Comparing Gradient Boosting Algorithms to Forecast Sales in Retail

Ana Clara Chaves Sousa[1] **, Thaís Gaudencio do Rêgo**[1] **,**
Yuri de Almeida Malheiros Barbosa[1] **, Telmo de Menezes e Silva Filho**[2]

[1]Universidade Federal da Paraíba (UFPB)

[2]University of Bristol, UK

ana.chaves@academico.ufpb.br, gaudenciothais@gmail.com

yuri@ci.ufpb.br, telmo.silvafilho@bristol.ac.uk

Abstract. *The availability of data and the increased processing power of computers have made it easier to make decisions based on data, specially with Artificial Intelligence. One area where AI is widely applicable in companies is Supply Chain Management, particularly in demand forecasting. This paper aims to forecast sales for a company in the Cosmetic, Fragrance, and Toiletry market. Data from 2019 to 2023 were used from two different sales channel. To predict the demand, three Gradient Boosting algorithms (CatBoost, LightGBM, and XGBoost) were compared, and forecasts were made for three different time horizons (next period, five and ten periods ahead). After the experiments, LightGBM showed more stability compared to the other models.*

1. Introduction

Big data and the advancement of computer processing power have facilitated data-driven decision-making, which involves making decisions based on data analysis instead of intuition [Provost and Fawcett 2013a]. Over the last twenty years, there has been a significant amount of investment in business infrastructure, resulting in a better capacity to gather data. Companies are contemplating the possibility of leveraging their data expertise to gain a competitive edge [Provost and Fawcett 2013b].

One area where data-driven decisions can be applied is Supply Chain Management (SCM), which includes logistics, transportation, operations management, procurement, engineering, research, and development [Schoenherr and Speier-Pero 2015]. Customer behavior analysis, trend analysis, and demand forecasting are among the various possibilities in this field [Seyedan and Mafakheri 2020].

In this paper, the focus will be on demand forecasting in a real-world context using Machine Learning techniques. The proposed methodology will be implemented within a company operating in the Cosmetic, Fragrance, and Toiletry (CFT) industry. The name of the company will be kept confidential to preserve its privacy and integrity, which is a necessary condition for continued collaboration in the research and obtaining more detailed and accurate information.

Demand forecasting is a complex task that can have significant impacts on inventory management. If the demand is overestimated, it may cause waste reduction, whereas if it is underestimated, it is possible for stockouts to occur [Andrade and Cunha 2022].

Planning for product availability for consumer purchase requires months of advance planning, depending on the production lead time. The supply chain refers to the series of processes that connect suppliers and customers, as well as the companies involved, from the point of origin of raw materials to the final consumption of the finished product [Cox et al. 1995]. Having said that, demand forecasting affects the entire supply chain by influencing decisions on raw material purchases, storage, and transportation.

The goal of this study is to compare different Gradient Boosting (GB) methods for demand forecasting over multiple time horizons, which are: one period ahead, five periods ahead and ten periods ahead. While comparing the methods, these being CatBoost, LightGBM and XGBoost, it was analyzed which one presented the better performance considering all horizons. GB was used due to the complexity of the data available and its flexibility to handle with categorical features and outliers. The performance of the models will be evaluated using Weighted Absolute Percentage Error (WAPE).

The structure of this paper is presented as follows: In Section 2, five related works on demand forecasting using machine learning techniques are outlined. Section 3 outlines the methodology employed in this study. The results are reported in Section 4. Finally, Section 5 provides concluding remarks.

2. Related Work

Robustness is a big challenge faced in the long-term demand forecast, which makes this task very complex [Zhou et al. 2022]. The model's reliability in detecting seasonality can be affected by patterns found in noisy data and that is the reason why robustness is difficult in this kind of problem [Wu et al. 2021].

A research project used Walmart sales data, which is available in Kaggle's M5 competition dataset, to compare various models for predicting the next 28 days. The dataset contains five years of sales data for three categories and three states in the United States of America. The study examined both parametric and non-parametric models, including Autoregressive Integrated Moving Average (ARIMA), LightGBM, and Prophet. The performance of these models was analyzed and compared, with ARIMA showing the best results presenting a Root Mean Squared Error (RMSE) of 1.09. Although LightGBM had a higher RMSE of 1.18, it was more computationally efficient [Hasan et al. 2022].

