Product Based Monthly Prediction and Tracking of Plastic Packaging Weight

Technical Documentation

By Ersan Kucukoglu

Submitted to
Central European University
Department of Economics and Business

Master of Science in
Business Analytics

June 2022

Table of Contents

Background (Summary)

The overall aim of the project is to support Lidl's making business decisions by monitoring and predicting the amount of plastic in plastic packaged products, with a data analytics approach. In this project, the client was the retail chain company Lidl. Lidl is desperate to lower the negative effect on the environment. To do that the business goal is to lower the sold plastic packaging with our product by 20% by 2025. By developing a data product that calculates the KPI of sold plastic packaging by the HU-contracted suppliers in tones and compares it to the goals an instant intervention would be possible in case of a risk.

Problem Definition

Plastic packaging has a lot of environmental impacts, the company has been committed to tackling the important issue of plastic waste, and the detrimental impact that this is having on the environment. The evaluation of the business goals is done quarterly and there is only a high effort possible to follow the measures within the year, no forecast is available.

Key problems

- evaluating the business goals is done quarterly
- there is only a high effort possible to follow the measures within the year, no forecast is available

The project aims to build a data product, which allows the user to see the sold plastic packaging in tones and compare it with the business goals also by seeing the predicted outcome till the end of the business year.

What questions do I need to answer?

To meet the project's requirements, I had to answer following questions.

- What is the sold plastic packaging quantity in tones?
- What is the estimation of sold plastic packaging till the end of the business year?
- Which items are significantly high and should be optimized (packaging change, like packaging change) based on the past / predicted values? This process should be automated and implemented with a dashboard for visualization.

Project Plan

Interim deliverables

- Python notebooks which can be rerun
- The dashboard uses the generated data
- Possibility to upload main item data
- Possibility to upload targeted values

Final deliverables

A dashboard with the following functions:

- allows the user to see the plastic packaging weight in tones
- allows the user to compare it to the business goals
- uses two calculation methods: sales and goods in based
- sales figures can be loaded from the database, and goods should be imported manually (csv)
- give estimation till the end of the business year for the run-out
- marks top items, which past / predicted values are significantly high and should be optimized (recipe change)

Notebooks:

- possible to re-run every month
- possible upload (csv) for: main item data, targeted values, goods-in values

Methods of approach

Data-science methods to be used:

- Model-based prediction, data-cleaning, dashboarding. (Sales and goods-in based.)
    - Receipt data analysis
    - Predicting yearly outcome
    - Importing main item data
    - Dashboarding, creating visualization

Software:

- Databricks platform
- Databricks SQL-Dashboard

Reproducibility:

Since I am developing the project in Databricks platform, I plan to utilize Python and notebooks for data exploration and modeling. It makes it possible to re-run every month to

see the predicted outcomes till the end of the business year. These notebooks will contain all of the code that I use, as well as descriptions, assumptions, and explanations in markdown cells as needed.

Timeline

- April 6: Student submits Project Initiation Document, any request for faculty supervision to the program coordinator. The student works with the project sponsor to develop the draft PID.
- April 20: Project begins. The project kickoff documents (PID, letter of terms, NDA) are finalized and signed. The Program Head appoints the faculty supervisor (if any) based on the suggestion of the Capstone Project Manager.
- May 15: Student submits interim progress report to the program coordinator. The report should discuss the project's status, interim outcomes, work to be done, and any problems or issues.
- June 13: Student submits final deliverables to project sponsor and program coordinator.

Stakeholders

The following individuals from Lidl oversaw my progress and guided me throughout the project:

1. Project Sponsor
2. Data Scientist

Interaction with stakeholders was carried out in individual meetings when necessary. The meetings consisted of discussing project updates, next steps, problems, and possible solutions.

Data

Three different data sets were used to reach the outputs specified in the project. These bonpos (receipt data), item, and status of the plastic packaging optimization data. Receipt and item data sets from these data sets can be accessed from the company's Databricks database system. A database (schema) in Azure Databricks is a collection of tables. A table in Azure Databricks is a collection of structured data. On Azure Databricks tables, you may cache, filter, and conduct any Apache Spark DataFrames-supported actions. Tables may be queried using Spark APIs and Spark SQL. The third dataset status of the packaging optimization table which shows the plastic content of the plastic packaged materials is given by the stakeholder as an excel file.

Bonpos Data

The bonpos table contains the data from all the transactions on the receipt level since 2014. Negative transactions (product returns) are also presented in this dataset, but they will be removed in the data cleaning process as they do not mean anything for analysis. The dataset has all info about the receipts, only the variables used in the analysis will be given here. The variables are RECEIPT_ID (key), ITEM_NUMBER, ITEM_NAME, RECEIPT_DT (date), SALES_FG (if 1, the item was sold), SALES_PIECE_QTY (number of items sold), SALES_WEIGHT_QTY, TURNOVER_RELEVANT_FG (if the purchase is relevant to the turnover), WEIGHT_ITEM_FG (flag for weighted items). For bonpos data, the spark was used until the data is getting aggregated, because it would be too big for pandas.

Item Metadata

Item metadata contains names and categories of all products. In this dataset, item number, valid from, and item group columns were used. The underlying item for a specific item number can change over time. In addition, products can be renamed as well. These data cleaning-related problems will be in the data cleaning part.

Actual Status

The actual status of the plastic packaging optimization table mainly includes the weights of the plastic packaging for the products that can be optimized. There are 41 different products in total in this table. In this excel table, item number, item description, brand, and plastic packaging weight previously and new, and reduction of plastic packaging material (%) variables were used for the calculation of the plastic packaging weight in this analysis. The plastic packaging weight of the product can be found by multiplying the new plastic packaging weight variable, which is the most important variable in this table, with the monthly total sales quantity of the item.

Data Cleaning and Exploration

Data Cleaning

After glimpsing the tables under the core_data database in Databricks Platform, the necessary libraries were imported into the notebook and the datasets were loaded using spark. First, bonpos data was examined. To find the weight of plastic packaging sold, it is necessary to focus on the number of items sold first, until the calculation. For the products sold, first, the sales flag, deposit flag, and turnover relevant variables are filtered as 1. In this way, deposited, i.e., not sold, returned products were deleted from the data. In the next step, the variable SALES_QTY to be used in the calculation is obtained from the product of SALES_WEIGHT_QTY and SALES_PIECE_QTY variables.

In the item data set, considering the possibility that the item number and name may have changed over time, according to the information given by stakeholders, duplicate values were checked. The VALID_FROM variable displays the date from which the properties given to the item are valid. Since in this case, it makes sense to base the most recent data, the duplicated values were removed and were kept the ones with the current date.

After the optimization table was imported, the explanation lines before the column names were removed, and the characters were changed to uppercase to be compatible with other datasets. From this data set, item number, the column that is common with item data, and columns related to plastic packaging weights are selected and merged with item data. This table has been created as df_plastic _content table under the db_ersan database to be merged with the data after item-level sales forecasts are made in the forecasting part. By merging this with bonpos data, the receipt data and plastic content data of the items were collected in a data frame.

The next step is to obtain a monthly based plastic package weight data for each item. The previous and new plastic packaging weights (in tons) were obtained by multiplying the previous and new plastic packaging weights (in grams) with the Sales Quantity variable. Also, to see how many tones of packaging were saved per item, the Reduction variable was obtained with the difference between the previous and new plastic packaging variables. To make the data monthly, the RECEIPT_DT variables were split and obtained year, month, and day variables.

Finally, to obtain the monthly aggregated data, the monthly sales quantity, previously plastic packaging weight (in tons), and new plastic packaging weight (in tons) variables were calculated by grouping the data set according to item number, year, and month. This monthly aggregated data table has been saved to the database as df_monthly to be used in modeling and dashboard. It includes 2401 observations and 10 variables.

