444
Age	0.442058	0.384968	0.410496
TypeTravel	0.956598	0.430403	0.545778
RoomType_double	0.000000	1.000000	0.000000
RoomType_single	1.000000	0.000000	0.000000
RoomType_suite	0.000000	0.000000	1.000000
Longevity	0.852056	0.763357	0.900444
RewardPoints_Bins	0.714638	0.790996	0.802667

Figure 8: Customer Characteristics DBSCAN results: mean values for each cluster obtained for the analysis with *eps* = 1.2 and *min_samples* = 15. Note that the analysis were performed on the normalized data set.

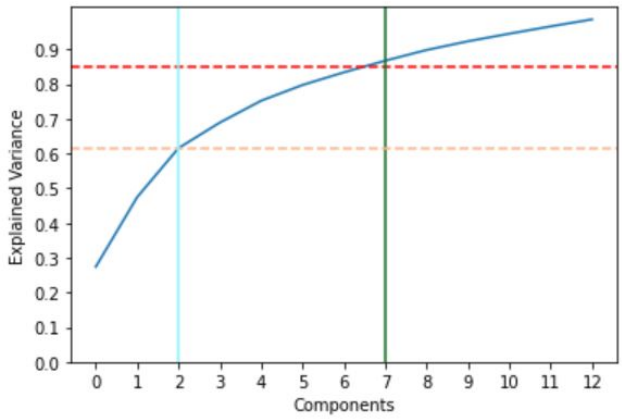

Figure 9: Customer Satisfaction PCA: cumulative explained variance according to the number of PCs.

	0	1	2	3	4	5	6	7
Comfort	-0.26	0.39	0.05	-0.23	0.26	-0.16	0.34	0.03
ReceptionSchedule	-0.14	0.50	-0.02	0.32	-0.32	0.03	-0.30	0.61
FoodDrink	-0.19	0.50	0.06	-0.17	0.16	-0.04	0.14	-0.07
Location	-0.11	0.43	0.05	0.37	-0.12	0.16	-0.05	-0.66
Wifi	-0.31	-0.18	0.35	0.23	0.01	-0.10	0.46	0.29
Amenities	-0.31	0.05	0.12	-0.64	0.11	-0.05	-0.32	0.03
Staff	-0.36	-0.17	0.29	-0.02	-0.11	0.17	-0.45	-0.19
OnlineBooking	-0.42	-0.20	0.02	0.24	0.17	-0.07	0.05	-0.04
PriceQuality	-0.25	-0.07	-0.42	0.04	0.05	-0.28	-0.35	0.09
RoomSpace	-0.22	-0.05	-0.38	-0.03	0.22	0.84	0.13	0.14
CheckOut	-0.21	-0.06	-0.41	0.10	0.01	-0.23	0.09	-0.13
Checkin	-0.18	-0.04	-0.16	-0.33	-0.82	0.06	0.31	-0.08
Cleanliness	-0.21	-0.06	-0.42	0.10	0.01	-0.24	0.07	-0.10
BarService	-0.36	-0.19	0.30	0.14	-0.09	0.03	-0.02	-0.04

Moderate Correlation: >=0.5

Strong Correlation: >=0.75

Figure 10: Customer Satisfaction PCA: component/correlation matrix (loading scores) for the PCs explaining more than 85% of the variance and original variables in the data set. Greater correlations between variables and PCs are highlighted in light blue (moderate correlations) and dark blue (strong correlations).

label	0	1	2
Comfort	2.43	4.04	2.01
ReceptionSchedule	3.06	3.97	1.95
FoodDrink	2.77	3.97	1.76
Location	3.01	3.80	2.00
Wifi	2.18	3.69	3.89
Amenities	2.60	4.13	3.41
Staff	2.31	4.08	4.15
OnlineBooking	2.06	4.14	4.18
PriceQuality	2.78	3.89	3.69
RoomSpace	2.90	3.87	3.63
CheckOut	3.16	4.05	3.88
Checkin	2.85	3.65	3.47
Cleanliness	3.15	4.05	3.86
BarService	2.11	3.90	4.06

label	0	1	2	3
Comfort	2.59	4.26	2.12	2.44
ReceptionSchedule	2.65	4.23	2.14	3.14
FoodDrink	2.62	4.22	1.98	2.69
Location	2.84	4.03	2.08	2.99
Wifi	3.53	3.69	3.79	1.76
Amenities	3.22	4.12	3.70	2.35
Staff	3.64	4.05	4.20	1.87
OnlineBooking	3.00	4.15	4.36	1.96
PriceQuality	2.22	3.94	4.17	3.23
RoomSpace	2.39	3.91	4.08	3.27
CheckOut	2.51	4.10	4.30	3.64
Checkin	2.74	3.65	3.74	3.01
Cleanliness	2.48	4.11	4.29	3.65
BarService	3.50	3.89	4.03	1.72

