



Degree in Data Science and Engineering

Data Science Project 24/25-S1

“Sales forecasting with machine learning”

Ana Real Terradez (100451138)

Olga Bonachera del Pozo (100472782)

Francisco José Landa Ortega (100483174)

Alejo González García (100454351)

January, 2025

Table of Contents

Acknowledgements	2
Abstract	2
Introduction	3
State of the art	3
Problem Understanding: The M5 Kaggle Competition	4
Preprocessing	5
EDA	6
Models	7
Outliers	20
Problem Understanding: Sandoz's Dataset	22
EDA	22
Components Classification	25
Outliers	28
Models	33
Final Model Structure	44
Results & Discussion	44
Conclusions	45
References	47

Acknowledgements

We would like to begin this report by thanking David Méndez, Juan Pérez-Prim and Carlos Salle, our links with Sandoz, for tutoring and coaching us during the develop of this Data Science project, also, to our professors Javier Nogales and Fernando Díaz; and Carlos III de Madrid University for preparing the subject. It has been really encouraging for us students to have a small contact with real-world data.

Abstract

Time series are a sequence of data points collected at regular time intervals. Unlike other forms of data, time series explicitly considers the temporal order of observations, making it ideal for analyzing patterns over time. The key components of the time series are the trend (long-term upward or downward movement in the data), seasonality (regular and repeating patterns or cycles over a fixed period, such as monthly or yearly), cyclic patterns (fluctuations that are not fixed to a specific time period but occur over longer horizons) and noise (random variation or irregular fluctuations that cannot be explained by trends, seasonality, or cycles).

Time series forecasting is the process of using past observations to predict future values. This project aims to analyze and develop an appropriate analysis of Time Series with cutting edge ML algorithms.

Introduction

Our team has worked in a collaboration between Carlos III University of Madrid and Sandoz on a project called ‘Sales forecasting with machine learning’.

Sandoz is a Swiss company that focuses on generic and biosimilar medicine. The aim of the project is to optimize the number of medicaments that the company produces and sends to each country, so that there is no surplus or shortage.

For this we need to correctly understand Time Series, as well as preprocessing and modeling the data accordingly. In order to develop models we will focus on the “Darts” library framework to predict the future sales values.

As we lacked knowledge in Time Series, after some deep research on the topic we needed to test our abilities in order to prepare for the Sandoz dataset, which was more complex. Consequently, we applied these recently learnt techniques on to the “M5 Kaggle Competition” dataset to further understand the recently learnt concepts. After that we analyzed the Sandoz dataset and made the forecast.

State of the art

Sales forecasting in the pharmaceutical industry has evolved significantly, with advanced methodologies enhancing accuracy and decision-making. Traditional statistical models like ARIMA (AutoRegressive Integrated Moving Average) have been foundational in predicting drug sales. However, the integration of machine learning and deep learning techniques together with significantly more powerful machines and graphic cards, has led to notable improvements and more complex models.

Deep Neural Networks (DNNs) such as Multilayer Perceptron (MLP), Convolutional Neural Networks (CNNs), and Long Short-Term Memory (LSTM) networks have been applied to forecast drug sales and pricing. Among these, LSTM networks have demonstrated superior performance, effectively capturing complex temporal patterns in sales data. **Hybrid Models** combine traditional statistical methods with machine learning approaches and have yielded promising results. For instance, integrating ARIMA with LSTM models leverages the strengths of both, enhancing forecasting accuracy.

Machine Learning Algorithms like XGBoost have been employed to analyze large datasets, improving the precision of sales forecasts. These models can handle various factors influencing sales, such as seasonality and market dynamics.

Sales forecasting enhances predictive accuracy, supporting better inventory management, strategic planning, and pricing strategies within the industry.

A 2024 study compared traditional forecasting methods with advanced machine learning techniques, highlighting the effectiveness of models like Facebook Prophet and LSTM Neural Networks in predicting pharmaceutical sales. [1]

Research has demonstrated that models optimized using algorithms like Particle Swarm Optimization (PSO) can achieve prediction accuracies exceeding 96%, outperforming traditional neural network models. [2]

Problem Understanding: The M5 Kaggle Competition

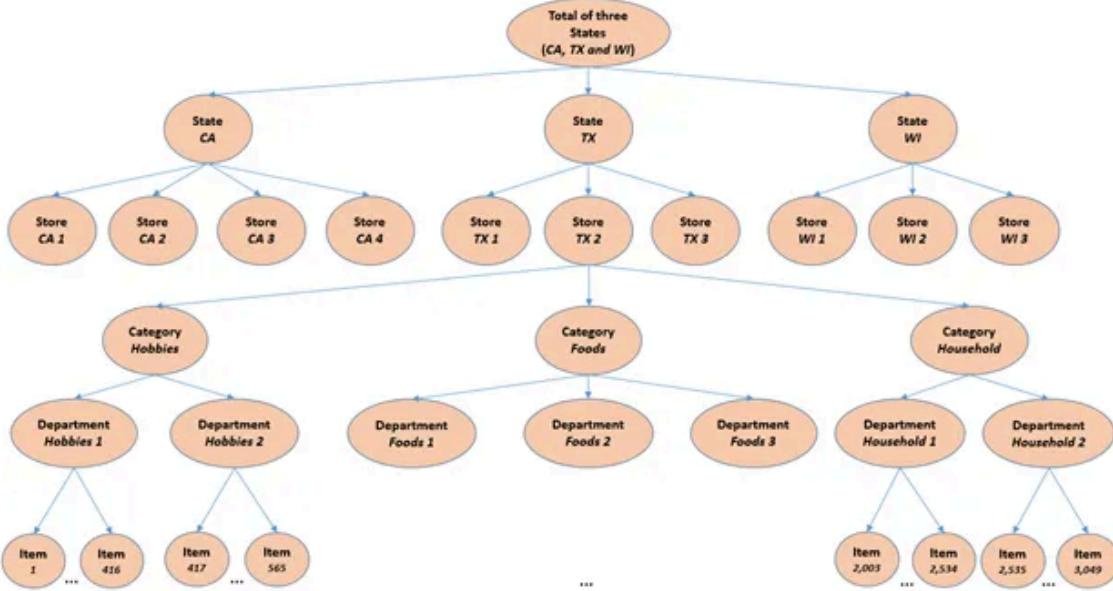
A question that arises is: *Why have we been working on the M5 Dataset?* Well the answer is simple: to practice and enhance our knowledge.

The idea that Sandoz proposed was to firstly work with this dataset, so that we could learn and test the Forecasting techniques in a controlled environment, where we knew what results to expect, and where we could search for examples and learn a lot before applying all of the techniques and knowledge acquired to the Sandoz dataset.

The M5 Dataset contains Walmart's sales data across different states in the US. It's divided into different tables:

- Sales: it has the daily sales of approximately 30,000 products across three states, a total of ten stores, divided in three categories over a period of 5+ years.
- Calendar: information about dates, including holidays and promotional events, which can influence sales. It helps to identify spikes or drops in sales to find patterns related to a specific event.
- Sell: we find the prices of each of the products sold across different stores and over time. With this we understand price fluctuations, which is crucial for our task since price changes can affect the demand of a product.

The dataset is hierarchical, meaning we can forecast sales at different levels of aggregation, such as total sales for all products, sales per state, or sales per store and product combination. This structure allows us to forecast for both granular and global, making the M5 dataset good for testing different models.



In this plot we can see the aggregation of the M5 dataset. We have 3 different states, California, Texas and Wisconsin; and each state has several stores where they sell either Hobbies, Foods or HouseHold. (And each of them has several departments that are not relevant in our study).

Preprocessing

This dataset contains information ranging from 2011 to 2016, initially comprising 58 million rows in the training set and 59 million in the test set. To handle this large volume of data on personal computers with limited RAM (16 GB), we reduced the dataset to 43.7 million rows and 9 columns. This was achieved by excluding data from 2015 in the train set and 2016 in the test set, focusing on data from 2011 to 2014 for model training. Additionally, we transformed and saved the datasets as `parquet` files to optimize storage and processing efficiency, enabling smoother execution on standard hardware.

Given the computational intensity, we utilized CUDA dependencies to leverage GPUs and reduce training time. Despite these challenges, the smaller size of the actual Sandoz dataset is expected to minimize potential issues with resource limitations, ensuring the feasibility of this approach for real-world application.

EDA

We aggregated sales data at monthly, yearly, and daily levels to identify trends, seasonality, and patterns across Walmart's 10 stores. Interactive visualizations revealed consistent trends, such as zero sales on December 25th (Walmart closure) and significant fluctuations around Black Friday. A heatmap highlighted sales distribution across stores, while time series plots showed aggregated trends by state, with notable growth in Wisconsin's WI_2 store from 2012, possibly due to expanded offerings.

Key insights include weekend sales spikes, higher sales in summer, and lower sales in winter, reflecting strong seasonality. These findings, along with event-driven fluctuations, will guide our analysis of Sandoz's dataset to account for similar patterns and behaviors.



On this plot we can see the different trends for each of the stores. On Store Wisconsin_2 there is a huge increase in sales in April 2012, same case for the end of 2012 in Wisconsin_1 and most of them oscillate and follow a similar pattern during time, thanks to the large amount of points that we have, it will be probably feasible to predict accurately most of the future sales.



In this other plot we can see the monthly sales distribution per year for all the shops. It's an interactive graph so we can see the specific values. We also have a dot line which represents the mean between all years. We can see that the overall sales increased from 2011 to 2012 and from 2012 to 2013, but 2013 and 2014 are more similar.

Models

For this dataset, we evaluated several models to forecast sales data effectively, considering the seasonality, trends, and potential anomalies inherent in the dataset. The primary goal was to identify models that could provide accurate predictions while balancing complexity and computational efficiency. Each model was assessed based on some error metrics, Mean Absolute Error (**MAE**), Root Mean Squared Error (**RMSE**), and Mean Absolute Percentage Error (**MAPE**).

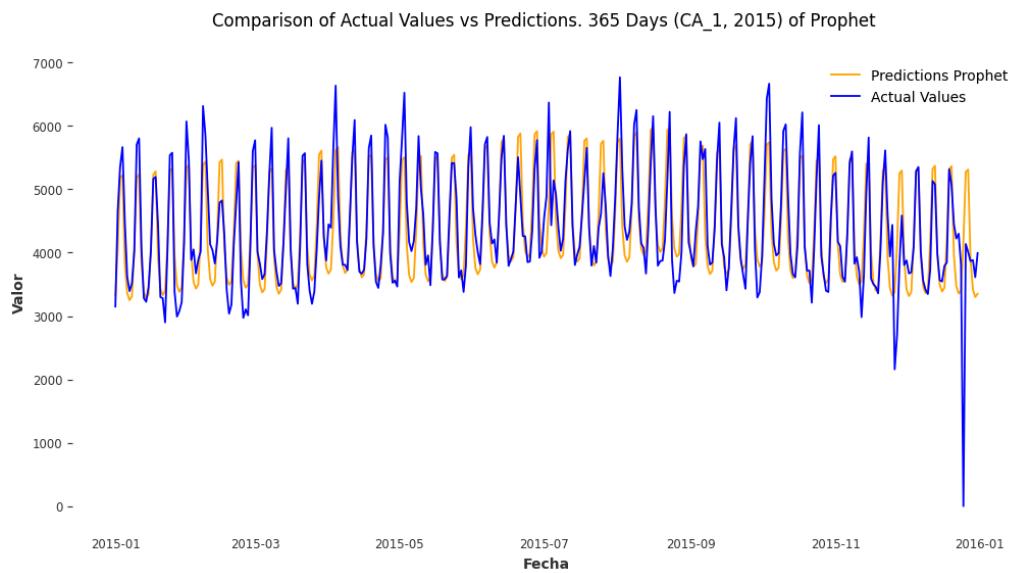
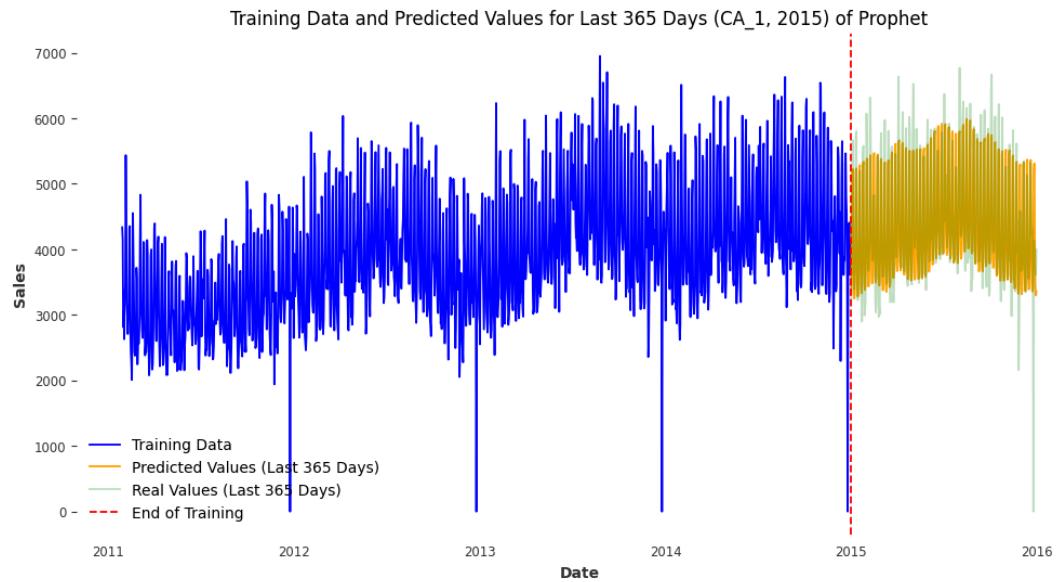
One of the first models we implemented was **Prophet**, a time-series forecasting tool developed by Facebook in 2017, it combines trends, seasonality and holidays (key aspects in the Time Series analysis), following this expression:

$$y(t) = g(t) + s(t) + h(t) + \varepsilon_t$$

Where $g(t)$ is the trend function, which models non-periodic changes in the value of the time series; $s(t)$ models the periodic changes (weekly, yearly...); $h(t)$ represents special dates like holidays or potentially irregular schedules over one or more days; and ε_t , the error term that will accommodate those terms not accommodated by the model. Deeper explanation on this

expression is far from the scope of this project, but find referenced the paper where Prophet was proposed for further analysis on it. [\[3\]](#)

Prophet is particularly suited for datasets with strong seasonal components and allows for the incorporation of holidays and events as additional predictors. For our dataset, we employed a multiplicative seasonality mode to account for variations that scaled with the overall trend. Prophet demonstrated strong performance in capturing yearly sales patterns, producing accurate forecasts while remaining interpretable.



MAE: 300.90739685517315

RMSE: 407.9821581391165

MAPE: 6.8482688100973315

We also explored **Exponential Smoothing** (ES), a classical forecasting technique that models level, trend, and seasonal components by applying exponentially decreasing weights to past observations, emphasizing recent data points.

ES has several variants for different types of time series:

- Simple Exponential Smoothing (SES) is suited for data with no clear trend or seasonality. Follows this equation, [\[4\]](#)

$$s_t = \alpha x_t + (1 - \alpha)s_{t-1}$$

Being s_t the smoothed value at time t , x_t the observed value at time t and α the smoothing parameter for the level (remember that $0 < \alpha < 1$), it controls the weight given to both, current and previous observations. Higher values assign more weight to the current observation, values close to 0 assign more weight to past smoothed values.

- Holt's Linear Trend Model captures data with linear trends but no seasonal patterns. Also called Double Exponential Smoothing, follows this expression, [\[5\]](#)

$$s_t = \alpha x_t + (1 - \alpha)(s_{t-1} + b_{t-1})$$

$$b_t = \beta(s_t - s_{t-1}) + (1 - \beta)b_{t-1}$$

Where s_t , x_t and α are the same as before, b_t is the slope and best estimate of the trend at time t ; and β ($0 < \beta < 1$) is the smoothing parameter for the trend.

- Holt-Winters (Additive or Multiplicative) is effective for data with both trend and seasonality, allowing flexibility in modeling seasonal variations. Now, the mathematical formulation adds a third component to the Exponential smoothing. The formulation [\[6\]](#)

$$s_t = \alpha(x_t - c_{t-m}) + (1 - \alpha)(s_{t-1} + b_{t-1})$$

$$b_t = \beta(s_t - s_{t-1}) + (1 - \beta)b_{t-1}$$

$$c_t = \gamma(x_t - s_t) + (1 - \gamma)c_{t-m}$$

(additive)

$$s_t = \alpha \frac{x_t}{c_{t-m}} + (1 - \alpha)(s_{t-1} + b_{t-1})$$

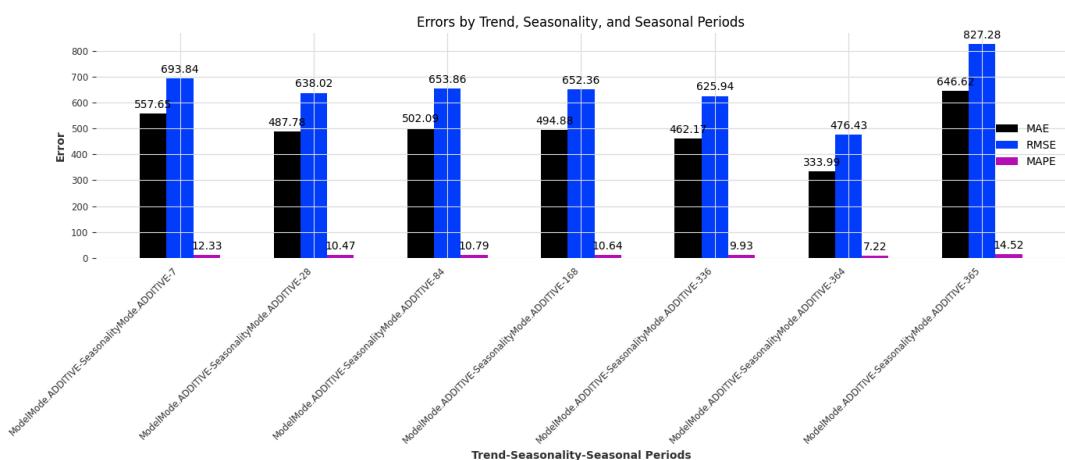
$$b_t = \beta(s_t - s_{t-1}) + (1 - \beta)b_{t-1}$$

$$c_t = \gamma \frac{x_t}{s_t} + (1 - \gamma)c_{t-m}$$

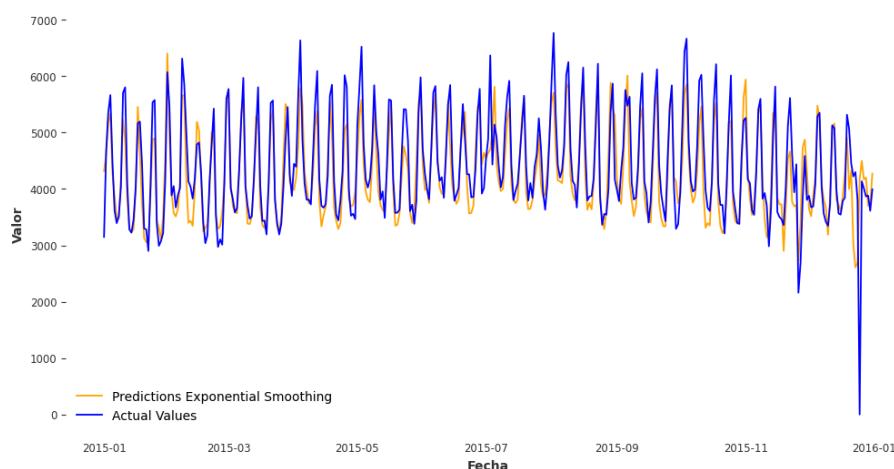
(multiplicative)

Where c_t is the seasonal component at time t, γ a smoothing parameter for the seasonality between 0 and 1 and m the seasonal period.