	ITEM_NUMBER	ITEM_NAME	WG_NAME	YEAR	MONTH	TOTAL_SALES_QTY	TOTAL_PRE_PLASTIC_T	TOTAL_NEW_PLASTIC_T	DATE	REDUCTION_T
0	1709	Bruehwurst Stapelpack QS SK24	Kühlung	2022	6	11255.0000000000	0.090715	0.079573	2022-06-01	0.011142
1	1709	Bruehwurst Stapelpack QS SK24	Kühlung	2022	5	46258.0000000000	0.372839	0.327044	2022-05-01	0.045795
2	1709	Bruehwurst Stapelpack QS SK24	Kühlung	2022	4	73219.0000000000	0.590145	0.517658	2022-04-01	0.072487
3	1709	Bruehwurst Stapelpack QS SK24	Kühlung	2022	3	56854.0000000000	0.458243	0.401958	2022-03-01	0.056285
4	1709	Bruehwurst Stapelpack QS SK24	Kühlung	2022	2	50579.0000000000	0.407667	0.357594	2022-02-01	0.050073

Figure 1 Monthly Aggregated Sample Data

Data Exploration

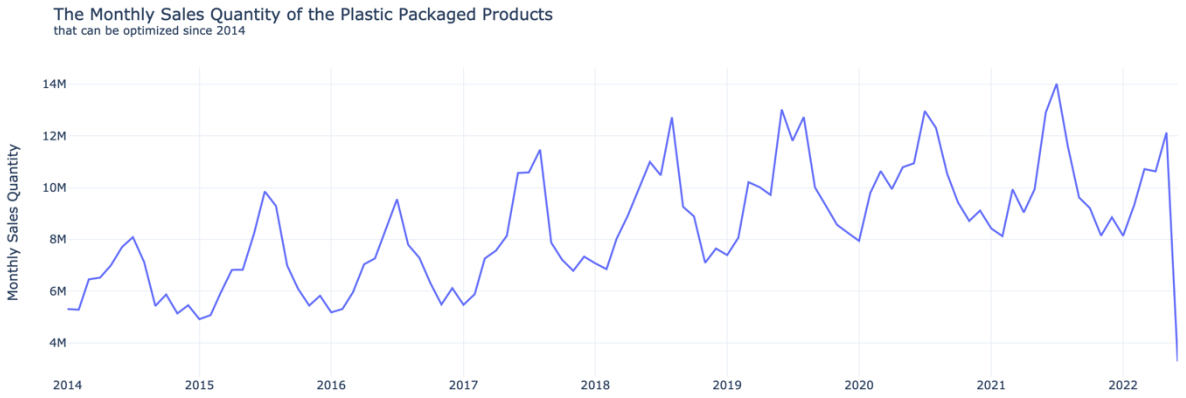
Looking at the summary statistics table of the plastic packaging amount of each item, the count value of some products is 102 because there are 102 months from 2014 to today. Those whose count value is not 102 are newly added products. For example, product number 8126 has data for 32 months. When the mean value in the statistical table is examined, the highest average monthly plastic packaging amount (34.84 tons) of the product is Mineralwasser mit CO2 1.5l, numbered 2338.

Figure 2 Summary Statistics Table

ITEM_NUMBER	count	mean	std	min	25%	50%	75%	max
2338	102.0	34.847059	9.307626	10.83	27.7925	32.630	39.4775	62.99
2336	102.0	32.078333	10.105795	12.02	23.1625	31.460	39.2625	56.84
3290	102.0	30.442549	6.968845	9.98	25.7250	29.390	35.0700	45.84
3025	102.0	29.052353	8.537643	8.09	22.8475	26.950	33.5850	55.32
3293	102.0	15.026667	4.234808	0.05	11.7950	14.995	17.4350	26.14
8126	32.0	12.344375	6.019103	1.98	7.1100	13.170	18.2150	20.65
8107	28.0	8.380000	2.763000	0.90	7.2750	8.395	9.6400	15.88
4579	102.0	7.755882	3.084927	2.58	5.2600	7.470	10.0125	14.60
3564	102.0	7.698922	3.284728	2.67	5.5300	6.990	9.7075	16.34

To find the future monthly plastic packaging amount of the products, first, the monthly sales quantity will be found. To get an idea about how the overall quantity of sales changes over time, a line chart of the monthly aggregated data set with the products that can be optimized has been created.

Figure 3 Monthly Sales Quantity over the time



As can be seen from the line graph, the data generally has an upward trend and seasonality. In general, the sales quantity of the products increases every year in the summer months. The behavior of time series is examined in more detail in the time series analysis section.

In the bar chart below, there are 10 products with the highest amount of plastic packaging in 2021 and 2022. According to the chart, the product with the highest plastic packaging last year is Mineralwasser ohne CO2 1.5l with 494.61 tons, while it is again the same product this year. Considering the bar values and considering that we are currently in the middle of the year, the 2022 data is the same as the previous year.

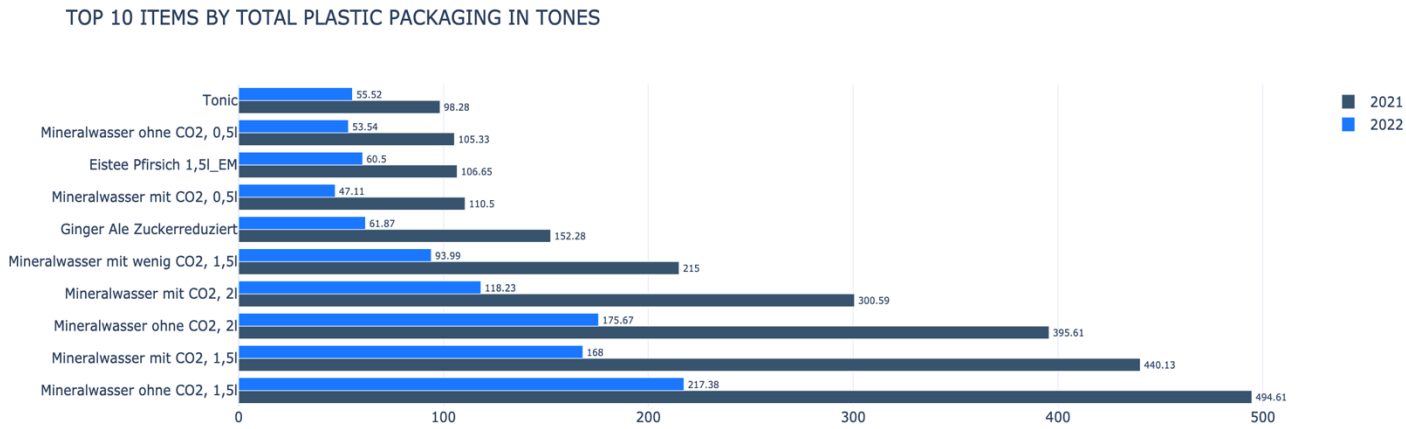
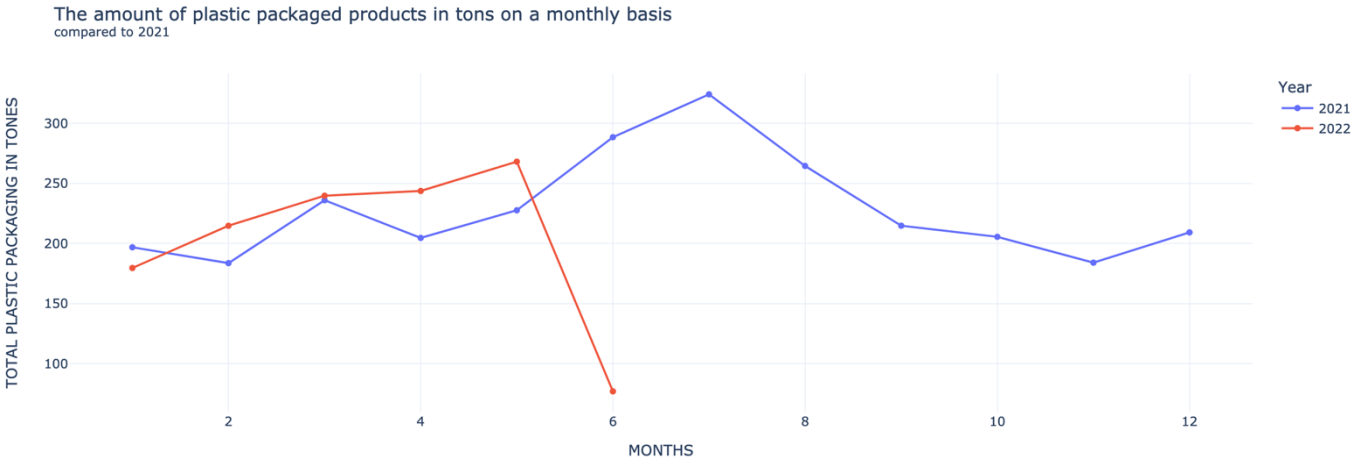


