FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Developing a Forecasting Analysis Framework to Fast Track Analytical Developments

Verónica Sofia Marcos Fradique

Mestrado Integrado em Engenharia Informática e Computação

Supervisor: Prof. José Luís Borges

July 31, 2020

Developing a Forecasting Analysis Framework to Fast Track Analytical Developments

Verónica Sofia Marcos Fradique

Mestrado Integrado em Engenharia Informática e Computação

Approved in oral examination by the committee:

Chair: Prof. João Moreira
External Examiner: Prof. João Vinagre
Supervisor: Prof. José Luís Borges

July 31, 2020

Abstract

Digitalization is an emerging concept and is increasingly getting relevance among companies. Due to this, some companies realize a need to change their own business models, in order to adapt to market impositions and requirements. One of the most important factors in this process is the reduction of lead times. To achieve this, it is necessary to create an environment integrating all the conditions to allow fast development and deployment of the solutions.

Forecasting plays a pivotal role in several industries that deal with uncertainties in the global and domestic markets. It helps in different domains, from business operations optimization to anticipating changes in the markets. In the context of LTPlabs, a boutique consultancy company that develops data-driven solutions to assist clients in making better decisions, forecasting is a typical problem being addressed daily. When a forecasting project starts, the business requirements and needs are defined. Then data is collected among the different sources, cleaned from missing values, and analyzed to understand it in terms of overall statistical analysis. After this process, data is selected and prepared to start forecasting. The next two steps are to select a forecasting algorithm and evaluate its performance. The process is iterative until a satisfactory evaluation is accomplished, proceeding to the deployment of the solution. Across projects, there are implementations of tools that support preprocessing, forecasting algorithms, and evaluation metrics that are not being reused since they are tailored to the specific project's data.

Actual forecasting performance, already achieved with the forecasting techniques used, and how they are chosen and tested, can be leveraged with a high-level framework aggregating the most common and some state-of-the-art techniques and evaluation metrics as well as preprocessing tools. In the reviewed literature, there is not a tool able to answer the identified needs, providing a fast analysis with lower user inputs and, at the same time, able to customize it to control and track all the stages of the development. Aiming to achieve better forecasting results and address a wider range of projects, an automatic approach to forecast was proposed and a framework developed including, also, grouped structure forecasting and inclusion of exogenous variables.

The high-level framework developed was tested with analysts on ongoing projects and was easily accepted and adopted, leading to a significant reduction in development times. Additional side effects include reducing costs maintaining and improving the high quality of the solutions developed and deployed. Moreover, the standardization of the forecasting process throughout the company, and systematization of existing knowledge related to the topic result from the implementation of the framework, ensuring quicker and easier access to main forecasting techniques.

Keywords: Forecasting, Hierarchical Forecasting, Grouped Forecasting

Resumo

A digitalização é um conceito emergente e cada vez mais presente no quotidiano das empresas. Para acompanhar este ciclo de mudança, várias empresas veem-se na necessidade de alterar os seus modelos de negócio, adaptando-se, assim, às novas necessidades e imposições do mercado. Um dos principais fatores tidos em conta nesta mudança é a redução do tempo que decorre entre o pedido do cliente e a entrega da solução personalizada que satisfaz as suas necessidades. Para tornar este processo mais rápido, é necessário criar um ambiente com as condições adequadas para permitir um rápido desenvolvimento.

A previsão é uma componente importante em diversas empresas, de diferentes setores que lidam com fatores de incerteza no mercado. A previsão permite o auxílio a diferentes niveis, desde a otimização da operações internas, até à previsão estimada de vendas, antecipando as tendências nos mercados.

No contexto específico da LTPlabs, uma empresa que desenvolve soluções analíticas para apoiar clientes na tomada de decisões mais informadas, as previsões estão presentes diariamente. Quando se inicia um projeto, definem-se as necessidades do negócio e os seus objetivos. Após isto, os dados são recolhidos e analisados estatisticamente, de uma forma global. Seguidamente, são selecionados e preparados para dar início ao processo de previsão. As duas fases seguintes consistem em escolher um algoritmo, implementá-lo e avaliar a sua performance. Este processo segue iterativamente até o resultado da fase de avaliação ser favorável, passando, assim, à implementação da solução final. De projeto para projeto, há implementações de ferramentas de pré processamento dos dados, algoritmos e métricas de avaliação que não são reutilizadas por estarem demasiado adapatadas aos dados de cada projeto específico.

Na revisão de literatura efetuada, não se encontrou nenhuma ferramenta capaz de responder às necessidades identificadas, providenciando uma análise rápida, com um número diminuto de comandos introduzidos pelo utilizador, e, ao mesmo tempo, personalizável para controlar e monitorizar todas as etapas do processo de previsão. Assim, foi proposta uma abordagem automática para o processo e desenvolvida uma ferramenta, incluindo ainda estruturas agrupadas e previsões com variáveis exógenas.

A ferramenta desenvolvida foi testada com analistas no decurso dos seus projetos, tendo sido aceite e incorporada nas metodologias, levando a uma melhoria significativa dos tempos desenvolvimento. Com isto, espera-se uma redução dos custos, assim como um aumento da qualidade das soluções desenvolvidas e implementadas. Em acréscimo, destacam-se as melhorias na padronização do processo de desenvolvimento de previsões, bem como na sistematização do conhecimento relacionado com o tema, permitindo um acesso fácil e rápido às principais técnicas de previsão.

Keywords: Previsão, Previsão Hierárquica

Acknowledgements

Que viagem! É esta a sensação ao olhar para os últimos 5 anos! Não é o fim que imaginei, se é que alguma vez imaginara o final disto. Não sei se levo os melhores 5 anos da minha vida neste pedaço de história, mas levo anos incríveis! Não podia estar mais grata a todos os que me ajudaram a chegar até aqui! Cá estamos, conseguimos!

Obrigada à LTPlabs pela oportunidade de passar os últimos meses a contribuir para um projeto tão ambicioso, numa equipa incrível, rodeada de pessoas sempre à procura de fazer mais e melhor!

Obrigada ao Francisco Amorim por todo o apoio e por ter partilhado tanto do que sabe! Sem dúvida que foi essencial para tudo o que aprendi e para toda a dissertação.

Obrigada ao João Alves pela forma como me acompanhou ao longo desta dissertação, estando sempre disposto a esclarecer todas as dúvidas.

Obrigada ao Professor José Luís Borges, pela disponibilidade e apoio ao longo de todo este período, contribuindo sempre com valiosos conselhos.

Obrigada a todos os amigos que me acompanharam ao longo do percurso. Poder contar convosco, partilhar experiências e criar memórias é, definitivamente, incrível!

Obrigada à família por terem estado sempre lá! Sem dúvida que são o meu pilar em todas as etapas e esta conquista também é vossa! Obrigada à minha irmã por ser a minha inspiração e a minha motivação, em todas as alturas. Obrigada à minha mãe por ser o porto seguro, pelo apoio incondicional e por viver comigo todos os momentos. Obrigada ao meu pai por toda a motivação e por me incentivar sempre a desafiar os limites e a deixar a zona de conforto bem distante!

Obrigada ao João por todo o apoio, por toda a motivação e por termos vivido tanto ao longo destes anos.

Verónica Fradique

"She stood in the storm and when the wind did not blow her way,
she adjusted her sails."

Elizabeth Edwards

Contents

List of Figures

List of Tables

Abbreviations

ADI	Average Demand Interval
ARIMA	Autoregressive Integrated Moving Average
CRISP-DM	Cross-Industry Standard Process for Data Mining
CV^2	Squared Coefficient of Variation
GBM	Gradient Boosting Machine
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
MASE	Mean Absolute Scaled Error
ML	Machine Learning
NN	Neural Networks
pp	Percentage Points
RMSE	Root Mean Squared Error
SARIMAX	Seasonal Autoregressive Integrated Moving Average with exogenous variables
SMA	Simple Moving Average

Chapter 1

Introduction

Increased competitiveness is a trend spreading in the last decades across several industries. Customers demand better service levels, companies press suppliers for decreased lead times and brand loyalty is not what it was a few years ago.

A fundamental interest in deploying an effective data-driven decision-making framework is having general knowledge on how the future is going to play out, regarding some key business variables such as sales, stock consumption and even to understand where is the best place to spot the next company building.

1.1 Context

The forecasting process is hard and consuming an increased quantity of resources, human and computational, for long periods. In business consultancy companies, a substantial part of the projects is associated with forecasting.

This project was conducted with LTPlabs, a Portuguese consultancy company that creates data-driven solutions to leverage clients' procedures in a vast area. Most of the projects this company develops are forecasting projects or other domain projects that require a forecasting module to support some part of them.

The methodology the company adopted, formally or informally, to overcome the challenges of each forecasting process is Cross-Industry Standard Process for Data Mining (CRISP-DM). This methodology divides the forecasting projects into six distinct phases, that are highly correlated with each other. Succinctly describing each step of CRISP-DM:

- Business understanding covers the familiarization with forecasting problems, being the crucial point to understand the goals and to define requirements from a business perspective;

- Data understanding consists of collect, clean and standardize data as well as explore it across different analysis, being an enduring task;

- Data preparation focus is on the definition of the final data structure with all the relevant information;
- Modelling consists of the selection of an algorithm to use as the forecasting strategy and its parameters;
- Evaluation comprises critically analyzing the results obtained previously. When the results addressed in the last phase are satisfactory to meet business needs, the algorithm under analysis is deployed and the forecasting process is complete.

In each project developed, the first and the last phases are unique since they depend on specific business needs and requirements. However, remain four phases are similar independently of the project nature: understanding data, statistically, preparing it, choosing an algorithm, and evaluating it. From a consultancy company's perspective, after each project developed, there are implementations of algorithms, data analysis tools, and evaluation metrics that are not being reused and, consequently, developed multiple times. It leads to previously acquired knowledge that is not being properly used if that are just some users managing it, from an organization's point of view.

In the reviewed literature it was not found a solution to allow a forecasting analysis from end to end with a reduced set of commands. In the market, there several solutions, such as Forecast Pro, [21], that provide support to a significant part of the forecast analysis but are not open source. To combine the knowledge previously acquired by the company, such as recommended strategies, algorithms, or even evaluation metrics, the existence of a framework developed and implemented from the beginning is important, to allow its expansion to adapt to their needs.

Aiming to combine a plethora of algorithms, data analysis tools, and evaluation metrics in a single framework the main goal set for this project is to reduce forecasting projects lead times. Along with this main goal, comes a set of issues that are addressed in the process, as the standardization of steps followed to forecast, an increase of the number of algorithms tested, and systematization of different techniques and knowledge acquired that remain accessible in a single point, reducing the programming needed by providing a high-level framework.

To cover a wide spectrum of projects, it is important to gear the framework with the most used and some state-of-the-art algorithms, a set of tools to help data diagnosis, and a vast quantity of evaluation metrics. Also, the interpretation of a hierarchical structure of data and best level selection as well as the inclusion of exogenous variables, since they can lead to higher accuracy of the forecasts, are included.

To address the advantages and the impact of a framework as the one briefly summarized, there are four case studies to clearly state the reduction of development times, the diminished programming skills required to implement an end to end forecast and simulation of an analyst interpretation of the framework output.

1.2 Dissertation structure

This document is divided into six chapters. It is organized with the following structure:

The present chapter introduced the project, contextualized the problem and the main goals are stated.

Chapter 2 is dedicated to providing an overview over a wide range of forecasting techniques along with a discussion of the complementary needs to give support both to forecasting and implementation.

Chapter 3 describes in detail the current methodology used, the main drawbacks of it as well as a brief description of the proposed approach.

Chapter 4 details the previous proposed methodology. It covers the initial structure of the information, the implementation, and the integration of different tools, algorithms, grouped forecasting, and exogenous variables inclusion.

Then, in chapter 5 the results of the implementation are presented and discussed based on the analysis of four case studies.

Concluding, chapter 6 draws the main conclusions of the project alongside a set of future guidelines for improvements.

Chapter 2

Literature Review

The present chapter aims to provide a general overview of the forecasting techniques available, data analysis tools, evaluation metrics, and implementation. Section 2.1 introduces forecasting techniques from simple to complex ones and addresses the challenges in the selection of the most adequate for a given problem. Data diagnosis tools and data classifications to assess time series characteristics are presented in section 2.2. In section 2.3 the evaluation metrics to obtain a statistical performance of the forecasts are described. To address the implementation of the concepts of the previous section, in section 2.4 technologies are discussed. To complete the forecasting overview, a review of the usage of exogenous variables is described in section 2.5. The chapter concludes with a summary of the main topics discussed, in 2.6.

2.1 Forecasting techniques

Forecasting is an extensive topic and can be used in a wide range of contexts. During this chapter, the term forecasting is applied to time series forecasting. Time series consists of a sequence of values collected over time maintaining the respective chronological order, for a given item, [38], and are studied in different areas, such as statistics, machine learning, and data mining [23].

In the field of time series, there are univariate and multivariate types of forecasting. When modeling univariate time series data, one is concerned with a single target variable's evolution through time [41]. When entering the domain of multivariate forecasting, exogenous variables are taken into account to predict the target variable, addressed in section 2.5.

For a better comprehension of the different techniques available to create univariate forecasts, they are usually divided into two subgroups: **qualitative** and **quantitative** methods [28]. Qualitative methods are based on judgments while quantitative methods are numerical and based on statistics. The following sections focus on data-driven quantitative methods. First, there is an overview of the Naïve and classical techniques, machine learning models, ensemble approaches, and methods for the selection of the most suitable forecasting algorithm is explored. After this, the notion of hierarchical structure is also presented.

Table 2.1: Example of first order differencing.

Date	Series	1st Difference
2020/03/12	100	Nan
2020/03/13	120	20
2020/03/14	145	25
2020/03/15	220	75
2020/03/16	180	-30

2.1.1 Naïve and classical forecasting methods

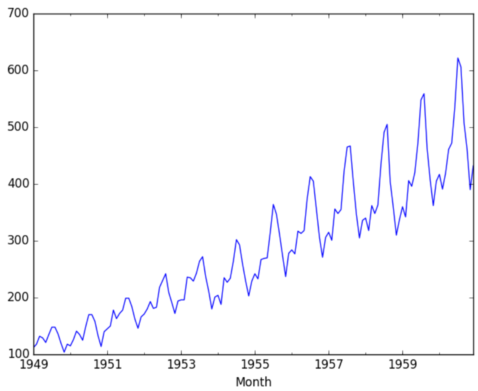
Naïve models applied to forecasting consist of simpler alternatives. Albeit its simplicity they are still used, specially when data is very constant and stable, due to their low complexity. On the other hand, there is evidence that they can be optimal when data follow a random walk [30]. There are several naïve methods, as **seasonal naïve**, predicting the exact value as any other previous period and **drift method** which allows the forecasts to increase or decrease over time based on a drift calculated as the average change observed in previous real observations [30].

Classical forecasting techniques present a wide range of algorithms from simple methods, such as Simple Moving Average (SMA), that calculates the forecasting value based on an average of a defined number of previous observations, to more complex ones. SMA, whence many other methods evolve, is connoted as the base of a high quantity of methods to analyze time series [30].

