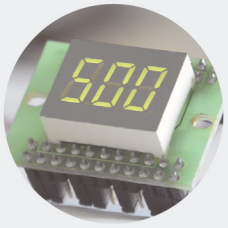
INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO
MESTRADO EM ENGENHARIA ELECTROTÉCNICA E DE COMPUTADORES






Top-Up Forecasting of Pre-Paid Mobile Subscribers

PEDRO MIGUEL FERREIRA ALVES
novembro de 2021

POLITÉCNICO
DO PORTO

Politécnico do Porto
Instituto Superior de Engenharia do Porto

Top-Up Forecasting of Pre-Paid Mobile Subscribers

Pedro Miguel Ferreira Alves

Master in Electrical and Computer Engineering
Specialization Area of Telecommunications

Departamento de Engenharia Eletrotécnica
Instituto Superior de Engenharia do Porto

November, 2021

This dissertation partially satisfies the requirements of the Thesis/Dissertation course of the program Master in Electrical and Computer Engineering, Specialization Area of Telecommunications.

Candidate: Pedro Miguel Ferreira Alves, No. 1161571, 1161571@isep.ipp.pt

Scientific Guidance: Maria Benedita Campos Neves Malheiro, mbm@isep.ipp.pt

Company: Altice Labs

Advisor: Ricardo Ângelo Filipe, ricardo-a-filipe@alticelabs.com

Departamento de Engenharia Eletrotécnica
Instituto Superior de Engenharia do Porto
Rua Dr. António Bernardino de Almeida, 431, 4200–072 Porto

November, 2021

Education is not the filling of a pail, but the lighting of a fire.
- William Butler Yeats

Acknowledgements

Since no man is ever self-made there are a few people I would like to thank for helping reach this far.

A special thank you to my family for continuous support throughout my academic years.

Since friends help shape our character, I would like to thank all my friends for all the lessons and making my journey enjoyable.

Last but not least, I would like to extend my gratitude to the engineers who worked alongside me on this dissertation for all the patience, lessons and making me work harder than ever before.

Pedro Alves

Abstract

In an ever-evolving technology world, telecommunications operators must attend to client needs in an effective and speedy manner to strengthen their relationship. The difficulty of this challenge is heightened in Big Data environments where there is a necessity to make sense of the valuable information within data. In the pre-paid telco environment, also known as pay-as-you-go, it is imperative for operators to predict client behaviour efficiently to meet their needs and improve campaigns and notifications, thus improving communication, client retention and revenue.

In this dissertation, a novel top-up date and value prediction solution for the pre-paid telco environment, is presented. This solution aims to dynamically estimate, for each client, the top-up date and value for the upcoming month. For this, the initial data goes through the developed processing pipeline. The first step is pre-processing, where data is cleaned and transformed. After this, it undergoes a feature engineering and selection step to identify the most relevant features for the prediction of the monthly frequency and value. For the prediction of the targets, several regression techniques were studied both on the offline and online scenario with the help of sliding windows. Using the most efficient technique, the monthly target predictions undergo a processing stage in which they are transformed into the individual top-up date range and top-up monetary value range for the following month. The evaluation of these predicted ranges is based on verifying if the observed event falls within the predicted interval.

The solution is implemented in Python and the Jupyter Notebooks environment for data analysis, dimensionality reduction and offline learning experiments. The online learning experiments make use of the Massive Online Analysis (MOA) graphical user interface (GUI) framework. In the end, the designed solution is able to estimate individual top-up activity with an accuracy of approximately 80 % for the date and 70 % for the monetary value.

Keywords: Telecommunications, Top-up, Forecasting, Big Data, Machine Learning

Resumo

Num mundo de tecnologia em constante evolução, as operadoras de telecomunicações devem atender às necessidades dos clientes de forma eficaz e rápida para satisfazê-los. A dificuldade deste desafio é aumentada em ambientes de *Big Data*, pois surge a necessidade de entender quais são as informações valiosas nos dados. No ambiente de telecomunicações pré-pago, também conhecido como *pay as you go*, é imperativo que as operadoras prevejam o comportamento de seus clientes de forma eficiente para que suas necessidades sejam atendidas eficientemente e campanhas e notificações aprimoradas possam ser-lhes enviadas com base na sua atividade, melhorando assim a comunicação, retenção de clientes e receita.

Nesta dissertação, uma nova solução de previsão de valor e data de recarga, para o ambiente de telecomunicações pré-pago, é apresentada. Esta solução tem como objetivo prever de forma dinâmica a data de recarga e o valor de cada cliente para o mês seguinte. Para isso, os dados iniciais, após serem estudados, são colocados numa *pipeline* desenvolvida onde a primeira etapa é o pré-processamento onde os dados são limpos e transformados. De seguida, passam por uma etapa de engenharia e seleção de variáveis para obter apenas os variáveis mais relevantes para a previsão dos *targets*, frequência mensal e valor, respectivamente. Para a previsão dos *targets*, diversas técnicas de regressão são estudadas tanto no cenário *offline* como no *online* com o auxílio de uma janela deslizante. Depois de escolhida a técnica mais eficiente, as previsões mensais previstas passam por uma etapa de processamento na qual são transformadas de modo a obter-se um intervalo de dias e de valor monetário definido para cada cliente para o mês seguinte. A avaliação dessas estimativas é definida com base em averiguar se o evento observado se encontra dentro do intervalo previsto.

A solução é implementada utilizando *Python* no ambiente *Jupyter Notebooks* para análise de dados, redução de dimensionalidade e experiências de aprendizagem *offline*. As experiências de aprendizagem *online* fazem uso da interface gráfica MOA. No final, a solução desenvolvida é capaz de prever a atividade de recarga dos clientes com uma precisão de aproximadamente 80 % para a data e 70 % para o valor.

Palavras-Chave: Telecomunicações, Recarga, Previsão, Big Data, Machine Learning

Contents

List of Figures

List of Tables

List of Acronyms

ACM	*Active Campaign Manager*
AI	*Artificial Intelligence*
ALB	*Altice Labs*
AMR	*Adaptive Model Rules*
ATM	*Automated Teller Machine*
AUC	*Area Under Curve*
BI	*Business Intelligence*
BMTR	*Basic Multi-Target Regressor*
CRISP-DM	*Cross Industry Standard Process for Data Mining*
CLV	*Customer Lifetime Value*
CRM	*Customer Relationship Management*
DBMS	*Database Management System*
DM	*Data Mining*
DSMS	*Data Stream Management System*
DT	*Decision Tree*
EDA	*Exploratory Data Analysis*
ESW	*Event Sliding Window*
GPON	*Gigabit Passive Optical Network*
GUI	*Graphical User Interface*
KM	*Knowledge Management*
KPI	*Key Performance Indicator*
LLM	*Logit Leaf Model*
MA	*Moving Average*
MAE	*Mean Absolute Error*
ML	*Machine Learning*
MLP	*Multi-Layer Perceptron*
MOA	*Massive Online Analysis*
MOLS	*Multiple Ordinary Least Squares*
MSW	*Monthly Sliding Window*
MTR	*Multi-Target Regression*
MTPR	*Multi-Target Perceptron Regressor*
OHE	*One Hot Encoder*
PH	*Page-Hinckle*

PT	*Portugal Telecom*
RFE	*Recursive Feature Elimination*
RFECV	*Recursive Feature Elimination Cross-Validation*
RF	*Random Forest*
RFM	*Recency Frenquency Monetary*
RL	*Reinforcement Learning*
RMSE	*Root Mean Squared Error*
ROC	*Receiver Operating Characteristic*
RSS	*Residual Sum of Squares*
STR	*Single-Target Regression*
SIM	*Subscriber Identification Module*
TDL	*Top Decile Lift*
TSA	*Time Series Analysis*

Chapter 1

Introduction

This dissertation was developed within the scope of the Thesis/Dissertation curricular unit (TEDI), integrated in the 2nd cycle of studies on the Master in Electrical and Computer Engineering. The study/application project presented here was developed in a practical business context at Altice Labs, an Altice Group company, dedicated to the production of advanced solutions with an innovative approach supported by an ecosystem built around research and development entities, startups and industrial partners.

In this chapter, the introduction of the work will be presented, starting with a contextual framework, moving on to an introduction to the problem under study, followed by the definition of the objectives/motivations of the project, ending with the presentation of the dissertation structure.

1.1 Hosting Entity

In 2001, the Altice Europe group was founded by Patrick Drahi, who still leads it today, and who is dedicated to the development of solutions and services in the areas of telecommunications, media and content as well as data analysis and advertising. Responsible for a wide range of developments and innovations, it currently has companies in four territories and thirty million customers [17].

Altice Labs (ALB) was founded in 2016 when the Altice Europe group expanded its territory of operations and acquired Portugal Telecom (PT), renaming the then

PT Inovação to Altice Labs, occupying it's space in Aveiro, visible in Figure 1.1, and also exploring new areas of operation.

Figure 1.1: Altice Labs campus in Aveiro [1]

In 2017, it created the first gateway for optical communications, the Gigabit Passive Optical Network (GPON) Gateway 802.11ac 4x4, which won the Technology Leadership Award 2017 [3].

Since then, it has continued to shape the future of technology, allowing communication service providers and companies to offer advanced and differentiated services to their customers and users. Currently, it is led by Alcino Lavrador and the management bodies presented in Figure 1.2, and has around five hundred employees.

Altice Labs is part of an innovation ecosystem, continuously engaging in collaborative projects aimed at strategic leadership. The areas of research and development are varied, as can be seen in Figure 1.3. This internship took place with the team from SRP5, the area of Big Data & Monetization.

The Big Data Analytics area focuses on creating solutions and data analysis optimisation services not only for subsidiary companies of the Altice Europa group but also for external companies in various sectors.

1.2 Context and Problem

Knowledge Management (KM), is a broad term that refers to the ability to identify, store and retrieve knowledge. Therefore, KM involves understanding what knowledge is important to the organisation, understanding systems that are important for organisational decision making, database management and analytical Data Mining (DM) tools.

Currently, the era of big data is being lived. Davenport [18] defines big data as information that is too large to be stored on a single server, unstructured to the

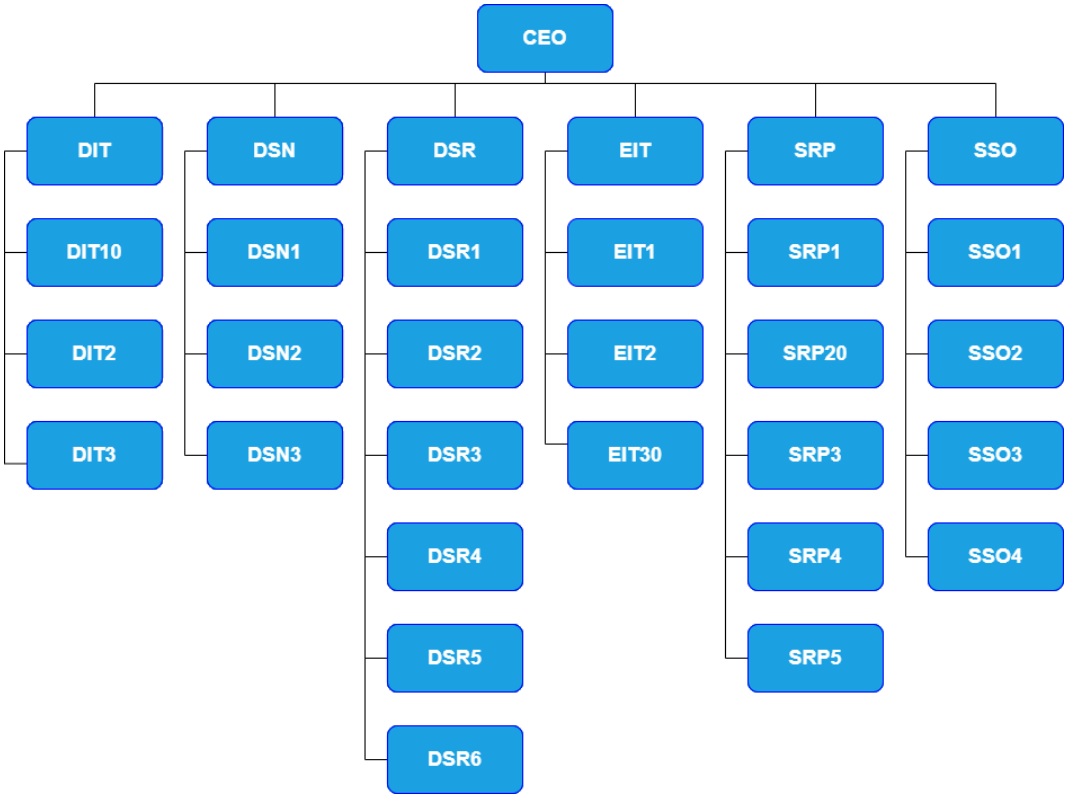


Figure 1.2: Altice Labs organizational chart [2]

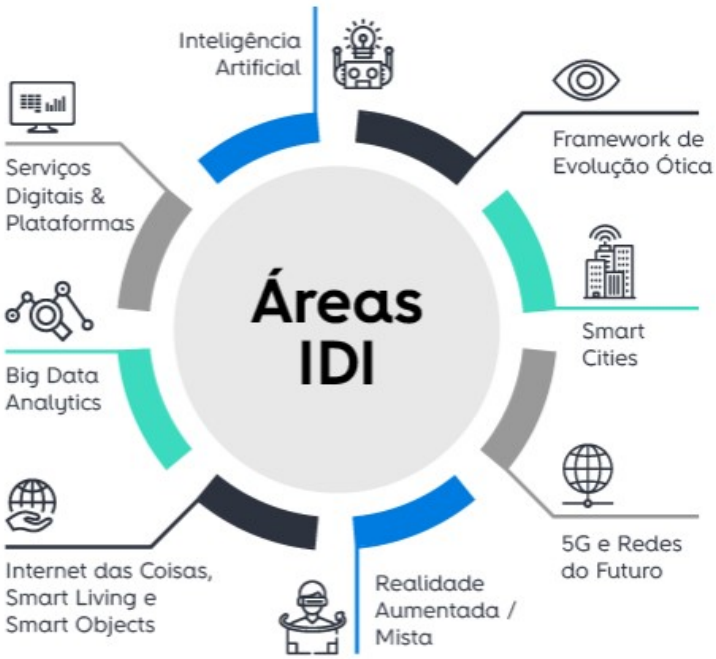


Figure 1.3: Altice Labs research and development areas [3]

point that it cannot fit into a database of rows and columns, that flows continuously in ways that cannot be stored in a data warehouse and that has characteristics of lack of structure.

Knowledge Management deals with big data, identifying and managing knowledge assets within organisations. KM is process oriented, reflects on how knowledge can be acquired, as well as the tools to assist in decision making. The purpose of big data is to analyse, converting data into insights, innovation and business value. It can add value by providing performance measures in real time, providing more timely analysis based on more complete data and leading to more solid decisions.

The Business Intelligence (BI) area is limited to seeking and gaining an understanding of the business environment in order to make the right decisions. For this, it involves the process of systematic acquisition, classification, analysis, interpretation and exploration of information [19]. The quantitative side of this development is business analysis, with a focus on providing better responses to business decisions based on access to large amounts of information, ideally in real time.

With this in mind, Altice Labs developed a platform, in the Customer Relationship Management (CRM) area, for designing and launching campaigns, the Active Campaign Manager (ACM). Aimed at telecommunications operators, this platform allows them to independently configure and launch promotional actions with the following objectives: increasing customer satisfaction, increasing revenues, promoting the acquisition of products/services, reducing costs of operation, among others.

One of the challenges of a company that uses advertising campaigns in order to promote its products or services is to ensure a strong adhesion of customers to the campaigns to which they are encouraged. However, in the prepaid telecommunications system, the offer of campaigns must be very refined so that the customer does not fail to top-up.

If the platform is aware of the times of top-up and the amounts that customers will top-up, it can recommend relevant campaigns for each customer, taking into account their previous behaviour. This leads to greater customer satisfaction, seeing that the clients are not bombarded with uninteresting campaigns. Additionally, it becomes less computationally expensive due to the fact that less hardware and network resources are required, providing greater system efficiency.

To this end, the goal of the current work is forecast client top-up date and amount, so that campaigns can be further tailored.

1.3 Motivation

Knowledge Management is part of the general field of knowledge, epistemology, and refers to the means to register and retrieve it, namely using computer systems, and

quantitative analysis to understand it in business contexts, such as business analysis example.

There are many applications of quantitative analysis, accommodating the general structure of the term business analytics, such as: analytics, descriptive analysis, predictive analysis, diagnostic analysis and prescriptive analysis. Data Mining includes descriptive and predictive modelling. In this respect, this work focuses on the predictive component.

Predictive analysis expands onto statistical and/or artificial intelligence to provide predictive capabilities. It also includes classification modelling, application of models for process optimisation, to include the identification of the most likely customer profiles to send marketing content or to flag suspicious insurance claims or many other applications.

Currently, predictive analysis involving forecasting models is present in practically all areas of human activity, from the governmental area, to the area of industry and engineering to that of telecommunications. In the area of telecommunications, the application of forecasting systems can applied to several areas, but the area of forecasting customer behaviour is essential to adapt the campaigns offered.

The present work attempts to enrich the CRM platform with top-up prediction as a way to improve the quality of service provided to clients by tailoring individual campaigns to promote and motivate engagement at the right time, thus reducing churn[1] and discontentment as a result of spam communication.

1.4 Objectives

The main goal of this dissertation is to explore the domain of prediction for top-up forecasting. The preparation for this dissertation involved the study of the currently implemented forecasting algorithms in the telecommunications industry and similar implementations from other industries. Furthermore, a good understanding of the basic concepts of Data Mining techniques and Machine Learning (ML) predictive algorithms was obtained alongside an extensive experimental work. Therefore, the present dissertation starts with the literature review, covering different techniques and Machine Learning algorithms. The next step is to develop, test and optimise forecasting models, inspired by published works as well as designing innovative approaches that can contribute to the state of the art solutions.

[1] Clients who decide to stop using a service offered by the company and use another company's service.

1.5 Contributions

This dissertation aims to contribute to forecasting client top-up on the pre-paid Telecom environment in four ways:

- Literature review of forecasting client activity on the Telecom world;
- Study and comparison of various feature engineering and feature selection techniques;
- Implementation and comparison of various regression techniques both for offline and online learning scenarios;
- Development and implementation of several methods for date and value interval prediction.

Additionally, from the study carried out in this dissertation resulted the article *Towards Top-Up Prediction on Telco Operators* [20], accepted on the *20th EPIA Conference on Artificial Intelligence*, a European conference on artificial intelligence where cross-industry and research articles are presented. The article focused on the feature engineering and selection, optimal window size as well as sliding window regression, described in Chapter 3.

1.6 Document Structure

The rest of the present document is structured in Chapters, each one with a specific purpose and divided in different Subsections to allow an organised reading.

Chapter 2 gives a summary of Data Mining and traditional programming and their differences, a comprehensive study of the DM pipeline alongside the current problem as well as the state of the art of top-up in Telecom environment.

Chapter 3 analyses client characteristics and behaviour throughout time by analysing a real data set. It also presents the proposed solution for top-up prediction.

Chapter 4 presents and discusses the results obtained. Several experiments are presented, including experiments that aim to evaluate the proposed solution and to validate the algorithms behaviour.

Chapter 5 finalises this dissertation by drawing the final conclusions. It also makes a brief summary of the workflow of this dissertation and presents possible future work.

Chapter 2

Literature Review

The practical application of knowledge is only possible after assimilation through one or more sources of knowledge. Thus, this chapter addresses Data Mining in general, going through the Data Mining pipeline, time series, regression models, top-up and, finally, the prediction interval.

2.1 Context

Progress in digital data acquisition and storage technology has resulted in the emergence of huge databases. This phenomenon appears to have expanded over various areas of human endeavour from day-to-day activities such as telephone call details, credit card usage records and supermarket transaction data, to more exotic ones just like cloud computing traffic, images of astronomical bodies, medical records and molecular databases. It comes as no surprise then, the data owners are excited of extracting information that might be of value from databases. Figure 2.1 shows the growth of the market size from 2011 until 2018 and a forecast until the year 2027, displaying of constant growth.

To this end, there has been a continuous development of more sophisticated and autonomous computational tools that reduce the need for human intervention and the dependence on data analysis experts. The trick is to extract valuable information from the uninteresting numbers so that data owners can capitalise on it. For this, the developed techniques must be able to create autonomously and based on past

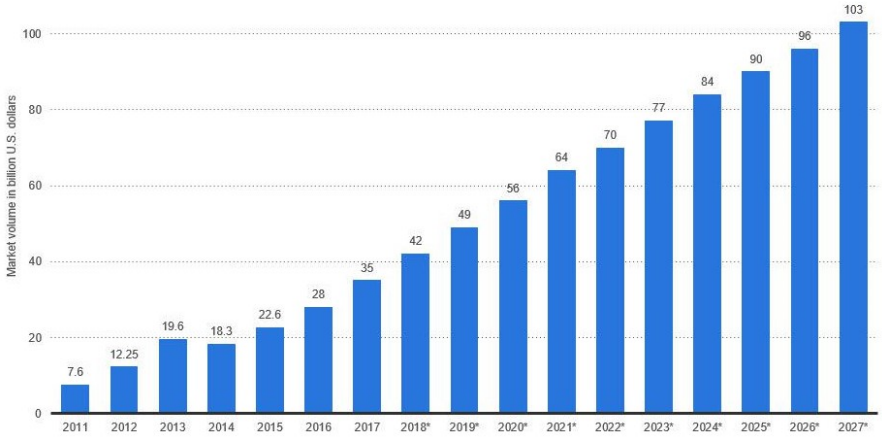


Figure 2.1: Big data market size revenue forecast worldwide [4]

experience, a hypothesis or function capable of solving the problem. This process of inducing a hypothesis from previous experiences is called Data Mining.

2.2 Data Mining and Traditional Programming

Taking the example of traditional programming, in order to solve a problem, the engineer starts by devising the algorithm and writing the corresponding code. Then, the input parameters are added and the algorithm produces the desired output.

In DM, computers are programmed to learn from past experience and apply a principle of inference called induction to draw broad conclusions through a particular set of examples. Algorithms learn to induce a function, or hypothesis, capable of solving the problem when at first all that exists was data representing instances of the problem.

This process has been applied in different areas, such as voice recognition [21], autonomous driving [22], and tools capable of defeating champions in board games [23]. In the telecommunications area, this process has been used for several purposes, including fraud detection [24], network fault isolation [24], improving market efficiency [25].

Data Mining can be defined as the analysis of observational data sets, usually large, to find unsuspecting relationships and summarise the data in new ways that are understandable and useful to the data owner. For this, it utilises Machine Learning algorithms.

To solve such problems, algorithms have different characteristics, thus being included in different categories of learning models. Learning tasks can be divided into predictive, descriptive and reinforcement.

Predictive tasks aim to find a model, from training data, capable of being used to predict a new label, or value, that characterises a new example, based on the values of its input attributes. For this purpose, it is necessary that each object in the training data set has input and output attributes. Consequently, predictive algorithms follow the supervised learning paradigm. This type of learning implies the presence of an external supervisor, who knows the output associated with each example, and can assess the capacity of the induced hypothesis to predict the output value for new examples.

When it comes to descriptive tasks, the objective is to explore, or describe, a set of data and, in these algorithms, the output attribute is ignored. For this reason, these algorithms follow the unsupervised learning paradigm.

Figure 2.2 presents a learning hierarchy according to the types of learning tasks. It is possible to see that inductive learning encompasses both supervised and unsupervised learning. The supervised tasks are distinguished by the type of labels in the data: discrete, in the case of classification; and continuous, in the case of regression. Descriptive tasks are usually divided into: clustering, in which data is grouped, in clusters, according to their similarity; summarisation, whose objective is to find a simple and compact description of a data set; and association, which consists of finding frequent patterns of associations between attributes in a data set.

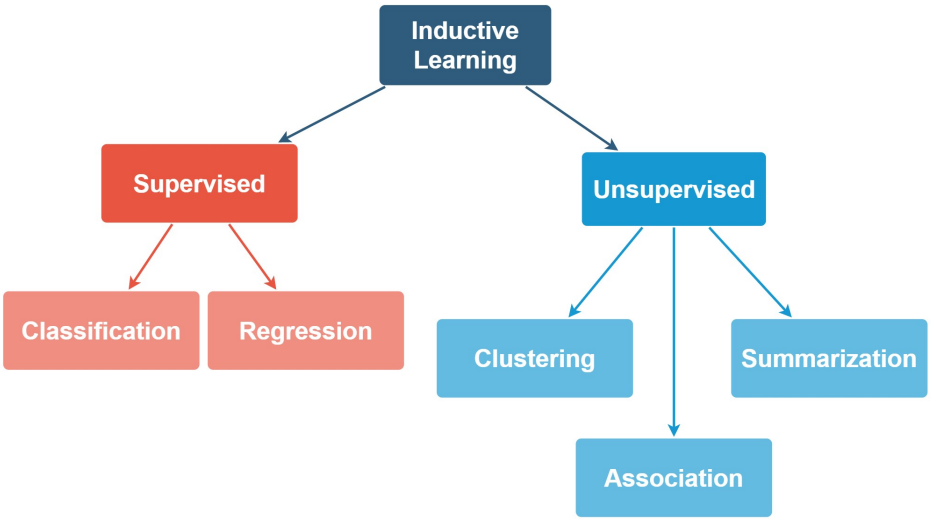


Figure 2.2: Inductive learning hierarchy (adapted from [5])

Another learning task, that does not fit in with the above tasks, is Reinforcement Learning (RL). This task is intended to reinforce, or reward, an action considered positive, and punish an action considered negative. The learning algorithms used in this task punishes the passage through unpromising paths and rewards the passage through promising paths. RL algorithms must, like unsupervised learning tasks, learn the expected output on their own. Nevertheless, on top of that, a reward

function is applied [26]. Figure 2.3, displays the various Machine Learning tasks and their most suitable applications.

