NOVA IMS Information Management School

MAA

Mestrado em Métodos Analíticos Avançados
Master Program in Advanced Analytics

Identifying clients' bad experiences with their internet service

Susana Margarida Silva Ferreira Lavado

Internship report presented as partial requirement for obtaining the Master's degree in Advanced Analytics

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

NOVA Information Management School

Instituto Superior de Estatística e Gestão de Informação

Universidade Nova de Lisboa

IDENTIFYING CLIENTS' BAD EXPERIENCES WITH THEIR INTERNET SERVICE

by

Susana Lavado

Internship report presented as partial requirement for obtaining the Master's degree in Advanced Analytics

Advisor / Co Advisor: Leonardo Vanneschi

Co Advisor*:* Sabina Zejnilovic

October 2018

ACKNOWLEDGEMENTS

Professor Leonardo, thank you for all your help during my internship and for believing in me from day zero when I applied to this master program. Your passion for machine learning and teaching is inspiring, and you gave me the confidence to become a data scientist.

Sabina, I will never forget my first interview with you for this internship. I remember being at the same time overwhelmed and really excited, which is what tells you have a great challenge ahead. Since that interview, I knew I would learn immensely from you and I was not mistaken. You were the best advisor I could hope for. Thank you for the guidance, the patience and the occasional chocolate!

Kevin, usually internships are little rollercoasters: we learn a lot, make a ton of mistakes and get frustrated occasionally, but always have fun. Thank you for amplifying all the good moments and making the less good moments easier. I had a blast sharing these nine months with you, and I am very grateful for all you taught me.

Sofia, I want to be just like you when I grow up! Only you combine that super smart brain with that giant heart (and the looks too!). Can we please, please, please do one more advanced analytics project together? Thank you for always having my back and being an awesome friend. And thanks for the patience to bear with me and David when we mispronounce famous mathematicians' names!

David, apparently you survived the craziness that was working with Sofia and I on almost all advanced analytics projects, which I am impressed by. You were the calmness of the group and I can still hear you assuring us that it would be ok in the end. I loved working with you and I'll try to keep your chillness with me for life.

Pedro, this is the third thesis where you are in the acknowledgements. You should have learned by now not to support me in this crazy path of mine, but I am so grateful that you have not. I promise not to do a fourth thesis anytime soon!

ABSTRACT

Identifying clients who had experienced a bad internet service is important for network providers, as bad service experiences may lead to less client satisfaction. It is possible to measure quality of service by looking at objective network quality measures. However, a decrease in the quality of service will not translate into a bad quality of experience for all clients at all times. This is because a) if the client does not try to use the internet, he or she would not notice the deterioration of the service; and b) different clients have different needs in terms of service quality; a slight decrease in network quality maybe be noticed by an intensive user but not by a light user, even if the latter is using the internet. In the present report, we describe the work we have done to develop: a) a segmentation the clients according to their typical internet usage; b) a probability that a given client would use the internet at a given time. These two features were then fed to a classifier, along with the objective network quality measures. This classifier, a gradient boosted model, was able to classify clients who filled a service request due to lack of access to the internet with an accuracy of 0.98, sensitivity of 0.87 and specificity of 0.98. The results of the classifier and the role of the special features we developed is discussed, along with future directions for this work.

KEYWORDS

Quality of experience; Internet service; Gradient boosted models; Clustering; Internet usage

RESUMO

A identificação dos clientes que tiveram uma má experiência de serviço é importante para as empresas de comunicações, uma vez que uma má experiência pode levar a menor satisfação dos clientes. É possível medir a qualidade do serviço através da análise das medidas objetivas de qualidade de rede. No entanto, uma diminuição da qualidade do serviço não se traduz numa má experiência de utilização para todos os clientes e em todos os momentos, por dois motivos: a) se o cliente não tenta utilizar a internet, ele/ela não se apercebe que houve uma deterioração do serviço; e b) diferentes clientes têm necessidades diferentes em termos de qualidade do serviço; uma deterioração ligeira na qualidade da rede de internet pode ser detetada por um cliente que usa o serviço intensamente, mas não por um cliente que usa a internet para tarefas menos exigentes, mesmo que este último esteja a usar a internet. Neste relatório, descrevemos o desenvolvimento de a) uma segmentação de clientes de acordo com o seu uso típico da internet; b) uma probabilidade de o cliente utilizar a internet numa dada hora. Estes dois atributos foram depois utilizados num algoritmo de classificação, em conjunto com medidas objetivas de qualidade de rede. Este algoritmo de classificação, um *gradient boosted model*, foi capaz de classificar clientes que fizeram um pedido de apoio técnico devido a falha no acesso à internet com uma taxa de acerto de 98% (*sensitivity* = 0.87, *specificity* = 0.98). Os resultados do classificador e o papel dos atributos desenvolvidos são discutidos, assim como futuras direções para o trabalho.

PALAVRAS-CHAVE

Qualidade de experiência; Serviço de internet; Gradient boosted models; Análise de clusters; Utilização da internet

Contents

LIST OF FIGURES

LIST OF TABLES

LIST OF EQUATIONS

1. INTRODUCTION

1.1. PROJECT DESCRIPTION

During the second year of the Nova University master program in Advanced Analytics, I enrolled in a 9-month internship at a network provider company. The present report summarizes some of the analytic activities developed during that period.

During the internship, I was part of a data science team mainly focused on understanding the factors related to customer's experiences with the provided services. My work focused on identifying clients who may have had a bad experience the internet. More specifically, we tried to identify the customers' who had no internet access.

The most obvious way to identify clients who may have had a bad experience with a service is to assess the service quality. In the case of the internet service, service quality is measured using data from the network monitoring systems. The company has different monitoring systems implemented, which register data from the quality of the service at different points in the network, and we had access to data from three of those monitoring systems. By combining the measures of these three systems, it may be possible to identify clients whose internet equipment may have had problems connecting to the internet.

However, a deterioration of the service may not always translate into a bad experience. Several other factors may affect the relationship between the quality of the service and the quality of experience, and some deteriorations of service may not be detected by clients (Fiedler, Hossfeld, & Tran-Gia, 2010). While the former is objective, in the sense that it is based on actual measurements of the network, the latter is subjective, depending on individual characteristics and behavior of the client.

The first aspect that may influence the translation of the quality of the service into the quality of experience is the way the client typically uses the service. Clients that use the internet for tasks that require fast connections with no transmission errors, such as calls or online games, may be more demanding than clients that use the internet mainly for emailing or browsing. Thus, the same service may translate as acceptable for some clients and unacceptable for others, depending on their usage.

Still, there will be times when even a frequent user is not using the internet. If the internet signal degradation occurs at a time when the client did not use or attempted to use the internet, he or she will not notice the decrease in the quality of the service. This means that the client would not have a bad experience, no matter how bad the quality of the service is. Thus, usage of the service is a second aspect that influences the relationship between the quality of the service and the quality of the experience.

In the work reported in this document, we aimed to combine the objective parameters related to network quality of service with variables related to the usage of that service by the clients, to predict bad experience with the internet service. Ultimately, we aimed to identify a subset of clients who had multiple bad experiences with their internet access and may be dissatisfied with the internet service. Dissatisfaction with the internet service is one of the main key drivers of customer satisfaction identified by the company (internal communication); therefore, addressing the issues affecting satisfaction with the internet service is one of the companies' priorities.

1.2. BUSINESS CONTEXTUALIZATION

The work presented in this report was developed in a network provider company. The company provides television, fixed voice, fixed internet, mobile voice and mobile internet services. Clients can subscribe a single service or a bundle; usually, the latter options offers better prices and different bundles are available to tailor to different needs. Clients can also subscribe, rent or buy extra TV content (e.g., video on demand, premium channels), rent equipment, and or pay for communication and internet traffic not contained in their subscription.

Each service can be provided with different settings – for example, an internet client can have different internet speeds and there are TV equipment with different functionalities. Importantly, the service can be provided by a direct to home technology (satellite service) or by cable and/or fiber.

On the present report, we will focus on all internet clients that receive their service through a hybrid cable and fiber connection, which are the majority of the internet clients of the company.

The data used on the work described on this report was anonymized. We could not track any data to specific people, nor did we have access to any personal or sensitive information of the clients.

1.3. STRUCTURE OF THE PRESENT REPORT

In next chapter (*Chapter 2 - theoretical framework*), we will present a summary of the literature supporting our work. We will briefly review the available literature on quality of service and quality of experience and on segmentation of internet usage. We will also present a theoretical summary of the methodology of the algorithms used on this work (k-means and gradient boosted trees).

The empirical part of the work is described in chapters 3, 4 and 5. Because the empirical work required the application of different procedures and algorithms, we opted to present its methodological details within each chapter, instead of creating a separate methods chapter. In our perspective, this makes the report clearer and easier to read.

First, we started by segmenting clients based on their typical internet usage. While no information was available about what clients typically do online, we were able to look at the typical intensity and frequency of internet usage for each client, based on volume of upstream and downstream traffic. This work aimed to help identifying the clients that would have higher or lower needs and demands regarding the internet. Clients with lower typical usage are expected to fill less service requests for technical reasons, compared with clients who use the internet more frequently and intensely. If the client uses the internet service very seldom or for tasks that do not require much speed or large data streams, it is less likely that he or she would notice service degradation. Thus, a segmentation of the clients into different clusters can help identify clients who are more likely to have had a bad internet experience. This work is presented in chapter 3 (*Segmentation of internet clients according to their upstream and downstream traffic*).

In chapter 4 (*Predicting if clients would use the internet at a given time*), we present the work done in developing and validating a special feature that represents the probability that the client would try to use the internet at a given hour. Different clients are likely to use the internet at different times. For instance, a client may always use the internet after 11 p.m., while other client may typically go to sleep before that time, and a third client may only use the internet at that time in alternate days. If a problem

in the network arises at 11 p.m., the first client will most certainly notice it; the second client probably would not notice it; and the third client may or may not notice it. This is important because a deterioration in the quality of internet service only translates to a deterioration in the service experience if the client tries to use the service. To contact a client because of a deterioration of service that the client did not notice could be damageable for the company image and a nuisance for the client.

The work described in chapters 3 and 4 is complementary. In the segmentation analysis, clients are categorized clients according to their typical hourly usage. The most important variable is the volume of upstream and downstream traffic, on average. The probability of usage predicts whether the client typically uses the internet at a given hour, regardless of the volume of traffic generated. A client that streams internet content every day at a given hour will have the same score as a client who simply checks his or her email every day at that same hour. While the segmentation analysis helps us understand what the internet needs of the client are, the probability of usage helps us understand if the client would have noticed an eventual degradation of service.

In the last empirical chapter (*chapter 5 – Predicting clients with no access to the internet*), we describe the work we did developing and evaluating a classifier to identify clients who did not have access to the internet at a given time. In this model, we combined information about the network signals on three different network-monitoring systems with data indicating the segment the client belonged to and his or her probability of trying to use the internet at a given time (the work described in chapters 3 and 4, respectively).

The last two chapters are dedicated to a reflection on the strengths and limitations of our results. In chapter 6, we provide our conclusions about the work developed in the context of the internship, summarizing the main findings and their impact for the business. Finally, in chapter 7, we present the limitations of the work and suggest future work paths.

2. THEORETICAL FRAMEWORK

In this chapter, we aimed to give an overview of the theoretical background that supports the current report. To organize the chapter better, we divided its content into two main parts. In the first part, we focus on the framework of the segmentation of internet users. We present an overview of related work and a theoretical summary of cluster analysis. In the second part, we focus on the framework of the identification of internet users with a bad internet experience. We present a theoretical summary of the algorithms we used (decision trees, random forests and gradient boosted trees) and of the evaluation and interpretation of classification models.

2.1. INTERNET USERS' SEGMENTATION: CLUSTERING ALGORITHM AND RELATED WORK

2.1.1. Cluster analysis

Cluster analysis aims to divide the dataset into clusters that are meaningful. That is, clusters that capture the internal structure of data and, therefore, can adequately describe the dataset through the characterization of the clusters, rather than individual points (Tan, Steinbach, & Kumar, 2006). Using this technique, it possible to reduce a large dataset into smaller, more interpretable groups (Hand, Mannila, & Smyth, 2001).

The goal in cluster analysis is to group cases that are similar to each other in the same cluster and, at the same time, maximize the difference between points in different clusters (Tan et al., 2006). Usually, similarity is established using distance-based measures, such as the Euclidean or the Manhattan distance, or correlation coefficients.

Clustering analysis is an unsupervised technique, meaning that there is no ground-truth to which the results of the analysis can be compared (Hand et al., 2001). Thus, evaluating the performance of the analysis is a little more challenging. Typically, the algorithm is evaluated through measures such as intra-cluster distance (the distance to a point to every point in the same cluster) and inter-cluster distance (the distance between a point and the closest point that belongs to a different cluster. Clusters can also be evaluated in terms of their external validation, that is, how meaningful they are in predicting variables that were not included in the initial cluster analysis; how stable they are in different time-points; and more subjective measures such as how the results match the users' domain-knowledge and the purpose of the analysis.

Clustering algorithms can be classified into hierarchical methods and partition methods. In hierarchical methods, clusters are nested into each other, and can be viewed as a hierarchical tree. In partition methods, each point is assigned to a single, non-overlapping cluster.

In the work described in this report, we mainly used k-means, an algorithm that follows a partition method. K-means is one of the oldest and most common clustering algorithms, due to its simplicity and ability to handle large datasets (Tan, Steinbach & Kumar., 2006).

2.1.2. The K-Means algorithm

The first step in k-means is to randomly select k points in the dataset. The position of those points in the space will be the initial centers of the clusters, called centroids. Then, the following steps are completed:

1. For every point in the dataset, compute the distance between that point and each of the previously defined centroids.
2. Attributed the point to the closest centroid.
3. When all points have been attributed, recalculate each centroid, by averaging all the points that have been attributed to that cluster.
4. Calculate the distance between each new centroid and the centroid it is replacing. If the distance is bigger than a user-defined constant, go back to step one. If not, the algorithm terminates.

One of its biggest challenges in k-means is the definition of the number of clusters, k, beforehand. To define the number of clusters, several approaches can be followed, which include operational concerns (e.g., making sure that the number of clusters is actionable) and more objective metrics. Below, we present three of the most common metrics that help to select a k value:

- **Within-cluster sum of squares (WSS):** WSS is a cohesion measure. It represents the sum of the distance of each point to the centroid of the cluster they were assigned to, summed across all clusters. It is calculated through the following formula:

$$\sum_{k=1}^{k} W(C_k) = \sum_{k=1}^{k} \sum_{x_i \in C_k} (x_i - \mu_k)^2$$

Equation 2.1 – Within-cluster sum of squares

Where C_k is a cluster resulting from the analysis, $W(C_k)$ is the within-cluster sum of squares of cluster C_k, x_i is a point belonging to cluster C_k, and μ_k is the centroid of cluster C_k, corresponding to the average of all points in the cluster.

Increasing k typically reduces WSS, to the point where WSS is zero (each data point is in its own cluster). It is important to balance the benefit in the increase of cohesion and the cost of adding one more cluster to the solution. This is usually done by plotting the WSS and observing where the line starts to plateau (the "elbow").

- **Average silhouette:** silhouette is a measure of the similarity of a data point to its own cluster (cohesion), compared to its similarity to other clusters (separation). The silhouette can be calculated with any distance measure (e.g., Euclidean, Manhattan). It produces values between -1 and 1 and the highest the value, the better the clustering solution (Rousseeuw, 1987). For each cluster, it is calculated through the following formula:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i),\ b(i)\}}$$

Equation 2.2 - Silhouette

Where a(i) is the average distance between a data point i and all points assigned to the same cluster as i, and b(i) is the lowest average distance between i and all points in any other cluster to which i was not assign to (Rousseeuw, 1987). The silhouette value across all clusters is calculated by averaging the silhouette values of all clusters.

- **Gap statistic:** the gap statistic was developed by Tibshirani, Walther, and Hastie (2001). It compares the logarithm of the obtained intra-cluster variation[1] $\log(W_k)$ with the expected logarithm of the intra-cluster variation of a reference null distribution ($E_n^* \log(W_k)$), a uniform distribution with no obvious clustering structure. The bigger the gap statistic, the better the clustering solution. We present the formula for the gap statistic below.

$$Gap_n(k) = E_n^* \log(W_k) - \log(W_k)$$

Equation 2.3 – Gap statistic

Typically, the user rans several iterations of the algorithm with different values of k, computing one or more of these measures for each k. Then, the k that leads to the best metric(s) is selected. These metrics can also be combined with more subjective evaluations of cluster interpretability, cluster size and even total number of clusters, if the main goal of the analysis is to define actionable and interpretable clusters.

K-means results may also vary a lot depending on the initial seeds, which are usually selected at random. Therefore, it is recommended that the algorithm be ran several times, with different initializations seeds. Since it is a distance-based algorithm, outliers have a lot of influence on k-means results.

Since the attribution of points to the cluster is based on distance, k-means assumes that clusters are spherical and has trouble identifying clusters with different shapes.

2.1.3. Related work on Internet users' segmentation:

Several authors have tried to segment people according to their internet usage. However, most studies have used subjective measures of users, that is, they have conducted surveys that ask participants what they do online and with what frequency.

For instance, Ortega Egea, Recio Menéndez, and Román González (2007) used survey data on how frequently people accessed the internet and how frequently they engaged in activities such as online shopping and usage of eGovernment services. They used a two-step cluster algorithm, which first pre-clusters the datapoint into many small clusters (using a modified cluster feature tree), and then re-cluster these clusters (using an agglomerative hierarchical clustering method), in order to be scalable and able to handle large datasets. The users were clustered in five segments, ranging from non-users (44% of the sample) to advanced users (19% of the sample). They then used demographic variables to perform external validation of the clusters and did discriminant analysis of the clusters (discriminant analysis uses the same variables used for clustering to predict the cluster to which each data point was attributed).

Brandtzæga, Heim, and Karahasanovic (2011) also clustered people according to their self-reported online behavior but included more questions regarding the frequency of specific online activities. They used k-means to identify five segments of users, according to frequency of internet usage and most common online activity: non-uses, sporadic users, entertainment users, instrumental users, and advanced users. To validate the clusters, they performed a logistic regression predicting the

[1] Intra-cluster variation is the average of the distances between a point to all the other points in the same cluster, summed across clusters.

membership to a given clusters using variables such as age, gender and number of people in the household.

However, the most relevant study for our work is the analysis developed by Oliveira, Valadas, Pacheco, and Salvador (2007) and by Kihl, Lagerstedt, Aurelius, and Ödling (2010), who used objective variables to segment users.

Oliveira et al. (2007) segmented internet users according to their download transfer rate, measured every half-hour on a single day. They applied a principal components analysis to the measures of transfer rate for each half-hour and extracted two factors. The first factor was an average of the utilization throughout the day, and the second factor corresponded to the difference between the morning and the afternoon usage. Then, they applied an agglomerative hierarchical clustering method to the data, which they separately combined with two methods to decide which clusters to merge: The Ward method and the partitioning around medoids method. The two methods gave similar results, which consisted of three clusters:

- Cluster 1: High transfer rate in all periods.
- Cluster 2: Low transfer rate in the morning, high transfer rate in the afternoon.
- Cluster 3: Low transfer rate in all periods.

The authors used discriminant analysis to validate the obtained clusters. In addition, the clusters were also externally validated by checking the most used applications in each cluster (e.g., file sharing, HTTP, Games).

Kihl et al. (2010) clustered users according to their average daily inbound and outbound traffic, and the number of applications used over one month (without further classifying the applications according to their type). They identified three clusters:

- Cluster 1: Lower inbound and outbound traffic, low number of applications used;
- Cluster 2: Moderate inbound and outbound traffic, moderate number of applications used;
- Cluster 3: Higher inbound and outbound traffic, higher number of applications used.

It is noteworthy that 80% of the users were on cluster 2, the cluster with moderate usage. The authors did not perform any further validation analysis of the clusters, nor did they specify the clustering algorithm used.

2.2. IDENTIFYING CLIENTS WITH A BAD INTERNET EXPERIENCE: DECISION TREES, RANDOM FORESTS AND GRADIENT BOOSTED TREES ALGORITHMS

Machine learning algorithms are generally classified as supervised or unsupervised learning algorithms. Unsupervised learning algorithms are unsupervised in the sense that there is no ground truth they can "learn". For instance, the clustering algorithms described in the previous section are unsupervised, because they aim to extract patterns from the data without those patterns being presented to them beforehand. Conversely, supervised learning algorithms are algorithms that have a ground truth – they aim to "learn" that ground truth and then apply it to new data. In other words, the algorithm is told what a given pattern looks like and then it tries to apply that pattern to novel situations.

One typical supervised learning problem is a classification problem. In these cases, the algorithm is presented with a dataset composed of a set of attributes and a target variable, which correspond to a

discrete class. The algorithm tries to learn which attributes correspond to which classification, to then classify new instances in one of the available classes. If there are only two classes available, it is a binary classification problem; if there are more than two classes in the dataset, it is a multi-classification problem.

Thus, to learn, the algorithm needs to be presented with some instances of data (a training set). Then, the algorithm performance is tested in new instances, which were not used to train the algorithm (the test set). This allows us to understand whether the patterns learned by the algorithm are generalizable outside the training set or if, instead, the algorithm over-fitted to the data. Over-fitting means that the algorithm learned the particularities of a given dataset too-well, in the sense that it learned patterns that are only present in that dataset (i.e., it learned noise in the data).

To test for the ability of the algorithm to generalize, two methods are generally used. The first method consists in dividing the dataset into two partitions: a training set and a test set. This is faster, but it may be biased depending on which instances end up in the training and testing sets. For instance, if the dataset is small, the test set may not be representative of the entire dataset, and the algorithm could perform much better (or much worse) on that particular subset of cases. Another way to validate the generalizability of the classifier is k-fold cross-validation. In k-fold cross-validation, the dataset is partitioned into k folds. Then, the k-1 folds are used to train the model, while the one partition is used to test the model. The procedure is then repeated until all folds were used to test the model, as represented in the following schema (Terribile, 2017):

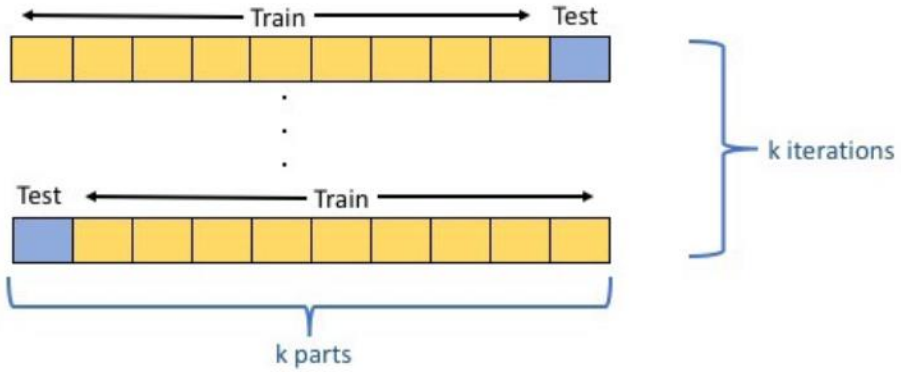


Figure 2.1 – K-fold cross-validation schema

The performance of a given algorithm corresponds to the average of the performance in each of the test folds, also taking standard deviation into consideration (because a classifier can perform very well on some folds but poorly on others, which would lead to a high standard deviation; an ideal classifier would perform well across folds). This allows the user to have more confidence that the model would generalize well on different subsets of the data.

In the remaining of this section, we will present some theoretical background on tree-based classifiers: decision-trees, random forests and gradient boosted trees. These were the algorithms we used on the work we are presenting in this report, for interpretability (in the case of the decision trees) and performance reasons (in the case of random forests and gradient boosted trees models).

2.2.1. Decision trees

Decision trees are popular algorithms that can be used for classification (binary and multiclass) and regression problems. Decision trees can be conceptualized as a set of if-then rules that classify each data point according to its attributes. Each node of the tree is a rule that indicates the "path" on the tree a data point must follow, depending on the value the data point has on a given attribute. Ultimately, each data point will end on a "leaf", that is, a final node of the tree, which has no children, and will be classified according to the value on that leaf. Below, we reproduce a classic example of a tree, to help visualize the algorithm.

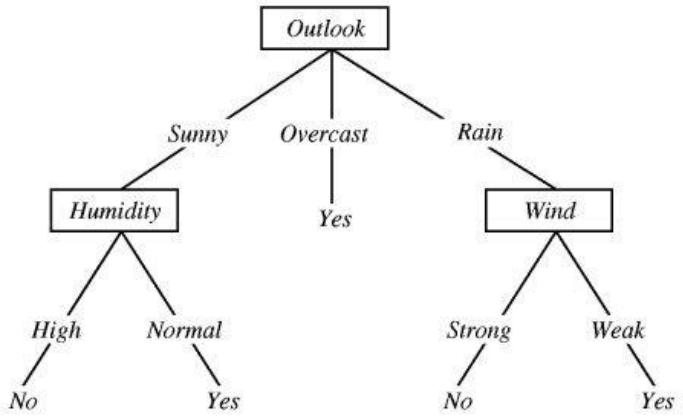


Figure 2.2 – The "play tennis" decision tree.[2]

The value of the leaf is defined based on the proportion of train cases that ended up on that leaf and belong to each class (if it is a classification tree) or based on the average target-value of the train cases that ended up on that leaf (for regression trees).

Decision trees are built through recursive partitioning, meaning that the data space is sequentially partitioned into smaller spaces, with increasingly higher homogeneity and to which simpler models can be applied (Bishop, 2006). Depending on the decision tree algorithm, those splits can be binary, meaning that only two branches are generated at each split (as is the case with, for example, the CART algorithm) or have more branches (as is the case of the ID3 algorithm).

However, a typical dataset has many variables, and each variable is a candidate for a given node. We need some criteria to select the variable (and the cut-off points on that variable) that will be used in each partition. The most traditional tree-based classification algorithms (e.g., ID3, CART) are greedy, in the sense that the variable selected is the one that maximizes the information gain at that split, or, equivalently, minimizes the error of the model at that split. These measures capture the increase in "purity" of the resulting node, that is, the homogeneity of classes in that node. Typical measures used to measure information gain/error reduction in classification trees are entropy and the Gini index, which are based on the proportion of cases of each class in the resulting nodes. The formula for information gain is:

[2] Obtained from https://nullpointerexception1.wordpress.com/2017/12/16/a-tutorial-to-understand-decision-tree-id3-learning-algorithm/

$$Gain(S, A) \equiv Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

Equation 2.4 – Information gain

Where S is the population of a given node, A is a given attribute with v possible values and S_v is the subset of S for which attribute A has value v.

Entropy for an attribute that has c possible values can be calculated using:

$$Entropy(S) \equiv \sum_{i=1}^{c} -p_i \log_2 p_i$$

Equation 2.5 - Entropy

Where p_i is the proportion of instances in the node S that have a given value i for the attribute.

Similarly, to entropy, Gini is also an impurity measure. If Gini index is the criteria selected for selecting the variable for a given split, the reduction in the overall Gini index is considered. Gini index can be calculated using the following formula, where p_j is the proportion of cases in each node:

$$\phi(p) = \sum_{j} p_j (1 - p_j)$$

Equation 2.6 – Gini index

The algorithm keeps producing splits until some stopping criteria is reached. That stopping criteria can be defined in different ways. It can be:

- A limit in the maximum depth of the tree, that is, in the number of edges from the lowest node to the tree's root node;
- The number of observations in a leaf node for an extra split to be attempted;
- The minimum number of observations on each leaf;
- A complexity parameter that defines how much error the split needs to reduce to be considered;

These stopping criteria prevent the tree overfitting the training data. If the tree was allowed to grow indefinitely, it would eventually get a perfect classification of the training data, as it would cover all possible cases. However, this would typically mean that it would not generalize well to unseen data, because it had learned how to identify specific cases instead of more general rules of the dataset that are applicable to other cases.

Alternatively, some decision tree algorithms allow the tree to grow very large and, posteriorly, prune the tree, i.e., remove sections that add little value to the model (which is evaluated through a validation set). This technique also prevents overfitting and has the advantage of not forcing the user to estimate when the tree should stop growing.

2.2.2. Random forests

As the name suggests, random forests are groups of decision trees. In other words, they are an ensemble method, which rely in developing several (usually hundreds) of decision trees and then combining the results of all those trees. The results can be combined in several ways, such as voting or taking the average of the predictions of all trees. We note, however, that ensembles do not need to consist of tree-base algorithms (or any single type of algorithm for that matter) – any type of classifier can be part of an ensemble, which usually outperforms the individual classifiers in the ensemble.