Another recent investigation from 2022 also used data from Kaggle's M5 competition to compare tree models: Decision Tree, Random Forest, and Gradient Boosting (GB). The metric used was the Mean Average Percentage Error (MAPE). The model with the lowest error was GB (5.5%), while Random Forest made an error of 5.9% and Decision Tree made an error of 7.5%. The significance of the feature engineering stage in obtaining satisfactory results from tree models was emphasized. Furthermore, this model type is interpretable since its rules can be visualized, enabling decision makers to comprehend the factors that impact the predictions [Spiliotis et al. 2022].

In the same year, a study also discussed the impact of variable engineering on predictive sales analysis. Genetic algorithms were used to achieve the goal. Four tree-based algorithms, Random Forest, XGBoost, LightGBM, and CatBoost, were employed to predict sales for the following month. The performance of the four models was compared using MAPE before and after feature engineering. Then, the models with the lowest errors

(Random Forest and LightGBM) were evaluated for predictions at further time horizons, namely the second and third months. At the conclusion of the study, it was determined that the Random Forest model achieved the best outcomes, with a MAPE of 10.2% in contrast to LightGBM's 10.8%, CatBoost's 13.5%, and XGBoost's 13.8% [Li 2022].

A study was made focusing on comparing the performance of various models across multiple time horizons. The dataset contains all card transactions captured by a Brazilian acquiring company from January 1, 2014, to July 31, 2018, with a total of 1,673 daily observations. Predictions were made for four different time horizons: the next day, seven days ahead, thirty days ahead, and ninety days ahead. The significance of considering holidays, special dates, and other calendar effects was highlighted. Traditional time series forecasting models such as Naïve, HoltWinters, TBATS (acronym for Trigonometric seasonality, Box-Cox transformation, ARIMA errors, Trend, Seasonal components), and Seasonal Autoregressive Integrated Moving Average (SARIMA) were evaluated, along with established Machine Learning and Deep Learning models, such as Multilayer Perceptron (MLP), Long-Short Term Memory (LSTM), XGBoost, and Prophet. Among all the models, Prophet achieved the lowest error at the furthest time horizon, with an RMSE of 17.2 [Lopes 2022].

In another research aimed at assessing the reliability of predictive analytics across various time horizons, XGBoost was employed to generate daily sales volume forecasts for a week, resulting in seven forecasts in total. The data was sourced from a partner company and consisted of sales records for five products. The performance of the model was evaluated using mean absolute error (MAE) and weighted mean absolute error (WMAE) metrics. Four of the five products showed higher errors in the forecasts on day 7 compared to day 1. The average WMAE was found to be 2.76, while the MAE was 2.44 [Baržić et al. 2022].

The primary commonality between the current research and previous studies is their focus on predicting sales within the retail industry using at least one Gradient Boosting algorithm. Nevertheless, there are also discrepancies. The studies conducted by [Hasan et al. 2022] and [Spiliotis et al. 2022] did not explore forecasts across various time horizons. [Li 2022] performed an experiment for distant time horizons of up to three months but did not focus on it. Although [Baržić et al. 2022] also compared sales volume forecasts at different time intervals, they only did so for a maximum of one week. The study that is most similar to the current one is [Lopes 2022], but an essential distinction to note is that while [Lopes 2022] had a maximum time horizon equivalent to three months, the present study has a more extended term forecast of around eight months ahead.

A significant differentiation between the current study and previous research is the presence of distinct contextual factors and data characteristics. Unlike previous studies that typically dealt with regularly spaced time series data (such as daily or weekly data), this study focuses on a time context aligned with business strategies, referred to as cycles. These cycles do not have equal intervals, which poses challenges when employing traditional time series methods. Furthermore, the study encompasses diverse consumer attraction campaigns and considers products that are not classified as essential commodities.

3. Methodology

The methodology of this work is described in Figure 1, with all the steps used to conduct the study.

1. **Data Preprocessing:** All data cleaning and formatting were carried out in this step, which is detailed in Section 3.3.
2. **Feature Engineering:** Section 3.4 provides a detailed explanation of this step. Creating new variables is particularly important for tree-based methods, as previously mentioned in Section 2. In addition, they are necessary to forecast sales when traditional time series methods are not being used.
3. **Data Preparation for Multiple Time Horizons:** In this step, three datasets were created separately for each time horizon. Details are provided in Section 3.5.
4. **Hyperparameter Optimization with Validation Set:** This step involved separating the data into several training and validation sets for each horizon and model, and then the hyperparameters were optimized. Details are also in Section 3.5.
5. **Model Training and Testing:** In this step, the GB algorithms were trained and tested. Section 3.5 provides further details on this part.
6. **Model Evaluation:** Finally, an analysis was carried out to understand the errors of the models and compare them, as outlined in Section 4.