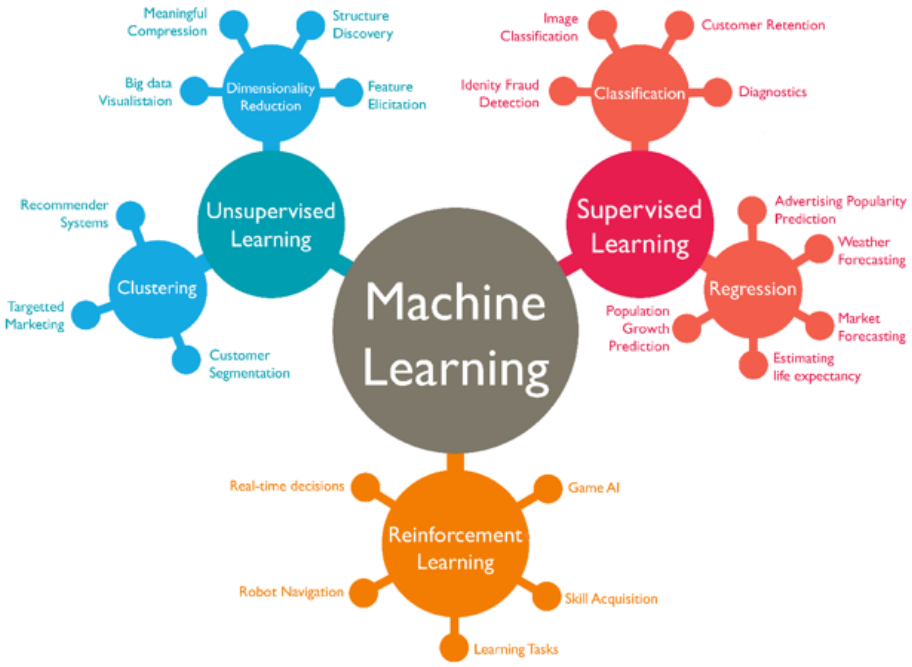


Figure 2.3: Types of learning and their applications [6]

2.3 Data Mining Pipeline

The processes that are used to organise and understand the proposed type of problems, measure progress and achieve the best results must be properly documented in order to provide the correct understanding. In order to establish a standard process, Cross Industry Standard Process for Data Mining (CRISP-DM) developed a methodology described in the Figure 2.4 identifying the involved phases and their interactions [7].

The process diagram leaves no doubt that iterative processing is the rule rather than the exception.

2.3.1 Business Understanding

Initially, it is vital to understand the problem. Therefore, the analyst's first objective is to deeply understand, from a business perspective, what the client really wants. The analyst's objective is to discover important factors at the beginning of the project that can influence the final result. Included in this phase are the identification of the resources available and associated constraints, overall goals, and

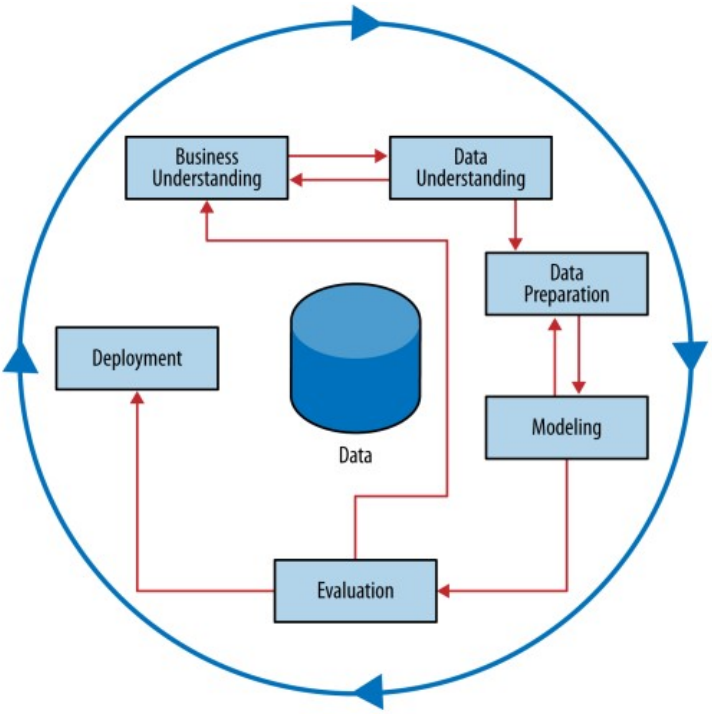


Figure 2.4: CRISP-DM standard process [7]

specific metrics that can be used to evaluate the success or failure of the project [27]. A likely consequence of neglecting this step would be to expend a lot of effort producing the correct answers to the wrong questions [28].

At this stage, the key to success is the analyst's creative formulation of a problem as to how to convert the business problem into one or more data science problems. High-level knowledge of the fundamentals helps creative business analysts see new formulations.

2.3.2 Data Understanding

During this phase, the data is collected and the analyst begins to explore and gain familiarity with the data, including form, content, and structure to understand its strengths and weaknesses. In the data analysis phase, it is necessary to deepen the study to discover the structure of the business problem and the data that are available [7].

Finally, it is through this preliminary exploration that the analyst acquires an understanding of and familiarity with the data that will be used in subsequent steps to guide the analytical process, including any modelling, evaluate the results, and prepare the output and reports [27].

2.3.3 Data Preparation

Although the analytical technologies available are quite powerful, they impose certain requirements on the data. That is, they often require that the data be in a different format than the one that is usually provided, which implies data processing.

After the data have been examined and characterised in a preliminary fashion during the data understanding stage, they are then prepared for subsequent mining and analysis. The treatment of data is done, as a rule, through transformations in the way the data is represented since some Data Mining techniques operate with symbols or categories, while others with numerical values. When operating with numerical values, it is necessary to bear in mind that the values must be normalised to allow comparison between them [29]. It is also during this stage that any necessary merging or aggregating of data sets or elements is done. This data preparation includes any cleaning and recoding as well as the selection of any necessary training and test samples.

The goal of this step is the creation of the data set that will be used in the subsequent modelling phase of the process.

2.3.4 Modelling

The modelling phase is where the Machine Learning techniques are applied to the data. It is important to have some understanding of the fundamental ideas of ML, including the types of techniques and algorithms that exist, so that there is a quick and effective implementation of the problem resolution [28]. Selection of the specific algorithms employed in the Data Mining process is based on the nature of the question and outputs desired. Additional considerations in model selection and creation include the ability to balance accuracy and comprehensibility [27].

2.3.5 Evaluation

The objective of the evaluation phase is to evaluate the results rigorously and to gain the confidence that they are valid and reliable before proceeding. This model evaluation should be performed in a controlled laboratory environment due to the fact that it is much easier, cheaper, faster and safer to test [28].

However, success in the laboratory environment does not guarantee direct passage to production, as the model may contain irregularities that go against the objectives of the project, such as fraud detection, detection of spam or monitoring intrusion [7].

For these reasons, the assessment phase also serves to help ensure that the model is attentive to the original business objectives, making the model's transparency

crucial so that analysts and stakeholders are able to make a correct assessment in anticipation of future catastrophes.

2.3.6 Deployment

On this phase, the results of the model are put to real use to obtain return on investment.

Model creation is generally not the end of the project. Even if the objective of the model is to increase the knowledge of the data, the knowledge acquired will need to be organised and presented in such a way that the client can use it [30].

The installation of a model in a production system usually requires that the model undergoes changes to the production environment, usually for greater speed or compatibility with an existing system. This can lead to substantial expenses and investments. Therefore, it is important that the project is followed from an initial stage by members of the production team [7].

Upon deployment, a plan for monitoring and maintenance is due. This phase refers to the monitoring of algorithm behaviour for example with accuracy and specific Key Performance Indicator's (KPI) monitoring and attention to concept drift [1]. This implies an awareness which contributes to an overtime adaptation to business objectives which can vary for various reasons such as market demand and topicality. Consequently, an update to the Machine Learning model may be needed or even a new Data Mining project.

Knowing that this phase is the consolidation of the previous ones, it is common for the previous processes to be repeated. Repeating the phases can provide new perspectives for better solutions.

With the pipeline established, the next section focuses on data understanding. In this case the business understanding section was reserved toward the end of the literature review so a perception of the real life implementations is easily understood since the analysis and modelling techniques are previously explained.

2.4 Exploratory Data Analysis

The analysis of the characteristics present in a data set allows the discovery of patterns and trends that can provide valuable information that help to understand the data generation process [5]. Many of these characteristics can be obtained by applying simple static formulas.

A plethora of statistical hypothesis testing procedures is available in the statistical analysis literature. However, analysts do not always have *a priori* notions of the expected relationships between variables. Especially when faced with large,

[1] Phenomenon in which the statistical properties of the target variable, which the model is trying to predict, change over time in unforeseen ways.

unknown databases, analysts often prefer to use Exploratory Data Analysis (EDA), or graphical data analysis. This type of analysis allows the analyst [31]:

- Deepen the knowledge of the data set;
- Examine the internal relationships between attributes;
- Identify interesting subsets of the observations;
- Develop an initial idea of possible associations between predictive factors, as well as between predictive factors and the objective variable.

2.4.1 Type and Scale Attribute

The domain of an attribute, that is, the possible values that an attribute can assume, determines the type of analysis that is possible to do. Two aspects of the attributes are the type and the scale.

The type defines whether the attribute represents quantities, being then called quantitative or numerical, or qualities, being then called qualitative, symbolic or categorical, as their values can be associated with categories [5].

The values of a quantitative attribute are ordered and can be used in arithmetic operations. The quantitative attributes fall within one of the following definitions [32]:

- Numeric, as in the countable set of {6, 45, 238};
- Continuous, can assume an infinite number of values. They are usually the result of measurements such as weight, sizes or distances;
- Discrete, contain a finite or infinite countable number of values. Some examples of these cases are the binary or boolean attributes.

Qualitative attributes are, typically, represented by a finite number of symbols or names and in some cases may be represented by numbers. However, these numbers cannot be used in arithmetic operations because they do not represent quantities [32].

The scale defines the operations that can be performed on the attribute values. Regarding the scale, the attributes can be of the qualitative type, classified as nominal and ordinal, and of the quantitative type, classified as interval and rational. These four scales are defined below in detail [32].

- Nominal scale, the values consist of only different names, and there is no order relationship between their values. Consequently, the operations most used in manipulating their values are those of equality and inequality between values.

- Ordinal scale, the values in this scale reflect an order of categories. Therefore, in addition to the previous operators, it is also possible to use $<$,$>$, $\leq$, $\geq$.
- Interval scale, attributes are represented by numbers that are measured at equal intervals from a point of origin. However, the origin does not imply a true absence of the measured characteristic. Thus, it is possible to define both the order and the difference in magnitude between two values.
- Rational scale, the values are similar to the interval scale, however, there is an absolute zero along with a unit that gives meaning to the quotient, that is, it implies a true absence of the measured characteristic.

2.4.2 Univariate Data

Univariate data is data that has only one input attribute. The analyses that can be performed on these types of data are through centrality measures, dispersion measures and distribution measures.

Centrality measures define reference points in the data and vary for numerical and symbolic data. For symbolic data, fashion is generally used. In the case of numerical data, the most used measures are the mean, the median and the percentile.

Dispersion measures measure the variability of a set of values. They allow to verify if the values are widely dispersed or relatively concentrated around a value. The most widely used dispersion measures are: the interval, the variance and the standard deviation.

Data distribution considers: the moment, the skewness and the kurtosis. The moment is a characteristic of the statistic that allows to characterise probability distributions. The first moment, whatever the data set, is always equal to zero and the second moment is equal to the sample variance. Skewness measures the symmetry of the data distribution around the mean and can take several values as Table 2.1 depicts. Finally, kurtosis is a measure that captures the flattening of the distribution function and can also take several values visible in Table 2.2.

Table 2.1: Types of skewness

Value	Description
Positive	Observed when the distribution has a thicker right tail
Negative	Observed when the distribution has a thicker left tail
Zero	Observed when the distribution is symmetric about its mean

Table 2.2: Types of kurtosis

Value	Description
Positive	Distribution has a sharp peak and is called a leptokurtic distribution
Negative	Distribution has a flat peak and is called a platykurtic distribution
Zero	Distribution follows a normal distribution and is also called a mesokurtic distribution

The most common ways of visualising the distribution of univariate data are histograms, boxplots and pie graphs.

2.4.3 Multivariate Data

Multivariate data has more than one independent input attribute. In these cases, the measures of centrality can be obtained by calculating the measure of centrality of each attribute separately. Dispersion measures can be calculated for each attribute independently of the others using any dispersion measure. Multivariate data allow analysis of the relationship between two or more attributes. The correlation, presented in the Equation 2.1, is the most used indicator of the linear relationship between two attributes, where the x^i and x^j are the attributes of study and s_i and s_j is the standard deviation of each of the attributes. The covariance is given by the Equation 2.2, where $\bar{x}^i$ is the mean value of the i^{th} attribute and x_k^i the value of the i^{th} attribute for the k^{th} object.

$$Correlation(x^i, x^j) = \frac{Covariance(x^i, x^j)}{s_i s_j} \tag{2.1}$$

$$Covariance(x^i, x^j) = \frac{1}{n-1} \sum_{k=1}^{n} (x_k^i - \bar{x}^i)(x_k^i - \bar{x}^j) \tag{2.2}$$

The analysis of multivariate data can be facilitated by the use of visualisation resources, particularly for the relationship between the different attributes.

For example, in a scatter plot, the linear correlation between two attributes is presented. In this case, each object, is associated with a position or point on a two-dimensional plane. The values of the attributes, which can be represented by integers or real numbers, define the coordinates of that point.

To conclude, the analysis of a data set can be performed using statistical and visualisation techniques, provides a better understanding of the distribution of the data and helps to choose ways to model the problem.

2.5 Data Pre-Processing

Pre-processing of data is a mandatory the step that encompasses data preparation as well as data reduction. These two methodologies will be presented below as well as the involved tasks.

2.5.1 Data Preparation

Data preparation converts previously incompatible data into data that can be used in a Data Mining process. If the data is not prepared, the algorithms may not read and process them. At best, the algorithms will work, but the results will not make sense or be considered accurate knowledge.

Thus, the data pre-processing step faces the challenge of correcting errors in the data as well as trying to shape them in order to be integrated into the DM algorithms. For this, the data goes through the following phases: cleaning the data; data transformation; data integration; normalisation of data; lack of data imputation; and noise identification. Figure 2.5 presents an illustration of the phases of this step [33].

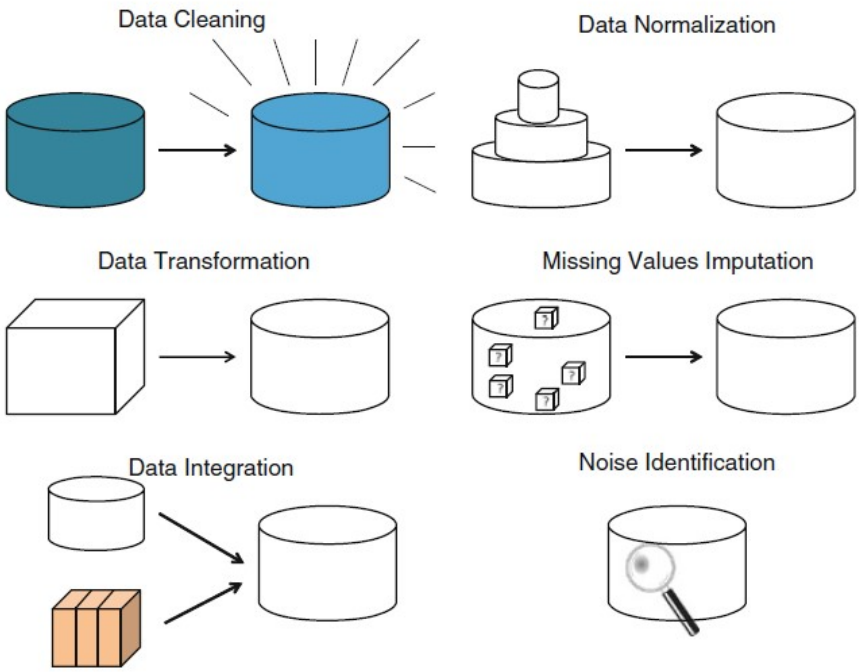


Figure 2.5: Types of data preparation [8]

- **Data cleaning** applies operations that correct invalid data, filter out some defective data from the data set and reduce the level of unnecessary detail in the data. Other data cleaning tasks such as detecting discrepancies and fragments of the original data that do not make sense are subject to audit.
- **Data transformation** converts or consolidates so that the result of the Data Mining algorithm can be applied or optimised. Sub-tasks within data transformation are smoothing, resource building, aggregating or summarising data,

normalisation, discretization and generalisation. Those tasks that require human supervision and are more dependent on the data are the classic techniques of data transformation, such as the generation of reports, new attributes that add the existing ones and generalisation of concepts especially in categorical attributes, such as the replacement of complete dates in the base of data by year numbers only.

- **Data integration** comprises the merging of data from various storage sources. This process requires a lot of care to avoid the introduction of redundancies and inconsistencies in the resulting data set. The most common operations performed in data integration are the identification and unification of variables and domains, the analysis of attribute correlation, the duplication of tuples and the detection of conflicts in data values from different sources.

- **Data normalisation** is responsible for distributing the weight of the attributes so that they all have the same weight. Normalisation is particularly useful for statistical methods. The importance of this step can be seen in the fact that the attributes must be expressed in the same units of measurement as well as using the same scale or interval.

- **Lack of data imputation** is a form of data cleansing, where the objective is to fill in missing values with some intuitive data. In most cases, adding reasonable estimate is preferable to leaving them blank.

- **Noise identification** detects random errors or variations in a measured variable. Once noise is identified, it is possible to apply a corrective process that may involve some type of underlying operation.

2.5.2 Data Reduction

The data reduction step, illustrated in Figure 2.6, comprises the various techniques used to obtain a reduced representation of the original data. This step should not be optional as there are several Data Mining algorithms that have a size limit, thus making the data reduction task as crucial as the data preparation. Taking into account other factors such as reducing complexity and improving the quality of the models produced, the role of data reduction also becomes decisive. Therefore, this stage includes the following tasks: selection of characteristics, selection of instances; discretization; and extracting characteristics and/or generating instances [34].

- **Feature selection** achieves data reduction by removing irrelevant or redundant features. The goal is to find a minimum set of data that produces the same results as the whole data to facilitate the understanding of the extracted

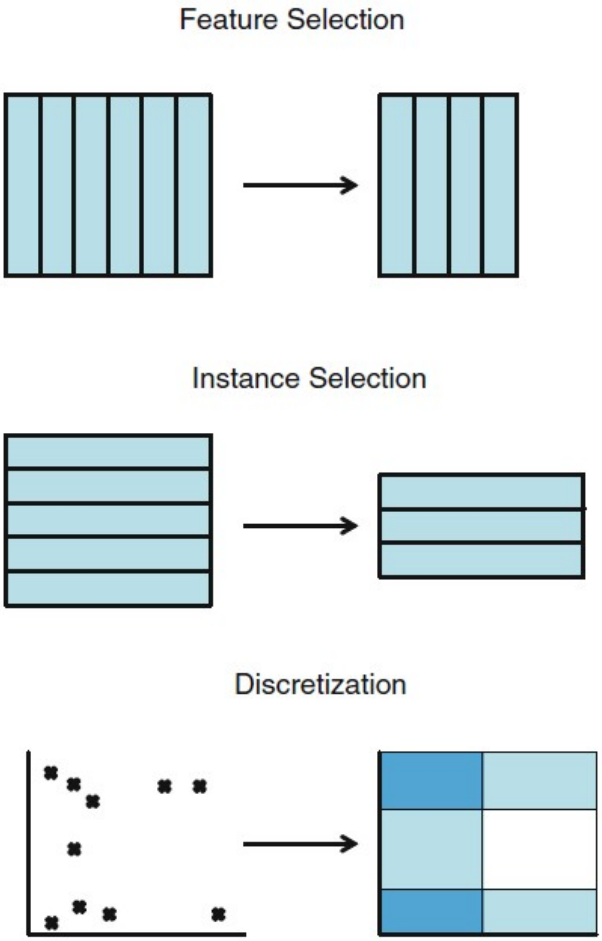


Figure 2.6: Types of dimensionality reduction [8]

pattern and increase the speed of the learning phase. Feature selection techniques are typically categorised as filters, wrappers and embedded approaches [35] [36]. Filters use measures of association between each predictor variable and the target to examine its predictive power. Wrappers look for the optimal subset of features by using predictive or trained algorithms. Specifically, they use different combinations or subsets of attributes to find the best subset of features. The embedded approach explores the advantages of both wrappers and filters to identify the best features, using attribute subsets and checking the performance of the corresponding models [35].

- **Selection of instances** consists of choosing a data set from the total to use in the algorithm. This technique is present in several models of Data Mining to check internal validation and avoid over-fitting.

- **Discretization** transforms quantitative data into qualitative data, that is, numerical attributes into discrete or nominal attributes with a finite number

of intervals, obtaining a non-overlapping partition of a continuous domain.

- **Extraction of characteristics and/or generation of instances** includes the removal of attributes, grouping subsets of attributes can be together or creating artificial substitute attributes. In relation to the generation of instances, it allows the creation or adjustment of artificial substitute examples that could better represent the decision limits in supervised learning.

2.6 Types of Learning

Forecasting can be performed offline, also known as batch learning, or online, also known as stream learning.

2.6.1 Batch Learning

In the last decades, research and practise of Machine Learning has focused on batch learning, typically, with small dimension data sets [2]. In batch learning, all the data which is intended for training is given to the algorithm, which produces a decision model after processing them, usually, several times. Firstly, the model is trained and then launched in production where it no new knowledge is acquired, *i.e.*, the model is static. In case it is necessary to teach the model again, a new version of the model must be trained with the complete data set, that is, with the old data and the new data, stop and replace the model in production. The training process, evaluation and launch can be automatised with relative ease, therefore, the models which follow this approach can be adapted to change [37]. This pipeline is further illustrated in Figure 2.7.

In other words, this approach uses a Dtr training set to generate an output hypothesis, which is a F function that maps instances of an input set X to a set of Y labels. Thus, these algorithms construct a statistical assumption about a probability distribution over the product space $X \times Y$. The batch learning algorithm is expected to generalise, in the sense that its output hypothesis predicts the Y labels of previously unseen examples X sampled from the distribution. To sum up, this approach is based on the random generation of examples according to some stationary probability distribution. This implies, in most cases, a great waste of time and computational resources, because of this it is done offline. Most learners that practice this approach use a greedy and hill climb search in the space of models for model optimisation. This means that the chosen models are prone to high variance and over-fitting problems [15].

[2] Sets of related information that are composed of separate elements, but can be manipulated as a unit by a computer.

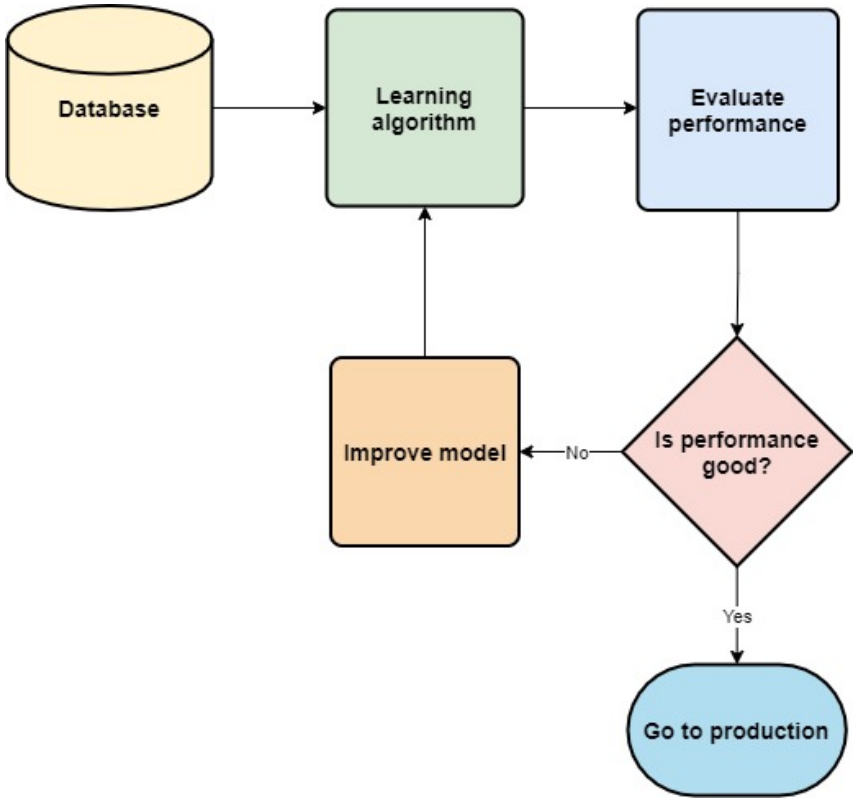


Figure 2.7: Batch learning pipeline [9]

When using small data sets, the main problem is the reduction of variance, while in learning with large data sets it can become more effective to use algorithms that place greater emphasis on bias management [38]. In contrast, automatic data feeds distinguish the current data set from previous ones. This is due to the fact that information is not only entered into computers by people but also by other computers. Some examples of these applications are telecommunications data management, sensor networks, network monitoring and financial applications. In these applications, it is not feasible to load the arriving data into a traditional Database Management System (DBMS). The reason for this is a DBMS is not traditionally designed to directly support the continuous queries required in these applications [39].

The constraints enumerated imply switching to a new perspective, one that can adapt better to aspects which entail characteristics such as [15]:

- Data made available through continuous streams that flow at high speeds over time;
- Data can no longer be considered as independent and identically distributed;
- Data are now often spatially as well as time distributed;

The online processing approach explores these approaches.

2.6.2 Stream Learning

Traditional DBMS are not designed for rapid and continuous loading of events, and do not support the continuous queries that are typical of data stream applications. They are built on the concept of persistent data that are stored reliably in stable storage and queried/updated several times throughout their lifetime. Furthermore, it is recognised that both approximation and adaptability are key ingredients in executing queries and performing other processing, e.g., data analysis and mining, over rapid data streams, while traditional DBMS focus largely on the opposite goal of precise answers computed by stable query plans [39].

In order to solve these problems, the database community has developed Data Stream Management Systems (DSMS), also called STREAM and, for continuous querying, compact data structures, and sub-linear algorithms for massive data set analysis [40]. In the data stream model, some or all of the input data are not available for random access from disk or memory, but rather arrive as one or more continuous data streams. Data streams differ from the conventional stored relation model in several ways [15]:

- The data elements in the stream arrive online;
- Data streams are potentially unbounded in size;
- Once an element from a data stream has been processed it is discarded or archived, *i.e.*, it cannot be retrieved easily unless it is explicitly stored in memory, which typically is small relative to the size of the data streams;
- The system has no control over the order in which data elements arrive to be processed, either within a data stream or across data streams.

The first three constraints limit the amount of memory and time-per-item that the streaming algorithm can use. The last one imposes the need to adapt to time changes. The main differences between DBMS and DSMS are summarised in Table 2.3.

Table 2.3: Comparison between DBMS and DSMS [15]

Data Base Management Systems	**Data Stream Manage Systems**
Persistent relations	Transient streams (and persistent relations)
One-time queries	Continuous queries
Random access	Sequential access
Access plan determined by query processor and physical DB design	Unpredictable data characteristics and arrival patterns

In the streaming model the input elements $a_1, a_2, ..., a_j, ...$ arrive sequentially, item by item, and describe an underlying function F. Streaming models on how f_i describes F. Regarding these models, three distinctions can be made:

- **Insert Only Model**: once an element f_i is seen, it can not be changed;
- **Insert-Delete Model**: elements f_i can be deleted or updated;
- **Additive Model**: each f_i is an increment to $F[j] = F[j] + j_i$.

