UNIVERSITAT DE BARCELONA

FUNDAMENTAL PRINCIPLES OF DATA SCIENCE MASTER'S THESIS

Demand Forecasting in Pharmaceutical Supply Chains: Novo Nordisk Case Study

---

Author:
Mattia TONELLI

Supervisor
Martin CHRISTIANSEN
Dr. Oriol PUJOL VILA

A thesis submitted in partial fulfillment of the requirements for the degree of MSc in Fundamental Principles of Data Science

in the

Facultat de Matemàtiques i Informàtica

January 18, 2021

UNIVERSITAT DE BARCELONA

Abstract

Facultat de Matemàtiques i Informàtica

MSc

Demand Forecasting in Pharmaceutical Supply Chains: Novo Nordisk Case Study

by Mattia TONELLI

Forecasting is a common use case in the field of Predictive Analytics and one of the key building blocks of any Supply Chain. This relevance is even magnified in the pharmaceutical industry, where a stock-out does not merely carry a monetary impact but might also tragically affect people's health.
In light of the aforementioned, this thesis has a twofold aim. Firstly, improving Sourcing Operations' forecasting process in terms of both accuracy, standing at 59%, and efficiency, currently a 5-day process. Secondly, helping to shed some light on the univariate-multivariate debate in the forecasting realm.

Attaining these goals required uncovering the best methods in the forecasting realm by scouring the existent literature; both univariate and multivariate applications were eventually pursued. With respect to the former, classic techniques such as simple average, Autoregressive Integrated Moving Average and Exponential Smoothing were chosen. These were also combined in an ensemble in order to leverage each model's strengths and keep each other in balance.
Amongst the several multivariate techniques available, the choice fell upon Gaussian Process Regression and its capability to model complex functions by means of kernels. Identifying these complex structures also required combining such kernels, and given the sheer amount of time series, a mechanical greedy search strategy operating as a forward selection method was devised.

Results showed how a multivariate approach (68%) outperformed the univariate models (63%), albeit the former (18 hours) was much slower than the latter (30 minutes). Finally, combining the "best of both worlds" enhanced accuracy up to 71%.

With respect to the first goal, these outcomes increase accuracy by 12 percentage points and slash forecasting time to less than a day. In terms of the second goal, these results seem to argue in favor of multivariate methods, demonstrating that these perform better since they can leverage external information; yet, the best-of-both-worlds approach also shows the lack of a clear-cut answer on the matter: each class of models might outperform the other under certain conditions.

Acknowledgements

With these lines I take the opportunity to assign the merit to those who deserve it, and made my goal of becoming a Data Scientist possible. First and foremost, I thank my family, who supported me throughout my whole study career and most of all continuously stressed the fact that, culture, is the most important prize a person can earn. Thanks for everything you have done for me.

Then comes Professor Oriol; his insights were incredibly useful, not only in steering this manuscript in the right direction, but have also made my working days much easier. Thanks for accepting to supervise my endeavor and for sharing your knowledge with me.

Third, thanks to the Supply Chain Analytics team, who made me feel part of the team from the very first moment. Mentioning someone specifically would be unfair towards the others, however, Martin deserves to be singled out. As my supervisor, he has been incredibly patient, even when I asked for his help despite the solution was simply in front of my eyes but I was incapable of seeing it. Thanks for all you taught me.
In a similar vein, thanks to my new manager Lars, who has granted me a very soft start in the new job in order to finish this thesis.

Another shout out goes to all of my Master's classmates. Strictly in alphabetical order, I am particularly grateful to Àlex Arcas Cuerda, Pablo Granatiero and Alejandro Matheus Hernandez who were there when I needed help during the studies. Thanks for having the patience to spend your time explaining me some concepts, and all the laughs we had together.

Fifth, thanks to my parents-in-law, who during the first pandemic wave hosted me at their house and treated me better than a son. Thanks for the great times we had together, and I look forward to many more to come.

Then, thanks to Davide, Giulio and Massimiliano. The first for proofreading this thesis - by the way, mistakes still remain mine - and spending long evenings playing board games during the long Danish winter. The second for sharing with me the same type of humour and for understanding each other without talking; I truly miss the amazing times we had as teammates. The third for his empathy and for the large chats over how to improve ourselves as human beings. Thanks for being good friends.

Seventh, a hug to my friends back home in Italy. Despite I have been abroad for many years now - and the fact that we hardly keep in touch on a daily basis - we still maintain the same relationship as when we first met in kindergarten. Thanks for sticking with me regardless of the distance and being always there when I need you.

Finally, the greatest of my appreciation goes to my girlfriend Cristina, who has stood me for almost 5 years now - despite my many flaws. She supported my decision to quit my job in Denmark in order to pursue these studies, although this meant having a long-distance relationship for almost a year. Thanks for bearing this, I promise your sacrifice has been done for an even better future together.

Chapter 1

Introduction

1.1 Introduction

It has only been in the past few decades that advances in technology - increased data storage, more powerful processors, faster internet connections - have reached a stage capable of supporting Artificial Intelligence applications (Walch, 2019). The removal of previous constraints has in fact allowed computers to process the sheer amount of data created nowadays, thereby facilitating – if not enabling altogether – Machine Learning (ML) exploitation. As a result, real-life applications that appeared tremendously complex or computationally expensive to solve are now within companies' reach (Usuga Cadavid, Lamouri, & Grabot, 2018).

This growing combination of resources and tools has also carried profound implications in the field of Supply Chain (SC) management (Waller & Fawcett, 2013). The term SC Analytics is an umbrella term referring to the application of advanced ML in SC and encompasses three main branches (Souza, 2014; Tiwari, Wee & Daryanto, 2018):

- *Descriptive Analytics* (DA) deal with the question of *"what has happened, what is happening, and why"* by generating reports that provide historical insights;
- *Predictive Analytics* (PredA) deal with the question of *"what is likely to happen"* by exploring data pattern using statistics, simulation, and programming;
- *Prescriptive Analytics* (PresA) deal with the question of *"what should be happening and how to influence it"* by driving decisions based on DA and PredA through mathematical optimization.

Forecasting - the focus of this thesis - is a common use case in the field of PredA (Usuga Cadavid et al., 2018) and one of the key building blocks of any SC (Chopra & Meindl, 2013, p.178; Heizer, Render & Munson, 2017, p. 147).
By providing several benefits such as bullwhip effect [1] mitigation, efficient production capacity planning, inventory control and reduction of stock shortages or oversupply costs (Chopra & Meindl, 2013, p.178-179; Heizer et al., 2017, pp. 147-148), it forms the basis of all managerial decisions in SC. This relevance is even magnified in the pharmaceutical industry, where a stock-out does not merely carry a monetary impact but might also tragically affect people's health (Merkuryeva, Valberga & Smirnov, 2019).

[1] The bullwhip effect is a concept for explaining inventory fluctuations or inefficient asset allocation as a result of demand changes as you move further up the supply chain. As such, upstream manufacturers often experience a decrease in forecast accuracy as the buffer increases between the customer and the manufacturer

A variety of forecasting methods have been developed, and amongst the many available, traditional models (TM) and Explanatory Models (EM) emerge as the most relevant (Hyndman & Athanasopoulos, 2018). The former, based on statistical techniques, are applied under the hypothesis that past demand can statistically estimate the future demand. This means that they look at past data patterns and attempt to predict the future based upon these underlying patterns.
The latter, on the other hand, assume that the variable being forecasted does not uniquely depend on its own past values but is also related to other variables in the environment. Methods belonging to this family normally include other endogenous and exogenous variables, allowing for a more comprehensive representation of reality. In this vein, ML itself could be considered as explanatory modeling[2] (Usuga Cadavid et al., 2018).

Both methods are largely implemented considering the SC as a whole (Chopra & Meindl, 2013, p.180) as well as in the specific domain of pharma (Merkuryeva et al., 2019). Each carries its own advantages and limitations. TM are easy to apply and typically serve as an adequate starting point. They offer huge advantages in terms of simplicity as they can perform demand forecasting in a matter of seconds for several SKUs (Stock Keeping Units).
TM, however, struggle when demand also depends on exogenous factors that are not effectively represented by its own lagged values. This situation is where ML helps to fill the gap, albeit at a considerable computational price.

The literature has not yet returned a clear-cut verdict on the most suitable approach (Huber & Stuckenschmidt, 2020). Regardless of the technique, however, it is fundamental to bear in mind that forecasting is "a prophecy, estimate, or prediction of a future happening or condition" (Mirriam Webster, n.d.). Intrinsic in its definition, thus, resides the concept that neither method can perfectly predict the future.

In light of the aforementioned, the goal of this piece of work is twofold[3]:

1. Firstly, to improve the forecasting process of Sourcing Operations (SoOP), a specific department belonging to Novo Nordisk, the pharmaceutical company object of the study;

2. Secondly, to a lesser extent, help shedding some light on the method debate.

The rest of this thesis is organized as follows. The remainder of Chapter 1 introduces the reader to the company and the specific department object of the study, also presenting the forecasting setup currently in place. Chapter 2 aims at providing an overview of the different techniques that scholars have implemented to predict demand, thereby offering a review of methods. Chapter 3, based on the literature review findings, describes the chosen techniques from both univariate and multivariate methodologies. Chapter 4 presents the dataset and the processes required in order to make it suitable for the different analyses carried out. Chapter 5 analyses the findings of both methodologies and compares their results. Finally, Chapter 6 summarizes the findings of this thesis.

[2]Please bear in mind that unless some causal-aware techniques are implemented, ML models of this sort can only predict *what* will happen and not *why* it happens (thanks Professor Pujol for raising this point).

[3]The code used to achieve it can be found in the *private* repository at the following address: `https://bitbucket.org/mattia_tonelli/forecasting/src/master`

1.2 Novo Nordisk, the Sourcing Operations Department and its Forecasting Process

Headquartered in Denmark, Novo Nordisk is a global healthcare company with a strong leadership in diabetes care and expertise in obesity, growth disorders and haemophilia, marketing its products in more than 170 countries (Who we are, 2020). The Sourcing Operations (SoOp) department is instrumental in enabling Novo Nordisk to help people defeat such serious chronic conditions: by managing "the global inbound supply chain to deliver packaging and raw materials to production [...]" (Sourcing Operations, 2020), it guarantees a timely and cost-efficient drug manufacturing.

1.2.1 The Forecasting Process

The current forecasting process is run twice a year, namely in October and February, and its output covers a period ranging from M+1 until the first half of Y+2[4], split into three prediction buckets. In both instances, an initial version is produced, serving as the baseline upon which inputs from subject matter experts, such as supply planners and consuming sites, will be incorporated. The resulting refined forecast then becomes the final version.

Depending on some criteria, different forecasting methods are implemented. These are presented below together with the corresponding percentage out of the total forecasted raw materials:

- *Manual*: Each production site relies on its experience and own forecasting tools, that could be as advanced as an optimization software or as simple as manually checking each single time series in an Excel file (19%);
- *Production Planning System*: The Advanced Planner and Optimizer (APO) software plans production quantities by taking into account many complex variables, such as the delivery schedule of raw materials and productions cycles (5%);
- *Predictive Models*: The time series are loaded into Alteryx, a self-service analytics tool, where a host of common, preset ML models are automatically and "blindly" run on each series (25%);
- *Historical Consumption*: The demand for a future prediction bucket is simply assumed to be equal to the past year's consumption in the same bucket (51%).

