A RETAIL DEMAND FORECASTING SYSTEM OF PRODUCT GROUPS CHARACTERIZED BY TIME SERIES BASED ON "ENSEMBLE MACHINE LEARNING" TECHNIQUES WITH FEATURE ENGINEERING

Santiago Mejía Chitiva[1] **and Jose Lisandro Aguilar Castro**[1, 2, 3]

[1]GIDITIC, Universidad EAFIT, Medellín, Colombia
[2]Dpto de Automática, Universidad de Alcalá, España
[3]CEMISID, Universidad de Los Andes, Mérida, Venezuela

smejiac3@eafit.edu.co, jlaguilarc@eafit.edu.co

Abstract

Retail companies face major problems in the estimation of their product's future demand due to the high diversity of sales behavior that each good presents. Different forecasting models are implemented by this sector to meet the demands requirements for an efficient inventory management. However, in most of the proposed works, a single model approach is applied to forecast all products, ignoring that some methods are better adapted for certain characteristics of the demand time series of each product, and consequently, generating inefficiencies in the supply chain. The proposed forecasting system address this problem, by implementing a two-phase methodology that initially groups the products with the application of an unsupervised learning approach using the extracted demand characteristics of each good, and then, implements a second phase where, after a feature engineering process, a set of different forecasting methods are evaluated to identify those that best performs for each group. Finally, ensemble machine learning models are implemented using the top-performing models of each group to carry out the demand estimation. The results indicate that the proposed forecasting system improves the demand estimation over the single forecasting approaches when evaluating the R2, MSE and MASE quality measures.

Keywords: Demand forecasting; machine learning, feature engineering; time series; ensemble learning methods.

1 INTRODUCTION

The responsible people for supply planning in retail companies deal with big challenges due to the large number of products that must be monitored to provide a timely and efficient inventory availability. Therefore, demand forecasting plays a central role in a successful supply chain plan and the service levels offered to customers. Most of the proposals to estimate demand are made from a single forecast model, ignoring the different consumption behaviors of the offered products. Therefore, this work proposes a forecasting system for product groups based on ensemble machine learning techniques, integrated with a feature engineering process for time series, which adapts to the different consumption behaviors experienced by each group of products. In this way, the forecasting system will autonomously consider several machine learning techniques and ensemble models for a demand estimation for groups of products.

1.1 Problem Statement

Retail companies are characterized by the high variety of products that they offer. This situation forces these companies to develop complex structures and processes that depend on the supply plans based on the demand forecasts. Due to the aforementioned characteristics, any forecast error can be detrimental to this sector, so it must be as accurate as possible (Ivanov, Tsipoulanidis, & Schönberger, 2017) (Moon, 2013). Forecasts considerably higher than actual demand will impact higher capital flow and inventory management costs, while lower forecasts will lead to lower service levels (Ivanov, Tsipoulanidis, & Schönberger, 2017).

On the other hand, articles/products show very different supply times and consumption behaviors. Thus, the daily forecast of the sale of each product becomes a challenging task, characterized by different attributes defined in its time series (Spiliotis, Makridakis, Semenoglou, & Assimakopoulos, 2020). Then again, the categorization of demand patterns facilitates the definition of forecasting methods for product groups. However, very limited attention has been carried out in the literature (Syntetos, Boylan, & Croston, 2004) (Nikolopoulos, 2021). In general, there is a large number of articles that have used different machine learning approaches to forecast demand (Kiefer, Grimm, Bauer, & Van Dinther, 2021), but a few studies have been focused on evaluating the model quality considering groups of products according to the demand characteristics evidenced in the time series. Consequently, the time series of product groups have not been sufficiently investigated in the literature (Nikolopoulos, 2021), nor the use of ensemble learning techniques in this context (Spiliotis, Makridakis, Semenoglou, & Assimakopoulos, 2020).

1.2 Justification

Estimating demand is of critical importance for an effective management of retail companies. This, due to the great diversity of goods that must be supplied on time, responding to the different behavior of the demand of each product. Thus, effective forecasting models adapted to the particularities of the different demand behaviors can represent a competitive advantage in supporting decision-making (Ivanov, Tsipoulanidis, & Schönberger, 2017). This is particularly relevant in retail companies where improvements in inventory management have a significant impact on customer service levels (Syntetos, Boylan, & Croston, 2004).

This paper defines an approach to select the most suitable models to estimate demand according to the characteristics of consumption of the products. Specifically, it evaluates several forecasting models based on machine learning techniques, from a perspective of groups of products, according to the behavior of their demand time series, after a feature engineering process. Particularly, this work seeks to define the most appropriate forecasting models for product categories based on the following points:

- Addressing the demand forecasting problem considering machine learning techniques and a product grouping approach based on a feature engineering process for time series.
- Developing ensemble machine learning models for each generated category.

1.3 Objectives

1.3.1 General objective:

Define a demand forecasting system for the retail sector, from a product grouping approach based on a time series feature engineering process, using ensemble learning techniques.

1.3.2 Specific objectives:

1. Performing an autonomic product grouping approach based on the demand behavior, using a feature engineering process for time series.

2. Implementing forecasting models based on machine learning techniques for each of the previously grouped products.

3. Implementing and evaluating ensemble learning models for an autonomous demand forecasting system by product category.

2 THEORETICAL FRAMEWORK

The theoretical framework is divided into four sections. The first part will highlight the relevance of the feature engineering process for time series. The second part will present the most used statistical and machine learning techniques in time series forecasting. The third part will discuss some successful existing ensemble learning

models for demand estimation. The last part presents the academic articles that have proposed demand forecasting models for time series by product categories.

2.1 Feature engineering in time series

Feature engineering is the process of defining the descriptors that make machine learning algorithms work better. Particularly, in time series, Christ, Braun, Neuffer, & Kempa-Liehr (2018) highlight that the feature engineering process plays an important role during the early phases of data science projects since it is a process that allows extracting and exploring different variables to evaluate its importance in predicting the desired objective.

However, due to the high complexity and variety of time series, feature engineering process becomes complicated (Cáceres & Rodríguez, 2011). That implies that an important part of the development of an adequate forecasting model depends on the set of characteristics associated with the behavior of the time series (Talagala, Hyndman, & Athanasopoulos, 2018). Due to this, some articles have developed proposals for the selection of characteristics in time series in different contexts. For example, Jiménez, Aguilar, Monsalve-Pulido, & Montoya (2021) developed a hybrid audio descriptor extraction scheme, based on principles of statistical measures of central tendency and variability, sound engineering and time series, to automatically select the best descriptors in a given audio context. The results obtained showed that, for sound grouping and classification problems, this approach can select the most relevant characteristics. In consequence, the audio descriptor proposal obtained competitive results with fewer numbers of descriptors and lower computation times compared to other traditional alternatives.

Due to the importance of the feature engineering process in time series, it is relevant to mention that several libraries have been developed to automatically generate different features for time series. Some of the most popular are TS-Features (Hyndman et al, 2019) and TS-fresh (Christ, Braun, Neuffer, & Kempa-Liehr, 2018), which have been used in various articles about time series forecasting such as Hewamalage, Bergmeir, & Bandara (2021), Abolghasemi, Beh, Tarr, & Gerlach (2020), Ma & Fildes (2021), Montero-Manso, Athanasopoulos, Hyndman, & Talagala (2020), Chen & Han (2019) and Theodorou, Wang, Kang, & Spiliotis (2021).

It is relevant to mention that time series can be decomposed into four parts, each of which expresses a particular component. These four components are: The trend, which describes the long term movement; seasonality, which is evidenced in patterns that repeat over a fixed period; cyclical behavior, which is identified in periodic but not seasonal patterns; and the irregularity that comes from non-random sources of variations in the series (Dodge, 2008). Consequently, the objective of a feature engineering process for time series is to automate the creation of the variables that describe the characteristics of the four previously mentioned components. In this way, variables such as the strength of the peaks, the maximums,

the minimums and the entropy, among others, help to identify and describe these components for different scenarios.

2.2 Statistical and machine learning models for demand forecasting

The accelerated growth of forecasting models has led the process of selecting the best model alternative to be an open problem (Petropoulos, Kourentzes, Nikolopoulos, & Siemsen, 2018). The retail sector is affected by that situation since its products tend to present strong seasonal variations (Fildes, Ma, & Kolassac, 2019) and have different characteristics in the demand behavior. Historically, the prediction of seasonal data has received significant research efforts, and in recent decades, many theoretical methods have been developed for its modeling (Ching-Wu & Guoqiang Peter, 2003). Among the most frequently used statistical methods for sales forecasting is Exponential Smoothing and its extensions (S.E. simple, Holt, Pegels and Holt-Winters) (Fildes, Ma, & Kolassac, 2019). The method averages (smoothes) time series data, but unlike a simple moving average, it assigns more weight to recent observations and exponentially decreases the weight of observations over time (Taylor, 2003). Like Exponential Smoothing models, the ARIMA models are recognized as one of the most widely used methods for time series forecasting. This type of model and its variants (SARIMA, ARIMAX, ARMA-GARCH, ARFIMA, among others) are characterized by representing different types of seasonal and non-seasonal stochastic time series, such as pure autoregressive processes (AR), pure moving average (MA) and mixed AR and MA processes (Abolghasemi, Hurley, Eshragh, & Fahimniab, 2020).

