Predictive models for in-store workforce optimization

Viviana de Oliveira Dias
Engenharia Matemática
Departamento de Matemática
2019

Superviser
Prof. Dr. Joaquim Pinto da Costa, Faculdade de Ciências da Universidade do Porto

Company superviser
Eduardo Coelho, Area Manager - Corporate Functions, Business Information and Technology Sonae MC

FACULDADE DE CIÊNCIAS
UNIVERSIDADE DO PORTO

Todas as correções determinadas pelo júri, e só essas, foram efetuadas.

O Presidente do Júri,

Porto, ______/______/________

MSc

Resumo

No retalho, o sucesso não depende apenas da qualidade e preço dos produtos mas também da qualidade do serviço ao cliente prestada. Para empresas de retalho de escala elevada é necessária uma solução de gestão da força de trabalho para atingir uma melhor resposta à afluência de clientes e aumentar a produtividade e eficiência.

Este relatório de estágio fornece os resultados obtidos no âmbito do projeto de 5 meses *Call for Solutions* proposto pela Sonae, o qual teve como principal objetivo uma solução do gestão da força de trabalho nas áreas de frescos do retalho. Este processo consistiu em dois desafios principais: desenvolver um modelo de previsão de vendas e converter a previsão numa estimação do número ótimo de trabalhadores a estarem presentes em cada secção por cada 15 minutos.

Uma combinação de métodos estatísticos, *Machine Learning*, otimização e algoritmos simples foram usados a fim de cumprir os objetivos deste projeto.

Key-words
Gestão da força de trabalho; Modelos de previsão; Retalho alimentar.

Abstract

In retail, success depends not only on product's quality and prices but also on providing high quality customer service. A workforce management solution is necessary for large-scale retail companies to achieve a better response to customers influx and an increase productivity and efficiency.

This internship report provides the results obtained for the 5 months challenge *Call for Solutions* proposed by Sonae, and it had the main goal of providing a workforce optimization solution in the fresh food retail area. This process consisted in two main challenges: develop a multi-step sales forecast and convert the sales prediction into an optimal number of employees for time lags of 15 minutes.

A combination of statistical methods, Machine Learning, optimization and simple algorithms were used in order to meet the goals of this project.

Key-words
Workforce Management; Forecast; Food Retail.

Contents

List of Figures

List of Tables

Glossary

Hue (Hadoop User Experience) is an open-source web interface that supports Hadoop and its ecosystem. 24

KPI (Key Performance Indicator) is a measure of your performance against key business objectives. 23

Microsoft SQL Server is a relational database management system with the primary function of storing and retrieving data. 24

POS (Point of Sale) is where a retail transaction is completed. 24, 29, 30

Python is an object-oriented programming language widely used, designed by Guido van Rossum in 1991 and developed by Python Software Foundation. 24, 27

SKU (Stock Keeping Unit) is an unique identifier or code for internal use that refers to a particular product. 14, 28–30

SQL (Structured Query Language) is a domain-specific language used in programming and designed for managing data held in a relational database. 24

1 Introduction

Increasingly, retail stores are expected to do more with less, to reduce costs while growing sales and at the same time provide their customers with a high quality service. The right number of personnel at the right time is essential to provide a good service to costumers (Sauer and Schumann 2007 and Günther and Nissen 2010).

In regard of experience and comfort on the sales floor, according to the study *Observatório Shopper Experience* done in 2017, promoted by in-Store Media and developed by Netsonda which involved 500 portuguese families, 83% of consumers want to complete purchases rapidly and free of queues and 81% say they want to meet helpful and kind employees. Concerning the importance of products' attributes, 76% answered they want to find in hyper/supermarket chains fresh products as good as the fresh products in local marketplaces. The study also identifies the importance given to the social responsibility, 70% want to know how the hyper/supermarket chains treat their employees and if their wellness is taken into account.

Therefore, the right number of employees at the balcony service is needed to match the density curve of costumers throughout the day in order to keep the service fast and the queues short. It's important to find a good equilibrium to reduce over staffing and under staffing. If the operation is overstaffed, it will not reach a good efficiency. Being understaffed will force the personnel to work with stress and that is not desirable either.

Not less important is to have the right employee density prepared for the replenishment from the backdoor to the shelves throughout the day. It's crucial to prevent products' shelf based out-of-stocks (Grubor et al. 2017) since the product will not sell if it's not available on the shelve and having it at the store's backdoor can cause shrinkage. Furthermore, having time to see if the products on the shelves are in good conditions for the costumers to buy is essential too.

Both assignments are directly dependent on sales throughout the day since more sales imply more clients to service and also trigger the need of replenishment. Consequently, having a strong and confident sales forecast is the foundation to build a good workforce management solution. In this project we will focus on these both tasks that represent most of the daytime's workload. We will also explore an estimation of the staff level required during off hours.

That said, this report will present a 11 weeks ahead predictive model of the staff workforce needs aimed at fresh food area of Sonae's hypermarkets. This 11 weeks time planning is requested to allow a full month schedule delivery to employees at least one month ahead of the schedule's first day. The process will be composed by the construction of a sales forecast to implement on the respective study case data and posterior calculation of the number of employees needed for time periods of 15 minutes.

2 Literature review

In the past few years, workforce management has become a relevant topic. Satisfying the demands of work and business objectives with a financial balance and a efficient approach is recognized by enterprises to be an urge to meet.

A workforce management process involves calculating a prediction of the optimal staff number per a determined period of time based on the respective forecast demand. Subsequently, it also extends into creating staff schedules, attempting to fulfill the needs previously calculated and taking into account schedules' legislation. Lastly, it also should track and manage continuously the performance of employees' workforce and the model. This project will embrace the steps of forecasting and calculation of optimal staff levels (see Figure 1).

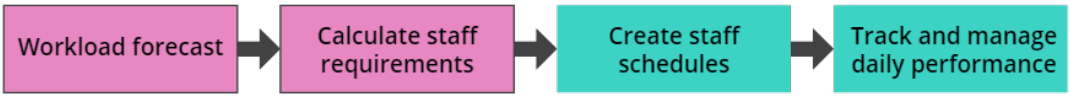


Figure 1: Basic steps of the workforce management process.

Historically, as stated on Infor 2007, companies addressed employee's management with deprecated procedures like spreadsheet-based analyses. This approaches lacks a confident sales forecast, has a subjective store operation's needs and consumes unnecessary time. Automating employee management processes can improve significantly the prediction of the real staffing levels needed throughout the day and reduces valuable time wastage that could be used on other productive activities. In compliance with Infor 2007, a study from WorkJam 2015 involving 500 U.S. service company managers and more than 700 U.S. hourly employees, states that 39% of service companies use paper schedules to create hourly worker schedules, 28% use spreadsheets, 19% use shift management software and 12% use free online tools. Besides that, 46 % of employers claim being understaffed sometimes or quite often. Over the course of a couple days, even a few understaffed shifts can have a lasting effect.

According to a research performed by *HR.com* (mentioned in Tommy Tonkins' blog post *7 ways a workforce management solution will immediately benefit your business*), companies that deploy a workforce management solution can reach a cost saving from 6% to 10% within the first year. This research also gathered some benefits noticed by the organizations involved after a solution has been implemented: increased administrative efficiency, cost saving, improved compliance, higher customer satisfaction, employee productivity boosting, reduced absenteeism and powerful business insight.

With that said, it is clear that better information about a company's workforce can help boost up its business. However, as said in McLean et al. 2016, organizations have to be willing to accept the data findings and be prepared to change their systems and processes. The expense and commitment of transition programs to roll out an operational change may be required and should not be underestimate.

2.1 Retail sector

One of the biggest sectors in the EU economy is retail. As mention in a press release of European Commission (April 2018), retail sector provides 8.6% of all jobs in the EU and has over 3.6 million enterprises (Figure 2).
Responding to customers demand flow is fundamental for retail companies to be successful in the com-

Figure 2: Retailing's contribution by size of firm in EU. *Source: European Retail Round Table*.

petitive business environment. Suitable store labor influences sales directly by having an impact on the level of sales assistance provided to costumers, and indirectly, through execution of store operational tasks such as labeling merchandise, stocking shelves and keeping the overall store ambiance (Fisher and Raman 2010). Moreover, besides its impact on sales, it also affects store's expenses being a significant amount of a store's operating expense.

2.1.1 In-store logistics

Retail sector covers many different types of stores, including department stores, supermarkets, discount stores and speciality stores. However, retail workers' basic tasks are similar:

- stocking shelves - certify products are available on shelves, refill them if products are running low on shelves but available in the backroom, check expiration dates and remove expired products, change prices labels when promotions or alterations occur;
- merchandising - create and prepare product displays to enhance its visibility or to make customers aware of some promotional offer, ensure displays remain with good aspect;
- customer assistance - help costumers locate products, explain products features, give advice;
- checkout operation - scan items, process costumers' payment, help customers pack their items;
- customer service - receive requests from customers such as exchange or return products, home delivery, warranty claims;
- store operations - organize backroom, open and close store, put any relevant signs, supervise deliveries.

Include each one of in-store tasks to build a model to optimize the staff level during the day is not necessary. In fact, its focus should be concentrated on activities that represent a considerable amount of employees' work time (Daniel Läubli and Silén 2015), such as activities related to replenishment which can reach 70% of the workload in a store.

Replenishment truly has a huge role on in-store processes. In fact, a product cannot be sold if it is not available on the sales floor to costumers to buy. Shelf-based out-of-stocks not only can reduce sales but also can lead to wastage if stock is still located in the backroom. Defective number of store employees is one of the reasons for products to be with low on-shelf availability (Grubor et al. 2017). Continuous shelf replenishment throughout the day helps reduce the amount of over-stock (difference between stock delivered to the store and stock that can be displayed on the shelf) helping the backroom not get congested and keeping products' availability on a high level (McKinnon, Mendes, and Nababteh

2007). This is especially important for perishable food since it has a low shelf life and a lost opportunity to sell it has a greater probability to lead into wastage.

Customers can react in different ways when the product they pretended to buy is out-of-stock. The outcome of Sloot, Verhoef, and Hans Franses 2005 research shows that 36% of costumers react by buying the same product from another brand, 23% postpone the purchase, 19% buy it in another store, 18% buy another product and 3% simply cancel the purchase (Figure 3).

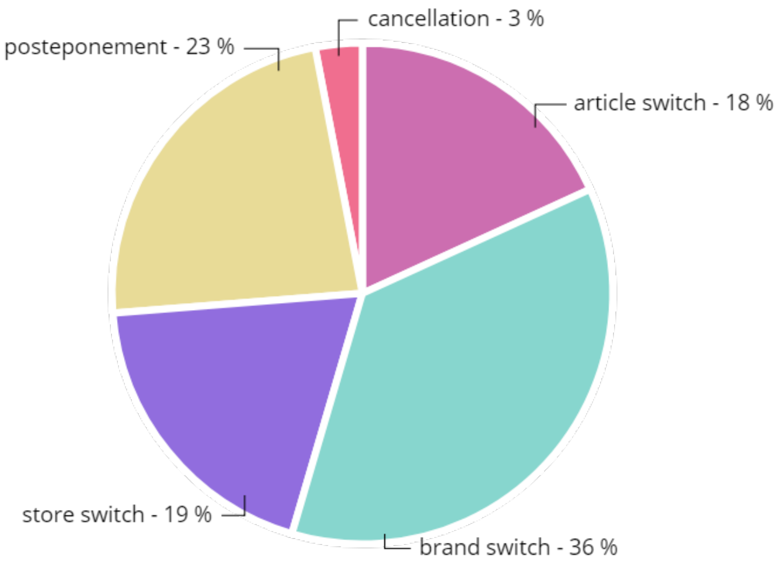


Figure 3: Customer stock-out reactions - groceries (Sloot, Verhoef, and Hans Franses 2005)

Customer service is also one of the most important components of store employees' activities. In addition to check-outs, when heading to a service balcony, costumers expect to be served quick and nicely. Service balcony includes services such as information, devolution, butcher, delicatessen, bakery, etc. Waiting for long periods of time in a queue to be attended is unpleasant for costumers and can lead into quitting the waiting line. Analysis results from Bielen and Demoulin 2007 show that waiting time satisfaction besides being a service satisfaction factor, it also influences the customer satisfaction-loyalty relationship.