In our analysis, we focused on the Holt-Winters method, leveraging its ability to handle the dataset's seasonal and trending characteristics. By tuning parameters such as seasonal periods and smoothing coefficients, the model provided robust forecasts, smoothing out noise while accurately capturing underlying patterns. The simplicity and speed of Exponential Smoothing made it a reliable choice for this forecasting task.



Comparison of Actual Values vs Predictions. 364 Days (CA_1, 2015) of Exponential Smoothing

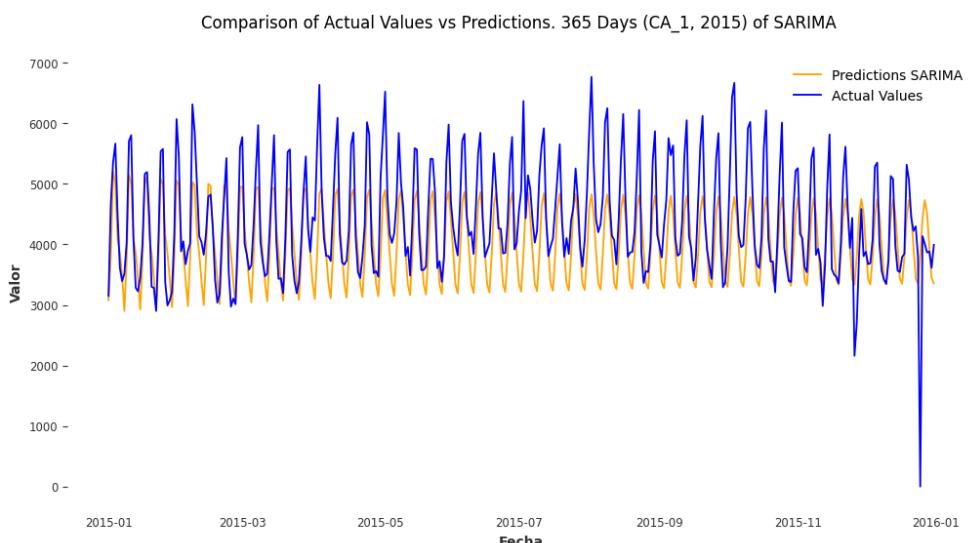
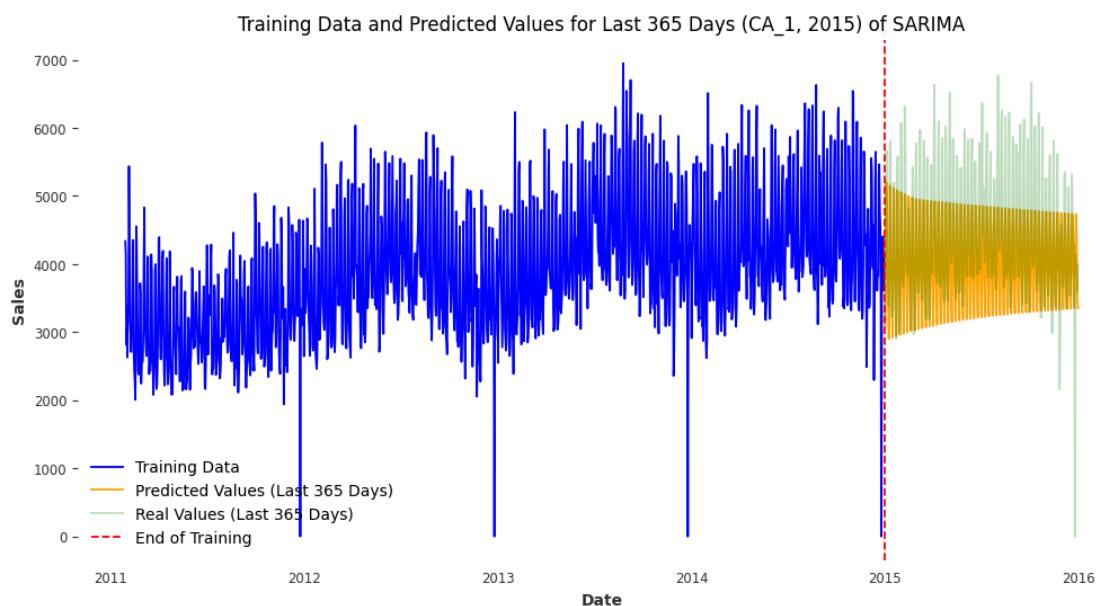


MAE: 333.99458554826396

RMSE: 476.42663498932313

MAPE: 7.216084430513189

To address both seasonal and non-seasonal patterns, we implemented the **SARIMA** (Seasonal AutoRegressive Integrated Moving Average) model, an extension of the **ARIMA** model. SARIMA is specifically designed to handle complex time series data with seasonal cycles. Using AutoARIMA, we automated the parameter selection process, searching across combinations of (p, d, q) for non-seasonal components and (P, D, Q, m) for seasonal components. This optimization is guided by performance metrics such as Akaike Information Criterion (AIC), which ensures the model balances accuracy and simplicity.



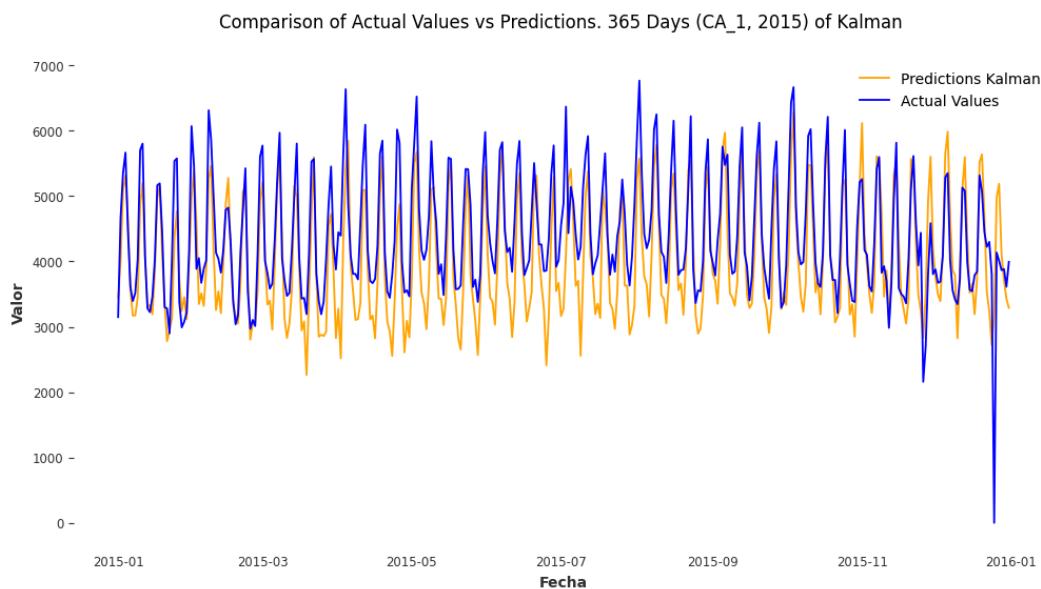
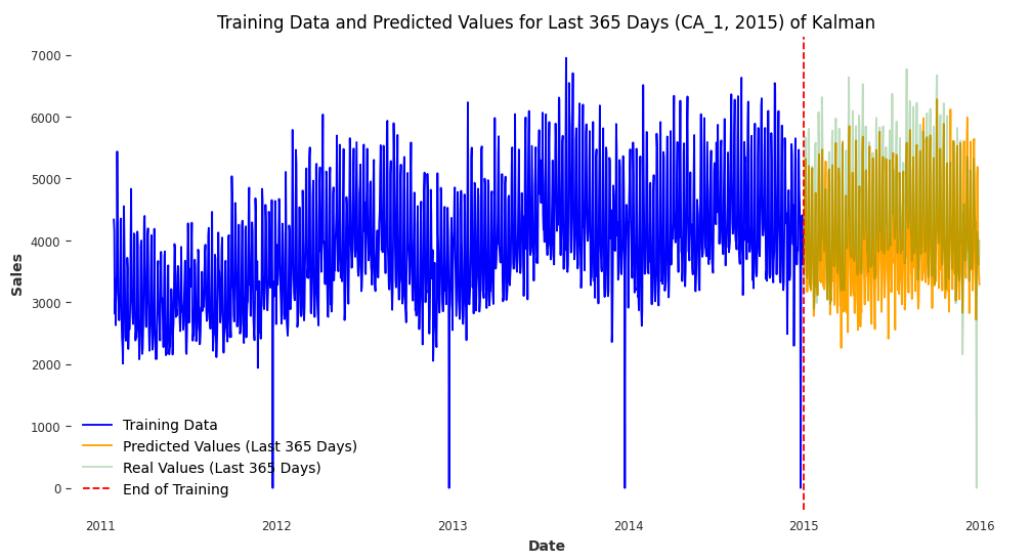
MAE: 550.5024345085926

RMSE: 706.6511876843424

MAPE: 11.731843172469917

However, while SARIMA captured general trends and seasonality, its performance fell short compared to models like Prophet and LightGBM. The model struggled with anomalies and sharp changes, leading to higher deviations in test forecasts, especially during irregular periods. Although SARIMA followed overall trends, its predictions lacked precision, highlighting its limitations for datasets with complex patterns. This demonstrates that advanced models are not always the best choice for real-world data with significant variability.

Additionally, we evaluated the **Kalman Filter**, a probabilistic model used for forecasting and smoothing. It offered a flexible framework for capturing dynamic time-series patterns but required careful tuning to avoid overfitting. Despite its theoretical strengths, the Kalman model's performance was comparable to simpler models, highlighting the trade-off between complexity and effectiveness.



MAE: 529.2371448810622

RMSE: 660.6848774506659

MAPE: 11.913529894478964

For more complex patterns, we applied **machine learning models**, including **XGBoost** and **LightGBM**.

On the one hand, XGBoost or Extreme Gradient Boosting utilizes a gradient boosting framework that combines multiple decision trees to optimize a defined objective function. The objective function consists of two components: a loss function $l(y_i, \hat{y}_i)$ and a regularization term $\Omega(f_k)$, which controls model complexity. Mathematically, this is expressed as:

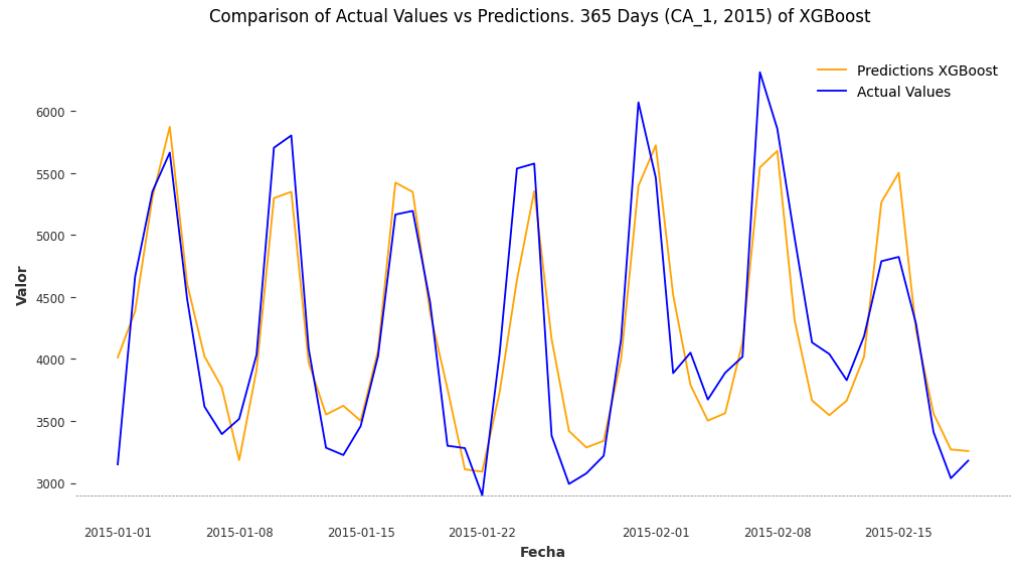
$$\mathcal{L}(\phi) = \sum_{i=1}^n l(y_i, \hat{y}_i) + \sum_{k=1}^T \Omega(f_k),$$

where $\Omega(f_k) = \gamma T + \frac{1}{2} \lambda \sum_{j=1}^T w_j^2$; again find more details on the decomposition of these factors in the related paper. [\[7\]](#)

XGBoost is built on Decision Trees, Ensemble Learning and Gradient Boosting. This combination of models and ML procedures enables XGBoost to outperform any other models when facing Structured Data. [enlace a nvidia]

On the other hand, LightGBM, or Light Gradient Boosting Machine, despite being optimized for high performance in distributed systems, also relies on Decision Trees for prediction tasks. It uses a histogram-based approach in order to build the DTs leafwise. Find more information on the mathematical basis and structure for LightGBM, it will help to understand the strengths and weaknesses of this algorithm. [\[8\]](#)

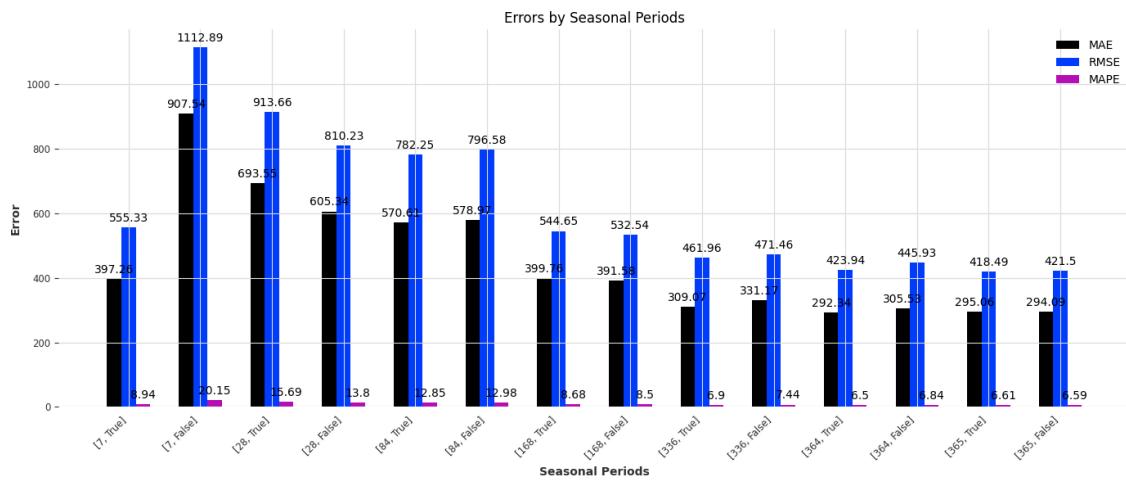
These models excel in handling non-linear relationships and large datasets. By transforming the time series into a supervised learning format with lagged features, we enabled these models to predict future sales based on historical patterns. Hyperparameter tuning was performed using grid search, and both models demonstrated strong predictive accuracy as well as high demand for computational resources, with LightGBM slightly outperforming XGBoost in certain scenarios.



MAE: 299.1251605308219

RMSE: 420.0005051497648

MAPE: 6.7428811795190295



This table corresponds to the errors obtained for LightGBM being '*True*' with scaling and '*False*' without it. We can see that if we consider the last year, the last 365 days as the lags for the model, we are going to be getting the best results. If we take 1 week, 4 (1 month), or multiples of months the results are also good, but not as a complete year.