The current baseline accuracy stands at 59% across time series, calculated as:

$$\frac{1}{n}\sum_{t=1}^{n} 1 - MAPE$$

where MAPE stands for Mean Absolute Percentage error[5].

MAPE will be introduced more in detail in a dedicated section, but for now it is

[4] M stands for current *month* whereas Y for current *year*. For example, if we were in July 2018, M+1 would refer to August 2018, whereas Y+2 to year 2020.

[5] In this thesis, the terms *MAPE* and *accuracy* will be used "interchangeably", that is, "MAPE of 30%" might as well be mentioned as "accuracy of 70%".

sufficient to bear in mind that its selection reflects the nature of the time series. In fact, their units of measure spans from grams and kilos, to milliliters and liters. Thus, the unit-of-measurement-free nature of MAPE enables meaningful comparisons and aggregations.

item	M+1	M+2	M+3	M+4	M+5	M+6	M+7	M+8	M+9	M+10	M+11	M+12	yearly_sum	actuals	accuracy
A	3	17	1	25	6	6	15	28	15	6	3	28	153	157	96.84
B	16	26	26	26	14	14	7	11	13	20	17	27	217	284	30.87
overall_accuracy															63.85

FIGURE 1.1: For every time series, the monthly predicted values (red) are aggregated into a yearly sum (green) and used with the yearly actuals (orange) to calculate the MAPE and subsequently the accuracy. Eventually, the baseline accuracy is computed (violet). Source: own creation

Figure 1.1 helps understanding more in detail how the accuracy is assessed. Per each item taken singularly, monthly predictions are aggregated into a yearly sum. Together with the year's actuals, these quantities are employed to compute the MAPE - and subsequently its accuracy. The rationale behind the evaluation being carried out at a year level resides in the fact that SoOp purchases quantities with an annual horizon.

Summing up, the depicted situation clearly highlights the importance of achieving the first goal of this thesis, quantified by the SoOp team as increasing forecast accuracy to at least 70% and reducing significantly the manual intervention[6].
The next chapter will touch upon the different methods and techniques through which this can be achieved, by presenting an overview of their usage in the literature.

[6]Despite there are no official figures availables, SoOp team estimates that currently the forecasting process takes about 5 days

Chapter 2

Literature Review

The previous chapter has already acquainted the reader with the relevance of the forecasting process in SC as well as with the different methods at disposal and the difficulties that come in tow. This chapter will walk the reader through researches investigating demand forecasting in an attempt to offer an exhaustive overview about forecasting methods from a scientific standpoint. Related works will include demand forecasting within SC in general or, in other words, regardless of the industry.

The topic already boasts several attempts at classifying papers along various perspectives. Wang et al. (2016) scour the existing literature and categorize researches based on the application of the different SCA branches on demand forecasting. Usuga Cadavid et al. (2018) retrieve the most relevant papers which make use of ML applications in order to spot new ML trends and techniques applied in the domain; from these, pieces of work comparing ML performance to TMs are then extracted. Finally, Wenzel, Smitt & Sardesai (2019) present a snapshot of the current state by mapping applied ML methods to the different areas composing SC, showing that different ML techniques can be applied to solve a common goal.
For this thesis, the selected approach reflects largely Usuga Cadavid et al.'s (2018) as deemed more in line with its goals and identifies an additional grouping, leading to a total of three broad categories.

The first group encompasses studies pitting TM against ML results. Somehow pioneering this stream of research, in 2001 Alon, Qi & Sadowski decide to compare Neural Networks (NN) and TMs – including exponential smoothing (ETS), Autoregressive Integrated Moving Average (ARIMA), and linear regression (LR). This comparison is carried out on aggregate monthly retail sales time series containing both trend and seasonal patterns, thereby providing a valuable testing ground for forecasting evaluations. Further, the authors validate the robustness of the alternative forecasting methods by splitting the time series into two time periods, each featuring different economic conditions. The data suggest that during turbulent economic times, NNs generally provide superior forecasts over the traditional methods, but the ARIMA and ETS remain formidable competitors, especially when conditions are relatively stable.
NNs also represent the ML model of choice for Gutierrez, Solis & Mukhopadhyay (2008). In the context of intermittent demand forecasting, their performance is related to three TMs – namely, ETS, Croston's, and the Syntetos–Boylan approximation. Based on the results, the academics conclude that NN models prove to be superior to the TMs on all the error measures evaluated.
Carbonneau, Laframboise & Vahidov (2008) seek to investigate the applicability of non-linear ML techniques – such as NNs, Recurrent NNs (RNN) and Support Vector Machines (SVM) – to forecast demand in the peculiar context of extended supply

chains. Their performance is pitted against traditional approaches including naıve forecasting, moving average, and LR models. Findings reveal a slightly greater accuracy obtained by ML techniques. Yet, the authors warn that these marginal gains should be weighed against the conceptual and computational simplicity of the traditional approaches.
In an initial study, Kandananond (2012a) first indicates that SVM achieves greater accuracies than NN models in forecasting consumer product demand. The study is then (2012b) extended to also encompass ARIMA. Findings confirm SVM as the most accurate model, also outperforming the selected TMs.
Huber & Stuckenschmidt (2020) address the challenges connected to demand forecasting on special days (public holidays, the days before and after, etc..), where daily demand diverges sharply from regular days as customers modify their daily routines. The scholars opt for NN, RNN and Gradient-Boosting techniques. These ML approaches provide more accurate predictions than LR and ETS since the sales historical series can be enriched with external information reflecting special calendar events.

A second stream of research focuses on applying uniquely ML models to solve forecasting problems. Ahmed, Atiya, Gayar, & El-Shishiny (2010) compare a variety of ML methods on monthly time series, including NNs, BNNs, kernel regression, K-nearest neighbour regression, CART regression trees, SVM, and Gaussian Processes (GP). Their study shows significant differences between these models, with NNs and GPs ranking as best models.
Sarhani & El Afia (2014) seek to overcome the main drawbacks linked to SVM parameter tuning in highly non-linear spaces, namely, computational expensiveness and unguaranteed convergence to the globally optimal solution. In their quest, the researchers employ monthly retail sales data finding how leveraging optimizing algorithms can efficiently find optimal or near-optimal solutions in large search spaces, thereby improving SVM's performance.
Ampazis (2015) presents an approach integrating data from various sources of information to advanced ML algorithms for lowering uncertainty in forecasting supply chain demand. In trying to predict DVD movie rental demand during a critical period for sales, such as Christmas holidays, the academic utilizes NNs and SVMs. The analysis supports how including relevant features improves the performance of both ML models.
Yang & Sutrisno (2018) attempt at forecasting short-term sales to fine-tune the daily replenishment strategy of a Chinese bakery chain. In this study, LR models are judged against NNs, with the latter attaining more accurate predictions on actual sales than the former due to its flexibility, capable of modelling the sales daily fluctuations.

A third and final group of studies concentrates its focus on combined approaches that involve different techniques, thereby leveraging their respective strengths. Aburto & Weber (2007) aim at improving the forecasting capabilities of a Chilean supermarket chain by means of a hybrid model. By training NNs on the residuals of the ARIMA model, results exhibit how the former outperforms the latter. Finally, Adhikari et al. (2017) propose a revised ensemble technique. The authors generate two forecasts: one stemming from a selection of TMs, and another resulting from several classical ML regression-based models. These are subsequently pooled into an ensemble and, based on historical performances, different weights are allocated in

order to penalize algorithms deviating from the actual sales. The forecast combination outstrips each of its component taken singularly since it evens out over- and under-forecasted values, thus bringing the aggregated predictions near to the actuals.

Summing up, this chapter hinted at how ML models outperform traditional models, findings also corroborated by the literature review found in Usuga Cadavid et al. (2018).
Yet, there is no shared consensus around this. Much debate surrounds the relative performance of TM and ML methods, and it is difficult to draw general conclusions about their efficacy. TMs have in fact been successfully applied to many forecasting problems, and there is no definite evidence about their inferiority compared to ML methods (e.g. Ahmed et al., 2010; Makridakis, Spiliotis & Assimakopoulos, 2018). Perhaps, truth lies in the middle, where each class of models might outperform others under certain conditions (Crone, Hibon & Nikolopoulos, 2011).

This literature outline will serve as the basis for the next chapter, which will introduce the different methods implemented to solve this business case.

Chapter 3

Methods

In this chapter the findings from the literature review will translate into concrete decisions. Equipped with the consideration that each class of models might outperform others under certain conditions, the following sections will present the chosen techniques from both sides and offer a brief description of their "engine room".
But first, MAPE - the chosen metric briefly mentioned in Chapter 2 - will be described in more detail, followed by some common potential time series pitfalls, and how to avoid them.

3.1 Metrics

The goodness of the generated forecasts will be evaluated via MAPE:

$$\frac{1}{n}\sum_{t=1}^{n}\left|\frac{Actuals_t - Forecasted_t}{Actuals_t}\right|$$

The absolute value in this calculation is summed for every forecasted point in time and divided by the number of fitted points n. Multiplying by 100 makes it a percentage error.
There are several reasons that brought to the decision of implementing MAPE as metric. Firstly, it is already used by the SoOp department, thereby enabling a "fair" performance comparison. Secondly, it comes under percentage errors which are scale independent, thus allowing to compare series with different units of measure. Lastly, because of its wide popularity in the forecasting literature (Alon et al., 2001; Hyndman & Koehler, 2006).

MAPE also carries along some limitations though (Hyndman & Koehler, 2006). As all measures based on percentage errors, MAPE is undefined if the actuals are zero or can take on huge values - thus larger than 100% - if the actuals are very close to zero. Finally, they also have the disadvantage of placing a heavier penalty on negative errors than on positive errors.
Despite these shortcomings, Hyndman & Koehler (2006) still consider MAPE as the preferred metric, in particular in situations like this one, where all data are positive.

Before delving into both univariate and multivariate methods, the next section will introduce two critical concepts in ML time series, namely cross-validation and data leakage.

3.2 Time Series-related Problems

3.2.1 Cross Validation

Applying ML to time series forecasting, like any other ML application, also requires a model evaluation aiming at estimating the model performance on unseen future data. And as any other ML application, it is crucial to prevent - or at least mitigate - the randomness involved in the train-test split from generating 'lucky' predictions, that is, prediction results depending on pure coincidence.
However, the "traditional" cross-validation approach cannot be directly employed with time series data. The rationale behind this limitation stems from the intrinsic nature of a time series, setting it apart from a "classic" ML regression problem: its sequential nature. The time dimension of observations, in fact, impedes a random split since this would assume the absence of a relationship between observations.

