Technical Report Data Science in Action

"Unieuro / Jakala - Business Case 2: Market Projections"

Domenico Albertini 762411 – domenico.albertini@studenti.luiss.it

Simone De Benedictis 764941 – simone.debenedictis@studenti.luiss.it

Project: "The Crystal Ball of Market Forecasting"

SECTION 1: INTRODUCTION

With our project, "The Crystal Ball of Market Forecasting", we use sales market data provided to us by Unieuro from January 2015 to February 2022 to predict sales for the following year (from March 2022 to February 2023). The provided data contains historical time series of the sales for each Product Group, Brand, and Sector. Therefore, our goal is to predict *offline sale*s (sales in physical retail stores) and *online sales* (sales made through the website) individually for each product group, brand, and sector during the mentioned period. This goal is achievable thanks to methods and models of time series forecasting.

SECTION 2: METHODS & SECTION 3: EXPERIMENTAL DESIGN

Approach to Model Definition: Techniques, Criteria, and Assumptions of our Analysis

2.1 Exploratory Data Analysis

Our dataset, in addition to the aforementioned offline and online sales, consisted of other variables, such as Mass Merchandiser sales, Technical Superstore sales, sales divided by North, South, East, and West, and others. However, for our analysis and the specific task required by the project, these variables were not important. Since we were only asked to predict online and offline sales, we decided to keep only these latter variables and the total sales (simply the sum of offline sales and online sales).

How many time series?

Our first objective was to understand the number of *product groups* for each *brand* in each *sector*, in order to determine the number of time series we would be working with. We achieved this by using the *groupby* function to group the data by *brand and product group* and creating a dictionary where each element had the key "brand, product group" and as the value, the *dataframe* with the specific *groupby* of that key (it is shown in Figure 1 to be clearer). This allowed us to identify a total of 3217 *brand-product group* combinations for which we needed to work. Since both online and offline sales data were available for each combination, we had to double the number of time series, resulting in a total of 6434. This dictionary proved to be a useful tool for us as we were able to apply the models to the *dataframes* within the dictionary, avoiding the need to create 6434 different *dataframes* and call each one individually.

Next, we filtered the time series, selecting only those that contained at least 12 months of data. This was necessary as the models cannot make accurate predictions with too few data. We identified a total of 1880 time series that met this criteria, which we doubled to account for both online and offline sales data.

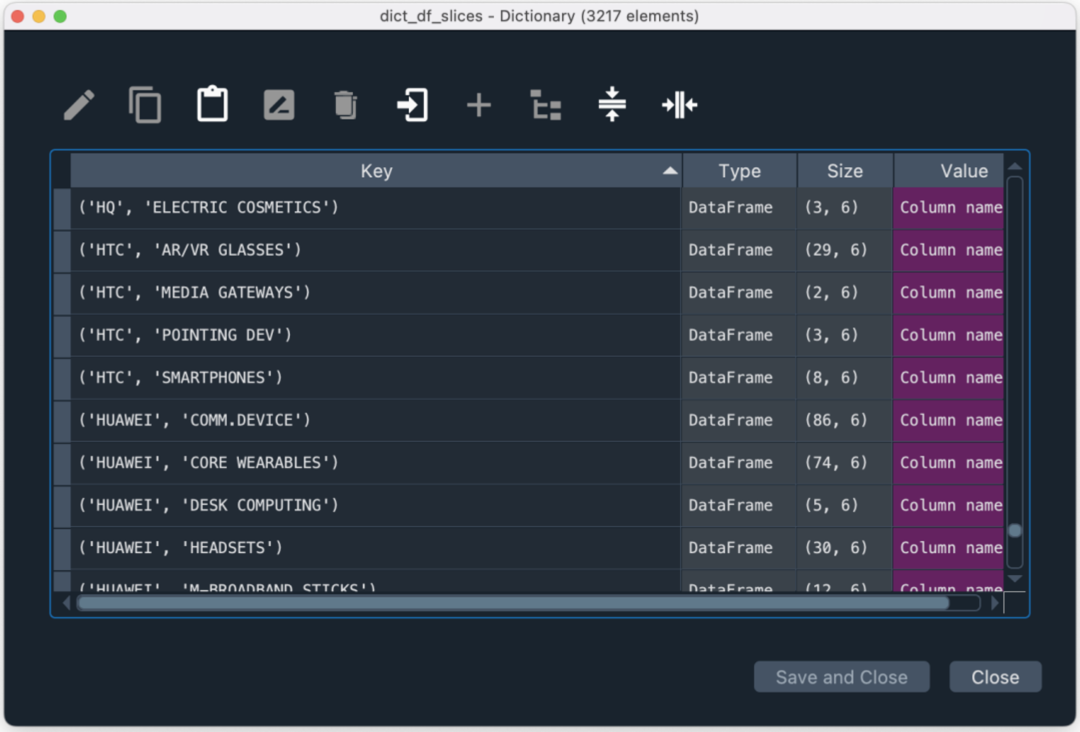


Figure 1

For the remaining time series, which contained less than 12 months of data, we applied a mean and used this as the prediction for future years. This approach was taken as these time series largely represented product groups that were no longer being sold and were thus no longer of interest to Unieuro.

Analysis of Time Series Length and Decision on Train-Test Splitting

To determine the appropriate train and test splitting method, we examined the size of our 1880 time series that we filtered before. *Figure 2* shows the distribution of time series size, with the left column indicating the number of months in each series, and the right column indicating the number of time series with that size. As we can easily see, the vast majority of our time series contained 86 months of data, indicating that they were complete. After conducting research and consulting with other projects, we decided to use the last year of data (13 months) as *test set*, and the remaining months (86-13=73) as *training set*. Upon applying the model to a subset of complete time series, we found that this splitting method seemed to work well for our purposes. We observed that this proportion corresponded to an 85% training and 15% test split, and we decided to keep this proportion to the rest of the time series with fewer months of data, where using an entire year of data as test set was not reasonable.

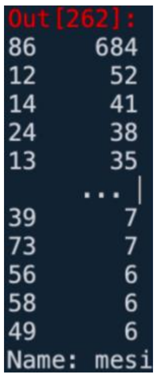


Figure 2

Identification of missing values

Another useful analysis in the *EDA* stage was to identify how many time series had missing values. By missing values, we do not mean *NaNs*, but the missing months that would make the series incomplete, thus generating time jumps. We noticed, in fact, that out of the total 3217 time series, 1668 were incomplete. Considering only the 1880 time series on which we will apply the models, 985 had missing months. We believed it was premature to make decisions regarding the handling of values in the missing months, as we found in our research that some models were able to handle them without any imputation, while others required it. Therefore, we deferred the decision until we applied the models, managing them based on the attempted and utilized model.

Regarding other analyses that are typically useful in the EDA phase for machine learning models (such as boxplots for outlier detection or correlation analysis), we did not perform them, as they are not particularly helpful for time series data. Carrying out these analyses just for the sake of doing them, without drawing any meaningful conclusions, did not seem sensible to us.

General Trend

To choose, test, and analyze the models that best fit our case, we decided to study and train them using a specific time series that had a trend similar to the trend of the general cumulative sales, both for offline and online sales. Based on our EDA insights, we chose the "SMARTPHONE" product group (the most-selling one) and the "APPLE" brand (the second-most-selling one for the offline sales and the most-selling one for the online). *Figures 3* and *Figure 4* show the trend of the overall cumulative sales for all product groups and the time series trend for *Apple Smartphone*, respectively. As can be seen, the time series for *Apple Smartphone* appears to have a similar seasonality to that of the overall sales, with positive and negative peaks occurring in the same months. Therefore, we identified *Apple Smartphone* as a series that reflected the general trend of all the available series for both offline and online sales and we decided to study the models using this one.