label	0	1	2	3	4
Comfort	4.30	2.34	2.10	3.02	2.36
ReceptionSchedule	4.21	3.00	2.18	3.79	1.99
FoodDrink	4.19	2.46	2.06	3.80	1.90
Location	4.01	2.84	2.06	3.68	2.37
Wifi	3.72	1.74	3.72	2.85	3.94
Amenities	4.15	2.24	3.88	3.42	2.96
Staff	4.09	1.81	4.24	3.17	3.88
OnlineBooking	4.23	1.95	4.41	2.39	3.61
PriceQuality	4.05	3.36	4.32	2.19	2.59
RoomSpace	3.96	3.32	4.23	2.61	2.63
CheckOut	4.19	3.78	4.43	2.45	2.93
Checkin	3.71	3.11	3.84	2.71	2.84
Cleanliness	4.20	3.78	4.43	2.40	2.89
BarService	3.93	1.70	4.01	2.81	3.90

Figure 11: Customer Satisfaction KMeans clustering results: mean values for each cluster obtained for the analyses with k = 3, k = 4 and k = 5, from left to right respectively. Higher values than the mid-point value (3) are highlighted in different shades of blue, helping to understand the profiles of the clusters.

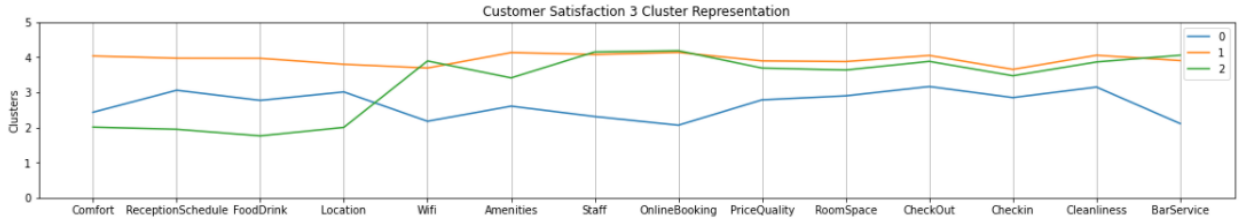

Figure 12: Customer Satisfaction KMeans results: 3 clusters.

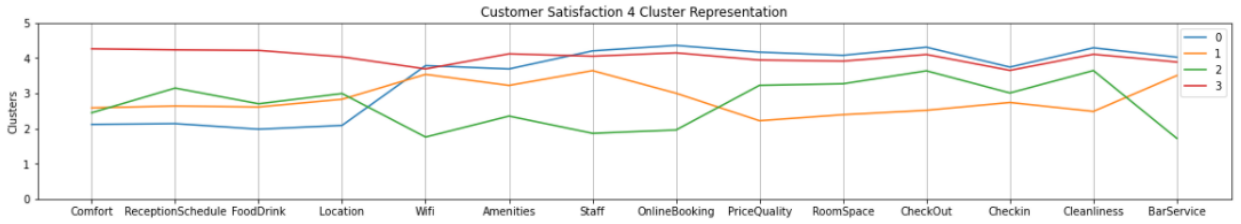

Figure 13: Customer Satisfaction KMeans results: 4 clusters.

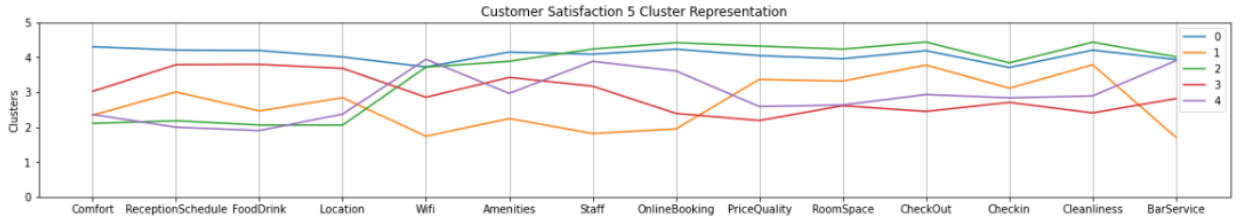

Figure 14: Customer Satisfaction KMeans results: 5 clusters.