Each tree is built on a random subset of the data, which are usually drawn with replacement (meaning that each datapoint can be included in more than one tree), a technique called bagging. This could lead to much higher accuracy of the model, but this is only true if the classifier is unstable, that is, if its output varies a lot with small changes in the input data (see Breiman, 1996).

In addition, it is common that each tree in a random forest is built using only a subset of the features in the dataset, to increase diversity in the trees. The number of variables randomly selected for each tree is user-defined. In addition, the user may control all the parameters of each decision tree, as described in the above section.

2.2.3. Gradient Boosting Models

Like random forests, gradient boosted models are algorithms based on the ensemble of several weak classifiers to produce a better classifier. Typically, those classifiers are also decision trees (but can be any other algorithm) and are considered weak in the sense that their individual performance is only slightly better than chance. However, while in random forests the trees are built independently from one another, in boosted models the trees are built sequentially. Each additional classifier attempts to correct the classification errors of the previous classifier, by attributing more weight to the previously misclassified cases. Data points that are misclassified by successive classifiers receive an increasingly greater weight. After the training of all classifiers, the prediction is derived through a weighted majority voting scheme, where the weight given to each classifier depends on its performance (more accurate classifiers receive greater weight). This is the base of boosting algorithms such as AdaBoost (Bishop, 2006).

In gradient boosting (Friedman, 2001), a gradient descent (also called steepest-descent) function is applied, which is an optimization algorithm to find local optima. In this algorithm, a parameter vector w is calculated using the formula:

$$w^{(\tau+1)} = w^{(\tau)} - \eta \nabla E_n$$

Equation 2.7 – Gradient descent formula

Where ∇E_n is the gradient of the error function, η is a learning rate parameter and τ is the iteration number (Bishop, 2006). In the case of gradient boosted models, this means that the algorithm will converge to a local minimum in the error function.

Stochastic gradient boosting, which was proposed by Friedman (2002) mixes bagging and boosting procedures. This model, uses gradient boosting, but adds additional randomness by randomly selecting a subset of the sample at each interaction (without repetition). This means that only a subsample of

observation will be randomly selected for fitting each tree, which may improve the accuracy of the classifier, especially for small samples and high-capacity base classifiers (Friedman, 2002).

Several parameters can be tuned when applying a stochastic gradient boosted tree model. Below, we list the most common:

- Depth of each tree: what is the maximum depth that each individual tree can achieve;
- Number of trees: how many trees should be sequentially grown;
- Minimum number of observations in each node: minimum number of observations in the trees terminal nodes;
- Bag fraction: the percentage of observations that is used to fit each tree;
- Shrinkage: also called learning rate, is a weighting factor for the corrections by new trees when added to the model. It can be roughly understood as how fast the algorithm learns; a larger learning rate may mean that the model "misses" the optimum and starts to overfit; a smaller learning rate means that the model needs more step (in gradient boosted trees, more trees) to get to a certain error reduction (Laurae, 2016).

2.2.4. Evaluation of classification models

Several metrics can be used to assess a binary classification model performance. Most of these measures are based on the concept of true and false positives or negatives. True positives (TP) are cases that were classified by the model as positives and were actual positives. Accordingly, false positives are cases that were classified by the model as positives but were actual negatives. True and false negatives are cases that were classified as negative and were actual negative and positive, respectively. The distribution of cases among these classes is usually organized in a confusion matrix, which is schematized below.

		PREDICTED CLASS	
		P	N
ACTUAL CLASS	P	True positives (TP)	False negatives (FN)
	N	False positives (FN)	True negatives (TN)

Figure 2.3 – Confusion matrix schema

Based on these classes, it is possible to derive several measures that inform about the model quality. Next, we summarize some of these measures.

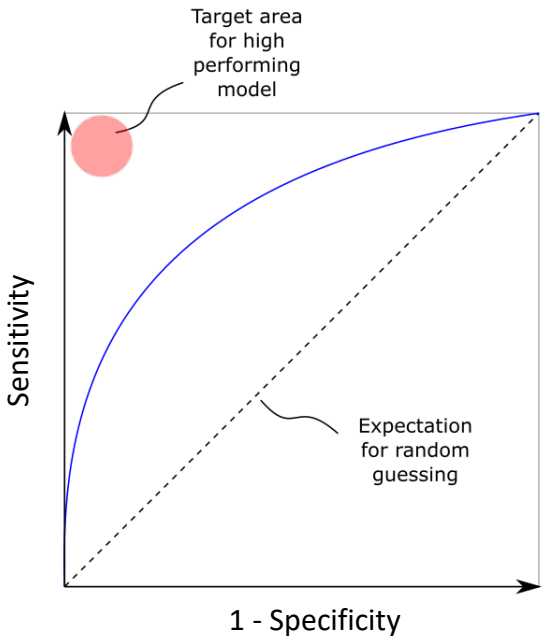
Measure	Description	Formula
Accuracy	Measures the total number of correct classifications, over the total cases	$\frac{TP + TN}{TP + TN + FP + FN}$
Sensitivity	Rate of positives cases correctly identified as positive, out of the actual positive cases. Also known as Hit rate, Recall, or True Positive Rate.	$\frac{TP}{TP + FN}$
Specificity	Rate of negative cases correctly identified as negative, out of the actual negative cases. Also known as True Negative Rate.	$\frac{TN}{TN + FP}$
Precision	Rate of true positives out of the total number of cases classified as positives.	$\frac{TP}{TP + FP}$
F1	Harmonic mean of precision and sensitivity. By combining the two measures, it allows	$\frac{2TP}{2TP + FP + FN}$

Table 2.1 – Summary of model performance measures

To compute these measures, it is necessary to establish a cutoff point for the probability attributed by the model. By default, most binary models assume a 0.5 cut-off point. If the probability is above this cut-off, the case is attributed to a class; if it is below, the case is attributed to the other class. However, the user can adjust the cut-off point, which is especially useful when the cost of mispredicting a class is higher than the cost of mispredicting the other. For instance, the cost of a false negative may be higher than the cost of a false positive, such as when failing to detect the onset of a disease is higher than the cost of sending the patient for further analysis.

Usually, the establishment of the cut-off point depends on a trade-off between sensitivity and specificity. To help define such a threshold, it is common to plot the Receiving Operator Characteristic (ROC) curve. The ROC curve plots the values of sensitivity and (1 – specificity) that would result from choosing different cutoff points.

The more the resulting curve approximates of the left corner of the plot (i.e., the more sensitivity = 1 and 1 – specificity = 0), the better the performance of the model. For comparison, it is usually also plotted the line corresponding to a random classifier. The picture below, adapted from Parkes (2018) helps understand this description better.



Besides helping to select a cut-off point (by selecting a point that maximizes a metric, without compromising the other metric too much), the ROC curve also provides a visual information about the model perform.

As an additional measure of model performance, it is also possible to calculate the Area Under the ROC Curve (AUROC). The AUROC represents the probability that a model will classify a randomly chosen positive instance higher than a randomly chosen negative one. Thus, a value of 1 for AUROC corresponds to a perfect classifier, and a value of 0.5 means that the model performance is at chance level.

2.2.5. Interpretation of classification models

When fitting a given model, it is important to understand how the predictors are related to the target value, to make sure the model learned something sensible and to help making model improvements. A frequent critic to complex machine learning algorithms such as gradient boosted models is that they are hard to visualize and interpret. To help overcome this difficulty, several techniques have been proposed.

One way to understand how variables are being used in the model is to look at the variable importance. In tree-based models, variable importance is usually based on the error reduction each variable is responsible for, considering not only the splits it is included in but also the splits to which it is one of the top candidates for the split. On boosted trees, the final variable importance is the sum of the importance in each boosting iteration (Kuhn, 2007). It is also common to scale variable importance by setting the importance of the most important variable as 100, and the importance values of other variables relative to that one.

However, variable importance does not inform about the way each input variable relates to the target variable. This limitation may be address by, for example, visualizing the target variable in relation with

the input variable in a scatter plot, to understand how certain values of the input variable are related to the target variable. But this is also not a good solution, because it does not take into consideration the effect of other variables included in the model.

Partial dependence plots (Friedman, 2001) try to overcome these challenges, by representing the relation between predictors and the target, while considering the average effect of other predictors (Greenwell, 2017). Partial dependence plots can be interpreted in a way that is similar to the coefficients in linear or logistic regression, but they can be used with any model.

After fitting the model, partial dependence plots can be generated by varying the values of a given predictor and estimating the effect on the target value or the probability of the classification, across all observations (Becker, 2017). More specifically, the pseudo-code for partial dependence plots is the following (Greenwell, 2017):

Let x_1 be the predictor variable of interest with unique values $\{x_{11,}\, x_{12,}\, \dots, x_{1k}\}$, and $\hat{f}(x)$ the prediction function.

1. For $i \in \{1, 2, \dots, k\}$:
    a. Copy the training data and replace the original values of x_1 with the constant $x_{1i.}$
    b. Compute the predicted values using the dataset resulting of step a.
    c. Compute the average prediction to obtain $\overline{f_1}\,(x_{1i})$.
2. Plot the pairs $\{x_{1i}, \overline{f_1}\,(x_{1i})\}$ for $i = 1, 2, \dots, k$.

3. SEGMENTATION OF INTERNET CLIENTS BASED ON THEIR UPSTREAM AND DOWNSTREAM TRAFFIC

Different clients may have different perceptions of the quality of the internet at a given time, even when objective parameters of that quality are the same, based on how much and for what they use the internet. However, to investigate these potentially different perceptions we need to first identify patterns of internet usage among clients. To do so, we performed a cluster analysis aiming to group clients based on the amount of uploads and downloads their modems registered per hour.

3.1. DATA SELECTION AND PREPARATION

For data selection and transformation, we used Microsoft SQL server, which is a database management system that uses the SQL language to query, transform and extract data (Microsoft, 2018).

We retrieved the data describing the hourly traffic (uploads and downloads) associated with a clients' cable modem MAC addresses (from now on, designated as MACs). We focused on non-business clients with an active HFC internet subscription in the month of May.

We note that data was anonymized, in the sense that we had only access to the MAC of each device and the account ID it was associated with, but not access to the actual name or address of the account holder. In addition, we had only access to the volume of traffic generated by each MAC; besides volume, we had no information about the online activity of the client (for instance if the internet was being used for browsing, gaming or streaming).

The data we used had a temporal granularity of an hour. After excluding null values and duplicate rows, we created the features for the clustering analysis, computing the following transformations of the traffic data:

- Average hourly rate of uploads and downloads;
- Standard deviation of the hourly rate of uploads and downloads;
- Average hourly ratio of uploads to downloads; if the average of downloads was zero, we considered this ratio to be zero as well;

Each of these transformations was performed for several date/time periods, based on business insights:

- Across the full month;
- For working and for non-working days;
- For the morning/afternoon, evening and night period;
- For working days morning/afternoon, evening and night periods;
- For non-working days morning/afternoon, evening and night periods.

The available traffic data was computed in bytes. After the calculation of averages and standard deviations, we converted the values to kilobytes.

We also computed four additional variables that represented, for each MAC, the number of hours where the traffic was higher than a given threshold (calculated separately for uploads and downloads).

These thresholds were the approximated average hourly upload traffic and the average hourly download traffic (for the full month) of the top 90% and 99% of the sample. These thresholds are depicted in the figures below.

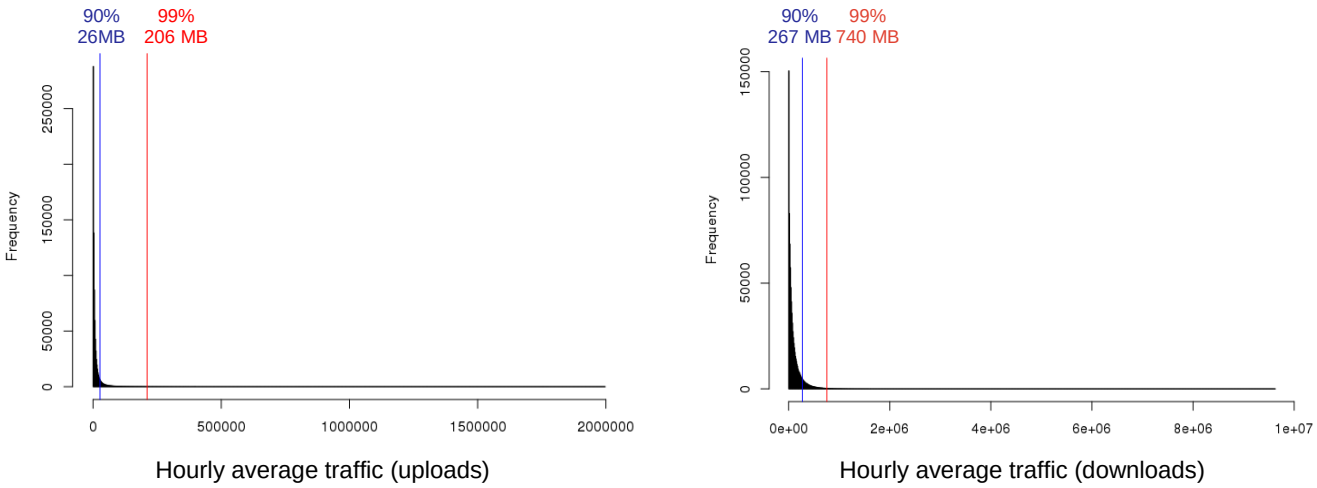


Figure 3.1a and b - Values considered for the 90 and 99 thresholds

With these thresholds, we hoped to create variables that would reflect the intensity of usage of each client, since, for example, two users can have similar hourly average traffic if they used the internet moderately for many hours or very intensely for only a couple of hours.

Finally, we created a variable measuring the number of days each device had a significant traffic, that is, the number of days the device was actively used. To calculate this variable, we first analyzed the traffic of cable modems belonging to customers who do not have an internet subscription; the cable modems of these clients are only used to make voice calls. For 92% of these devices, the average daily consumption was less than 1MB, for both variables. We considered this value as our threshold and counted the number of days where each MAC had an upload and download traffic higher than 1MB. We then divided that number by the number of days the MAC had entries.

3.2. DATA EXPLORATION

After computing the variables described above, we proceeded to the exploration of the obtained data. To do so, we used Open Database Connectivity (ODBC) to read data from SQL Server into R. ODBC is an Application Programming Interface (API) that allows access to database management systems, designed specifically for relational data stores (Milener & Guyer, 2017). R is a programming language and an open-source environment typically used in data analysis and visualization (R Project, 2018).

3.2.1. Initial analysis

We first checked for data inconsistencies (e.g. values too big or too small to be possible; null values). After verifying that the data was consistent, we counted the number of clients who had zero hourly

average consumption across the different date/time periods. We verified that these were less than 1% of the sample for all the considered variables.

3.2.2. Outliers

Regarding outliers, even though we had very high maximum values, we could not find a clear breaking point in the data that would remove enough data points to be meaningful. To illustrate this point, we present below a scatter plot relating the average hourly traffic for uploads and downloads, across the full month.

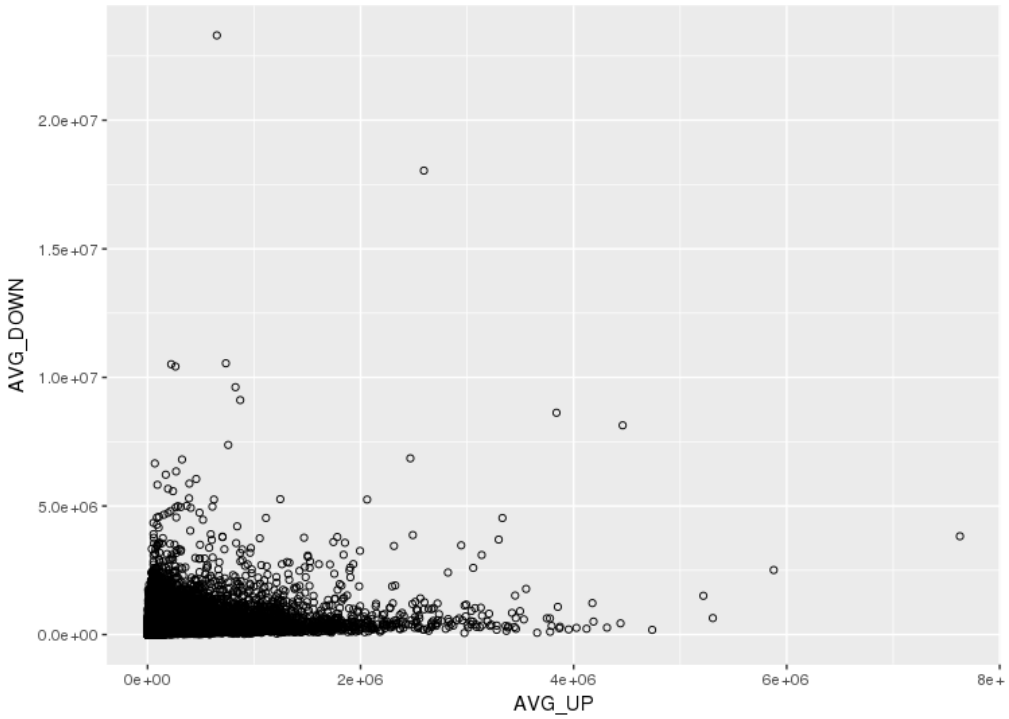


Figure 3.2 - Hourly average of up and downstream traffic, across the full month

Even though it was possible to identify cutting points that would remove perhaps around 10 outliers, it would not have much impact, considering that there over eight hundred thousand MAC addresses in the population. This finding was similar for the other traffic variables considered. Therefore, we decided to use the entire population in the analysis. This decision also addresses a business concern: each outlier was an actual client and we wanted to include all of them. In addition, high usage outliers are especially interesting from the business point of view.

3.2.3. Data distribution

The table below contain information about the average and spread of the average traffic for the full month.

	Average Uploads (Month)	Average Downloads (Month)	SD Uploads (Month)	SD Downloads (Month)
Average	15.1 MB	98.9 MB	45.0 MB	242.4 MB
Min.	0.0 MB	0.0 MB	0.0 MB	0.0 MB
Median	3.7 MB	50.2 MB	11.4 MB	134.5 MB
Max.	7447.8 MB	22754.1 MB	3826.9 MB	15723.4 MB

Table 3.1 - Average and spread measures for the hourly averages and SD of traffic across the full month, in MB.

It is possible to observe that the average, median and maximum for the hourly average of downloads was much higher than uploads. In addition, the spread of the variables was big, with very large maximum values for uploads and downloads.

The histogram of the average variables also suggested that these variables follow a long-tail distribution.

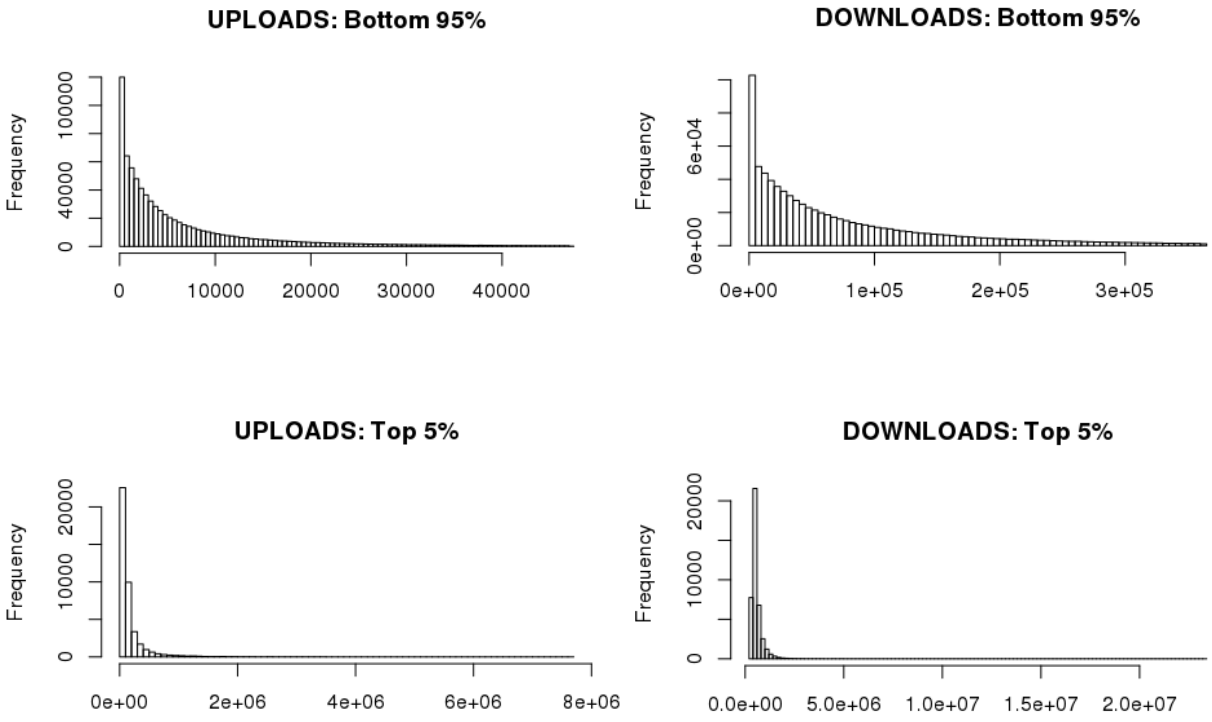


Figure 3.3a, b, c, and d - Histograms representing the distribution of the hourly average traffic for the bottom 95% and the top 5%, for uploads and downloads.

The variables measuring average traffic in specific date/time periods had similar average and spread values among themselves; the average and spread was also similar to the values of the monthly

average presented above. For the sake of simplicity, we will not present them. Traffic was higher for evenings of non-working days and lower for the day period of working days, as expected.

The following histogram presents the distribution of the upload to download ratio variable:

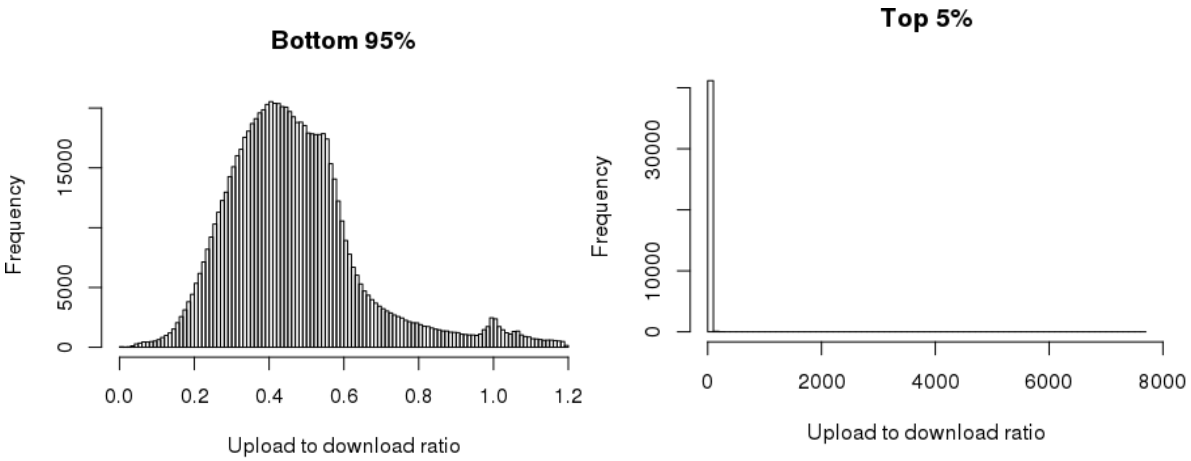


Figure 3.4a and b - Histograms representing the distribution of the upload to download ratio for the bottom 95% and the top 5%, for uploads and downloads.

As it is visible from the right-hand plot, the distribution had a very long tail (up to 8000 uploads to downloads), but most clients had a much smaller upload to download ratio. 92% of the sample has an average upload to download ratio for the full month lower than 1, and 99% of the sample has a ratio lower than 4.5.

Finally, we analyzed the distribution of the threshold variables. We started by looking at the number of days each user had active internet usage, defined as the number of days where uploads and downloads were higher than 1MB.

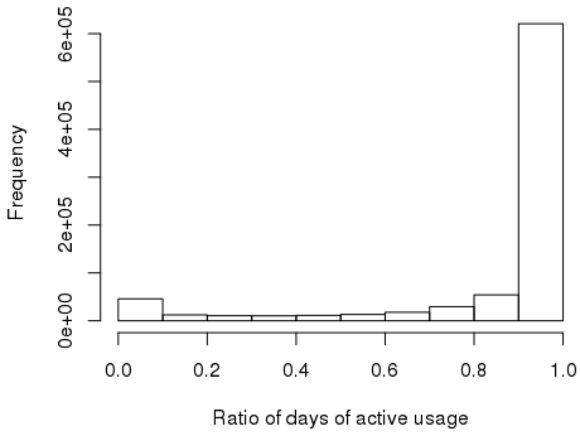


Figure 3.5 - Histogram of the variable representing the ratio of number of days with active internet usage.

63% of the population had active internet usage for all the days there were entries available; 89% of the population had active usage for at least half of the days.

We present the histograms representing the distribution of the hourly threshold variables below. Recall that these variables indicated the number of hours each user generated traffic above the average hourly traffic of 90% and 99% of the population.

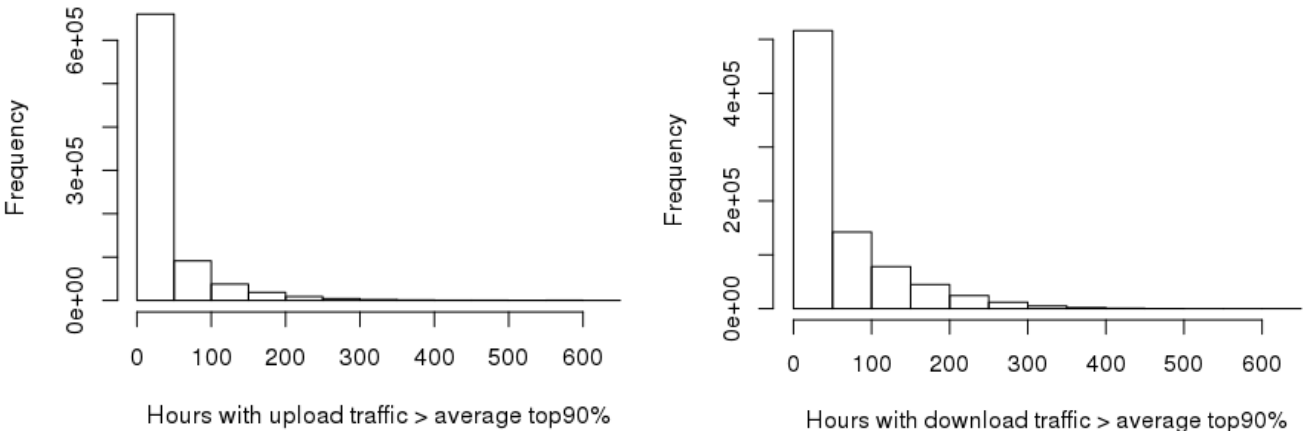


Figure 3.6a and b - Histogram of the variables representing the number of hours with traffic higher than the average traffic of 90% of the sample, for uploads and downloads, respectively.

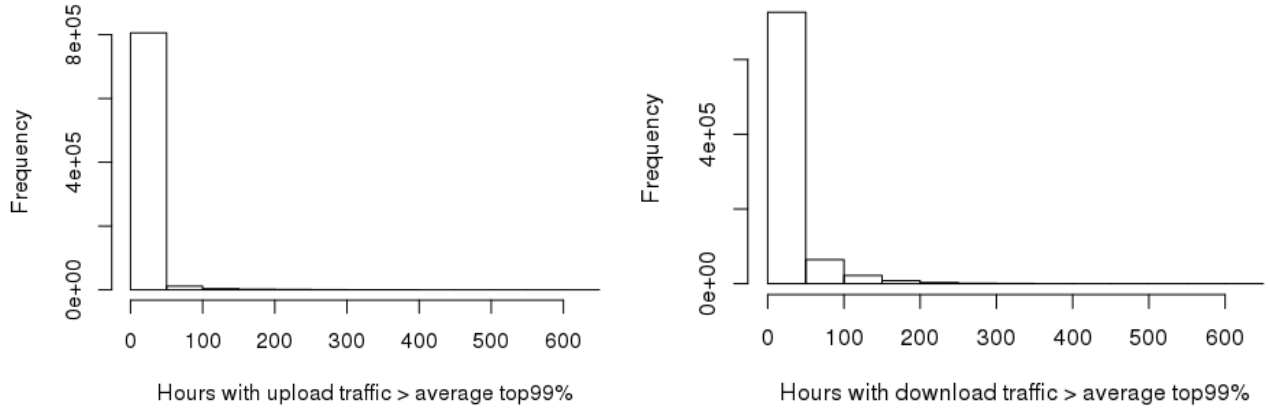


Figure 3.7a and b - Histogram of the variables representing the number of hours with traffic higher than the average traffic of 99% of the sample, for uploads and downloads, respectively.

