VIETNAM NATIONAL UNIVERSITY – HCMC
INTERNATIONAL UNIVERSITY
SCHOOL OF INDUSTRIAL ENGINEERING & MANAGEMENT

COURSE : TIME SERIES & FORECASTING TECHNIQUES

GROUP : 09

Foreasting Sale in Corporación Favorita stores in Ecuador

Student	ID	Contribution
Nguyen Van Phong	IELSIU21357	100%
Nguyen Thi Bich Ngoc	IELSIU21334	100%
Duong Binh Duong	IELSIU21189	100%
Nguyen Minh Duc	IELSIU21276	100%
Le Huy Hoang	IELSIU21299	100%
Van Ngoc Hoang Hai	IELSIU21288	100%

Ho Chi Minh City, 09th January, 2024

Table of Contents

LIST OF FIGURES

PART 1 : PROBLEM DESCRIPTION

1.1 Problem statement

Forecasting is an essential part of any successful business strategy. It involves predicting future outcomes based on past data and trends, allowing businesses to make informed decisions and plan for the future. Whether you are a small startup or a multinational corporation, forecasting plays a crucial role in determining your success. The purpose of this project is to provide knowledge about time series and forecasting techniques by doing research on the Favorita stores located in Ecuador and following the requirements of the project. In order to complete the project, we have employed analytical abilities to choose the most reliable source of information, and we have also used our whole attention to gather and synthesize data to complete the job effectively.

During the preparation and presentation stage, we gained experiences and skills in time management, adapting with pressure, and self-learning. would provide a good basis for future work in our major of Logistics & Supply Chain Management or other curricula.

1.2 Objective

In this project, our team will forecast sales for products offered at Favorita shops throughout Ecuador. The results can be utilized by the procurement department in the following season or period to import the right quantity of this product category. Past data will be taken into consideration, analysed trend, seasonality, and cyclical features. The training data contains dates, store and product information, whether the item was marketed, and sales figures. We have analysed the given data sets, then we predicted sales using forecasting approaches, sensitivity analysis, and compared the errors between the data of different forecasting techniques in order to select the one with the lowest error. Then, we can reduce loss sales and backorder cost, meet the demand during seasons and operate the products' flow effectively.

1.3 Scope

After conducting extensive investigation, we discovered a case on Kaggle, a platform that specialises in delivering data files from many majors, that was acceptable for our group to complete the purpose of the assignment. We choose to forecast sales for thousands of product families sold in Favorita stores located in Ecuador. The training data contains dates, store and product information, whether the item was marketed, and sales figures. Additional files include supplemental information that may be relevant in the construction of our models.

1.4 Limitation

There are various details to which we should pay attention while working on this project. To begin, in the public sector, wages are paid every two weeks on the 15th and final day of the month. This might have an impact on supermarket sales. Second, on April 16, 2016, a magnitude 7.8 earthquake rocked Ecuador. People came together to help with relief efforts, providing water and other basic necessities, which had a significant impact on grocery sales for several weeks following the earthquake.

Furthermore, the lack of knowledge in this subject is also a barrier in the project process, especially how to handle data sets to meet requirements.

PART 2 : TIME SERIES DATASET

Regarding to this project, our team have found some dataset that show statistics about the store' sales - time series forecasting on KAGGLE. Here we will explain specifically about the data we have shown at this report. This data will be the key to forecast sales for thousands of product families offered in Ecuadorian Favorita outlets. Dates, product and shop details, whether the item was being marketed, and sales figures are all included in the training data.

File Descriptions and Data Field Information:

id	date	store_nbr	family	sales	onpromotion
0	1/1/2013	1	AUTOMOTIVE	0	0
1	1/1/2013	1	BABY CARE	0	0
2	1/1/2013	1	BEAUTY	0	0
3	1/1/2013	1	BEVERAGES	0	0
4	1/1/2013	1	BOOKS	0	0
5	1/1/2013	1	BREAD/BAKERY	0	0
6	1/1/2013	1	CELEBRATION	0	0
7	1/1/2013	1	CLEANING	0	0
8	1/1/2013	1	DAIRY	0	0
9	1/1/2013	1	DELI	0	0
10	1/1/2013	1	EGGS	0	0
11	1/1/2013	1	FROZEN FOODS	0	0
12	1/1/2013	1	GROCERY I	0	0
13	1/1/2013	1	GROCERY II	0	0
14	1/1/2013	1	HARDWARE	0	0
15	1/1/2013	1	HOME AND KITCHEN I	0	0
16	1/1/2013	1	HOME AND KITCHEN II	0	0

Figure 1 : Data train.csv

The **training data** consists of the goal **sales** as well as time series of the features **store_nbr, family, and onpromotion.**

store_nbr identified the store where the products are sold.

family indicates the kind of goods being sold.

sales provides the total amount of money sold for a product family at a specific retailer on a specific date. Since things can be sold in fractional units (1.5 kg of cheese as opposed to 1 bag of chips, for example), fractional values are possible.

onpromotion indicated the quantity of products in a product family that were on promotion at a retailer on a specific day.

id	date	store_nbr	family	onpromotion
3000888	8/16/2017	1	AUTOMOTIVE	0
3000889	8/16/2017	1	BABY CARE	0
3000890	8/16/2017	1	BEAUTY	2
3000891	8/16/2017	1	BEVERAGES	20
3000892	8/16/2017	1	BOOKS	0
3000893	8/16/2017	1	BREAD/BAKERY	12
3000894	8/16/2017	1	CELEBRATION	0
3000895	8/16/2017	1	CLEANING	25
3000896	8/16/2017	1	DAIRY	45
3000897	8/16/2017	1	DELI	18
3000898	8/16/2017	1	EGGS	1
3000899	8/16/2017	1	FROZEN FOODS	1
3000900	8/16/2017	1	GROCERY I	64
3000901	8/16/2017	1	GROCERY II	0
3000902	8/16/2017	1	HARDWARE	0
3000903	8/16/2017	1	HOME AND KITCHEN	2
3000904	8/16/2017	1	HOME AND KITCHEN	6

Figure 2 : Data test.csv

The **test data** is identical to the **training data** in terms of characteristics. For the dates in this file, it will give us the information to forecast the goal sales.

The **test data** contains dates for the fifteen days following the final date in the training data.

store_nbr	city	state	type	cluster
1	Quito	Pichincha	D	13
2	Quito	Pichincha	D	13
3	Quito	Pichincha	D	8
4	Quito	Pichincha	D	9
5	Santo Domingo	Santo Domingo de los Tsachilas	D	4
6	Quito	Pichincha	D	13
7	Quito	Pichincha	D	8
8	Quito	Pichincha	D	8
9	Quito	Pichincha	B	6
10	Quito	Pichincha	C	15
11	Cayambe	Pichincha	B	6
12	Latacunga	Cotopaxi	C	15
13	Latacunga	Cotopaxi	C	15
14	Riobamba	Chimborazo	C	7
15	Ibarra	Imbabura	C	15
16	Santo Domingo	Santo Domingo de los Tsachilas	C	3
17	Quito	Pichincha	C	12

Figure 3 : Data store.csv

This file is the store metadata, including **city, state, type** and **cluster**

City, **state** is the place where the store (identified by **store_nbr)** located in.

Type of store can be indicated by alphabet**(A,B,C,D,E)**
cluster is a grouping of similar stores, identified by number

date	dcoilwtico
1/1/2013	
1/2/2013	93.14
1/3/2013	92.97
1/4/2013	93.12
1/7/2013	93.2
1/8/2013	93.21
1/9/2013	93.08
1/10/2013	93.81
1/11/2013	93.6
1/14/2013	94.27
1/15/2013	93.26
1/16/2013	94.28
1/17/2013	95.49
1/18/2013	95.61
1/21/2013	
1/22/2013	96.09
1/23/2013	95.06

Figure 4 : Data oil.csv

This file shows the **dcoilwtico** (price of oil per day) including values from the test and train data periods. (As an oil-dependent nation, Ecuador's economy is extremely susceptible to fluctuations in oil prices.)

holidays_events.csv:

This file shows the holidays and events with metadata.
Pay special attention to the transferred column. A holiday that was officially shifted by the government to a different day falls on that calendar day. A transferred day is not so much a holiday as it is an ordinary day. Find the matching row where type is Transfer to determine the day that it was truly honored. For instance, the Guayaquil Independence Day was moved from October 9, 2012, to October 12, 2012, meaning that it was observed on October 12, 2012. Bridge days are additional days added to holidays, such as to stretch the break over a long weekend. These are usually created by the type Work Day, which is a day (such as Saturday) that is not usually designated for work but is intended to payback the Bridge.

Additional holidays are days added a regular calendar holiday, for example, as typically happens around Christmas (making Christmas Eve a holiday).

date	type	locale	locale_name	description	transferred
3/2/2012	Holiday	Local	Manta	Fundacion de Manta	FALSE
4/1/2012	Holiday	Regional	Cotopaxi	Provincializacion de Cotopaxi	FALSE
4/12/2012	Holiday	Local	Cuenca	Fundacion de Cuenca	FALSE
4/14/2012	Holiday	Local	Libertad	Cantonizacion de Libertad	FALSE
4/21/2012	Holiday	Local	Riobamba	Cantonizacion de Riobamba	FALSE
5/12/2012	Holiday	Local	Puyo	Cantonizacion del Puyo	FALSE
6/23/2012	Holiday	Local	Guaranda	Cantonizacion de Guaranda	FALSE
6/25/2012	Holiday	Regional	Imbabura	Provincializacion de Imbabura	FALSE
6/25/2012	Holiday	Local	Latacunga	Cantonizacion de Latacunga	FALSE
6/25/2012	Holiday	Local	Machala	Fundacion de Machala	FALSE
7/3/2012	Holiday	Local	Santo Domingo	Fundacion de Santo Domingo	FALSE
7/3/2012	Holiday	Local	El Carmen	Cantonizacion de El Carmen	FALSE
7/23/2012	Holiday	Local	Cayambe	Cantonizacion de Cayambe	FALSE
8/5/2012	Holiday	Local	Esmeraldas	Fundacion de Esmeraldas	FALSE
8/10/2012	Holiday	National	Ecuador	Primer Grito de Independencia	FALSE
8/15/2012	Holiday	Local	Riobamba	Fundacion de Riobamba	FALSE
8/24/2012	Holiday	Local	Ambato	Fundacion de Ambato	FALSE
9/28/2012	Holiday	Local	Ibarra	Fundacion de Ibarra	FALSE
10/7/2012	Holiday	Local	Quevedo	Cantonizacion de Quevedo	FALSE
10/9/2012	Holiday	National	Ecuador	Independencia de Guayaquil	TRUE
10/12/2012	Transfer	National	Ecuador	Traslado Independencia de Guayaquil	FALSE

2.1 Pre-processing data

At first, we employed the following libraries in the particular Python language for data analysis:

+ Use pandas and numpy to import and process multidimensional and tabular data.

+ Data visualization tools: plotly, seaborn, and matplotlib.

+ Format data, use XGBRegressor, statistics model, sklearn, and linear regression to assess the model's fitness.

+ Format time: datetime, calendar,

The data should then be uploaded to the group workspace, read, and returned as variables of type pd.DataFrame, int64+.

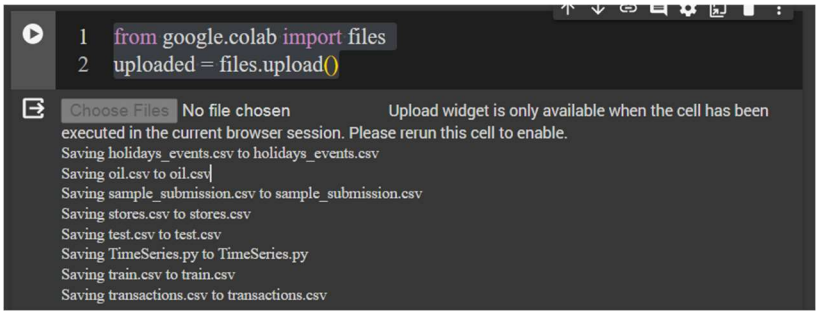


Figure 5 : Upload file into workspace

```
train = pd.read_csv("train.csv")
test = pd.read_csv("test.csv")
oil = pd.read_csv("oil.csv")
transactions = pd.read_csv("transactions.csv")
holidays_events = pd.read_csv("holidays_events.csv")
stores = pd.read_csv("stores.csv")
```

Figure 6 : Read data

Combining the data into a single, shared table that we will refer to as df_data is crucial. Since there are references (overlapping data columns) between data files, combining data will be considerably simpler and more logical. The data files will be combined using the pd.merge() procedure. When the parameter "on" is set, the data will match one another in order to find values that are identical.

```
df_data = pd.concat([train, test], sort=True)
df_data = df_data.merge(stores, how="left", on='store_nbr')
df_data = df_data.merge(oil, how="left", on='date')
df_data = df_data.merge(transactions, how="left", on=['date','store_nbr'])
df_data = df_data.merge(holidays_events,on='date',how='left')
df_data = df_data.rename(columns={'type_x' : 'store_type','type_y':'holiday_type'})
```

Figure 7 Merge all data into one

	date	family	id	onpromotion	sales	store_nbr	city	state	store_type	cluster	...	holiday_type	locale	locale_name	description	transferred	year	month	week	quarter	day_of_week
0	2013-01-01	AUTOMOTIVE	0	0	0.0	1	Quito	Pichincha	D	13	...	Holiday	National	Ecuador	Primer dia del ano	False	2013	1	1	1	Tuesday
1	2013-01-01	BABY CARE	1	0	0.0	1	Quito	Pichincha	D	13	...	Holiday	National	Ecuador	Primer dia del ano	False	2013	1	1	1	Tuesday
2	2013-01-01	BEAUTY	2	0	0.0	1	Quito	Pichincha	D	13	...	Holiday	National	Ecuador	Primer dia del ano	False	2013	1	1	1	Tuesday
3	2013-01-01	BEVERAGES	3	0	0.0	1	Quito	Pichincha	D	13	...	Holiday	National	Ecuador	Primer dia del ano	False	2013	1	1	1	Tuesday
4	2013-01-01	BOOKS	4	0	0.0	1	Quito	Pichincha	D	13	...	Holiday	National	Ecuador	Primer dia del ano	False	2013	1	1	1	Tuesday

5 rows × 22 columns

Figure 8 Data Frame: df_data

2.1.1 Format time

The time units in the data file must be reformatted since pandas will require compliance with a date format. Here, for convenience of implementation, we shall split it into two sections. In particular, convert the files' timestamps to the following format: "%Y-%m-%d".

```
# Convert all 'date' columns to datetime Pandas format
holidays_events['date'] = pd.to_datetime(holidays_events['date'], format = "%Y-%m-%d")
oil['date'] = pd.to_datetime(oil['date'], format = "%Y-%m-%d")
transactions['date'] = pd.to_datetime(transactions['date'], format = "%Y-%m-%d")
train['date'] = pd.to_datetime(train['date'], format = "%Y-%m-%d")
test['date'] = pd.to_datetime(test['date'], format = "%Y-%m-%d")
```

Figure 9 Data Frame: df_data

With respect to the whole data file df_data, we shall format it as follows:

```
df_data.date = pd.to_datetime(df_data.date)
df_data['year'] = df_data['date'].dt.year
df_data['month'] = df_data['date'].dt.month
df_data['week'] = df_data['date'].dt.isocalendar().week
df_data['quarter'] = df_data['date'].dt.quarter
df_data['day_of_week'] = df_data['date'].dt.day_name()
df_data.head()
```

Figure 10 Format time

We will be able to represent datatime when exhibited in the image in more ways if we format in two separate approaches.

2.2.2 Fill NaN value

Now, taking into account the issue of missing values for oil price, we are going to fill them by **backward fill technique**. That means filling missing values with next data point (Forward filling means fill missing values with previous data).

```
1  df_data['dcoilwtico'] = df_data['dcoilwtico'].fillna(method='bfill')
2  df_data.dcoilwtico.isnull().sum()
```

Figure 11 Fill NaN in oil data

With respect to transactions, we understand that since there is no data recorded, this is 0.

```
4  df_data.transactions = df_data.transactions.replace(np.nan,0)
```

Figure 12 Fill NaN in transaction data

As we can see above the **holidays_events** DataFrame contains a row for each of the national, regional or local holidays. The transferred column refers to whether the holiday has been moved or not. We assume then that the missing data corresponding to this DataFrame in the training set correspond to those days for which no public holiday has been recorded. Therefore, we will replace the type by **Working day**. The rest of the categorical variables in this DataFrame will be changed to the empty string, and in transferred we will set all values to **false.**

```
6  df_data[['locale','locale_name', 'description']] = df_data[['locale','locale_name', 'description']].replace(np.nan,'')
7  df_data['holiday_type'] = df_data['holiday_type'].replace(np.nan,'Working Day')
8  df_data['transferred'] = df_data['transferred'].replace(np.nan,False)
```

Figure 13 Fill NaN in holiday_events into working day value

2.1.3 Overview data

In the train data set, this code determines the average sales level of shop groups, item groups, and cluster groups. The data is separated, grouped into "store, family, and cluster" groups, and the "mean" value is taken into consideration in "ascending" order. In addition,

To obtain a portion of the data from the start to the train.shape[0] row, which corresponds to the train data set, use df_data[:train.shape[0]].

- Group names are arranged into distinct columns via reset_index()

```
df_st_sa = df_data[:train.shape[0]].groupby('store_type').agg({"sales" : "mean"}).reset_index().sort_values(by='sales', ascending=False)
df_fa_sa = df_data[:train.shape[0]].groupby('family').agg({"sales" : "mean"}).reset_index().sort_values(by='sales', ascending=False)[:10]
df_cl_sa = df_data[:train.shape[0]].groupby('cluster').agg({"sales" : "mean"}).reset_index()
# chart color
df_fa_sa['color'] = '#496595'
df_fa_sa['color'][2:] = '#c6ccd8'
df_cl_sa['color'] = '#c6ccd8'

# chart
fig = make_subplots(rows=2, cols=2,
                    specs=[[ {"type": "bar"}, {"type": "pie"}],
                           [{"colspan": 2}, None]],
                    column_widths=[0.7, 0.3], vertical_spacing=0, horizontal_spacing=0.02,
                    subplot_titles=("Top 10 Highest Product Sales", "Highest Sales in Stores", "Clusters Vs Sales"))
fig.add_trace(go.Bar(x=df_fa_sa['sales'], y=df_fa_sa['family'], marker=dict(color= df_fa_sa['color']),
                     name='Family', orientation='h'),
                     row=1, col=1)
fig.add_trace(go.Pie(values=df_st_sa['sales'], labels=df_st_sa['store_type'], name='Store type',
                     marker=dict(colors=['#334668','#496595','#6D83AA','#91A2BF','#C8D0DF']), hole=0.7,
                     hoverinfo='label+percent+value', textinfo='label'),
                     row=1, col=2)
fig.add_trace(go.Bar(x=df_cl_sa['cluster'], y=df_cl_sa['sales'],
                     marker=dict(color= df_cl_sa['color']), name='Cluster'),
                     row=2, col=1)
# styling
fig.update_yaxes(showgrid=False, ticksuffix=' ', categoryorder='total ascending', row=1, col=1)
fig.update_xaxes(visible=False, row=1, col=1)
fig.update_xaxes(tickmode = 'array', tickvals=df_cl_sa.cluster, ticktext=[i for i in range(1,17)], row=2, col=1)
fig.update_yaxes(visible=False, row=2, col=1)
fig.update_layout(height=500, bargap=0.2,
                  margin=dict(b=0,r=20,l=20), xaxis=dict(tickmode='linear'),
                  title_text="Average Sales Analysis",
                  template="plotly_white",
                  title_font=dict(size=29, color='#8a8d93', family="Lato, sans-serif"),
                  font=dict(color='#8a8d93'),
                  hoverlabel=dict(bgcolor="#f2f2f2", font_size=13, font_family="Lato, sans-serif"),
                  showlegend=False)
fig.show()
```

Figure 14 Code for visualize overall business

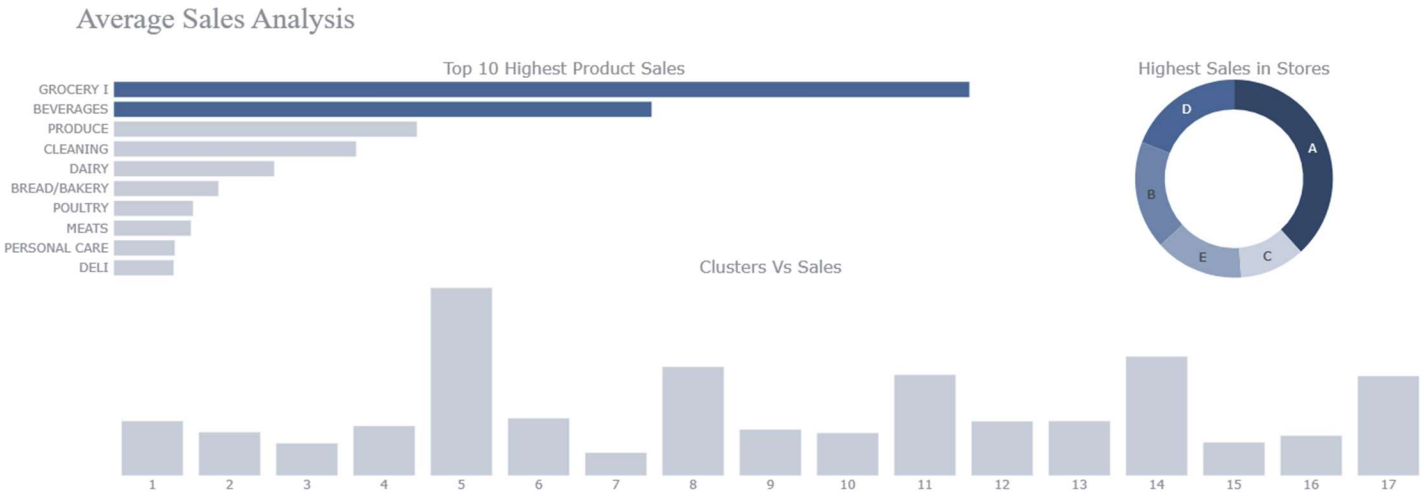


Figure 15 Average Sale Analysis

As can be observed, the two product categories that sell the best at stores are groceries (3.8 million sold) and beverages (2.8 million sold), with produce and cleaning supplies coming in second and third. With 1.1 million sales, store number five has the highest sales volume. Points 8, 11, 14, and 17 are next in line, in that order. With 38% of sales, product code A has the largest volume, followed by code D, and code C has the lowest.

Different product groups have fairly big average sales. This indicates that the company sells a wide variety of goods.The "Grocery I" product group has the greatest average sales, which could be attributed to a number of things, including strong consumer demand, aggressive pricing, or successful marketing techniques. The average sales of the remaining product groupings are lower, which could be caused by a variety of factors such high prices, intense competition, or low client demand.

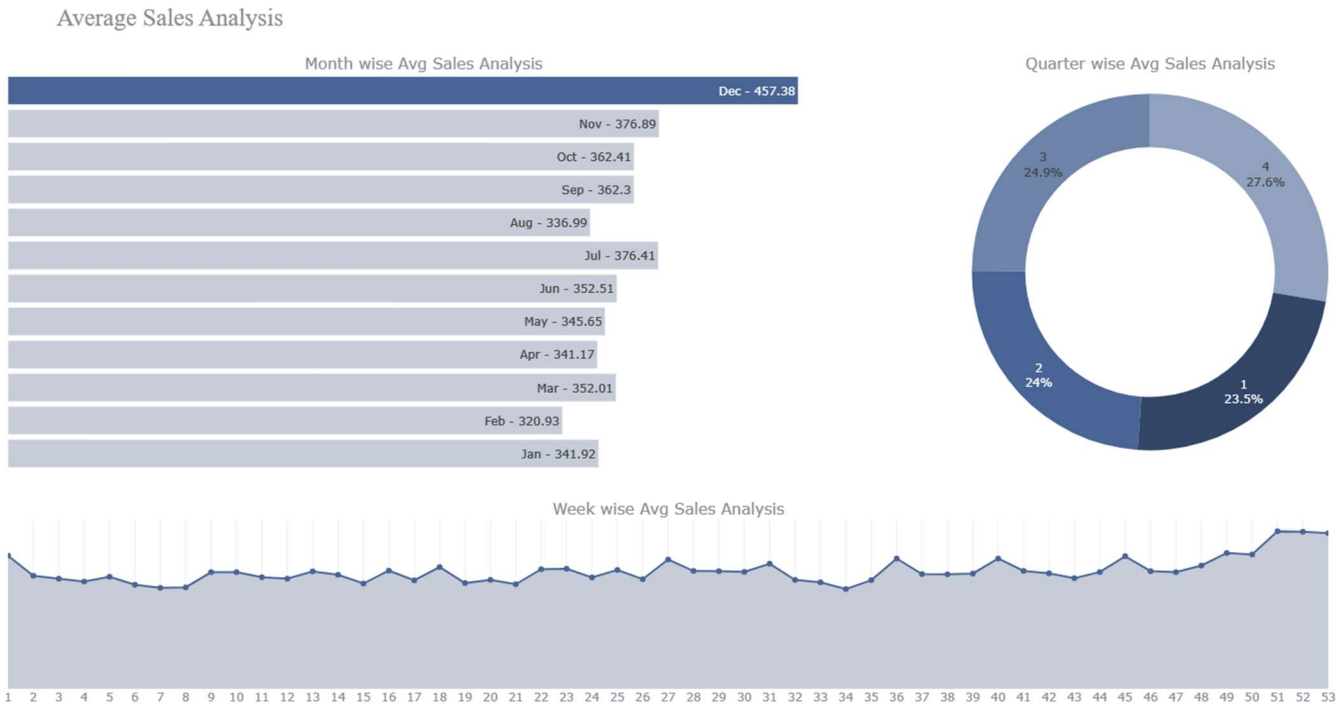


Figure 16 Average value of sales by (month, week, quarter).

The average monthly sales climbed progressively from January to December, with the largest rise of 27.6% occurring between September and October, as seen in the column chart on the left. With $457.38 in average sales, December was the month with the highest sales.

Periodic evaluation:

The average quarterly sales climbed progressively from quarter 1 to quarter 4, with the largest rise of 24.9% occurring from quarter 3 to quarter 4, as shown by the column chart on the right. With an average sales of 1,369.52 USD, the fourth quarter has the greatest average sales.

Analysis by cluster group: With 483.22 USD in average sales, cluster group 1 has the largest sales, as seen by the pie chart. At 322.99 USD, cluster group 2 has the lowest average sales. Generally speaking, this company's typical sales rise from January to December, from the first to the fourth quarter, and based on cluster groups. The growth was not uniform, though, with the largest increase occurring between September and October at a rate of 27.6%.

- This company has very high average revenues, particularly in the fourth quarter. This indicates that the company is doing well and has a lot of room to grow.
- Over time, sales growth varies. Numerous things, including the weather, special occasions, or marketing initiatives, could be to blame for this.
- The various cluster groups' average sales are relatively high. This demonstrates that the company caters to a wide range of clientele.

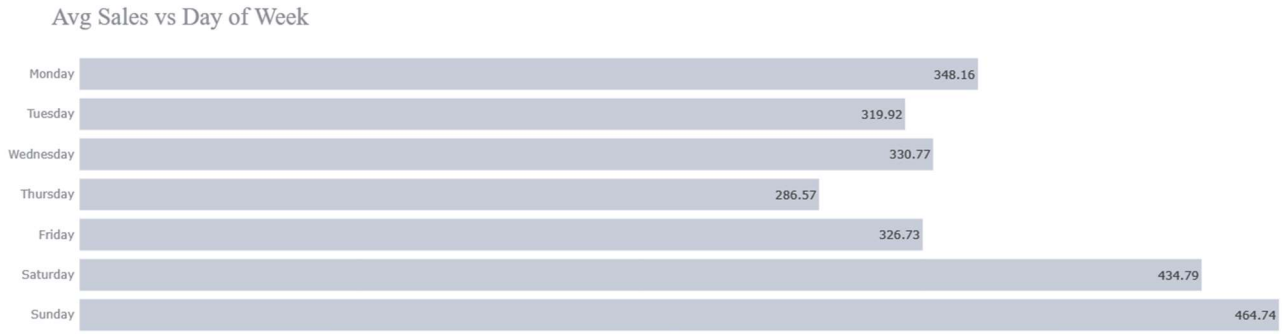


Figure 17 Average sales by day of the week.

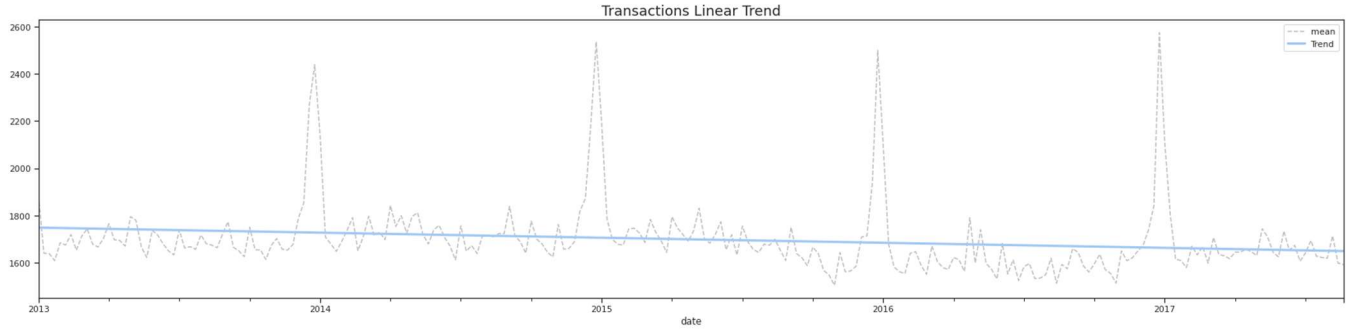
The two days with the highest average sales, Saturday and Sunday, were $434.79 and $464.74, respectively. Wednesday had the lowest average sales, $330.77.

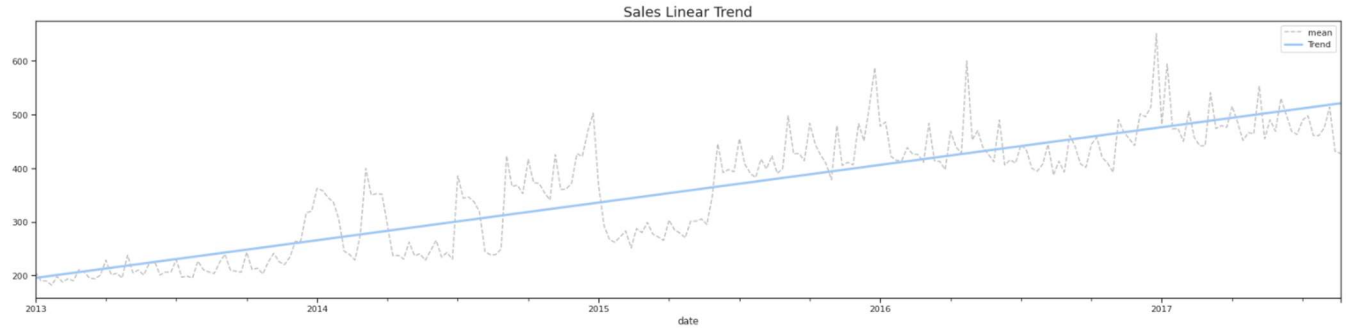
PART 3: ANALYSIS OUTLIERS

3.1 Trend

The trend component of a time series represents a persistent, long-term change in the mean of the series. The trend is the slowest-moving part of a series, the part representing the largest time scale of importance. In a time series of product sales, an increasing trend might be the effect of a market expansion as more people become aware of the product year by year.