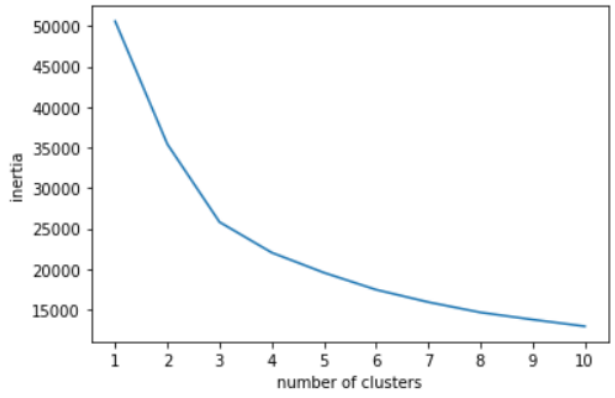

Figure 15: Quality of Service KMeans: elbow method.

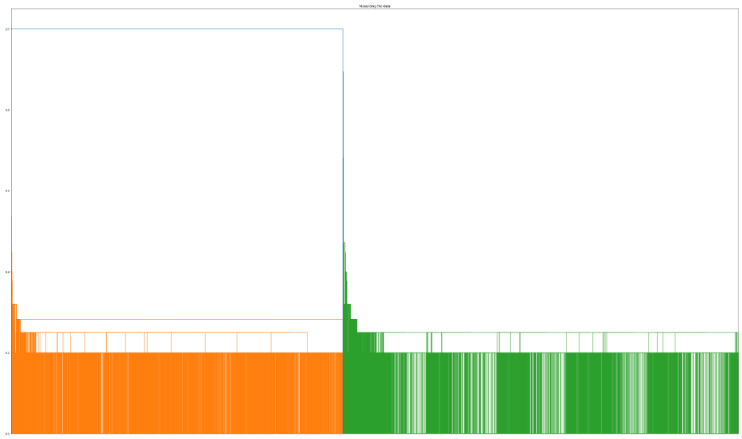

Figure 16: Quality of Service Agglomerative Hierarchical Clustering results: dendogram.

label	0	1
Churn	0.201325	0.796108
Rating1	3.154559	2.678518
Rating2	4.077229	2.474102
Rating3	3.919335	3.012605

label	0	1	2
Churn	0.230499	0.854197	0.238362
Rating1	4.143855	2.704582	2.101410
Rating2	3.922905	2.332425	4.032026
Rating3	3.812911	2.972023	3.869036

label	0	1	2	3
Churn	0.202022	0.832505	0.168170	0.777417
Rating1	4.183483	2.746711	2.067777	2.663620
Rating2	3.957618	1.928617	4.062133	3.263267
Rating3	3.890337	3.433499	4.076660	2.461203

Figure 17: Quality of Service KMeans results: mean values for each cluster obtained for the analyses with k = 2, k = 3 and k = 4, from left to right respectively.

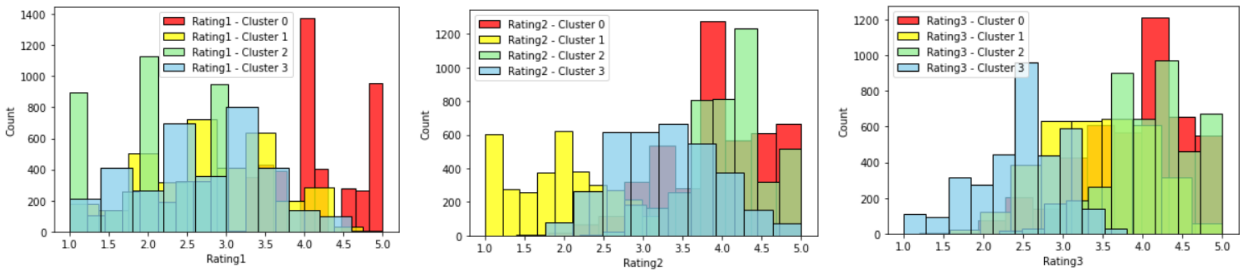

Figure 18: Quality of Service KMeans results for k = 4: histograms for the variables *Rating1*, *Rating2* and *Rating3*, respectively, for each one of the obtained clusters.