2.2 Forecasting

Forecast is a prediction of future events, usually constructed by analyzing past and current data. Forecasting has applications in many sectors such as finance, team performance in sports, political, weather, economics, sales, supply chain, transport planning, earthquake predictions.

2.2.1 Workload Forecast and Staff Requirements

In order to build a prediction of staff levels we must forecast the amount of workload we will need to fulfill. Workload forecast is the foundation to the workforce management process. A poor forecast will never generate a good prediction of the actual number of employees required, so it is extremely importance for this process.

In the ambit of retail, sales volume and store's traffic are driver components for staff's workload. Replenishment during open store hours, checkout operation, customer assistance and costumer service are highly correlated with sales and store traffic. In addition, promotions, weather, holidays, events and some other variables should also be used as an input to the forecast. Likewise, to reduce bias, the occurrence of queues and items stock-outs on sales historical data should also be evaluated since it might have decreased the potential sales.

Regarding staff levels, according to Daniel Läubli and Silén 2015 and Infor 2007, to achieve a more accurate projection, retailers should use an activity-based approach (bottom-up) rather than using an arbitrary percentage of the sales' volume (top-down). Therefore, it provides an interpretable and easily adjustable forecast enabling a continuous constructive discussion based on the premises used. In fact, it can be used as a guide to management teams to understand which store activities could be speeded up or relaxed. It should be taken into account that time can differ between different stores since each store has its own structure characteristics and logistics. All that summed up, the projection seeks to provide the right number of employees at the right time performing the right activities.

2.2.2 Forecasting Models

Regarding this project, we will address some forecasting models for time series forecasting. Time series is a sequence of data points indexed in time. For instance, sales data can be considered a time series having sales (in quantity or currency) as the measured variable and the time the transaction was made as indexes.

Sales forecasting is considered hierarchical since sales can be aggregated in different levels: by SKU (Stock Keeping Unit), by section, by store, etc. There are three main planning approaches for hierarchical forecast (Rostami-Tabar et al. 2013): "top-down", "bottom-up" and hybrid approach. The concept of top-down method is starting with macro view - going general to specific. The bottom-up approach requires forecasting at the lowest level of the hierarchy and then aggregate the forecasts by summing them up - going specific to general. Between top-down and bottom-up is the hybrid approach that consists in forecasting at an intermediate level and aggregate or disaggregate as needed. Since sales are time series, they can also be aggregated in distinct time ranges. Different levels of aggregation can result in quite different performances (Zotteri, Kalchschmidt, and Caniato 2005). The choice of the appropriate level of hierarchical and temporal aggregations can be done by testing and evaluating the results and it also depends on the forecasting purpose.

Besides the aggregation options, an important thing to have in mind while choosing a forecasting model is how many time steps into the future we want to predict. Some models have extremely good results predicting one step ahead but can be weak predicting in long term since multistep-ahead prediction is more complex and difficult than one-step-ahead. With the purpose of this project in mind, planning the optimal staff levels requires an intraday forecast of several days to posterior generation of the schedules so they can be handed over to employees a couple days in advance.

Traditionally, statistical methods are used to forecast time-series. ARIMA, which is a statistical analysis model, is considered to be the most common approach for time series forecasting (Masum, Liu, and Chiverton 2018). However, the use of machine learning methods has been proposed as an alternative to statistical methods and has significantly increased in this area Bontempi, Taieb, and Le Borgne 2012; Tyralis and Papacharalampous 2017. There are a lot of controversial opinions about using machine learning methods to model time-series predictions but there are articles and challenges results where the machine learning methods have outperformed the statistical ones. In order to know which method is more suitable to our data we are going to try out some models from statistical methods and machine learning and compare their performances. Therefore, we will present theoretical foundations about each model.

2.2.2.1 Statistical methods

We will discuss the most common forecasting model, ARIMA (Box and Jenkins 1976).

ARIMA

This model was proposed in 1976 by George Box and Gwilym Jenkins and it is composed by discrete time linear equations with noise. The acronym stands for auto-regressive (AR) integrated (I) moving average (MA). Each of these components has a representative hyper-parameter that can be tuned to our data.

The auto-regressive component indicates that the variable we want to predict is regressed on its own previous values and its associated hyper-parameter p determines the number of lags to be used in the model:

$$X_t = \alpha_1 X_{t-1} + ... + \alpha_p X_{t-p} + \epsilon_t$$

where $\alpha_1, ..., \alpha_p$ are the parameters and ϵ_t is an error term. This equation could also be written like $(1 - \sum_{k=1}^{p} \alpha_k L^k) X_t = \epsilon_t$, where L^k is a power of k of the time lag operator. The time lag operator is a linear operator which transforms sequences in sequences and can be defined as $Lx_t = x_{t-1}$, for all $t \in \mathbb{Z}$. Its powers, L^k, are defined as $L^k x_t = x_{t-k}$, for all $t \in \mathbb{Z}$.

The integrated part is related to differencing. In order to apply an ARIMA model data must be stationary. A time series is said to be stationary when its statistical properties (mean, variance, autocorrelation, etc.) are all constant over time - see Figure 4. If the time series is not stationary, it can be transformed by a differencing process (replacing the current data values with the difference between their values and the previous values). Its corresponding hyper-parameter is d and it indicates the number of times the series has suffered differentiation. The first difference operator, Δ, is defined by $\Delta X_t = X_t - X_{t-1} = (1-L)X_t$. The second difference operator is defined by $\Delta^2 X_t = (1-L)^2 X_t$. This can be generalized to Δ^d.

The moving average component is a explicit formula for X_t in terms of the noise, ϵ. Basically, it is a (weighted) average of the last $q+1$ error terms, where q is the hyper-parameter. The explicit formula in this case is

$$X_t = \epsilon_t + \beta_1 \epsilon_{t-1} + ... + \beta_q \epsilon_{t-q} = \Big(1 + \sum_{k=1}^{q} \beta_k L^k\Big)\epsilon_t.$$

The union of all this concepts originates the ARIMA (p, d, q) model which form is given by

$$\Big(1 - \sum_{k=1}^{p} \alpha_k L^k\Big)(1-L)^d X_t = \Big(1 + \sum_{k=1}^{q} \beta_k L^k\Big)\epsilon_t.$$

Additionally, if the data appears to have seasonality we can use seasonal ARIMA (SARIMA) instead of increasing the order of AR or MA model's parts. Within the framework of time series, seasonality is a

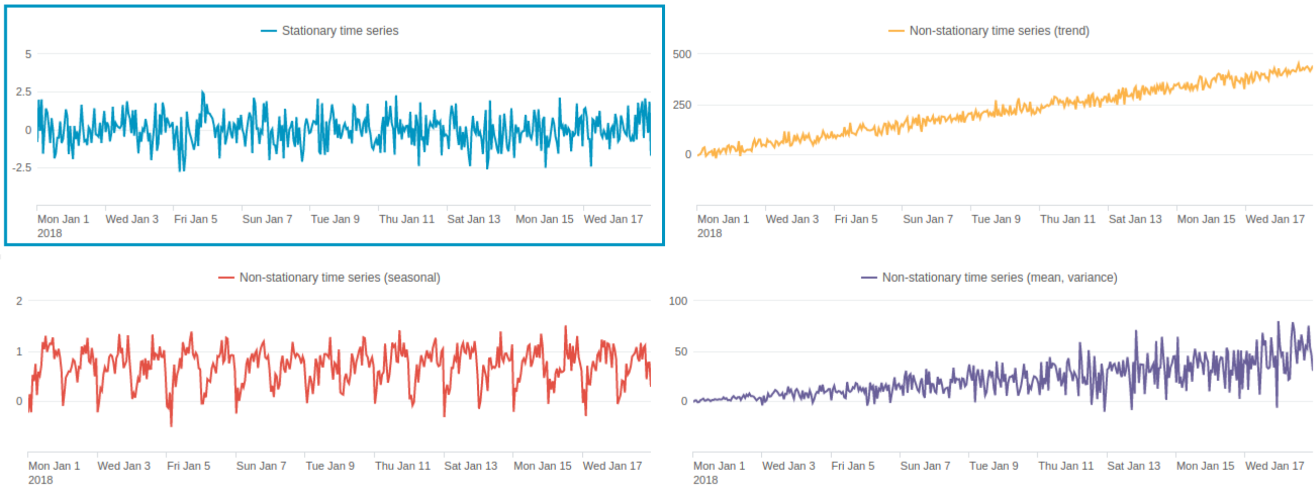


Figure 4: Comparison between stationary and non stationary time-series.

repetitive pattern of changes that occurs over S time periods and takes S time periods to be repeated again. In this case, instead of ARIMA (p, d, q) we have four additional hyperparameters – SARIMA $(p, d, q)(P, D, Q)S$, where S is the number of time periods of the seasonality and P, D, Q are the respective p, d, q for the seasonal part. SARIMA's generalized form can be written as (Box, Jenkins, et al. 2015):

$$\left(1-\sum_{k=1}^{p}\alpha_k L^k\right)(1-L)^d\left(1-\sum_{k=1}^{P}\mathsf{A}_k L^{Sk}\right)(1-L^S)^D X_t=\left(1+\sum_{k=1}^{q}\beta_k L^k\right)\left(1+\sum_{k=1}^{Q}\mathsf{B}_k L^{Sk}\right)\epsilon_t$$

.

In order to tune the hyperparameters the first step is to determine if the time-series is stationary and if seasonality exists. Then, we can use autocorrelation and the partial autocorrelation plots (Hyndman and Athanasopoulos 2018) and/or AIC_c (Akaike information criterion with correction) to identify the best p, q (and P, Q), as claimed on Brockwell, Davis, and Fienberg 1991. In the case of AIC_c, the objective is to estimate the hyperparameters that minimize $AIC_c = -2\log(L) + 2(p+q+k+1) + \frac{2k^2+2k}{n-k-1}$, where p is the autoregressive parameter, q is the moving average parameter, L is the the maximum of the likelihood function, k is the number of model parameters and n is the data sample size. The autocorrelation and partial autocorrelation plot summarize the correlation between an observation in a time series and the observations at prior time steps.

2.2.2.2 Machine Learning methods

In machine learning models the observations are independent from one another. That is not the case with time series since the closer in time observations are to each other, the more codependent they are. However, we can manipulate the problem into a machine learning supervised learning case by creating time related features. The process of creating features using domain knowledge of the data is called **feature engineering**. This does not add any new data but makes the data we have more useful. In this section we will mention a few models among the wide existing range.