Here we use the data of transactions and sales as sample for illustrating their trend, as shown in the below figures





In the illustration, the dashed line represents the average of sales and transactions for all stores over the time axis on a weekly basis.

It's obviously that the trend line of average sales is going up which implies a surge in sales value while there is a plunge in the number of average transactions per week as the trend line is downward. This is possibly due to the change in customer's buying behavior; they tend to buy in a larger quantity each time or maybe there is an increase in the sale of high value items.

Recognition of trend can enable the firm to have better control on the inventory,

3.2 Seasonality

Seasonality is a pattern of variation in a time series that occurs at regular intervals, such as daily, weekly, monthly, or yearly. Seasonality can be caused by various factors, such as weather, holidays, or business cycles. Seasonality can affect the analysis and forecasting of time series data, so it is important to identify and measure it. There are several graphical and statistical methods for detecting seasonality in time series data. Graphical methods include plotting the data over time, using seasonal subseries plots, or using seasonal decomposition plots. Statistical methods include using autocorrelation functions, periodograms, or tests for seasonality, such as the Kruskal-Wallis test or the ANOVA test.

In this report, we used the data from Kaggle to explore some methods for detecting and quantifying seasonality in store sales data. We learned how to use seasonal decomposition, autocorrelation plots, and seasonal subseries plots to identify the seasonal patterns and their strength. We also practiced how to remove the seasonal component from the data using differencing and STL decomposition. By doing so, we were able to isolate the trend and the irregular components of the time series and analyze them more clearly.

Specifically, we use seasonal trend plots about the data of sales and transactions to discover seasonal patterns. The possible code and explanation using for detecting the seasonal trend of the data will be given in Part 4.3.

PART 4: APPLICATION OF MULTIPLE LEARNED FORECASTING MODELS/METHODS

4. 1 Linear Regression

For time series data forecasting, use the linear regression algorithm.

- A distinct split of the data is shown by each subplot in the picture. The dark orange line (darkorange) in each subplot represents validation data, while the blue line (dodgerblue) represents training data.
- Each subplot's title indicates the breakdown of the data within that subplot.

- A moving window of variable size is used to divide the data in the subplot in the first row (Rolling Window with Adjusting Training Size, first column). The movable window size in this division is increased from one data point to ten data points for each division. This implies that the training set's size will vary with each split.

The method of dividing the data with a moving window of a fixed size is illustrated in the second column (Rolling Window with Constant Training Size) of the subplot in the first row. The movable window size in this split is 5 data points. This implies that there will always be five data points in the training set.

- The second column (Rolling Window with Constant Training Size) in the subplot in the first row illustrates how to divide the data using a moving window with a constant size. The movable window size in this split is 5 data points. This implies that there will always be five data points in the training set.

In this instance, splitting the data using a rolling window with constant training size—a moving window of fixed size—can yield more accurate forecasts than dividing the data using a fixed-size moving window. Rolling window with training size adjustment.

This might be the result of the second case's training set size fluctuating with each split, which makes it more challenging for the model to identify long-term patterns in the data.

This might be the result of the second case's training set size fluctuating with each split, which makes it more challenging for the model to identify long-term patterns in the data.

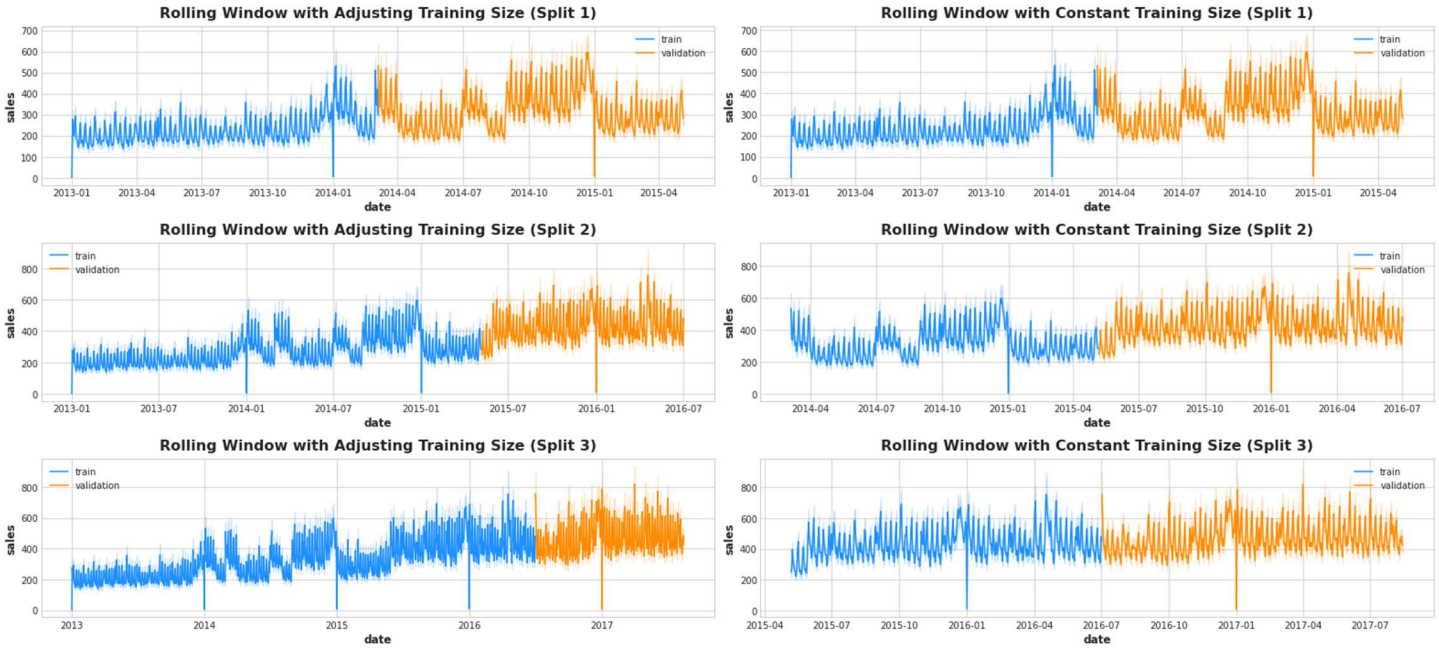


Figure 18 Rolling window with adjusting train size

4.2 ACF/ PACF

In our problem, we have multiple time series and each time series have different pattern of course. It is that those time series consists of store-product family combinations and we have 54 stores and 33 product families. We can't examine all of them one by one. For this reason, we look at average sales for each product but it will be store independent. In addition, the test data contains 15 days for each family. We should be careful when selecting lag features. We can't create new lag features from 1 lag to 15 lag. It must be starting 16.

Our expression for the product of ACF and PACF is as follows:
- Compile the sales column's average data according to the product category and date.
- Repeat for every kind of product.

- Create ACF and PACF sales graphs for every product type, lagging by 365 days.
- If there is an issue with a particular product type, disregard the errors.

Remove any rows from df_data where the sales column contains null values.
Determine the sales column's average value based on the product type (family) and date.
Make the date column the time axis and convert the results to a dataframe type.
This line iterates across each of the unique values in dataframe a's family column. I is the value of the current product type, and num is the variable that represents the iteration order.
plt.subplots(1,2,figsize=(15,5)) = fig, ax: Make two 15 by 5 subplots in a row.
temp = a[i = a.family]A.Sales.NotNull() #(a): filters data from dataframe a containing product type I data, removing null values from the sales column (commented).

sm.graphics.tsa.plot_acf(temp.sales, lags=365, ax=ax[0], title = "AUTOCORRELATION" + i): Plot the autocorrelation (ACF) of the sales column on the first subplot, titled "AUTOCORRELATION" + product type name, in the temp dataframe with lags up to 365 days. Plot the partial autocorrelation (PACF) of the sales column in a temp dataframe with lags up to 365 days on the second subplot, titled "PARTIAL AUTOCORRELATION" + product type name. sm.graphics.tsa.plot_pacf(temp.sales, lags=365, ax=ax[1], title = "PARTIAL AUTOCORRELATION" + i)

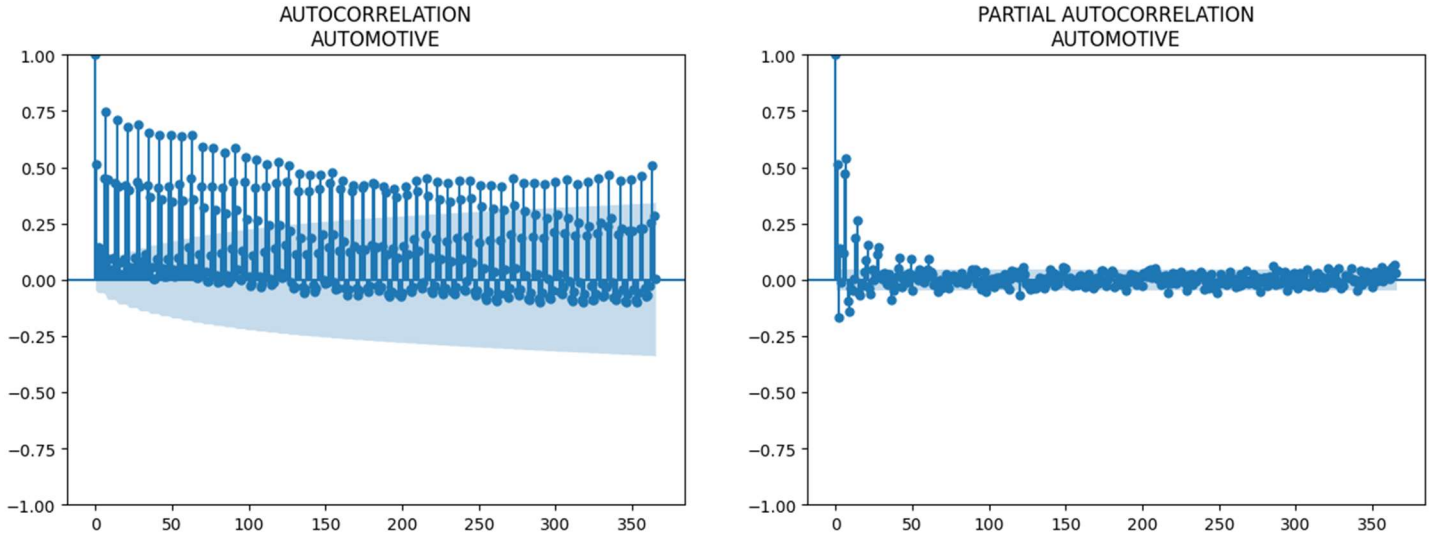


Figure 19 ACF & PACF of Product : Automative

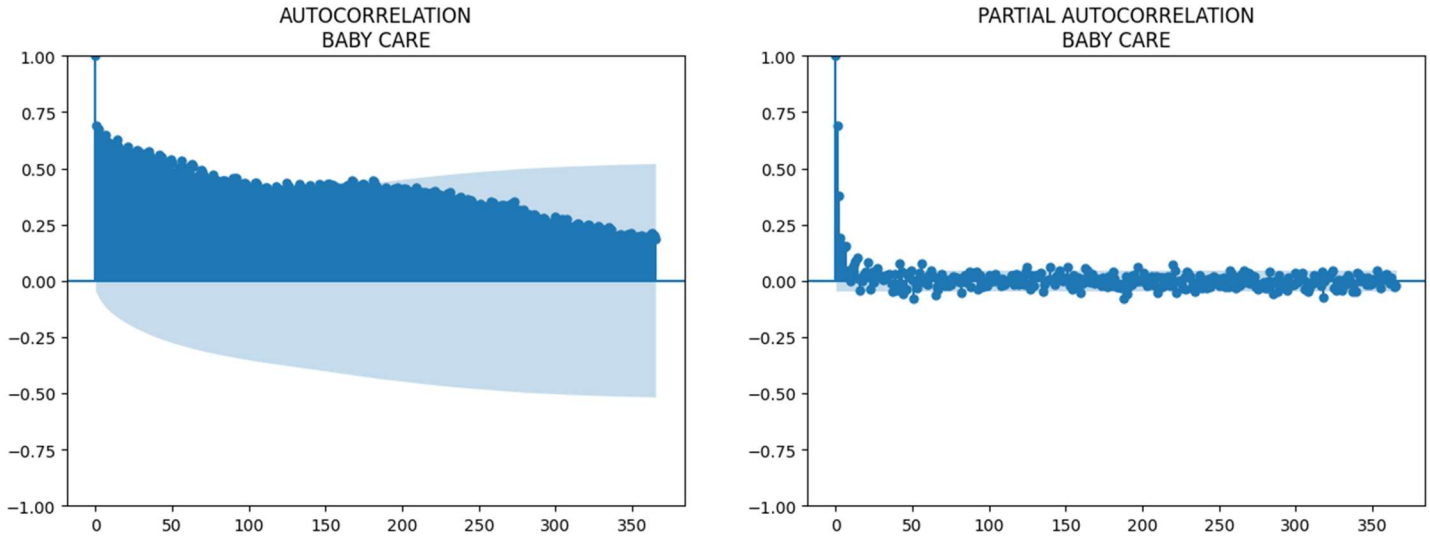


Figure 20 ACF & PACF of Product : Baby Care

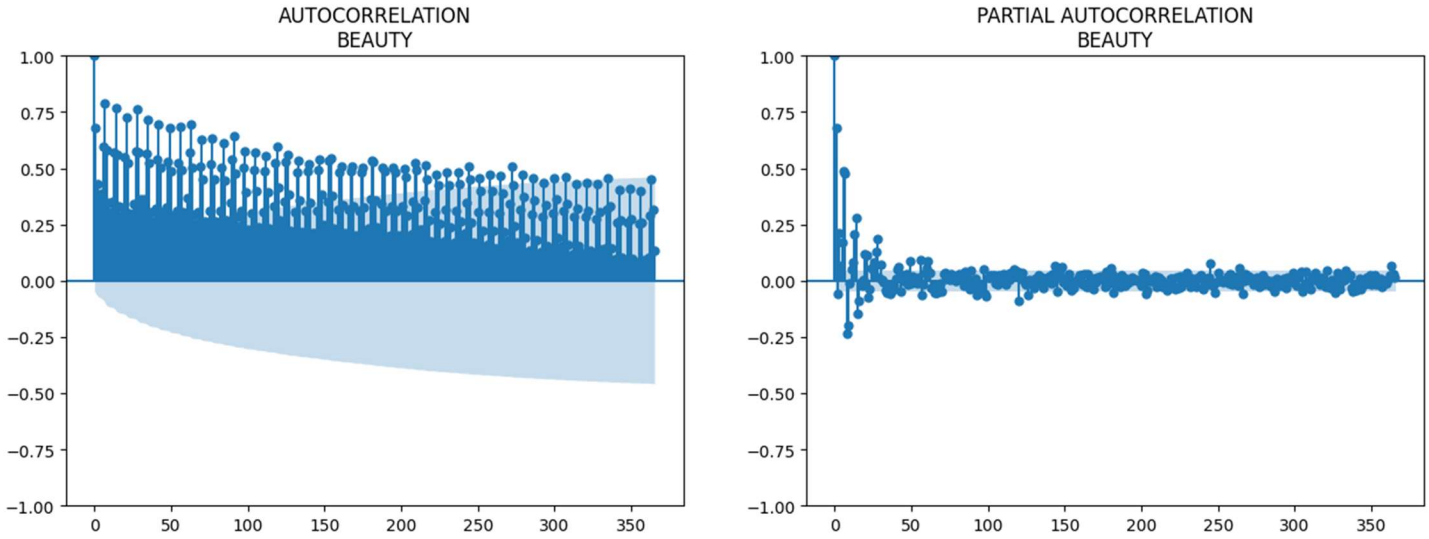


Figure 21 ACF & PACF of Product : Beauty

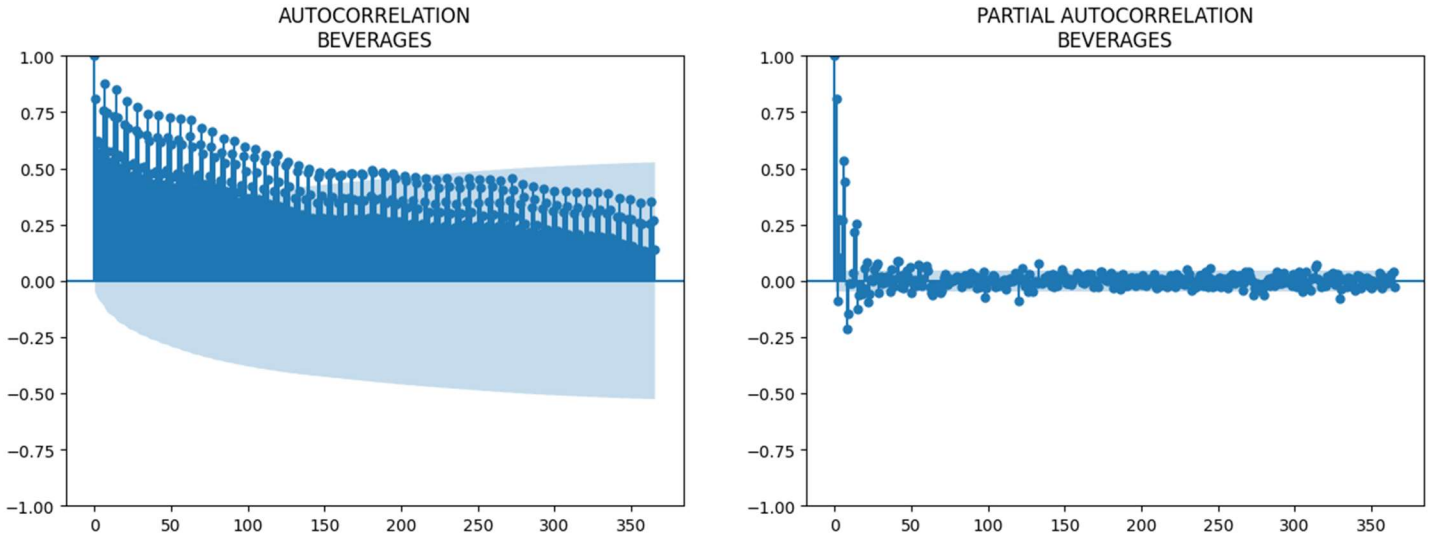
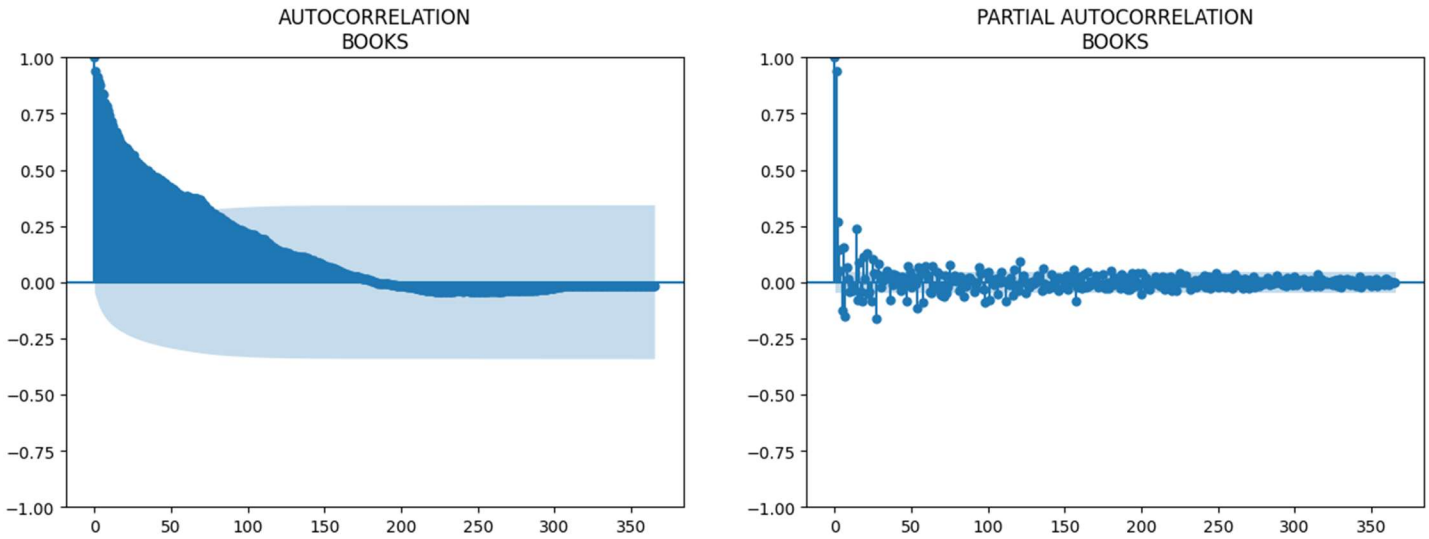