On the other hand, in recent years, machine learning methods have been proposed as an alternative to statistical methods for forecasting time series (Makridakis, Hyndman, & Petropoulos, 2020), due to their ability to identify non-linear patterns and the few assumptions that the data must meet for a correct implementation (Kuvulmaz, Usanmaz, & Engin, 2005) (Barker, 2020). The most popular models come from Artificial Neural Networks (ANN). Various types of ANN have been developed to estimate demand in the retail sector, such as Multi-Layered Perceptron - MLP (Gutierrez, Solis, & Mukhopadhyay, 2008), Bayesian Neural Networks - BNN (Liang, 2005) and Long Short Term Memory Neural Network - LSTM, which is considered one of the best methods for time series forecasting (Li, Zhua, Kong, Han, & Zhao, 2019). The above is because the LSTM model differs in its ability to capture non-linear time series patterns in short and long time ranges while considering the inherent characteristics of non-stationary time data.

It is important to mention that models for demand forecasting have mostly focused on neural networks models (Spiliotis, Makridakis, Semenoglou, & Assimakopoulos, 2020). However, it is relevant to point out other works based on other machine learning methods, such as Linear Regression (Hong, Gui, Baran, & Willis, 2010), Random Forest (Kumar & Thenmozhi, 2006) (Mei, He, Harley, Thomas, & Qu, 2014), KNNR (Martínez, Frías, Pérez-Godoy, & Rivera, 2018) (Ban, Zhang, Pang,

Sarrafzadeh, & Inoue, 2013), Light GBM (Deng, Zhao, Wang, & Yu, 2021), among others.

2.3 Ensemble learning models in demand forecasting

In addition to statistical and machine learning techniques, it is important to analyze ensemble learning models that combine different machine learning techniques. Researchers such as Armstrong (2001), Clemen (1989), and Makridakis & Winkler (1983), have empirically demonstrated that these ensemble forecasting models are effective in real-world settings. For example, in the M4 competition held in 2018, ensemble forecasting models were the big winners, occupying the first 12 places in the competition (Atiya, 2020).

In the aforementioned competition, Jaganathan & Prakash (2020) experimented with simple combinations of forecasts (Exponential Smoothing, ARIMA, Temporal HIErarchical Forecasting - THIEF, naive model, TBATS, Theta, among others) with equal weights based on the trimmed mean, the median and mean, to implement ensemble forecasts. Based on the performance of those combinations, the authors decided to use the median as the operator to generate the ensemble forecasts. Similarly, Petropoulos & Svetunkov (2020) proposed a simple combination of univariate models (SCUM - Simple Combination of Univariate Models) from the median of the forecast points and the prediction intervals of four traditional models: Exponential Smoothing (ES), Complex Exponential Smoothing (CES), ARIMA and Dynamic Theta Optimization (DOTM). Their submission finished in sixth place in the M4 Competition and was singled out as the simplest combined model out of the top 5.

Stacking and Blending techniques are also ensemble models that have been implemented in sales forecasting. Stacking uses two levels or layers (level-0 and level-1) to combine the models. In level-0, different models are trained, and subsequently, the prediction of the response variable for each one is performed. These forecasts are used as an input set for the level-1 model. This model, too called the meta-model, is trained, and its prediction is the desired result (Ribeiro & dos-Santos, 2020). Blending follows the same approach as stacking but uses only a holdout (validation) set from the train set to make predictions. In other words, unlike stacking, the predictions are made on the holdout set only. The holdout set and the predictions are used to build a model (level-1), which is run on the test set. Pavlyshenko (2019) used a stacking approach to forecast the sales from “Rossmann Store”, from a public dataset that is available on Kaggle. The level-0 models used in the implementation were Extra Trees Regression, Lasso Regression and a Neural Network Model, while the meta-model was another Lasso Regression. The results in terms of Relative Mean Absolute Error showed that the stacking model error was lower than the three base models.

2.4 Grouping products methods according to demand behavior.

There are a large number of proposed models to improve the accuracy of demand estimation in the retail industry. However, most studies have aimed to propose a single forecasting method that is used for all sales time series under their study (Fildes, Ma, & Kolassac, 2019).

Among the articles that have proposed forecasting methods according to the different characteristics of the product time series, is the one presented by Syntetos et al. (2004), in which a product categorization scheme was proposed based on the squared coefficient of variation of demand sizes (CV2) and the average inter-demand interval (p). This forecast scheme was evaluated with 3000 products from a company in the automotive industry, using 3 demand forecasting methods: Croston method, Syntetos-Boylan method (Variant of the Croston method) and EWMA (simple Exponentially Weighted Moving Averages). The results indicated that the Syntetos-Boylan method has better performance for the intermittent, lumpy and erratic categories, while the Croston method obtained better results for the products of the smooth category.

Spiliotis et al. (2020) used the categorization proposed by Syntetos et al. (2004) and evaluated eleven statistical and seven machine learning methods for a set of 3,300 products from a retail company in Greece. The results obtained by evaluating the RMSSE (Root Mean Squared Scaled Error) quality metric, indicated that for intermittent products the Random Forest models, GBT (Gradient Boosting Trees), SVR (Support Vector Regression) and the Syntetos-Boylan method had the best results; for the lumpy type, the best methods were the same four previous ones and the KNNR (K-Nearest Neighbors Regression) model; regarding the smooth products, the best methods were GBT and Random Forests; and finally, for the erratic class products, the best models were SVR, Random Forests, GBT and KNNR. However, the study recognizes that only two characteristics of time series were explored in their proposal (squared coefficient of variation of demand sizes and average inter-demand interval), so the article highlighted the importance of examining additional variables of the time series for future analysis

The most recent article, to the best of our knowledge, that conducted a comparative forecasting model study for different product groupings, is Kiefer, Grimm, Bauer & Van Dinther (2021). The study used the categorization proposed by Williams (1984), in which the products can be segmented according to their intermittency and the variability of the order size, into the following groups: "A" for non-intermittent demands and constant orders sizes, “B” for very intermittent products and constant order sizes, “C” for little intermittent demands and variable order sizes, “D1” for products with very intermittent orders and with variable order sizes, and “D2” for very intermittent demand behaviors with highly variable order sizes. It is important to clarify that the thresholds proposed by Williams (1984) for the definition of categories, unlike the Syntetos et al. proposal, must be defined according to the sector, the type of item, and the inventory management policy. However, Kiefer et

al. (2021) implemented the same parameters used by Williams in 1984, and additionally joined the categories D1 and D2, renaming it as category “D”. The results obtained in this research when evaluating the Croston, Holt-Winter, Auto-ARIMA, Random Forest, XGBoost, SVR, Multi-Layer Perceptron (MLP) and LSTM models with the SPEC (Stock-keeping-oriented Prediction Error Costs) metric, proposed by Martin, Spitzer, & Kühl (2020), showed that for all classes the best performing model was Croston. Additionally, the second-best model for classes A, C and D, was LSTM, while for class B was AUTO-ARIMA. It is important to mention that the Croston method was not among the best models in the work carried out by Spiliotis et al. (2020), evidencing the lack of consistency in the results between the different investigations.

3 METHODOLOGY

3.1 System architecture

Our methodology is defined by seven steps. The following section explains the objective of each step.

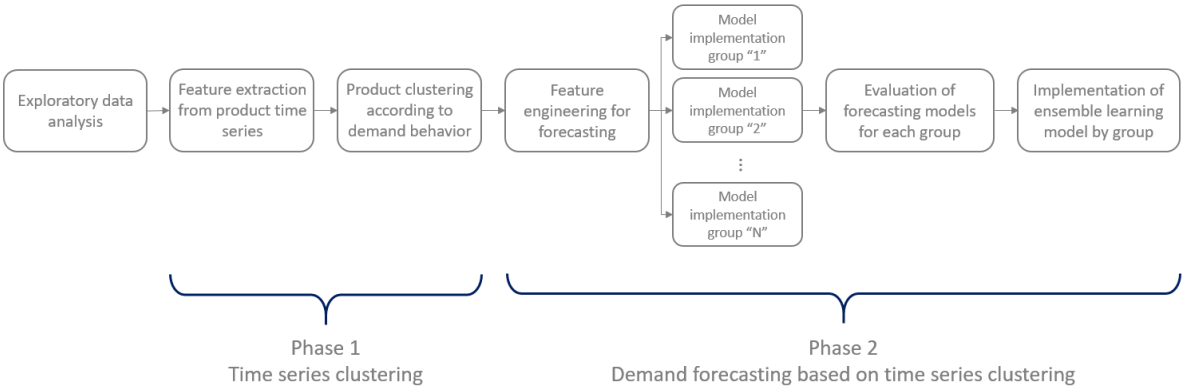

Figure 1 *Methodology of the project.*

1. Exploratory data analysis:
   Obtaining a general understanding of the dataset based on the identification of the types of variables, the statistical analysis of central tendency and dispersion measures, the identification of anomalous data that may be present in the information, the analysis of seasonality, cyclic and trend characteristics in the time series, among others. The objective is to obtain a general understanding of the business information.

2. Feature extraction from product time series
   Developing an autonomous feature extraction process to capture different variables that are associated with the demand of the products. The idea is to generate variables that could explain the cyclical, trend, seasonal and

irregular behavior of the time series. In this way, the objective is to find different variables that describe the behavior of the demand.

3. Product clustering according to demand behavior
   Developing a product grouping process based on similar characteristics of the demand behavior that are identified in the time series. To do this, different clustering methods can be evaluated to select the one that best fits the data.

4. Feature engineering for forecasting
   This step is composed of three processes: First, the feature extraction process to define the input features (X) that relate with the output (Y). Second, a filtering process where strongly correlated features will be eliminated. Finally, the feature importance step, where the most important input variables will be selected according to how useful they are at predicting the target variable. This step will be done with each group because each of them has different time series characteristics, so the best input variables that the models need to predict the demand could be different in each group.

5. Model implementation by group:
   This step will select different forecasting methods for each product group. In that way, different statistical and machine learning models will be implemented.

6. Evaluation of forecasting models for each group:
   Several quality metrics will be selected to evaluate the performance of all the implemented models in each group of products. Particularly, the next metrics will be considered: R2, MSE (Mean Squared Error) and MASE. According to the results in each group of products, a ranking will be made to identify the best model for each group.