Linear Regression

Although being developed in the field of statistics, linear regression can also be referred to as a machine learning model. It is both a statistical algorithm and a machine learning algorithm and it is widely used given its simplicity and interpretability. Being Y the variable we want to predict, the linear relationship between Y and other variables X_1, ..., X_p ($p \in \mathbb{N}$) can be described as

$$Y = \beta_1 X_1 + ... + \beta_p X_p,$$

where β_1, ..., $\beta_p \in \mathbb{R}_0^+$ are the parameters. In practice, this equation has an error term (ϵ) which is a random and non-observable variable. The mean squared error (MSE) is the loss function used to estimate the coefficients' values. This loss is given by the average squared difference between the true values of the variable and the predicted values,

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2, \tag{1}$$

where $\hat{Y}_i = \hat{\beta}_1 X_{i1} + ... + \hat{\beta}_p X_{ip}$ and n is the number of samples. That being said, the values for the coefficients are estimated by minimizing the loss function.

Ridge Regression

This model is similar to the linear regression model but with an addition of a regularization term. To fully understand the importance of this term, we will introduce the concept of bias-variation trade-off. Bias and variance are characteristics of the estimators. The bias is the difference between the expected estimator value and the true population parameter, measuring the estimates' accuracy, while variance measures the uncertainty in these estimates. The model's error can actually be decomposed into three parts: bias, variance and an unexplainable term. In order to have a good model's performance, we want both variance and bias to be as low as possible. A model with high bias underestimates the information of the data used to train the model and oversimplifies it. This is called underfitting since the model is unable to capture the underlying pattern of the data. However, a model with high variance holds too much on the information given by the training data so it captures the noise along with the underlying pattern. In this case the model is not able to generalize, not returning good predictions for new data. Being the opposite of underfitting, this occasion is called overfitting. The desired model has to have a good balance without overfitting and underfitting the data but there is a tradeoff between bias and variance.

The loss used for the linear regression model has the property of being unbiased but it can have a significant variance. This might happen for two main reasons: some predictor variables are highly correlated with each other, or there are too many predictors. Reducing variance at the cost of increasing some bias is a common solution for this problem and it usually has a positive result on the predictive performance of the model. This method is known as regularization.

The only difference between ridge regression and linear regression is on the loss function. So, instead of minimizing the MSE to find the best parameters, they are estimated by minimizing the MSE plus a regularization term. So, the loss is given by

$$\frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 + \lambda\sum_{j=1}^{p}\beta_j^2,$$

where n is the number of samples, p is the number of predictors/parameters and $\lambda \in \mathbb{R}_0^+$ is a tuning parameter which controls the weight of the regularization term (Saleh, Arashi, and Kibria 2019). In practice, this shrinks the coefficients towards zero, although they can never reach an exact null value.

LASSO Regression

LASSO Regression, just like Ridge Regression, is similar to linear regression with a regularization term. In addition to its regularization task, it also performs feature selection. In fact, the name LASSO stands for Least Absolute Shrinkage and Selection Operator. Similarly to Ridge Regression it penalizes the coefficients of the regression variables, shrinking some of them. However, in contrast to Ridge, it can shrink a coefficient to zero. Therefore, it is known to have the property of feature selection. This allows an improvement of the model interpretability by removing irrelevant features that are not connected with the variable we want to predict. Its loss function is given by the following formula:

$$\frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 + \lambda \sum_{j=1}^{p} |\beta_j|,$$

where n is the number of samples, p is the number of predictors/parameters and $\lambda \in \mathbb{R}_0^+$ is the tuning parameter (Fonti and Belitser 2017).

Elastic Net Regression

Elastic Net is an extension of LASSO Regression proposed by Zou and Hastie in 2005 (Zou and Hastie 2005). They state that Elastic Net frequently outperforms LASSO Regression on the grounds of real world data and a simulation study. LASSO might not be suitable when the number of samples (n) is much smaller than the number of predictors (p) or when there are grouped variables (high correlation between each other). Elastic Net overcomes these limitations by merging LASSO and Ridge regression models.

Same as Linear Regression, the relationship between the variable we want to predict and the predictors is described by

$$\hat{Y}_i = \hat{\beta}_1 X_{i1} + ... + \hat{\beta}_p X_{ip},$$

however, in this case we find the parameters β by minimizing the following function:

$$\frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 + \lambda_1 \sum_{j=1}^{p} |\beta_j| + \lambda_2 \sum_{j=1}^{p} \beta_j^2,$$

where λ_1 and $\lambda_2 \in \mathbb{R}_0^+$ are regularization terms we can tune.

Gradient Tree Boosting Machines

This model uses an ensemble technique called boosting. The idea of boosting is to train a certain K number of weak models in a gradual, additive and sequential manner. After training the base model with the given data, it adjusts the observations' weights giving a higher weight if the observation was poorly predicted by the model. Then, we build another model by training again but taking into account the new weights. The new model becomes the sum of the latest trained model with the previous one. The process is repeated for a specified number of iterations (see Figure 5). A weighted sum of the predictions made by each model are taken as the final predictions of the ensemble model.

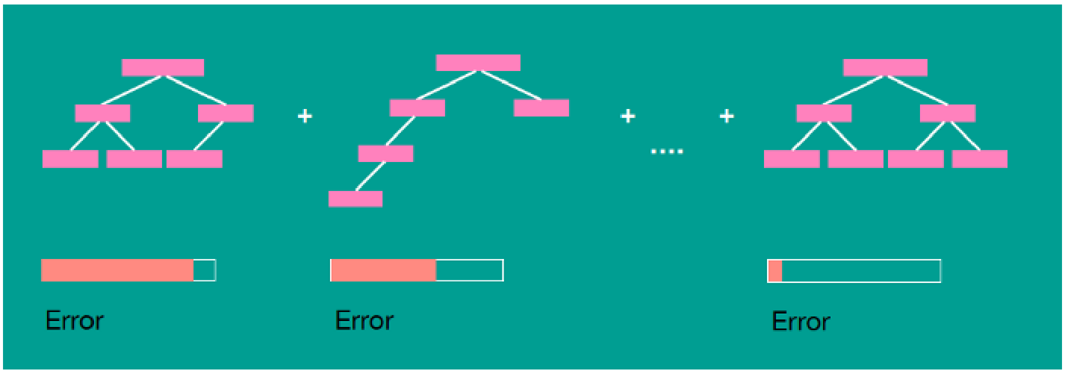


Figure 5: Simplified representation of the boosting algorithm using trees. *Source: Towards Data Science*.

That being said, the first model is build by trying to find the weak model $b(X)$ that minimizes the squared error (or another convex loss function) between the predictions and the real values:

$$b_1(X) = \text{arg min}_b \sum_{i=1}^{n} (b(X_i) - y_i)^2$$

,

being X the set of features, n the number of samples and y the variable we want to predict. Then, the subsequent model $b_2(X)$ is fitted in a way to minimize the composition of itself and the previous one, $b_1(X)$:

$$b_2(X) = \text{arg min}_b \sum_{i=1}^{n} (b_1(X_i) + b(X_i) - y_i)^2 = \text{arg min}_b \sum_{i=1}^{n} (b(X_i) - (y_i - b_1(X_i)))^2$$

.

Intuitively, $b_K(X)$ is chosen as follows:

$$b_K(X) = \text{arg min}_b \sum_{i=1}^{n} \left(b(X_i) - (y_i - \sum_{l=1}^{L-1} b_l(X_i)) \right)^2.$$

Usually, regression trees are used as weak models for the boosting process. Regression trees are a non-linear model which is achieved by successively dividing the data in smaller subgroups (recursive partitioning) based on different features. Given the whole dataset it searches the best split value and the feature that partitions the data into two regions, R_1 and R_2, that minimizes a cost function. This division leads to two nodes that are named as children of their previous node. This process is repeated successively, creating binary divisions on nodes with no children, until it reaches a predefined stopping criterion. Additionally, the resulting tree can be pruned to reduce its complexity. Different stopping criteria and pruning techniques are used depending on the purpose of the model. Having the final tree, nodes with no children are called leaves. Based on the splitting features and values, a data point will reach a leaf. The value associated with that leaf is the resulting prediction of that data point (see Figure 6).

Regularization is another important part of this model. Being a sequential addictive model of regression (or decision) trees, each of the trees should be short and simple to not overfit. Having $T \in \mathbb{N}$ as number of leaves of the corresponding b tree, $w_j \in \mathbb{R}$ as the score of the j-th leaf, $\gamma, \lambda \in \mathbb{R}_0^+$ as tuning parameters, the regularization term Ω is defined by the following formula (Chen and Guestrin 2016):

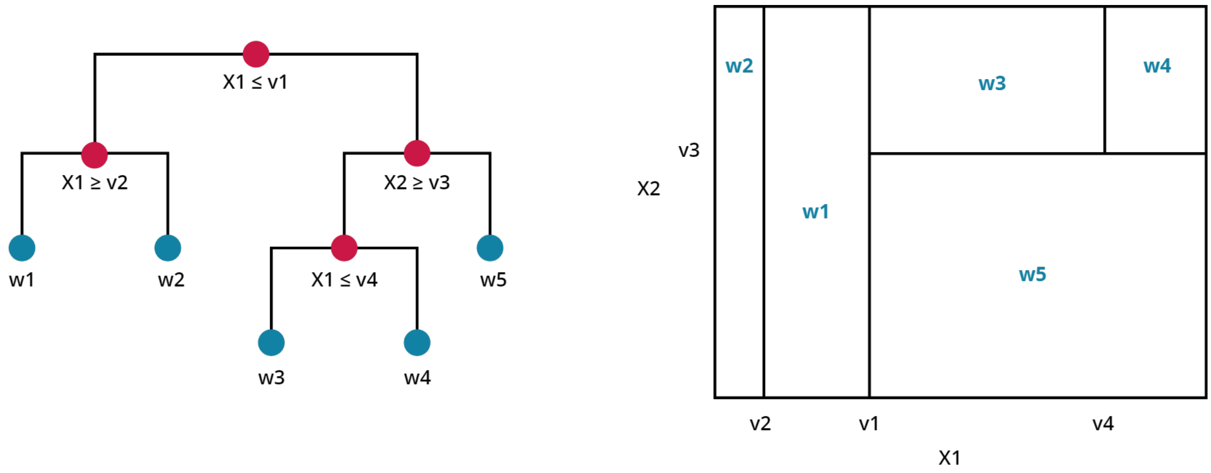


Figure 6: Example of a graph representation of a regression tree and its variable space partitions with fictional data being X_1 and X_2 features; v_1, ..., v_4 splitting values and w_1, ..., w_5 the leaves' values.

$$\Omega(b) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2.$$

Combining the loss and regularization, we obtain the objective function of the model. Therefore, in each t-th iteration, we want to find the model $b_t(X)$ that minimizes the following function (Chen and Guestrin 2016):

$$Obj^{(t)} = \sum_{i=1}^{n} \Big(b_t(X_i) - (y_i - \sum_{l=1}^{t-1} b_l(X_i))\Big)^2 + \sum_{l=1}^{t} \Omega(b_l).$$

As its name suggest, this model uses gradient descent to optimize the objective function. At each iteration t it calculates the gradient and improves b_t along the direction of the gradient to minimize the objective. Besides calculating the first order gradient $\partial_{b_t} Obj^{(t)}$, it also calculates the second order $\partial_{b_t}^2 Obj^{(t)}$ to improve the performance. However, since the objective function is not always derivable, it is considered its second order Taylor approximation:

$$Obj^{(t)} \simeq \sum_{i=1}^{n} \Big[L(y_i, b_{t-1}) + g_i b_t(X_i) + \frac{1}{2} h_i b_t^2(X_i)\Big] + \sum_{l=1}^{t} \Omega(b_l),$$

where $g_i = \partial b_{t-1} L(y_i, b_{t-1})$, $h_i = \partial^2 b_{t-1} L(y_i, b_{t-1})$ and $L(y_i, b_{t-1}) = \Big(b_{t-1}(X_i) - (y_i - \sum_{l=1}^{t-2} b_l(X_i))\Big)^2$.

