



NOVA

IMS

Information
Management
School

MAAA

Mestrado em Métodos Analíticos Avançados
Master Program in Advanced Analytics

**Predictive Maintenance use case
employing Survival Analysis in a
telecommunication company**

Marta Carochó de Sousa Costa

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

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

A big thank you to all my friends and family.

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

**PREDICTIVE MAINTENANCE USE CASE EMPLOYING SURVIVAL
ANALYSIS IN A TELECOMMUNICATION COMPANY**

by

Marta Carochó de Sousa Costa

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

Advisor: *Prof Doutor* Flávio Luis Portas Pinheiro

External Advisor: *Doutor* Luís Carlos Santos

03 2021

ABSTRACT

Driven by the digital revolution, telecommunications companies need to adopt innovative technologies and services to be competitive. In this context, the company invests in its first **Predictive Maintenance** solution, intelligent anticipation of device failure through sensorial data. This solution has the power to anticipate and plan **Reactive Maintenance** measures that extend equipment's life, reduce downtime, aim for cost savings, and avoid negative feedback, consequently improving the service quality. This project explores a Fault Prediction tool such as **Survival Analysis**. It undergoes the six phases of a Data Science Project following the **CRISP-DM** methodology.

For applying the **Survival Analysis** technique (e.g. **Kaplan-Meier**), it is crucial to identify two key events using the equipment's historical data (e.g. **STB**): The beginning of the anomalous event and the exact moment of the fault event. Several techniques, such as Statistical Smoothing models and Anomaly Detection models, were analysed and compared in detail to detect the beginning of the device malfunction. The best results to detect the devices' anomalous event were employed by a Statistical technique, the **SMA** where the anomalous event is reaching 50 degrees for the one-day smooth average.

Therefore, it is possible to obtain an acceptable anticipation period of 38 days for future equipment maintenance intervention. In this sense, employing a **Predictive Maintenance** solution guarantees the reduction of 71% of the actual emergency interventions. Consequently, the company saves more money rather than not making any prediction at all.

Moreover, it was also developed a visualisation tool to demonstrate the solution and explore it, where it employs the different models to detect the beginning of the anomalous event's. Consequently, all the proposed goals of the company were accomplished.

Keywords: Predictive Maintenance, Survival Analysis, Anomaly Detection, Fault Prediction

CONTENTS

List of Figures	xi
List of Tables	xiii
Acronyms	xv
1 Introduction	1
1.1 Project Goals	3
1.2 Structure and Methodology	4
1.3 Company overview	5
2 Theoretical Framework	7
2.1 Data Understanding	7
2.1.1 Time Series data	8
2.2 Data Preparation	10
2.2.1 Data Cleaning	10
2.2.2 Data Construction: Statistical	11
2.2.3 Data Construction: Anomaly Detection	12
2.3 Modelling - Survival Analysis	17
2.3.1 Kaplan-Meier	20
2.4 Evaluation	21
3 Project Development	23
3.1 Data Understanding	24
3.1.1 Data Description	25
3.1.2 Data Exploration	27
3.2 Data Preparation	31
3.2.1 Data Cleaning	32
3.2.2 Data Construction	34
3.3 Data Modelling and Evaluation	39
3.4 Deployment	43
4 Conclusions	47
4.1 Report Evaluation and Lessons Learned	48

CONTENTS

4.2	Limitations	49
4.3	Future Work	49
	Bibliography	51
	Appendices	55
A	Appendix 1	55

LIST OF FIGURES

1.1	A RM example: STB running for an unknown reason and the client has to call to the technical support, this unplanned intervention is more expensive than scheduled one for the company.	2
1.2	The structure of this report divided into six different phases.	4
2.1	Representation the Isolation Forest model.	13
2.2	Representation the Autoencoder model.	16
2.3	Types of censored data that are, Right-censored, Left-censored and Interval-censored.	18
2.4	Representation of a Survival Curve, that defines the survival function $S(t)$ on the y-axis, and time t on the x-axis.	19
3.1	The project timeline with the expected and the real time represented in weeks.	24
3.2	Each client historical data can be associated to one of A, B, C or D scenario.	26
3.3	Correlation matrix that expresses the relationship between the numerical features employing Pearson coefficient.	27
3.4	On the right represents a box plot for each numerical feature for the inactive devices. On the left represents a box plot for each numerical feature for the active devices.	28
3.5	Two line plots with time against the $CPUT_p$ value, they show two distinct behaviours. The grey line (Normal) corresponds to scenario A and the red line (Changed) corresponds to scenario C.	29
3.6	Bivariate analysis with a scatter and an histogram plots, the top plot is $DMFRE$ against $CPULV$ and the bottom plot is $AMUSE$ vs $CPUT_p$	30
3.7	Process of constructing the final data set for the model with all the important columns event , end date , start date , time event process and feature .	34
3.8	Line plot with time against $CPUT_p$ that compares the behaviour of a device with grey shadow and the effect of applying EMA technique with different lags represented by the remaining colours.	35
3.9	Line plot with time against $CPUT_p$ that compares the behaviour of a device with grey shadow and the effect of applying SMA technique with different lags represented by the remaining colours.	35

LIST OF FIGURES

3.10	Process of getting the start date by means of Anomaly Detection Approach models, with input the time series and the output the start date	36
3.11	A multiple Survival Curve comparison employing the Kaplan-Meier Estimator.	40
3.12	Parallel box-plot representing the distribution of time event process for the inactive devices for the different data construction models.	40
3.13	A Survival Curve employing the Kaplan-Meier Estimator.	42
3.14	An example of an interactive dashboard illustrating the solution for the proposed business problem.	44
A.1	A flow that represents the architecture of the package pipeline.	56
A.2	The structure of the package pipeline that followed the cookiecutter structure.	57

LIST OF TABLES

1.1	Identification of the company's main three goals that goes through the theoretical, resolution and visualization for solving the PM problem; and their respective location on the report.	3
2.1	Table that describes the characteristic of the different type of features. . .	8
2.2	Table that describes the terms that accomplish the quality of the data. . .	9
2.3	Evaluation metrics to understand if a model is well predicted, such as TPR, FNR, TNR and FPR.	22
3.1	Description of the features of the initial dataset, as well as the type of data and feature.	25
3.2	Data Cleaning report with all the data transformations applied to the features, including the data quality report.	32
3.3	Data Cleaning report for removing the inconsistent devices, including the data quality report.	33
3.4	Comparison of the models used to calculate the start date Event.	37
3.5	An example of the structure of data used as input of the Survival model.	38
3.6	Results of employing the SMA for different temperature threshold for detecting the start of the anomalous event.	41

ACRONYMS

AD	Anomaly Detection Approach
AE	Autoencoder
AI	Artificial Intelligence
ANACOM	Autoridade Nacional de Comunicações
ASSOFT	Associação Portuguesa Software
CRISP-DM	Cross-Industry Standard Process for Data Mining
CSP	Communication Service Providers
EMA	Exponencial Moving Average
IDE	Integrated Development Environment
IF	Isolation Forest
IoT	Internet Of Things
KM	Kaplan-Meier
LSTM	Long-Short Memory Term
MAD	Median Absolute Deviation
ML	Machine Learning
OPEX	Operational Expenditure
PCA	Principal Component Analysis
PM	Predictive Maintenance
RM	Reactive Maintenance
RPCA	Robust Principal Component Analysis

ACRONYMS

SA	Survival Analysis
SAS	Statistical Analysis System
SEMMA	Sampling, Exploring, Modifying, Modeling, and Assessing
SMA	Simple Moving Average
SPSS	Statistical Package for the Social Sciences
SR	Spectral Residual
STA	Statistical Approach
STB	Set-Top Box

INTRODUCTION

*"In the long run, all machines
break down" - Maynard Keynes*

Nowadays, telecommunications companies, also known as **Communication Service Providers (CSP)** face an oppressive pressure to compete for costumers leading to a constant demand to innovate.

Indeed, **CSP** 's competed in a saturated market populated by young companies that rely on the empowerment by the new **Internet Of Things (IoT)** technologies (e.g. sensors), which provide real-time access to large scale datasets.

The access to the equipment's hardware data enabled to reduce of the number of claims. As stated by **Autoridade Nacional de Comunicações (ANACOM)** [3] the Portuguese telecommunications companies' customer claims on the first semester of 2020 were respectively 21% and 8% for customer service and equipment faults. These type of claims are critical issues since *"the network equipment reliability is one of the major critical issues for telecommunication operators, not only because it may affect the company's outlook in terms of customer service and reputation- but also because the repairing activities can be extremely expensive"*, according to Andrea from Hewlett Packard Enterprise [19].

Figure 1.1 exemplifies these client claims of the equipment hardware fault.

Let us assume, within a **CSP** client's residence, his **Set-Top Box (STB)** (that enables cable or satellite television broadcast) stopped running for an unknown reason. Therefore the client has to call the **CSP** technical support and ask for the equipment to be repaired or replaced. The technician has to reply to the customer within an urgent time frame of 48 hours according to service level agreements [1][2], including both the technician's travel time and the box repair or replacement. The technician can probably exceed the repairing time, which leads to overtime labour. For each extra hour,



Figure 1.1: A RM example: STB running for an unknown reason and the client has to call to the technical support, this unplanned intervention is more expensive than scheduled one for the company.

the company pays an additional fee of 50 euros, unlike the scheduled interventions that are 25 euros per hour as stated by ANACOM [1][2]. During that time frame, there are unsatisfied costumers without service. Consequently, this leads to two significant problems: The customer may churn, meaning he can abandon the company to a competitor and the company loses money with each new unplanned intervention are more expensive.

Currently, on the telecommunication company where this internship took place, these urgent interventions are treated with the **Reactive Maintenance (RM)** strategy [28] where the maintenance operation occurs after a failure or a break-down as the example in Figure 1.1. RM is the most straightforward method to employ and understand, with no meaningful effort to implement within companies. Still, it is too costly due to the increase in the human task force the **OPEX**.

To improve on this domain, the company needed to answer the following question: *"How long will equipment live before a failure event happens?"*. This answer would allow anticipating the faults on services and equipment's before they happen. In this sense, the **Predictive Maintenance (PM)** strategy appeared since it optimises the intelligence of the network planning and operations by using the analytical capabilities that take advantage of the mathematical algorithms of **Artificial Intelligence (AI)** and **Machine Learning (ML)**. One possible solution for employing this strategy is using a **Survival Analysis** model that estimates a Survival Curve indicating the equipment's probability of surviving at a particular time based on an anomalous event. Although these automation techniques are now beginning to emerge on the networking and service domains within telecommunication operators, PM strategies, have proven to be 35% more cost-effective than employing RM strategy, as reported by TEOCO [13].

This improvement is only possible due to the access to historical and real-time data that records the machine's process to fail. Consequently, by reinforcing the advantages mentioned before, it is possible to avoid the increase in complaints, churn of clients, profit and maintenance costs.

	Description	Location
Goal 1	Identify, understand and apply Data Science models and concepts to solve the PM problem;	Chp 2: Theoretical Framework & Chp 3: Project Development
Goal 2	Obtain an acceptable anticipation period for at least 70% of the possible urgent interventions;	Chp 3: Project Development
Goal 3	Demonstrate the solution in an uncomplicated form, using data visualization tools .	Chp 3: Project Development

Table 1.1: Identification of the company’s main three goals that goes through the theoretical, resolution and visualization for solving the **PM** problem; and their respective location on the report.

1.1 Project Goals

This project report aims to contribute to the first **Predictive Maintenance** strategy within the company using **Survival Analysis (SA)** technique to anticipate an acceptable period before the **STB** failure happens and for the technical team to arrive at the customer residence. This initiative project’s outcomes will enable (if successful) to have a long-term project based on more resources.

In order to guarantee that accomplishment, it is essential to establish with the company the main goals to fulfil by the end of the project period, that is, between March and July of 2020. Whereas this period restricts the project development to some concepts, time and available resources.

Table 1.1 describes each of the proposed three main goals, and the column *Location* indicates where each goal’s development can be found on this report.

All the milestones together pass by all the domains that a **PM** strategy should have, employing a Predictive algorithm, more precisely the **SA** technique. Each goal aims to answer the following questions, respectively: "How to do it?" describes the essential research techniques to solve the proposed problem; "How to solve it?" contains the resolution by employing the acknowledge theoretical techniques, which is only successful if it reaches that acceptance criteria referred on the Table 1.1; and finally "How to demonstrate it?" contains the visualisation and the demonstration of the obtained results using the researched techniques.

Moreover, it will be introduced an extra milestone on the Project Development Chapter 3 that is not represented on Table 1.1.

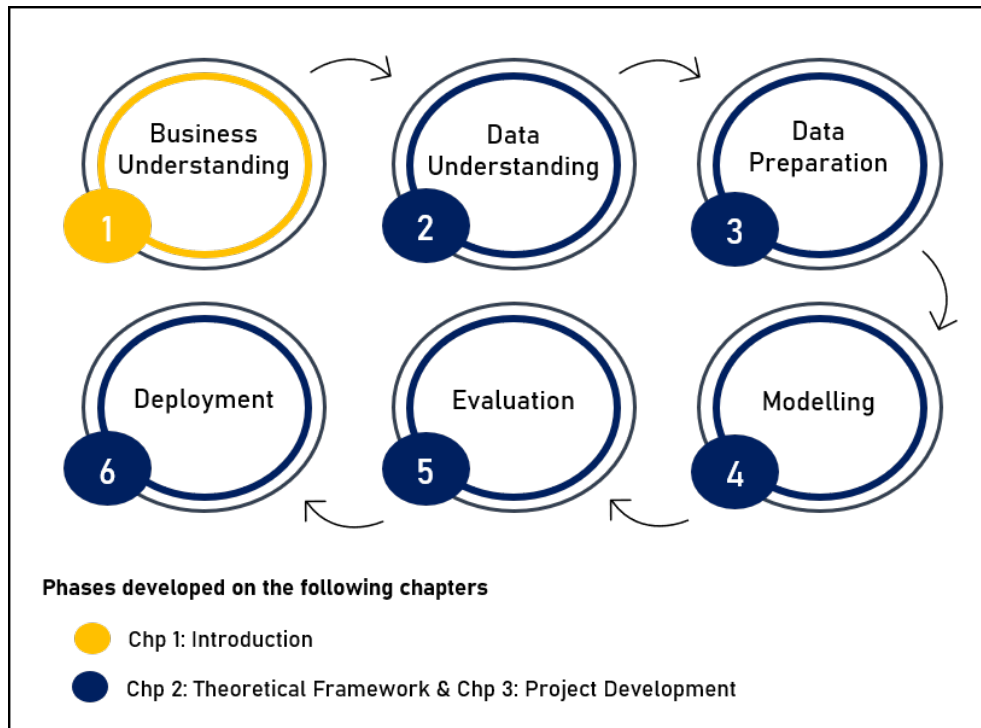


Figure 1.2: The structure of this report following the [CRISP-DM](#) methodology created by Chapman [20] divided into six different phases developed within three distinct chapters.

1.2 Structure and Methodology

The report follows a methodology containing all the six phases of a project's life cycle. Based on the [Cross-Industry Standard Process for Data Mining \(CRISP-DM\)](#) methodology, created by Chapman [20]. Founded in the year 2000 for systematising processes, it has become one of the most recognised methodologies for data science, analytics and data mining. There is another similar methodology, [Sampling, Exploring, Modifying, Modeling, and Assessing \(SEMMA\)](#) developed by [SAS Institute](#) [0]. Still, [CRISP-DM](#), proof to be the proper solution for this business problem, as shown in this [SPSS](#) software step-by-step guide [20].

This structure is displayed in [Figure 1.2](#) in order to accomplish the aforementioned goals through acceptable practices. These phases are Business Understanding, Data Understanding, Data Preparation, Modelling, Evaluation and Deployment. All these six phases except for the first are extended into the two subsequent Chapters 2 and 3, Theoretical Framework and Project Development.

At the beginning of this Chapter 1, describes the business challenge, thus addressing the first phase. The subsequent phases from flow [Figure 1.2](#), are addressed on a theoretical level in the Theoretical Framework (Chapter 2). In Project Development (Chapter 3), each phase is presented in detail in terms of applying the preferred techniques alongside with the respective results and discussion.

Finally, the Conclusions (Chapter 4) contains a brief reflection on the achievements and improvement points, the project's outcomes, and what is left as future work.

1.3 Company overview

The company associated with the project is known for delivering high-tech transactions, solutions, products, consulting, technology services, system integration and managed services. This company, linked with a significant Portuguese company of Information Technology, develops and implement technology solutions that help CSP under Digital Transformation. Their methodology basis reduces the risk in IT projects and achieves a high success rate.