Figure 22 ACF & PACF of Product : Beverages



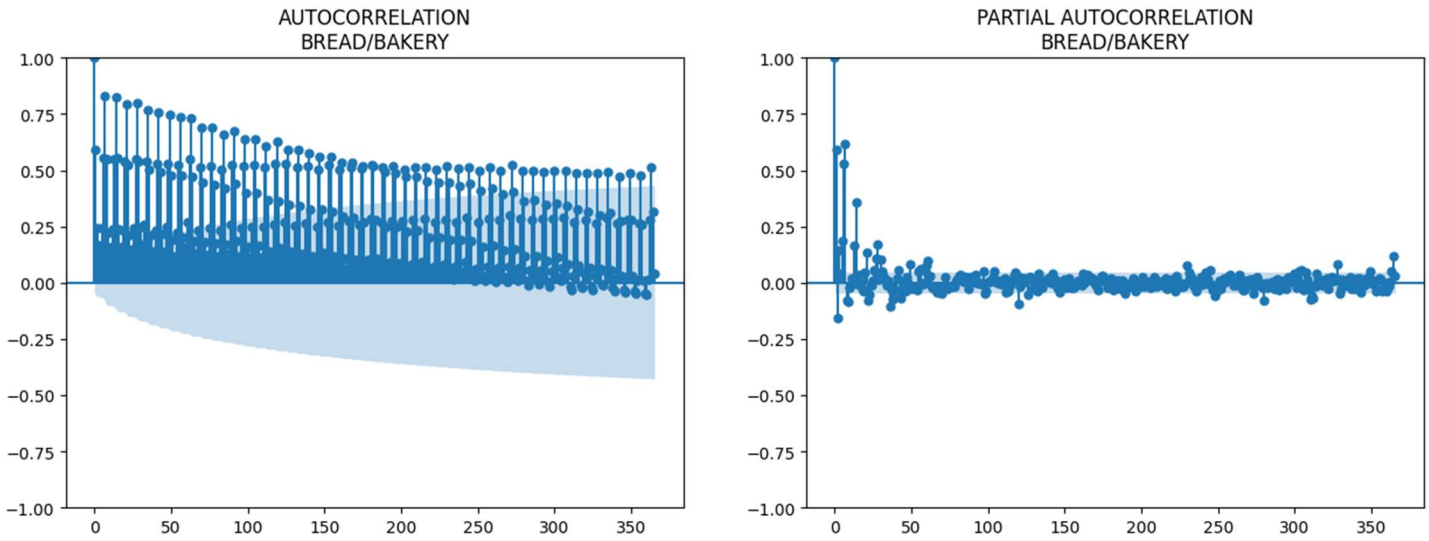


Figure 23 ACF & PACF of Product : Bread / Bakery

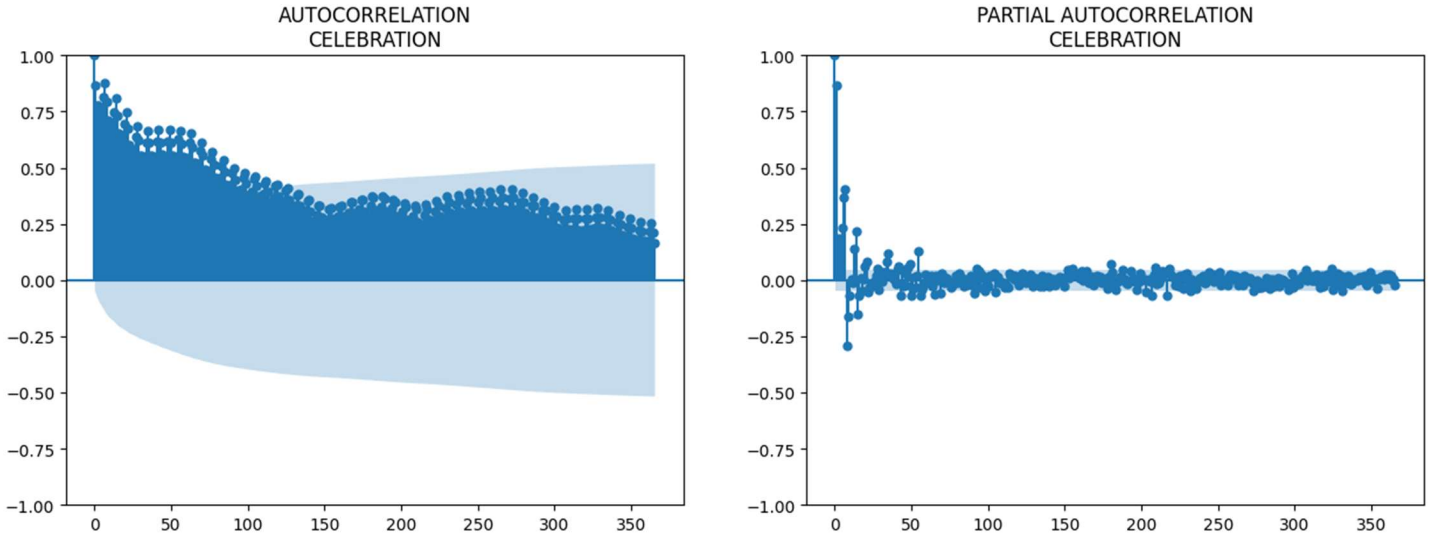


Figure 24 ACF & PACF of Product : Celebration

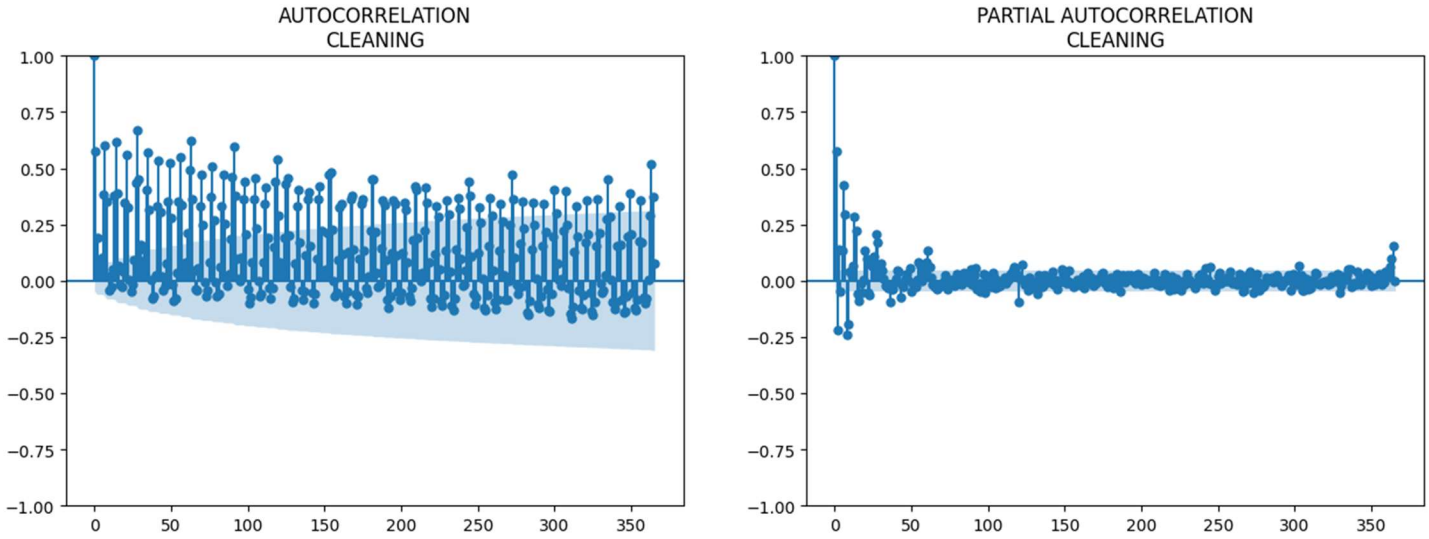


Figure 25 ACF & PACF of Product : Cleaning

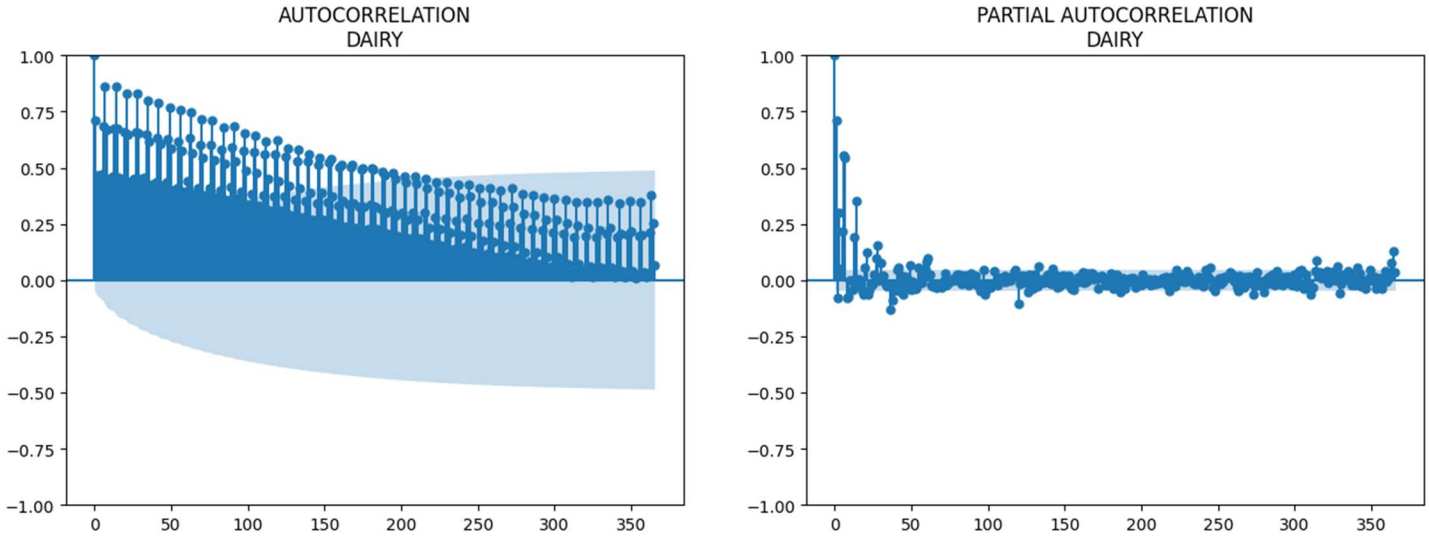


Figure 26 ACF & PACF of Product : Dairy

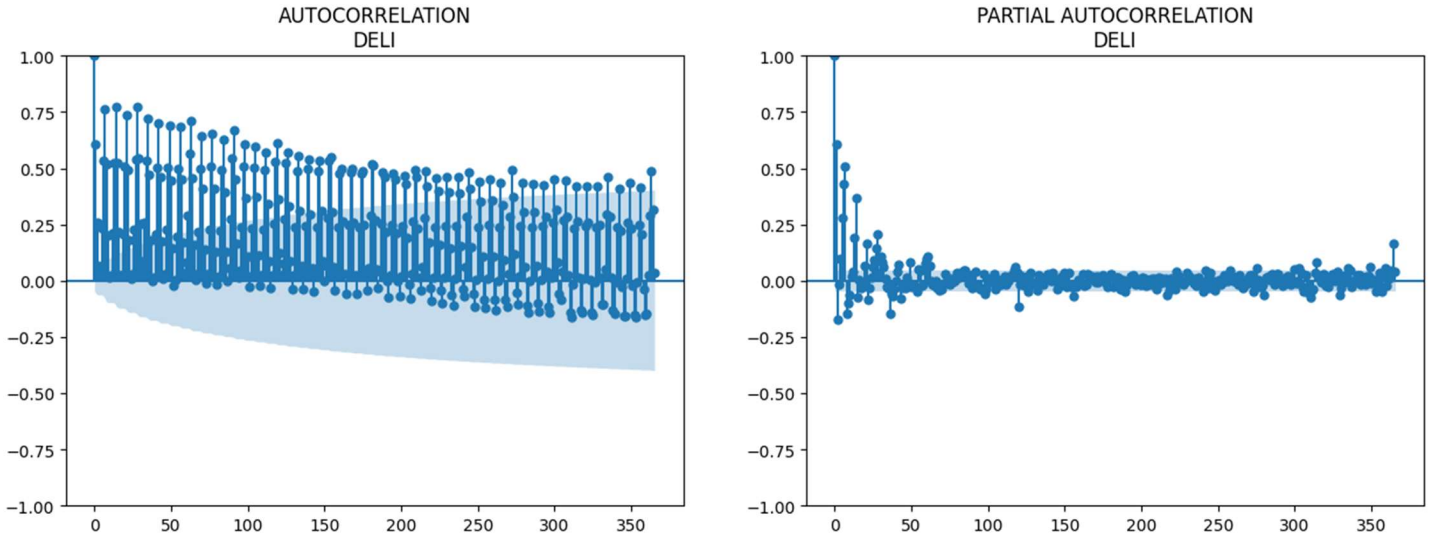


Figure 27 ACF & PACF of Product : Deli

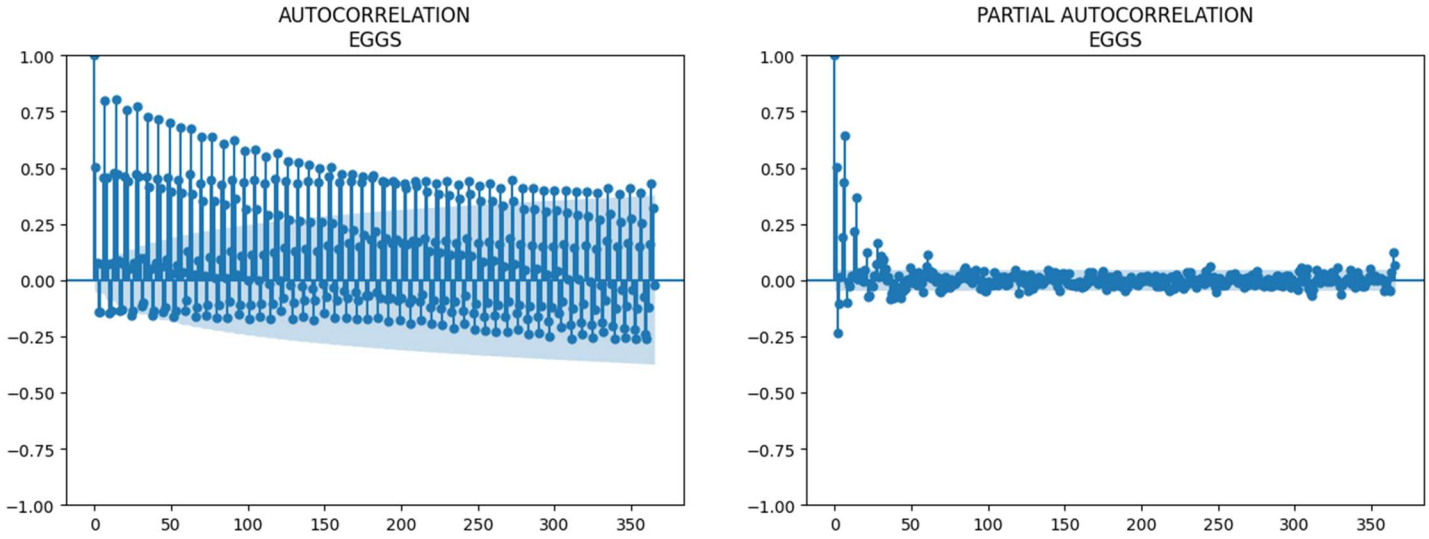


Figure 28 ACF & PACF of Product : Eggs

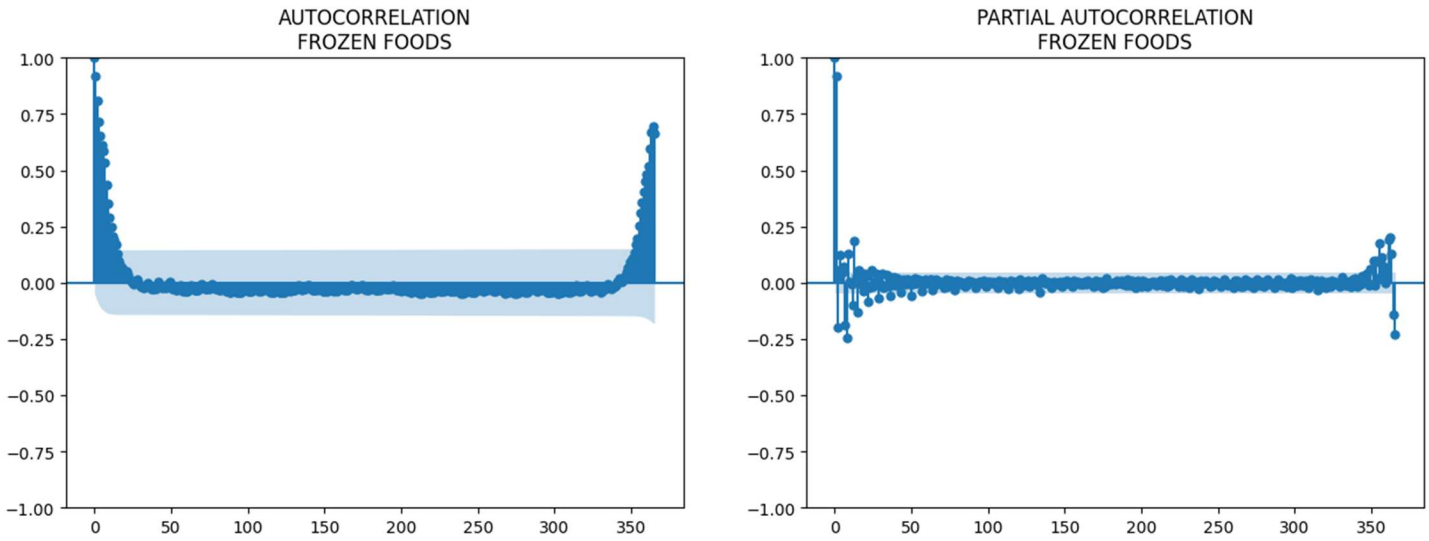


Figure 29 ACF & PACF of Product : Frozen Foods

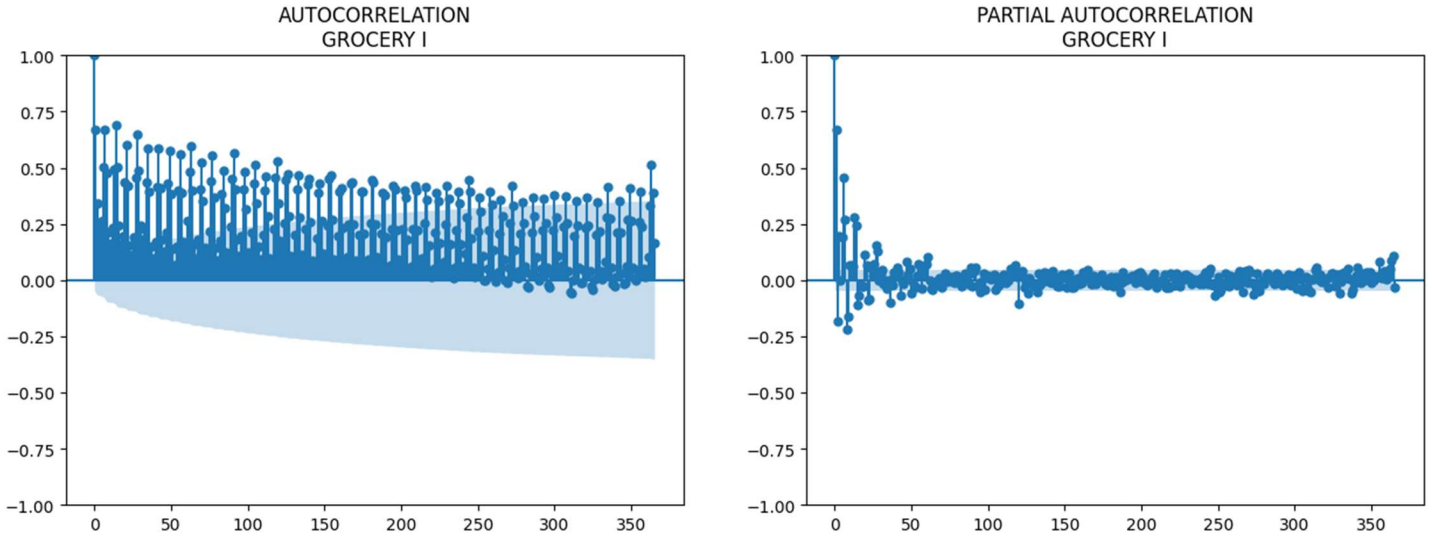


Figure 30 ACF & PACF of Product : Grocery I

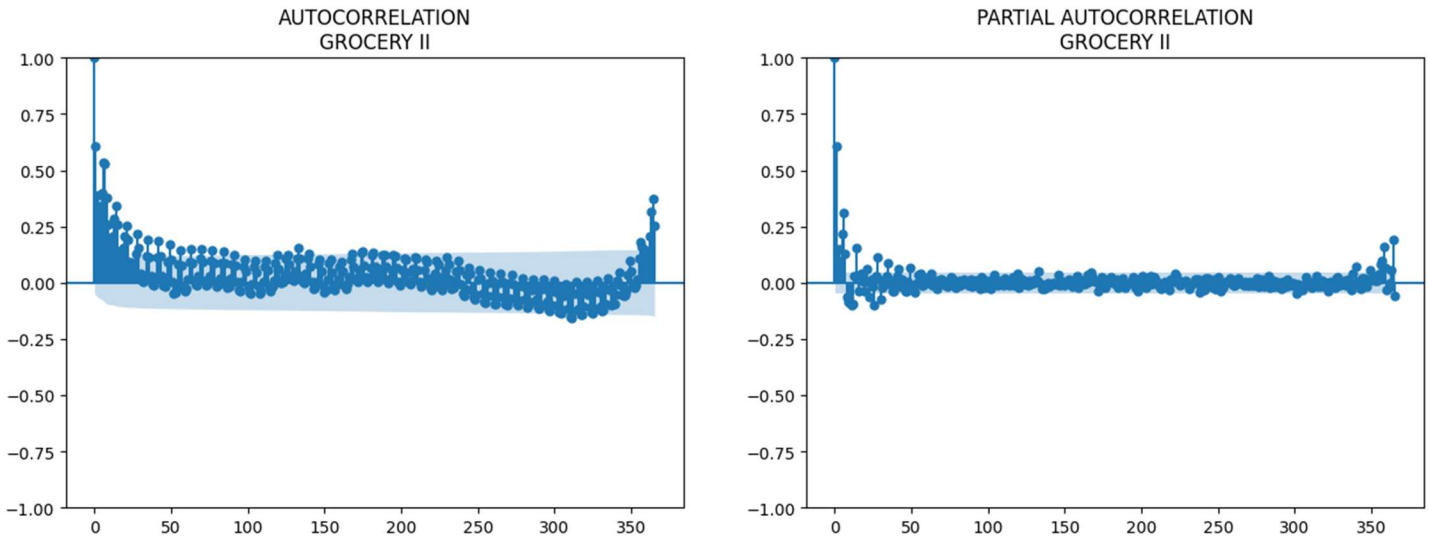


Figure 31 ACF & PACF of Product : Grocery II

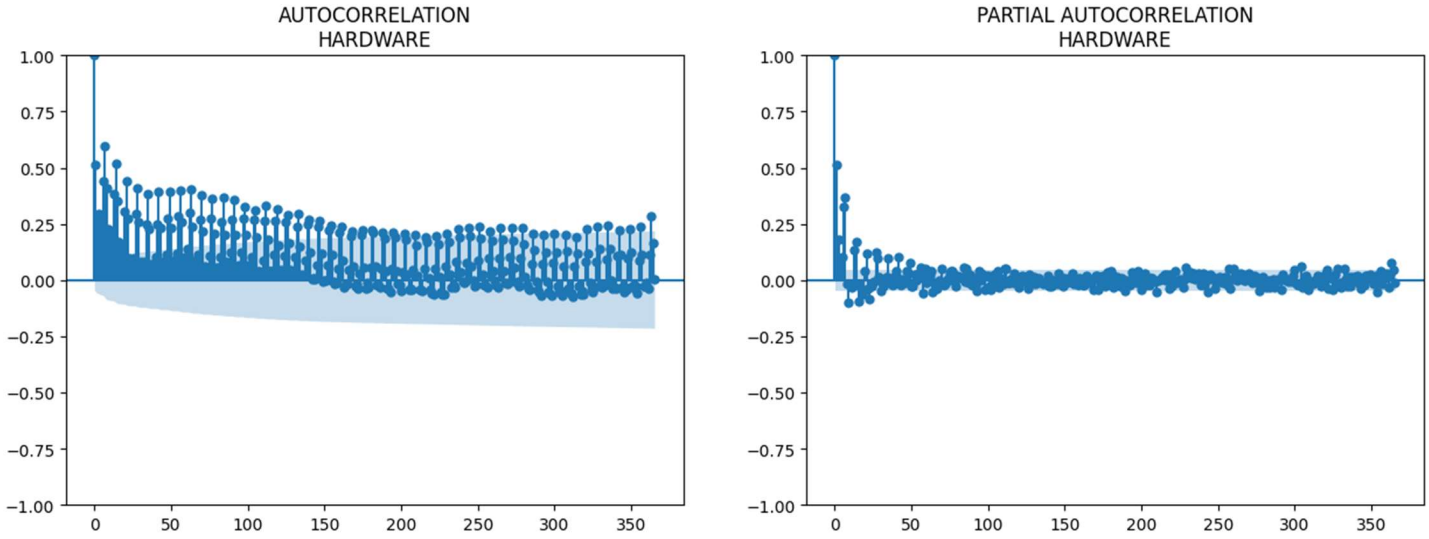


Figure 32 ACF & PACF of Product : Hardware

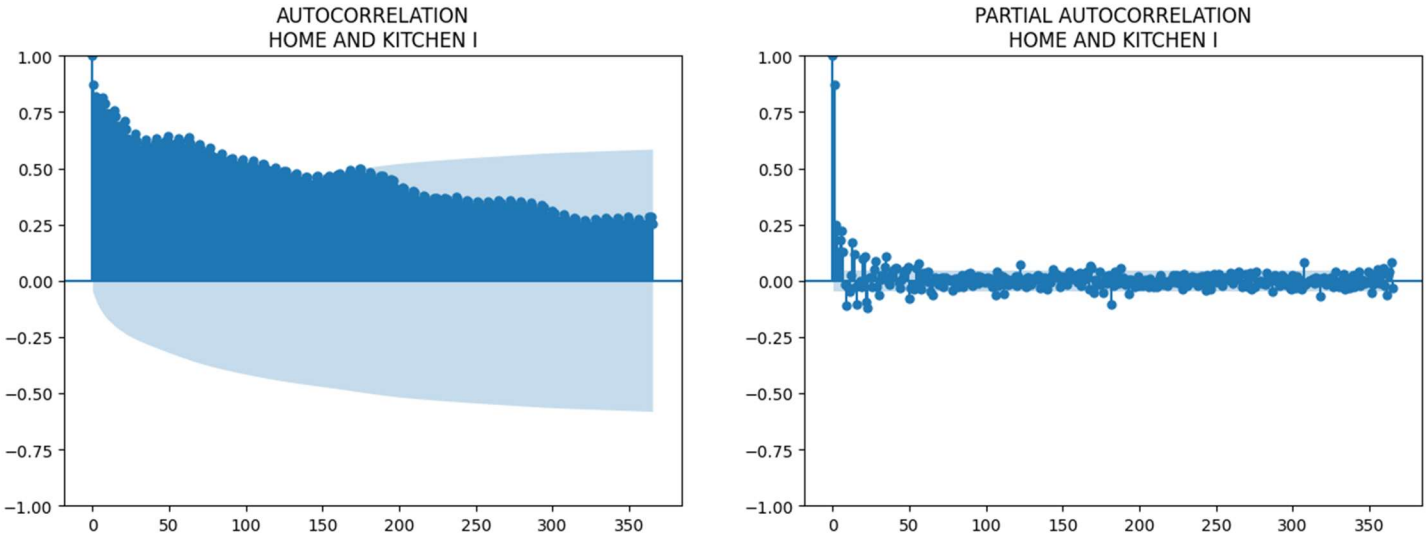


Figure 33 ACF & PACF of Product : Home and Kitchen I

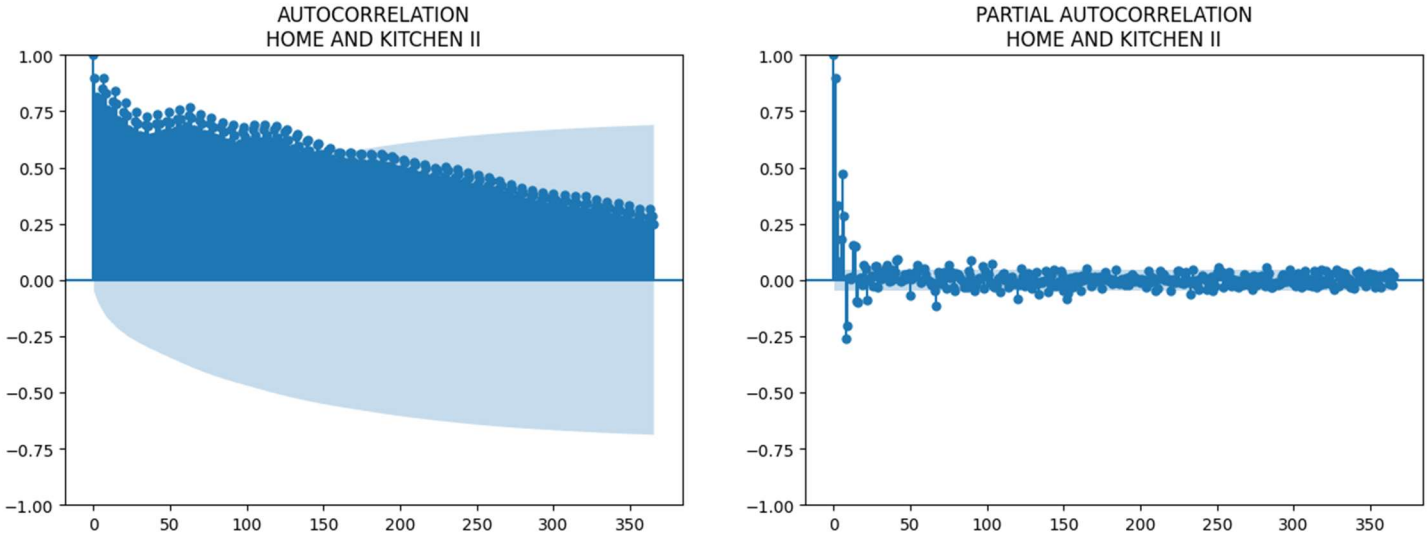


Figure 34 ACF & PACF of Product : Home and Kitchen II

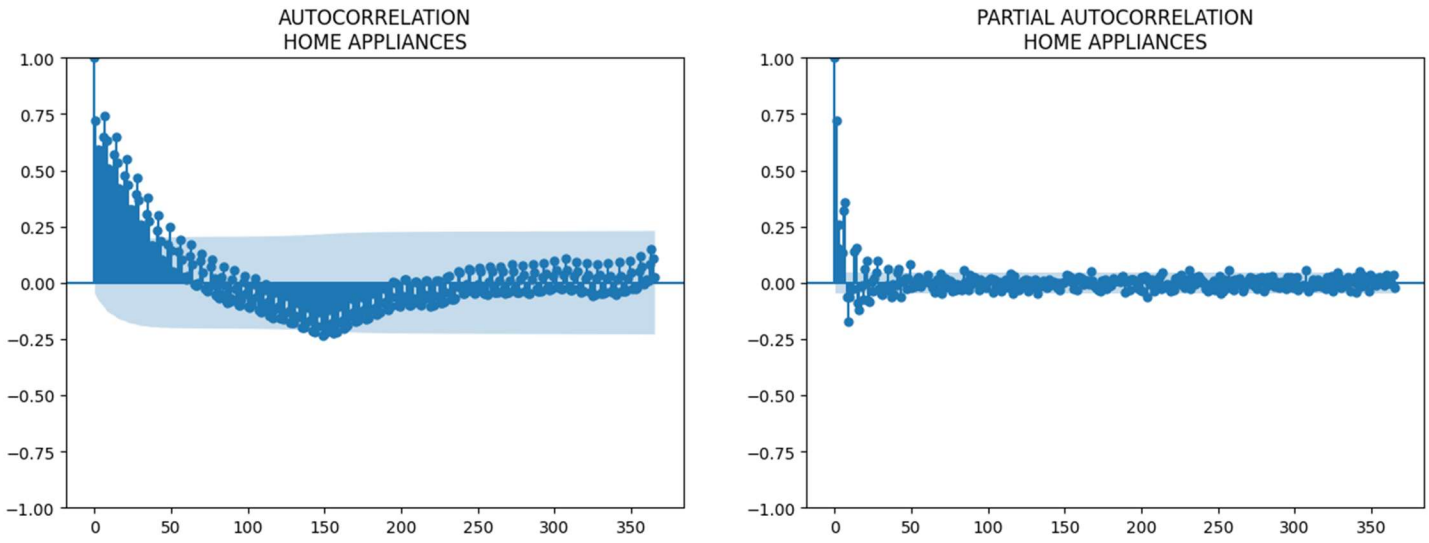


Figure 35 ACF & PACF of Product : Home Appliance

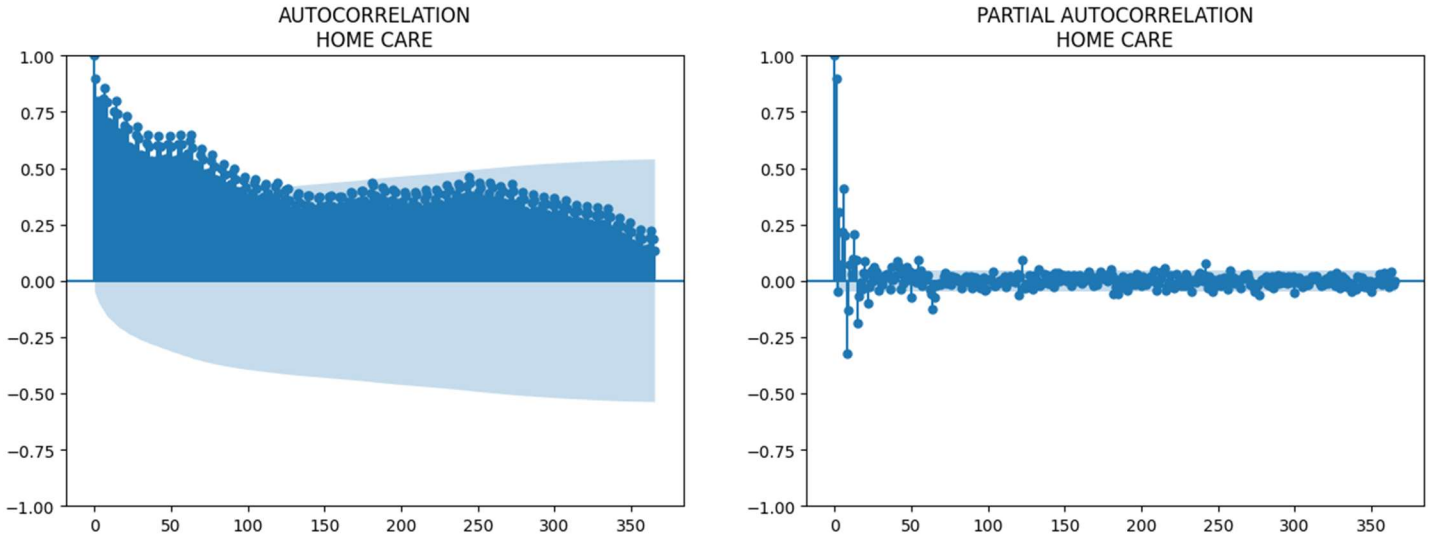


Figure 36 ACF & PACF of Product : Home Care

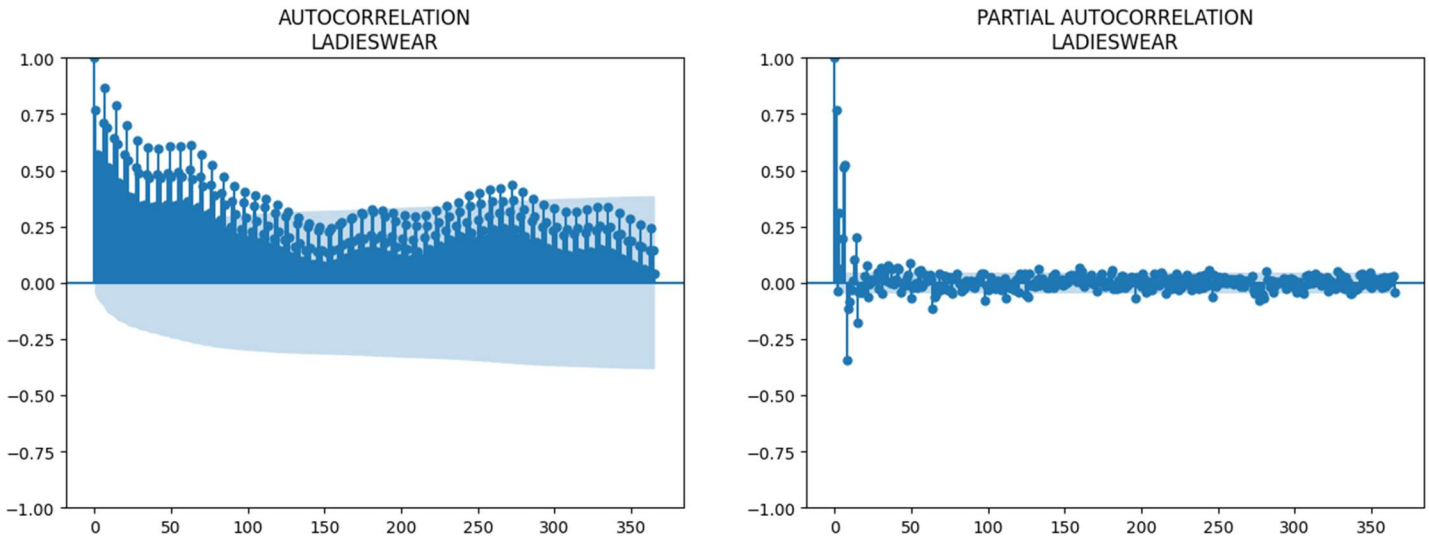


Figure 37 ACF & PACF of Product : Ladieswear

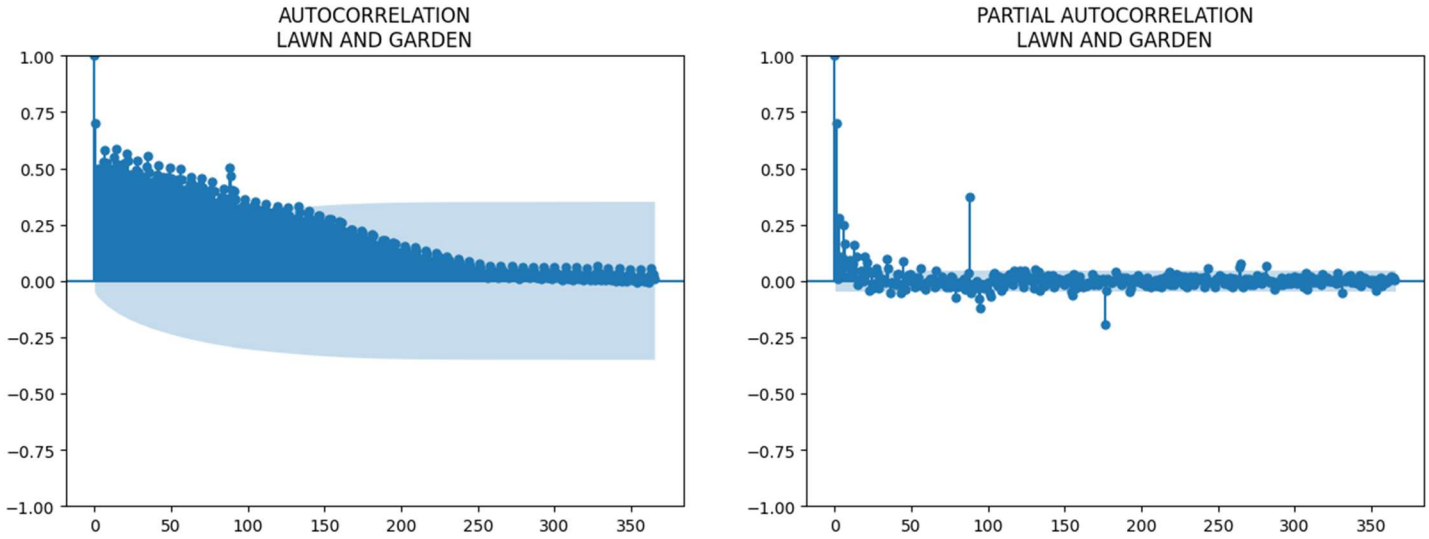


Figure 38 ACF & PACF of Product : Lawn and garden

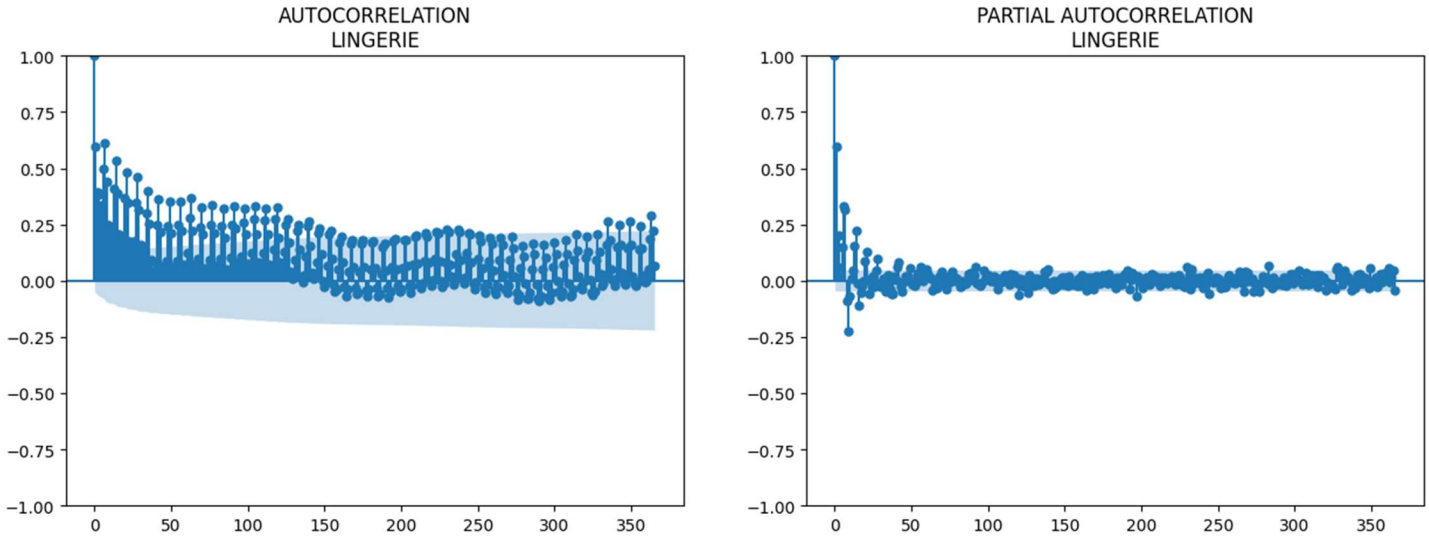


Figure 39 ACF & PACF of Product : Lingerie

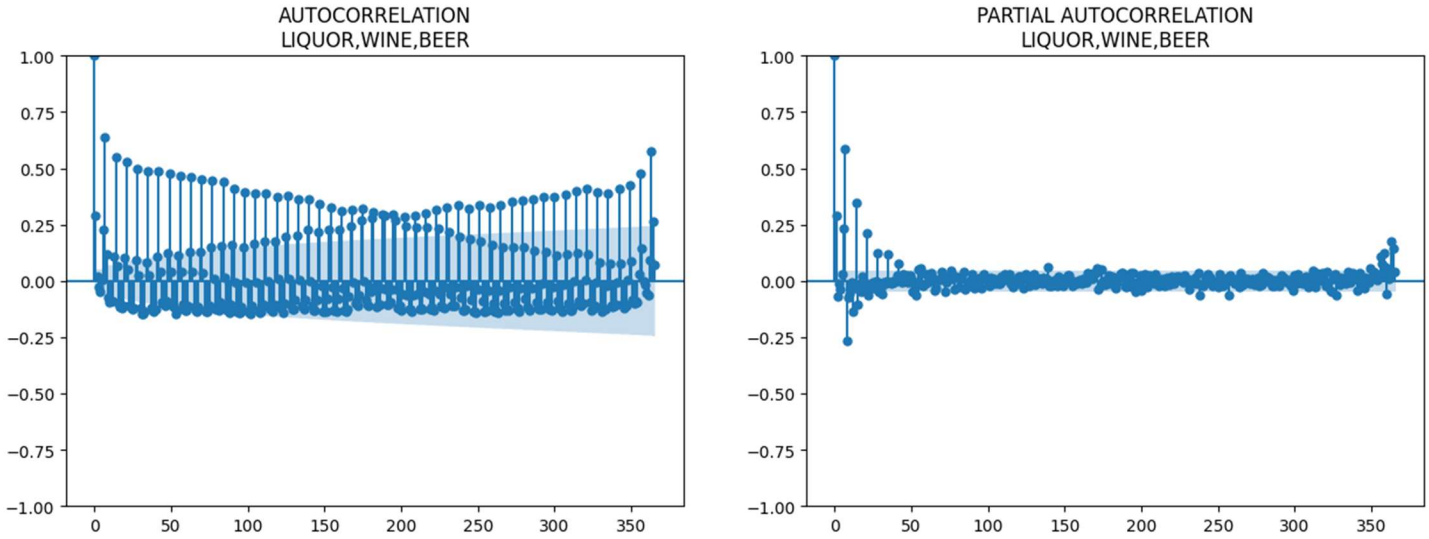


Figure 40 ACF & PACF of Product : Liquor,Wine,Beer

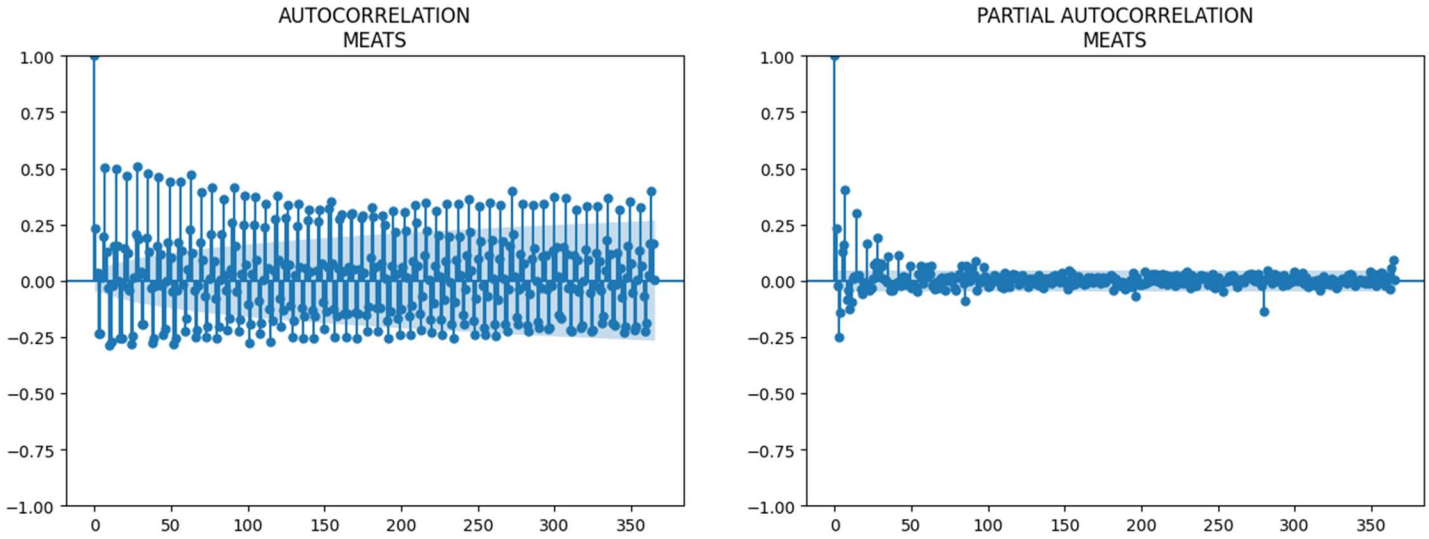


Figure 41 ACF & PACF of Product : Meats

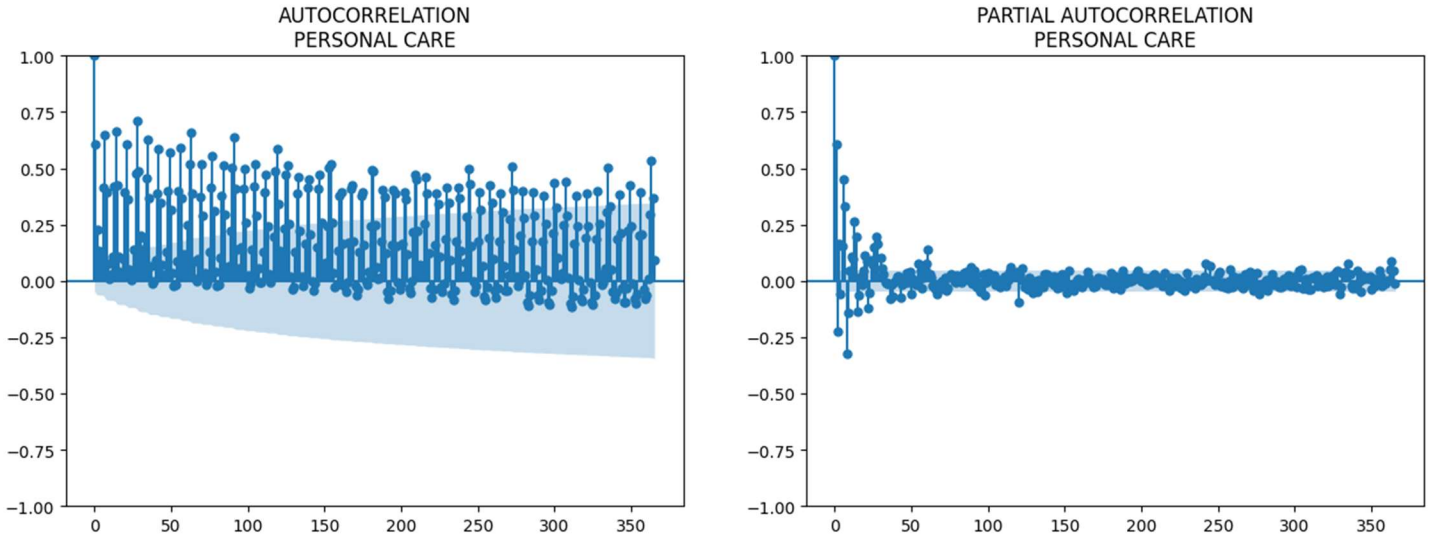


Figure 42 ACF & PACF of Product : Personal Care

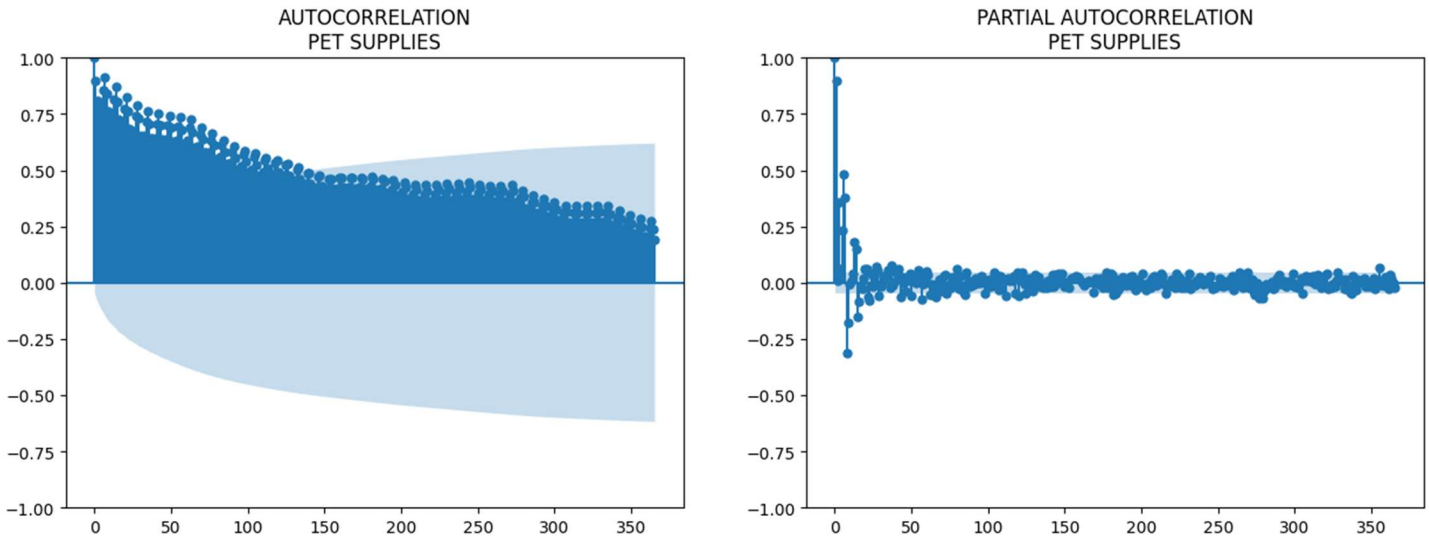


Figure 43 ACF & PACF of Product : Pet supplies

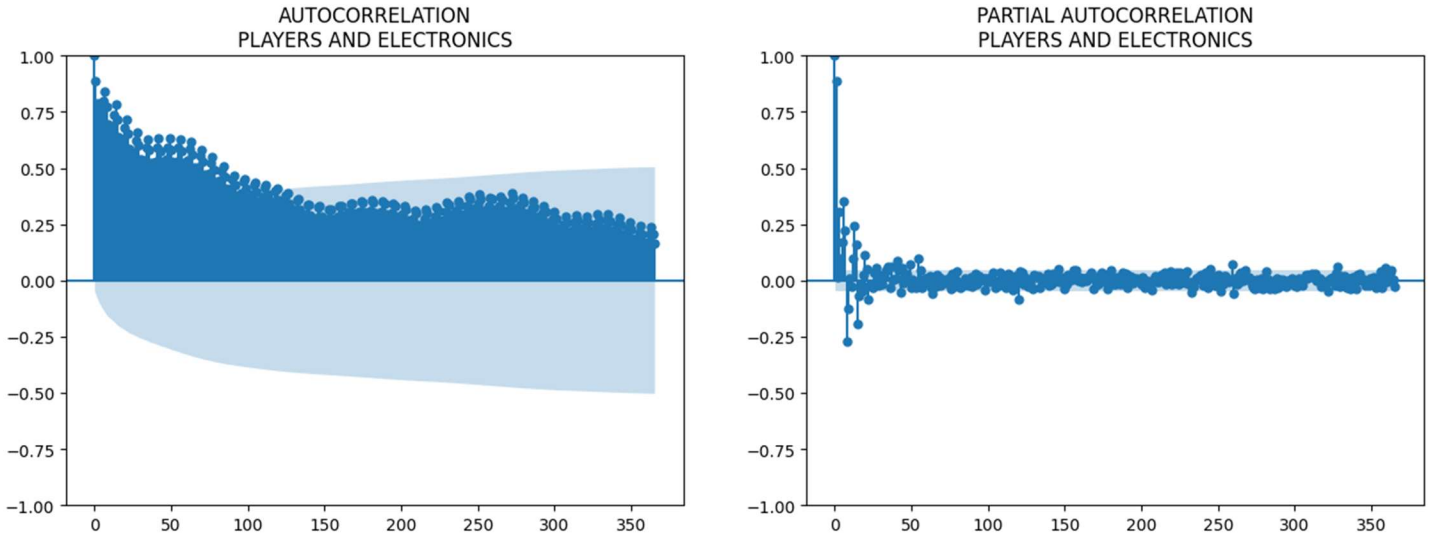