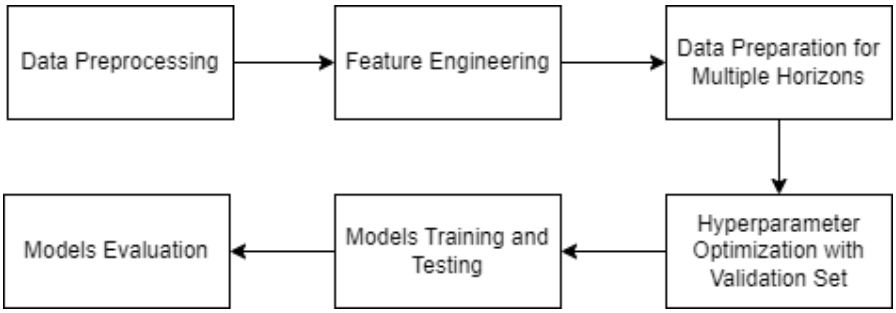

Figure 1. Methodology.

Section 3.1 shows information about the data used in this study. Section 3.2 shows more detailed information about the machine that was used to run this work and other tools, such as programming languages, libraries and cloud.

3.1. Data

This study used a dataset from a partner company in the CFT market, which included originally around 8.8 million instances (in the cycle/state/channel/product level) and 16 variables. The granularity of the raw data can be seen with more details in Table 1. The data pertained to sales made between 2018 and 2023. The data from 2018 was excluded from the training set and used solely for generating lag features, as described in Section 3.4. The data available from 2023 was used as holdout set to test the models.

Table 1 displays the available raw variables. To conduct the study, products were grouped by brand, subcategory, and category. Consequently, the modeling was carried out based on the following granularity: cycle, channel, state, category, subcategory, and brand.

Table 1. Variables available in the dataset.

Feature	Description
Cycle	Period of the year, with each year consisting of 17 cycles.
Cycle description	Cycle type definition.
Date of cycle start	Date on which cycle was started.
Date of cycle end	Date cycle ends.
Channel	Sales channel, may be Direct Sales or Stores.
State	Federative Unit of Brazil.
Category	Category of the product.
Subcategory	Subcategory of the product.
Brand	Product brand.
Product code	Product code/SKU.
Sold amount	**Target**. Quantity of products sold.
Practiced value	Sale value with discounts.
Table value	Sale value without discounts.
Discount value	Discount value applied.
Discount percentage	Discount percentage.
Unit Price	Unit price considering the discounts.
Promotion	Indicates if the product is promoted or not.
Focus	Indicates if the product is in focus campaign or not.
Subfocus	Indicates if the product is in campaign subfocus or not.

3.2. Hardware and Technologies

To run the experiments, JupyterLab on the GCP (Google Cloud Platform) workbench was used. The hardware utilized was 8 vCPUs and 30 GB RAM. Python and SQL were the programming languages used in this study. The following libraries were employed within Python 3.7:

- **pandas**, for manipulating the data through dataframes;
- **numpy**, for performing mathematical operations and using arrays;
- **matplotlib.pyplot**, for data visualization through charts;
- **seaborn**, for data visualization as well;
- **google.cloud**, for connecting the data to Google BigQuery;
- **lightgbm**, for running the experiments using the LightGBM algorithm;
- **xgboost**, to run the experiments using the XGBoost algorithm;
- **catboost**, for the execution of experiments using the CatBoost algorithm;
- **sklearn**, for performing general applications of Machine Learning;
- **optuna**, for hyperparameter optimization.

3.3. Data Preprocessing

During this step, the data was initially formatted according to each variable. Since the data had been previously structured by the Data Engineering team of the company, not much processing was required at this stage. Following the formatting process, the missing values were analyzed and their proportion was determined concerning to the complete dataset. Subsequently, the rows with the missing values were removed. In total, the dataset was reduced by only 0.28% after cleaning the nulls.

In addition to cleaning missing data, a further cleaning process was carried out to eliminate deactivated products, categories, subcategories, and brands. This was done because there was no need to forecast demand for products that were no longer part of the company's portfolio. As a result of this cleaning process, 5.78% of the dataset was removed.