The company comprises teams specialised in Business Support Systems, Operations Support Systems, IoT, Digital Television and Unified Communications. They work in different offices, such as Lisbon, Oporto, Newbury, Dubai, and Dusseldorf; It employs between 1001 to 5000 people, focuses on telecommunication services, and serves clients in more than 25 countries.

The company has a department of Analytics composed of several teams of different projects and domains, like the Data Science team. The Data Science team comprises ten people. The PM project was developed within an initiative project under the Project Manager. During the project period, the teamwork provided a foundation for collaboration and communication, crucial to achieve the proposed goals.

THEORETICAL FRAMEWORK

Theoretical Framework (Chapter 2) explains all the concepts and general information behind each phase of the Project Development (Chapter 3). Moreover, it fulfils the first goal that identifies and understands Data Science concepts and models to solve the PM problem, stated in the previous Chapter 1 in Table 1.1.

In summary, this Theoretical Framework is organised in four sections; each one is a phase of the CRISP-DM method according to the Figure 1.2 referred in the Introduction Chapter 1, being the following: Data Understanding (Section 2.1) that describes the type of data (in this case Time Series data), and all the concepts to understand the data quality; Data Preparation (Section 2.2) explains the methods to extract the relevant features and some techniques employed for constructing the data to be used on the Survival Analysis (SA) model alike Anomaly Detection models and Statistical Smoothing techniques; Modelling (Section 2.3) that describes the chosen technique to solve the Predictive Maintenance problem (Survival Analysis) and the last Section is Evaluation (Section 2.4) that describes essential metrics to evaluate the performance of the models.

2.1 Data Understanding

Data Understanding is the second phase of the CRISP-DM methodology Figure 1.2, a stage for the fundamental data exploration. The book *"Data Mining. Concepts and Technique"* [10] inspired all the concepts or techniques disclosed in this section.

The section starts by identifying the different types of analysis that are important for discovering patterns. Then, it describes the features types for understanding more enhanced the data. Then, this section will evaluate data quality problems managing to explain reasonable conclusions throughout the development. Additionally, at the

Type		Description
Nominal		Also named as categorical, represents some kind of category. Not being possible to do mathematical operations.
Ordinal		The features have an order to rank, but the magnitude between successive value is not known.
Boolean		It's a binary feature only with two categories 0 or 1.
Numeric	Ratio	It is a quantitative being the ratio of another value having the point of origin.
	Interval	Is quantitative value being a scale of equal size units ordered.

Table 2.1: Table that describes the characteristic of the different type of features stated on [10].

end of this section will focus on a particular type of data that tracks the movement of data points (historical data) applied to the **PM** problem that is Time Series data.

In Data Science, the number of variables/ features used in a study will define the type of analysis. **Univariate Analysis** utilises only one feature, employing analysis like frequency distribution tables, bar charts, histograms, frequency polygons, and pie charts. **Bivariate Analysis** examines the relationship between two features, for instance, applying correlation matrix or regression. Finally, **Multivariate Analysis** represents the relationship between three or more features, for instance, using Cluster Analysis or Principal Components Analysis.

Table 2.1 describes the different features that populate a dataset can be composed. The feature type **nominal**, **ordinal**, **binary** or **numeric** helps to understand the information that belongs to a feature.

Table 2.2 describes the meanings within different data quality terms that involve **accuracy**, **completeness**, **consistency**, **timeliness**, **believability** and **interpretability**. The quality of the data is essential because low-quality data leads to low-quality mining results. The data is exposed to noise, missing values and inconsistencies on real-world data because of the typically colossal size and the multiple and heterogeneous sources, as stated by Han, Jiawei Kamber, Micheline Pei, and Jian [10].

2.1.1 Time Series data

Time Series data is a sequence of numeric data points ordered in time. It can be distributed at fixed or random time intervals, like per minute, hour, days or months. Nonetheless, time is an independent variable. This type of data is generated naturally, such as stock markets and technical observations. In real-world applications, Time

Terms	Description
Accuracy	Avoids inaccurate attribute values, human error, equipment malfunction transmitting data, or even missing values;
Completeness	Avoids the absence of values of interest, perhaps due to unavailability of the data or disregarded historical data;
Consistency	Avoids incorrect data, essentially naming, formatting input fields or even avoid duplicated data;
Timeliness	Avoids data being incomplete after or before a certain period;
Belivability	How much the users trust the data;
Interpretability	How easy the data is understand.

Table 2.2: Table that describes the terms that accomplish the quality of the data stated on [10].

Series analysis is used for detecting trends in financial markets, analysing electricity consumption or even sales forecasting.

Time Series analysis focuses on finding different patterns in data according to BBC article [27], *"Finding patterns is extremely important. Patterns make our task simpler. Problems are easier to solve when they share patterns because we can use the same problem-solving solution wherever the pattern exists. The more patterns we can find, the easier and quicker our overall task of problem-solving will be"*. Chatfield [6] states four different patterns to characterise the time series data t in:

1. **Trend, $T(t)$:** This pattern exists when there is a tendency to increase, decrease, or be static over a long period on a Time Series. For example, the number of houses in Lisbon over time can show an upward trend.
2. **Cyclical, $C(t)$** This pattern exists when data shows ups and downs that are not of a fixed period. For example, business cycles can last several years, and it is unknown the length of the cycle.
3. **Seasonal, $S(t)$:** This pattern exists when a Time Series is influenced by a fixed and known period. For instance, a month, day of the week, parts of the day, the weather season or traditional holidays. For example, online sales increase during Christmas before decreasing again, or electricity consumption is higher during the day and lower during the night.
4. **Irregular or random variations, $I(t)$:** This pattern exists when there is an unpredictable event, without recurrence. For example, an earthquake, war or anomaly events.

Time Series are non-deterministic; this means that we can not predict with certainty what happens in the future. However, they follow some regular pattern in the long term, so they are suitable for applying time series forecasting methods.

2.2 Data Preparation

Data Preparation is the third phase of **CRISP-DM** methodology (Figure 1.2), and it consists of transforming data to be perceptible and reformatting all the data towards the Model Phase.

This section will be divided into three parts. **Data Cleaning** (Section 2.2.1) that includes techniques for extracting relevant features. The **Data Construction: Statistical** (Section 2.2.2) and **Data Construction: Anomaly Detection** (Section 2.2.3) both include techniques for detecting an important feature that is the beginning of an anomalous period. This is a key feature in order to construct the dataset to apply the **Survival Analysis** technique on Section 2.3.

2.2.1 Data Cleaning

Redundancy is an important issue in an enormous number of datasets, which implies an inconsistent dataset. The technique used in this kind of situations is correlation analysis. Correlation evaluates given two variables, how strong one attribute implies the other, it can be used the following two types of analysis, and for different types of features (Table 2.1) [10]:

- **Cramér's V:** Created by Harald Cramér, is used for nominal data. It measures the relationship between two variables, between an interval from 0 to 1. If Cramér's V's (ϕ_c) value is close to 0, the variables do not have an association between them; otherwise, it means the two variables have a strong association. This value is based on Pearson Correlation's chi-squared test derives the value χ^2 , where N is the sample size, and k is the lesser number of categories between the two variables. The following Equation (2.1) gives this association value:

$$\phi_c = \sqrt{\frac{\chi^2}{N(k-1)}} \quad (2.1)$$

- **Correlation coefficient:** It is used for numeric data, that evaluates the relationship between two variables. Despite, existing other methods the Pearson's was the chosen technique for computing the correlation coefficient since it is the most used. Pearson's coefficient measure for linear relationship as seen in the following Equation (2.2):

$$r = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 (y_i - \bar{y})^2}} \quad (2.2)$$

$$\begin{cases} r = -1 & , X \text{ and } Y \text{ are strongly negatively correlated} \\ r = 0 & , X \text{ and } Y \text{ are independent} \\ r = 1 & , X \text{ and } Y \text{ strongly positive correlated} \end{cases} \quad (2.3)$$

Where, $X = (x_1, x_2, \dots, x_n)$ and $Y = (y_1, y_2, \dots, y_n)$ are the two variables, being \bar{x} and \bar{y} the respective mean values of X and Y . The correlation r , will give a value between -1 and 1 , as demonstrated by the system Equation (2.3).

2.2.2 Data Construction: Statistical

Smoothing techniques are popular approaches used to better understand the patterns on data; they smooth the effect of random variations of the time series, outlining the anomalies or abnormal patterns and highlighting long-term ones. Mainly, it a simple technique that only needs a threshold to define the anomaly. There are several techniques mentioned by Chatfield in [6] like [Simple Moving Average \(SMA\)](#), [Exponential Moving Average \(EMA\)](#), [Weighted Moving Average](#) or [Cumulative Moving Average](#), regardless, only the most known technique will be explained:

- [Simple Moving Average \(SMA\)](#), is an arithmetic moving average that generates a series of averages of distinct subsets of the general data, being a rolling mean by moving a period of t , as shown by the formula:

$$SMA_t = \frac{p_{t+1} + p_{t+2} + \dots + p_{t+n}}{n} \quad (2.4)$$

So for each time t of a sequence $P = (p_1, \dots, p_m)$ of m elements, where $n < m$ and $n, m \in \mathbb{N}$, it calculates the average of the last n observations, being n the number of periods to be averaged. An abnormal event here is when a value exceeds a predefined threshold.

- [Exponential Moving Average \(EMA\)](#), is a type of the moving average that gives more weight to the most recent data points from the time series sequence, instead of what SMA does, as shown by the formula:

$$EMA_t = \alpha p_{t+1} + \alpha(1 - \alpha)p_{t+2} + \dots + \alpha(1 - \alpha)^{t+n-1} p_{t+n-1} + (1 - \alpha)^{t+n} p_{t+n} \quad (2.5)$$

where $\alpha = \frac{2}{N + 1}$

So for each time t of a sequence $P = (p_1, \dots, p_m)$ of m elements, where $n < m$ and $n, m \in \mathbb{N}$, calculates the average of the last n exponential observations, being n the number of periods to be averaged and $N \in \mathbb{N}^*$. An abnormal event here, is when a value exceeds a predefined threshold.

2.2.3 Data Construction: Anomaly Detection

Anomaly Detection identifies unusual behaviour in a given set, known as outliers, exceptions, surprises, or even contaminants [0]. This technique is massive due to anomalies translating critical information for various applications, for example, an abnormal traffic pattern in a computer network; this could mean that a hacked computer is sending sensitive data to an unauthorised destination.

However, there are some challenges of applying this technique according to Chandola [0] that are: the availability of labelled data for training the models; removing noisy data that can mask real anomalies and the difficulty of creating a precise boundary for the anomalies.

Another important aspect is how anomalies are analysed and reported; there are two known techniques referred to by Chandola. The **scores** being the most common technique, where each instance of the test data has assigned an anomaly score, the score depends on how much that value is considered an anomaly. The output of such a technique is a sorted list of anomalies. Then it is possible to select the top anomalies or to use a cut-off threshold to select them. Finally, **labels** technique that assigns a label as "normal" or "abnormal" to each instance in the test data, usually represented by a binary flag.

In the following sections, only unsupervised models will be mentioned, without knowing a priori which data points are anomalies.

2.2.3.1 Median Absolute Deviation

Median Absolute Deviation is a Robust Z-Score that, instead of using mean and standard deviation like the Z-Score model [16], uses median. The mean value is a non-robust statistic and consequently highly influenced by outliers. Therefore, the necessity of creating a robust version of the standard Z-Score using median emerged. **MAD** facilitates a more consistent measure of central tendency of a time series with a high number of anomalies or outliers, defined as follows:

$$MAD = med|x_i - \tilde{x}| \quad (2.6)$$

Median Absolute Deviation (MAD) it is the median of the absolute difference between each point x and the median of the raw data \tilde{x} , knowing this value, it is possible to estimate the standard deviation, that is:

$$\hat{\sigma} = b \times MAD \quad (2.7)$$

On the previous equation b is a constant that is multiplied by the value **MAD** for approximation to the standard deviation value, with a normal distributed data. For $MAD \neq 0$, the constant $b \approx 1.4826$ so the robust Z-Score can be defined as follows:

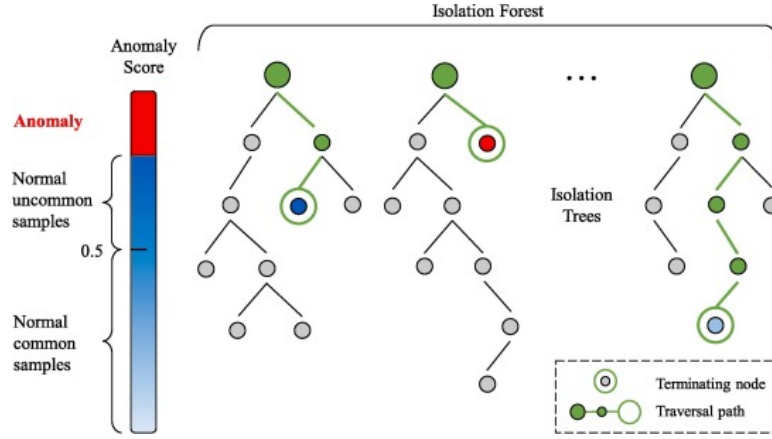


Figure 2.1: Representation of the **Isolation Forest** anomaly detection algorithm retrieved from [18].

$$M_i = cx_i - \bar{x}b \times MAD \quad (2.8)$$

The robust Z-score is calculated for each point of the data sample. On Anomaly Detection, a cut-off parameter represents how many standard deviations are required if the robust Z-score has a value more significant than the cut-of for each point is considered an anomaly as stated on [16].

2.2.3.2 Isolation Forest

Isolation Forest (IF) can also be called iForest based on the decision tree algorithm's principle being an unsupervised algorithm. The word isolation means "to be separated from the rest" since anomalies are rare and different, they are more susceptible to isolation.

This IF algorithm as illustrated in Figure 2.1 [18] creates an ensemble of Isolation Trees - a collection of vertices and edges where any two vertices are connected by one path. This Isolation Trees use sub-samples of the dataset, and trees are created recursively partitioning the data until instances are isolated. There are two parameters at this stage: the **sub-sampling size** that controls the data size for the trees, and the **number of trees** that controls the ensemble size. A forest joins several trees in the same graph.

The anomalies score is the expected traversal path length for each tree; each path length is obtained by counting the root's paths until the terminating node. Therefore, the anomalies will be the points with a shorter path in the tree, represented with red colour.

According to F. Tony Liu, K. Ming Ting, and Z.-H. Zhou [24] IF performance converges quickly with a small number of trees and requires just a tiny sub-sampling size to achieve high detection performance with high efficiency. For big dimensional problems with many irrelevant attributes, IF can be efficient and can have the lowest

computational complexity than other models. On the other hand, Isolation Forest can "mask" anomalies.

Isolation Forest can have time-series data as stated in this article written by Y. W. Liu and Lei [18], which proposes an algorithm based on sequential data. Summarily, it divides the time series data into several sub-sequences according to a sliding window, which will be the algorithm's third parameter. For each sub-sequence, the **IF** algorithm is applied. The main challenge is defining this parameter.

2.2.3.3 Spectral Residual

Spectral Residual developed by Hansheng Ren from Microsoft [23], is an unsupervised algorithm suitable for Anomaly Detection in a univariate time series, based on Fast Fourier Transform. This method based on a human visual system's ability to detect visual saliency by analysing the log spectral obtaining the spectral residual.

The **SR** algorithm consists of the Fourier Transformation to get the log amplitude spectrum than calculates the Spectral Residual and the Inverse Fourier Transform that transforms the sequence the spatial domain. This sequence is called the saliency map $S(x)$.

The saliency map $S(x)$ can identify anomalies based on a threshold of Γ , as shown in Equation 2.9. The anomaly score is the relative difference between the saliency map values and their moving averages. If the score is above the threshold Γ the value is flag as an anomaly.

$$score = \begin{cases} 1, & \text{if } \frac{S(x_i) - \overline{S(x_i)}}{S(x_i)} > \Gamma \\ 0, & \text{otherwise} \end{cases} \quad (2.9)$$

The x_i is an arbitrary point of the sequence x .

Spectral Residual has four parameters to tune, the sliding window, the Anomaly Detection threshold, the k of the kernel convolution and finally z , which is the preceding points of $S(x_i)$.

Based on the article written by Hansheng Ren [23], **SR** is used at Microsoft to monitor millions of metrics coming from Bing, Office and Azure, such as page views and reviews in real-time, in order to solve faster the issues on those sites. Microsoft teams saved manual effort and accelerated the process of diagnosis. This algorithm had an exceptional performance and robustness in detecting anomalies compared with other Anomaly Detection models, SPOT, DSPOT, DONUT, FFT, Twitter-AD and Luminol.