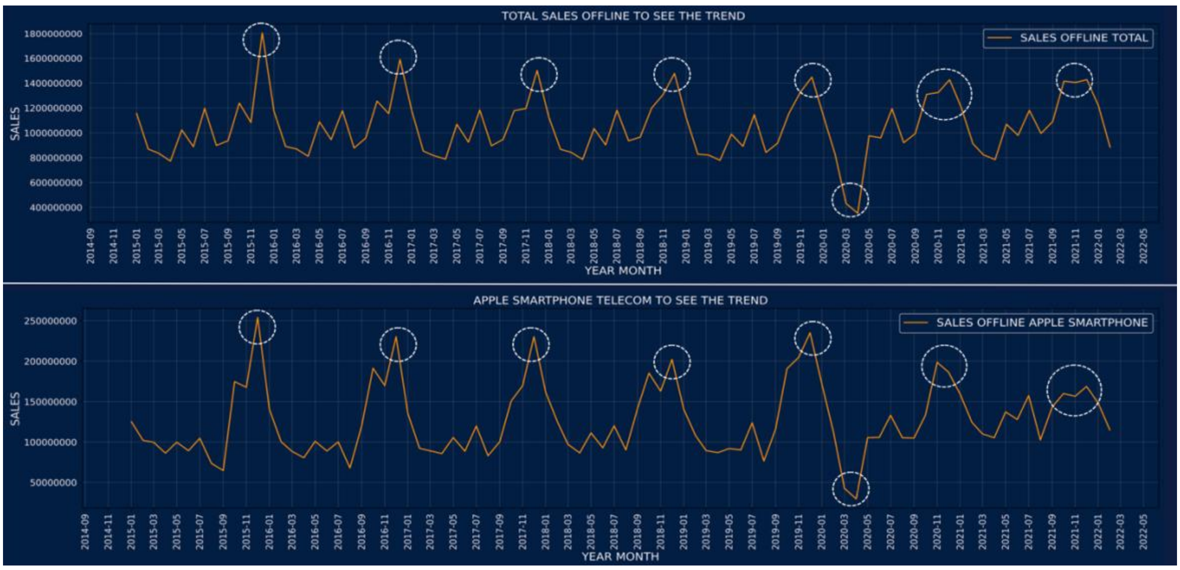


Figure 3

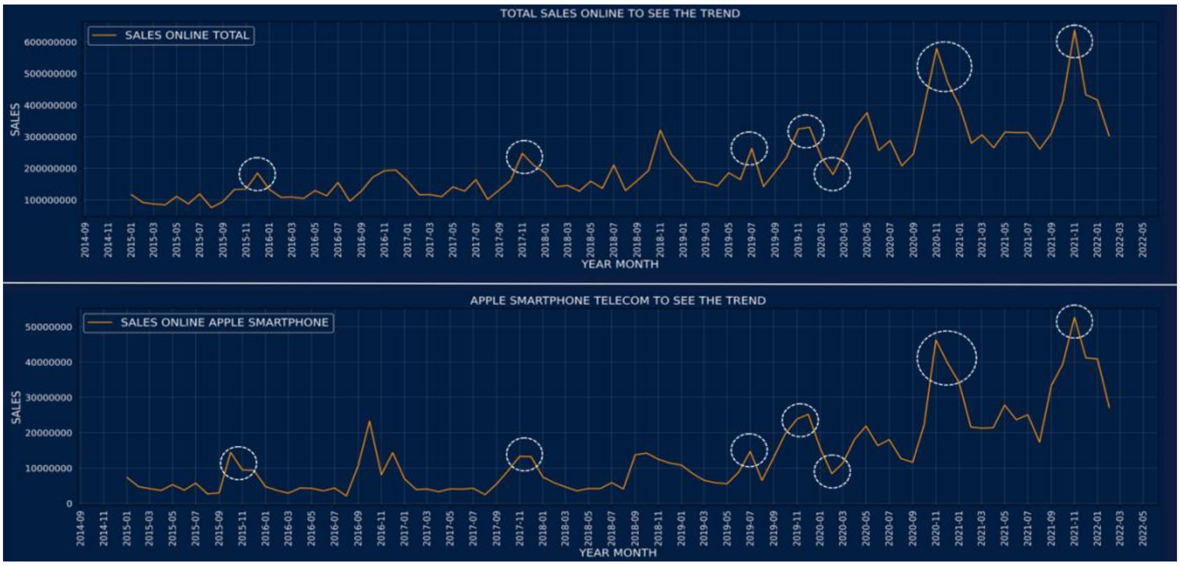


Figure 4

Business Analysis

Before delving into the description of our approach in defining and applying models, we would like to dedicate this brief paragraph to examining some economic characteristics that we explored during the EDA phase. These insights will be invaluable for drawing business conclusions later on for the company that commissioned the study.

Starting with an analysis of the sectors, as seen in *Figure A* below, there are a total of nine sectors, with the market share of each being displayed. Furthermore, the top five sectors account for 95% of total sales, while the top seven encompass 99%.

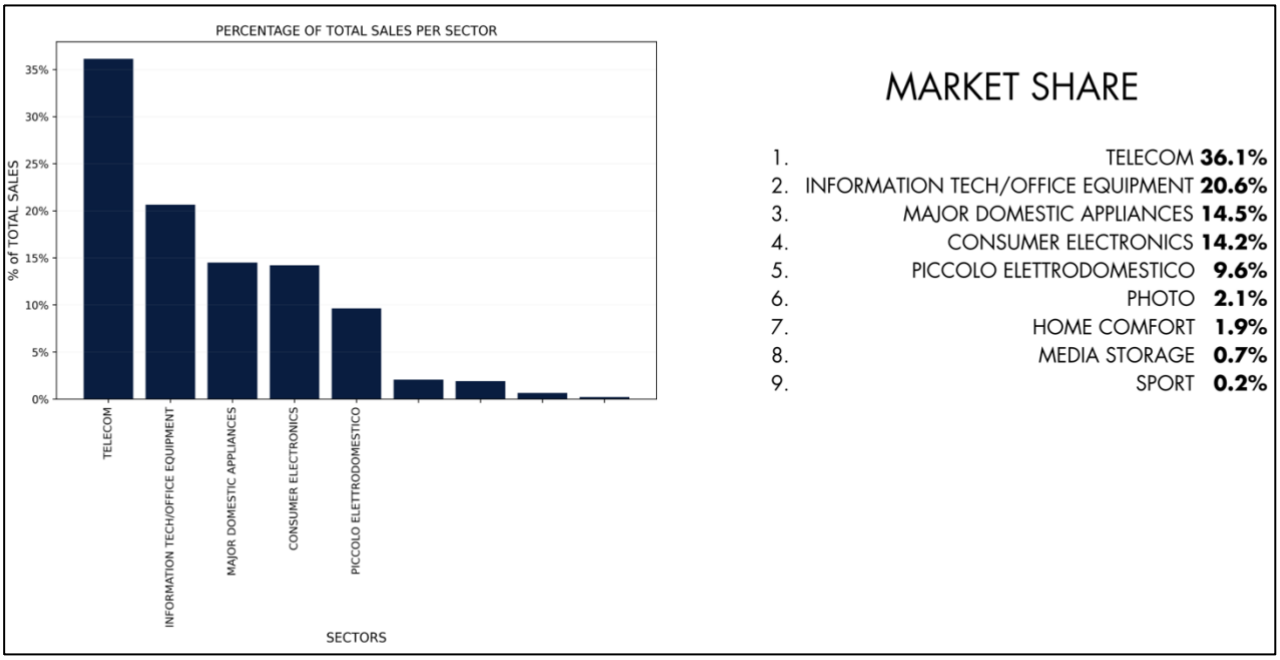


Figura A

Regarding the Product Groups, out of a total of 117 product groups, the top 5 product groups represent 58% of total sales, as shown in *Figure B* below. The market shares of these top 5 product groups can also be seen in the same figure.

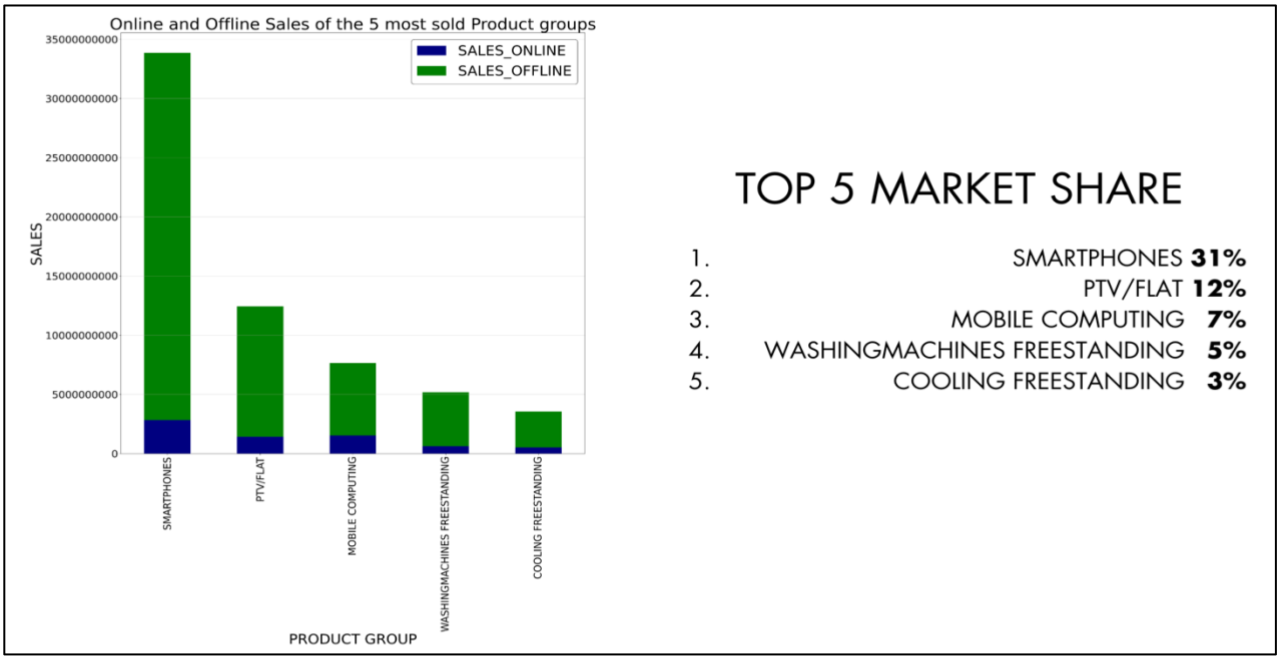


Figure B

2.2 Model selection

The metric chosen to evaluate the performance of the model is the MAPE (Mean Absolute Percentage Error), that is a common metric used to evaluate the performance of time series forecasting models. It expresses the average percentage error between the observed values and those predicted by the model. MAPE is a good metric for time series because it takes into account the proportion of errors relative to the observed values. In other words, MAPE penalizes larger percentage errors, making it particularly useful for models that need to predict time series values where the values themselves can vary significantly. This makes MAPE a useful metric for evaluating model performance and communicating results to business decision makers. Indeed we have chosen the MAPE as our evaluation metric, even because it is considered one of the most business-friendly. This means that it is highly interpretable, and given that we must deliver the data to the company's business department as well, it seemed like an excellent choice for our purposes.

