

# **DATATHON 2023**

Group Name: **Data Dazzlers**

University: **SLIIT**

Group Members:

- Siyumi Rivinma Pathirana
- Hiruni Imesha Saparamadu
- Thamasha Bandara
- Sewwandi Anjalika Meriyan

## Table of Contents

<b>Problem Statement .....</b>	<b>3</b>
<b>Dataset.....</b>	<b>3</b>
<b>Data cleaning and preprocessing .....</b>	<b>3</b>
<b>Exploratory data analysis and visualization .....</b>	<b>5</b>
<b>Sales Performance by Category, Department, and Store .....</b>	<b>6</b>
<b>Sales Performance by Location .....</b>	<b>7</b>
<b>Sales forecasting.....</b>	<b>8</b>
<b>Trend Analysis.....</b>	<b>8</b>
<b>Seasonal Decomposition of Time Series (STL).....</b>	<b>9</b>
<b>Sales Forecasting with ARIMA and SARIMA .....</b>	<b>10</b>
<b>Sales Forecasting with Holt-Winters Exponential Smoothing.....</b>	<b>11</b>

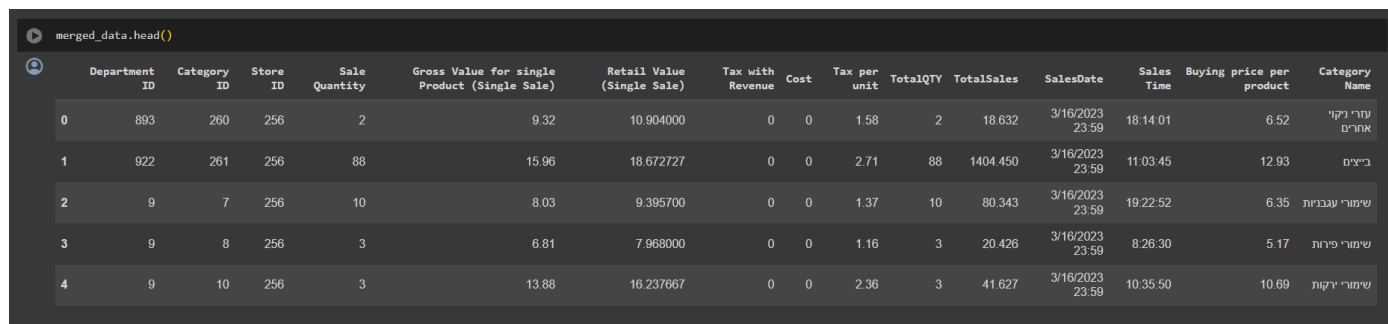
## Problem Statement

Orel IT is a leading IT solution provider, has a client who needs descriptive data insight about a dataset that was generated during the POS transactions of supermarket product sales. The client needs to present the dataset to the management and the investors with convincing insights. Orel IT Data team needs to create these insights with the help of data scientists.

## Dataset

- General sales data
- Department data
- Store data
- Category data

First, we have integrated all 4 datasets by using pandas library. Department data, store data and category data were linked with the general sales data using id columns so we have combined datasets using their id columns.



The screenshot shows a Jupyter Notebook interface with a code cell containing `merged_data.head()`. Below the code, the first five rows of a merged dataset are displayed. The columns include Department ID, Category ID, Store ID, Sale Quantity, Gross Value for single Product (Single Sale), Retail Value (Single Sale), Tax with Revenue, Cost, Tax per unit, TotalQTY, TotalSales, SalesDate, Sales Time, Buying price per product, and Category Name. The data is presented in a dark-themed table.

	Department ID	Category ID	Store ID	Sale Quantity	Gross Value for single Product (Single Sale)	Retail Value (Single Sale)	Tax with Revenue	Cost	Tax per unit	TotalQTY	TotalSales	SalesDate	Sales Time	Buying price per product	Category Name
0	893	260	256	2	9.32	10.904000	0	0	1.58	2	18.632	3/16/2023 23:59	18:14:01	6.52	נקני אחרים
1	922	261	256	88	15.96	18.672727	0	0	2.71	88	1404.450	3/16/2023 23:59	11:03:45	12.93	ביצים
2	9	7	256	10	8.03	9.395700	0	0	1.37	10	80.343	3/16/2023 23:59	19:22:52	6.35	שימור עגבניות
3	9	8	256	3	6.81	7.968000	0	0	1.16	3	20.426	3/16/2023 23:59	8:26:30	5.17	שימור פירות
4	9	10	256	3	13.88	16.237667	0	0	2.36	3	41.627	3/16/2023 23:59	10:35:50	10.69	שימור ירקות

## Data cleaning and preprocessing

- Initially we checked for missing values and there was only one missing value in the department dataset, so we ignored the row.
- There were no any duplicate values.

- To concatenate general sales and department datasets, we had to convert the datatype of the id column, since there were 2 different datatypes for the department id column, in 2 datasets. Therefore, we converted department id datatype into int64.