2.2.3.4 Robust Principal Component Analysis

Robust Principal Component Analysis (RPCA), created by Zhou [32][4], is an improvement of the statistical method **Principal Component Analysis (PCA)**, but is not overly sensitive to anomalies; in other words, it is robust to outliers.

The classical **PCA** finds the data's maximal variance, applied in data analysis and dimensional reduction. **PCA** linearly projects into a lower-dimensional space and separates the signal from the noise. A point far away from the rest of the data is considered an anomaly. **PCA** is highly sensitive to perturbations; a final point can change the projection's orientation. This way, it can mask the anomalies since eigenvectors and eigenvalues are estimated from the sample covariance matrix, highly sensitive to outliers.

Because of this, Zhou [4] proposed **RPCA** for the minimisation of the impact of masking the anomalies. Decomposes the original data into the accurate data of M , the low-rank matrix L , into a sparse matrix S that contains the noisy data and finally into a random noise matrix W . Demonstrated in the following equation:

$$M = L + S + W \quad (2.10)$$

According to Zhou [32][4] to recover the low-rank matrix L from a high dimensional data matrix M with several errors, it is computed the convex program named Principal Component Pursuit (PCP) that considers M with the Equation (2.10). Unlike **PCA**, for the matrix L , it is unknown the low-dimension column and the row space, and for matrix S it is unknown the place and the amount of non-zero entries. To recover both matrices, PCP solves the following optimisation problem (Equation 2.11) using an Augmented Lagrange Multiplier (ALM) to more details check the Article [32].

$$\begin{aligned} \min_{L, S} \quad & \|L\|_* + \lambda \|S\|_1 \\ \text{subject to} \quad & L + S = M \end{aligned} \quad (2.11)$$

The most important details from the **RPCA** are the choice of μ and the stopping criterion to not converge. For the first one it's applied $\mu = \frac{n_1 n_2}{4\|M\|_1}$ (the number 4 can change, it's just what they used in Zhou's paper) and the second the algorithm ends when $\|M - L - S\|_F \leq \delta \|M\|_F$ with $\delta = 10^{-7}$.

Netflix [14] handled with a high cardinality of data and implemented this algorithm to detect anomalies. For example, the process of millions of transactions happening every day across several banking institutions in real-time, using batch environments, to detect anomalies in the payment failure. This way, **RPCA**, helped business managers follow up and helped Netflix teams to understand and react faster to anomalies. Although dealing with a high cardinality of data, **RPCA** also minimises the mask anomalies, deals with seasonality and finally, data that is not always normally distributed.

2.2.3.5 Autoencoder

Autoencoder (AE) based the book written by Goodfellow [9] is an unsupervised learning algorithm, a particular case of the feed-forward neural network, a non-recurrent

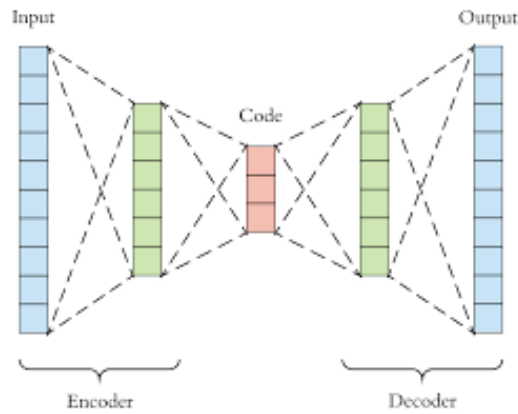


Figure 2.2: Representation of the **Autoencoder** model, the first two layers represent the encoder, the red layer the code and the last two layers represents the decoder.

neural network. An illustration of this algorithm is shown in Figure 2.2. This neural network algorithm aims to learn a representation of the data by training the data to copy the input to output (the output and input layer have the same number of nodes) approximating the input data since the model has to learn the essential aspects of the input data and not all its features of the input data being the main goal to minimize the reconstruction error. The architecture of an Autoencoder is composed by three parts as shown in Figure 2.2:

1. **Encoder** function: $h = f(x) = f(W_{xh}x + b_{xh})$.
 h is the internally hidden layer, also known as **code**, W is the weight and b the bias;
2. **Decoder** function that produces the reconstruction: $r = g(h) = g(f(x)) = f(W_{hx}h + b_{hx}) = x$, the output layer;
3. **Reconstruction Error (RE)**: $\|x - r\|$.

In summary, the goal of an **AE** is to map an input x to an output, the reconstruction r , through an internal representation, h . Where the function f maps x to h and the function g maps h to r .

The **Autoencoder** learns to minimise the **RE** for Anomaly Detection; this **RE** value is seen as the anomaly score. Thus data points with a high value of RE are considered anomalies. The following Algorithm 1 summarises how this is applied in Anomaly Detection.

Autoencoder initially was created for dimensional reduction and feature selection; now, they are used in Anomaly Detection. One application of **AE**'s that supports sequence data as input data is **Long-Short Memory Term (LSTM) AE**, which can learn a compressed representation of temporal data and an internal memory to remember.

Typically, this encoder layer's architecture is an **LSTM** model that reads the input sequence (that can be of more than one layer). This layer's output represents the hidden

Algorithm 1 AE on Anomaly Detection

Init: Anomalous dataset x , threshold α ;
For $i = 1$ **to** N :
 1. $RE(i) = ||x(i) - g(f(x(i)))||$;
 2. **If** $RE(i) > \alpha$: $x(i)$ anomaly **else** $x(i)$ not anomaly;
end if, end for
output: Reconstruction error $||x - r||$.

layer that is the internal learned representation of the entire input sequence, that can be a *repeat_vector* of a fixed length. This layer is the input of the LSTM decoder model layer that generates the output sequences. The *dropout* layers help prevent overfitting (happens when a model learns the detail and noise in the training data). A suggestion of this type of architecture is given on the Keras documentation [21], which is the one used in Project Development (Section 3.3).

There are other AE models used in Anomaly Detection, such as the Robust Deep Auto Encoder [31], DONUT [29], or any other state of the art algorithms referred on [30].

2.3 Modelling - Survival Analysis

The modelling phase is the fourth step on the CRISP-DM methodology (Figure 1.2), and it refers to the modelling technique selected to approach the Predictive Maintenance (PM) problem, which is, Survival Analysis (SA).

According to the dictionary, to survive is the "act or fact of surviving, especially under adverse or unusual circumstances". Although surviving is a property of living beings, it can be applied to other contexts like things that have beginnings, transformations, and then deaths [17]. In this light, we can talk about the life span of a device or a customer churn (the client abandons the company and moves to the competitor).

Survival Analysis (SA) is a collection of statistical techniques that have as the outcome: the time until an event occurs or the length of time until the change of an event's occurrence. For example, from healthy to sick or from normal behaviour to irregular behaviour.

The Survival Analysis technique was previously applied for solving this PM problem [25]. Nonetheless, other techniques could also be used for the same purpose, namely, Logistic Regression. Logistic regression can be applied to estimate the probability of experience a particular event within a limited period. However, it does not consider when the fault event occurred; therefore, it ignores the length of the survival process [17].

There is no supervision on learning the machine learning problem in unsupervised learning, which means that outcomes are not labelled. Supervised learning is the opposite; there is supervision on the learning model where it contains labelled data.

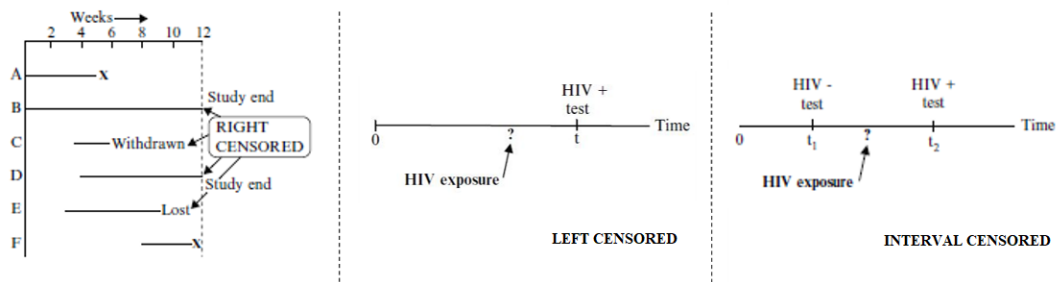


Figure 2.3: Types of censored data, extracted from the book written by Kleinbaum and Klein, that are, Right-censored, Left-censored and Interval-censored respectively [15].

Semi-Supervised contains labelled and unlabeled examples used while learning the model.

For constructing the data for any SA model, the dataset structure must contain two primary features: The **time event process** and **censoring**.

In order to understand **Survival Analysis**'s critical concepts, two use cases will be described referred on [15]. (1) A study that follows leukaemia patients in remission over several weeks to see how long they stay on that status. (2) The duration of patients surviving after receiving a heart transplant.

The **time event process** is the time until the event occurs. On the scope of previous examples: (1) The event is going out of remission, and time is the period in weeks until an individual goes out of remission. (2) The event is death, and time is the period until death.

The **censoring** occurs when there is no knowledge about the survival time, therefore no event. On the scope of the previous examples: (1) The patient's survival time is censored if the study ends while the patient is still in remission. So at that time, the event did not occur or will happen after the study ends. (2) Censored if the patient did not die when the study ends. A flag typically represents **censoring**, 1 for the event that happened (not censored) and 0 for the opposite (censored).

Furthermore, there are several **censoring** types, the right, left, and interval censoring, as illustrated in Figure 2.3 retrieve from [15].

- **Right-Censored:** The right side of Figure 2.3 illustrates a right-censored example. The y-axis is the events A to F, and the x-axis is the event process's time. At some point in time, some events are still in the study after week 12. So it is unknown how long these events will last. These observations are right-censored, being the events B, C, D and E. So the failure, in this case, will occur after the recorded time.
- **Left-Censored:** The centre of Figure 2.3, regarding a left-censored example, illustrates the follow up of an individual until he becomes HIV positive. In this case, the study ends at time t , and it is known that the individual tested positive.

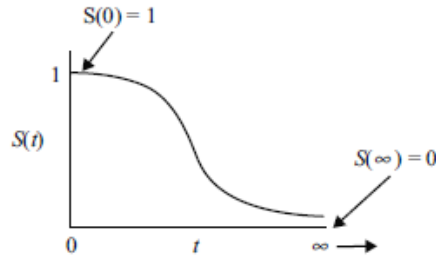


Figure 2.4: Representation of a Survival Curve, that defines the survival function $S(t)$ on the y-axis, and time t on the x-axis that starts from 0 to infinity. Figure taken from [15]

However, it is unknown exactly when the first exposure to HIV occurred. These cases are named left-censored observations when the failure happened before a particular time and without knowing when.

- **Interval-Censored:** In the left side of Figure 2.3, an HIV interval-censored example is displayed. In this case, it is unknown when the individual was exposed for the first time, but it is known within some time interval. Between his negative HIV test on time t_1 and his positive HIV test on time t_2 . These observations are interval-censored. So it is known within a period when the failure occurred.

To detect the beginning of the **time event process** (time until the abnormal event) on this project, two approaches were considered **Statistical Approach (STA)** and **Anomaly Detection Approach (AD)**, already mentioned in the past two Sections 2.2.2 and 2.2.3. The implementation of these techniques will be explained in more detail in the next Chapter 3.

Survival Function is also an important concept in SA, denoted by $S(t)$ where t is a specific value of interest from a random variable T that is the survival time, with $T \geq 0$. $S(t)$ gives the probability of a random variable T to survive longer than some specified time t , represented by Equation 2.12.

$$S(t) = P(T \geq t) = 1 - P(T \leq t) \quad (2.12)$$

The representation of a Survival Curve in Figure 2.4 defines the survival function $S(t)$ on the y-axis, and time t is the x-axis goes from 0 to infinity. Accordingly, survival function decreases when t increases, which means at the start of the curve, there is a higher surviving probability on that instance than at the end of the curve, which tends to zero. Notwithstanding, there is no smooth curve in real-world data like Figure 2.4, and the time may not converge to zero.

The **Hazard Function** denoted by $h(t)$, gives the instantaneous rate of failure at time t , and is mathematically defined as:

$$h(t) = \lim_{\Delta t \rightarrow 0} \frac{P(t \leq T < t + \Delta t | T \geq t)}{\Delta t} \quad (2.13)$$

So $h(t)$ represents the hazard rate as the ratio of the conditional probability of experiencing a particular event at time t given the condition $T \geq t$ over a small interval of time Δt .

This function gives the opposite information given by the survival function $S(t)$, and the formula gives a rate between 0 and infinity. This function is always non-negative, and the curve does not need to start at value one and go down to zero. Instead, it can start anywhere and go up and down. Consequently, this survival function $S(t)$ is more appealing for analysing the survival data, focusing on surviving. The hazard function $h(t)$ measures instantaneous potential or can identify a specific model that focuses on failing. There is a relationship between the two functions. From one, it is possible to derive the other one.

There are several state-of-the-art models like [Kaplan-Meier \(KM\)](#), Nelson Aalen and Cox Proportional Hazard. Or more complex models like Neural Multi-Task Logistic Regression [25], Random Survival Forest [12], or Linear Support Vector Machines for SA [8]. Although the only one that will be used is [Kaplan-Meier \(KM\)](#).

2.3.1 Kaplan-Meier

The [Kaplan-Meier \(KM\)](#) estimator, also known as a product-limit method, is a [SA](#) model that calculates the survival function $S(t)$. [KM](#) is a univariate and non-parametric model used to estimate a Survival Curve of the population, based on the independence between **censoring** and the real survival times.

For a sample of n individuals, there are n survival times until the event; these survival times can be ordered by a rank $t_1 \leq t_2 \leq \dots \leq t_n$, where t_i represents the time at which the individual i experiences an event. Since t_i individuals with censored time smaller than t_i have already exited, the remaining survivors are exposed to the risk event at time t_i , the n_i . On [KM](#), the term *tied observation time* refers to the existence of many events to rank, and some subjects may share the same survival time. The following Equation 2.14 is the survival function estimator that gives the probability of living longer than t .

$$\widehat{S} = \prod_{i:t_i \leq t} \left(1 - \frac{d_i}{n_i}\right) \quad (2.14)$$

The variable d_i represents the number of events at the time t_i , considering the tied observations and n_i being the population at risk just before t_i .

Furthermore, it is possible to find the 95% Confidence Intervals with the following Equation 2.15 that gives the variance of an estimated [KM](#) survival probability.

$$\widehat{S}_{KM}(t) \pm 1.96 \sqrt{\text{Var} \left[\widehat{S}_{KM}(t) \right]} \quad (2.15)$$

The $\text{Var} \left[\widehat{S}_{KM}(t) \right]$ denotes the variance of [KM](#) survival estimate at time t , the most common approach to calculate this variance is according with Greenwood's formula:

$$Var [\widehat{S}_{KM}(t)] = \widehat{S}_{KM}(t)^2 \sum_{f:t_{(f)} \leq t} \left[\frac{m_f}{n_f(n_f - m_f)} \right] \quad (2.16)$$

Where $t_{(f)}$ is the f -ordered failure time, m_f is the number of failures at $t_{(f)}$ and n_f is the number in the risk set at $t_{(f)}$.

Kaplan-Meier (KM) is a state of the art model of **SA**. However, it is mainly descriptive; it only uses one variable. On the other hand, it describes a Survival Curve, it compares two study populations, it is intuitive, and it is a low complex model according to X. Liu [17].

2.4 Evaluation

The evaluation phase evaluates and understands how well the model performed to achieve the third goal from Table 1.1.

A metric to evaluate and compare the Survival Curves obtained by the **KM** estimator is the Log-Rank Test. The Log-Rank Test is suitable for non-parametric models alike **KM**. This test deals with Right-Censored problems, evaluating if the Survival Curve of two or more groups is statistically equivalent distributed. The summary of this test is displayed in Equation 2.17.

$$\begin{aligned} H_0 : & \text{No difference between Survival Curves.} \\ \text{Log-Rank statistic} & \sim \chi^2 \text{ with } G - 1 \text{ df under } H_0 \end{aligned} \quad (2.17)$$

The test statistic is approximately a chi-square (χ^2) in large samples with $G-1$ degrees of freedom (df), where G denotes the number of groups being compared. Under the null hypothesis, H_0 if there is no overall difference between the Survival Curves, under this H_0 the log-rank is approximately chi-squared with one degree of freedom, the p-value is determined from the chi-squared distribution tables. Other alternative tests used for evaluating the Survival Curves are Wilcoxon, Tarone-Ware, Peto and Fleming-Harrington [17].