As stated in Section 3.1, the modeling was conducted at the following levels of granularity: cycle, channel, state, category, subcategory, and brand. It is worth noting that the data were grouped at this level, resulting in a reduced dataset of approximately 950k instances.

3.4. Feature Engineering

At this stage, different features were created to serve as input to the models. Table 2 shows the features that were created considering the products, promotions and campaigns.

Table 2. Features created considering the products, promotions and campaigns.

Original Features	Features Created	Description
Product, Cycle, Category, Subcategory, Brand	Quantity of Products (1 feature)	How many products are in that category/subcategory/brand.
Promotion, Cycle, Product, Category, Subcategory, Brand, State	Products on Promotion (1 feature)	Quantity of products on promotion within the primary key.
Discount Percentage, Cycle, Product, Category, Subcategory, Brand, State	Discount Range (4 features created)	Number of products belonging to each discount range (0%, 10%, 20%, 30%) within the primary key.
Focus, Cycle, Product, Category, Subcategory, Brand, State	Products in Focus (1 feature)	Quantity of products in focus within the primary key.
Subfocus, Cycle, Product, Category, Subcategory, Brand, State	Products in Subfocus (1 feature)	Quantity of products in subfocus.

Lag features were constructed since they are the ones that make the regression time-aware, that is, that extract properties that can sort the data. The reason for creating such features is that GB methods are not like traditional time series models. Therefore, to enable the models to make predictions, it is necessary to create this type of feature, as pointed out by [Bergmeir and Benítez 2012]. The lag futures can be seen in Table 3.

21 features were created considering the cycle used in prediction. It is crucial to emphasize that there is no time leap in the first time horizon prediction, meaning that the cycle to be predicted and the cycle used in prediction are identical. Table 4 shows the features built over cycle information, whereby three of them are used in the next cycle prediction and six are used in the other horizons.

Table 3. Lag features created considering the cycle used in prediction.

Original Features	Features Created	Description
Amount Sold, Cycle, Product, Category, Subcategory, Brand, State	Target 1 year lag (1 feature)	Amount sold in the year before.
Amount Sold, Cycle, Product, Category, Subcategory, Brand, State	Target 1 to 10 cycle lag (10 features)	Amount sold in 1, 2, 3, 4, 5, 6, 7, 8, 9 and 10 cycles before.
Amount Sold, Cycle, Product Code, Category, Subcategory, Brand, State	Target moving average of 5 and 10 cycles (2 features)	Amount sold moving average considering the previous 5 and 10 cycles.

Table 4. Features created considering cycle information.

Original Features	Features Created	Description
Cycle Start Date, Cycle End Date	Cycle Length (1 feature)	Cycle duration in days, usually 21 days. (Cycle used in prediction)
Cycle Start Date, Cycle End Date	Cycle Length (1 feature)	Cycle duration in days, usually 21 days. (Cycle to be predicted)
Cycle	Cycle Year, Cycle Number (2 features)	Information about the cycle used in prediction.
Cycle	Cycle Year, Cycle Number (2 features)	Information about the cycle to be predicted.

Other feature related to cycle's information that was used as input to the model is the cycle description. It was not showed in Table 4 because it is an original feature from the data. For 5 and 10 cycles ahead, there is also the description of the cycle to be predicted.

3.5. Data Modeling

This section describes three of the main steps showed in Figure 1: data preparation for multiple time horizons, hyperparameter optimization, training the models and testing them.

The time horizons predictions were made separately, so there are three datasets: one for predicting the next period, one for predicting 5 cycles ahead and one for predicting 10 cycles ahead. The next cycle prediction dataset had 31 input features, that being:

- 6 features regarding the granularity (cycle, state, channel, category, subcategory and brand);
- 1 original feature with the cycle description;
- 3 features created considering the cycle information (Table 4);
- 8 features created regarding the products, promotions and campaigns (Table 2);
- 13 lag features (Table 3).

The predictions for 5 and 10 cycles ahead had 45 input features each, that being:

- 6 features regarding the granularity (cycle, state, channel, category, subcategory and brand);
- 1 feature with the cycle to be predicted;
- 2 original features with the description of the cycle used in prediction and the cycle to be predicted;
- 6 features created considering the cycle information (Table 4) for both cycle used in prediction and cycle to be predicted;
- 8 features created regarding the products, promotions and campaigns considering the cycle used in prediction (Table 2);
- 8 features created regarding the products, promotions and campaigns considering the cycle to be predicted (the same shown in Table 2);
- 13 lag features (Table 3) considering the cycle used in prediction;
- 1 feature with the amount sold in the cycle used in prediction;

It is worth noting that all 14 additional features used for predicting future cycles will be available in the future, enabling their utilization in the models. At the time of making the prediction, all relevant information is known, including details such as the number of products in the portfolio, the specific products to be promoted, the corresponding discount amounts, information about cycle characteristics, and more.