Now, removing the constants terms, the objective function becomes:

$$Obj^{(t)} = \sum_{i=1}^{n} [g_i b_t(X_i) + \frac{1}{2} h_i^2 b_t(X_i)] + \Omega(b_t).$$

Therefore, we have to find the structure of the tree b_t and the scores to assign to its leaves that minimizes the function. We can define the tree as

$$b_t(X) = w_{q(X)},$$

where $q : \mathbb{R}^p \to T$ is the directing function which assigns X to the $q(X)$-th leaf and $w_{q(X)}$ its the respective leaf score.

We can define an index set, which contains the indices of all the data points that are assigned to the j-th leaf, as $I_j = \{i|q(X_i) = j\}$. Given this, the objective function can be rewritten as:

$$\begin{aligned} Obj^{(t)} &= \sum_{i=1}^{n}[g_i b_t(X_i) + \frac{1}{2}h_i^2 b_t(X_i)] + \gamma T + \frac{1}{2}\lambda\sum_{j=1}^{T} w_j^2 \\ &= \sum_{j=1}^{T}\big[(\sum_{i\in I_j} g_i)w_j + \frac{1}{2}(\sum_{i\in I_j} h_i + \lambda)w_j^2\big] + \gamma T. \end{aligned}$$

For a fixed structure $q(X)$, the optimal weight w_j^* of the leaf j is easily calculated by resolving a quadratic problem. So,

$$w_j^* = -\frac{\sum_{i\in I_j} g_i}{\sum_{i\in I_j} h_i + \lambda},$$

and the optimal value is given by

$$Obj^{(t)}(q) = -\frac{1}{2}\sum_{j=1}^{T}\frac{(\sum_{i\in I_j} g_i)^2}{\sum_{i\in I_j} h_i + \lambda} + \gamma T. \tag{2}$$

This equation 2 is used to evaluate the quality of a tree structure q. An approximate algorithm (see Chen and Guestrin 2016 for more details) is used to obtain candidates for the splitting points and features. To evaluate the candidates the following formula of the loss reduction after the split is calculated:

$$Obj_{split} = \frac{1}{2}\Big[\frac{(\sum_{i\in I_L} g_i)^2}{\sum_{i\in I_L} h_i + \lambda} + \frac{(\sum_{i\in I_R} g_i)^2}{\sum_{i\in I_R} h_i + \lambda} - \frac{(\sum_{i\in I} g_i)^2}{\sum_{i\in I} h_i + \lambda}\Big],$$

assuming I_L and I_R are the instance sets of left and right nodes after the split, and $I = I_L \cup I_R$.

The best splitting point is found recursively until it is reached a maximum tree depth (defined by the user as tuning parameter), followed by pruning out the nodes with a negative gain in a bottom-up order.

3 Case study

In this chapter we pretend to give the reader a context about the company this study relies on and explore some characteristics that may be relevant to the motivation of this project.

3.1 The company - Sonae

Sonae - *Sociedade Nacional de Estratificados* - was established in 1959. At that time the company had only one business unit regarding the production of decorative laminated products. Sonae opened the very first hypermarket in Portugal in 1985 and it was located in *Matosinhos*. Currently, Sonae is a multi-national company being present in 98 countries (Figure 7) and coordinates a variety of services in distinct business units such as retail, financial services, technology, shopping centers and telecommunications (Figure 8). In 2018, retail activity sales of Sonae have increased 7.6% and reached 6 317 million euros, being the best year of retail with every segment from the group showing a positive performance.

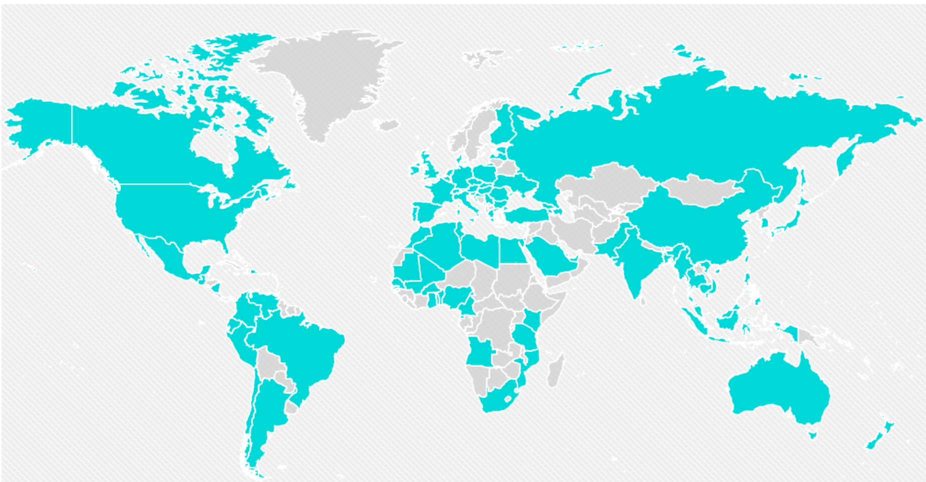


Figure 7: Sonae's global presence - 98 countries - updated in january of 2019. *Source: official Sonae's site*

Figure 8: Sonae's portfolio. *Source: official Sonae's site*

Food retailing is the focus of this project which belongs to Sonae MC, one of Sonae's affiliates. Sonae MC - Sonae *Modelo e Continente* - is the national leader in food retailing and embraces several brands. *Continente* (urban hypermarkets) is the main focus brand in this study. It has the wider range at the category and product level and it counts with 41 open stores that represent 38% of the total sell regarding all the stores of Sonae MC's food retailing - *Continente*, *Continente Modelo*, *Continente Bom dia*, *Meu Super* (franchise).

3.2 Fresh food sections

Hypermarkets *Continente* can be divided by 3 types of offering: fast-moving consumer goods, fresh food and non food. Representing approximately 25% of *Continente*'s hypermarkets sales area, fresh food involves the following sections: bakery, butchery, cafeteria, delicatessen, fish market, fruits and vegetables and take-away. Each one of them possesses a service balcony to provide specialized customer service.

In recent years, fresh food has become a focus to Sonae MC due to being a distinction point between supermarket and hypermarket chains and due to the urge of improving the costumer perception of the fresh food area in comparison with fresh food from local marketplaces. Also, the increased awareness of consuming healthy food is changing people's eating habits and there is a constant competitive need of following costumers' behavior changes and preferences.

3.3 Problem identification

Sonae MC already has a workforce management software, which roll out occurred in the beginning of 2018. It is a user-friendly software that provides a forecast of needs, work schedules optimization respecting corporate policy and legislation, reports and KPI (Key Performance Indicator) dashboards. However, its full capacity is only implemented on checkouts sections. Currently, there is an ongoing project to extend it to all fresh food sections. Within the scope of this extension, the project *Call for Solutions - Predicted models for in-store workforce optimization* arised being an independent project but having some complementary role on the ongoing project.

Accordingly with Sonae's policy, schedules must contain the work timetable of an entire month and be delivered to employees at least 1 month in advance. Also, they are managed at time intervals of 15 minutes. That being said, it means that a prevision of a least 2 months in periods of 15 minutes is needed.

Each one of the fresh food sections has specific operation tasks. On opening hours there are two main tasks that are common to almost all of them (Table 1): replenishment from backroom to sales floor shelves and customer service at the service balcony. Specially in the concern of the fresh food, replenishment is a constant needed task for avoiding having this perishable food in the backroom where the costumers can't buy it. Both tasks have a significant importance on sales and costumer satisfaction, occupying most of the employees' workforce. Before the opening hours, making sure the levels of staff are filled is vital to prepare the shelves and balconies before the opening hour to make the store and products look appealing to the costumers. After the close hour, the sections and the backroom must be cleaned and organized, allowing the next day to start fluently and lacking delayed tasks.

fresh food section	service balcony	shelves replenishment
bakery	yes	yes
butchery	yes	yes
cafeteria	yes	no
delicatessen	yes	yes
fish	yes	yes
fruits and vegetables	no	yes
take-away	yes	yes

Table 1: Major tasks for each fresh food section of hypermarkets *Continente*.

4 Methodology

To answer the proposed problem a set of algorithms were constructed. Having access to some data, together they output a timetable 11 weeks long with time intervals of 15 minutes with the predicted optimal number of employees for each section, for each hypermarket. We separated the problem of calculating the optimal staff level into two groups - staff level on store closing hours and staff level on store opening hours - since we used a substantial distinct approaching method. The algorithms were written in Python.

4.1 Data collection

On the first two weeks, an Onboarding plan allowed me to visit some stores and talk to employees of each function about their day-to-day tasks. This was very helpful since I could see their tasks being done which allowed me to have a deeper insight of the problem. After that I could explore which data was available and explore it. I had access to Hue (Hadoop User Experience) and Microsoft SQL Server were the data were storage. On both platforms, visualizing the data must be done by queries using SQL (Structured Query Language). In sum, the available data was the following: sales data (by currency and number of items), stock per day, schedules of employees, number of items per pallet, and data of the balconies service ticketing system.

Unfortunately, I did not have access to promotional data but the majority of promotions cannot be known 3 months in advance anyway. However, it could help to know its impact to possibly make posterior changes on the prevision after having knowledge of a future promotion.

4.2 Sales forecast model

The sales are counted at the POS (Point of Sale) when the purchase is finished. This data will be used as base of this methodology since the sales are highly correlated with the amount of work to be done by the staff (as referred in 2.2.1). Store traffic could bring more information since sales can be lost or delayed by deficient costumer service, queues or stock-outs, however there is no data related to the store's traffic.

We have information about the date and the 15 time interval when the purchase was done, the section each item belongs, the number of items and how much it costs. We choose to only take into account the data since 2016. The variable to predict will be number of items per section per 15 minutes. Number of items is an unit of measure more appropriated than the cost because the price can be affected by the inflation rate, promotions, etc. This prediction will be made separately for each section of each hypermarket since they can have different behaviours.

For the preprocessing of the data we zeroed the sales that were off the store opening store. This can be caused when the employees are testing if the product is being recognized by the POS. Then, we cap the sales that were over the 99 percentile to mitigate some extreme rare values.

We will try different models in order to choose which is more suitable for our data. To evaluate the models, we choose to leave out the last 12 weeks of the data and only train (optimize the parameters) with the remaining data. This last 12 weeks will be used to test the performance of our models' forecast and compare them.

First, we tried to use the ARIMA model giving the date-time variable and the number of items as the time series input. However, this model was not giving good predictive results, was very time consuming and we couldn't use a lot of data to train because it would run out of memory.

With that being said, we moved forward and tried some machine learning models since it can handle a lot of variables and samples with a relatively good computational time performance. With this type of models we can easily add or experiment different features and evaluate their impact on the predictive results.