We computed two additional variables, representing the difference between the average traffic on working days and the average traffic on non-working days, for uploads and downloads. To calculate these variables, we subtracted the average hourly traffic on working days from the average hourly traffic on non-working days.

As shown in the following histograms, many users have similar traffic on working and non-working days:

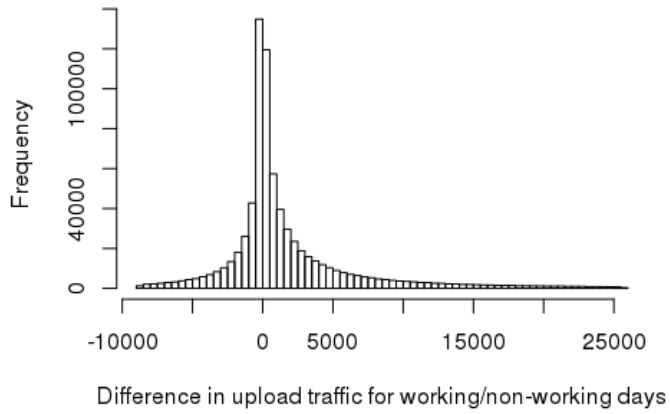


Figure 3.8 - Histogram representing the difference in upload traffic for working and non-working days, for the middle 90% of the sample

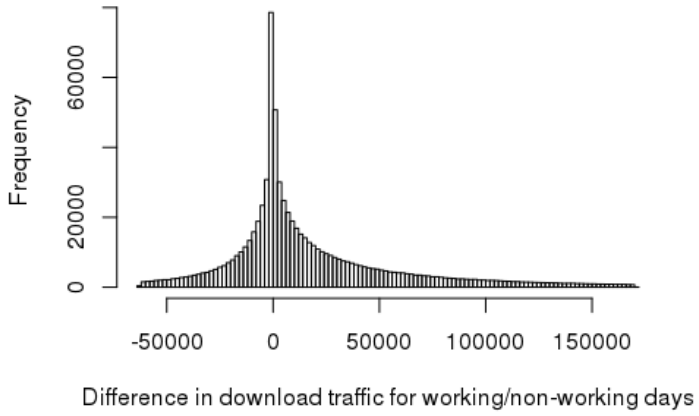


Figure 3.9 - Histogram representing the difference in download traffic for working and non-working days, for the middle 90% of the sample

There are more users on the right side of the distribution, and the right tail of the distribution is larger, for both histograms. This means that more user traffic is larger on non-working days than working days.

3.2.4. Correlation among variables

For clustering, to include variables that are highly correlated may not only add unnecessary complexity to the analysis (because each cluster will behave very similarly in variables with high collinearity, which means that adding more variables does not help to differentiate the groups in the analysis) but also bias the results, particularly on distance-based clustering algorithms. Adding many variables that are highly correlated skews the analysis towards these variables, giving more importance to what these variables are measuring than to other aspects that can be equally important in the analysis. This tends to produce clusters who have only different average values in the highly correlated variables, leading to less interesting segments (Sambandam, 2003).

Through a correlation analysis, we verified that most of the variables related to average traffic are highly correlated among themselves. The average and the standard deviation of the hourly traffic for the same date/time period are highly correlated for all date/time periods considered ($r > .70$). Similarly, the average traffic per hour of the different date/time periods is also correlated for uploads ($r > .70$) and downloads ($r > .53$), as showed in the tables below.

UPLOADS	WD, Day	NWD, Day	WD, Eve	NWD, Eve	WD, Nig	NWD, Nig
WD, Day	1					
NWD, Day	0.83	1				
WD, Eve	0.85	0.80	1			
NWD, Eve	0.70	0.84	0.83	1		
WD, Nig	0.80	0.78	0.85	0.79	1	
NWD, Nig	0.72	0.82	0.79	0.83	0.85	1

Table 3.2 - Correlation among the hourly average uploads for different date/time periods

DOWNLOADS	WD, Day	NWD, Day	WD, Eve	NWD, Eve	WD, Nig	NWD, Nig
WD, Day	1					
NWD, Day	0.67	1				
WD, Eve	0.71	0.72	1			
NWD, Eve	0.57	0.74	0.77	1		
WD, Nig	0.62	0.60	0.73	0.67	1	
NWD, Nig	0.53	0.64	0.69	0.71	0.75	1

Table 3.3 - Correlation among the hourly average downloads for different date/time periods

This analysis suggests that clients' internet usage is similar (in terms of volume) across the different time-periods considered (different times of the day and working/non-working days). Clients' who generate relatively more average traffic at a given time period also tend to generate relatively more average traffic at a different time period, and the same for clients who generate relatively less traffic.

However, the average hourly traffic variables are less correlated with other variables such as the minimum days with active internet usage, the upload to download ratio or the difference in traffic between working and non-working days, as showed in the next table.

	Uploads (month)	Downloads (month)	Ratio Up/down.	Diff. workdays uploads	Diff. workdays down.	Days active usage
Uploads (month)	1					
Downloads (month)	0.34	1				
Ratio Up/down.	0.06	0.01	1			
Diff. workdays uploads	0.22	0.11	0.01	1		
Diff. workdays down.	0.06	0.24	-0.01	0.30	1	
Days active usage	0.09	0.27	-0.01	0.04	0.10	1

Table 7. Correlation among traffic for the full month, ratio of uploads to downloads for the full month, difference between traffic in non-working and working days and ratio of days with active usage

Thus, these variables may add some value to the clustering analysis, when added to the variables measuring upstream and downstream traffic.

3.3. PRINCIPAL COMPONENTS ANALYSIS (PCA)

In order to reduce the dimensionality of the dataset, and to better understand variance in the data, we performed a principal components analysis. Thus, we followed a similar approach of Oliveira and colleagues (2007), who also used PCA prior to clustering the internet users (see the theoretical framework of the current thesis for more details). However, while those authors had only a day of data, and used the actual volume of traffic generated in each half-hour of that day, we were using hourly traffic averaged across different time periods of a month. In addition, the authors included only downstream traffic, while we also included upstream traffic, a count of the days with active usage, and variables that reflected the usage relative to the other clients (number of hours with traffic higher than the average traffic of 90 and 99% of the sample).

To systematize, we included two sets of variables (a total of 17 variables):

- Average hourly traffic for all the smallest periods of date/time considered (working days, during the day; non-working days, during the day; working days, during evening …), for uploads and downloads (12 variables total);
- Hours with higher traffic than the bottom 90 and 99% of the sample, for uploads and downloads; days with active usage (5 variables total).

Since our data was in very different scales (for example, hourly average of uploads for working days and night hours variable ranged from 0 to 21806 MB; days with active usage variable ranged from 0 to 1), we scaled and centered the data prior to performing the PCA.

One possible way of selecting the number of PCs to retain is to analyze the plot of the variance explained by each PC, commonly known as the scree plot. The number of PCs to retain is the point in the where an “elbow” is formed, that is, when the amount of variance explain by each additional PC

starts to plateau. The scree plot of the current analysis suggested that three principal components (PCs) should be retained. These PCs explained 79% of the total variance.

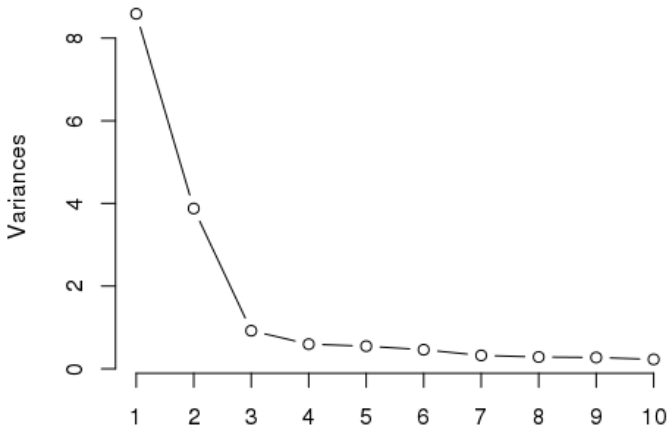


Figure 3.10 - Eigenvalues of the first ten principal components

A table containing the loadings of each variable in the three PCs is in the appendix, as along with a table containing the variables names and the corresponding descriptions. The analysis of the loadings of each variable in the three retained PCs suggest that the following interpretations of the PCs:

PC	PC description	PC naming
PC1	All average traffic variables loaded on this factor, plus the thresholds representing the number of hours with traffic higher than the average traffic of 90 or 99% of the sample. A higher value on this PC means high overall traffic, relative to the rest of the population.	Overall traffic
PC2	The average upload variables loaded negatively in this factor, while the average download variables loaded positively; high values suggest low relative upload and high download traffic; low values suggest high relative upload and low download traffic; and values close to zero suggest the same relative rate of uploads and downloads.	Difference uploads downloads
PC3	The variable measuring the number of days with active internet usage loaded very highly in this principal component; high values suggest frequent usage.	Frequency of usage

Table 3.4 - Description and naming of the three retained PCs

The PCs were easily interpretable and intuitive, strengthening the decision to use them in the clustering analysis. This way, we were able to reduce the dimensionality of the data to three PCs, which independent linear combinations of the 17 original variables. This helped us better understand the patterns of usage of the clients, and simplified the subsequent clustering analysis, while retaining much of the variability in the data.

3.4. CLUSTERING ANALYSIS

3.4.1. Clustering process

To perform the clustering analysis, we used the k-means algorithm. K-means is a well-known clustering algorithm, which has the advantages of a) being easy to implement and to interpret and b) being able to handle with (REF), two important features considering the business goals of this analysis (being able to generate easily explainable clusters in a limited timeframe, for a high number of customers). We also attempted using spectral clustering, but it was computationally much more intense and harder to implement for the entire sample. Therefore, we did not proceed using that algorithm.

To select the appropriate number of clusters (k), we ran the algorithm setting different values of k. For efficiency reasons, we started by selecting a sample of 5% of the population, on which we computed the following measures:

- **Within-cluster sum of squares (WSS):** WSS is a cohesion measure. It represents the average distance of each point to its cluster center. Increasing k typically reduces WSS, to the point where WSS is zero (each data point is the center of its own cluster). It is important to balance the benefit in the increase of cohesion and the cost of adding one more cluster to the solution. This is usually done by plotting the WSS and observing where the line starts to plateau (the "elbow").
- **Average silhouette:** silhouette is a measure of the similarity of a data point to its own cluster (cohesion), compared to its similarity to other clusters (separation). The silhouette can be calculated with any distance measure (e.g., Euclidean, Manhattan). It produces values between -1 and 1 and the highest the value, the better the clustering solution. It is calculated through the following formula:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

Equation 3.1 - Silhouette formula

Where a(i) is the average distance between a data point i and all points assigned to the same cluster as i, and b(i) is the lowest average distance between i and all points in any other cluster to which i was not assign to.

We also made sure that the selected k did not produce clusters that were too small to be meaningful from a business point of view; and that the results generalized well (to different samples and to the entire population).

In terms of the variables included in the cluster analysis, we followed three main approaches:

- Cluster using all the available variables (this was also used to understand the data better and set a baseline of cluster results);
- Cluster with different subsets of variables;

- Cluster using the PCs described in section 4.

We always scaled the data prior to clustering, since the variables had different ranges, which could influence the results.

To compare the cluster results when using different variables, we used the following metrics:

- **Average silhouette** across all dataset (see above description);
- **Silhouette for each cluster** in the solution (silhouette can also be calculated separately for each cluster, giving a measure of how cohesive and distinct from the other cluster it is);
- **Between sum of squares / total sum of squares** (total sum of squares is the average distance of each point to the global dataset average; between sum of squares is the distance between each cluster average to the global dataset average. The higher this ratio, the more distinct is each cluster average to the global average, suggesting a good cluster solution).

In addition, we also evaluated the clusters results qualitatively, considering how meaningful and useful the different clusters produced were.

3.4.2. Clustering using all the variables

To establish a baseline, we started by conducting a cluster analysis with all the variables in the dataset (a total of 68 variables; see section 2 and 3 of the present report). Based on the within-cluster sum of squares plot, the silhouette plot and the size of the resulting clusters, we decided to select k = 2.

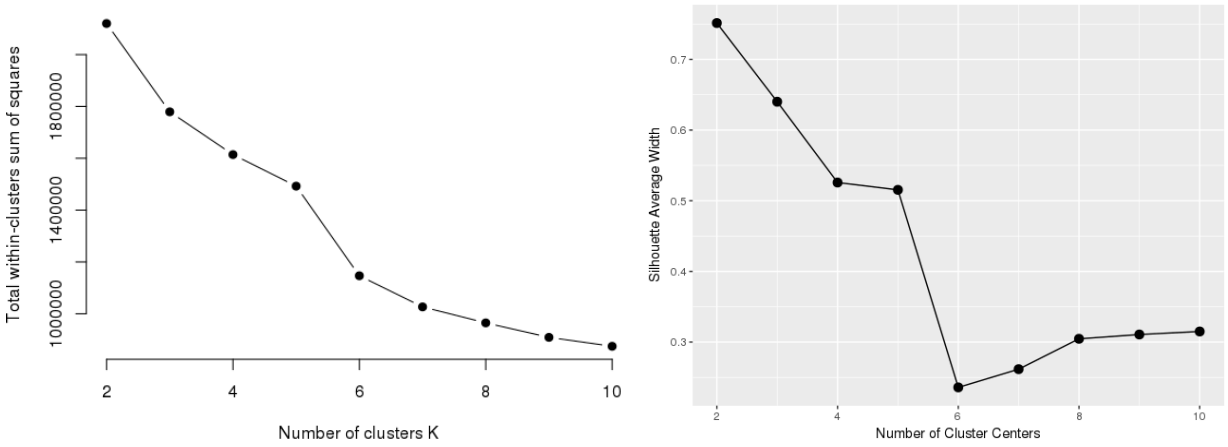


Figure 3.11a and b - Within-cluster sum of square and silhouette plot for the cluster analysis using all available variables, for different sizes of k

We interpreted the obtained clusters based on the average and range of each clusters on each variable. Below, we present the description, size (in %) and silhouette value of the clusters.

Cluster	Cluster Description	Cluster size	Silhouette
1	High consumption. More uploads than downloads. More usage on non-working days.	6.1%	-0.13
2	Vast majority of users. Average consumption. More downloads than uploads.	93.9%	0.81

Table 3.5 - Cluster description, size and silhouette measure

Even though the overall silhouette value was high (0.75), the smaller cluster had a negative silhouette value, suggesting that some points were closer to the other cluster center than, on average, to the points in their own cluster. In addition, the percentage of between SS over the total SS was low (23%)). From the business perspective, a solution with only two clusters where one cluster aggregates 94% of the clients was also not rich enough. Thus, there was margin to improve the results.

3.4.3. Clustering using a subset of variables

The correlation analysis of the variables in the subset had revealed that many variables were highly correlated. With this finding in mind, we performed several cluster analyses with different subsets of less correlated variables (since using many highly correlated variables may bias the analysis; Sambandam, 2003), and compared their results. However, none of these analyses produced a clustering solution with better metrics than the solution using the PCs described in section 4, which is detailed in the next section.

3.4.4. Clustering using the PCs

Using the three PCs previously described, the best cluster solution had k = 4.

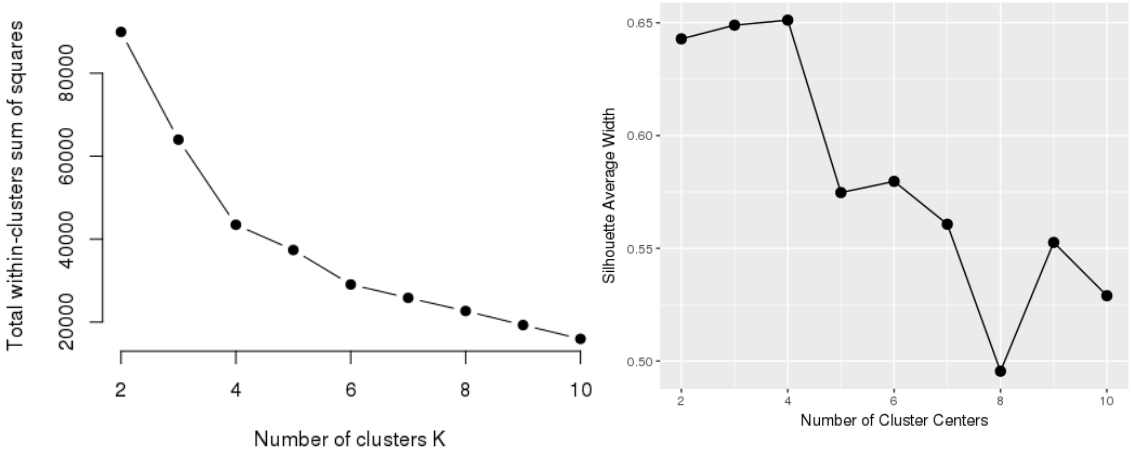


Figure 3.12a and b - Within-cluster sum of square and silhouette plot for the cluster analysis using the PCs, for different sizes of k

The table below presents the size, interpretation, and silhouette of each cluster (see section 9 for further characterization of the clusters). We also present the cluster label, which we will use throughout this report.

Cluster	Cluster Description	Label	Cluster size	Silhouette
1	Moderate-high traffic. More relative traffic of downloads than uploads. Less frequent usage.	Moderate traffic, downloads	11.0%	0.30
2	High traffic. More relative traffic of uploads than downloads. Frequent usage.	High traffic, uploads	0.3%	0.33
3	Low traffic. Approximately the same relative traffic for uploads and downloads. Infrequent usage.	Low traffic	12.1%	0.67
4	Majority of the sample. Moderate to low usage. Same relative traffic of uploads and downloads. Frequent usage.	Majority	76.6%	0.70

Table 3.6 - Description, size and silhouette of the clusters obtained using the PCs

The obtained clusters have acceptable silhouette values, and a relatively high between SS / total SS ratio. The following table summarizes the evaluation measures for the analysis using the PCs, compared with the metrics obtained when using all the variables.

Cluster analysis	Between SS / Total SS	Average silhouette	Range of silhouette per cluster
All variables	23.4%	0.751	[-0.13, 0.81]
PCs	64.9%	0.651	[0.30, 0.70]

Table 3.7 - Cluster metrics when clustering using all variables, a subset of variables, and the PCs

The cluster analysis with the PCs held acceptable silhouette values for all clusters (even though average was lower), and a better ratio between between-SS and total-SS. It was also more interesting from the business point of view (more differentiated clusters, including a cluster with low usage and two clusters with high usage, for uploads and downloads). Thus, this analysis was selected as the best solution to segment the clients according to their traffic.

3.5. VALIDATION OF THE RESULTS IN A DIFFERENT DATASET (DATA FROM SEPTEMBER/OCTOBER)

To check if the results would be consistent in a different and wider time-period, we replicated the analysis for the two most recent months with data available (September and October).

3.5.1. PCA with data from September/October

The PCA results with the new dataset are very similar to the results obtained for May. We again selected three PCs, which explain 80% of the variance of the original dataset. The interpretation of these PCs is the same as the interpretation of the PCs obtained with May data (see appendix for a table with the loadings of each variable in each PC).

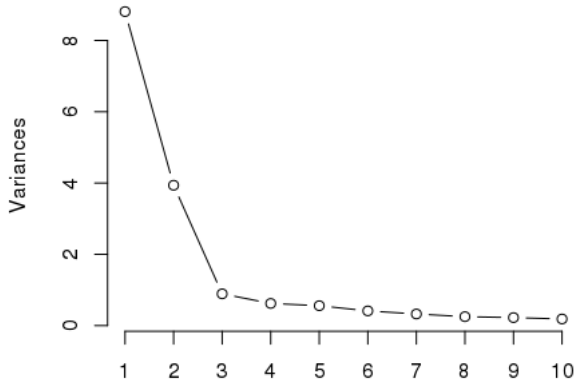


Figure 3.13 - Eigenvalues of the first ten principal components, for the September/October dataset

3.5.2. Cluster analysis with data from September/October – 5% sample

We then used the PCs in a cluster analysis, using 5% of the sample (for efficiency reasons, and due to limitation in computing the silhouette for the entire sample). As for May data, the elbow and the silhouette plot suggest k = 4.

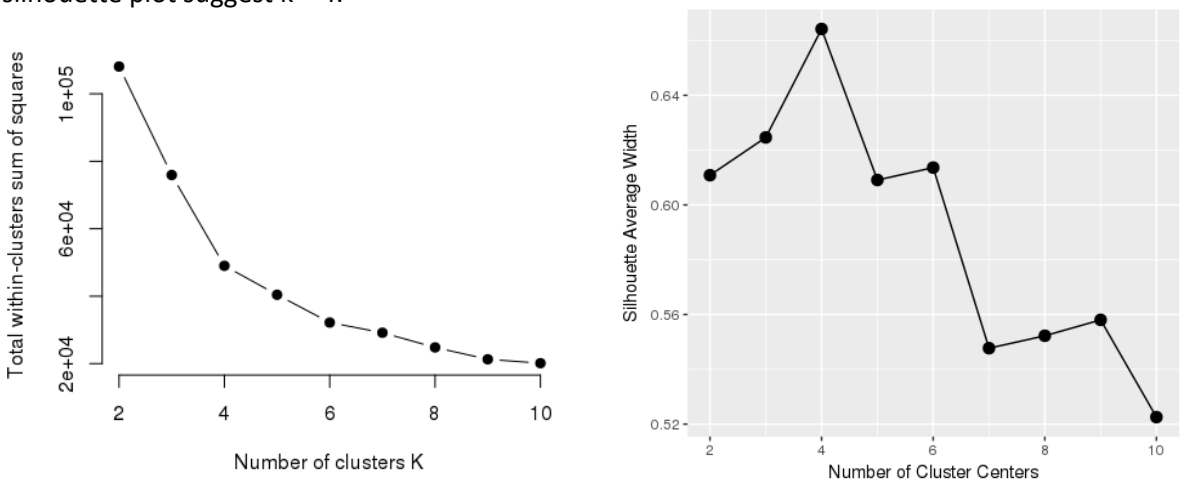


Figure 3.14a and b - Within-cluster sum of square and silhouette plot, for different sizes of k, for the cluster analysis using September/October data

We obtained similar clusters to those obtained for May data, with similar sizes and silhouette values.

Cluster	Cluster label	Cluster size - May	Cluster size - Sept/Oct	Silhouette - May	Silhouette – Sept/Oct
1	Moderate traffic, downloads	11.0%	9.2%	0.30	0.29
2	High traffic, uploads	0.3 %	0.5%	0.33	0.31
3	Low traffic	12.1%	18.3%	0.67	0.80
4	Majority	76.6%	72.0%	0.70	0.68

Table 3.8 - Comparison between cluster sizes and silhouette values of May and September/October

The analysis of averages and ranges of the clusters suggests an interpretation similar to the presented for the May data (which is detailed in the section 3.8, final results). Below, we present a table comparing the metrics of the cluster analysis using May and September/October data.

Cluster analysis	Between SS / Total SS	Silhouette
May data	64.9%	0.651
Sep/Oct data	68.0%	0.664

Table 3.9 - Cluster metrics for cluster analysis using May and September/October data

Thus, the two analyses using different datasets led to similar conclusions, suggesting that the clustering solution we adopted was stable across time and that the clusters are well defined. Therefore, we applied this analysis to the entire population of September/October.

3.5.3. Cluster analysis with data from September/October – entire population

We found similar results using the entire population for within-cluster sum of squares. We could not compute the silhouette values for the entire population, due to technical limitations.

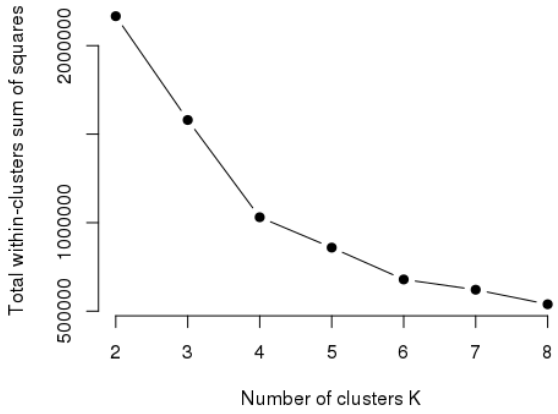


Figure 3.15 - Within-cluster sum of squares for September/October data, for the entire population

Cluster	Cluster label	Cluster size (%) – 5% sample	Cluster size (%) – entire pop.	Cluster size (N) – entire pop.
1	Moderate traffic, downloads	9.2%	9.6%	98459
2	High traffic, uploads	0.5%	0.3%	3483
3	Low traffic	18.3%	18.4%	187878
4	Majority	72.0%	71.6%	731906

Table 3.10 - Comparison between the distribution of MAC addresses per cluster using 5% or 100% of the population

The average and range of the clusters when using the entire population were also similar to the ones obtained with 5% of the data.

3.5.4. Match between May clusters and September/October clusters

Ideally, for the analysis to show some robustness and be useful, the same cable modems should be in the same cluster in the two periods of data (May and September/October). We merged the two datasets and found that 74% of the MAC addresses included in September/October analysis had been in the May analysis. Of these MAC addresses, 88% belonged to the same cluster in both analyses.

Below, we present visualizations showing the distribution of the MAC addresses in the two analysis. We created a pie chart with all the MACs in a given cluster in May data; and for each pie chart, we represent the percentage of those users in each cluster in September/October data.

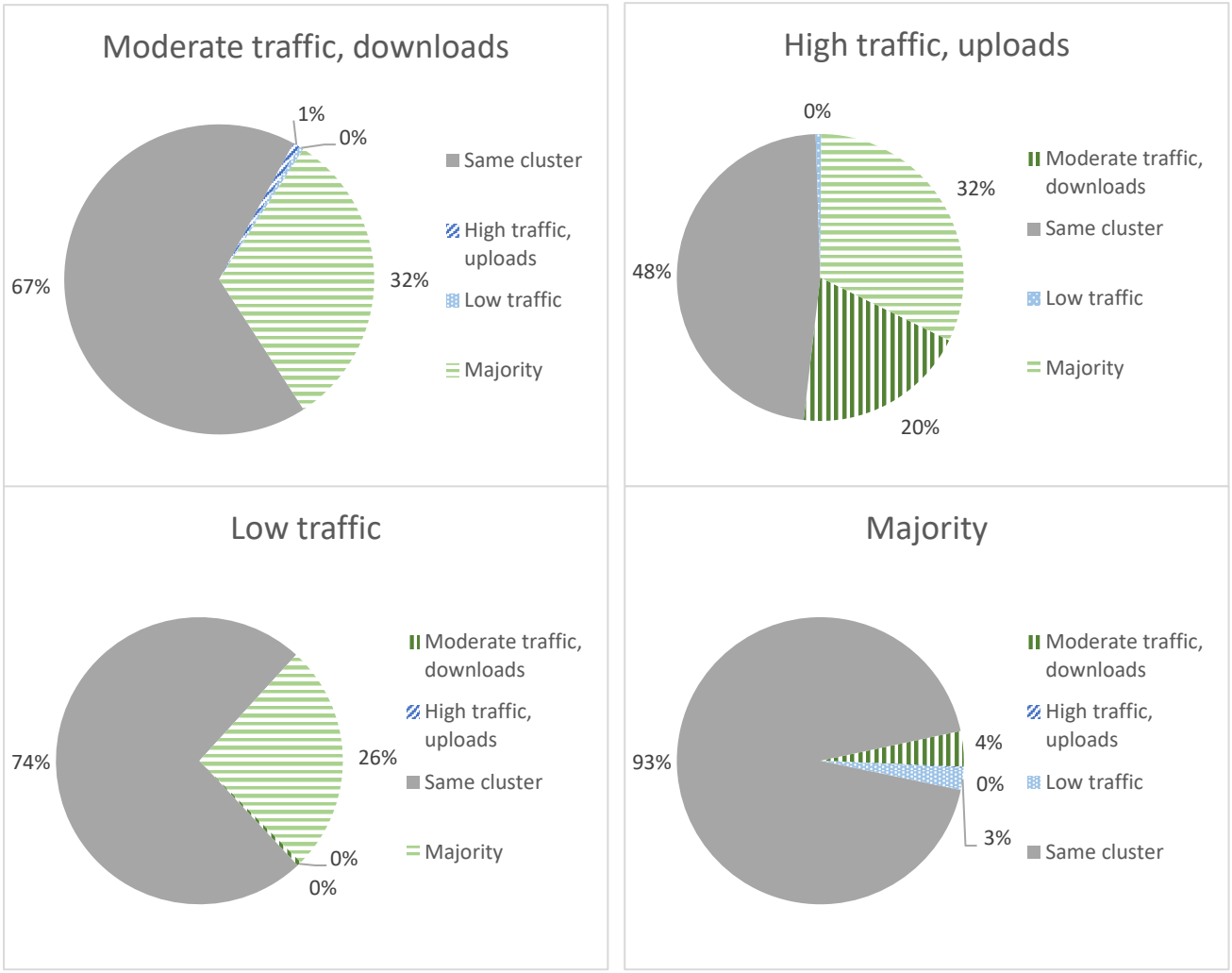


Figure 3.16a, b, c & d - Percentage of MAC addresses that belong to each cluster using Sept/October data, divided by their cluster in May (each pie chart corresponds to a cluster in the analysis using May data)

We note that the percentage of MACs who changed from the cluster high traffic, uploads, to other clusters is the highest. However, this is the smallest cluster (representing around 0.5% of the data). Only 1.6% of the cases that changed cluster belonged to this cluster. The majority of devices migrated from cluster Moderate traffic, downloads to cluster Majority (29% of the cases who changed cluster); from cluster Low traffic to cluster Majority (26% of the cases); and from cluster "Majority" to cluster Moderate traffic, downloads (26% of the cases). We also note that the migration to the Majority cluster was the biggest change for all the other three clusters. This may be because the majority cluster is not only the cluster with the most clients but also the cluster that corresponds to the most moderate and typical usage traffic; it is more expectable for someone from a high usage cluster to migrate to this cluster than to the low usage cluster, for instance.