Figure 44 ACF & PACF of Product : Players and Electronics

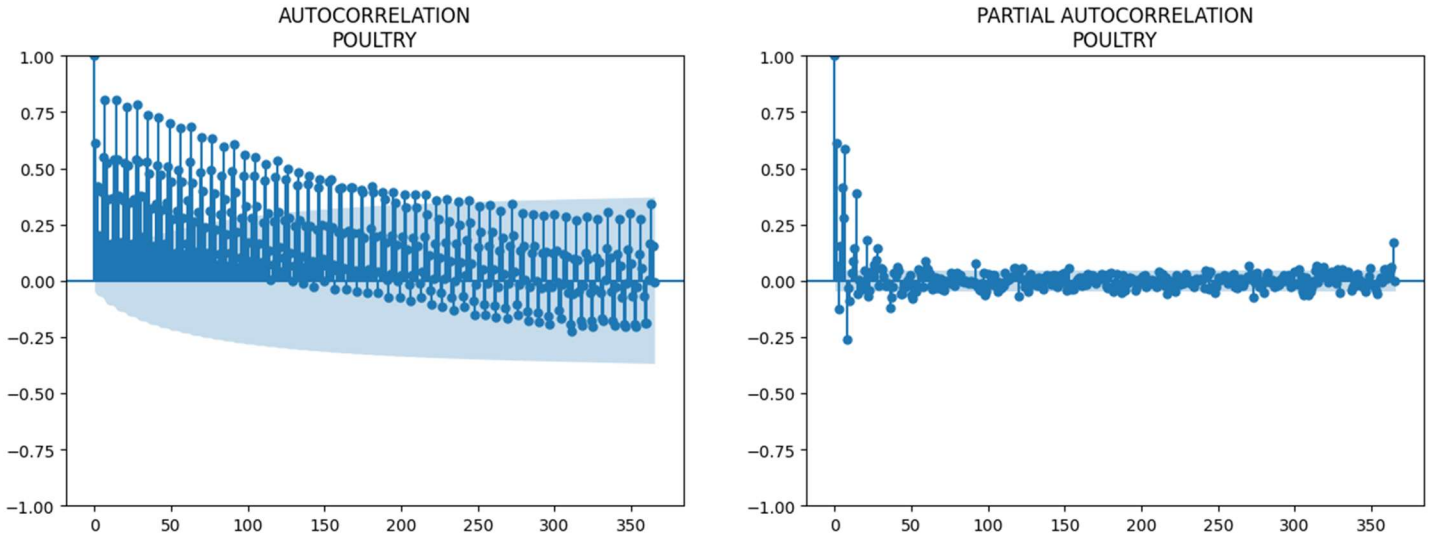


Figure 45 ACF & PACF of Product : Poulitry

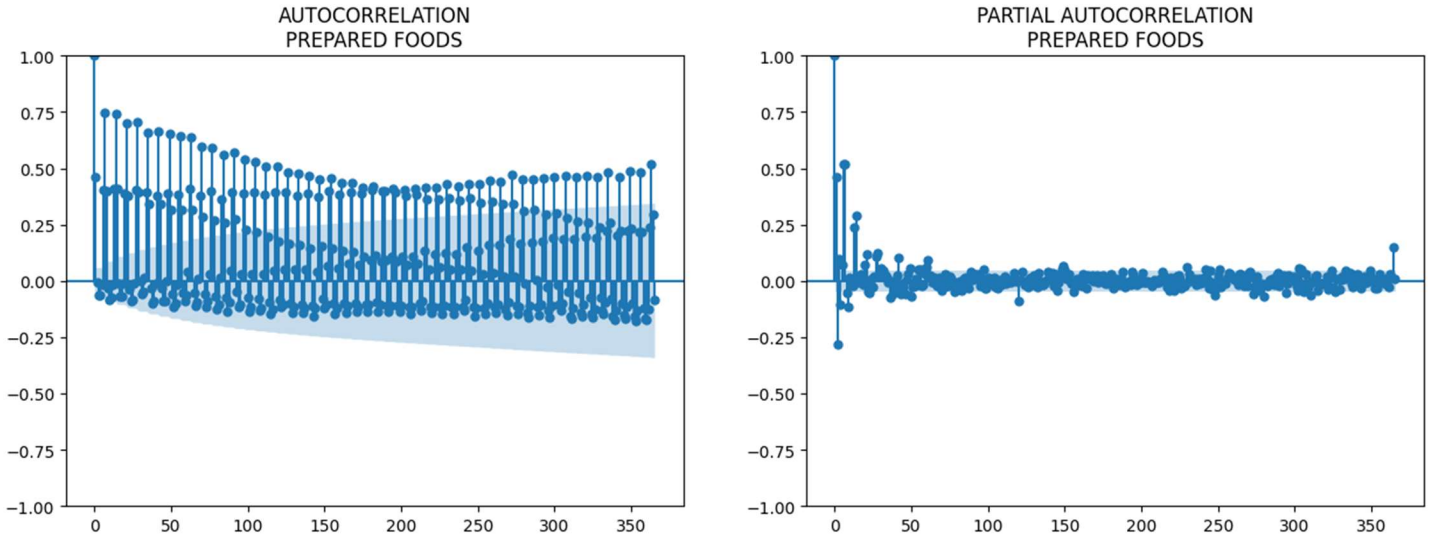


Figure 46 ACF & PACF of Product : Prepared foods

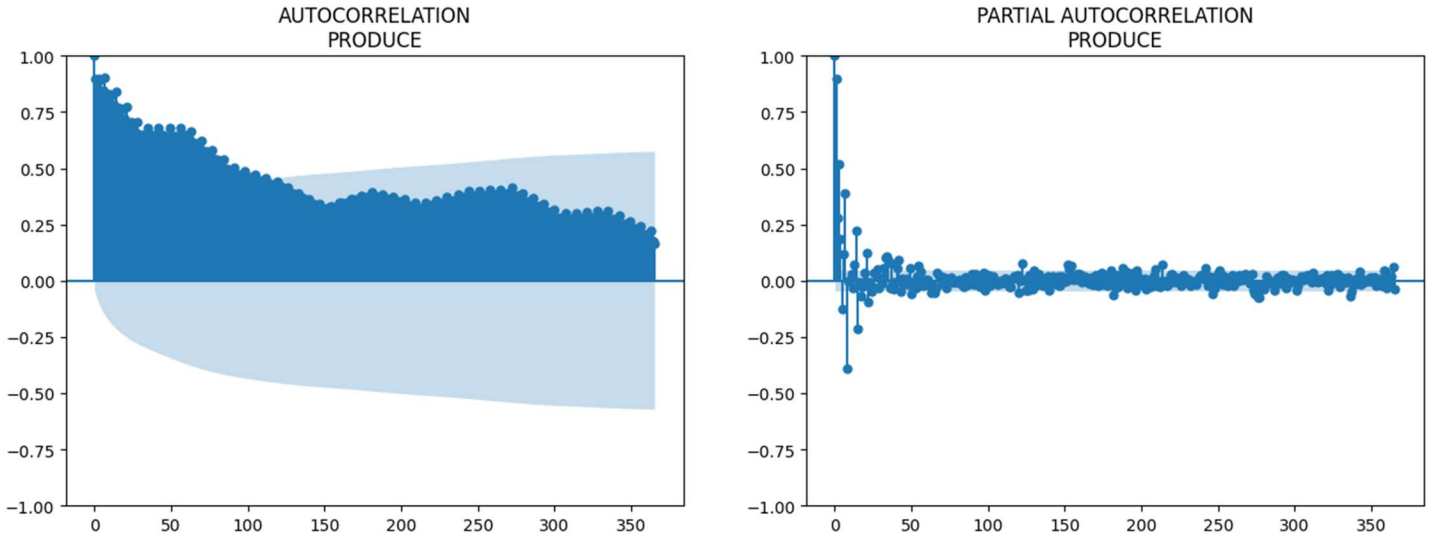


Figure 47 ACF & PACF of Product : Produce

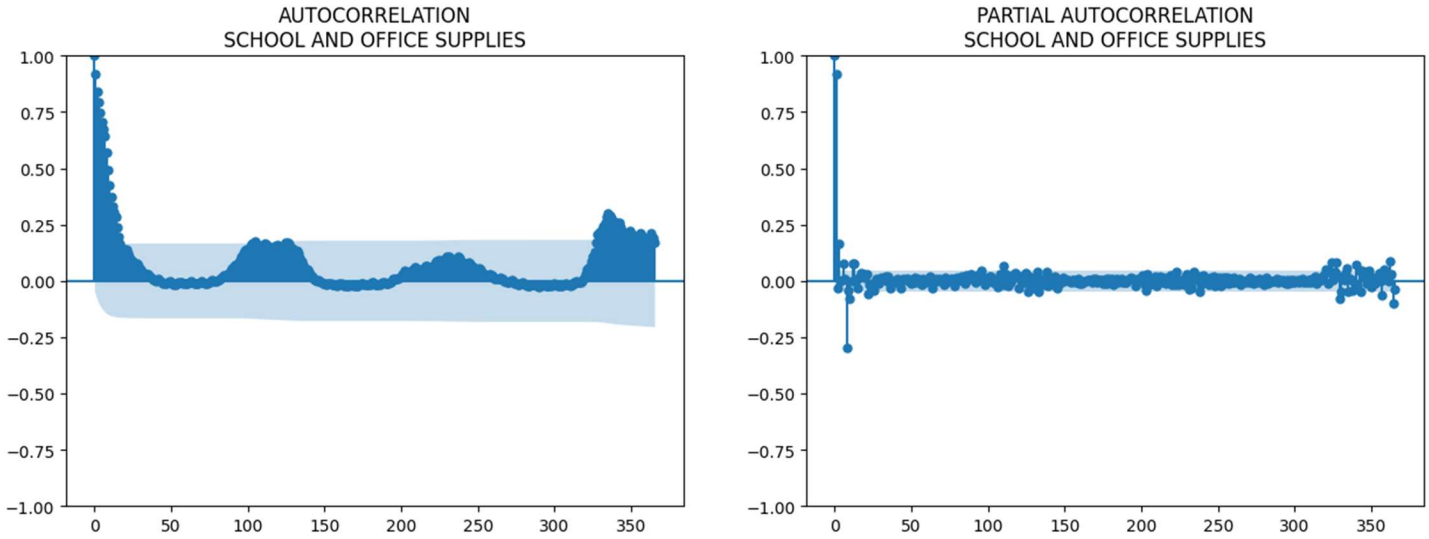


Figure 48 ACF & PACF of Product : School office supplies

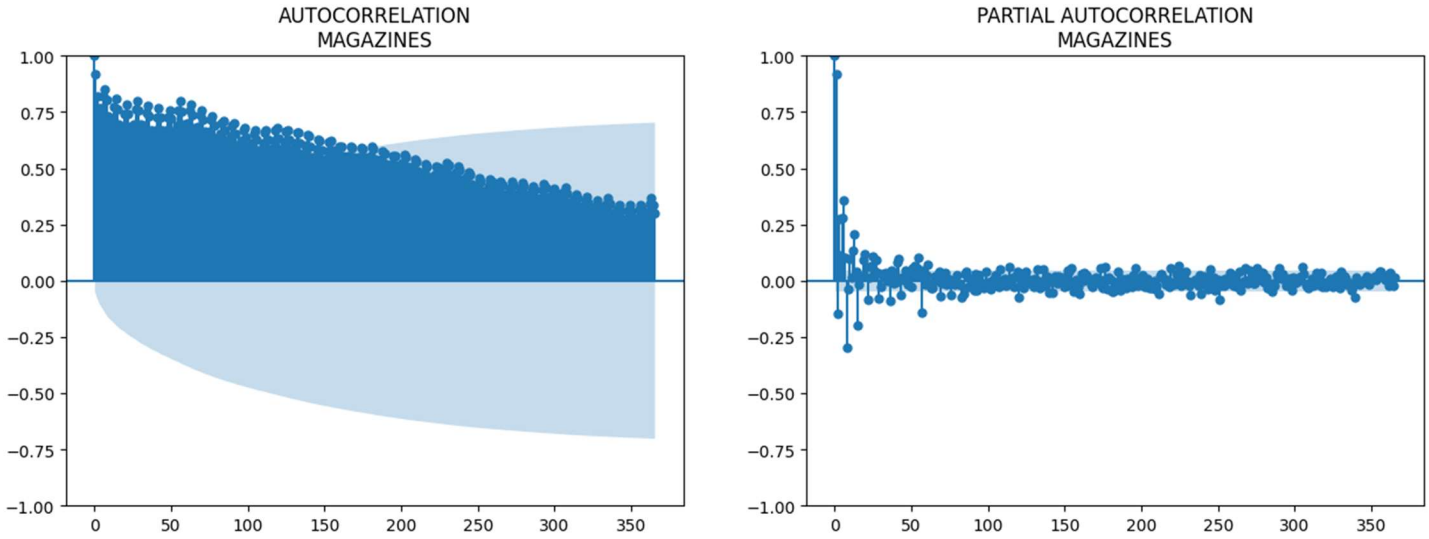


Figure 49 ACF & PACF of Product : Magazines

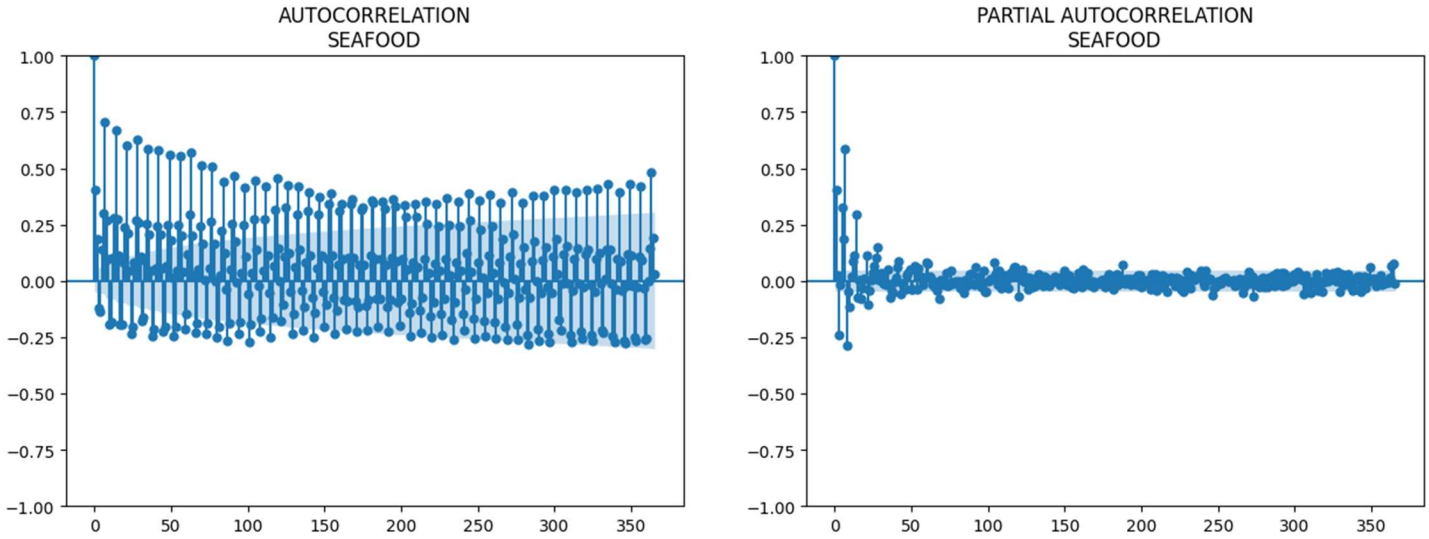


Figure 50 ACF & PACF of Product : Seafood

ACF: There is positive autocorrelation at lags 1, 2, 3, 4, 5, and 6. This shows that the sales of this product type tend to depend on the sales of previous days, with the cycle period of about 1 week.

PACF: There is positive autocorrelation at lags 1, 2, 3, 4, and 5. This also shows that the sales of this product type tend to depend on the sales of previous days, with the cycle about 1 week.

4.3 Trend Forecasting

We'll employ the Deterministic Process function from the stats models library. By using this function, we can steer clear of a few problematic situations when time series and linear regression go wrong. Polynomial order is indicated by the order argument, which ranges from 1 for linear to 3 for cubic.

Using linear regression for training and forecasting:

- Line y1 = df_grouped["mean"] assigns the column col average value to y1 (the target value).

- A linear regression model without constants is produced by the line model = LinearRegression(fit_intercept=False) (already in DeterministicProcess).
- Using data from X1 and the goal value y1, the line model.fit(X1, y1) trains the model.
- The target value for each time point in the sample is predicted by the line y1_pred = pd.Series(model.predict(X1), index=X1.index).

The goal value for out-of-sample prediction steps is predicted by the line y2_fore = pd.Series(model.predict(X2), index=X2.index).