With features and **censoring**, it is possible to evaluate the survival performance using the test set (data that is not used for fitting the model) to calculate some known metrics referred on Table 2.3 taken from the book [10].

This Table 2.3 mentions four important terms that are normally represented on a confusion matrix (visualization tool) and are used to calculate each one of the metrics shown on the Table 2.3, and can be defined as follow:

- **True Positive (TP)**: The number of predictions classify as positive that were correctly classified;
- **True Negative (TN)**: The number of predictions classify as negative that were correctly classified;

Measure	Formula
True Positive Rate (TPR)	$\frac{TP}{TP + FN}$
False Negative Rate (FNR)	$\frac{FN}{FN + TP} = 1 - TPR$
True Negative Rate (TNR)	$\frac{TN}{TN + FP}$
False Positive Rate (FPR)	$\frac{FP}{FP + TN} = 1 - TNR$

Table 2.3: Evaluation metrics where TP, TN, FP, P, N refer to the number of true positive, true negative, false positive, positive, and negative samples, respectively. Adapted from "Data Mining. Concepts and Technique" [10].

- **False Positive (FP):** The number of negative predictions mislabeled classify as positive;
- **False Negative (FN):** Finally, the number of positive predictions mislabeled classify as negative.

The metrics from Table 2.3 help understanding if a model is well predicted according to real labels.

PROJECT DEVELOPMENT

As stated in the Introduction Chapter 1, the main problem is to develop the first **Predictive Maintenance** application within the company.

At this stage, the first goal from Table 1.1 is depicted on Theoretical Framework Chapter 2, which answered the question "How to do it?". Consequently, the second goal applies the aforementioned techniques to reduce at least 70% of the emergency interventions on the **Set-Top Box** with maximum anticipation. Moreover, this chapter will also include the third goal's outcomes by demonstrating the achieved solution through a dashboard.

According to the previous chapters, the project is divided into six stages, alike **CRISP-DM** methodology. Therefore each section of this Chapter 3 will describe the phases between 2 and 5. Respectively, the Data Understanding in Section 3.1, the Data Preparation in Section 3.2, the Data Modelling and Evaluation in Section 3.3 and finally Deployment in Section 3.4.

Considering that the project duration was between March and June of 2020, the different phases were distributed along the timeline. Figure 3.1 illustrates the expected timeline (blue shade) distributed within the different phases alongside the respective duration, compared with the real timeline (yellow shade).

By relating the two timelines, a delay between both is apparent. Because the company integrates a real-world industry, it is unlikely to predict sudden situations; for instance, there was one week of delay at the beginning of the project since the company did not have the data available to start the data analysis. Besides, in June, there was another goal proposed (that is explained on the last Section 3.4 Deployment), so the delivery was extended until the end of July, a three weeks delay. However, some other constraints could also delay the project, such as data confidentiality, the data set's size,

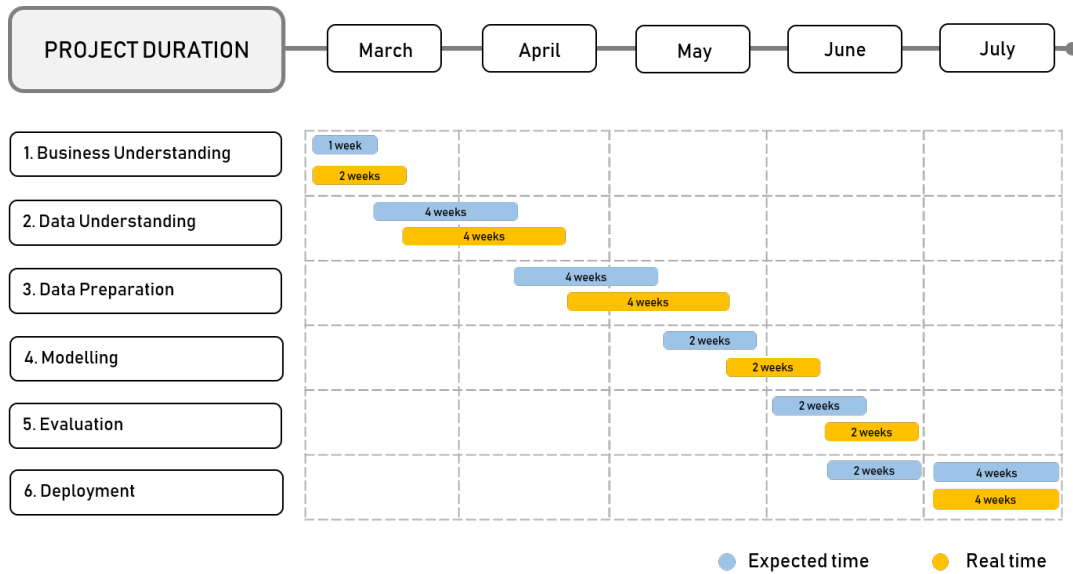


Figure 3.1: The project timeline with the expected and the real time represented in weeks.

and the low computing resources. In the last chapter of this report, these problems are described in more detail.

The project aligned to the project duration was developed in the Pycharm IDE (an integrated development environment used in computer programming), employing the well known Python language open-source, easy to implement and gives the possibility of being creative. For this project pipeline, which is discussed in more detail in Appendix A, the main Python tools used were:

- Pandas and Numpy for Data Manipulation;
- Scikit-Learn for Machine Learning concepts, feature selection, and preprocessing;
- Keras for Deep Learning model;
- PySurvival for SA;
- Dash, Matplotlib and Seaborn for Visualization.

Finally, for saving the several versions of the project, it was used GIT, which is a version control system that tracks and records all the decisions made along with the project development. GIT also allows reverting to previous specific versions.

3.1 Data Understanding

Data Understanding is the second stage of the workflow in Figure 3.1, being one of the most prominent. This section is divided into Data Description (Section 3.1.1) and

Short Name	Feature Name	Description	Data Type	Feature Type
Active	Active Device	It is true if the device is active at the data extraction time.	Integer	Boolean
Active Date	Latest Active device date	The last record date when the device was active.	Datetime	Interval
AMUSE	App Memory Usage	Last registered record of memory usage from the app in megabytes.	Integer	Interval
BN	Brand Name	The brand of the device.	Object	Nominal
CPULV	CPU Percentage Level	Last registered device's CPU percentage.	Integer	Ratio
CPUTp	CPU Temperature	Last registered device's temperature level in celsius.	Integer	Interval
Device	Device ID	Device identification.	Integer	Nominal
DMFRE	Device memory free	Last registered device's memory available.	Integer	Interval
DMUSE	Device memory used	Last registered device's memory usage.	Integer	Interval
Household	Household ID	Client Identification with at least one device.	Integer	Nominal
HW	Hardware Version	Last registered device's hardware version.	Object	Ordinal
MN	Model Name	The name of the model device.	Object	Nominal
PSmode	Power Saving Mode	Power saving mode configured on device.	Object	Nominal
STBmode	STB Mode	Actual STB mode.	Object	Nominal
SW	Software Version	Last registered device software version.	Object	Nominal
Time	Timestamp	The timestamp the information of the box is received.	Datetime	Interval

Table 3.1: Description of the features of the initial dataset, as well as the type of data and feature.

Data Exploration (Section 3.1.2). The first goes through collecting and describing the initial data, and the second explores important findings or patterns.

Due to the IoT technologies, various sensors on the Set-Top Box generate data sent to a data centre, so it is possible to obtain information about the hardware and software components. Therefore, the extracted subset originates from this information. As an essential remark, since we are dealing with operational data, it was necessary to declare the data's privacy by masking the real values with percentage values.

The principal demand for this project was to obtain all the historical data of the equipments. Unluckily, the data centre did not have the information starting from the device's first event. It was only possible to have data between August 2019 and March 2020, which comprises eight months of historical data.

The following demand implied the extraction of features; concerning that, it was necessary to discuss with the business specialists to understand the most impacted features to prevent faults in STBs. Hence, the selected features extracted from the data centre were the hardware features since the machine elements are essential to determine the STB lifetime.

Due to the project's dimension and limited resources, the extracted data only contains a few variables concerning equipment's information.

3.1.1 Data Description

The collected data resulted on Table 3.1, that includes the **Short Name** referred to the **Feature Name** and the respective **Description**. Additionally, the columns **Data Type** and **Feature Type** describes extra information about the features (concepts detailed earlier in Table 2.1).

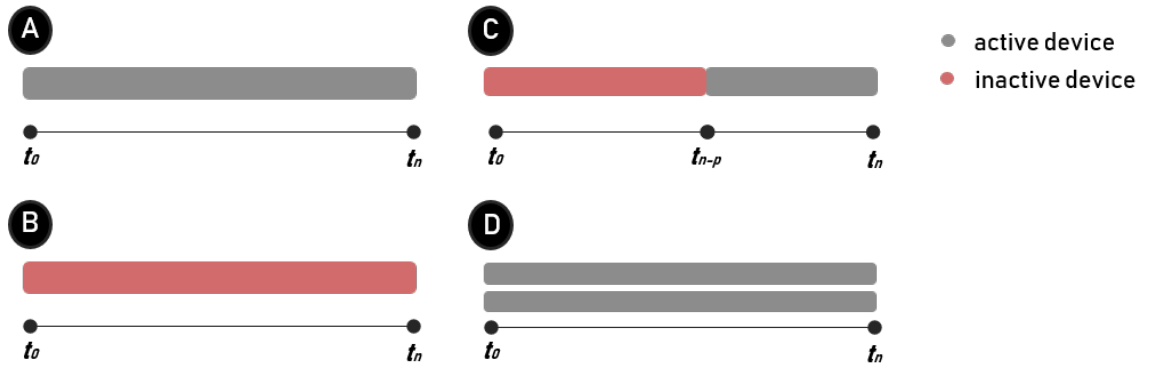


Figure 3.2: Each client historical data can be associated to one of A, B, C or D scenario.

Before advancing to the subsequent section, it is inevitable to establish general knowledge about the data. Observing Table 3.1, a feature titled *Time*, manifests of being a time series problem (described in Section 2.1.1), where every equipment owns one time series. Consequently, a dataset with multiple devices is a dataset holding multiple time series.

Each client can be associated with one of the following four distinct scenarios illustrated in Figure 3.2. The scenarios were based on the information from features *Active*, *Active Date*, *Device* and *Household*. Regarding the feature *Household*, the client identification is associated with one or more devices id (feature *Device*). For simplicity, we only consider the clients with one or two devices associated. The feature *Active* registers if the device is active at the data extraction moment or not, and the last registered active date is referred on feature *Active Date*.

- **Scenario A:** At the moment of data extraction, a collection of households were using one active device.
- **Scenario B:** At the moment of data extraction, a collection of households has an inactive device associated.
- **Scenario C:** At the moment of data extraction, a collection of households had two devices associated. One stopped being active at the instance t_{n-p} . The other started being active at that instance, indicating the household only has one active device and one inactive device associated.
- **Scenario D:** At the moment of data extraction, a collection of households had two devices associated, both active at the same time with different historical data, indicating the household has two active devices associated.

The initial subset contains around 72% of the clients holding two devices and 28%, holding only one device. From this 72% of the clients, all were assigned to the scenario C. This phenomenon can declare a possible circumstance for having a changed of



Figure 3.3: Correlation matrix that expresses the relationship between the numerical features employing Pearson coefficient.

boxes at the instance t_{n-p} . However, there is a lack of validation from the business specialists for the reason of those circumstances—Arose the assumption of the equipment breakdown and therefore the replacement by new equipment on that instance t_{n-p} . On the other hand, from the 28% with a single device, all were assigned the scenario **A**, they are all active boxes, assuming those are the "good" devices. Therefore, there are two different data groups, the subset of clients with two different boxes associated, one active and another inactive and the subset with only one active equipment. Clients from scenario **B** and **D** were irrelevant for the problem solution.

3.1.2 Data Exploration

The data exploration detects unnecessary variables, recognise outliers, missing values, noisy data, relationships among variables, and other patterns. This stage is essential for the Data Preparation phase (Section 3.2); considering that the knowledge behind the exploration supports the choices for refining the dataset, only some critical findings will be explained throughout this section.

It was employed the Pearson coefficient method Equation 2.2, to analyse the correlation between all the numerical features, for which the results are displayed in the matrix in Figure 3.3. This matrix reveals a strong negative correlation between the feature *DMFRE* and *DMUSE* of value 1. The coefficient value suggests those features are inversely proportional; consequently, both give the same information. Due to the names of the features, this relationship is evident. Therefore in the remaining analysis, the feature *DMUSE* will not be considered.

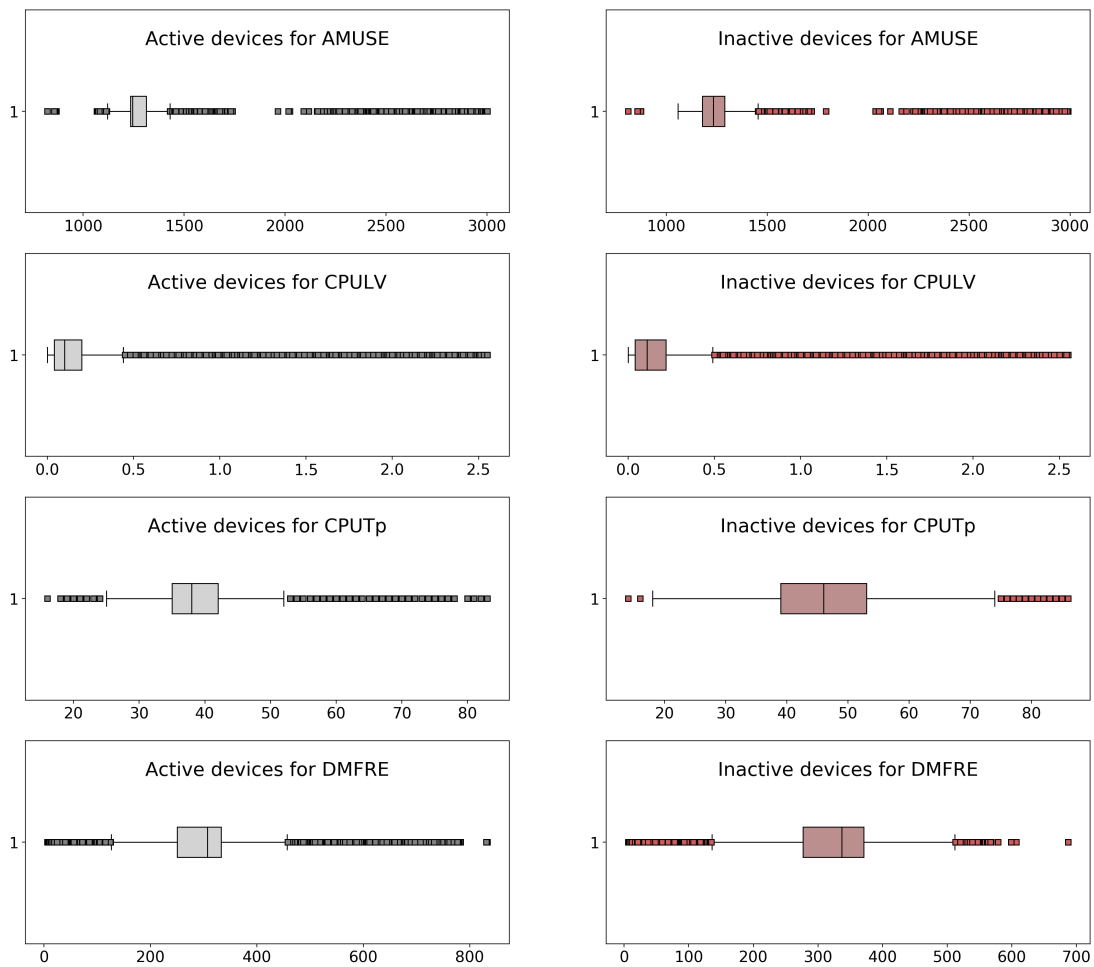


Figure 3.4: On the right represents a box plot for each numerical feature for the inactive devices. On the left represents a box plot for each numerical feature for the active devices.

Regarding the conclusion of the last section, where the devices can only be active or inactive, a comparison using boxplots for each numerical feature is illustrated in Figure 3.4. The inactive devices are displayed on the right and the active devices on the left of the figure.

Through the observation of Figure 3.4 witnesses the features' distribution, the outliers, average and the standard deviation. These variables appear to be similar among the groups, except for the $CPUTp$. For the $CPUTp$ the average is around 55 degrees in the inactive devices and around 40 degrees in the active devices, apparently caused by the higher values of the outliers or having more inactive devices reaching higher temperatures.