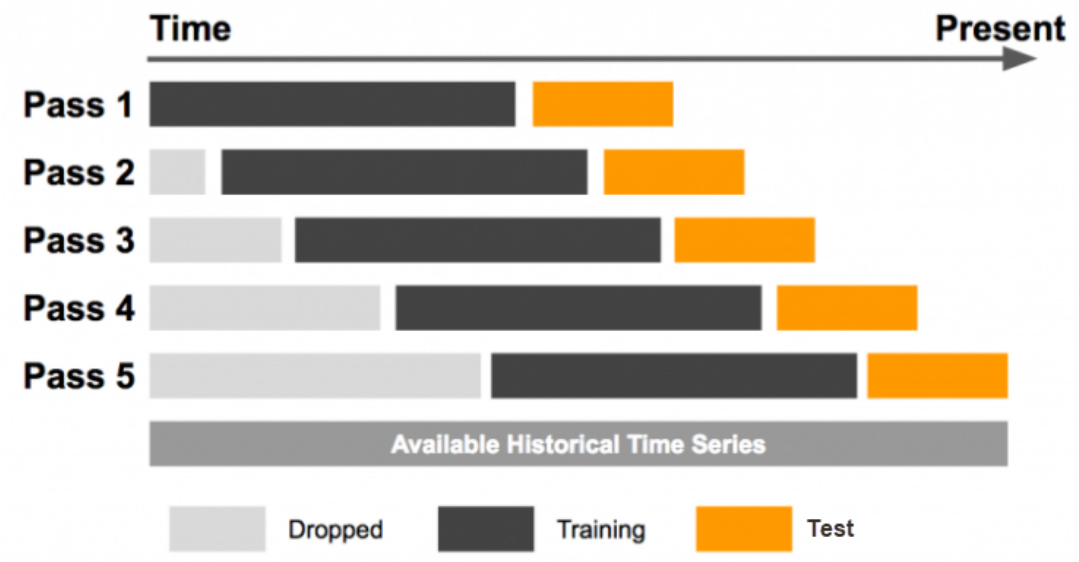

FIGURE 3.1: A fixed-size training window (dark grey) slides over the entire time series horizon and is repeatedly tested against a test set (orange), while older data points dropped (light grey). Source: here

In light of what just described, the evaluation of time series models requires another approach, the so-called *backtesting*, which applies a cross validation logic while accounting for the temporal order in which values were observed.
For this endeavour, the type of backtesting employed is the *sliding window*, depicted in Figure 3.1. This process is iterative and applies a rolling forecast evaluation by splitting a time series over multiple time points. At each splitting point, the time series is divided between training and test sets, both of *constant* size: as usual, the model trains on the former and the resulting forecasted values are compared with the actuals obtained from the latter.

Each of the splits, thus, yields a specific MAPE per each forecasting technique. Once all the time splits have been performed, an overall MAPE is calculated by averaging each technique results over time (Figure 3.2). This averaged MAPE can be interpreted as an estimation of the generalisation error associated with each model in production, thereby serving as a model selection instrument.

model	2016	2017	2018	2019	overall_MAPE
ARIMA	0.69	0.89	0.69	0.21	0.62
ETS	0.25	0.88	0.62	0.69	0.61
MEAN	0.71	0.96	0.59	0.57	0.71
ENSEMBLE	0.63	0.14	0.42	0.75	0.48

FIGURE 3.2: The orange rectangle enables a MAPE comparison by model over the same year. However, the best model is chosen by calculating an overall MAPE across years (in red). This is simply the average of the MAPEs within the green rectangle. Source: own creation.

3.2.2 Data Leakage

Time series problems are prone to introduce data leakage, if carefulness is not employed when designing the ML model. In the case of introducing data leakage in time series, we allow our model to use both past *AND* future data, for predicting today's value.
This phenomenon can happen in a time series context for the following reasons. The first reason has been just explained in the above subsection and consists of employing a random split neglecting the temporal nature of the data to validate the model, instead of using the backtesting approach.
A second one involves adding explanatory features that were not "legitimately" available during the actual target value generation. Let's hypothesize that *item_A* is used as raw material in the manufacturing of *med_A* and *med_D*; this entails that leveraging the medicines sales numbers would represent a great help in trying to figure out the amount of *item_A* required for a month. However, since the total sales would only be available at the end of the forecasted month, including *sales_med_A* and *sales_med_D* in the training phase would leak information.
To combat this potential issue, the features entering the multivariate model have been carefully selected. These will be introduced in the chapter 4.

Now, this chapter moves onto explaining the chosen techniques from univariate and multivariate methods.

3.3 Univariate

The term *univariate* implies that forecasting is based on a sample of observations of the dependent variable without taking into account the effect of other variables. The underlying concept bolstering this methodology is that the effect of exogenous variables is embodied in, and reflected by, the actual behaviour of the dependent variable. The data and the computational requirements of these models are normally smaller than in the case of multivariate models.

The three univariate time series techniques chosen are simple average (or MEAN), ETS and ARIMA, belonging respectively to the averaging methods, the smoothing methods and time series decomposition (Moosa, 2000). Further, a "selective" ensemble is built based on the combination of the aforementioned.
Their implementation is enabled in `R` through the packages *fable* and *fabletools* which, more importantly, support automatically selected ETS and ARIMA models. The next

subsections will present them in more detail, following the order they were introduced here.

3.3.1 MEAN

The simple average is an extremely simple, yet a surprisingly effective technique (Hyndman & Athanasopoulos, 2018). It is obtained by extracting the mean computed from all the actual observations and then using it as an input for the forecast of the next point in time.
The forecast (F) for time $t+1$ is the average of the actuals (A), thereby assuming equal relevance for each observation over the period extending between 1 and t.

$$F_{t+1} = \frac{1}{t} \sum_{i=1}^{t} A_i$$

When the forecast is required for points in time beyond $t+1$, as actuals are not available for these, previous forecast values become an input to calculate the simple average. The forecast for $t+n$, thus, is calculated from a sample of $t+n-1$ observations consisting of the t actuals and the $n-1$ forecasts for the period between $t+1$ and $t+n-1$.

$$F_{t+n} = \frac{1}{t-n+1} \left(\sum_{i=1}^{t} A_i + \sum_{j=1}^{n-1} F_{t+j} \right)$$

In this case, the forecast quantity will be the same regardless of the forecasting horizon since the forecast values are used. In other words, adding to a sum its mean does not change the mean value itself.
Despite being incredibly simple to implement, a drawback of this method is its appropriateness only if the observed time series has no trend nor seasonality (Moosa, 2000). The next two techniques enable overcoming this shortcoming.

3.3.2 ETS

ETS is based on averaging (i.e. smoothing) past values of the dependent variable in an exponentially decreasing manner. The principle behind smoothing is that demand observations that are in temporal proximity are likely to be similar in value. Forecasts produced using ETS are thus weighted averages of past observations, with the weight relevance decaying exponentially as the observations come from a further distant past. In other words, the more recent the observation the higher the associated weight.
Its simplest application, exactly like the MEAN, does not assume any systematic structures in the data. However, its extensions render ETS a forecasting technique suited to handle a time series with trend and/or seasonal component. By considering variations in the combinations of the trend and seasonal components, nine methods are identified[1]. These can be broadly be grouped into three main ETS families (Hyndman & Athanasopoulos, 2018):

[1]Their detailed explanation goes beyond the aim of this thesis. The interested reader can found a comprehensive overview in Hyndman et al. (2008).

- *Single ETS (or Simple ETS)* is suitable for forecasting data with no clear trend or seasonal pattern. It represents a middle way between those forecasting methods assigning all the weight to the last observation (naïve) and those allocating equal importance to all observations (mean);
- *Double ETS (or Holt's Linear Trend)* extends single ETS to allow the forecasting of data with a trend. It supports trends that change in different ways: additive or multiplicative, depending on whether the trend is linear or exponential, respectively. Its forecasts display a constant trend (increasing or decreasing) indefinitely into the future, which is a rather unrealistic assumption. For longer forecast horizons, then, it can be useful to dampen the trend over time, that is, reduce the trend to a flat line at some point in the future;
- *Triple ETS (or Holt-Winters ETS)* builds on top of the double ETS to capture also seasonality. The seasonal component may be also be modeled as either an additive process, when seasonal variations are roughly constant through the series, or multiplicative when these variations change proportionally to the level of the series.

This framework generates reliable forecasts quickly and for a wide range of time series, which is a great advantage and of major importance to applications in industry (Hyndman & Athanasopoulos, 2018).
The next subsection will present another widespread and powerful methodology in the forecasting industry, ARIMA.

3.3.3 ARIMA

The ARIMA methodology is appropriate if the observations of a time series are statistically dependent on each other. Models belonging to this family are based on the idea of transforming the time series to achieve stationarity by means of a differencing process, thereby rendering its statistical properties (mean and variance) constant over time.
Its acronym is self explanatory, embodying the key elements of the model itself:

- *Autoregression (AR)* since it uses a linear combination of past values of the dependent variable, indicating that it is a regression of the variable against itself.
- *Integrated (I)* since it differences raw observations (e.g. subtracting an observation from another at the previous time step) in order to make the time series stationary. A time series that requires differencing is said to be an "integrated" version of a stationary series.
- *Moving Average (MA)* since it uses past forecast errors in a regression-like model.

Each of these components are respectively expressed in the standard notation *ARIMA(p,d,q)*, where integer values substitute the parameters to quickly indicate the specific ARIMA model: in other words, a linear regression model constructed including the specified number and type of terms. If necessary, before applying the regression, the data is prepared by one, or more, degree of differencing.
In short, the ARIMA equation for a time series is a linear equation in which the input consists of lags of the dependent variable along with lags of the forecast error.

3.3.4 Ensemble

Ensemble methods combine predictions from multiple forecasting techniques to improve the accuracy of a simple prediction and avoid possible overfitting by reducing the impact of any specific model. Diversity among the individual components of ensembles is in fact known to be the key element that makes ensemble a successful model.
There are many ways in which allocating the importance to each model inside the ensemble. Yet, simple average of these forecasts represents a standard approach. It is easy to implement, relatively fast to compute, and often provides an excellent forecast (Hyndman, 2018).

$$\frac{1}{n}\sum_{model=1}^{n} model$$

In this work, a "selective" ensemble has been coupled to a classic, "all-inclusive" one. The logic behind the selective is that a model, in order to become part of the ensemble, must achieve a determined performance. This threshold has been set to an accuracy of 60%, decision driven by the desire to *at least* beat the current baseline.
However, if only one model makes it past the threshold, then a selective ensemble simply becomes the model itself. In this case, a classic ensemble including the three basic models is constructed instead. Figure 3.3 helps better understanding this approach.

	year	ARIMA	ETS	MEAN
2015		0.63	0.79	0.85
2016		0.71	0.59	0.51
2017		0.06	0.34	0.79
2018		0.54	0.41	0.37
2019		0.1	0.77	0.83
selective_count		2	2	3

FIGURE 3.3: The ensemble selection process. For every year the models that have made it past the accuracy cut are marked in red. Their count over time is marked in green at the bottom and serves to identify the components of the selective ensemble for the whole time horizon. Source: own creation.

For this specific time series, the accuracy results are reported for each of the three base models. If the score was above the threshold, it is marked in red. For example, in 2015 and 2019, where respectively three and two models scored above the selective threshold, a selective ensemble will be constructed using such models. Further, their presence will be counted in order to keep track of the number of times a model entered a selective ensemble.
On the other hand, in 2017 and 2018, where respectively one and no models qualified to enter the selective ensemble, the calculated ensemble will include all of the models. However, the selective ensemble counter will not be updated to reflect their inclusion.

Regardless of the ensemble type, the usual backtesting approach is employed to obtain the overall MAPE. If the ensemble happens to provide the best performance,

then the issue of selecting a general ensemble for the whole time horizon arises. In fact, most likely, every year the ensemble will be composed by different models.
The solution lies in the selective counter. Bringing the attention back to Figure 3.3, it can be noted how the MEAN enters the selective ensemble three times, followed by both ARIMA and ETS tied at two. If there were two - or more - models tied at the first place, the ensemble will include these. Since in this instance there is only one model, then the ensemble will include such a model plus whichever model(s) ranked second.
Obviously, if no model made it past the threshold at any point in time (i.e. selective counter equals to zero for every model), then the ensemble would simply be a classic one.

Now that has been shown how the several univariate techniques somehow compensate for each other's limitations (MEAN, ETS, ARIMA) and even can enhance their strengths if aggregated (ensemble), the next section will introduce the multivariate part of the thesis.