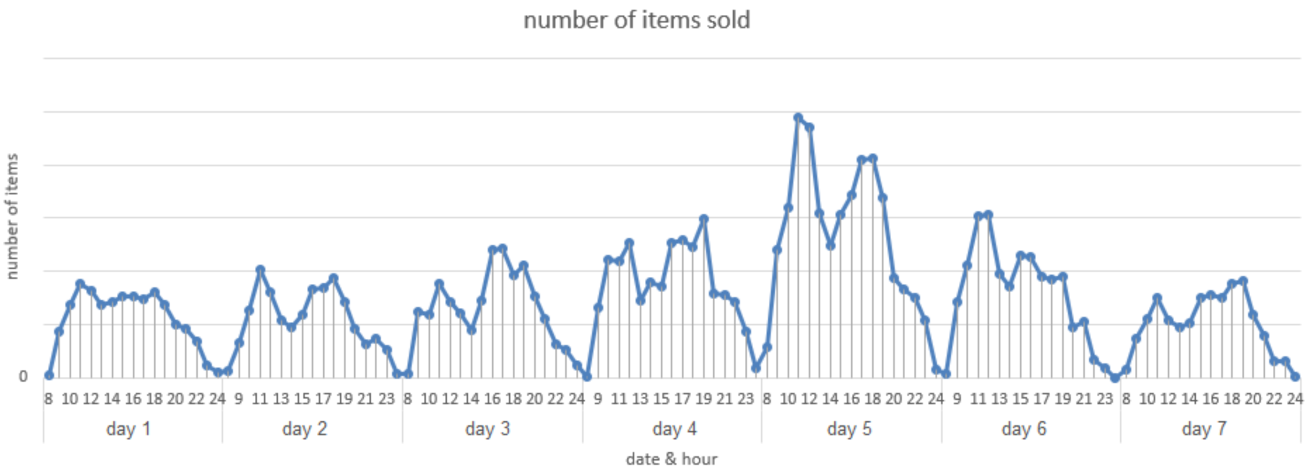
The date-time feature cannot be used in its raw format due to this type of models don't handle date or time variables. Instead, we must create time related feature from the date-time information. Therefore, we plot the sales of some sections to understand the behaviour of the sales throughout time. They have a strong daily seasonality, with two peaks of sales per day (see Figure 9a); a strong weekly seasonality with an increasing number of sales on Saturdays (see Figure 9b); and, some sections, an yearly seasonality (see Figure 9c). However, it does not seem to exist a trend throughout the years.
Therefore, some categorical features were created to help the model capture these seasonalities: hour of the day, day of the week, number of the week, month and season. The categorical for year was also created having in mind the possibility of an increased/decreased number of sales on certain years. Not every machine learning model can handle categorical features so we expand them to binary features where ρ categories result in $\rho - 1$ binary features. This process is known as one-hot-encoding. We also add two more binary features: holiday and holiday's eve.

Numerical features were also created out of the date-time and number of items sold information. The creation of this features had into account not letting any information of the train subset to leak into the test subset. They are the following:

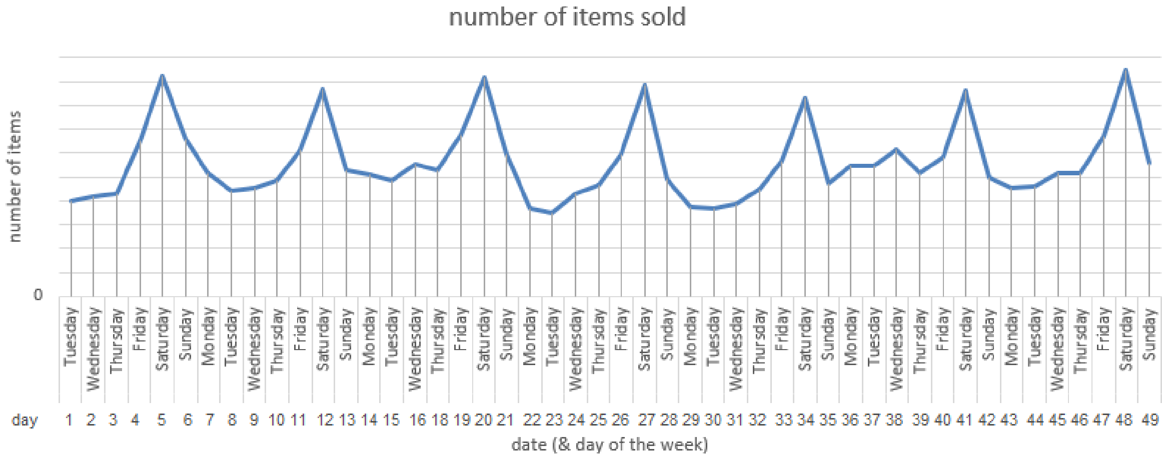
- value of the number of items sold:
  - of the corresponding partition of same day from the previous year,
  - of the corresponding partition of the same day of the week from the previous year;
- mean value of the number of items sold:
  - per partition,
  - per hour,
  - per day of the week,
  - per partition per number of the week;
- maximum value of the number of items sold:
  - per partition,
  - per day of the week,
  - per month,
  - per partition per day of the week;
- mean value of the daily sum of the number of items per day of the week per month.

Given that this forecast model has to fit different sections of different hypermarkets, we cannot expect that the variables will have the same importance for every section/hyper. Therefore, we choose to only try machine learning models which have regularization. This way the model will be able to support different high correlations between the variables and we don't have to choose between variables where one can have a bigger importance on certain section/hyper and a minor importance on other section/hyper.

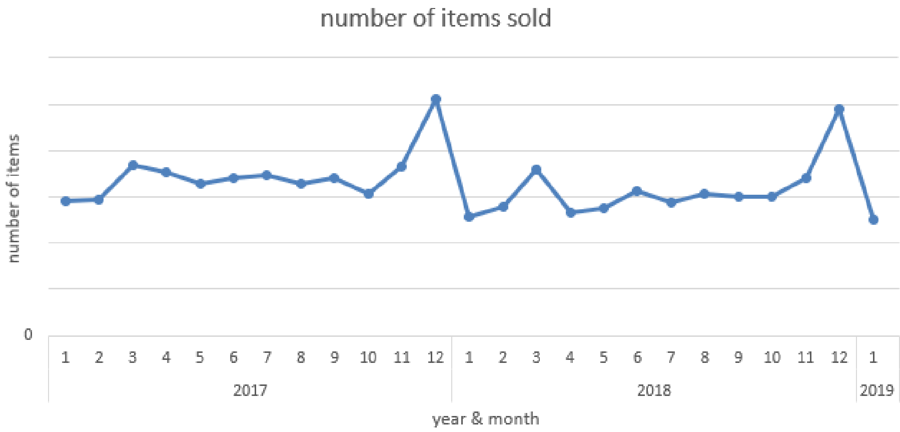
Most models require the range of values between features to be similar to prevent some variables superimpose others; specially models with regularization, so that the penalization can be fair. That said,



(a) Number of items sold per hour to visualize the daily seasonality.



(b) Number of items sold per day to visualize the weekly seasonality.



(c) Number of items sold per month to visualize the annual seasonality.

Figure 9: Example of the number of items sold on a section of a hypermarket with different time aggregations to visualize the seasonality. Dates and number of items sold were omitted, proportions were kept.

we standardize the train subset. So for each feature (except the one we want to predict), every x_i of the N observations becomes:

$$z_i = \frac{x_i - \overline{x}}{s},$$

where mean $\overline{x} = \frac{1}{N}\sum_{i=1}^{N}(x_i)$ and standard deviation $s = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2}$. Then, we apply the same standardization values to standardize the test subset.

After that, the features of the train set are ready to be given to a model in order to fit its parameters that minimize its loss function. We will tune and evaluate the following models: Ridge Regression, LASSO Regression, ElasticNet and Gradient Boosting Trees (the Python libraries used for this task were *sklearn* and *xgboost*). In order to tune the models' hyperparameters, a gridsearch with cross-validation was performed. The metrics used were Mean Squared Error (MSE) - with the predictions grouped by partition - and symetric Mean Average Percentage Error (sMAPE) - with the predictions grouped by day, with major weight on MSE. We choose to add a metric aggregated by day to not allow an accentuated bias. MSE is formulated as seen in equation 1, as sMAPE is formulated as follows:

$$sMAPE = \frac{100\%}{n}\sum_{i=1}^{n}\frac{|\hat{Y}_i - Y_i|}{(|\hat{Y}_i| + |Y_i|)/2}.$$

After some testing, the features mean value per partition, maximum value per partition and mean value per hour were dropped since they were not allowing predictions to reach values too further away from its averages.

Finally, we build the data features with the whole dataset (both train and test set), estimate the parameters of the previous chosen model by training it again but this time with the whole dataset, and use this final model to make the sales forecast for the next 3 months.

4.3 Conversion into number of employees on store's open hours

On opening hours, the prediction of the number of items the section will sell will be the driven variable to the workforce necessity. Replenishment and service at the section's balcony are the major activities on opening hours. This methodology will focus on those. Minor activities or tasks which cannot be predicted in advanced such as inventory count, food security (audit, inspection and evaluation) and supply chain tasks will not be taken into account in this project.

4.3.1 Service balcony assignment

Balcony service is an important task where the employee directly interacts with the costumer to fulfill their requests. Having the prevision of number of items that will be sold, to convert them to number of employees we had to calculate some key values and assumed some simplifications. Percentage of balcony service items costumers bought, service time per costumer and number of balcony service items bought per costumer were the key values to be found per section. Every section and every hypermarket has different structures and particularities so the values must be calculated per section for each hypermarket.

Most hypermarkets have a ticket dispenser where the costumers take their ticket to a specify section's service. Although there is a ticket service dataset which includes: store, service section, date, serve time and service duration; it was too noisy. The noise can happen due to many reasons. An employee calls and serves the client without changing to the next ticket number, a costumer takes a ticket by mistake, the ticket counting is delayed so an employee quickly calls many tickets to reach a certain ticket number. However, an average service time per section was calculated. This calculation only took into account serve times where maximum of one employee had their schedule associated with the service task, and service duration between 5 and 480 seconds. The values obtained seemed reasonable but could be more precise if measured by the store's section manager.

In relation to the percentage of items that are sold in the balcony's sector service, we used the detailed dataset of sales by SKU where each SKU had a field which specified if it was a balcony service item or a store shelf (self-service) item. We calculated the mean percentage for the months July, August, September and October 2018 per section. These values do not vary much in the examined months but can be updated from time to time.

To calculate the value of items bought per costumer on balcony service, we first calculated the number of costumers that bought at the very least one balcony service item in the respective section. This was done by counting the number of sale receipts, using once more the samples of the months July, August, September and October 2018. Dividing the number of items sold on balcony service by the number of costumers, we get the number of balcony service items sold per costumer. Just like the percentage value previously calculated, this value does not vary much per month and can be updated.

The average number of items that can be sold in 15 minutes related to the balcony service is given by: 15 times the average number of balcony service items per costumer dividing by the average service time per costumer. We called this value the service balcony productivity.

Using this productivity value we can now easily convert the prevision of items that will be sold into number of employees by taking the following formula: percentage of service balcony items sales times the prevision of number of items that will be sold dividing by the productivity. The result can be a decimal number, so we take its ceiling as the number of employees to fulfill in this task.