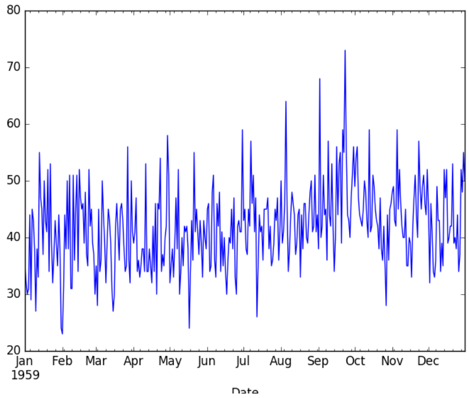
SMA does not adapt quickly to changes in trends and requires the storage of big quantities of data [18]. Exponential Smoothing appears as an improvement of SMA, since it uses the same principle but gives greater weight to the most recent data points [18]. These methods are part of time series models, that also aggregates Autoregressive Integrated Moving Average (ARIMA) and its variations.

ARIMA is complementary to Exponential Smoothing models and aims to describe correlation in the data [30]. This algorithm is a form of regression analysis that gauges the strength of one dependent variable relative to other changing variables. It can be subdivided into three capabilities: Autoregression (AR), Integrated (I) and Moving Average (MA). This method rests on the assumption that the time series is stationary [5]. Time series can be stationary or it can achieve such status through **differencing**. The first-order differenced data takes the increases between timesteps in the original data. Subsequent higher order differencing (order N) applies the same principle to order N-1 differenced data. The original time series is obtained through integration. Differencing concept is illustrated in Table 2.1. In Figure 2.1 there are two graphics representing a non stationary time series 2.1a and a stationary one 2.1b .

Exponential Smoothing methods are weighted averages of previous observations whose weights decrease exponentially as the observations get older. **Holt Winter's Models** is presented as an algorithm integrating exponential smoothing methods, with two variations Additive and Multiplicative. This method is also known as **triple exponential smoothing** due the three smoothing equations that describes it for level, trend and seasonality [30] and the three correspondent parameters α , β and γ. Trend is the component that changes over time and does not repeat itself



(a) Non stationary time series due to visible positive trend.



(b) Stationary time series.

Figure 2.1: Example of time series non stationary and stationary.

and seasonality is the tendency of time series data to have behavior that repeats over time [35]. Summarizing the differences between additive and multiplicative approaches, the additive model is used when seasonality is expressed with the sum of a value, while multiplicative is used when seasonality corresponds to an increase of a percentage.

The previously referred methods have been widely used in different contexts. It is possible to find a considerable number of works that test different algorithms applied to a wide range of areas. For example, in electricity demand preview several works have been published over the past decades. Javedani et al. [33] compared five classical methods, Naïve, Holt Winter's Multiplicative and Additive methods, Decomposition both Addictive and Multiplicative and regression, ARIMA. This study used 66 quarterly data observations with an outlier and was divided into two groups. The first group was used to train the model and the second group to test and compare the results. Concluding, the authors realized that Holt Winter's Multiplicative method obtains more accurate forecasts than the others.

Da Veiga et al. [20] conducted a study comparing ARIMA and Holt Winter's Multiplicative Model, within the scope of food retail demand forecasting. This research counted with 8 years of data of the specific type of products with a short life cycle. The results show that Holt Winter's Multiplicative method has more accuracy when the predictions horizon does not extend the seasonal cycle of the time series. ARIMA model has many assumptions as persistence of historical patterns that, when true, guides to increased accuracy and very good results.

A survey on modeling and forecasting call center arrivals provides some insights on the most used techniques, in practice. Although the evolution of research and development of more sophisticated forecasting methods applied to this domain, the results are not being implemented in real contexts [32]. In this domain, ARIMA was presented as one of the earliest efforts to predict it. With the evolution in standard techniques, this algorithm continued being applied with some improvements such as the inclusion of exogenous variables. Contrasting, the same author indicates that a considerable number of call centers were using simple methods implemented in Excel, at the moment of the study.

A recent study [51] on sales forecasting aimed at presenting a new algorithm selection, compared several forecasting techniques. Focusing on the methods without external factors, 15 methods, ranging from classical to machine learning, where compared. Vouching for classical methods, researchers concluded that ARIMA with additional seasonal components and Exponential Smoothing outperformed the rest.

A more recent approach to address time series forecasting is Prophet, proposed by Facebook, to handle common features of business time series which are characterized by trend, multiple seasonality, and holidays [49]. Prophet is considered interesting to process daily periodicity data with large outliers and shifts in trends with the advantage of modeling several seasonality periods simultaneously [54]. Papacharalampous et al. [42] compared a set of methods to predict hydrometeorological time series, and concluded that Prophet is a competitive method in that domain, despite nothing similar was previously applied. Zhao et al[54] applied Prophet to predict the concentration of fine particulate matter and considered that the algorithm reduces the impacts of problems such as data missing and unexpected outliers. Analyzing the trends of energy consumption on buildings Prophet was compared with ARIMA model Gong et al. [27] concluded Prophet presents lower MAPE then ARIMA. Another study on retail sales forecasting used Prophet to predict the sales of 200 products of the 400 in their catalog. The results obtained with lower MAPE in most of the products for monthly and quarterly predictions denote satisfactory results, also, on this area [55]. Although it is a recently available solution, its implementations and results, denote the potential to be applied to business time series, considering the lower needs of preprocessing and the ability to deal with multi-temporal patterns and trends.

From the main categories of forecasting problems which are highly documented in the reviewed literature, ARIMA models are presented as one that can achieve good performance, in many situations, being interpretable and with low complexity. Simple Exponential Smoothing and, its variations, also figure in the classical algorithms with the highest performance. Simple Exponential Smoothing is suggested also to Intermittent demand, described in section 2.2 , along with Croston's method [46]. These two methods can be compared once they are indicated to the same target. Prophet is presented as an alternative state-of-art with advantages in time series data with strong seasonal effects and several seasons of historical data.

2.1.2 Machine learning applied to forecasting

Machine Learning (ML) can be defined as the ability a computer has to learn from data, and then apply the results of that learning to new information without being explicitly programmed to it. The machine receives a set of inputs and, then, learns from experience [29]. With high access to big quantities of data, and machine learning improving and getting good results, there is growing interest in exploring these algorithms [17]. Due to the capacity of ML, the forecasting domain also can benefit from its usage.

There are several comparisons, in literature, between different algorithms available. These algorithms include, but are not limited to, Neural Networks (NN), Recurrent Neural Networks

(RNN), Random Forests, Gradient Boosting Machine (GBM) and Generalized Linear Model (GLM), briefly described in the following list:

- **Neural Networks** are an approach inspired by how a brain and nerve system works capturing the idea that the system can be nonlinear and non-parametric. It has capabilities to deal with more complex and non-linear relationships.
- **Recurrent Neural Networks** can be described as learning machines that recursively compute new states by applying transfer functions to previous states and inputs.
- **Random Forest** algorithm consists of a set of decision trees which are structured regressors, created during the training and whose output is, for regression methods, the mean of them. Breiman [16] considers random forests effective in prediction with the advantage of do not overfit.
- **Gradient Boosting Machine** (GBM) is a tree-based method that intends to improve prediction accuracy by combining the responses of previously executed algorithms [40] and is used in the context of ensemble techniques as detailed in the next subsection. Inside the context of ML, GBM produces interpretable results, once provides each predictor weight on the final model [53].

Some authors conducted studies comparing different algorithms in different contexts, contributing to an overview of the performance of each one of the methods. Alon [7] compares Neural Networks and traditional methods with the former outperforming the latter in three of the four cases evaluated in the context of aggregate retail sales.

In the context of food sales predictions, Tsoumakas [50] describes some machine learning techniques. This area cares of special attention when forecasting once the products have a short shelf-life and each error can lead to losses, when under or over predicting. Although the author specifies that the machine learning technique is more powerful and more flexible, when implementing in real business contexts the increase of lead times due to the complexity of the algorithms, is not always reasonable.

Forecasting inside a business context cares of an explanation of obtained results and statistics and evaluation metrics may not clarify what is happening to obtain the presented results. Ease of interpretation is considered one of the more important characteristics when selecting a forecasting technique [52]. Moreover, complex techniques are considered a challenge in terms of implementation due to the users' understandability and openness to them [6]. Also, NN has been criticized for the lack of interpretability, as Lasek [37] argued.

2.1.3 Ensemble techniques

The aforementioned algorithms produce forecasts that can be combined in order to obtain more accurate predictions. There are two types of strategies to address this technique, competitive and cooperative [44]. **Competitive ensemble** forecasting aims to combine the values from different models, models constructed with different initial conditions or different parameters. The final predictions are given by a combination of the predictions of the selected methods. This strategy gains with diversity though similar inputs lead to a combination also similar. **Cooperative ensemble** is obtained by aggregate the different predictions obtained by different models when forecasting subtasks of the all prediction objective. There are different strategies to answer the dataset division [44].

Focusing on competitive ensembles, there are several types of ensembles varying from linear combination to tree-based methods. A simple ensemble approach is to take a set of predictions and combine them equally weighted, resulting in a forecasting that is the average of the others. On the other side, GBM calculates the best combination of the predictors, within certain restrictions such as time and number of trees generated.

2.1.4 Algorithm selection

There are several forecasting methods, each one with its specific characteristics, and it is not possible to select the "best" one [22]. One method can produce valuable results under specific criteria and present failures under others.

When it comes to selecting a forecasting method the decision is rather intricate. In [8], Armstrong introduces six different ways of selecting the forecasting algorithm to use, covering criteria that range from qualitative to statistical-based ones. First, the **convenience** by choosing a model the forecaster is used to work with, for example. This strategy might be useful when it is important to have a fast solution or there is not a significant impact induced by forecasting errors. Second, the **Market Popularity** which consists of select a method based on what is being used by others. This strategy can be based on the assumption that if a specific method is being largely applied it may derive from others' research and experience which is also, a risky approach. **Structured Judgement** consists of developing a set of explicit criteria and based on this, compare the different methods. From a previous study inquiring researchers, educators, practitioners, and decision-makers the top-rated criteria used when selecting a forecasting method, accuracy was on the top for all [52]. **Statistical Criteria** strategy is used when comparing quantitative methods and is based on statistics and due to its nature, based on a restricted selection, it can lead to misleading important information. **Relative Track records** consists of comparing the performance of different methods and is a good choice when the predictions have a high impact and, consequently, high errors too. The author considers the last approach useful and reliable but recognizes it imposes higher lead times and costs.

There are more complex approaches to select the correct forecasting method, such as develop a meta-learner that is able to learn from a set of time series the features that characterize each time series, and learn the appropriate forecasting model for each one of them, based on the lowest forecast error [36]. Although this approach leads to a better selection model on the tests developed, comparing four exponential smoothing models, it needs the development of the meta-learner and a wide range of methods is going to increase complexity.

Summarizing, model selection is a complex task and can be executed in different ways. From a statistical and data-driven point of view, the **Relative Track records** can be an accurate choice if the complexity of the tests and the time can be reduced.

2.1.5 Hierarchical structure

Although the algorithm selection is a very important process, under business conditions there is another category with high impact: **data hierarchy**. This concept exists on data that can be aggregated into different levels being an approach capable of integrating domain knowledge [21].

Under the domain of retail sales forecasting, the data can be easily aggregated into different levels. For example, the bottom level corresponds to each product (e.g milk, cheese), that can be aggregated into their category (e.g dairy products), which can be combined with other categories. This is a common practice in retailers once there are effects derived from small changes in prices, for example, whose impact on total revenue, at a top-level, is more easily captured than on-demand [25].

It is possible to have in the same dataset two types of hierarchies, for example, product, as the previous example, and geographic location of the stores that sell the products. The existence of two or more types of hierarchies that can be combined to produce different levels of aggregation leads to grouped time series [30].

There are three different approaches to obtain forecasts when predicting with different levels of aggregation:

- **Bottom-up** approach generates forecasts for each time series at the bottom-level and then sum them to obtain the predictions for all the structure levels.
- **Top-down** approach generates forecasts to the top level, first, and then disaggregates until reaches the lower level in the structure.
- **Middle-out** approach starts by selecting a middle level to generate forecasts. Then, combines both approaches, bottom-up to meet the higher levels and top-down to disaggregate downwards.

2.2 Diagnosis and classification

When conducting forecasting analysis there are several steps that can be performed before applying forecasting methods. The data preprocessing can give important insights and lead to better forecasts. It can vary from highlight interesting relationships between data, detect trends, and identify and omitting outliers [10]. Classifications are important to understand the overall dataset composition and mostly used to obtain information relevant for inventory control [14].

Classifications

Items classification plays an important role in several industries. It can help to understand the inventory items better through different possibilities of classification. Through them, it is possible to obtain valuable information on the items, represented by time series, forecastability and to infer important information to supply chain management, for example. The main classifications are:

ABC: rank items according to the periodic turnover. They can be classified by two approaches - quantity or sales. The principle consists in sorting the items descending by one of the categories, quantity or sales, and the cumulative sum is calculated. Then, to attribute the classification of A, B, or C to each item the distribution is usually based on ideal Pareto principle (80:20) [45]. This leads to less but more valuable A-items and numerous C items with low impact on sold quantity or sales amount.

XYZ: evaluates each item variations in consumption or demand. To obtain it the coefficient of variation is calculated and sorted by it. Then, it calculated the cumulative sum of all the items and, three groups are created based on defined boundaries [4]. Usually, the items are equally divided so each category has 33% of the items. Analyzing, X-items are constant, while Y-items have strong fluctuations in consumption which can be associated with trend or seasonal conditions and Z-items are irregular.

ABC analysis presents a rough classification due to the reduced number of categories [1]. Its quality is improved when ABC is combined with XYZ, constituting a matrix of nine possible classifications. With it, the classification is now based on values and demand frequency. Describing the mapping between these two types of classifications, an AX item has high sales value and constant demand while a CZ has low sales value and sporadic demand.

Smooth, Erratic, Intermittent, and Lumpy classification is based on two values: average inter-demand interval (ADI) and squared coefficient variation (CV^2) [48] and appears as an alternative to classify items forecastability. Each category is defined as follows:

- **Smooth:** ADI < 1.32 and $(CV^2) < 0.49$
- **Erratic:** ADI < 1.32 and $(CV^2) > 0.49$
- **Intemittent:** ADI > 1.32 and $(CV^2) < 0.49$
- **Lumpy:** ADI > 1.32 and $(CV^2) > 0.49$

The graphical aspect of a time series of each category is presented in Figure 2.2, where Variability in demand quantity corresponds to (CV^2) and Variability in demand timing relates to ADI. Smooth items are the ones where forecasting accuracy is expected to be better, while lumpy items refer to the opposite.

All the classifications can be used in data analysis before forecasting, but also to evaluate the performance in each category after it or even to create different models for each category, as suggested in 4.

Outliers

Outliers can be defined as data points with a high deviation from the remaining ones, denoting an abnormal behavior. Outliers can be present in the x-axis and y-axis although, in time series analysis the ones in the y-axis are the most critical.