For hyperparameter optimization, the data was divided into four training sets and four validation sets for each horizon. The validation sets had six cycles each with the most recent data as it is a forecasting problem. The training sets had different sizes for each horizon: 43 cycles for the next cycle predicted, 38 cycles for the 5 cycles ahead prediction and 33 cycles for the 10 cycles ahead prediction. Figure 2 shows how the data was split off.

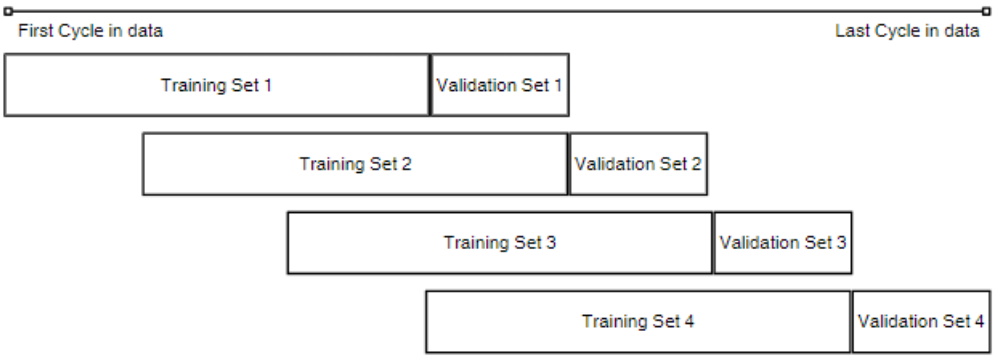

Figure 2. Training and Validation Sets for Hyperparameter Optimization.

In this step, optuna was used and the goal was to find the best hyperparameters that minimize the Weighted Absolute Percentage Error (WAPE). Its calculation is showed in

Equation (1). Since it is weighted, the forecast errors in products with more sales would be larger than in products with fewer sales. WAPE can range from 0% to any positive number in percent, the lower the better.

$$\frac{\sum_{t=1}^{n} |A_t - F_t|}{\sum_{t=1}^{n} |A_t|}, \tag{1}$$

where A_t and F_t are the observed and predicted values at instant t, respectively, and n is the maximum instant of time.

Five trials were made for each model and horizon. Table 5 shows the hyperparameters that were optimized and its ranges, as well as what each hyperparameter does. Unlike LightGBM and XGBoost, CatBoost uses a strategy called lossguide for the construction of trees, so it is not possible to specify the max_leaves hyperparameter for it.

Table 5. Range of hyperparameters to be optimized.

Hyperparameter	Description	Range
max_leaves / num_leaves	It limits the maximum amount of leaves a tree can have.	[20, 3000]
max_depth	Limits the maximum depth of decision trees.	[3, 12]
min_data_in_leaf / min_child_weight	Specifies the minimum number of samples required in a bin to be considered valid in histogram-based methods.	[100, 2000]
n_estimators / num_boost_rounds	Defines the maximum number of decision trees to be built.	[80, 300]
learning_rate	Controls the rate at which the model learns during training.	[0.01, 0.3]

Table 6 shows the final hyperparameters for CatBoost, Table 7 presents the hyperparameters used for LightGBM and Table 8 displays the XGBoost hyperparameters. The categorical features hyperparameter was also used in all models, so there was no need to encode the categories in the data preprocessing stage. All other hyperparameters were set as default.

Table 6. CatBoost hyperparameters.

Hyperparameter	Next cycle	5 cycles ahead	10 cycles ahead
max_depth	9	11	9
min_data_in_leaf	243	1462	1982
n_estimators	140	130	247
learning_rate	0.116	0.179	0.238
random_state	42	42	42

Table 7. LightGBM hyperparameters.

Hyperparameter	Next cycle	5 cycles ahead	10 cycles ahead
num_leaves	570	2275	2626
max_depth	6	10	9
min_data_in_leaf	707	937	205
n_estimators	229	185	219
learning_rate	0.190	0.221	0.164
random_state	42	42	42

Table 8. XGBoost hyperparameters.