3.4 Multivariate

A multivariate time series encompasses more than one time-dependent variable, and each of these variables does not exclusively depend on its past values but also features a certain degree of dependency on the others. Such a dependency is used for forecasting future values. Thus, techniques belonging to this family typically introduce other endogenous and exogenous variables in order to model the actual behaviour of the target function.
In this vein, multivariate time series could be solved as a supervised ML problem. But before ML can be employed, time series forecasting problems must be re-framed as supervised learning problems. The first subsection will exhibit how the original time series have been modified in order to enable the application of the chosen ML method, namely GP. Then, the selected forecasting strategy is presented, followed by a large section about GP and its peculiarities. Finally, the search strategy selected in order to maximize the performance of the GP models is explained.
All multivariate tasks will be carried out in `Python`, in order to leverage the well-equipped package *scikit-learn*.

3.4.1 Problem Re-framing

The two figures below clearly exhibit the result of the transformation from a long (3.4) to a wide (3.5) format. Since we set the training window size to 12, the first row comprising the first twelve observations - marked in red - become the input to predict the target observation, which is the 13^{th}. Then, in the second row, the window slides forward by one month so that observations from the 2^{nd} to the 13^{th} are used to predict the new target, namely the 14th observation, and so forth.
Important to note is that the temporal order between the observations is preserved. However, the re-framing entails the "loss" of one year of observations, namely 2015, which exclusively participates as input to predict the months of 2016.

3.4.2 The Multi-Step Forecasting Strategy

A multiple-step forecasting strategy fits this thesis' goal for several reasons. First, multiple time steps ahead must be predicted as per business requirements by SoOp

item_id	demand	consumption_month
1000012-2044	80825.20000	2015-01-01
1000012-2044	91782.30000	2015-02-01
1000012-2044	90931.51900	2015-03-01
1000012-2044	98501.70000	2015-04-01
1000012-2044	80030.80000	2015-05-01
1000012-2044	106222.88100	2015-06-01
1000012-2044	46381.00000	2015-07-01
1000012-2044	45911.50000	2015-08-01
1000012-2044	159473.10000	2015-09-01
1000012-2044	85504.00000	2015-10-01
1000012-2044	128610.50000	2015-11-01
1000012-2044	67228.50000	2015-12-01
1000012-2044	0.00000	2016-01-01
1000012-2044	73994.20000	2016-02-01
1000012-2044	180650.50000	2016-03-01
1000012-2044	86413.00000	2016-04-01
1000012-2044	87378.00000	2016-05-01
1000012-2044	92594.40000	2016-06-01
1000012-2044	60463.80000	2016-07-01
1000012-2044	55468.75000	2016-08-01
1000012-2044	49893.00000	2016-09-01
1000012-2044	77093.50000	2016-10-01
1000012-2044	0.00000	2016-11-01
1000012-2044	117888.20000	2016-12-01

FIGURE 3.4: The long format required for univariate forecasting. The variable of interest is reported chronologically from oldest to newest date. In green are the observations of the whole 2015, whereas in red the observation of January 2016. Source: own creation.

team. Compared to the more "classic" one-step forecast, the Multiple Input Multiple Output (MIMO) technique involves developing a single model capable of predicting the entire forecast sequence in a one-shot fashion, where the predicted value is no more a scalar quantity but a vector of future values of the time series.
Second, when a long term horizon is at stake like in this specific instance, the modeling of a single-output mapping neglects the existence of stochastic dependencies between future values (e.g. between y_{t+1} and y_{t+2}), consequently biasing the prediction accuracy. On the contrary, MIMO avoids the simplistic assumption of conditional independence between future values made by the Direct strategy, where a specific model is developed to predict each of the different time points within the forecast horizon.
Third, it does not suffer from the accumulation of errors plaguing the Recursive strategy, where the forecasted values become the input for the subsequent predictions. In fact, the model has a tendency to accumulate errors and therefore forecasting accuracy may drop significantly as the forecasting horizon increases.
In sum, MIMO fits our forecasting goal as well as overcomes the limitations of the Direct and Recursive strategies (Bontempi, Ben Taieb, Le Borgne, 2013).

Having defined the "technical" aspects linked to multivariate forecasting, the next subsection presents the chosen ML technique.

	M-12	M-11	M-10	M-9	M-8	M-7	M-6	M-5	M-4	M-3	M-2	M-1	target
Jan 2016	80825.2	91782.3	90931.5	98501.7	80030.8	106222.9	46381.0	45911.5	159473.1	85504.0	128610.5	67228.5	0.0
Feb 2016	91782.3	90931.5	98501.7	80030.8	106222.9	46381.0	45911.5	159473.1	85504.0	128610.5	67228.5	0.0	73994.2
Mar 2016	90931.5	98501.7	80030.8	106222.9	46381.0	45911.5	159473.1	85504.0	128610.5	67228.5	0.0	73994.2	180650.5
Apr 2016	98501.7	80030.8	106222.9	46381.0	45911.5	159473.1	85504.0	128610.5	67228.5	0.0	73994.2	180650.5	86413.0
May 2016	80030.8	106222.9	46381.0	45911.5	159473.1	85504.0	128610.5	67228.5	0.0	73994.2	180650.5	86413.0	87378.0
Jun 2016	106222.9	46381.0	45911.5	159473.1	85504.0	128610.5	67228.5	0.0	73994.2	180650.5	86413.0	87378.0	92594.4
Jul 2016	46381.0	45911.5	159473.1	85504.0	128610.5	67228.5	0.0	73994.2	180650.5	86413.0	87378.0	92594.4	60463.8
Aug 2016	45911.5	159473.1	85504.0	128610.5	67228.5	0.0	73994.2	180650.5	86413.0	87378.0	92594.4	60463.8	55468.8
Sep 2016	159473.1	85504.0	128610.5	67228.5	0.0	73994.2	180650.5	86413.0	87378.0	92594.4	60463.8	55468.8	49893.0
Oct 2016	85504.0	128610.5	67228.5	0.0	73994.2	180650.5	86413.0	87378.0	92594.4	60463.8	55468.8	49893.0	77093.5
Nov 2016	128610.5	67228.5	0.0	73994.2	180650.5	86413.0	87378.0	92594.4	60463.8	55468.8	49893.0	77093.5	0.0
Dec 2016	67228.5	0.0	73994.2	180650.5	86413.0	87378.0	92594.4	60463.8	55468.8	49893.0	77093.5	0.0	117888.2
Jan 2017	0.0	73994.2	180650.5	86413.0	87378.0	92594.4	60463.8	55468.8	49893.0	77093.5	0.0	117888.2	161379.1
Feb 2017	73994.2	180650.5	86413.0	87378.0	92594.4	60463.8	55468.8	49893.0	77093.5	0.0	117888.2	161379.1	185548.2

FIGURE 3.5: The wide format required to apply multivariate forecasting. Each line now is composed of 12 observations, in green, and the variable of interest, in red, is the target. As we try to predict the next month's value, the observations shift backwards by one step until they do not belong to the the required one-year horizon anymore (orange). Source: own creation.

3.4.3 Gaussian Process Regression

GP regression is a non-parametric method that generates predictions by finding a distribution over the possible functions, consistent with the observed data, through Bayesian inference. Each of these functions is allotted a probability and the weighted average of this probability distribution then represents the most probable function underlying the set of data points at hand.

Starting with an assumed prior distribution, the training set enables the incorporation of additional information into our model. This lets us first form the joint distribution $P(train \cap test)$ between the test and the training points, resulting in a multivariate normal distribution spanning the space of all possible function values for the target function.
Such a joint distribution is then instrumental to obtain the posterior $P(train|test)$, which is also distributed normally. The key importance linked to this step is that the resulting set of functions is forced to precisely pass through each training point.

Just as a multivariate normal distribution is completely specified by a mean vector and covariance matrix, a GP is fully specified by a mean function and a covariance function.

$$f(x) \sim GP(m(x), k(x_i, x_j))$$

Configuring m is straightforward since it is common practice to assume it equal to 0, whereas setting the covariance matrix is a bit more intriguing.
This matrix is determined by its covariance function k – also called *kernel* – which receives pairwise combinations of all available points as an input and returns a similarity measure between each couple as a scalar. Since the kernel describes the similarity between the values of the function, it therefore controls the possible form the predicted function can assume by determining which type of functions, from the

space of all possible functions, are more probable.

To sum up, for a given training set, there are potentially infinitely many functions that fit the observations. GPs identifies a posterior distribution over the most probable functions that represents the training set as close as possible.
Making a prediction using a GP, thus, eventually boils down to drawing samples from this distribution, and such a probabilistic approach also enables to integrate the confidence of the prediction into the regression results.

In this section, we had a glimpse on how the GP regression can model diverse functions by just defining a covariance function. More details about the most common covariance functions and their combinations to model complex functions will be discussed in the next paragraph.

3.4.3.1 Common Kernels and their Combinations

The previous subsection has already presented how the covariance function measures similarity between two points and determines which functions are likely under the GP prior - subsequently determining the generalization properties of the model. In other words, selecting a kernel entails making an implicit assumption about the shape of the function to be encoded with the GP. This belief could for instance regard the smoothness of the function or its periodicity.
In this thesis, four "basic" kernels will be employed. These are:

- *Radial Basis Function (RBF)* assumes that the underlying function is smooth and infinitely differentiable;
- *Periodic (Per)* enables modeling periodic functions by producing perfectly repeating patterns;
- *Rational Quadratic (RQ)*, unlike the RBF kernel, does not assume that the function is smooth. This makes the RQ kernel appropriate to model a non-smooth, rough function;
- *Linear (Lin)* simply models linear functions.

Appendix A explains these kernels more in detail by introducing their parameters, and Figure A.1 graphically depicts some samples from these common kernels priors at varying parameters.

These so-called basic covariance functions are powerful in their own right. In many cases, however, the true data structure might not be represented by any known kernel singularly. To fill this gap, Duvenaud (2014) presents a comprehensive "tutorial" on modelling a more complex time series structure by combining several kernels together through two operations, addition and multiplication. The unique constraint placed upon these combinations is maintaining the necessary positive semi-definiteness property intact in the resulting covariance matrix.

$$k_{\text{sum}}(x_i, x_j) = k_1(x_i, x_j) + k_2(x_i, x_j)$$

$$k_{\text{product}}(x_i, x_j) = k_1(x_i, x_j) \cdot k_2(x_i, x_j)$$

Addition and multiplication create two distinct effects on the resulting kernel. In the

addition process, the characteristic of two added kernels is retained, and both traits are strongly apparent in the resulting structure.
In the multiplication process, unlike the former, the two kernels actually merge their peculiarities and only show their joint effect. Table 3.1 exhibits an overview of the the different combinations utilized in this thesis - on top of the four basic kernels - and the time series peculiarity they seek to model[2].

TABLE 3.1: Relevant Kernel Combinations

Expressed Structure	Kernel Combination
Locally Periodic	RBF x Per
Increasing Variation	RBF x Lin
Growing Amplitude	Lin x Per
Periodic Trend	Lin + Per
Periodic Noise	RBF + Per
Linear Variation	RBF + Lin
Slow-Fast Variation	RBF(long length scale) + RBF(short length scale)

Handcrafting the kernel combination is a perfectly applicable strategy to identify the true underlying structure. Yet, its feasibility on this specific problem encounters some obstacles. Firstly, there is a lack of deep knowledge on kernel combinations - and their properties - by the thesis' author. Secondly, the sheer amount of time series to be analyzed renders the manual task almost impossible. In light of the aforementioned, it sounds a reasonable approach to mechanically search amongst all kernel combinations in an efficient manner.
The next paragraph will present different search strategies available and, eventually, the chosen one more in detail.