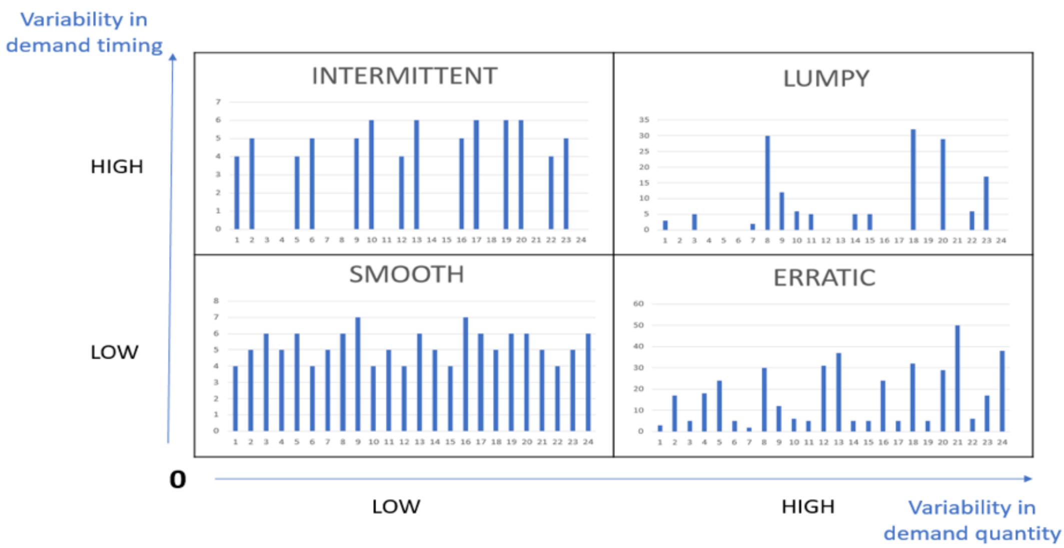


Figure 2.2: Graphical representation of demand in each category of SEIL classification, [26].

Time series data usually has values considered outliers that can affect the different steps of a forecasting analysis like model identification, estimation, and forecasting [47]. The presence of outliers can lead to worst forecasting accuracy [34].

The treatment of outliers can be to remove these data points or replace them with a defined value that can be based on some time series metric.

2.3 Evaluation metrics

Metrics are necessary to compare different models in order to identify which of them produce better forecasts. The metrics selected must give a clear overview of the accuracy of the results.

The process of comparing algorithms starts with a common principle: every model uses part of the data available as a training set, also called in-sample, while the reminiscent data is used to test the model [30]. While the first set is used to fit the model, the second sample uses the previously defined model to forecast and, then, it is possible to compare the resulting set of forecasted values with observed values.

Evaluation metrics can be divided into three types **scale-dependent** errors, **percentage** errors and **scaled** errors [30]. In the first type, the errors are measured on the same scale as the data which means that cannot be compared with other series with different units. Percentage errors, due to its nature, are unit-free and allows comparison between datasets with different units. The last type is an alternative to compare different datasets based on percentage errors. Each type contains different metrics that can be relevant to address different forecasting problems. Amidst those metrics are:

- **Mean Absolute Error, MAE**, which is used when forecast models were applied to a single time series or several, using the same units [30]. As it is not a percentage, its interpretation depends on the specific cases.

- **Mean Squared Error, MSE** is the average of the squared error.
- **Root Mean Squared Error, RMSE**, corresponds to the standard deviation of prediction errors.
- **Mean Absolute Percentage Error, MAPE**, that represents errors as a percentage, for each value forecast.
- **Symmetric Mean Absolute Percentage Error, SMAPE**, is an adaptation of MAPE aiming to equally weight under and over forecasts
- **Mean Percentage Error, Bias**, represents a historical average error, to understand if predicted values are, on average, over or under forecasted.
- **Mean Absolute Scaled Error, MASE**, consists of scaling error based on the training data MAE from the naïve method.
- **Theils' U2 Statistic** is a relative measure that addresses the forecasted values against the result of predict using a naïve strategy. It intends to measure the series forecastability. This statistic penalizes more large errors.

In table 2.2 formulas to calculate error and scaled error and previous described evaluation metrics are presented:

Table 2.2: Evaluation Metrics

Metric	**Formula**	**Type**
Mean Absolute Error, MAE	$\text{mean}(\lvert e_t \rvert)$	Scale-dependent
Mean Squared Error	$\text{mean}({e_t}^2)$	Scale-dependent
Root Mean Squared Error, RMSE	$\sqrt{\text{mean}({e_t}^2)}$	Scale-dependent
Mean Absolute Percentage Error, MAPE	$\text{mean}(\lvert 100\, e_t / y_t \rvert)$	Percentage
Symmetric Mean Absolute Percentage Error, SMAPE	$\frac{100}{N} \sum_{t=1}^{N} \frac{\lvert y_t - \hat{y}_t \rvert}{(\lvert y_t \rvert + \lvert \hat{y}_t \rvert)}$	Percentage
Mean Percentage Error	$\text{mean}(100\, e_t / y_t)$	Percentage
Mean Absolute Scaled Error	$\text{mean}(\lvert q_t \rvert) q_t = \frac{e_t}{\frac{1}{n-1} \sum_{i=2}^{N} \lvert Y_i - Y_{i-1} \rvert}$	Scaled

In the formulas in Table 2.2, the following symbols are:

- **e:** difference between real value, observed, and forecast value;
- **y:** forecast value;

- **N:** total number of forecast values.

These evaluation metrics differ and have advantages in different circumstances and are going to be implemented to calculate forecast accuracy.

Percentage errors, such as MAPE, are scale independent and, for that, can compare the performance of different data sets. Although this advantage, it is not performing so well once exist a predicted null value [31]. Positive errors have a higher penalty than negative errors, using MAPE. Afterward, MAPE is still widely used and preferred, when data has higher positive values, for its simplicity.

Once MAPE weights more over forecasted values, SMAPE is created trying to equal-weight forecasting errors. Although, Hyndman et al. [31] claim that this approach is not correct since it can be undetermined with a relatively common case: when predicted value is 0 and real value is 0, this metric is undertermined.

Hyndman and Koehler compare several accuracy measures and explains the pros and cons of them and also suggested a new measure, MASE. It is suggested when there are different scales and data points negative or close to null values. MASE analysis is, also, quite simple: when is higher than one, it concludes that, on average, naïve method performs well than method under analysis.

When comparing different time series there is a reduced set of evaluation metrics to help to compare the performance of algorithms between them. These methods are MASE and Theil's U2 which intend to measure the time series forecastability. The main difference between them is that MASE is robust when there are actual or predicted values 0 while Theil's U2 is undetermined when the naïve forecasts have a 0 value.

Evaluation metrics are crucial to lead to a complete and informed analysis addressing the quality of the forecasts. Each one of them can be useful under the restrictions and conditions, so it is important to have a wide range of them.

2.4 Technologies

Data-driven solutions are developed using large quantities of data collected through various methods. Its storage is important to ensure data is accessible and fully integrated [24]. Data can be stored into Data Warehouses which can aggregate data from multiple sources. When comparing Amazon Redshift with Azure SQL Data Warehouse, two cloud data warehousing solutions available, Ferreira [24] concludes that Amazon Redshift is easy to configure and setup but is more expensive when there is a necessity to scale data. Otherwise, the second solution computes and storage units are separated which makes it cheaper to increase only one of the components. On the other hand, it has an initial setup that takes longer.

There are available different programming languages and this number continues increasing. According to the IEEE Spectrum ranking of programming languages popularity, Python is leading since 2017. This popularity can be due to many specialized libraries of artificial intelligence [3]. Stack Overflow field a survey, every year, about different topics related to code. From the

2019 survey, Python is considered the fastest-growing major programming language and second preferred language [2].

Python aggregates libraries applied to forecasting and to data analysis that is efficient and effective. When compared to R, another language that has an extensive catalog of libraries applied to data science and forecast, Python is more general-purpose and more practical [29]. Python also has an extensive and active community developing content that can, drastically, decrease development lead times.

Concluding, Python is more indicated to deployment while R is more indicated to the implementation of forecasting algorithms.

2.5 Exogenous variables in forecasts

Exogenous variables can add extra information to the time series that can improve forecast accuracy. Their nature can be internal such as promotional information and last periods sales and also external for example macroeconomic factors or weather, supposing a sales forecasting problem.

VanCalster et al.[51] evaluate the value added by external variables comparing different models with and without them. It is recognized that external variables increase explanation but also leads to higher maintainability costs. The execution of the different models indicates Seasonal Autoregressive Integrated Moving Average with exogenous variables (SARIMAX) as the third best when comparing all models tested and the best forecasting method with external factors, based on MAPE. This model is considered transparent due to the understandability allowing to obtain the different weights of the exogenous variables. SARIMAX is an extension of ARIMA to deal with seasonal effects and exogenous variables. It is defined by seven parameters, three to classify trend, as in ARIMA, and four to characterize seasonality.

Weather is an external factor that can be, also, used as an external regressor. It impacts the economy [11] and is considered as an uncontrollable external factor, multiple times. Although this impact exists, it is a challenge to quantify sales related to weather conditions [9]. Weather can also be addressed as weather risk and can be divided into catastrophic and non-catastrophic [12]. Catastrophic refers to weather conditions not normal, such as storms, hurricanes, and floods among others. Otherwise, non-catastrophic refers to rain, heat, and cold, typical changes, and weather conditions that occur quite frequently [43]. Bertrand and Parnaudeau studied the impact of abnormal weather and stated that if companies that are more exposed to be influenced by weather variability understand how they can mitigate this risk and losses related to **abnormal** weather, they would get improvements and be well prepared to face problems [15].

Adding variables that in the future are also predictions, as weather, which is not known but only forecasted into the future, cares of special attention. In the train sets the values must be, also the predictions and not the verified conditions, predicted with the same time window.

When selecting external variables it means, usually, to increase complexity. Although these models can outperform simple benchmarks there is not enough evidence that this is an excellent approach [25].

2.6 Summary

Concluding the literature review chapter, this section intends to present an overview of the aforementioned topics.

Along this section, the different forecast techniques and some of their field applications were described. From the naïve algorithms, the simpler methods that can either work as forecasting methods or a benchmark to address how well are other methods performing when compared with them. The classical forecasting techniques are explored and it is verified that they are still used and obtaining good results. This can be a consequence of their high interpretability, reduced complexity, and overall implementation and reduced execution times. This is a motivation to take these models into account when forecasting. Inside the domain of ML algorithms, the field results are relevant but when centering on real data and in a business context the increase of complexity, the challenge in creating a generalization applicable to different time series, and the loss of comprehension are barriers to their implementation. Ensemble techniques appear as an alternative to combine previous methods predictions which can happen in more traditional formats or using an ML algorithm, GBM, that can explain how the predictions are generated. With a high variety of methods, the algorithm selection is not an obvious choice and cares for statistical evaluation of accuracy metrics as well as some business expertise. An additional characteristic, hierarchical structure, can add business information due to the grouping of the different items being predicted.

Each forecasting process concerns of analysis before the implementation of algorithms. These analyses can be more general purpose or more specific as classifications to characterize the different time series contained in a single dataset.

The analysis of time series followed by an implementation of forecasting techniques cares of evaluation, in order to simplify the choice of the final algorithm. Evaluation Metrics are the main measure to compare different forecasting strategies. Each one has advantages and disadvantages so, due to the number of different methods that can be tested before selecting the more adequate, it is important to have a wide range of them.

As time series forecasting is a complex process, adding external information as predictors lead to accuracy improvements, which makes it crucial to extend forecasting algorithms to consider these factors. For example, weather forecasting is selected as one variable that could impact most demand and sales in the retail industry, as well as business related characteristics.

Considering that there is not a tool compiling several forecasting techniques allowing users to test and compare results at once having the hierarchical concept implemented automatically and exogenous variables support, the need for a tool combining several algorithms and allowing agile development of forecasting techniques, arises. It is also noted that the inclusion of an analysis of statistics for data preparation and classifications can add value to building a powerful tool to answer business forecasting needs.

Chapter 3

Problem Description

Forecasting constitutes a necessity for multiple industries in order to improve their processes in a wide range of areas. The ability to learn from past events, applying that findings in modeling the future is a great advantage to leverage business. Organizations need to be prepared to face the future under uncertainty with forecasting systems tailored to their business needs. The development of these systems needs expertise in the process of forecasting to: apply a set of different algorithms, evaluate them and after selecting the one best fitting their needs, control its behavior throughout the future [30].

The iterative process mentioned becomes more complex when it is not executed one time to predict an organization future, but multiple times by a consultancy company operating in the development of data-driven solutions to help clients make more informed and better decisions across different areas. This is the case of LTPlabs, whose projects have a high incidence of forecasting. These projects are present on a daily basis, where they can be the main goal of the project or just a step to meet a need in other domains. There is space for improvements on forecasting processes since there are several algorithms and processing techniques to address them, however, they are not combined into a single place in an accessible and standardized way.

This chapter aims at detailing the identified problem and needs. The first section encompasses a detailed description of the forecasting approach followed at LTPlabs, in section 3.1. In section 3.2 the application of the methodology is presented along with the problems verified using it. In the end, holistic description of the proposed approach to mitigate the issues is provided.

3.1 Current forecasting approach

The current forecasting methodology adopted at LTPlabs can be described as a long and complex process. The current approach, followed by a considerable part of the forecasters, is illustrated in Figure 3.1. The Cross-Industry Standard Process for Data Mining, CRISP-DM, is the most used methodology to guide the development of data mining and exploratory projects. Although it was conceived in 1996, it gained popularity in the market around 2000. It is analyzed by Matinez-Plumed[39] twenty years later and is considered still appropriate for predictive processes.

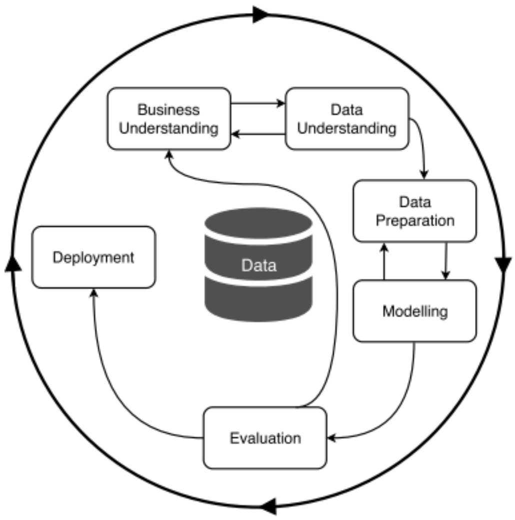


Figure 3.1: The CRISP-DM process model of data mining, still applied to forecasting.

The initial step executed is **Business Understanding**. It consists of understanding the main goals of a project and, consequently, collecting the requirements from a business perspective. The accumulated expertise of those who are familiar with data and have business knowledge is important to help in defining the main objectives and focus of forecasts. The results of Business Understanding stage can change the direction of the forecasting strategy and approach. It is important that business needs and constraints are well defined at the beginning to allow a faster process. Though, the need to complete or exploit information from the business is an available option that can be followed once the evaluation of a model denote it.

The next stage, **Data Understanding** starts with the collection of data and continues with a set of procedures to allow a better comprehension of it. First, it is necessary to collect data that can be from different sources which leads to a need for standardization. The next procedures can be from different natures such as interpret all data metrics, as mean, maximum and minimum values, check size, and attributes or even analyzing it graphically plotting different characteristics. This phase is the most lingering, focusing on increase familiarization with the data.

The **Data Preparation** covers all the steps to create the final time series. At this moment it is necessary to select data, to collect additional data sources, which can be either external, such as weather variables, or internal, as to whether a product is under a promotional campaign or not. Moreover, it is during this phase that is usually identified and removed outliers and selected the aggregation levels.

The **Modelling** phase incorporates the selection of an algorithm to apply and its parameters, the preparation of the train and test sets, as well as selecting the time windows for each one of them, and there is space for its optimization, by choosing more adequate values. This phase is executed multiple times, being selected a different algorithm at each cycle.

Evaluation stage is a deeper analysis of the previously selected algorithm results. It is the moment to confirm the impact of the strategy, comparing with the defined forecasting goals. Moreover, it gives insights on how to improve the model.

When the Evaluation is favorable, the **Deployment** occurs. It comprises the process of transforming the defined algorithm and the test created into the forecasting process of some entity.