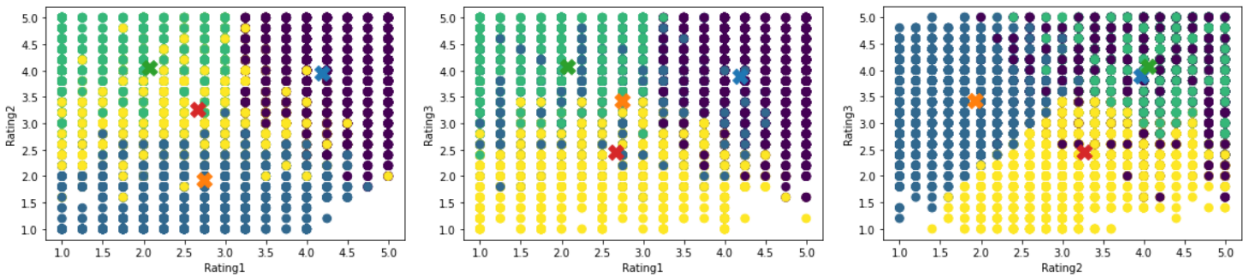

Figure 19: Quality of Service KMeans results for k = 4: scatter plots of variables *Rating1*, *Rating2* and *Rating3* against each others, allowing a visual interpretation of the resulting clusters. The centroids of each cluster are identified with a cross mark.

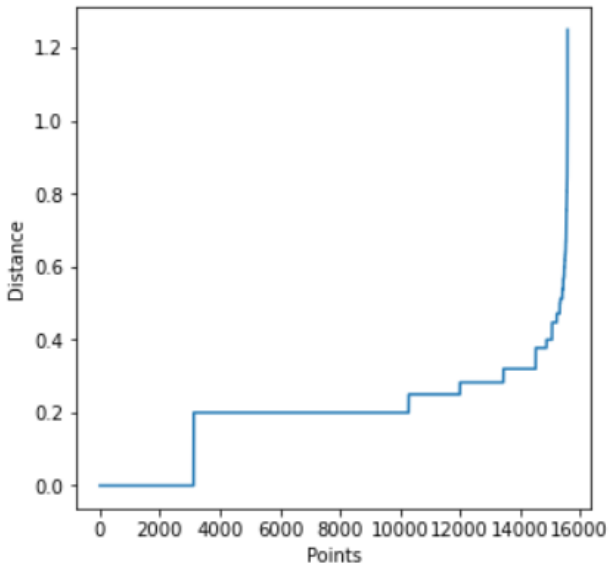

Figure 20: Quality of Service DBSCAN: estimation of the *eps* parameter through performing a *NearestNeighbors* analysis. The average distance between each point and its k-nearest neighbors is calculated where k = *min_samples* selected by the user. The plot shows the average k-distances in ascending order. A good value for *eps* would be the point associated to the maximum curvature or bend (greatest slope).

label	-1	0	1	2	3
Churn	0.321429	1.000000	0.000000	0.000000	0.00
Rating1	2.794643	2.809953	3.072193	2.191176	1.95
Rating2	2.228571	2.827887	3.877980	1.023529	1.04
Rating3	2.407143	3.151022	3.857968	3.870588	2.84

Figure 21: Quality of Service DBSCAN results: mean values for each cluster obtained for the analysis with *eps* = 0.4 and *min_samples* = 5. Four clusters were obtained for this set of parameters. Note that the group of customers labeled with -1 are considered to be ”noisy” or outliers, being customers that are not assigned to any cluster.

Table 1: Percentage of customers within each customer segment (relatively to the total number of customers in the data set) after concatenating the clusters resulting from the ”customer characteristics” and ”quality of service” perspectives. Note that this was the most frequent output from our simulations, but these percentages may vary when running the code. The main focus with this is properly establishing the segments, which did not vary in any simulation.

		Customer characteristics			
		Business Single	Leisure Double	Business Double	
Quality of service	No churn despite low review in Rating 1	18,16%	6,83%	2,74%	27,74%
	Churn with low reviews in Ratings 1 and 2 (improve Rating 2)	7,40%	7,89%	6,66%	21,95%
	No churn with good reviews in all ratings	16,02%	8,82%	3,71%	28,54%
	Churn with low reviews in Ratings 1 and 3 (improve Rating 3)	9,39%	5,40%	6,99%	21,77%
		50,98%	28,93%	20,09%	

6.2 *k-Nearest Neighbors (KNN) Imputer*

The k-Nearest Neighbors (KNN) Imputer is a data transformation method that relies on the k-nearest neighbors algorithm, a supervised machine learning algorithm that assumes similar things exist in close proximity in order to replace the missing values in the data set. In a nutshell, it finds the k closest neighbors to the observation with missing data and imputes it based on the the non-missing values in the neighbors.

The process is initialized by the selection of the k value, in which it is good to understand that using a low value for k will increase the influence of noise, reaching less ”accurate” results; whereas choosing a high k will tend to blur local effects, which in the case of our project it was exactly what we were looking for. One way to define it is by assigning it the square-root of the total number of sample.