3.4.3.2 Search Strategy

Automating the kernel composition selection can be likened to a search problem. Its ambition is identifying the most suitable kernel combination - according to a metric - out of all possible combinations. The search space of possible kernels can be depicted as a search tree (Figure 3.6).

There are three search strategies that could be implemented to scour the ideal kernel combination: the exhaustive, random, and greedy search strategies, each with its own strengths and weaknesses. The exhaustive search assesses every single node of the search tree as a candidate model. It offers the guarantee of finding the optimal combination, but at the cost of generating an enormous search space - especially when the tree level is deep - making the search very expensive.
Exactly on the other end of the strategy spectrum lies the random search strategy. This method randomly samples n nodes to be evaluated, clearly not providing any guarantee to find the ideal model.

At a middle ground stands the greedy strategy (Figure 3.7). Proposed by Duvenaud et al. (2013), this method identifies the local optimum choice at every iteration

[2]These combinations are graphically shown in Appendix A (Figures A.2 and A.3). For their detailed explanation the reader is advised to consult Duvenaud's (2014) chapter *Expressing Structure with Kernels*.

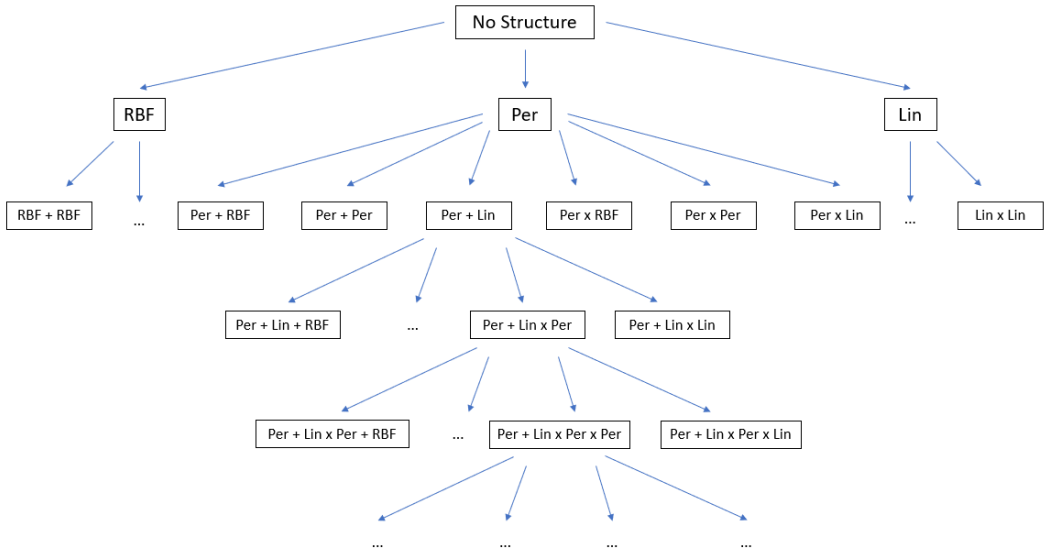

FIGURE 3.6: The tree structure that graphically represents the kernel search space. Source: own creation inspired by Duvenaud et al. (2013).

- or level. It starts by evaluating each single kernel out of all base kernels and the one returning the best metric score is chosen. In the next step, each of the base kernels taken individually is once added and once multiplied to the current best model. The optimal combination becomes the basis for the next level, and so forth.
In other words, the greedy search selects only a single branch to continue the search iteration - corresponding to the best kernel - and disregards the remaining branches.

Similarly to the random search, this approach is sub-optimal. There is in fact no guarantee that the chosen candidates will lead to the optimum solution because this might actually reside in one of the pruned branches. On the other hand, the computational complexity of the greedy technique is much lower than the exhaustive search, since the number of candidates to evaluate is significantly trimmed.
Because it explores a large number of combinations at a reasonable computational cost, the greedy search represents a valid trade-off, thereby becoming the chosen search strategy.

As a final step, this search strategy requires a stopping criterion to decide when to terminate the exploration. In this endeavor, the stopping criterion relies on the overall MAPE obtained by means of backtesting; specifically, the difference between the lowest MAPE of the current search level and the lowest MAPE of the previous search level. If such a differential has not decreased below a certain threshold, or stayed the same - thereby not showing an improvement that justifies the more complex kernel combination - the search will stop.
In this thesis, the differential is set at 1%, since the driving logic is to gain as much benefit as possible.
Before moving onto chapter 4, the next and last subsection will clarify a potential misunderstanding by differentiating between what *optimal kernel* means in terms of parameter selection and in terms of model selection.

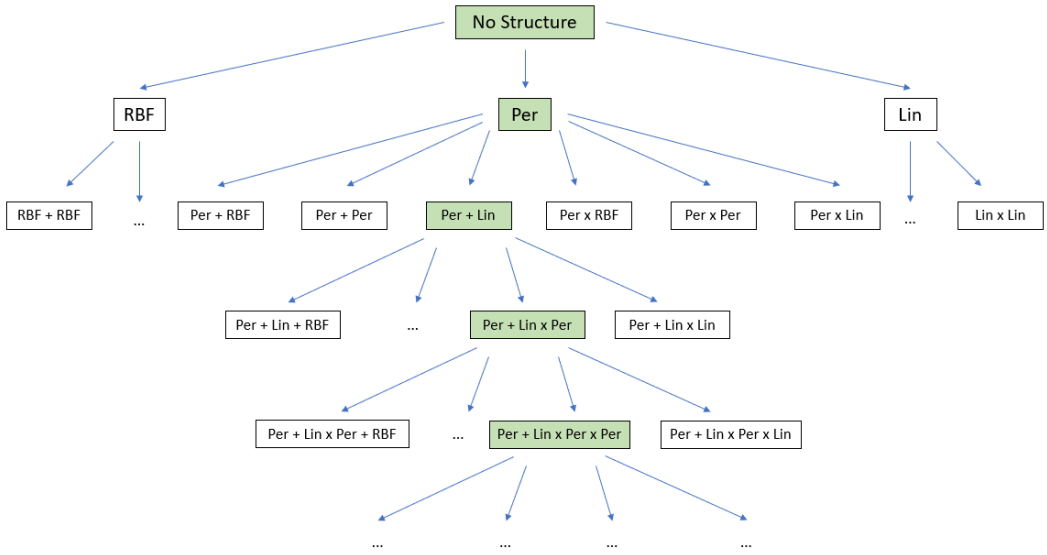

FIGURE 3.7: The greedy search selects a local optimal node at every iteration, denoted by the green rectangles. The other branches at the same level are pruned. Source: own creation inspired from Duvenaud et al. (2013).

3.4.3.3 Optimal Kernel

Distinguishing between the parameter selection of a GP model and the model selection in our search strategy is of crucial relevance. The former finds the optimal parameters of the covariance functions by maximising the marginal likelihood of a GP model, thereby achieving the right match between the capacity of a model and its fit to the data. In other words, different parameter combinations - all belonging to the *same kernel* - are compared.

On the other hand, once the optimal parameters for each of several kernel combinations are found, the latter aims at finding the optimal kernel in terms of prediction accuracy. The yearly predicted values are used to calculate the overall MAPE by means of backtesting, and the kernel combination achieving the lowest becomes the optimal kernel for a specific level of the tree. Doing so forces the model selection to focus on the forecasting performance.
Otherwise, comparing kernel combinations by the maximum likelihood value obtained after optimizing their parameters, all else being equal, will favor larger numbers of free parameters - and consequently more complex models (Duvenaud, 2014).

Unfortunately, finding the optimal parameter values is not a convex optimization problem and many local optima can be present. Overcoming this obstacle is performed in two ways. First, when searching for the optimal kernel combination, the parameter optimization process is repeated several times (20, in this case) attempting different initializations to maximize the likelihood of any given model.
Second, the logic behind the greedy search procedure also serves as a tool to provide reasonable initializations. In fact, the optimal parameters of the best kernel combination at a specific level are kept, so that the search at the next level will start the

optimization from this set of parameters; only newly introduced parameters are initialized randomly. Such a procedure is a commonly used heuristic when seeking to model residuals (Duvenaud, 2014).

Summing up, with respect to univariate techiniques MEAN, ARIMA and ETS are chosen. These are also combined in an ensemble in order to leverage each model's strengths and keep each other in balance. Amongst the several multivariate techniques available, GP regression and its capability to model complex functions by means of kernels has been selected. Identifying these complex structures requires combining such kernels, and given the sheer amount of time series, a greedy search strategy has been devised.
Operating with time series also implied handling some context-specific challenges. For one, a forecasting strategy must be defined that takes into account the business goal. Another involves the sequential nature of its observations, which impedes a classic cross-validation application since this would assume the absence of a relationship between observations. Overcoming this issue necessitates implementing a backtesting approach. Further, applying ML in this context also demands adapting the data format from long to wide, so that previous demand observations can become features.

The next chapter introduces the data set and the data preprocessing.

Chapter 4

Dataset

This section presents the database object of the study, briefly explaining the variables that compose it. It also introduces the pre-processing actions required as well as the features engineered to provide additional information in the multivariate approach.

4.1 Data Preprocessing

The datatset is extracted by the SoOp team from SAP, Novo Nordisk enterprise resource planning software. It originally contained 3690 unique raw materials - each presenting monthly observations from January 2015 to August 2020 - and 11 features. Their overview begins by introducing the core variables, that is, the ones used in both univariate and multivariate:

- *item_id*: the specific raw material identifier, composed of three parts. For instance, *1019508-2040-CMC* can be destructured into *1019508* (the item itself), *2040* (the purchase group) and *CMC* (the Senior Vice President (SVP) area);
- *demand*: the amount consumed of a given raw material during the whole month. This is our target variable;
- *consumption_month*: the month and year combination (e.g., 05-2017) referring to month the feature values were recorded;

With a mental exercise, the rest of the dataset can be divided into “dynamic” and “static” features. Features whose values modify *within* the same raw material are treated as dynamic. On the other hand, the statics might actually show dynamism between one item and another. However, once the *item_id* is singled out, these features display a constant value throughout the whole forecasting horizon, thereby not helping to pick up differences between months in the multivariate forecasting. The following variables are considered to be dynamic:

- *apo_demand*: the planned demand according to APO, Novo Nordisk production optimizer;
- *stock_level*: the level of stock available. This value is actually reflecting the stock level at the end of the previous month (M-1), thereby showing the amount at disposal at the beginning of the month object of the prediction.

On the other hand, among the "statics" can be found:

- *purchase_group*: the unique number that identifies each specific planner. Every item is purchased solely and exclusively by a determined planner;
- *planning_category*: the way a product is reordered, for instance manually or automatically after reaching a certain stock level;

- *purchasing_category*: groups of products that share aspects within the purchasing process;
- *purchasing_subcategory*: a sub-group of the above that further increase commonalities shared within a group;
- *svp_area*: the SVP area, or basically the company section whose activities consume the product;
- *plant*: the sourcing location.

Traditional data polishing tasks were performed, such as setting the right formatting, assigning the correct data type and sorting the observations in a chronological order, just to name a few. Further, it is worth mentioning that there were no missing values since the data was pre-cleansed by the SoOp team.