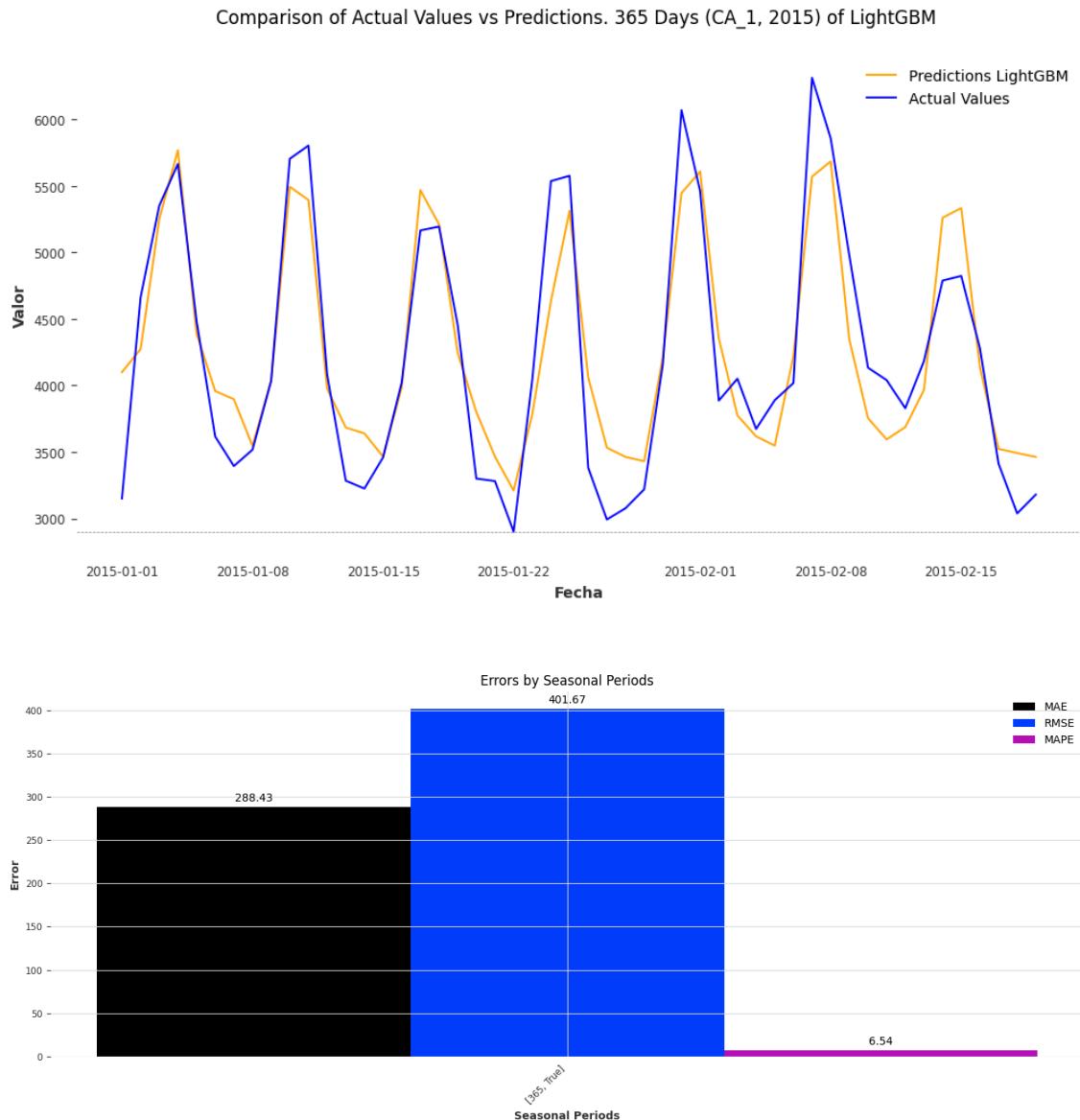
MAE: 292.34056861841873

RMSE: 423.94366485566667

MAPE: 6.503331730673157

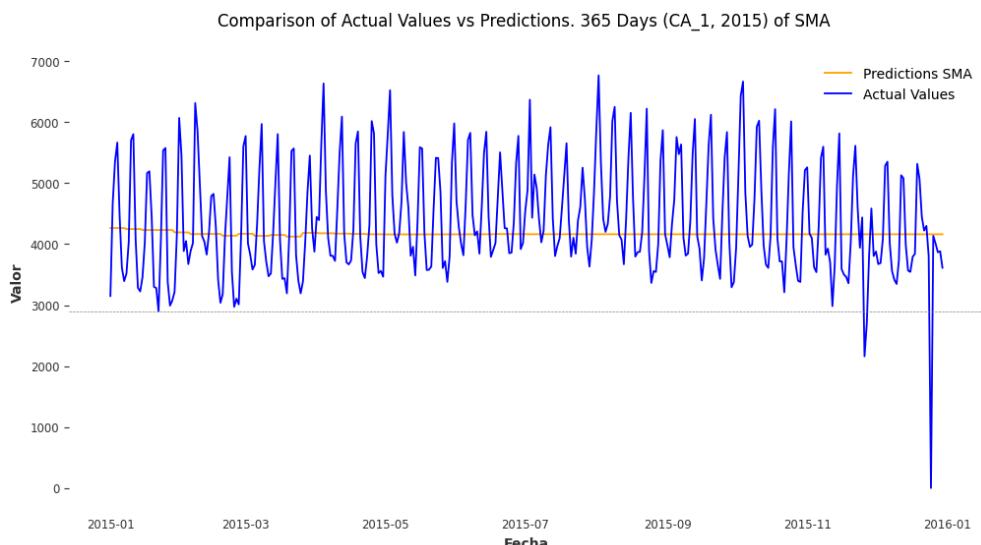
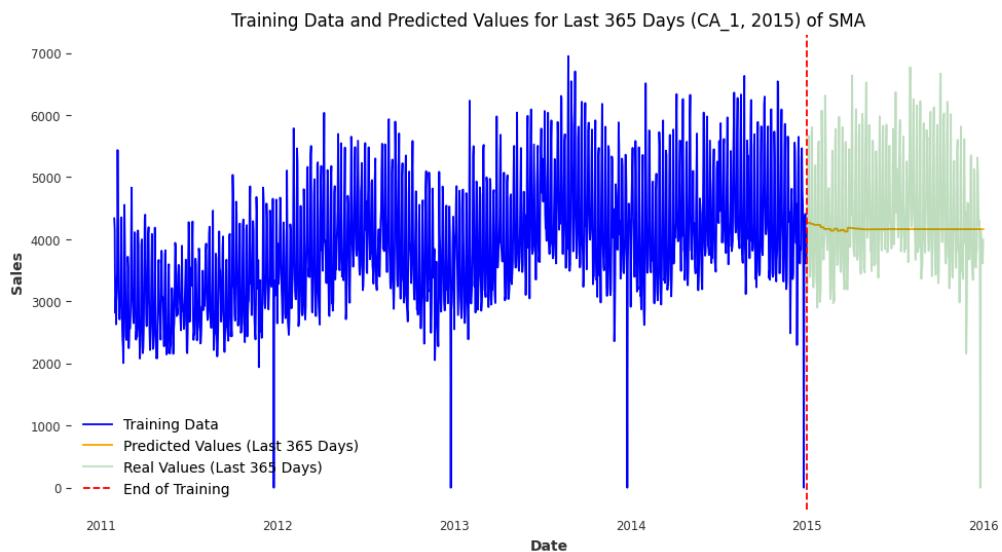
We also explored **Random Forest**, a robust ensemble learning technique that builds multiple decision trees and aggregates their results to make predictions. Random Forest is well-suited for capturing non-linear relationships in time series data, especially when transformed into a supervised learning format with lagged features.

In our analysis, we tested Random Forest by varying key hyperparameters, such as the number of trees and the maximum depth of each tree, as well as by experimenting with scaling and different lag structures. While the model demonstrated strong performance in capturing non-linear dependencies and trends, it struggled to account for the seasonal components of the dataset as effectively as other models like LightGBM or Prophet. However, Random Forest remained competitive, particularly when combined with scaled features and carefully chosen lagged variables, making it a valuable tool for forecasting in scenarios with less pronounced seasonality.



Finally, we included baseline models as benchmarks to compare the performance of more advanced techniques. Among these, the **Naive Forecast** and **Seasonal Naive Forecast** provided straightforward approaches for forecasting. The Naive Forecast simply carries forward the last observed value, while the Seasonal Naive Forecast repeats values from the same period in the previous year. Surprisingly, the Seasonal Naive model performed competitively, highlighting the importance of seasonality in our dataset and serving as a useful reference for evaluating other models.

In addition, we implemented the **Simple Moving Average (SMA)**, which calculates the average of sales over a sliding window of recent periods. This approach smooths out short-term fluctuations while retaining longer-term trends. By varying the window size, we tailored SMA to better reflect the dataset's characteristics. While SMA is less sophisticated than other models, it provides a reliable baseline for comparison, especially for data with minimal irregularities.

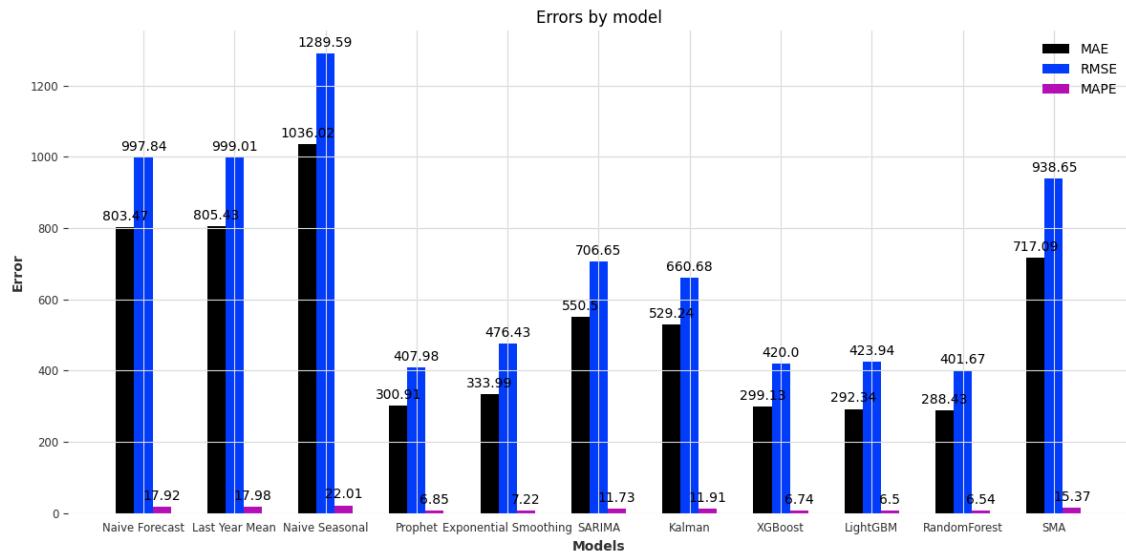


MAE: 717.0912661745542

RMSE: 938.6482779501315

MAPE: 15.371368018429244

By looking at both plots, we can clearly see how SMA works. After a couple of iterations it smooths and just takes as prediction the average value, that we can see in the flat yellow line. This method is quite useful for situations where the series is unpredictable and does not follow any kind of trend, being random. When used in an ensemble, it provides a high robustness.



After evaluating all models, we got this table with the best metrics errors for each model to compare them.

If we compare the baseline models (Naive Forecast, Last Year Mean, Naive Seasonal, and Simple Moving Average) with the more advanced models, it's evident that the latter consistently outperformed the baselines across all error metrics (MAE, RMSE, and MAPE). Among the baseline models, Simple Moving Average (SMA) achieved the best performance. However, even SMA fell short compared to the more sophisticated models.

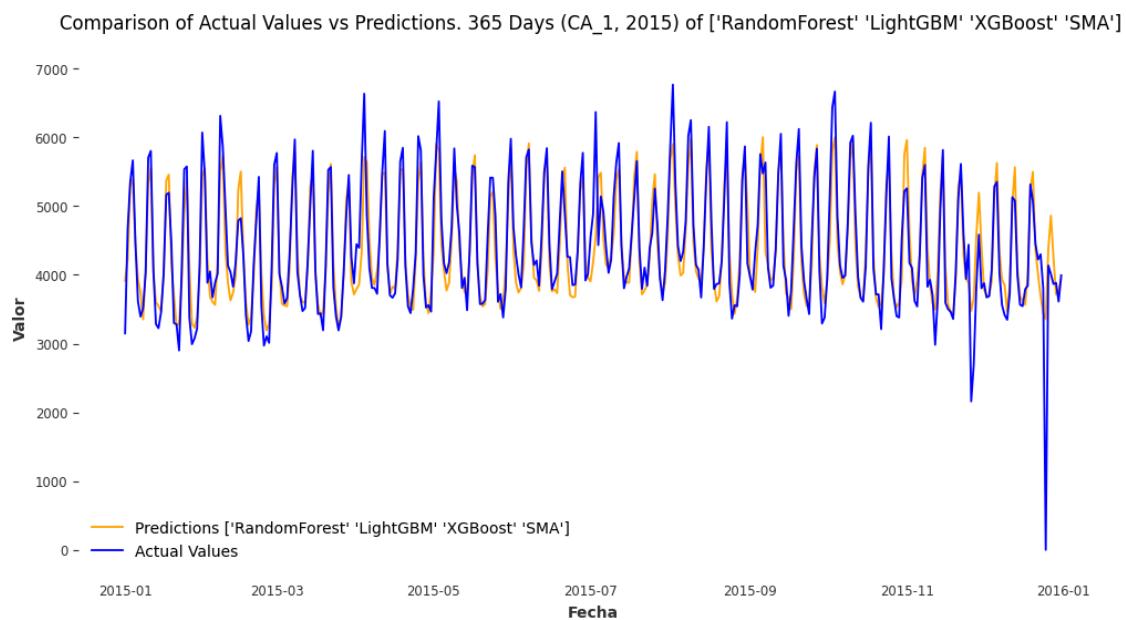
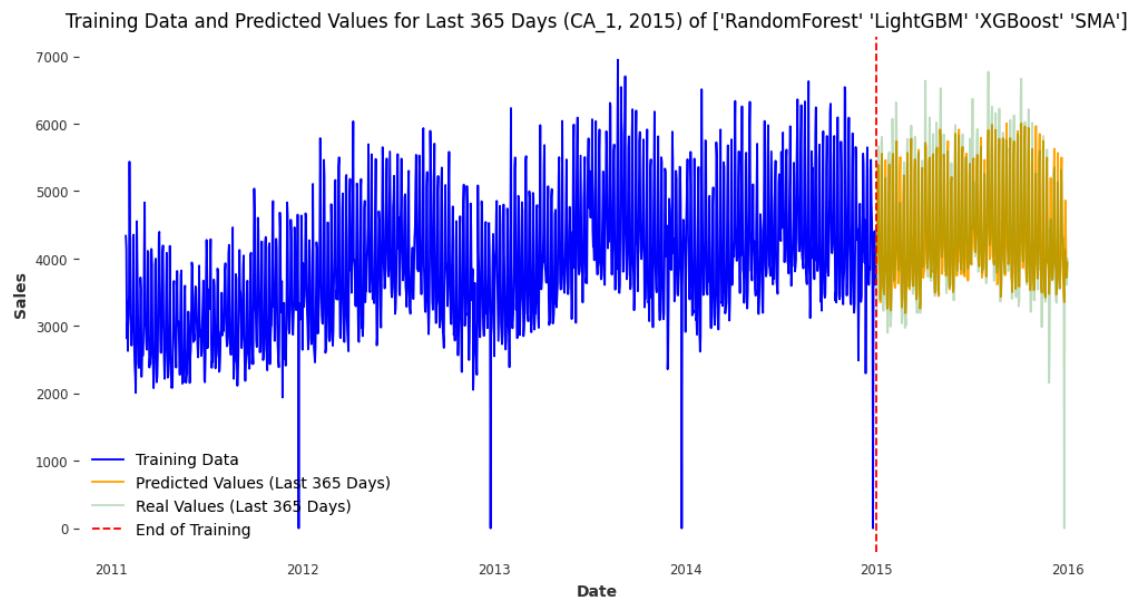
Based on the results, we decided to proceed with the four best-performing models: **Exponential Smoothing**, **XGBoost**, **LightGBM**, and **Random Forest**. These models demonstrated significantly lower errors and greater robustness, making them strong candidates for use in forecasting on the Sandoz dataset. Their ability to capture trends, seasonality, and complex patterns in the data further justifies their inclusion in the final ensemble approach.

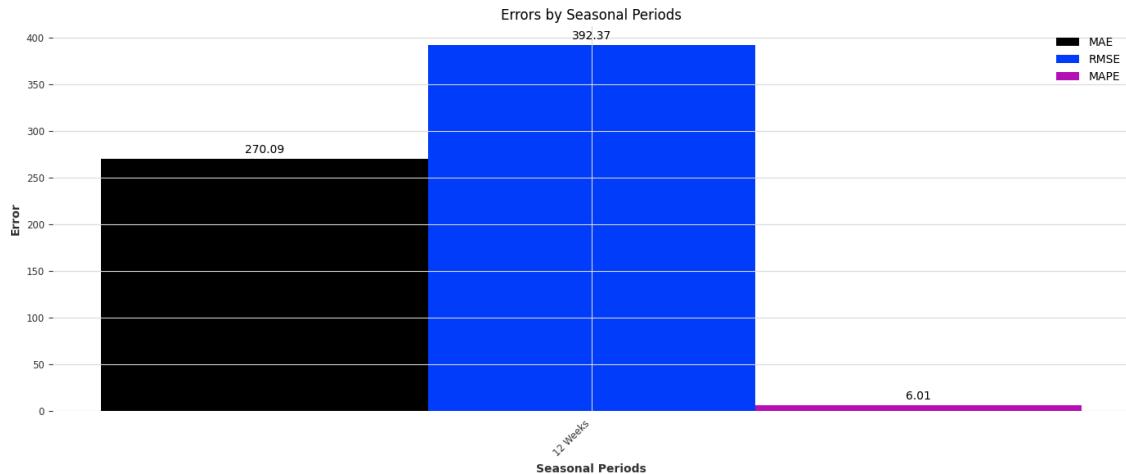
After identifying the best-performing models, we proceeded to create an ensemble using the **top 3 advanced models** (Exponential Smoothing, XGBoost, and LightGBM) combined with the **Simple Moving Average (SMA)**. The inclusion of SMA in the ensemble added an additional

layer of robustness by integrating a simple baseline model that effectively captures general trends and seasonality, avoiding overfitting.

The ensemble approach leverages the strengths of each individual model: the ability of advanced models to handle complex patterns and anomalies, and the simplicity and stability of SMA. This combination aimed to reduce individual model biases, improve forecast accuracy, and increase robustness against variability in the data.

By blending the predictions of these models, the ensemble produced forecasts that consistently outperformed any single model, validating its efficacy as a reliable forecasting strategy for the Sandoz dataset.





We compare all the errors again, now with the ensemble too.



We can see here that to ensemble the models is the best option in this case.