Hyperparameter	Next cycle	5 cycles ahead	10 cycles ahead
max_leaves	1322	2382	2131
max_depth	8	7	9
min_child_weight	1291	313	323
num_boost_rounds	211	147	277
learning_rate	0.130	0.103	0.208
seed	42	42	42
tree_method	hist	hist	hist

After the hyperparameter optimization, the final models were trained using data from 2019 to 2022. Six cycles from 2023 were used as test set. The final training sets and test sets were separated considering the cycle to be predicted.

4. Results and Discussions

The metric chosen to evaluate and compare the three models results through the different time horizons was WAPE, as mentioned in Section 1.

Table 9 presents the WAPE for each channel and for each time horizon. Overall, it can be observed that the predictions for the store channel tend to be more accurate compared to direct sales. Among the three models, XGBoost consistently achieved the lowest error across all prediction horizons for the store channel. LightGBM also demonstrated a strong performance in predicting for the store channel, while CatBoost showed a significantly higher error when predicting for the 10 cycles horizon compared to the other two models.

When examining the errors for direct sales, LightGBM yielded superior results in predicting 10 cycles ahead. On the other hand, XGBoost surpassed the other models in the remaining two time horizons. Regarding the range of error, XGBoost showed almost the same variation for both channels: 2.97% (stores) and 2.96% (direct sales). CatBoost had the highest error range in both channels (7.76% for stores and 6.73% for direct sales), while LightGBM had the lower error range in stores (2.31%) and a value of 4.97% in direct sales. CatBoost had the least acceptable results taking into account not just the channels but also the different horizons.

Table 9. WAPE per channel.

Model	Channel	Next cycle	5 cycles ahead	10 cycles ahead
CatBoost	Stores	3.49%	3.12%	10.88%
LightGBM	Stores	4.22%	2.77%	5.08%
XGBoost	Stores	**3.07%**	**0.10%**	**2.62%**
CatBoost	Direct Sales	17.37%	17.46%	10.73%
LightGBM	Direct Sales	10.77%	13.80%	**8.83%**
XGBoost	Direct Sales	**8.75%**	**11.71%**	9.91%

To discuss the results for each cycle in each one of the three horizons, Figures 3, 4 and 5 should be investigated. Figure 3 shows the cycles predicted considering the next period ahead, Figure 4 shows the cycles to be predicted in 5 periods ahead and Figure 5 has the 10 cycles ahead that were predicted. For each test set, as discussed in Section 3.5, there are six cycles. The results from cycles 202301 (first cycle of 2023) to 202306 (sixth cycle of 2023) were evaluated.

For the closest prediction (Figure 3), all errors are below 18%. While LightGBM's and XGBoost's errors are all lower than 11%, CatBoost shows peaks in certain cycles. Both LightGBM and XGBoost had 3 out of 6 cycles with the lowest error, whereas CatBoost presents higher errors in all cycles. Once more, both LightGBM and XGBoost had better results.

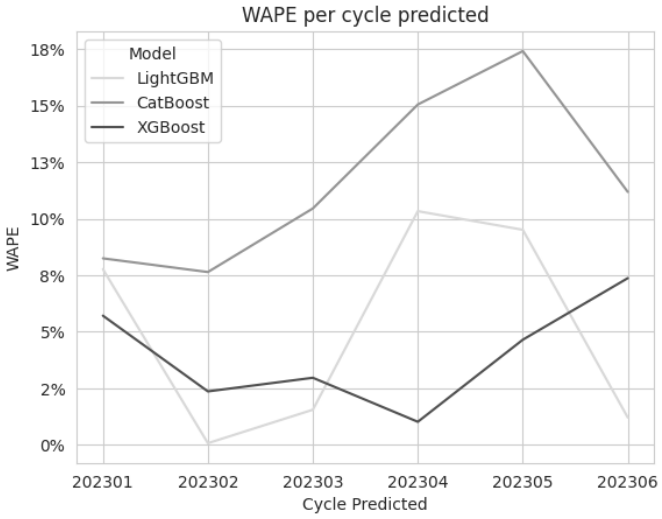

Figure 3. WAPE per cycle predicted: next cycle.

When predicting 5 cycles ahead, Figure 4 illustrates a notable peak in cycle 202304 across all models. Regarding the range of errors, when comparing the maximum and minimum WAPE for each model, all ranges fall between 13% and 18%. These results indicate no significant differences among the models in this particular scenario.