Figure 4 Top 10 Items by Plastic Packaging Weight , 2021 and 2022

Considering the plastic packaging weight in 2021 and 2022 data on a monthly basis, the amount of plastic packaging sold in 2022 almost follows the line chart of 2021. As can be seen from this graph, since the quantity of sales increases in the summer months, the weight of plastic packaging that is sold increases in direct proportion. Note that since June of 2022 is not over yet, it is normal for the

red line to be opposite the blue one. At the end of June, it is predicted that it will be close to the value in 2021.



Time Series Analysis

Time series are numerical quantities in which the values of the variables are observed consecutively from one period to the next. Time series data should be acquired at regular intervals to see the progress of the series. The most distinctive feature of time series data being different from other series data is that the observed values in the series are interdependent over time. Time Series Analysis summarizes the characteristics of a series and tries to reveal the outstanding structure of the series. Time series use line charts to show seasonal patterns, trends, and relationships with external factors. The long-term increase or decrease of a variable observed over time is called a trend. Depending on the unit in which they are measured, time series can have a similarly repeating course. This course is called the seasonal element. The time dimension of a wave is defined as the recurrence period. The recurrence period of a seasonal element is at most one year.

Time Series Decomposition with ETS (Error-Trend-Seasonality)

ETS Decomposition was used to separate the different components of the time series. Statmodels library provides a seasonal decomposition tool we can use to separate the different components. Visualizing the data based on its ETS is a good way to build an understanding of its behavior. After importing the seasonal decompose, the decompose result object is set. We apply an additive model when it seems that the trend is more linear and the seasonality and trend components seem to be constant over time.

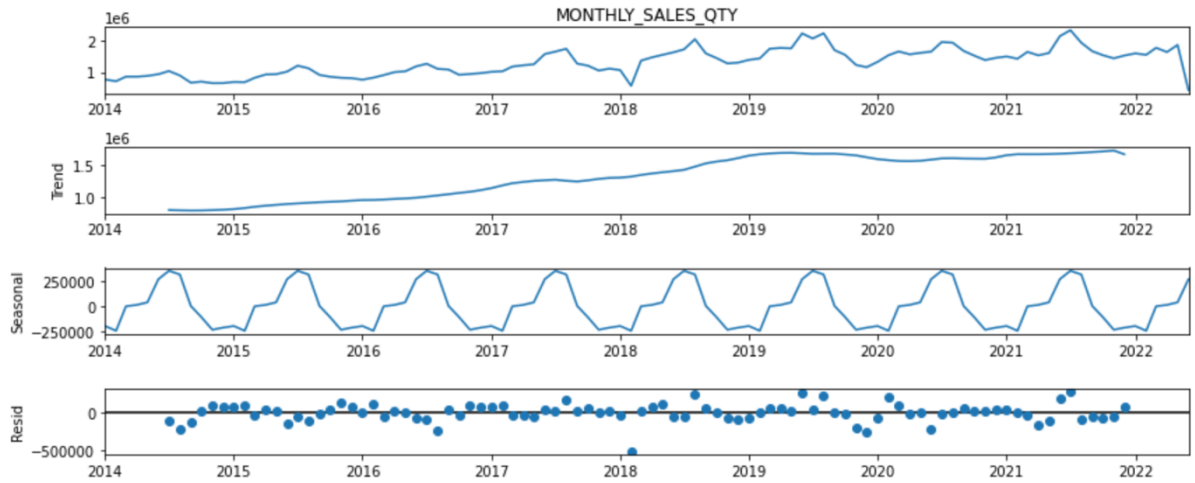


Figure 5 Seasonal decompose results

We can see that there is a general trend upwards. That's the trend component. There can be a linear trend or exponential trend and that can be either downwards or upwards. It looks like it could be slightly exponential, and the reason it's hard to tell from raw data here is that we can see there is also seasonality to the dataset. We can see that based on the month of the year, there tend to be fluctuations that are repeating. There are peak sales during the summer months. We see the original observed data at the top. The trending term is showing the general growth pattern or decline pattern, depending on the data of the actual real observed data points. We can see that it does look to have a general trend upwards and we focus more on the trend components to see if it's exponential or linear. Then we can see the isolated seasonal component. That is perfectly repeating and that allows us to analyze seasonality. We have the last residual component and this residual component is for those not explained by trend or seasonality. Any sort of residual or error that is not explained by trend term or seasonal term is going to be a residual.

EWMA (Exponentially Weighted Moving Average)

The smoothing process is essential to reduce the noise present in time series and to indicate actual patterns that may appear over time. There are three important smoothing methods in time series analysis: Single Exponential smoothing, Double Exponential Smoothing, and Triple Exponential Smoothing. Triple Exponential Smoothing is an extension of Exponential Smoothing that explicitly adds support for seasonality to univariate time series. EWMA will allow us to reduce the lag effect from SMA and it will put more weight on values that occurred more recently. Basic SMA has some weaknesses, smaller windows will lead to more noise, rather than signal. It is hard to balance the window size because as we go smaller, we are being

more accurate as far as what we can model from that generalized time series, behaviors general trends in the current data. Extreme historical values can skew the SMA significantly.

Holts -Winters Method

Holt-Winters methods were implemented using statmodels.
Peviously with EWMA we applied simple exponential smoothing using just one smoothing factor-alpha. This failed to account for other contributing factors like trend and seasonality. The Holt-Winters seasonal method comprises the forecast equation and three smoothing equations. (1) level l_t, (2) trend b_t and (3) seasonal s_t. The idea is that it's separating the time series data into two main components, the actual level or value component and then the trend component. If we use Holts Method, just double exponential smoothing, it would be able to predict that sales quantity is increasing.

There is Holts Winter method, which is the winters add on to this, which is known as triple exponential smoothing. With that, we are going to introduce a smoothing factor called gamma that addresses there changes due to seasonality. We added one more component, s_t.

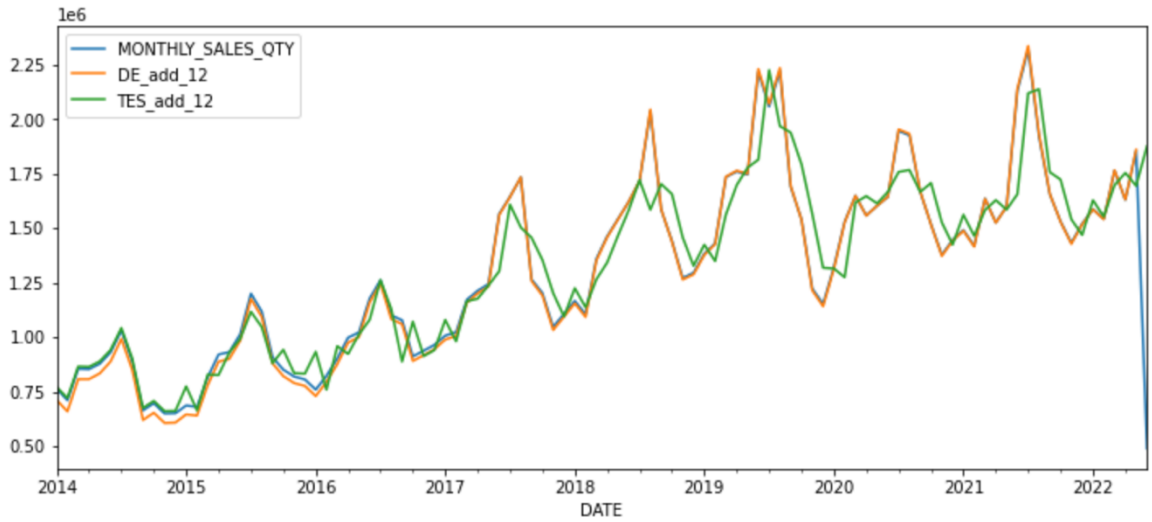


Figure 6 Comparison of the Smoothing

Forecasting Models

In this section, prediction models are created and evaluated. The aim here is to select the best model for the monthly plastic packaging weight prediction, which is one of the aims of the project, and to obtain future forecasts for the next 10 months (March 2023).