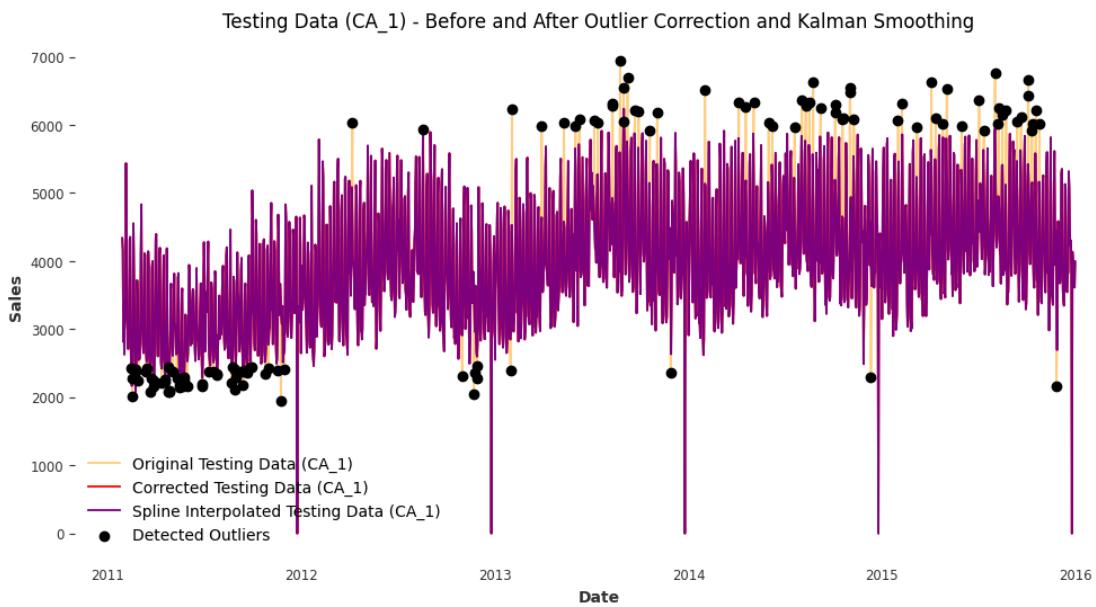
MAE: 270.0902131

RMSE: 392.3701812

MAPE: 6.01321

Outliers

Before moving on to the Sandoz dataset, we tried an initial approach to outlier detection and correction, as depicted in the figure below, we employed a straightforward threshold-based method. This method involved identifying outliers as data points that fell beyond a predefined range of acceptable values. The detected anomalies were then replaced with interpolated values to smooth the data. While this approach allowed for a quick identification and correction process, it suffered from several limitations.



The title in this graph is not correct since we tried different methods (Kalman, isolation forest, z-score and IQR) but we didn't change the title, this one corresponds to isolation forest.

One major limitation of this approach was its lack of consideration for the context within the time series. By using a simple threshold, the method often misidentified meaningful spikes, like those caused by promotions or seasonal trends, as outliers. This resulted in the potential loss of valuable information tied to natural sales patterns.

Another issue was its reliance on fixed rules, which made it inflexible when the data changed. Sales data often shows dynamic patterns, and this rigid approach struggled to handle real-world complexities, especially when outliers followed the overall trend or seasonality rather than deviating sharply from it.

Because of these limitations, we chose not to use this method in our main analysis. Instead, we focused on techniques that incorporate domain knowledge and are better suited to account for the structure and characteristics of time series data. These alternative methods preserved

meaningful patterns while handling anomalies more effectively, we applied it directly to the Sandoz dataset, which will be explained later.

Problem Understanding: Sandoz's Dataset

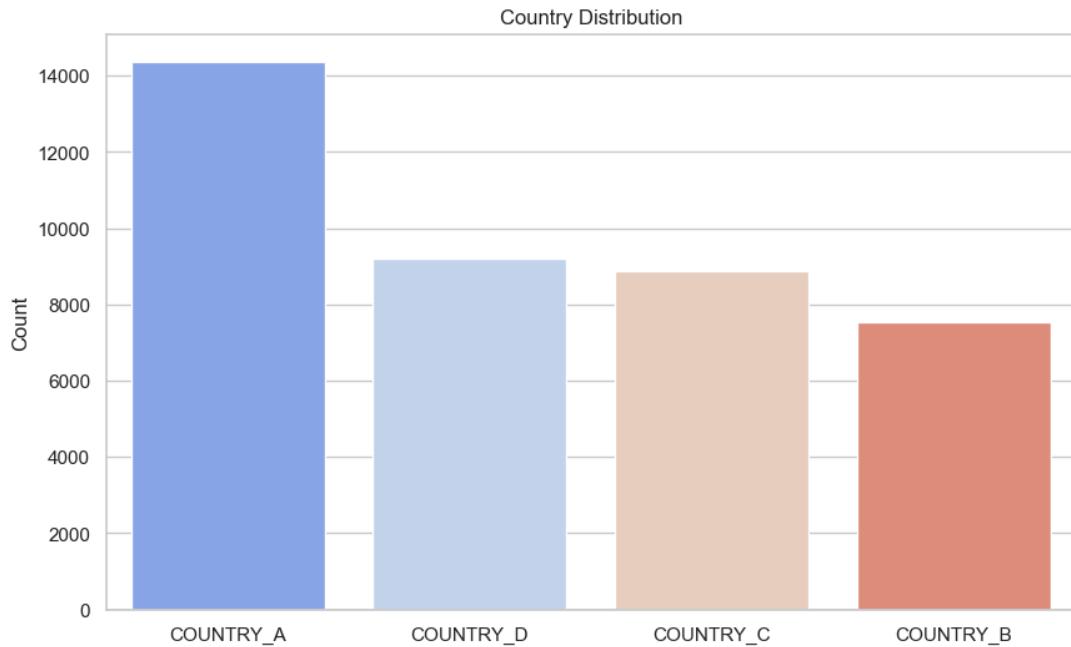
Now the focus is on the provided Sandoz's Dataset, the target of our work.

As previously mentioned, Sandoz belongs to the Novartis Group and it is focused on generics and biosimilar medicine. It includes information about their sales from 2013 to 2024.

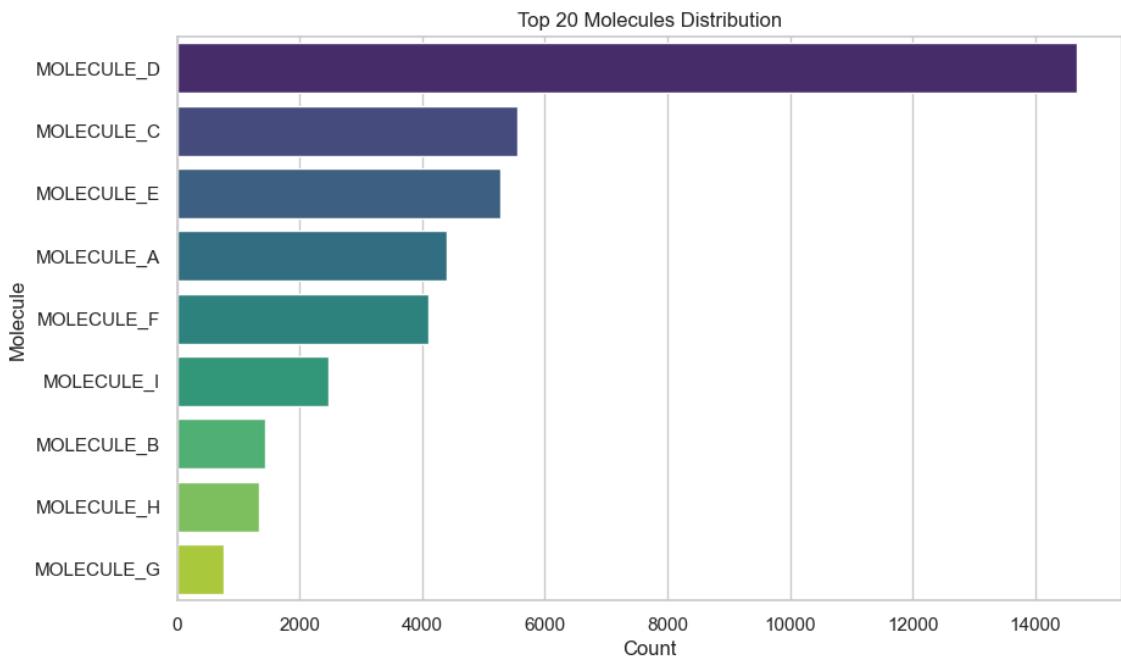
The dataset consists of 40008 rows of data with 6 columns which specify the name of the country, the main molecule, the dosage form, strength, the amount sold and a time stamp. Here we have a small overview of the data.

	country_name	molecule	dosage_form	strength	standard_units	date
0	COUNTRY_A	MOLECULE_A	DOSAGE_FORM_1	STRENGTH_1	3.354448e+07	2013-02-01
1	COUNTRY_B	MOLECULE_A	DOSAGE_FORM_1	STRENGTH_2	6.284162e+06	2013-02-01
2	COUNTRY_B	MOLECULE_A	DOSAGE_FORM_1	STRENGTH_3	6.968987e+06	2013-02-01
3	COUNTRY_A	MOLECULE_A	DOSAGE_FORM_2	STRENGTH_4	1.979899e+03	2013-02-01
4	COUNTRY_C	MOLECULE_A	DOSAGE_FORM_2	STRENGTH_4	1.332210e+06	2013-02-01

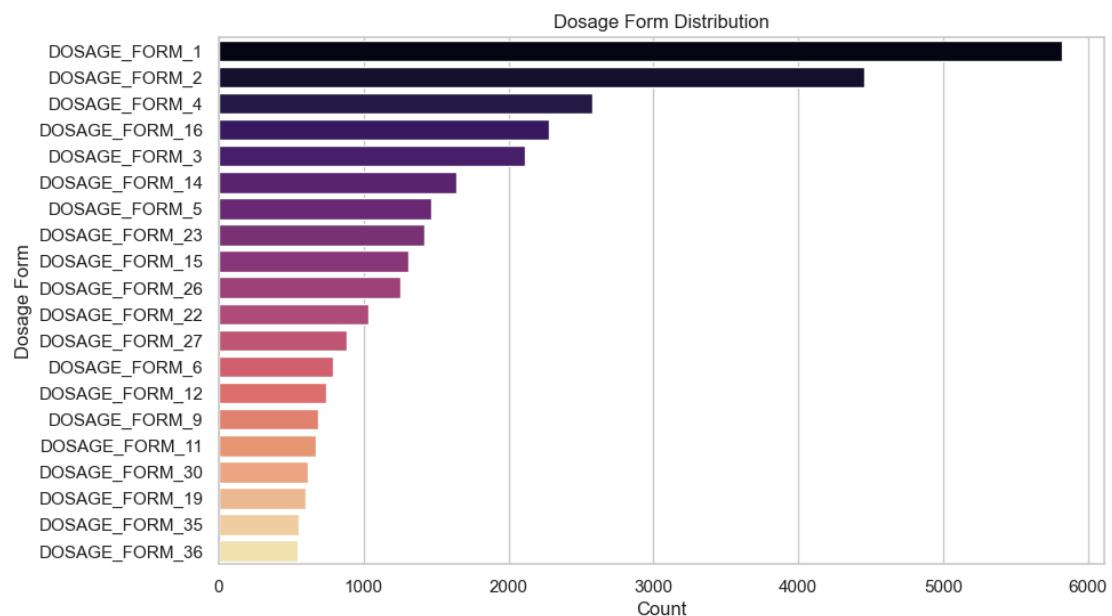
EDA



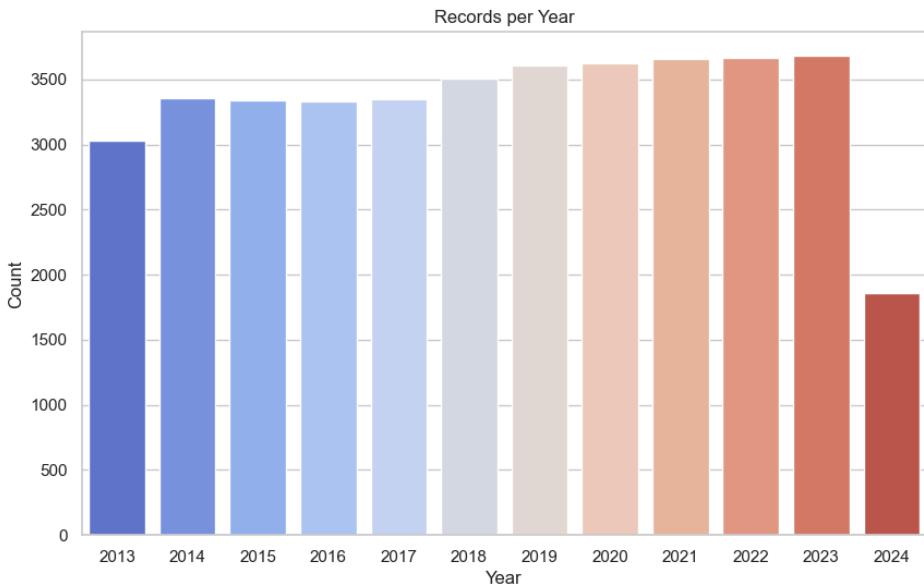
As shown in the previous plot we face 4 different countries, if we take a closer look at the most sold molecules, we can see that that one in particular arises from the rest. Of course the data is anonymized but we could take a guess on what molecule could be.



We face both lots of molecules and lots of dosage forms so predicting at a very low level is going to be a challenge.

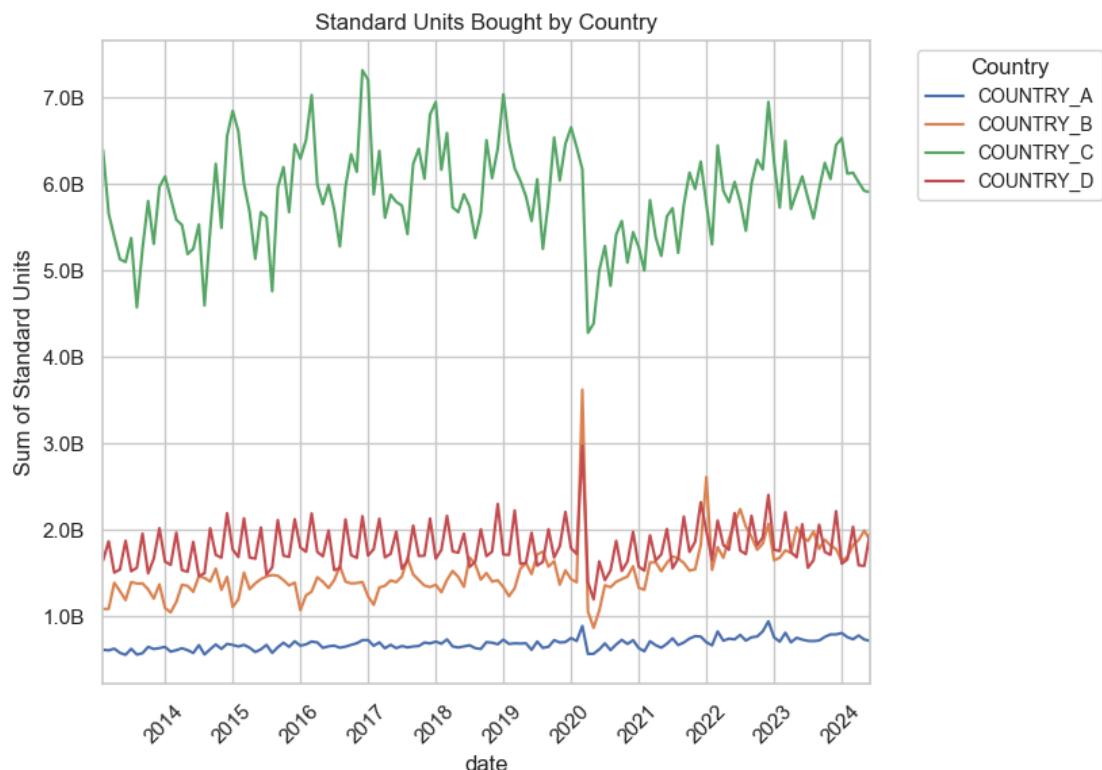


Now the focus is on the Temporal distribution of the data:

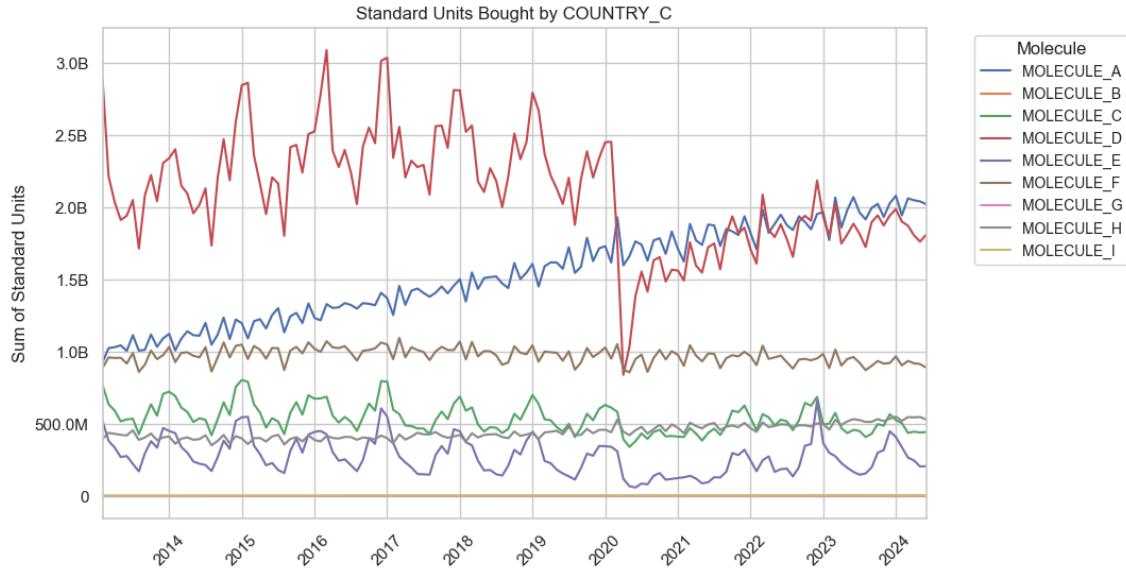


Check that we face an approximate even distribution of instances per year, except for 2024, for obvious reasons.

If we plotted the sales during all the time series we can appreciate the seasonality of the purchases of certain countries and also an outlier produced by the COVID-19 outbreak. We will need to deal with it so that our predictions can be more accurate.



By focusing on Country C, for example, we can better understand some other properties of the time series, increasing trends. It is interesting to highlight that despite the fact that around March 2020 we face a huge outlier, it hasn't affected all the molecule sales in each country in the same way, so we must be careful when preparing our models.



To make it similar to the M5 dataset we studied it at country-molecule level, so dosage and strength columns were removed.

Components Classification

As part of the forecasting project with Sandoz, an analysis on the time series components for each country-molecule combination has been carried out. This involved decomposing the series into their core components—trend, seasonality, and residuals (noise). The primary objective was to better understand the structure of the data and leverage this information to improve model selection and performance.

This has been done with the STL (Seasonal and Trend decomposition using Loess) library that splits the series into three additive components: Trend, Seasonal and Residual.