For all the models we will present in the following illustrations, we will focus on the *Apple Smartphone* time series for the reasons mentioned in the "General Trend" paragraph. It is common practice when illustrating time series to show the entire series (including both train and test data), but in this case, we do not want to focus on the specific numbers or general trend, but exclusively on how well or poorly the models perform on the test data. Therefore, in the following graphs, we will only show the test values of *Apple Smartphone* time series compared to the predicted values of the models.

To evaluate the goodness of the performance of our models, we defined a benchmark to compare them against. The chosen benchmark was the mean, so we computed the mean of the training values and kept it constant for the time window covered by the test set. We then compared this mean to the test set values and calculated the MAPE. For the specific *Apple Smartphone* time series, *Figure 5* shows the test set values and the mean, which has a MAPE of 14.79%. The goal is therefore to find one or more models that significantly improve this performance by reducing this percentage.

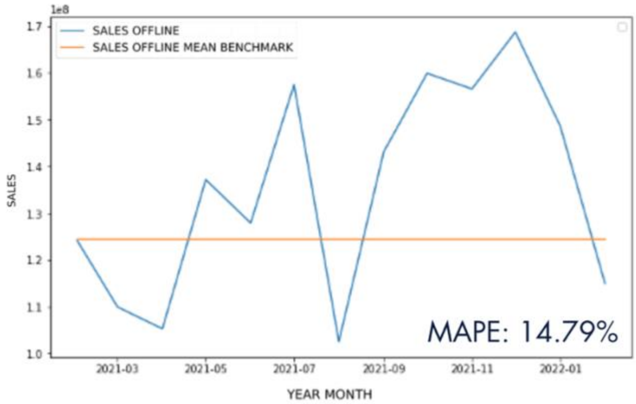


Figure 5

As a first approach in the study and application of models, we started with ARIMA (Autoregressive Integrated Moving Average) models. ARIMA is a time series analysis model that combines the components of an autoregressive model (AR), a moving average model (MA), and an integration process (I). Its popularity primarily derives from the fact that it is a simple yet powerful model for time series analysis, as it can capture many of the characteristics of time series, including trends, seasonality, and cyclicality.
Running the first ARIMA model with a brief *stepwise* of *autoarima* with three parameters to obtain the optimal order, the first result was rather disappointing. The MAPE was similar to that of our benchmark, and as can be seen from Figure 6, the model was not able to capture the seasonality.

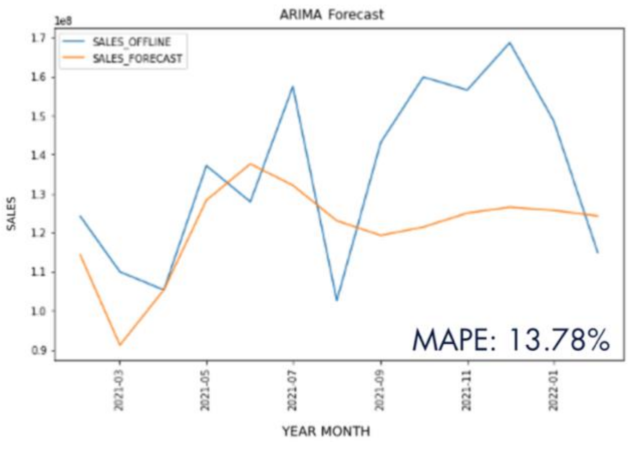


Figure 6

Upon further investigation, we have read that the first assumption of ARIMA models is that the data should be stationary (meaning that the mean and variance must be constant over time). We conducted the *Augmented Dickey-Fuller (ADF) test*, which is a statistical test that determines whether a time series is stationary or not if the p-value is less than a threshold of 0.05 or 0.1. As can be seen from *Figure 7*, based on the p-value, our original data were non-stationary; therefore, we conducted some transformations to make them stationary.

Test Statistic	-2.514373
p-value	0.112016
#Lags Used	12.000000
Number of Observations Used	60.000000
Critical Value (1%)	-3.544369
Critical Value (5%)	-2.911073
Critical Value (10%)	-2.593190

Figure 7

After applying a *log transformation* and a *lag shift transformation* (i.e., t-t-1), we were able to make our data stationary and applied the model. After transforming the data back to the original scale to evaluate the model performance, surprisingly, the model performed even worse than before (*Figure 8*).

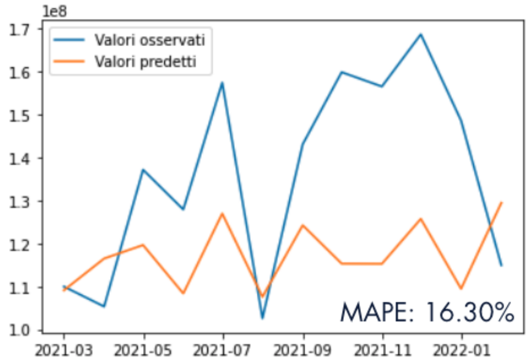


Figure 8

Initially considering our data unsuitable for ARIMA models, we opted to use the FB Prophet model. Prophet is particularly useful for time series that exhibit seasonality, long-term trends, and irregular fluctuations, as it can capture these patterns with greater accuracy than other time series forecasting models. Flexibility, robustness, interpretability, and ease of use are some of the reasons why Prophet is convenient for our case study. Upon studying Prophet, we learned that we could include a *holiday_off* in the model, which is a time window to which the model should assign less weight or even zero weight. This insight was perfect for our case where the months of the COVID lockdown from March 2020 to May 2020 caused an unexpected negative peak in offline sales due to the closure of retail stores and strict circulation limits. By treating COVID as *holiday_off* and conducting a brief *grid search*, we already noticed that the Prophet model was better able to capture seasonality and reduce the MAPE to 10.36% (*Figure 9*).

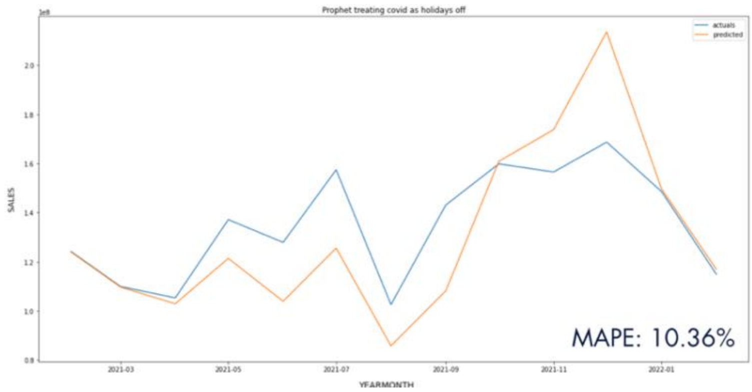


Figure 9

Going back to our previous steps, we decided to retry the ARIMA model with the insight of treating the Covid pandemic as a *holiday_off*, as what was done with Prophet. Upon further research, we discovered that we could perform a parallel stepwise search for the seasonal order in addition to the other search for the parameters order and suggesting a seasonality of 12 months to the model. The result was finally positive, with the MAPE decreasing further to 8.64%, and the model successfully capturing the seasonality (*Figure 10*).

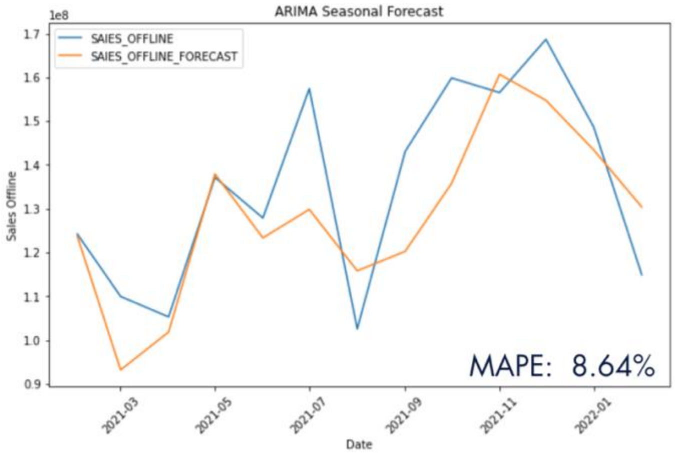


Figure 10

We wanted to go further and push the boundaries. We try an XG Boost model that used *n* previous *time steps* to predict the next value, appending the predicted value as a time step to predict the next one to avoid the *look ahead bias*, i.e. the use of test data during the training. Although the model had plenty of room for improvement and despite the fact that the seasonality was well captured (Figure 11), we noticed that the computational cost was very high to obtain predictions with a similar or higher MAPE compared to those obtained with the ARIMA and Prophet models. These last two had managed to obtain similar or better accuracy results with a few minutes of computation, unlike XG Boost.