The process described, constitutes a long journey, because it is an iterative process that needs high investment in understanding the problem from a business perspective, to analyze data and then to select different algorithms and test them. Each test of a new algorithm requires high effort, which can lead to a reduction of the algorithms tested under time constraints. There is not a suggested number of algorithms to test until selecting the one that is going to be deployed, so the user expertise can be the trigger to understand when the results are good enough. Lack of time can be another trigger to stop the iterative.

Modelling and evaluation stages are repetitive tasks. These two processes are strongly connected since every model tested cares of an evaluation that can be, statistically, based on several metrics as previously described in section 2.3. Evaluation can also benefit from user expertise to critical extract information from them and to map it with the Business Understanding.

3.2 Forecasting application

The current framework followed to address forecasting projects works with projects highly correlated with business, due to the strong correlation with it, aiming to clearly define the business context needs.

Under a consultancy context that creates forecasting models to support multiple areas, these processes lead to time invested in the implementation of algorithms and evaluation metrics for each new project. Those implementations are, in general, tailored to the specific data available which can have different formats and come from many different sources. Although, the algorithms selected can not be the same performing the final solution, the path to achieve it has a number of them coincident.

After each forecasting project, there is knowledge acquired by practitioners. It can provide insights on best practices, model selection for specific cases or conditions, and on the accuracy metrics interpretability, for example.

Each forecasting project can have a different team assigned and the final solution is dependent on them. The previous experiences, the research background, and even the programming skills influence the way that the final solution is going to be traced and carried out. The change of teams, also difficult the spread of feedback, best practices, and strategies followed in previous works.

With the previously referred methodology to address forecasting inside business, there is space to leverage this process as there are some inherent **drawbacks**:

Rework: In each project, there are several concepts implemented such as analysis, algorithms, and evaluation metrics that are common, independently from the context of the project.

Non Standardization: Due to the different teams constituted to solve each new project, the methods used in the implementation of each technique are not concise across the organization. When there is no standardization, each solution will tend to a reaction to the specific problem instead of an application of a deliberated structure common to the entire organization. Although the methodology implemented is the same, there are no guidelines on best practices and recommended approaches for each one of the phases to follow. One of the goals, with standardization, is to reduce costs and consequently improving competitiveness in the market.

High lead times: This is, first, a consequence of the previous two problems. There is a part of the process explained in the previous section, that have to be addressed individually to correctly define the main goals and the focus of the forecasting. However, the implementations of common processes already done can be reused to reduce the development times. Thus, it is possible, from a business perspective, to reduce inherent costs.

Analyzing the forecasting projects completed at LTPlabs, the duration of the four phases, executed iteratively, from Data Understanding to Evaluation is, on average **seven working days**. Throughout an internal survey, it was possible to conclude that there is a work being repeated, mostly due to difficulties in the interpretation of the previous implementations of algorithms and due to being completely focused on the specific data of the project.

The selection of the models to implement is mostly guided by three factors. First, using the previous expertise on similar projects and selecting the ones with the best performance, then, on time series characteristics, obtained by the application of different classification techniques. Lastly, it is taken into consideration if the models being implemented are similar in terms of implementation and requirements or are inside the **same interface**, to reduce development times.

The implementation of an algorithm is usually based on a library from a specific programming language that can contain a single algorithm or a set of algorithms linked in the same way. For example, when selecting a classical algorithm such as ARIMA, the user chooses a library with it and format the data to meet its requirements. Then, all the algorithms present in there can be used in a similar way, since there is consistency in data inputs in the most used libraries. However, if the user intends to experiment a model from a different interface, to address the problem with an ML strategy, for example, the initial data preparation has to be executed again to fulfill the requirements of the new library.

When focusing on the evaluation phase, it is agreed that standardization of this process can lead to a fast observation of the metrics without implementing them saving time to spend in the following steps of the project, when the accuracy is not the focus for the final solution.

3.3 Proposed approach

In this section, it is described the proposed approach to tackle the problems defined aiming to reduce high lead times and contribute to other related processes improvement.

It consists of the development of a framework capable of allowing users o have a solid base to produce forecasts, both for junior users, that can obtain an overview of the forecasting process with lower inputs, and expert users who can add their expertise to obtain more insights or to guide the forecasting process in their own way.

At the initial phase, it is necessary to collect the requirements allowing a better understanding of the needs and main features among the forecasters. Then, it is necessary to structure the framework base of implementation and to define the structure of the information to be incorporated. Entering the forecasting domain, the implementation of data analysis tools is needed as well as the evaluation metrics that are going to allow an overview of the behavior of the algorithms throughout the development. For that reason, the next step is implementing each algorithm and, in the end, extend the framework with additional features to support grouped forecasting and the inclusion of exogenous variables. These sequence of steps is detailed along the following section

When this project started, at LTPlabs, the two initial phases were already completed existing the basic structure and definition of the principal framework units.

Evaluating the performance of the framework proposed, it is tested in four case studies, being three of them real business problems, previously addressed there.

Chapter 4

Methodology

The current chapter aims at detailing the proposed approach to reduce forecasting development effort and ensure thorough testing while generating forecasts in time constraint environments. Section 4.1 details the proposed approach. The high-level framework structure and information organization are explained along section 4.2. The development starts along with the implementation of the diagnostic and classifications described in section 4.3 and section 4.4. In section 4.5 it is covered the inclusion of different evaluation metrics. Section 4.6 describes the process that ensues the integration of forecasting algorithms in a standardized way. In section 4.7 the framework is generalized to include exogenous predictors in an effort to yield more accurate forecasts of the target variable. The chapter concludes with remarks on hierarchical forecasting strategies, in section 4.8. Across all the sections the implementation is described, along with the defined default behavior when some specific information is not defined by the user.

4.1 Proposed approach

The framework succinctly described in the last chapter intends to group the main forecasting techniques, a set of preprocessing tools and evaluation metrics into a high-level framework, mitigating the issues of the current forecasting approach.

The forecasting procedure is extensive, in most of the cases, since it covers plenty of steps until the effective output of a prediction. There is a need to prepare data, which can come from different sources and to analyze it in order to extract valuable information. These initial processes can be executed to reduce the number of human input mistakes, or due to machine incorrect outputs or, even, errors. The initial effort to clean the data and to analyze it is one of the most difficult parts of the procedure. Although there are several packages incorporating forecasting techniques, there are no tools to completely address those steps, into the same packages. The proposed approach to develop the high-level framework is illustrated in 4.1, whose organization is created and projected to support the complete implementation of all functionalities that are going to support the forecasting procedure. The steps guiding the implementation are similar to the followed when

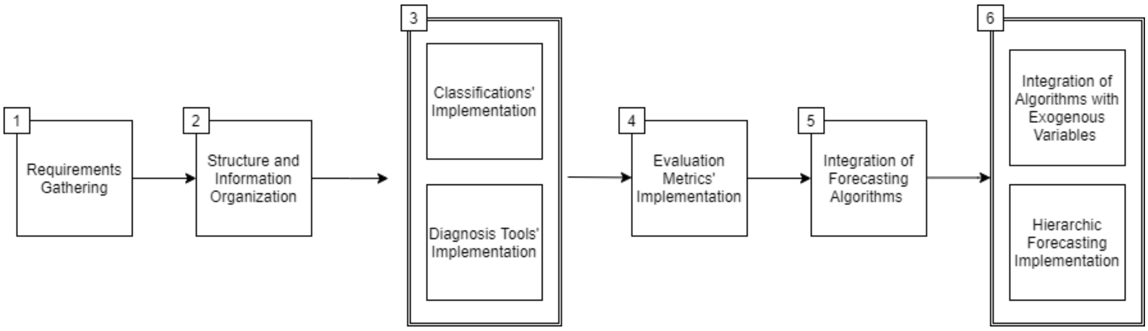


Figure 4.1: Proposed development approach of the framework.

executing an end to end forecasting analysis, except that, during development, evaluation metrics are implemented before the algorithms, as explained in section 4.5.

The first phase consists in gathering requirements in order to determine the main functionalities expected. During this step, the path to follow throughout the development became more clear.

On the next stage, phase 2, structure and information organization, the framework is technically detailed using the previously collected requirements, turning them into a system-specific description. At this point, the main constraints are defined in order to allow a link to all the functions within the module. Thus, the development starts with the implementation of the common structures that underline all the development and are responsible for the framework orchestration. With the main workflow defined, a strong and well-defined base fast-tracks ensuing developments. With an emphasis on grasping the general behavior of time series, the next development steps, phase 3, focus on algorithmic and diagnosis tools' implementations. Phase 4 is the implementation of several evaluation metrics, that are useful to characterize forecasts. From the results of the evaluation metrics, it is possible to compare different algorithms used and to choose the one which best fits the data. At this point, the Framework contains the structure to do the early-stage analysis and to evaluate forecasts. These implementations care for several tests and validations that occur in parallel with the development.

From phase 3 onwards, the development includes testing to ensure a high level of generalization of the solution that has been developed each stage.

The proposed approach addresses what is the expected development of a forecast analysis, however, the evaluation metrics are implemented before the models themselves. This is required to guarantee that during the integration of algorithms, detailed along section 4.6, there are enough tools to evaluate if these implementations are producing the expected results.

4.2 Structure and information organization

To support all the functionalities with different goals, it is necessary to define a structure of the information allowing seamless communication between all the framework components. This structure must be able to let the user communicate with all its components with a reduced set of commands, keeping both, customization and simplicity.

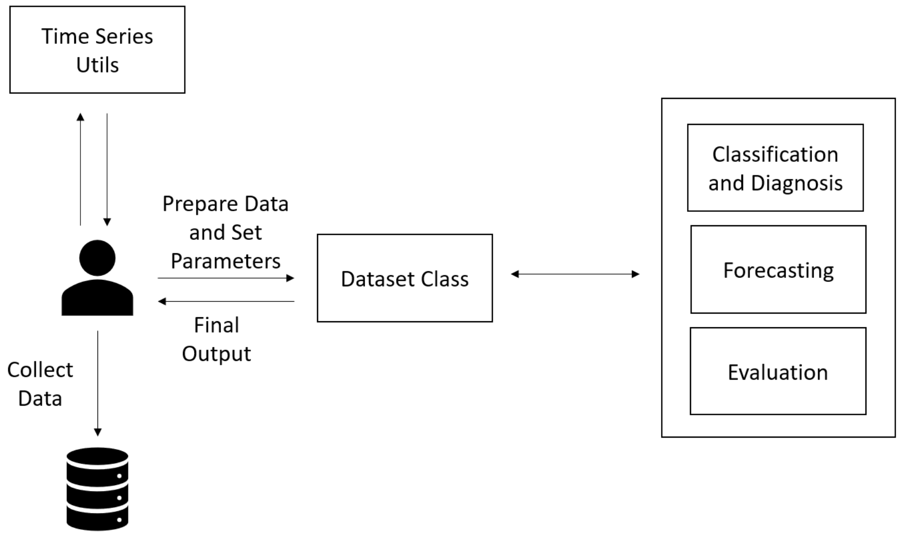


Figure 4.2: Framework interactions.

The user only interacts with data sources, in the early stages of the analysis, and two classes, as illustrated in Figure 4.2. The dataset class works as a wrapper that communicates with the three main components of the framework. All the steps performed by the user consist of direct interaction with this class that calls the respective functions. The user also has access to time series utils class containing a set of analyses to perform data preprocessing. When forecasting grouped time series, instead of a dataset there are multiple dataset instances, forming a grouped dataset object, instance of a grouped dataset class described in section 4.8 with similar behavior to the illustrated in Figure 4.2.

The implementation of this object-oriented structure occurs with Python and R. Python is responsible for all the framework orchestration while R is used for specific tasks such as the implementation of algorithms when it proves to be more adequate.

The basic unit to start forecasts is a dataset object, which consists of a unique instance of dataset class. The user creates it providing the following initial information:

- **Id**: identifier of the dataset
- **Prediction Start**: date/time stamp identifying the earliest point in time for which to generate predictions,
- **Prediction end**: date/time stamp identifying the latest point in time for which to generate predictions,
- **Prediction length**: determines the forecast horizon, which corresponds to the number of predictions to be generated at each time. If N=1 one-step-ahead forecast will be generated. For N>1 one is in the multi-step forecasting domain.

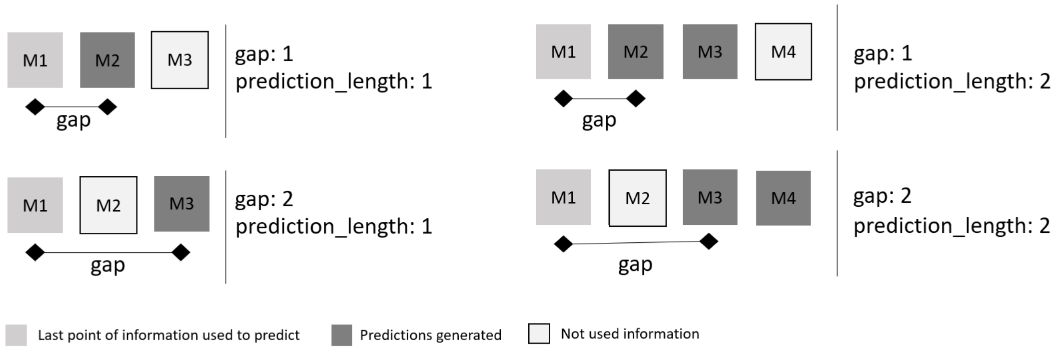


Figure 4.3: Interaction between gap and prediction length.

- **Gap**: an additional time gap between the last observation given to the model for training or fitting and the desired forecast output (with gap = 1 and prediction length = 1, a model is fitted with information up to T=N-1 to generate a prediction for T=N). In Figure 4.3 there are two examples of the outputs generated based on the variation of gap and prediction length.

- **Training end**: delimits the train and test data used to train ensemble models and evaluate the performance of all models used, respectively.

- **Target frequency**: frequency of data ('W', 'M', 'Y', for example). If this argument is not provided, a best-effort heuristic is employed to try to guess it.

- **Sales table**: tabular item containing all the data necessary to create predictions. It must contain an arbitrarily large set of identifiers, followed by a timestamp or date identifier as the index and a single column with the target variable.

Dataset class is the interface between the user and all the framework functionalities. It has the previously referred parameters, that aggregate the information necessary to support all the forecasting processes implemented. It has the main method of prediction, that is responsible to call the algorithm functions that execute each one of them, ensuring parallelization in the process in order to reduce execution times. This is possible since each time series forecast is independent of the others. After completing an algorithm execution on all the time series, their predictions are stored as an attribute of the dataset object. Each classification calculus is a method of this class and each classification is stored in the respective parameter of the object with the classification name.

The defined structure and information organization are crucial to understanding the interactions occurring in the framework and also to have a solid base to implement all the functionalities, detailed in the next sections.

4.3 Diagnosis tools integration

The application of diagnosis methods are considered important steps before developing a forecasting algorithm but are not mandatory. In different scenarios, the use of one or more of them to extract information on data behavior can improve the accuracy of the forecasts. In this set of methods, one performs to identify and plot outliers and is detailed in this section. ACF and PACF help to understand, graphically, how data is correlated with its past values and is implemented in time series utils class, along with Granger Causality Test that analyzes whether a time series can be considered useful to forecast other. Also, a Variance Inflation Factor is implemented to help quantify the collinearity between a set of predictors and a target variable. The latter analysis is particularly insightful when forecasting with exogenous variables, covered in section 4.7.

Prior to forecasting generation, it is recommended to detect **outliers**. The emergence of extreme values can derive from input errors or unexpected or uncontrolled situations [13]. Besides leading to a better comprehension of data, the analysis of outliers can lead to improvements in forecasting accuracy.