So far, the behaviors and the patterns of the time series were explored with tools such as pandas, numpy and statmodels in the Time Series Analysis section. In this section, different forecasting models were built for the time series. General forecasting procedure is:

1. Choose a Model
2. Split data into training and test sets
3. Fit the model on the training set
4. Evaluate the model on the test set
5. Re-fit the model on entire dataset
6. Forecast for the future data

With the Persistence Algorithm 6 different models created, Holts-Winters, SARIMA, SARIMAX, PROPHET and LSTM for the time series data by following the procedure that given above. For the evaluating the predictions, Root Mean Square Error (RMSE) metric is used. RMSE is more sensitive to outliers and RMSE has the benefit of penalizing large errors more so can be more appropriate in some cases. An issue with MAE though, is that simply averaging the residuals will not alert if the forecast was off for a few points.

Dickey- Fuller Test

In order to select the best model, the data augmented Dickey-Fuller test implemented to find if the data is stationary or non-stationary. This performs a test in the form of a classic null hypothesis test and returns a p-value. If p-value is low (<0.05) , reject the H0, so the dataset is stationary. The p-value is 0.52, weak evidence against the null hypothesis and fail to reject the H0.

Persistence Algorithm

A baseline in forecast performance allows for comparison. The Zero Rule algorithm is the most often used baseline approach for supervised machine learning. In the case of classification, this method predicts the majority class, whereas in the case of regression, it predicts the average outcome. This might be used for time series; however, it disregards the serial correlation pattern found in time series datasets. The persistence algorithm is the equivalent approach for use with time series datasets. The persistence method predicts the expected outcome at the next time step (t+1) based on the value at the previous time step (t-1). After evaluating the model with test data, RMSE value found as 379432.49.

Holt-Winters with Exponential Smoothing

Firstly, the data was split into train and test set. Train set includes the data from 2014 until 2021, while test set includes the rest of the main dataset. The model was fitted to the training

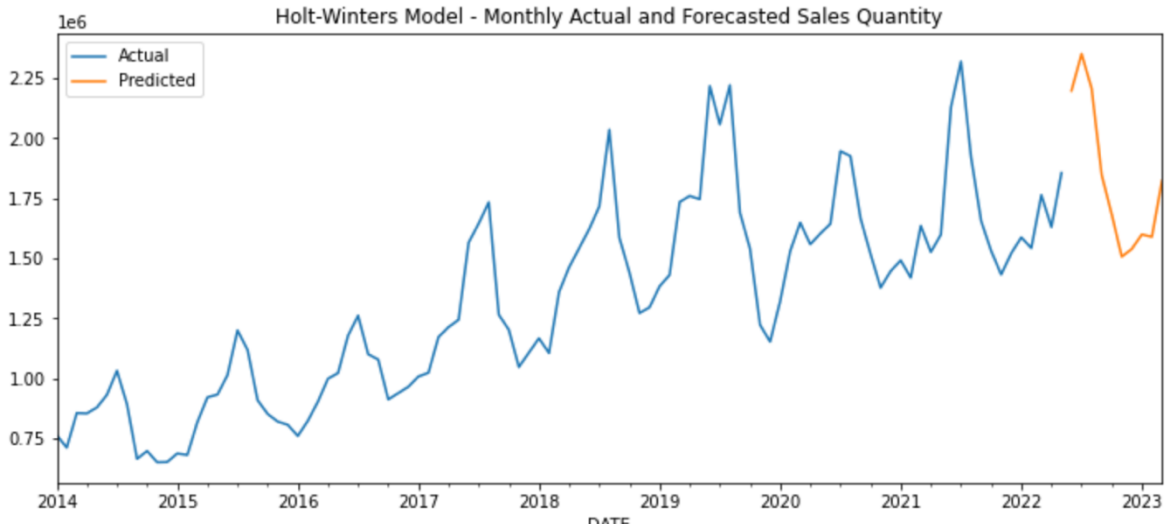


Figure 7 Holt-Winters Model - Forecast until March, 2023

data and the trend and seasonal components were chosen as additive and multiplicative, respectively, while creating the model. For this, both options have been tried and the best performing has been selected. For the evaluation stage, the model is evaluated with the test data set. Train, test, and prediction values were shown on the same line graph and compared, and the RMSE value of the model was found as 175892.03.

The last step is to fit the created model to the entire data set and predict for the next 10 months. Forecasted values follow the trend and seasonality as can be seen from the line plot below.

ARIMA Models

Autocorrelation is used for the correlation between the values of the same variable itself. ARIMA models, like Exponential Correction, can be divided into seasonal and non-seasonal models. Since there is seasonality in the data, it is thought that a seasonal ARIMA model would be more appropriate.

ARIMA models consist of three main elements. The AR() element, the MA() element, and the "d" element, that is, the differentiation element. The small "p" here indicates the non-seasonal AR() rank (how many lag observations will be used) while the large "P" indicates the seasonal AR() rank (how many seasonal lag observations will be used). Likewise, the small "q" indicates

the non-seasonal MA() order (how many lags forecast errors will be used), while the large "Q" indicates the seasonal MA() order (how many seasonal lag forecast errors will be used). The small "d" value indicates how many normal offsets will be applied until the data is stationary, and the large "D" value shows how many seasonal offsets will be applied until the seasonal effect is removed from the data. Finally, "m" is the observation frequency on a yearly basis. For example, it will be 12 for monthly data.

To find these auto arima function is used instead of reading ACF and PACF plots. Identification of an AR model is often done with PACF, MA model with ACF. The pmdarima is separate library designed to perform grid searches across multiple combinations of p,d,q and P,D,Q parameters. It utilizes AIC as a metric to compare the performance of various ARIMA based models.

- **Use auto.arima() to find the best ARIMA Model**

SARIMAX Results

Dep. Variable:	y			**No. Observations:**	101	
Model:	SARIMAX(1, 1, 1)x(0, 1, 1, 12)			**Log Likelihood**	-1163.527	
Date:	Fri, 10 Jun 2022			**AIC**	2335.055	
Time:	21:40:21			**BIC**	2344.964	
Sample:	0			**HQIC**	2339.047	
	- 101					
Covariance Type:	opg					

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	0.4855	0.271	1.794	0.073	-0.045	1.016
ma.L1	-0.7844	0.201	-3.896	0.000	-1.179	-0.390
ma.S.L12	-0.4867	0.117	-4.160	0.000	-0.716	-0.257
sigma2	2.188e+10	7.13e-12	3.07e+21	0.000	2.19e+10	2.19e+10

Ljung-Box (L1) (Q):	1.02	**Jarque-Bera (JB):**	3.79
Prob(Q):	0.31	**Prob(JB):**	0.15
Heteroskedasticity (H):	2.49	**Skew:**	0.14
Prob(H) (two-sided):	0.02	**Kurtosis:**	3.98