Yet, a two-round filtering proved necessary in order to identify what we have defined as the "unpredictables", that is, material with the count of non-zero demand observations below an arbitrary threshold.
First and foremost, *item_ids* with *all* zero observations were removed. After this action the number of unique items dropped to 1989. The second round aimed at actually targeting the presence of a minimum number of non-zero demand observations in each year taken singularly. The logic supporting this is the idea that to make a prediction, as a bare minimum, at least two points are needed to draw a line. The cleaning slashed the data set by half, keeping 891 unique *item_ids*.

item_number	consumption_month	DAPI	OTHER	CMC	aggregate
1010042	2015-01-01	816049.0000	660.0000	380.0000	817089.0000
1010042	2015-02-01	1129244.0000	460.0000	300.0000	1130004.0000
1010042	2015-03-01	1714437.0000	940.0000	360.0000	1715737.0000
1010042	2015-04-01	1090490.0000	500.0000	400.0000	1091390.0000
1010042	2015-05-01	1054005.0000	100.0000	1100.0000	1055205.0000
1010042	2015-06-01	1177486.0000	240.0000	80.0000	1177806.0000
1010042	2015-07-01	447726.0000	440.0000	120.0000	448286.0000
1010042	2015-08-01	865828.0000	480.0000	0.0000	866308.0000
1010042	2015-09-01	1275330.0000	480.0000	480.0000	1276290.0000
1010042	2015-10-01	1043545.0000	1220.0000	340.0000	1045105.0000
1010042	2015-11-01	1374709.0000	1460.0000	120.0000	1376289.0000
1010042	2015-12-01	620848.0000	200.0000	220.0000	621268.0000

FIGURE 4.1: The demand generated by each *svp_area* (DAPI, OTHER, CMC) for the same *item_number* is summed horizontally according to the month and its result is stored in column *aggregate*. Source: own creation.

Another "trick" applied has sought to leverage the centrality of purchases made by SoOp. Raw materials are in fact bought as a unique entity, and then the different *svp_area* will consume them from a shared stock. Therefore, in order to enhance the information available relative to raw materials employed in more than one SVP areas, their value was grouped along these areas. Figure 4.1 exhibits it neatly.

Harmonizing the dataset, then, implied the application of this approach to another quantitative variable requiring such an aggregation level, namely *apo_demand*. On the contrary, *stock_level* is already reported as a whole, and not split according to the SVP area.

Further, the reader can recall the problem re-framing carried out in subsection 3.4.1, also belonging to data pre-processing. Linked to this, considering our backtesting fixed-window of twelwe months and that 2020 was not over yet, its available eight months did not participate in the backtesting. This choice therefore made 2019 the last available year for training purposes.

Having leveraged the existing features, the next section will show how some additional features were extracted from the the aforementioned in order to enhance the predictive capability of GP.

4.2 Feature Selection and Engineering

After selecting the suitable features for the multivariate part, new variables were devised in an attempt to provide the algorithm with an additional help in its quest to uncover patterns.
These features - dynamic as intend it in this thesis - aim at capturing the existence of a clear periodicity:

- *order_gap*: cumulative count of months with zero demand between two observations with non-zero demand. For example, four months all with non-zero demand would obtain a value of zero. On the other hand, if between the first and the fourth month there are two zero-demand observations, then these four months would receive a value of zero, one, two and three, respectively;
- *month_number*: the number of the month being predicted.

Other variables could have been engineered as well. However, due to computational constraints[1], they were not included to avoid an additional burden. For instance:

- *holidays*: if the month includes many holiday days, production would reduce, at least in theory;
- *quarter of the year*: in a similar way to the above feature, depending on the business quarter, more output could be produced or less. For example some production plants might step up their quantity produced in order to achieve some Key Performance Indicators when the current year is about to close;
- *consumption-to-stock*: the ratio between consumption and the stock. This feature could help understand if there is a clear pattern between the amount of a material consumed and its stock level.

To sum up, the selected variables are: each month all the way to the same month of the previous year (cfr. subsection 3.4.1), *apo_demand*, *stock_level*, *order_gap* and *month_number*.

At this point, models from both methods are ready to be run. Their results will be presented in the next chapter.

[1] As of January 2021, there are no cloud computing solutions in place at Novo Nordisk. This has been the major constraint in terms of trying out different variable and hyperparameters combinations.

Chapter 5

Results

This chapter will simply present the results according to the method. These will be dissected in order to extrapolate relevant information such as which model performed better in the univariate, and empirically assess whether the greedy search actually brought improvements.

5.1 Univariate

The univariate forecasting scores a 63% accuracy across series, corresponding to a 4 percentage point improvement with respect to the baseline accuracy. Table 5.1 exhibits the split in terms of number of times the technique was chosen, as well as both its average and maximum accuracy. These numbers are graphically shown in Figure 5.1.

TABLE 5.1: Univariate Count

Techniques	Count	Average Accuracy	Max Accuracy
MEAN	250	59%	94%
ETS	287	66%	95%
ARIMA	182	61%	96%
Ensemble	172	64%	97%

ETS ranks first as the most used technique as well as the most accurate. Despite being the least utilized, the ensemble methodology is second in terms of average accuracy. Digging a bit deeper, we can also retrieve such values for the "all-inclusive" and the "selective" types. The former represents 97 of the ensemble models employed, achieving an accuracy of 63%, whereas the 75 models belonging to the latter attains 65%. Their performance differential is not as striking to justify the selective approach; perhaps, setting the threshold (subsection 3.3.4) higher might actually make this type of ensemble perform better.
With respect to the two remaining techniques, ARIMA performs perfectly in line with the whole lot, and MEAN, albeit the simplest model, proves to be almost as effective as the other techniques.
Finally, a note on the whole script efficiency. From data preparation to forecast generation took about 30 minutes, an astounding improvement compared to the previously required 5 days. Further, human's decisions relying on "gut feeling" are limited to basically none.

In the next section, the multivariate results are presented.

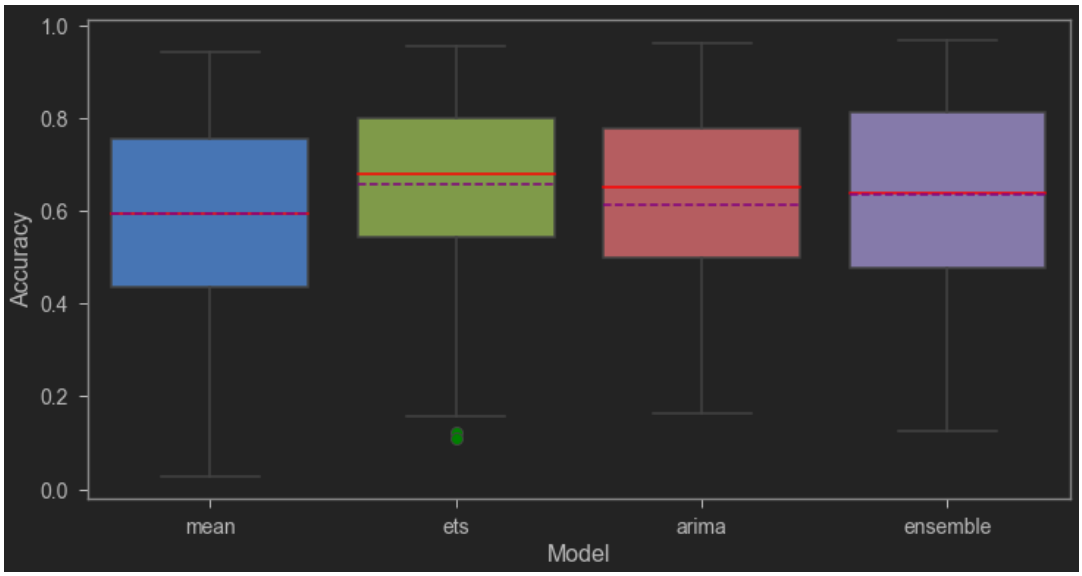

FIGURE 5.1: Boxplot of univariate accuracy by model. The red line is the median, whereas the violet dotted line represents the mean. Source: own creation.

5.2 Multivariate

Applying GP Regression enables achieving an overall accuracy of 68%, only 1 percentage point short of the target accuracy - set at 69%. Running the whole script took roughly 18 hours, with the time being affected by the number of random parameter starts - to avoid local maxima - and the number of forward selection rounds, which entails fitting more parameters at each new iteration.
A key aspect to assess, now, is whether the model forward selection (FS) implemented by means of greedy search actually proved valuable. The number of FS pushed beyond 1 round is 418, accounting for roughly 47% of the total. Table 5.2 presents many interesting statics, grouped according to the number of FS completed, and Figure 5.2 depicts it in a boxplot.

TABLE 5.2: Forward Selection

Rounds	Count	Average Accuracy	Max Accuracy	Average Gain	Max Gain
1	473	67%	99%	NA	NA
2	285	70%	98%	6%	42%
3	99	68%	95%	10%	37%
4	23	72%	95%	15%	47%
5	8	66%	90%	16%	29%
6	3	84%	94%	31%	45%

First of all, it can be noted that the number of rounds are inversely proportional to number of series that required such additional rounds. In other words, as more complicated models are trying to be built, an ever smaller number of series secure an accuracy improvement that justifies this additional level of complexity.
Some other general statistics of interest are the maximum gain reached, namely 47%,

and the maximum accuracy achieved equal to 99%.

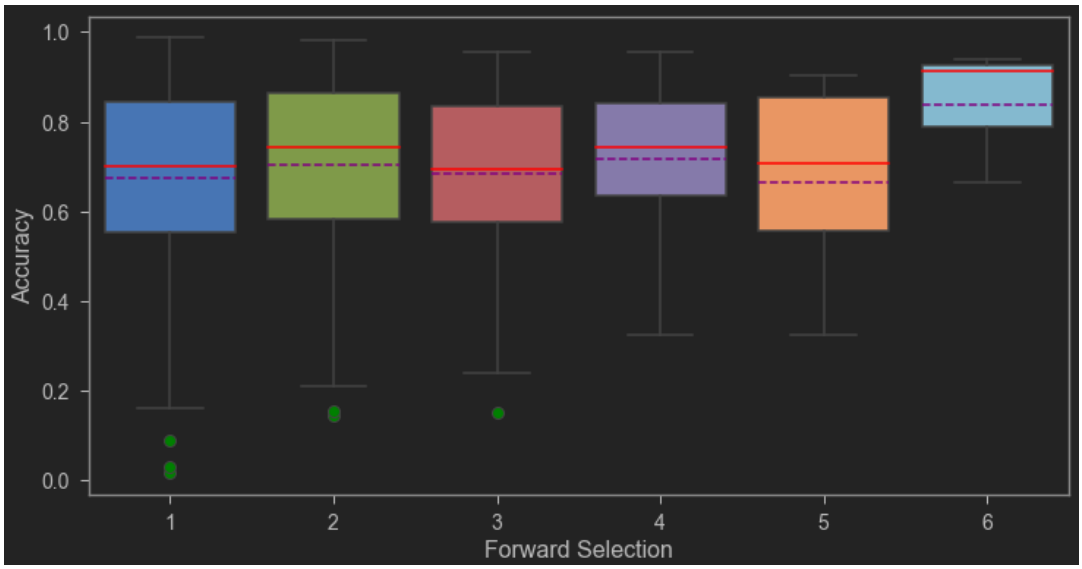

FIGURE 5.2: Boxplot of multivariate accuracy by forward selection rounds. The red line is the median, whereas the violet dotted line represents the mean. Source: own creation.