Among the plethora of ways to identify outliers and, if need should arise, remove them, the following two are implemented, given their frequent use:

- **Standard Deviation Method**: consider outliers the values positioned a certain number of standard deviations above or below mean.

- **Interquartile Range Method**: consists in create a box-and-whisker plot, which corresponds to the interval between the third quartile and the first quartile. The values out of this interval multiplied by defined value, usually 1.5, can be spread-out. This method is the framework default algorithm, with factor 1.5.

At this point, the framework is equipped with a set of tools to improve data understanding. From now on, the user contains a reliable foundation to create the classifications and forecasts on top of it.

4.4 Classifications' integration

The classifications ABC, XYZ, and SEIL analysis are useful to understand how data is grouped into different categories and can have an important role in forecasting, selecting different models for each category inside a classification, for example, or after forecasting, understanding how the methods are performing on each category of classification, providing interesting insights.

ABC and XYZ classifications divide the dataset into three groups, based on two thresholds that are used against the cumulative volume of item weights. The default values selected for ABC consist of 80 percent to the first group, A, 95 percent to the second group, B, and the last 5 percent consists of C, which means to classify based on item quantities. XYZ default cut values selected to the framework consist of 33 percent for each category.

SEIL classification is based on average demand interval and square of the coefficient of variation follows the rules defined by Syntetos, Boylan, and Croston [48].

All the classifications can be executed at any time of the process and can occur either inside a time window when a lower and upper bound are defined or inside the whole dataset. The calculus of these analyses is integrated within the dataset class thus the classifications for each item are stored as members of the dataset object, remaining accessible at any moment.

Beyond the usage of classifications to do prior data analysis, they are also used as predictors in ML ensemble algorithms, explained in section 4.6.3.2. In this context they are also used to predict inside of each category, using a different ML model trained only on items from each category, aiming to provide important insights due to the higher similarity between the items used during training.

At this moment, the framework allows complete preprocessing of the data. The implemented resources can be used not only to help the forecasting process but, also, in other contexts or, even, isolated to characterize and address different types of problems. From this point onwards, the methodology focuses on detailing the implementations to evaluate and generate forecasts.

4.5 Evaluation metrics' integration

Predictions evaluation is a crucial step to understand if what is being tested is producing valuable insights or not. The most common metrics are implemented, along with some other metrics recommended by experts in the field.

The set of evaluation metrics, whose importance is already described along section 2.3 has MAPE, Bias, MASE, SMAPE, MSE, RMSE. All of them receive two parameters, the set of real values and the predicted values. The standardization in their implementation allows the user to select which of them use.

The evaluation metrics agile the integration of the upcoming algorithms once is possible to track them and to evaluate the outputs while implementing. They work as benchmarks to guide the generalization of each algorithm to make them robust to tackle the different datasets with different types of information.

4.6 Algorithms integration

Every algorithm presented within the framework is into one of four main categories: Naïve, Classical, Ensemble Methods, and Multivariate forecasting. Each forecasting algorithm produces predictions that are stored in the dataset object, so they remain accessible after the execution.

4.6.1 Naïve algorithms

Naïve algorithms are used to generate predictions and can, even if they are not considered good forecasting methods, provide a useful benchmark for more robust methods. Comparing performance using different accuracy measures between these methods and more complex ones is useful

Table 4.1: Default parameters for Naïve methods, according to time series frequency.

Method	**Daily Frequency**	**Weekly Frequency**	**Monthly Frequency**
Naïve lag N	1, 7, 28	1, 4, 52	1, 12
Naïve mean	7, 28	4, 52	3, 12

to evaluate how well are complex methods performing.

The naïve algorithms figuring in the module are divided into two groups: naïve algorithms based on a lag of a defined number of periods, N, and another group, based on an average of N periods.

With naïve lag N algorithms, the next forecast is the same as N periods before, as defined in Equation 4.1.

$$\hat{y}_t = y_{t-N} \tag{4.1}$$

Naïve mean forecast is to predict the actual value with the mean of the last N periods, as described in Equation 4.2.

$$\hat{y}_{t+N} = \bar{y} = \frac{y_1 + \cdots + y_N}{N} \tag{4.2}$$

In both equations, ŷ corresponds to the forecast value, while y represents the real value at a defined time t.

For each of the alternatives described, it is possible to test different scenarios by selecting the different desired values for N. When this selection is not performed by the user, there are a set of default values applied to each of the methods, based on the data frequency. To allow a simple suggestion covering the most common cases, some assumptions can be made based on data frequency. As an example, if daily data and naïve lag N method, the most common pattern would be to predict for Monday, the sales of the last day (N = 1), the sales of last Monday (N = 7) and the same as one month before, translated as four weeks earlier (N = 28). The default selections for each frequency are detailed in Table 4.1.

At this stage, the framework is equipped with simpler forecast algorithms. These algorithms are still used, nowadays, due to its comprehension, and ease of implementation, and when the data is considered stable. They also stand as benchmarks to compare with complex methods, which are detailed in the following sections.

4.6.2 Classical algorithms from R Forecast package

The framework has available all the forecasting classical algorithms present in the R Forecast package. The use of the classical algorithms from this source, all implemented in a standardized way within a unique package, allows a generalization and a wide range of methods implemented and available in a concise way. Once it is implemented for one method, the use of any other method present in the package is possible. Another motivation to use the set of models in the package is that they are already implemented in R, which has a high performance for univariate time series analysis and forecasting, already tested and due to package constant updates and improvements. R Forecast package also includes other types of algorithms, such as the one presented in 4.7.

Once the Forecast package is developed in R it is necessary to link it with Python which supports the framework implementation. This is achieved by using an interface that runs embedded R in Python. Through it, the conversion between the data structures of both languages is direct ensuring simple communication, both in sending and receiving information. The workflow consists of sending the data structure to R with the information needed to predict, such as time series and the relevant parameters. Then, once the process has executed, it returns the point forecasts which are then stored as members of the dataset object.

The default models selected, when the user does not provide a list of models, are Exponential smoothing state space model with Box-Cox transformation, ARMA errors, Trend and Seasonal components, Exponential Smoothing State Space Model, Theta Method, Exponential Smoothing, ARIMA, Croston and Neural Networks.

4.6.3 Ensemble methods

Ensemble methods generate predictions by combining the output of several less sophisticated algorithms to achieve superlative forecast performance. These combinations of methods are flexible to group a wide range of methods being, sometimes, conceptually distinct. There are two types of ensemble techniques, Linear combination, described in 4.6.3.1 and a more complex, Machine Learning Ensemble, detailed along 4.6.3.2.

4.6.3.1 Linear ensemble methods

Linear ensemble models apply linear combination, such as weighted averages, to the outputs of stage-one learners. To cover a wide range of combinations, they can be specified:

Explicitly with a list of all the base models and weights to combine them.

Automatically with a single parameter regarding the number of base models the ensemble is given.

The default selection provided for linear ensemble methods is an automatic ensemble of the top 5 models, based on MAPE, and of all models available. The explicit option has no default values since it is available for user customization.

4.6.3.2 Machine learning ensemble methods

The machine learning algorithms available are selected from the open-source H2O.ai library for ML. Since the algorithms are implemented in a standardized way in this library, it is possible to keep the module able to use several models from there, with no restrictions.

The algorithm receives the predictions previously generated by all the other algorithms and, also, the classifications available. Thus, the ML algorithm contains the base to combine and generate a model based on it. Another functionality implemented, inside this context consists of forecast inside the different groups of classification. In this approach, the data given to training a model is only of a specific category. To obtain the predictions of the entire dataset it is trained a different model for each category, which can improve outcomes but, also, increases the running time. For example, for ABC classification, A items predictions are obtained by a model only trained with A time series, and the same for B and C categories. In the end, each item has its prediction, based on the model trained within its category. Due to the split of the time series items in several categories, if the number of items in at least one of them is lower than 20, the ML ensemble based on classification, is not performed.

The framework contains, at this point, all the functionalities to conduct a forecasting analysis from end to end covering a set of different strategies to predict, from the simpler ones to ensembles with machine learning based on classifications. To complete it, the inclusion of external variables and the ability to group the information into different levels is detailed along the following sections.

4.7 Exogenous variables integration

The inclusion of exogenous variables can help to capture different aspects that can influence the behavior of a time series. For example, the number of ice creams sold increases in sunny days with higher temperatures, so it may be important to consider temperature when forecasting ice cream sales. The algorithm to deal with the external factors selected is SARIMAX. To implement it, there are different approaches and suggestions in the literature. Aiming to compare two libraries found with SARIMAX implementations R Forecast package and Stasmodels in Python, the first step was to conduct some tests to understand their performances. After comparing execution times, predictions accuracy, and ease of generalization in two distinct datasets, the one selected is SARIMAX from R Forecast package. The predictions vary from both implementations due to the different approaches followed to estimate its orders.

SARIMAX algorithm works differently from the classical ones, present in the same package, once it needs external components to define the model. To obtain the orders that define each of its components is used an automatic method provided in the same package. It allows a generalization once it analyzes data and optimizes the parameters accordingly, being faster and not user-dependent. At the same time, it is important to allow user inputs to specify characteristics that can be previously known, so, it is possible to select SARIMAX orders and skip the automatic selection process.

To allow SARIMAX to have the expected behavior, the following restrictions and verifications are implemented, as explained:

- **Minimum Historic:** At least, there must be 22 observations.

- **Remove predictors with no information:** The predictors with 0 in all observations or in one of the train or test sets are removed since they do not add valuable information and conduct the algorithm to throw errors.

- **Predictors Information exists in the predictions period:** SARIMAX is used to predict, only, the target variable, consequently it is expected that all the predictors used to create the model exist in the defined forecast period.

When time series does not have enough historic, execution proceeds, leaving a warning and, in the end, predictions for that specific item appear blank. When there are predictors introduced that are not aligned with the second constraint, they are removed and the execution proceeds with the remain ones.

Along with SARIMAX, VARX appears has a good competitor in terms of accuracy and execution times. VARX implementation is more complex, once it assumes the data is, already stationary. This is not always true, so it needs a preprocessing to imply this feature on data and then, to unroll the differences to obtain real values from the predictions of a differenced time series.

Predicting with exogenous variables imposes a different data structure. Beyond date, and target variable, a set of regressors are introduced. The developments to improve the framework consist of, benefiting from the already implemented features, introducing grouped forecasting strategies.

4.8 Grouped forecasting

Grouped Forecasting expands previously referred forecasting algorithms to adjust the concept to time series that frequently occur and can be aggregated or disaggregated into different levels. This process is based on descriptors provided for each item. As an example, for two descriptors - product and brand - the sales of a specific product can be aggregated into the sales of each brand and several brands can be aggregated for total sales.

The hierarchic levels of a grouped dataset can be divided into three layers:

- **Top:** The upper level of the Group Dataset where information is only aggregated by time or date column with the frequency defined for the Dataset.

- **Middle:** The intermediate layer where all relevant combinations of descriptors are stored.

- **Bottom:** The most disaggregated data available (often each row has a combination of all the available descriptors.

When predicting a specific level of a dataset with a hierarchical structure, it can be achieved by top-down disaggregation or through a bottom-up aggregation of lower levels. The bottom-up strategy is always a sum of the lower levels that give the upper level forecasts. On the other side, the disaggregation approach has two distinct methods:

- **Top-Down Moving-Average proportions** (TDMA): the proportions used for disaggregating the upper level forecasts are calculated through the historical moving-average (over a window of arbitrary length) weight of each.
- **Top-Down Forecasted Proportions** (TDFP): the proportions used for disaggregating the upper level forecasts are calculated through the estimated future weight of each target level using the best performing direct forecasting method.

To produce a grouped forecasting it is possible to select a set of forecasting algorithms or, as an alternative, if no model is provided, there is a default set of models that will be executed, as in the simpler version. Other parameters are optional, but when fulfilled can give important insights about what and how it is being forecasted.

Chapter 5

Results

The current chapter presents the results of the implementation of the proposed methodology divided into three major groups. The first one demonstrates the impact of the adoption of the framework on development times. In section 5.3, to address standardization and ease of use, the results presented demonstrate the sequence of steps necessary to obtain a complete analysis as well as outputs of each step. Finally, a simulation of the high level analysis performed by an analyst after using the framework is demonstrated in one of the case studies.

5.1 Case studies description

To ensure robustness of the framework presented in the methodology, four unique datasets varying in number of time series, observations, frequency, number of forecasting periods, external predictors and number of possible aggregation levels were used. The composition in terms of these categories is presented in Table 5.1.

Case study I supported framework developmentk from the beginning of the project until the total implementation of basic forecasting, which consists of predict without grouped forecasting and external regressors. It corresponds to sales data from a retail company ranging from 2017 to 2019.

Table 5.1: Case studies characteristics.

Case study	Number of time series	Total observations	Frequency	Forecasting period	External predictors	Number of aggregation levels
Case study I	1521	53527	Monthly	6	-	-
Case study II	3795	399654	Weekly	4	-	2
Case study III	1324	49041	Monthly	4	-	6
Case study IV	279	32085	Weekly	4	15	-

Case study II is a public dataset from a UK non-store operating online and contains one year sales of occasion gifts between 2010 and 2011 [19]. The data available was divided into 8 different columns. An item description, invoice date and number, stock code, the quantity sold, unit price, customer and country. The first change in calculate the total amount of purchase, by multiplying unit price by quantity sold in each invoice. Due to the nature of the analysis being performed, the relevant information is items description, invoice date and calculated sales. Therefore, remaining columns were discarded and items description and invoice date were considered the identifiers. With a year of information, the prediction time considered was last 4 weeks. The structure of data only allows for two hierarchies, top and bottom. Through this case study, a strategy of univariate with grouped forecasting is selected to test if the grouped strategy can improve results with minimum information to aggregate.

Case study III introduces the hierarchy concept in a more complete structure when compared with case study II, with six levels of aggregation. It contains sales of a drinks company between 2016 and 2019. The set of descriptors that allow the different hierarchical levels are: product, brand and client. When combining them the result is: top-level, which means predict the total amount of sales independently from each specific product or category, middle layer with four levels for material, brand, client and the combination brand-client and, at last, bottom layer which consists on predicting each item sales individually. In the middle layer, there are combinations skipped, for example, brand-product since each product has only one brand so predict this combination is the same as predicting only product.

At last, case study IV introduces exogenous predictors to complement the sales information. It corresponds to a retail store sales between 2018 and 2019. Predictors are promotional campaigns, lagged sales figures, trend and seasonal components. When addressing this case study, in section 5.4, also a grouped forecasting strategy is implemented with the bottom and top levels only. With this strategy, it is possible to compare the three different strategies in a single case study.

The four case studies were selected aiming to address all the framework capabilities, demonstrating the behavior and performance of it. All of them have dataset parameters `gap` and `prediction length` set to 1. This ensures execution times can be extrapolated as a baseline and allow the future projects duration calculus based on it.

In all the studies conducted, the algorithms selected were automatically recommended by the framework. This default selection, as explained along the previous section differs only on naïve methods options, due to data frequency.

Over the next sections a case will be made on the adequacy of the proposed framework to solve forecasting tasks by demonstrating the results obtained with the datasets along with a comparison with a traditional forecast workflow.