7. Implementation of an ensemble learning model by group:
   Different ensemble learning models will be implemented for each product group, based on the models that showed the best performance in the previous step. Recent ensemble learning models will be studied to select those with satisfactory performance in the demand forecasting context.

4 RESULTS

4.1 Dataset and Case study

The dataset involves the unit sales of various products sold by a major retailer in the USA, organized in the form of grouped time series. More specifically, the dataset

involves the unit sales of 3,049 products, classified into 3 product categories (Hobbies, Foods, and Household) and 7 product departments, in which the above-mentioned categories are disaggregated. The products are sold across ten stores, located in three states (CA, TX, and WI). The historical data range from 2011-01-29 to 2016-05-22. Thus, the products have a (maximum) selling history of 1.941 days (5.4 years). This information is publicly available in Kaggle website as “M5 Forecasting – Accuracy”.

This paper evaluates 100 products of one of the stores located in California. The goods were randomly selected and included products from the 3 categories. The objective of the case study is to forecast the last 28 days of the dataset (4 weeks). That is, estimating the demand for all the 100 products, from 25/04/2016 to 22/05/2016.

The reason for using 100 products from the data set was due to the processing times involved in the entire process of generating variables for each time series and the implementation of the different forecast models. Currently, the entire proposed process takes about 4 hours using the standard computing capacity provided by google colab (which is variable).

4.2 Methodology implementation

4.2.1 Exploratory data analysis

The descriptive study was carried out starting from the most general groups of the dataset, and then progressively descending to subsets that allowed identifying more specific behaviors for certain groups. This initial understanding provides relevant information to the design of the forecasting system since it shows relevant trends in the data and suggests important input variables to be used.

General Analysis of the dataset

The aggregated sales of the 100 products of the California store show that the average demand is 532 units per day. Additionally, it is noted in figure 2 that there is one day for each year that presents an atypical value, where the total demand obtains values equal to or close to zero. This happens on December 25, where stores are closed due to the Christmas holidays.

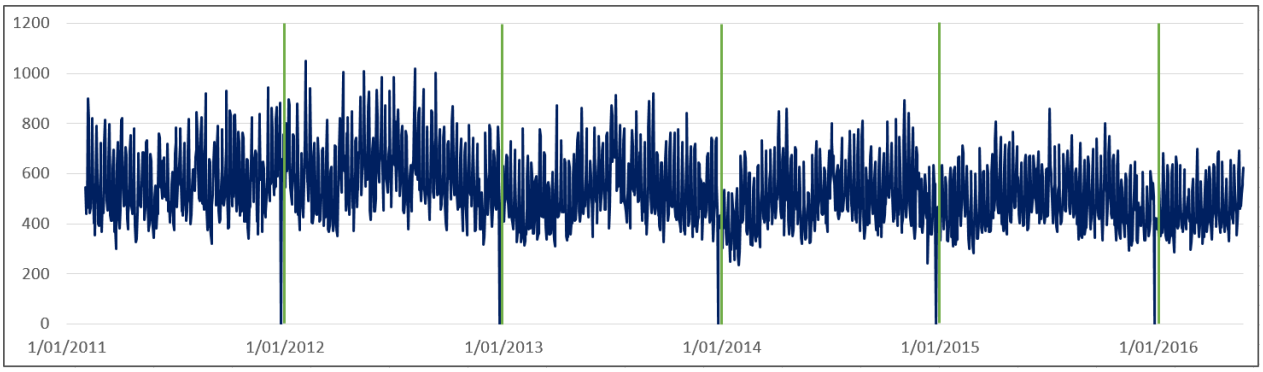

Figure 2 *Daily aggregated sales.*

It is also important to mention that there is a cyclical behavior in annual sales. It can be seen as a repetitive behavior each year. The year sales start with a progressive decrease in unit sales from January to April, and then an increase in demand begins until September. Finally, the sales start to decline again from September to next year's April. Additionally, it can be seen that the aggregate sales of the 100 products for the California store have a negative trend. Figure 3 is a smoothed graph with a 90-day moving average so that the described phenomenon can be more easily appreciated.

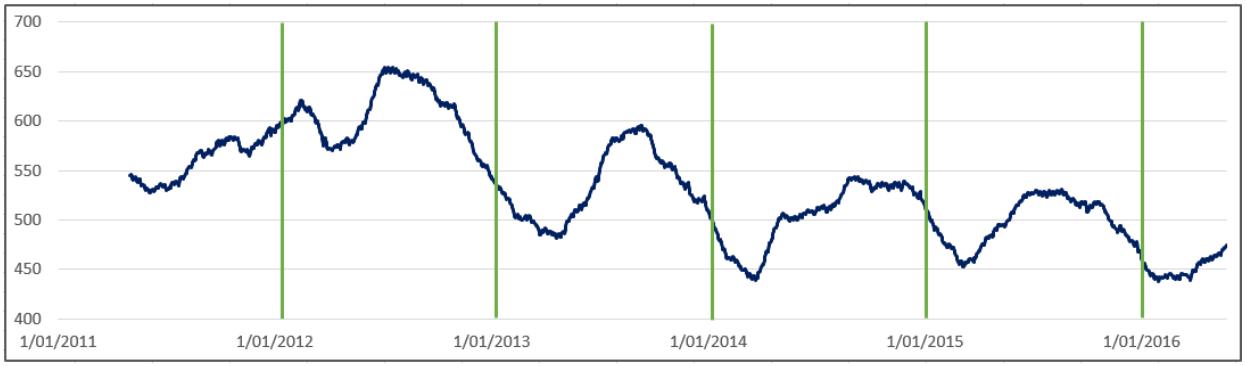

Figure 3 *Moving average (90-days) of the aggregated sales.*

Descriptive analysis by category

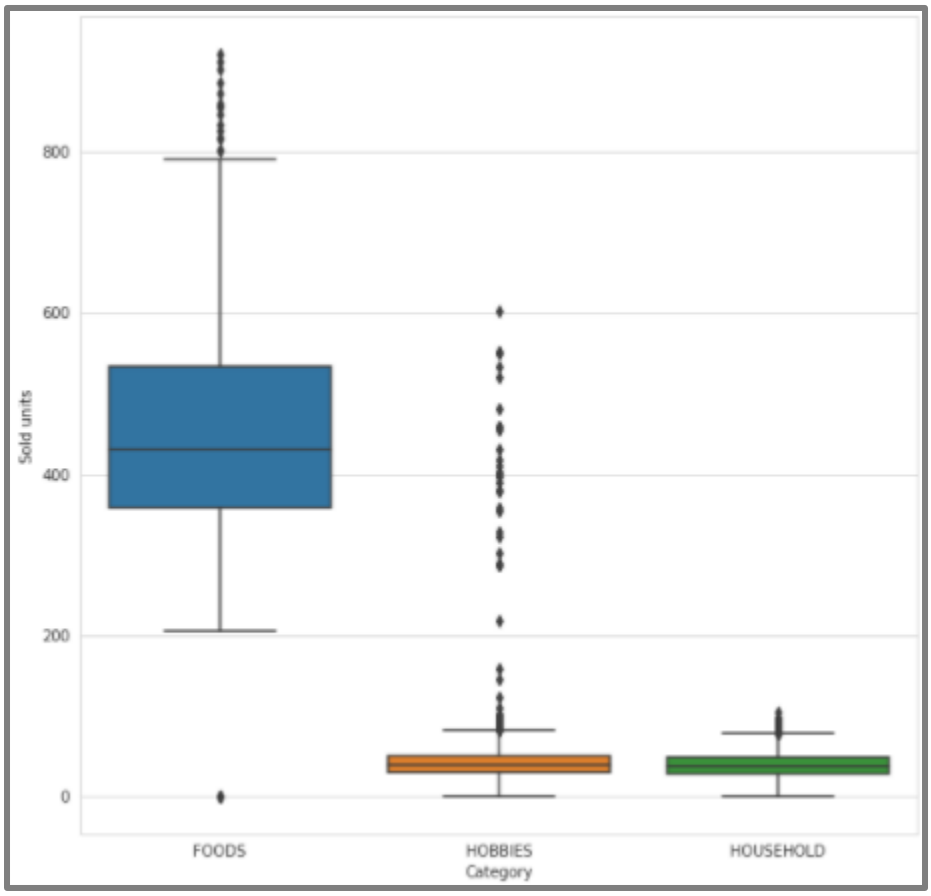

Figure 4 *Boxplot of average daily sales for each product category.*

The category with the highest average demand per day and the highest variance is the “Foods” category (see Figure 4). Regarding the “Household” and “Hobbies” categories, they have a similar average daily demand, which is around 40 units sold per day. However, it is important to mention that the “Hobbies” category presents sporadic over-demand on certain days, due to the large number of outliers that can be observed in the boxplot.

The above information is relevant to the proposed forecasting system, since products of the “Hobbies” and "Household" categories will have the highest product proportion with intermittent demand, and consequently, will be the most difficult products to forecast (as will be explained in the analysis of the results).

Sales behavior in temporary variables

Figure 5 *Average daily sales for different periods.*

It is important to understand how these product categories behave in different time windows to include date-time features that could be important as input information for the forecasting models. Here are the most relevant findings (see Figure 5):

- Day's analysis: It is observed that the three categories have very similar behavior in the average demand sales of the days of the week. As expected, sales have their highest peak on weekends, while for weekdays demand remains at the lowest levels (Except for Friday, where demand begins its growth due to being the weekend eve). Therefore, it is evident that for a large number of products the demand will be affected depending on the day of the week that is evaluated.
- Month's analysis: It can be seen that the "Household" category has two sales peaks throughout the year, one around April and the other in August. This information is relevant since the forecast dates for the case study are between April and May. Regarding the "Hobbies" category, it is observed that the highest peak of sales is in April and it notably drops in May. Additionally, a decrease in sales can be observed for the months that coincide with the coldest seasons of the year. Finally, regarding the "Foods" category, it is

important to take into account that sales begin to increase from April to July (when demand reaches its maximum value).