This function forecasts future target values by analyzing the average trend of data over time using linear regression and determination. Use this function to make two graphs:

- Graph 1: Displays the forecast trend and average value for the given data range.
- Graph2: Displays the mean value

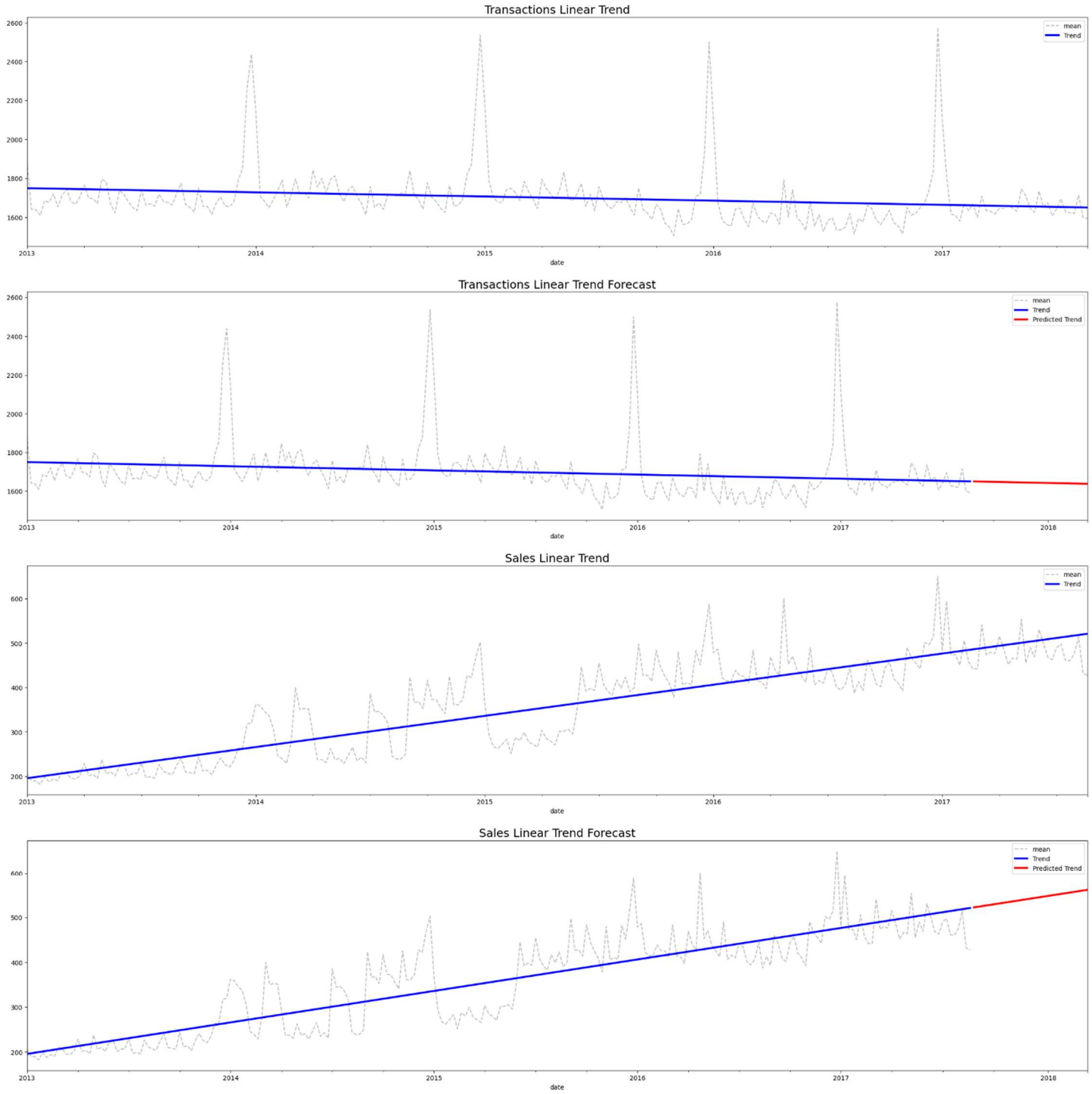


Figure 51 Sale and Transaction Trend Forecast

The data's average value over time is depicted in the image. The actual mean is shown by the dashed blue line, the model prediction trend is shown by the solid green line, and the predicted value for out-of-sample steps is shown by the red line.

It is evident that with time, the actual average number tends to rise. A trend toward growth is also seen in the forecast model, albeit at a somewhat slower pace. This implies that in the future, the real value can exceed the prediction. It is possible to forecast from the graph that the

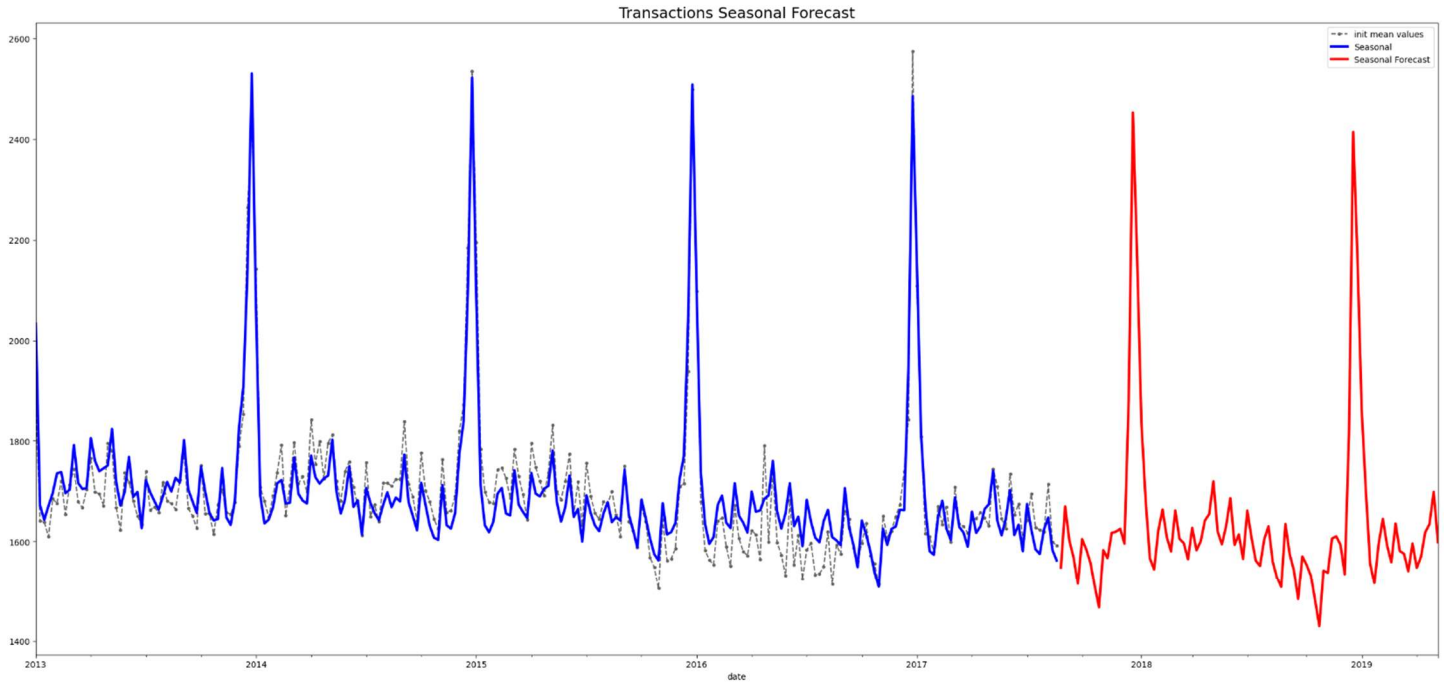
average value of the data will rise steadily in the coming years. It is possible that the actual number will exceed the projection by a small amount.

4.4 Seasonality Forecasting

Make the annual cycle's Fourier component:
In order to model annual seasonality, fourier = CalendarFourier(freq="A", order=10) generates a CalendarFourier object with frequency freq="A" (annual) and order=10 sin/cos pairings.
An object is created using the expression dp = DeterministicProcess(index=df_grouped['date'],...).

Forecast data for the following 90 steps (out of sample) are produced by the line X1_fore = dp.out_of_sample(steps=90).
The DeterministicProcess function forecasts future target values while accounting for annual seasonality by analyzing average trends and seasonality of data over time using deterministic processes and linear regression.

Average value: The true average value of the data is shown by the dotted blue line. The average value is shown to have a tendency to rise with time.
- Trend: The model's predicted trend is shown by the bold blue line. It is evident that, like the actual mean value, the anticipated trend likewise tends to climb steadily, albeit a little more slowly.
- Forecasting: The data's predicted value for out-of-sample steps is shown by the bold red line. It is evident that, while increasing somewhat more slowly than the anticipated trend, the forecast value also tends to rise steadily.



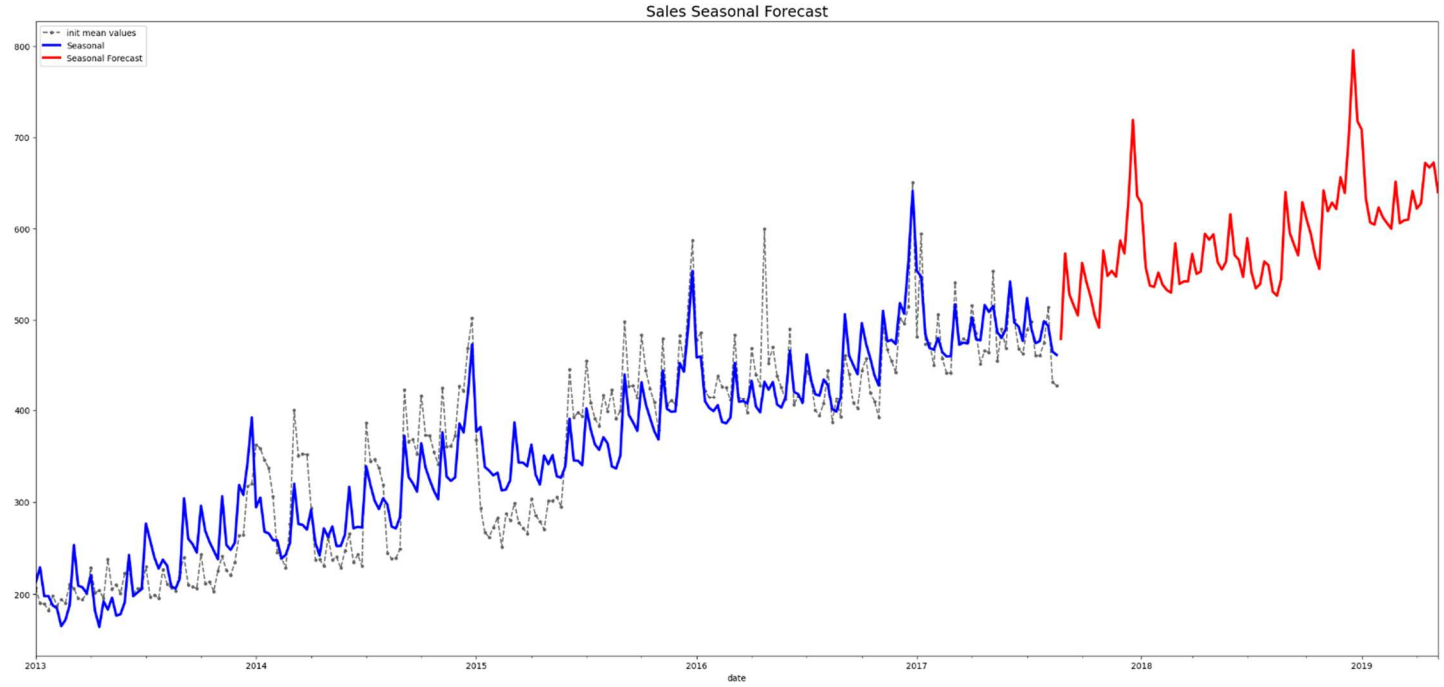


Figure 52 Sale and Transaction Seasonal Foreacast

4.5 Simple Moving Average

SMA is a basic forecast that we utilize to provide an overview of each distinct product. Considering over thirty products, based on previous sales figures. The following values will be taken into account for various adjustments; in essence, the form just changes.If you consider every component in isolation, it will be rather generic. Each image's pattern lines depict variations from one year to the next.

Certain items exhibit extreme volatility, which could be attributed to both the product's attributes and the state of the market's trading in those particular years. Additionally, some products, such as those in the beauty and bakery industries, can clearly display the effects of the seasons. Their cycles of increase and fall are consistent in their strength, and the business cycle of a product can be observed throughout time.

It is possible to see notable pauses, increases, and drops for particular product lines, such as beverage, dairy, and frozen food. This can be as a result of how the earthquake affected people's ability to go shopping. In the latter months of the year, product categories including groceries and home and kitchen might experience a sharp rise in sales.

By presenting SMA in this way as a straightforward projection, we hope to provide readers with the broadest possible picture of each product type's circumstances, enabling them to devise effective solutions.

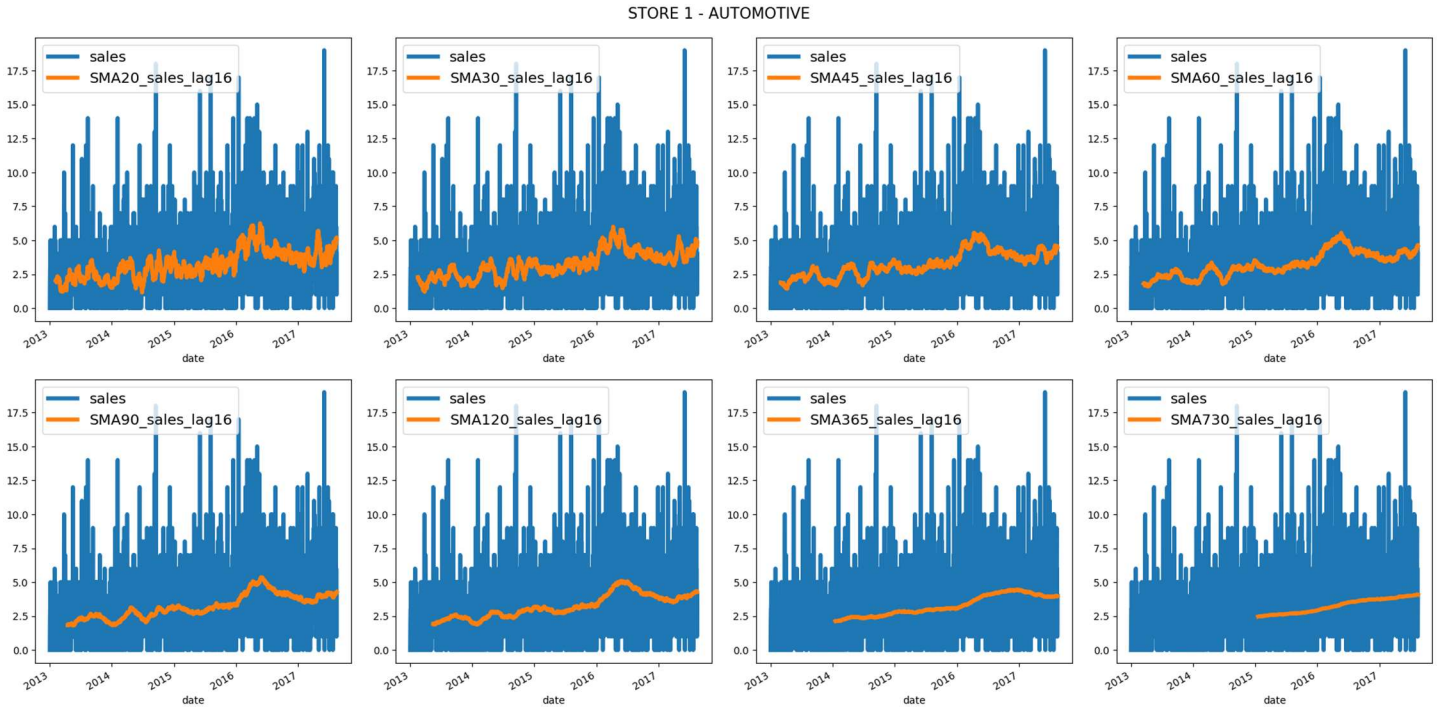
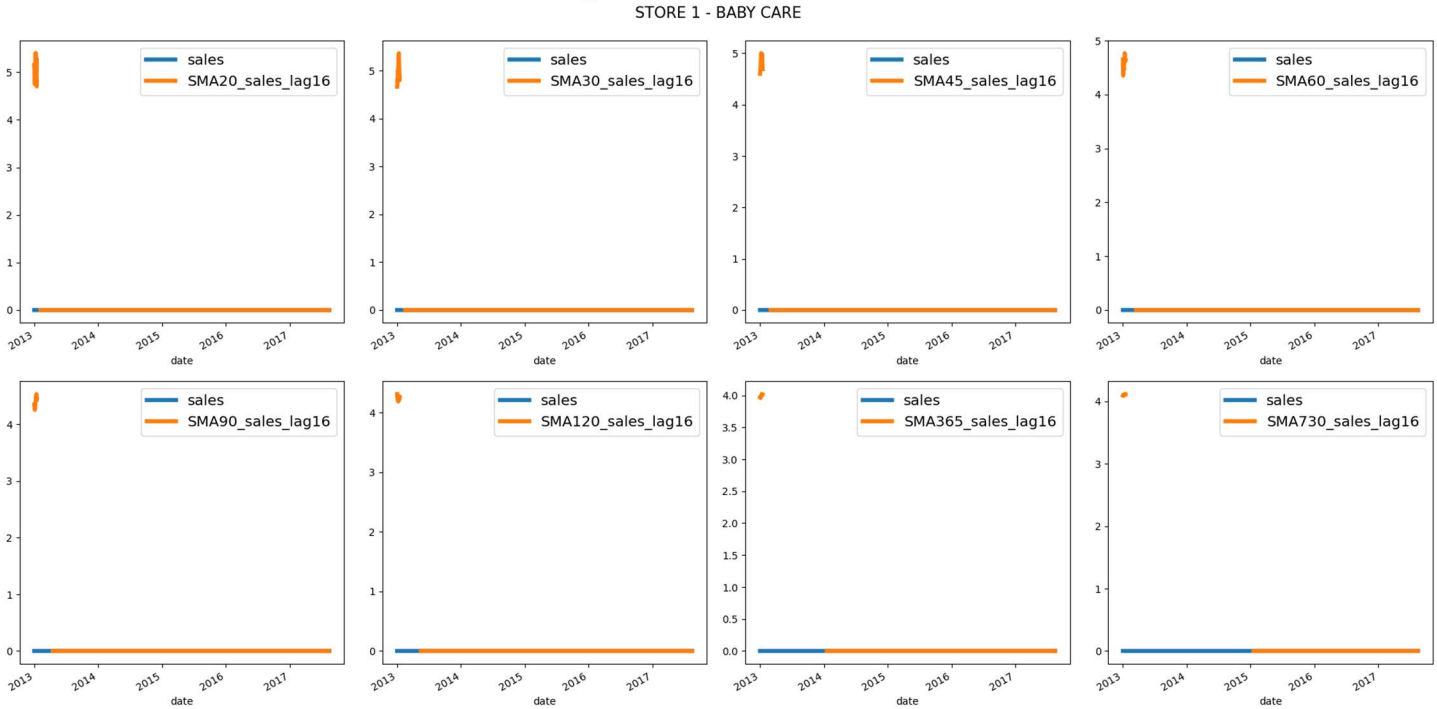
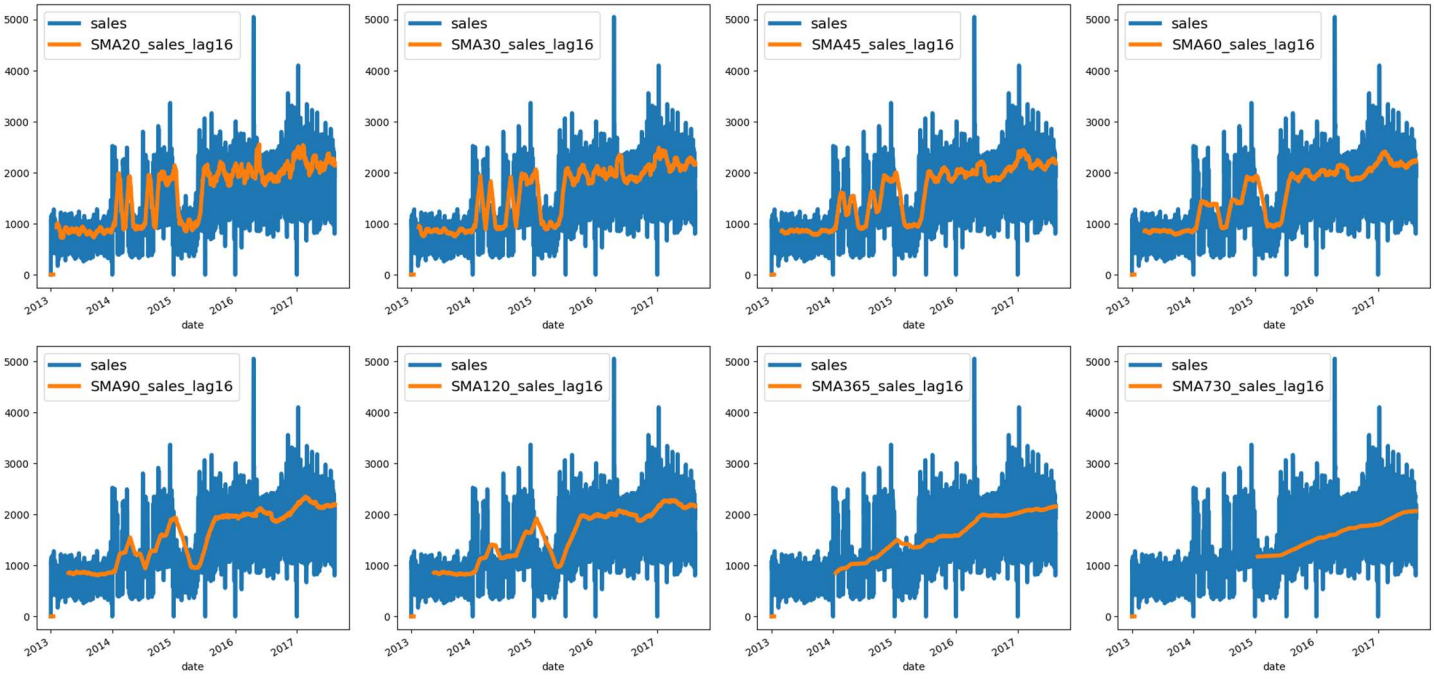