Stream learning is illustrated in Figure 2.8, where data streams (either individually or in mini-batches) flow into the learning algorithm and update the model.

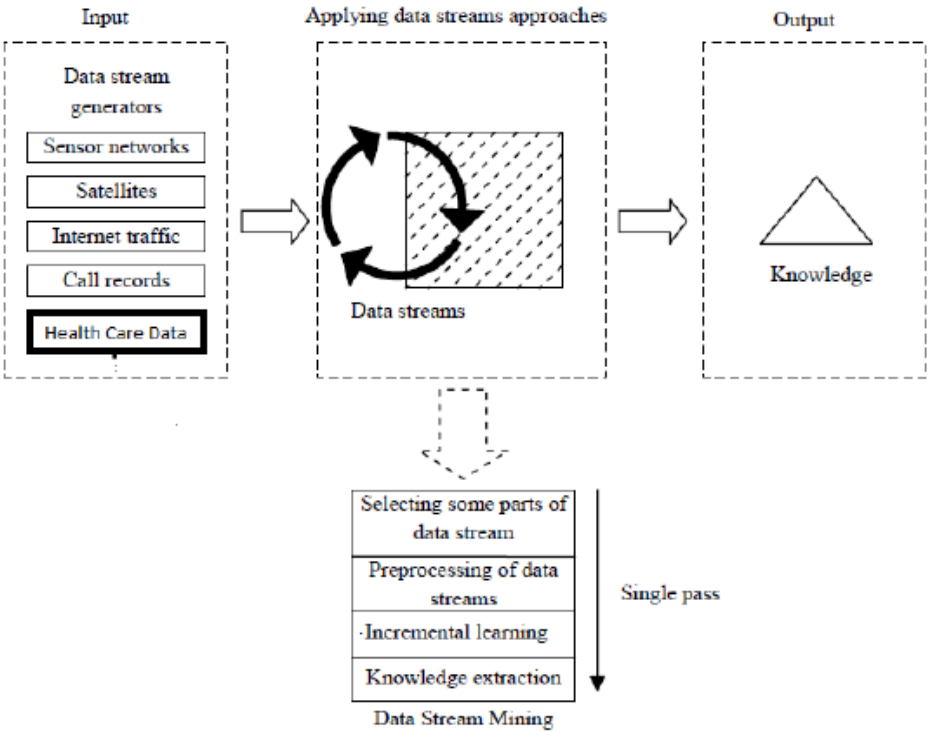


Figure 2.8: Stream learning pipeline [10]

Building a general-purpose DSMS poses many interesting challenges [40, 15]:

- Approximate query processing techniques to evaluate queries that require unbounded amount of memory.
- Sliding window query processing both as an approximation technique and as an option in the query language.
- Sampling to handle situations where the flow rate of the input stream is faster than the query processor.
- The meaning and implementation of blocking operators (e.g., aggregation and sorting) in the presence of unending streams.

- Declarative queries must be translated into physical query plans that are flexible enough to support optimisations and fine-grained scheduling decisions.
- Achieving high performance requires that the DSMS exploit possibilities for sharing state and computation within and across query plans. In addition, constraints on stream data (e.g., ordering, clustering, referential integrity) can be inferred and used to reduce resource usage.
- Since data, system characteristics, and query load may fluctuate over the lifetime of a single continuous query, an adaptive approach to query execution is essential for good performance.
- When incoming data rates exceed the DSMS's ability to provide exact results for the active queries, the system should perform load-shedding by introducing approximations that gracefully degrade accuracy.
- Due to the long-running nature of continuous queries, DSMS administrators and users require tools to monitor and manipulate query plans as they run.

2.7 Time Series

A time series is a set of observations X_t, each one being recorded at a specific time t. The analysis of time series is based on the assumption that successive values of a random variable represent consecutive measurements taken at spaced time intervals.

Time series variables can display a wide variety of patterns, therefore, Time Series Analysis (TSA) refers to applying data analysis techniques to model dependencies in the sequence of measurements.

A time series can be classified into two different types [41]:

- **Stock** series is a measure of certain attributes at a point in time and can be thought of as *stocktakes*. For example, the Monthly Labour Force Survey is a stock measure because it takes stock of whether a person was employed in the reference week.
- **Flow** series measure the activity over a given period. For example, a telco top-up activity or stock trading activity.

The main difference between a stock and a flow series is that flow series can contain effects related to the calendar (for example trading day effects). Both types of series can still be seasonally adjusted using the same seasonal adjustment process.

An observed time series can be decomposed into three components: the trend (long term direction); the seasonal (systematic, calendar related movements); and the aberrant observations (unsystematic, short term fluctuations) [41].

2.7.1 Trend

The trend is defined as the long term movement in a time series without calendar related and irregular effects, and is a reflection of the underlying level [42]. Depending on the problem being analysed, it can be the result of influences such as population growth, price inflation and general economic changes.

When decomposing a time series, the trend on the models is typically additive or multiplicative [41].

Additive

In some time series, the amplitude of both the seasonal and irregular variations do not change as the level of the trend rises or falls. In such cases, an additive model is appropriate.

In the additive model, the observed time series O_t is considered to be the sum of three independent components: the seasonal S_t, the trend T_t and the irregular I_t. The equation for this model is displayed is:

$$O_t = S_t + T_t + I_t \tag{2.3}$$

Each of the three components has the same units as the original series. The seasonally adjusted series is obtained by estimating and removing the seasonal effects from the original time series. The estimated seasonal component is denoted by $\hat{S}_t$. The seasonally adjusted estimates can be expressed by:

$$SA_t = O_t - \hat{S}_t = T_t + I_t \tag{2.4}$$

Multiplicative

In many time series, the amplitude of both the seasonal and irregular variations increase as the level of the trend rises. In this situation, a multiplicative model is usually appropriate.

In the multiplicative model, the original time series is expressed as the product of trend, seasonal and irregular components

$$O_t = S_t \times T_t \times I_t \tag{2.5}$$

The seasonally adjusted data then becomes:

$$SA_t = \frac{O_t}{\hat{S}_t} = T_t \times I_t \tag{2.6}$$

Under this model, the trend has the same units as the original series, but the seasonal and irregular components are unit-less factors, distributed around 1.

In essence, the additive decomposition is the most appropriate if the magnitude of the seasonal fluctuations, or the variation around the trend-cycle, does not vary with the level of the time series. When the variation in the seasonal pattern, or the variation around the trend-cycle, appears to be proportional to the level of the time series, then a multiplicative decomposition is more appropriate [42]. As visual reference, Figure 2.9 displays a examples of additive and multiplicative trend.

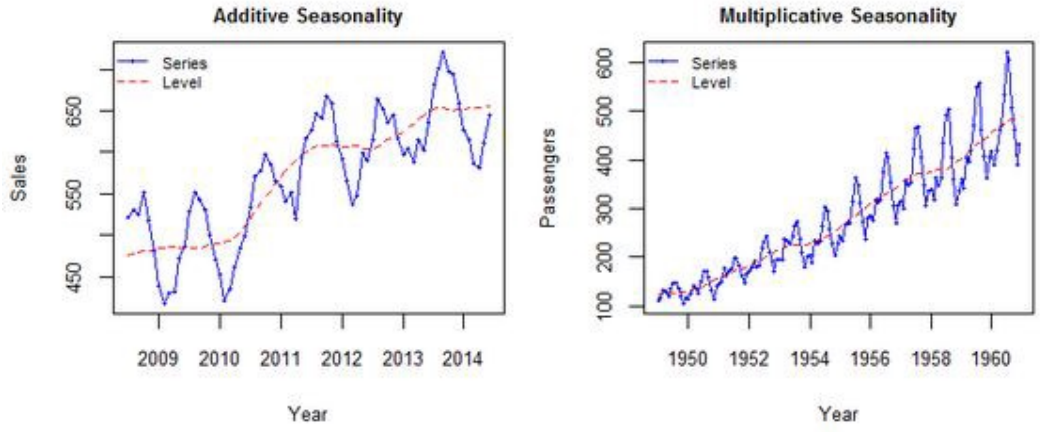


Figure 2.9: Example of additive and multiplicative trend [11]

2.7.2 Seasonality

A seasonal pattern occurs when a time series is affected by seasonal factors, *i.e.*, reasonably stable with respect to timing, direction and magnitude, such as the time of the year or the day of the week [42]. Seasonality is always of a fixed and known period and it arises from systematic, calendar related influences such as: natural conditions (ex: weather seasons); business and administrative procedures (ex: start and end of the school term); and social and cultural behaviour (ex: Christmas). It also includes calendar related systematic effects that are not stable in their annual timing or are caused by variations in the calendar from year to year, such as: trading day effects (ex: number of weeks in a month will differ from year to year); moving holiday effects (ex: Easter) [41].

2.7.3 Aberrant Observations

For many time series variables it can happen that one or more observations are markedly different from the other observations. This often is due to the occurrence of exceptional and usually unpredictable events. Such outliers occur rarely, are (often) unforecastable, and are (assumed to be) caused by exogenous influences. These outliers are known as aberrant observations or residuals.

Sometimes aberrant observations are part of the process which one wants to model, that is, they are in fact the most interesting observations in a time series. For example, the effect of substantial price discounts on product sales makes the low price data very informative. Other effects that can lead to outliers are more difficult to capture with a time series forecasting model. For example, again price discounts but now by competitors generally are impossible to predict by the own company. In sum, aberrant observations often are influential, and hence should be dealt with [43].

2.8 Regression Models

The point of Data Mining is to have a variety of tools available to assist the analyst and user in understanding what the data consists of. For that purpose it uses many different methods that can originate from both classical statistics as well as Artificial Intelligence (AI). Statistical techniques have strong diagnostic tools that can be used for development of confidence intervals on parameter estimates and hypothesis testing, for example. Artificial Intelligence techniques require fewer assumptions about the data, and are generally more automatic. Table 2.4 seeks to demonstrate this evidence with a few examples.

Table 2.4: Comparison between Data Mining models and their basis [16]

Algorithm	**Function**	**Basis**	**Task**
Cluster Detection	Cluster analysis	Statistics	Classification
Regression	Linear regression	Statistics	Prediction
	Logistic Regression	Statistics	Classification
	Discriminant analysis	Statistics	Classification
Neural Networks	Neural networks	AI	Classification
	Kohonen netsai	AI	Cluster
Decision Trees	Association rules	AI	Classification
Rule Induction	Association rules	AI	Description
Link Analysis			Description
	Query tools		Description
	Descriptive statistics	Statistics	Description
	Visualization tools	Statistics	Description

Descriptive modelling are usually applied to initial data analysis, where the intent is to gain initial understanding of the data, or to special kinds of data involving relationships or links between objects (hence why after acquiring the data, the first step is the previously presented EDA). There are cases where a specific problem is best treated with a particular type of algorithm. And other cases where different types of algorithm types can be used for the same problem.

Most of the work in Machine Learning focuses on individual learning tasks. While great success has been achieved in this type of framework, it is clear that having a learner work on several tasks simultaneously, instead of sequentially, should certainly be some advantage, especially if the tasks are closely related in some way. The

approach of learning multiple tasks simultaneously is given the name of multi-task learning or multi-target learning.

Existing methods for multi-target regression can be categorised as: problem transformation methods, which transform the multi-target problem into independent single-target problems each solved using a single-target regression algorithm; and algorithm adaptation methods, which adapt a specific single-target method to directly handle multi-target data sets.

Adaptation methods are based on the idea of simultaneously predicting all the targets using a single model that is able to capture all dependencies and internal relationships between them. This actually has several advantages over transformation methods: it is easier to interpret a single multi-target model than many single-target models and it ensures better predictive performance especially when the targets are correlated [44]. For this reason, only the adaptation algorithm methods were studied.

Regression analysis is a technique for estimating a functional relationship between one or more dependent variables and a set of independent variables. It has been widely studied in statistics, pattern recognition, Machine Learning and Data Mining. Since this work addresses a forecasting problem, regression techniques were studied.

The following sections present offline learning techniques for single-target and multi-target regression and online learning techniques for multi-target regression.

2.8.1 Multiple Linear Regression

Ordinary regression is the popular technique for predicting a quantitative outcome, *i.e.*, takes on numerical variables, such as profit and sales. This modelling task aims to create a (linear or non-linear) map between the independent variables (i.e. the various features) and a set of continuous dependent variables (i.e. the variable you want to predict) by estimating a set of parameters [45]. Regression is used on a variety of data types. For example time series data often uses regression models for forecasting. It is considered the workhorse of profit modelling as its results are taken as the gold standard. Moreover, the ordinary regression model is used as the benchmark for assessing the superiority of new and improved techniques.

The multiple linear regression or Multiple Ordinary Least Squares (MOLS) model is defined in Equation 2.7 [16], where Y is the dependent variable (the one being forecast), X_n are the n independent (explanatory) variables β_0 is the intercept term, β_n are the n coefficients for the independent variables, ε is the error term.

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_n X_n + \varepsilon \tag{2.7}$$

MOLS regression is the straight line (with intercept and slope coefficients β_n) which minimises the sum of squared error terms ε_i over all i observations. The idea is to look at past data to determine the β coefficients which worked best. The model gives the most likely future value of the dependent variable given knowledge of the X_n for future observations. This approach assumes a linear relationship, and error terms that are normally distributed around zero without patterns. While these assumptions are often unrealistic, regression is highly attractive because of the existence of widely available computer packages as well as highly developed statistical theory. Statistical packages provide the probability that estimated parameters differ from zero [16].

Advantages and Disadvantages

The ordinary regression model has the following advantages [46]:

- **Simplicity.** Linear regression is a very simple algorithm that can be implemented very easily to give satisfactory results. Furthermore, these models can be trained easily and efficiently even on systems with relatively low computational power when compared to other complex algorithms. Linear regression has a considerably lower time complexity when compared to some of the other Machine Learning algorithms. The mathematical equations of linear regression are also fairly easy to understand and interpret. Hence linear regression is very easy to master.

- **Performance on linearly separable data sets.** Linear regression fits linearly separable data sets almost perfectly and is often used to find the nature of the relationship between variables.

- **Overfitting can be reduced by regularisation.** Overfitting is a situation that arises when a Machine Learning model fits a data set very closely and hence captures the noisy data as well. This negatively impacts the performance of model and reduces its accuracy on the test set. Regularisation is a technique that can be easily implemented and is capable of effectively reducing the complexity of a function so as to reduce the risk of overfitting.

However, the ordinary regression model has several limitations [46]:

- **Underfitting.** A situation that arises when a Machine Learning model fails to capture the data properly. This typically occurs when the hypothesis function cannot fit the data well. Since linear regression assumes a linear relationship between the input and output variables, it fails to fit complex data sets properly. In most real life scenarios the relationship between the variables of the

data set is not linear and, hence, a straight line does not fit the data properly. In such situations a more complex function can capture the data more effectively. For this reason, most linear regression models have low accuracy.

- **Outliers.** They are anomalies or extreme values that deviate from the other data points of the distribution. Data outliers can damage the performance of a Machine Learning model drastically and can often lead to models with low accuracy. Outliers can have a very big impact on linear regression's performance and, thus must be dealt with appropriately before linear regression is applied on the data set.

- **Linear Regression assumes that the data is independent.** Very often the inputs are not independent of each other and, hence, any multi-collinearity must be removed before applying linear regression.

2.8.2 Multivariate Linear Regression

The distinction between multivariate linear models and standard (univariate) linear models is simply that multivariate linear models involve more than one dependent variable. Let the dependent variables be $y_1, \cdots, y_q$. If n observations are taken on each dependent variable, we have $Y_{il}, \cdots, Y_{iq}, i = 1, \cdots, n$. Let $Y_1 = [y_u, \cdots, Y_{n1l}]$' and, in general, $Y_h = [Y_{lh}, \cdots, Y_{nhl}]', h = 1, \cdots, q$. For each h, the vector Y_h is the vector of n responses on the variable Y_h and can be used as the response vector for a linear model. For $h = 1, \cdots, q$, write the linear model [47], where X is a known $n \times p$ matrix that is the same for all dependent variables, but β_h vector and the error vector $e_h = [e_{lh}, \cdots, e_{nh}]$' are associated to the dependent variable.

$$Y_h = X\beta_h + e_h, E(e_h) = 0, Cov(e_h) = \sigma_{hh}I \tag{2.8}$$

The model can be rewritten as:

$$\begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_q \end{bmatrix} = \begin{bmatrix} X & 0 & \cdots & 0 \\ 0 & X & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & 0 & & X \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_q \end{bmatrix} + \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_q \end{bmatrix},$$

where the error vector has mean zero ($E(eh) = 0$) and the covariance matrix $Cov(eh)$ is given by:

$$\begin{bmatrix} \sigma_{11}I_n & \sigma_{12}I_n & \cdots & \sigma_{1q}I_n \\ \sigma_{12}I_n & \sigma_{22}I_n & \cdots & \sigma_{2q}I_n \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{1q}I_n & \sigma_{2q}I_n & \cdots & \sigma_{qq}I_n \end{bmatrix}$$

In essence, this model consists of fitting the q linear models simultaneously. For this, the following matrices are used:

$$Y_{n\times q} = [Y_1, \cdots, Y_q],$$
$$B_{p\times q} = [\beta_1, \cdots, \beta_q],$$

$$e_{n\times q} = [e_1, \cdots, e_q]$$

As a result, the multivariate linear model is given by Equation 2.9:

$$Y = XB + e \tag{2.9}$$

Advantages and Disadvantages

The advantages of the Multivariate Linear Regression (MLR) model are an extension of the ones previously mentioned for the MOLS model. That is, additionally to the aforementioned advantages, the MLR model has the ability to predict multiple dependent variables with multiple independent variables which eliminates the need to perform several multiple linear regressions to solve a problem. This models limitations are [48]:

- Multivariate techniques are a bit complex and require a high-levels of mathematical calculation.
- The output of a multivariate regression model is not always easy to interpret, because the loss and error output are not identical.
- This model requires large data sets.

Learning from data streams requires incremental learning, using limited computational resources, and the ability to adapt to changes in the process generating data. As such, this section presents multi-target regression techniques for time evolving data.

2.8.3 Decision Tree

Trees can be used for classification or regression. However, only the case regression will be explored in this study. The basic idea of trees – or, more formally, recursive partitioning – is to chop the space of explanatory variables and their values into subsets and, then, assign a response value for each point conditionally on the subset to which the point belongs.

The process of building a regression tree can be broken down into two steps [49]:

- Divide the predictor space - that is, the set of possible values for each feature into J distinct and non-overlapping regions, $R_1, R_2, \cdots, R_J$.

- For every observation falling in a region, a prediction is made, corresponding to the mean of the values for the response values of the observations in the region.

The regions are built by dividing the predictor space into high-dimensional rectangles, for simplicity and for ease of interpretation of the resulting predictive model. In essence, the goal is to find the regions that minimise the Residual Sum of Squares (RSS), given by Equation 2.10 [49], where $\hat{y}_{R_j}$ is the mean response for the training observations within the j^{th} box.

$$\sum_{j=1}^{J} \sum_{i \in R_j} (y_i - \hat{y}_{R_j})^2 \tag{2.10}$$

Unfortunately, it is computationally infeasible to consider every possible partition of the feature space into J boxes. For this reason, a top-down, greedy approach known as recursive binary splitting is employed. The approach is top-down because it begins at the top of the tree (at which point all observations belong to a single region) and, then, successively splits the predictor space; each split is indicated via two new branches further down on the tree. It is greedy because at each step of the tree-building process, the best split is made at that particular step, rather than looking ahead and picking a split that will lead to a better tree in some future step.

Decision Trees (DT) can easily be extended towards the case of multi-target prediction, by extending the notion of variance towards the multi-dimensional case. They define the variance of a set as the mean squared distance between any element of a set and a centroid of the set. Depending on the definition of distance, which could be Euclidean distance in a multidimensional target space, a decision tree will be built that gives accurate predictions for multiple target variables [50].

2.8.4 Random Forest

Bagging or bootstrap aggregation is a technique for reducing the variance of a statistical learning method. This technique works specially well for high-variance, low-bias procedures, such as trees. In the case of regression, it fits the same regression tree many times to bootstrapped sampled versions of the training data, and average the result. The essential idea in bagging is to average many noisy but approximately unbiased models, and hence reduce the variance.

Trees are ideal candidates for bagging, since they can capture complex interaction structures in the data, and if grown sufficiently deep, have relatively low bias. Since trees are notoriously noisy, they benefit greatly from the averaging.

Random Forest (RF) is an ensemble ML method where weak learners are grouped into a strong learner to make a final decision.

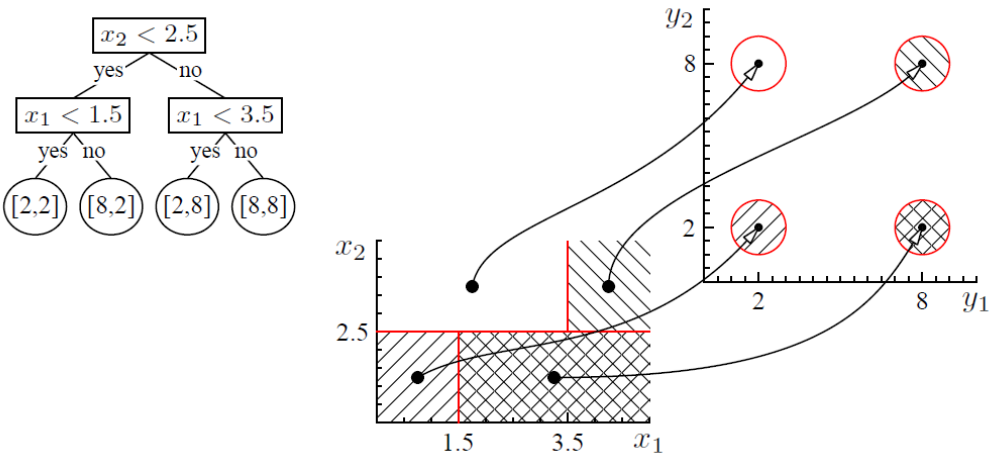


Figure 2.10: A multi-target regression tree together with its mapping from the input to the target space [12]

The idea in random forests is to improve the variance reduction of bagging by reducing the correlation between the trees, without increasing the variance too much. This is achieved in the tree-growing process through random selection of the input variables. When growing a tree on a bootstrapped data set, before each split, it selects a combination of the input variables at random as candidates for splitting.

Thereby, there are two important parameters to be tuned, in order to meet performance and avoid computational constraint:

- **Number of estimators**. The number of estimators, *i.e.* DT instances, is important to be defined with a high value, typically between 100 and 500. Increasing the value for this parameter, increases the computational requirements for the model which, at a certain point, is no longer beneficial;
- **Number of features sampled**. At each node, a DT instance makes a decision taking into account features randomly sampled, leading to the feature and split threshold that grants the best division of the classes.

2.8.5 Multi-Layer Perceptron

Neural networks are inspired on the human brain and consist of input, hidden and output layers. The objective of the neural network is to transform the inputs into meaningful outputs. Multi-Layer Perceptron (MLP) is the most common neural network model.

A perceptron, also known as neuron, is a linear function which, given an input x, will produce an output based on some internal parameters. It can be implemented mathematically through Equation 2.11, where w corresponds to the weight and b to

the bias. Nevertheless, unlike regular functions, a perceptron can learn the optimal values for w and b. This is achieved by minimising the average output error for a set of right example pairs $(x, f(x))$.

$$f(x) = w \times x + b \tag{2.11}$$

MLP neural networks consist of units arranged in layers. Each layer is composed of nodes. Each MLP is composed of a minimum of three layers consisting of an input layer, one or more hidden layer(s) and an output layer. The input layer distributes the inputs to subsequent layers. Input nodes have linear activation functions and no thresholds. Each hidden unit node and each output node have thresholds associated with them in addition to the weights. The hidden unit nodes have nonlinear activation functions and the outputs have linear activation functions [51]. A nonlinear activation function is an optimisation algorithm which aims to minimise loss. This algorithm uses the Adam optimiser which presents the following benefits [52]:

- computationally efficient and low memory requirements;
- well suited for problems that are large in terms of data and/or features;
- invariant to diagonal rescaling of the gradients;
- hyper-parameters have intuitive interpretation and typically require little tuning.
- appropriate for problems with very noisy/or sparse gradients;

Hence, each signal feeding into a node in a subsequent layer has the original input multiplied by a weight with a threshold added and then is passed through an activation function. A typical three-layer network is shown in Figure 2.11.

2.8.6 Adaptive Model Rules

The Adaptive Model Rules (AMR) algorithm can learn ordered or unordered rules. The antecedent of a rule is a set of conditions based on the attribute values (literals) and the consequent is a function that minimises the mean square error of the target attribute computed from the set of examples covered by rule.

The algorithm begins with an empty rule set, and a default rule $\{\} \rightarrow \zeta$. Every time a new training example is available the algorithm proceeds with checking whether, the example is covered by any rule in the rule set, *i.e.* if all the literals are true for the example. The target values of the examples covered by a rule are used to update the sufficient statistic of the rule. Before an example is covered by any rule, change detection tests are updated with every example of this rule. For the change detection, the Page-Hinckley (PH) change detection test is used to monitor

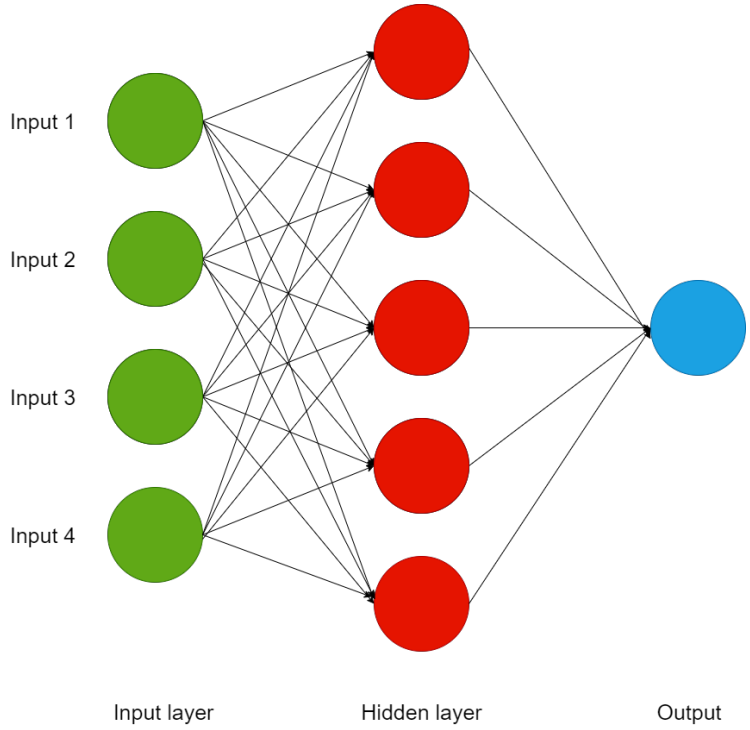


Figure 2.11: Example of multi-layer perceptron

the online error of each rule. If a change is detected, the rule is removed from the rule set. Otherwise, the rule is expanded. The expansion of the rule is considered only after certain period (N_{min} number of example). The set of rules is learned in parallel and two cases are considered: learning ordered or unordered set of rules. In the former, every example updates statistics of the first rule that covers it. In the latter, every example updates statistics of all the rules that covers it. If an example is not covered by any rule, the default rule is updated [53].