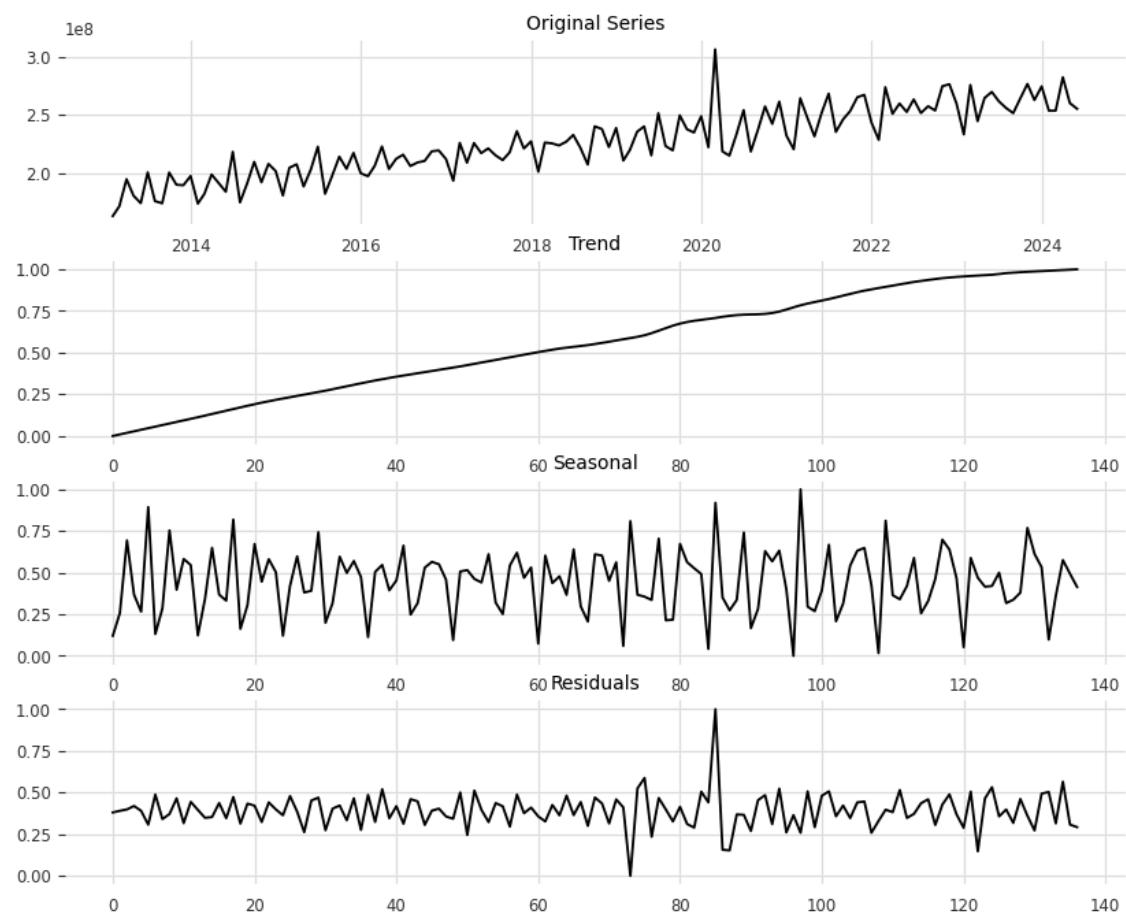
$$Y(t) = T(t) + S(t) + R(t)$$

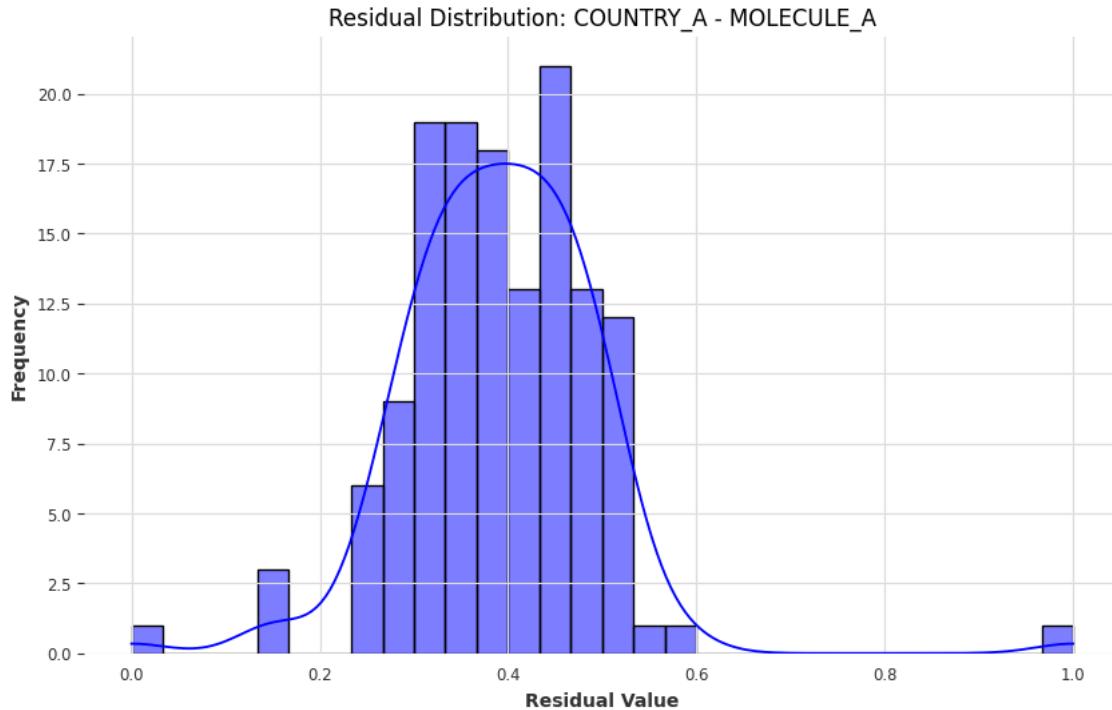
The seasonal component is estimated by applying Loess smoothing to the time series, grouping data points by their seasonal periods (e.g., months for monthly data), Then subtract the seasonality to the series to produce a seasonally adjusted series. The trend component is computed by applying Loess smoothing to the seasonally adjusted series. The residual is obtained by subtracting both the trend and seasonal components from the original series.

The seasonal and trend estimates are refined iteratively until convergence or a specified number of iterations is reached.

STL is robust to outliers, meaning it can handle irregularities without drastically affecting the decomposition and it works for time series with non-standard seasonal periods, unlike classical decomposition methods that assume fixed periodicity.

Using decomposition, we were able to identify the underlying patterns in the data. For instance, some datasets exhibited clear trends, such as consistent growth or decline over time, while others displayed strong seasonal cycles. The residuals highlighted irregular fluctuations or noise that could not be explained by the trend or seasonality components.





The analysis was done with each component normalized with min-max normalization so that they can be compared properly allowing to group series based on their dominant components. For example, country-molecule combinations with significant seasonality were likely better suited for models like Exponential Smoothing, while datasets with less-defined patterns might benefit from tree-based models such as XGBoost or LightGBM. This grouping provided a structured way to evaluate model performance based on the data's inherent features.

For this grouping, a classification of the normalized components of the series was made, dividing the series into upwards, downwards and stable trend, weak and strong seasonality and noisy and not noisy series.

After this classification implementation, clustering with K-NN was done in order to cluster series based on the classification of the components.

	Time Series	Trend	Seasonality	Noise	Cluster
0	COUNTRY_A - MOLECULE_A	upwards	strong	Not noisy	2
1	COUNTRY_A - MOLECULE_B	upwards	strong	Noisy	3
2	COUNTRY_A - MOLECULE_C	upwards	strong	Not noisy	2
3	COUNTRY_A - MOLECULE_D	upwards	strong	Not noisy	2
4	COUNTRY_A - MOLECULE_E	stable	weak	Noisy	1
5	COUNTRY_A - MOLECULE_F	downwards	strong	Not noisy	4
6	COUNTRY_A - MOLECULE_G	upwards	strong	Not noisy	2
7	COUNTRY_A - MOLECULE_H	upwards	strong	Noisy	3
8	COUNTRY_A - MOLECULE_I	upwards	strong	Noisy	3
9	COUNTRY_B - MOLECULE_A	upwards	strong	Not noisy	2
10	COUNTRY_B - MOLECULE_B	upwards	strong	Not noisy	2
11	COUNTRY_B - MOLECULE_C	stable	strong	Not noisy	0
12	COUNTRY_B - MOLECULE_D	stable	strong	Not noisy	0
13	COUNTRY_B - MOLECULE_E	stable	strong	Not noisy	0
14	COUNTRY_B - MOLECULE_F	downwards	strong	Not noisy	4
15	COUNTRY_B - MOLECULE_G	upwards	weak	Noisy	5
16	COUNTRY_B - MOLECULE_H	stable	strong	Not noisy	0
17	COUNTRY_B - MOLECULE_I	upwards	strong	Not noisy	2
18	COUNTRY_C - MOLECULE_A	upwards	strong	Not noisy	2
19	COUNTRY_C - MOLECULE_B	stable	strong	Not noisy	0
20	COUNTRY_C - MOLECULE_C	stable	weak	Noisy	1
21	COUNTRY_C - MOLECULE_D	stable	strong	Not noisy	0
22	COUNTRY_C - MOLECULE_E	stable	strong	Not noisy	0
23	COUNTRY_C - MOLECULE_F	stable	weak	Noisy	1
24	COUNTRY_C - MOLECULE_G	upwards	strong	Not noisy	2
25	COUNTRY_C - MOLECULE_H	upwards	strong	Not noisy	2
26	COUNTRY_C - MOLECULE_I	upwards	strong	Noisy	3
27	COUNTRY_D - MOLECULE_A	upwards	strong	Not noisy	2
28	COUNTRY_D - MOLECULE_B	stable	weak	Not noisy	1
29	COUNTRY_D - MOLECULE_C	stable	weak	Noisy	1
30	COUNTRY_D - MOLECULE_D	stable	strong	Not noisy	0
31	COUNTRY_D - MOLECULE_E	stable	strong	Noisy	3
32	COUNTRY_D - MOLECULE_F	stable	strong	Not noisy	0
33	COUNTRY_D - MOLECULE_G	stable	strong	Noisy	3
34	COUNTRY_D - MOLECULE_H	upwards	strong	Noisy	3
35	COUNTRY_D - MOLECULE_I	upwards	strong	Noisy	3

Outliers

In this second approach (first explained with M5) we selected the outliers as data points that significantly deviate from the general pattern of the dataset. In sales forecasting, outliers can occur due to various reasons, such as:

- Sudden spikes or drops in sales due to promotions or stockouts.
- Data entry errors or missing values replaced with placeholders.
- External factors like economic shifts or unexpected events (e.g., pandemics, natural disasters).

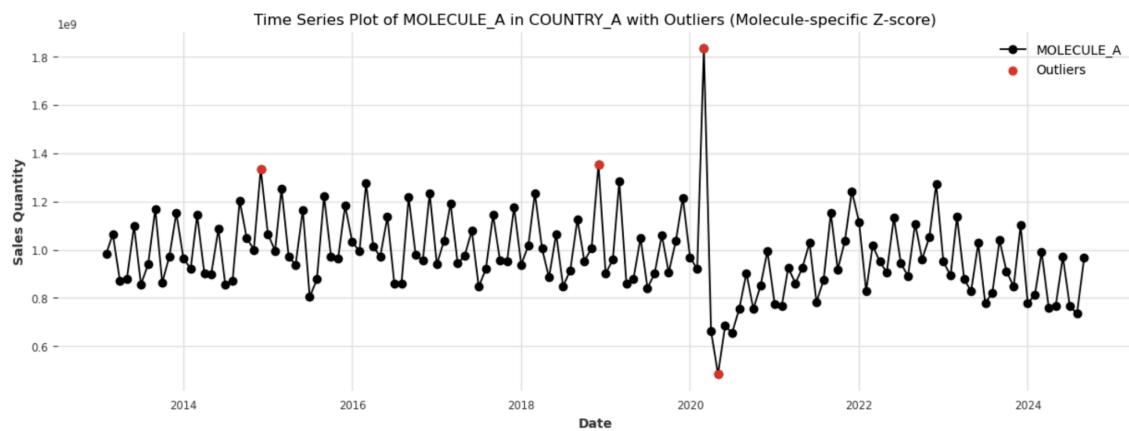
Outliers are crucial to identify and handle because they can:

- Distort summary statistics like mean and variance.
- Negatively impact forecasting models, leading to poor performance.
- Represent actionable insights, such as detecting high-demand periods or identifying unusual customer behavior.

Knowing this we can't treat them as normal outliers, like set a threshold and cut the points outside, because for some series those spikes could be normal and essential, so we need to do another procedure.

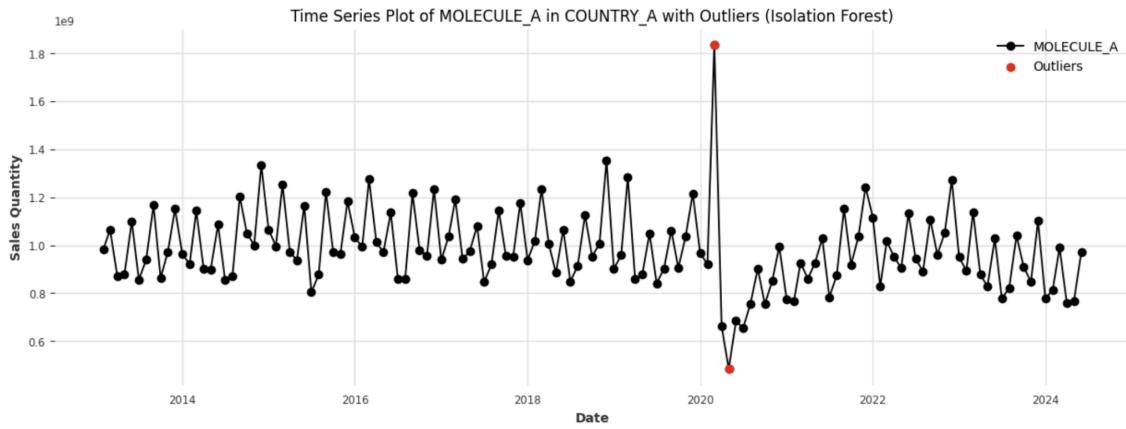
First we need to identify the outliers by analyzing each series, for this we tried two different methods to find the outliers **Z-score** and **Isolation Forest**.

The **Z-score** method is a statistical approach that measures how far a data point deviates from the mean, expressed in terms of standard deviations. It assumes that the data follows a normal distribution, which makes it straightforward to apply for datasets with symmetric patterns. In our analysis, we calculated the Z-scores for each data point and flagged those exceeding a threshold of $|Z|>2$ as outliers. This approach is computationally efficient and easy to interpret, as it highlights anomalies that are significantly different from the average behavior. However, the Z-score method's reliance on the assumption of normality may limit its effectiveness for skewed datasets like ours, where sales data often exhibit seasonal trends.



The red points in the picture were detected as outliers, we did this for every country-molecule series to see how well the method detected the outliers.

On the other hand, the **Isolation Forest** is a machine-learning-based method that detects outliers by isolating them in a tree structure. The algorithm works by randomly splitting the dataset and calculating the anomaly score for each data point based on how quickly it is isolated. Outliers, which are sparse and distinct, tend to be isolated faster than regular data points. Unlike the Z-score, the Isolation Forest does not require assumptions about the data's distribution, making it more robust for complex and non-linear patterns. This method was particularly effective in identifying localized anomalies, such as unexpected sales spikes in specific regions or molecules. However, it required careful tuning of hyperparameters, such as the number of trees and the contamination level, to balance detection accuracy and computational efficiency.



Same as we did with the Z-score but with this other method, here the parameters (contamination, which is the percentage of outliers expected) are not yet tuned so we used the same for every series, but they need to be treated differently

By applying both methods, we could compare their strengths and limitations. While the Z-score method provided a simple and interpretable approach to detect extreme anomalies, the Isolation Forest captured more subtle patterns that statistical methods might overlook. Combining insights from both approaches helped ensure that our outlier detection process was comprehensive and adaptable to the diverse characteristics of our sales data.

To handle the detected outliers in our sales data, we tried some substitution methods to ensure that the adjusted dataset maintained its integrity and continuity for forecasting. The primary goal was to replace the anomalous values with more representative figures while preserving the underlying trends and patterns.

One of the simplest and most robust approaches we applied was the **median substitution**. This method replaces the outlier values with the median of the dataset or a specific segment of the data. The median is less sensitive to extreme values compared to the mean, making it a reliable choice for handling outliers without distorting the overall distribution. By substituting outliers with the median, we effectively reduced their impact on subsequent analyses while maintaining consistency in the dataset.

We also explored various **interpolation methods** to reconstruct the values of outliers based on their surrounding data points. These methods estimate the missing or anomalous values by fitting curves or lines through neighboring points, thereby preserving temporal trends. The interpolation techniques we tested included:

- **Linear:** A straightforward approach that fits a straight line between two adjacent points.

- **Spline and Cubic:** Methods that use smooth polynomial curves for interpolation, offering a more flexible fit for capturing non-linear trends.
- **Polynomial:** Fits a polynomial function to the data, providing a broader representation of trends but requiring careful selection of the degree to avoid overfitting.
- **Nearest and Pad:** Approaches that substitute outliers with the nearest valid value or propagate the last observed value, respectively. These are particularly useful for maintaining short-term stability in time series data.
- **Akima:** A specialized spline interpolation method that provides a more localized and smooth fit, particularly effective for unevenly spaced data.

After trying them we found that, while median substitution offered simplicity and robustness, interpolation methods allowed for greater adaptability by leveraging the temporal context of the data. Among the interpolation techniques, smoother methods like cubic or spline proved effective for datasets with clear seasonal or cyclical patterns, while linear and nearest methods were more suited for simpler trends.

The problem is that, since we have a series per each country-molecule combination, we could not just use the same method to find and substitute the outliers for all of them.

To determine the best outlier handling method for each series, we implemented a systematic evaluation pipeline that compared the detection and correction techniques mentioned before. The goal was to identify the combination that minimized forecasting errors, ensuring optimal data quality for model training and prediction.