Considering scenario C from Figure 3.2, the instance t_{n-p} , marks the appearance of a new box. To apply scenario C, a plot similar to Figure 3.5 was done for each household and feature. Therefore, it will show a comparison between the inactive STB (changed) behaviour with the active STB (not changed).

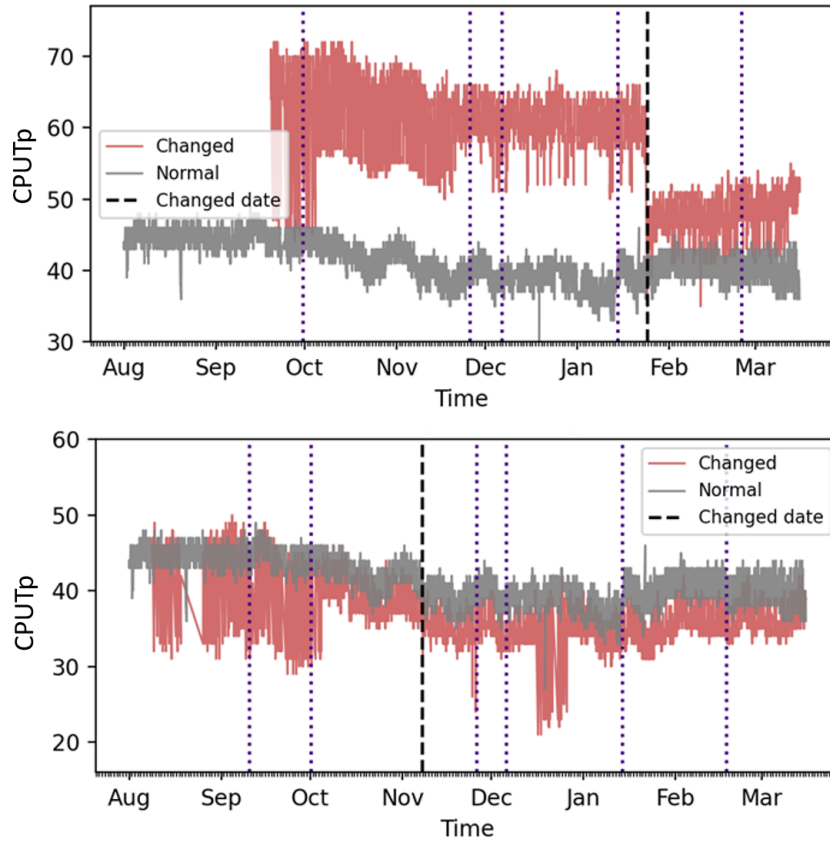


Figure 3.5: Two line plots with time against the $CPUTp$ value, they show two distinct behaviours. The grey line (Normal) corresponds to scenario A and the red line (Changed) corresponds to scenario C.

This Figure 3.5 describes two plots, and both have a red line (Changed) that symbolises one household with two devices active and inactive, respectively, like the scenario C. The grey line (Normal) symbolises a different household with one device active alike the scenario A. The line plot displays the evolution of the feature $CPUTp$ against *Time*. The dotted vertical black line (Changed date) is the instance t_{n-p} from scenario C, referred has the possible changed date. Finally, the purple dotted lines are the SW dates (to understand if this feature influences the temperature changes).

The two plots suggest distinct behaviours, the upper plot from the Figure 3.5, $CPUTp$ decreases after the dotted vertical black line that it is the moment the records of the active device starts (t_{n-p}). The distinct behaviour between the two households begins with lower temperatures and then increases to higher temperatures between 70 and 60 degrees and compared with the grey line (normal box) that reaches 50 degrees. The SW dates seems not to influence the $CPUTp$ variations within the devices. For further confidence regarding these statements, the analysis occurred for other households. Therefore it was observed 85% of all the clients from scenario C a behaviour similar to the client on the upper plot in Figure 3.5 represented with the red line. This behaviour can be a good pointer towards temperature's influencing on having a device

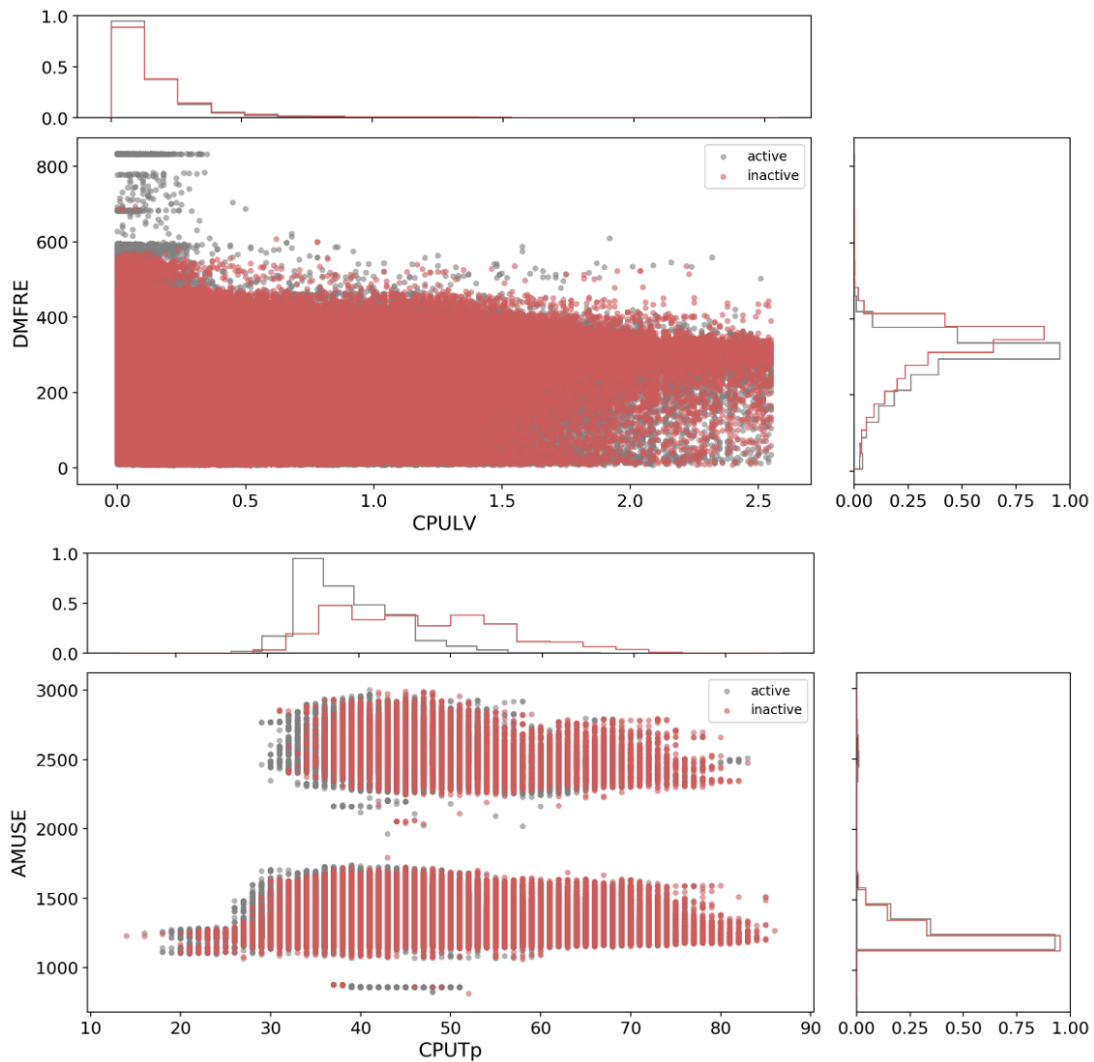


Figure 3.6: Bivariate analysis with a scatter and an histogram plots, the top plot is *DMFRE* against *CPULV* and the bottom plot is *AMUSE* vs *CPUTp* .

exchange. However, there are still 15% where this behaviour does not happen as the red household on the bottom plot in Figure 3.5.

Additionally, for some clients devices alike scenario C, there was not enough historical data from the equipments. Moreover, for other clients alike scenario C it was found long gaps within the historical data. These findings can guide two assumptions, or there are no records or an error in the data centre. Thus, it is essential to consider this in the Data Cleaning phase (Section 3.2.1).

As previously mentioned, this Figure 3.5 was done for the other features. However, they did not show any significant unusual performance before and after the changed date.

Reinforcing these conclusions, Figure 3.6 displays a scatter plot and histogram plots corresponding to each axis. The scatter plot shows the correlation between two variables, and the histogram plot the distribution of the associated axis. The grey

shadow is the active devices, and the red shadow is the inactive devices.

The top plot from Figure 3.6 represents the relationship between *DMFRE* and *CPULV*, there is no explicit distinct behaviour between the two histograms.

The bottom plot from Figure 3.6 represents the relationship between *AMUSE* and *CPUTp*, there is a clear distinction between the two groups, and the inactive devices hold higher temperatures than the active devices. The *AMUSE* only splits the data into two clusters. However, both clusters own the equivalent performance for active and inactive. Therefore this feature may not produce any impact on fault prediction.

This analysis was also done for other combinations of numerical features; moreover, there were only distinct patterns when the feature *CPUTp* appeared. From this arose the importance of understanding the impact of this feature among the inactive devices. Reinforcing this theory, 75% of the inactive devices and 3% of the active devices reached temperatures higher than 60 degrees.

Additionally, since the variable *Time* exists, a time series analysis was done. Firstly, it was observed if there are distinct repeated patterns among frequent intervals due to seasonal factors. The data was grouped in seasonal intervals, part of the days, weekdays and months. The only different finding was the device reaching lower temperatures in all features on the clients' custom bedtime. Additional to this exploration analysis, the autocorrelation and partial autocorrelation of the features did not show any pattern that could indicate the presence of seasonality or trend.

In summary, after giving these initial results, principal points were concluded to bear in memory during the subsequent sections.

1. The *CPUTp* feature revealed to be very relevant on this analysis and can lead to a significant impact on fault prediction;
2. There are some gaps in some devices which can be considered when cleaning the data in the next section;
3. There are devices with a small number of historical data which can be considered when cleaning the data in the next section;
4. There are no seasonality or trend patterns in the time series data;
5. The case with clients alike the scenario C, where the devices are breakdown, are cases that need future validation within the company.

3.2 Data Preparation

Data Preparation is the third stage of the workflow Figure 3.1, and the second-longest stage of the Project Development. According to the theoretical Chapter 2 is modifying the data towards the model. Therefore, the reconstruction of the data should have the attention of the *SA* model.

Process Name	Description	How much?	Data Quality
Duplicate rows	Remove rows that are exactly the same.	0.02% of all dataset	Consistency
Change features format	Change the format of the features for datetime, float or int.	All features	Consistency
Remove noisy data	Remove noisy words, e.g. 'undefined' or strange symbols, e.g. '%' or lowercase.	All features	Consistency
Missing values by feature	Remove features that have a high % of missing values.	No features	Accuracy
Constant features	Remove the features with no variance (constant).	<i>BN</i>	Consistency
High correlated numerical features	Remove high correlated numerical features, erasing the ones that own the same knowledge employing Pearson Coefficient (equation 2.2).	<i>DMUSE</i>	Interpretability
High correlated categorical features	Remove high correlated categorical features, this involves erasing the ones that own a strong association employing Cramér's V Coefficient (equation 2.1).	<i>MN</i>	Interpretability

Table 3.2: Data Cleaning report with all the data transformations applied to the features, including the data quality report based on the concepts from Table 2.2.

Data Preparation will branch into two steps: Data Cleaning 3.2.1 and Data Construction 3.2.2. On the first step, all the cleaning transformation steps are reported, and on the second step is the transformation of the cleaned data on the final structure to be used on the SA model.

3.2.1 Data Cleaning

The data cleaning process is an essential step for the subsequent steps. All the decisions here can determine the final results.

The cleaning processes are divided into two tables. On both tables **Process** column identifies the performed transformation. The second column **Description** describes the transformation process. The third column **How much?** relates the impact on the dataset of doing that transformation, for instance, the percentage of removed devices or the number of removed features. Finally, the column **Data Quality** indicates the

Process Name	Description	How much?	Data Quality
Filter devices	Remove equipments that are not <i>STB</i> .	No devices	Consistency
Missing values by household	For each client, if there is a high % of missing values in the respective <i>CPUT_p</i> feature, the device is deleted; Otherwise, the imputation is done, in this case using Median and <i>MAD</i> .	3% devices	Accuracy
Low historical data by device	The equipments with a significant small historical timeline compared to the others were removed.	22% devices	Timeliness
Low number of events by device	The equipments that have a significant small number of events comparing to the others, were deleted.	25% devices	Completeness

Table 3.3: Data Cleaning report for removing the inconsistent devices, including the data quality report based on the concepts from Table 2.2.

data quality term that is succeeded by that data cleaning process. The definitions of such concepts were explained earlier in Table 2.2.

Table 3.2 describes the processes applied to the features by removing all the noisy data, missing values, inconsistencies and feature selection transformations. Moreover, Table 3.3 describes the processes applied as criteria for deleting inconsistent devices.

Both tables guarantee the quality of data despite having a significant reduction of 50% of the devices—ending up to be the decision performed to have a consistent dataset. However, this reduction leads to less time series to analyse, consequently, fewer devices. At this point, the dataset is composed of 52% active devices and 48% inactive devices, being a balanced dataset (the two groups have almost the same weight).

After the transformation process, only the features *Device*, *Time*, *Active*, and *CPUT_p* are used to determine the input data for the model phase that will be described in the next Data Construction Section 3.2.2. It was only considered *CPUT_p* because, as previously concluded, it is a good indicator for the device malfunction, and due to the limited amount of time constraints, only a univariate model will be applied. However, Table 3.2 mentions all the transformations made for all the features (since all can be used in future work for multivariate analysis).

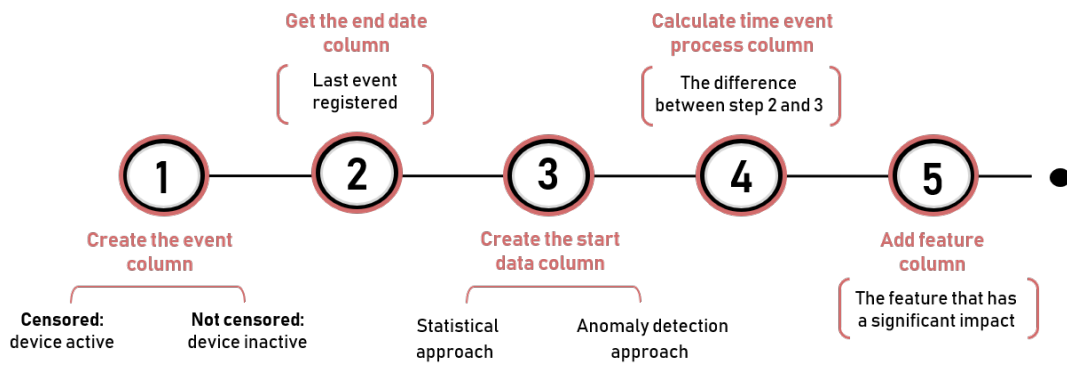


Figure 3.7: Process of constructing the final data set for the model with all the important columns **event**, **end date**, **start date**, **time event process** and **feature**.

3.2.2 Data Construction

The next step is constructing the dataset used as input on the Model phase (Section 3.3). As previously explained in Section 2.3, the **Survival Analysis** depends on two mandatory features regarding each device **ensorship** and the **time event process**.

The **time event process** is the number of days from the beginning of the anomalous event until the fault event occurred (device breakdown). Moreover, the **censoring** is a Right-Censoring problem, where the device is censored if it never encountered an event until the data extraction day (active devices). It is unknown if any event happened after the extraction day.

All the phases to accomplish this final dataset is summed in the flow Figure 3.7. The primary step implies creating a column **event**, signed with value one if the event did occur. Alternatively, it is signed with a value of zero if censored. This feature is based on the *Active* feature, which implies being censored if active at the data extraction moment.

The following step is to create a column **end date**, which is the date of the last registered event from each device, obtained by the column *Time*.

The third step is creating a column **start date**, the estimation regarding the timestamp or date that the device begins to malfunction (the event of interest). The main challenge is to establish this event since it is unknown when the device begins to break. So two different paths were adopted to identify this **start date**:

1. For the censored devices (active devices), the **start date** is fixed as the first date recorded by the device, acquired from the column *Time*.
2. For devices not censored (inactive devices), the **start date** implies a prediction employing a **Statistical Approach** or the use of **Anomaly Detection Approach**.