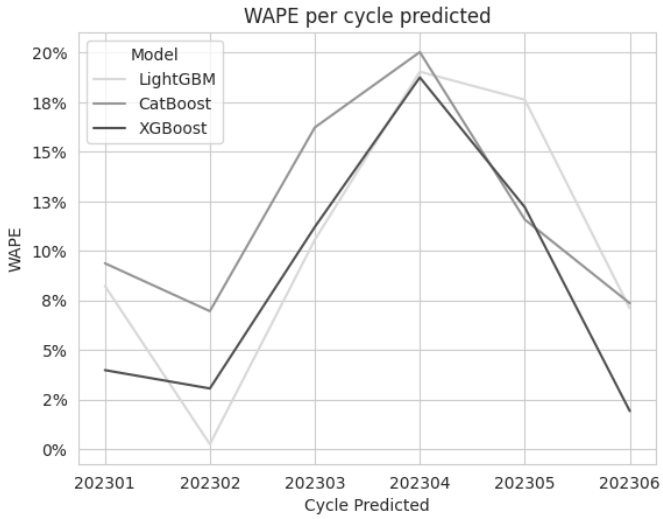

Figure 4. WAPE per cycle predicted: 5 cycles ahead.

For the long-term forecast, 10 cycles ahead were predicted (equivalent to eight months). It is possible to see the WAPE in the test set in Figure 5. All models shows a peak in the last cycle predicted, but LightGBM had a smaller error increase from one cycle to the next. CatBoost had the lower error in cycles 202303, 202304 and 202305. LightGBM was better in 202301 and 202306. XGBoost had a lower WAPE only in cycle 202302. In terms of error range, LightGBM has a range of 6.9%, followed by 10.1% in XGBoost and 11.2% in CatBoost.

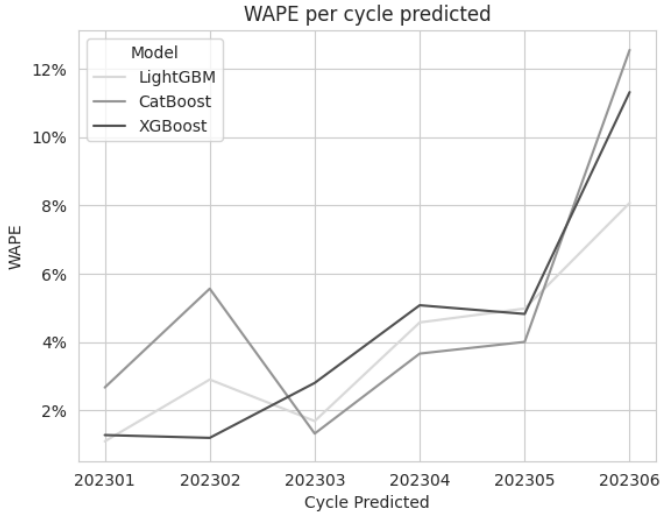

Figure 5. WAPE per cycle predicted: 10 cycles ahead.

As the goal of this paper is to compare three GB models over three time horizons, the model consistency considering the different scenarios is very important. The model that showed more stability between the analysis was LightGBM. XGBoost had a good overall performance as well, but Figure 5 shows that LightGBM is superior, especially because the biggest concern of this study is the long-term forecasting.

Since the feature engineering stage was a focus on the study, it is important to understand how the features had an impact on the most stable model performance. Figure 6 shows a feature importance chart for the LightGBM model predicting 10 cycles ahead. The features were divided between two big groups: original features and created features. Within the created features shown in the chart, three of them were lag features and the other three were some of the ones that were built upon information about products, promotions and campaigns. The names of the features will not be displayed due to confidentiality. It is evident that out of the top 10 most significant features for the model, 6 of them are derived from the applied methodology. This shows that the methodology used with a strong focus on feature engineering had a considerable impact on the results.