In sum, the formula to estimate the optimal number of employees to fulfill on balcony service per section per 15 minutes is the following:

$$\left\lceil \frac{Per_{sb} \times \hat{Y}}{Prod_{sb}} \right\rceil,$$

where $\hat{Y}$ is the prevision of items that will be sold on that section per 15 minutes;

$$Per_{sb} = \frac{1}{lenght(M)} \sum_{m \in M} \frac{Y_{m_{sb}}}{Y_m}$$

with M being the set of months used for the estimation, Y_M is the number of items sold on month m and $Y_{M_{sb}}$ is the number of service balcony items sold on month m;

$$Prod_{sb} = 15 \times \frac{IP}{T}$$

where T is the estimated service time and

$$IP = \frac{1}{lenght(M)} \sum_{m \in M} \frac{Y_{m_{sb}}}{p_m},$$

p_m is the number of different sale's receipts on month m that have a minimum of one service balcony item.

This approach was constructed with the following assumptions:

- The number of items sold represents the potencial sales (due to lack of information of queues ocurrence).
- Items that are sold on balcony service are not sold on self-service shelves with the same SKU (due to impossibility of knowing if the item was dislocated).
- Average percentage of balcony service items remains the same through the day and throughout the days of the month (due to simplification).
- Average number of items bought per costumer on service balcony remains the same through the day and throughout the days of the month (due to simplification).
- The conversion to number of employees is linear (doubling the employees will double the productivity) (due to lack of information of the service time).
- The time interval between the balcony service and the moment of the purchase at the POS is in the same partition of 15 minutes (due to lack of information of the costumer time distribution inside the store).

4.3.2 Replenishment assignment

Replenishment is strongly related with sales and with the items shelf space. However, in contrast to fast-moving consumer goods and non food, fresh food does not have a strict shelf space allocation plan. That said, it does not exist a shelf space dataset of fresh food.

The replenishment from the back room to the shelves is done by transporting the items on pallets. In general, different items can be put in the same pallet matching the sales floor replenishment necessity. This does not happen in the case of fruits and vegetables section since the pallets are transported to the sales floor completely filled by the same item. For each item we have the information of the quantity per pallet each one normally arrives to the store. Although employees can put more or less items per pallet when transporting them to the shelves, that information is a good approximation to how many items a pallet can hold.

On account of not knowing the shelf space given to each item, we established to trigger replenishment every time the number of items sold can fill one pallet. This can be adapted once there is information about the shelves allocation.

Each section has different items, with different sizes, so the quantity needed to fill a pallet can differ per item inside the same section. However, a sales prevision per item would not be accurate on a forecast per 15 minutes for 3 months long. Instead, we calculated an average value of the number of items that fill a pallet per section. This average value is estimated by an weighted average, being the item's sales the weights. If an item has a bigger volume of sales there is more probability of having to replenish it.

So, the average number of pallets we can fill per 15 minutes concerning the sales is given by

$$ANP = \frac{Per_{ss} \times \hat{Y}}{FP},$$

where $\hat{Y}$ is the prevision of items that will be sold on that section per 15 minutes; $Per_{ss} = 1 - Per_{sb}$, which corresponds to the percentage of shelf items sold; and FP is the average number of items a pallet can transport.

Like referred before, we established that replenishment is triggered when the sales can fill an entire pallet. That said we take the ceiling of the average number of pallets (ANP) on the first time lag of 15 minutes and add the decimal number to the next 15 minutes ANP. So on the next 15 minutes, we take the ceiling of the ANP plus the decimal part that came from the previous time lag, and so on. This gives the number of pallets to replenish (NPR) per 15 minutes.

Given an input of the average time to spend on replenishment per pallet ($ATRP$), or an average number of pallets than can be replenished by one employee per each 15 minutes (APR), we can estimate the number of employees to be in a certain section with replenishment as their task by:

$$\left\lceil \frac{NPR}{15/ATPR} \right\rceil \text{ or } \left\lceil \frac{NPR}{APR} \right\rceil.$$

This approach was constructed with the following assumptions:

- The number of items sold represents the potencial sales (due to lack of information of shelf stock-outs).
- Items that are sold on balcony service are not sold on self-service shelves with the same SKU (due to impossibility of knowing if the item was dislocated).
- Average percentage of shelf items remains the same through the day and throughout the days of the month (due to simplification).
- The conversion to number of employees is linear (doubling the employees will double the productivity) (due to lack of information of the replenishment time).
- The time interval between picking an item of the shelf and the moment of the purchase at the POS is in the same partition of 15 minutes (due to lack of information of the costumer time distribution inside the store).

4.4 Conversion into number of employees on store's closed hours

In the case of pre-opening hours we will not divide the workload by tasks due to the lack of detailed information about them. However it will be bearing in mind that some tasks are somewhat sales' dependent - receiving inventory, arranging the inventory for posterior replenish, etc - and others are fixed - cleaning the floor area, checking if shelves are refilled properly, checking electric equipment, etc. Post store opening hours can be defined by minimal requirements given the reduced duration of employees' schedules after store's closing time.

Instead of trying to predict the optimal number of employees per 15 minutes, we pursue a different approach on closing hours. Fixing partitions of time in this case did not make sense since the entrance hour of the first employee to arrive at the store can differ a lot depending on the day, section and store. We assume that each 15 minutes worked per employee on store closing hours has the same workforce and we will call it an *work unity*. For instance, if for an interval of 1 hour before the store opening we have 2 employees on a certain section, it represents 2×4 ($\times 15$ minutes) $= 8$ *work unities*. Therefore, we will try to predict the needed working units per day and then they will be distributed depending on the store's human resource manager inputs such arriving hour, maximal and minimum number of employees per partition of 15 minutes.

Sales and average stock per section will be the drivers to estimate the workforce necessity. Average stock can represent fixed tasks which depends on the section's sales floor and backroom area, such as cleaning up and organizing the section. On the other hand, sales will represent tasks related to the storage of the items that arrive to the store. The number of items that arrive to the store is strongly related

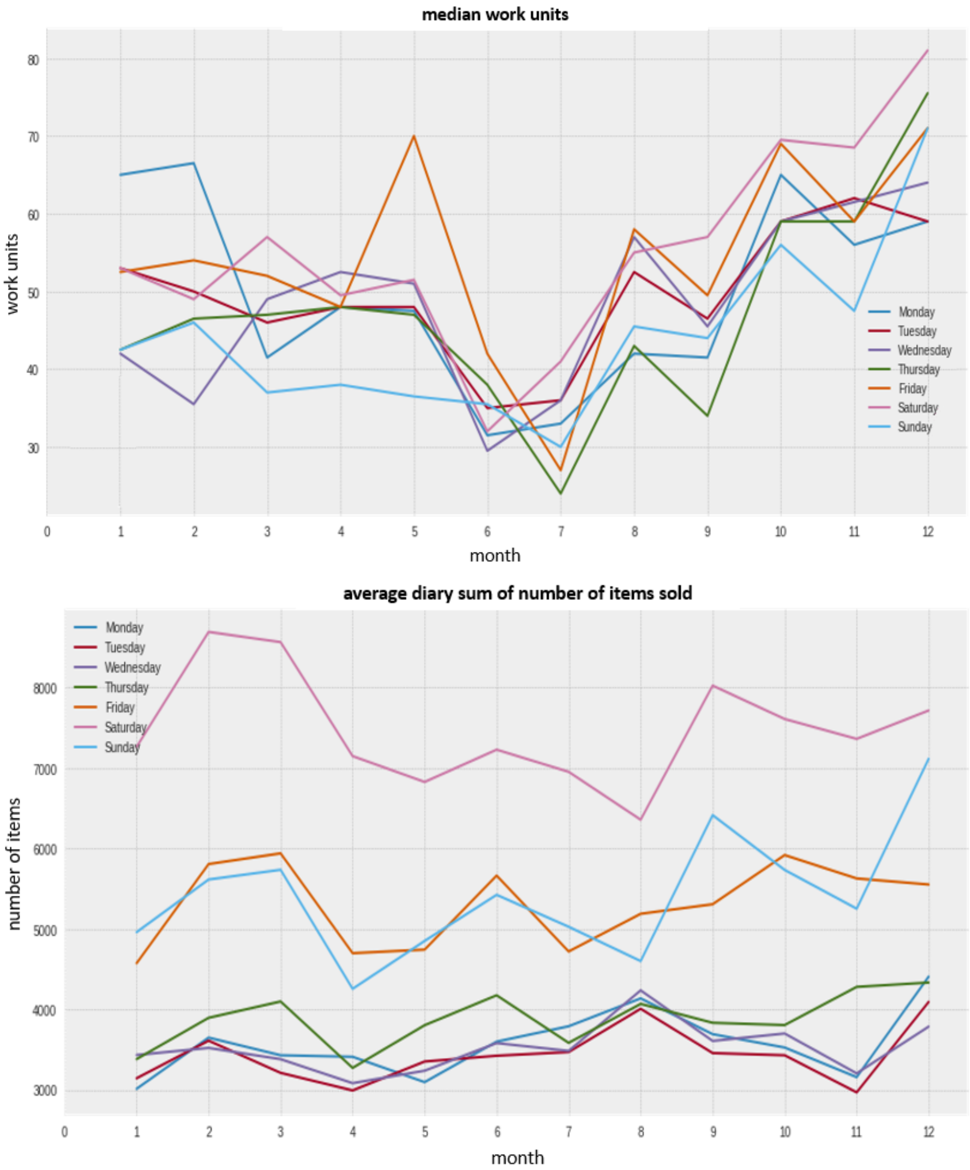


Figure 10: Comparison between the median work units and the mean daily sum sales in 2018 of a sample section from a hypermarket.

to sales in the case of perishable items, which is a characteristic of most part of fresh food. The orders are made with particular care to try to sell most of the items quickly in order to avoid food surplus which can lead to spoilage.

To find the relation between sales and average stock with the workforce, we used the historical data since 2018 of the employees' schedules because it was the only data that gives some information about the staff level requirements on store's closed hours. However, visualizing the median historical work units per day of the week per month we found controversy values comparing to the respective sales on some sections from some stores (see Figure 10).
Schedules can have a lot of oscillations and not reflect the optimal number of employees because they have to respect legal rules and they can be affected by employees' breaks and holidays. Therefore, we decided to estimate the relation by only comparing with the historical data per day of the week. This way, the possible divergences concerning the optimal number of employees are mitigated.

Thus, we decided to use the following formula to calculate the optimal number of work units:

$$\frac{x \times average\, stock + y \times sales\,(per\, days\, of\, the\, week)}{productivity} = work\, units\,(per\, days\, of\, the\, week),$$

where x is a coefficient associated with the workforce coming from the tasks represented by the average stock and y is a coefficient associated with the consequent tasks of the sales' volume. We have to estimate x, y and the $productivity$. Concerning x and y, we will estimate each one for each section (independently of the hypermarket) since the influence of the average stock and the sales on the amount of work must be similar between different hypermarkets, and can be different between sections. On the other hand, the $productivity$ will have a different value for each section of each hypermarket, since the store's logistics and structure may influence the productivity.

In order to estimate x, y and $productivity$ we use the historical data as follows:

$$\frac{x \times average\, stock + y \times sales\,(per\, days\, of\, the\, week)}{median\, work\, units\,(per\, days\, of\, the\, week)} = productivity\,(per\, days\, of\, the\, week).$$

The idea is to minimize the divergence between the productivity of the different days of the week, because if the coefficients are expressing the workforce well the productivity between the different days should not vary too much. Thus, the objective is to minimize the coefficient of variation between the productivity per day of the week. The final productivity will be the average of the values of the productivity per day of the week.

We used 5 hypermarkets as sample for the estimation of this parameters since the extraction of the required data was not automated and, consequently, was too time consuming. To achieve better results a greater number of samples would be preferred.