5.2 Lead times impact

One of the main issues that the developed framework proposes to achieve is a reduction in lead times. From forecasting project to forecasting project, the needs vary and there is not an easy

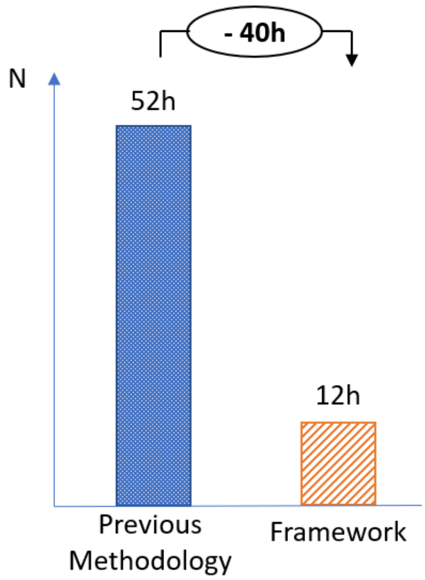


Figure 5.1: Comparison between previous forecasting projects duration and current approach.

mapping between the project constraints or characteristics and the duration of the project. The strategy, here, was to find a duration common to different forecasters based on their previous experiences.

To keep a consistent baseline, this duration lays on a project whose data is already collected from the different sources and cleaned from missing values. It consists of Data Understanding, Data Preparation, Modelling and Evaluation phases of CRISP-DM methodology described in chapter 3. Thus, it is assumed that data is formatted as described in the previous chapter, with identifiers defined, time field is the last position and target variable is the last column of the table. All analysis and classifications are present in the process and taken into account to calculate development times. Solution deployment is not considered in project duration since it depends on external factors, such as specific client needs or rules.

From an internal survey to forecasters at LTPlabs, the average duration agreed for a forecasting project was **seven working days**. Implementing the high-level framework developed the times are reduced to **one day and 4 hours**, approximately 12 hours, as demonstrated in Figure 5.1. This duration is divided into one working day to learning and familiarization with the framework through provided documentation and to produce the analysis and classifications allowing to better understand data and algorithms and evaluation metrics selection, if necessary. The framework has diversity in models and evaluation metrics to cover a high number number of cases, although the user can reduce this sets, based on previous knowledge, and thus reducing execution times. The remaining four hours are related to the complete execution of the forecasting process. Though the case studies conducted last for 2h30 to three hours approximately, a bigger time window is suggested to work as an average when considering similar cases. The case studies execution times are presented in Table 5.2.

Table 5.2: Case Studies execution times.

Case Study	**Duration**
Case Study I	2h30
Case Study II	2h40
Case Study III	3h15
Case Study IV	0h34

During forecasting framework developments, it started to be included on ongoing business projects with a successful performance and acceptance by the analysts. The framework leveraged forecasting procedures reducing development times and providing a fast-track analysis of the performance with a vast selection of methods. This process, also increased feedback during development allowing to incorporate resulting insights.

5.3 High-level framework

To perform an end to end analysis there is a set of commands available, allowing the user to execute each one of the functionalities incorporated in the framework. It is considered an high-level framework due to the level of abstraction provided. The framework can be installed locally. With it, there are three instruction files covering installation and configuration of the environment, complete executions of a basic and a grouped forecasting. The instructions summarized in the following subsections are the minimum to obtain the forecasts, with lower user inputs. The explanation is divided into introduction to forecasting basics and grouped forecasting to map both approaches and results individually.

5.3.1 Introduction to forecasting basics

A basic forecasting process has a set of time series without external predictors and hierarchical structure. It was considered the prototype that, then, was extended to incorporate more advanced techniques and knowledge.

The instructions presented in Listing 1 provide an overview of the programming commands needed for a basic forecasting procedure.

Listing 1: Introduction to forecasting instructions

```
# (1) Dataset Creation
ds = dataset.Dataset(
    id = 'ds'
    , sales_table= sales_table
    , prediction_start= '2011-11'
    , prediction_end= '2011-12-11'
    , training_end='2011-11-20'
    , prediction_length=1
    , gap=1
```

```
)

# (2) Classifications
ds.calculate_<type_of_analysis>() #individually
ds.get_all_classifiers() #all classifiers

# (3) Outliers
ds.plot_outliers(method = 'IQR', identifier = <tuple_of_descriptors>)

# (4) Predictions
ds.predict(models_dict = ds.get_recommended_models_dict())

#(5) Evaluation Metrics
ds.assess_per_model()
ds.assess_per_class(
    model = 'gbm_abc'
    , gap = 1
    , metrics = ['mape', 'bias']
    , target_class = ['abc', 'xyz']
)
```

The presented set of commands covers the forecasting procedure with lower parametrization, where only mandatory parameters are inputted. Moreover, the predictions are executed combined in a single command but the framework allows to predict with each one of the algorithms, individually. This feature can be useful to create a fast check of an algorithm, for example. The following topics summarize the set of commands described in Listing 1.

The first step **(1) Dataset Creation** consists of the creation of a dataset object. To do this, the information needed, as mentioned in section 4.2 and detailed in the instruction are an id, the sales table, prediction window, defined by a start and an end, training end, gap and prediction length.

With the dataset created, the next step **(2) Classifications** parses the different time series in different groups to understand how the data is grouped. Note that the visual outputs only appear when the user sets the `plot` option to True. The output of an ABC analysis is illustrated in Figure 5.2. There are two personalizations available in these classifications:

- Set of the period to calculate classifications which is, by default, using all data points, but can be reduced with two extra parameters: `up_bound` and `lower_bound`.
- Definition of the thresholds by passing a parameter `thresholds` with the two cutoffs. This option is only available for ABC and XYZ once SEIL has fixed values to classify the time series.

These two customizations are, also, available for the instruction to obtain all classifiers. The visual output of this instruction consist of two cross-analysis plots, one showing the percentage of X, Y and Z items in each class the ABC classification, in Figure 5.3, and other addressing the percentage of A, B and C items in each one of SEIL categories.

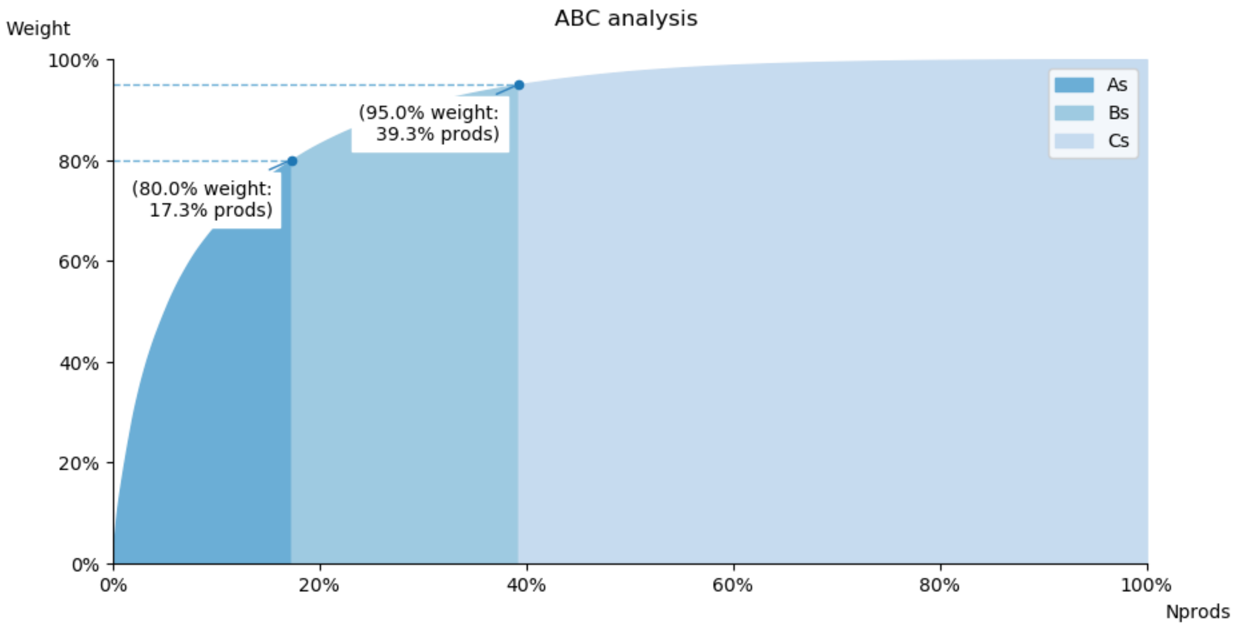


Figure 5.2: ABC analysis plot.

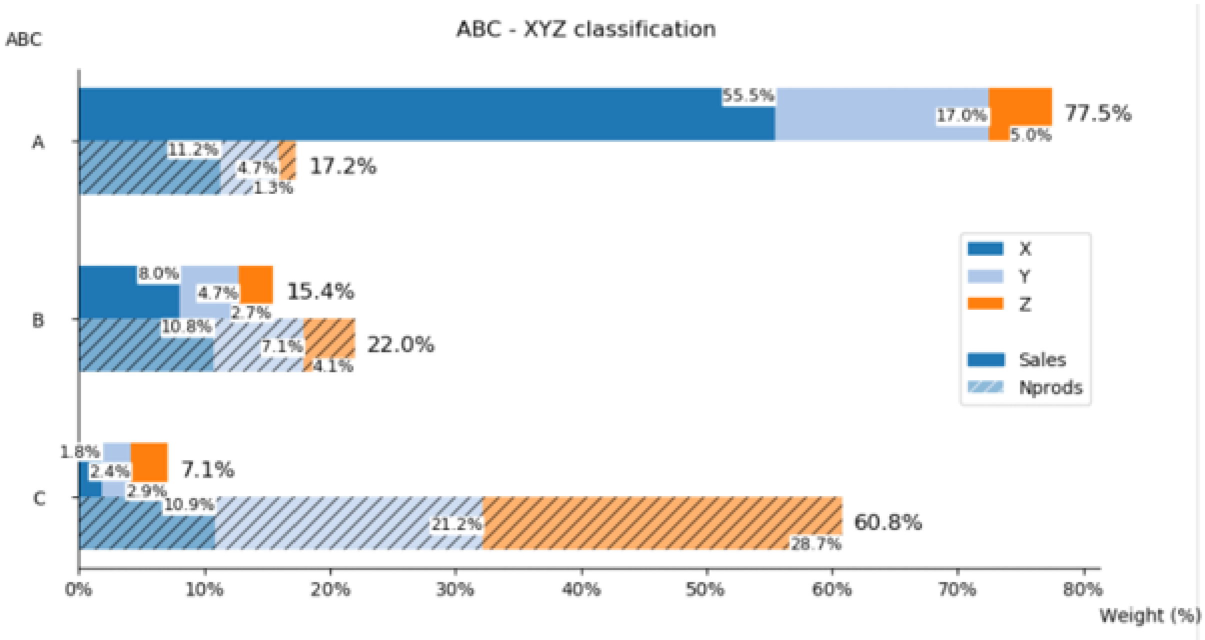


Figure 5.3: Percentage of X, Y and Z in each ABC categories.

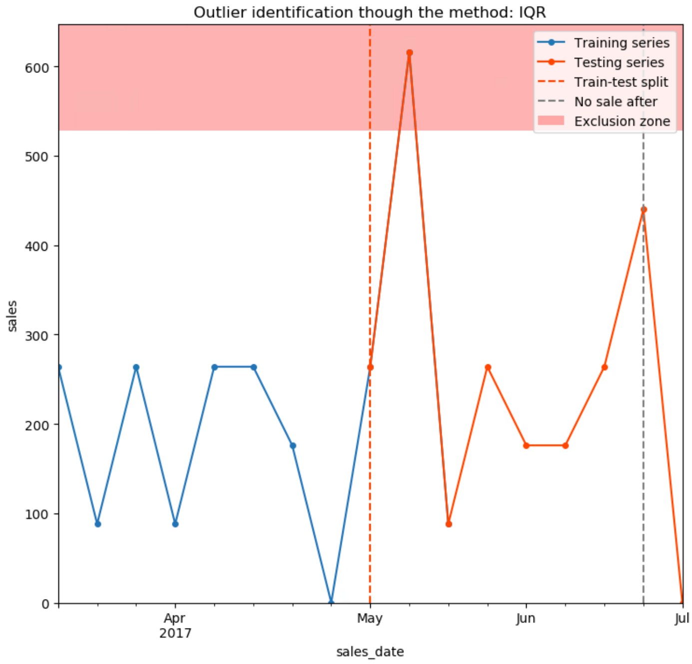


Figure 5.4: Outliers plot through IQR method.

The detection of outliers can give valuable insights into the data. To obtain them with instruction **(3) Outliers**, there are two available methods, Standard Deviation Method, Z, and Interquartile Range Method, IQR, which can be selected through the `method` parameter. Outliers can be obtained for each time series present in the dataset. To select it, the parameter `identifier` should have identifiers of the specific item. A plot of the output of this analysis is depicted in Figure 5.4.

Then, the user has the conditions to start forecasting **(4) Predictions**.To select a specific set of methods, users must fulfill a dictionary with them, with a structure similar to the presented in Listing 2, which represents the default selection. Otherwise, the default set of models is obtained by `.get_recommended_models_dict()`.

Listing 2: Models dictionary with the default models to weekly data.

```
models_dict={
    'r_models':['arima', 'ets', 'tbats', 'thetaf', 'ses', 'nnetar', 'croston']
    , 'naive_models' :[
                ('lagn_naive' , [1, 4, 52]),
                ('naive_mean', [4,52])]
    ,'linear_ensemble_models' : {
                'ens_auto' : [-1]}
                ,'ml_ensembles' : ['gbm', 'glm']
    ,'ml_per_class_ensemble' : {'gbm' : ['abc', 'seil']}}
```

After predicting, results are stored in the dataset object and the final output provided to the user is obtained containing each model evaluation metrics. The results are presented in a table format

	point_forecast	actual	err	abs_err	err_sq	n_forecasts	mae
model							
gbm_seil	3.5109e+06	3631178	-120317.1834	1.1780e+06	2.9547e+09	5280	223.0972
gbm_abc	3.5125e+06	3631178	-118632.1281	1.1838e+06	2.9368e+09	5280	224.1966
gbm	3.5154e+06	3631178	-115803.9100	1.1907e+06	2.9746e+09	5280	225.5077
naive_mean_last_3	3.5840e+06	3631178	-47136.6667	1.2478e+06	2.1839e+09	5280	236.3331
ens_all	3.6491e+06	3631178	17920.7470	1.2495e+06	2.1683e+09	5280	236.6474
ses	3.5340e+06	3631178	-97216.9888	1.2733e+06	2.3150e+09	5280	241.1603
ets	3.6430e+06	3631178	11868.2996	1.3076e+06	2.7233e+09	5280	247.6558
glm	3.5903e+06	3631178	-40844.5437	1.3463e+06	2.5254e+09	5280	254.9810
thetaf	3.6866e+06	3631178	55465.1397	1.3502e+06	4.1348e+09	5280	255.7114
croston	3.6866e+06	3631178	55465.1397	1.3502e+06	4.1348e+09	5280	255.7114
arima	3.7581e+06	3631178	126922.3929	1.3903e+06	3.5149e+09	5280	263.3052
lag_1_naive	3.7792e+06	3631178	148067.0000	1.4126e+06	2.9997e+09	5280	267.5301
naive_mean_last_12	3.4254e+06	3631178	-205780.2826	1.4998e+06	4.1893e+09	5280	284.0610
nnetar	3.9060e+06	3631178	274809.9507	1.6586e+06	6.8108e+09	5280	314.1228
naive_mean	3.3270e+06	3631178	-304182.9724	1.7824e+06	6.4917e+09	5280	337.5698
lag_12_naive	3.8979e+06	3631178	266706.0000	2.0916e+06	7.6310e+09	5280	396.1337
tbats	4.6226e+06	3631178	991441.3877	2.1746e+06	1.3791e+10	5280	411.8481

Figure 5.5: Part of evaluation metrics table provided by the framework.

and are extensive, anticipating the needs of various stakeholders: analysts that should have all the information to allow informed decisions and business-oriented stakeholders who focus on easily interpretable metrics. The results presented only correspond to `gap` equal to 1. For higher gaps, there is a similar evaluation table for each one. The performance of the models tends to deteriorate with the increasing distance between last given observation and requested prediction. Although, predicting with gaps higher than 1 is a common task when, for example, the production planning has to be anticipated. In Figure 5.5 displays a part of the results table. The complete evaluation table is presented in appendix A.