The **Statistical Approach** is based on the results of the exploratory analysis described during the last Section 3.1.2 that proved $CPUT_p$ feature implied to be decisive towards device failure.

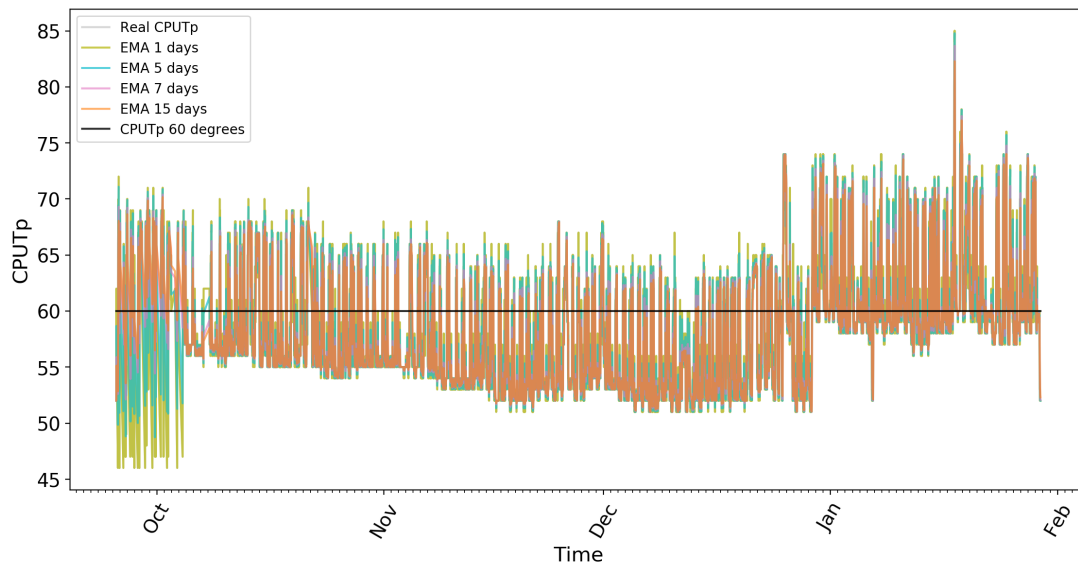


Figure 3.8: Line plot with time against $CPUTp$ that compares the behaviour of a device with grey shadow and the effect of applying EMA technique with different lags represented by the remaining colours.

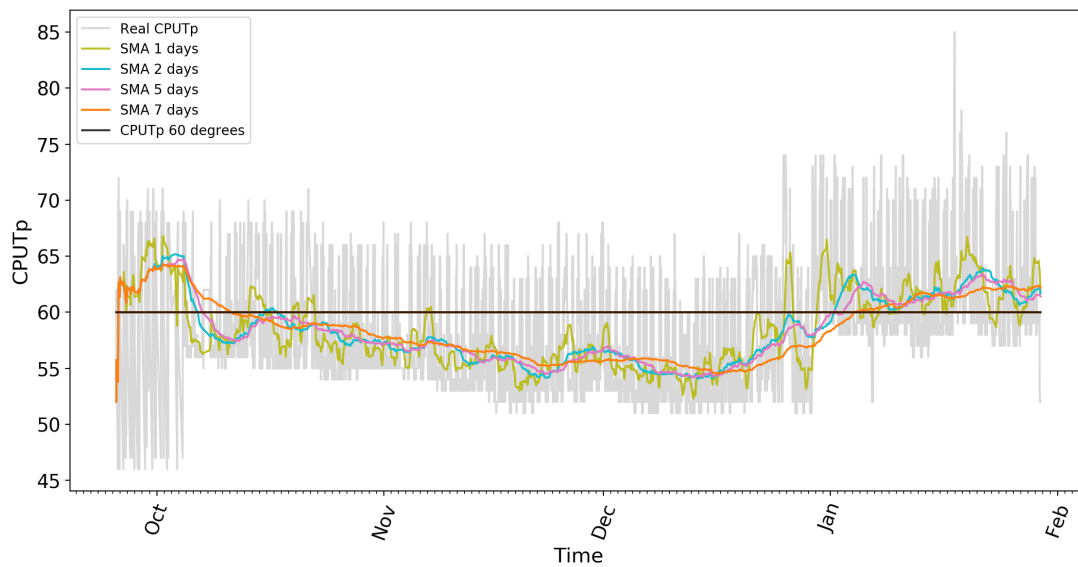


Figure 3.9: Line plot with time against $CPUTp$ that compares the behaviour of a device with grey shadow and the effect of applying SMA technique with different lags represented by the remaining colours.

The statistical technique is used as a simpler model for selecting the **start date** by employing one of two distinct Moving Average techniques EMA and SMA that were covered earlier in Section 2.1.1. Each model application employs a sliding window to compute the average over a period collection of n days. The event's **start date** using these techniques will be the first occurrence reaching a specific temperature.

The respective application of EMA and SMA is represented by Figure 3.8 and

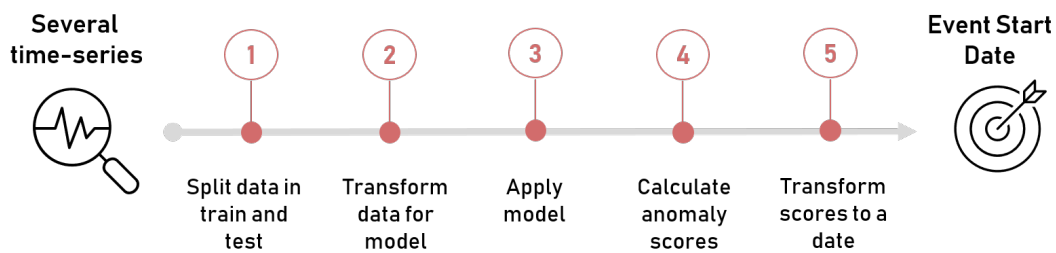


Figure 3.10: Process of getting the **start date** by means of [Anomaly Detection Approach](#) models, with input the time series and the output the **start date**.

Figure 3.9 for the example of an inactive device.

Both plots represent on the x -axis the *Time* and on y -axis the *CPUP*. The grey line represents the raw time-series of the inactive device. The black line represents the temperature threshold based on the conclusions previously explained (Section 3.1.2), where the temperature in the inactive devices reaches more than 60 degrees. The remaining colours represent the application of each technique for different n days.

When comparing both figures, the [SMA](#) is smoother than the [EMA](#) approach since it filters the effect of random variations. Whereas [EMA](#) with different lags, it does not differentiate much from the real behaviour, which does not outline the abnormal patterns; this probably happens because [EMA](#) gives more weight to the most recent data points.

Therefore, it is plausible to conclude that [SMA](#) removes better the noise, and it is the most suitable technique for detecting the **start date** of the [Statistical Approach](#). [SMA](#) technique will be employed in the modelling phase, using different temperature thresholds to understand which one yields better results.

The chosen lag for [SMA](#) will be one day since it can show slightly more variation when compared with other lags, as seen in Figure 3.9. These analyses for [EMA](#) and [SMA](#) were based on all inactive devices.

On the other hand [Anomaly Detection Approach \(AD\)](#) uses a more complex model for detecting the **start date** of the event using only the *CPUP* feature, the models can be: [AE](#), [RPCA](#), [SR](#), [IF](#) or [MAD](#) already explained on Section 2.2.3. As opposed to the [Statistical Approach](#) that the **start date** is obtained without any model learning. All the models need to go through different steps from the time series representation until they reach the **start date**. A precise idea of this process is designed in the Diagram 3.10.

Each step can be slightly different from this diagram depending on the implemented model, except for Step 1 and Step 5. On the **Step 1**, data is divided into two data sets, the train and the test set. The first receives the devices that hold normal behaviour (active devices), and the latter receives the devices that hold abnormal behaviour (inactive devices). The split acts in this direction since it is essential to train the models among "normal" pattern devices and test among inactive devices to identify the

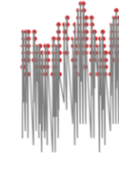
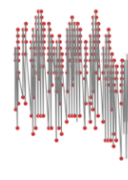
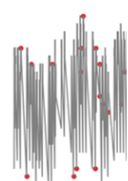
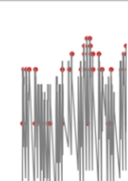
Model Name	Step 2: Transform data for model	Step 3: Apply model	Step 4: Calculate Anomaly score	Anomaly Threshold	Anomaly detection behaviour	Example of one time series displaying anomalies
AE	The input data are multiple time series in an array of a sliding windows of a fixed length with shape (batch_size, window size, nr features).	<ul style="list-style-type: none"> Batch_size = 128 Nr_ephocs = 10 Optimizer = "Adam" Loss = "mse" Window size = 300 nr features = 1 	The reconstruction error is the mean square error between the predicted and the real values of each time series. Returns scores between 0 and 1.	0.7	Low, high and medium peaks are detected	
RPCA	Input data is the original matrix M with shape (n° rows, n°columns) for each time series, the rows are the time, hh:mm:ss and columns the date dd-mm-yyyy.	<ul style="list-style-type: none"> Sensitivity = 4 Stop criterion=10e-7 	Results based on the sparse matrix S, the values are between 0 and 1 and expresse how anomalous each point is of the time series.	0.8	The higher peaks are detected, and only in some signal waves of the time series	
SR	The input array is the time series for a univariate time series.	<ul style="list-style-type: none"> Kernel size = 3 z = 21 	Results based on the saliency map, the values are between 0 and 1, expresse how anomalous each point is of the time series.	0.8	Just detect random low and high peaks, but in general detect different signals	
IF	The input array is the time series for a univariate time series.	<ul style="list-style-type: none"> nr_estimators= 100 max_samples= auto max_features=1 contamination=auto 	Returns value 1 if is na anomaly otherwise is -1. Those were transform in 0 or 1 respectively, to expresse how anomalous each point is of the time series.	1	Detect low and high peaks, do not detect different signals	
MAD	The input array is the time series for a univariate time series	<ul style="list-style-type: none"> Cut off = 3 	Results based on the cut-off, the values are 0 or 1, expresse how anomalous each point is of the time series.	1	Detect low and high peaks, do not detect different signals	No anomalies detect for this time series

Table 3.4: Comparison of the models used to calculate the **start date** Event, such as Autoencoder (AE), Robust Principal Component Analysis (RPCA), Spectral Residual (SR), Isolation Forest (IF) and Median Absolute Deviation (MAD).

abnormal pattern. On the **Step 5**, it is the transformation of the scores (outcome from Step 4) towards the **start date** of the event for each device by following the subsequent process:

1. A threshold for the anomalies scores, it is an empirical metric;
2. If a score of the time series is above the threshold, it is an anomaly;
3. For each point a cumulative sum of scores among a window of n days is delivered, it was considered a window of 5 days;

Identification	Event	End Date	Start Date	Time Event Process
A	0	2020-03-15 23:44:27	2019-08-13 20:26:45	215
B	0	2020-03-15 23:29:27	2019-08-28 10:50:29	200
C	0	2020-03-15 23:30:23	2019-08-03 15:20:48	225
D	0	2020-03-15 23:37:45	2019-08-06 19:38:24	222
E	1	2019-11-07 11:23:07	2019-09-08 11:21:22	60

Table 3.5: An example of the structure of data used as input of the Survival model, including the mandatory features the time event process (in days) and event (censoring).

4. The respective date of the highest cumulative score minus the n days is considered the **start date** of that time series, implying the interval with the highest anomaly scores.

The **Step 2, 3 and 4**, they may depend on the adopted model, are summarised in the first three columns represented in Table 3.4. The column **Step 2: Transform data for model** describes the transformation made to the data as input for the model. The **Step 3: Apply model** shows all the parameters used and the respective value. Finally, the **Step 4: Calculate Anomaly score** explains how the anomaly score was given.

Concerning models implementation, each row of the Table 3.4 suggests the following models **RPCA**, **SR** and **MAD** were applied using a package developed by the company's Data Science team. The **IF** model used the scikit-learn package. Moreover, the chosen **AE** architecture was a **LSTM AE** based on Keras documentation [21].

As it was mentioned in Section 2.2.3, the model parameters (Table 3.4) on Step 3, were retrieved from the literature.

When running the models, the **AE** required a more amount of data and a meaningful computational effort in time and memory compared with the remaining models.

The columns **Anomaly Threshold**, **Anomaly general behaviour** and **plot for one time series**, characterise the behaviour of the anomalies regarding each model. Those irregularities are limited by a threshold that classifies if a point is an anomaly (**Anomaly Threshold**). The remaining two columns (**Anomaly general behaviour** and **plot for one time series**) describes each model's overall behaviour, including an example of a device covered with anomalies (illustrated with a red dot) detected by each model. However, it is challenging to conclude the most reliable **AD** model to detect the **start date**. The model selection would require parameter tuning and optimisation, which were used the default parameters without tuning from this work.

The absolute difference between the **start date** and **end date** columns originates the fourth step on the flow 3.7 that is the **time event process** column. The fifth step, **feature column** adds the $CPUT_p$ feature since it is the feature that distinguishes the

device's behaviour for future tractability. Finally, the dataset is constructed and ready to be used on the model phase, as displayed in Table 3.5.

3.3 Data Modelling and Evaluation

This section is the data modelling and evaluation phase, stage four and five, based on the workflow in Figure 3.1. This section addresses the second goal from Table 1.1, of obtaining an acceptable anticipation period for at least 70% of the urgent interventions for the tech team to intervene in the client's house.

At this stage, the data was carefully assembled as displayed in Table 3.5 and it is ready for modelling. The data was split between two datasets, the train set and test set with 80 % and 20%. The model uses the firsts to estimate an optimal anticipation period, and the latter helps to evaluate the model.

As previously explained, due to a limit amount of time, it was only applied one SA univariate model, [Kaplan-Meier \(KM\)](#), based on the apriori knowledge acquired in Section 2.3.

With the **time event process**, it is possible to estimate a Survival Curve using the [Kaplan-Meier](#) model, which gives the probability of each device surviving longer at each time.

This Survival Curve was done for the several datasets that use each **start date** approach defined in the last section. Such as a more complicated approach using [Anomaly Detection Approach](#) models alike [AE](#), [SR](#), [IF](#), [MAD](#) and [RPCA](#). Alternatively, a more straightforward approach using the [SMA](#) technique employing a $CPUT_p$ threshold by reaching 50, 55 or 60 degrees in an average of 1 day ([SMA_50_1](#), [SMA_55_1](#), [SMA_60_1](#)).

Figure 3.11 represents the Survival Curve comparison employing multiple data construction techniques presented with distinct shadows, with the time on days against the probability of surviving at that time.

Observing the plot is plausible to verify that the two approaches behave in two distinct forms. The [AD](#) models start to decrease earlier than the [SMA](#) Survival Curves when using the different temperature thresholds. In the [Statistical Approach](#), the probability of surviving above 99% is held for a more extended period so the technician would have more time to assist the client's house. Whereas [AD](#) starts to decrease with a shorter lag, suggesting the technician would have less time to intervene before the failure.

To reinforce the above conclusion, the **time event process** (y-axis) calculated with each model (x-axis) is displayed in Figure 3.12 in a parallel box-plot. The box-plots of the [Statistical Approach's](#) [SMA_50_1](#), [SMA_55_1](#) and [SMA_60_1](#), the first quantile starts around 50 days later than the [AD](#) models.

The Log-Rank Test explained earlier in Section 2.3.1 proofs this conclusion. The test was performed to compare the different curves between the different approaches.

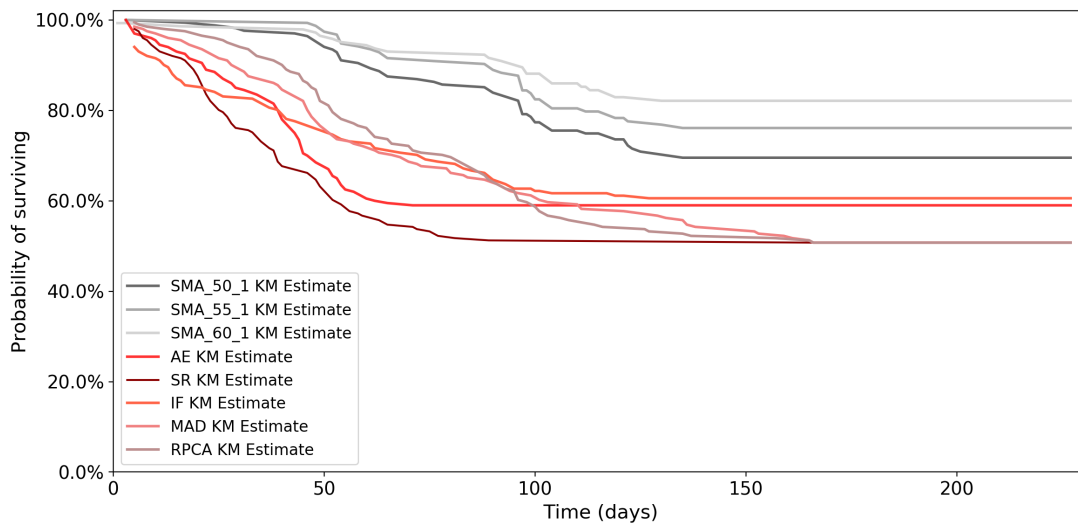


Figure 3.11: A multiple Survival Curve comparison employing the [Kaplan-Meier Estimator](#) with the use of the [AD](#) and [Statistical Approach](#) models for detecting the **start date** event.