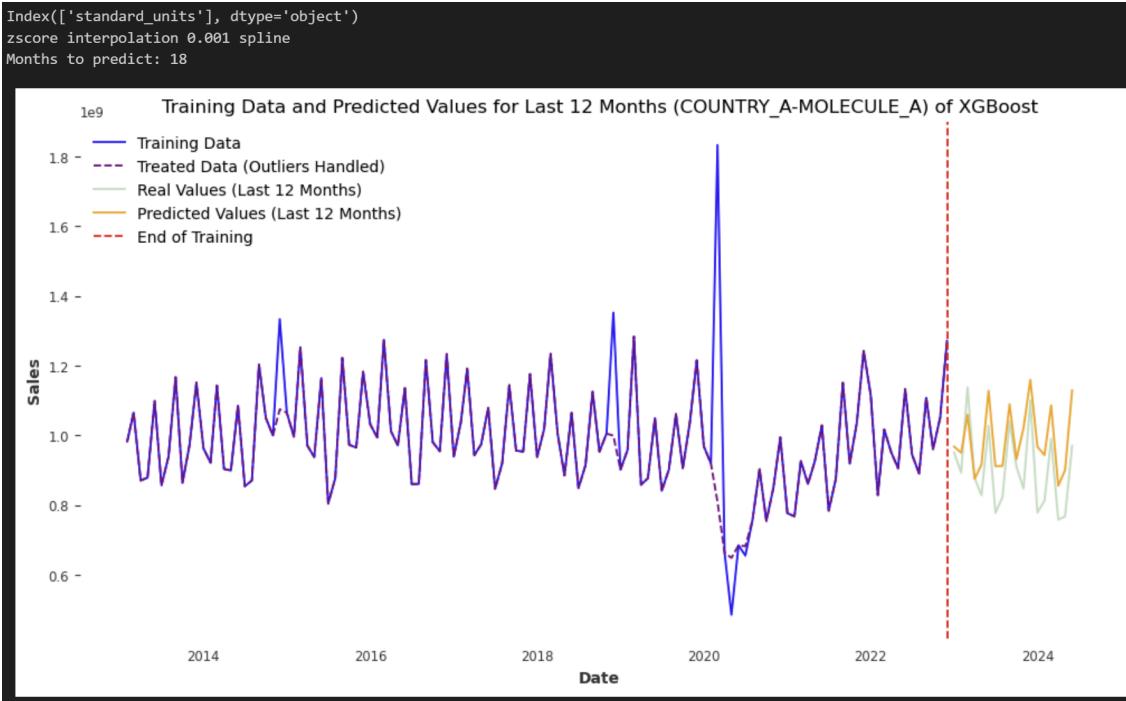
The pipeline began by loading sales data for a specific country and molecule, filtering and preparing it for time series analysis. We defined a function (`handle_outliers`) to apply different combinations of outlier detection and correction methods. For detection, we tested the Z-score and Isolation Forest methods, while for correction, we evaluated median substitution and several interpolation techniques, including linear, spline, polynomial, nearest, pad, cubic, and Akima methods. Each method was parameterized for flexibility, such as varying the contamination levels and the order of interpolation for polynomial or spline methods.

To evaluate the effectiveness of each combination, we split the data into training (2013–2022) and testing (2023 onwards) sets. Using an XGBoost model (just because it was the first one we took from the models part), we first trained the model on the raw data to establish a baseline of mean absolute error (MAE), root mean squared error (RMSE), and mean absolute percentage error (MAPE). We then applied each combination of outlier handling techniques to the training data, retrained the model, and forecasted the test set. The results for each combination were compared against the baseline to assess improvements.

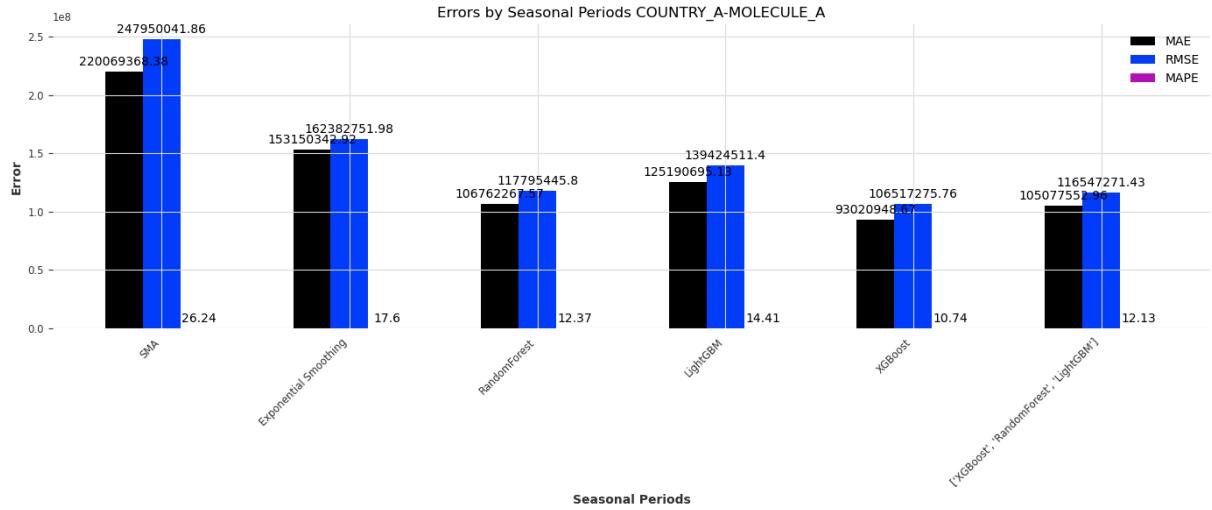
A decision rule was used to select the best method for each series. We would say that a combination was better if at least two out of the three metrics (MAE, RMSE, MAPE) showed improvement. This ensured a balanced evaluation, considering both absolute and relative errors.

The results of this process were stored for each molecule in a summary table, capturing details such as the outlier detection method, correction technique, contamination level, interpolation method, and corresponding forecasting metrics. The best-performing combination for each molecule was saved in a csv to use the chosen method with the models testing, ensuring the highest accuracy in sales forecasting.

Country	Molecule	Outlier Handling	Detection	Correction	Contamination	Interpolation Method	MAE	RMSE	MAPE
COUNTRY_A	MOLECULE_A	Yes	zscore	interpolation	0.001	spline	96829746.15671426	110414597.28758001	11.3813035611062
COUNTRY_A	MOLECULE_B	Yes	isolation_forest	interpolation	0.01	linear	135247.23267041112	182660.2190092243	7.971853927462147
COUNTRY_A	MOLECULE_C	Yes	isolation_forest	interpolation	0.001	linear	532820.3136842584	678838.4921921842	13.453842919386995
COUNTRY_A	MOLECULE_D	Yes	isolation_forest	interpolation	0.04	spline	12069009.355396725	17213607.20567134	11.527176243238555
COUNTRY_A	MOLECULE_E	Yes	isolation_forest	interpolation	0.01	linear	5480.192240027523	5873.635513835299	15.60377916669913
COUNTRY_A	MOLECULE_F	Yes	isolation_forest	interpolation	0.001	linear	11233514.331722954	13577638.085860286	5.643374210582393
COUNTRY_A	MOLECULE_G	Yes	isolation_forest	interpolation	0.05	linear	30196730.434696957	34794228.36505246	5.235069209767256
COUNTRY_A	MOLECULE_H	Yes	isolation_forest	interpolation	0.02	akima	1259037.1006598833	1684006.4913578264	7.129256486447371
COUNTRY_A	MOLECULE_I	Yes	isolation_forest	interpolation	0.05	linear	3770.62709450099	4861.232776607977	89.86680772971226
COUNTRY_B	MOLECULE_A	Yes	isolation_forest	interpolation	0.05	nearest	102675149.2343904	145955402.71303642	7.985177949182058
COUNTRY_B	MOLECULE_B	Yes	zscore	interpolation	0.001	polynomial	282833.315837907	627496.6201270189	24.702751112026096



As explained above, we can see on the plot that the COVID effect was significantly reduced for this molecule A, and the forecasted values (in orange) seem to return good results (for the XGBoost model). We have the original data in blue and the treated one (outliers) doted in purple.



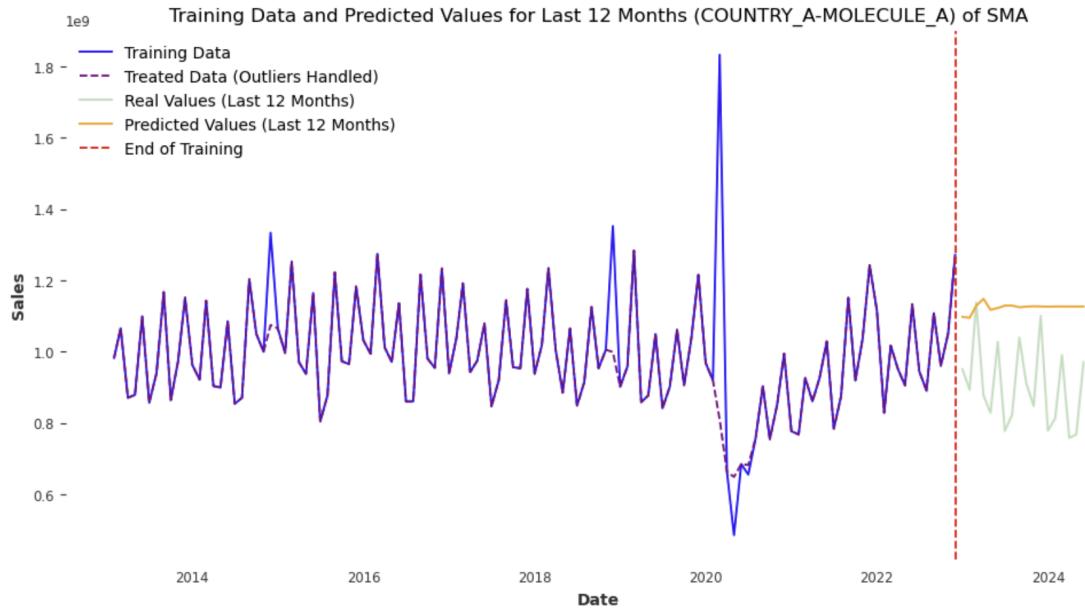
Here we include a comparison for Molecule A again, with all the models we have used to predict, having as first value a Simple Moving Average (SMA) as baseline. A final ensemble with the best models is also detailed. We will see this process in detail now.

Models

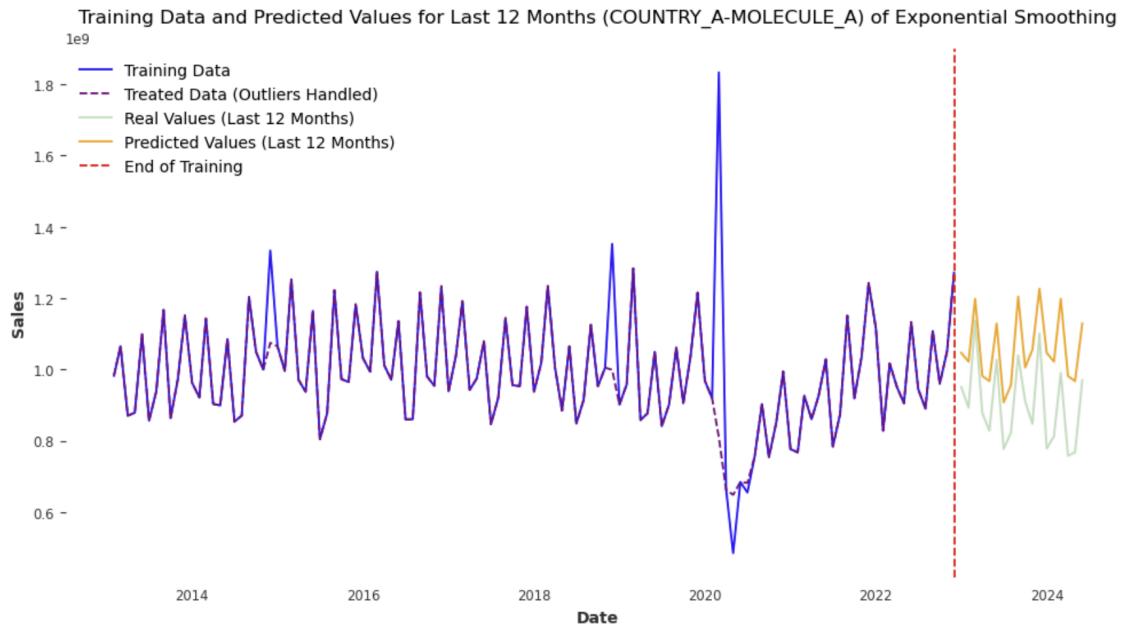
After trying and learning with the M5 we implemented everything with this other data, we used the best models obtained before (Exponential Smoothing, XGBoost, LightGBM, and Random Forest) and SMA for comparison. We created a function that evaluates all models, identifies the top three performers, generates an ensemble forecast, and produces a graph summarizing the errors for each country-molecule combination.

The modeling process was conducted over three different train-test splits:

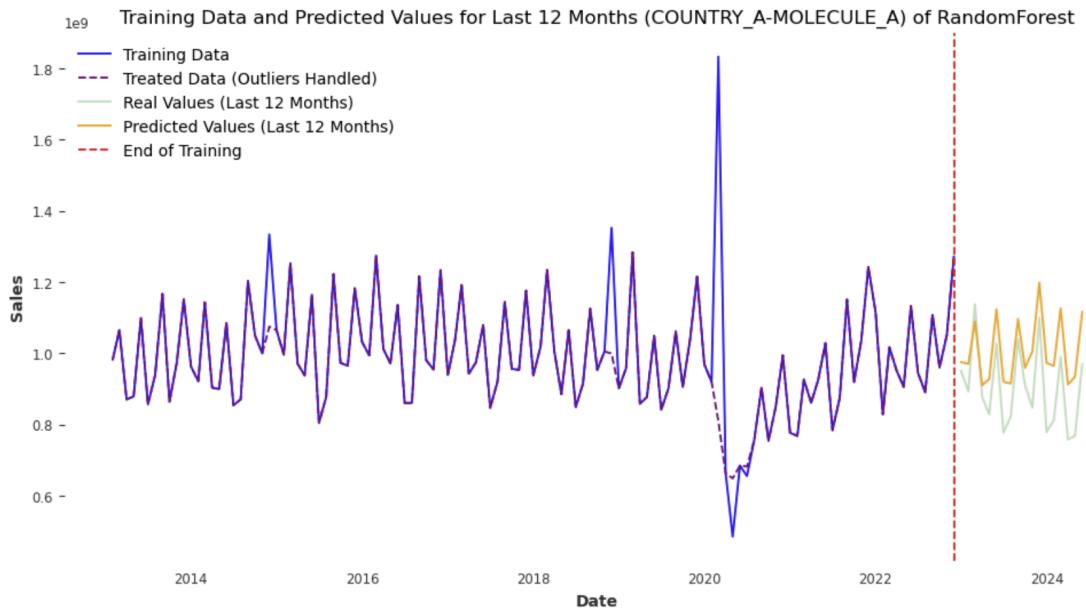
- First Split:** Training data spanned 2013 to 2022, with the testing period covering one and a half years (2023 to June 2024). This approach utilized the original data provided by Sandoz, and it served as the foundation for identifying the best-performing models for each country-molecule combination and saved in a csv. We decided this using something similar to the outlier one, by comparing the errors and the best one was the one where two out of the three errors were better.



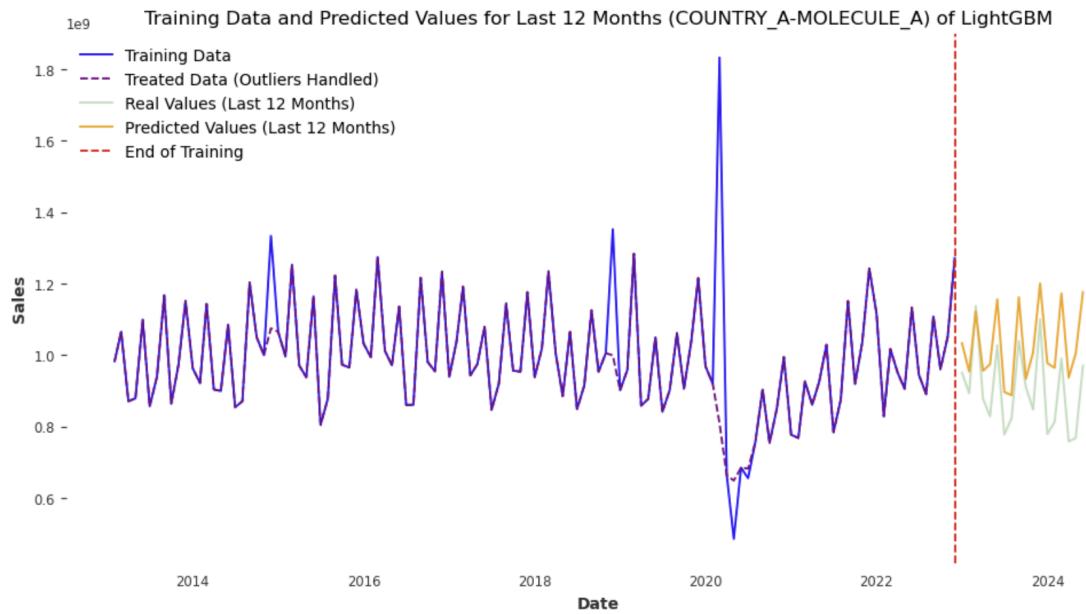
On the plot above, we display the results for the SMA model for the Country A - Molecule A (same combination for the following graphs). For this particular case, as the data has a higher trend, the average value of the previous years is larger than the testing period, so the result is poor.