2.8.7 Basic Multi-Target Regressor

The Basic Multi-Target Regressor (BMTR) algorithm is centred on rule learning. Rule learning is based on implications called rules, where antecedent A_r is a conjunction of conditions, also called literals, that create partitions in the input variables x_i space and the consequent C_r is a predicting function, $R_r = (A_r \Rightarrow C_r)$. The literals present different forms whether the data is numerical or nominal. For numerical data, they can be, for example, $X_j \leq v$ and $X_j > v$. In the case of nominal data, an instance can be $X_j = v$ and $X_j \neq v$. Where X_j represents the j^{th} input variable. A rule, R_r, is set to cover all the features, if all the features meet all the conditions. Support $S(x_i)$ corresponds to a set of rules that cover x_i and ζ_r returns a prediction $\hat{y}_i$ if a rule $R_r \in S(x_i)$ [13].

In Figure 2.12, associated to each rule is a data structure, ζ_r, which contains the necessary statistics to the algorithm's training and prediction. Specifically, ζ_r contains the input variables statistics I_r. The default rule D exists for initial conditions and for the case of none of the current rules covers the example $S(x_i) = \emptyset$. The antecedent of D and is initially empty. Rule set is formed by a set of U learned rules defined as $R = \{R_1, \cdots, R_r, \cdots, R_U\}$ and a default rule D.

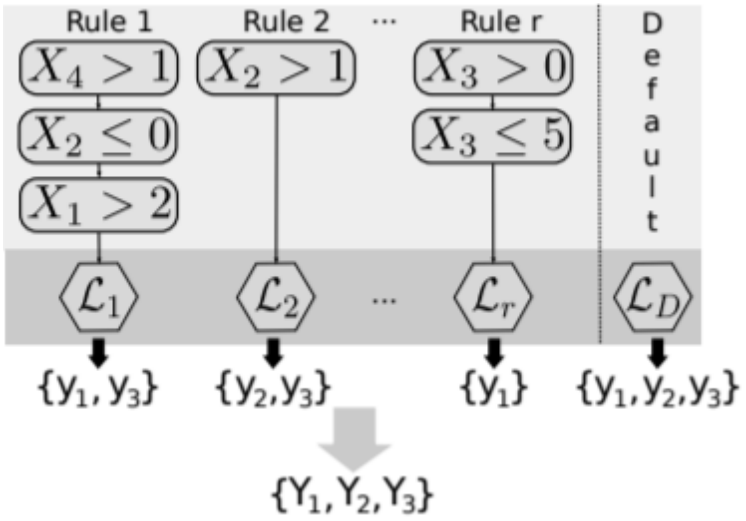


Figure 2.12: Rule learning in basic multi-target regression [13]

2.8.8 iSOUP Tree

iSOUP-Tree is a supervised incremental tree-based learner that utilises the Hoefding inequality and a variance-reduction-based splitting heuristic. iSOUP-Trees have been used to address the MTR task [54], as well as the multi-label classification task [55], in the online learning setting [56].

The iSOUP maintains, on one hand, a multi-target perceptron and on the other hand the multi-target mean predictor that computes the prediction as the mean value of each of the targets observed at a given leaf [57].

2.8.9 Multi-Target Perceptron Regressor

The Multi-Target Perceptron Regressor (MTPR) algorithm consists of a neural network made up of only one neuron and it uses the sigmoid function instead of a threshold. Since it is used for online learning, instead of implementing batch updates, it makes use of stochastic gradient descent, for every new instance, to update the model. In environments with a multitude of targets, it provides one perceptron per target. This version of the perceptron was used in conjunction with decision trees [58].

2.9 Prediction Interval

Prediction intervals are used in both frequentist and Bayesian statistics: a prediction interval bears the same relationship to a future observation that a frequentist confidence interval or Bayesian credible interval bears to an unobservable population parameter. Prediction intervals predict the distribution of individual future points, whereas confidence intervals and credible intervals of parameters predict the distribution of estimates of the true population mean or other quantity of interest that cannot be observed [59].

In fact, the frequentist believes in data - if one can't show a result with data, then one can't believe the result. On the other hand, the Bayesian says that if one has information beyond data, specifically a prior probability, then this should be used. The issue arises when the prior probability comes from personal belief rather than data, that is, is subjective rather than objective [60].

All in all, since prediction intervals are only concerned with past and future observations, rather than unobservable population parameters, they are advocated as a better method than confidence intervals by some statisticians.

Given a sample from a normal distribution, whose parameters are unknown, it is possible to predict intervals in the frequentist sense, *i.e.*, an interval $[a, b]$ based on statistics of the sample such that, on repeated experiments, X_{n+1} falls in the interval the desired percentage of the time. These intervals can be named predictive confidence intervals.

A general technique of frequentist prediction intervals is to find and compute a pivotal quantity of the observables $X_1, \cdots, X_n, X_{n+1}$ – a function of observables and parameters whose probability distribution does not depend on the parameters – that can be inverted to give a probability of the future observation X_{n+1} falling in some interval computed in terms of the observed values so far, $X_1, \cdots, X_n$. Such a pivotal quantity, depending only on observables, is called an ancillary statistic. The usual method of constructing pivotal quantities is to take the difference of two variables that depend on location, so that location cancels out, and then take the ratio of two variables that depend on scale, so that scale cancels out [61].

It is a fact that a large standard deviation indicates that the data samples can spread far from the mean and a small standard deviation indicates that they are clustered closely around the mean. Standard deviation may serve as a measure of uncertainty and is visible in several real life industries.

Population standard deviation is used to set the width of Bollinger Bands, a widely adopted technical analysis tool. Bollinger Bands are a type of statistical chart characterising the prices and volatility over time of a financial instrument or commodity, using a formulaic method. Financial traders employ these charts as a methodical tool to inform trading decisions, control automated trading systems, or

as a component of technical analysis. Bollinger Bands display a graphical band and volatility in one two-dimensional chart [62].

Two input parameters chosen independently by the user govern how a given chart summarises the known historical price data, allowing the user to vary the response of the chart to the magnitude and frequency of price changes, similar to parametric equations in signal processing or control systems.

Bollinger Bands consist of an N-period Moving Average (MA), an upper band at K times, an N-period standard deviation above the moving average ($MA + K\sigma$) and a lower band at K times with an N-period standard deviation below the moving average ($MA - K\sigma$) [62].

The chart thus expresses arbitrary choices or assumptions of the user, and is not strictly about the price data alone. Figure 2.13, represents a chart of a Bollinger Band of the American Express stock (NYSE: AXP) from 2008 where, in colour blue, it can be seen the the moving average and the upper and lower bands.

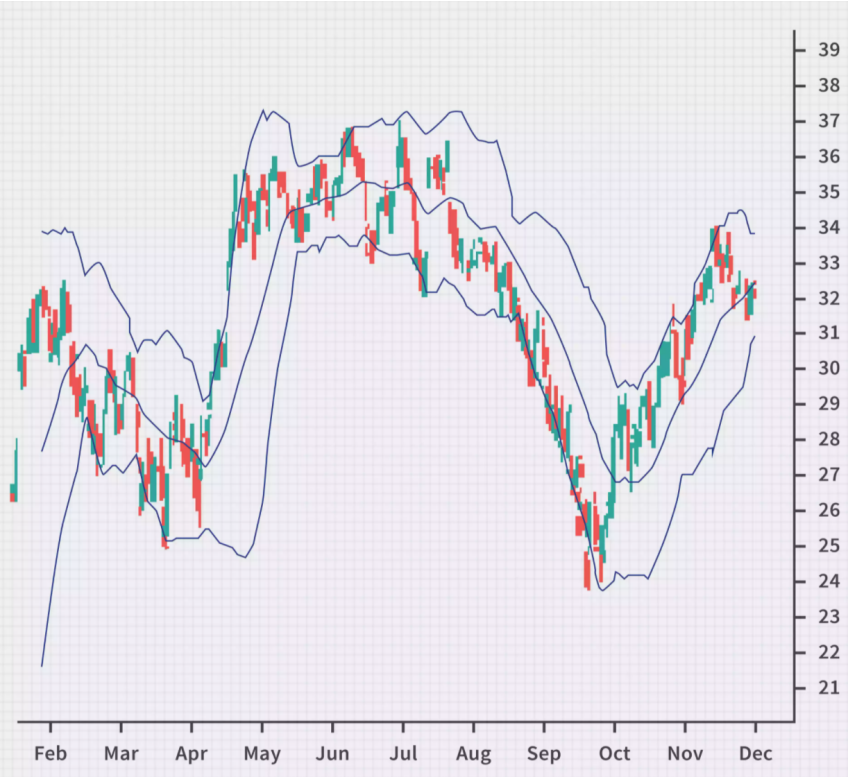


Figure 2.13: Bollinger bands in American Express stock from 2008 [14]

Financial time series, just like telecommunications time series, are known to be non-stationary, whereas the statistical calculations above, such as standard deviation, apply only to stationary series.

2.10 Top-Up

The literature review focused on the profiling of prepaid telco subscribers and prediction of their behaviour, both offline and online. However, not only the works found address mostly the problem of customer churn prediction, *i.e.*, customers migrating to a competitor, rather than customer top-up prediction, but the majority of the research works adopt batch rather than stream processing. By predicting future churners, telcos can implement proactive service retention measures, namely, personalised marketing actions, involving the offer better service conditions [63]. In this field, it is also common to profile customers through segmentation [63, 64, 65] and, then, launch tailored retention campaigns targeting the identified churners. Although customer churn and top-up are related, since both rely on customer profiling and proactive tailored marketing as retention strategies, the outcome of a top-up predictor is continuous while that of a churner predictor is binary.

2.10.1 Offline

Offline batch processing creates static models from stored data sets to generate predictions for a limited time frame. The following works implement offline processing.

Caigny *et al.* propose a hybrid algorithm, the Logit Leaf Model (LLM), to better classify data [66]. The idea behind the LLM is that different models constructed on segments of the data rather than on the entire data set lead to better predictive performance while maintaining the comprehensibility from the models constructed in the leaves. The LLM consists of two stages: a segmentation phase and a prediction phase. In the first stage customer segments are identified using decision rules and in the second stage a model is created for every leaf of this tree.

Nie *et al.* implement a churn prediction model using credit card data collected from a real Chinese bank [67]. After a feature selection process, logistic regression and decision trees are studied using the accuracy of analytic results, and a misclassification cost measurement by taking the two types error and the economic sense into account. The test result shows that regression performs a little better than decision tree.

Jain *et al.* studied how two machine-learning techniques were used for predicting customer churn logistic regression and logit Boost using Orange an American telecommunication company database [68]. The results show that both techniques outperformed and have similar accuracy measures.

In a recent survey, Jain *et al.* [69], identify several offline Machine Learning algorithms used to build churn prediction models. They include logistic regression, multi-layer perceptrons, rough set theory, support vector machines, k-means and fuzzy C-mean, Bayesian belief networks, decision trees, convolutional neural networks as well as ensemble methods like random forest, bagging or AdaBoost. A

large number of churn prediction models rely on logistic regression and decision trees to classify customers into churners and non-churners [70, 71].

Yang *et al.* propose a random forest prediction model together with a monthly sliding window to recognise churn customers in two months time based on the current month data [72]. The idea is to provide the telco with a one month interval to convince the future churner to remain a customer.

Diettrichand *et al.* implement a credit score system for a telco company targetting pre-paid clients [73]. On a first stage a decision tree was utilised to create a minimum viable product for the credit score based on features from top-up date and usage data. On the second stage, techniques such as logistic regression, random forest and decision trees was utilised, taking into account the credit scores created for each client on stage one, to predict if the client would make a purchase or not. The random forest technique in combination with the credit score displayed the best performance.

Sundsøy *et al.*, in [74], apply techniques such as random forest, gradient boosting machines and deep learning to customer phone data to predict household income. Firstly, a group of features is selected and then experiments are made in which the objective is to predict if the target is below/above poverty level or below/above median income. In this study, the top algorithm was deep learning.

Table 2.5, sums up the offline research found.

Another interesting approach related with the problem at hand is the Customer Lifetime Value (CLV). In business analysis, CLV is the general measure of the projected revenue that a client will bring over the lifespan of the established contract, and can be used to predict repeated client purchases. However, its main drawback is that, in most cases, it requires a decent-sized investment of time, coordination, and organisational alignment to determine and continue to analyse CLV [75]. This is aggravated when there is an extended time between purchases just like in the pre-paid pay-as-you-go telecommunications environment. Another drawback is the fact that CLV is no longer supported by Python development environments [76].

2.10.2 Online

The few online processing research works found are also dedicated to customer churn. Manzano *et al.* perform churn prediction using a stream Hoeffding adaptive tree classifier. It detects new churn patterns in real-time high-speed data streams and adapts quickly to a changing reality [77].

Machado *et al.* adopt data stream clustering, where customers are grouped by their activity patterns, and use customer behaviour change (concept drifts) to identify churners [78]. Similarly, Tatar *et al.* rely on anomaly detection together with online clustering for customer churn prediction [79].

Table 2.6, presents the results from the research for online approaches.

Furthermore, Telco data sets can be poked with the help of marketing methodologies such as RFM analysis [80] to better understand client behaviour. RFM stands for Recency, Frequency, and Monetary value, where recency represents the engagement with the operator and frequency and monetary value characterise the overall top-up behaviour.

2.11 Summary

Since the focus of this dissertation is to find a solution to predict top-up activity in data following the concept of a time series, four main concepts for the literature review were considered: Data Mining, time series, regression models, top-up forecasting implementation and prediction intervals.

The final section displays the literature review of implementations of telco top-up activity forecasting. These concepts will be further explored in the following chapter.

Table 2.5: Offline literature review

Author(s)	Title	Year	What?	Data Set	Techniques	Metrics
Arno De Caigny, Kristof Coussement, Koen W. De Bock	A New Hybrid Classification Algorithm for Customer Churn Prediction Based On Logistic Regression and Decision Trees	2018	A new hybrid algorithm (Logit Leaf Model) for customer churn prediction. It is designed to perform well in terms of both accuracy and interpretability with competitive performance from an extensive benchmarking experiment.	Financial services, retail, DIY, newspaper, telecom, energy	Logistic Regression and Decision Tree	Area Under Curve (AUC) and Top Decile Lift (TDL)
Hemlata Jaina, Ajay Khuntetab, Sumit Srivastavac	Churn Prediction in Telecommunication using Logistic Regression and Logit Boost	2020	Predict customer churn with real data from and american company Orange.	Telecom data from American company Orange	Logistic Regression and Logit Boost	Kappa Statistic, Mean Absolute Error, Root Mean Square Error, Relative Absolute Error, Root Relative Square Error, Mean Rel. Region size, True Positive Rate, False Positive Rate, Precision, Recall, F-Measures, AUC
Guangli Nie, Wei Rowe, Lingling Zhang, Yingjie Tian, Yong Shi	Credit Card Churn Forecasting by Logistic Regression and Decision Tree	2011	Examination of the performance of two data mining algorithms in credit card churn forecasting. Analysis of process of dealing with variables. Conclusion that cost should be taken into account in model evaluation and regression performs a little better than decision tree.	Credit card data from Chinese bank	Logistic Regression and Decision Tree	Misclassification cost measurement to indicate the loss caused by the error of the model
Lingling Yanga, Dongyang Lia, Yao Lu	Prediction Modelling and Analysis for Telecom Customer Churn in Two Months	2019	A new $T + 2$ churn customer prediction model, in which the churn customers in two months are recognized and the one-month window $T + 1$ is reserved to carry out churn management strategies.	China Unicom Telecom Company Guangdong Branch	Random Forest Classifier	Precision, Recall
Luciano Dietrich, Fábio de Souza, André Guerreiro	Development of Credit Scores with Telco Data Using Machine Learning and Agile Methodology in Brazil	2020	Creation of a credit score and use of Machine Learning models to predict if clients would wether or not make a purchase.	Claro Brazil	Logistic Regression, Decision Tree and Random Forest	Non-parametric statistical test, the KolmogorovSmirnov test, the Gini and ROC coefficients
Pål Sundsøy, Johannes Bjelland, Bjørn-Atle Reme, Asif M.Iqbal, Eaman Jahani	Deep Learning Applied to Mobile Phone Data for Individual Income Classification	2016	How socio-economic status in large de-identified mobile phone datasets can be accurately classified using deep learning, thus avoiding the cumbersome and manual feature engineering process	Basic phone usage, Top-up transactions, Location/-mobility, Social Network, Handset type, Revenue, Advanced phone usage	Gradient Boosting Machines, Random Forest, Deep Learning	AUC

Table 2.6: Online literature review

Author(s)	Title	Year	What?	Data Set	Techniques	Metrics
Borja Balle, Bernardino Casas, Alex Catarineu, Ricard Gavaldà	The Architecture of a Churn Prediction System Based on Stream Mining	2013	A prototype for churn prediction using stream mining methods, which offer the additional promise of detecting new patterns of churn in real-time streams of high-speed data, and adapting quickly to a changing reality.	Synthetic telecom data	Hoeffding trees	Recall, Precision
Serdar Baran Tatar, Andrew McIntyre, Nur Zincir-Heywood, Malcolm Heywood	Benchmarking Stream Clustering for Churn Detection in Dynamic Networks	2015	Exploration of the use of anomaly detection for churn prediction.	Cell phone data and online gaming data	Bio-inspired and deterministic online clustering algorithms	

Chapter 3

Presentation of the Problem and Proposed Solution

Since a proper study and understanding of the data is crucial to a well designed and implemented solution, this chapter focuses on analysing the data set and extracting insights of value for the end goal.

3.1 Problem Statement

Pay-as-you-go services are a Telcom business model used by millions around the globe on a daily basis. Achieving a fine-grained characterisation of these clients is a *must* have for telecommunication operators, to ensure better quality-of-service in an extremely competitive environment with a plethora of products and companies. Additionally, in high countries, with market penetration around 100 %, studies show that the cost to retain a client is lower that the cost to acquire a new one [81].

In fact, the goal of telecommunication companies is to enlarge their subscriber base by establishing and strengthening customer relationships. This means improving customer experience and streamlining operations. In this respect, telcos are making extensive use of Artificial Intelligence to reap the benefits of increased customer satisfaction and loyalty while decreasing fraud and improving the quality of service [82]. Operators continuously collect call, customer and network related data,

mining these data streams to analyse a number of scenarios from predictive customer support, fleet management, fraud detection, customer retention, to optimised marketing [83].

Hence, it is very appealing to identify patterns, and understand if a client will or not make a top-up in the near future. The possibility to predict top-up values allows to understand the health and evolution of the market share of the operator.

3.2 Specification

The characterisation of pay-as-you-go clients is a hard task due to their volatility. To design a predictor algorithm, we resorted to a data set from a Portuguese mobile network operator with around 400 000 clients and historical thirty-month data comprising top-up events. From the raw data, several new features were created. The Recency, Frequency, and Monetary value analysis [80] was applied to better understand the data set.

3.2.1 Data Set

The data set expands over a period of thirty months, from the beginning of January 2019 to the end of June 2021 and includes information of 374 717 pre-paid clients and has a total of 2 875 099 events. All sensitive personal and business information was anonymised. The top-up data holds, apart the individual card identification, the categorical type of top-up, the type of tariff, the date of the top-up, value of the top-up, the card balance after top-up and the age of subscription in months.

Table 3.1 displays the various types of tariff codes. Tariff codes differ on the frequency of top-up demanded or the type of top-up, *e.g.*, the top-up must be made trough an Automated Teller Machine (ATM). The combination of these different types of tariff on the same data set introduces variety in client top-up activity.

Table 3.1: Tariff codes

Tariff Code	Description	No. Clients
165	Tariff code A	20888
166	Tariff code B	11892
167	Tariff code C	187822
177	Tariff code D	7161
190	Tariff code E	1868
191	Tariff code F	21

The types of top-up were also anonymised (see Table 3.2). This information is connected client contract.

Figure 3.1 presents the distribution of top-up events per month, which has an average of 95 836.63 events per month with a coefficient of variance c_v of 8.19. Typically, the volume of top-up data displays a tendency to increase in the middle and in the end of the year. This seasonal behaviour is expected, that is, the number of

Table 3.2: Types of top-up

Type of Top-Up Code	Description	No. Clients
AUZ	Type of top-up A	132485
OYS	Type of top-up B	78187
ATO	Type of top-up C	14547
ACW	Type of top-up D	613
O	Type of top-up E	2181
7	Type of top-up F	1174
PAU	Type of top-up G	40
AMO	Type of top-up H	305
500	Type of top-up I	61
FDI	Type of top-up J	57
OCC	Type of top-up K	2

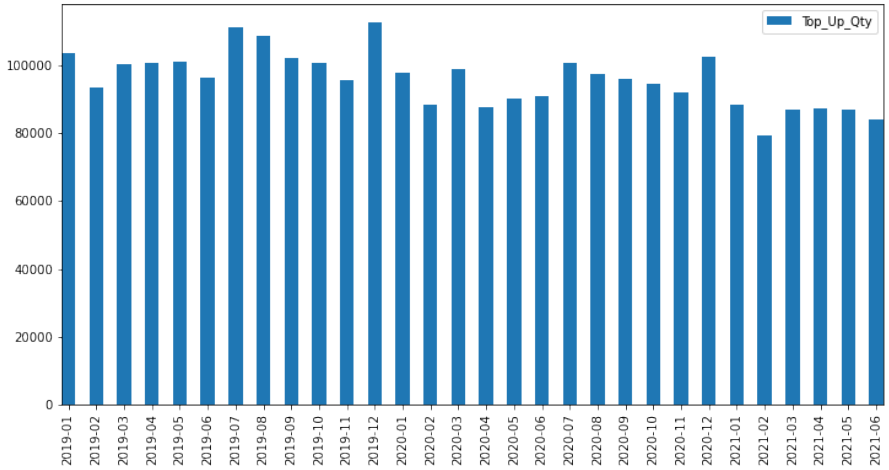


Figure 3.1: Top-up data

client events is higher during holiday seasons. Between these periods, client activity tends to reduce. As a consequence, the quantity of top-ups tends to follow the same trend.

Figure 3.2 unveils the trend, seasonality and aberrant observations of the time series. Based on this analysis, the following conclusions can be drawn:

- There is a visible downwards additive trend which is explained by the client lack of interest on Subscriber Identification Module (SIM) cards and by the migration the post-paid environment;
- The data is seasonal and the pattern repeats roughly every twelve months, although with lower activity every year;
- Aberrant observations can be found mainly in months where client activity tends to fluctuate more, *i.e.* the middle and the end of the year. This behaviour is expected in these months.

Figure 3.3 shows the range, in months, of the client subscription time and the RFM visual analysis over the thirty-month period. The displayed power loss RFM

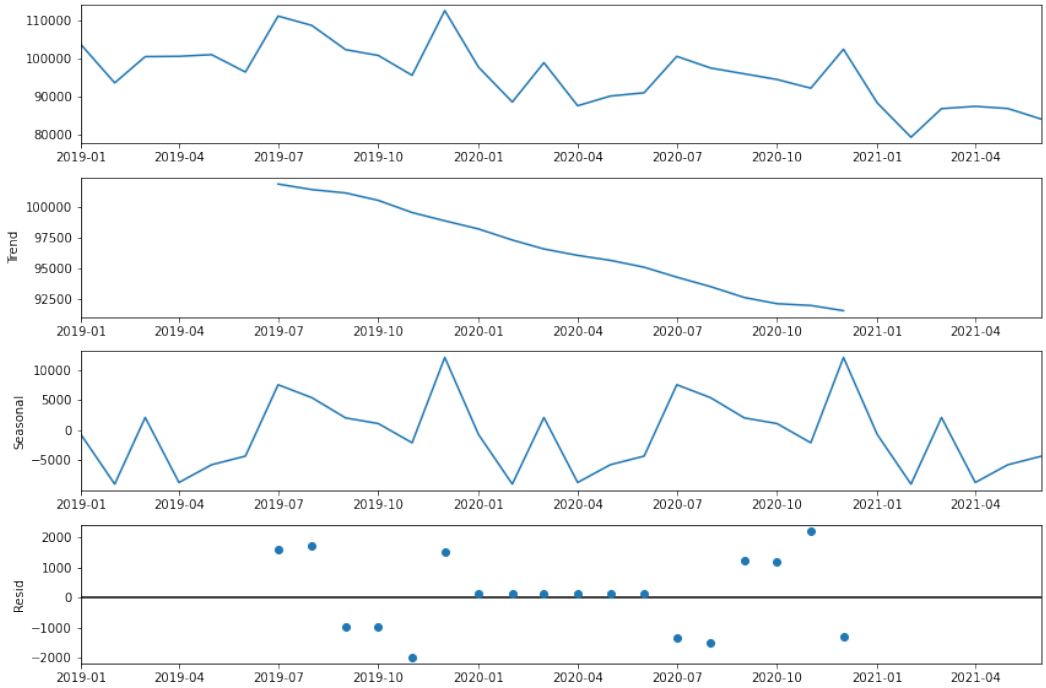


Figure 3.2: Trend, seasonality and aberrant observations analysis

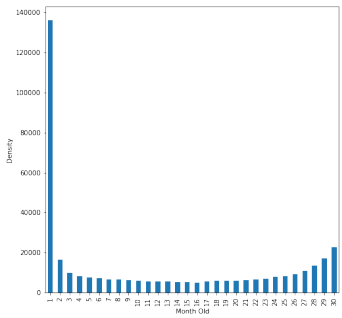
curves indicate that client activity, with the exception of few outliers, is infrequent and top-up values are low. Such evidence is common in pre-paid telecommunication subscriptions.

To have a better understanding of the data and how the clients behave, individual client profiles were built. Additional variables such as frequency, mean and standard deviation, maximum, minimum and total values were calculated from numerical top-up features. These derived features listed in Table 4.4 were calculated incrementally and monthly since the target features, *i.e.*, the monthly frequency and monthly value of top-up are very sporadic. To be able to analyse the potential impact of all features on the dependent variables, the categorical features were converted to numeric features through One-Hot-Encoder (OHE) [84]. Moreover, the corresponding global variables were calculated, taking all clients into account, making use of cumulative calculus. The client profiling provided a better understanding of where clients stand globally, allowing a general client classification.