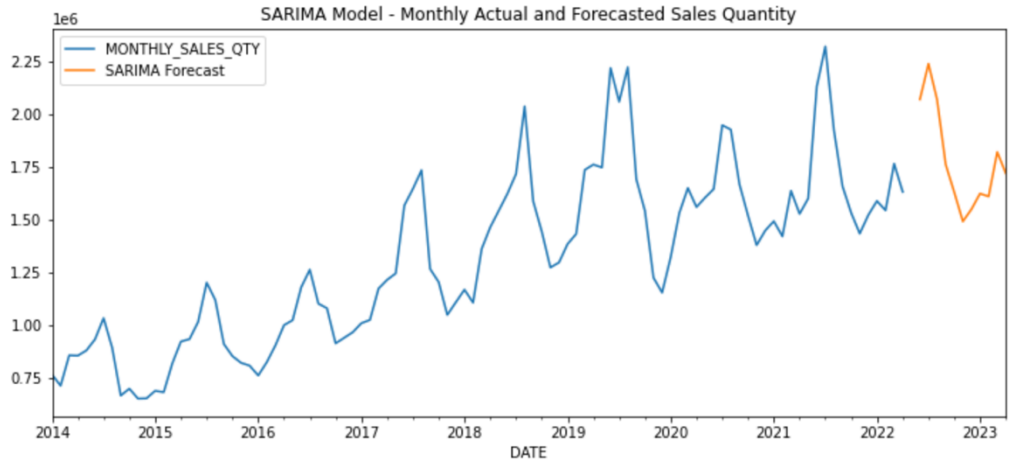
Figure 8 Auto arima test results

The results give two different orders, ARIMA order and seasonal order. In this case for the given best model, ARIMA (1,1,1) and seasonal order (0,1,1)[12]. Basically, this informs that these parameters should be passed to the statmodels.

SARIMA

After the auto ARIMA process, the parameters were found. A SARIMA model was created with the obtained p, d, q and P, D, Q parameters and fitted to the training dataset. Model evaluation with the test set is 222427.5 as RMSE. After, the forecasting was done to predict for the next 10 months.

Figure 9 SARIMA Model - Forecast until March 2023



SARIMAX

The statmodels implementation of SARIMA is called SARIMAX. The X added to the name means that the function also supports exogenous regressor variables. Apart from these seasonal (P, D, Q, m) factors, this model introduces the idea that external factors (environmental, economic, etc.) can also influence a time series and be used in forecasting. Month dummy variables and summer months were added in exogenous variables. After the Auto Arima test the suggested best model is SARIMAX (1,0,0) x (0,1,1,12). Based on these orders, the model was created and fitted. The RMSE value of the model is 147512.6. According to SARIMA, the RMSE value is quite low, and it is the lowest RMSE value obtained so far.

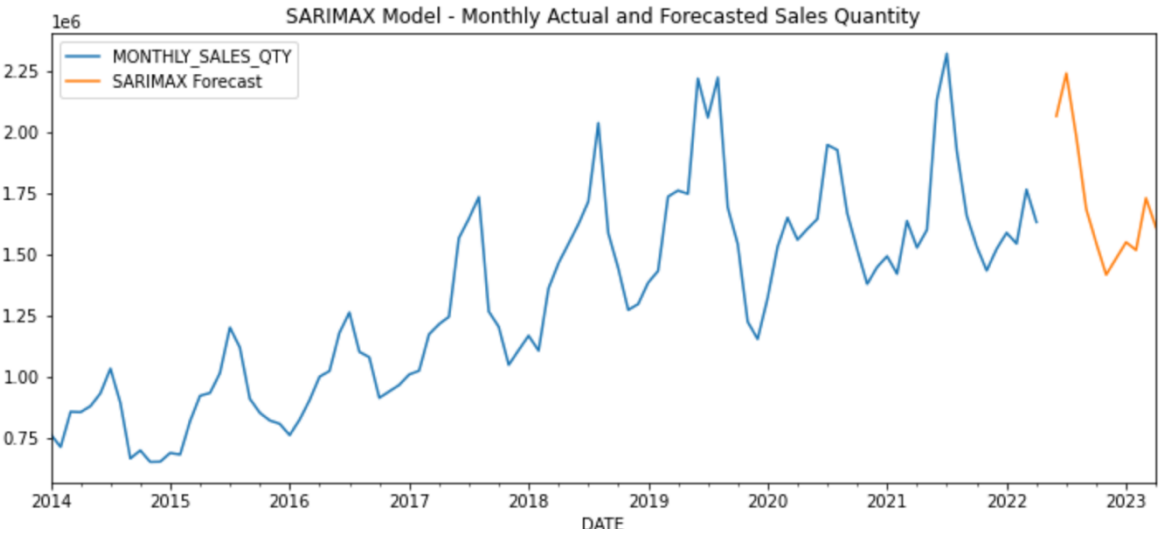


Figure 10 SARIMAX Model - Forecast until March 2023

PROPHET

The Prophet model is an open-source forecasting application developed by the Facebook data science team. It includes procedures that enable making annual, quarterly, weekly, and daily forecasts on non-linear time series data. In this study, Python language modules of the Prophet model were used. Prophet takes a dataset with two attributes as input. The first of these features is the ds timestamp, and it supports time formats that can be handled by pandas, a Python data manipulation library. The other feature, y, is the numerical measurement value that is the subject of the estimation. The model trained with the data set containing ds and y values can produce predictions in different periods. The intuitive approach to the operation of the Prophet model allows to create effective predictions without drowning in the details of the data.

Prepare the data

As input for the Prophet model, it needs to enter 'ds': date and 'y': numeric value that we want to predict. Changed DATE column to ds and TOTAL_SALES_QTY to y. Then, fit() and predict() functions are used as in sklearn models. Since the data is monthly freq is passed as MS into the model.

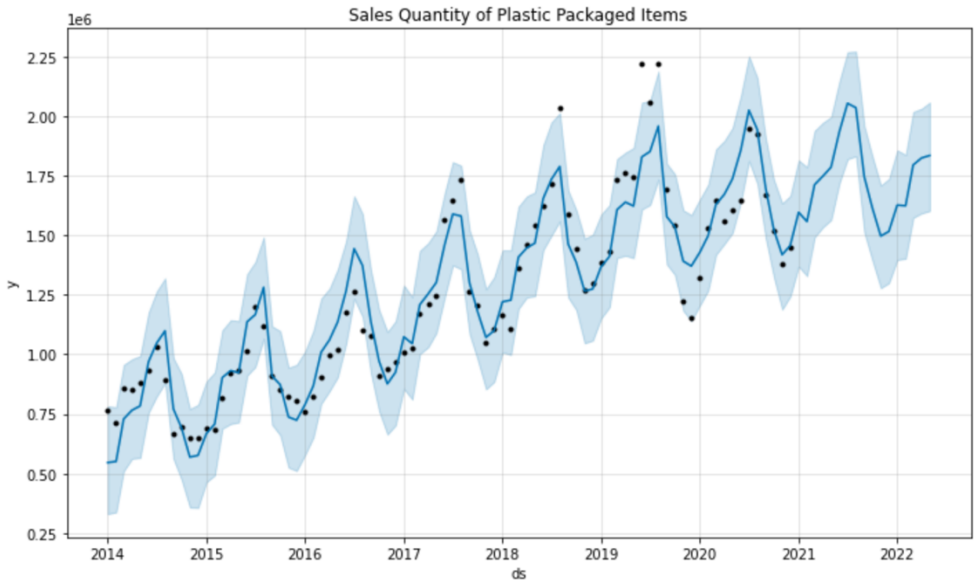


Figure 11 Prophet Model - True and Test Predictions

It can be seen the true data as far as on the left-hand side with the black points of the true data points and the way the model is fitting to it. Forecasting points can be seen as well as an upper and lower bound. It can be found that the data has clear upward trend between 2014 and 2019. After 2019, the momentum of the trend decreased, which may be due to the emergence of the

covid virus in that year. In addition, when the yearly plots below are examined, the peak sales in the summer months attract attention.