3.6. FINAL RESULTS: CLUSTER CHARACTERIZATION

The cluster analysis using the PCs seemed to be the most appropriate to segment the entire population of cable modems according to their traffic and was stable when applied to datasets from different time-periods. Therefore, this was selected as the final cluster solution.

To further explore and understand the resulting segments, we developed several visualizations, using the most up-to-date data (September/October data, for the entire population of MAC addresses).

Since the PCs were a transformation of the variables, it is harder to interpret the clusters using them. In addition, because data was scaled for PCA and cluster analysis, it is harder to understand the magnitude of the differences between clusters using those variables. Thus, to further aid the interpretation of clusters, we calculated the average and range for the untransformed variables that were more similar to the PCs: average hourly traffic across the whole month, for uploads and downloads; difference between average hourly upload traffic and download traffic; and number of days with active usage. These plots are presented below.

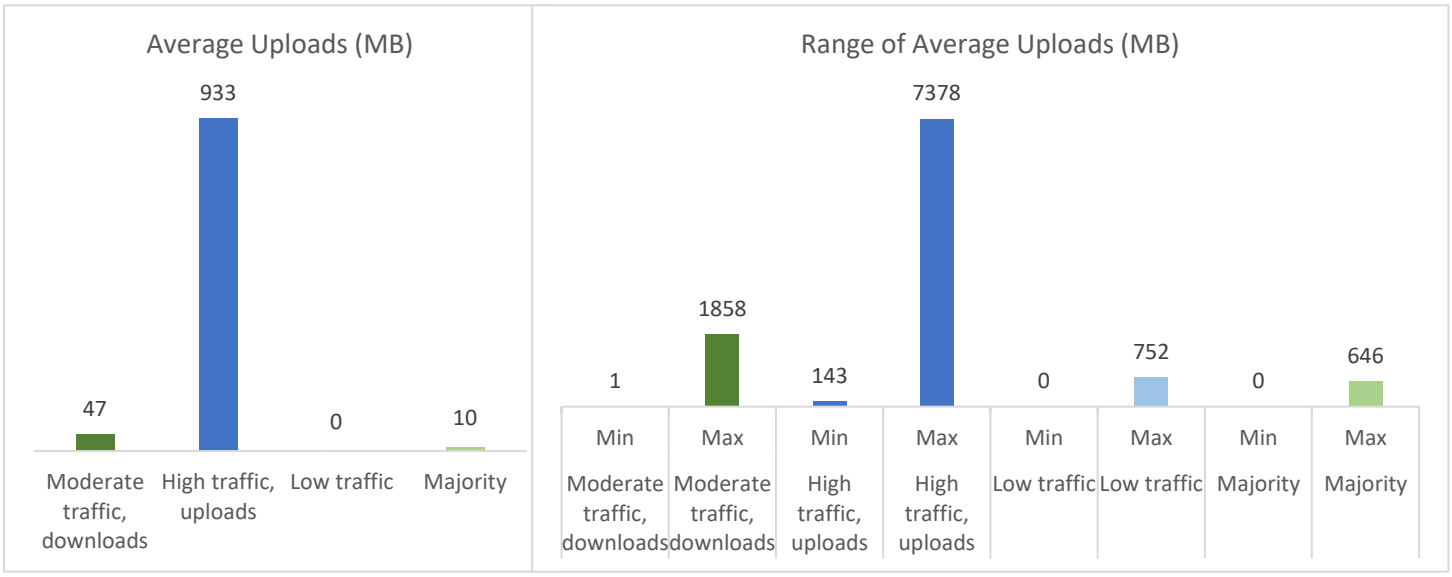


Figure 3.17a, b - Averages and range of hourly averages for uploads, across the entire month, per cluster

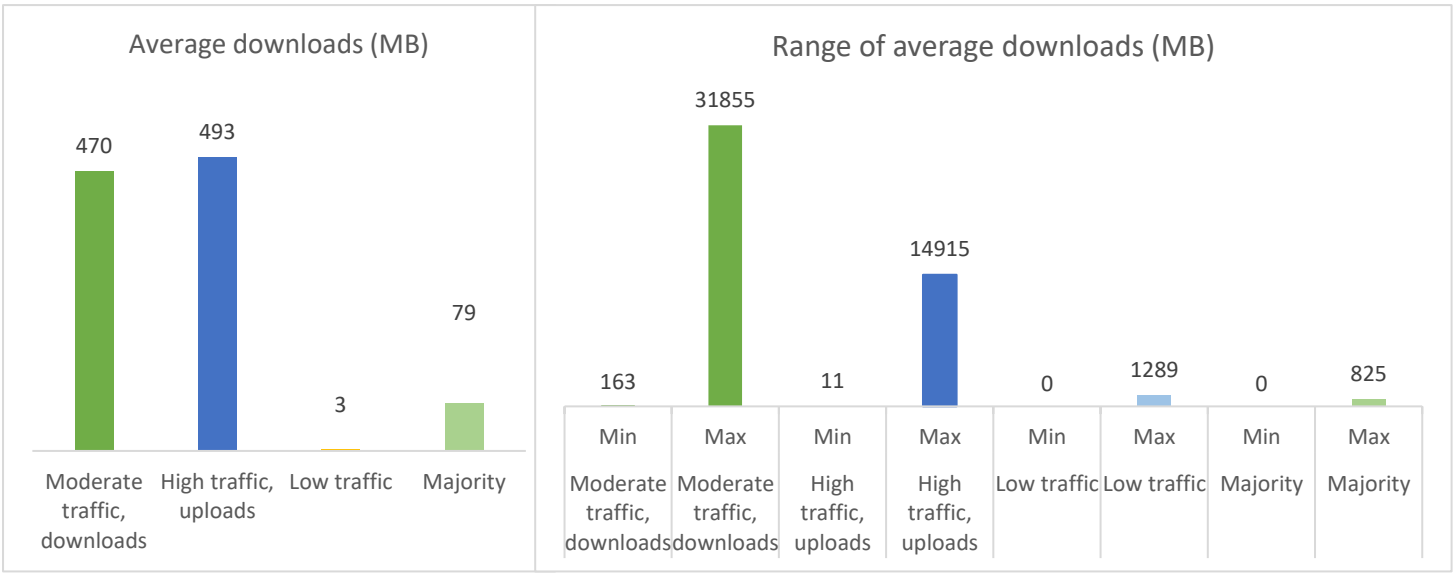


Figure 3.18a, b - Averages and range of hourly averages for downloads, across the entire month, per cluster

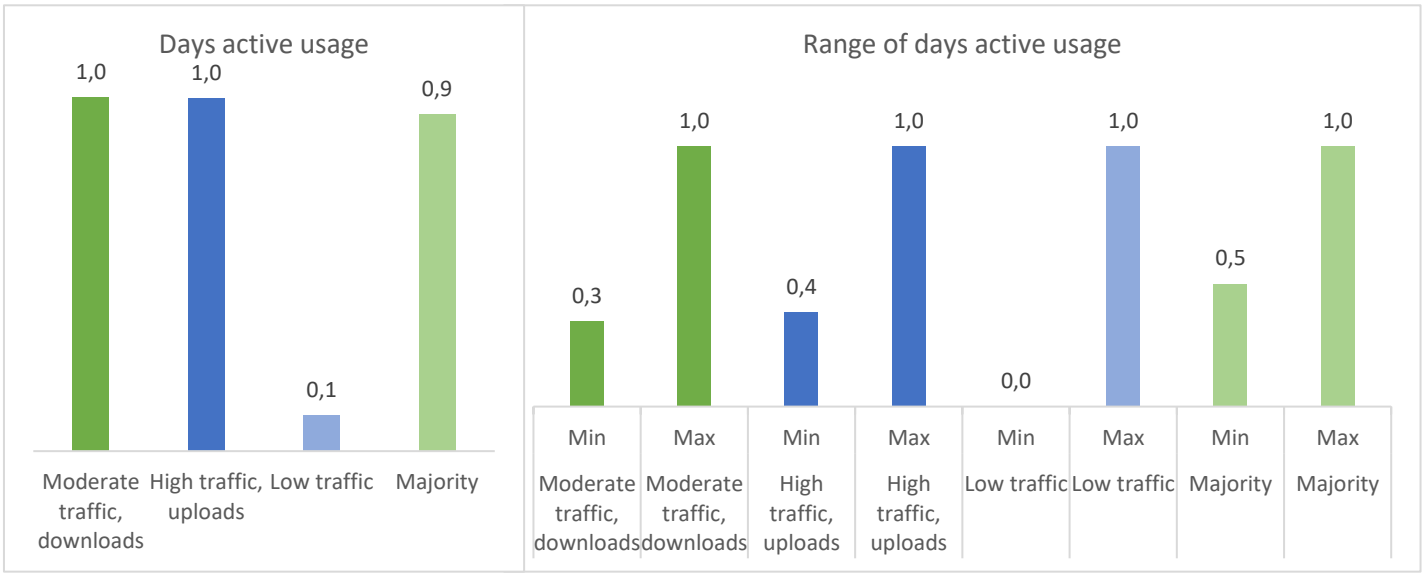


Figure 3.19a, b - Averages and range of days with active usage, per cluster

3.7. EXTERNAL VALIDITY OF THE CLUSTERS

In order to check for external validity of the clusters, we analyzed each cluster behavior in variables that are expected to be related to internet traffic but were not included in the cluster analysis.

3.7.1. Validation of the clusters using service request variables

The main goal of this cluster analysis is to predict the quality of experience with internet, based on parameters of the service. One of the variables associated with the quality of experience of internet is

the internet-related, technical service requests made by customers. That is, if a customer experiences poor internet quality, he or she might complain, which generates a service request. However, different users, with different patterns of internet traffic, may be more or less demanding relatively to their internet quality.

Given its relevance to the project, we started by analyzing if there were differences in the clusters' service requests. We selected data from service requests for the previous year of the analysis (September 2016 to October 2017). The following figures depict this analysis.

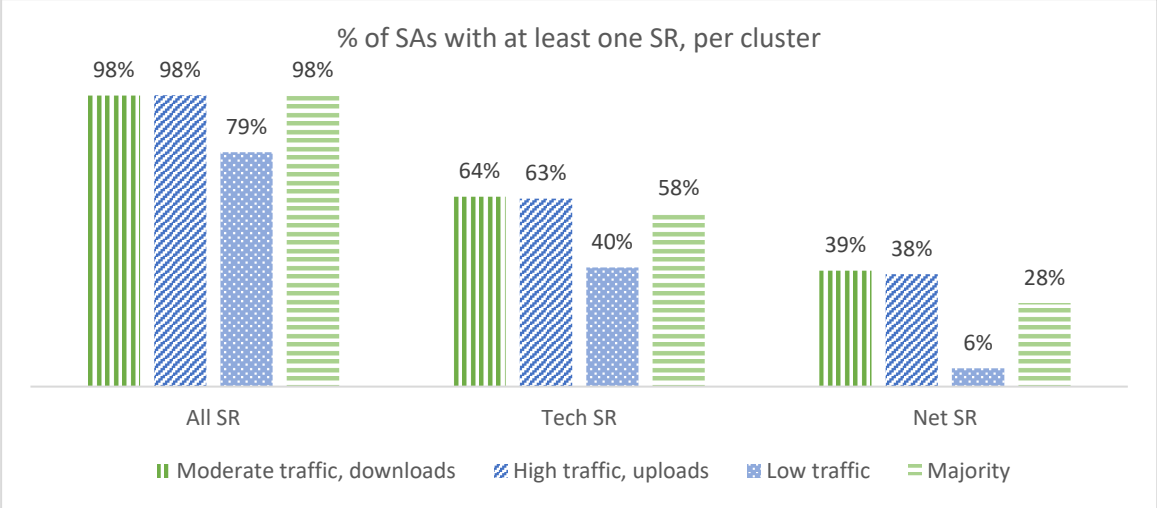


Figure 3.20 - Percentage of clients who made at least on service request, per cluster and per type of service request (all service requests, only technical service requests and only internet service requests)

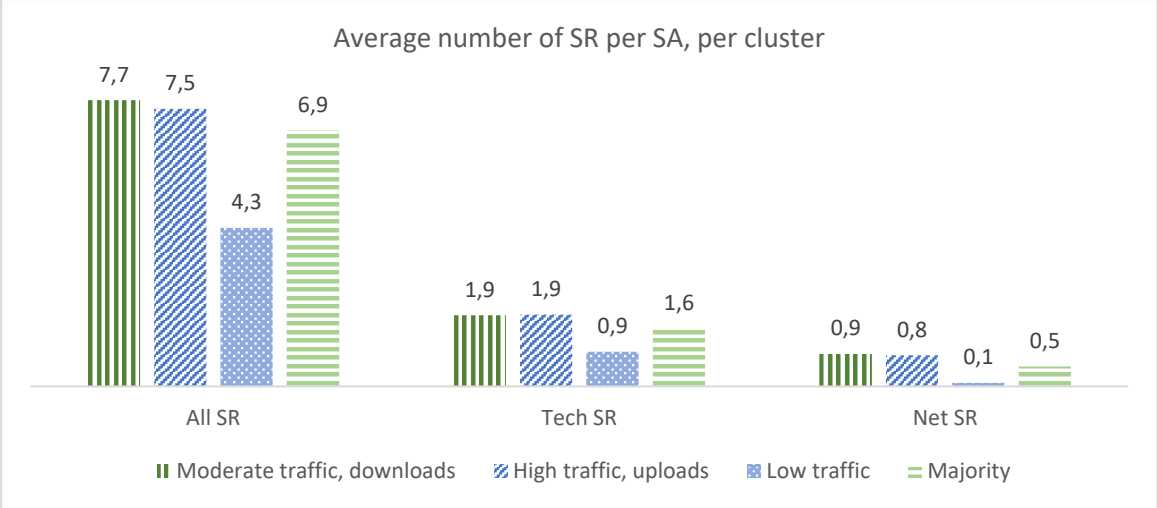


Figure 3.21 - Average number of service requests, per cluster and per type of service request (all service requests, only technical service requests and only internet service requests)

Even though clients in the low traffic cluster tend to file less service requests overall, this difference is more pronounced when considering only the internet service requests. We see this both when looking

at the percentage of SAs who filed at least one service requests, and at the average of service requests per cluster. Conversely, the clusters with higher traffic (moderate traffic, downloads and high traffic, uploads) are the clusters with more service requests related to internet. This is consistent to what we would expect: people with higher internet usage may be more likely to identify problems in the network (because they use more frequently) and may be more demanding in terms of quality of service, leading to more technical complaints.

3.7.2. Validation of the cluster using the subscription of a mobile internet service

It is plausible to think that customers with higher traffic in their cable modem are also more active in other internet sources. For instance, clients with higher traffic may be more likely to have subscribed a mobile internet service (i.e., a USB-stick that provides broadband internet service). Below, we present the percentage of clients in each cluster who subscribed a mobile internet service in the months of September and October 2017.

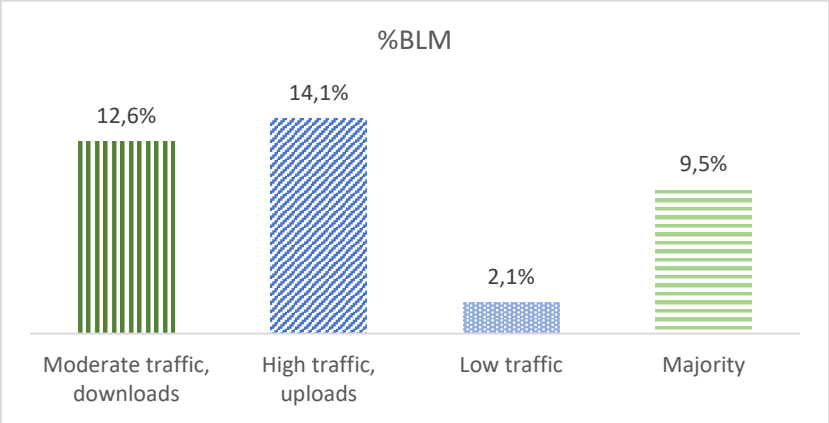


Figure 3.22 - Percentage of SAs who subscribed a mobile internet service, per cluster

As expected, a higher percentage of SAs in the clusters with higher traffic subscribed to a mobile internet service, while the cluster with lower traffic had a lower percentage of clients who subscribed to this service, which further validates our clustering results.

4. PREDICTING IF CLIENTS WOULD USE THE INTERNET AT A GIVEN TIME

While network signals are fundamental for identifying bad internet service, they are not sufficient to identify clients who had a bad internet experience. A cable modem may have lost network signal, but that only translates into a bad experience for a client if he or she tries to connect to the internet. Contacting clients who did not realize their cable modem lost access to the internet may constitute a nuisance for the client and an unnecessary cost for the company.

However, if the cable modem is not connected to the monitoring systems because it has no access to the internet, we have no way of knowing if the client was trying to use the internet. We can only try to infer, based on the clients' past behavior, if it was a time where he or she would typically use the internet. Therefore, we attempted to define a probability that a client would be using at the time the problem was identified.

4.1. WHAT CONSTITUTES INTERNET USAGE?

We did not have access to a variable that indicates whether a client is actively using the internet. The only variable that has some indication of internet usage is the volume of traffic generated by the cable modem in a given hour. However, even when a client is not actively using the internet service, typically some upstream and downstream traffic is registered, because of the communication between the modem and the monitoring systems. In addition, there may be updates to the modem software, which also generate traffic.

Therefore, the first challenge was to define a threshold above which we would consider there was active usage by the client. In order to do so, we followed two main approaches, presented below. We note that, after an additional exploration, we decided to consider only downstream traffic, for three main reasons:

- Upstream and downstream traffic typically convey the same information; if anything, upstream traffic is usually lower;
- Focusing on only one of the two was more efficient in terms of the required resources;
- From a business perspective, download traffic is more relevant.

4.1.1. First approach: Consulting with the technical team

The technical team informed us that there was no specific technical information available regarding the volume of traffic generated by the cable modem alone (without usage by the client), or any previous analysis done on this subject. As an educated guess, they suggested that 5 MB could be a reasonable threshold, since it is fairly easy to reach this volume of traffic by simply loading a webpage.

To check how these thresholds translated in practice, we verified the percentage of clients that had downstream traffic above 5 MB and on three key hours, where we expected the percentage of users to vary: during the night (2 a.m.), during the afternoon (3 p.m.) and during the peak hour (9 p.m.). Below we present the results obtained for a working and a non-working day:

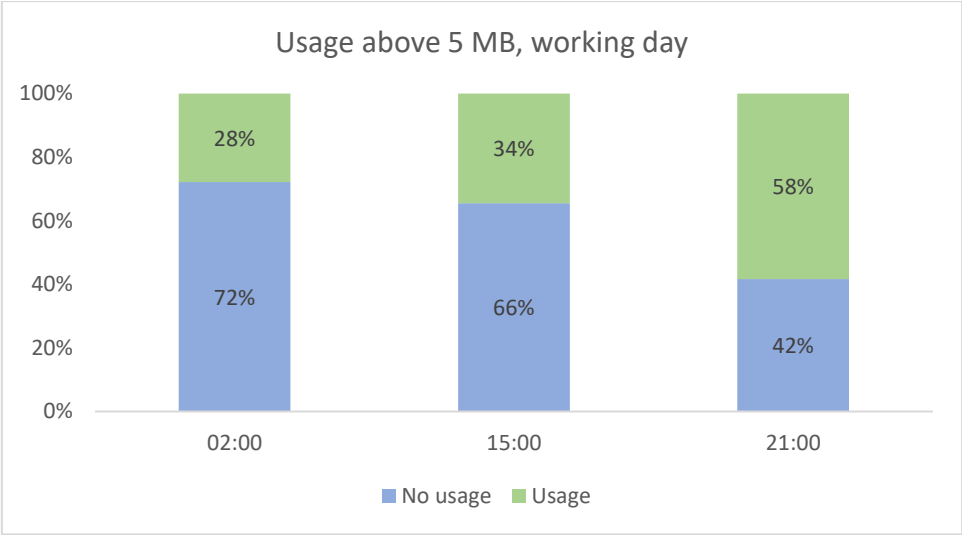


Figure 4.1 – Percentage of clients using the internet according to the 5MB threshold, on a working day

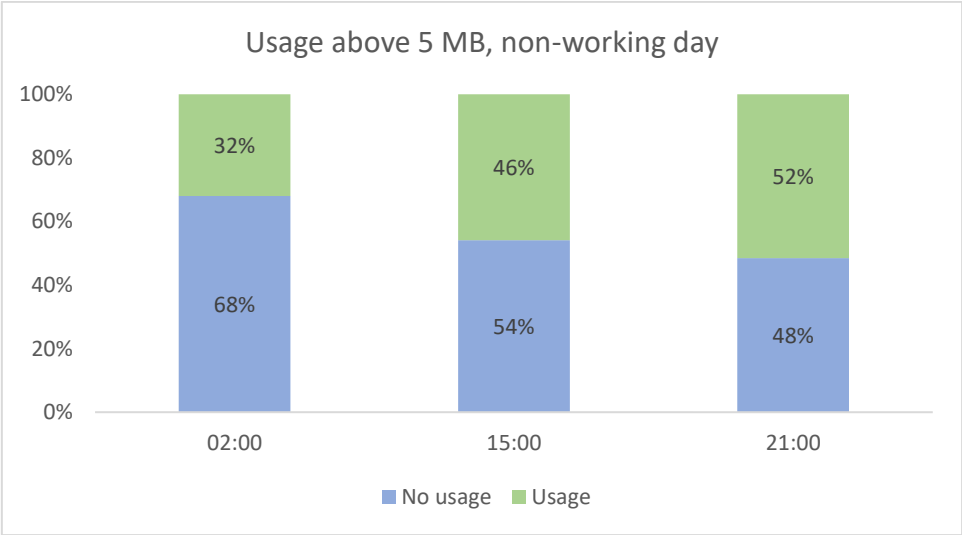


Figure 4.2 – Percentage of clients using the internet according to the 5MB threshold, on a non-working day

These numbers seem reasonable, since they suggest that:

- More clients would be using during the peak hour (about half), in both working and non-working days;
- Only a third of the clients would be using during the night;
- More clients would use the internet during the afternoon on non-working than working days.

We also tested a more conservative threshold of 1 MB. The percentage of clients using the internet in the different time segments considered was similar, albeit slightly higher for the 1 MB threshold. The table below presents the comparison of the results using two thresholds:

	5 MB Threshold	1 MB Threshold
Working day, 2 a.m.	28 %	36 %
Non-working day, 2 a.m.	32 %	39 %
Working day, 3 p.m.	34 %	40 %
Non-working day, 3 p.m.	46 %	52 %
Working day, 9 p.m.	58 %	65 %
Non-working day, 9 p.m.	52 %	57 %

Table 4.1 – Comparison of the percentage of clients using the internet when considering 5 MB or 1MB as the thresholds

These results suggest that both thresholds seem reasonable candidates for being the threshold separating clients who are actively using and not using the internet.

4.1.2. Second approach: Examining the traffic of a cable modem rarely used

We had a cable modem in the office that was registered in the network but typically no one used. We looked at maximum hourly traffic generated in a given day, for 7 months (between September 2017 and March 2018, corresponding to 190 registered days). To reduce the probability someone was using the cable modem, we looked only at non-working days and non-working hours of working days (between 8pm and 8am).

The plot below presents the maximum hourly traffic generated by the cable modem for each of the days considered (each point represents a day).

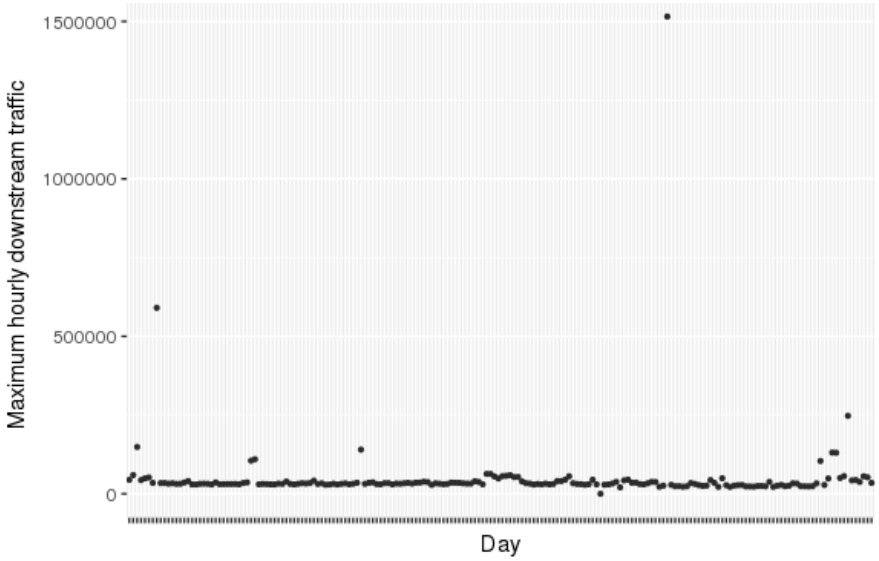


Figure 4.3 – Maximum hourly downstream traffic generated by a cable modem rarely used, per day, for 190 different days

In one of the days considered, the traffic generated was much higher than in the rest of the days (around 1.5 MB). It is possible that this is due to some update in the cable modem software. Nevertheless, this happened only once during the period under analysis. In 97% of the cases, the

generated traffic was below 70 KB, which led us to select this as another candidate for the usage threshold.

Repeating the procedure for the 5 MB threshold, we analyze the percentage of users that had traffic above 70 KB, in key hours of the day and in working and non-working days.