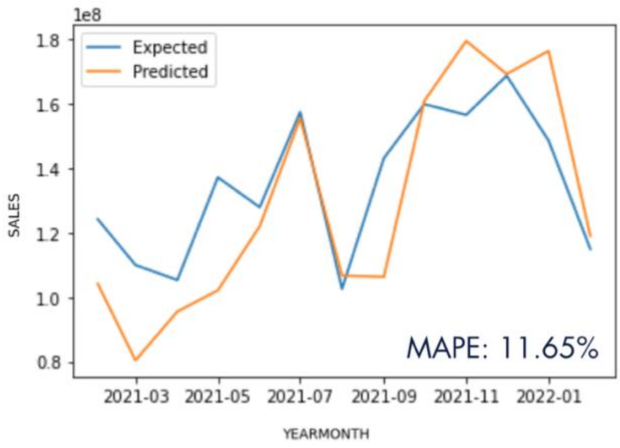


Figure 11

To indulge our curiosity, we decided to also try a Neural Network model: LSTM. We chose to apply this type because, unlike traditional neural networks, which process each input data independently of the others, recurrent neural networks like LSTM can process sequential data and maintain a sort of "memory" of past information. This makes them particularly suitable for time series, where the order of events is important and past observations can influence future observations. However, the conclusions were similar to those of XG Boost. The training time of the model and the grid search was extremely high, even taking hours for this single time series. Another consideration is that models like LSTM generally perform better than simpler models (such as ARIMA and Prophet) with a large amount of data. In our case, having only 86 values for the longest time series, the performance was not even excellent considering the high computational cost. These arguments can be proved from *Figure 12*, which shows several LSTM models trained differently.

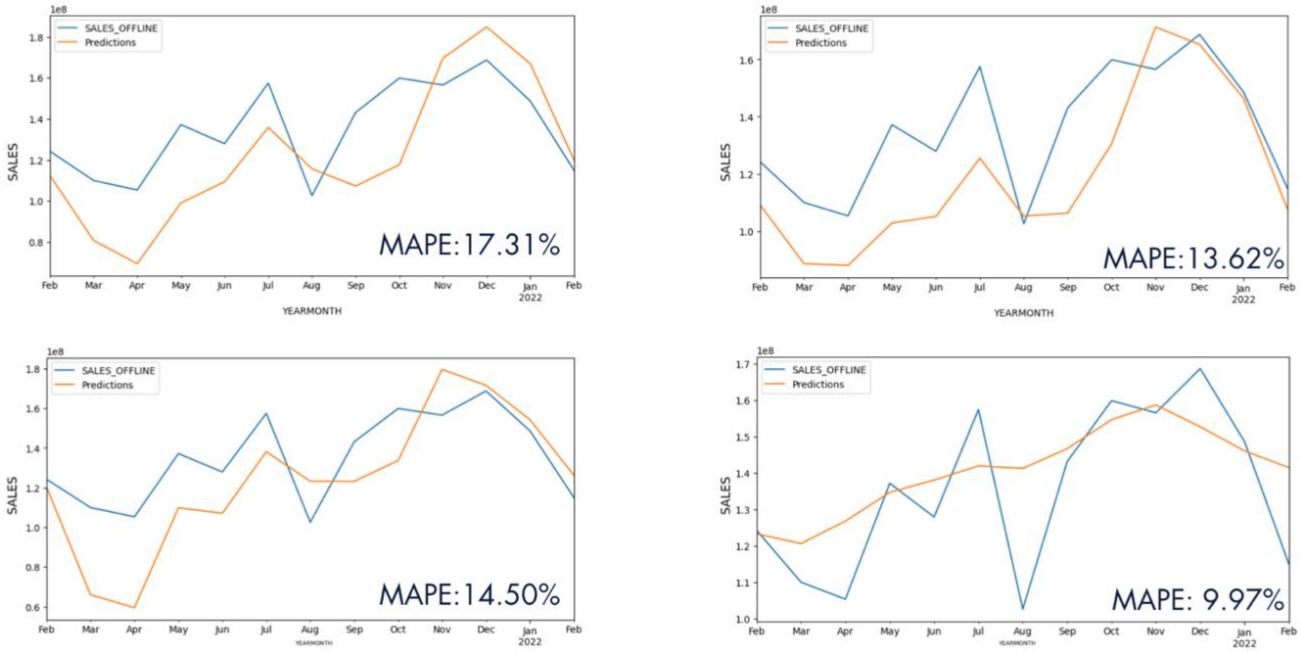


Figure 12

Key points to address before finalizing our approach.

After testing the previously mentioned models on other time series, we observed that in many of them Prophet exhibited superior performance over ARIMA. Therefore, we arrived at the definition of a final approach to pursue, in which we will apply both ARIMA and Prophet models, which will be described in the next paragraph.

Moreover, As previously mentioned, we would handle the missing values in the time series data based on the models we would be using. By applying the model to time series with missing months, we observed that Prophet was able to handle them without the need for imputation, while ARIMA required it. As we already described in the EDA paragraph, we have observed that there were as many as 985 such time series among those to which we planned to apply our models. We decided to handle this issue by using linear interpolation when we would apply the ARIMA models. This involves connecting the point before and after the missing value, taking an average of these two values, and imputing the missing value in this way.

SECTION 4: RESULTS

Final approach and application of final models

As a final approach, we applied the ARIMA and Prophet models to the 1880 offline sales time series (treating COVID as a *holiday_off*) and to the other 1880 online sales time series. For Prophet we tuned for each time series the Prophet model with a Grid Search. For ARIMA we tuned for each time series the model with the two-stepwise described earlier and interpolating the missing values.

Then, we compared the MAPE of ARIMA, of Prophet and of the benchmark mean for each time series. Based on these comparisons, we used an if-statement to select the best-performing model for each time series and we predict the future sales using that model. In cases where the benchmark mean performed better than ARIMA and Prophet, we used the *Naive* method of predicting future sales by copying the values from the previous year. We considered *Naive* to be better than the mean as it is reasonable to assume that sales values from the previous year will not differ significantly from future values in terms of seasonal peaks. This is precisely because the mean is not able to represent seasonality being a constant. We also used the Naive method in cases where the MAPE of the models was smaller than that of the mean, but still considered excessive, i.e., above 100%. This was mainly observed in time series with few months of data or low sales volume, where seasonal peaks were unpredictable or random. The table below shows the approach of comparing MAPEs and choosing the best model, displaying only the top five *product groups-brands* in term of offline sales due to space constraints, but the same approach was used for the online sales.

TIME SERIES (BRAND, PRODUCT GROUP)	**MAPE MEAN**	**MAPE ARIMA**	**MAPE PROPHET**	**CHOSEN MODEL**
SAMSUNG, SMARTPHONE	36.8 %	13.7%	<u>10.0 %</u>	PROPHET
APPLE, SMARTPHONE	14.8 %	<u>9.6 %</u>	14.3 %	ARIMA
HUAWEI, SMARTPHONE	955.4%	165.1%	125.1%	NAIVE
SAMSUNG, PTV/FLAT	31.7 %	19.3 %	<u>14.8 %</u>	PROPHET
LG, PTV/FLAT	33.5 %	21.9 %	<u>15.7 %</u>	PROPHET

Overall, concerning offline sales, for the target period forecast, were used: 395 ARIMA models (21% of the total), 1002 Prophet models (53%), and 483 Naive (26%). For online sales, instead, 469 ARIMA models (25%), 673 Prophet models (36%), and 738 Naive (39%) were used.

On the other hand, as previously mentioned, for the remaining 1337 time series, those with less than 12 months of data, we applied the mean for both offline and online sales.

SECTION 5: CONCLUSION

Results and Business Considerations after Forecasting

To present the results to Unieuro's business department, we decided to conduct additional predictions by grouping the economic features described earlier in the EDA. This allowed us to perform a more macro-level analysis compared to the micro-level analysis for each *brand-product group* previously explained.

Sector Analysis

For the sector analysis, we applied the models to the total sales of the individual nine sectors, obtaining their respective time series through a groupby operation. Since we discovered during the EDA that the last two sectors by market share ("media storage" and "sport") did not even account for 1% of sales, we excluded them from this analysis. This was primarily because these two, without a recurring seasonal trend and having data significantly different from the other time series on which we had studied the models, both Prophet and ARIMA were unable to perform as desired. Therefore, our focus was on the remaining seven sectors, which together represent 99.1% of total sales. In *Figures 13* and *Figure 14*, only the models applied to two sectors are shown due to space constraints, but the approach was the same for the others as well. For both figures, in the first plot, we show the model and its performance on the test data, while in the second plot, we present the forecast for the following year alongside the growth percentage predicted in our target year.