After preparing the data and fitting it to the train set, evaluation is done and the RMSE value

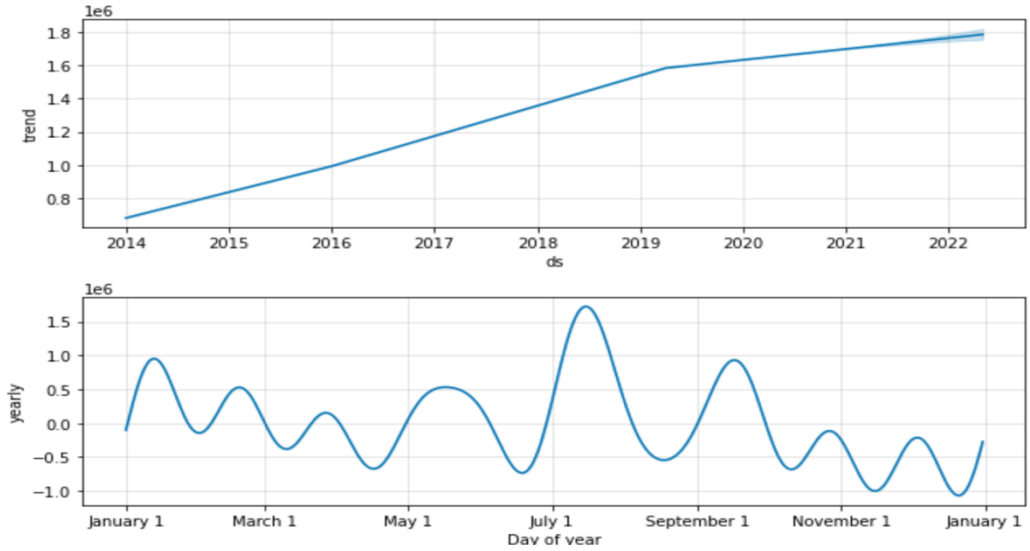


Figure 12 Components of the Prophet Model

found as 134645.6. This is the lowest RMSE value among others. In the chart below with the actual and predicted sales amounts per month, forecasted data shows the seasonality behavior but it does not match the general trend.

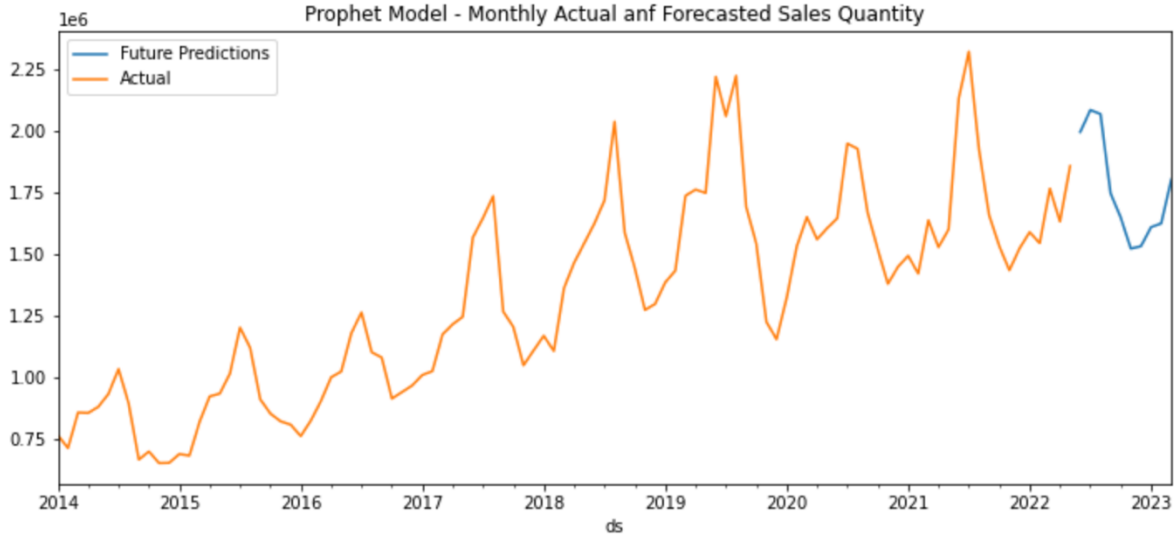


Figure 13 Prophet Model - Forecast until March 2023

LSTM

RNN ensures that each output it produces progresses depending on the previous step. It tries to keep the results calculated in the previous steps in its memory. LSTM networks are an extension of recurrent neural networks (RNNs) mainly introduced to handle situations where

RNNs fail. Talking about RNN, it is a network that works on the present input by taking into consideration the previous output (feedback) and storing in its memory for a short period of time.

Scale and Transform Data

The data was prepared for LSTM modelling and, and it is split into train and test set. For the next step, the data need to be scaled and transformed to get ready for modelling. This can be done with sklearn. Firstly, the scaler object was created with MinMaxScaler. The scaler object can either fit or transform, so the scaler object was fitted to the training data, this process finds the maximum value in the training data. After scaling, the next step is transforming the training and test data. For that scaled train and scaled test data were created with scaler object and transforming the train and test sets. This process basically divides by the maximum value.

After transforming the data, we must figure out how we are going to feed batches of the time series along the label. RNN, basically want something like t1 (time step 1), t2, t3, and it wants some sort of label for t4. Keras has preprocessing time series generator object that does this automatically. After importing Time Series Generator object from Keras library, which is going to be able to read in an entire time sequence like scaled train data, then split out batches. For the generator components, number of inputs was defined as 12, that way we go ahead and look at 12 months before predicting 13^{th} month, number of features was defined as 1. Time Series has also another component, which is batch size. Batch is how many times are these batches it produces. For time series analysis, smaller batch sizes lead to better training. If the batch size is too large, the neural network might end up overfitting the training data. In this case batch size was set firstly 1 and large numbers, and as a result batch size 1 performed better than others.

Built a Model

LSTM model is created and fitted to the trained time generator. Then loss plot is created. It can be found that, there is significant reduction over the first couple of epochs.

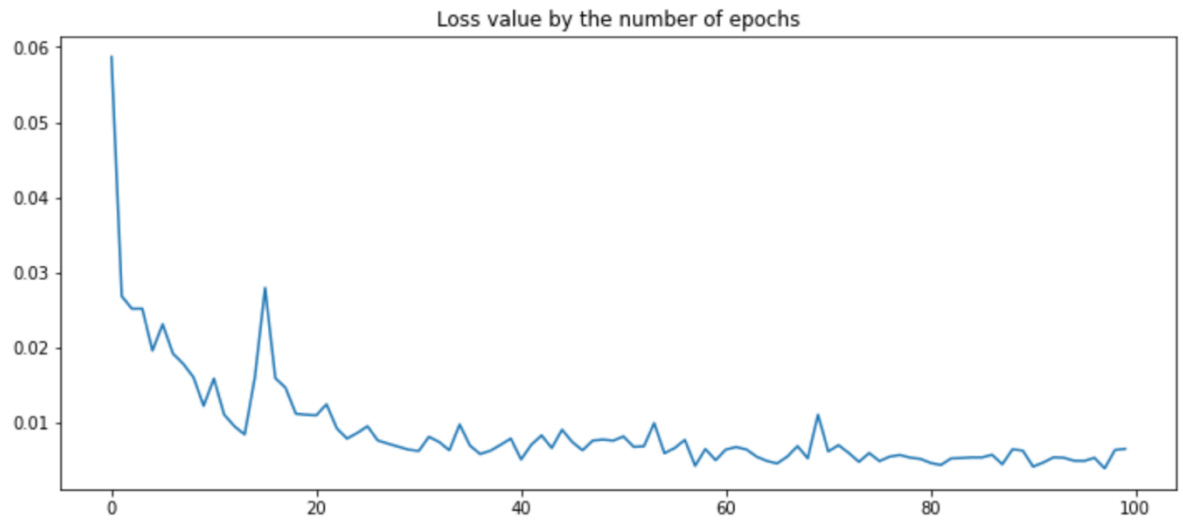


Figure 14 Loss value by the number of epochs

The model is trained, and the next step is the forecasting. After prediction, compare the test prediction and true predictions values in the below plot.