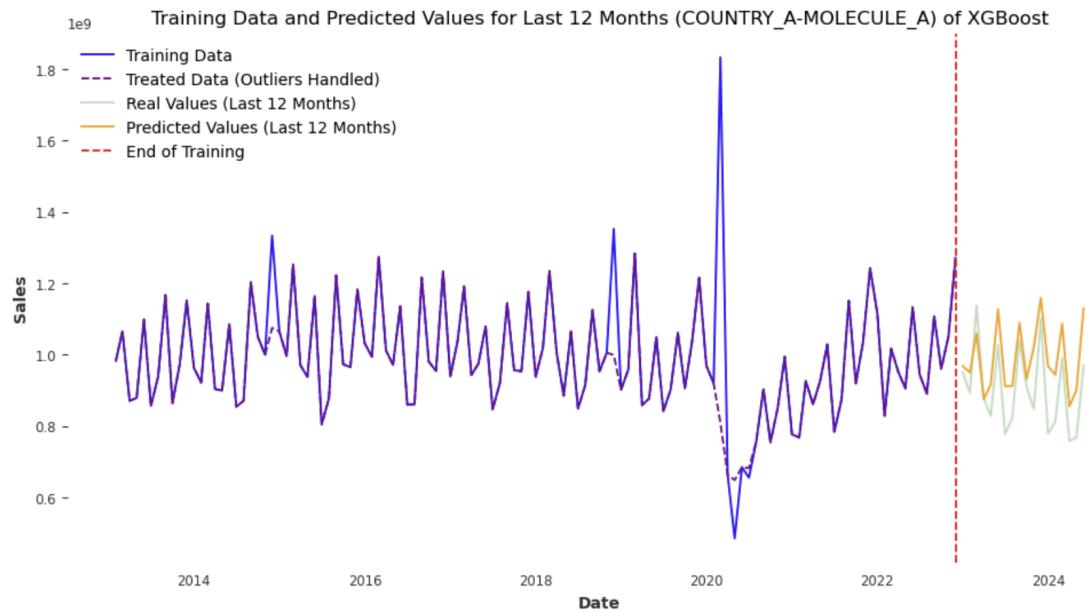
Next is the result for the Exponential Smoothing, we can see the real values in light green and the forecasted ones in orange. It seems to be a decent prediction, slightly higher sales number than the Real Value.



Proceeded by the model Random Forest. Together with the next one, it is quite accurate and fits really well the test data.

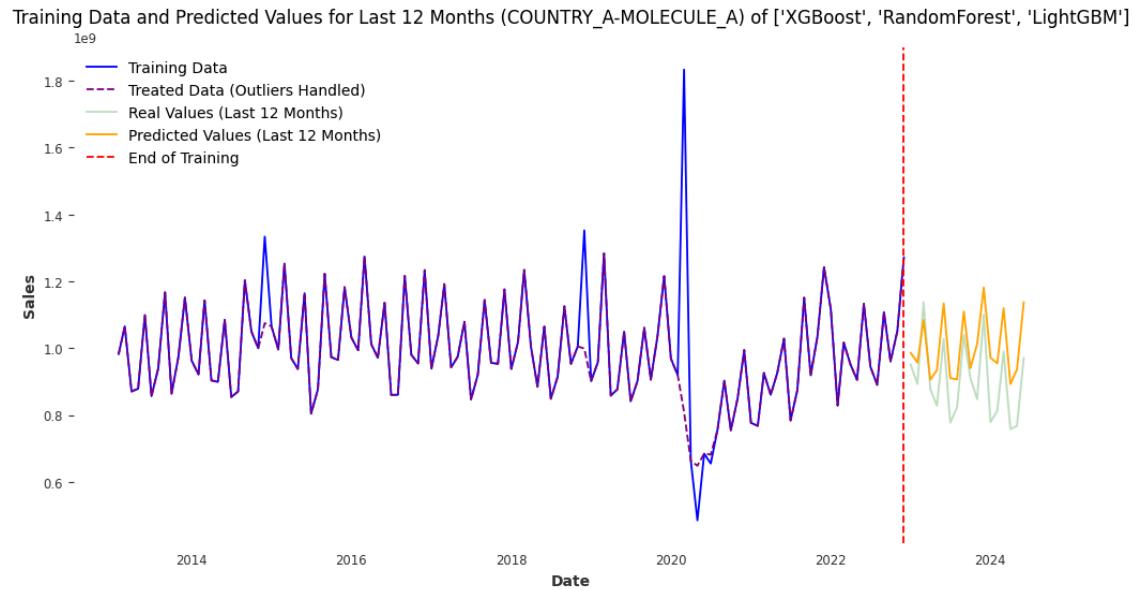


Next, LightGBM produces similar results to Random Forest, thus, good results.

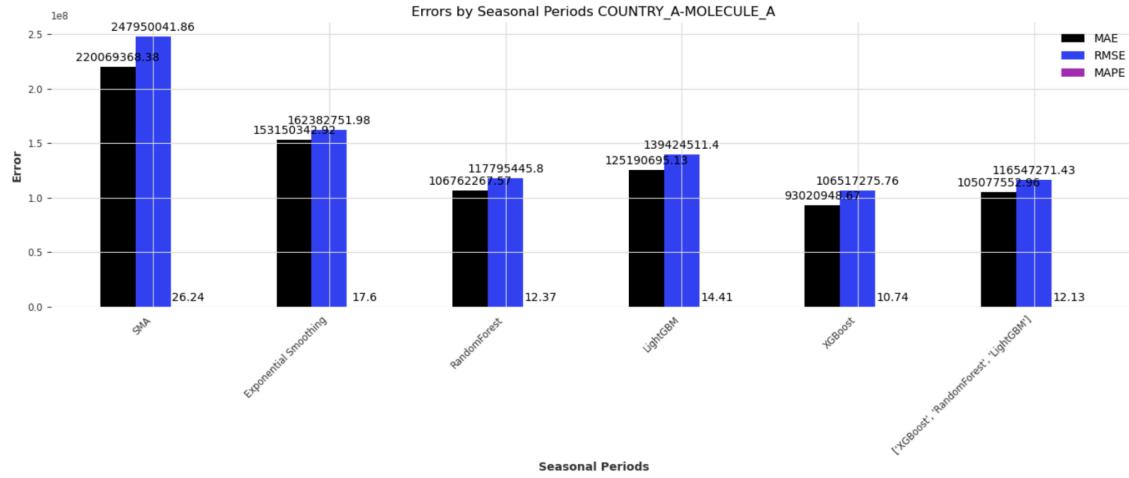


Finally we tested XGBoost, which, as we can see in the error table below, is the model with the lowest error terms.

We are now going to plot the ensemble function:



Here we have added the **Ensemble**, it uses the predictions for the three best performing models of each molecule and weights them as 45% for the best one, 30% for the second one and 25% for the last model, LightGBM in this case.



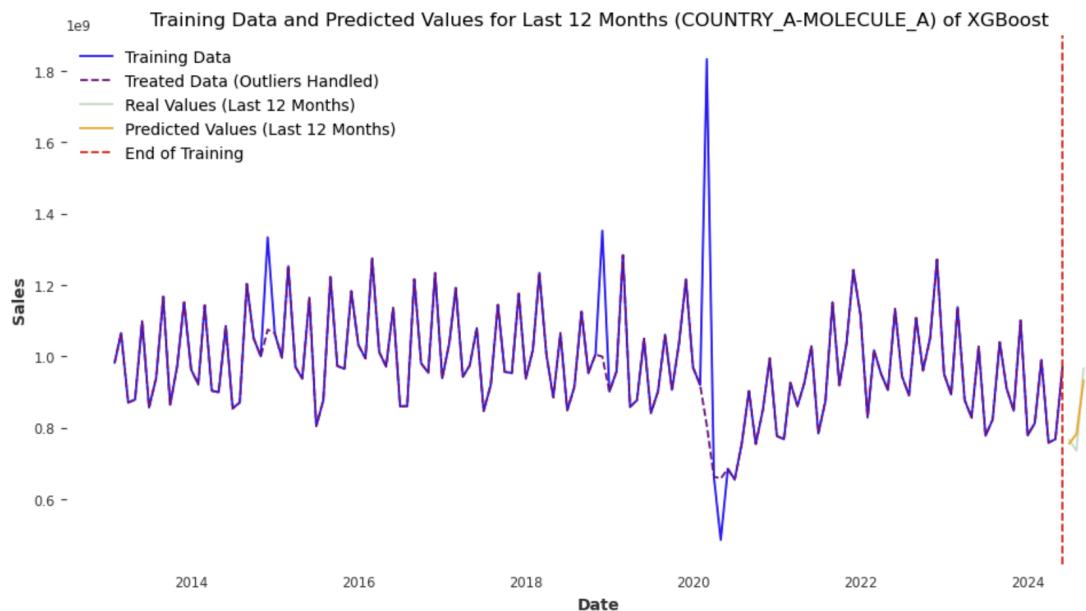
As referenced before, with this graph we can compare the obtained error values for each of the methods performed. The ensemble provides a worse numerical result than the best performing model, but makes it stronger to new real data. For the case where the difference between the best model and the rest (usual case when mathematical models like Exponential Smoothing are better than the machine learning ones), we will be taking just the best model.

	country	molecule	model
0	COUNTRY_A	MOLECULE_A	XGBoost
1	COUNTRY_A	MOLECULE_B	Exponential Smoothing
2	COUNTRY_A	MOLECULE_C	(Exponential Smoothing, RandomForest, SMA)
3	COUNTRY_A	MOLECULE_D	XGBoost
4	COUNTRY_A	MOLECULE_E	RandomForest
5	COUNTRY_A	MOLECULE_F	Exponential Smoothing
6	COUNTRY_A	MOLECULE_G	Exponential Smoothing
7	COUNTRY_A	MOLECULE_H	Exponential Smoothing
8	COUNTRY_A	MOLECULE_I	RandomForest
9	COUNTRY_B	MOLECULE_A	(LightGBM, Exponential Smoothing, RandomForest)
10	COUNTRY_B	MOLECULE_B	XGBoost
11	COUNTRY_B	MOLECULE_C	XGBoost
12	COUNTRY_B	MOLECULE_D	Exponential Smoothing
13	COUNTRY_B	MOLECULE_E	(SMA, RandomForest, XGBoost)
14	COUNTRY_B	MOLECULE_F	RandomForest
15	COUNTRY_B	MOLECULE_G	Exponential Smoothing
16	COUNTRY_B	MOLECULE_H	Exponential Smoothing
17	COUNTRY_B	MOLECULE_I	LightGBM
18	COUNTRY_C	MOLECULE_A	RandomForest
19	COUNTRY_C	MOLECULE_B	Exponential Smoothing
20	COUNTRY_C	MOLECULE_C	Exponential Smoothing
21	COUNTRY_C	MOLECULE_D	(XGBoost, RandomForest, LightGBM)
22	COUNTRY_C	MOLECULE_E	(Exponential Smoothing, RandomForest, XGBoost)
23	COUNTRY_C	MOLECULE_F	Exponential Smoothing
24	COUNTRY_C	MOLECULE_G	(LightGBM, Exponential Smoothing, XGBoost)
25	COUNTRY_C	MOLECULE_H	Exponential Smoothing
26	COUNTRY_C	MOLECULE_I	Exponential Smoothing
27	COUNTRY_D	MOLECULE_A	Exponential Smoothing
28	COUNTRY_D	MOLECULE_B	Exponential Smoothing
29	COUNTRY_D	MOLECULE_C	(RandomForest, XGBoost, LightGBM)
30	COUNTRY_D	MOLECULE_D	(XGBoost, RandomForest, SMA)
31	COUNTRY_D	MOLECULE_E	Exponential Smoothing
32	COUNTRY_D	MOLECULE_F	(Exponential Smoothing, XGBoost, RandomForest)
33	COUNTRY_D	MOLECULE_G	Exponential Smoothing
34	COUNTRY_D	MOLECULE_H	LightGBM
35	COUNTRY_D	MOLECULE_I	Exponential Smoothing

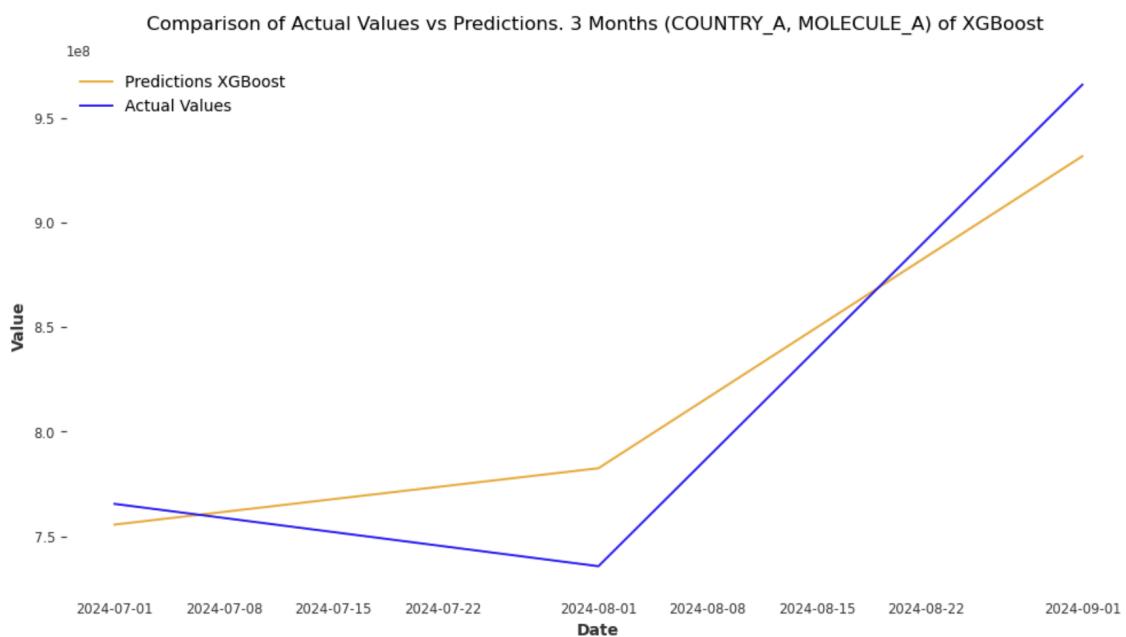
In this image you can see the table saved with the best model for each country-molecule combination.

2. **Second Split:** Training data extended from 2013 to June 2024, with the testing period covering July to September 2024. This split was made because of the reception of additional data in December (covering July to September 2024). Using the best models

identified from the first split, we updated the training data and reevaluated the models to incorporate the new information.



Example of the first combination using the best model obtained from the first split.

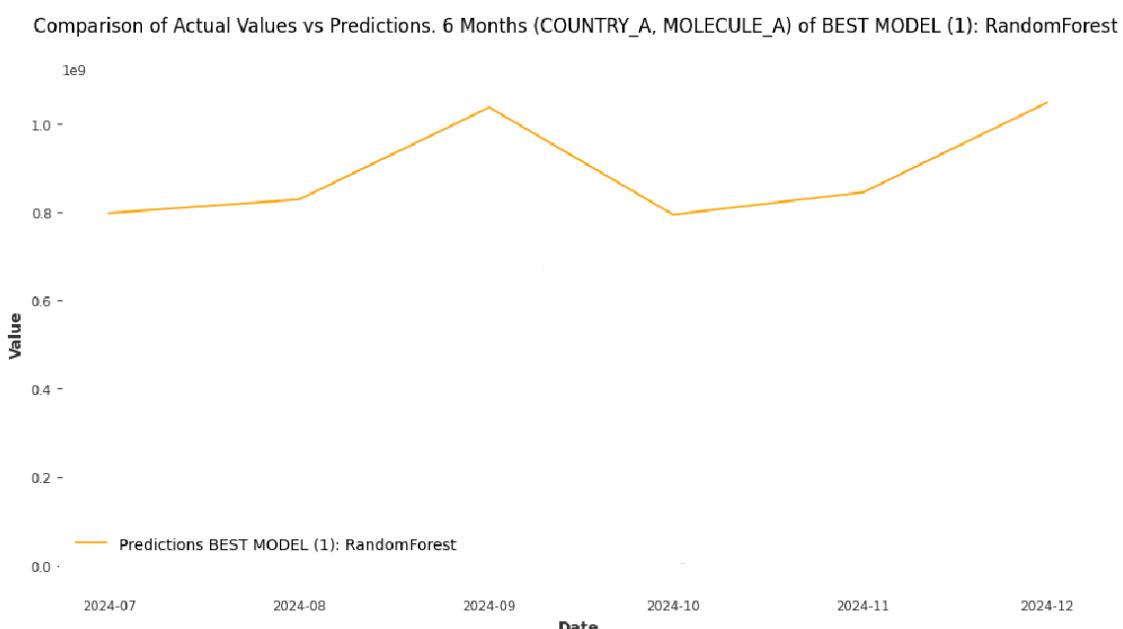
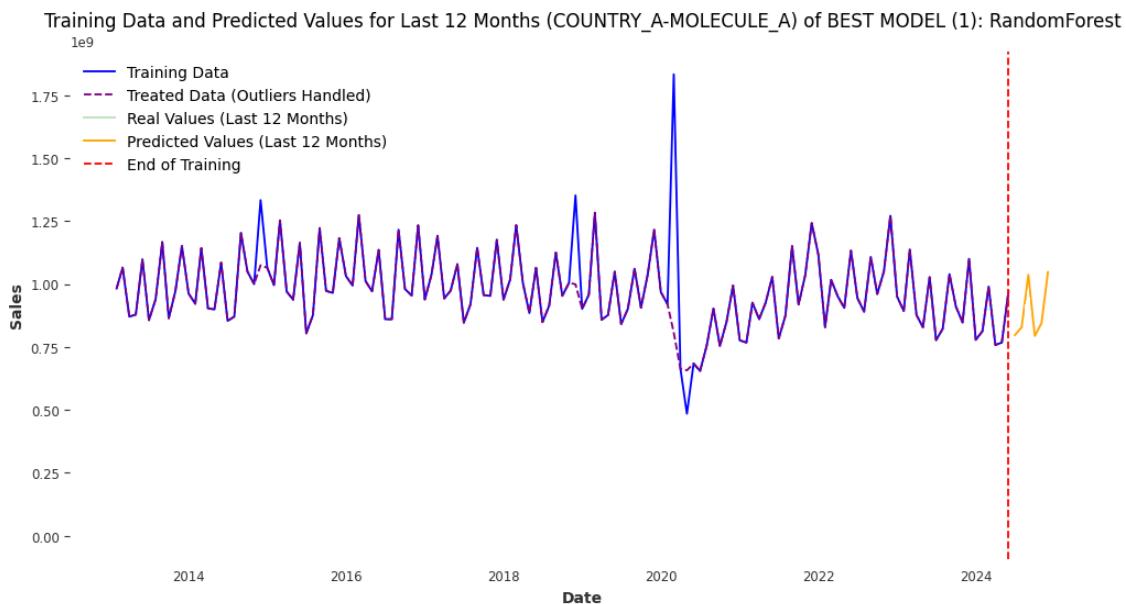


This image is a close up to the real (blue) and predicted (orange) values.

Errors: [('XGBoost', 30323109.084677577, 33975114.81672318, 3.73462231420145)]

First number corresponds to MAE, second to RMSE and last MAPE errors.