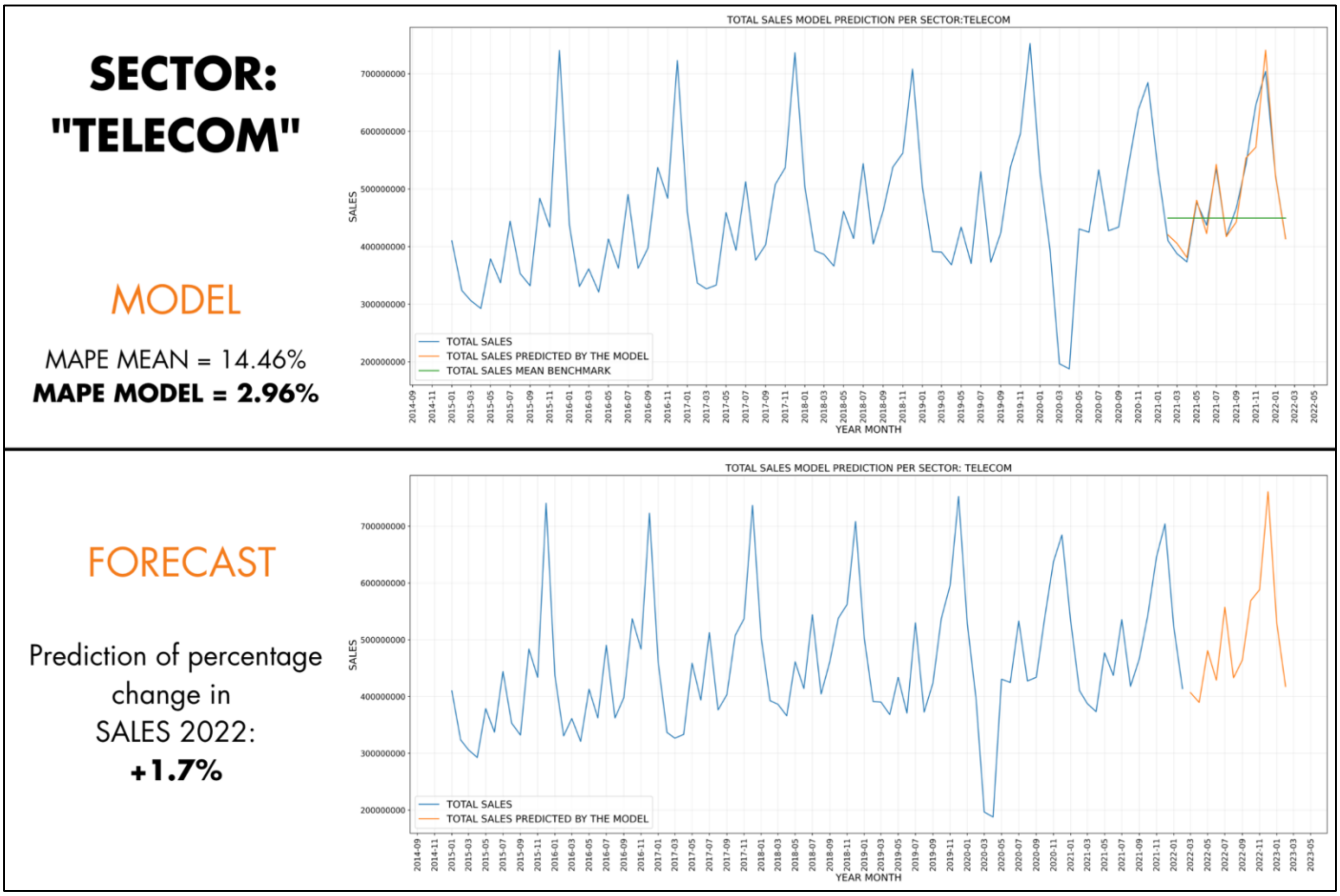


Figure 13

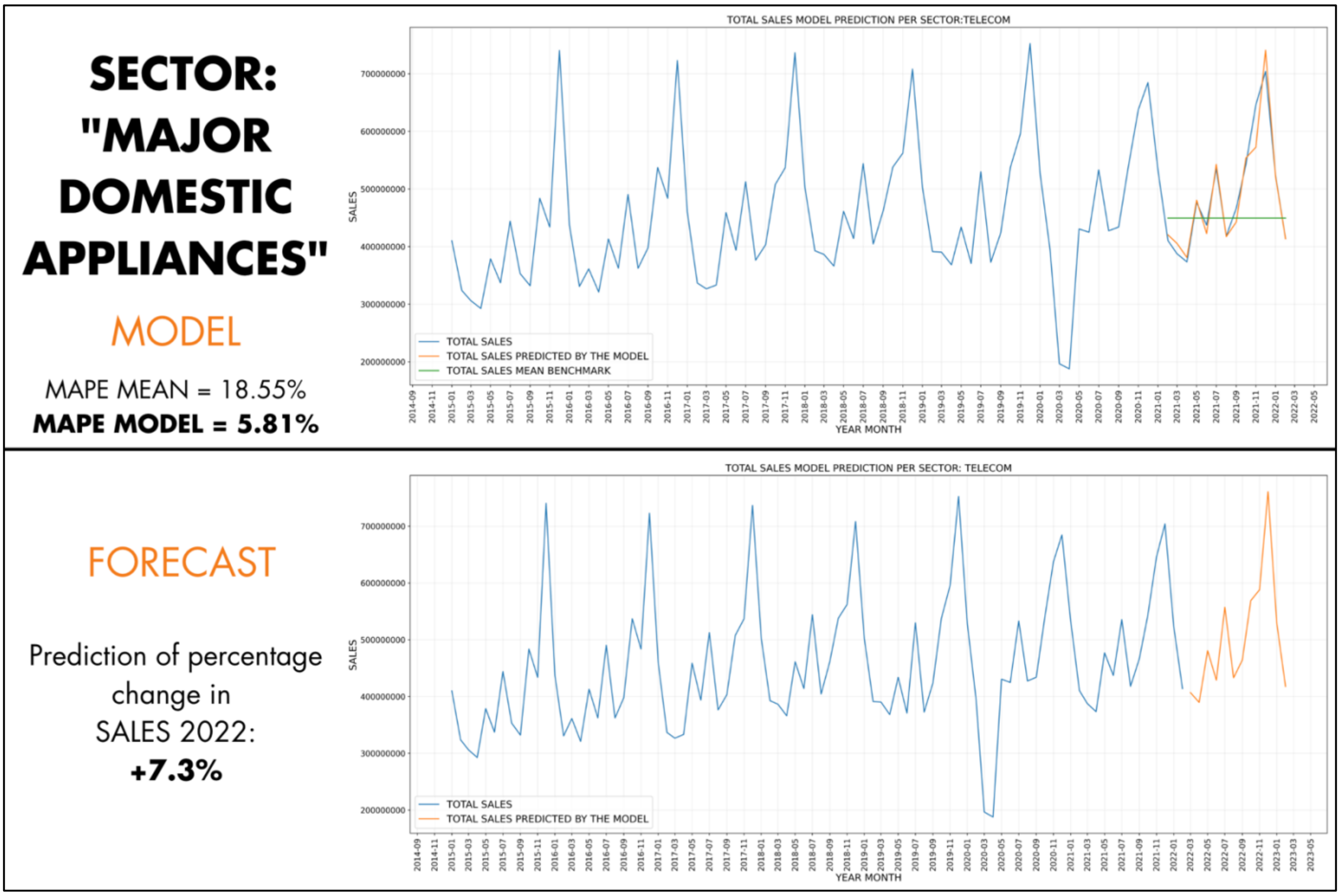


Figure 14

As can be seen, the models perform pretty well, and using the same approach for the others, we have found that: the sector with the highest growth will be "consumer electronics" (+34%), followed by "major domestic appliances" (+7.3%) and "home comfort" (+5.1%). The sector that will experience the most severe decline is "information technology/office equipment" (-20.9%). The others, however, will remain more or less at the same level as the previous year. The growth/decline percentages for these seven sectors can be seen in the bar plot shown below (*Figure 15*).

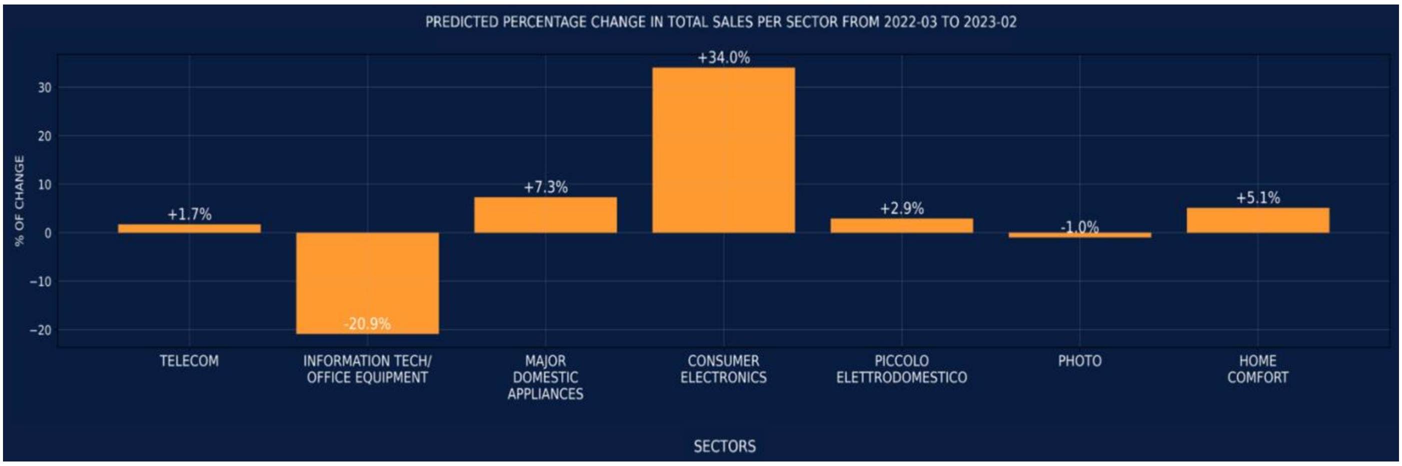


Figure 15

To even compare how the market share of the sectors will change, *Figure 16* displays the market shares for the last year of data 2021, and the predicted ones for 2022. As can be seen, the sector that will experience the most significant increase in its market share is "consumer electronics," rising from 14.5% to 21.6%. On the other hand, the sector that will see the most substantial decrease is "information technology/office equipment," dropping from 21.3% to 16.3%. The remaining sectors will not undergo any significant change in their market share.

Figure 16

Product Group Analysis

Regarding the Product Groups, we used the same approach as in the sector analysis to calculate the growth of total sales in the following year. The focus was on the top 5 product groups, which, as already seen in the EDA, cover 58% of total sales. In *Figures 17* and *Figure 18*, we show only two of these five product groups, as with the sectors, demonstrating both the performance of the applied model and the prediction for the following year with the percentage change.