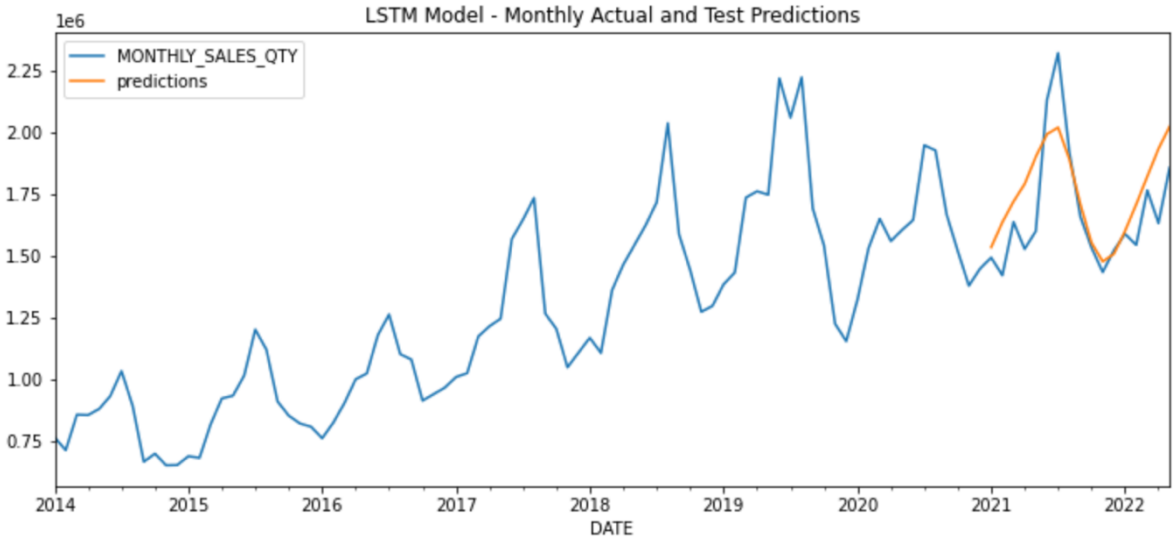


Figure 15 LSTM Model - Comparison True - Test Predictions

As can be seen from the graph above, the prediction values are close to the test data, but in the summer months, when the sales peak, the prediction values are below compared to the actual true values. Finally, after the LSTM model was evaluated with the test data set, the RMSE value was found to be 140982.14.

Model Selection

In this section, the data has been prepared for five different models, all data has been divided into train and test sets. After the models were fitted with train data, the RMSE values of the

model were found with the test set. According to the table below, the model that works best according to the RMSE value is the Prophet Model. By examining the components of the prophet model, we can see that it can identify key trends in the data. The overall trend represents the overall increase in sales. It accurately depicts the annual seasonality, so it's understandable why the performance was reasonably good. Prophet's is more convenient, as LSTMs are trained on time-series windows out of context and do not necessarily account for seasonality in the data. In the next stage, Prophet will be used as the forecasting model in the calculation of the amount of plastic packaging.

Figure 16 RMSE Values of the Models

	Model	RMSE
4	PROPHET	134645.475557
5	LSTM	136013.259667
3	SARIMAX	147512.645315
1	Holt Winter	175892.038341
2	SARIMA	222427.576872
0	Baseline	379432.497591

Monthly Predicted Plastic Packaging Calculation

Up to this point, we attempted to forecast future sales volumes for that product by choosing a single item from the data set and applying several forecasting models. The Prophet model with the highest performance was chosen among the other models we tested, and the first step in this part is to apply the model to all items and create a datagram (prediction_df) including the forecasted values up to the end of the business year, March 2023. First, the function named prophet_predict_item, which has data and item components, was created to apply prophet model to the data. Later, the model was applied to all other products with the for loop.

The table of optimized plastic packaged products was imported and merged with the dataset containing the forecast values. As a result, the values required for calculation in the merged data set are forecasted value and plastic packaging weight. By simply multiplying these two values, the Predicted Plastic Packaging Weight variable was created for each product.

	DATE	YHAT_LOWER	YHAT_UPPER	YHAT	ITEM_NUMBER	ITEM_NAME	ITEM_DESCRIPTION	PREDICTED_PLASTIC_PACK_T
339	2022-07-31	-137260.954504	-135927.880110	0.000000	112144	Wattestaeb. mit Papierschaft in Mot	Fültisztító pálcika	0.000000
340	2022-08-31	-195797.824170	-194099.852079	0.000000	112144	Wattestaeb. mit Papierschaft in Mot	Fültisztító pálcika	0.000000
341	2022-09-30	319963.776735	322021.817585	320977.725096	112144	Wattestaeb. mit Papierschaft in Mot	Fültisztító pálcika	2.407333
342	2022-10-31	-8492.159715	-5975.468918	0.000000	112144	Wattestaeb. mit Papierschaft in Mot	Fültisztító pálcika	0.000000
343	2022-11-30	-116114.112100	-113092.801200	0.000000	112144	Wattestaeb. mit Papierschaft in Mot	Fültisztító pálcika	0.000000

Figure 17 Sample data of the main data that has forecasted values

Outcomes

As a primary deliverable of the project, I constructed a SQL Dashboard that allows the user to see the sold plastic packaging quantity in tones, give estimation till the end of business year for the run-out, and show top items, which past / predicted values are significantly high and should be optimized.

At the beginning of the Dashboard, 3 different metrics appear. As it is written on the counters, these show the (I) total weight of plastic packaging sold for the current year and previous year, (II)the total amount of plastic weight saved during the year, (III) the monthly current plastic packaging weight used according to the selected month and the predicted value for that month. With these 3 data cards, the dashboard gives general information about the weight of plastic packaging on a monthly and yearly basis.

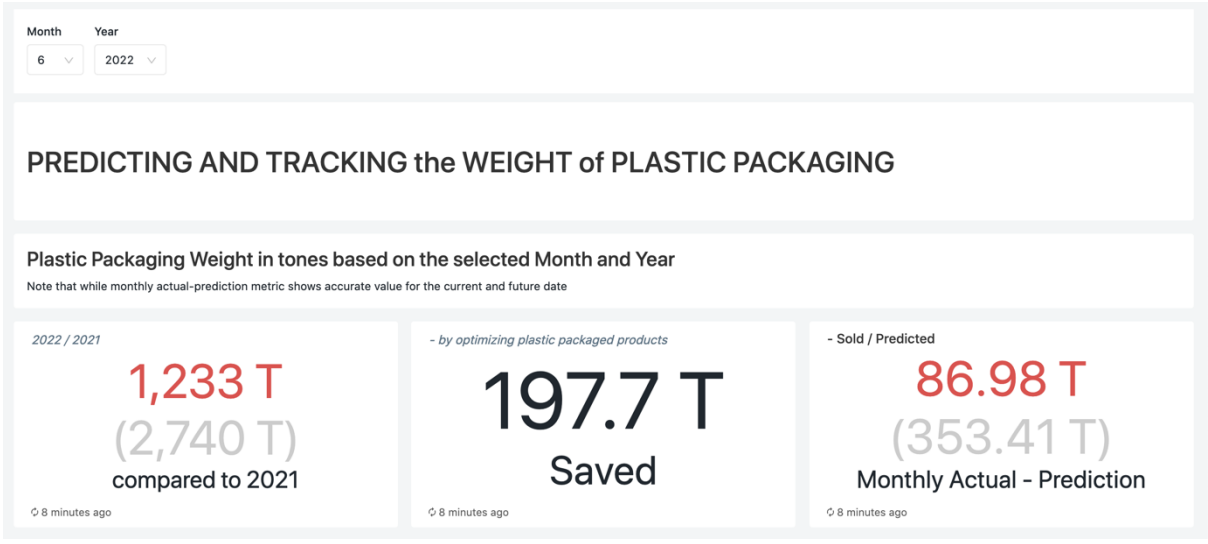


Figure 18- Dashboard (Part1) - Counters

Dashboard Figure 19 also shows item-based data. Here, the 10 products with the highest plastic packaging weight on a monthly basis are shown with their predicted value for that month. The bars are ordered by actual value to see the weight of the Top 10 items for the current month. At the same time, the dashboard shows you the top 10 products from the past months (already sold) and the next months (predicted), with the help of widgets.