Figure 53 SMA of Store 1

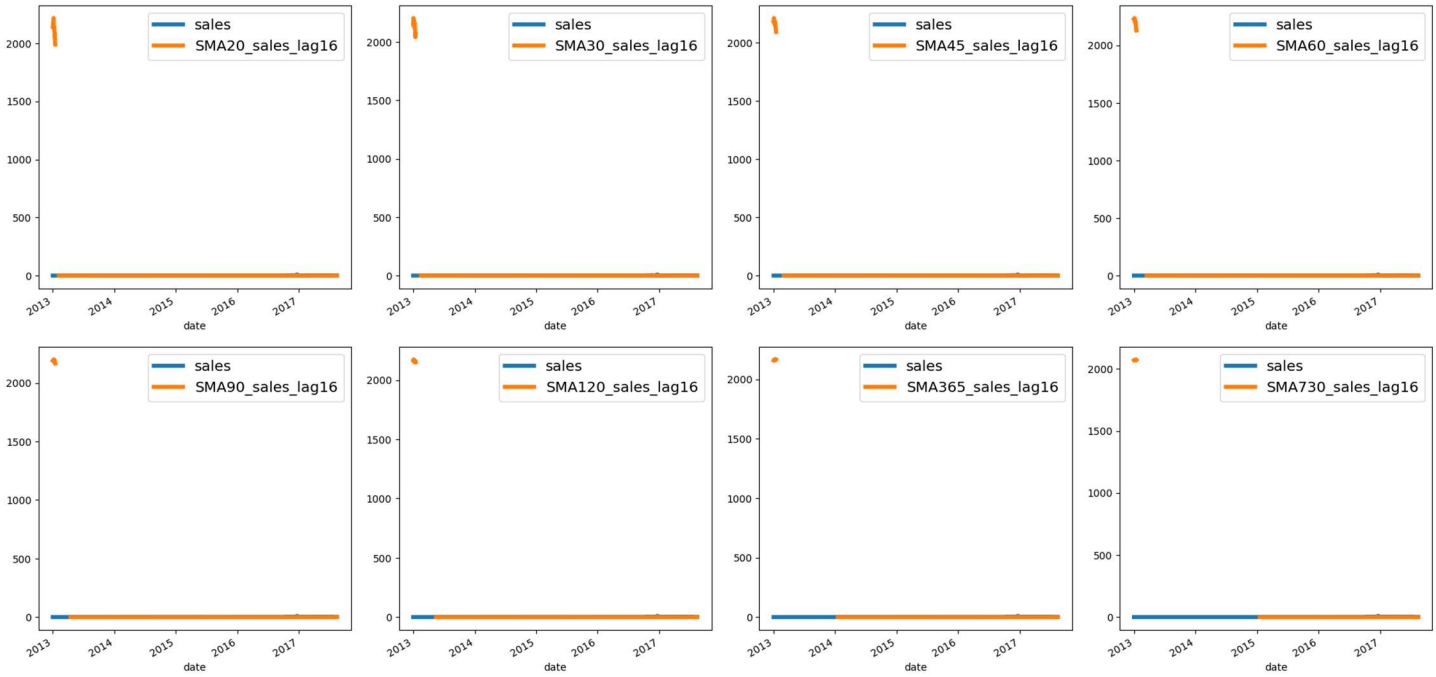


STORE 1 - BEAUTY

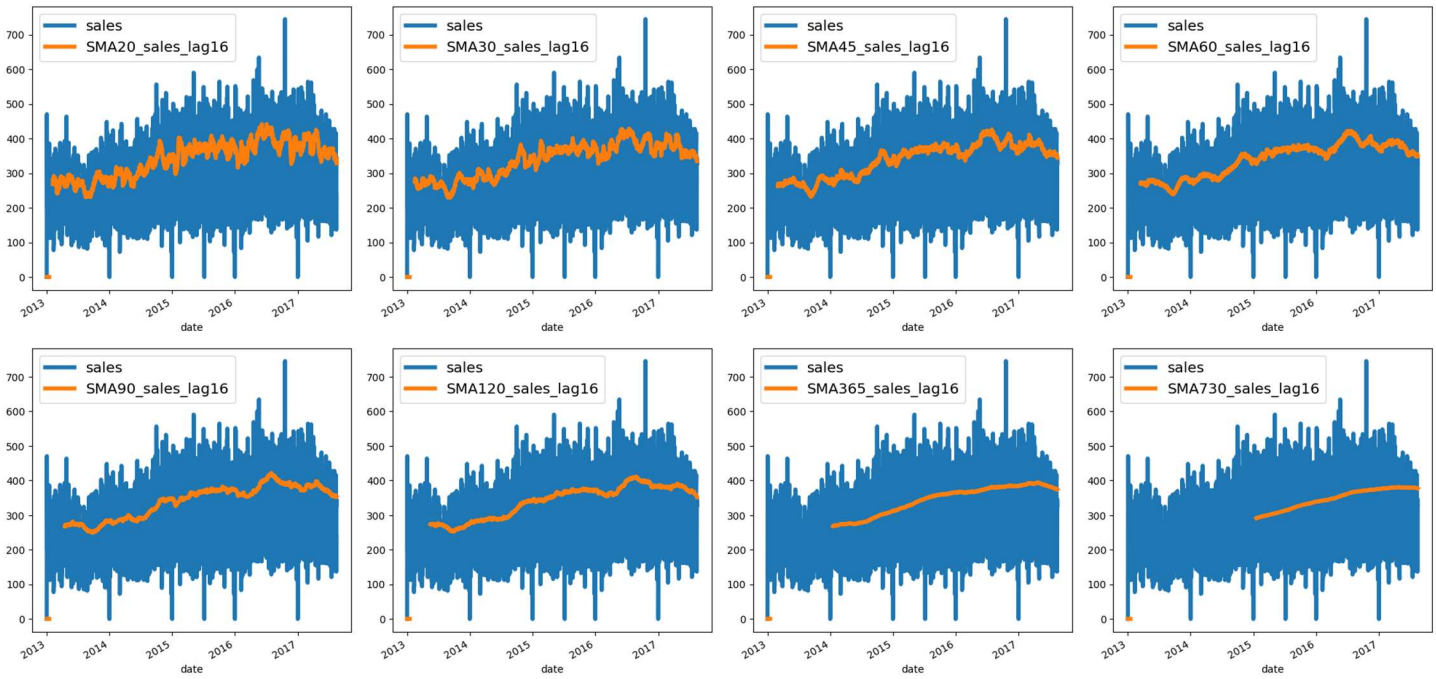
STORE 1 - BEVERAGES



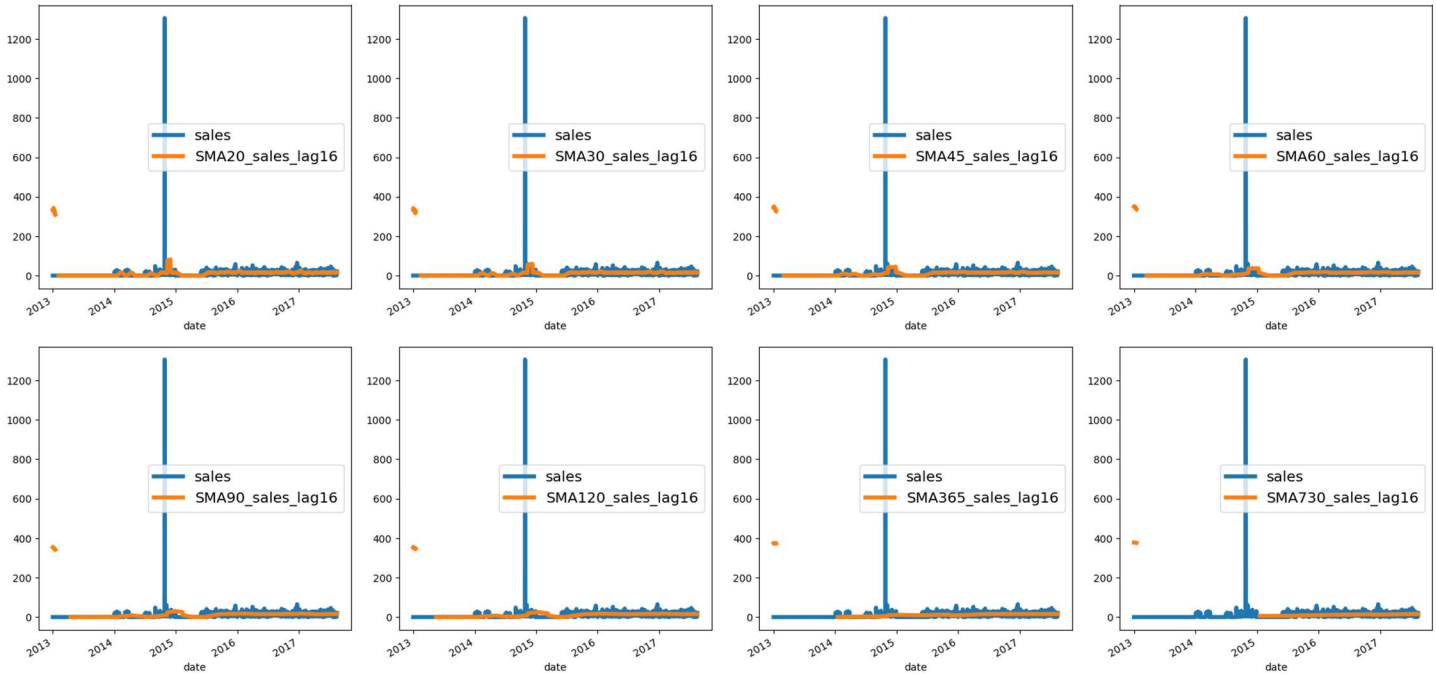
STORE 1 - BOOKS



STORE 1 - BREAD/BAKERY

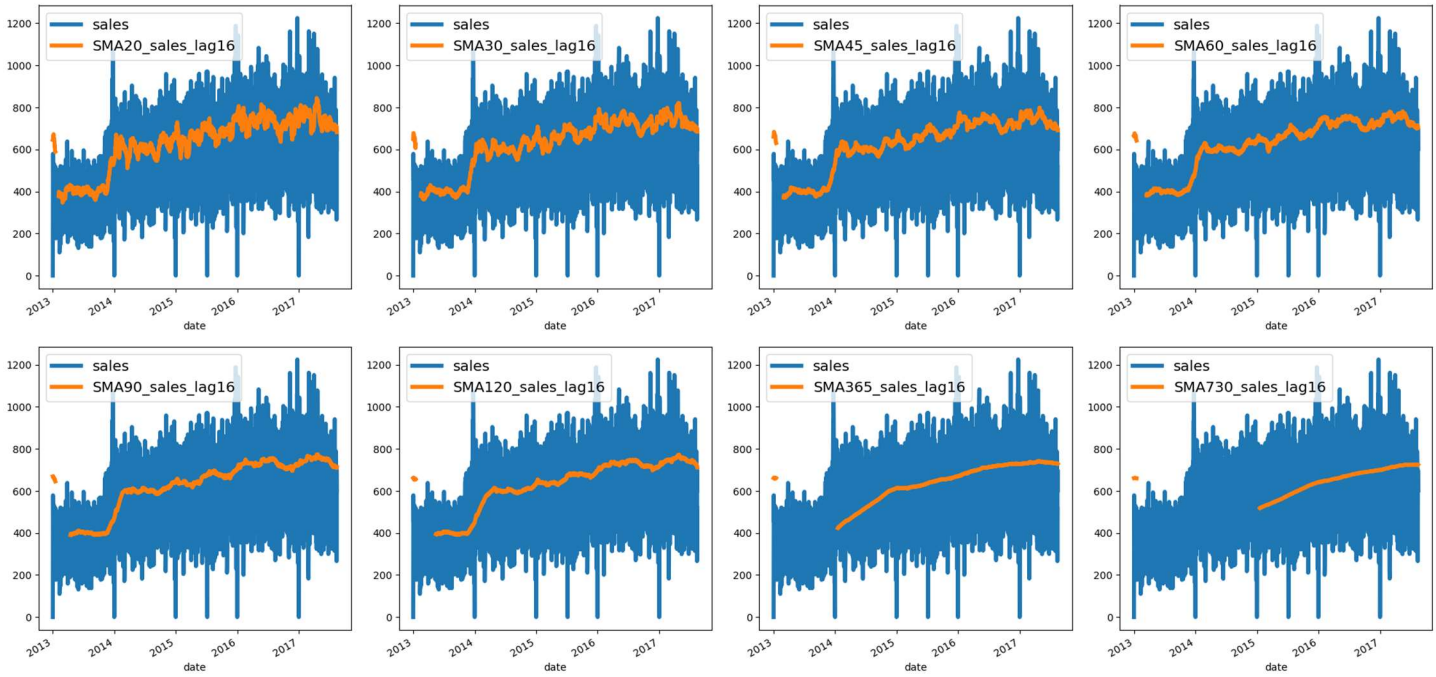


STORE 1 - CELEBRATION

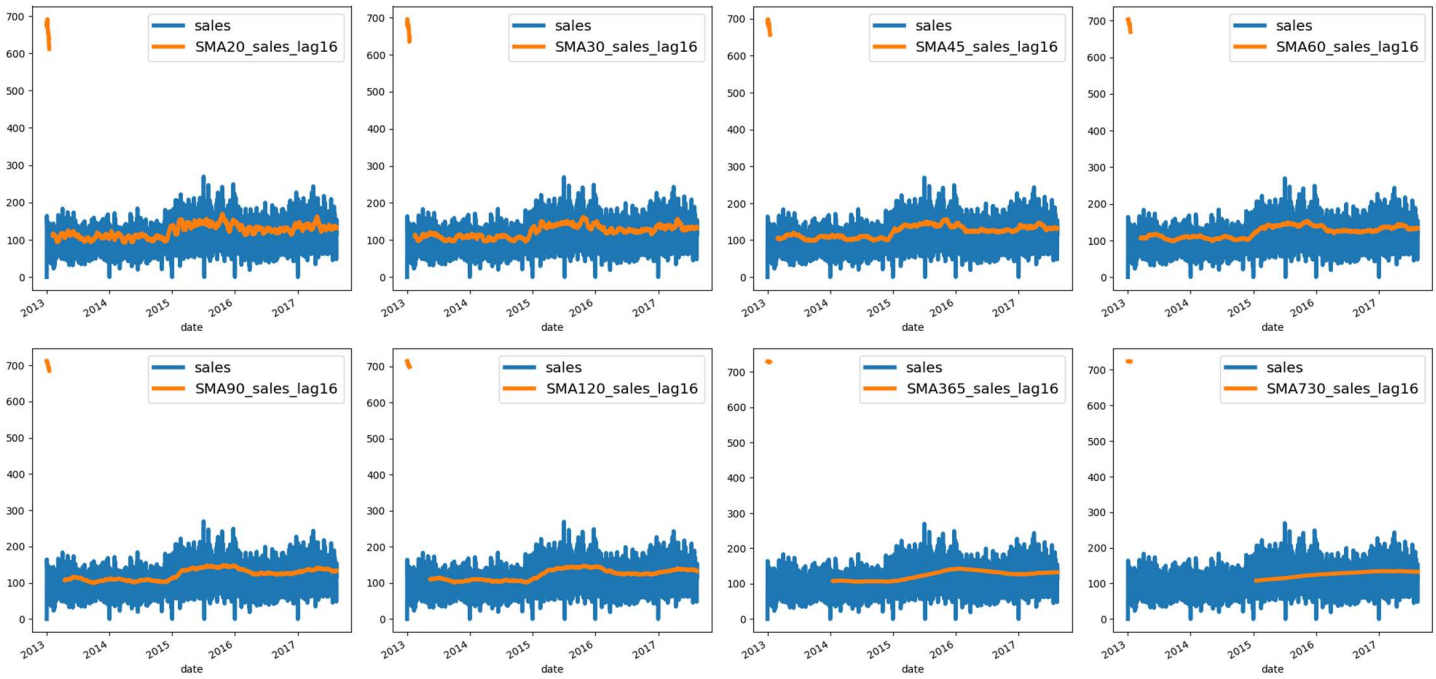


STORE 1 - CLEANING

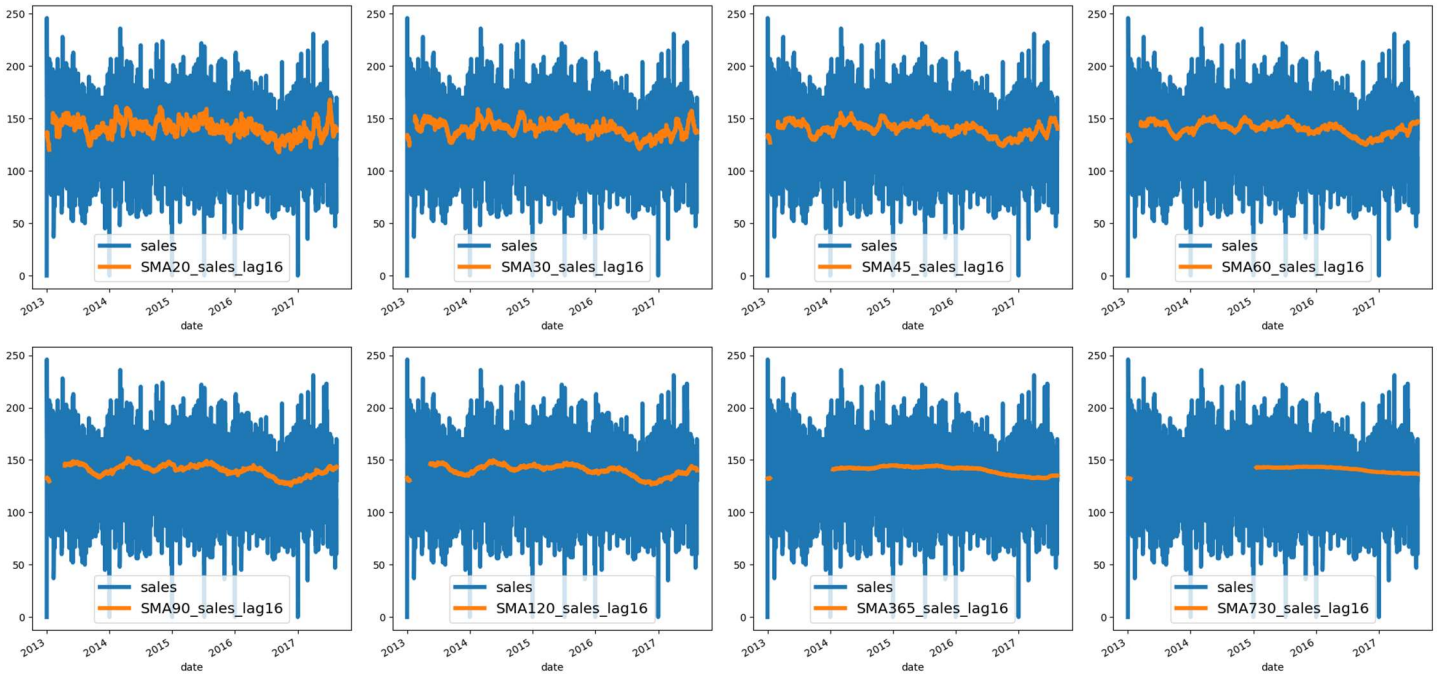
STORE 1 - DAIRY



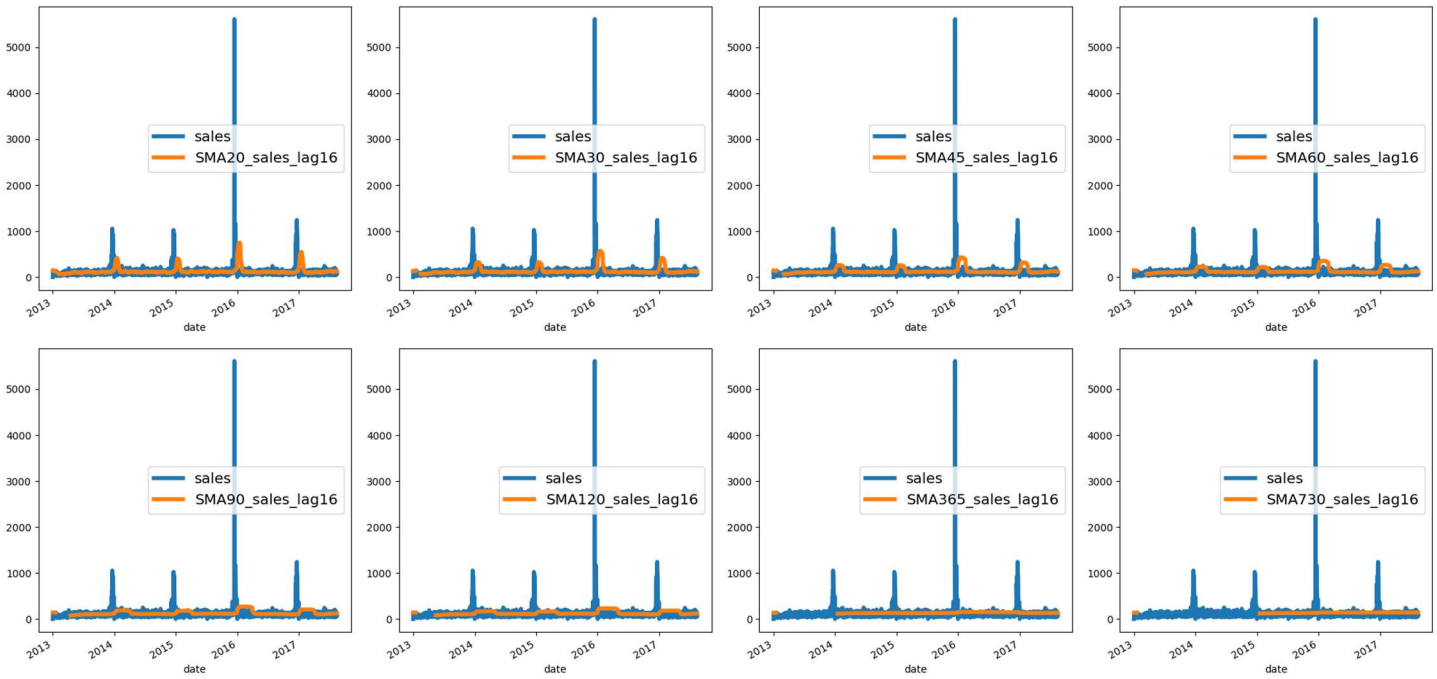
STORE 1 - DELI



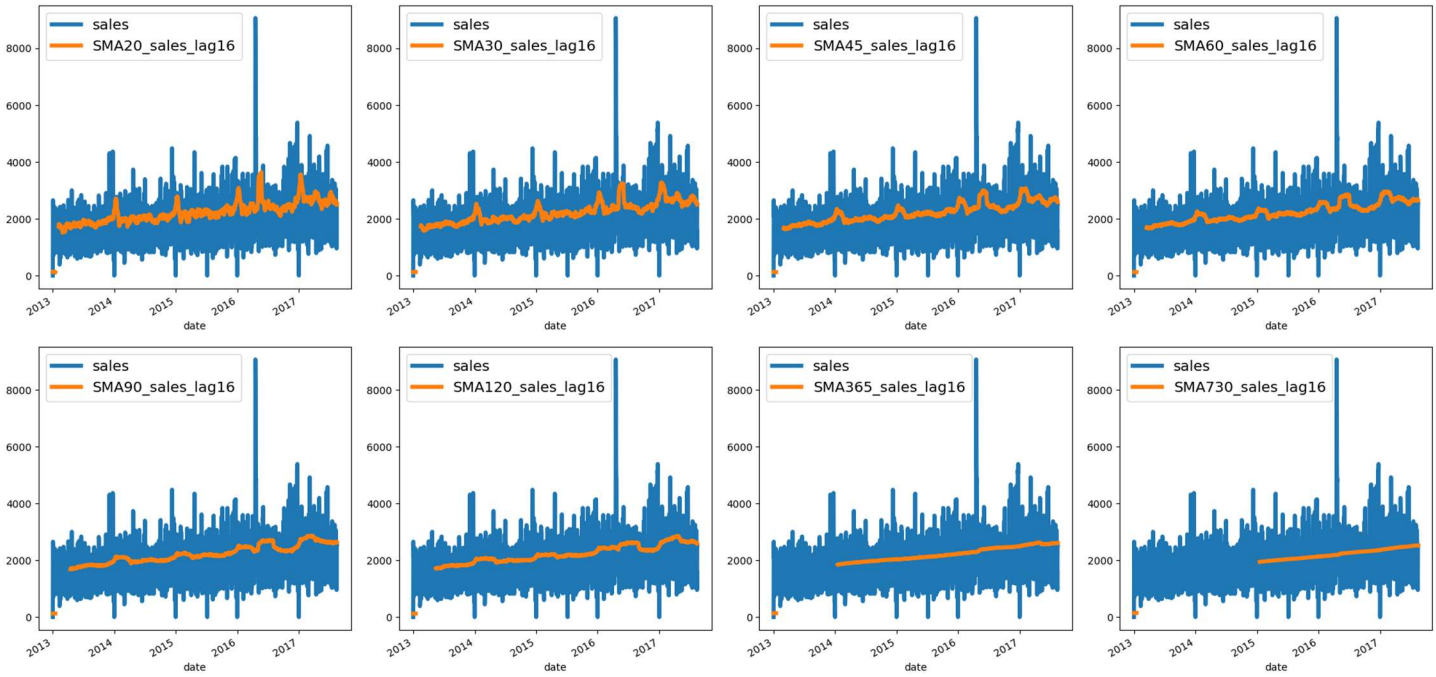
STORE 1 - EGGS



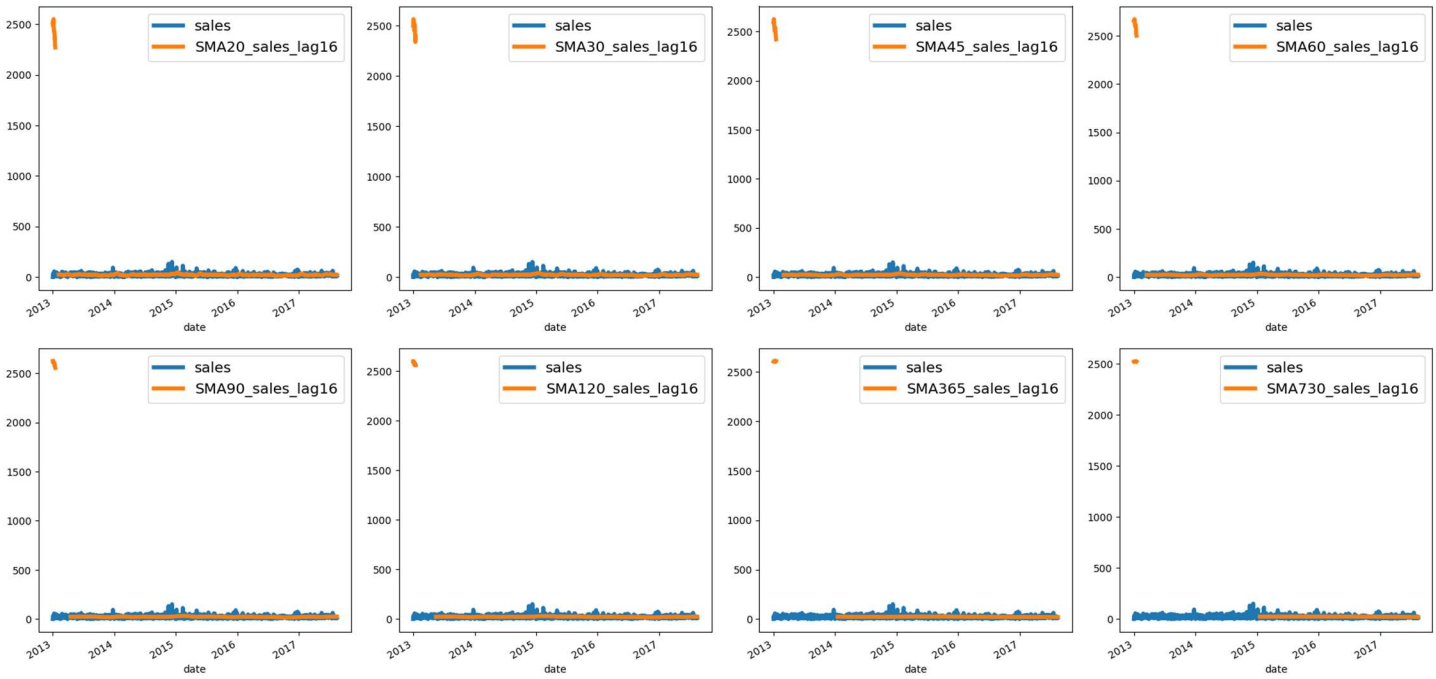
STORE 1 - FROZEN FOODS



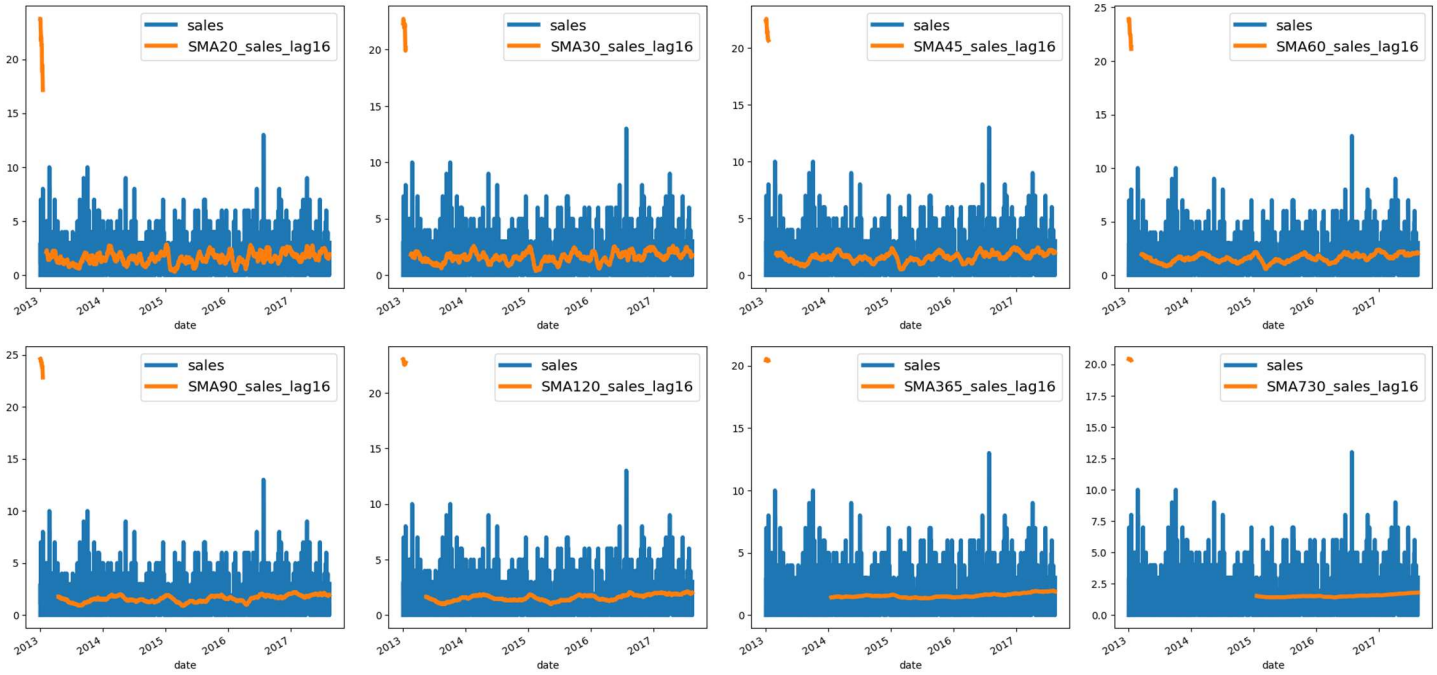
STORE 1 - GROCERY I



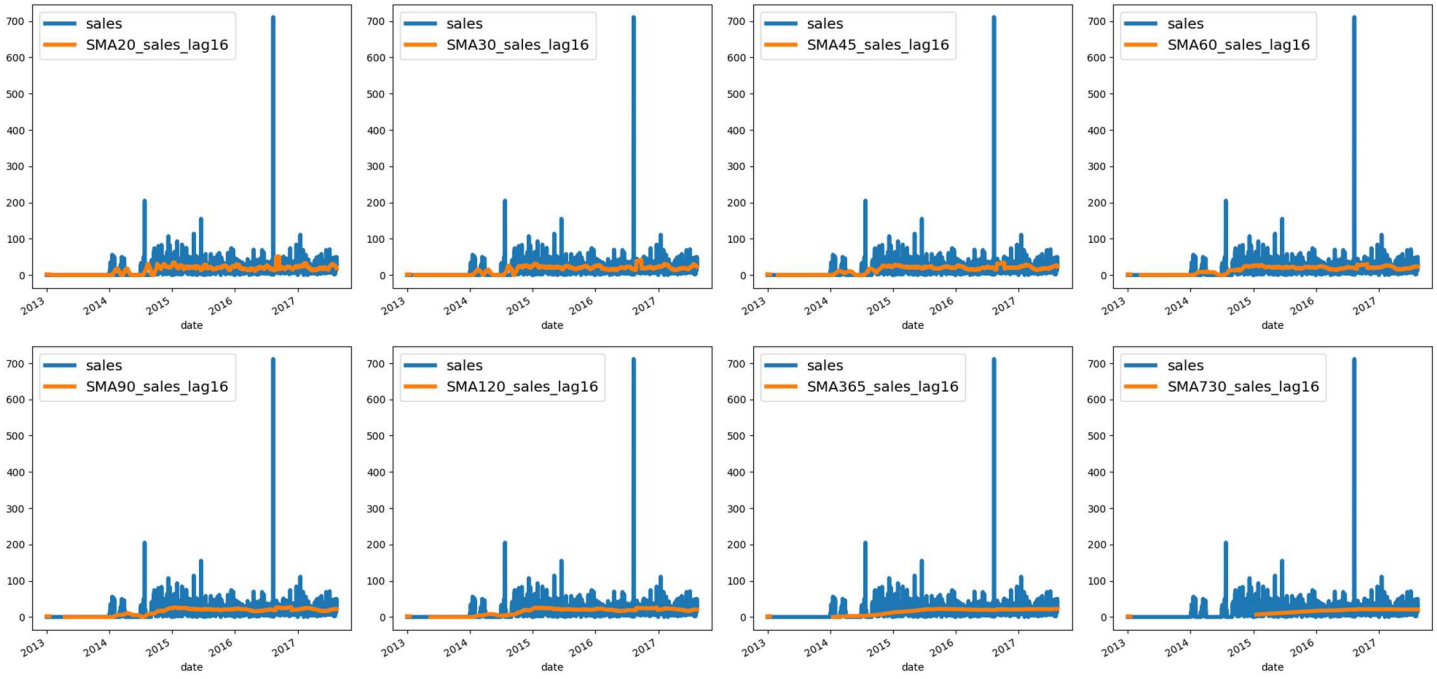
STORE 1 - GROCERY II



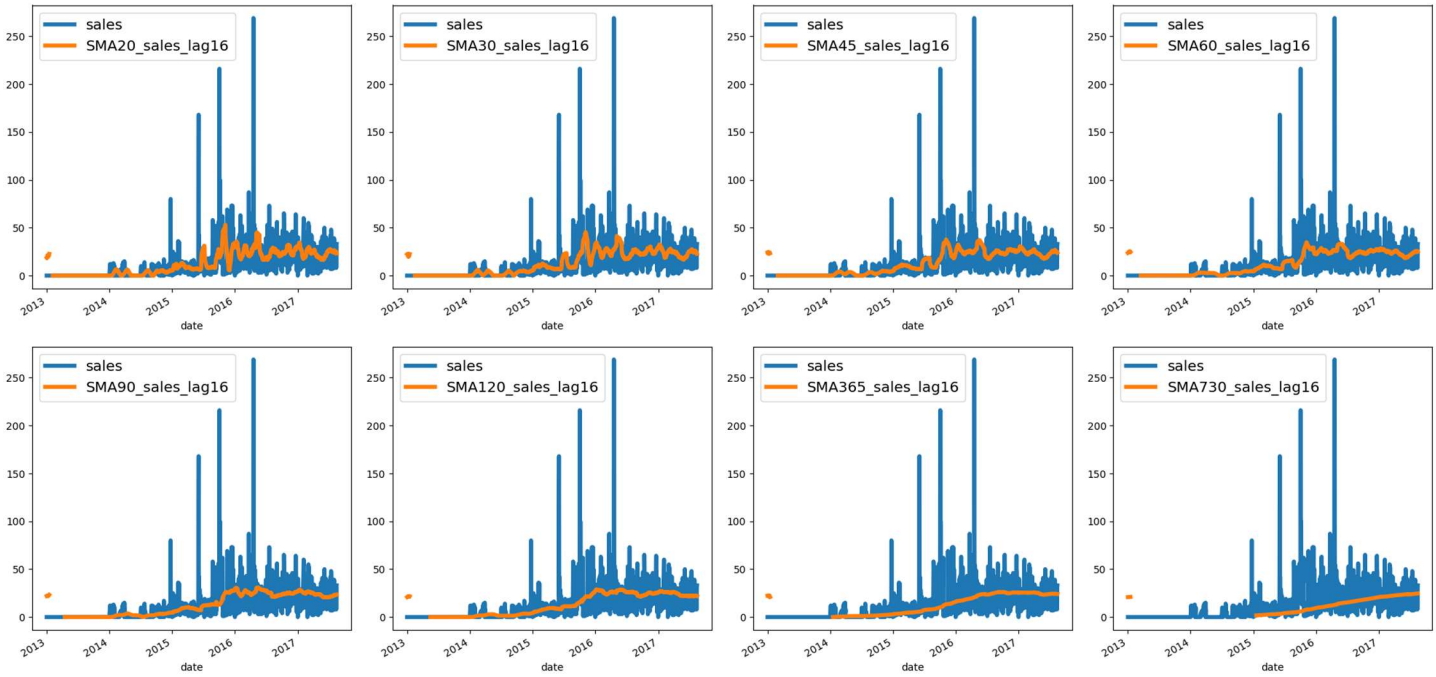
STORE 1 - HARDWARE



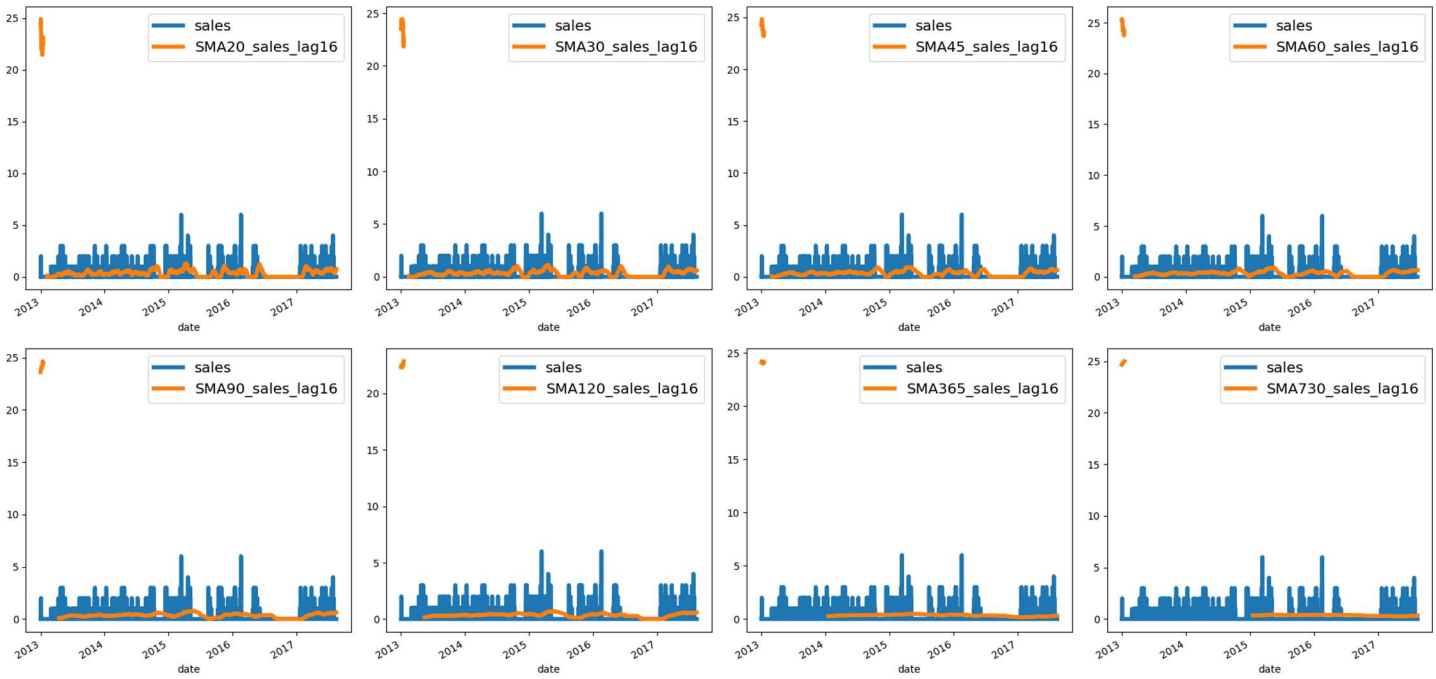
STORE 1 - HOME AND KITCHEN I



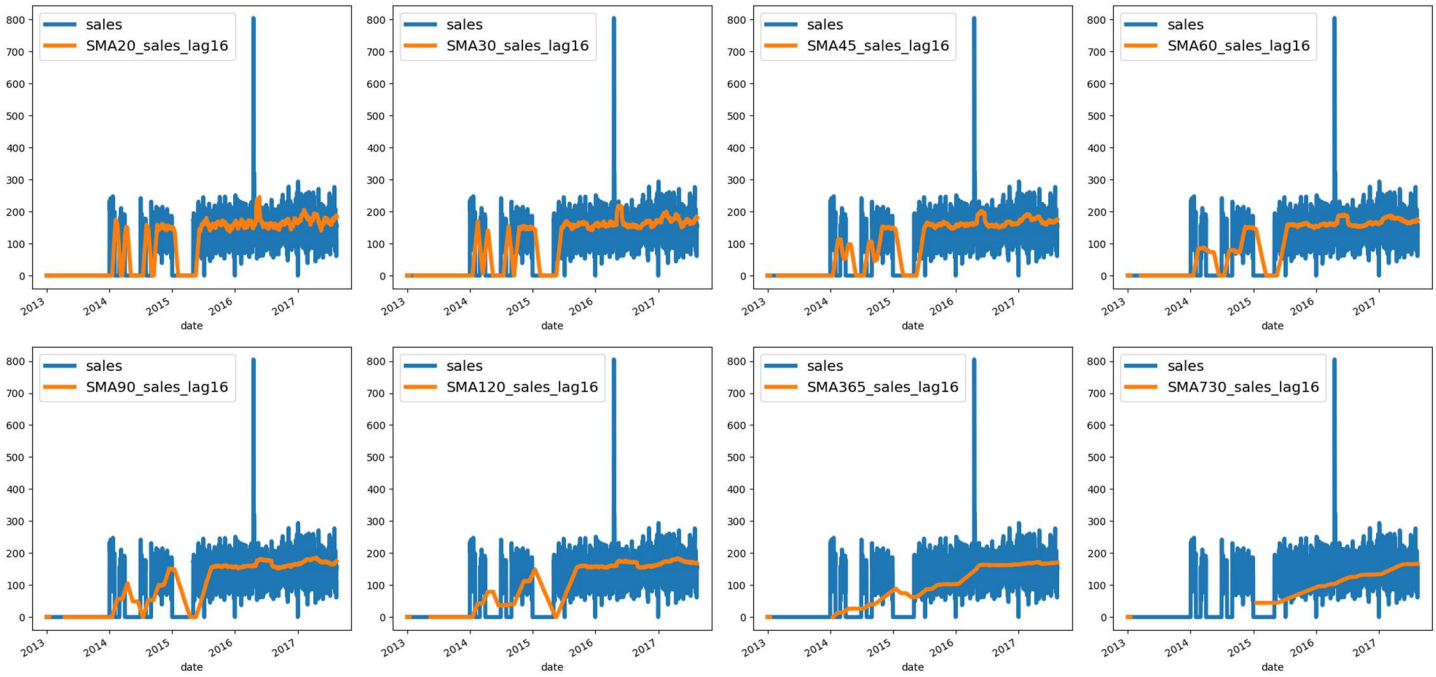
STORE 1 - HOME AND KITCHEN II



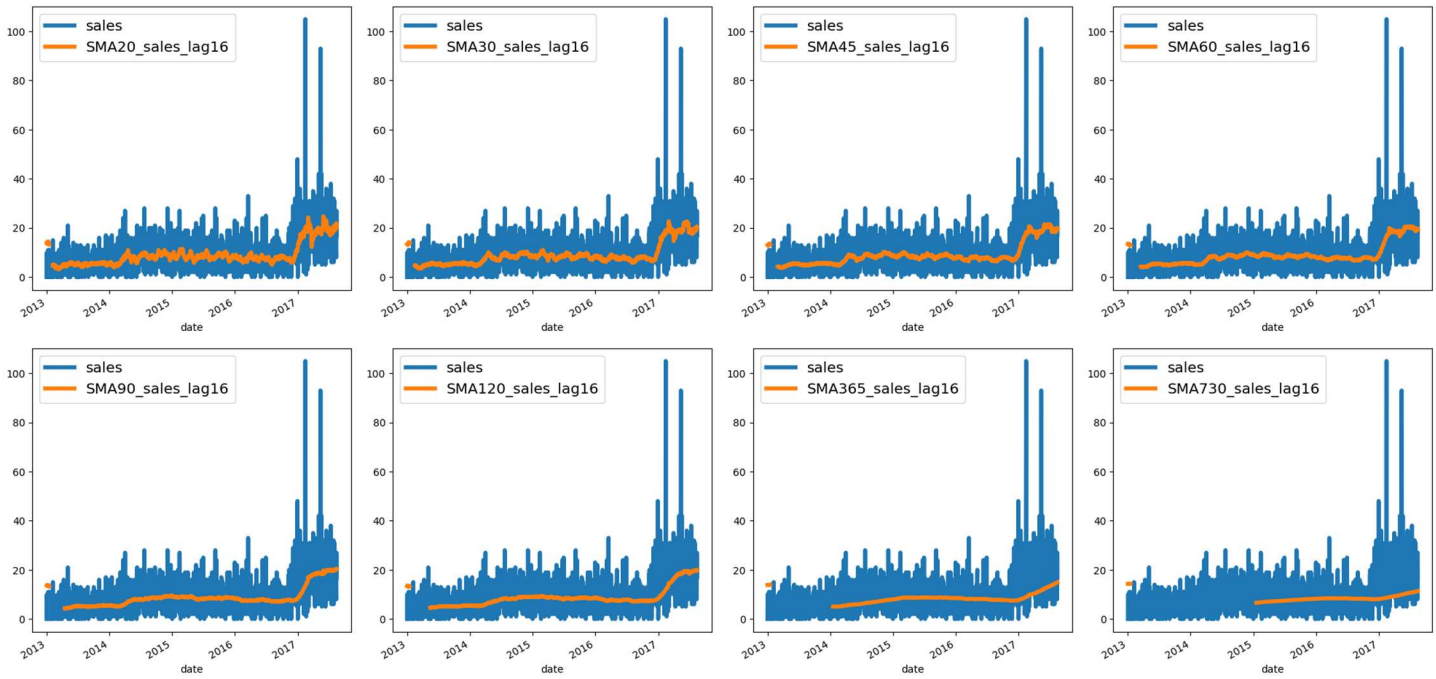
STORE 1 - HOME APPLIANCES



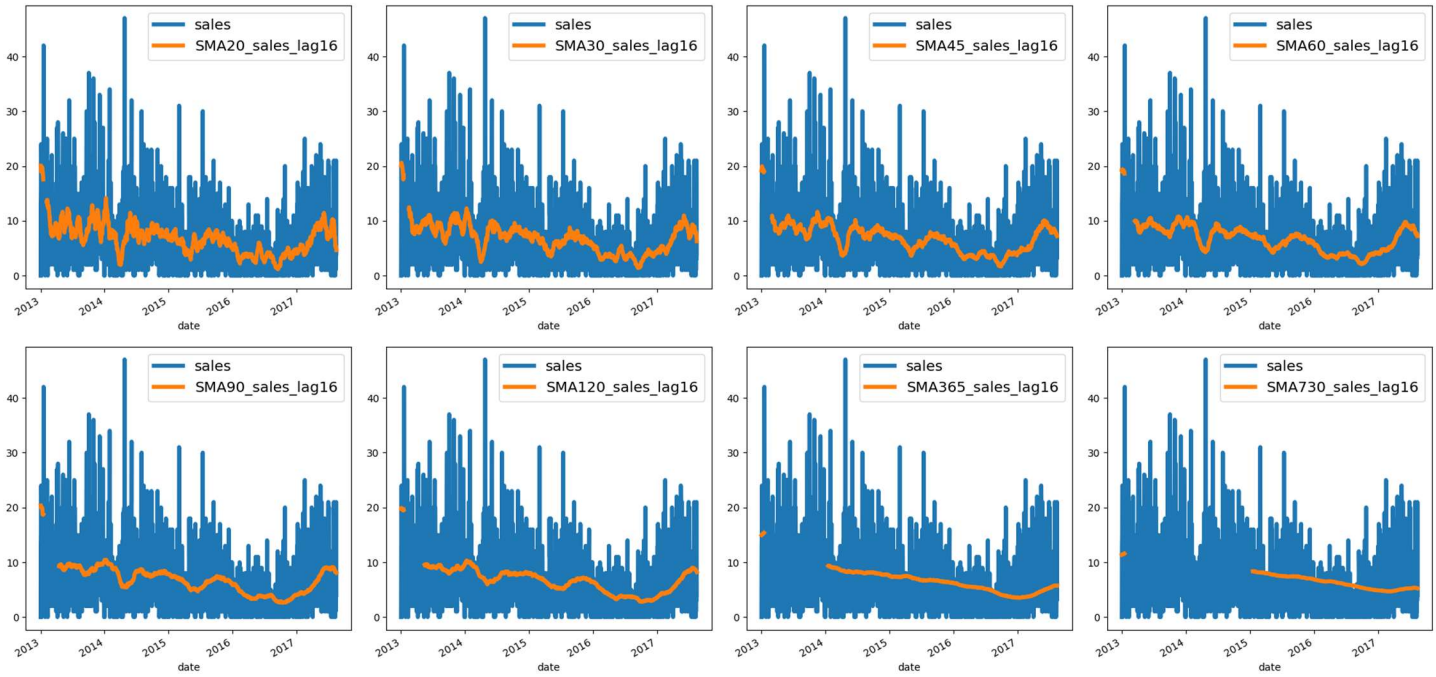
STORE 1 - HOME CARE



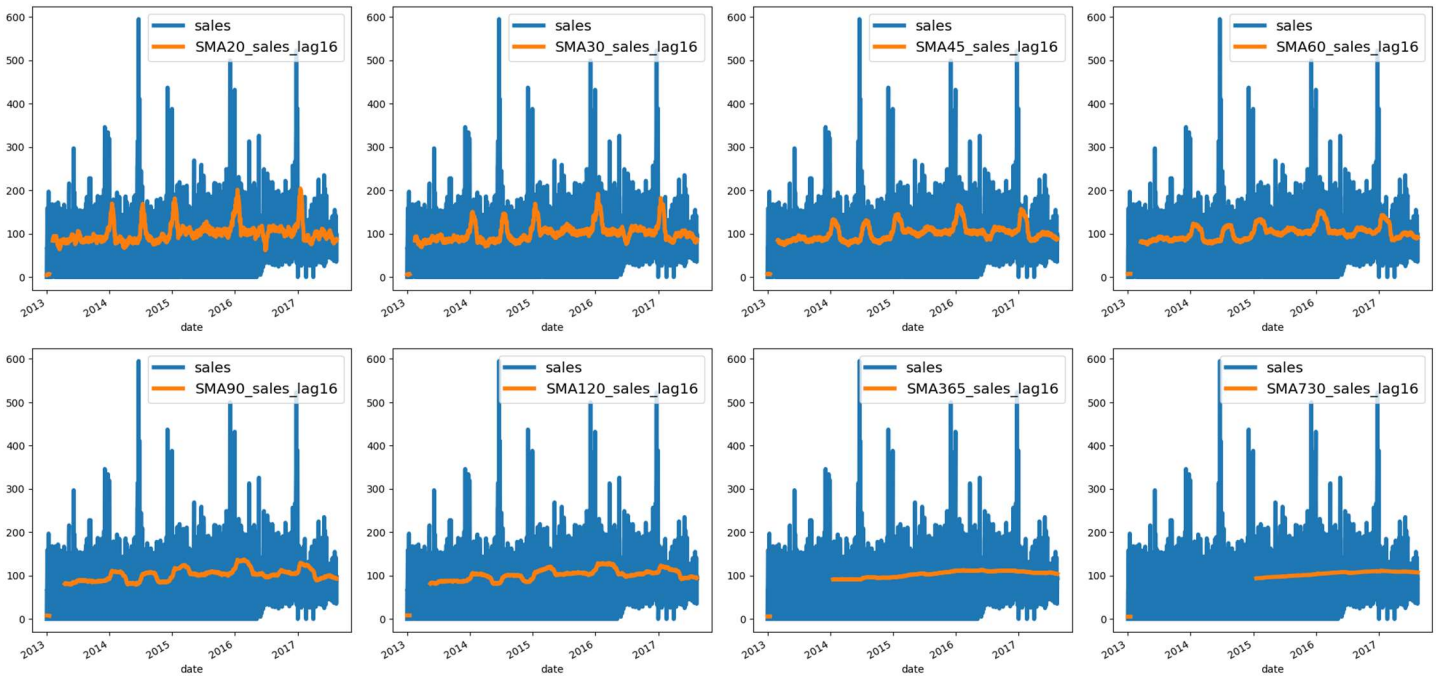
STORE 1 - LAWN AND GARDEN



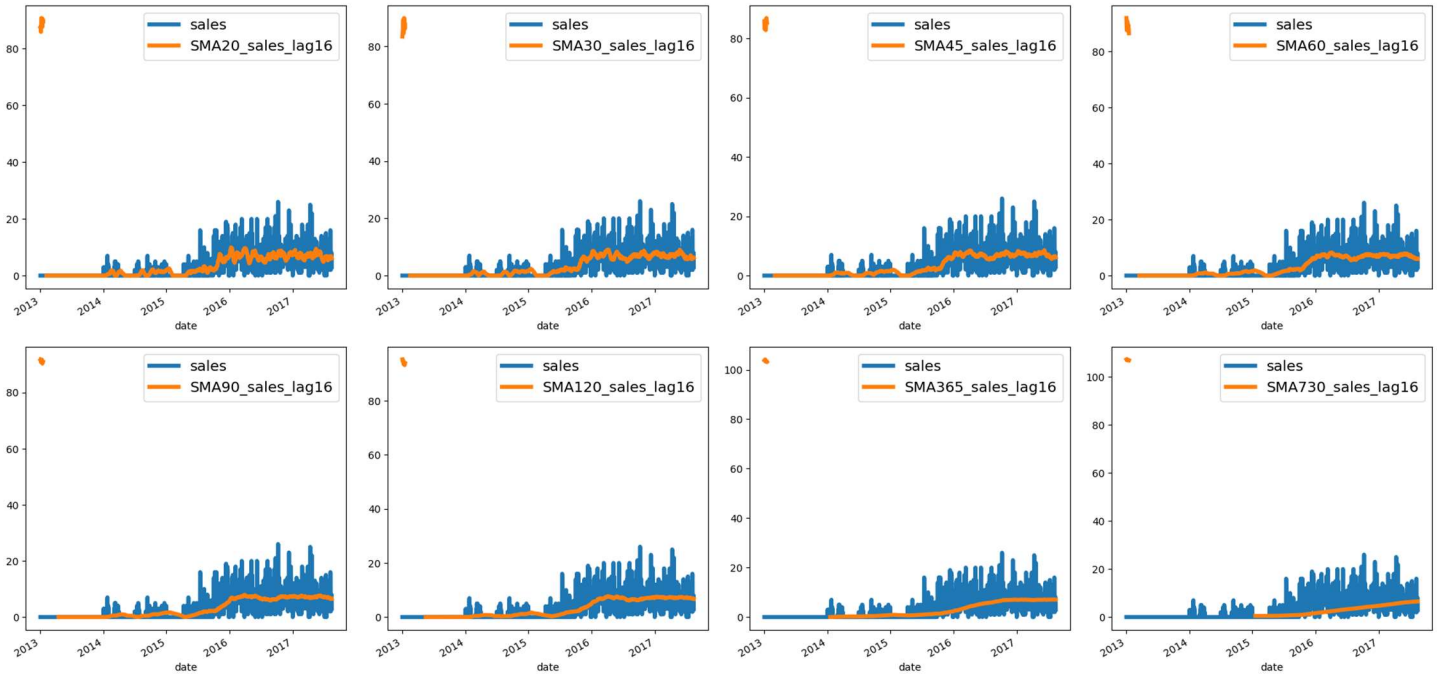
STORE 1 - LINGERIE



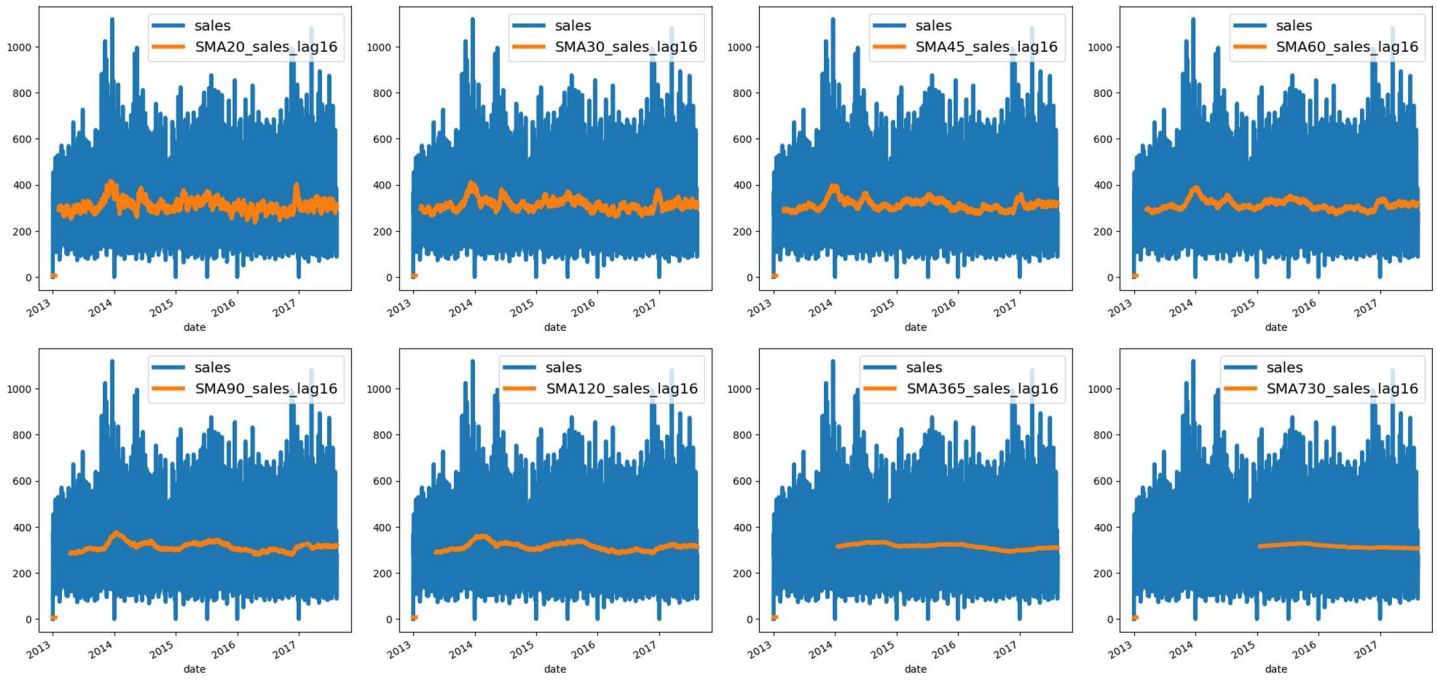
STORE 1 - LIQUOR,WINE,BEER



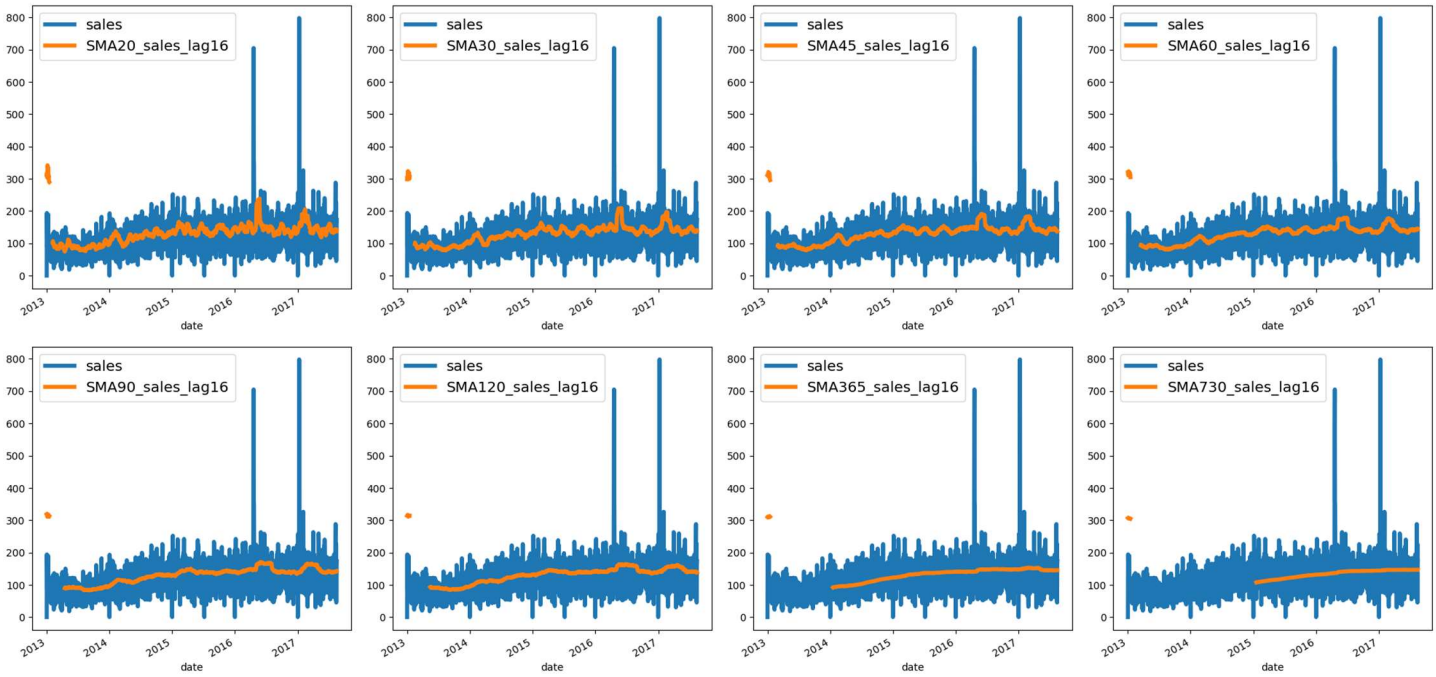
STORE 1 - MAGAZINES



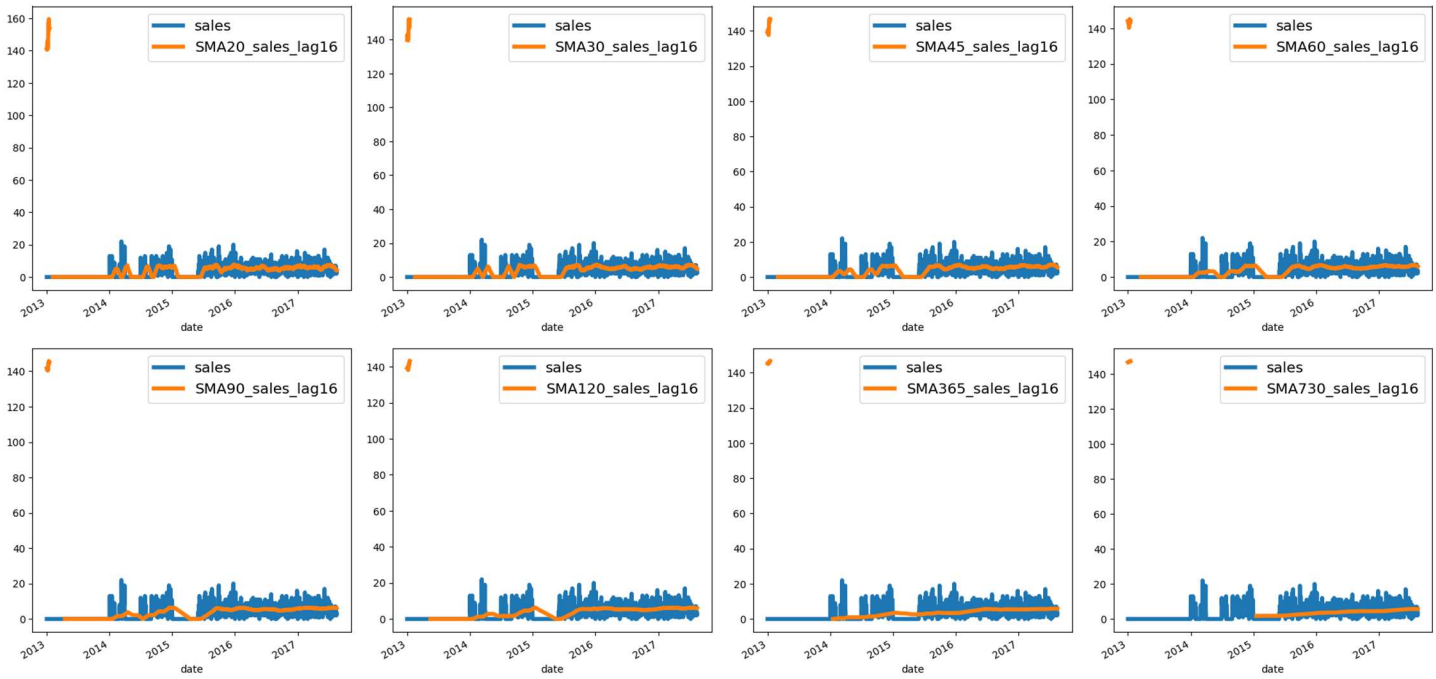
STORE 1 - MEATS



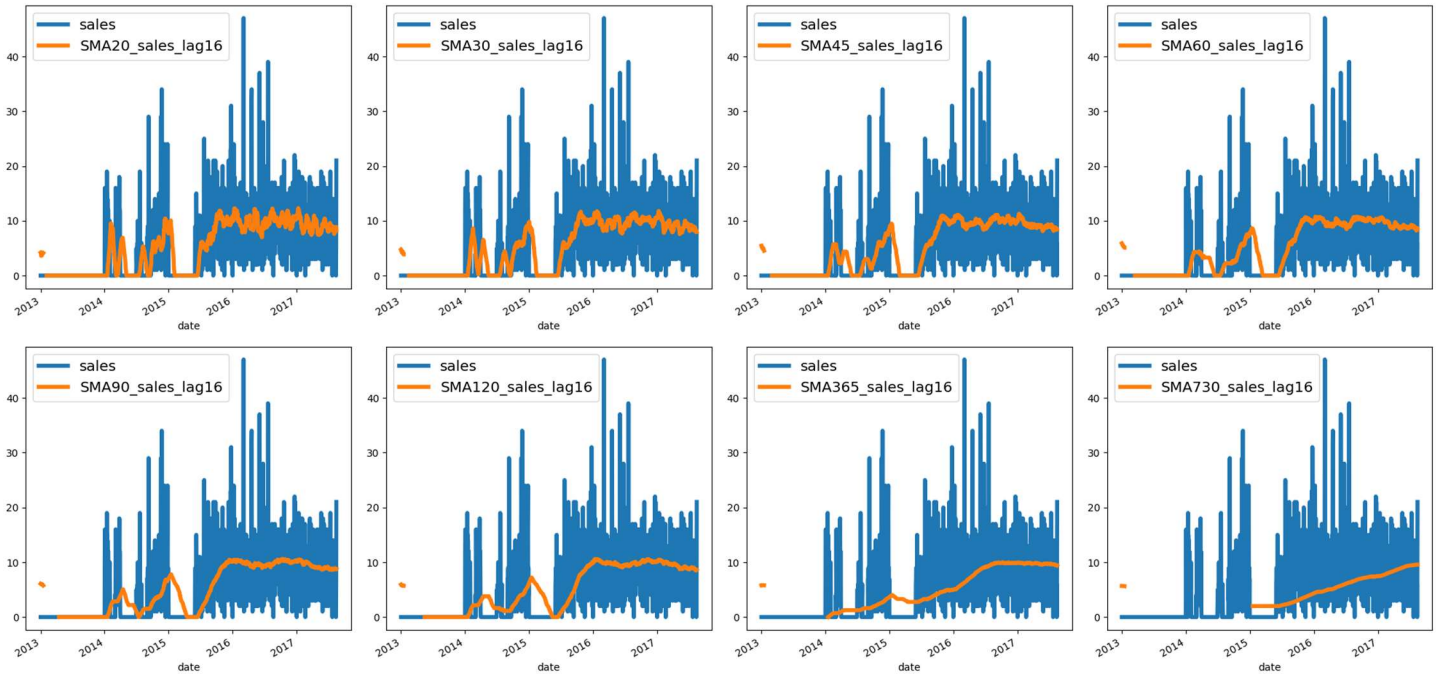
STORE 1 - PERSONAL CARE



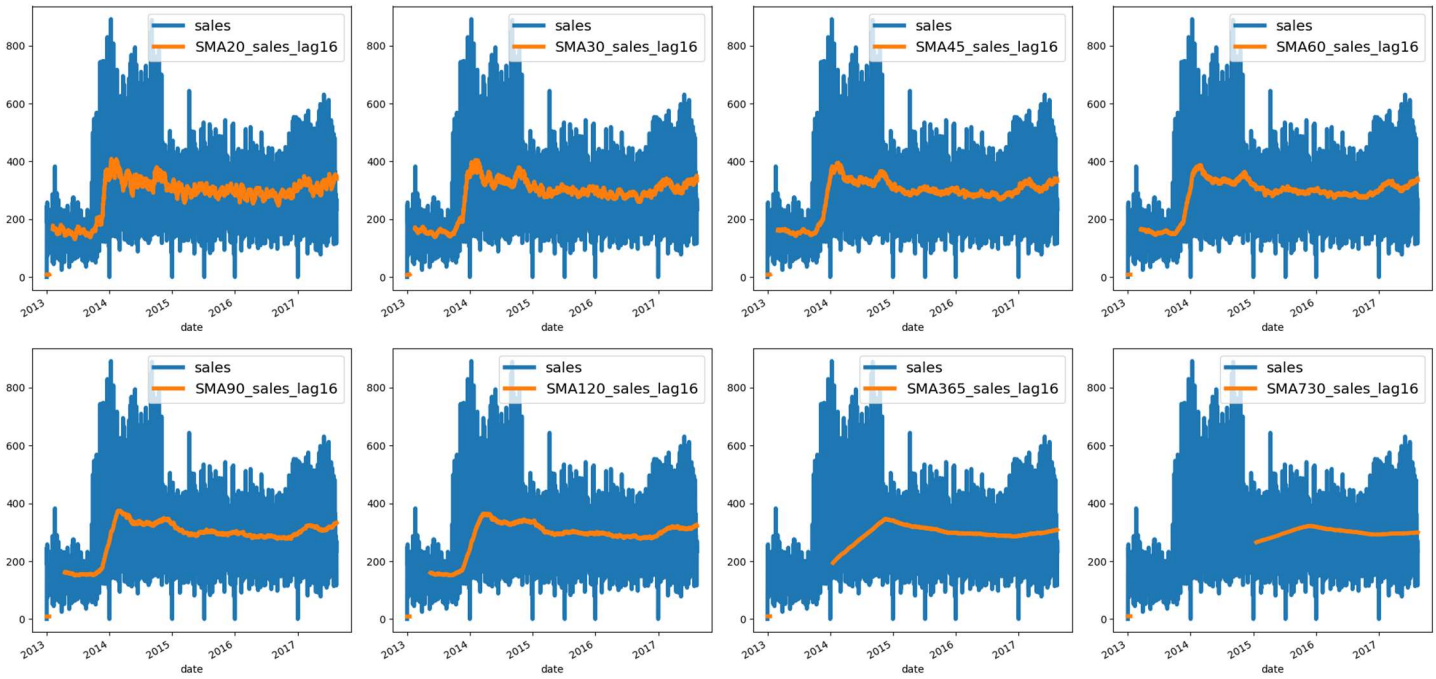
STORE 1 - PET SUPPLIES



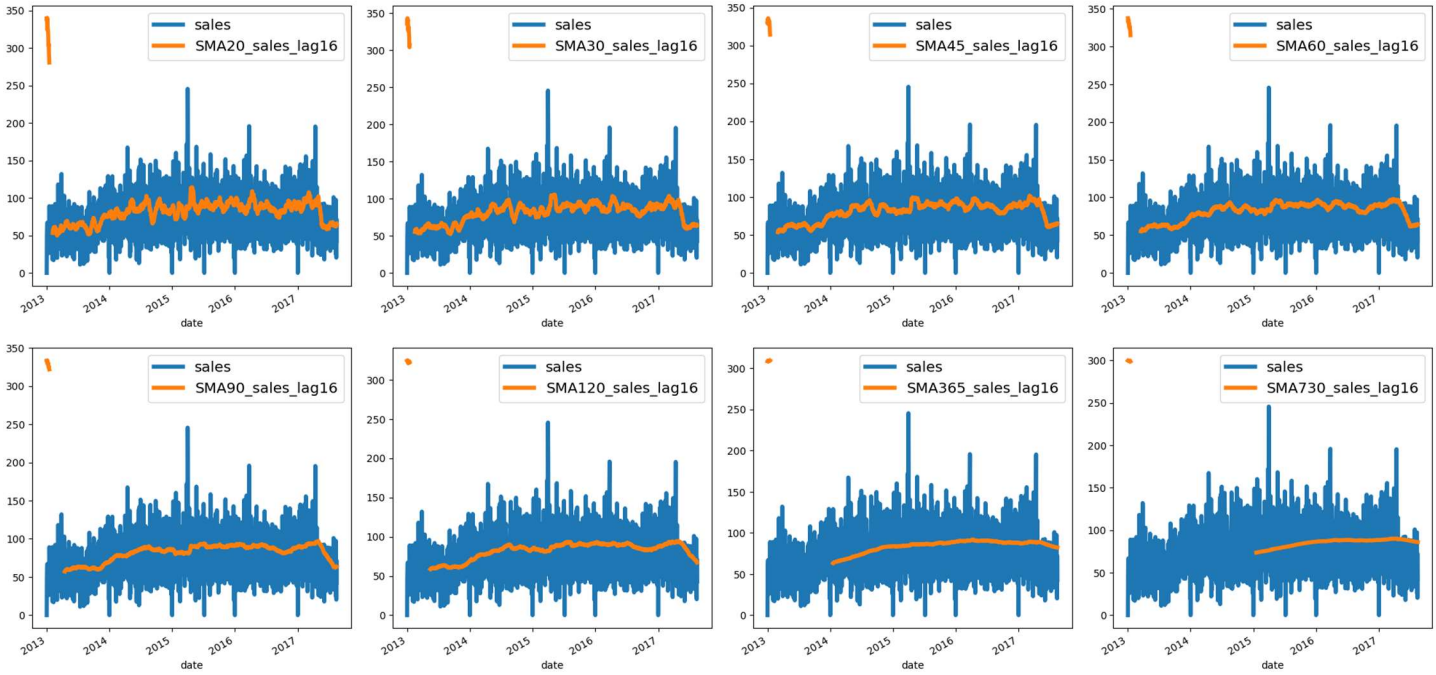
STORE 1 - PLAYERS AND ELECTRONICS



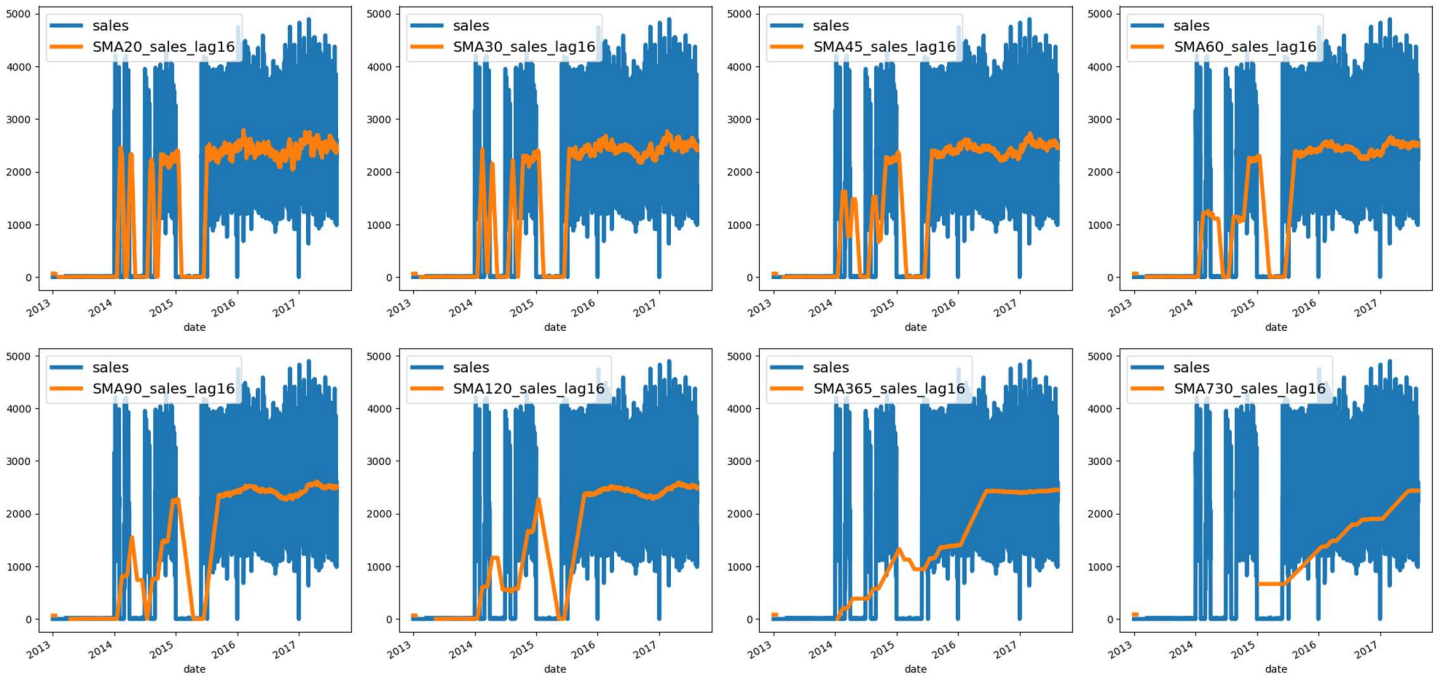
STORE 1 - POULTRY



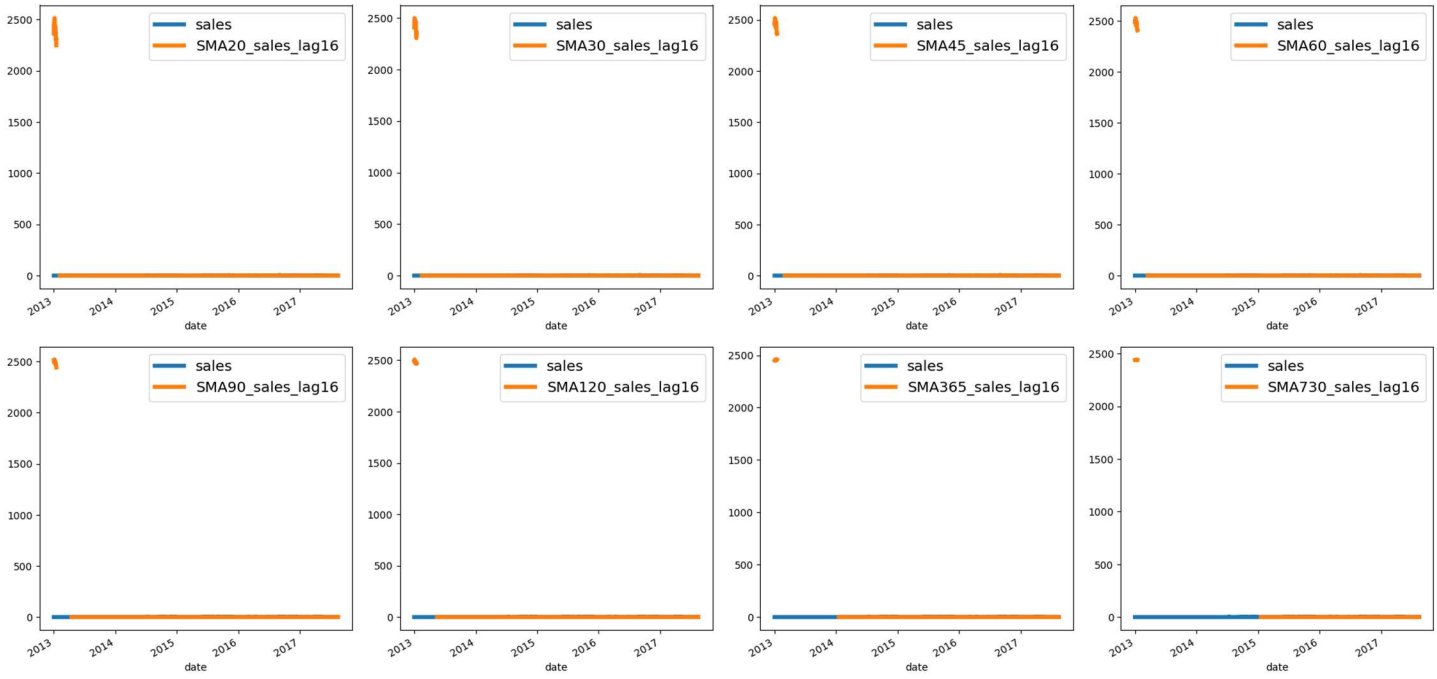
STORE 1 - PREPARED FOODS



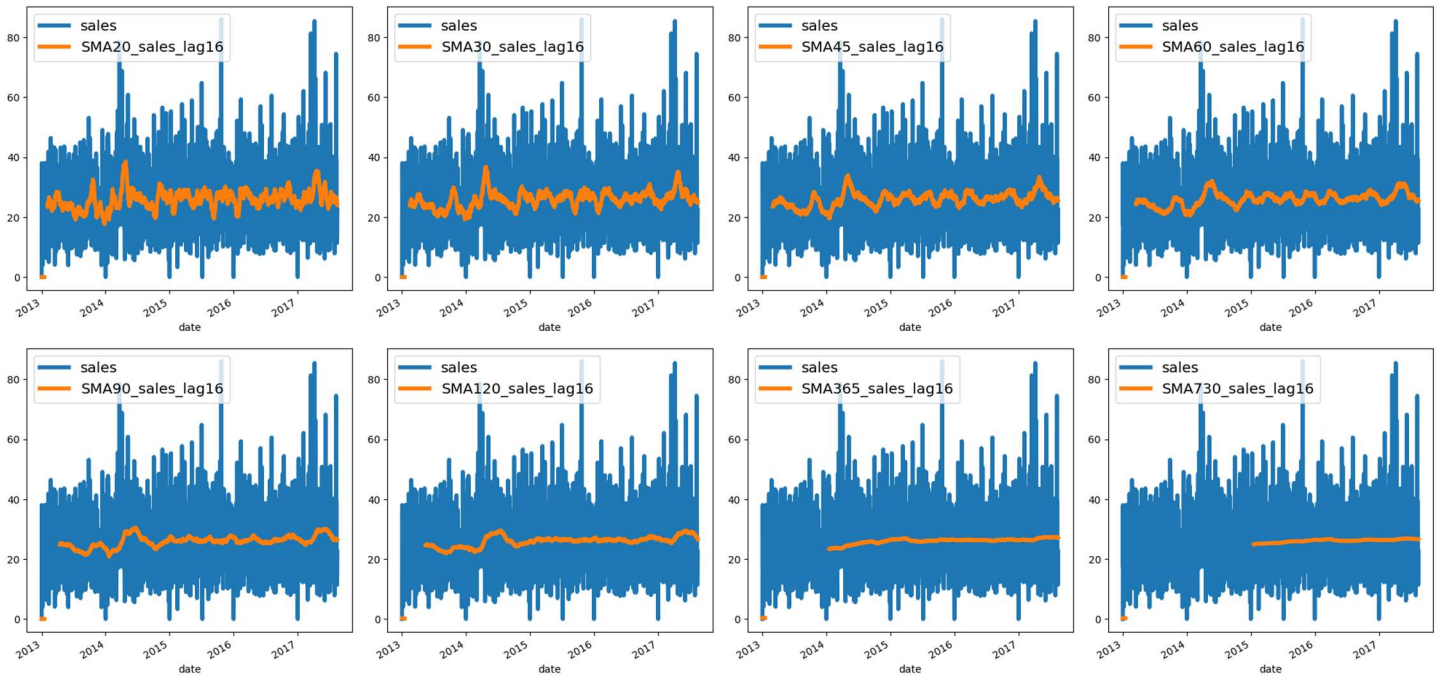
STORE 1 - PRODUCE



STORE 1 - SCHOOL AND OFFICE SUPPLIES



STORE 1 - SEAFOOD



4.6 Exponential Smoothing Forecast

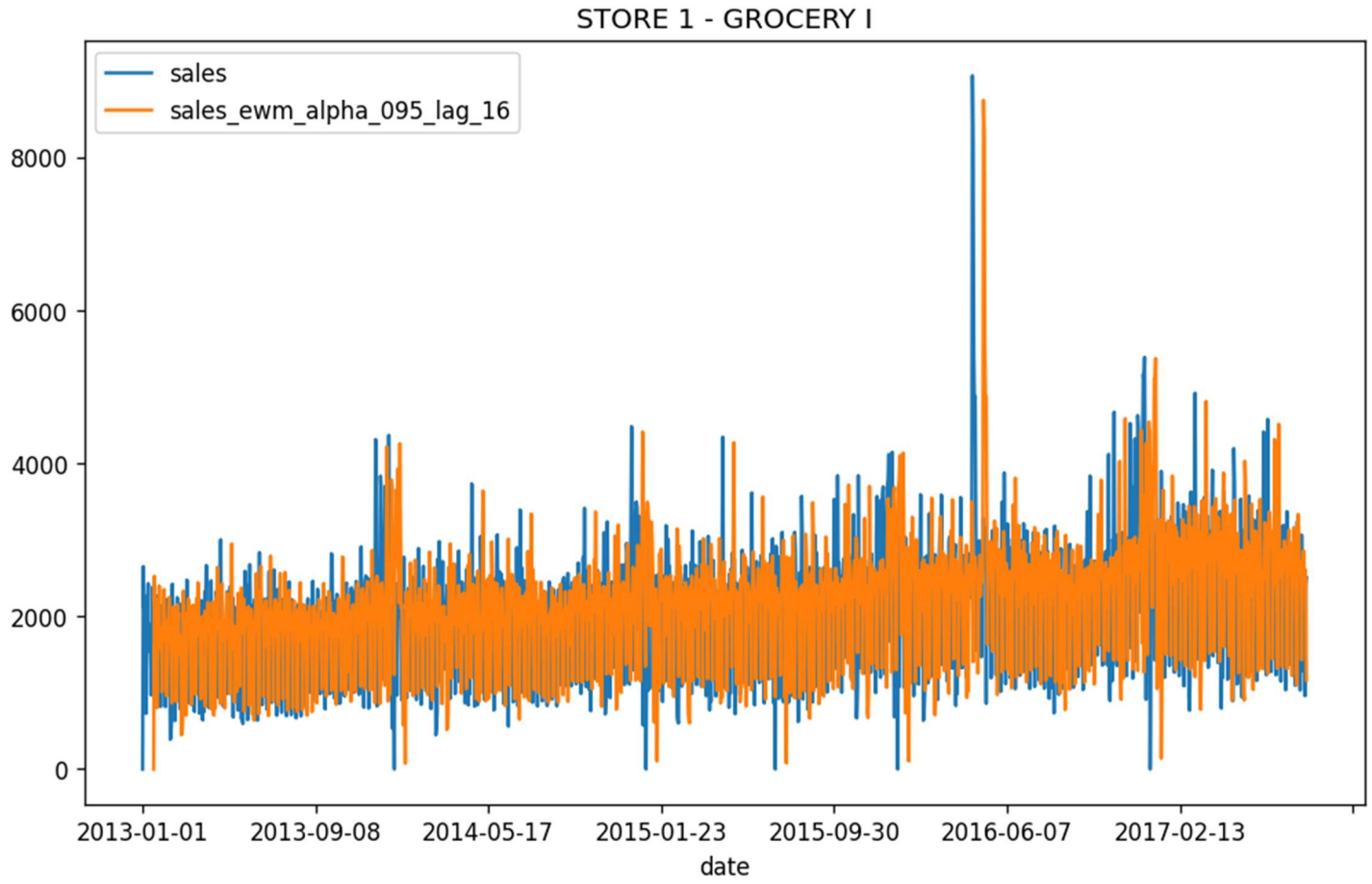


Figure 54 Exponential Smoothing Forecast

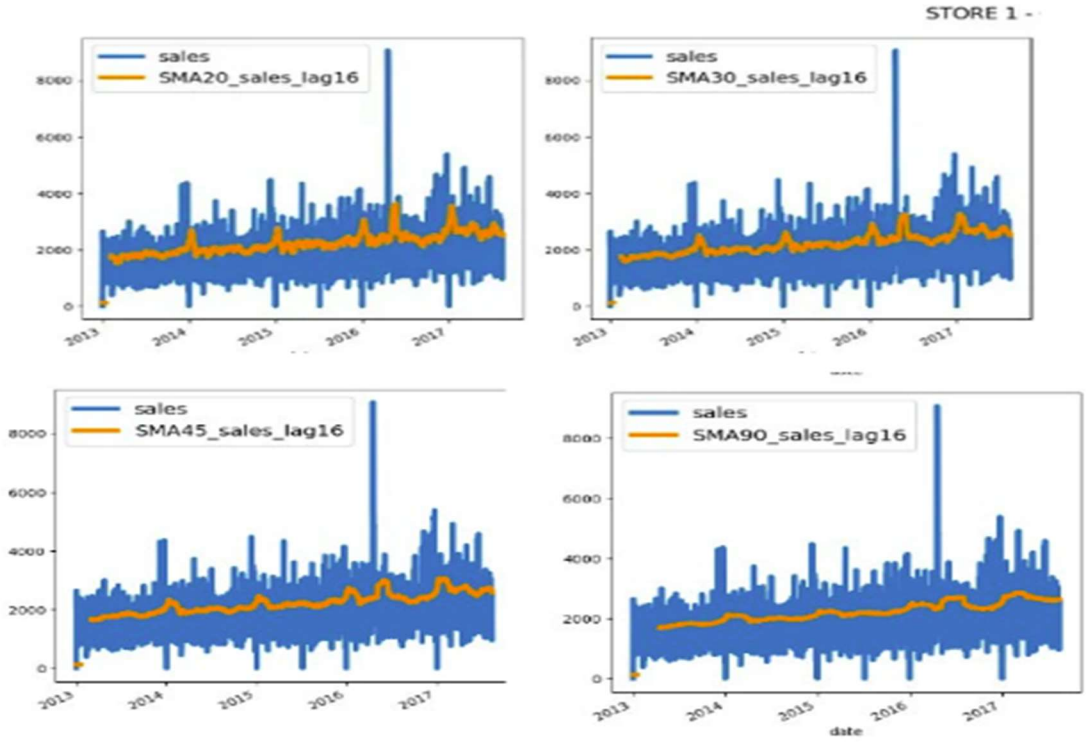
Similar to using a simple moving average, we can forecast using exponential smoothing, however in this part we want to display the data's fit results. In general, both approaches will produce the best results, allowing readers to quickly see a summary of how each product grows or shrinks or changes over time.

Note: The computations and formulas will be displayed in the appendix section.

PART 5: SENSITIVITY ANALYSIS & COMPARISON OF FORECASTING ERRORS

Sensitivity analysis is a crucial tool in various fields, including finance, engineering, economics, and decision-making processes. It involves assessing the impact of changes in input variables on the output of a model, system, or decision. In this project, we want to present an orientation of a new method of analyzing the sensitivity of the model: comparing the correlation value between SMAs with different jumps. Theocratically, Simple Moving Averages (SMAs) is a type of technical indicator used in financial analysis to smooth out price data and identify trends over a specific period. They are calculated by averaging the prices over that period, with the "jump" or "window size" determining the number of data points included in the average. On the other hand, correlation measures the statistical relationship between two variables. A correlation value ranges from -1 to 1, where -1 indicates a perfect negative correlation, 1 indicates a perfect positive correlation, and 0 indicates no correlation.