It is possible to address results per class with the second command at **(5) Evaluation Metrics**. It gives an overview of overall performance by each category of the classification, as illustrated in Figure 5.6. This type of statistics can be based, also, on SEIL classification mapped with any metric, or in a single classification.

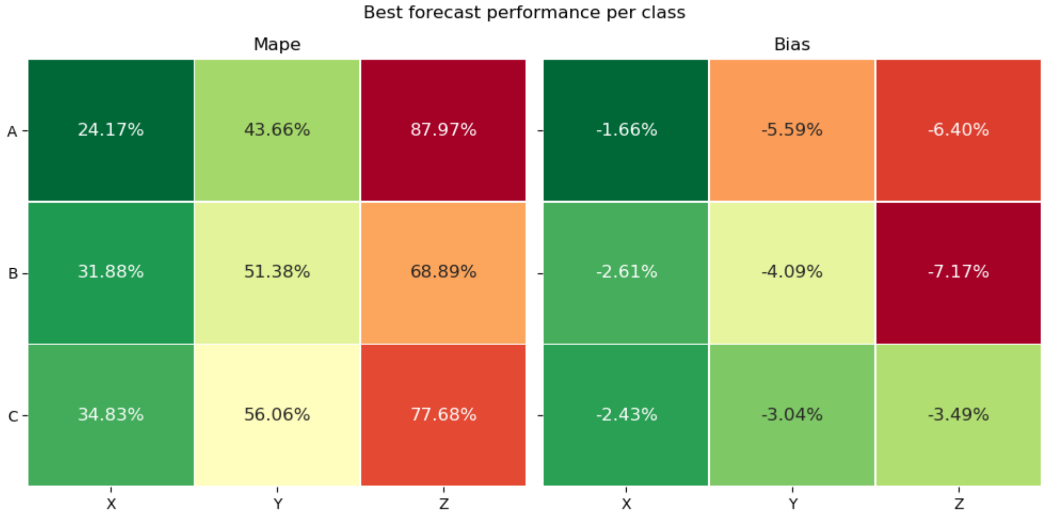


Figure 5.6: Evaluation of forecasts in terms of MAPE and Bias in classifications ABC and XYZ.

5.3.2 Grouped forecasting

Grouped forecasting extends the previous concepts to adjust them to a reality where time series can be naturally aggregated or disaggregated by different attributes. Listing 3 presents the set of instructions to obtain grouped forecasts. As in the previous example, it contains only the essential steps to predict.

Listing 3: Grouped Forecasting Instructions

```
# (1) Grouped dataset creation
g_dataset = gdataset.GroupedDataset(
    id = 'ds'
    , sales_table= sales_table
    , prediction_start= '2011-11'
    , prediction_end= '2011-12-11'
    , training_end='2011-11-20'
    , prediction_length=1
    , gap=1
    , freq='D'
)

# (2) Data Organization
g_dataset.get_summary()
g_dataset.get_gsales_object()

# (3) Predictions
g_dataset.grouped_forecast()

# (4) Evaluations
fmethods, summary = g_dataset.get_best_forecast_models()
```

```
summary[gap]['bottom'] #assess bottom level
g_dataset.which_is_better('bottom') #best model predicting bottom
```

When predicting data with a hierarchical structure, the flow is similar to basic forecasting. The analysis and classifications were removed to focus on the core of a grouped forecasting approach. The first step **(1) Grouped dataset creation** is essential to provide the information necessary to predict. The main difference is creation of an instance of `GroupedDataset` class. In the set of parameters, the user can provide `freq` with data frequency. If it is not present, a best-effort strategy is followed to infer it from the data.

The first instruction of **(2) Data Organization** provides detail dataset organization and composition of the distinct classes, while the second one aggregates the target amount for each level within the hierarchical structure. The second command is also responsible to create a dataset, which is an object of dataset class introduced in the previous subsection, for each level, being accessible by `.datasets`.

Predictions are obtained with instruction **(3) Predictions** that has an optional `levels` parameter to select desired levels to predict. When this parameter is not provided, the framework performs calculations for all relevant levels automatically obtained.

In terms of evaluation, set of instructions **(4) Evaluations** contains three instructions. The first one evaluates all the available direct forecasts for each level of the hierarchy and each gap, returning the best forecasting method, per model, per gap, along with a leaderboard of the evaluation metrics calculated (sorted by an evaluation metric, where 'MASE' is the default). There are two outputs of this instruction: `fmethods` with the best method per level according to the defined evaluation metric and `summary` with the complete evaluation metrics tables for all the levels whose output is presented in Figure 5.7. The specific table for a level can be obtained with the second instruction inside **(4) Evaluations**. The last instruction calculates automatically the best path to reach the forecasts at a given target level. To achieve it, the command explores all viable aggregation and disaggregation paths available.

The inclusion of exogenous variables through SARIMAX is implemented but is not incorporated in the framework. Although, it was possible to conduct tests addressing its behavior, with case study IV.

5.4 Case study overview

In the presented case studies all the models were selected, to provide an overview of the capabilities of the framework. In a real context, some models could be avoided, reducing execution times. The type of analysis demonstrated was performed for each case study, however the detailed results presented in this section refer to case study III.

Typifying an analysis performed by a user of the framework, an high-level overview of the evaluation metrics table, presented in the previous section, was created. The main purpose of it is to summarize the results of a case study and slightly demonstrate the performance of each category

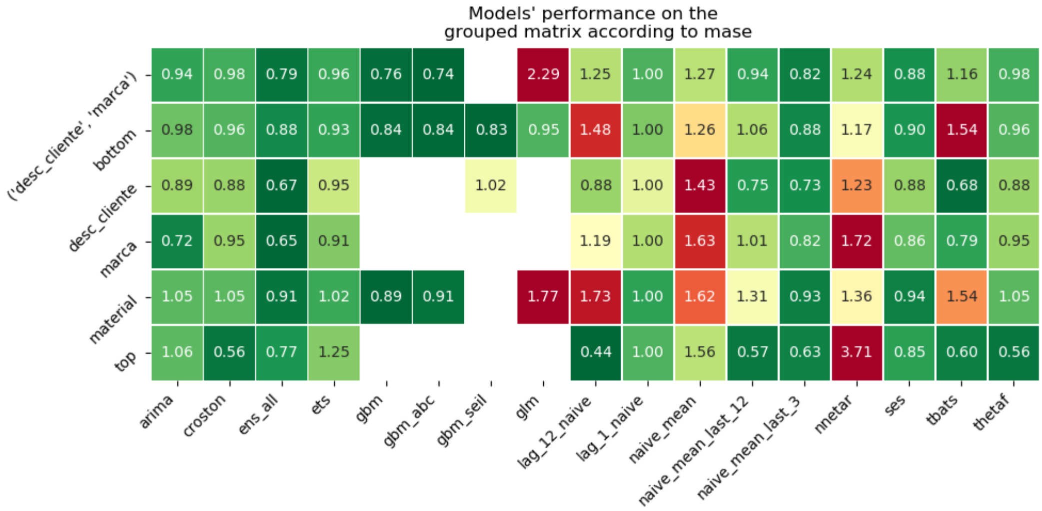


Figure 5.7: Evaluation metrics per level and model with gradient color from dark red (worst) to green (best). The blank spaces refer to ML algorithms which have a minimum of 20 items per model.

of models. Accordingly with this, there are six categories for a univariate analysis without external predictors. Each one represents the following:

- **Lag 1 naïve** is considered a benchmark. When any other model is performing worst than predicting the same as the last period, that algorithm can be discarded once its complexity is necessarily higher. This method is the starting point of this evaluation and comparison.

- **Best Learner** is the method from the R Package performing well on the evaluation metric selected.

- **Linear Ensemble** corresponds to the ensemble model whose performance was better.

- **ML Ensemble - GBM** is chosen because has demonstrated higher performance when comparing with GLM over the realized tests during developments. This model is used, also, to extract insights on the power of predicting based on each classification.

- **GBM Classification** is selected because enters another category because is a specification of the GBM case. At this point, it is possible to compare both, knowing that the second approach is complex since it trains three or four different models.

- **Best Direct** is present to allow an overview of the previously referred methods. With its metric, it is possible to understand the evolution from the beginning, with the simplest method, to the best solution provided in the univariate selection of models. This model is going to be the reference when comparing with grouped alternatives.

- **Best TD** is referred when a grouped forecasting strategy was conducted. As the target level of the case studies was the bottom level, in the summary results it is presented the best performance with a Top-Down approach.

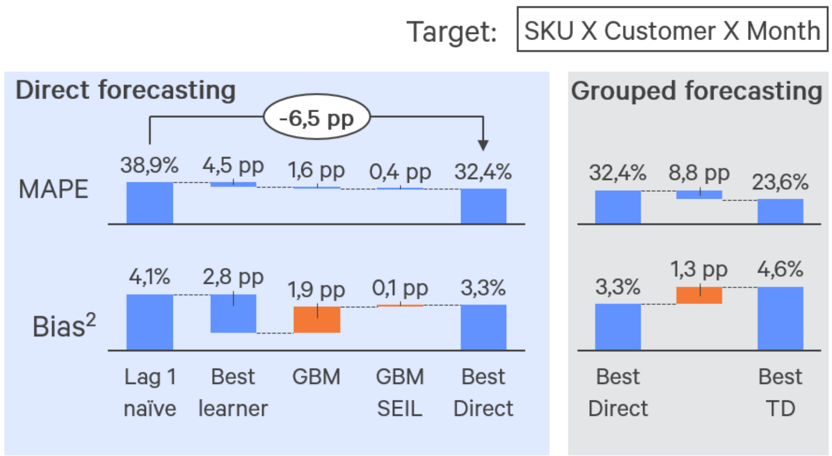


Figure 5.8: Evaluation metrics on case study III, with grouped forecasting.

The main outputs of this analysis is to address what an end-user of the framework can expect to obtain in a reduced time window. It was possible to interpret results in order to understand if the accuracy was improving with the increase of complexity. The results for case study III are presented based on two evaluation metrics: MAPE and Bias.

The evaluation in terms of accuracy in grouped forecasting is created once the dataset contains identifiers that comport a meaningful hierarchical structure. The comparison is between the Best Direct, previously selected, and the best aggregation level performance then disaggregated, if it was not the bottom level until reach it. In Figure 5.8 the overall comparison between the direct forecasting ends with the Best Direct model as GBM based on SEIL classification and due to the predictions within the hierarchical structure, the best approach reduced MAPE in 8,8pp. The Bias increased but continued acceptable being lower than 5%.

Table 5.3 presents a summary of the results obtained for case studies I, II and IV with a similar analysis as the one previously described. The three categories stand for the best performing model, according to MAPE and the total improvement, from Lag 1 naïve to the best model in MAPE and Bias.

In case study I a basic forecasting approach was executed, resulting in the GBM based on classificafion ABC as the top algorithm.

In case study II and IV the implemented strategies were grouped forecasting, although the datasets do not present an hierarchical structure. They only present two levels, top and bottom.

Table 5.3: Forecasting results of case studies I, II and IV.

Case study	**Best Model**	**Best MAPE**	**Total Improvement (MAPE)**	**Total Improvement (Bias)**
Case study I	GBM ABC	37.4%	13.3 pp	-3pp
Case study II	tdfp, top	63.6%	7.2pp	1.9 pp
Case study IV	tdfp, top	25.8%	13.5pp	-4.1pp

In both cases predict aggregated sales and disaggregate, based on Top-Down Forecasted Proportions, to bottom level resulted in best results than predict directly the bottom level. However, the improvements in MAPE are only 0.2pp in case study II and 0.6pp in case study IV. The decision between select the top performing algorithm at bottom level, performed as a univariate basic forecasting or the grouped approach is up to the user, based on business needs and other relevant metrics.

Case study IV included external predictors, with SARIMAX, and the performance, in terms of MAPE, was 31,1%, similar to Lag 4 naïve. This result may be associated with the type of predictors used.

Chapter 6

Conclusions

Agile processes within organizations lead to less time spent and also to the transmission of knowledge and information within them. If the knowledge acquired is spread inside organizations, it can help to solve future similar problems.

The current forecasting methodology is a complex process and depends on the experience of users. A combination of algorithms at a unique platform allowing a standardized baseline to help selection of the best solutions would improve forecast analysis.

The developed project focused on improving forecasting strategies being practiced by forecasters, based on the existent needs of the market and confirmed by LTPlabs experience. The objective was to propose and develop a methodology to combine analytical knowledge under the domain of forecasting covering the procedure from end to end. The main goal was the reduction of lead times in all types of forecasting projects. Along with this objective, a set of minor nice-to-haves arose to improve the expected gain.

This closing chapter begins by exposing the main implications of the solution obtained. Then, the last section describes future work.

6.1 Implications for practice

The key result of this development is the leverage of forecasting processes characterized as long and expensive due to their exploratory and iterative nature. The current methodology being followed at LTPlabs is, officially or, usually, in an informal way, CRISP-DM. It recognizes the strong connection between business needs and the project development stating this clearly by considering it the first part of the process and that at the end of each cycle there is space to revise the business objectives. The developed high-level framework focus on improving four phases of CRISP-DM methodology: Data Understanding, Data Processing, Modelling and Evaluation.

The combination of a vast set of tools prepared to work automatically with low user input but, also, allowing users to incorporate expertise and to access each of the functionalities individually was not found in the current market options for forecasting projects. Although there are a set of libraries combining forecasting procedures or evaluation metrics and even outliers detection, it was not possible to find one that provides an overview of the entire needs of a real forecasting project. These libraries are used in the framework development to agile implementations and due to the advantages from its own implementations and design that are robust.

The developed framework contains a baseline that starts with tools to exploratory analysis of data, helping to a better characterization of it. Then, covers the classifications that are used to how data is grouped in different categories, and that are going to support the ensemble forecasting based on each one. At this point, the user contains the knowledge to underline a forecasting process. Inside forecasting, specifically, there is a wide range of methods that can address most of business problems starting by simpler ones, in most of the cases, to establish a baseline, the classical ones from autoregression to exponential smoothing whose results are still applied in different industries and a class of linear and machine learning ensemble methods. Then, to address the results of each model were implemented a set of evaluation metrics to address results. It is possible to extend the framework to include different tools and algorithms, increasing the available alternatives to enrich it, allowing to incorporate users knowledge into it.

To complement the basic forecasting structure, an automatic treatment of hierarchical and grouped datasets was implemented. Along with this extension of the framework, an implementation of an algorithm to deal with exogenous variable was implemented. At this point, it is not yet implemented inside the framework .

The automatic forecasting process aims at reducing development times, generalizing the framework and at providing an overview of a higher set of possible solutions. When forecasting, parameters optimization can lead to accuracy improvements but, usually, the gain is not significant. Accordingly, the followed approach was to use automatic forecasting generation as the default option.

In practice, a reduction of 40h in the forecasting projects at LTPlabs is achieved when using the forecasting framework, along with the advantages of the standardization of the inherent processes. The framework was used to address ongoing projects, during developments, and its impact was positive, leveraging forecast analysis. Its adoption by analysts at LTPlabs is motivated by the issues they find when repeating processes from previous projects, spending time in implementation details that could be used to focus on forecasting issues and strategies definition.