Delving into the different groupings, except for the last one which somehow sticks out from the rest, FS rounds exhibit rather similar figures in terms of average and largest accuracy (Figure 5.2).
Finally, what actually assumes a great importance is the average gain, since it further empirically supports the usefulness of FS. In fact, as neatly depicted in Figure 5.3, it can be seen how the average (and median) gain surges as the number of rounds increases.

Another characteristic of crucial relevance in a multivariate setting is evaluating feature importance. Figure 5.4 reports these findings[1]. The counts reported under each column refers to the number of times a feature has obtained a relevance score of 1 - that is, maximum relevance - calculated per each time series. Overall, the four exogenous features jointly accumulate 67% of the total relevance. Three of these stand on the podium. The first is *apo_demand*. Being APO an optimizer, it could be speculated that this feature actually represents a proxy for the effect of variables such as delivery schedules, lead times, productions cycles and other production constraints. Second comes *order_gap*. It seems that this feature manages to pick up recurring consumption patterns by counting the number of months without consumption prior to a month with consumption. Finally, *stock_level* completes the podium. The rationale behind its relevance reasonably seems to be linked to the fact that a raw material consumption is upper bounded by its quantity in stock.

In the next and last section a quick one-to-one comparison between the two methods is carried out, in order to further investigate their performances.

[1]Feature importance has been calculated relying on an adapatation of Paananen, Piironen, Andersen & Vehtari's (2019) code to fit *scikit-learn*'s GP library.

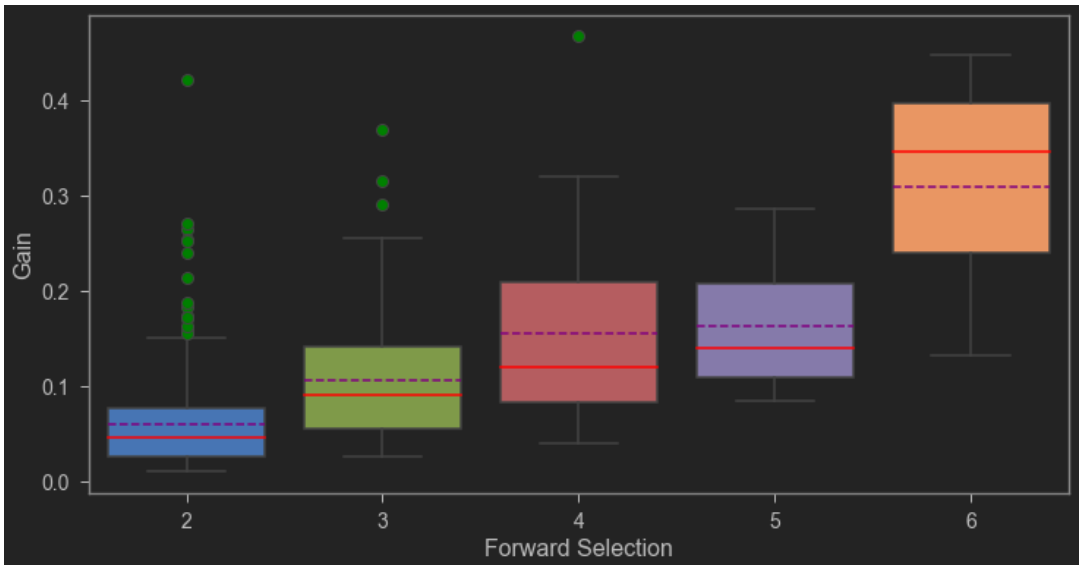

FIGURE 5.3: Boxplot of multivariate accuracy gain by forward selection rounds. The red line is the median, whereas the violet dotted line represents the mean. Round 1 is not reported, since it is the base upon which calculate gains. Source: own creation.

5.3 Results Comparison

A last angle to be investigated remains drawing a comparison at time series level. Comparing the accuracy attained by both methods, it emerges that GP outperformed the univariate models for 581 series, whereas the opposite happened 310 times.
This hints at the fact that there is a margin to further improve the accuracy of the overall forecasting process. In fact, selecting the "best of both worlds", regardless of the method, enables reaching an accuracy of 71%.

In conclusion, this thesis' results are in line with literature findings that argue in favor of multivariate methods, demonstrating that these perform better since they can leverage external information.
At the same time though, it also stresses the fact that each method could outperform the other depending on the time series.

M-12	M-11	M-10	M-9	M-8	M-7	M-6	M-5	M-4	M-3	M-2	M-1	order_gap	stock_level	month_number	apo_demand
45	43	38	35	20	21	21	26	15	14	16	27	207	68	27	363

FIGURE 5.4: The count, across time series, of how many times a feature was the most relevant. Source: own creation.

Chapter 6

Conclusions

This thesis had a twofold aim. First, improving SoOp's forecasting process in terms of both accuracy and efficiency. Second, helping to shed some light on the univariate-multivariate debate in the forecasting realm. It has started by briefly introducing Novo Nordisk, SoOp department and its current forecasting process setup, characterized by a heavy reliance on human decisions. Another distinctive peculiarity of this forecasting process is dealing with time series spanning many different unit of measurements, thereby requiring the adoption of MAPE. Based on this metric, the current baseline accuracy stood at 59% and the goal was to achieve at least a 10 percentage point improvement.

Attaining a larger accuracy required uncovering the best methods in the forecasting realm by scouring the existent literature. The review outcome has largely leaned in favor of ML applications. Yet, it has also shown the lack of a clear-cut answer on the matter: each class of models might outperform the other under certain conditions.

In light of these findings, both univariate and multivariate applications have been pursued. With respect to the former classic techniques such as MEAN, ARIMA and ETS were chosen. These were also combined in an ensemble in order to leverage each model's strengths and keep each other in balance. Amongst the several multivariate techniques available, GP regression and its capability to model complex functions by means of kernels was selected. Identifying these complex structures required combining such kernels, and given the sheer amount of time series, a mechanical greedy search strategy operating as a forward selection method was devised.
Operating with time series also implied handling some context-specific challenges. The sequential nature of its observations impedes a classic cross-validation application since this would assume the absence of a relationship between observations. Overcoming this issue required implementing a backtesting approach, in other words, a rolling forecast evaluation that splits a time series over multiple time points. Further, applying ML in this context also demanded adapting the data format from long to wide, so that previous past demand values could enter the model as features.

This problem re-framing was only one of the actions taken during the data preprocessing phase. Besides the usual tasks, worth of mention has been aggregating demand regardless of the company area that consumed the product. This enabled leveraging SoOp's centralization of purchases, thereby enhancing the information available. In this vein, additional features aimed at picking up periodicity were engineered.

Results have shown how a multivariate approach (68%) outperformed the univariate models (63%), albeit the former (18 hours) being much slower than the latter (30 minutes). Additional analysis were carried out, dissecting the results of both methods. Important to note, in particular, is that the four exogenous features actually were the most relevant predictors 67% of the times. Finally, combining the "best of both worlds" enhanced accuracy up to 71%.

To sum up, by converting these findings into answers to our research goals, it can be claimed that:

1. Accuracy has been improved by 11 percentage points and the forecasting process time drastically reduced from 5 days to less than one;

2. GP regression delivers better overall results than traditional statistical models, since the former leverages additional information carried out by features other than demand itself.
   Yet, it seems clear that each class of models might outperform the other depending on the time series at hand.

Appendix A

Kernels

A.1 Basic Kernels

In Chapter 3 paragraph 3.4.3.1 some basic kernel functions were briefly introduced. These are explained here more in detail (Duvenaud, 2014) and displayed in Figure A.1.

A.1.1 Radial Basis Function

A Radial Basis Function (RBF) kernel - also known as Squared Exponential - is given by:

$$k(x_i, x_j) = \sigma^2 \exp\left(-\frac{d(x_i, x_j)^2}{2\lambda^2}\right)$$

This kernel exhibits two parameters. The noise variance, σ^2, shapes the vertical scale changes, whereas the length scale parameter λ^2 determines function changes along the horizontal scale.
The latter is actually more interesting since it plays an important role in shaping our prior belief about how the underlying function would look like. A small length scale causes a more rapidly changing, suitable for rapidly varying functions. In contrast, a large length scale implies a slow change, thereby creating a very smooth function.

Notably, a RBF kernel assumes that the underlying function is smooth and infinitely differentiable.

A.1.2 Periodic

A periodic kernel is given by:

$$k(x_i, x_j) = \sigma^2 \exp\left(-\frac{2\sin^2(\pi d(x_i, x_j)/p)}{\lambda^2}\right)$$

Such a kernel enables modeling functions repeating themselves exactly. Its σ^2 and λ parameters carry out the same effect as in the RBF kernel, whereas the p is responsible of determining the distance between repetitions of the function.
A problem of the Periodic kernel is that generates an exactly repeating structure. Since repeating patterns in real world data usually do not have precise oscillations, this limitation can be overcome by multiplying with an RBF kernel. Their combination adds some flexibility, enabling the repeating part of the function to vary its shape over time.

A.1.3 Rational Quadratic

A rational quadratic (RQ) can be seen as an infinite sum of RBF kernels with multiple length scales (Rasmussen and is given by:

$$k(x_i, x_j) = \sigma^2 \left(1 + \frac{d(x_i, x_j)^2}{2\alpha\lambda^2}\right)^{-\alpha}$$

Unlike the RBF kernel, an RQ kernel does not assume the function underlying to be smooth. Besides the noise variance, σ^2, and the length scale parameter λ, this kernel boasts an additional parameter compared to the RBF, the the power parameter α. The long-term variation is controlled by λ as in the RBF, whereas α enables controlling for the rapidity of the local variation: the greater its value the quicker the such a variation.

A.1.4 Linear

The linear kernel is given by:

$$k(x_i, x_j) = \sigma^2 + x_i \cdot x_j$$

Perhaps the simplest kernel, where the only parameter is the noise variance σ^2, which shapes the vertical length scale of the function. It is commonly combined with itself to achieve a desired level of exponentiation.

A.2 Samples from Basic Kernels Priors

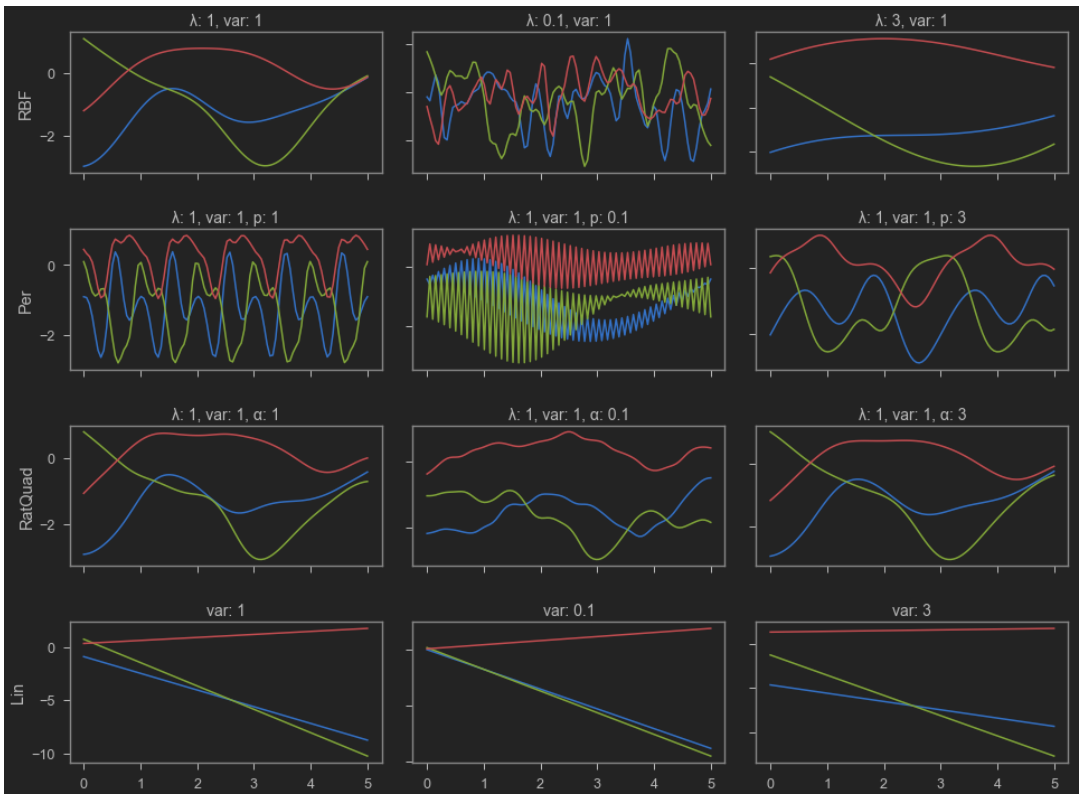

FIGURE A.1: Random samples from selected basic covariance functions. Each row represents a specific kernel and each column features a different parameter set. Source: own creation.