- Year's Analysis: Regarding the "Hobbies" category, it is important to mention that the unit sales tend to be higher on odd years than even years. However, it could be observed that the unit sales have a negative trend as the years go by. The "Household" category seems to be recovering from a sales drop in 2014 and is having the best daily average sales in 2015 and 2016. Lastly, the "Foods" category has a negative trend that started in 2013 and is extending until 2016. This category is the one that shows the greatest unit sales reduction of the three product categories.

4.2.2 Feature extraction for product grouping

Different features were extracted from each product's time series using the tsfresh package. The main objective was to generate variables that could explain the cyclical, trend, seasonal and irregular behavior of the time series to later cluster the products based on the extracted characteristics. In this way, the demand from day 1 to day 1,941 of every product was analyzed to extract features like peaks, maximums, minimums, trend, entropy, quartiles, among others, to identify and describe the components of each time series. Table 1 explains all the extracted features that have been used in the project, specifying whether its origin is statistical (ST) or from time series (TS).

Feature	Description	Type
abs_energy	Returns the absolute energy of the time series which is the sum over the squared values.	TS
absolute_sum_of _changes	Returns the sum over the absolute value of consecutive changes in the series x.	TS
Autocorrelation	Calculates the value of an aggregation function f_{agg} (e.g. the variance or the mean) over the autocorrelation R(l) for different lags.	TS
count_above_ mean	Returns the number of values in x that are higher than the mean of x.	ST
count_below_ mean	Returns the number of values in x that are lower than the mean of x	ST
Kurtosis	Returns the kurtosis of x (calculated with the adjusted Fisher-Pearson standardized moment coefficient G2). Kurtosis measures the peakedness of a time series.	ST
linear_trend	Calculate a linear least-squares regression for the values of the time series versus the sequence from 0 to the length of the time series minus one.	ST
longest_strike_ above_mean	Returns the length of the longest consecutive subsequence in x that is bigger than the mean of x.	ST
longest_strike_ below_mean	Returns the length of the longest consecutive subsequence in x that is smaller than the mean of x.	ST
Maximum	Calculates the highest value of the time series x.	ST
Mean	Returns the mean of x.	ST

mean_abs_ change	Returns the mean over the absolute differences between subsequent time series values.	ST
mean_change	Returns the mean over the differences between subsequent time series values.	ST
Median	Returns the median of x.	ST
number_peaks	Calculates the number of peaks of at least support n in the time series x. A peak of support n is defined as a subsequence of x where a value occurs, which is bigger than its n neighbors to the left and the right.	TS
Quantile	Calculates the q quantile of x. This is the value of x greater than q% of the ordered values from x.	ST
sample_entropy	Calculate and return sample entropy of x. Used for assessing the complexity of physiological time-series signals, diagnosing diseased states.	ST
Skewness	Indicates the symmetry of the probability density function (PDF) of the amplitude of a time series.	ST
standard_ deviation	Returns the standard deviation of x.	ST
sum_values	Calculates the sum over the time series values.	ST
value_count_0	Count occurrences of 0 values in time series x.	TS
variation_ coefficient	Returns the variation coefficient (standard error / mean, give a relative value of variation around mean) of x.	TS
x: the time series to calculate the feature of.		

Table 1 *Description of the extracted features.*

At the end of the feature extraction process for each product's time series, a total of 44 characteristics described different aspects of the time series behavior. Due to the high number of variables, efficiency issues and information quality, an elimination process of correlated variables was carried out so that there were no characteristics that would explain similar phenomena of the time series. For this, the Pearson correlation was used and it was defined that variables with greater values than a defined correlation threshold were filtered (For this work, 0.8 was defined). The resulting variables are specified below:

1. Skewness
2. Mean
3. Autocorrelationlag_6
4. Number_peaks_n_6
5. Linear_trend_attr_"slope"
6. Linear_trend_attr_"pvalue"
7. Linear_trend_attr_"rvalue"
8. Count_above_mean
9. Longest_strike_above_mean
10. Longest_strike_below_mean
11. Mean_change
12. Sample_entropy

It is important to mention that the filtering process of correlated variables was also evaluated with the Kendall method. However, the results obtained in step 3 showed that the silhouette method reached better results with the selected variables from Pearson's correlation.

4.2.3 Time series clustering

Two different clustering methods were implemented to evaluate the quality of the suggested groups: K-means and hierarchical clustering. For each clustering method, 2 alternatives were evaluated: A scenario using all the variables selected in the previous step and another using PCA. The reason for a second scenario with PCA was to avoid problems due to the curse of dimensionality. It is important to mention that the distance-based grouping methods, K-Means and hierarchical clustering, were selected over other methods based on densities, such as DBSCAN or OPTICS, since they guarantee that all products are assigned to a cluster (DBSCAN and OPTICS do not classify an element to a specific group when it is far from the high-density areas). Regarding the method of evaluation of the suggested groups, it was decided to implement the silhouette method since it is also based on distances between the elements and has the advantage of being a self-contained metric (it varies from -1 for overlapping clusters to 1 for defined and separate clusters).

The results using the silhouette metric were consistent in the number of suggested clusters in each of the methods, as shown in Table 2.

Number of Clusters	K-Means		Hierarchical Clustering	
	All variables	PCA	All variables	PCA
2	0,1958	0,2601	0,1804	0,2395
3	0,2106	0,2751	0,1874	0,2595
4	**0,2334**	**0,3123**	**0,2078**	**0,2793**
5	0,2018	0,2463	0,1396	0,2476
6	0,1961	0,2961	0,1461	0,2791

Table 2 *Silhouette method results of the implemented clustering methods.*

The results obtained in Table 2 show that the suggested number of clusters is 4 for each of the evaluated methods. Since the best result was obtained with the K-Means and PCA implementation, it was decided to use the suggested cluster by that method in this work.

After the cluster definition process, a study of the average characteristics of each group was carried out. The objective was to understand the general properties of the products that composed each of the groups. Table 3 shows some of the main characteristics.

Group	Kurtosis	Mean	Auto-correlation lag 7	Number peak n_6	Linear trend "slope"	Value count 0
0	13,4	1,5	0,33	131	0,000	1.050
1	2,8	17,0	0,48	198	-0,004	225
2	5,8	4,5	0,19	183	0,000	473
3	13,3	2,4	0,26	170	0,000	830

Table 3 *Average of the main characteristics in each group.*

In group 0, products are characterized by high intermittency due to a large number of null demands that its time series has. This peculiarity influences it to be the one with the lowest demand average of all groups. On the contrary, group 1 is characterized by a continuous demand since it has the highest mean value of the 4 groups and also has the lowest null demand cases in its time series. Additionally, it is important to mention that this group is the only one with a negative trend. Group 2 stands out for having the lowest auto-correlation value against the demand of the previous weekday. This results in less useful lagged variables for these type of products, and in consequence, make them more complex to forecast. Group 3 has similar characteristics to group 0. However, it has a lower presence of null demand and a much higher number of demand peaks, which makes its auto-correlation lower than the group 0 value.

4.2.4 Feature engineering for forecasting

Time series datasets must be converted to build the forecasting model. To achieve that, a feature engineering process was developed to provide strong relationships between input features and real demand value. However, the choice of the most appropriate set of features depends on the nature of the analyzed time series (Talagala, Hyndman, & Athanasopoulos, 2018). In consequence, this work implemented three different classes of features, so that the different groups of products will use the most relevant variables according to demand characteristics. Table 4 provides a brief description of the features used in this work.

Feature Class	Description	Feature	Type
Date Time Features	Information from the date/time values of each observation.	Wday_n: Day of the week. Wday_1 is Saturday, Wday_2 is Sunday, Wday_3 is Monday and so forth.	Bool
		DX_DY: Day of the month Range. That is, if it is a day between day X and day Y of the month.	Bool
		Event_Today: If there is an important event on the day.	Bool
		Event_Tomorrow: If there is an important event on the next day.	Bool
		Event_type: if the event is Cultural, National, Religious or Sporting.	Bool
		SNAP: A variable indicating whether the stores allow SNAP purchases on the examined date. SNAP is a nutrition assistance benefit for low-income families.	Bool
Lag Features	Values at prior time steps	D_lag_x: Lagged demand values for previous day (D_lag_1), 7 days ago (D_lag_7), 14 days ago(D_lag_14), 21 days ago(D_lag_21) and 28 days ago (D_lag_28).	Cont
		P_lag_x: Lagged price values for previous day (P_lag_1), 7 days ago (P_lag_7), 14 days ago (P_lag_14), 21 days ago (P_lag_21) and 28 days ago (P_lag_28).	Cont
Window Features	Summary of values over a fixed window of prior time steps.	D_rolling_mean_x: Demands mean of the previous 7 days, 14 days, 21 days and 28 days.	Cont
		D_rolling_std_x: Demands standard deviation of the previous 7 days, 14 days, 21 days and 28 days.	Cont
		P_rolling_mean_x: Prices mean of the previous 7 days, 14 days, 21 days and 28 days.	Cont
		P_rolling_std_x: Prices standard deviation of the previous 7 days, 14 days, 21 days and 28 days.	Cont
		All TS-Fresh features that were extracted in the "Feature extraction for product grouping" step with a rolling window of the last 28 days.	Cont

Table 4 *Description of the generated features for the forecasting models.*

After the feature extraction process, each product ended up with 93 features that described different demand behaviors. Therefore, a feature selection process was needed to choose the best characteristics for each of the product groups. This

process was done in two steps. First, a Pearson correlation process was implemented to eliminate variables with a higher correlation from a defined threshold (For this work, 0.8 was defined). After that, a feature selection process based on Random Forest was implemented. It is important to mention, that when training the Random Forest algorithm, the model can identify the most important features in the training step according to those variables that were most influential on the target variable. In this way, this work left those features with a higher coefficient than a defined threshold (the project used a 0.01 threshold). Table 5 shows the resulting variables after the feature selection process.