In this method, the greater the moving average fluctuates, the higher the jump is and the stronger the correlation is. For example, in this project we tried to compare MA of 4 different jumps: 20 days, 30 days, 45 days, 90 days.



Overall, from observation, we can conclude that the correlation value increases as the jump increases. When the jump is up, SMA will catch more long-term trends, resulting in a higher correlation value

Performing stepwise search to minimize aic

ARIMA(2,0,2)(1,0,1)[52] intercept : AIC=inf, Time=12.72 sec

ARIMA(0,0,0)(0,0,0)[52] intercept : AIC=1854.305, Time=0.03 sec

ARIMA(1,0,0)(1,0,0)[52] intercept : AIC=1577.148, Time=3.77 sec

ARIMA(0,0,1)(0,0,1)[52] intercept : AIC=inf, Time=4.29 sec

ARIMA(0,0,0)(0,0,0)[52] : AIC=2241.066, Time=0.05 sec

ARIMA(1,0,0)(0,0,0)[52] intercept : AIC=1602.051, Time=0.13 sec

ARIMA(1,0,0)(2,0,0)[52] intercept : AIC=inf, Time=23.69 sec

ARIMA(1,0,0)(1,0,1)[52] intercept : AIC=1578.210, Time=7.57 sec

ARIMA(1,0,0)(0,0,1)[52] intercept : AIC=1579.720, Time=2.43 sec

ARIMA(1,0,0)(2,0,1)[52] intercept : AIC=inf, Time=30.03 sec

ARIMA(0,0,0)(1,0,0)[52] intercept : AIC=1847.699, Time=3.78 sec

ARIMA(2,0,0)(1,0,0)[52] intercept : AIC=1578.688, Time=4.33 sec

ARIMA(1,0,1)(1,0,0)[52] intercept : AIC=1578.728, Time=4.79 sec

ARIMA(0,0,1)(1,0,0)[52] intercept : AIC=1724.147, Time=6.58 sec

ARIMA(2,0,1)(1,0,0)[52] intercept : AIC=inf, Time=7.36 sec

ARIMA(1,0,0)(1,0,0)[52] : AIC=1581.537, Time=2.65 sec

Best model: ARIMA(1,0,0)(1,0,0)[52] intercept

Total fit time: 114.303 seconds

SARIMAX Results

Dep. Variable: y No. Observations: 156

Model: SARIMAX(1, 0, 0)x(1, 0, 0, 52) Log Likelihood -784.574

Date: Thu, 04 Jan 2024 AIC 1577.148

Time: 16:08:50 BIC 1589.348

Sample: 0 HQIC 1582.103

- 156

Covariance Type: opg

coef std errz P>|z| [0.025 0.975]

intercept	12.8597	8.997	1.429	0.153	-4.774	30.493
ar.L1	0.9253	0.043	21.519	0.000	0.841	1.010
ar.S.L52	0.4384	0.092	4.766	0.000	0.258	0.619
sigma2	1257.7645	91.993	13.672	0.000	1077.461	1438.068

Ljung-Box (L1) (Q):	0.39	Jarque-Bera (JB):	220.58
Prob(Q):	0.53	Prob(JB):	0.00
Heteroskedasticity (H):	4.31	Skew:	1.06
Prob(H) (two-sided):	0.00	Kurtosis:	8.43

Warnings:

[1] Covariance matrix calculated using the outer product of gradients (complex-step).

PART 6: CONCLUSION ON HOW THESE MODELS FIT THE DATASET & YOUR PROPOSED PROBLEM

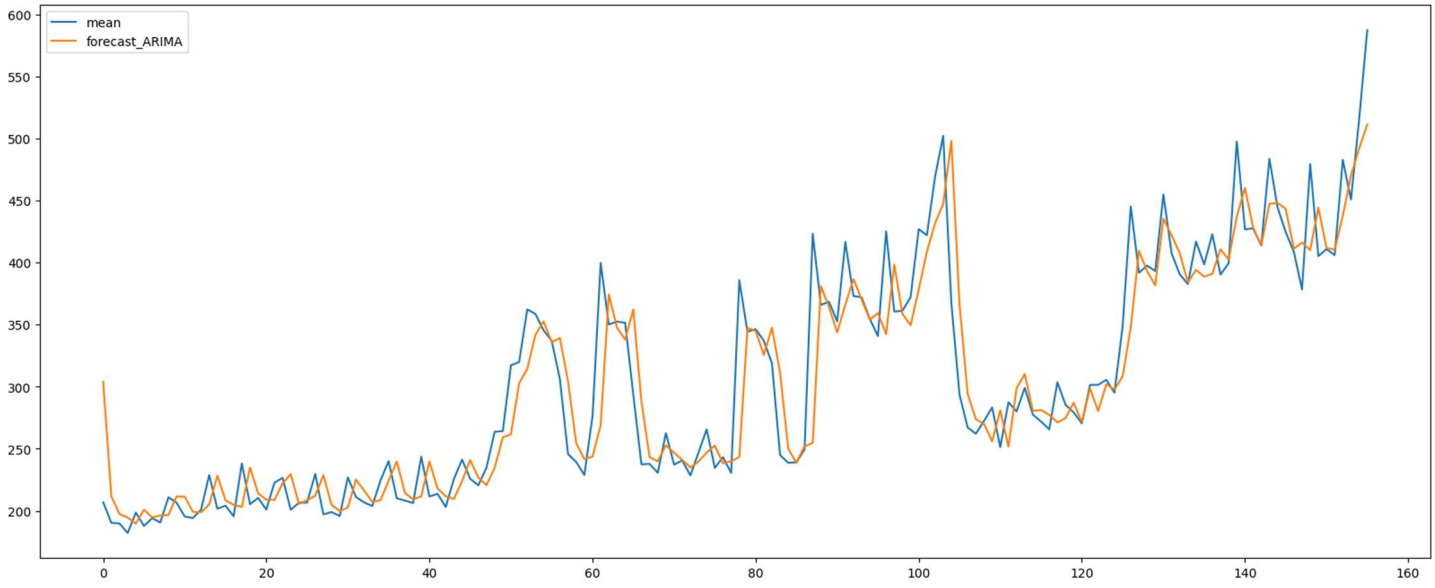


Figure 55 RMSE figure

Based on the computed error values and the values displayed on the chart in the earlier sections, we can say that the model has a decent fit, making it a useful foundation for predicting. contributes to future deep learning.

```
RMSE of Auto ARIMA: 74.11849912967759
MAE of Auto ARIMA: 60.58182312088649
```

It should be noted that future forecasting models may utilize the smoothed value files utilized in the simple moving average and exponential moving average as input data.

We have also partially practiced the subject's key concepts through the project, such as data smoothing, trend and season observation, and the use of simple and exponential moving averages to get desired outcomes. forecasts for the future. Further research is required to be able to translate the images into statistics and develop a primary forecast method, as the data obtained are only at a basic level.

PART 7: RECOMMENDATION FOR FURTHER RESEARCH.

To analyse the given data better and more precise, we suggest applying the Multiple Linear Regression. The first step is collecting data. We will define all variables as below:

Yt: sales over time (units)

X1t Price of Grocery I (USD)

X2t Price of Beverages (USD)