Then, for H hypermarkets as sample, the objective function to optimize per section is:

$$\sum_{h=1}^{H} std(\{P_{h_{dw}},\, dw \in DW\})/average(\{P_{h_{dw}},\, dw \in DW\}),$$

where $DW =$ {Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday},
and has the following constraints:

- $\{(x \times STOCK_h + y \times SALES_{h_{dw}})/EMP_{h_{dw}} - P_{h_{dw}} = 0,\, dw \in DW,\, h \in [1, ..., H]\}$
- $x \geq 0$
- $y \geq 0$
- $\{P_{h_{dw}} > 0,\, dw \in DW,\, h \in [1, ..., H]\}$,

where $STOCK$ is the average stock (value calculated by the daily stock from the previous 3 months), $SALES_{dw}$ is the average daily sales per days of the week dw, EMP_{dw} is the median work units per day of the week dw, and P is a productivity value.

The algorithm used to optimize was Sequential Least SQuares Programming (SLSQP). Given this optimization we obtain the values x, y and $\{P_{h_{dw}},\, dw \in DW,\, h \in [1, ..., H]\}$. Now, we can calculate a productivity per hypermarket (per section):

$$P_h = average(\{P_{h_{dw}},\, dw \in DW\}).$$

Finally, we have all the values needed to our goal function. However, the results obtained are related to days of the week. It can be extended by days of the week per month, by using the sales per days of the week per month, instead of only sales per days of the week, and maintaining the other estimated values. This would give the following formula:

$$\frac{x \times average\, stock + y \times sales\, (per\, days\, of\, the\, week\, per\, month)}{productivity} =$$

$$= work\, units\, (per\, days\, of\, the\, week\, per\, month).$$

It could also be extended to give the work units per day, following the same principle. This calculations would be a good approximation of the optimal number of work units to fulfill if the historical employees' schedules per days of the week actually corresponded to the workforce needed. Unfortunately, like referred before, the schedules are affected by many employee schedule's rules it has to comply, and not only by the real workforce needs. Given the lack of data and not existing a way to evaluate these results, the resulting work units in this process can only be a guideline to the actual workforce needed and has to be constant monitored by the store's section manager or by the store's human resource manager.

After having the predicted work units , we created an algorithm to distribute them by employees and by the partitions of 15 minutes. Having the arriving hour, maximal and minimum number of employees per partition of 15 minutes as input, the algorithm distributes the work units in the following way:

1. Fills the partitions of 15 minutes starting at the arriving hour until the store's opening hour with the minimum number of employees.
   – Warning message if after this the number of work units is exceeded.
2. If there are remaining work units, fills the closest partition of 15 minutes to the store's opening hour up to the maximal number of employees while the number of work units is not exceeded.
3. If there are remaining work units, fills the previous partition of 15 minutes in the same way of point 2, and so on, until the last partition is reached.
   – Warning message if after this the number of work units is not fulfilled.
   – An exception is made and the maximal number is increased by one unit.
   – If the number of units is still not fulfilled, the maximal number of employees is increased one more unit.
   – Warning message if after increasing the maximal number of employees 2 times, the number of work units was still not fulfilled.

5 Results and Discussion

We will show the results for all sections of two random hypermarkets. To protect the confidentiality of the information we will name them as hypermarket A and B. We did not have an automatic process to gather all the data needed so it was not possible to present the results for all the hypermarkets since this was considerable time consuming.

To evaluate the final forecast model for each section we divided the data 3 times like shown in Figure 11, except for sections where the samples start only after 2018, in this case it is only divided 1 time.

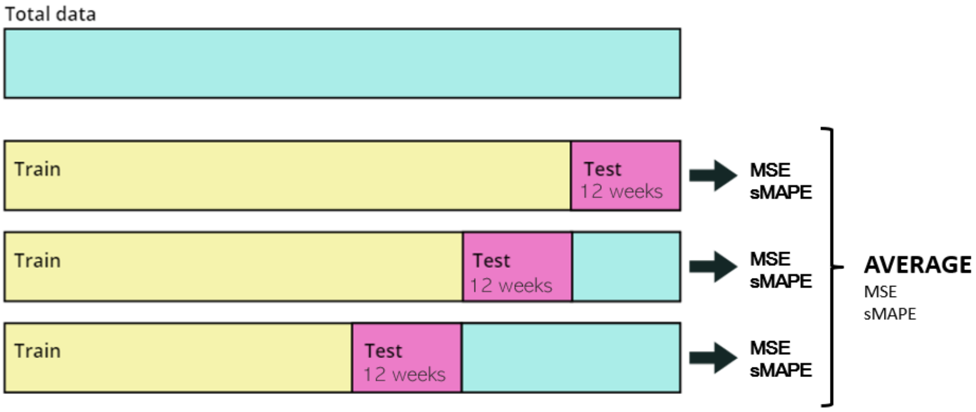


Figure 11: Evaluation and division of the data for the sales forecast model per section.

The results for the hypermarket A are presented in the following table 2.

Section	**Best model**	**MSE (per partition)**	**sMAPE (group by day)**
Butchery	Gradient Tree Boosting	319.53	10.53 %
Fish	Gradient Tree Boosting	242.58	15.74 %
Delicatessen	Gradient Tree Boosting	560.05	8.81 %
Fruits and Vegetables	Gradient Tree Boosting	2113.51	14.31 %
Bakery	Gradient Tree Boosting	2215.88	8.71 %
Cafeteria	Gradient Tree Boosting	178.90	26.37 %
Take-Away	Gradient Tree Boosting	107.11	16.44 %

Table 2: Evaluation results for all the fresh food section in hypermarket A.

MSE is not a good metric to interpret and compare the performance of the models for each section given the scale of sales can differ a lot. For instance, Cafeteria which has one of the lowest MSE scores, does not present good results - we can also see that the sMAPE value is really high. However, the data of this hypermarket related to Cafeteria seems to have an anomaly as we can see on Figure 12, maybe an error occurred while mapping the correct items on some months.

To better interpret the forecast results, we compare the absolute difference of the number of employees after our conversion method of the sales real values and the sales predicted values into number of employees (see 3). For comparing purposes we needed a replenishment time for each section with that task, so we chose a number that minimizes the difference between the schedule data and the predicted of the maximal and average number of employees.
In relation to the Hypermarket B the forecast results are presented in tables 4 and 5.
Gradient Boosting Trees was overall the model with better results. It was able to capture the non-linearity

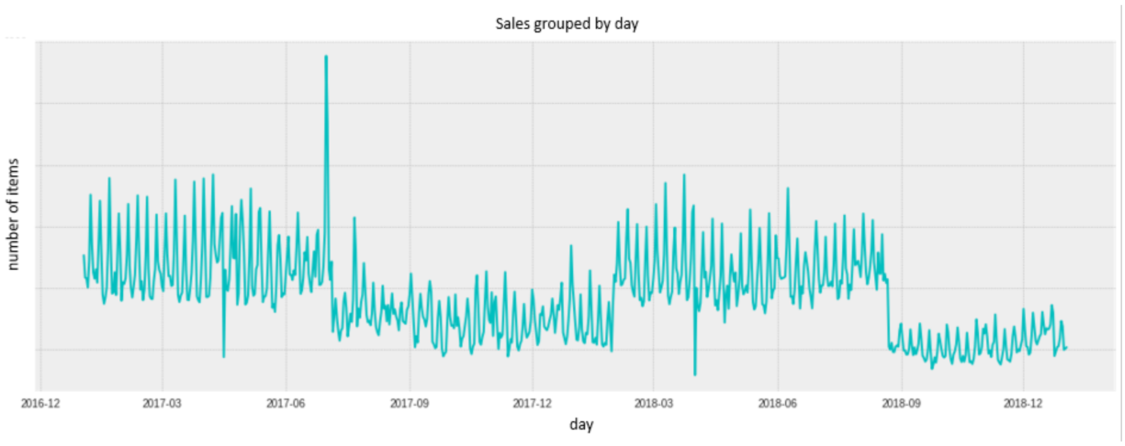


Figure 12: Sales grouped by day from Cafeteria of Hypermarket A.

Section	Task	% difference equals zero	% difference greater than 1
Butchery	Service Balcony	56.7	2.6
	Replenishment	61.9	2.2
Fish	Service Balcony	47.2	7.9
	Replenishment	61.6	1.2
Delicatessen	Service Balcony	65.3	1.0
	Replenishment	66.7	1.0
Fruits and Vegetables	Replenishment	49.3	8.6
Bakery	Service Balcony	57.4	1.7
	Replenishment	78.3	0.0
Cafeteria	Service Balcony	42.1	10.2
Take-Away	Service Balcony	68.6	0.0
	Replenishment	86.1	0.0

Table 3: Percentage of partitions of 15 minutes in relation to the differences between the conversion into number of employees by using the real sales values and the predicted sales values - Hypermarket A.

Section	Best model	MSE (per partition)	sMAPE (group by day)
Butchery	Gradient Tree Boosting	202.30	10.73 %
Fish	Gradient Tree Boosting	127.95	12.03 %
Delicatessen	Gradient Tree Boosting	395.36	7.09 %
Fruits and Vegetables	Gradient Tree Boosting	1550.10	12.73 %
Bakery	Gradient Tree Boosting	2395.22	8.20 %
Take-Away	Elastic Net	309.62	9.80 %

Table 4: Evaluation results for all the fresh food section in hypermarket B.

of the sales data unlike the LASSO, Ridge and ElasticNet regressions. Although the high volatility of the data the model could capture the general sales behaviour as we can see in Figure 13.

The final output consisted in presenting the estimated optimal number of employees for each section, in every 15 minutes, for every day of the 3 months required, separated by: before opening hours, replenishment and service balcony. We can see an example of the output for 1 sample day in Figure 14.

Section	Task	% difference equals zero	% difference greater than 1
Butchery	Service Balcony	58.0	2.3
	Replenishment	64.5	1.9
Fish	Service Balcony	55.5	3.8
	Replenishment	61.7	1.2
Delicatessen	Service Balcony	71.1	0.3
	Replenishment	65.8	0.8
Fruits and Vegetables	Replenishment	51.4	5.7
Bakery	Service Balcony	55.2	2.7
	Replenishment	80.9	0.0
Take-Away	Service Balcony	50.3	6.9
	Replenishment	86.2	0.1

Table 5: Percentage of partitions of 15 minutes in relation to the differences between the conversion into number of employees by using the real sales values and the predicted sales values - Hypermarket B.