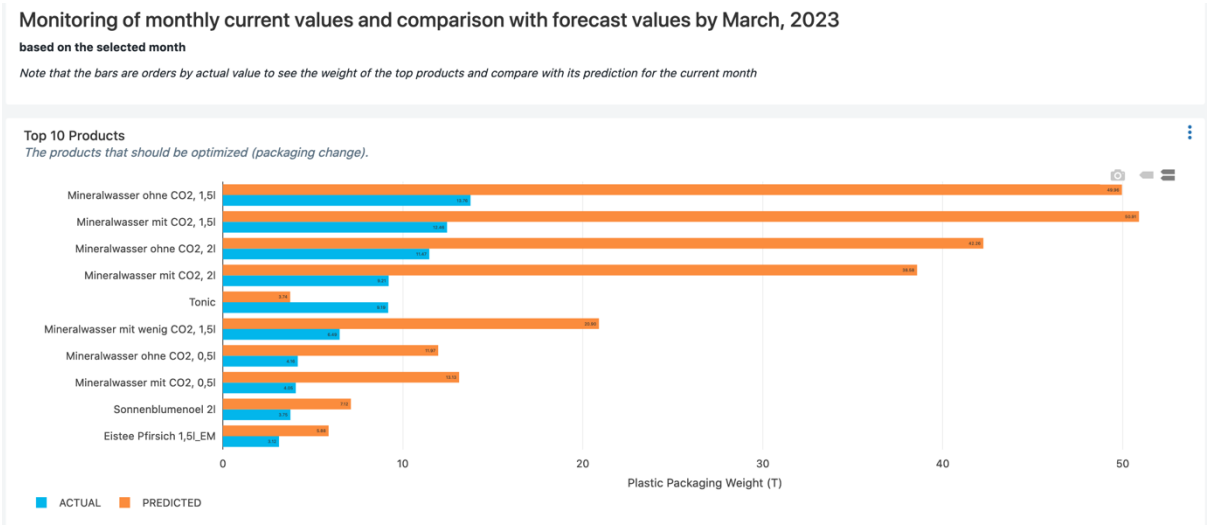


Figure 19 Top 10 Items that should be optimized (packaging change) with their predictions

Figure 20 shows the forecasted, actual values of the total weight of plastic packaging in tons per month, and also provides the opportunity to compare with the data for 2021. For example, there is a prediction of how many tons of plastic has been used so far in June this year, how many tons were used last year, and how many tons will be used. Meanwhile, the predicted value shows well that June is in the trend of the 2021 value.

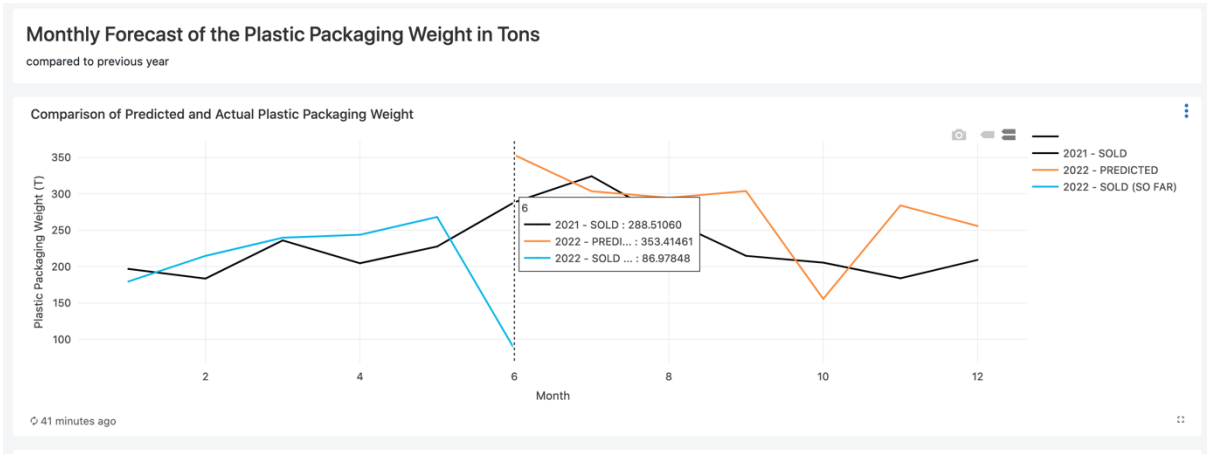


Figure 20 Monthly Forecast of the Plastic Packaging Weight in Tons

Figure 21 is the bar graph showing the monthly total weight of the selected items. After selecting the product or more than one product with the help of the widget, the dashboard presents the data for this year and last year for that product according to the months. How the sales of the product changed according to the months and a comparison with the last year can be made.

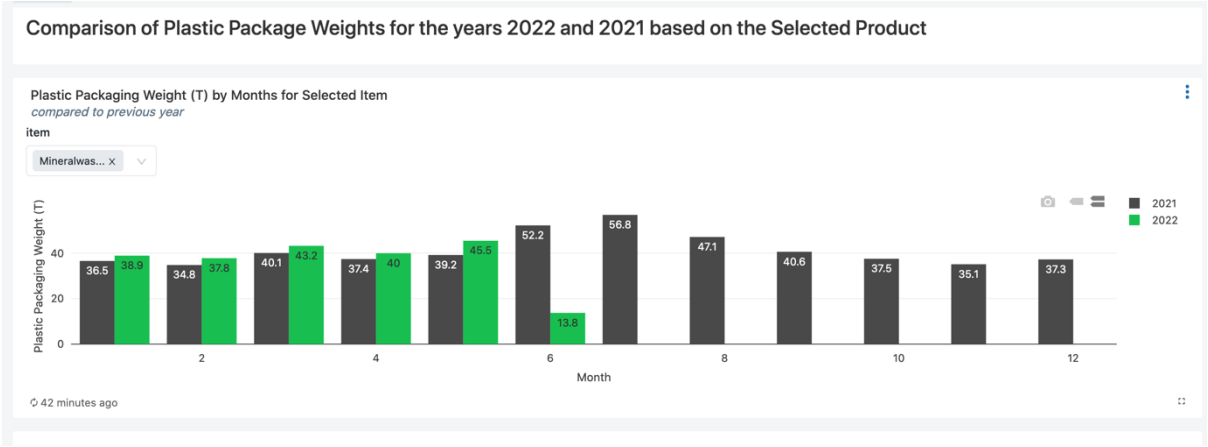


Figure 21 Plastic Packaging Weight by Months for Selected Item

Figure 22 shows the amount of plastic packaging of the selected product over time and the forecasted data for the future. For the Mineralwasser product selected here, the forecast follows the seasonality by considering the 2021 data. Note that the Sold quantity for June 2022 is expected to be close to the predicted value at the end of the month, as it has not been completed yet.

Figure 22 Sold and Predicted Values based on the Item

Summary

In this project, it was aimed to reduce the plastic packaging sold (in t), and in this context, the client aimed for a data product as project output that allows the user to see the plastic packaging sold in tons and compares it with the business targets by seeing the predicted result until the end of the business year. In order to achieve these goals, Time Series Analysis was performed according to the data provided by the client, using tools such as Databricks, Python, and SQL. After data cleaning, data exploration, and time series analysis, forecasting models were built to predict the plastic packaging weight for each item. With the Prophet model with the best performance, monthly predicted values were created until March 2023, at the end of the business year. In the final, a SQL Dashboard was created, with reproducible notebooks as output, in which plastic packaging weight can be observed according to various filters such as a month, year, and item, showing forecasts for the coming months, showing the top 10 items based on most plastic packaging is used for the current month and next months. Some important outputs are as follows; according to the data provided by the Dashboard, the top 10 items by weight of plastic packaging are generally plastic packaged products such as mineral water, tonic, oil. According to the forecast data, this year, the most plastic packaged products will be sold in June. 197.7 T plastic was saved this year with packaging optimization.