6.2 Future work

Despite the significant improvements by using the developed framework it still remains space for improvements and to achieve superlative results. The first step is to incorporate SARIMAX into the developed framework covering all the analysis proposed inside the same framework. Also, at this point, ML algorithms are not fully explored and covered due to the hard task of creating a strategy able to generalize and address different problems without implying users to discover and manually set their parameters. A strategy of automatically selecting the best performing algorithm as the one explained in [51] aiming to underline the decision on improving sales profit would be interesting to compare with the current approach based on statistical evaluation metrics and analysts expertise. In terms of implementation, the process can evolve to a distributed implementation, with advantages in scalability and in the resilience provided.

Appendix A

Complete evaluation metrics table

To keep the body of the thesis at a reasonable size, the complete table of evaluation metrics, that is one of the outputs of an end to end forecasting analysis performed with the developed high-level framework, is presented in A.1. It provides evaluation metrics sorted by a specific metric, being MASE the default and the one selected in this example.

	point_forecast	venda	err	abs_err	err_sq	n_forecasts	mae	mse	me	bias	mape	accuracy	smape	mase
model														
gbm_seil	3.5109e+06	3631178	-120317.1834	1.1780e+06	2.9547e+09	5280	223.0972	5.5960e+05	-22.7873	-0.0331	0.3244	0.6756	0.6766	0.8339
gbm_abc	3.5125e+06	3631178	-118632.1281	1.1838e+06	2.9368e+09	5280	224.1966	5.5622e+05	-22.4682	-0.0327	0.3260	0.6740	0.6123	0.8380
gbm	3.5154e+06	3631178	-115803.9100	1.1907e+06	2.9746e+09	5280	225.5077	5.6336e+05	-21.9326	-0.0319	0.3279	0.6721	0.6890	0.8429
naive_mean_last_3	3.5840e+06	3631178	-47136.6667	1.2478e+06	2.1839e+09	5280	236.3331	4.1362e+05	-8.9274	-0.0130	0.3436	0.6564	0.5707	0.8834
ens_all	3.6491e+06	3631178	17920.7470	1.2495e+06	2.1683e+09	5280	236.6474	4.1066e+05	3.3941	0.0049	0.3441	0.6559	0.6102	0.8846
ses	3.5340e+06	3631178	-97216.9888	1.2733e+06	2.3150e+09	5280	241.1603	4.3846e+05	-18.4123	-0.0268	0.3507	0.6493	0.6361	0.9014
ets	3.6430e+06	3631178	11868.2996	1.3076e+06	2.7233e+09	5280	247.6558	5.1577e+05	2.2478	0.0033	0.3601	0.6399	0.6360	0.9257
glm	3.5903e+06	3631178	-40844.5437	1.3463e+06	2.5254e+09	5280	254.9810	4.7830e+05	-7.7357	-0.0112	0.3708	0.6292	0.7515	0.9531
thetaf	3.6866e+06	3631178	55465.1397	1.3502e+06	4.1348e+09	5280	255.7114	7.8310e+05	10.5048	0.0153	0.3718	0.6282	0.6224	0.9558
croston	3.6866e+06	3631178	55465.1397	1.3502e+06	4.1348e+09	5280	255.7114	7.8310e+05	10.5048	0.0153	0.3718	0.6282	0.6224	0.9558
arima	3.7581e+06	3631178	126922.3929	1.3903e+06	3.5149e+09	5280	263.3052	6.6571e+05	24.0383	0.0350	0.3829	0.6171	0.6023	0.9842
lag_1_naive	3.7792e+06	3631178	148067.0000	1.4126e+06	2.9997e+09	5280	267.5301	5.6813e+05	28.0430	0.0408	0.3890	0.6110	0.5853	1.0000
naive_mean_last_12	3.4254e+06	3631178	-205780.2826	1.4998e+06	4.1893e+09	5280	284.0610	7.9344e+05	-38.9735	-0.0567	0.4130	0.5870	0.6330	1.0618
nnetar	3.9060e+06	3631178	274809.9507	1.6586e+06	6.8108e+09	5280	314.1228	1.2899e+06	52.0473	0.0757	0.4568	0.5432	0.6853	1.1741
naive_mean	3.3270e+06	3631178	-304182.9724	1.7824e+06	6.4917e+09	5280	337.5698	1.2295e+06	-57.6104	-0.0838	0.4909	0.5091	0.6642	1.2618
lag_12_naive	3.8979e+06	3631178	266706.0000	2.0916e+06	7.6310e+09	5280	396.1337	1.4453e+06	50.5125	0.0734	0.5760	0.4240	0.8132	1.4807
tbats	4.6226e+06	3631178	991441.3877	2.1746e+06	1.3791e+10	5280	411.8481	2.6120e+06	187.7730	0.2730	0.5989	0.4011	0.7323	1.5394

Figure A.1: Comparison between previous forecasting projects duration and current approach.

References

[1] A Review on Inventory Management Control Techniques: ABC-XYZ Analysis. Technical report.

[2] Stack Overflow Developer Survey 2019.

[3] The Top Programming Languages 2019 - IEEE Spectrum.

[4] Esra Agca Aktunc, Meltem Basaran, Gozde Ari, Mumin Irican, and Sahna Gungor. Inventory Control Through ABC/XYZ Analysis. pages 175–187. Springer, Cham, 2019.

[5] Hesham K. Alfares and Mohammad Nazeeruddin. Electric load forecasting: Literature survey and classification of methods. *International Journal of Systems Science*, 33(1):23–34, jan 2002.

[6] Özden Gür Ali, Serpil Sayin, Tom van Woensel, and Jan Fransoo. SKU demand forecasting in the presence of promotions. *Expert Systems with Applications*, 36(10):12340–12348, dec 2009.

[7] Ilan Alon, Min Qi, and Robert J. Sadowski. Forecasting aggregate retail sales:. *Journal of Retailing and Consumer Services*, 8(3):147–156, may 2001.

[8] J. Scott Armstrong. Selecting Forecasting Methods. *SSRN Electronic Journal*, (August), 2012.

[9] Nari Sivanandam Arunraj and Diane Ahrens. Estimation of non-catastrophic weather impacts for retail industry. *International Journal of Retail and Distribution Management*, 44(7):731–753, 2016.

[10] A Azadeh, M Sheikhalishahi, M Tabesh, and A Negahban. The Effects of Pre-Processing Methods on Forecasting Improvement of Artificial Neural Networks. *Australian Journal of Basic and Applied Sciences*, 5(6):570–580, 2011.

[11] Florian Badorf and Kai Hoberg. The impact of daily weather on retail sales: An empirical study in brick-and-mortar stores. *Journal of Retailing and Consumer Services*, 52, jan 2020.

[12] Erik. Banks and Inc. Element Re Capital Products. *Weather risk management : markets, products, and applications*. Palgrave, 2002.

[13] Sabyasachi Basu and Martin Meckesheimer. Knowledge and Information Systems Automatic outlier detection for time series: an application to sensor data. *Knowl Inf Syst*, 11(2):137–154, 2007.

[14] D. Clay Whybark Benito E. Flores. *Multiple Criteria ABC Analysis*. International Journal of Operations & Production Management, 1986.

[15] Jean Louis Bertrand and Miia Parnaudeau. Understanding the economic effects of abnormal weather to mitigate the risk of business failures. *Journal of Business Research*, 98:391–402, may 2019.

[16] Leo Breiman. Random Forests. Technical report, 2001.

[17] Jenna Burrell. How the machine 'thinks': Understanding opacity in machine learning algorithms. *Big Data & Society*, 3(1):205395171562251, jan 2016.

[18] John C Chambers, Satinder K Mullick, and Donald D Smith. How to Choose the Right Forecasting Technique. Technical report.

[19] Dr Daqing Chen. Online retail dataset, 2015. Data retrieved from The UCI Machine Learning Repository, https://archive.ics.uci.edu/ml/datasets/Online%20Retail,.

[20] Claudimar Pereira Da Veiga, Cássia Rita Pereira Da Veiga, Anderson Catapan, Ubiratã Tortato, and Wesley Vieira Da Silva. Demand forecasting in food retail: A comparison between the Holt-Winters and ARIMA models. *WSEAS Transactions on Business and Economics*, 11(1):608–614, 2014.

[21] Sarah Goodrich Darin and Eric Stellwagen. Forecasting the M4 competition weekly data: Forecast Pro's winning approach. *International Journal of Forecasting*, 36(1):135–141, jan 2020.

[22] Neville Davies and C. Chatfield. *The Analysis of Time Series: An Introduction*, volume 74. 1990.

[23] Christos Faloutsos, Jan Gasthaus, Tim Januschowski, and Yuyang Wang. Forecasting Big Time Series: Old and New. 11(12):2102–2105, 2018.

[24] Pedro Joel Ferreira, Ana Almeida, and Jorge Bernardino. Data Warehousing in the Cloud: Amazon Redshift vs Microsoft Azure SQL.

[25] Robert Fildes, Shaohui Ma, and Stephan Kolassa. Retail forecasting: Research and practice. *International Journal of Forecasting*, dec 2019.

[26] Frepple. Demand classification: why forecastability matters, feb 2020. Online; accessed on 12 february 2020.

[27] Feixiang Gong, Ninghui Han, Dezhi Li, and Shiming Tian. Trend Analysis of Building Power Consumption Based on Prophet Algorithm. In *2020 Asia Energy and Electrical Engineering Symposium (AEEES)*, pages 1002–1006. IEEE, may 2020.

[28] José Fernando Gonçales. *Gestão de Aprovisionamentos*. Publindustria, 2000.

[29] Peter Hahn. *Artificial intelligence and machine learning*, volume 51. 2019.

[30] Rob J Hyndman and George Athanasopoulos. Forecasting : Principles and Practice.

[31] Rob J. Hyndman and Anne B. Koehler. Another look at measures of forecast accuracy. *International Journal of Forecasting*, 22(4):679–688, oct 2006.

[32] Rouba Ibrahim, Han Ye, Pierre L'Ecuyer, and Haipeng Shen. Modeling and forecasting call center arrivals: A literature survey and a case study. *International Journal of Forecasting*, 32(3):865–874, jul 2016.

[33] Hossein Javedani, Muhammad Hisyam Lee, and Suhartono. An Evaluation of Some Classical Methods for Forecasting Electricity Usage on Specific Problem. *Sciences-New York*, 2010(June):47–56, 2010.

[34] Chawalit Jeenanunta, K. Darshana Abeyrathna, M. H. M. R. Shyamali Dilhani, Su Wutyi Hnin, and Pyae Pyae Phyo. Time Series Outlier Detection for Short-Term Electricity Load Demand Forecasting. *INTERNATIONAL SCIENTIFIC JOURNAL OF ENGINEERING AND TECHNOLOGY (ISJET)*, 2(1):37–50, 2018.

[35] Ps Kalekar. Time series forecasting using Holt-Winters exponential smoothing. *Kanwal Rekhi School of Information Technology*, (04329008):1–13, 2004.

[36] Mirko Kuck, Sven F. Crone, and Michael Freitag. Meta-learning with neural networks and landmarking for forecasting model selection an empirical evaluation of different feature sets applied to industry data. *Proceedings of the International Joint Conference on Neural Networks*, 2016-October:1499–1506, 2016.

[37] Agnieszka Lasek, Nick Cercone, and Jim Saunders. Sales and customer demand forecasting: Literature survey and categorization of methods. In *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST*, volume 166, pages 479–491. Springer Verlag, oct 2016.

[38] Francisco Martínez-Álvarez, Alicia Troncoso, Gualberto Asencio-Cortés, and José Riquelme. A Survey on Data Mining Techniques Applied to Electricity-Related Time Series Forecasting. *Energies*, 8(11):13162–13193, nov 2015.

[39] Fernando Martinez-Plumed, Lidia Contreras-Ochando, Cesar Ferri, Jose Hernandez Orallo, Meelis Kull, Nicolas Lachiche, Maria Jose Ramirez Quintana, and Peter A. Flach. CRISP-DM Twenty Years Later: From Data Mining Processes to Data Science Trajectories. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–1, dec 2019.

[40] Victor Mayrink and Henrique S. Hippert. A hybrid method using Exponential Smoothing and Gradient Boosting for electrical short-term load forecasting. In *2016 IEEE Latin American Conference on Computational Intelligence, LA-CCI 2016 - Proceedings*. Institute of Electrical and Electronics Engineers Inc., mar 2017.

[41] Imad A. Moosa and Imad A. Moosa. Univariate Time Series Techniques. In *Exchange Rate Forecasting*, pages 62–97. Palgrave Macmillan UK, 2000.

[42] Georgia Papacharalampous, Hristos Tyralis, and Demetris Koutsoyiannis. Predictability of monthly temperature and precipitation using automatic time series forecasting methods. *Acta Geophysica*, 66(4):807–831, 2018.

[43] Juliusz Pres. Measuring non-catastrophic weather risks for businesses. *Geneva Papers on Risk and Insurance: Issues and Practice*, 34(3):425–439, jul 2009.

[44] Ye Ren, P. N. Suganthan, and N. Srikanth. Ensemble methods for wind and solar power forecasting - A state-of-the-art review, oct 2015.

[45] Bernd Scholz-Reiter, Jens Heger, Christian Meinecke, and Johann Bergmann. Integration of demand forecasts in ABC-XYZ analysis: Practical investigation at an industrial company. *International Journal of Productivity and Performance Management*, 61(4):445–451, apr 2012.

[46] Adriano O Solis, Letizia Nicoletti, Somnath Mukhopadhyay, Laura Agosteo, Antonio Delfino, and Mirko Sartiano. INTERMITTENT DEMAND FORECASTING AND STOCK CONTROL :. (c):367–374, 2012.

[47] Milan Stojanović and Dušan Regodić. The Significance of the Integrated Multicriteria ABC-XYZ Method for the Inventory Management Process. Technical Report 5.

[48] Boylan J. & Croston J. Syntetos, A. On the categorization of demand patterns. *Journal of the Operational Research Society*, page 495–503, 2005.

[49] Sean J Taylor and Benjamin Letham. Forecasting at Scale.

[50] Grigorios Tsoumakas. A survey of machine learning techniques for food sales prediction. *Artificial Intelligence Review*, 52(1):441–447, jun 2019.

[51] Tine Van Calster, Filip Van den Bossche, Bart Baesens, and Wilfried Lemahieu. Profit-oriented sales forecasting: a comparison of forecasting techniques from a business perspective. *arXiv preprint arXiv ...*, feb 2020.

[52] J. Thomas Yokuma and J. Scott Armstrong. Beyond accuracy: Comparison of criteria used to select forecasting methods. *International Journal of Forecasting*, 11(4):591–597, dec 1995.

[53] Yanru Zhang and Ali Haghani. A gradient boosting method to improve travel time prediction. *Transportation Research Part C: Emerging Technologies*, 58:308–324, sep 2015.

[54] Naizhuo Zhao, Ying Liu, Jennifer K. Vanos, and Guofeng Cao. Day-of-week and seasonal patterns of PM2.5 concentrations over the United States: Time-series analyses using the Prophet procedure. *Atmospheric Environment*, 192:116–127, nov 2018.

[55] Emir Zunic, Kemal Korjenic, Kerim Hodzic, and Dzenana Donko. Application of Facebook's Prophet Algorithm for Successful Sales Forecasting Based on Real-world Data. *International Journal of Computer Science and Information Technology*, 12(2):23–36, may 2020.