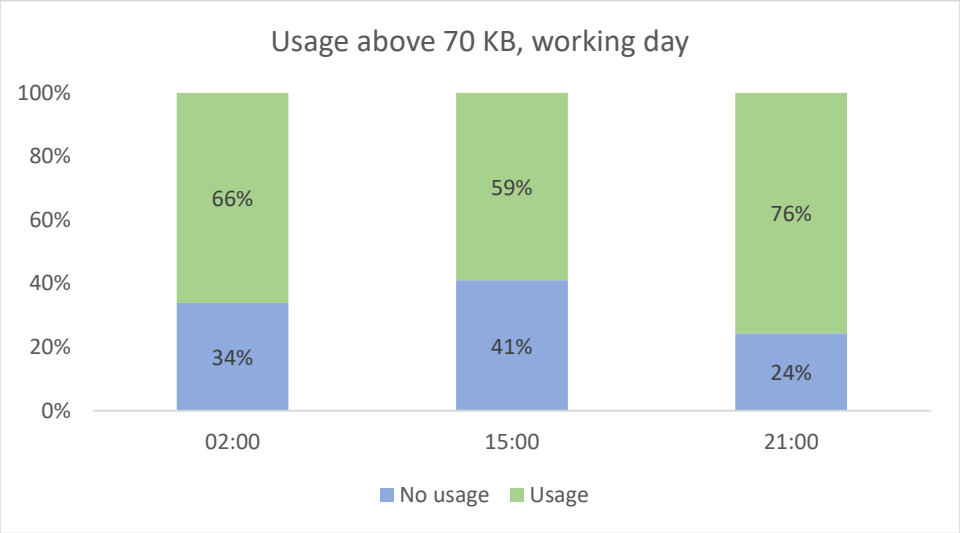


Figure 4.4 – Percentage of clients using the internet according to the 70KB threshold, on a working day

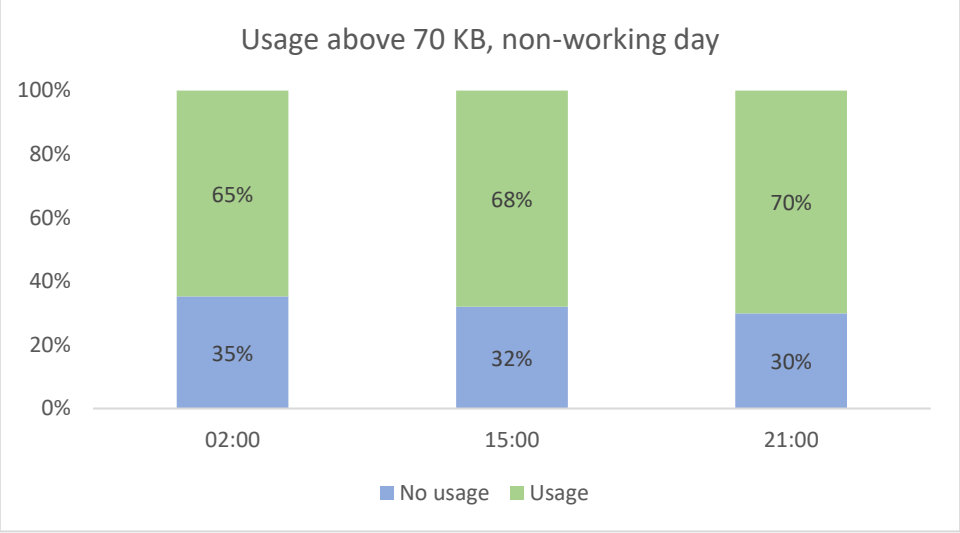


Figure 4.5 - Percentage of clients using the internet according to the 70KB threshold, on a non-working day

Contrary to the previous threshold, this threshold does a worst job differentiating between hours where we would expect more or less clients to be using the internet. In addition, according to this threshold, two thirds of the clients would be using the internet at 2 a.m. of a working day, which seems unlikely. This suggests that this threshold is too low. Therefore, we did not consider this threshold further.

4.1.3. Final decision

Based on the above analysis and on business understanding, we decided to use the more conservative approach of 1 MB as the threshold of usage. While this might mean that some hours are misidentified, we thought it would be more problematic to fail to identify usage when it actually occurred than the opposite. Even when a person only uses the internet for simple things like checking the e-mail or chatting with their friends (generating very low volumes of data), having no access to the internet might be a very unpleasant experience.

4.2. FEATURE ENGINEERING USING THE 1MB THRESHOLD

After establishing the threshold of internet usage at 1MB, we proceeded to create a feature that could predict whether a client would be using the internet and, ultimately, be included in the model for identifying clients without access to the internet. More specifically, we wanted to create a variable that would indicate whether a client would be likely using the internet at a given hour or not.

4.2.1. Unsuccessful approaches to build features

Before deciding to predict each hour individually, we tried to group hours that could behave similarly. Our rational was that we could have more robust probabilities of usage if you would predict day segments, instead of predicting individual hours. For example, a client could always use in the period between 9 p.m. and 12 a.m., but only use the internet sometimes between 9 p.m. and 10 p.m.

To group the hours, we tried two approaches: principal components analysis (PCA) and clustering. For PCA, we constructed a dataset with the hourly usage of 2% of the clients during a week. Each hour and day were used as input variables, for a total of 92 variables[3]. We tested different approaches, such as using the actual traffic in bytes or recoding it according to the 1MB threshold; using only complete cases or imputing the missing values with a constant. However, we found that to reach 80% of the variance explained, we needed to retain around 30 PCs, a number too big to be useful. In addition, when we analyzed the PC, we could not extract meaningful patterns.

For the cluster analysis, we cast the dataset such that each hour of all days would be a row and each cable modem would be a column. We then applied k-means with different values of k. However, when we analyzed the values of silhouette for different values of k, all values were below what is considered acceptable (i.e., were all below 0.25). Similarly, the gap measure was inconclusive. Therefore, we also did not pursue this approach further.

4.2.2. Final feature computation

Since we could not group the hours and days in a meaningful away, we decided to build a feature representing the probability of usage for each client and each individual hour. To build this feature, we looked at each client's traffic for the month of January. For each client and each hour of the day, we counted the number of times there was traffic above 1 MB throughout the month and divided by the total of entries for that hour. That is, if a given client had measures for, say, 5 p.m. on 20 days of

[3] 92 variables instead of 168 (7 x 24) because we did not have access to registers for all hours of all days.

January, and on 10 of these days the traffic was above 1MB, the probability of usage for 5 p.m. for this client would be estimated as 0.5.

4.3. PREDICTING INTERNET USAGE

4.3.1. Variables used

We checked how well the probability of usage constructed using January data would predict usage on February, for specific hours and clients. We created a dataset containing:

a. The hourly usage, for each client at a February 11th, recoded according to the 1MB threshold into 0 – no usage and 1 – usage.
b. The corresponding probabilities of usage per hour, constructed using January data.
c. A variable containing the hour each usage and probability corresponded to.

Thus, we had one target variable (a. the hourly usage, recoded into a binary variable) and two input variables (b. the probability of usage and c. the hour it corresponded to). The reason why we also included the variable containing the hour was that the threshold could have been different for specific hours.

4.3.2. Model implementation

We fed the two input variables to a classification decision tree, using the R package rpart (Therneau, Atkinson, & Ripley, 2018). This package implements a Classification And Regression Tree (CART) algorithm, first introduced by Breiman, Friedman, Olshen, and Stone (1984). CART produces binary trees, that is, trees with nodes that have at most two children. To produce each split in the tree, a greedy technique is used, in the sense that the variable picked for the split is always the one that reduces the "impurity" of the node, that is, the split that separates cases from different classes the most. In our tree, we used the Gini index as the impurity measure we want to minimize. As for the stopping criteria (i.e., the criteria that prevents the tree to continue to grow indefinitely), we chose to limit the number of observations in a node for a split to be attempted, the number of observations at any leaf (i.e., terminal node) and the complexity parameter. As mentioned in the introduction, the complexity parameter

In the rpart package, the main model parameters that can be tuned are the following:

- Minsplit: the minimum number of observations in a node for a split to be attempted.
- Minbucket: the minimum number of observations in any leaf.
- CP: Complexity parameter. A split that does not improve the fit by a factor of cp is not attempted.
- Usesurrogate: the action to perform when an observation has missing values.
- Surrogatestyle: the way to select a variable as a surrogate, when observations have missing values.
- Maxdepth: maximum depth of any node of the tree (root node has depth = 0).

Initially, we left all parameters as default, which are presented below. However, it should be noted that our dataset did not have any missing observation; therefore, the actions to take for missing values were irrelevant.

- Minsplit: 20
- Minbucket: 7
- CP: 0.01
- Usesurrogate: use surrogate variables; if all surrogates are missing, then send the observation in the majority direction
- Surrogatestyle: select the variable based on the total number of correct classification for a potential surrogate variable.
- Maxdepth: 30

To build and validate the decision tree, we took the approach of dividing the dataset into train and test sets, which was appropriate considering we had a very large dataset (several millions of rows).

4.3.3. Results

The algorithm produced the following tree, for the train set:

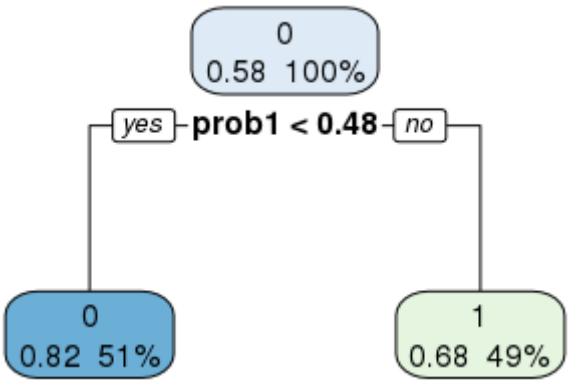


Figure 4.6 – Decision tree predicting usage on February 1st, when complexity parameter = 0.01.

When we applied the tree to classify the test set (containing all the available hours of the day), the model had the following values for the evaluation measures:

Metric	Value
Accuracy	0.75
Sensitivity	0.78
Specificity	0.73

Table 4.2 – Evaluation metrics of the predicting usage model

We also analyzed the model performance separately for each hour of the day, as depicted in the next plot. We note that some hours were missing on the x-axis. This is due to the missing data in the database.

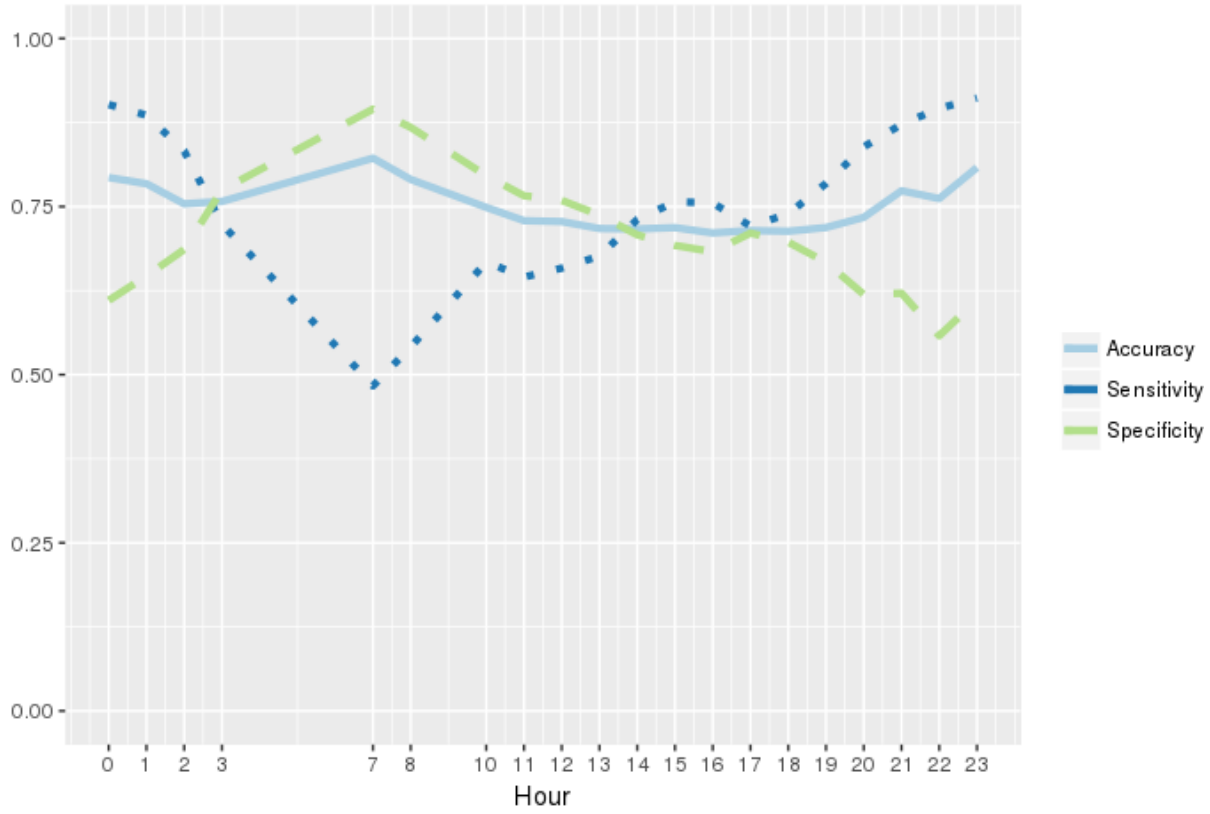


Figure 4.7 – Model evaluation metrics for each hour of the day

Analyzing the plot, we verified that accuracy (solid line) does not vary much among the hours considered. On the other hand, sensitivity (dotted line) and specificity (dashed line) vary more. This is probably due to the imbalance of the positive and negative classes in these hours. On the hours where the percentage of clients using is lower (e.g., late night and morning), the sensitivity (i.e., the percentage of cases that truly positive, out of the total number of cases classified as positive) tends to be lower, while the specificity is higher. On the hours where the percentage of clients using is higher (e.g., evenings) the specificity (i.e., the percentage of cases that truly negative, out of the total number of cases classified as negative) is lower, while the sensitivity is higher.

This model was very simple, and we used only one month of data to calculate the probability of usage, and human behavior has typically a lot of variability and is hard to predict. Therefore, we deemed these values acceptable. Nevertheless, we decided to adjust the parameters of the tree to allow it to grow more and to see if it would define different thresholds for different hours. We reduced the complexity parameter (i.e., the minimum reduction of error that a splitting needs to produce in order to be applied) to 0.001, which produce the following tree:

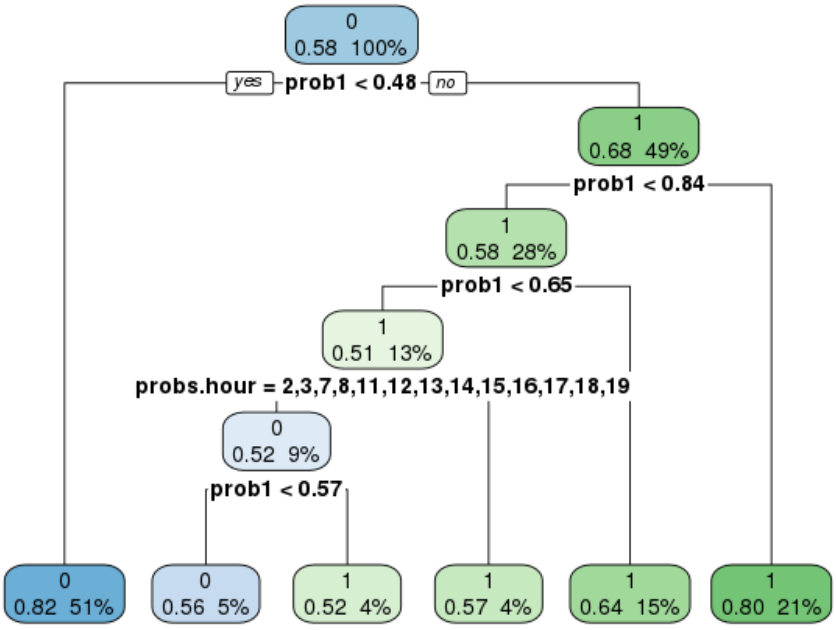


Figure 4.8 – Decision tree predicting usage on February 1st, when complexity parameter = 0.001.

When the complexity parameter is smaller, the tree did consider a separate split for hours with less usage (nighttime and during the day). However, that particular split did not reduce the error by much, as it split a node with probability 0.51 into nodes with probability 0.56, 0.52 and 0.57. The main gain of this tree lies in the first split of the right part of the tree. If the probability of usage was higher than 0.84, the client would use the internet with a 0.8 probability. This node represents roughly a fifth of the cases.

The following table compares the performance of the tree when cp = 0.01 and when cp = 0.001.

Measure	Tree with cp = 0.01	Tree with cp = 0.001
Accuracy	0.75	0.75
Sensitivity	0.78	0.73
Specificity	0.73	0.77

Table 4.3 – Evaluation metrics of the predicting usage model

The more complex tree had a similar accuracy, but with a worst sensitivity (i.e., it flagged more cases as positives when they were in fact negatives) and a better specificity (i.e., it flagged more cases as negative when they were in fact positive). However, the tree is still informative, as it suggests more defined cutoff points that may be used on the prediction of bad internet experience. More specifically, it shows that if the probability of usage is below 0.48, then the user would probably not be using. On the contrary, if the probability of usage is above 0.84, then the user would probably be using. If the probability lays between these two values, then we cannot predict usage with much accuracy. This is important because for some business problems, it might be more relevant to reduce the false positives, while for other business problems false negatives might be costlier.

4.4. PROBABILITY OF USAGE AND SERVICE REQUESTS

Our next step was to see the distribution of people with service requests of no access on this variable. We would expect people who complained to have a high probability of usage on the hour where they complained, given that to complain, people need to try to use the internet.

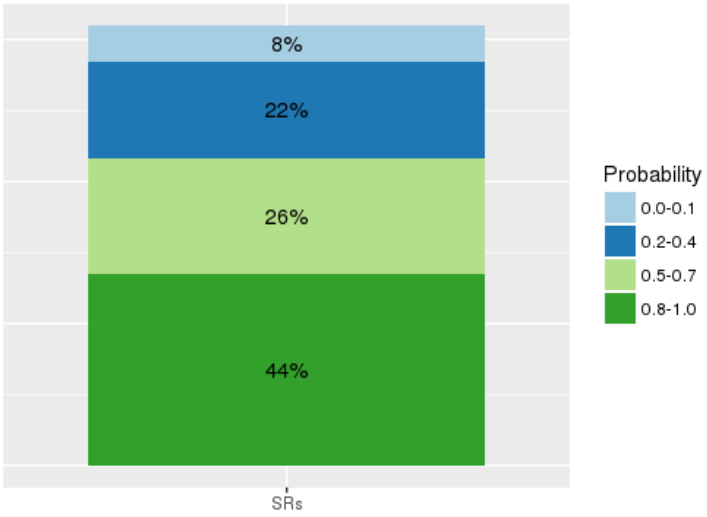


Figure 4.9 – Distribution of clients with a service request for No Access, by the probability of usage they had on the hour where they made a service request

As expected, most clients had a probability of usage higher than 0.5 (70%). The majority of clients had a very high probability of usage (between 0.8 and 1.0). Less than 10% of the clients had a probability of usage equal or less than 0.1. This plot suggests that there is a relation between the probability of usage and the filling of a service request. To reinforce this idea further, we compared the distribution of the clients who filled a service requests with the distribution of the clients who did not fill a service request, on two hours of the day using a density plot. Data for one of the populations consisted of measurements taken between 9 and 10 a.m. (the morning population) and data for the other population was taken between 7 and 8 p.m. (the peak hour population).

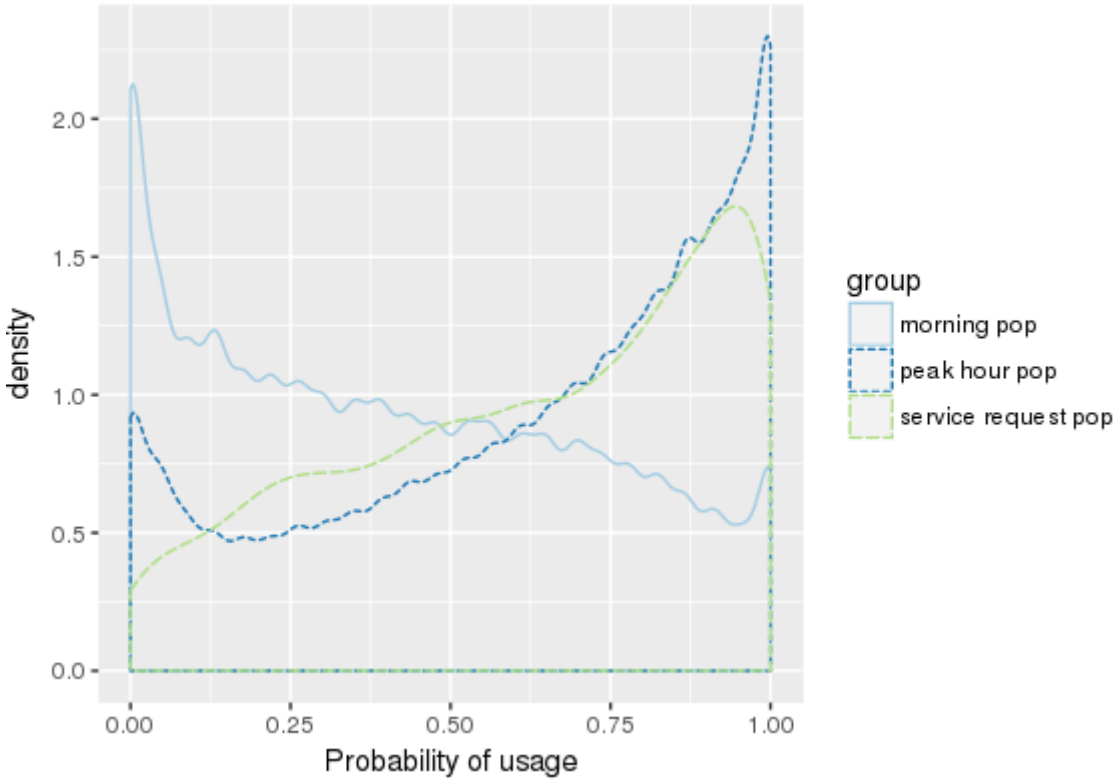


Figure 4.10 – Density plots of the probability of usage variable

It is possible to see that clients who filled a SR had a distribution in the hour of the service request more similar to the distribution of all clients in the peak hour (vs. the non-peak hour). Nevertheless, there was a smaller percentage of clients with a probability equal or less than 0.1 in the population of clients with a SR than in the population of all clients in a peak hour. We note that, while most SR occurred in a peak hour, they occurred in all hours of the day.

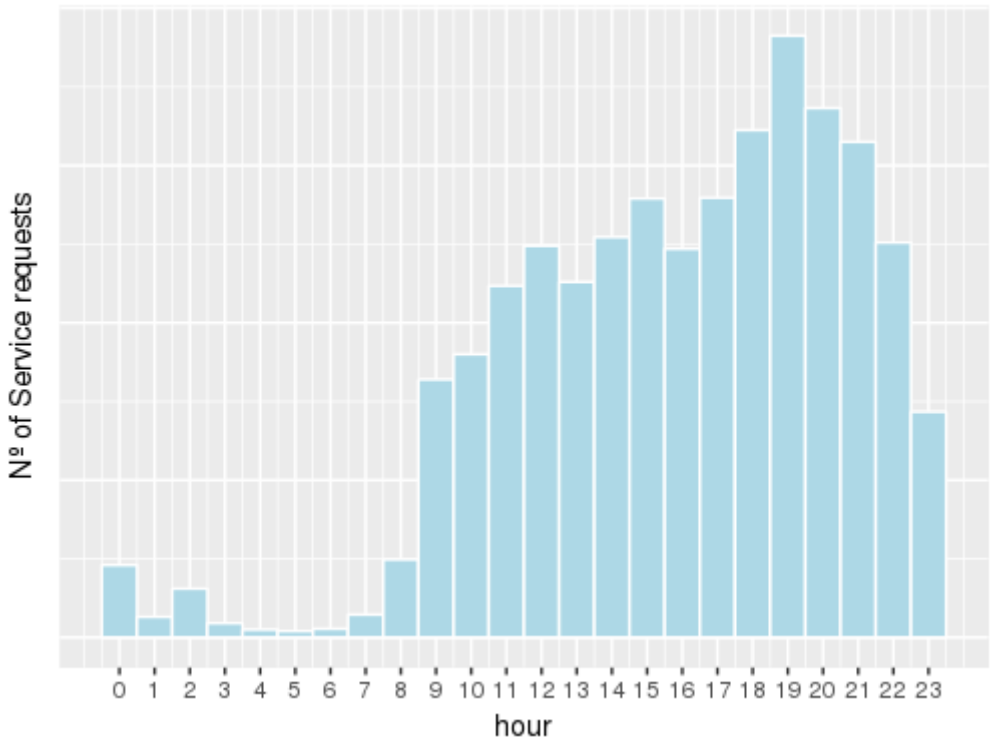


Figure 4.11 – Histogram of the service requests per hour, for the month of January

These results were encouraging, and, therefore we had one more feature to include in the model to identify customers with no access to the internet. Even though we used a cutoff point of 0.5 to predict usage in a day of February, we decided to include the probability of usage as a continuous variable in the model, so that we would not create artificial binning in the variable. For instance, if we flagged clients with a probability of less than 0.5 as “0” and all the others as “1”, we would lose information and pack probabilities as dissimilar as 0.50 and 1 in the same bin, while simultaneously saying that a probability of 0.49 is much different from a probability of 0.50. By keeping the variable as continuous, we let the model pick the cutoff that would be optimal for the model performance. We will present the development and evaluation of this model in the next chapter of the thesis.

5. IDENTIFYING CLIENTS WITH NO ACCESS TO THE INTERNET

The goal of the company was to identify the clients who may have experienced lack of access to the internet at a given time. In order to do so, we had technical information that came from the network monitoring systems. This information pertained to aspects such as the quality of the upstream and downstream signals and the state of the modem in a given hour (e.g., online, offline or in partial mode). As described in the previous two empirical chapters, we also had two additional sources of data. We included data about the segment the client belonged to in terms of his or her typical internet usage (see chapter three for a description of these segments); and information about the likelihood that the client would try to use the internet at a given hour (see chapter four for a description of how the variable was calculated and validated). We note that all data was anonymized, and we had no away to trace back any of the information to specific clients.

5.1. DATA GATHERING AND CLEANING

For this part of the work, we gathered data using Hive, which is a data warehouse software for Apache Hadoop for "reading, writing, and managing large datasets residing in distributed storage using SQL" (The Apache Software Foundation, 2018). We then explored the data and developed the model using R, which is both a programming language and an open-source environment typically used in data analysis and visualization (R Project, 2018). We selected data only from individual clients with an active internet usage provided by cable modem, since service provided using other network technologies were monitored using different systems.

We also cleaned from the dataset other cases. Namely:

a) We removed the cable modems that never appeared in the registers of one of the monitoring systems, because these were probably cable modems not in use;
b) We removed clients that started being a client after the day considered for the dataset;
c) We removed cable modems from clients whose service was suspended due to lack of payment.

This led to a dataset containing approximately one million rows.

5.1.1. Target variable

We did not have direct information about the clients who had experienced no access to the internet. The most similar information we had that could be used as a ground truth was the list of clients who had filled a service request due to lack of access to the internet. While at a first glance this could seem the same, there is an important difference: not all clients complain when they face service issues (Garín-Muñoz, Pérez-Amaral, Gijón, & López, 2016; Nimako, & Mensah, 2012). Therefore, it is likely that the clients who complained are only a subset of the clients who experienced trouble with the service. This means that some of the clients marked as zero (because they did not fill a service request) should have been marked as one, in the sense that they experienced no access to the internet. Keeping this limitation in mind, we moved on to create the population on which we would train the model.

5.1.2. Population without experiences of no access

Clients who had filled a technical service request classified by the call center as a “no access to the internet” service request during the month of February were classified as positive cases in the model (i.e., were marked as “1”). However, we still needed to create the negative cases (i.e., cases marked as “0”) for the model to learn the patterns distinguishing the two. This was less straightforward than one might expect, because, as mentioned in the introduction, people who do not complain may still experience no access issues, due to individual variables related, for example, with personality and internet usage.

We experimented with two approaches to generate the negative cases. The first approach was to take all clients who complained of no access to the internet only once in February and never in January and get their data exactly two weeks prior to the complaint. This way, we controlled for personal differences that might influence the decision to fill a service request (because “0” and “1” were the same clients).

The second approach considered all clients who did not complain about no access to the internet in the month of February as zero. To generate this dataset, we got all clients with an active internet subscription and filtered out the ones that filled a service request in February. To gather the network measurements, we selected two time points: one during a peak hour in a non-working day and one during a non-peak hour during a working day. We gather data for half of the clients on first day and for the remaining clients in the second day.

It was possible to select a third, and maybe preferable approach. Instead of selecting a random day, we would match each positive case to a subset of negative cases at the same day and hour, controlling factors related to the monitoring systems that could affect all clients in that particular time point. However, it was not possible to follow this approach due to business-related demands.

The approach where the negative cases were all the clients who did not fill a service request in February (i.e., the second approach) proved to yield better results (for more details on how we evaluated the models, please see section 4.4). All the results described in the following sections were obtained using this approach.