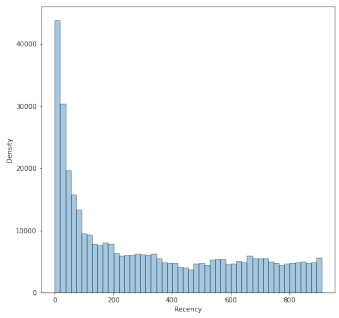
To further enhance the analysis of the clients profiles, they were grouped by their monthly top-up frequency. For this, the global average frequency was calculated as well as the average of the upper and lower halves, resulting in four client categories.

A visual representation of each of the categories, ranked in ascending order from lowest to highest activity, can be found in the Figure 3.4.

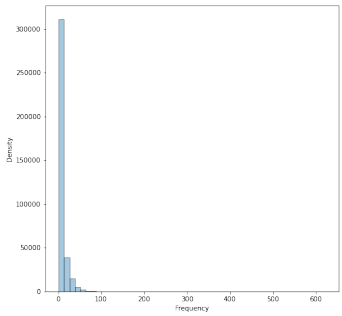
From these results, it is possible to conclude that the majority of the clients presents a top-up frequency below the global average. The number of those above the



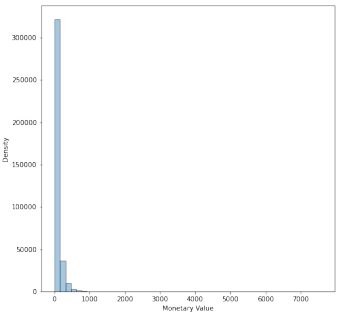
(a) Subscription age



(b) Recency



(c) Frequency



(d) Monetary value

Figure 3.3: Subscription age and RFM analysis top-up

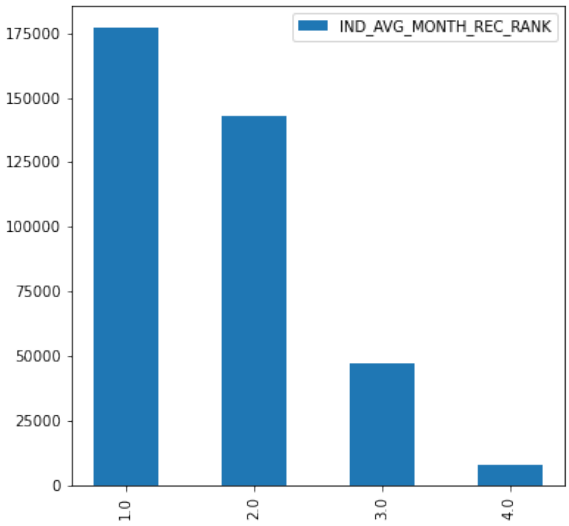


Figure 3.4: Distribution of clients by top-up monthly frequency

average top-up activity is significantly smaller, evidencing two distant extremes. The distribution is somewhat expected in the pre-paid environment since is populated by very different types of clients with a tendency for scarce top-up activity.

Table 3.3: Raw and manufactured features

Feature code	Description
CARD_ID	Client identification
EVENT_DATE	Date of event
IND_TARIFF_CODE	Tariff code
IND_QTD_LAST_REC	Quantity of top-ups on last date
IND_VAL_LAST_REC	Value of top-up on last date
IND_CARD_BALANCE	Card balance after top-up
IND_REC_TYPE_500	Type of top-up I
IND_REC_TYPE_7	Type of top-up F
IND_REC_TYPE_ACW	Type of top-up D
IND_REC_TYPE_AMO	Type of top-up H
IND_REC_TYPE_ATO	Type of top-up C
IND_REC_TYPE_AUZ	Type of top-up A
IND_REC_TYPE_FDI	Type of top-up J
IND_REC_TYPE_O	Type of top-up E
IND_REC_TYPE_OCC	Type of top-up K
IND_REC_TYPE_OYS	Type of top-up B
IND_REC_TYPE_PAU	Type of top-up G
MONTH_OLD	Time of activity
IND_MIN_REC	Minimum of top-ups on a certain date
IND_MAX_REC	Maximum of top-ups on a certain date
IND_TOTAL_REC	Total number of top-ups
IND_AVG_MONTH_REC	Monthly top-up average
IND_STD_MONTH_REC	Monthly top-up standard deviation
IND_MIN_REC_VAL	Minimum value of top-up on a certain date
IND_MAX_REC_VAL	Maximum value of top-up on a certain date
IND_TOTAL_REC_VAL	Total value of top-ups
IND_AVG_MONTH_REC_VAL	Monthly average of top-up value
IND_STD_MONTH_REC_VAL	Monthly standard deviation of top-up value

3.3 Solution Proposal

The prediction of the individual top-up activity involves two tightly related targets: the upcoming top-up date and the top-up value intervals.

Both tasks identified are achieved by implementing a processing pipeline comprising pre-processing, incremental profiling, modelling, evaluation, top-up date and value interval calculus as well as feature selection and window size optimisation.

3.3.1 Pre-Processing

The pre-processing of data, can include several operations so the built models learn better and exhibit improved performance. These operations comprise cleaning, data integration and normalisation, data imputation, noise identification steps, or even data transformations that improve data quality.

The used data set presented some noise as pre-paid and post-paid subscribers were mixed together. To solve this problem, the post-paid clients were identified by their tariff code and removed.

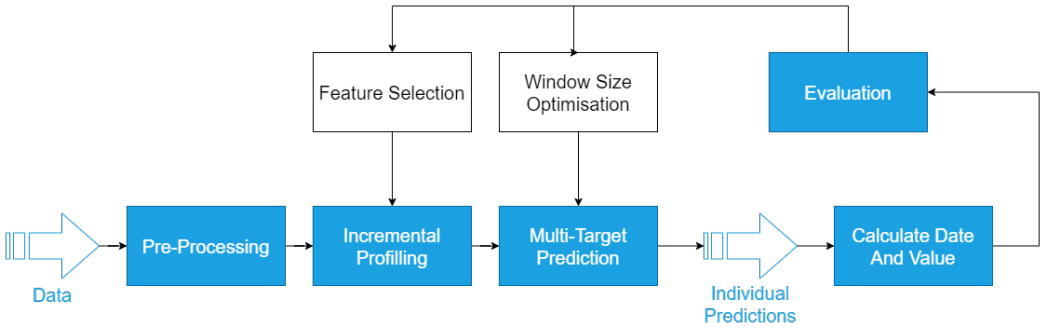


Figure 3.5: Processing pipeline

Additionally, since all features ideally should be numerical for optimised model results, the categorical features were transformed in numerical.

3.3.2 Dimensionality Reduction

As previously mentioned, the refinement of the individual profiles was performed by using the raw numeric features to create new individual features corresponding to monthly average and standard deviation, total, maximum and minimum values. The resulting data set presents the features depicted on the Table 3.3.

Using the total number of features in regression tasks can result in high dimensionality, which encumbers processing. The presence of too many features is a drawback to most inducers, even when these attributes are relevant for the task, not to mention irrelevant or redundant features which can obscure existing patterns [85].

In previous Telco-related works, feature selection techniques are typically categorised as filters, wrappers and embedded approaches [35] [36]. Filters act before and are independent of the learning process; wrappers use the specified learning algorithm to evaluate sub-groups of features; and embedded techniques perform the search as part of the learning process itself. Wrappers methods tend to be more accurate than filters, but also more complex. Embedded methods are less costly than wrappers, but require direct modifications of the learning procedure [86].

To find the best combination of features for the prediction of the individual top-up monthly frequency and monthly value, several feature selection techniques were explored, including wrappers (Forward Selection, Backward Selection, Recursive Feature Elimination and Recursive Feature Elimination Cross Validation), a filter (Univariate Selection) and an embedded approach (Selection using Shrinkage) [87]. A brief description of these methods follows.

- Forward Selection is an iterative method which starts without features. In each iteration, it adds the feature which best improves the model until the addition of a new variable no longer improves the performance of the model.

- Backward Selection starts with all features and removes the least significant feature at each iteration which improves the performance of the model. This is repeated until no improvement is observed with the removal of features.
- Recursive Feature Elimination (RFE) is a greedy optimisation algorithm which aims to find the best performing feature subset. It repeatedly creates models and keeps aside the best or the worst performing feature after each iteration. Then, constructs the next model, using the remaining features, until all features have been eliminated. Finally, the features are ranked based on the order of their elimination.
- Recursive Feature Elimination Cross-Validation (RFECV) ranks features with the help of recursive feature elimination and cross-validated selection of the best number of features. Cross-validation is a technique for evaluating Machine Learning models by training and evaluating several models on subsets of the available input data, using the remaining data subset.
- Univariate Selection selects the best features based on univariate statistical tests, in this case, according to the k highest scores.
- Selection using Shrinkage applies, during the learning process, the least absolute shrinkage and selection operator to choose the features to include based on importance weights and cross-validation [88].

3.3.3 Monthly Sliding Window Regression

Since this is a forecasting problem, the regression technique was chosen to predict individual client *monthly top-up frequency* and the *monthly top-up value*.

The proposed method uses a monthly sliding window (MSW) of size $n+1$ months, where the first n window months are for training and the last window month to test, as proposed by [89]. Specifically, the model is trained using the independent observations of the first $n-1$ months and the target observations of month n and is tested with the $n+1$ month. The window then slides one month and repeats the process till the end of the data set. Considering a data set with m months of data, MSW predicts a total of p months, where $p = m - n$. Figure 3.6 displays the adopted sliding window.

The feature engineering step tested various feature selection techniques along with a single-target regression. Thus, the feature selection techniques select the best group of features for each target based on single-target regression. The predicted values are then compared to the observed ones to determine the best group of features.

To find the best sliding window dimension, a set of single-target experiments was performed using, this time, different sliding window sizes together with the best

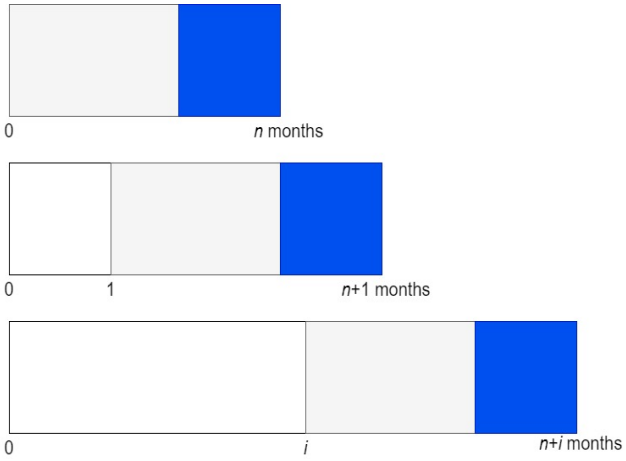


Figure 3.6: Sliding window of size $n + 1$ months

set of features obtained. The sliding window is applied exclusively to the original features, whereas the manufactured features retain the historical perspective.

The MSW is also utilised for the multi-target experiments, with fewer hardware resources to achieve similar results to the single-target experiments.

3.3.4 Event Sliding Window Regression

In case of the online experiments, Multi-Target Regression (MTR) technique was employed due to the fact that it is the most comparable technique to the offline regression experiments.

MTR works with a continuous stream of data $S = (\vec{X}_t, \vec{Y}_t)|t = 1, \ldots, T$ where $T \to \infty$, $\vec{X}_t$ is a feature vector and $\vec{Y}_t$ the corresponding target vector. The objective is to predict the target $\widehat{\vec{Y}}_t$ for an unknown $\vec{X}_t$.

The explored MTR algorithms include AMR by [90], BMTR and MTPR, all implemented in [91].

The MTR-ESW technique implements event-driven incremental test and training. As soon as a new $(\vec{X}_t, \vec{Y}_t)$ data pair becomes available, it is used to incrementally test and, then, update the regression model. This interleaved-test-then-train evaluation, also known as prequential evaluation, is a popular performance evaluation method for the stream setting, where tests are performed on new data before using it to train the model. Considering a data set with e events, MTR-ESW predicts every single event. Figure 3.7, displays the adopted sliding window.

3.3.5 Prediction Interval

The next step of the pipeline, once the regression predictions are made, is to estimate an interval in which a future observation will fall. In other words, estimate the top-up date and values prediction intervals for each client. For this purpose, known as

Figure 3.7: Sliding window of size $n + 1$ events

well as tailor-made methods were developed and implemented.

Date Interval

For the date interval, the already studied method named Bollinger Bands was applied [62]. The individual top-up date interval is based on the predicted average monthly top-up frequency and its standard deviation. For this, a three step process was applied. The Bollinger Bands concept was applied to predict the individual top-up date interval, where the moving average is the predicted individual top-up date, the standard deviation corresponds to the individual standard deviation of the frequency in days and K is set to one. Firstly, the next top-up date is calculated using Equation 3.1, where the number of days until the next predicted top-up is given by dividing the number of days of the prediction month, n, by the predicted frequency, f'. The number of days is then added to the date of the last top-up event, resulting the next predicted top-up date.

$$NextDate = LastTopUpDate + \frac{n}{f'} \tag{3.1}$$

Next, the lower and upper limits of the interval of the next top-up is given by Equation 3.2 and Equation 3.3, where $NextDate$ is given by Equation 3.1, n is the number of days in a month, f' is the predicted monthly top-up frequency and σ_f is the standard deviation of the monthly top-up frequency.

$$MinDate = NextDate - \frac{n}{f' + \sigma_f} \tag{3.2}$$

$$MaxDate = NextDate + \frac{n}{f' + \sigma_f} \tag{3.3}$$

Lastly, the individual date interval, in days, is specified by the difference between the upper and lower band, as stated in Equation 3.4:

$$DateInterval = MaxDate - MinDate \tag{3.4}$$

Thereby, the predicted top-up date interval of a client can be represented as:

$$NextDate - \frac{n}{f' + \sigma_f} \leq NextDate \leq NextDate + \frac{n}{f' + \sigma_f} \tag{3.5}$$

Or:

$$MinDate \leq NextDate \leq MaxDate \tag{3.6}$$

Value Interval

In the case of the value interval, the first approach was to apply the Bollinger Bands method. The predicted value interval of the next top-up is estimated by adding and subtracting the standard deviation to the predicted monthly top-up value. As such, the minimum and maximum values for the interval, in €, are provided by the following Equation 3.7 and Equation 3.8, where v' the predicted monthly top-up value and σ_v is the standard deviation of the monthly top-up value.

$$MinValue = v' - \sigma_v \tag{3.7}$$

$$MaxValue = v' + \sigma_v \tag{3.8}$$

The predicted interval of values for the next top-up can be written as:

$$v' - \sigma_v \leq NextTopUpValue \leq v' + \sigma_v \tag{3.9}$$

Or:

$$MinValue \leq NextTopUpValue \leq MaxValue \tag{3.10}$$

Furthermore, techniques revolving client aggregation were also implemented. In fact, two distinct techniques were experimented: with dynamic and fixed intervals. The dynamic intervals aggregates clients by their monthly top-up value in relation to the global average top-up value. Four categories were created, two above the global monthly average value and two below.

Specifically, three global averages were determined to define the four categories: the global average (μ), the average of all clients above the global average (μ^+), and the average of the clients below the global average (μ^-). The four categories

correspond to the following intervals: (i) $[0, \mu^-[$; (ii) $[\mu^-, \mu[$; (iii) $[\mu, \mu^+[$; and (iv) $[\mu^+, \infty[$.

This method attempts to aggregate by their monthly top-up value and predict the top-up value interval taking into consideration the category of the client. In order to implement this, the global error of each category, *i.e.* the average of the difference between the estimated and the observed values, is added to each individual interval. After this client aggregation into categories, the prediction interval is inspired on the Bollinger Bands method. The method is adapted by adding the global error to the standard deviation of the category. Figure 3.8 displays this logic where μ represents the global average, μ^+ the average of the upper side and μ^- the average of the lower side.

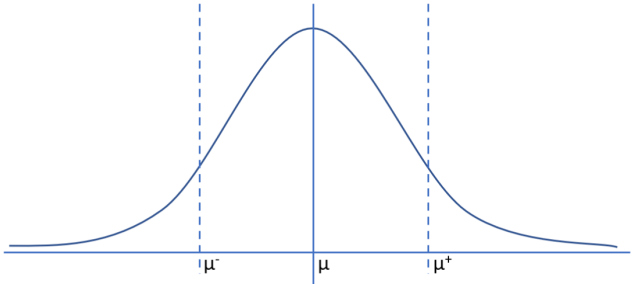


Figure 3.8: Four category method representation

The fixed intervals is based on the lowest top-up value existent, 5 €. In reality, most top-ups are of small amount slightly over the minimum limit. Evidence of this is that, over the course of thirty months, 98.68 % of the top-up events are equal or smaller than 20 € and 86.45 % of equal or under 10 €. Ergo, this method defined three categories corresponding to the intervals $[5; 10]$, $]10; 20]$ and $]20; \infty[$. This evidence is displayed in Figure 3.9 where it is possible to see the percentages of top-up values for these categories, for the whole data set, named in ascending order. Upon estimating the individual category, the global error of the category is added to the individual standard deviation as mentioned.

Lastly, a method combining the predicted monthly top-up value with the last top-up value of each client was explored. This method highlights the contribution of the last payment in relation to the historical payment profile of each client. The top-up value interval $[MinValue, MaxValue]$ is obtained through Equation 3.11 and Equation 3.12, where v' is the prediction value, v_{last} is the value of the last top-up and $\beta \in [0, 1]$ defines the weight given to the last top-up value.

$$MinValue = v' - v_{last} \times \beta \tag{3.11}$$

$$MaxValue = v' + v_{last} \times \beta \tag{3.12}$$

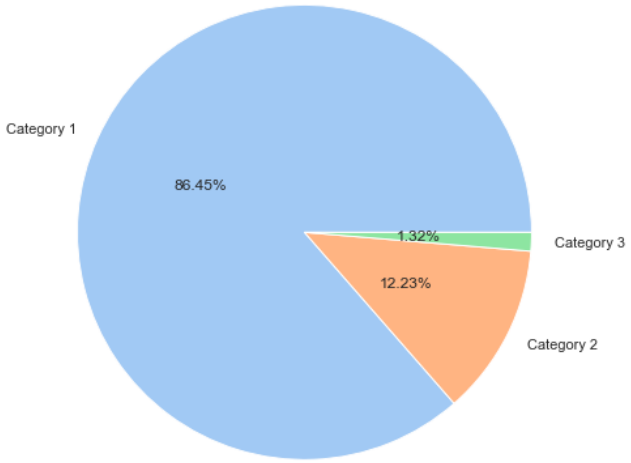


Figure 3.9: Percentage of top-up values for three fixed intervals

3.3.6 Evaluation

The performance of the regression models is measured according to some loss function that evaluates the difference between the observed and predicted values. This way, the evaluation of the predictive models was based on the *Mean Absolute Error* (MAE) and *Root Mean Squared Error* (RMSE) accuracy metrics. These metrics measure the closeness (error) between the predicted target features (dependent variables) and the observed values. Specifically, RMSE determines the standard deviation and MAE the average of these errors within the test partition. On the batch experiments, the calculated error values correspond to the weighted average error of the t tested months, where each weight w_i represents the number of top-up events of that month. Equation 3.13 and Equation 3.14 present the MAE and RMSE of the sliding window regression.

$$MAE = \frac{\sum_{i=1}^{t} w_i MAE_i}{\sum_{i=1}^{t} w_i} \tag{3.13}$$

$$RMSE = \frac{\sum_{i=1}^{t} w_i RMSE_i}{\sum_{i=1}^{t} w_i} \tag{3.14}$$

For the stream experiments, the accuracy metrics are calculated using sliding windows of size e events. The accuracy for these experiments follow the predictive sequential (prequential) evaluation protocol proposed by [92]. In this evaluation protocol, each individual example can be used to test the model before it is used for

training and from this the accuracy can be incrementally updated. When intentionally performed in this order, the model is always being tested on examples it has not seen.

With regard to the prediction intervals, the evaluation is based on the accuracy of the interval. For both cases, it is considered a good prediction if the observed date or value is within the predicted interval. In addition, clients whose activity exceeded and fell short were also evaluated.

Figure 3.10 contains a visual representation of the conditions for predictions whose observed value is within, above and below the estimated interval.

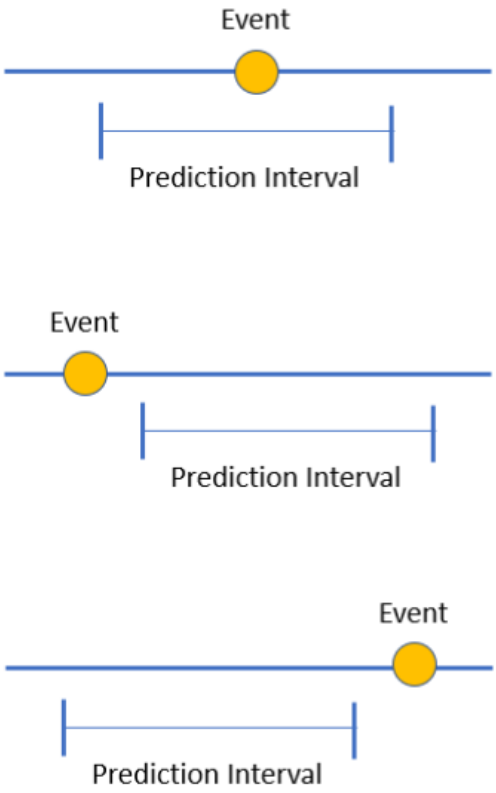


Figure 3.10: Prediction interval accuracy logic

Equation 3.15, shows the calculus for the accuracy of the prediction intervals, where N represents the number of clients and P the percentage of correct predictions for each i category.

$$Accuracy = \frac{\sum_{i=1}^{t} N_i P_i}{\sum_{i=1}^{t} N_i} \tag{3.15}$$

3.4 Tools Used

The visual analysis was performed with the Seaborn library [93]. The offline processing and experiments were implemented using Python with Jupyter Notebooks [94], and the Scikit-Learn library [95]. The online experiments were performed with Massive Online Analysis (MOA) framework. For all parameters which no direct alteration was required were left as the tools default values. The hardware specifications of the computer in which the experiments were executed, to serve as reference, are: 32.00 GB of RAM, Intel Core i7-8750H CPU @ 2.20 GHz processor, 256 GB of physical memory and Microsoft Windows 10 Home operating system.

3.5 Summary

This chapter explores the problem the various steps of the proposed solution. The steps for the proposed solution follow the guidelines for a Data Mining problem just as:

- Business understanding, in presenting the problem statement;
- Data understanding, in studying the data set;
- Data preparation, in the pre-processing and dimensionality reduction steps;
- Modelling, in applying a sliding window for various regression techniques both in offline and online scenarios;
- Evaluation, in comparing the predicted values to the real values with various error metrics.

The deployment stage isn't mentioned as this dissertation sits on a proof-of-concept whether then a deployed implementation.

The following chapter presents the experimental work consequence of applying the aforementioned pipeline to the data.

Chapter 4

Experimental Work

This chapter starts with a brief explanation of some challenges that pre-paid clients reveal which impact the forecasting of their behaviour. Next, the results of various experiments described in Chapter 3 are presented.

4.1 Pre-Processing

The first step upon reading the data was to clean it and transform all the categorical features in numerical features. Features such as the tariff code had to be cleaned since some clients with post-paid tariffs were mixed on the data. Furthermore, the categorical feature, i.e. the type of top-up, was transformed into a numerical feature by making use of OHE, which created a new numerical feature for each previous categorical value. Although this technique can be counter productive if there is a large number of categorical values, this was not the case with this data set as only eleven categorical values were present. This step ensures that all the features correspond to a type which is optimal for the regression technique.

For as much as the raw features are relevant, they can be somewhat limiting if not aided by manufactured features. Manufactured features reveal knowledge that before was hidden. For this reason, statistical features such as the mean, the standard deviation, the maximum, the minimum and the sum were manufactured for the top-up activity and the value of the top-ups, on a monthly basis except for the sum, except for individual accumulator.

4.2 Single-Target Offline Experiments

Bearing in mind that to perform an offline experiments each sample on the data set introduced to the algorithm must be referred to a client's most updated features, the data was prepared by calculating the features for the whole length of the data and then limiting the time windows to the desired interval of months. This method implicates that the manufactured features contain historic data from the beginning of the data set and the raw features make reference only to the time window specified, since these are the ones who provoke changes on the manufactured ones. As such, the resulting data frame has a length equal to the number of clients of the specified window, in which the raw features refer to the clients most recent activity and the manufactured ones present the most updated statistics. In all regression experiments, the data was normalised to improve algorithm performance.

4.2.1 Dimensionality Reduction

Seeing that the process of manufacturing new features leads to an increase in the number of features on the data set which consequently leads to an increase in processing speed and time, dimensionality reduction techniques, in this case feature selection techniques, were utilised to select only the best features. In order to select the best features, a learning technique must be chosen due to the fact that the features are selected according to the learning technique. Since the problem at hand is a forecasting problem and the data presents a linear distribution, confirmed by Figure 4.1, a multiple linear regression was first implemented to test the various feature selection techniques.

Feature Selection

The feature selection step considers the monthly sliding window as described in Chapter 3. At this stage, the maximum window size was considered for training, in this case, thirty months for the reason.

These experiments, additionally to predicting with all the features and the most correlated features, also included techniques such as Univariate Selection, RFE, RFECV, Shrinkage and Forward and Backward Selection. The number of features for the techniques to select was dictated by the number of most correlated features. Initially only the features with a correlation with the target above 0.3 were to be selected. However the number of features which satisfied this condition was low, hence the fifteen most correlated features were selected. As such, it was required of the feature selection techniques, to select the same number of features. The targets, at this point, are the monthly top-up frequency and the monthly top-up value. Table 4.1, shows the best results were obtained with Shrinkage. The complete results with all techniques can be found in Appendix A.