A.3 Common Kernels Combinations

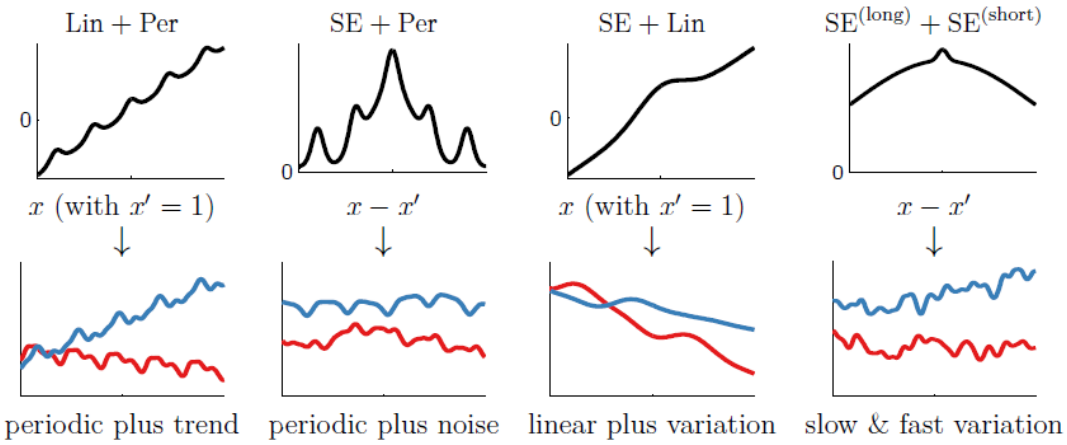

FIGURE A.2: Examples of structures expressible by adding kernels. Source: Duvenaud (2014).

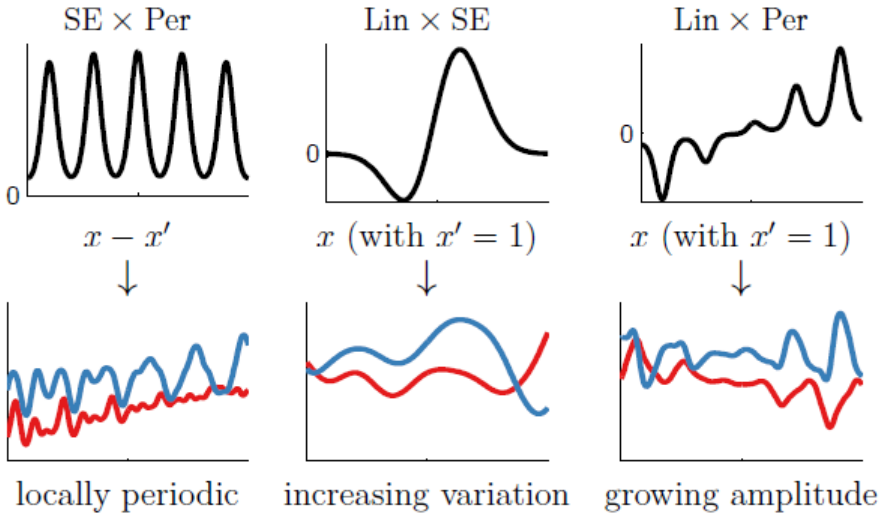

FIGURE A.3: Examples of structures expressible by multiplying kernels. Source: Duvenaud (2014).

Appendix B

Bibliography

Alon, I., Qi, M., & Sadowski, R. J. (2001). Forecasting Aggregate Retail Sales: a Comparison of Artificial Neural Networks and Traditional Methods. *Journal of Retailing and Consumer Services*, Vol. 8(3), pp. 147–156.

Ampazis, N. (2015). Forecasting Demand in Supply Chain Using Machine Learning Algorithms. *International Journal of Artificial Life Research (IJALR)*, Vol. 5(1), pp. 56-73.

Bontempi G., Ben Taieb S. & Le Borgne YA. (2013). Machine Learning Strategies for Time Series Forecasting. In: Aufaure MA., Zimányi E. (eds) *Business Intelligence*. Lecture Notes in Business Information Processing, Vol 138. Springer, Berlin, Heidelberg

Carbonneau, R., Laframboise, K., & Vahidov, R. (2008). Application of Machine Learning Techniques for Supply Chain Demand Forecasting. *European Journal of Operational Research*, Vol. 3, pp. 1140–1154.

Chopra, S. & Meindl, P., (2013). *Supply Chain Management*. 5th ed. Upper Saddle River, N.J.: Pearson Prentice Hall.

Crone, S. F., Hibon, M. & Nikolopoulos, K. (2011). Advances in Forecasting With Neural Networks? Empirical Evidence from the NN3 Competition on Time Series Prediction. *International Journal of Forecasting*, Elsevier, Vol. 27(3), pp. 635-660.

Duvenaud, D. K., Lloyd, J. R., Grosse, R. B., Tenenbaum, J. B. & Ghahramani, Z. (2013). Structure Discovery in Nonparametric Regression through Compositional Kernel Search. In *ICML*, Vol. 3, pp. 1166–1174.

Duvenaud, D. (2014). Automatic Model Construction with Gaussian processes (PhD thesis). Retrieved from: https://www.cs.toronto.edu/ duvenaud/thesis.pdf.

Heizer, J., Render, B. & Munson, C. (2017). *Principles Of Operations Management - Sustainability And Supply Chain Management*. 10th ed. Essex, England: Pearson Education Limited.

Hyndman, R. J (2018. June 24). A Forecast Ensemble Benchmark. *robjhyndman*. Retrieved from: https://robjhyndman.com/hyndsight/benchmark-combination/.

Hyndman, R.J., & Athanasopoulos, G. (2018). *Forecasting: principles and practice*. 2nd ed. Melbourne, Australia: OTexts. Retrieved from: https://otexts.com/fpp2/.

Hyndman, R. J., & Koehler, A. B. (2006). Another Look at Measures of Forecast Accuracy. *International Journal of Forecasting*, Vol. 22, pp. 679–688.

Hyndman, R. J., Koehler, A. B., Ord, J. K., & Snyder, R. D. (2008). *Forecasting with Exponential Smoothing: The State Space Approach*. Springer-Verlag.

Kandananond, K. (2012a). A Comparison of Various Forecasting Methods for Autocorrelated Time Series, *International Journal of Engineering Business Management*, Vol. 4, no. 1, pp. 187–192.

Kandananond, K. (2012b). Consumer Product Demand Forecasting Based on Artificial Neural Network and Support Vector Machine. *World Academy of Science, Engineering and Technology* 63 (2012), 372–375.

Makridakis S., Spiliotis E, Assimakopoulos, V. (2018). Statistical and Machine Learning Forecasting Methods: Concerns and Ways Forward. *PLoS ONE 13(3): e0194889*. Retrieved from: //doi.org/10.1371/journal.pone.0194889.

Merkuryeva, G., Valberga, A., Smirnov, A. (2019). Demand Forecasting in Pharmaceutical Supply Chains: A Case Study. *Procedia Comput. Sci*, Vol. 149, pp. 3–10.

Mirriam Webster (n.d.). Forecast. In *Merriam-Webster's collegiate dictionary*. Retrieved from: https://www.merriam-webster.com/dictionary/forecast

Moosa I.A. (2000). Univariate Time Series Techniques. In: *Exchange Rate Forecasting: Techniques and Applications*. Finance and Capital Markets Series. Palgrave Macmillan, London

Paananen, T., Piironen, J., Andersen, M. R. & Vehtari, A. (2019). Variable Selection for Gaussian Processes via Sensitivity Analysis of the Posterior Predictive Distribution, in *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 1743–1752.

Sarhani, M. & El Afia, A. (2014). Intelligent System Based Support Vector Regression for Supply Chain Demand Forecasting. In: *Second World Conference on Complex Systems (WCCS)*: IEEE, pp. 79–83.

Sourcing Operations. (2020). Internal Novo Nordisk presentation: Unpublihsed.

Souza, G.C. (2014). Supply Chain Analytics. *Business Horizons*, Vol. 57(5), pp. 595–605.

Tiwari, S., Wee, H.M., & Daryanto, Y. (2018). Big Data Analytics in Supply Chain Management Between 2010 and 2016: Insights to Industries. *Computers Industrial Engineering*, Vol. 115, pp. 319–330.

Usuga Cadavid, J. P., Lamouri, S. & Grabot, B. (2018). Trends in Machine Learning Applied to Demand Sales Forecasting: A Review. In: *"International Conference on Information Systems, Logistics and Supply Chain"*.

Walch, K. (2019, October 20). Are We Heading For Another AI Winter Soon?. *Forbes*.

Retrieved from: https://www.forbes.com/sites/cognitiveworld/2019/10/20/are-we-heading-for-another-ai-winter-soon/?sh=3344252d56d6.

Waller, M. A. & Fawcett, S. E., (2013). Data Science, Predictive Analytics, and Big Data: A Revolution That Will Transform Supply Chain Design and Management. *Journal of Business Logistics*, Vol. 34, pp. 77–84.

Wang, G., Gunasekaran, A., Ngai, E. W., & Papadopoulos, T. (2016a). Big Data Analytics in Logistics and Supply Chain Management: Certain investigations for Research and Applications. *International Journal of Production Economics*, Vol. 176, pp. 98–110.

Wenzel, H., Smit, D., Sardesai, S. (2019). A Literature Review on Machine Learning in Supply Chain Management, In: Kersten, Wolfgang Blecker, Thorsten Ringle, Christian M. (Ed.): *Artificial Intelligence and Digital Transformation in Supply Chain Management: Innovative Approaches for Supply Chains*. Proceedings of the Hamburg International Conference of Logistics (HICL), Vol. 27, ISBN 978-3-7502-4947-9, epubli GmbH, Berlin, pp. 413-441.

Who we are. (2020). Retrieved from: https://www.novonordisk.com/about/who-we-are.html.

Yang, C. L. & Sutrisno, H., (2018). Short-Term Sales Forecast of Perishable Goods for Franchise Business. In: *"Cybernetics in the next decades"*. Piscataway, NJ, pp. 101–105.