5.1.3.Input variables

The monitoring of the Hybrid Fiber Coax (HFC) network is done through three main systems, which monitor different points of the network, as is represented in the following diagram of the customer premise equipment (CPE):

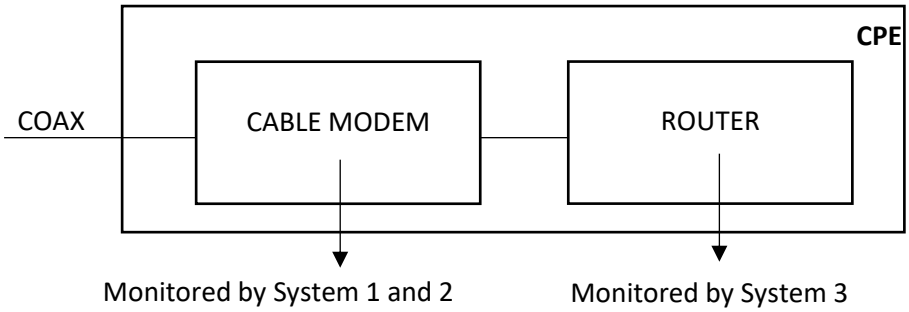


Figure 5.1 – Monitoring points between the cable modem and network monitoring systems

For identifying clients without access to the internet, we used network information from the three systems. This information broadly corresponds to measures of the quality, quantity and reliability of the data transmitted through the network, as well as information about the state of the cable model and if it lost its connection (totally or partially) to the network at any point.

The following table systematizes the input variables used in the model, from each of the three systems. Note that a field in the table may correspond to more than one variable as, typically, the same variable is measured for upstream and downstream traffic.

Monitoring system	Variables	Description
System 1	Signal-to-Noise Ratio (up and downstream)	Signal level of the network; compares the level of the signal the CM is receiving to the level of background noise.
	Transmitting power Receiving power (up and downstream)	Power with which a modem transmits its signal. Power of the incoming signal level being received by the cable model
	Codeword errors (up and downstream)	Codeword error rate of all packets received/sent from the cable modem
	Correctable codeword errors (up and downstream)	Correctable codeword error rate of all packets received/sent from the cable modem
	Traffic volume (upstream and downstream)	Traffic generated by the cable modem
	Resets	Number of times the equipment lost connection to the system
	Occupancy rate (upstream and downstream)	Interface capacity in use, over the total capacity of the interface
	State	State of the modem in the system
	Partial mode downstream	Whether the cable modem was partially online (that is, some cable modem carriers were not connected to the system) for downstream traffic
System 2	State of the modem	State of the modem in the system
	Partially online for upstream traffic	Whether the cable modem was partially online (that is, some cable modem carriers were not connected to the system) for upstream traffic
System 3	Nº of events	Nº of times the cable modem lost and/or recovered the connection to the system

Table 5.1 - Input variables included in the model

Gathering the measures in a consistent way was one of the challenges we encountered. Information on each system was measured and stored using different rules. For instance, in one system the data was measure roughly every hour and was stored under a timestamp with minute precision. For another system, data was collected three or four times during an hour, but only one of those measurements is retained and stored under a timestamp with hourly precision – we had no information of which one or the minutes it was taken. Finally, in the third system data was only collected if there was a change in the status of the cable modem in the system, i.e., if the modem lost or recovered its connection to the system. The data was stored in a timestamp with precision to the minute. In the end, we dealt with these differences by:

- Keeping the last two measurements of the first system;
- Keeping the measurement taken at the hour where the service request was filled (even if it meant that it was possible the service request occurred before the measurement was taken) and the previous hour for the second system;
- Counting the number of events that were registered in the third system on each of the hours considered in the model.

We decided not to aggregate data in any way (for example, to compute the average of numeric variables in the hours before) because typically clients complain within a short interval from the time where the internet goes off. We also noted that we tested whether including an additional hour of measurements (i.e., considering three hours before the service request, instead of two) would improve the model. Even though the model performed better when three-hour measurements, the improvement was small (0.001 points in accuracy) and did not justify the increase in computational cost. Conversely, if we consider only one hour of measurements in the model, the accuracy would be reduced in 0.01 points. For this reason, we kept using the measurements corresponding to two hours in the model.

Besides the variables gathered from these three systems, we also included the information of which segment the client belonged to (see chapter 3 for more details) and the probability that the user would be using in the considered hour (see chapter 4 for more details).

5.2. DATA EXPLORATION

We started by exploring the data, aiming to understand:

- How the three systems were related among themselves (e.g., when a cable modem is offline in System 1, is it also offline in System 2? Can a modem be offline in System 2 and still be registered in System 3?).
- How the data from clients with “no access” complaints differed from the data from clients who did not complain of no access in key variables from each monitoring systems.
- What is the distribution of values for each variable? What percentage of data is typically missing in each variable?
- How can we further transform each variable?

5.3. ALGORITHM SELECTION AND IMPLEMENTATION

To solve our binary classification problem, we chose to use a gradient boosted tree model (see chapter 2, theoretical framework, for more details on this algorithm). Specifically, we used the implementation of CARET R package of a gradient boosted model. CARET (Classification And REgression Training; Kuhn, 2018) is one of the most widely used machine learning packages in R. More specifically, CARET utilizes several R packages that implement a variety of algorithms, making the process of applying different algorithms more streamlined.

In addition, CARET has the following features:

- Allows tuning the model parameters through a search grid; the best parameters are selected using cross-validation;
- Allows different methods of sampling within the model implementation (see section 5.5.1. Dealing with an unbalanced dataset);
- Provides tools to evaluate the model that are easy to implement;

Within CARET, we chose to use the implementation of GBM, which uses the GBM package. GBM implements extensions to AdaBoost algorithm and Friedman's gradient boosting machine (Ridgeway, 2017). For more information about gradient boosting models, please see chapter 2 (theoretical introduction).

Also using CARET, we implemented a CART decision tree and random forest algorithm (see chapter 2, theoretical introduction, for further details about these models). See table 5.3 in section 5.6.2. for the comparison of the models.

5.4. MODEL DEVELOPMENT

5.4.1. Dealing with an unbalanced dataset

The number of clients who complained about having no access to the internet in a given month is a very small fraction of the total number of internet clients (typically, less than 1%). This meant that we ended with a very unbalanced dataset, where more than 99% of the cases were negative cases and less than 1% were positive cases. In practice, an algorithm that classifies everyone as not having a complaint would still have an accuracy of 0.99, an accuracy level that would be very hard to beat by any other strategy that attempts to differentiate between the positive and the negative cases (Provost, 2000).

To deal with this problem, we again used two different approaches. The first approach was to re-build the training set by simply randomly under-sampling the cases classified as “0” (i.e., the negative cases) to form a sample of roughly the same size as all the cases classified as “1” (i.e., the positive cases). In order to evaluate the algorithm with conditions more similar to reality, we kept the test set unbalanced.

The second approach is to use SMOTE. SMOTE is a technique that oversamples the minority group. To overcome problems in over-fitting that could arise from simply oversampling with repetition the minority group, SMOTE uses k-nearest neighbors to generate new synthetic cases. More specifically, the algorithm generates cases that lay in between each minority case and its nearest neighbors, by

computing the difference between the feature vector of a case and the feature vector from one or more of its neighbors, multiplying it by a random number between 0 and 1 and adding it back to the feature vector of the original case (Chawla, Bowyer, Hall, & Kegelmeyer. 2002; Kotsiantis, Kanellopoulos, & Pintelas, 2006). SMOTE can also be combined with random undersampling of the majority.

Since finding the k-neighbors in SMOTE is based on distance, the results might change if we scale the data prior to running the algorithm, yet another approach we tried.

As mentioned, one of the features of the CARET package is to allow for different sampling techniques when training the model. Thus, we implemented the three sampling approaches directly using the package. The models developed using SMOTE performed better than the model using only downscaling of the majority sample. There was practically no difference between the performance of the algorithm when using SMOTE with and without previously scaling the data. For simplicity, we used SMOTE without data scaling on the final model. See table 5.3. in section 5.6.2. for the comparison of the models.

5.4.2. Dealing with missing values

Data was stored under the same timestamps for all the cable modems in the three systems. Therefore, we could know if a modem was supposed to have a measurement for a given hour or if, for some reason, that measurement was missing. In this particular problem, missing values were very informative, because they could be due to problems in the network. Therefore, imputing them using the mean or through other variables was not a good option.

The implementation of GBM we chose allowed us to pass the missing values to a third node on each of the generated trees (i.e., each split in a given tree originates a left node, a right node and a missing node; Ridgeway, 2017). Alternatively, we could set the missing values as additional levels in the categorical variables and impute them with a number that we knew was outside of the variable range for numeric variables. We tested both approaches and realized that the latter approach had a better performance on unseen data.

5.4.3. Tuning model parameters

The caret package GBM implementation allows the user to define a grid with several parameters ranges for the algorithm to test. Specifically, the package will run multiple times with every possible combination of parameters and use cross-validation and ROC to select the best combination of parameters (for a brief overview of the model parameters see chapter 2, theoretical framework). However, this method is very resource intensive, limiting the number of parameters that can be tested in a reasonable time. More specifically, we tested the following parameters[4] to tune the model:

- Interaction depth: 8, 9 and 10
- Number of trees: 800, 900, and 1000

[4] These parameters were choose based on previous versions of the model, where we tested a broader range of values.

- Minimum number of observations in each node: 10
- Percentage of sample to fit each tree: 50
- Shrinkage: all numbers between 0.05 and 0.1, with increments of 0.01

We used a 10-fold cross-validation to test these parameters. The best model was chosen based on the area under the receiver operating characteristic curve (AUROC) value obtained with each combination of parameters, across the ten folds. AUROC has the advantage of being less sensitive to class imbalance than accuracy, which for very unbalanced datasets would be very high even for very low values of sensitivity (Horton, 2016).

Recall that the receiver operating characteristic curve (ROC curve) is the plotted values for sensitive and specificity for different cutoff points. AUROC is the area below that curve. A value of 0.5 for a binary classification means that the model performance is at chance level; a value of 1 corresponds to a perfect classifier. Accuracy is the number of cases correctly classified by the model, out of the total cases. Sensitivity (also called recall) is the number of cases correctly identified as positive out of all positive cases. Specificity is the number of correctly identified negative cases out of all negative cases. Precision is the number of true positives of all cases identified as positive. F1 is the harmonic mean of precision and recall (see also chapter 2, Theoretical Framework).

5.5. MODEL EVALUATION

5.5.1. Validating the model

With the training set, we used a 10-fold cross-validation to choose the best training parameters, as mentioned. We defined that the metric that the model would try to optimize would be AUROC. This means that the parameters chosen by cross-validation would be the ones that maximize AUROC.

The best model had an interaction depth of 10, 1000 trees, shrinkage of 0.1 and a minimum of 10 observations in each node. Across the 10-folds, the model had the following performance (considering a 0.5 cutoff point):

Measure	Average	Standard Deviation
AUROC	0.985	0.002
Accuracy	0.978	0.001
Sensitivity	0.868	0.017
Specificity	0.979	0.001
Precision	0.240	0.005
F1	0.376	0.008

Table 5.2 – Cross-validation results

The model obtained very good values across the metrics, with an accuracy close to 1. It obtained the lowest scores on precision, which is not surprising considering that the dataset was very unbalanced. The small standard deviation obtained across the measures suggests that the model is robust for different partitions of the data. In addition, the average values for AUROC, sensitivity and specificity suggest that the model will be good at predicting people who filled a service request due to lack of access to the internet.

5.5.2. Comparison with other models

Below, we systematize the comparison in the performance between the final model and other models tested (SD is in parenthesis). All GBM were ran with the same parameters. The random forest values were the best obtained, after trying with three different sizes for the subset of variables used in each tree. The decision tree served as the benchmark model. We only varied the complexity parameter of the tree, which did not affect the performance of the model, because having an additional partition in the tree would reduce the error in only a very small fraction.

	AUROC	Accuracy	Sensitivity	Specificity	Precision	F1
GBM with SMOTE (Final model)	**0.985 (0.002)**	**0.978 (0.001)**	**0.868 (0.017)**	**0.979 (0.001)**	**0.240 (0.005)**	**0.376 (0.008)**
GBM with SMOTE on scaled data	0.985 (0.002)	0.978 (0.001)	0.869 (0.018)	0.979 (0.001)	0.241 (0.006)	0.377 (0.008)
GBM with down-sampling	0.981 (0.002)	0.956 (0.002)	0.906 (0.012)	0.957 (0.002)	0.140 (0.005)	0.243 (0.008)
Random forest with SMOTE	0.962 (0.004)	0.971 (0.001)	0.822 (0.020)	0.972 (0.001)	0.188 (0.005)	0.306 (0.008)
Decision tree (benchmark model)	0.865 (0.009)	0.937 (0.003)	0.776 (0.017)	0.938 (0.003)	0.089 (0.003)	0.159 (0.005)

Table 5.3 – Comparison of the performance of different algorithms

5.5.3. ROC CURVE and model output

The GBM model outputs a probability of being a positive case for each of the cases in the train set. For classifying a case as positive or as negative, we need to define a cut-off point on that probability. Any cases with a probability higher than the threshold would be classified as a positive case; cases with a probability lower than the threshold would be classified as negative cases. Below, we plot of the ROC curve, with the predictions across the 10-folds. The plot represents the values of specificity and sensitivity when different probability cut-offs are considered.

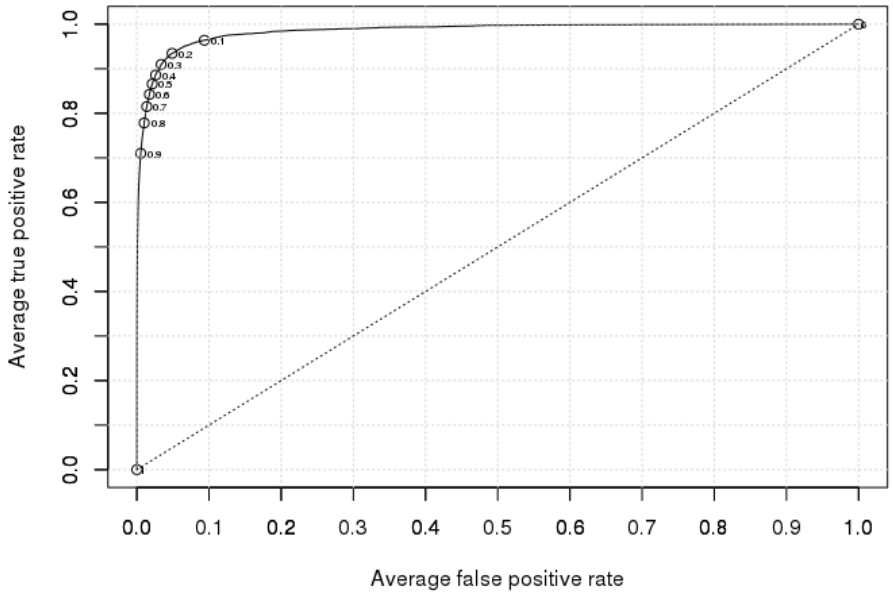


Figure 5.2 – ROC curve

However, we note that the ROC curve is not sensitive to class imbalance in the dataset. A decrease of a decimal point in sensitivity would represent much less people than a decrease of a decimal point in specificity. From a business perspective, the cost of false positives and false negatives needs to be defined when defining the threshold above which the company should intervene on a client. Different costs for the false positives and the false negatives could be defined, assuming that it could be, for example, be costlier to contact a client who did not experience lack of access to the internet than to miss the chance to make amendments to a client who had experienced a bad service. However, it was a business decision to attribute the same cost to false positives and false negatives.

We also looked at the distribution of the probability attributed by the model, for the positive and the negative cases.

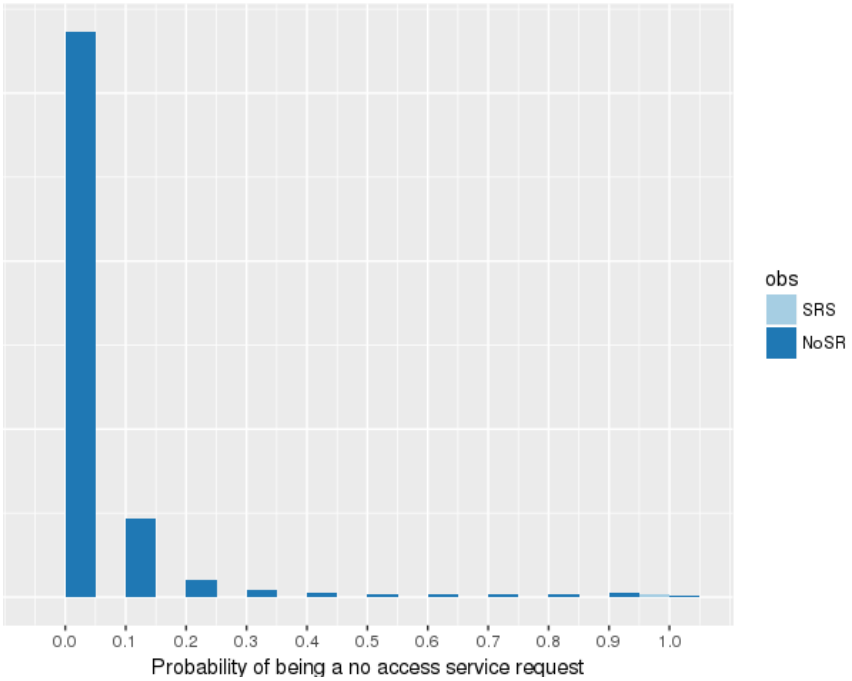


Figure 5.3 – Distribution of positive and negative cases according to the probability of being a positive case attributed by the model

We note that the majority of the negative cases (no SR) were classified by the model as having a very low probability of being a client who filled a service request by no access (93% of the cases had a probability lower than 0.15). Given the small proportion of cases in the test set that were positive cases (SRs), we zoomed in the plot to grasp their distribution better (see below).

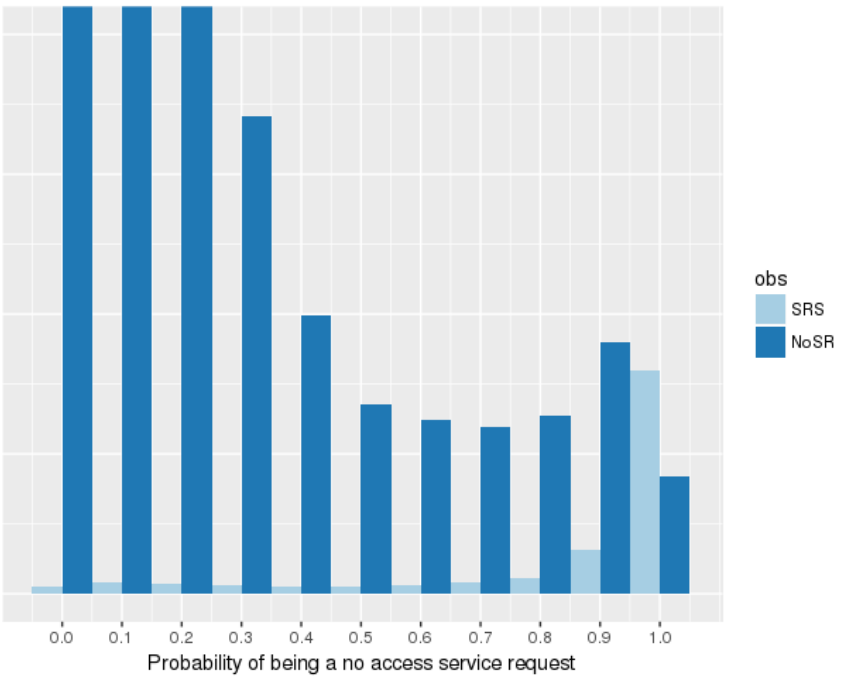


Figure 5.4 - Distribution of positive and negative cases according to the probability of being a positive case attributed by the model (limiting the y-axis)

The vast majority of SR cases (74%) was classified with a very high probability by the model (i.e., probability higher than 0.85). We also note that less than 1% of the No SR cases were classified with a very high probability. Similarly, 5% of the SRs were classified with a very low probability (< 0.15).

We will analyze these extreme cases in the next section. Since it is hard to visualize and interpret GBM results, we will rely on the analyses of key variables of the model, to gain some insight on the decisions made by the model.

5.6. MODEL INTERPRETATION

To better understand the produced model, we started by looking at the variable importance of the model, which we plot next.

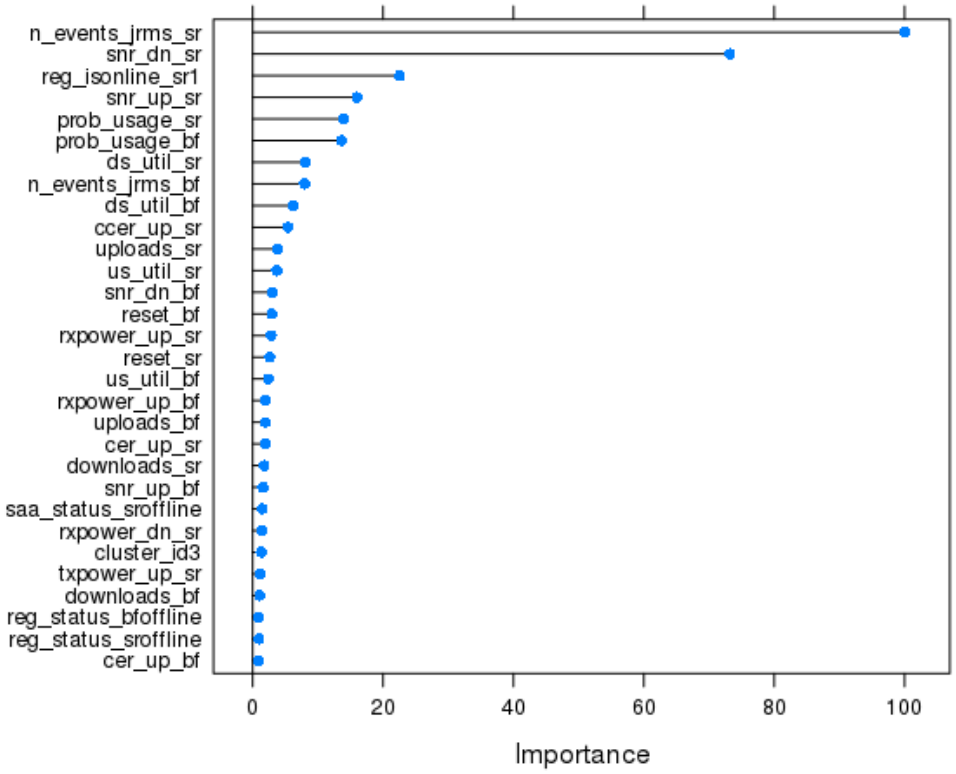


Figure 5.5 – Scaled variable importance for the GBM model (top 30 variables).

The majority of the top 10 variables in terms of importance were the measurements taken at the hour of the service request. Of these, the number of connects and disconnects and the signal-to-noise ratio in the hour where the service request was made were the most important variables, by a relatively large margin. Interestingly, the top three variables came from the three different monitoring sources considered.

The impact of each variable can be estimated using partial dependence plots. As mentioned in the introduction, partial dependence plots are graphical translations of the prediction function, helping to visualize the relation between one or more predictors and a) the target, for regression problems and b) the class probability, for classification problems. Importantly, partial dependence plots take into account the average effect of other predictors (Greenwell, 2017). Below, we present the plots for the top three variables.

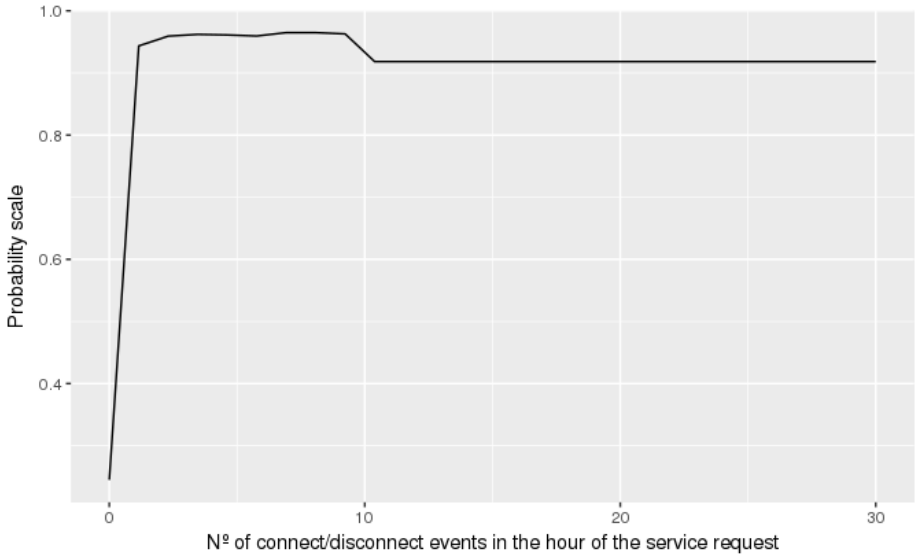


Figure 5.6 – Partial dependence plot for the number of events variable

If there were no registered connect or disconnect events, the probability that the client filled a no access service request was much lower and increased abruptly with one event. Above one or two events, the number of events registered seemed to matter little.

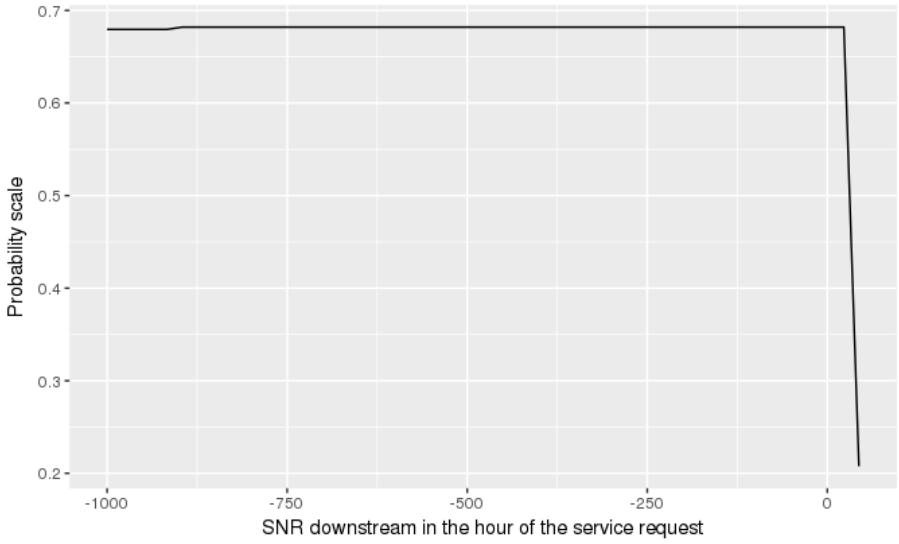


Figure 5.7 – Partial dependence plot of the Signal-to-Noise Ratio variable

To interpret the above plot, please recall that we replaced the missing values in numeric variables by -1000. What the above plot suggests is that the probability of being considered a cable modem with no access to the internet is much higher if the signal-to-noise ratio (SNR) measurement is missing. If we exclude the missing cases, the following plot is produced:

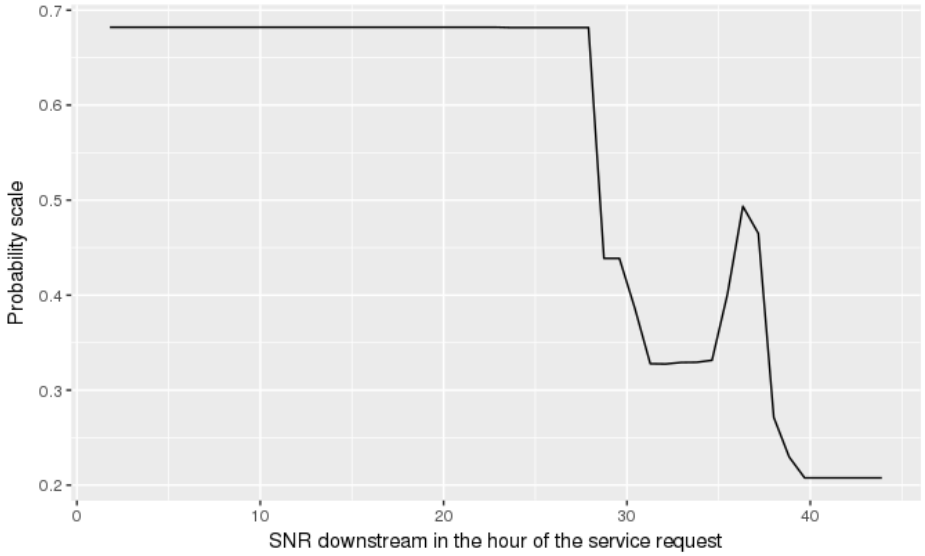


Figure 5.8 – Partial dependence plot of the signal-to-noise ratio variable, not considering missing values

The probability of being considered a service request was lower for higher values of SNR. This is not surprising, because the higher the SNR downstream, the better the signal. Typically, the SNR downstream value was considered acceptable when is equal or above 35. However, the partial dependence plot suggested a value slightly higher. We note that the drop in the probability of being a no-access service request when SNR downstream is lower than 34 might have been due to the low number of cases that had these values (only 1% of the cases had a value lower than 34, while 6% of the cases had a missing value; the remaining cases had SNR downstream above 34).