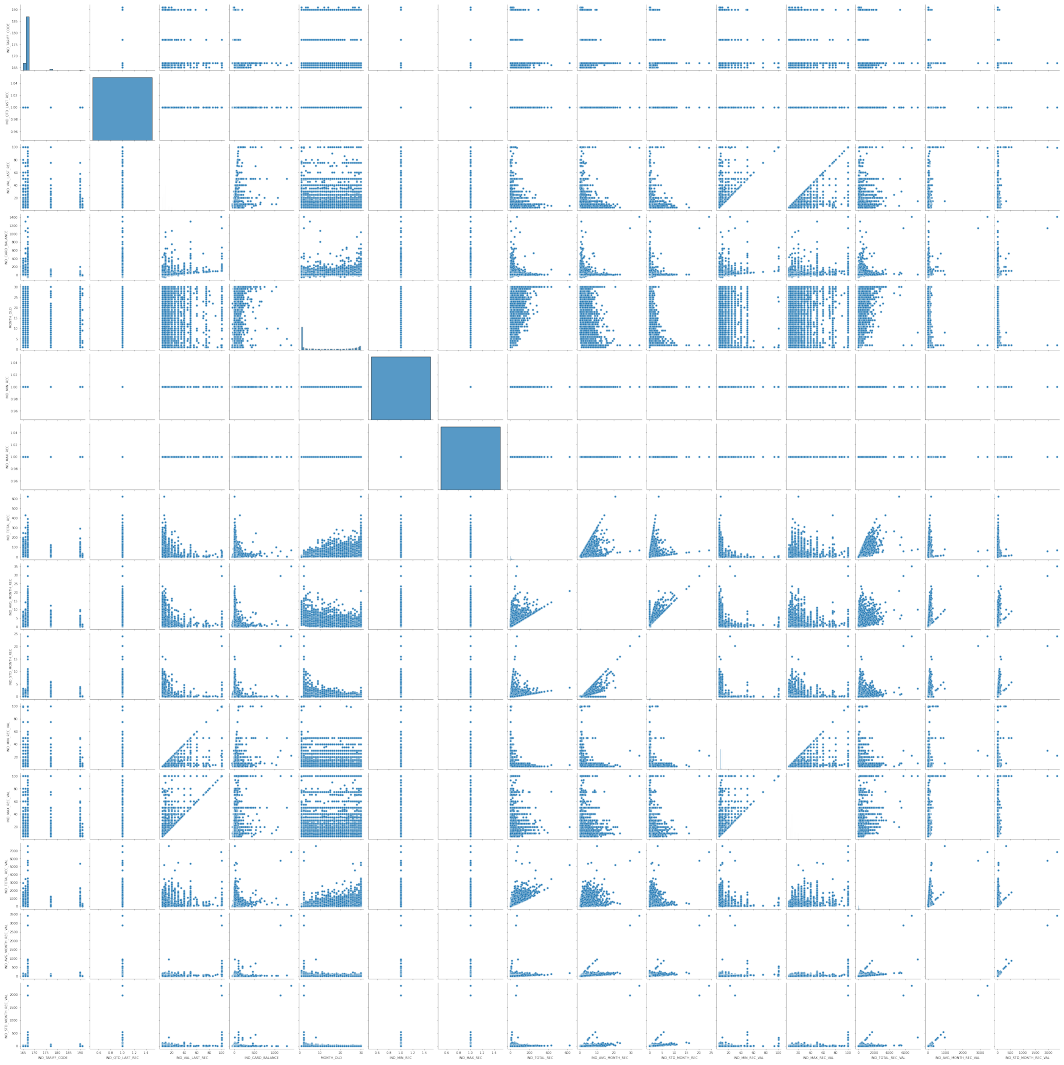


Figure 4.1: Linear relation between features on the data

Table 4.1: Target predictions with a thirty month sliding window

		Monthly Frequency			Monthly Value		
		#	MAE	RMSE	#	MAE	RMSE
Shrinkage	OLS	15	0.00439	0.00766	15	0.00043	0.00080
	DT		0.00060	0.00229		**0.00006**	**0.00029**
	RF		**0.00057**	**0.00218**		**0.00006**	0.00056
	MLP		0.00257	0.00382		0.00053	0.00071

According to Table 4.1, the best technique is Random Forest when the target is the monthly frequency and decision trees when the target is the monthly value. The features selected by this technique are displayed in the Table 4.4.

4.2.2 Optimal Window Size

To determine the best number of months for the sliding window. Experiments with multiple linear regression were carried out for both targets, in which different window

Table 4.2: Selected features by top-up target variable

	Monthly Frequency	Monthly Value
1:	Value of last top-up	Value of last top-up
2:	Time of activity	Time of activity
3:	Number of all top-ups made	Number of all top-ups made
4:	Standard deviation of monthly frequency	Standard deviation of monthly top-up frequency
5:	Minimum value of a top-up	Minimum value of a top-up
6:	Maximum value of a top-up	Maximum value of a top-up
7:	Value of all top-ups made	Value of all top-ups made
8:	Standard deviation of monthly value	Standard deviation of top-up value
9:	Type of top-up D	Type of top-up D
10:	Type of top-up H	Type of top-up H
11:	Type of top-up C	Type of top-up C
12:	Type of top-up B	Type of top-up B
13:	*Type of top-up J*	*Card balance*
14:	*Type of top-up G*	*Tariff code*
15:	*Monthly top-up value average*	*Monthly top-up average*

sizes were considered, ranging from five to thirty months. Table 4.3 presents the best window size of thirty months. The full list of results can be found in Appendix B.

Table 4.3: Offline regression: best MLR-MSW results

		Monthly Frequency			Monthly Value		
		MAE	**RMSE**	**Time** (s)	**MAE**	**RMSE**	**Time** (s)
30	MOLS	0.004 39	0.007 66	0.534 11	0.000 43	0.000 80	0.188 08
	MLP	0.001 44	0.003 04	14.878 05	0.000 36	0.000 72	15.334 16
	DT	0.000 60	0.002 32	1.575 75	**0.000 06**	**0.000 29**	2.445 72
	RF	**0.000 57**	**0.002 17**	94.930 22	0.000 06	0.000 59	148.701 22

4.3 Multi-Target Offline Experiments

Although the aforementioned experiments exhibit low error metrics they require the execution of two regression tasks to predict the desired targets. For this reason, a multi-target approach is a natural follow up.

Since this approach is able to predict both targets with one task, it is imperative to select the most promising group of features. The previously selected features for the single-target tasks were united, resulting on the set of features presented in Table 4.4.

Taking into consideration the better intrinsic characteristics of the MTR compared to the STR, the process of finding the ideal time window in months was performed again. This step was implemented not only to verify if applying the group of features actually presents a favourable approach, but also to verify the behaviour of the algorithm, when it comes to optimal window size, in comparison to the previously tested technique. Therefore, experiments to find the ideal window size were repeated. The best results are shown in Table 4.5. The full list of results can be found in Appendix A.

Table 4.4: Regression features and targets

		Features
Raw	1:	Value of last top-up
	2:	Time of activity
	3:	Tariff code
	4:	Card balance
	5:	Type of top-up D
	6:	Type of top-up B
	7:	Type of top-up C
	8:	Type of top-up J
	9:	Type of top-up G
	10:	Type of top-up H
Mnf.	11:	Maximum value of a top-up
	12:	Minimum value of a top-up
	13:	Standard deviation of monthly value
	14:	Standard deviation of monthly frequency
	15:	Number of all top-ups made
	16:	Value of all top-up made
		Targets
Mnf.	1:	Monthly value
	2:	Monthly frequency

Table 4.5: Offline multi-target regression: MTR-MSW results

Window (month)	**Technique**	**MAE**	**RMSE**	**Time** (s)
30	MOLS	0.003 92	0.006 57	0.283 64
	MLP	0.001 37	0.002 20	19.696 69
	DT	**0.000 30**	**0.001 35**	**2.286 20**
	RF	0.000 31	0.001 37	158.537 85

As expected, the thirty month window presents the lower error. Figure 4.2 displays the plot of the trend and the relation between the predicted and observed values, displaying a linear relation.

A quick verification of the results confirms the before established premise that larger time windows predict more accurately. This conclusion cements the theory that MTR is preferable to multiple STR, not only because of the matching time windows and similar prediction error.

Until now all experiments are based on offline learning. Chapter 2, however reveals that online learning presents several advantages to offline learning. As such, the following section focuses on predicting with online learning techniques.

4.4 Multi-Target Online Experiments

As aforementioned, the online experiments use an event sliding window. The experiments comprise several techniques for multi-target regression through the MOA GUI.

Taking into account the fact that online learning represents an event based approach, experiments require an initial data preparation. As such, firstly, the manufactured features are calculated incrementally event-wise and the whole stream of

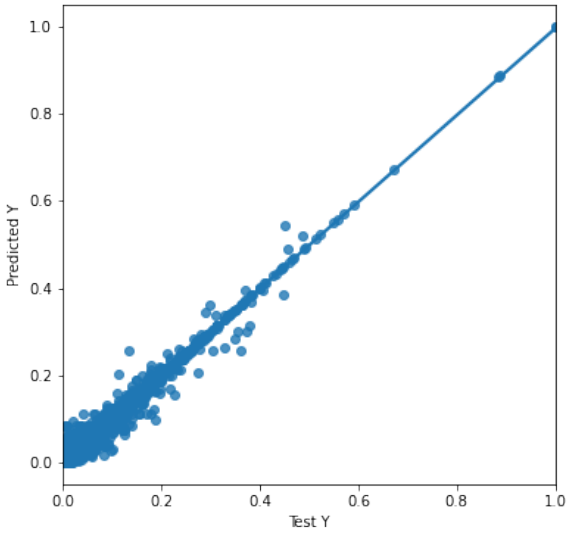


Figure 4.2: Trend and relation between predicted and observed values for the DT with a thirty month window size

events is used in the regression task. Secondly, the previously chosen optimal group of features is normalised, concluding the data preparation step.

As referred in Chapter 3, seasonal effects, such as the middle of the year and the end of the year, associated with summer and Christmas holidays respectively, have a great impact in client top-up activity. Client activity is seasonal and the data set provided is somewhat small for predicting individual behaviour, specially since there are many novel clients whose activity is reduced. This particular behaviour hinders the task of forecasting as it causes big deviation in the estimations.

The experiments for the online scenario with the optimal group of features previously selected were performed with the intent to find the optimal technique and event window size. Since the average number of events per month is approximately 100 000 events, the experiments were performed with this increment until the end of the data set. The complete list of results of these experiments can be found in Appendix C. Table 4.6, summarises the best results in which the AMR technique displays the lowest error with an event window size of just 500 000 events.

Table 4.6: Online multi-target regression: MTR-ESW results

Window (events)	**Technique**	**MAE**	**RMSE**	**Time** (s)
500 000	BMTR	0.001 08	0.004 11	260.312 50
	MTPR	0.001 41	0.004 22	27.250 00
	iSOUP	0.007 89	0.021 93	46.187 50
	AMR	**0.001 14**	**0.004 15**	140.718 75

In the offline experiments, the optimal window size was thirty months and the run-time was 2.286 20 seconds. This requires larger memory and processing requirements, making the online scenario stand out.

Additionally, it is expected that the online error improves throughout time since the model is incrementally trained and tested each new event. Figure 4.3 illustrates the fluctuations of the RMSE with the 500 000 event window.

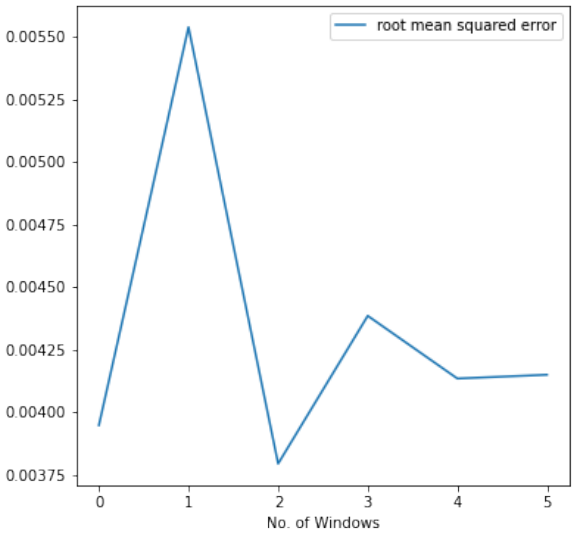


Figure 4.3: Variation of the RMSE

Finally, Figure 4.4 presents the predicted and real values. In this case, the trend line presents a bigger deviation compared to the offline scenario. This is expected since stream models start from scratch, *i.e.*, with a void model, in which the prequential evaluation processes every event and compares the predicted against the observed values.

All in all, this technique offers very good predictions. The prediction error may not be as low but the number of events utilised in order to obtain the predictions is roughly six times smaller, *i.e.*, has lower hardware requirements and has a tendency to adapt to change far superior than the offline techniques studied. For these reasons, this technique was chosen for the calculus of the prediction of the individual top-up date and monetary value intervals.

4.5 Prediction Interval

In order to properly implement the logic perpetrated in the Bollinger Bands method, it is important that the data presents a distribution close to a normal distribution.

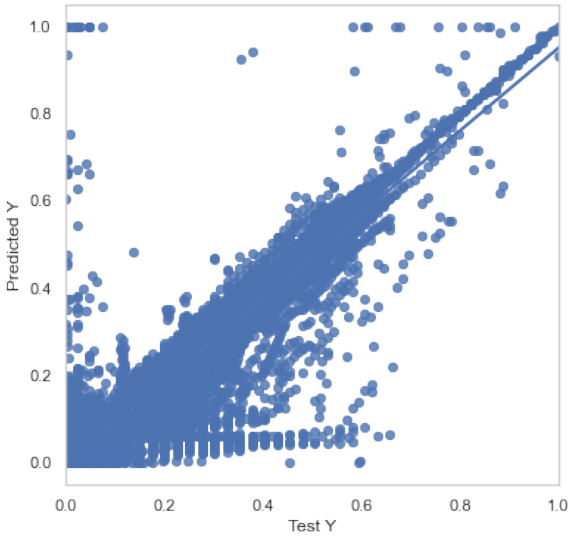
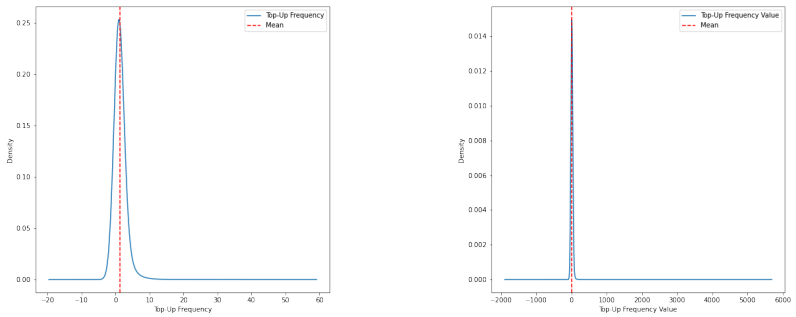


Figure 4.4: Trend and relation between predicted and observed values for the AMR with a 500 000 event window size

As such, before the implementation of the method, an analysis of the distribution of the targets was performed. In Figure 4.5, it is possible to verify that both targets present a leptokurtic distribution dictated by the sharp peaks and a positive skewness by the thick right tail. In addition, the values for the second moment, which corresponds to the variance, are 0.52 and 117.76 for the frequency and the value, respectively, which emphasise the many types of clients.



(a) Top-up frequency distribution

(b) Top-up frequency value distribution

Figure 4.5: Distribution of the top-up frequency and the top-up frequency value

With this assurance, the predicted top-up date and top-up value intervals were calculated according to the method aforementioned.

4.5.1 Date Interval

In the case of the date interval, the method formulas were applied to the prediction month of June 2021. On the first attempt, no multiplicative factor K was applied to obtain the results with the least possible deviation. Additionally, and to properly assess the results, given that clients may make more than one top-up per month, the experiment considered the first top-up made by the client, in the prediction month.

In theory, the more data the algorithm has about each client, *i.e.*, the more active a client is, the easier it should be to predict a correct interval. Figure 4.6 displays the relation between the number of top-ups and the size of the prediction interval in days, where it is possible to conclude that more client activity leads to more accurate estimations.

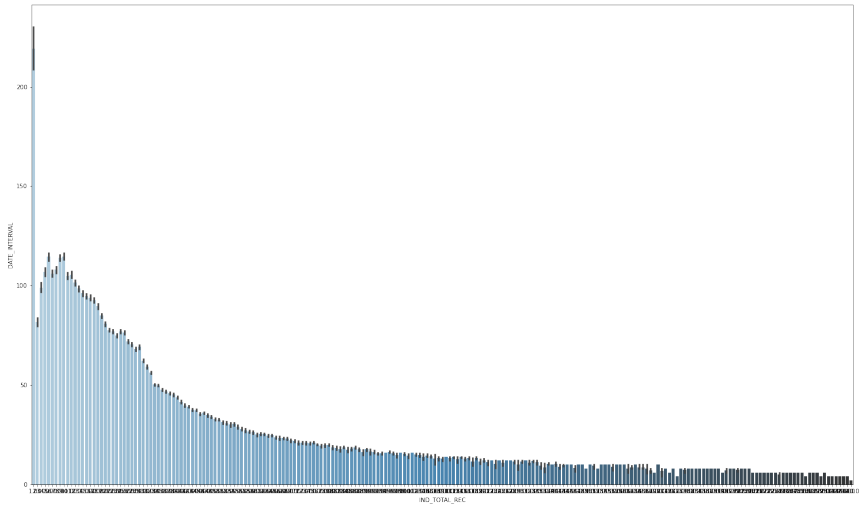


Figure 4.6: Relation between the total number of top-ups and the prediction interval

Due to the fact that there is a large variety of clients profiles on the data set, and since the intervals are tailored to each client, there is a considerable variety of interval lengths. In fact, the prediction interval has a mean of 90 days and a standard deviation of 78 days, with a minimum interval of 2 days and a maximum of 900 days.

Upon this analysis, the accuracy of the predictions was calculated as the number of observed values within the prediction interval for every client. Table 4.7 shows the accuracy results for the predicted month.

Results show an accuracy of 80.68 % for observed values within the prediction interval. Most of the clients whose activity is outside the predicted interval have later

Table 4.7: Date interval prediction for June 2021 with Bollinger Bands

Date interval prediction results				
Window	**Right**	**Wrong**	**Total**	**Accuracy (%)**
Interval	48047	11505	59552	80.68
Above	10399	-	-	17.46
Below	1106	-	-	1.56

observation than the maximum predicted date. This scenario can be problematic since it can indicate a client is less active. For occurrences of this type the telco operator should be alerted to take the necessary measures (tailor-made campaigns).

All in all, since this result was deemed satisfactory. No further K values were tested since they would introduce bigger deviations to the intervals, decreasing precision.

4.5.2 Value Interval

In the case of the value interval, there are a few annotations to be made:

- To have a congruent analysis, only the clients whose top-up date interval was correct were utilised on the experiments for the value interval (48 047 clients);
- There is a minimum top-up value of 5 €, so this value was used as the lower limit;
- While analysing the accuracy of the intervals, predictions under the observed value were deemed, from the perspective of the telco operator, less important than those above since the later are actually beneficial.

To determine the best method for the value interval several experiments were performed from applying the Bollinger Bands, client aggregation to a combination of the individual predicted average with the last top-up.

Table 4.8 displays the results of the Bollinger Bands method without the multiplicative factor.

Table 4.8: Value interval prediction for June 2021 with Bollinger Bands

Value interval prediction results				
Window	**Right**	**Wrong**	**Total**	**Accuracy (%)**
Interval	10474	37573	48047	21.80
Above	29331	-	-	61.05
Below	8242	-	-	17.15

Due to the fact that the target is the individual monthly average, this method falls short when the client is inactive for a few months between top-ups, in such cases, the predicted value can be significantly lower than the observed top-up value, presenting low accuracy of 21.80 %. The fact that the majority of the clients observed

value is no cause of concern for the Telecom, on the contrary. However, these results fall short for the objective. Hence, experiments which aggregate clients according to their monthly expenditure were made.

The first attempt to aggregate clients was executed taking the global average monthly value and subdividing the upper and lower parts of the global value, creating four client categories. This led to the distribution displayed in Figure 4.7, where the first three categories contains around the same number of clients and the last significantly less. This is expected since clients who fall in this category are scarce.

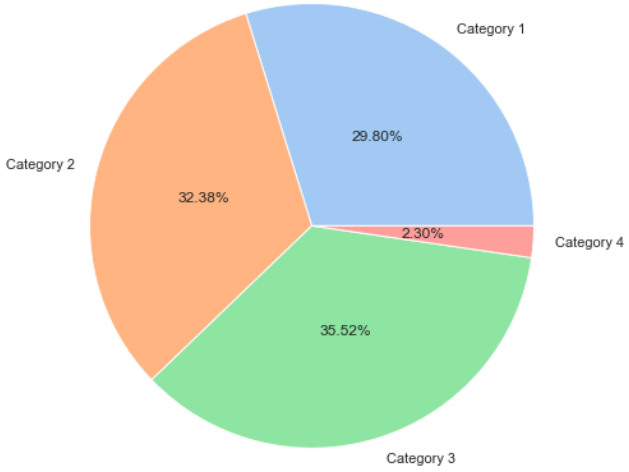


Figure 4.7: Distribution of monthly value for the prediction month in dynamic intervals

As such, Table 4.9 presents the results for this experiment.

Table 4.9: Value prediction with four category client aggregation

		Right	Wrong	Total	Accuracy (%)
Category 1	**Interval**	6785	7533	14318	47.39
	Above	7518	-		52.51
	Below	15	-		0.10
Category 2	**Interval**	7649	7910	15559	49.16
	Above	7870	-		50.58
	Below	40	-		0.26
Category 3	**Interval**	11620	5445	17065	68.09
	Above	2691	-		15.77
	Below	2754	-		16.14
Category 4	**Interval**	233	872	1105	21.09
	Above	26	-		2.35
	Below	846	-		76.56

The majority of the clients is below the global average and it is difficult for accuracy varies throughout the categories. The accuracy of first two categories is reasonable and the observed values above the prediction interval as well. The third category has high accuracy for the prediction interval and the observed values are

equally distributed above and below the interval. The fourth category displays a low accuracy and most of the observed values are below the forecasted interval. Overall, the weighted accuracy of this method is 54.73 %. Moreover, it will be slow to adapt to changes in client activity.

The following category experiment took into consideration fixed monetary intervals based on the minimum top-up value. Clients are then attributed to a category based on their observed monthly value which can be $[5; 10]$, $[10; 20]$ and $[20; \infty]$. For the prediction month, the distribution of client's average value is presented in Figure 4.8. The vast majority of clients is located in the first category and the second category is roughly four times bigger than the third.

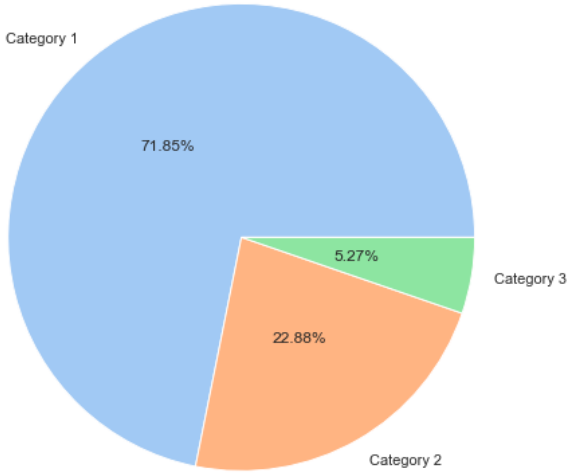


Figure 4.8: Distribution of monthly value for the prediction month in fixed monetary intervals

Table 4.10 shows the results from this experiment.

Table 4.10: Value prediction with 3 category client aggregation

		Right	Wrong	Total	Accuracy (%)
Category 1	**Interval**	17429	17091	34520	50.49
	Above	16926	-		49.03
	Below	165	-		0.48
Category 2	**Interval**	7753	3241	10994	70.52
	Above	1502	-		13.66
	Below	1739	-		15.82
Category 3	**Interval**	658	1875	2533	25.98
	Above	90	-		3.55
	Below	1785	-		70.47

In this case, the vast majority of clients is located in the first category. The weighted accuracy of this experiment is 53.78 %, which is slightly lower than the

previous method. This method presents an even bigger challenge to client activity adaptation.

These experiments lead to the conclusion that client aggregation is tricky since the majority of clients displays scarce activity, concentrating in certain categories.

Lastly, as an attempt to mitigate the disadvantages of the aforementioned methods, the last experiment makes use of the individual predicted monthly value and the last top-up value. The accuracy is expected to be higher due to the fact that it takes into consideration the recent past (value of the last top-up) which can be beneficial since most clients usually top-up the same amount. Table 4.11 presents the results of the several experiments with different values of β, the weight applied to the last top-up value.

Table 4.11: Top-up value interval prediction with last top-up value

Date interval prediction results					
β	**Window**	**Right**	**Wrong**	**Total**	**Accuracy (%)**
1/3	**Interval**	24089	23958	48047	50.14
	Above	18937	-		39.41
	Below	5021	-		10.45
1/2	**Interval**	29316	18731		61.02
	Above	13872	-		28.87
	Below	4859	-		10.11
2/3	**Interval**	33792	14255		70.33
	Above	11679	-		24.31
	Below	2576	-		5.36

As anticipated, this method offers better accuracy than the aggregation methods. In fact, by applying just the weight of 1/3 to the client's last top-up value, a similar accuracy is given. If, for instance, the top-up values of a client are irregular, that is, they change significantly from top-up to top-up, a higher weight should be placed on the last top-up value with the intent of mitigating this variance. For cases such as this, the weights of 1/2 and 2/3 of the last top-up were tested which produced an accuracy of 61.02 % and 70.33 %. The best accuracy was achieved with a $\beta = 2/3$, which highlights considerably the contribution of the last top-up value.

4.6 Summary

To sum up, this Chapter showed that:

- Single-target feature selection and optimal window size experiments provide good insights and benchmark values for multi-target offline experiments;
- Multi-target offline experiments are able to, in one task, combine the features which resulted from single-target feature selection and produce better results for the same window size;

- Multi-target online experiments have far less hardware requirements, are capable of learning and improving throughout the window of learning and still produce approximate error metrics results to the offline scenario;
- Bollinger Bands method is suitable for date interval but falls short for value interval due to clients with scarce top-up activity;
- For the case of the value interval, the combination of the recent past (last top-up value) with the average predicted value presented the best results

Chapter 5

Final Considerations

This chapter presents the balance of this work against the initial objectives. Some opportunities for improvement and optimisation to be taken into account in the future are also identified.

5.1 Conclusions

The world of telecommunications has always had the goal of bringing people together and it has always been demanded of it to have the most recent technology and intelligence. More recently, the advent of telecommunication systems which acquire data from every day activity has led to the exponential growth to process such data and extract good insights from it. Despite the vast literature in Data Mining applications for the telecommunications industry when it comes to the churn use case or binary top-up, for the use case of predicting individual top-up date and value nothing was found. Thus, exploring this uncharted area demands plenty of literature review in similar industries as well.

Taking this into account, this dissertation aimed to study an approach that predicts individual top-up date and value, also addressing different learning techniques. The ideal method should achieve a small and accurate prediction interval in which only the important features, that is, the relevant features for the algorithm, and present a low error metric when comparing predicted values to observed values. The extensive literature review for the most updated use cases made it possible to

acknowledge the scarcity of applications for the individual top-up prediction. Consequently, the emergence of techniques to achieve the outcome proves to have a great potential and plenty of room to grow. That being the case, a creative approach to the problem was undergone making use of novelty statistical and Machine Learning techniques.