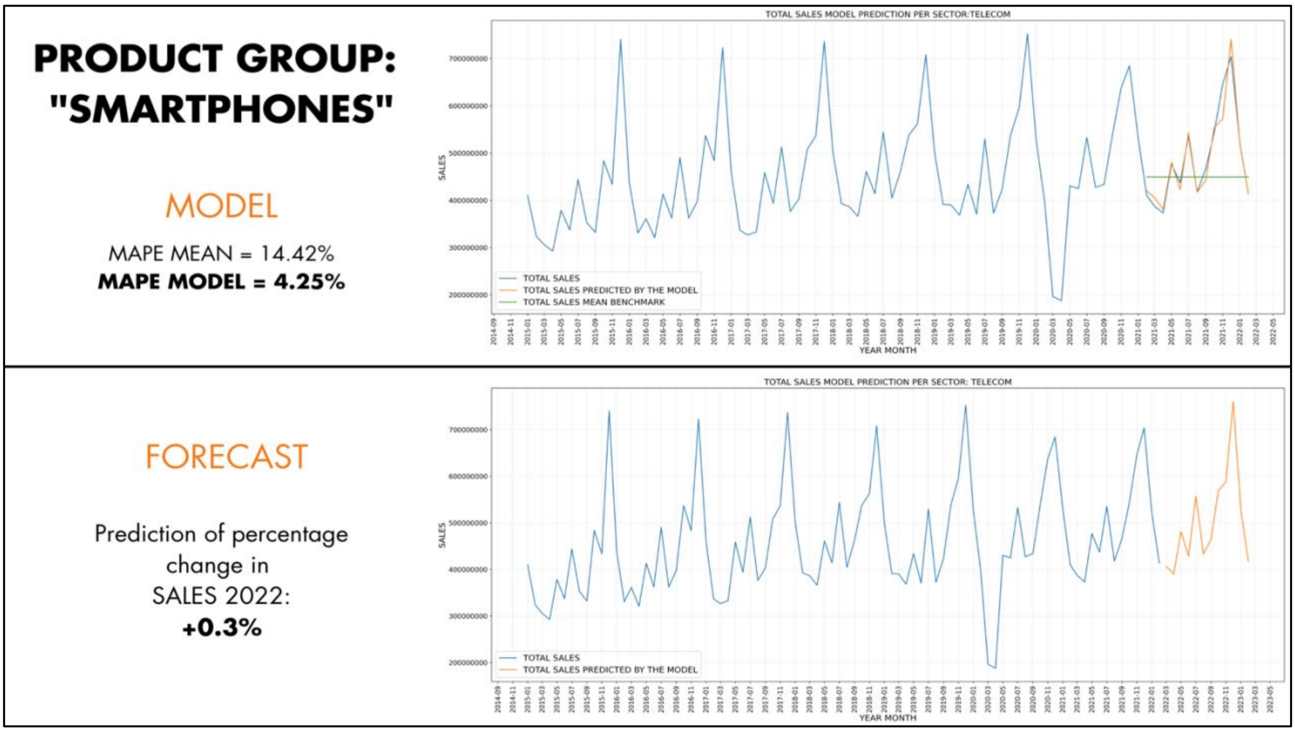


Figure 17

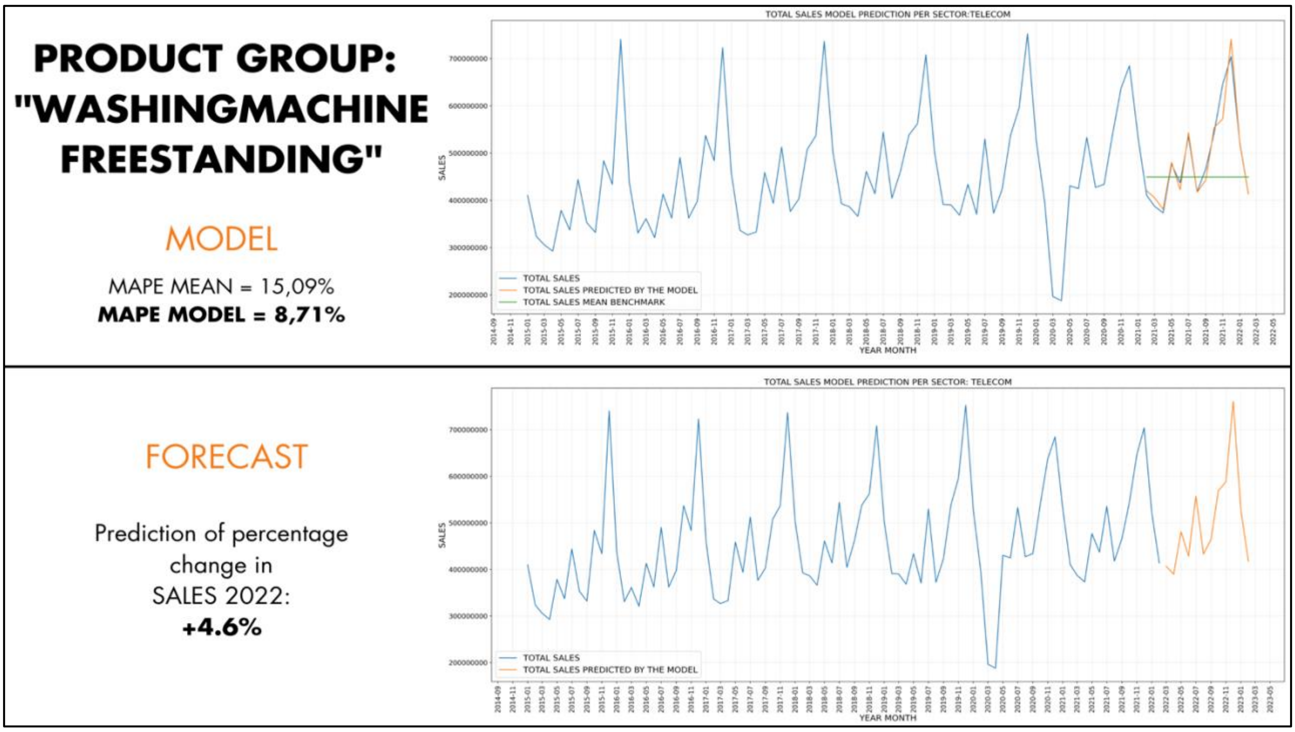


Figure 18

In this case as well, the models' performances are satisfactory. Therefore, applying them to the other three of the top 5 product groups, we can observe that: the product group with the highest growth in terms of total sales is "Washing machine freestanding" (+4.6%). "Mobile computing" and "IPTV/FLAT" will experience a negative growth of -14.7% and -4.3%, respectively. The remaining two product groups will not undergo any significant change. These percentages are shown in the bar plot below (*Figure 19*).

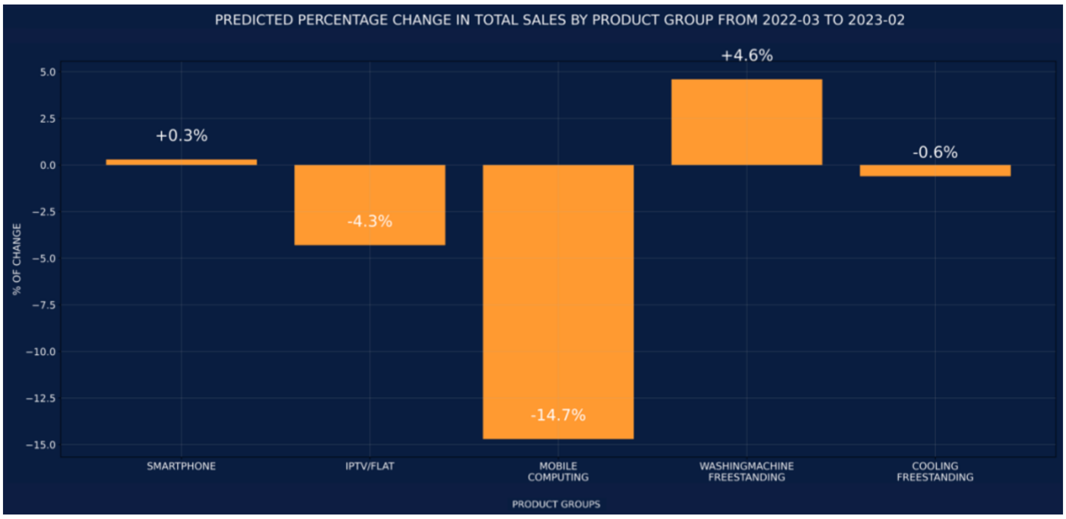


Figure 19

As can be seen, three out of these five product groups will experience a negative change. However, it is essential to specify that these are the top 5 product groups. Consequently, this does not mean that they will lose appeal among consumers (they will still be the most-selling ones), but rather that the gap with the others will be slightly reduced.

Offline and Online sales Analysis

As a final business analysis, we conducted an even more macro-level study, aiming to predict how sales by channel (offline and online) will change in the target period, comparing the percentage changes recorded in previous years as well.

Regarding offline sales, we applied the Prophet model to the series generated by summing up all offline sales for all product groups and brands on a month-by-month basis. In *Figure 20*, the model's performance on the test is shown, compared to that of the benchmark, along with their respective MAPE values.