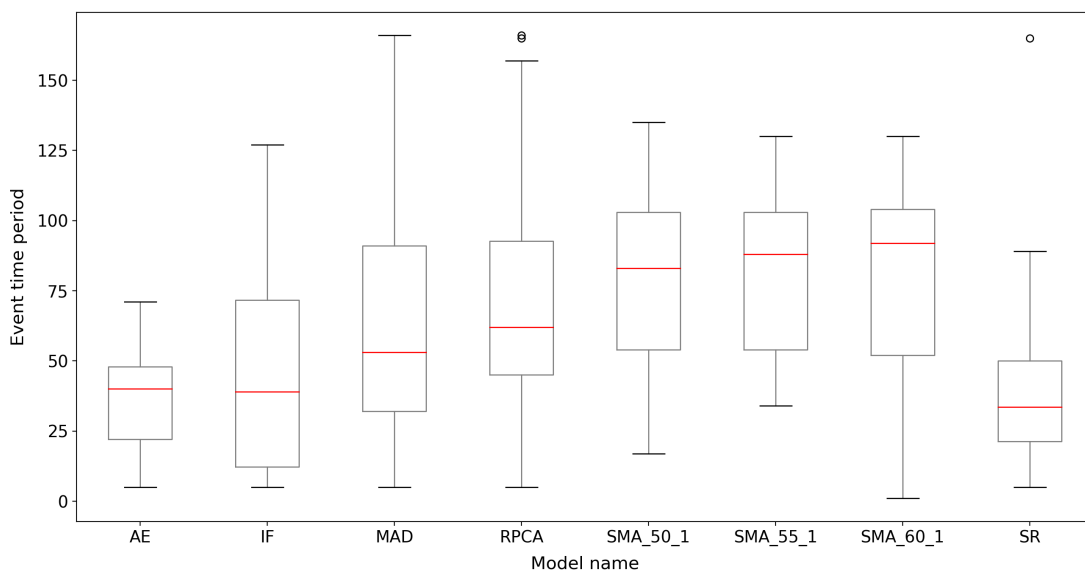


Figure 3.12: Parallel box-plot representing the distribution of **time event process** for the inactive devices for the different data construction models.

The results suggested that within the [AD](#) Survival Curves, the test did not reject the null hypothesis with a level of significance of 5%, which means that the curves are similar. The same happened within the [Statistical Approach](#) Survival Curves. However, when doing the test between curves from different approaches, the opposite occurred, which means the curves have distinct behaviours.

The Survival Curves being different within approaches can be because of: The [Anomaly Detection Approach](#) detects the start of the anomalous event as the beginning of the window with the highest number of anomalies. Contrary to the [Statistical](#)

Start event date model	Interval	TPR	FPR	TNR	FNR
SMA_50_1	[38; 227]	71 %	0 %	100 %	29 %
SMA_55_1	[36; 227]	67 %	0 %	100 %	33 %
SMA_60_1	[29; 227]	67 %	0 %	100 %	33 %

Table 3.6: Results of employing the **SMA** for different temperature threshold for detecting the start of the anomalous event, indicating the final interval, True Positive Rate (TPR), False Positive Rate (FPR), True Negative Rate (TNR) and False Negative Rate (FNR).

Approach, which considers as the **start date**, the first day where the temperature reached the threshold. The second fact is that the **AD** models did not suffer any optimisation. Finally, only a tiny sample of data was used. Thus, if any of these facts were addressed, the **AD** Survival Curves could present different behaviours.

The final results were analysed only for the **Statistical Approach** reaching temperatures of 50, 55, and 60 degrees since it yield consistently better. Moreover, the company stated its preference toward a lower complex model as a starting point.

Table 3.6 illustrates all the results for each one of the **SMA** thresholds. The column **Interval** indicates the interval determined by using the estimations from the Survival Curve employing the Train set. Based on this interval, the proportion of true positives, false positives, true negatives and false negatives that are calculated for the test set are represented on the last four columns on the Table 3.6 respectively:

- **True Positives Rate (TPR)** is the percentage of inactive devices, within the interval, classified correctly;
- **False Positives Rate (FPR)** is the percentage of active devices that already failed before the beginning of the anticipation period (lower boundary). In other words, the active devices that encounter the fault event (reaching a certain temperature) but did not fail;
- **True Negative Rate (TNR)** is the percentage of active devices within the interval classified correctly;
- **False Negative Rate (FNR)** is the percentage of inactive devices that failed before the anticipation period (lower boundary) and did not encounter the fault event, that is, reaching high temperatures.

With these metrics, it is reasonable to compare all **Statistical Approaches** from Table 3.6 and conjure the best approach concerning the company's main goals. The **FNR** and the **TPR** are the only columns with different scores. The **FNR** can suggest that devices' failure can be caused by other problems besides overheating since it gives

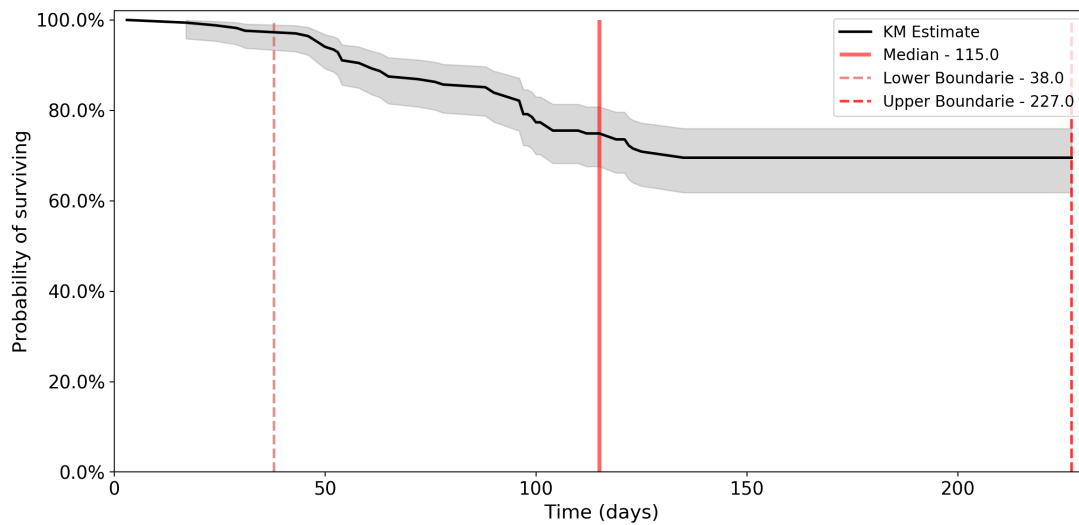


Figure 3.13: A Survival Curve employing the [Kaplan-Meier](#) Estimator with the use of the `SMA_50_1` technique for detecting the **start date** event

the percentage of inactive devices that never encountered temperatures higher than 50, 55 and 60 degrees.

The **TPR** suggests the most important score for deciding which is the best approach. Since it indicates the percentage of inactive devices within the interval that encounter the fault event, in other words, the reduced number of emergency interventions when applying this **PM** technique. Observing all the metrics from [Table 3.6](#), the approach that has higher scores is the `SMA_50_1`.

[Figure 3.13](#) displays an isolated example of the Survival Curve for the case of `SMA_50_1`. The red vertical lines were achieved with the **MAD** technique, where the middle line is the median and the dotted lines are the lower and upper boundaries. The lower boundary is calculated based on the equation ($median - mad$). The upper boundary is the last recorded *Time*.

The grey shadow is the 95% Confidence Interval explained earlier in [Equation 2.15](#). With the lower boundary, it is possible to obtain an anticipation period of at least 38 days for the technical team to intervene. Therefore it will be possible to anticipate the fault for all the devices that fail after that period.

Considering all the conclusions obtained based on [Figure 3.13](#) and [Table 3.6](#). The **Statistical Approach** is the best choice using a temperature threshold of 50 degrees for the one-day smooth average (`SMA_50_1`) and possible to obtain an acceptable anticipation period of at least 38 days for the technical team to appear at the client's house. Consequently, it is possible to reduce the number of actual emergency interventions to about 71% when looking at the **TPR** score from [Table 3.6](#).

In the end, the straightforward approach, based on the initial data exploration, delivered the results that the company wanted. Despite being an approach based on thresholds, probably with more data, the results would not be the same. However,

AD models could be a more promising approach. With more data, better computer resources, and optimised parameters, the results could be better, leading to a more reliable approach since it is not based on a threshold. Better to be used in the long term. In future work, consider the combination of AD and [Statistical Approach](#) approach. Using the statistical models to smooth the time series before using any AD model smooths the effect of random variations highlighting the abnormal patterns and making it easier for the model to detect the anomalies.

In summary, the results achieved the second goal displayed in Table 1.1. Therefore, through this approach, the company could predict 70% of the possible emergency swaps, which can successfully lead to the company saving more money than not making any prediction.

3.4 Deployment

The deployment phase is the final stage, containing the descriptions of the final remarks and achievements. The last sections explained the accomplishment of goals 1 and 2 of Table 1.1 in a reasonable time (end of June). Additionally, at the end of this section, an extra goal is revealed suggested by the company.

In the business world, Data Science concepts tend to be challenging to explain to business individuals. Thus it is suitable to use the advantage of visualisation techniques to communicate data and results clearly and effectively through graphical representation.

Therefore, the proposed solution was demonstrated through an interactive dashboard developed in Python using the Dash framework [22], the Figure 3.14 represents a screenshot of that dashboard. This framework is ideal for building data visualisation apps with custom user interfaces; these apps are viewed locally. These Dash apps are composed of two parts: The app's layout that describes what the application looks like and its interactivity that uses callbacks.

This dashboard implementation is segmented into three parts, as shown in Figure 3.14:

1. In Figure 3.14, the top shows three steps to the data set creation for the Survival Curve composed by **(1) Select the model**, **(2) Select the Data of event** and **(3) Generate new data**. The **(1)** selects the Survival model, in this case, [Kaplan-Meier](#). The **(2)** selects how the devices fault event is triggered, represented by a switch button. If it is "Anomaly" it implements the [Anomaly Detection Approach](#) otherwise "I Define" uses the [Statistical Approach](#). The [Statistical Approach](#) has two sliders, one to define the Temperature threshold and the number of Days to smooth the average. Finally, the **(3)** generates data as input on Survival Curve (plot below) based on steps **(1)** and **(2)** by clicking on the "Update" button.

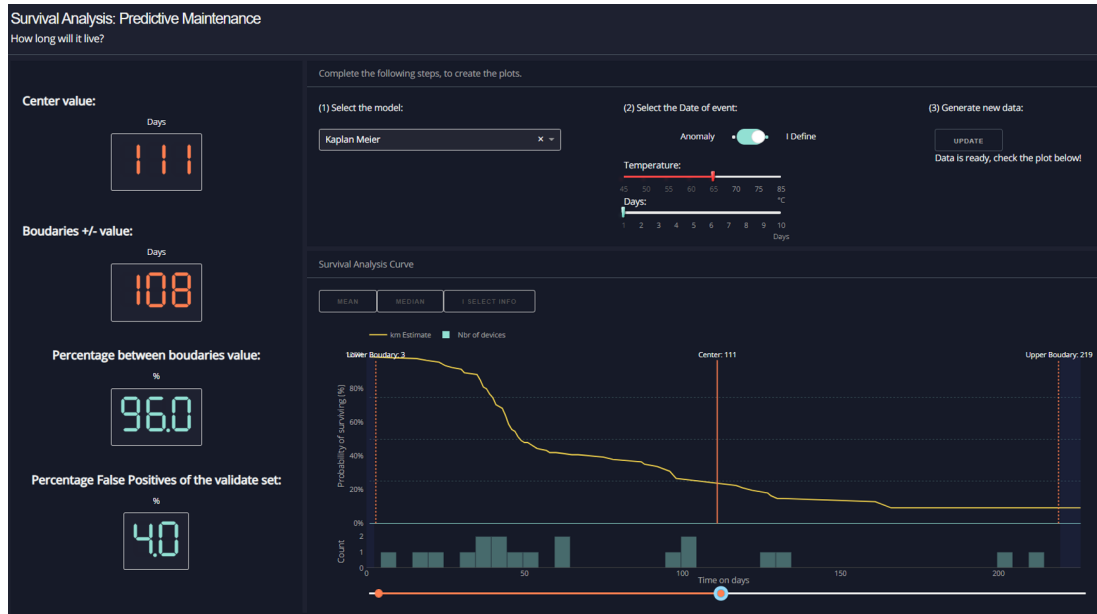


Figure 3.14: An example of an interactive dashboard illustrating the solution for the proposed business problem. Where it represents a Survival Curve with a customized data depending on the chosen approach.

2. The bottom of Figure 3.14 represents the generate Survival Curve based on the last point's definitions. The y-axis is the probability of surviving as a percentage, and the x-axis is the time on days. The vertical dotted lines illustrated on the plot represents the upper, middle, and lower boundary used to estimate the interval. These lines can change the place based on three buttons: the "mean" calculates the mean (centre) and standard deviation (boundaries); The "median" calculates the median (centre) and mad (boundaries); And the "I select info" is a popup button that explains the possibility of manually setting the boundaries using the orange slide range on the bottom. Additionally, below the Survival Curve, there is a bar plot displaying the number of devices that disrupt each instance.
3. Lastly, on the left side of Figure 3.14, represents the evaluation metrics that change based on the selected button: The middle value on the plot, the value of the boundary, the **True Positives Rate (TPR)** (proportion of inactive devices) concerning the test set within the boundaries and finally, the last box is the **False Negative Rate (FNR)** is the percentage of inactive devices that failed before the anticipation period concerning all the data set.

Creating this visualisation made it easier to understand the different models' behaviour with a click and gained the target audience attention by telling a story. That translated into a successful presentation, enabling us to move to the subsequent stage of a project with real-time data, in an Agile and bigger team and with access to more resources. So in the future, it will be possible to consider other features, try other SA

models and optimise the [Anomaly Detection Approach](#) models.

In the middle of July 2020, the dashboard visualisation was concluded. Emerged a new opportunity to develop until the end of July a software package with the pipeline used on this project to be applied in different Predictive Maintenance cases. Without considering the Data Cleaning phase since it will depend on the nature of the dataset or the use case, only the Data Construction, Modelling and Evaluation phases were implemented. Knowing more about this package's pipeline can be seen in [Appendix A](#).

Later on, this generic package was registered on the [Associação Portuguesa Software \(ASSOFT\)](#) framework, which records the author rights on the software to gain intellectual protection and gain more to the company. This achievement was another successful milestone for different use case projects within internal and outsourcing company projects.

CONCLUSIONS

The project started with the company's need to compete for more costumers, avoid customer churns, reduce the number of claims, the cost and increase innovation within the company projects. The need emerged to replace the [Reactive Maintenance](#) with [Predictive Maintenance](#) interventions to devices (e.g. [STB](#)).

This report suggests a solution to a [Predictive Maintenance](#) solution that aims to anticipate, within an acceptable period, the equipment ([STB](#)) failure. This anticipation enables planning ahead future maintenance to devices, thus reducing the company costs.

The goals on [Table 1.1](#) settled at the beginning of the report were fulfilled by the end of July 2020.

The first goal meant to answer the question "How to do it?" explored and identified the best approach to solve the proposed problem seen in Theoretical Framework [Chapter 2](#). That was using a [Survival Analysis](#) model, the [KM](#). The model's input data needed to have a specific structure by containing the period of the devices' anomalous event. However, the given dataset did not have the beginning of that event. So emerged the need to explore different techniques to detect the beginning of the anomalous event, such as a [Statistical Approach](#) techniques ([SMA](#) and [EMA](#)) or a [Anomaly Detection Approach](#) ([AE](#), [RPCA](#), [SR](#), [IF](#) and [MAD](#)).