Firstly an understanding of the data was undergone from which conclusions such as the following were derived: the fact that client activity is scarce and in most cases infrequent and with small amounts per top-up; client subscription age varies a lot being the majority of the clients active for a small window of time; the global trend is of a decrease of activity throughout time with a seasonal factor of around twelve months. To prepare the data, the data cleaning, data transformation and noise identification processes were implemented. After this, a feature generation and feature selection procedures were executed to a data set with only the most important features for the prediction of the targets. Upon having the data properly analysed and with the correct structure various modelling techniques were experimented in the offline and the online scenarios for predicting the individual monthly frequency and monthly value. In both scenarios, data normalisation was performed to optimise the regression algorithms. The predictions of the modelling technique which presented the most benefits were then utilised to calculate the prediction interval for the next top-up date and its value. The date interval estimation can be satisfied by the Bollinger Bands method however the same doesn't occur for the value interval for months absent of activity make a big impact on the monthly value. The value interval, due to big discrepancy in top-up amounts between clients undergone a study of client aggregation which culminated on the knowledge that although aggregation techniques possesses far better results than the previous study method also has the limitations of adapting to clients movement between categories and still not producing optimal accuracy. As such, because clients top-up amounts don't vary much in value, an experiment which adds to the predicted monthly value a fraction of the last top-up was done and ended up mitigating the aggregation methods limitation of properly predicting when clients monthly value varies, because it calculates individually and at the same time providing better accuracy.

To sum up, the work carried out in this dissertation provides the company a thorough literature review and study of several forecasting techniques as well as a practical implementation of a methodology which offers an accuracy of around 80 % and 70 % for the use case of date and value interval, respectively.

5.2 Future Work

All the work is incomplete if what is aimed for is perfection. With this motto, a chapter is introduced where the intention is to highlight the work that could not be

embraced during the project period, but which is crucial in order to continue the work developed.

For the case of the top-up value, perform a study in applying an average of the value of the last X amount of top-ups to mitigate possible deviation present on the proposed method when the last top-up is high or low for example.

Not having access to the traffic data, in the future it is suggested to utilise both the top-up and traffic data to improve predictions since on the pre-paid environment top-up activity tends to be made according to traffic generated.

References

[1] Terranova, "Altice labs celebra aniversário com apresentação de novos projetos. | terranova." https://www.terranova.pt/noticia/sociedade/altice-labs-celebra-aniversario-com-apresentacao-de-novos-projetos, 2020. (Accessed on 02/11/2021). [Cited on pages ix and 2]

[2] Altice Labs, "Welcome Abroad," 2021. (Accessed on 02/11/2021). [Cited on pages ix and 3]

[3] Altice Labs, "Altice labs | sobre nós." https://www.alticelabs.com/pt/sobre.html, 2021. (Accessed on 02/11/2021). [Cited on pages ix, 2, and 3]

[4] Statistica, "• global big data market size 2011-2027 | statista." https://www.statista.com/statistics/254266/global-big-data-market-forecast/, 2018. (Accessed on 10/05/2021). [Cited on pages ix and 8]

[5] J. Gama, A. C. Lorena, K. Faceli, M. Oliveira, and A. P. de Leon Carvalho, *Extração de Conhecimento de Dados*. Edições Sílabo, 09-2017 ed., 2017. [Cited on pages ix, 9, 13, and 14]

[6] A. GeekStyle, "Business intelligence and its relationship with the big data, data analytics and data science." https://www.linkedin.com/pulse/business-intelligence-its-relationship-big-data-geekstyle/, 2 2017. (Accessed on 07/01/2021). [Cited on pages ix and 10]

[7] F. Provost and T. Fawcett, "Data science for business: What you need to know about data mining and data-analytic thinking." file:///C:/Users/35191/Desktop/Data_Science_for_Business.pdf, 2013. (Accessed on 11/17/2020). [Cited on pages ix, 10, 11, 12, and 13]

[8] S. García, J. Luengo, and F. Herrera, "Data preprocessing in data mining.pdf." https://pzs.dstu.dp.ua/DataMining/preprocessing/bibl/Data%20Preprocessing%20in%20Data%20Mining.pdf, 2015. (Accessed on 11/28/2020). [Cited on pages ix, 17, and 19]

[9] E. Bisong, *Building Machine Learning and Deep Learning Models on Google Cloud Platform: A Comprehensive Guide for Beginners*. Apress, 1st ed., 2019. [Cited on pages ix and 21]

[10] D. Patil and V. Wadhai, "Adaptive real time data mining methodology for wireless body area network based healthcare applications," *Advanced Computing: An International Journal*, vol. 3, 07 2012. [Cited on pages ix and 23]

[11] Sigmundo Preissler Jr, "Seasonality in python: additive or multiplicative model? | by sigmundo preissler jr, phd | medium." `https://sigmundojr.medium.com/seasonality-in-python-additive-or-multiplicative-model-d4b9cf1f48a7`, 2018. (Accessed on 10/03/2021). [Cited on pages ix and 26]

[12] B. Piccart, *Algorithms for Multi-Target Learning (Algoritmes voor het leren van multi-target modellen)*. PhD thesis, Katholieke Universiteit Leuven, Belgium, 2012. [Cited on pages ix and 33]

[13] R. Sousa and J. Gama, "Online semi-supervised learning for multi-target regression in data streams using amrules," 10 2016. [Cited on pages ix, 35, and 36]

[14] Investopedia Team, "The basics of bollinger bands." `https://www.investopedia.com/articles/technical/102201.asp`, 2020. (Accessed on 07/23/2021). [Cited on pages ix and 38]

[15] J. Gama, *Knowledge Discovery from Data Streams*. Data Mining and Knowledge Discovery, Chapman and Hall/CRC, 1st ed., 2010. [Cited on pages xi, 20, 21, 22, and 23]

[16] D. L. Olson and D. Wu, *Predictive Data Mining Models [2nd ed.]*. Springer, 2020. [Cited on pages xi, 27, 28, and 29]

[17] Altice Europe, "Altice europe |." `https://altice.net/`, 2021. (Accessed on 02/11/2021). [Cited on page 1]

[18] T. Davenport, "Big data at work: Dispelling the myths, uncovering the opportunities." Harvard Business Review Press, 2014. (Accessed on 03/08/2021). [Cited on page 2]

[19] W. Chung, H. Chen, and J. F. N. Jr., "A visual framework for knowledge discovery on the web: An empirical study of business intelligence exploration," *Journal of Management Information Systems*, vol. 21, no. 4, pp. 57–84, 2005. [Cited on page 4]

[20] P. Alves, R. Filipe, and B. Malheiro, *Towards Top-Up Prediction on Telco Operators*, pp. 573–583. 09 2021. [Cited on page 6]

[21] D. Kanevsky, S. H. Maes, and J. S. Sorensen, "Conversational Data Mining," *International Business Machines Corporation*, vol. 1, no. 12/2003, 2003. [Cited on page 8]

[22] Z. Constantinescu, M. C., and M. Vladoiu, "Driving style analysis using data mining techniques," *International Journal of Computers, Communications and Control*, vol. 5, no. 5, pp. 654–663, 2010. [Cited on page 8]

[23] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Y. Chen, T. Lillicrap, F. Hui, L. Sifre, G. Van Den Driessche, T. Graepel, and D. Hassabis, "Mastering the game of Go without human knowledge," *Nature*, vol. 550, no. 7676, pp. 354–359, 2017. [Cited on page 8]

[24] M. V. Joseph, "Data mining and business intelligence applications in telecommunication industry." `https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.679.3160&rep=rep1&type=pdf`, 2013. (Accessed on 11/17/2020). [Cited on page 8]

[25] M. Nadaf and V. Kadam, "Data mining in telecommunication: Mohsin nadaf & vidya kadam | data mining | market segmentation." `https://www.scribd.com/document/413045590/Data-Mining-in-Telecommunication-India`, 2013. (Accessed on 12/16/2020). [Cited on page 8]

[26] P. V. Klaine, M. A. Imran, O. Onireti, and R. D. Souza, "A survey of machine learning techniques applied to self-organizing cellular networks," *IEEE Communications Surveys Tutorials*, vol. 19, no. 4, pp. 2392–2431, 2017. [Cited on page 10]

[27] C. McCue, *Data Mining and Predictive Analysis*. Butterworth-Heinemann, 1st ed., 2007. [Cited on pages 11 and 12]

[28] P. Chapman, J. Clinton, R. Kerber, T. Khabaza, T. Reinartz, C. Shearer, and R. Wirth, "Crisp-dm 1.0." `https://the-modeling-agency.com/crisp-dm.pdf`, 2000. (Accessed on 11/21/2020). [Cited on pages 11 and 12]

[29] R. Wirth and J. Hipp, "Crisp-dm: Towards a standard process model for data mining." `https://www.cs.unibo.it/~danilo.montesi/CBD/Beatriz/10.1.1.198.5133.pdf`, 2000. (Accessed on 11/21/2020). [Cited on page 12]

[30] A. Azevedo and M. Santos, "Kdd, semma and crisp-dm: a parallel overview." `https://recipp.ipp.pt/bitstream/10400.22/136/3/KDD-CRISP-SEMMA.pdf`, 2008. (Accessed on 11/21/2020). [Cited on page 13]

[31] D. T. Larose and C. D. Larose, *Discovering Knowledge in Data An Introduction to Data Mining*. Wiley, 2nd ed., 2014. (Accessed on 11/22/2020). [Cited on page 14]

[32] M. Bramer, *Principles of Data Mining*. Springer, 2nd ed., 2013. (Accessed on 11/23/2020). [Cited on page 14]

[33] J. Han, M. Kamber, and J. Pei, *Data Mining Concepts and Techniques (The Morgan Kaufmann Series in Data Management Systems)*. Morgan Kaufmann, 3rd ed., 2011. (Accessed on 11/30/2020). [Cited on page 17]

[34] E. Alpaydin, *Introduction to Machine Learning*. The MIT Press, 2nd ed., 2010. (Accessed on 11/30/2020). [Cited on page 18]

[35] V. E and D. P. Ravikumar, "Attribute selection for telecommunication churn prediction," *International Journal of Engineering & Technology*, vol. 7, no. 4.39, pp. 506–509, 2018. [Cited on pages 19 and 51]

[36] Y. Yulianti and A. Saifudin, "Sequential feature selection in customer churn prediction based on naive bayes," *IOP Conference Series: Materials Science and Engineering*, vol. 879, p. 012090, 2020. [Cited on pages 19 and 51]

[37] Z. Elhamraoui, "Batch and online learning." `https://medium.com/analytics-vidhya/batch-and-online-learning-bcb416fa898c`, 2020. (Accessed on 03/10/2021). [Cited on page 20]

[38] D. Brain and G. I. Webb, "The need for low bias algorithms in classification learning from large data sets," in *Principles of Data Mining and Knowledge Discovery* (T. Elomaa, H. Mannila, and H. Toivonen, eds.), (Berlin, Heidelberg), pp. 62–73, Springer Berlin Heidelberg, 2002. [Cited on page 21]

[39] B. Babcock, S. Babu, M. Datar, R. Motwani, and J. Widom, "Models and issues in data stream systems," in *Proceedings of the ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, pp. 1–16, 06 2002. [Cited on pages 21 and 22]

[40] M. Garofalakis, J. Gehrke, and R. Rastogi, *Data Stream Management: Processing High-Speed Data Streams*. Data-Centric Systems and Applications, Springer-Verlag Berlin Heidelberg, 1st ed., 2016. [Cited on pages 22 and 23]

[41] Australian Bureau of Statistics, "Time series analysis: The basics." `https://www.abs.gov.au/websitedbs/d3310114.nsf/home/time+series+analysis:+the+basics`, 2021. (Accessed on 06/18/2021). [Cited on pages 24, 25, and 26]

[42] R. J. Hyndman and G. Athanasopoulos, *Forecasting: Principles and Practice*. OTexts, 2nd ed., 2018. [Cited on pages 25 and 26]

[43] P. H. Franses, D. van Dijk, and A. Opschoor, *Time Series Models for Business and Economic Forecasting*. CUP, 2nd ed., 2014. [Cited on page 27]

[44] H. Borchani, G. Varando, C. Bielza, and P. Larrañaga, "A survey on multi-output regression," *WIREs Data Mining and Knowledge Discovery*, vol. 5, no. 5, pp. 216–233, 2015. [Cited on page 28]

[45] J. F. Trevor Hastie, Robert Tibshirani, *The elements of statistical learning: Data mining, inference, and prediction.* Springer Series in Statistics, Springer, 2nd ed., 2009. [Cited on page 28]

[46] N. Singh, "Advantages and disadvantages of linear regression." `https://iq.opengenus.org/advantages-and-disadvantages-of-linear-regression/`, 2021. (Accessed on 03/21/2021). [Cited on page 29]

[47] R. Christensen, *Advanced Linear Modeling: Multivariate, Time Series, and Spatial Data; Nonparametric Regression and Response Surface Maximization.* Springer Texts in Statistics, Springer-Verlag New York, 2nd ed., 2001. [Cited on page 30]

[48] Great Learning Team, "Introduction to multivariate regression analysis." `https://www.mygreatlearning.com/blog/introduction-to-multivariate-regression/`, 2020. (Accessed on 05/24/2021). [Cited on page 31]

[49] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning with Applications in R.* Springer Texts in Statistics, Vol. 103, Springer, 2013. [Cited on pages 31 and 32]

[50] H. Blockeel, L. De Raedt, and J. Ramon, "Top-down induction of clustering trees," *Proc. 15th Intl. Conf. on Machine Learning*, 12 2000. [Cited on page 32]

[51] W. H. Delashmit and M. T. Manry, "Recent developments in multilayer perceptron neural networks," in *Proceedings of the 7th Annual Memphis Area Engineering and Science Conference, MAESC*, 2005. [Cited on page 34]

[52] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations*, 12 2014. [Cited on page 34]

[53] E. Almeida, C. Ferreira, and J. Gama, "Adaptive model rules from data streams," in *Machine Learning and Knowledge Discovery in Databases* (H. Blockeel, K. Kersting, S. Nijssen, and F. Železný, eds.), (Berlin, Heidelberg), pp. 480–492, Springer Berlin Heidelberg, 2013. [Cited on page 35]

[54] A. Osojnik, P. Panov, and S. DundefinedEroski, "Tree-based methods for online multi-target regression," *Journal of Intelligent Information Systems*, vol. 50, p. 315–339, Apr. 2018. [Cited on page 36]

[55] A. Osojnik, P. Panov, and S. Džeroski, "Multi-label classification via multi-target regression on data streams," *Machine Learning*, vol. 106, pp. 745–770, Jun 2017. [Cited on page 36]

[56] A. Osojnik, P. Panov, and S. Džeroski, "Incremental predictive clustering trees for online semi-supervised multi-target regression," *Machine Learning*, vol. 109, pp. 2121–2139, Nov 2020. [Cited on page 36]

[57] D. Boulegane, A. Bifet, H. Elghazel, and G. Madhusudan, "Streaming time series forecasting using multi-target regression with dynamic ensemble selection," in *2020 IEEE International Conference on Big Data (Big Data)*, pp. 2170–2179, 2020. [Cited on page 36]

[58] A. Bifet, G. Holmes, B. Pfahringer, and E. Frank, "Fast perceptron decision tree learning from evolving data streams," Jun 2010. [Cited on page 36]

[59] S. de Rooij and P. D. Grünwald, "Luckiness and regret in minimum description length inference," in *Philosophy of Statistics* (P. S. Bandyopadhyay and M. R. Forster, eds.), vol. 7 of *Handbook of the Philosophy of Science*, pp. 865–900, Amsterdam: North-Holland, 2011. [Cited on page 37]

[60] R. Riffenburgh, "Chapter 17 - bayesian statistics," in *Statistics in Medicine (Third Edition)* (R. Riffenburgh, ed.), pp. 355–364, San Diego: Academic Press, 3rd ed., 2012. [Cited on page 37]

[61] S. Geisser, *Predictive Inference: An Introduction*. Monographs on Statistics and Applied Probability 55, Springer US, 1993. [Cited on page 37]

[62] Fidelity Investments, "What are bollinger bands?." `https://www.fidelity.com/learning-center/trading-investing/technical-analysis/technical-indicator-guide/bollinger-bands`, 2021. (Accessed on 07/23/2021). [Cited on pages 38 and 54]

[63] M. Poldrugač and M. Komadina, "Social analytics for mobile operators," in *2012 Proceedings of the 35th International Convention MIPRO*, pp. 624–628, May 2012. [Cited on page 39]

[64] C. Dullaghan and E. Rozaki, "Integration of machine learning techniques to evaluate dynamic customer segmentation analysis for mobile customers," *International Journal of Data Mining & Knowledge Management Process*, vol. 7, p. 13–24, Jan 2017. [Cited on page 39]

[65] I. Ullah, B. Raza, A. K. Malik, M. Imran, S. U. Islam, and S. W. Kim, "A churn prediction model using random forest: Analysis of machine learning techniques for churn prediction and factor identification in telecom sector," *IEEE Access*, vol. 7, pp. 60134–60149, 2019. [Cited on page 39]

[66] A. De Caigny, K. Coussement, and K. W. De Bock, "A new hybrid classification algorithm for customer churn prediction based on logistic regression and decision trees," *European Journal of Operational Research*, vol. 269, no. 2, pp. 760–772, 2018. [Cited on page 39]

[67] H. Jain, A. Khunteta, and S. Srivastava, "Churn prediction in telecommunication using logistic regression and logit boost," *Procedia Computer Science*, vol. 167, pp. 101–112, 2020. International Conference on Computational Intelligence and Data Science. [Cited on page 39]

[68] G. Nie, W. Rowe, L. Zhang, Y. Tian, and Y. Shi, "Credit card churn forecasting by logistic regression and decision tree," *Expert Systems with Applications*, vol. 38, no. 12, pp. 15273–15285, 2011. [Cited on page 39]

[69] H. Jain, A. Khunteta, and S. Srivastava, "Telecom churn prediction and used techniques, datasets and performance measures: a review," *Telecommunication Systems*, vol. 76, no. 4, pp. 613–630, 2021. [Cited on page 39]

[70] Z. Can and E. Albey, "Churn prediction for mobile prepaid subscribers," in *Proceedings of the 6th International Conference on Data Science, Technology and Applications*, pp. 67–74, INSTICC, SciTePress, 2017. [Cited on page 40]

[71] R. Vyas, B. G. M. Prasad, H. K. Vamshidhar, and S. Kumar, "Predicting inactiveness in telecom (prepaid) sector: A complex bigdata application," in *2018 International Conference on Information Technology (ICIT)*, pp. 39–43, 2018. [Cited on page 40]

[72] L. Yang, D. Li, and Y. Lu, "Prediction modeling and analysis for telecom customer churn in two months." `https://arxiv.org/ftp/arxiv/papers/1911/1911.00558.pdf`, 2019. [Accessed in May 2021]. [Cited on page 40]

[73] L. Diettrichand, F. de Souza, and A. Guerreiro, "Developing credit scores with telco data using machine learning and agile methodology." `https://www.sas.com/content/dam/SAS/support/en/sas-global-forum-proceedings/2020/4831-2020.pdf`, 2020. (Accessed on 06/04/2021). [Cited on page 40]

[74] P. Sundsøy, J. Bjelland, B.-A. Reme, A. M.Iqbal, and E. Jahani, "Deep learning applied to mobile phone data for individual income classification," in *Proceedings of the 2016 International Conference on Artificial Intelligence: Technologies and Applications*, pp. 96–99, Atlantis Press, 2016/01. [Cited on page 40]

[75] M. Pratt, "Customer lifetime value 101 | adroll blog." `https://www.adroll.com/blog/customer-experience/customer-lifetime-value-101`, 2019. (Accessed on 04/16/2021). [Cited on page 40]

[76] C. Davidson, "Github - camdavidsonpilon/lifetimes: Lifetime value in python." `https://github.com/CamDavidsonPilon/lifetimes`, 2020. (Accessed on 04/16/2021). [Cited on page 40]

[77] D. Manzano-Machob, "The architecture of a churn prediction system based on stream mining," in *Proc. Artif. Intell. Res. Develop., 16th Int. Conf. Catalan Assoc. Artif. Intell.*, vol. 256, p. 157, 2013. [Cited on page 40]

[78] N. L. R. Machado and D. D. A. Ruiz, "Customer: A novel customer churn prediction method based on mobile application usage," in *2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*, (Cham), pp. 284–298, Springer International Publishing, 2017. [Cited on page 40]

[79] S. B. Tatar, A. McIntyre, N. Zincir-Heywood, and M. Heywood, "Benchmarking stream clustering for churn detection," in *Discovery Science* (M. S. Japkowicz N., ed.), pp. 2146–2151, June 2015. [Cited on page 40]

[80] J.-T. Wei, S.-Y. Lin, and H.-H. Wu, "A review of the application of RFM model," *African Journal of Business Management*, vol. 4, no. 19, pp. 4199–4206, 2010. [Cited on pages 41 and 46]

[81] PwC network, "Analytical imperatives for telecom marketers in emerging markets." `https://www.strategyand.pwc.com/m1/en/reports/hitting-the-target.pdf`, 2014. (Accessed on 04/15/2021). [Cited on page 45]

[82] h2o.ai, "Telecommunications | h2o.ai." `https://www.h2o.ai/telecom/`, 2021. (Accessed on 06/03/2021). [Cited on page 45]

[83] G. M. Weiss, *Data Mining in Telecommunications*, pp. 1189–1201. Boston, MA: Springer US, 2005. [Cited on page 46]

[84] D. Vorotyntsev, "Benchmarking categorical encoders: Towards data science." `https://towardsdatascience.com/benchmarking-categorical-encoders-9c322bd77ee8`, 2019. (Accessed on 04/15/2021). [Cited on page 48]

[85] G. H. John, R. Kohavi, and K. Pfleger, "Irrelevant features and the subset selection problem," in *Machine Learning Proceedings 1994* (W. W. Cohen and H. Hirsh, eds.), pp. 121–129, San Francisco (CA): Morgan Kaufmann, 1994. [Cited on page 51]

[86] S. Ramírez-Gallego, B. Krawczyk, S. García, M. Woźniak, and F. Herrera, "A survey on data preprocessing for data stream mining: Current status and future directions," *Neurocomputing*, vol. 239, pp. 39–57, 2017. [Cited on page 51]

[87] J. Cai, J. Luo, S. Wang, and S. Yang, "Feature selection in machine learning: A new perspective," *Neurocomputing*, vol. 300, pp. 70–79, 2018. [Cited on page 51]

[88] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996. [Cited on page 52]

[89] P. M. Alves, R. Ângelo Filipe, and B. Malheiro, "Towards top-up prediction on telco operators," in *Progress in Artificial Intelligence. EPIA: 20th Portuguese Conference on Artificial Intelligence*, Lecture Notes in Artificial Intelligence, (Cham), APPIA, Springer International Publishing, 2021. [Cited on page 52]

[90] J. Duarte, J. Gama, and A. Bifet, "Adaptive model rules from high-speed data streams," *ACM Trans. Knowl. Discov. Data*, vol. 10, Jan. 2016. [Cited on page 53]

[91] A. Bifet, G. Holmes, R. Kirkby, and B. Pfahringer, "MOA: massive online analysis," *Journal of Machine Learning Research*, vol. 11, no. 52, pp. 1601–1604, 2010. [Cited on page 53]

[92] J. Gama, R. Sebastiao, and P. P. Rodrigues, "On evaluating stream learning algorithms," *Machine learning*, vol. 90, no. 3, pp. 317–346, 2013. [Cited on page 57]

[93] M. Waskom, "seaborn: statistical data visualization — seaborn 0.11.1 documentation." `https://seaborn.pydata.org/`, 2020. (Accessed on 04/15/2021). [Cited on page 59]

[94] Jupyter, "Project jupyter | home." `https://jupyter.org/`, 2021. (Accessed on 04/15/2021). [Cited on page 59]

[95] scikit learn, "scikit-learn: machine learning in python — scikit-learn 0.24.1 documentation." `https://scikit-learn.org/stable/`, 2021. (Accessed on 04/15/2021). [Cited on page 59]

Appendix A

Offline Single-Target Experiments

This Appendix is divided in two Sections: Section A.1 displays the various results for the experiments of feature selection so that future regression experiments are populated with the only the most representative features; Section A.2 presents the detailed outputs of the optimal window size experiments to find the appropriate window size for the single-target experiments so these can be compared with the multi-target results.

A.1 Feature Selection

The feature selection experiments are particularly important to reduce redundancy and complexity for the modelling step. In this case, various feature selection techniques were tested as Table A.1 shows. The number of features to be selected is 15, for reasons aforementioned in Chapter 4.2.1. From the results, it is possible to conclude that the Shrinkage selection technique presents the best results for both targets. Another visible trend is that Decision Tree and Random Forest techniques produce the lowest error metrics independently of the feature selection technique and target.

Table A.1: Feature selection with a thirty-month sliding window

		Monthly Frequency			Monthly Value		
		#	**MAE**	**RMSE**	#	**MAE**	**RMSE**
All Features	OLS	26	0.004 39	0.007 66	26	0.000 43	0.000 80
	DT		0.000 74	0.003 29		0.000 08	0.000 37
	RF		0.000 76	0.002 74		0.000 08	0.000 61
	MLP		0.002 03	0.003 64		0.000 42	0.000 70
Correlated	OLS	15	0.005 70	0.009 02	15	0.000 50	0.000 25
	DT		0.000 56	0.002 41		0.000 05	0.000 27
	RF		0.000 64	0.002 36		0.000 05	0.000 45
	MLP		0.002 33	0.004 05		0.000 32	0.000 56
Forward	OLS	15	0.004 45	0.007 73	15	0.000 44	0.000 80
	DT		0.000 76	0.003 32		0.000 08	0.000 38
	RF		0.000 73	0.002 63		0.000 08	0.000 55
	MLP		0.002 06	0.003 57		0.000 24	0.000 50
Backward	OLS	15	0.004 45	0.007 73	15	0.000 44	0.000 80
	DT		0.000 76	0.003 33		0.000 08	0.000 39
	RF		0.000 73	0.002 64		0.000 08	0.000 46
	MLP		0.001 69	0.003 31		0.000 20	0.000 48
RFE	OLS	15	0.006 87	0.013 22	15	0.000 44	0.000 80
	DT		0.000 74	0.003 29		0.000 08	0.000 39
	RF		0.003 57	0.007 73		0.000 06	0.000 52
	MLP		0.006 90	0.011 29		0.000 41	0.000 62
RFECV	OLS	15	0.004 39	0.007 66	15	0.000 43	0.000 80
	DT		0.000 73	0.003 27		0.000 08	0.000 38
	RF		0.000 57	0.002 18		0.000 08	0.000 56
	MLP		0.001 77	0.003 37		0.000 41	0.000 87
Univariate	OLS	15	0.004 43	0.007 70	15	0.000 44	0.000 80
	DT		0.000 82	0.003 47		0.000 08	0.000 38
	RF		0.000 78	0.002 83		0.000 08	0.000 48
	MLP		0.002 10	0.003 80		0.001 31	0.001 68
Shrinkage	OLS	15	0.004 39	0.007 66	15	0.000 43	0.000 80
	DT		0.000 60	0.002 29		**0.00006**	**0.00029**
	RF		**0.00057**	**0.00218**		**0.00006**	0.000 56
	MLP		0.002 57	0.003 82		0.000 53	0.000 71

A.2 Optimal Window Size Selection

The single-target offline optimal window size experiments have the intent of finding the window size which produces the best results. Additionally, the performance of the different regression techniques are also analysed to identify the one which returns the lowest error metrics for each target the fastest. This step also serves as benchmark for future multi-target offline regression experiments.