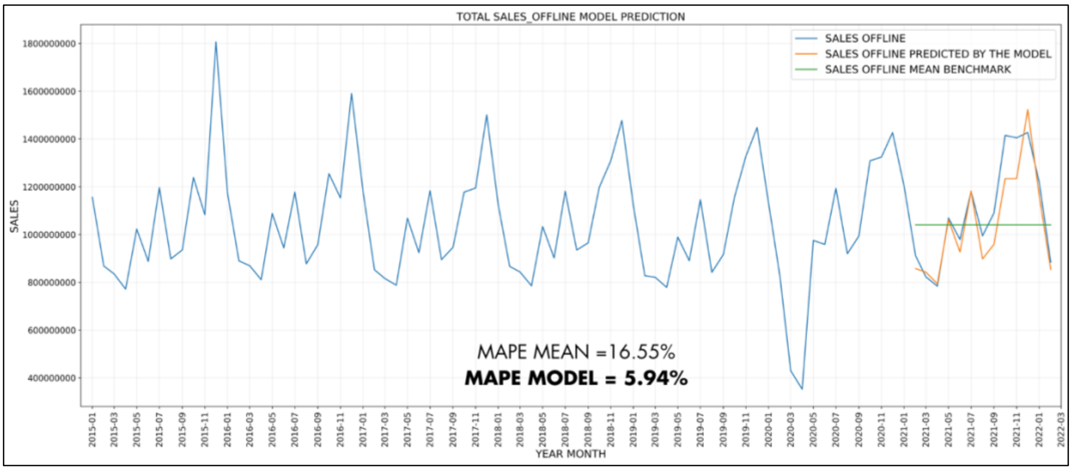


Figure 20

After completing the forecast and thus initiating the prediction for the following year, we calculated the annual growth percentage recorded in previous years and the one predicted for the future by the model. As shown in *Figure 21*, the forecasted growth for offline sales is +2%. This is quite plausible since the trend had already registered a significant growth of +10.6% in 2021, due to the substantial easing of lockdown restrictions imposed during the Covid pandemic in the two previous years. Obviously, the restrictive measures had prohibited and limited circulation in some months, leading to negative peaks in sales at physical retail stores. As previously mentioned, further growth is expected in 2022-23, when pandemic limitations will totally no longer be present.

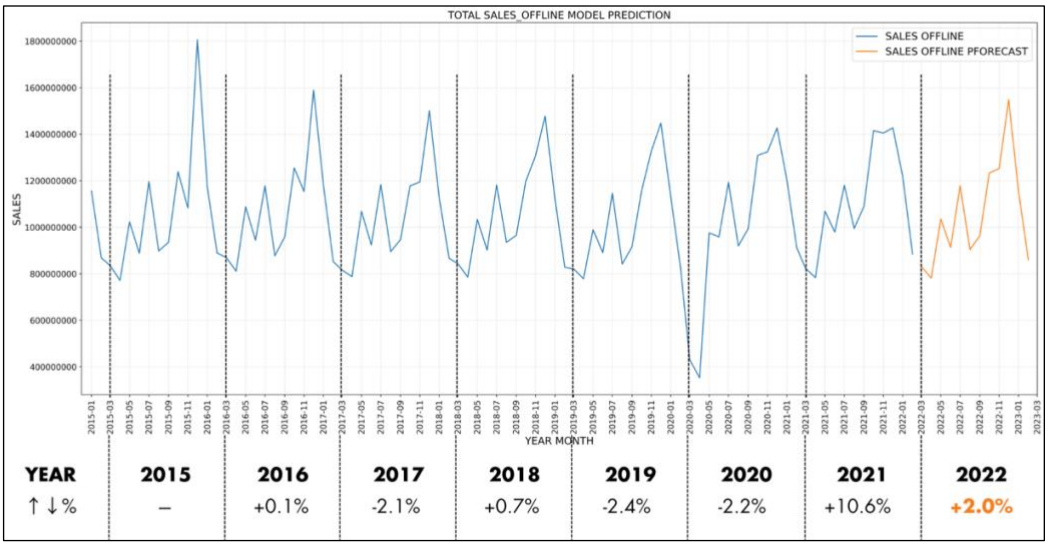


Figure 21

Regarding online sales, *Figure 22* displays the performance of the applied model, which is useful for conducting a similar analysis as described for offline sales and drawing further conclusions. However, it can be observed that the model was not entirely successful in capturing the positive peak of December perfectly, which should be taken into account in the subsequent analysis.

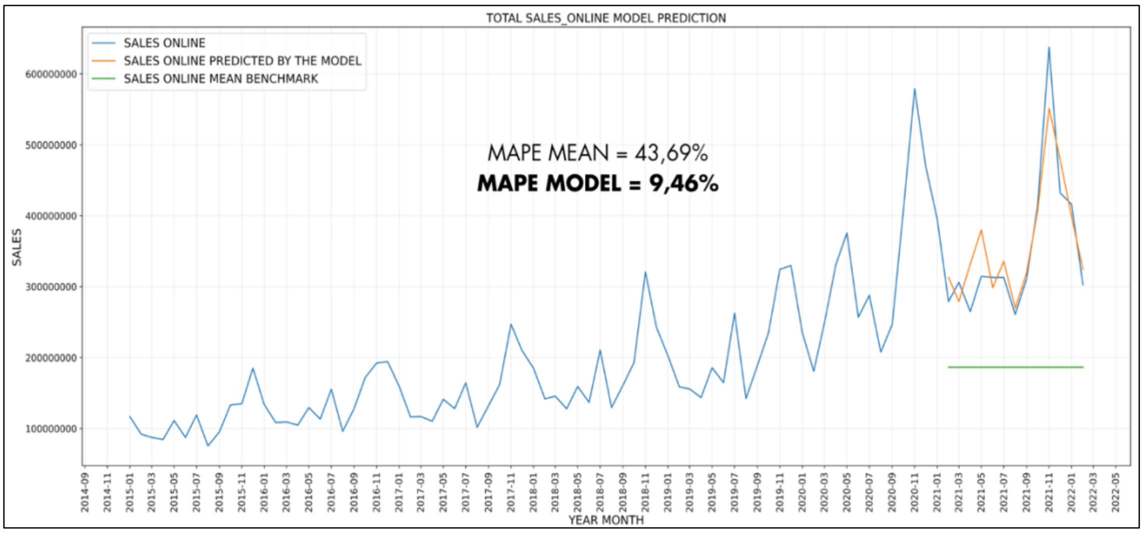


Figure 22

As can be seen from *Figure 23*, the trend of online sales has been consistently growing since 2015. In 2020, while offline sales experienced a decline due to COVID-19, online sales logically saw substantial growth (+53.4%), as online stores replaced offline ones entirely during certain months. In the target year, the model predicted a growth of +11.3%, in line with the positive trend of previous years. However, as mentioned earlier, the model was not able to capture the growth peak recorded in December. Therefore, we could expect the growth to be slightly higher than the predicted 11.3%.

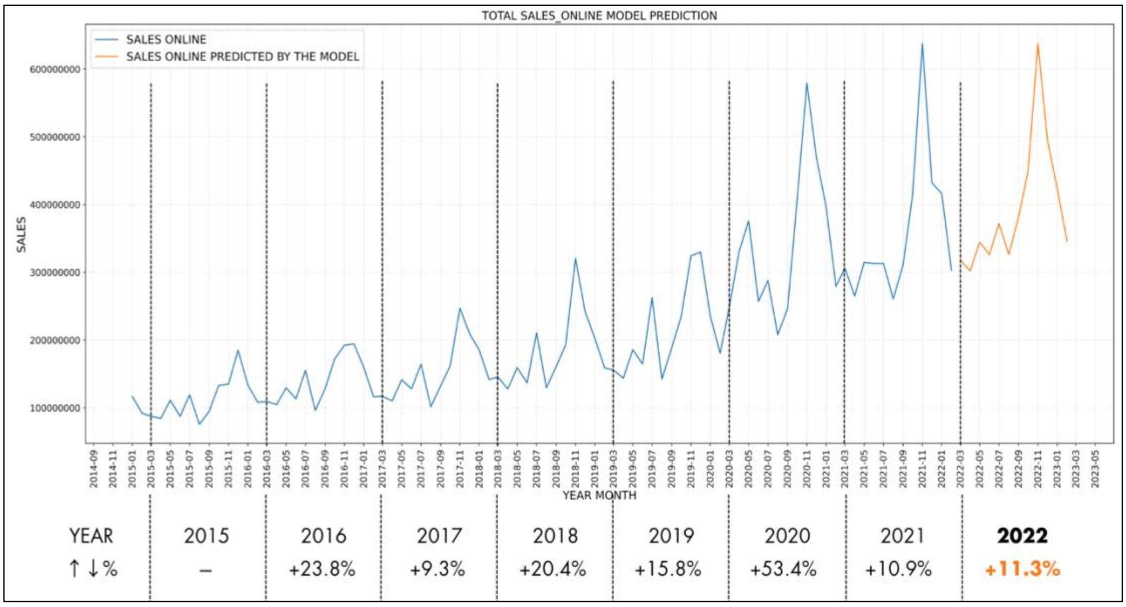


Figure 23

Moreover, since the model previously applied to the aggregated offline sales performed quite well (MAPE: 5.9%), we thought: why not compare this with the sum of the forecasts of the individual product group-brand for all the time series described in Section 2?