#	Group 0	Group 1	Group 2	Group 3
1	D_rolling_mean_t7	D_rolling_mean_t7	D_rolling_mean_t7	D_rolling_mean_t7
2	D_lag_1	D_lag_1	D_lag_1	D_lag_1
3	Demanda_abs_energy	D_lag_7	Demanda_variance	D_lag_7
4	D_rolling_std_t7	D_lag_14	D_lag_7	D_lag_14
5	Demanda_linear_ trend_attr_"slope"	D_lag_21	D_rolling_std_t7	Demanda_count_ above_t_0
6	Demanda_mean_ change	D_lag_28	D_lag_14	Demanda_variation_ coefficient
7	Demanda_linear_ trend_attr_"intercept"	D_rolling_std_t7	D_lag_21	D_lag_21
8	D_lag_7	Demanda_variation_ coefficient	D_lag_28	D_lag_28
9	Demanda_count_ above_t_0.1	Demanda_mean_ change	Demanda_variation _coefficient	Demanda_linear_ trend_attr_"slope"
10	Demanda_variation_ coefficient	P_lag_7	Demanda_linear_ trend_attr_"slope"	Demanda_sample_ entropy
11	Demanda_quantile_ q_0.1	Demanda_ autocorrelation_lag_27	Demanda_mean_ change	Demanda_mean_ change
12	D_lag_14	Demanda_linear_ trend_attr_"rvalue"	Demanda_count_ above_t_0.1	P_lag_7
13	Demanda_linear_ trend_attr_"rvalue"	Demanda_linear_ trend_attr_"pvalue"	P_lag_7	Demanda_linear_ trend_attr_"rvalue"
14	Demanda_linear_ trend_attr_"pvalue"	Demanda_sample_ entropy	Demanda_skewness	Demanda_skewness
15	D_lag_21	Demanda_ autocorrelation_lag_6	Demanda_ autocorrelation_lag_27	Demanda_ autocorrelation_lag_6
16	Demanda_skewness	Demanda_ autocorrelation_lag_13	Demanda_linear_ trend_attr_"rvalue"	Demanda_linear_ trend_attr_"pvalue"
17	Demanda_ autocorrelation_lag_27	Demanda_count_ above_t_0	Demanda_ autocorrelation_lag_20	Demanda_ autocorrelation_lag_13
18	Demanda_ autocorrelation_lag_20	Demanda_ autocorrelation_lag_20	Demanda_ autocorrelation_lag_13	Demanda_ autocorrelation_lag_27
19	Demanda_ autocorrelation_lag_6	Demanda_skewness	Demanda_ autocorrelation_lag_6	Demanda_ autocorrelation_lag_20
20	D_lag_28	Demanda_kurtosis	Demanda_linear_ trend_attr_"pvalue"	Demanda_count_ above_mean
21	Demanda_count_ below_mean		Demanda_sample_ entropy	Demanda_count_ below_mean
22	Demanda_longest_ strike_below_mean		Demanda_count_ above_mean	Demanda_longest_ strike_below_mean

23	Demanda_ autocorrelation_lag_13	Demanda_longest_ strike_below_mean	Demanda_longest_ strike_above_mean
24	Demanda_longest_ strike_above_mean	Demanda_count_ below_mean	Demanda_number_ cwt_peaks_n_6
25	P_lag_7		Demanda_number_cwt_ peaks_n_13
26	Demanda_number_ cwt_peaks_n_6		
27	Demanda_number_ cwt_peaks_n_13		

Table 5 *Resulting variables after the feature selection process.*

4.2.5 Model implementation

Various forecasting models were implemented to have different estimation approaches. This work implemented forecasting methods based on statistical, machine learning and deep learning models that capture different properties of the time series, so that they can obtain specific advantages for different forecasting scenarios that are presented in the studied products. This is also beneficial for the ensemble model step, where it is known that an increase in diversity among the base forecasting models can improve the accuracy of their combination (Lichtendahl & Winkler, 2020). The implemented forecast models and the parameters that were used in the work are mentioned below.

4.2.5.1 Naive

Setting all forecasts to be the value of the last observation (Hyndman & Athanasopoulos, 2013). In this way, the forecast at time t, $\widehat{y_t}$, is equal to the last known observation of the time series, y, as follows:

$$\widehat{y_t} = y_{t-1}$$

In the case study, the value of the last observation y_{t-1}would be the product demand of the previous day.

4.2.5.2 Naive – Previous weekday value

Setting all forecasts to be the value of the last weekday observation. In this way, the forecast at time t, $\widehat{y_t}$, is equal to the last weekday observation of the time series, y, as follows:

$$\widehat{y_t} = y_{t-8}$$

For example, the prediction of the next Wednesday's product demand would be the demand of the previous Wednesday. It was decided to use this type of method because one of the most relevant characteristics defined in step 4 was autocorrelation.

4.2.5.3 Moving average

The forecasts of all future values are equal to the average (or "mean") of the historical data (Hyndman & Athanasopoulos, 2013). Let the historical data be denoted by $y_1, y_2 \ldots, y_t$, then the forecast could be defined as:

$$\widehat{y_t} = (y_1 + y_2 + \cdots + y_{t-1})/(T-1)$$

Two time periods were implemented for the case study. One Moving average was defined with the last 7 observations, and the second one was set with the last 28 demand observations. This was defined by the importance of the autocorrelation variable detected in step 4.

4.2.5.4 Triple Exponential Smoothing

Forecasts produced using exponential smoothing methods are weighted averages of past observations, with the weights decaying exponentially as the observations get older (Hyndman & Athanasopoulos, 2013). Triple Exponential Smoothing is an extension of Exponential Smoothing that explicitly adds support for seasonality to the univariate time series. The Triple Exponential Smoothing parameters used in the case study were: “seasonal_periods” set equal to 7 due to the feature importance of the autocorrelation with previous weekday, “add” was chosen over “mult” as the “seasonal” parameter because, in the exploratory data analysis step, it wasn’t common to identify a product with an important seasonal increasing (In case the product had a seasonality). Regarding the “trend” parameter, it was set equal to “add” because the downward or upward trend was linear (Not exponential) in the majority of the products that were analyzed in the exploratory data analysis step. Lastly, the “smoothing_level”, “smoothing_slope” and “smoothing_seasonal” parameters were set to “optimized”, letting the algorithm choose the best parameters for each product.

It is important to mention that the “seasonal” and “trend” parameters were also set with “None” in the studied scenarios because an important number of products didn’t show those behaviors in their time series. However, the average of the quality metrics achieved lower results than the “add” parameters that were previously mentioned.

4.2.5.5 AUTOARIMA

ARIMA is composed of three parts, namely, AR, I, and MA, that can stand alone or be combined at will (Luceño & Peña, 2008). The “Auto Regressive” (AR) component helps to forecast the variable of interest using a linear combination of past values of the variable. The “Moving Average” (MA) part works by analyzing the error magnitude of the predicted values for the previous time periods, to make a better estimate for the current period. Finally, the last component, “Integrated (I)”, accounts for the overall trend in the data. AUTOARIMA was chosen for this case study because it helps to identify the optimal parameters for an ARIMA model. As the case study has distinct products with different time series characteristics, AUTOARIMA emerges as a relevant option to adapt the different model parameters to each product. The parameters that were selected in this project were: “AIC” as the “information_criterion” to seek the values that minimize the error; “start_p”, as the starting number of time lags of the AR model, was set in “7”; “start_q” as the order of the moving-average model, was set in “2”; “d”, as the order of first-differencing, was set in “1” to consider the differences between consecutive timesteps to eliminate upward/downward trend; and “Stepwise” set to “True” to use the stepwise algorithm

to identify the optimal model parameters (faster than fitting all hyper-parameter combinations and is less likely to over-fit the model).

4.2.5.6 Linear Regression

Linear Regression fits a linear model with coefficients $W = w_1, w_2, \ldots, w_n$ to minimize the residual sum of squares between the observed targets in the dataset, and the targets predicted by the linear approximation (Pedregosa et al., 2011). The case study implemented all the default parameters of the Scikit Learn package, which are "fit_intercept" set to "True" to calculate the intercept for the model, "Normalize" set to "False" because the data had been already normalized, "copy_X" set to "True", "n_jobs" set to "None", and "positive" set to "False" because it is not needed to force the coefficients to be positive.

4.2.5.7 KNN Regression

KNNR is a similarity-based method, generating forecasts according to the Euclidean distance computed between the points used for training and testing. Given x_i points as inputs, the method picks the closest k points of the training sample to them and then sets the prediction equal to the average of their corresponding target values (Spiliotis, Makridakis, Semenoglou, & Assimakopoulos, 2020). The case study used KNeighborsRegressor from the Scikit Learn package (Pedregosa et al., 2011), using the following parameters: Number of neighbors equals to "5", weights set to "distance" to assign greater influence to the closest neighbors; algorithm set to 'auto' to decide the most appropriate algorithm ('ball_tree', 'kd_tree', 'brute') based on the passed values to the fit method; and the parameter "p" defined in "2" to use the Euclidean distance.