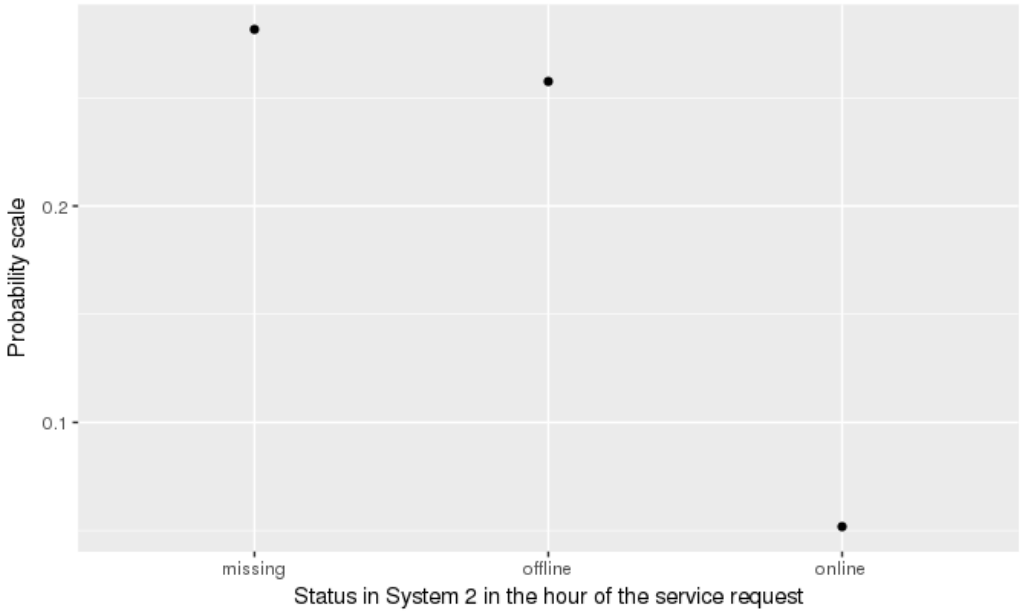


Figure 5.9 – Partial dependence plot of the cable modem status in System 2

If the cable modem is online, then it is less likely that it would be a no access service request, as expected.

5.6.1. Analyzing the false negatives

The no access to the internet service requests were further classified into subareas by the customer service (through an automatized system). By looking at those subareas, we could further understand the cases the model was misclassified. The plot below pictures this information.

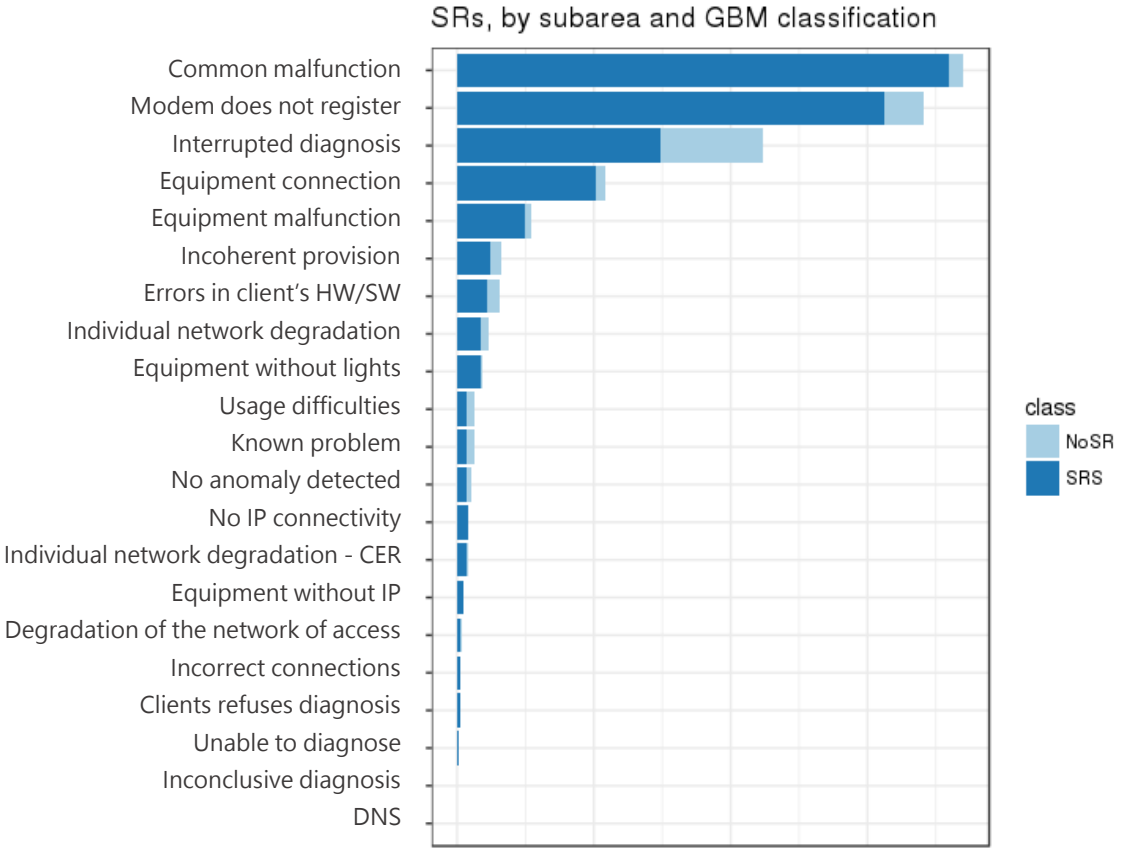


Figure 5.10 – Model classification by sub-area of the service request

Of the most frequent sub-classes of service requests, the model performs worst when the malfunction is classified by the customer service system as interrupted diagnosis. This type of classification happens when, for example, the client is away from the cable modem, is not available for completing the diagnosis at the moment, or if the call is terminated before the diagnosis is completed. Surprisingly, the model did quite well even when no anomaly was detected and when the problem was in the software or hardware of the client ("client's HW/SW errors"). This may be the case because we included in the model variables that capture connects and disconnects of the equipment from the monitoring systems. If the client thinks he or she does not have access to the internet, one of typical first behavior is to disconnect and reconnect the cable modem, to see if that solves the issue. The monitoring systems are not able to distinguish between reboots provoked by the client from reboots caused by malfunctioning. Thus, even if everything is working fine in the network, the model may be picking up the clients' behavior. In future versions, it may be beneficial to remove these categories of the service requests, as they may add noise to the model.

We also compared the distribution of the clients classified with a very high probability and the clients classified with a very low probability on the most important variables in the model: number of connects

and disconnects of the cable modem, signal-to-noise ratio, and whether the cable modem is online in one of the monitoring systems (System 2). The plot below shows these comparisons. The x-axis represents the percentage of people that has an abnormal value in each of the variables (a connect or disconnect event, a not online status or a signal-to-noise ratio below the acceptable threshold or not registered).

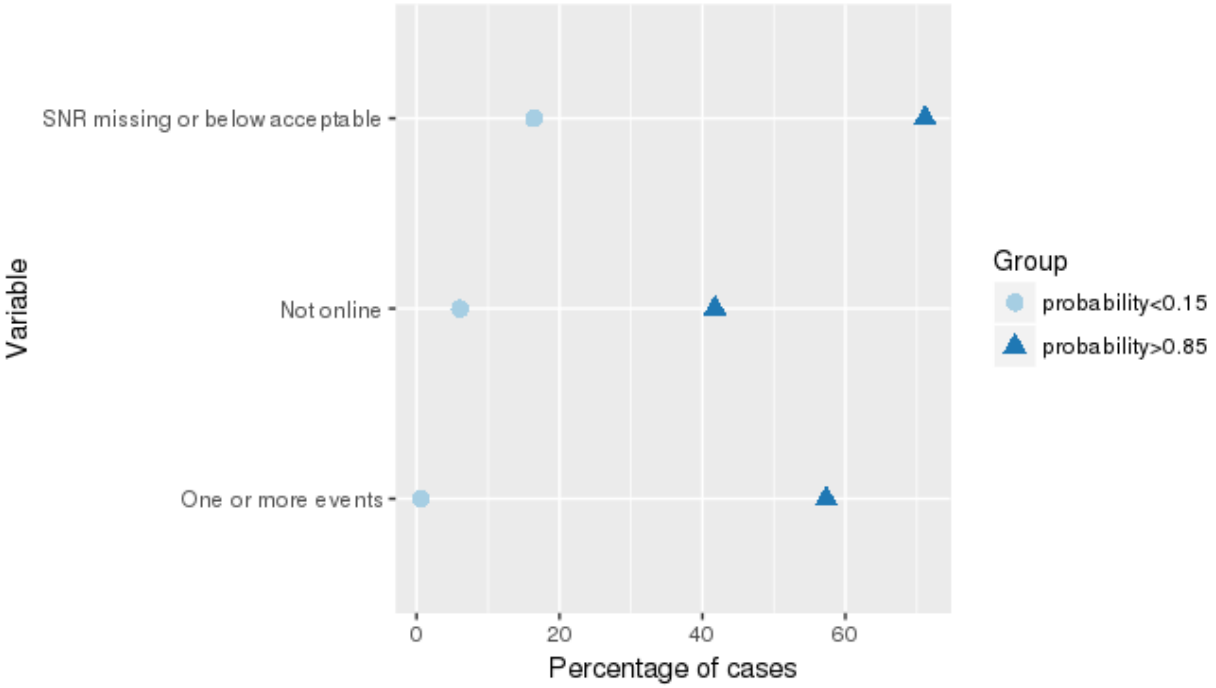


Figure 5.11 – Frequency of abnormal values in key variables, for the observed positive cases classified with lowest and highest probability

A much smaller percentage of the cable modems who were wrongly classified by the model has having a low probability of being a service request had abnormal values in the three variables. Less than 1% of the cable modems with a low probability had an event in SYSTEM 3, compared to almost 60% of the cable modems with a high probability. Similarly, more than 90% of the cable modems with a low probability had an online status and more than 80% of the modems had an acceptable value for signal-to-noise ratio. For cable modems with a high probability, these values were about 60% and 30%, respectively.

This data allows us to conclude that for some clients that filled a no access service request, the signals and state of the modem was as expected if everything was working normally, making it harder for the model to identify them. This is probably due to the fact that the cause of the lack of access to the internet was not in the network or in the cable modem, but in the clients' personal device, over which we have no visibility.

5.6.2. Analyzing the false positives

As mentioned, there was also some negative cases that were classified as having a high probability of being positive cases. We repeated the analysis we presented in the previous section, this time for the

clients that did not fill a service request. The plot below compares the negative cases that were classified by the model with a very low and a very high probability, on the three key variables of the model.

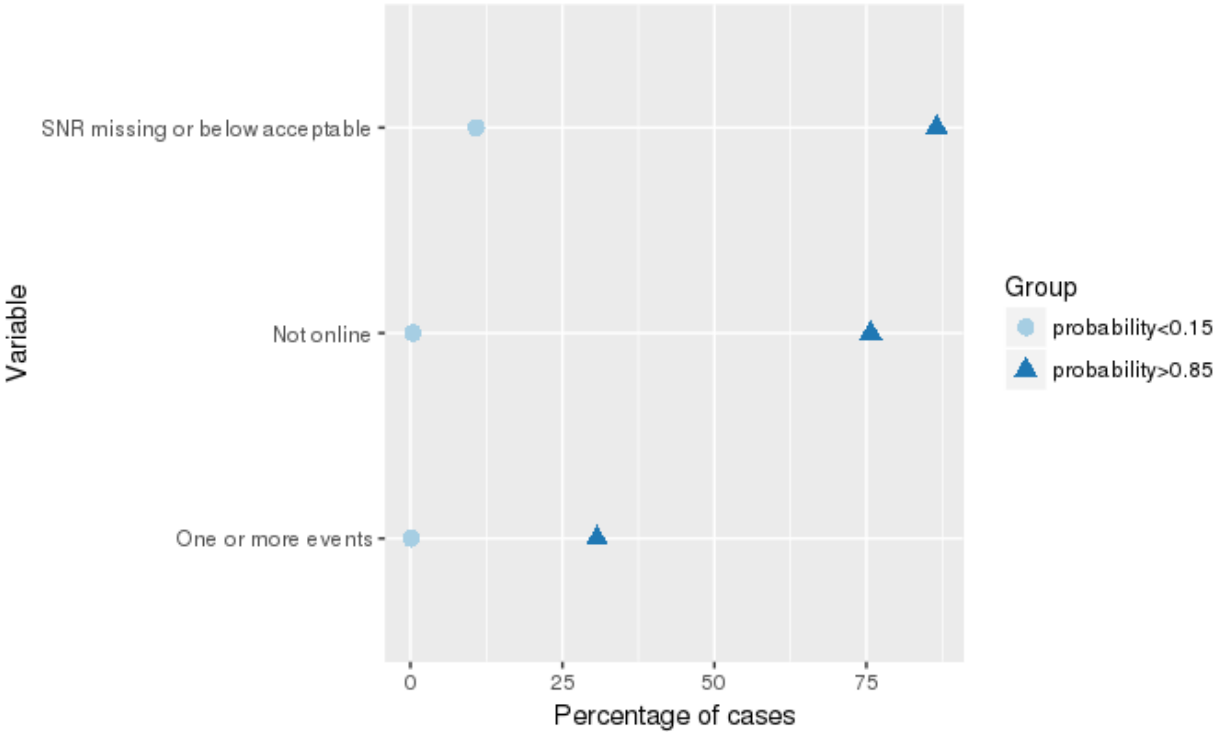


Figure 5.12 - Frequency of abnormal values in key variables, for the negative cases classified with lowest and highest probability

As expected, around 80% of the cable modems classified by the model with a very high probability had a signal-to-noise ratio value missing or below what was considered acceptable; and 75% of the cable modems were not registered as online. The percentage of cable modems with at least one connect or disconnect event was much lower (around 30%), when compared with the percentage of true positives with one or more events (close to 60%; see previous section). However, even this percentage is still much higher than the percentage of cable modems with at least one event to which the model attributed a lower probability (less than 1%).

Thus, the cable modems to which the model attributed a higher percentage had, indeed, some signs of malfunctioning. Besides being error of the model, these modems could indeed have some problems, but belong to clients that a) did not use the internet and did not realize there was a malfunction or b) noticed the problem but decided not to complain.

5.6.3. Analyzing the role of specific variables: probability of usage and segmentation of clients by internet usage

As presented in chapter 3 and 4 of the current report, we developed specific features to be included in the model. We computed a probability that the client would be using at the time under analysis and segmented the clients according to their typical usage. The model picked up both the variables, albeit with different importance. The probability of usage in the hour of the service request and the hour before appeared in the 5th and 6th position, respectively, and the cluster variable appeared in the 25th position, out of 46 variables (see figure 5.5).

5.6.3.1. Probability of usage variables

The partial dependence plots for the probability of usage variables were hard to interpret. We begin by examining the partial dependence plot for the probability of usage in the hour of the service request. The conclusions taken for the hour before the service request were similar as the ones presented.

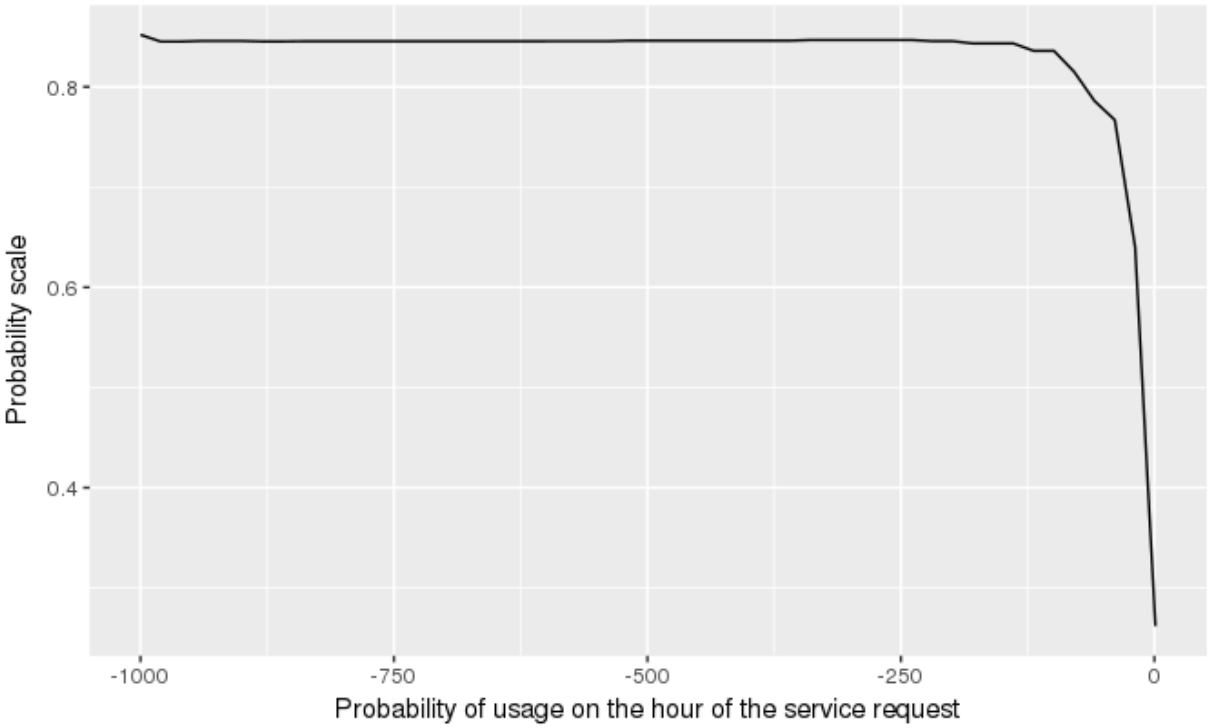


Figure 5.13 – Partial dependence plot for the probability of usage in the SR hour

Since we re-coded the missing numeric values as -1000, this plot shows that clients who did not have a probability of usage were more likely to be classified as a no-access service request. Clients without a probability of usage were clients whose cable modem was never connected to one of the monitoring systems in the previous months. They could be new clients or cable modems that were shut down for the entire month. A slightly higher probably for these clients is also intuitive, since these are likely new clients who are still getting to know the equipment, and/or unused equipment that might have some configuration issues.

We redid the plot without the missing cases, to allow the visualization of values between 0 and 1. To help interpreting the values, we smoothed the partial dependence line, using a LOESS regression. The grey area represents the confidence interval for the smoothing function.

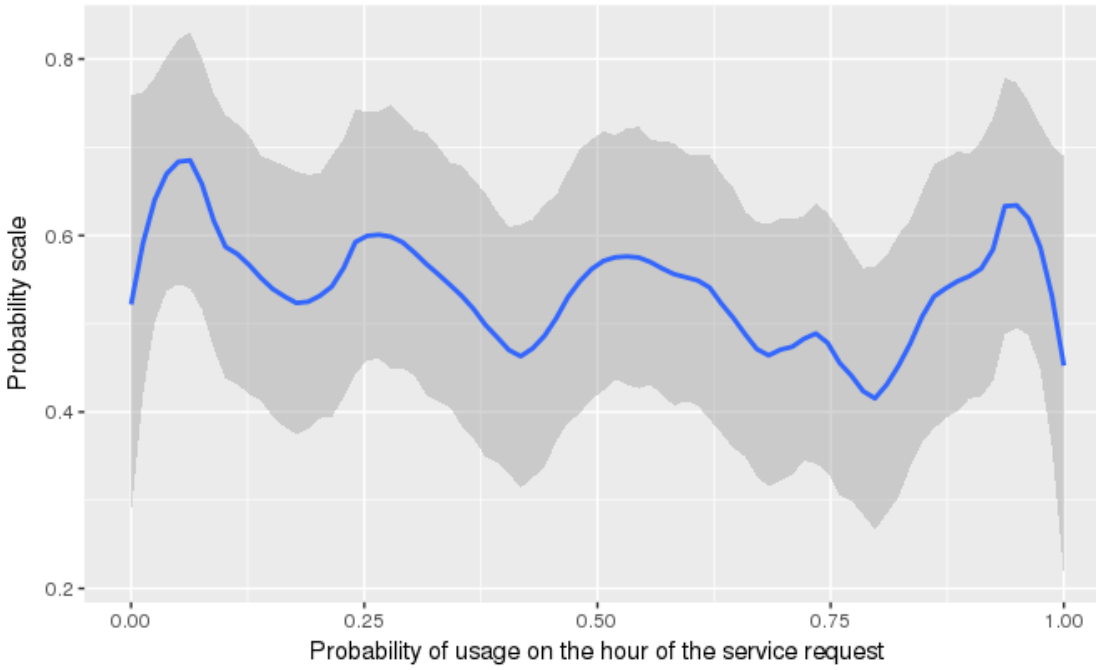


Figure 5.14 – Partial dependence plot for the Probability of Usage in the hour of the SR

At first glance, it seems that the probability of being a service request was lower for the extreme probabilities (0 and 1). The probability was higher when the probability was close to 0.9 but lower than 1, or when the probability was close to 0.1 but higher than 0. This is probably due to the way the negative cases population was build: half of the population came from a peak hour where most people had a very high probability of usage, and half of the population came from a non-peak hour (a morning of a working day), where the probability of usage was typically low. This was not the case for the population with a no-access service request.

The scatter plot of the probability of usage and the probability attributed to the model also helps to understand these values.

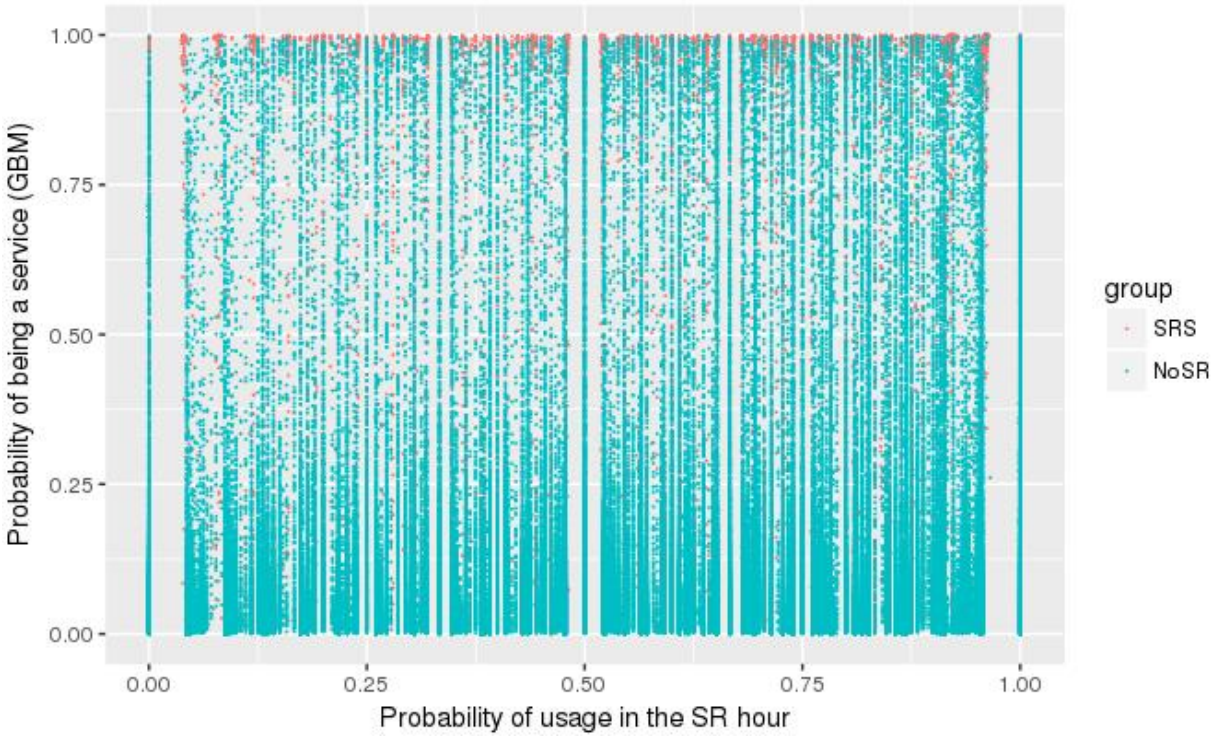


Figure 5.15 – Scatter plot of the probability of usage in the hour of the Service Request and the probability attributed by the model, by group

To try to shed more light to the role of the probability of usage in the model, we build a different plot. The plot below shows the proportion of cases that were classified as positive by the model minus the proportion of observed positive cases. Of course, this plot should be interpreted with care, since it is merely looking at the distribution of cases according to their probability of usage, without taking into consideration other variables in the model.

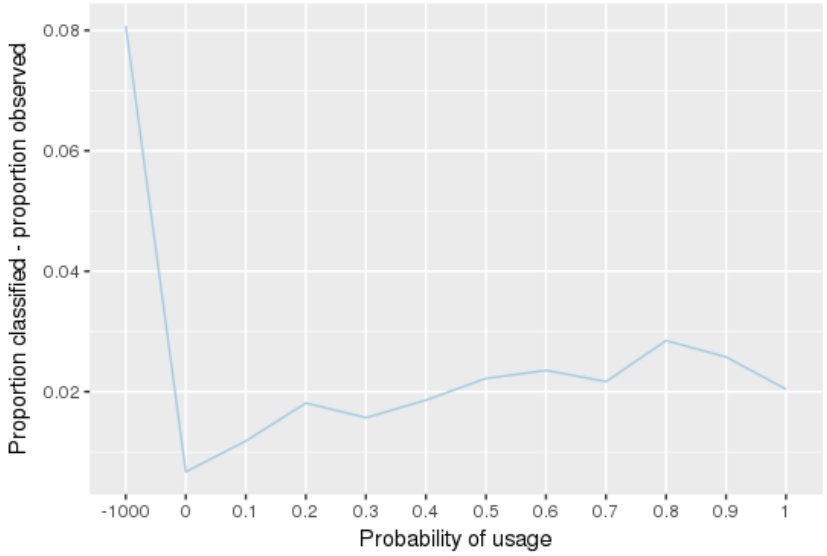


Figure 5.16 – Difference in the proportion of cases classified and observed as positive, by probability of usage in the hour of the service request

Two main points can be taken from the above plot. First, the proportion of cases with a missing probability of usage that were classified as a service request by the model is much higher than the actual proportion of service request cases in the dataset. Second, the proportion of cases classified as positive cases tends to increase with higher values of the probability of usage.

5.6.3.2. Segmentation of internet users variable

As mentioned, the variable that was produced through the segmentation of internet users had a relatively small importance. The partial dependence plot does not change a difference in the probability of being a service request among the levels of this variable. Note that cluster 1 and 2 corresponded to the clusters with higher usage, cluster 3 corresponded to the cluster with lower usage and cluster 4 corresponded to the cluster with moderate usage (see chapter 3 for more details).