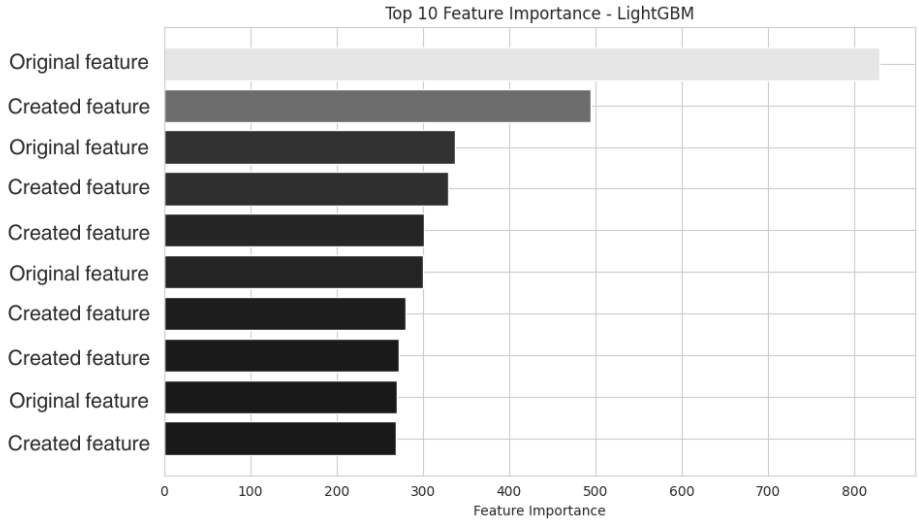

Figure 6. Feature Importance for LightGBM - 10 cycles ahead.

5. Conclusion

Demand forecasting is critical in various sectors and markets. For companies in retail, for example, knowing which and how many products are expected to be sold allow the business to be prepared to meet the customers needs. To forecast demand in a long-term scenario, it is even more challenging because of robustness [Zhou et al. 2022]. The aim of this research was to predict future demand using a real-world dataset in multiple time horizons, namely one period, five periods, and ten periods ahead. To achieve this goal, three Gradient Boosting algorithms (CatBoost, LightGBM, and XGBoost) were assessed, with a particular focus on feature engineering. The solution introduced in this paper indicated that LightGBM is able to provide good predictions in different steps ahead with consistency. XGBoost was also a remarkable alternative solution. In future work, enhancements could be derived from implementing the presented study across various retail datasets and also see how it behaves against other methodologies, such as ARIMA, Prophet and Deep Learning techniques.

References

Andrade, L. and Cunha, C. B. (2022). Disaggregated retail forecasting: A gradient boosting approach. *Available at SSRN 4129889*.

Baržić, M., Munitić, N.-F., Bronić, F., Jelić, L., and Lešić, V. (2022). Forecasting sales in retail with xgboost and iterated multi-step ahead method. In *2022 International Conference on Smart Systems and Technologies (SST)*, pages 153–158. IEEE.

Bergmeir, C. and Benítez, J. M. (2012). On the use of cross-validation for time series predictor evaluation. *Information Sciences*, 191:192–213.

Cox, J., Blackstone, J., Spencer, M., Production, A., and Society, I. C. (1995). *APICS Dictionary*. American Production and Inventory Control Society.

Hasan, M. R., Kabir, M. A., Shuvro, R. A., and Das, P. (2022). A comparative study on forecasting of retail sales. *arXiv preprint arXiv:2203.06848*.

Li, J. (2022). A feature engineering approach for tree-based machine learning sales forecast, optimized by a genetic algorithm based sales feature framework. In *2022 5th International Conference on Artificial Intelligence and Big Data (ICAIBD)*, pages 133–139. IEEE.

Lopes, G. (2022). The wisdom of crowds in forecasting at high-frequency for multiple time horizons: A case study of the brazilian retail sales. *Brazilian Review of Finance*, 20(2):77–115.

Provost, F. and Fawcett, T. (2013a). Data science and its relationship to big data and data-driven decision making. *Big data*, 1(1):51–59.

Provost, F. and Fawcett, T. (2013b). *Data Science for Business*. O'Reilly.

Schoenherr, T. and Speier-Pero, C. (2015). Data science, predictive analytics, and big data in supply chain management: Current state and future potential. *Journal of Business Logistics*, 36(1):120–132.

Seyedan, M. and Mafakheri, F. (2020). Predictive big data analytics for supply chain demand forecasting: methods, applications, and research opportunities. *Journal of Big Data*, 7(1):1–22.

Spiliotis, E. et al. (2022). Decision trees for time-series forecasting. *Foresight: The International Journal of Applied Forecasting*, (64):30–44.

Wu, H., Xu, J., Wang, J., and Long, M. (2021). Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in Neural Information Processing Systems*, 34:22419–22430.

Zhou, T., Zhu, J., Wang, X., Ma, Z., Wen, Q., Sun, L., and Jin, R. (2022). Treedrnet: A robust deep model for long term time series forecasting. *arXiv preprint arXiv:2206.12106*.