4.2.5.8 Random Forest

Random Forest is a combination of Regression Trees, each one depending on the values of a random vector sampled independently and with the same distribution (Breiman, 2001). The case study implemented Random Forest from the Scikit Learn package (Pedregosa et al., 2011). The parameters to train the algorithm were: the number of trees equals set to "600", "MSE" as the function to measure the quality of a Split, the number of features to consider when looking for the best split was set to "sqrt", and the Bootstrap sampling was done with replacement. All other parameters were equal to the default value defined by Scikit Learn.

4.2.5.9 Gradient Boosting Trees

GBT builds one tree at a time, each new tree correcting the errors made by the previously trained one (Freund & Schapire, 1997). The model was implemented using the LightGBM Package and the following parameters: the learning rate was set to 0.05 and the maximum tree depth was defined to "8", for slow learning and better generalization. Additionally, the maximum tree leaves for base learners was set to 128 and the number of boosted trees to fit was equal to "500". MSE was defined as the loss function to optimize the model performance. All other parameters were equal to the default value defined by LightGBM.

4.2.5.10 *Long Short-Term Memory neural networks (LSTM)*

LSTM as an extension of RNN has a strong capability in forecasting time series data. The main difference between an RNN and LSTM is that LSTM can store long-range time dependency information and can suitably map between input and output data (Greff, Srivastava, Koutník, Steunebrink, & Schmidhuber, 2017). The model that was implemented in the case study was a Stacked LSTM model, composed of two hidden LSTM layers. The first layer had 32 units while the second layer had 16 units. The output layer was a dense layer with one unit. The model was fit using the efficient Adam version of stochastic gradient descent and optimized using the "MSE" loss function. The lag size was set in 7, which means that the data was prepared with 7 previous time steps as input, and with the following time step used as output.

4.2.6 Evaluation of forecasting models for each group

The quality metrics MSE, R^2 and MASE were selected for the performance evaluation of the forecasting models. The reason for choosing these metrics was the different advantages provided by each of its characteristics. MSE is one of the most popular functions in demand forecasting and is widely used as an objective function. Additionally, due to the way it is constructed, it does not generate indeterminate values. On the other hand, R^2 is a self-contained metric since its maximum value is 1. Therefore, it can be used to compare results from data with different scales. Finally, the MASE metric was selected because of several recent articles, such as Kiefer, Grimm, Bauer & Van Dinther (2021) and Spiliotis, Makridakis, Semenoglou & Assimakopoulos (2020) have implemented it. The advantage of MASE is its ease of interpretation since it compares the result of the forecast model with the naive model. In this way, if MASE is greater than 1, then it means that the model has worse performance than the naive model; while if it is closer to 0, then the model estimates the demand better than the mentioned reference model. Table 6 shows the quality metrics results for the test dataset (from 25/04/2016 to 22/05/2016) in each of the four product groups.

	GROUP 0			GROUP 1			GROUP 2			GROUP 3		
Method	**R^2**	**MSE**	**MASE**	**R^2**	**MSE**	**MASE**	**R^2**	**MSE**	**MASE**	**R^2**	**MSE**	**MASE**
Naive	-0.98	6.49	1.53	-0.63	63.13	0.82	-0.76	25.58	0.98	-0.77	8.56	1.27
Naive7	-0.93	6.55	1.51	-0.81	58.01	0.86	-0.94	25.81	1.06	-0.96	8.76	1.37
MA7D	-0.18	3.94	1.29	-0.17	46.27	0.74	-0.17	16.66	0.87	-0.19	5.67	1.11
MA28D	-0.07	3.54	1.21	-0.10	45.31	0.72	-0.08	15.70	0.83	-0.07	5.09	1.03
TES	-0.10	3.54	1.19	-0.33	38.58	0.73	-0.09	15.82	0.82	-0.13	5.04	1.00
ARIMA	-0.22	4.08	1.23	-0.46	52.12	0.82	-0.20	18.18	0.86	-0.22	5.71	1.06
LR	**0.19**	**2.39**	**1.08**	**0.46**	**24.37**	**0.49**	0.20	10.64	0.71	0.09	4.09	0.95
KNN	-0.20	3.81	1.26	-0.09	39.32	0.69	-0.22	15.47	0.86	-0.18	5.55	1.04
RF	0.02	3.05	1.20	0.14	34.10	0.62	0.08	12.00	0.78	0.03	4.45	0.99
LGBM	-0.04	3.36	1.25	0.03	37.80	0.65	0.03	12.10	0.79	-0.05	4.88	1.03
LSTM	0.18	2.64	1.06	0.42	29.57	0.53	**0.22**	**9.97**	**0.69**	**0.19**	**3.65**	**0.89**

Table 6 *Results in each product group for all the implemented models*

According to Table 6, the top-three performing methods in terms of R2, MSE and MASE are the Linear Regression, LSTM and Random Forest. However, the Random Forest model does not have a good performance in quality metrics, which is

evidenced by its difference with the results obtained by linear regression and LSTM. Interesting findings are made when examining the results in terms of the product groups. Group 0, which was characterized for having the most intermittent products, showed that Linear Regression and LSTM models had a similar performance in the quality metrics results. However, it can be observed that this group achieved the lowest result because of the null-demand recurrency. Regarding group 1, which was composed of products with recurrent and the highest mean demand, the results showed a superiority of the linear regression model over LSTM and Random Forest methods. It is important to mention that this group had the highest score with the three evaluated metrics, which means that a high mean demand value combined with a high autocorrelation, helps the models to achieve better results. For groups 2 and 3, the best performance was obtained by the LSTM model.

4.2.7 Implementation of an ensemble learning model by group

Five different combined models were implemented from the 2 or 3 best models of each of the groups. Initially, 3 basic ensemble models were implemented. The first, based on the mean of the 2 best performing models in each group, the second based on the mean of the 3 best performing models in each group, and the last based on the median of the 3 best performing models in each group. Subsequently, two blending models were implemented. The first one, using the two best-performing models of each group as the base model, and the second one using the three best-performing models of each group as the base model. The base models were trained with the demand data from 01/29/2011 to 03/27/2016 and then, the predictions were made for the validation set between 03/28/2016 to 04/24/2016. From the obtained forecast of the base models in the validation set, a Linear Regression model (Meta-Model) was trained to predict the demand of the test set (Dates between 04/25/2016 to 05/22/2016). Table 7 shows the test results of each of the ensemble models, as well as the obtained result of the best model of each group in the sixth step of the methodology.

	GROUP 0			GROUP 1			GROUP 2			GROUP 3		
Method	**R^2**	**MSE**	**MASE**	**R^2**	**MSE**	**MASE**	**R^2**	**MSE**	**MASE**	**R^2**	**MSE**	**MASE**
Best_M	0.197	2.391	1.080	0.458	24.36	0.496	0.221	9.969	0.686	**0.191**	**3.650**	**0.888**
CMean2	0.224	2.394	1.052	**0.481**	**24.67**	**0.494**	0.238	9.78	0.684	0.180	3.704	0.910
CMean3	0.186	2.533	1.094	0.397	26.88	0.529	0.204	10.18	0.711	0.151	3.867	0.928
CMedi3	0.175	2.590	1.100	0.402	26.72	0.521	0.196	10.41	0.714	0.131	3.984	0.942
Blend2	**0.186**	**2.349**	**1.038**	0.466	26.10	0.508	**0.245**	**9.397**	**0.678**	0.163	3.688	0.862
Blend3	-1.76	3.918	1.366	0.453	25.61	0.499	0.196	9.914	0.689	0.127	4.068	0.899

Table 7 *Results of each of the ensemble models and the best model for each group*

Table 7 highlights the best-performing models for each of the product groups. For group 0, a very similar performance was observed between the blending and the mean model using the 2 best models in step 6. However, the blending model was selected as the best one in group 0 because it obtained better results in 2 of the 3 quality metrics (MSE and MASE). For group 1, the combined model with the mean of the two best models was the one that obtained the best performance of all the

proposed alternatives. However, the blending model with 2 base models also got better performance than the best method in step 6. Regarding the products of group 2, it was observed that the best model was the blending one with 2 base models since it got better results in each of the three quality metrics than the other proposals. Finally, in group 3 no ensemble model obtained better performance than the LSTM method. One of the likely reasons is the wide difference between the LSTM performance with the other implemented models. Table 6 shows that LSTM has an R2 greater than the double of the second model with the best performance in group 3 (Linear Regression).

4.3 Results Analysis

According to Table 6, the demand behavior influences the model performance. The product groups of the proposed forecasting system had better results with different types of forecasting models. Particularly, the products from group 0 and group 1, which were characterized by a high autocorrelation value, had a better performance with the linear regression model, while the products of group 2 and group 3, which were characterized by a low autocorrelation value, achieved better results with the LSTM model. However, the best forecasting results were derived from the ensemble models that combined the best-performing models of each group. In this case, the blending model and the combined mean model that used the LSTM and Linear Regression methods obtained better results than the individual models in group 0, group 1 and group 2. Regarding group 3, none of the ensemble models exceeded the performance of the best individual model. This likely happened because the LSTM model significantly outperforms the other individual models in the quality metrics result. The difference in R2 between LSTM and the second-best model in group 3 is 0.1. This does not happen in any of the other groups where the ensemble models had better performance.

In consequence, the proposed forecasting system shows a better general performance than the traditional approach in which a single model is implemented for all the products. This means that the combination of the clustering product process and the use of ensemble models based on the top-performing models of each group generates better results than the individual models. Table 8 shows the three quality metric comparisons between the proposed forecasting system and the individual models used in the sixth step. The proposed forecasting system considers the best model of each group to calculate the overall result. The quality metrics used to obtain the best model of each group are R2, MSE and MASE (highlighted in Table 7). For each model, the overall average for the quality metrics of all products is shown in Table 8.