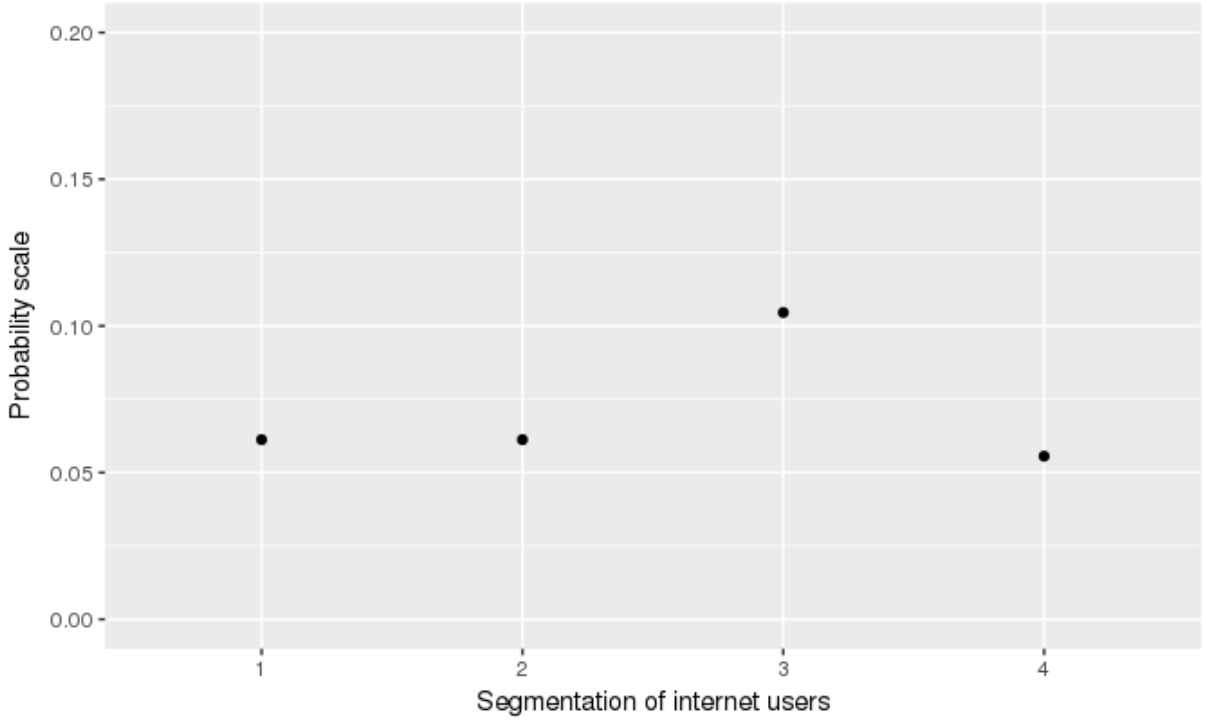


Figure 5.17 – Partial dependence plot for the variable corresponding to the segmentation of internet users

Surprisingly, cluster 3 was the cluster with higher probability of being a no access service request. We were not expecting this result, considering that this cluster corresponds to the lowest and less frequent usage. Therefore, we further analyzed the data.

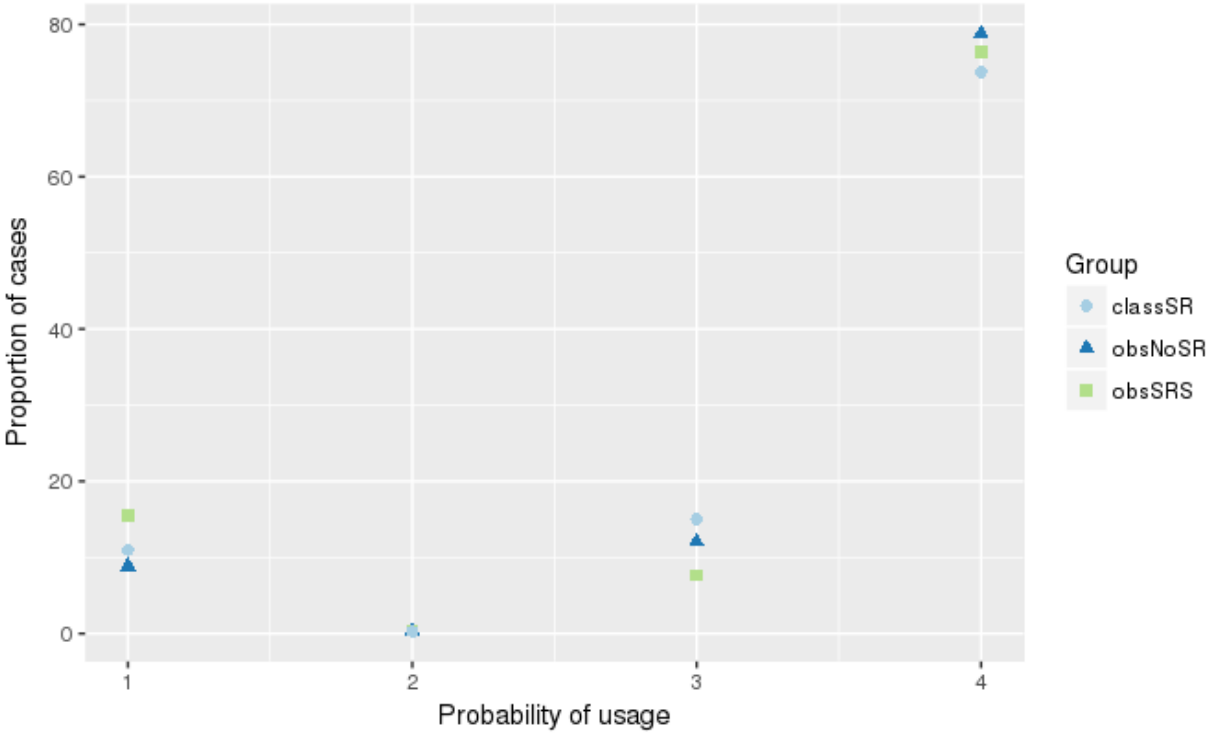


Figure 5.18 – Comparison of the proportion of clients on each cluster, for the population of clients with a no access service request (obsSRS), clients without a service request (obsNoSR) and clients classified as having a service request by the model (classSR)

Comparing the distribution of clients with and without a no access service request, we see, as expected that the proportion of clients in cluster 3 is smaller in the population with a no access service request than in the population without such service request. Conversely, the proportion of clients in cluster 1 (cluster with a high usage) is bigger among clients with a service request.

However, for cluster 3, the proportion of clients classified by the model as having a service request was almost the double of the actual observed proportion (15% vs. 8%). A possible explanation for this bias in the model is that cable modems of clients that are secondary cable modems, or cable modems that are seldom used, were attributed to this cluster (see chapter 3). If the cable modem was disconnected because it is not in use, its status in the monitoring systems would also be “abnormal”, and difficult to distinguish from a cable modem that was experiencing technical difficulties.

6. DISCUSSION

The work presented in the current report testifies the broad nature of the work that I had the opportunity to develop during the internship. Below, we discuss the results of that work.

6.1. SEGMENTATION OF INTERNET CLIENTS

In the first half of the internship, we employed non-supervised learning tools to develop a model to segment the company's clients based on their internet usage. The main challenge of this analysis was the scarcity of the information available. In fact, we only had information about the hourly upstream and downstream traffic generated by each client: we had no data on the type of activity the client was doing online, or even simpler variables such as the upstream or downstream velocity and the duration of each usage session. This limited the depth of the analysis, and how discriminant the model could be. As a result, the analysis produced only four clusters, mainly distinguishable by the volume of usage. In addition, around 75% of the clients ended up in one cluster, with moderate usage, which is another limitation of the analysis. These results are not surprising: Kihl et al (2010) also used almost exclusively variables related to daily upstream and downstream volume of traffic; in their analysis, 80% of the internet users were in a moderate usage cluster.

In our cluster analysis, we relied on the silhouette values and on the analysis of the within-cluster sum of squares values to define the number of clusters to retain. However, we could have further validated the number of clusters by analyzing other measures such as the gap statistic, which is a limitation of the current work. We did not use the gap statistic due to computation limitation and time constraints.

However, we note that our clustering solution had acceptable silhouette values, meaning that, for each cluster, the points on that cluster were closer among themselves than among points in other clusters. This suggests that the analysis was able to identify a latent structure in the data, and therefore, correctly characterized the dataset.

In addition, results were stable in time. Almost 90% of the cases were classified in the same cluster in the analysis done with data from four months apart. The least stable cluster was the cluster with high volume of upstream traffic, which was the cluster with the smallest percentage of the population. Future analysis could test the solution were the two clusters with higher usage were merged, for the sake of the stability of the solution.

Importantly, the clusters proved to have external validity, differentiating among clients in variables that were not used in the cluster analysis. Clients in the clusters with higher usage tended to submit more technical-related service requests. This was probably because clients that use the internet more are more likely to notice issues and are also more demanding in terms of quality of service. Clients in clusters that had more usage were also more likely to have subscribed to a mobile internet service, suggesting that these were, indeed, clients who use and value the internet more. This result was crucial to validate the analysis, since we used the segmentation results as a feature of the model that identified clients who made a service request due to lack of internet service.

Another way to validate the cluster solution was to use a discriminant analysis, which is a multivariate technique to find a function (called discriminant function) that best separates the cases into groups (clusters). This technique is a way to validate the cluster analysis results because if the clusters are

meaningful, there should be possible to use the features to classify each case into one group (Oliveira et al., 2007). Thus, implementing a discriminant analysis can be an important future task for this line of work.

Finally, we could have tried other clustering algorithms that are able to handle big data, to explore other clustering solutions. Although we did try to segment the dataset using spectral clustering, we could not do so efficiently given the volume of data we had.

6.2. IDENTIFYING CLIENTS WITH NO ACCESS TO THE INTERNET

In the second part of the work described in this report, we used a supervised-learning model to identify the clients who might have experienced bad internet service. We used a stochastic gradient boosting model, where the target variable indicated whether the client had filled a service request due to no access to the internet. As model features, we used the network signals of the hours before the service request. Furthermore, we included in the model two special features, which we developed specifically for this task: the cluster to which each client belonged and the probability that the client was using the internet at the hour the service might have failed.

The model we developed had very good performance, with an average accuracy over the cross-validation folds of 0.978 (SD = 0.001) and with an AUROC of 0.985 (SD = 0.002). Furthermore, specificity and recall were also very good (0.979 and 0.868, respectively). This suggests that the model was efficient in identifying the clients that had filled a no access service request, without classifying too many clients as false positives. Despite these very positive results, the work had also some limitations.

A deeper analysis of the predictors revealed that the lack of network measurements was a very important category within the model. If the cable modem had no measurements, it was more probable that it would be classified as a client with no access by the model. This proved to be one of the biggest challenges of the model, because it would not distinguish the clients that had voluntarily switched off their cable modem from the ones that were indeed having technical issues. In addition, if the measurements were not registered in the system for some reason in a given hour, the model would flag much more cases as positives, increasing the rate of false positives.

Another limitation of this work was the way the population without no access to the internet was selected. Since a lot of people who experience problems in the service do not complain (Garín-Muñoz, et al., 2016; Nimako, & Mensah, 2012), and clients may not notice deteriorations of service, we could not know for sure who were the clients that were true zeros, that is, the clients with no problems in their service. Therefore, some false positives of the model could actually be clients who had service deterioration but did not notice or did not complain.

To compose the population of clients who did not fill a service request, we selected two random hours (one peak and one non-peak hour). Half of the population had network measures at the first hour and the other half had network measures at the second hour. However, the population with a no-access service request was drawn at many different hours and days throughout a month. This might have introduced some biases in the model, because specific hours might have been influenced by specific factors related to the monitoring systems performance. Ideally, the population without service requests would have been drawn at the same hours of the population with a service request. However, we did not have the chance to implement this solution due to time constraints.

In our model, we did not do any variable selection, but we could have included previous procedures to first select the most important variables to include in the final model. For instance, we could first run a two step-algorithm and select the variables that 1) have the highest importance, i.e., reduce the classification error the most; and 2) select a smaller set of variables that would reduce redundancy, keeping prediction good enough, as detailed in Genuer, Poggi and Tuleau-Malot (2015; see also (Genuer, Poggi, Tuleau-Malot, 2010).

For the classification of the cases, we used a default 0.5 threshold. If the model outputted a probability higher or equal to 0.5, the case would be classified as without access to the internet; if the probability was lower than 0.5, the case would be classified as having no internet issues. However, we could have set the cut-off point at another value. For instance, Kuhn (2014) suggests that the best cut-off point could be selected by calculating the distance from specificity and sensitivity of the actual model to sensitivity and specificity of the best possible model (i.e., a sensitivity and specificity equal to 1). The selected cut-off point would be the one that minimizes that distance.

We could have selected a different way to validate our results. In our analysis, we used a ten-fold cross-validation, meaning that we split the dataset into 10 partitions, and used 9 of those partitions to train the model, and one partition to test the model. Then, we would use another nine partitions to train the model and one (different) partition to test the model, until all partitions were used to test the model. The final model performance was an average of the performance on each of the nine partitions. In theory, we also could have trained the model using all the data from one or more months (with or without cross-validation to select the best model parameters) and then test in the subsequent month, using a cross-validation schema suitable for temporal data (Hyndman, 2010). Future versions of the model, where the dataset is built differently, could test this approach, considering that there might be some seasonality in complaining behavior (for instance, in the hours that people complain, or even on the days or weeks).

Finally, we only applied algorithms that were based on decision trees. We could have applied other algorithms, such as neural networks or support vector machines to see if the performance could have been further improved. Nevertheless, the results obtained with the GBM were already very satisfactory. We were able to identify almost all the clients who made a complain, while keeping the rate of false positives low, which was the main goal of the analysis. Furthermore, we did a complete analysis of the results and made an effort to present the model in a comprehensible way within the company. The model was well-received within the company and, at the end of the internship, was ready to be implemented and tested on additional days and validated with the technical team as described in the next chapter (*Chapter 7, future work*).

7. FUTURE WORK

In the future, there were several lines of work we would like to pursue, given more time. Below, we present our suggestions for future work regarding the topics approached in this report.

7.1. SEGMENTATION OF INTERNET CLIENTS

Regarding the segmentation of internet clients, the main improvement that could be done in the future would be to have access to (anonymized) data regarding what the client would typically do online. This data could be aggregated in broad categories such as “browsing”, “streaming” and “gaming,” and anonymized, to preserve the privacy of the clients. In addition, it might be very helpful to have data about the duration of each online session and the rate of upstream and downstream traffic per second or minute (i.e., the volume of bytes uploaded or downloaded per second or minute).

These variables would enrich the segmentation analysis, providing segments that might be further characterized by the typical online activities of the clients and their rate of usage of the internet service. This segmentation analysis could be more useful to the company to further understand their clients and their online necessities.

7.2. PREDICTING INTERNET USAGE

As explained in chapter 4, we developed a feature that aimed to estimate the probability that a client would be actively using the internet at a given hour. Although we tried other approaches (principal components analysis and clustering; since chapter 4 for details), our feature ended up being just the number of hours where downstream traffic was above 1MB, divided by the total number of hours registered. Future work could improve this work in different ways. The first way would be to make distinct features for working and non-working days. The second way would be to include more variables to calculate the probability that a client uses the internet: for instance, the model could include the traffic in the hours before the target hour, in the previous day and in the same weekday in the previous week. That is, future work could produce a probability for each pair of client/hour that takes into consideration more variables regarding the clients’ typical usage.

7.3. IDENTIFYING CLIENTS WITH NO ACCESS TO THE INTERNET

The model to identify clients with no access to the internet had a good performance on the cross-validation, but future work should test its performance in subsequent days. Indeed, the most important test of the model would be its daily or even hourly implementation, to see if it is robust enough to perform well across different days. A further step to validate the model would also be an analysis of the results with the technical team, and through surveying the clients which were identified by the model, to confirm if they had experienced a bad service or not.

As mentioned, the model had some difficulty to distinguish between cable modems that were switched off and cable modems that were experiencing difficulties. One of the things that could be done to improve the model would be to include variables that represented a broader time window, leading to the time where the client experienced difficulties. In our model, we only looked at measurements

taken two hours before. We chose this time window because usually clients complain within a short time when they experience serious problems in the service. However, maybe there is some indication earlier on that the service will be deteriorated. To investigate this, future model could, for example, include aggregated measurements of a day or a week before the complaint.

7.4. OTHER LINES OF WORK

During the internship, we also had access to anonymized survey data about how satisfied the clients were with their internet service (on a scale from 0, completely dissatisfied to 10, completely satisfied). However, there was not enough time to explore the dataset in detail.

The goal was to relate clients' subjective satisfaction with the internet service and network measurements about the quality of the internet service, in addition to variables such as the value paid for the service, the clients' longevity and the clients' typical usage. Our first attempt at this problem as to apply a multi-nominal classification algorithm (a multiclass GBM), where the classes would be the clients that were dissatisfied, not satisfied nor dissatisfied, and satisfied. However, this model performed with an accuracy only slightly above chance. We then tried to identify only the clients that were dissatisfied. The performance improved, but accuracy was still close to 60%. Future work could improve the features in the model, including feature selection, and try other classification algorithms.

This is a promising avenue of work, because Identifying clients who are more or less satisfied is crucial for the company's ability to act toward that client. For instance, if we could identify the clients that would be more dissatisfied with their internet service, we could take proactive measurements to make amendments to those clients, providing a better service and preventing clients' churn. For clients that are only satisfied (rather than satisfied) the company could invest on strategies that would increase satisfaction, which may be different from the strategies to prevent dissatisfaction (Conklin, Powaga, & Lipovetsky, 2004).

Using objective variables to predict satisfaction, instead of subjective variables, would also be very efficient. Typically, satisfaction is predicted using variables collected in a survey, which is costly and reaches only a very small fraction of the clients. To have a way to predict satisfaction using only variables that are readily available within the company would be very important for managing clients' experiences. Furthermore, if the variables that are more important predictors of clients' satisfaction are identified, the company may directly intervene in those aspects of the service that are more impactful.

8. REFERENCES

Becker, D. (2017). Partial dependence plots [tutorial]. Retrieved from https://www.kaggle.com/dansbecker/partial-dependence-plots

Bishop, C. M. (2006). Pattern recognition and machine learning. Berlin, Heidelberg: Springer-Verlag.

Brandtzæg, P. B., Heim, J., & Karahasanović, A. (2011). Understanding the new digital divide—A typology of Internet users in Europe. *International Journal of Human-Computer Studies*, *69*(3), 123–138. https://doi.org/10.1016/j.ijhcs.2010.11.004

Breiman, L. (1996). *Bagging predictors. Machine learning, 24*, 123-140. doi: 10.1023/A:1018054314350

Breiman L., Friedman J. H., Olshen R. A., & Stone, C. J. (1984). Classification and regression trees. Wadsworth.

Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16, 321–357.

Conklin, M., Powaga, K., & Lipovetsky, S. (2004). Customer satisfaction analysis: Identification of key drivers. *European Journal of Operational Research*, *154*(3), 819–827. https://doi.org/10.1016/S0377-2217(02)00877-9

Fiedler, M., Hossfeld, T., & Tran-Gia, P. (2010). A generic quantitative relationship between quality of experience and quality of service. *IEEE Network, 24*(2), 36–41. https://doi.org/10.1109/MNET.2010.5430142

Friedman, J. H. (2001). Greedy function approximation: The gradient boosting machine. *The Annals of Statistics, 29*(5), 1189-1232.

Friedman, J. H. (2002). Stochastic Gradient Boosting. *Computational Statistics and Data Analysis, 38*(4):367-378.

Garín-Muñoz, T., Pérez-Amaral, T., Gijón, C., & López, R. (2016). Consumer complaint behaviour in telecommunications: The case of mobile phone users in Spain. *Telecommunications Policy*, *40*(8), 804–820. https://doi.org/10.1016/j.telpol.2015.05.002

Genuer, R. Poggi, J., Tuleau-Malot, C. (2010). Variable selection using random forests. *Pattern Recognition Letters, Elsevier, 31* (14), 2225-2236.

Genuer, R. Poggi, J., Tuleau-Malot, C. (2015). VSURF: An R package for variable selection using random forests. *The R journal, 7*(2), 19-33.

Greenwell, B. M. (2017). pdp: An R package for constructing partial dependence plots. *The R Journal*, *9*(1), 421–436.

Hand, D., Mannila, H., & Smyth, P. (2001). *Principles of data mining*. Cambridge, MA, USA.

Horton, B. (2016, November 22). *Calculating AUC: the area under a ROC Curve* [Blog post]. Retrieved from http://blog.revolutionanalytics.com/2016/11/calculating-auc.html

Hyndman, R. (2010). Why every statistician should know about cross-validation [Blog post]. Retrieved from https://robjhyndman.com/hyndsight/crossvalidation/

Kihl, M., Lagerstedt, C., Aurelius, A. & Ödling, P. (2010). Traffic analysis and characterization of Internet user behavior. In *International Congress on Ultra Modern Telecommunications and Control Systems* (pp. 224–231). https://doi.org/10.1109/ICUMT.2010.5676633 [Available at https://lup.lub.lu.se/search/publication/1734535]

Kuhn, M. (2007). Variable importance using the CARET package. Retrieved from http://ftp.uni-bayreuth.de/math/statlib/R/CRAN/doc/vignettes/caret/caretVarImp.pdf

Kuhn, M. (2014). *Optimizing probability thresholds for class imbalances.* [Blog post]. Retrieved from http://appliedpredictivemodeling.com/blog/2014/2/1/lw6har9oewknvus176q4o41alqw2ow

Kuhn, M. (2018). Package 'caret' (version 6.0-8.0). Retrieved from https://cran.r-project.org/web/packages/caret/caret.pdf

Kotsiantis, S., Kanellopoulos D., & Pintelas P. (2006). Handling imbalanced datasets: A review. *GESTS International Transactions on Computer Science and Engineering, Vol.30.*

Laurae (2016). Let me learn the learning rate (eta) in xgboost! (or in anything using Gradient Descent optimization) [Blog post]. Retrieved from https://medium.com/data-design/let-me-learn-the-learning-rate-eta-in-xgboost-d9ad6ec78363

Milener, G. & Guyer, C. (2017). *Microsoft Open Database Connectivity (ODBC).* Retrived from https://docs.microsoft.com/pt-pt/sql/odbc/microsoft-open-database-connectivity-odbc?view=sql-server-2017

Nimako, S.G., & Mensah, A.F. (2012). Motivation for customer complaining and non-complaining behavior towards mobile telecommunication services. *Asian Journal of Business Management, 4*(3), 310–320.

Oliveira, M. R., Valadas, R., Pacheco, A., & Salvador, P. (2007). Cluster analysis of internet users based on hourly traffic utilization. *IEICE Transactions on Communications*, *E90B*. https://doi.org/10.1093/ietcom/e90-b.7.1594 [Available at http://www.av.it.pt/rv/Papers/hetnet03_cluster.pdf]

Ortega Egea, J.M., Recio Menéndez, M., & Román González, M.V. (2007). Diffusion and usage patterns of Internet services in the European Union. *Information Research*, ***12***(2) paper 302. [Available at http://InformationR.net/ir/12-2/paper302.html]

Parkes, D. (2018). *The ROC curve.* [Blog post]. Retrieved from https://deparkes.co.uk/2018/02/16/the-roc-curve/

R-Project (2018). *What is R?* Retrieved from https://www.r-project.org/about.html

Ridgeway, G. (2017). *Package 'gbm' (version 2.1.3).* Retrieved from https://cran.r-project.org/web/packages/gbm/gbm.pdf

Rousseeuw, P. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, *20*, 53-65. doi: 10.1016/0377-0427(87)90125-7

Tan, P., Steinbach, M., & Kumar, V. (2006). *Introduction to data mining (1st edition).* Boston, MA, USA: Pearson.

Terribile, M. (2017). *Understanding Cross Validation's purpose.* [Blog post]. Retrieved from https://medium.com/@mtterribile/understanding-cross-validations-purpose-53490faf6a86s

Therneau, T., Atkinson, B., & Ripley, B. (2018). Package 'rpart.' Retrieved from https://cran.r-project.org/web/packages/rpart/rpart.pdf

Tibshirani, R., Walther, G., & Hastie, T. (2001). Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B, 63*, 411-423.

The Apache Software Foundation (2018). *Apache Hive.* Retrieved from https://hive.apache.org/

Appendix

1. **Name and description of the variables included in the PCA (May and September/October data)**

Variable name	Variable description
AVG_UP_WD_DAY	Hourly average of upload traffic, for working days and day hours (7:00 am to 5:59 pm)
AVG_DOWN_WD_DAY	Hourly average of download traffic, for working days and day hours (7:00 am to 5:59 pm)
AVG_UP_WD_EVE	Hourly average of upload traffic, for working days and evening hours (6:00 pm to 11:59 pm)
AVG_DOWN_WD_EVE	Hourly average of download traffic, for working days and evening hours (6:00 pm to 11:59 pm)
AVG_UP_WD_NIGHT	Hourly average of upload traffic, for working days and night hours (00:00 am to 6:59 am)
AVG_DOWN_WD_NIGHT	Hourly average of download traffic, for working days and night hours (00:00 am to 6:59 am)
AVG_UP_NWD_DAY	Hourly average of upload traffic, for non-working days and day hours (7:00 am to 5:59 pm)
AVG_DOWN_NWD_DAY	Hourly average of download traffic, for non-working days and day hours (7:00 am to 5:59 pm)
AVG_UP_NWD_EVE	Hourly average of upload traffic, for non-working days and evening hours (6:00 pm to 11:59 pm)
AVG_DOWN_NWD_EVE	Hourly average of download traffic, for non-working days and evening hours (6:00 pm to 11:59 pm)
AVG_UP_NWD_NIGHT	Hourly average of upload traffic, for non-working days and night hours (00:00 am to 6:59 am)
AVG_DOWN_NWD_NIGHT	Hourly average of download traffic, for non-working days and night hours (00:00 am to 6:59 am)
THRESH_DOWN_90	Number of hours with an average download traffic higher than the hourly average of the bottom 90% of the population
THRESH_UP_HIGHEST	Number of hours with an average upload traffic higher than the hourly average of the bottom 99% of the population
THRESH_DOWN_HIGHEST	Number of hours with an average download traffic higher than the hourly average of the bottom 99% of the population
THRESH_UP_90	Number of hours with an average upload traffic higher than the hourly average of the bottom 90% of the population
THRESH_MIN_DAY	Number of days with active internet usage, over the total number of days with any entry

2. Loadings of each variable in the three retained PCs

May

	PC1	PC2	PC3
AVG_UP_WD_DAY	0.22458820	-0.30654493	-0.017203159
AVG_DOWN_WD_DAY	0.23143063	0.18053399	-0.137766337
AVG_UP_WD_EVE	0.25005325	-0.29270056	0.005529070
AVG_DOWN_WD_EVE	0.26390615	0.23417382	-0.066187538
AVG_UP_WD_NIGHT	0.24751708	-0.28744231	0.013930012
AVG_DOWN_WD_NIGHT	0.24973603	0.21290895	-0.050036622
AVG_UP_NWD_DAY	0.23586838	-0.30581166	-0.005695844
AVG_DOWN_NWD_DAY	0.24248664	0.20951731	-0.103099358
AVG_UP_NWD_EVE	0.24589602	-0.27714525	0.016870558
AVG_DOWN_NWD_EVE	0.25161737	0.22538230	-0.052696955
AVG_UP_NWD_NIGHT	0.24373815	-0.28118414	0.013632225
AVG_DOWN_NWD_NIGHT	0.24145630	0.21560985	-0.057837102
THRESH_UP_90	0.28700764	0.07815231	0.033607589
THRESH_DOWN_90	0.25104814	0.25797100	0.091109656
THRESH_UP_HIGHEST	0.25087997	-0.24254268	0.011641110
THRESH_DOWN_HIGHEST	0.25915677	0.26854066	-0.074604060
THRESH_MIN_DAY	0.09366611	0.10391561	0.970169689

September/October

	PC1	PC2	PC3
AVG_UP_WD_DAY	0.2228026	-0.31134537	-0.019098849
AVG_DOWN_WD_DAY	0.2212387	0.17567401	-0.177884139
AVG_UP_NWD_DAY	0.2337730	-0.31182014	-0.003667179
AVG_DOWN_NWD_DAY	0.2488986	0.21919301	-0.111180885
AVG_UP_WD_EVE	0.2500250	-0.29284984	0.004968045
AVG_DOWN_WD_EVE	0.2667890	0.23481159	-0.077994329
AVG_UP_NWD_EVE	0.2465000	-0.27883727	0.018097164
AVG_DOWN_NWD_EVE	0.2566183	0.22934480	-0.056388359
AVG_UP_WD_NIGHT	0.2447343	-0.28958124	0.012668298
AVG_DOWN_WD_NIGHT	0.2510200	0.21458441	-0.069343465
AVG_UP_NWD_NIGHT	0.2446468	-0.28473926	0.006158909
AVG_DOWN_NWD_NIGHT	0.2473701	0.21929945	-0.081555970
THRESH_UP_90	0.2796563	0.06354704	0.054155232
THRESH_DOWN_90	0.2492867	0.24671642	0.130697097
THRESH_UP_HIGHEST	0.2469133	-0.22953887	0.004837020
THRESH_DOWN_HIGHEST	0.2584697	0.25619902	-0.079259897
THRESH_MIN_DAY	0.1115071	0.11564104	0.952903206