The second goal meant to answer the question "How to solve it?" contains applying the concepts explored on the first goal seen in [Project Development Chapter 3](#). During the [Data Exploration \(Section 3.1.2\)](#) phase, the temperature was a decisive feature for a changed device. Knowing this was only considered one feature during [Data Construction](#) for detecting the anomalous event when employing the [Statistical Approach \(STA\)](#) and [Anomaly Detection Approach \(AD\)](#). It was concluded that using different

approaches to construct data as input of the [Kaplan-Meier \(KM\)](#) model could be decisive for the final results. With the employment of [SMA](#) it was achieved the proposed goal of anticipating 70% of the urgent interventions. Besides model selection, several temperature thresholds were tested in [SMA](#) to understand their impact on the results. Thus our final proposal allowed for minimum anticipation of 38 days when devices achieve temperatures of 50 degrees in a one-day smooth average.

Finally, in Section 3.4, the third goal, which consists of answer the question "How to demonstrate it?" by applying a dashboard employing the Data Visualization tools that demonstrate the proposed problem's results. As an extra milestone, a software package was developed with the pipeline used on the project but for different use cases of [Predictive Maintenance](#) stated in Section 3.4. Later on, this package was registered on [Associação Portuguesa Software \(ASSOFT\)](#) framework with the possibility of being improved to use in internal and outsourcing projects within the company.

Lastly, all these accomplishments provided the ability to proceed to the subsequent stage consisting of a successful demonstration of this solution using the developed dashboard to a future project client with the exact use-case as in this report, with the possibility of employing the developed package. So it enabled the opportunity to go from an initiative project to a long-term project with access to real-time operational data in a more considerable team and access to more resources.

4.1 Report Evaluation and Lessons Learned

The initially proposed goals were accomplished until the end of the project time; yet, each goal's solution changed a lot during that period. The following lessons were learnt along the way:

1. Adaptation to the circumstances without delusions – Since the data was scarce (with only one relevant feature), and the time was short, the initial thought of a multivariate Survival Analysis was changed to a univariate [Survival Analysis](#). This decision allowed us to invest time in visualisation tools to present the project to future clients, leading to the second lesson learnt.
2. Importance of visualisation and adequately conveying the message – it was initially thought only to display the Survival Curve and the respective metrics. However, it ended up being a more complex dashboard that calculated at real-time the final Survival Curve with the final maintenance interval depending on the chosen approach. This change was made because future clients do not know the models or concepts most of the time. Thus, it is essential to demystify the solution through visualisation. Consequently, if the client sees it as a good investment, there is a possibility of obtaining more resources by accessing a more significant project in the future.

3. Investing time smartly and have the flexibility to change the initial plans – Overall, this project was a time to gain the "know-how" since the academic world is quite different from the business world. More time was spent on some project phases than others, which was not expected. As an example, the Data Visualization phase was a step that required considerable time that was initially unplanned. So the following important question emerged: How to invest the time? It can be invested in a low complex model and the structure for future developments or a more complex model without data visualisation and difficult reproducibility.
4. Personal growth and acceptance of mistakes - In conclusion, the organisation skills, time management, team spirit (even more in this Pandemic year, in order to continue to have motivation and support), the desire to learn and learn from mistakes are all critical aspects for continuous learning and self-improvement for those who start and want to continue working in the business world.

4.2 Limitations

As stated before, this project could have been solved in several different ways, but since this is a company project, a simpler approach like [Statistical Approach](#) works for the wanted results. During the project process, several limitations were found as described along with the report:

1. The project period was planned to be between March and June, but took one extra month due to the fourth milestone and all the following enumerated limitations, as shown in Chapter 3;
2. Lack of data. A small sample size of devices was considered, and due to poor data quality (e.g. the gaps in the historical data), a significant number of devices was deleted as shown in Section 3.2.1, leading to a smaller dataset;
3. There was a lack of historical data from the equipments. Only eight months was considered, as explained in Section 3.1.1;
4. Finally, the lack of validation of the problems that originated the devices swaps;

4.3 Future Work

This project has a lot to pick up and improve, as stated along with Project Development (Chapter 3). There are six new suggestions for future work:

1. Get more real-time data with more than eight months of historical data in order to obtain more trustful and accurate results;

2. Optimise the [Anomaly Detection Approach](#) models by tuning the parameters, for instance, using the Grid-Search (consider all parameter combinations) or employing a different architecture for the [AE](#) model;
3. Optimise the process of anomaly identification in order to avoid the use of the threshold to define the anomaly;
4. Application multivariate [SA](#), such as Cox Proportional-Hazards model, Neural Multi-Task Logistic Regression, Random Survival Forest or others that can be found already implemented in the `pysurvival` package;
5. Consider joining the two approaches the [Statistical Approach](#) and the [Anomaly Detection Approach](#), by using the smoothed curve as an input of the [AD](#) models;
6. Finally, have more available resources as team members and computational memory.

BIBLIOGRAPHY

- [1] *Anacom - Compensation for non-compliance*. 2010. URL: <https://www.anacom.pt/render.jsp?categoryId=339736> (visited on 11/20/2020).
- [2] *Anacom - Supervision of interventions (urgent and non-urgent) and of interventions*. 2011. URL: <https://www.anacom.pt/render.jsp?categoryId=339641> (visited on 11/20/2020).
- [4] E. J. Candes, X. Li, Y. Ma, and J. Wright. *Robust Principal Component Analysis?* Tech. rep. 2009.
- [0] V. Chandola, A. Banerjee, and V. Kumar. *Anomaly Detection: A Survey*. Tech. rep. 2007.
- [20] P. Chapman. *Step-by-step data mining guide*. Tech. rep. 2000.
- [6] C. Chatfield. *Problem Solving: A statistician's guide*. Paperback, 1995.
- [30] Z. Chunkai and Y. Chen. *Time Series Anomaly Detection with Variational Autoencoders*. 2019.
- [7] *Cookiecutter Data Science*. URL: <https://drivendata.github.io/cookiecutter-data-science/> (visited on 11/20/2020).
- [13] J. Crawshaw. *AI/ML for CSP Operations: From Reactive to Predictive Autonomous*. Tech. rep. 2019.
- [18] Z. Ding and M. Fei. "An anomaly detection approach based on isolation forest algorithm for streaming data using sliding window." In: vol. 3. Part 1. IFAC Secretariat, 2013, pp. 12–17.
- [19] A. Fabrizi. *Network equipment failure predictionf*. URL: <https://www.linkedin.com/pulse/network-equipment-failure-prediction-andrea-fabrizi/> (visited on 02/28/2020).
- [9] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016.
- [10] J. Han, M. Kamber, and J. Pei. *Data Mining. Concepts and Techniques, 3rd Edition*. Tech. rep. 2011, pp. 40–99; 364–373.
- [12] H. Ishwaran, U. B. Kogalur, E. H. Blackstone, and M. S. Lauer. "Random Survival Forest." In: *The Annals of Applied Statistics* 2.3 (2008), pp. 841–860.

- [14] W. Jeffrey, C. Chris, M. Elijah, and V. Shankar. *RAD — Outlier Detection on Big Data*. 2015. URL: <http://techblog.netflix.com/2015/02/rad-outlier-detection-on-big-data.html> (visited on 02/24/2020).
- [15] D. G. Kleinbaum and M. Klein. *Survival Analysis*. Springer-Verlag New York, 2012.
- [16] C. Leys, C. Ley, O. Klein, P. Bernard, and L. Licata. “Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median.” In: *Journal of Experimental Social Psychology* 49 (2013), pp. 764–766.
- [24] F. T. Liu, K. Ting, and Z.-H. Zhou. “Isolation Forest.” In: Jan. 2009, pp. 413–422.
- [17] X. Liu. *Survival Analysis: Models and Applications*. Higher Education Press, 2012.
- [22] Plotly. *Dash for Python Documentation*. URL: <https://dash.plotly.com/introduction>.
- [8] S. Pölsterl, N. Navab, and A. Katouzian. *Fast training of support vector machines for survival analysis*. Tech. rep. 2015, pp. 243–259.
- [3] *Reclamações no sector das comunicações Primeiro Semestre de 2020*. Tech. rep. 2020, pp. 18–21.
- [23] H. Ren, B. Xu, Y. Wang, C. Yi, C. Huang, X. Kou, T. Xing, M. Yang, J. Tong, and Q. Zhang. “Time-Series Anomaly Detection Service at Microsoft.” In: *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2019), pp. 3009–3017.
- [0] *SAS Help Center: Introduction to SEMMA*. 2017. URL: <https://documentation.sas.com/?docsetId=emref&docsetTarget=n061bzurmej4j3n1jn j8bbj j m1a2.htm&docsetVersion=14.3&locale=en> (visited on 11/20/2020).
- [21] P. Vijay. *Timeseries anomaly detection using an Autoencoder*. 2020. URL: https://keras.io/examples/timeseries/timeseries_anomaly_detection/ (visited on 01/09/2021).
- [25] J. Wang, C. Li, S. Han, S. Sarkar, and X. Zhou. “Predictive maintenance based on event-log analysis: A case study.” In: *IBM Journal of Research and Development* 61 (2017), pp. 121–132.
- [27] *Why do we need to look for patterns? - Pattern recognition*. URL: <https://www.bbc.co.uk/bitesize/guides/zxxbgk7/revision/2> (visited on 07/25/2020).
- [28] *Why Is Predictive Maintenance Important? - Digital Doughnut*.
- [29] H. Xu, Y. Feng, J. Chen, Z. Wang, H. Qiao, W. Chen, N. Zhao, Z. Li, J. Bu, Z. Li, and et al. “Unsupervised Anomaly Detection via Variational Auto-Encoder for Seasonal KPIs in Web Applications.” In: *Proceedings of the 2018 World Wide Web Conference on World Wide Web - WWW ’18* (2018).

- [31] C. Zhou and R. C. Paffenroth. “Anomaly Detection with Robust Deep Autoencoders.” In: *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Association for Computing Machinery, 2017, 665–674.
- [32] Z. Zhou, X. Li, J. Wright, E. Candes, and Y. Ma. “Stable Principal Component Pursuit.” In: *IEEE International Symposium on Information Theory - Proceedings* (2010), pp. 1518–1522.



APPENDIX 1

On the middle of July 2020, the dashboard visualization was concluded. It emerged a new opportunity, to develop a software package with the pipeline used on this project but for different Predictive Maintenance cases. Consequently, this is another proposed milestone by the company, therefore the goals Table 1.1 was updated by adding this new opportunity.

This extra milestone was successfully done until the end of July. This aims to be more generic, thus only the Data Construction, Modelling and Evaluation phases were implemented. Since the Data Cleaning phase will depend on the nature of the dataset or the use case.

Later on, it was registered this generic package on the [Associação Portuguesa Software \(ASSOFT\)](#) framework records the author rights on the software to gain intellectual protection and gain more value in the company. This achievement was another success story that can be employed on different use case projects within internal and outsourcing company projects, not only for the one described in this report.

The package followed the cookiecutter structure template developed by GitHub [7] and a representation of the pipeline flow is displayed in the Figure A.1, and follows the subsequent steps:

1. Import data already cleaned with the following input columns: *timestamp*, *identifier*, *event* and *feature_1* (alike the input shown on Section 3.2.2) and split between train and test set;
2. Construct the data as input of the model phase, that follows all the steps described earlier on Section 3.2.2, however, is necessary to choose the approach in order to estimate start date event that can be an Anomaly Detection model or

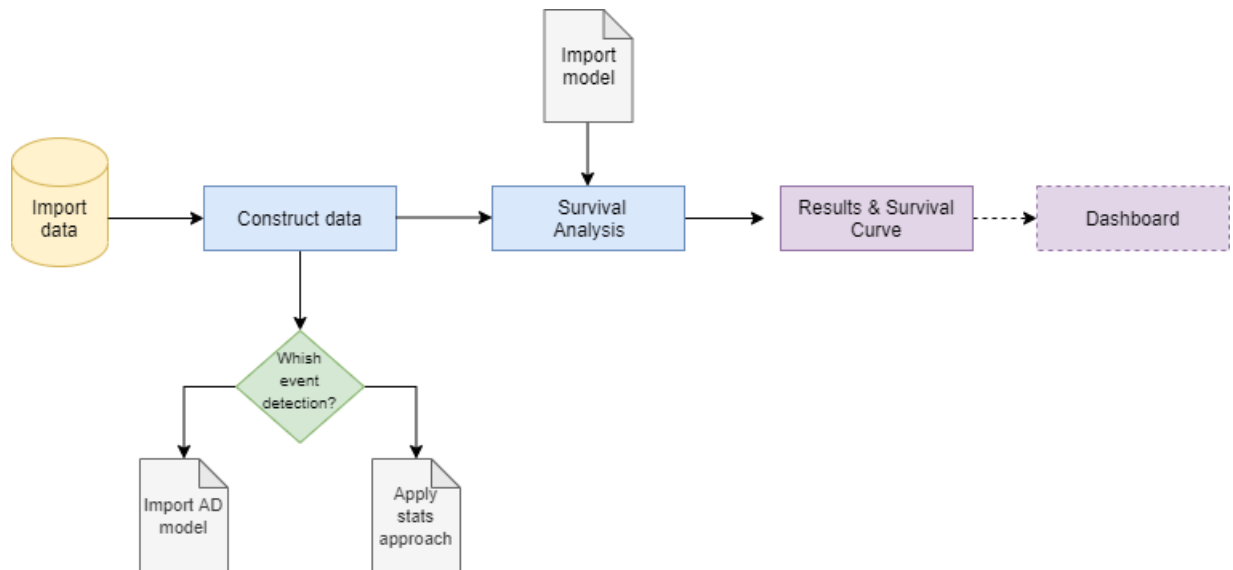


Figure A.1: A flow that represents the architecture of the package pipeline that followed the cookiecutter structure template developed by GitHub [7].

Statistical model - the definition of the respective parameters can be defined on a separate script;

3. Apply the Survival Analysis model with the output of the Construct Data (previous step) using the train set. The import model has Kaplan Meier as the only option described in Section 3.3;
4. Return the Survival Curve and the corresponding estimations for each identifier of the test set following the same strategy as described on the section 3.3;
5. Lastly, if there is a need, the dashboard can be displayed, similar to the one described in the last Section 3.4.

Each square step of the flow in Figure A.1 corresponds to a python script, and the folder structure is referenced in Figure A.2.

Later on, it was registered this generic package on the [Associação Portuguesa Software \(ASSOFT\)](#) framework records the author rights on the software to gain intellectual protection and gain more value in the company. This achievement was another success story that can be employed on different use case projects within internal and outsourcing company projects, not only for the one described in this report.


```

├── .gitignore          <- Files that should be ignored by git. Add separate .gitignore files in sub folders if
                        needed
├── LICENSE
├── README.md          <- The top-level README for developers using this project.
├── requirements.txt    <- The requirements file for reproducing the analysis environment, e.g.
                        generated with `pip freeze > requirements.txt`. Might not be needed if using conda.
├── data
│   ├── interim        <- Interim files - the data_event csv generated for each model
│   ├── models         <- Files relating to the training process of the model
│   ├── processed      <- The cleaned data frame with all the mandatory features
│   └── temp           <- Temporary files.
├── reports            <- All the generated figures are saved here
├── src                <- Code for use in this project.
│   ├── eventDetection <- This is the folder for generating the df_event
│   │   ├── __init__.py
│   │   ├── main_event.py <- This as the main function for generating the df_event
│   │   ├── main_event_aux.py <- Auxiliar function for the main_event
│   │   └── start_date_aux.py <- The auxiliar functions for the anomaly detection
│   ├── survivalAnalysis <- This is the folder for generating the SA model
│   │   ├── __init__.py
│   │   └── models      <- This is the folder for generating models
│   │       ├── kaplan_meier.py <- The class of the Kaplan meier model
│   │       ├── evaluation.py <- Plots and metrics for the results from the model
│   │       ├── main_survival_analysis.py <- The script with the main functions for fit and predict the models
│   │       └── sa_aux_functions.py <- The auxiliar functions for the survival analysis
│   ├── utils          <- This is the folder with all the utils functions and variables
│   │   ├── __init__.py
│   │   ├── aux_functions.py <- auxiliar functions regarding the files and scripts
│   │   ├── path_folders_definition.py <- for generating all the necessary paths for saving information
│   │   └── variables_definition.py <- For defining all the variables for the project
│   └── main.py        <- This is the main script where you run all the project

```

Figure A.2: The structure of the package pipeline that followed the cookiecutter structure.