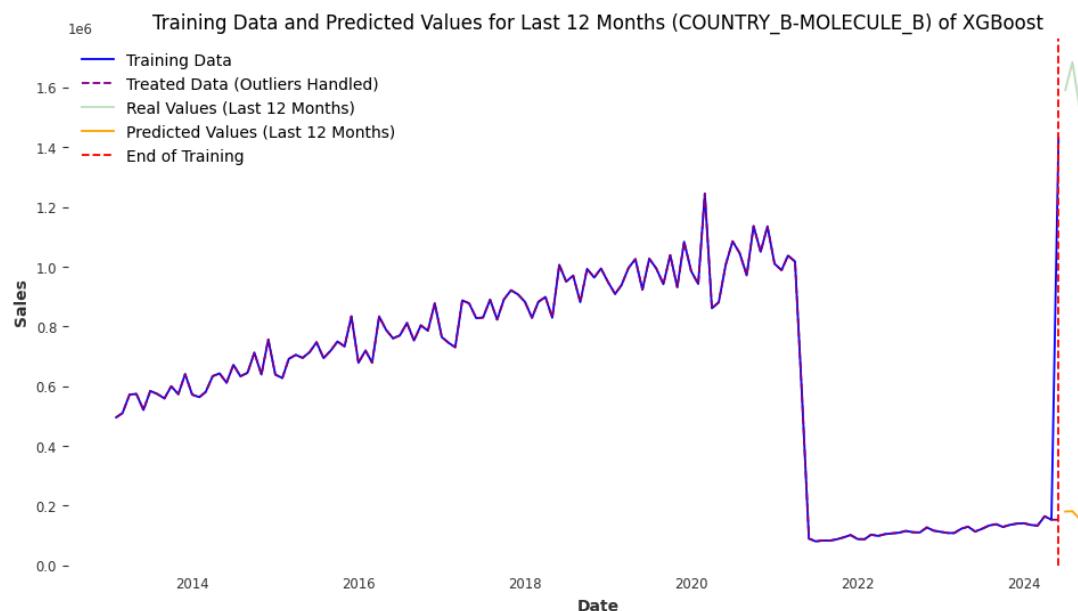
3. **Third Split:** Training data included the full range from 2013 to June 2024, and the testing period covered July to December 2024. This final approach aimed to provide Sandoz with a predictive baseline for the months where they currently lack data (October, November, and December 2024). These forecasts can be later compared to actual sales data as it becomes available, serving as a validation of the models' robustness.

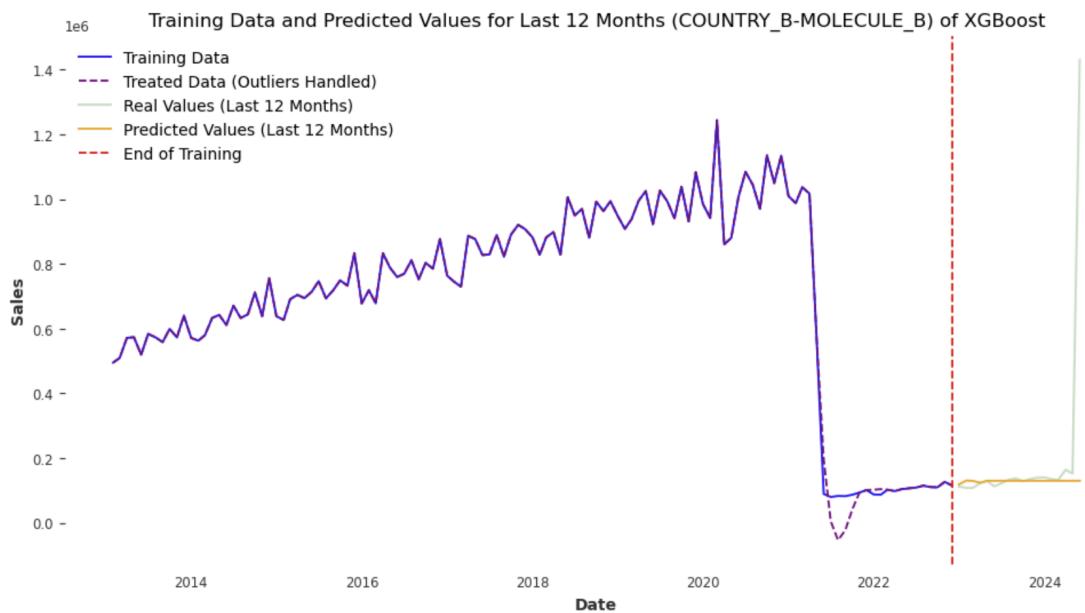


This structured progression ensured that we maximized the utility of the available data while adapting to the evolving dataset. The use of the SMA model in all comparisons further emphasized robustness and consistency across the evaluations. Additionally, the ensemble approach with the top three models allowed us to leverage their combined strengths, ensuring that the forecasts were as accurate and reliable as possible. This process not only provided actionable insights for the present but also established a framework for future analyses as more data becomes available.

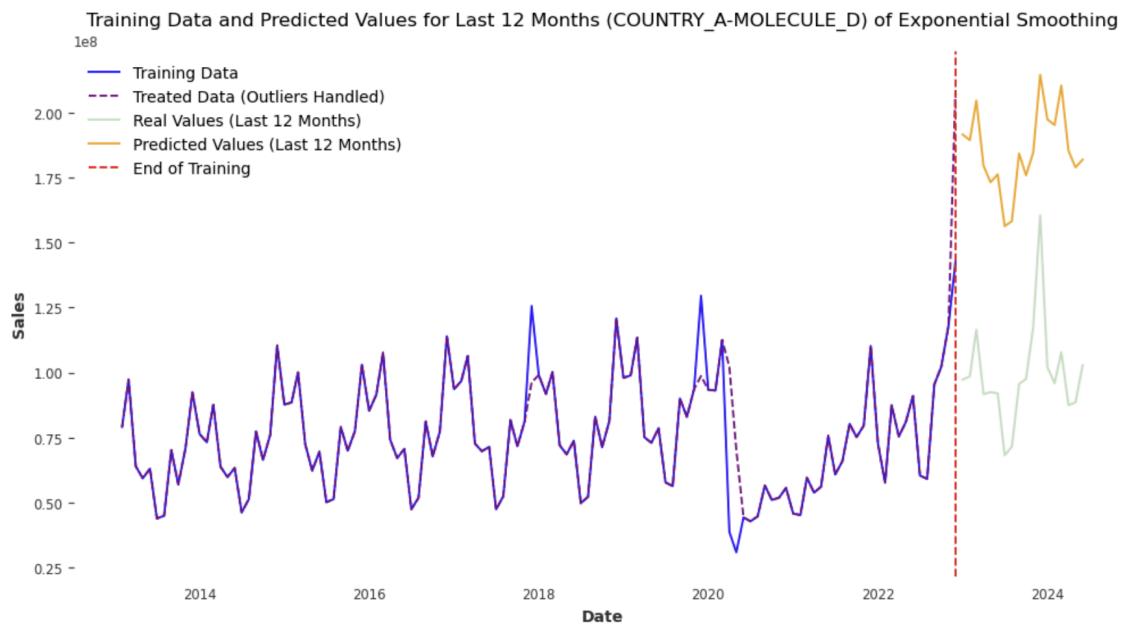
Remarkable cases

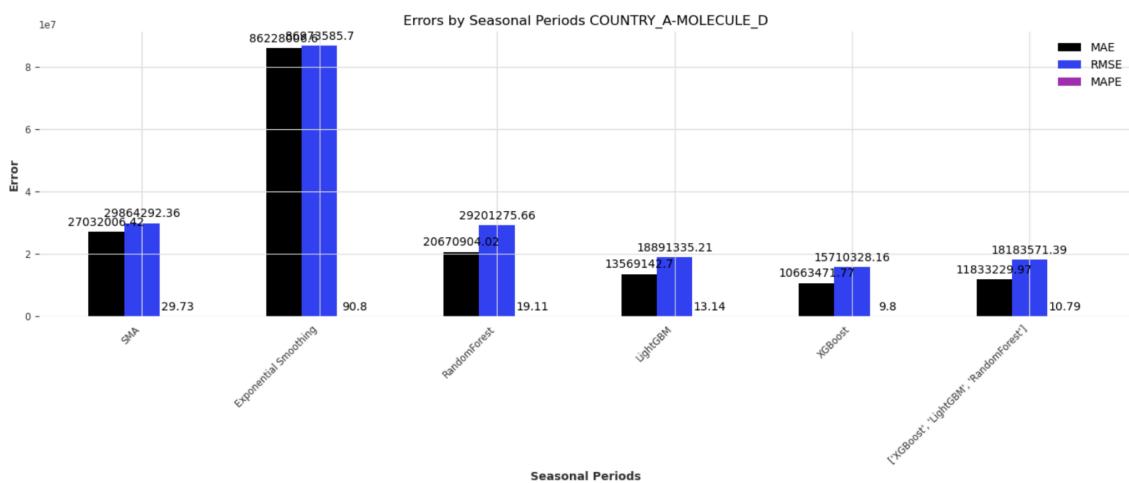
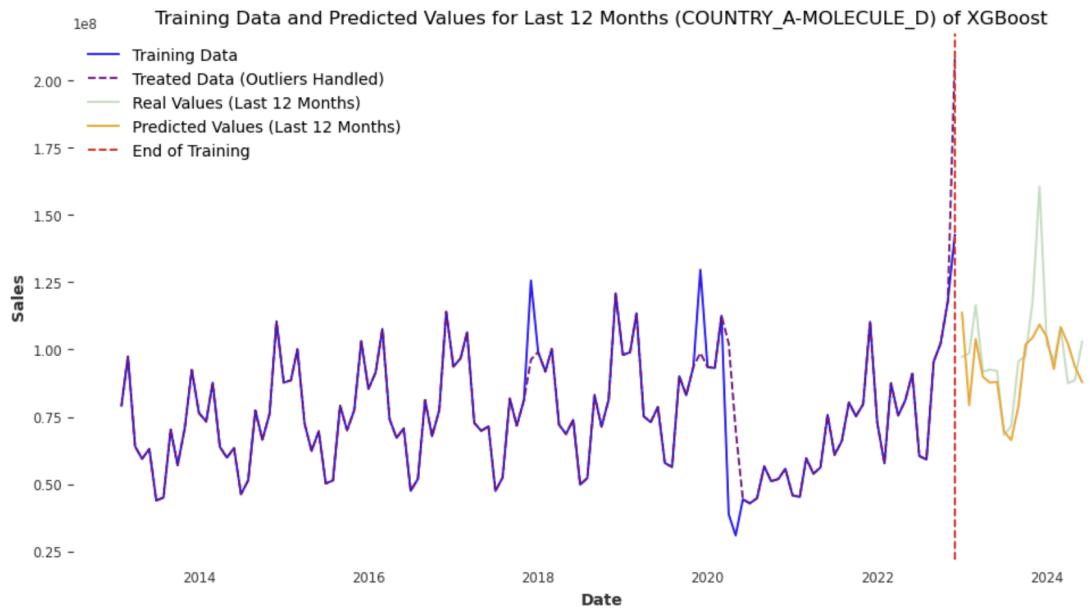
The following two plots present a visualization of forecasts with a significant error. However, upon closer observation, it becomes evident that the model is not entirely ineffective, as it closely follows prior patterns and the real values were particularly challenging to predict.





In this series, the only model affected by the last value (which is an outlier) is the Exponential Smoothing given that it assigns exponentially decreasing weights to older observations, meaning the most recent values have the greatest influence on the smoothed value.





Final Model Structure

In our project, we employed an ensemble approach to improve forecasting accuracy and robustness by leveraging the strengths of multiple models. This method aimed to address the complex patterns and variability present in the Sandoz dataset, which includes sales data spanning from 2013 to 2024.

The ensemble combined predictions from the Exponential Smoothing, XGBoost, LightGBM, Random Forest and Simple Moving Average (SMA). This ensemble approach is key in our project for many different reasons:

- **Reduced Individual Model Bias:** Each model addressed different aspects of the dataset. For example, while advanced models like LightGBM and XGBoost captured complex and non-linear patterns, Exponential Smoothing excelled at trend and seasonal adjustments, and SMA added stability, robustness.
- **Improved Forecasting Accuracy:** By blending predictions, the ensemble consistently outperformed any single model. It achieved lower error metrics across MAE, RMSE, and MAPE compared to standalone models.
- **Robustness to Variability:** The ensemble reduced the impact of outliers and anomalies by integrating diverse forecasting methodologies, ensuring reliable predictions even in irregular periods. We can see this better in some of the molecules of the Sandoz dataset.
- **Balanced Complexity:** Combining simple models like SMA with advanced machine learning models ensured a balance between computational efficiency and prediction precision.

Results & Discussion

Our project focused on improving sales forecasting for Sandoz, a pharmaceutical company, using time series data. The primary goal was to develop a forecasting model that would help forecast product sales for 2024. By utilizing machine learning algorithms alongside traditional time series models, we were able to analyze the data, preprocess it, and build models that could accurately predict future sales.

We started by familiarizing ourselves with the key components of time series analysis—trend, seasonality, patterns, and noise. Understanding these elements was crucial as they guided our approach to forecasting. We also learned the importance of proper data preprocessing to handle large datasets effectively. For instance, by reducing the size of the M5 Kaggle dataset, we made

it manageable for our computational resources, ensuring that our analysis would run smoothly on the hardware we had available (our current laptops and personal computers).

We tested a range of models to identify the most effective forecasting techniques for this problem. We began with traditional models like Prophet, Exponential Smoothing (Holt-Winters), and SARIMA. Among these, the Prophet was highly effective at capturing seasonal patterns and the effects of holidays and special events. It helped us produce forecasts that reflected yearly trends, though it wasn't as accurate when dealing with more irregular data. We then turned to Exponential Smoothing, which was designed to capture both trends and seasonality. While it provided good results, it struggled with handling abrupt changes or anomalies, particularly in irregular periods. The SARIMA model was another option we explored; it showed that it could manage general trends very poorly.

As we went into machine learning models, we saw a marked improvement in performance, especially in the M5 dataset. We implemented XGBoost, LightGBM, and Random Forest, all of which were better suited for handling non-linear patterns. We transformed the time series data into a supervised learning format, which allowed us to incorporate lagged features and use historical patterns for more accurate predictions. After hyperparameter tuning, LightGBM together with XGBoost delivered strong results, outperforming the traditional models.

Random Forest, another ensemble learning technique, showed potential but was less effective at capturing seasonal trends compared to these two. We also compared our results with baseline models like the Naive Forecast and Seasonal Naive Forecast. These simpler models, though not as sophisticated, provided a good baseline and highlighted the importance of seasonality in our data.

Ultimately, we decided to take an ensemble approach by combining the strengths of the top-performing models, LightGBM, XGBoost, and Exponential Smoothing for most of the cases. This strategy allowed us to improve our forecasting accuracy by combining the complementary capabilities of each model, which resulted in more robust and reliable predictions.

Conclusions

In conclusion, our project demonstrated the power of combining traditional statistical methods with advanced machine learning techniques for accurate sales forecasting. The models we developed can help Sandoz optimize its production and product distribution strategies, reducing the risk of both overproduction and stockouts. The lessons learned from this project not only

provide insights into sales forecasting but also set the stage for further improvements in handling real-world, dynamic data.

We must note that we started this project with a very low background in forecasting and temporal series, as we did not have previous experience in this particular sector, but we can confidently say that we have learned a lot and improved our knowledge during the process, something that will be really insightful for the future.

References

- [1] Fourkiotis, K. P., & Tsadiras, A. (2024). Applying Machine Learning and Statistical Forecasting Methods for Enhancing Pharmaceutical Sales Predictions. *Forecasting*, 6(1), 170-186. <https://doi.org/10.3390/forecast6010010>
- [2] Yu, C. (2022). Design of Drug Sales Forecasting Model Using Particle Swarm Optimization Neural Networks Model. *Computational Intelligence And Neuroscience*, 2022, 1-9. <https://doi.org/10.1155/2022/6836524>
- [3] Taylor SJ, Letham B. 2017. Forecasting at scale. *PeerJ Preprints* 5:e3190v2 <https://doi.org/10.7287/peerj.preprints.3190v2>
- [4] 7.1 Simple exponential smoothing | *Forecasting: Principles and Practice* (2nd ed). (s. f.). <https://otexts.com/fpp2/ses.html>
- [5] 7.2 Trend methods | *Forecasting: Principles and Practice* (2nd ed). (s. f.). <https://otexts.com/fpp2/holt.html>
- [6] 7.3 Holt-Winters' seasonal method | *Forecasting: Principles and Practice* (2nd ed). (s. f.). <https://otexts.com/fpp2/holt-winters.html>
- [7] XGBoost Paper. (2016, August 17). XGBoost: A Scalable Tree Boosting System. (s. f.) <https://www.kdd.org/kdd2016/papers/files/rfp0697-chenAemb.pdf>
- [8] LightGBM Paper. LightGBM: A Highly Efficient Gradient Boosting Decision Tree (s. f.) [LightGBM: A Highly Efficient Gradient Boosting Decision Tree](#)

[] InfoQ. (2018, 4 mayo). *Introduction to Forecasting in Machine Learning and Deep Learning* [Video]. YouTube. <https://www.youtube.com/watch?v=bn8rVBulcFg>

[] Ekhande, S. (2022, 5 enero). *M5 Forecasting-Accuracy - Shantanu Ekhande - Medium.* Medium.

<https://medium.com/@shantanuekhande19/m5-forecasting-accuracy-73343a873685>

[] DecisionForest. (2020, 20 octubre). *ARIMA Models for Stock Price Prediction*  How to Choose the p, d, q Terms to Build ARIMA Model (1/2) [Video]. YouTube.

<https://www.youtube.com/watch?v=gqryqlvEoM>

[] *M5 Forecasting - Accuracy. (s.f.).* Kaggle.

<https://www.kaggle.com/competitions/m5-forecasting-accuracy/code?competitionId=18599&sortBy=voteCount&excludeNonAccessedDatasources=true>

[] *Time Series Made Easy in Python — darts documentation. (s.f.).*

<https://unit8co.github.io/darts/>

[] Kaggle. (2020). *Kaggle Competition M5 Forecasting - Accuracy | Walmart Sales Unit Forecast* [YouTube playlist]. YouTube.

https://www.youtube.com/playlist?list=PLu4R9FIFLvWmRgHTH2Br-Di6koS_D7oL7