From Tables A.2, A.3, A.4 and A.5 it is possible to conclude that the regression techniques have the commonality of the optimal window size being the almost the largest if not the biggest. This behaviour is somewhat expected since client activity is scarce and, in smaller window sizes, many clients have no top-ups, making it difficult to accurately predict the targets.

Table A.2: MOLS optimal window size selection

		Monthly Frequency			Monthly Value		
Technique	**Window** (month)	**MAE**	**RMSE**	**Time** (s)	**MAE**	**RMSE**	**Time** (s)
MOLS	30	0.004 39	0.007 66	0.534 11	0.000 43	0.000 80	0.188 08
	29	**0.004 14**	**0.007 07**	0.948 97	0.000 43	0.000 80	0.048 44
	28	0.004 39	0.007 61	1.375 35	0.000 43	0.000 80	0.640 29
	27	0.004 39	0.007 59	1.853 79	0.000 43	0.000 80	0.840 95
	26	0.004 37	0.007 55	2.214 70	0.000 43	0.000 80	1.092 45
	25	0.004 36	0.007 53	1.283 81	0.003 61	0.005 11	1.282 40
	24	0.004 35	0.007 50	1.547 79	0.000 43	0.000 80	1.355 76
	23	0.004 56	0.007 86	1.763 79	0.005 60	0.001 03	1.482 59
	22	0.004 73	0.008 14	1.975 64	0.000 65	0.001 20	1.961 30
	21	0.004 86	0.008 35	2.013 38	0.007 20	0.001 35	1.954 98
	20	0.004 95	0.008 50	2.120 17	0.000 78	0.001 46	2.107 58
	19	0.005 08	0.008 73	2.348 67	0.000 85	0.001 57	2.143 07
	18	0.005 24	0.008 97	2.497 32	0.000 90	0.001 66	2.288 66
	17	0.005 31	0.009 04	2.653 86	0.001 11	0.002 08	2.533 57
	16	0.005 36	0.009 11	2.546 27	0.001 33	0.002 49	2.529 58
	15	0.005 47	0.009 29	2.759 53	0.001 50	0.002 86	2.661 50
	14	0.005 37	0.009 14	6.111 72	0.001 64	0.003 14	2.748 59
	13	0.005 27	0.008 98	5.883 33	0.001 83	0.003 52	2.790 36
	12	0.005 18	0.008 85	2.931 23	0.001 99	0.003 81	3.031 18
	11	0.005 12	0.008 74	2.910 95	0.002 09	0.004 03	2.970 70
	10	0.005 03	0.008 68	2.930 64	0.002 29	0.004 40	3.061 98
	9	0.006 84	0.011 12	2.933 15	0.002 56	0.004 89	2.994 58
	8	0.007 06	0.011 40	2.859 01	0.002 91	0.005 32	2.927 61
	7	0.005 07	0.008 69	2.829 89	0.003 26	0.005 93	2.897 56
	6	0.004 98	0.008 52	2.618 86	0.003 50	0.006 51	2.641 33
	5	0.004 81	0.008 18	2.430 28	0.003 98	0.007 13	2.420 46

Table A.3: MLP optimal window size selection

		Monthly Frequency			Monthly Value		
Technique	**Window** (month)	**MAE**	**RMSE**	**Time** (s)	**MAE**	**RMSE**	**Time** (s)
MLP	**30**	**0.001 44**	**0.003 04**	14.878 05	0.000 36	0.000 72	15.334 16
	29	0.002 10	0.003 49	34.959 23	0.001 20	0.001 47	29.195 68
	28	0.001 87	0.003 59	54.451 20	0.000 73	0.000 94	44.089 44
	27	0.001 78	0.003 57	59.630 81	0.000 33	0.000 71	58.008 93
	26	0.001 64	0.003 30	77.980 61	0.000 38	0.000 70	73.702 55
	25	0.001 69	0.003 32	101.932 44	0.003 48	0.005 01	83.123 39
	24	0.002 30	0.004 10	115.104 99	0.000 53	0.000 84	91.896 02
	23	0.002 47	0.003 97	132.254 99	0.000 44	0.000 95	104.648 83
	22	0.001 84	0.003 63	143.823 61	0.000 39	0.000 86	116.955 04
	21	0.001 88	0.003 81	147.419 20	0.000 80	0.001 21	123.994 28
	20	0.002 12	0.003 95	263.882 90	0.000 72	0.001 21	133.278 96
	19	0.001 99	0.003 98	305.085 71	0.000 80	0.001 31	143.665 83
	18	0.002 53	0.004 45	322.204 21	0.000 680	0.001 20	150.755 29
	17	0.002 60	0.004 56	301.909 30	0.000 87	0.001 56	151.091 22
	16	0.002 34	0.004 39	178.703 50	0.000 87	0.001 62	163.352 66
	15	0.002 43	0.004 64	181.912 87	0.000 98	0.001 85	163.709 42
	14	0.002 68	0.004 87	178.278 62	0.001 12	0.002 08	163.184 49
	13	0.003 60	0.005 80	180.824 08	0.001 46	0.002 54	159.593 86
	12	0.002 83	0.004 92	176.199 72	0.001 51	0.002 69	182.925 40
	11	0.002 93	0.004 99	187.115 11	0.001 42	0.002 65	188.925 27
	10	0.002 81	0.005 01	179.977 27	0.001 75	0.003 02	189.806 67
	9	0.003 03	0.005 46	178.970 60	0.001 82	0.003 24	185.656 91
	8	0.003 97	0.006 49	179.837 22	0.002 44	0.004 00	284.266 66
	7	0.004 88	0.010 30	14.602 85	0.002 46	0.004 19	295.651 64
	6	0.005 66	0.011 73	13.406 02	0.002 79	0.004 81	456.319 27
	5	0.006 85	0.014 09	11.248 45	0.003 38	0.005 52	376.771 58

Table A.4: DT optimal window size selection

		Monthly Frequency			Monthly Value		
Technique	**Window** (month)	**MAE**	**RMSE**	**Time** (s)	**MAE**	**RMSE**	**Time** (s)
DT	**30**	**0.000 60**	**0.002 32**	1.575 75	0.000 06	0.000 29	2.445 72
	29	0.000 62	0.002 36	3.132 52	0.000 06	0.000 29	4.684 74
	28	0.000 63	0.002 36	4.617 68	0.000 06	0.000 29	6.549 35
	27	0.000 64	0.002 37	6.001 55	0.000 06	0.000 29	9.056 62
	26	0.000 65	0.002 39	7.317 22	0.000 06	0.000 30	11.067 86
	25	0.000 67	0.002 45	8.604 13	0.003 31	0.004 70	12.842 95
	24	0.000 69	0.002 51	9.667 24	0.000 07	0.000 31	14.522 51
	23	0.000 77	0.002 70	10.422 75	0.000 10	0.000 43	15.732 75
	22	0.000 83	0.002 84	11.559 23	0.000 65	0.001 20	17.452 88
	21	0.000 89	0.003 00	12.409 73	0.000 16	0.000 65	19.313 10
	20	0.000 98	0.003 22	13.186 75	0.000 78	0.001 46	20.766 74
	19	0.001 04	0.003 34	13.916 74	0.000 25	0.000 84	23.176 30
	18	0.001 31	0.003 65	14.325 03	0.000 90	0.001 66	22.596 89
	17	0.001 62	0.004 07	15.181 11	0.000 36	0.001 17	22.743 13
	16	0.001 83	0.004 41	15.505 80	0.000 58	0.001 59	22.055 70
	15	0.002 33	0.004 99	16.433 28	0.000 59	0.001 85	24.399 04
	14	0.002 43	0.005 22	16.573 48	0.001 64	0.003 14	24.661 53
	13	0.002 48	0.005 32	16.794 43	0.000 82	0.002 28	24.940 51
	12	0.002 63	0.005 58	17.085 48	0.000 98	0.002 64	25.283 71
	11	0.002 92	0.006 13	17.421 61	0.001 06	0.002 92	25.348 62
	10	0.003 22	0.006 42	16.579 38	0.001 14	0.003 34	25.179 91
	9	0.005 62	0.011 34	16.294 12	0.001 62	0.004 29	24.738 23
	8	0.006 57	0.013 35	15.678 98	0.001 91	0.004 48	23.710 69
	7	0.004 88	0.010 30	14.602 85	0.002 18	0.005 00	22.104 20
	6	0.005 66	0.011 73	13.406 02	0.002 72	0.006 31	19.698 09
	5	0.006 85	0.014 09	11.248 45	0.003 33	0.007 41	16.664 47

Table A.5: RF optimal window size selection

		Monthly Frequency			Monthly Value		
Technique	**Window** (month)	**MAE**	**RMSE**	**Time** (s)	**MAE**	**RMSE**	**Time** (s)
RF	**30**	**0.000 57**	**0.002 17**	94.930 22	0.000 06	0.000 59	148.701 22
	29	0.000 60	0.002 26	197.776 47	0.000 07	0.000 44	284.337 39
	28	0.000 61	0.002 26	292.809 34	0.000 07	0.000 41	419.851 38
	27	0.000 63	0.002 28	381.707 67	0.000 07	0.000 49	547.938 61
	26	0.000 63	0.002 27	459.537 28	0.000 07	0.000 49	669.217 17
	25	0.000 65	0.002 31	568.163 05	0.003 31	0.004 86	788.619 40
	24	0.000 68	0.002 37	645.768 03	0.000 08	0.000 49	891.525 36
	23	0.000 74	0.002 57	698.838 86	0.000 10	0.000 58	978.990 72
	22	0.000 80	0.002 68	740.938 61	0.000 13	0.000 62	1095.160 44
	21	0.000 85	0.002 83	847.869 84	0.000 16	0.000 67	1202.345 00
	20	0.000 91	0.002 95	886.935 76	0.000 17	0.000 69	1288.746 59
	19	0.000 99	0.003 13	924.946 68	0.000 22	0.000 79	1284.217 87
	18	0.001 26	0.003 42	986.790 86	0.000 26	0.000 89	1344.858 03
	17	0.001 53	0.003 74	1015.898 77	0.000 33	0.001 08	3012.136 13
	16	0.001 71	0.004 05	1026.992 72	0.000 41	0.001 26	1426.505 17
	15	0.002 16	0.004 53	1033.395 28	0.000 48	0.001 48	1451.918 13
	14	0.002 26	0.004 70	1036.914 46	0.000 55	0.001 65	1560.573 16
	13	0.002 28	0.004 76	1067.188 84	0.000 67	0.001 87	1500.919 38
	12	0.002 41	0.004 99	1085.558 56	0.000 76	0.002 06	1543.466 44
	11	0.002 60	0.005 27	1045.415 28	0.000 84	0.002 20	1563.982 25
	10	0.002 96	0.005 70	1027.236 13	0.000 93	0.002 49	1483.303 10
	9	0.005 55	0.008 71	1007.883 71	0.001 32	0.003 24	1491.637 46
	8	0.006 29	0.010 13	973.266 58	0.001 71	0.003 71	1414.462 40
	7	0.004 45	0.008 40	933.265 46	0.001 84	0.003 94	1294.837 28
	6	0.005 16	0.009 55	821.915 63	0.002 20	0.004 57	1130.201 10
	5	0.005 97	0.010 56	635.213 35	0.002 71	0.005 40	964.033 21

Appendix B

Offline Multi-Target Experiments

This Appendix describes the results of the multi-target offline experiments related to the optimal window size selection so the results obtained with them can be compared to the online scenario. It is important to mention that the time values for these experiments were measured once the data was already loaded in to memory, hence making reference specifically to the training and testing. In case the time to read the data is taken into account, the run-time should be significantly higher, especially for bigger time windows.

B.1 Optimal Window Size Selection

In order to find the optimal window size experiments from the smallest window to biggest window possible were made. The goal is to determine the smallest window with lowest error metrics to meet the minimum hardware requirements. This requirement is particularly important for the batch learning scenario because, as it is possible to infer from the the processing time, it presents an heavy load to the hardware of the machine.

Table B.1: Offline single-target MOLS regression

Technique	**Window** (month)	**MAE**	**RMSE**	**Time** (s)
MOLS	30	0.003 92	0.006 57	0.613 63
	29	0.003 90	0.006 52	0.751 08
	28	0.003 88	0.006 48	2.439 80
	27	0.003 85	0.006 43	3.135 91
	26	0.003 81	0.006 38	1.950 45
	25	0.003 78	0.006 32	1.843 17
	24	**0.003 74**	**0.006 26**	2.780 72
	23	0.003 99	0.006 70	2.937 82
	22	0.004 19	0.007 03	2.577 04
	21	0.004 33	0.007 26	2.667 33
	20	0.004 42	0.007 42	2.957 08
	19	0.004 53	0.007 62	3.354 44
	18	0.004 62	0.007 79	3.453 25
	17	0.004 83	0.008 14	3.428 32
	16	0.004 98	0.008 44	3.566 96
	15	0.005 13	0.008 74	3.669 45
	14	0.005 17	0.008 83	3.857 59
	13	0.005 25	0.009 00	3.949 57
	12	0.005 28	0.009 08	3.952 77
	11	0.005 29	0.009 12	4.066 48
	10	0.005 34	0.009 22	4.127 50
	9	0.005 41	0.009 40	3.931 56
	8	0.005 61	0.009 64	3.783 09
	7	0.005 72	0.009 96	3.842 70
	6	0.005 76	0.010 05	3.650 94
	5	0.005 78	0.009 98	3.477 22

Table B.2: Offline single-target DT regression

Technique	**Window** (month)	**MAE**	**RMSE**	**Time** (s)
DT	30	**0.000 30**	**0.001 35**	2.286 20
	29	0.000 32	0.001 41	4.610 17
	28	0.000 32	0.001 40	9.792 84
	27	0.000 34	0.001 43	8.869 59
	26	0.000 34	0.001 42	10.516 75
	25	0.000 36	0.001 47	14.898 28
	24	0.000 37	0.001 50	15.272 51
	23	0.000 42	0.001 68	20.503 41
	22	0.000 48	0.004 88	23.890 52
	21	0.000 54	0.002 07	25.716 17
	20	0.000 58	0.002 16	31.778 63
	19	0.000 69	0.002 46	37.481 23
	18	0.000 85	0.002 66	28.764 94
	17	0.001 08	0.003 05	29.253 03
	16	0.001 25	0.003 40	30.289 25
	15	0.001 62	0.003 89	31.787 52
	14	0.001 74	0.004 20	32.772 25
	13	0.002 05	0.004 96	32.833 75
	12	0.002 36	0.005 57	33.266 12
	11	0.002 41	0.005 68	41.456 18
	10	0.002 74	0.006 29	37.690 98
	9	0.003 71	0.008 35	30.897 26
	8	0.004 02	0.008 58	32.945 16
	7	0.003 99	0.009 39	31.568 02
	6	0.004 33	0.009 08	27.909 34
	5	0.005 25	0.011 13	22.921 80

Table B.3: Offline single-target MLP regression

Technique	**Window** (month)	**MAE**	**RMSE**	**Time** (s)
MLP	30	0.001 37	0.002 20	19.696 69
	29	0.001 41	0.002 42	50.285 89
	28	0.001 44	0.002 35	80.322 93
	27	0.001 34	0.002 21	109.534 41
	26	0.001 81	0.002 69	158.630 94
	25	0.000 99	0.001 93	203.364 62
	24	0.001 35	0.002 22	252.409 50
	23	**0.001 25**	**0.002 21**	332.876 44
	22	0.001 40	0.002 46	298.887 53
	21	0.001 39	0.002 49	311.018 80
	20	0.002 35	0.003 51	567.568 07
	19	0.001 60	0.002 72	703.001 12
	18	0.001 70	0.002 93	512.446 73
	17	0.001 62	0.002 97	910.751 32
	16	0.001 82	0.003 26	543.905 21
	15	0.002 02	0.003 58	467.363 50
	14	0.002 12	0.003 70	786.928 55
	13	0.002 10	0.003 80	1436.022 04
	12	0.001 97	0.003 71	555.082 18
	11	0.002 17	0.004 73	5648.754 38
	10	0.002 48	0.004 35	617.667 92
	9	0.002 46	0.004 50	1474.405 61
	8	0.003 23	0.005 34	424.575 79
	7	0.003 00	0.005 29	354.519 65
	6	0.003 10	0.005 53	766.769 19
	5	0.003 61	0.006 22	203.526 46

Table B.4: Offline single-target RF regression

Technique	**Window** (month)	**MAE**	**RMSE**	**Time** (s)
RF	30	**0.000 31**	**0.001 37**	147.963 14
	29	0.000 33	0.001 47	302.058 29
	28	0.000 33	0.001 41	431.917 39
	27	0.000 34	0.001 42	586.964 54
	26	0.000 34	0.001 41	704.322 17
	25	0.000 36	0.001 44	867.447 68
	24	0.000 38	0.001 49	981.016 20
	23	0.000 43	0.001 65	1083.976 14
	22	0.000 48	0.001 75	1077.749 95
	21	0.000 54	0.001 91	1195.538 76
	20	0.000 57	0.001 98	1253.469 60
	19	0.000 66	0.002 18	1356.505 85
	18	0.000 79	0.002 35	1444.366 47
	17	0.001 04	0.002 69	1470.874 96
	16	0.001 22	0.002 98	1543.221 95
	15	0.001 49	0.003 31	1491.551 21
	14	0.001 67	0.003 66	1543.352 75
	13	0.001 83	0.004 03	1623.905 67
	12	0.002 17	0.004 75	1660.297 60
	11	0.002 17	0.004 73	1584.812 87
	10	0.002 48	0.005 37	1629.258 67
	9	0.003 39	0.007 15	1518.767 96
	8	0.003 79	0.007 49	1462.813 18
	7	0.003 49	0.007 31	1313.544 63
	6	0.003 84	0.007 42	1206.717 92
	5	0.004 66	0.008 95	999.467 55

Appendix C

Online Multi-Target Experiments

This Appendix describes the results for the multi-target online experiments related to the optimal window size selection so the results obtained with them can be compared with the offline scenario in order to chose the best regression technique and window size. In contrast to the offline multi-target experiments, the time in the following tables, is discriminated into the reading, training and testing. As a result, it is not possible to directly compare the run-time between scenarios. Nevertheless, it is possible to conclude that the online scenario is faster.

C.1 Optimal Window Size Selection

The following experiments for the online scenario have the intent of finding the window size, in number of events, which produces the lowest error metrics. The comparison between techniques considers not only the error metrics and run-time, but also through the size of the window, *i.e.* a smaller window of events and similar or lower error metrics is preferred over a larger window with identical error metrics.

Hence, Tables C.1, C.2, C.3 and C.4 display the results of the experiments with window sizes from 100 000 events to 2 875 099 events with an increment of 100 000 events. The increment was chosen with the intent of mimicking the number of events in a month so the comparison between the online and offline scenarios is more fair.

Table C.1: BMTR optimal window size selection

Technique	**Window** (event)	**MAE**	**RMSE**	**Time** (s)
BMTR	100 000	0.001 13	0.004 68	226.265 63
	200 000	0.001 09	0.004 43	249.234 38
	300 000	0.001 09	0.004 31	258.171 88
	400 000	0.001 08	0.004 24	253.531 25
	500 000	**0.001 08**	**0.004 11**	**260.312 50**
	600 000	0.001 11	0.004 40	253.562 50
	700 000	0.001 12	0.004 37	246.265 63
	800 000	0.001 12	0.004 36	252.937 50
	900 000	0.001 13	0.004 39	258.546 88
	1 000 000	0.001 13	0.004 34	255.109 38
	1 100 000	0.001 13	0.004 36	265.453 13
	1 200 000	0.001 13	0.004 31	246.906 25
	1 300 000	0.001 14	0.004 34	242.125 00
	1 400 000	0.001 15	0.004 33	241.796 88
	1 500 000	0.001 18	0.005 59	243.390 63
	1 600 000	0.001 26	0.025 23	241.562 50
	1 700 000	0.001 51	0.142 98	234.656 25
	1 800 000	0.001 58	0.163 48	242.312 50
	1 900 000	0.001 56	0.159 12	239.343 75
	2 000 000	0.001 53	0.155 10	248.484 38
	2 100 000	0.001 51	0.151 36	246.828 13
	2 200 000	0.001 49	0.147 88	237.234 38
	2 300 000	0.001 47	0.144 63	243.281 25
	2 400 000	0.001 45	0.141 59	237.062 50
	2 500 000	0.001 43	0.138 73	241.078 13
	2 600 000	0.001 41	0.136 04	240.203 13
	2 700 000	0.001 39	0.133 50	240.093 75
	2 800 000	0.001 38	0.131 10	236.968 75
	2 875 099	0.001 37	0.129 38	244.234 38

Table C.2: MTPR optimal window size selection

Technique	**Window** (event)	**MAE**	**RMSE**	**Time** (s)
MTPR	100 000	0.001 45	0.005 00	27.078 13
	200 000	0.001 43	0.004 61	27.500 00
	300 000	0.001 42	0.004 43	27.562 50
	400 000	0.001 42	0.004 35	27.406 25
	500 000	0.001 41	0.004 22	27.250 00
	600 000	0.001 42	0.004 49	27.375 00
	700 000	0.001 42	0.004 44	27.375 00
	800 000	0.001 42	0.004 41	27.593 75
	900 000	0.001 42	0.004 41	27.484 38
	1 000 000	0.001 41	0.004 32	27.734 38
	1 100 000	0.001 41	0.004 32	27.671 88
	1 200 000	0.001 40	0.004 24	27.562 50
	1 300 000	0.001 40	0.004 24	27.437 50
	1 400 000	**0.001 39**	**0.004 18**	**26.984 38**
	1 500 000	0.001 40	0.004 26	27.359 38
	1 600 000	0.001 40	0.004 25	27.031 25
	1 700 000	0.001 46	0.004 49	27.234 38
	1 800 000	0.001 71	0.015 67	27.015 63
	1 900 000	0.001 70	0.152 49	27.796 88
	2 000 000	0.001 68	0.148 63	27.231 30
	2 100 000	0.001 67	0.145 05	27.328 13
	2 200 000	0.001 65	0.141 72	27.468 75
	2 300 000	0.001 64	0.138 60	27.671 88
	2 400 000	0.001 63	0.135 69	27.578 13
	2 500 000	0.001 62	0.132 95	27.312 50
	2 600 000	0.001 61	0.130 37	27.687 50
	2 700 000	0.001 60	0.127 93	27.703 13
	2 800 000	0.001 59	0.125 63	26.921 88
	2 875 099	0.001 58	0.123 98	27.968 75

Table C.3: iSOUP optimal window size selection

Technique	**Window** (event)	**MAE**	**RMSE**	**Time** (s)
	100 000	0.008 00	0.022 52	89.015 63
	200 000	0.007 99	0.022 38	61.984 38
	300 000	0.007 96	0.022 13	51.609 38
	400 000	0.007 94	0.022 05	51.421 88
	500 000	**0.007 89**	**0.021 93**	**46.187 50**
	600 000	0.007 96	0.022 28	44.218 75
	700 000	0.008 00	0.022 45	45.578 13
	800 000	0.008 03	0.022 56	42.468 75
	900 000	0.008 05	0.022 65	44.046 88
	1 000 000	0.008 02	0.022 55	38.421 88
	1 100 000	0.008 01	0.022 57	40.484 38
	1 200 000	0.007 99	0.022 51	39.203 13
	1 300 000	0.008 00	0.022 54	40.265 63
	1 400 000	0.007 99	0.022 50	40.578 13
iSOUP	1 500 000	0.008 01	0.022 55	39.140 63
	1 600 000	0.008 00	0.022 52	38.737 50
	1 700 000	0.008 00	0.022 51	40.375 00
	1 800 000	0.008 00	0.022 49	40.578 13
	1 900 000	0.008 01	0.022 52	42.656 25
	2 000 000	0.008 01	0.022 47	39.531 25
	2 100 000	0.008 02	0.022 49	40.015 63
	2 200 000	0.008 01	0.022 45	40.687 50
	2 300 000	0.008 03	0.022 47	40.343 75
	2 400 000	0.008 05	0.022 50	41.968 75
	2 500 000	0.008 07	0.022 57	42.187 50
	2 600 000	0.008 09	0.022 63	43.218 75
	2 700 000	0.008 10	0.022 65	46.234 38
	2 800 000	0.008 11	0.022 67	45.437 50
	2 875 099	0.008 11	0.022 66	35.656 25

Table C.4: AMR optimal window size selection

Technique	**Window** (event)	**MAE**	**RMSE**	**Time** (s)
	100 000	0.001 18	0.004 67	137.140 63
	200 000	0.001 19	0.004 47	136.015 63
	300 000	0.001 16	0.004 34	138.750 00
	400 000	0.001 15	0.004 28	136.140 63
	500 000	**0.001 14**	**0.004 15**	**140.718 75**
	600 000	0.001 15	0.004 41	135.984 38
	700 000	0.001 15	0.004 37	138.687 50
	800 000	0.001 15	0.004 40	140.765 63
	900 000	0.001 16	0.004 41	140.812 50
	1 000 000	0.001 15	0.004 35	138.968 75
	1 100 000	0.001 15	0.004 37	141.906 25
	1 200 000	0.001 15	0.004 32	140.578 13
	1 300 000	0.001 16	0.004 34	137.390 63
	1 400 000	0.001 17	0.004 32	134.781 25
AMRules	1 500 000	0.001 21	0.013 09	137.250 00
	1 600 000	0.001 28	0.023 43	139.140 63
	1 700 000	0.001 46	0.076 51	140.250 00
	1 800 000	0.001 52	0.094 50	147.796 88
	1 900 000	0.001 50	0.091 99	144.406 25
	2 000 000	0.001 47	0.089 66	143.593 75
	2 100 000	0.001 45	0.087 51	143.500 00
	2 200 000	0.001 43	0.085 50	142.390 63
	2 300 000	0.001 41	0.083 63	142.640 63
	2 400 000	0.001 39	0.081 87	143.515 63
	2 500 000	0.001 38	0.080 22	141.328 13
	2 600 000	0.001 36	0.078 66	143.484 38
	2 700 000	0.001 34	0.077 19	142.750 00
	2 800 000	0.001 34	0.075 81	143.953 13
	2 875 099	0.001 32	0.075 82	143.687 50