The result was astonishing. The trend, seasonality, and shape of the line plot are almost identical. The comparison is shown in *Figure 24*, where the forecast of the model applied to the aggregated offline sales is displayed above, and the sum of the forecasts of the models applied to the individual time series is displayed below. This provides substantial evidence that the models on the individual time series have predicted well, and therefore, the single forecasts requested by Unieuro have proven to be particularly effective. Of course, there is a difference in the volume of sales, but this is due to the fact that the error on the individual 1880 time series is obviously greater than the error applied to the single offline sales time series offline sales.

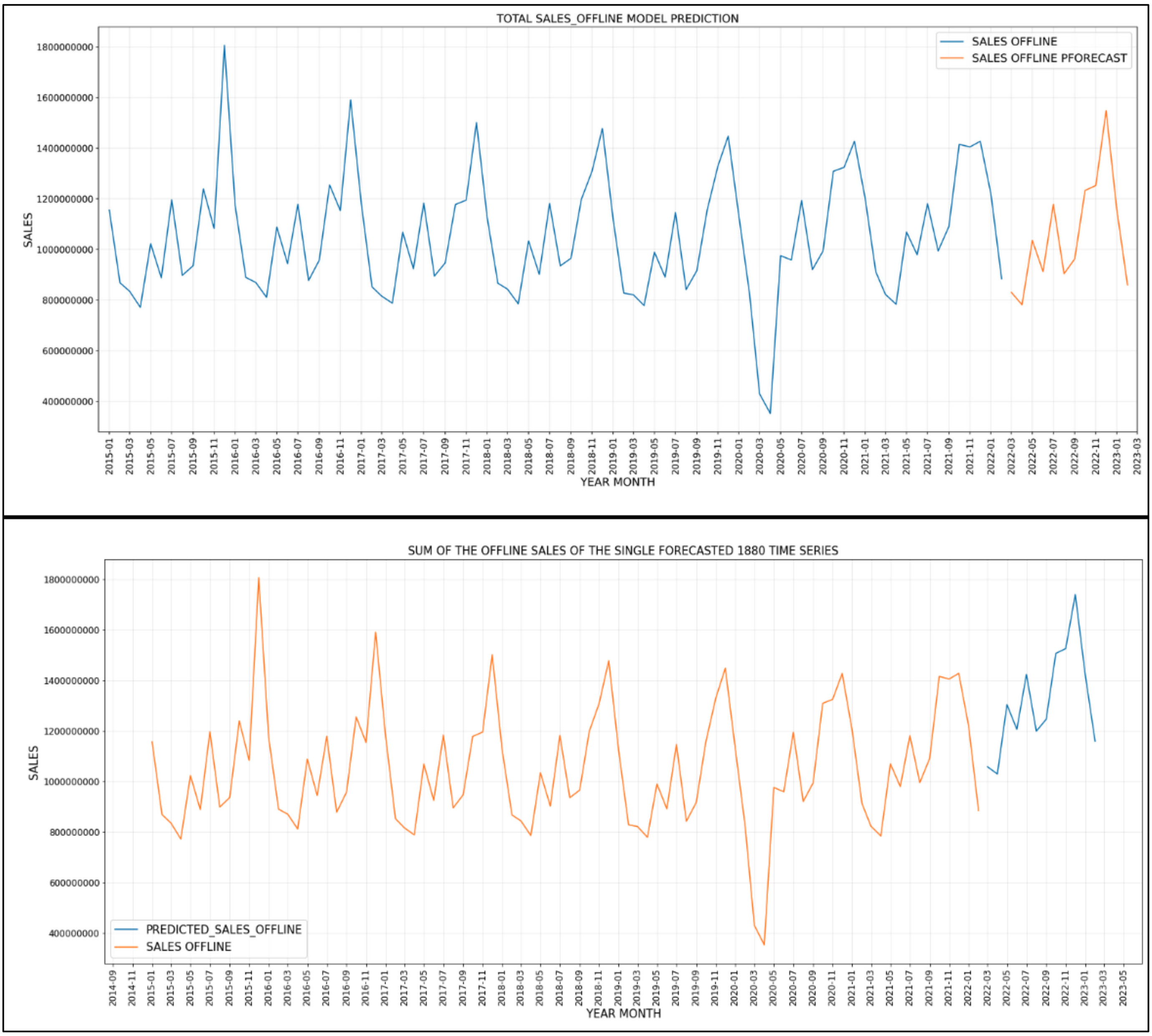


Figure 24

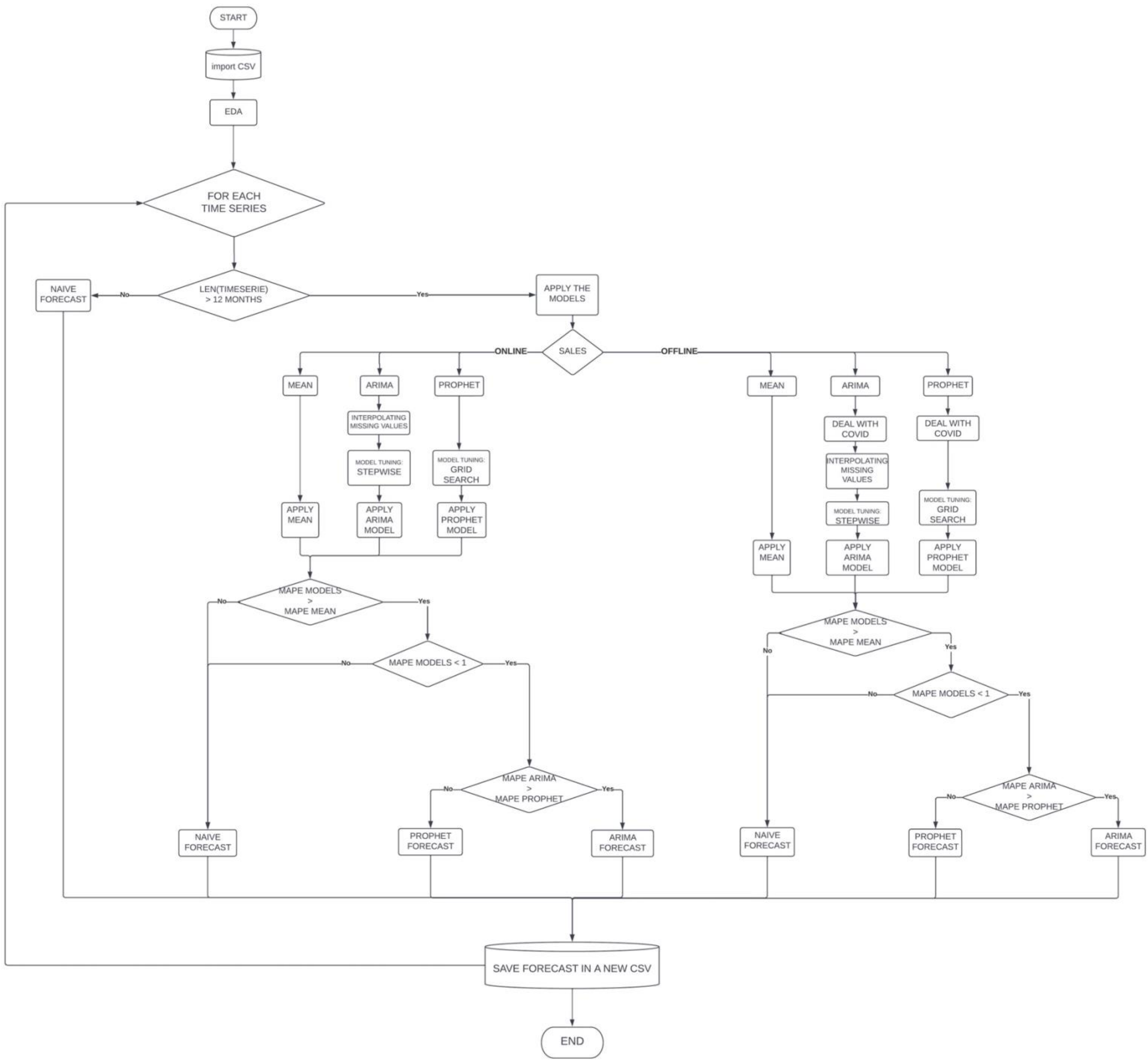
Natural next steps from the direction of future work.

Overall, considering the complexity of the task and the fact that we had never worked with time series before, we are satisfied with the results obtained from the models and how we have structured the work. However, improvements could certainly be made. If we had more time, we would have liked to integrate some variables such as monthly expected inflation and GDP into the predictions. In fact, ARIMA models could have been made more economically accurate by using SARIMAX models instead, which allow for the addition of exogenous variables such as those mentioned above. This would have likely made the prediction results less accurate in terms of MAPE but more accurate in empirical terms, as macroeconomic variables such as GDP and inflation actively influence consumer choices and thus sales growth/decline.

Regarding the purely technical aspect, we would have liked to streamline the code and make it faster in running. In particular, the grid searches for Prophet required a significant computational cost that could potentially be reduced by improving the model.

Additionally, we would have liked to work more on LSTM and found the idea of creating a single model for predicting all the time series (including product groups and brands as features) fascinating. However, in all honesty, our skills in recurrent neural networks and creating a single model for such a high number of time series to manage in parallel are not yet sufficient. This does not necessarily imply that these models would have performed better than our Prophet and ARIMA models, but we would have been intrigued by the idea of tackling this additional challenge, an approach that we will attempt in the future with time series.

Appendix A: Code Description



Appendix B: Author Contribution

To be completely honest, it is particularly difficult for us to attribute specific tasks to each other as we worked together on the project in its entirety. Everything was done jointly and with total collaboration from both of us. All attempts and approaches used were thought out and developed simultaneously by both of us.