#	MODEL	R^2	MSE	MASE
1	**Proposed forecasting system**	**0.2592**	**9.3262**	**0.7573**
2	LSTM	0.2371	10.3873	0.7742
3	Linear Regression	0.2250	9.9916	0.7873
4	Random Forest	0.0740	12.2687	0.8804
5	Gradient Boosting Trees	0.0054	12.9535	0.9058
6	Moving Average (28 last days)	-0.0817	15.9731	0.9232
7	Triple Exponential Smoothing	-0.1307	15.0921	0.9162
8	Moving Average (7 last days)	-0.1786	16.7641	0.9731
9	KNN Regression	-0.1935	15.1222	0.9469
10	ARIMA	-0.2446	18.4267	0.9630
11	Naive	-0.7920	24.7241	1.1172
12	Naive Previous Weekday	-0.9232	24.1600	1.1704

Table 8 *General result of the proposed forecasting system and the individual models*

Note that the proposed forecasting system obtained better quality metrics results in each of the evaluated measures. Comparing the proposed approach with the top individual forecasting model (LSTM) shows that the R^2 measure is 9.32% higher, the MSE value is 10.21% lower, and the MASE is 2.18% lower.

Additionally, a second comparison is proposed in which the general average of the quality metrics is calculated according to each product and its best individual model according to the group to which it belongs. That is, the results of the quality metrics for the products of groups 0 and 1 are taken from the linear regression model and for the products of groups 2 and 3 from the LSTM model, and then is calculated the general average of each metric. Finally, the comparison of the “best individual model of each group" and the proposed forecasting system shows the contribution of the ensemble models (see Table 9).

#	MODEL	R^2	MSE	MASE
1	**Proposed forecasting system**	**0.2592**	**9.3262**	**0.7573**
2	Best individual model of each group	0.2457	9.5975	0.7706
	Percentage of improvement	**5.49%**	**-2.82%**	**-1.73%**

Table 9 *Comparison between the proposed model with a composed model from the best individual model of each group*

The proposed forecasting system obtained better quality metrics results in each of the evaluated measures in this second comparison. Table 9 results showed that the

R^2 measure is 5.49% higher, the MSE value is 2.82% lower, and the MASE is 1,73% lower.

In this way, the proposed approach demonstrates that when different methods are combined, it is likely that the biases counteract each other, thereby improving accuracy (Armstrong, 2001). However, it is important to take into account that the base models that are used in the ensemble models should have similar results to avoid a performance deterioration as has been viewed in the results of the third group. In this case, we observed that when the difference in R^2 was lower or equal to 0,04, the ensemble models perform better than the best individual model. On the contrary, when the R2 difference was high (0.1), then the individual model performed better than the ensemble models.

5 COMPARISON WITH OTHER WORKS

The comparison of the proposed forecasting system with other studies was done with the following criteria:

- Implemented forecasting models: The forecasting techniques that were used in the study.
- Product grouping method: The method with which the products were categorized.
- Ensemble Models: If ensemble learning models were evaluated in the study.

Academic article	Implemented forecasting models.	Product grouping method.	Ensemble Models
Our forecasting system	Naive. Seasonal Naïve. Moving Average. Triple Exponential Smoothing. Auto-ARIMA. Linear Regression. KNNR. Random Forest. Gradient Boosting Trees. LSTM.	K-Means with uncorrelated demand features.	Yes
Kiefer, Grimm, Bauer & Van Dinther. (2021)	Croston. Triple Exponential Smoothing. Auto-ARIMA. Random Forest. XGBoost. Auto-SVR. MLP. LSTM.	Williams' categorization (1984).	No

Spiliotis, Makridakis, Semenoglou & Assimakopoulos. (2020)	Naive. Seasonal Naïve. Simple Exponential Smoothing. Moving Averages. Croston's method. Syntetos–Boylan Approximation. Shale–Boylan–Johnston Approximation. Teunter–Syntetos–Babai method. ADIDA. iMAPA. Multi-Layer Perceptron (MLP). Bayesian Neural Network. Random Forest. Gradient Boosting Trees. KNNR. Support Vector Regression. Gaussian Processes (GP).	Syntetos et al. categorization (2004).	No
Syntetos, Boylan and JD Croston (2004)	Croston method. Syntetos-Boylan method. EWMA (simple Exponentially Weighted Moving Averages).	Syntetos et al. categorization (2004).	No

Table 10 *Comparison with other works.*

Syntetos et al. (2004), based on the Johnston & Boylan study (1996), proposed an alternative approach to the categorization problem according to which direct comparison of the forecasting methods results in specifying the demand categories. They considered Croston's method, Syntetos and Boylan's method and EWMA to forecast 3000 real-intermittent demand data series. The results indicated that the Syntetos-Boylan method has better performance for the intermittent, lumpy and erratic categories, while the Croston method obtained better results for the products of the smooth category. Spiliotis et al. (2020), evaluated the performance of Statistical and Machine Learning methods for 3300 real-time series of various consumption goods sold by a major retailer in Greece. The results showed that ML methods can provide significantly less biased and more accurate forecasts than well-established, statistical methods, like the Croston's method and its variants. Particularly, using the proposed Syntetos et al (2004) categorization, the results showed that for intermittent products the Random Forest models, GBT (Gradient Boosting Trees), SVR and the Syntetos-Boylan method were the best models in terms of RMSSE; for the lumpy category, the best methods were the four previous

models and KNNR; regarding the smooth products, the best methods were GBT and Random Forest; and finally, for the erratic class, the top-performing models were SVR, Random Forests, GBT and KNNR. Kiefer et al. (2021) used Williams's demand categorization to compare statistics and machine learning methods by applying a novel metric, called Stock-keeping-oriented Prediction Error Costs (SPEC). The results evidenced that the Croston algorithm is well suited to demand forecasting of intermittent and lumpy time series because it was the top-performing model in each of the product categories.

Based on Table 10, the proposed forecasting system differs from the other articles because it proposes a dynamic grouping process based on different uncorrelated demand features that describe the cyclical, trend and seasonal components of the time series. In this way, the proposal selects the number of groups according to the silhouette score of a K-Means algorithm, while the other studies use fixed values on only two types of variables that describe the intermittence and lumpiness of the time series. In consequence, the proposed system can cluster similar products that could be categorized in the same group by having a similar value on the two variables that are examined in the fixed grouping proposal. Additionally, the forecasting system that is proposed in this project evaluates the implementation of ensemble learning models from the top-performing forecasting methods of each group. This is not evaluated in the other studies and demonstrates that the demand estimation is enhanced when the base models that are used for the ensemble model have similar performances in the quality metrics.

6 CONCLUSIONS

Limited research has been done about demand forecasting based on the comparison of different methods in groups of products with distinct time series characteristics. Additionally, to the best of our knowledge, there are no studies evaluating ensemble models within the framework of product clustering according to the characteristics of the demand. In this way, this study seeks to contribute by (1) proposing a novel product grouping process based on a dynamic feature selection process according to the demand behavior of the selected products, (2) identifying the best forecasting models for each group of products from 10 different method alternatives, (3) evaluating the estimation benefits of ensemble models from the top-performing methods of each of the composed groups.

The results showed that the proposed forecasting system can select the number of product groups according to the demand characteristics that are associated with the time series. In this way, an aggrupation of four different groups was recommended. Additionally, it is observed that after the feature engineering process, each of the groups has its own feature importance ranking and a different set of top-performing models, being the linear regression and LSTM the best forecasting models in every group of each scenario. However, it was identified that according to the group characteristics, the linear regression model performed better with products with a

high demand autocorrelation, while LSTM had better results with low autocorrelated demand.

Regarding the ensemble models that were implemented in the proposed forecasting system, it is observed that they are an important alternative to take into account when the base forecasting models have acceptable and similar performances in the quality metrics. Thus, the mean and blending models that used linear regression and LSTM as base models were the top-performing models in groups 0, 1 and 2 since the difference in their R2 was less or equal to 0.04. On the other hand, the mean and blending models that used base models with high-performance differences obtained lower results compared to the best individual model.

It is important to mention that due to the nature of intermittent demand that characterizes a large number of products in the dataset, the results of the quality metrics are, in general, low. However, it can be seen that the proposed method obtains better results than all the conventional forecasting models that were implemented in this study. Additionally, when comparing the results with Kiefer et al. (2021), who used the information from the same dataset, it is evident that the results obtained are competitive in terms of MASE.

Different aspects are worth studying for future work. First, it would be interesting to integrate more variables from external information related to this case study, such as macroeconomic and weather variables, to analyze if they could be useful to achieve better results. Another topic to analyze would be the evaluation of another type of correlation technique such as VIF (variance inflation factor) to eliminate highly correlated variables. This is because it does not have the Pearson limitation of being a pair-wise comparison, so it is worth analyzing if it could select better features. Regarding the feature selection process, it would be interesting to evaluate the performance of conventional regularization techniques where the variable selection problem is naturally incorporated. Another aspect to focus on is the evaluation of other forecasting models, like prophet or Croston, to identify those with similar or better performance than linear Regression and LSTM. This, for extending the knowledge about which models suit better to each group of products and use them as base models for the ensemble model process. Regarding the grouping product process, it would be interesting to compare our grouping process against Williams' and Syntetos' grouping method with different sets of products to evaluate its influence on the proposed forecasting system performance. Finally, as blending models allow the use of different options of meta-models (model that learns how to best combine the predictions of the base models), it would be interesting to examine other different options from the linear regression to evaluate if the blending models could reach better results.

7 REFERENCES

Abolghasemi, M., Beh, E., Tarr, G., & Gerlach, R. (2020). Demand forecasting in supply chain: The impact of demand volatility in the presence of promotion. *Computers & Industrial Engineering, 142*.

Abolghasemi, M., Hurley, J., Eshragh, A., & Fahimniab, B. (2020). Demand forecasting in the presence of systematic events: Cases in capturing sales promotions. *International Journal of Production Economics, 230*.