X3t Price of Produce (USD)

X4t Price of Cleaning (USD)

X5t Price of Dairy (USD)

X6t Price of Bread/ Bakery (USD)

X7t Price of Poultry (USD)

X8t Price of Meat (USD)

X9t Price of Personal Care (USD)

X10t Price of Deli (USD)

where all data is collected from 02/03/2012 to 26/12/2017

Yt is a dependent variable, X1t to X10t are independent variables

In the second step, Correlation matrix is formed by using Minitab, by observing the matrix, we can analyze the relationship between independent variable with independent variables.

In the third step, continuing to apply Minitab, Regression Equation is formed, for example: Yt = coef1 X1t + coef2 X2t + coef3 X3t + coef4 X4t + coef5 X5t + coef6 X6t + coef7 X7t + coef8 X8t + coef9 X9t + coef10 X10t

By using Minitab, we will have a Coefficient Table for Regression Model, which contains some such as: Coef SE, Coef, T-Value, P-Value and VIF.

Then, P-value and VIF will be taken into consideration.

P-Value: The P-value is the probability that we would observe the given t-value if statistics the null hypothesis is true. A p-value smaller than the significance level indicates that we can reject the null hypothesis.

VIF: Variance Inflation Factor is a measure of multicollinearity among the predictors in a regression model. A VIF of 1 indicates that there is no correlation among the predictor and the other predictors or the variance of that coefficient is not inflated at all. A larger VIF on an

independent variable indicates a highly collinear relationship to the other variables that should be considered or adjusted for in the structure of the model and selection of independent variables.

Finally, to determine how well the model fits our data, examine the goodness-of-fit statistics in the Model Summary table. In this table, there are values such as S, R-sq, R-sq (adj) and R-sq (pred). In our case, using R-sq (adj) to evaluate the strength of the model corresponded to our data. R-sq (adj): Use adjusted R-squared when we want to compare models that have different numbers of predictors. R-squared always increases when we add a predictor to the model, even when there is no real improvement to the model. The adjusted R-squared value incorporates the number of predictors in the model to help we choose the correct model.

PART 8 : APPENDIX: DATASET AND CODINGS FOR SOLVING AND ANALYZING THE PROBLEM.

8.1 Data

Thanks for kaggle with a great support providing free data for our to accomplish this project. Following the data in the link below : https://www.kaggle.com/competitions/store-sales-time-series-forecasting/data

- Holiday_events: https://www.kaggle.com/competitions/store-sales-time-series-forecasting/data?select=holidays_events.csv
- Oil data : https://www.kaggle.com/competitions/store-sales-time-series-forecasting/data?select=oil.csv
- Store data : https://www.kaggle.com/competitions/store-sales-time-series-forecasting/data?select=stores.csv
- Train data : https://www.kaggle.com/competitions/store-sales-time-series-forecasting/data?select=train.csv
- Test data : https://www.kaggle.com/competitions/store-sales-time-series-forecasting/data?select=test.csv
- Transaction : https://www.kaggle.com/competitions/store-sales-time-series-forecasting/data?select=transactions.csv

8.2 Coding

For easily observe entire the workspace, we put the link in the following link – using Google Colaboratory : https://colab.research.google.com/drive/178fEXv-vQhFAyMF8ux46_lsvxZ_-X9QL#scrollTo=vZP7HeHeRQOs

We also submit this coding.pdf beside the report for easily assess our work.