With k set, for every observation with a missing value in the data set, the algorithm will determine, based on a distance criterion between observations (the most commonly used is the euclidean distance), its k-nearest neighbors. Then, for the variable with missing value of that observation:

- If we are dealing with numerical data: If our variables is numerical, the algorithm calculate the mean value of the k-sample, in order to fill the missing value by a uniform locally computed mean. However, alternatively, one can choose to assign different weights to each neighbor according to its distance to the observation with the missing value, thus computing a locally weighted mean that will replace the missing value.

- If we are dealing with categorical data: If our variables is categorical, the algorithm will get the mode value of the k-sample to fill the missing data.

It might also be a good practice to apply KNN Imputer after normalizing the data, if needed, in order not to overemphasize variables that range between larger values.

6.3 *Principal Component Analysis (PCA)*

Principal Component Analysis (PCA) is a technique that seeks to explore the correlation structure of the original variables in a data set. The underlying idea behind PCA is to generate new variables, called principal components (PCs), constructed as linear combinations of the data set original variables, representing a smaller number of uncorrelated variables that can summarize the information in the larger original data set. Thus, two of its main purposes are to be used as a data-reduction technique and to unveil new correlations in the original data set, being often used in exploration and pre-processing stages in both supervised and unsupervised learning projects, and to attain better computational performances when dealing with very large data sets (millions of data points).

Therefore, when performing PCA, the total variability of a data set can often be mostly accounted by the PCs. In order to estimate the PCs, it is of major importance to normalize the data so that any variable is over-emphasized during the analysis. Thereafter, knowing the correlations (covariances) between all these variables, the PCs are linear combinations of the them , such that the variances of each PC are as large as possible, and the PCs are uncorrelated.

This way, the PCA statistical procedure aims to put the maximum possible information in the first component PC_1; then, to put the possible maximum remaining information in the second component PC_2; and so on, guaranteeing that the PCs are independents, and thus leading to uncorrelated PCs. In other words, in a multi-dimensional space, it finds the ”line” that best fits most of the information, by minimizing the distances from the data points to that ”line” (or, instead, by maximizing the distances from the projected data points in the ”line” to the origin). This first ”line”, representing PC_1, leans to an eigenvector, which determines the direction associated to the greatest variance of the information, and a corresponding eigenvalue, which represents the amount of variance carried in this PC. Afterwards, a second ”line”, PC_2, perpendicular (independent) to PC_1 is computed, as well as their eigenvector and eigenvalue. The process is repeated in order to generate as many PCs as desired.

For instance, each column in the component/correlation matrixes analysed in section 4 represent a PC (eigenvector). On their turn, each component of the matrix corresponds to the weight (component of the eigenvector) that a given variable has when determining that PC, then indicate a partial correlation between a given variable and the corresponding PC. Finally, the proportion of the total variance explained by a given PC is given by the ratio of its eigenvalue to the number of original variables in the data set.

6.4 *Density Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm*

The Density Based Spatial Clustering of Applications with Noise (DBSCAN) is a density-based clustering method, meaning it interprets clusters as areas of high density separated from each other by regions of low density. Therefore, unlike partition-based methods (as KMeans) or hierarchical clustering algorithms, DBSCAN is able to generate clusters of any shape and is robust enough not to be sensitive to the presence of outliers.

The underlying idea supporting DBSCAN is that belongs to a cluster if it close to many points within that cluster. This way, there are two main parameters of DBSCAN:

- *eps*: the maximum distance for two points to be considered neighbors.
- *min_samples*: The minimum number of points clustered together for a region to be considered dense.

Based on these parameters, each data point in the data set can be classified as:

- Core point, if there are at least *min_samples* points in its surrounding area with radius *eps*.
- Border point, if it is reachable from a core point, but is not classified as a core point.
- Outlier, it is not a core point nor reachable from any core points.

The concept of core points is thus crucial to the algorithmic steps of the DBSCAN algorithm. Broadly:

- DBSCAN proceeds by arbitrarily picking up a point in the data set (until all points have been visited). This point will be the starting point for generating a given cluster.
- After identifying which points are core and border points, a cluster is a set of core points that can be built and expanded by recursively taking a core point, finding all neighbors that are also core points, and then finding their neighboring core points, and so on. Border points will delimit the cluster's shape and outliers will not be assigned to any cluster, being part of a low density region. If a border points is already assigned to a cluster, it will not be assigned to another cluster.

Thus, there is no need of defining the number of clusters *a priori*. The greater the *min_samples* parameter and the lower the *eps* parameter are, a higher density is necessary to form a cluster. Such parameters should be tuned in order to meet each problem's needs and generate meaningful results, even though DBSCAN is somewhat more arduous to tune.