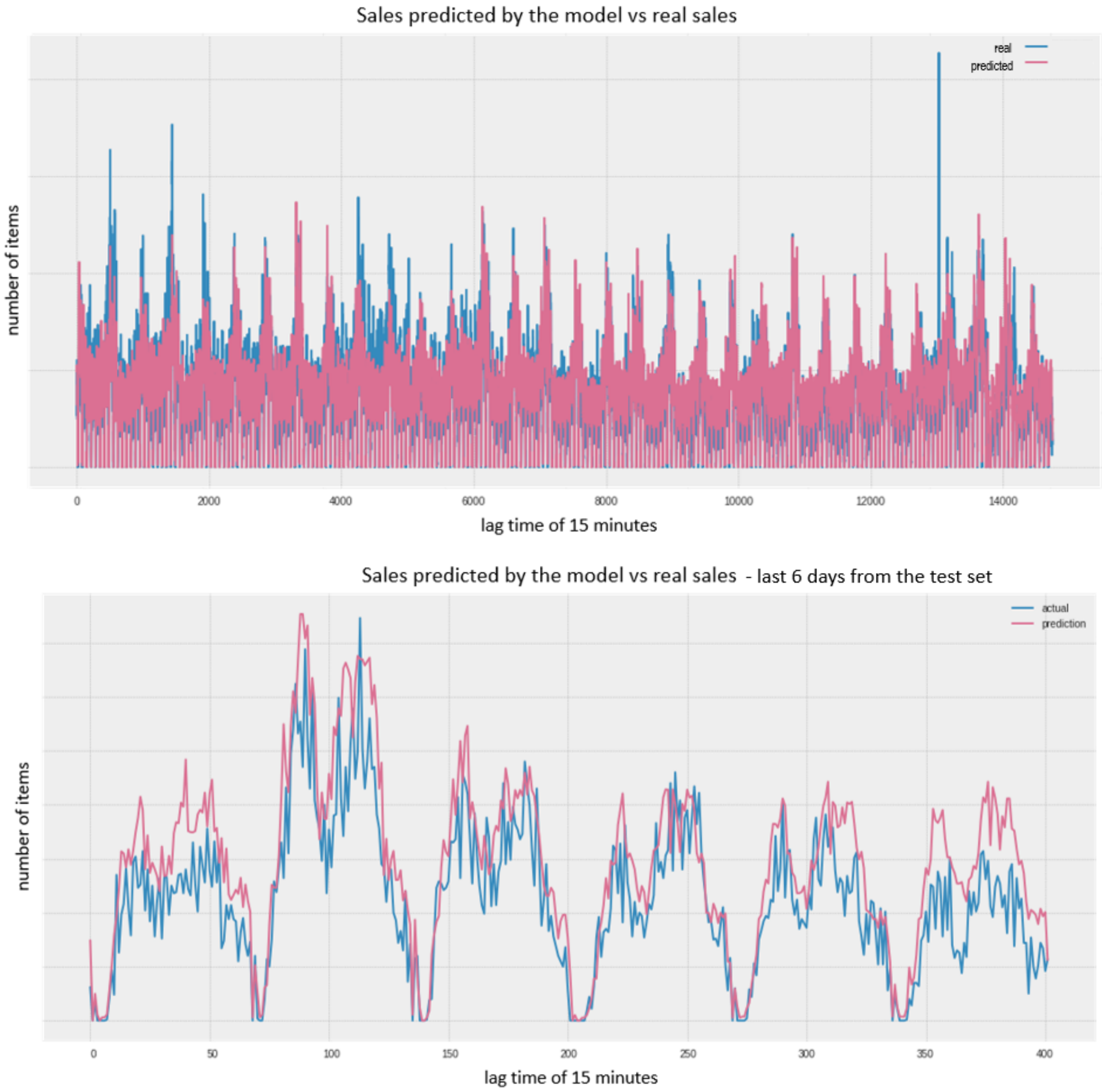


Figure 13: Forecast model sales predictions versus sales real values of hypermarket A Fruits and Vegetables.

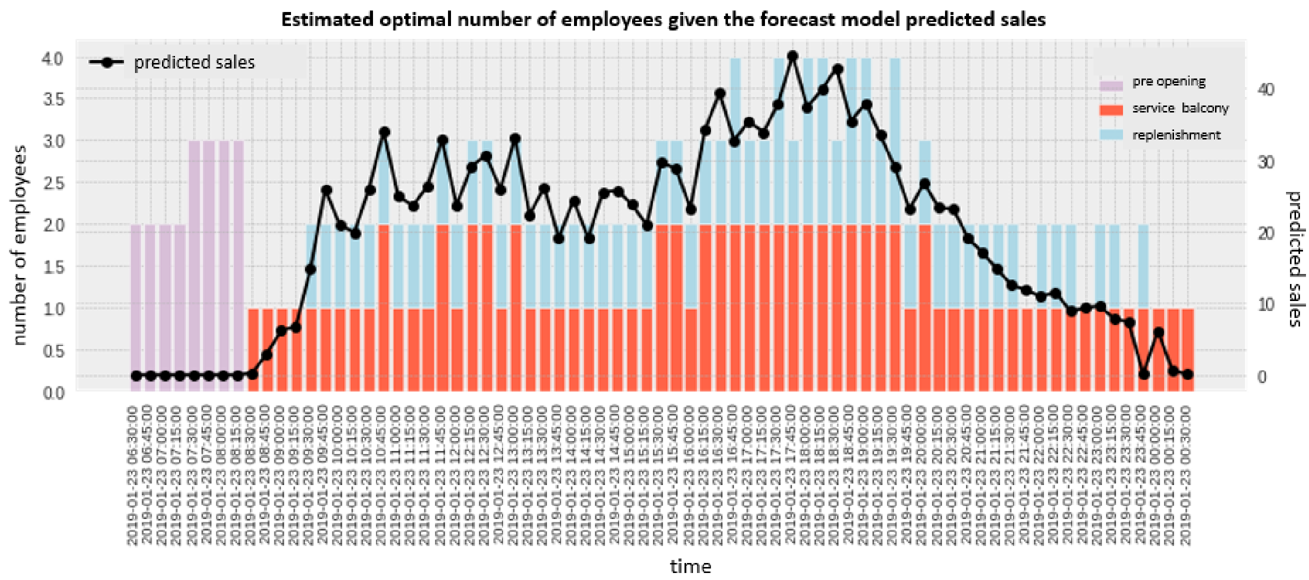


Figure 14: Estimated optimal number of employees for a certain section.

6 Conclusions and Future Work

This method allows the possibility of a continuous improvement and discussion for being a bottom-up approach of this problem. However, a set of things could be worked up to fortify this method. The use of store traffic information instead of sales could be a better driven for the required workforce since the unknown lost sales due to queues or stock-out products could not have been taken into account in this model. For the balcony service task, more data in relation to the times costumers takes the ticket, is assisted and concludes the assistance, would allow the study of a queue simulation. Information of replenishment tasks would help to understand if the productivity and the number of employees has a linear behaviour or not. Overall, with more data this method could be continuously improved.

For future work, besides the improvements we could do with the expansion of the available data, an implementation of this method could be done by creating an iterative platform so the human resource managers or the section's chief could work with.

References

Bielen, Frédéric and Nathalie Demoulin (2007). “Waiting time influence on the satisfaction-loyalty relationship in services”. In: *Managing Service Quality: An International Journal* 17.2, pp. 174–193.

Bontempi, Gianluca, Souhaib Ben Taieb, and Yann-Aël Le Borgne (2012). “Machine learning strategies for time series forecasting”. In: *European business intelligence summer school*. Springer, pp. 62–77.

Box, George EP and Gwilym M Jenkins (1976). “Time series analysis, control, and forecasting”. In: *San Francisco, CA: Holden Day* 3226.3228, p. 10.

Box, George EP, Gwilym M Jenkins, et al. (2015). *Time series analysis: forecasting and control*. John Wiley & Sons.

Brockwell, Peter J, Richard A Davis, and Stephen E Fienberg (1991). *Time Series: Theory and Methods*. Springer Science & Business Media.

Chen, Tianqi and Carlos Guestrin (2016). “Xgboost: A scalable tree boosting system”. In: *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. ACM, pp. 785–794.

Daniel Läubli, Gernot Schlögl and Patrik Silén (2015). “Smarter schedules, better budgets: How to improve store operations”. In:

Fisher, Marshall and Ananth Raman (2010). “The new science of retailing”. In: *Harvard Business School Press, Boston*.

Fonti, Valeria and Eduard Belitser (2017). “Feature selection using lasso”. In: *VU Amsterdam Research Paper in Business Analytics*.

Grubor, Aleksandar et al. (2017). “Shelf based out-of-stocks in the context of employee density”. In: *Engineering Economics* 28.4, pp. 446–454.

Günther, Maik and Volker Nissen (2010). “Combined working time model generation and personnel scheduling”. In: *Advanced Manufacturing and Sustainable Logistics*. Springer, pp. 210–221.

Hyndman, Rob J and George Athanasopoulos (2018). *Forecasting: principles and practice*. OTexts.

Infor (2007). *Labor Planning and Budgeting for Retail Workforce Agility*. Tech. rep. Infor Global Solutions GmbH and/or its affiliates and subsidiaries. URL: http://www.workbrain.com.

Masum, Shamsul, Ying Liu, and John Chiverton (2018). “Multi-step time series forecasting of electric load using machine learning models”. In: *International Conference on Artificial Intelligence and Soft Computing*. Springer, pp. 148–159.

McKinnon, Alan C, Daniela Mendes, and M Nababteh (2007). “In-store logistics: an analysis of on-shelf availability and stockout responses for three product groups”. In: *International Journal of Logistics Research and Applications* 10.3, pp. 251–268.

McLean, Susan et al. (2016). “BIG data and human resources: letting the computer decide”. In: *Scitech Lawyer* 12.2, p. 20.

Rostami-Tabar, B et al. (2013). “Forecasting aggregate ARMA (1, 1) demands: Theoretical analysis of top-down versus bottom-up”. In: *Industrial Engineering and Systems Management (IESM), Proceedings of 2013 International Conference on*. IEEE, pp. 1–8.

Saleh, A.K.Md.Ehsanes, M Arashi, and B M Golam Kibria (2019). “Introduction to Ridge Regression”. In: pp. 1–13. DOI: 10.1002/9781118644478.ch1.

Sauer, Jürgen and René Schumann (2007). “Modelling and solving workforce scheduling problems”. In: *Proc. of PUK*, pp. 93–101.

Sloot, Laurens, Peter Verhoef, and Philip Hans Franses (2005). “The Impact of Brand Equity and the Hedonic Level of Products on Consumer Stock-Out Reactions”. In: *Journal of Retailing* 81, pp. 15–34. DOI: 10.1016/j.jretai.2005.01.001.

Tyralis, Hristos and Georgia Papacharalampous (2017). “Variable selection in time series forecasting using random forests”. In: *Algorithms* 10.4, p. 114.

WorkJam (2015). *An Inside Look at the Hiring and Scheduling Crisis in the Hourly Workforce*. WorkJam. URL: https://www.workjam.com.

Zotteri, Giulio, Matteo Kalchschmidt, and Federico Caniato (2005). “The impact of aggregation level on forecasting performance”. In: *International Journal of Production Economics* 93, pp. 479–491.

Zou, Hui and Trevor Hastie (2005). “Regularization and variable selection via the elastic net”. In: *Journal of the royal statistical society: series B (statistical methodology)* 67.2, pp. 301–320.

MSc

O Presidente do Júri,

Porto,

______/______/________

Predictive models for in-store workforce optimization

Viviana de Oliveira Dias

Engenharia Matemática

Departamento de Matemática

2019

M2CMSC

Predictive models for in-store workforce optimization

Viviana de Oliveira Dias
Engenharia Matemática
Departamento de Matemática

U.PORTO
FACULDADE DE CIÊNCIAS
UNIVERSIDADE DO PORTO

MSc

MSc

2º
CICLO

FCUP
BIT
2019

U.PORTO

Predictive models for in-store workforce optimization

Viviana de Oliveira Dias

FC

U.PORTO
FACULDADE DE CIÊNCIAS
UNIVERSIDADE DO PORTO

bit BY
SONAE

Predictive models for in-store workforce optimization

Viviana de Oliveira Dias

Relatório de Estágio de Mestrado apresentado à
Faculdade de Ciências da Universidade do Porto e Business
Information and Technology Sonae MC em
Matemática

2019

MSc