Armstrong, J. S. (2001). *Principles of Forecasting. A Handbook for Researchers and Practitioners.* Pennsylvania: Springer.

Atiya, A. F. (2020). Why does forecast combination work so well? *International Journal of Forecasting, 36*, 197-200.

Ban, T., Zhang, R., Pang, S., Sarrafzadeh, A., & Inoue, D. (2013). Referential kNN Regression for Financial Time Series Forecasting. *International Conference on Neural Information Processing, 8226*, 601-608. doi:10.1007/978-3-642-42054-2_75

Barker, J. (2020). Machine learning in M4: What makes a good unstructured model? *International Journal of Forecasting, 36*, 150-155.

Breiman, L. (2001). Random Forests. *Machine Learning, 45*, 5–32. doi:10.1023/A:1010933404324

Cáceres, G., & Rodríguez, J. E. (2011). Clustering of time series data. State of the art. *Vinculos, 8*(1), 210-231.

Chen, X., & Han, T. (2019). Disruptive Technology Forecasting based on Gartner Hype Cycle. Atlanta: IEEE.

Ching-Wu, C., & Guoqiang Peter, Z. (2003). A comparative study of linear and nonlinear models for aggregate retail sales forecasting. *International Journal of Production Economics, 86*(3), 217-231.

Christ, M., Braun, N., Neuffer, J., & Kempa-Liehr, A. W. (2018). Time Series FeatuRe Extraction on basis of Scalable Hypothesis tests (tsfresh – A Python package). *Neurocomputing, 307*, 72-77.

Clemen, R. (1989). Combining forecasts: A review and annotated bibliography. *International Journal of Forecasting, 5*(4), 559-583.

Deng, T., Zhao, Y., Wang, S., & Yu, H. (2021). Sales Forecasting Based on LightGBM. *2021 IEEE International Conference on Consumer Electronics and Computer Engineering (ICCECE)* (págs. 383-386). Guangzhou, China: IEEE. doi:10.1109/ICCECE51280.2021.9342445

Dodge, Y. (2008). The Concise Encyclopedia of Statistics. En Y. Dodge. New York: Springer. Recuperado el 9 de Mayo de 2021, de https://doi.org/10.1007/978-0-387-32833-1_401

Fildes, R., Ma, S., & Kolassac, S. (2019). Retail forecasting: Research and practice. *International Journal of Forecasting*. doi:10.1016

Freund, Y., & Schapire, R. (1997). A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *Journal of Computer and System Sciences, 55*, 119-139. doi:10.1006/jcss.1997.1504

Greff, K., Srivastava, R., Koutník, J., Steunebrink, B., & Schmidhuber, J. (2017). LSTM: A Search Space Odyssey. *IEEE Transactions on Neural Networks and Learning Systems, 28*, 2222 - 2232. doi:10.1109/TNNLS.2016.2582924

Gutierrez, R. S., Solis, A. O., & Mukhopadhyay, S. (2008). Lumpy demand forecasting using neural networks. *International Journal of Production Economics*, 409-420.

Hewamalage, H., Bergmeir, C., & Bandara, K. (2021). Recurrent Neural Networks for Time Series Forecasting: Current status and future directions. *International Journal of Forecasting, 37*(1), 388-427.

Hong, T., Gui, M., Baran, M., & Willis, H. L. (2010). Modeling and forecasting hourly electric load by multiple linear regression with interactions. *IEEE PES General Meeting*, 1-8. doi:10.1109/PES.2010.5589959

Hyndman, R. J., & Athanasopoulos, G. (2013). *Forecasting: principles and practice.* Melbourne, Australia: otexts. Obtenido de https://otexts.com/fpp2/index.html

Hyndman, R., Kang, Y., Montero-Manso, P., Talagala, T., Wang, E., Yang, Y., & Moorman, J. (2019). Tsfeatures: Time Series Feature Extraction.

Ivanov, D., Tsipoulanidis, A., & Schönberger, J. (2017). *Global Supply Chain and Operations Management.* Springer.

Jaganathan, S., & Prakash, P. (2020). A combination-based forecasting method for the M4-competition. *International Journal of Forecasting, 36*, 98-104.

Jiménez, M., Aguilar, J., Monsalve-Pulido, J., & Montoya, E. (2021). An automatic approach of audio feature engineering for the extraction, analysis and selection of descriptors. *International Journal of Multimedia Information Retrieval, 10*, 33–42.

Johnston, F. R., & Boylan, J. E. (1996). Forecasting for items with intermittent demand. *Journal of the Operational Research Society, 46*, 113–121.

Kiefer, D., Grimm, F., Bauer, M., & Van Dinther, C. (2021). Demand Forecasting Intermittent and Lumpy Time Series: Comparing Statistical, Machine Learning and Deep Learning Methods. *Demand Forecasting Intermittent and Lumpy Time Series: Comparing Statistical, Machine Learning and Deep Learning Methods.* Hawaii. doi:10.24251

Kumar, M., & Thenmozhi, M. (2006). Forecasting Stock Index Movement: A Comparison of Support Vector Machines and Random Forest., (pág. 16). doi:10.2139

Kuvulmaz, J., Usanmaz, S., & Engin, S. N. (2005). Time-Series Forecasting by Means of Linear and Nonlinear Models. *MICAI 2005: Advances in Artificial Intelligence. 3789*, págs. 504-513. Springer.

Li, Y., Zhua, Z., Kong, D., Han, H., & Zhao, Y. (2019). EA-LSTM: Evolutionary attention-based LSTM for time series prediction. *Knowledge-Based Systems*. doi:10.1016

Liang, F. (2005). Bayesian neural networks for nonlinear time series forecasting. *Statistics and Computing, 15*, 13-29.

Lichtendahl, K. C., & Winkler, R. L. (2020). Why do some combinations perform better than others? *International Journal of Forecasting, 36*, 142-149. Obtenido de https://doi.org/10.1016/j.ijforecast.2019.03.027

Luceño, A., & Peña, D. (2008). Autoregressive Integrated Moving Average (ARIMA) Modeling. En *Encyclopedia of Statistics in Quality and Reliability.*

Ma, S., & Fildes, R. (2021). Retail sales forecasting with meta-learning. *European Journal of Operational Research, 288*(1), 111-128.

Makridakis, S., & Winkler, R. (1983). Averages of Forecasts: Some Empirical Results. *Management Science, 29*(9), 987-1112.

Makridakis, S., Hyndman, R. J., & Petropoulos, F. (2020). Forecasting in social settings: The state of the art. *International Journal of Forecasting, 36*(1), 15-28.

Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2020). The M5 Accuracy competition: Results, findings and conclusions. *International Journal of Forecasting*.

Martin, D., Spitzer, P., & Kühl, N. (2020). A New Metric for Lumpy and Intermittent Demand Forecasts: Stock-keeping-oriented Prediction Error Costs. *53rd Annual Hawaii International Conference on System Sciences (HICSS-53).* Hawaii.

Martínez, F., Frías, M. P., Pérez-Godoy, M. D., & Rivera, A. J. (2018). Dealing with seasonality by narrowing the training set in time series forecasting with kNN.

Expert Systems with Applications, 103, 38-48. doi:10.1016/j.eswa.2018.03.005

Mei, J., He, D., Harley, R., Thomas, H., & Qu, G. (2014). A random forest method for real-time price forecasting in New York electricity market. *2014 IEEE Power & Energy Society General Meeting.* Maryland. doi:10.1109

Montero-Manso, P., Athanasopoulos, G., Hyndman, R., & Talagala, T. S. (2020). FFORMA: Feature-based forecast model averaging. *International Journal of Forecasting, 36*(1), 86-92.

Moon, M. A. (2013). *Demand and Supply Integration: The Key to World-class Demand Forecasting.* New Jersey: FT Press.

Nikolopoulos, K. (2021). We need to talk about intermittent demand forecasting. *European Journal of Operational Research, 291*(2), 549-559.

Pavlyshenko, B. M. (2019). Machine-Learning Models for Sales Time Series Forecasting. *Data*, 4-15. doi:10.3390/data4010015

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., . . . Brucher. (2011). *Scikit-learn: Machine Learning in Python*. Obtenido de https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html?highlight=linear#sklearn.linear_model.LinearRegression

Petropoulos, F., & Svetunkov, I. (2020). A simple combination of univariate models. *International Journal of Forecasting, 36*(1), 110-115.

Petropoulos, F., Kourentzes, N., Nikolopoulos, K., & Siemsen, E. (2018). Judgmental selection of forecasting models. *Journal of Operations Management, 60*, 34-46. doi:10.1016

Ribeiro, M. H., & dos-Santos, L. (2020). Ensemble approach based on bagging, boosting and stacking for short-term prediction in agribusiness time series. *Applied Soft Computing, 86*. doi:10.1016/j.asoc.2019.105837

Spiliotis, E., Makridakis, S., Semenoglou, A.-A., & Assimakopoulos, V. (2020). Comparison of statistical and machine learning methods for daily SKU demand forecasting. *Operational Research*, 1-25.

Syntetos, A., Boylan, J., & Croston, J. (2004). On the categorization of demand patterns. *Journal of the Operational Research Society, 56*, 495-503.

Talagala, T., Hyndman, R., & Athanasopoulos, G. (May de 2018). Meta-learning how to forecast.

Taylor, J. W. (2003). Exponential smoothing with a damped multiplicative trend. *International Journal of Forecasting, 19*(4), 715-725.

Theodorou, E., Wang, S., Kang, Y., & Spiliotis, E. (2021). Exploring the representativeness of the M5 competition data.

Williams, T. (1984). Stock Control with Sporadic and Slow-Moving Demand. *Journal of the Operational Research Society, 35*, 939-948.