### Conversion of Data Types before concatenate

```
# Check the current data types
print("Before Conversion:")
print(Department_Data.dtypes)

# Convert 'Department ID' column to int
Department_Data['Department ID'] = Department_Data['Department ID'].astype(int)

# Check the data types after conversion
print("\nAfter Conversion:")
print(Department_Data.dtypes)
```

Before Conversion:

```
Department ID      object
Department Name    object
dtype: object
```

After Conversion:

```
Department ID      int64
Department Name    object
dtype: object
```

<ipython-input-487-a1d31e720025>:6: SettingWithCopyWarning:  
A value is trying to be set on a copy of a slice from a DataFrame.  
Try using .loc[row\_indexer,col\_indexer] = value instead

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/index.html](https://pandas.pydata.org/pandas-docs/stable/user_guide/index.html)  
Department\_Data['Department ID'] = Department\_Data['Department ID'].astype(int)

- After that we got the statistic for the merged dataset and it showed that both **'tax with revenue'** and **'cost'** columns are having only 0's. therefore we removed both the columns from the dataset.
- We detected that for the **'Sale quantity'**, **'Retail value (Single sale)'**, **'Tax per unit'**, **'TotalQty'**, **'Totalsales'** columns had negative values as well. Since it is not possible to have negative values for the mentioned columns, we checked number of rows which are having negative values.

```
[ ] # Specify the columns you want to check for negative values
columns_to_check = ['Sale Quantity', 'Retail Value (Single Sale)', 'Tax per unit', 'TotalQty', 'TotalSales']

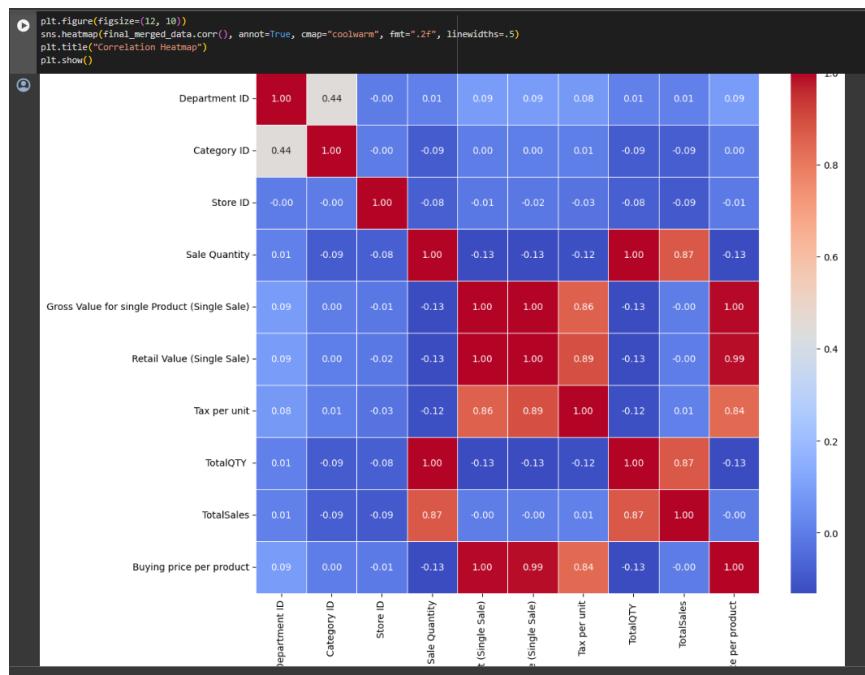
# Filter rows with at least one negative value in specified columns
rows_with_negatives = final_merged_data[(final_merged_data[columns_to_check] < 0).any(axis=1)]

# Print the resulting DataFrame
rows_with_negatives
```

	Department ID	Category ID	Store ID	Sale Quantity	Gross Value for single Product (Single Sale)	Retail Value (Single Sale)	Tax per unit	TotalQty	TotalSales	SalesDate	Sales Time	Buying price per product	Category Name	Department Name	Store Name	Location
742	867	56	261	1	0.00	0.000	0.00	0	-0.0001	3/16/2023 23:59	19:16:25	0.00	סיהור אוור	חומרי ניקוי אחרים	קניון 261	Central
1114	885	240	8	-1	12.74	14.900	2.16	-1	-12.7400	3/16/2023 23:59	21:42:42	9.56	סודה קלאב	כלי בית	קניון 8	North East

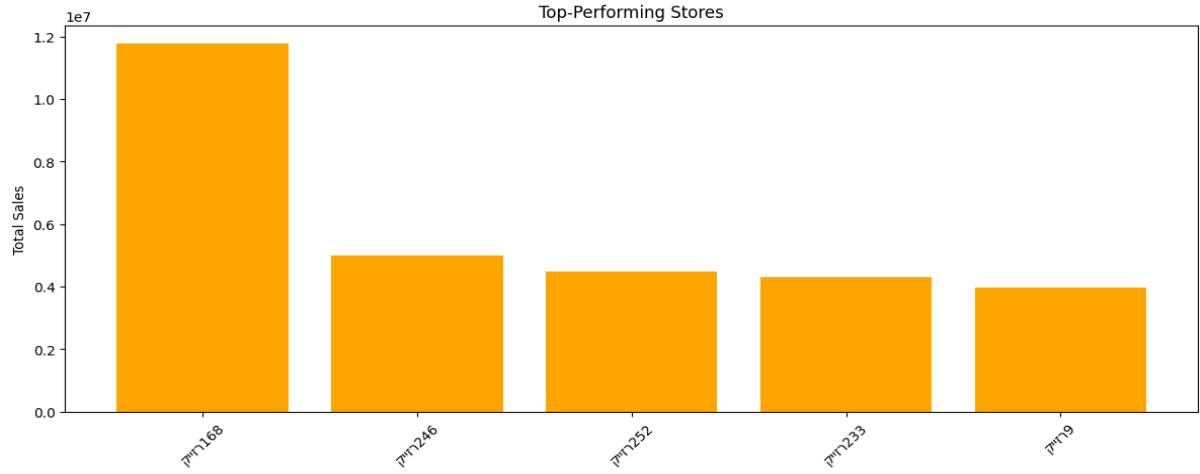
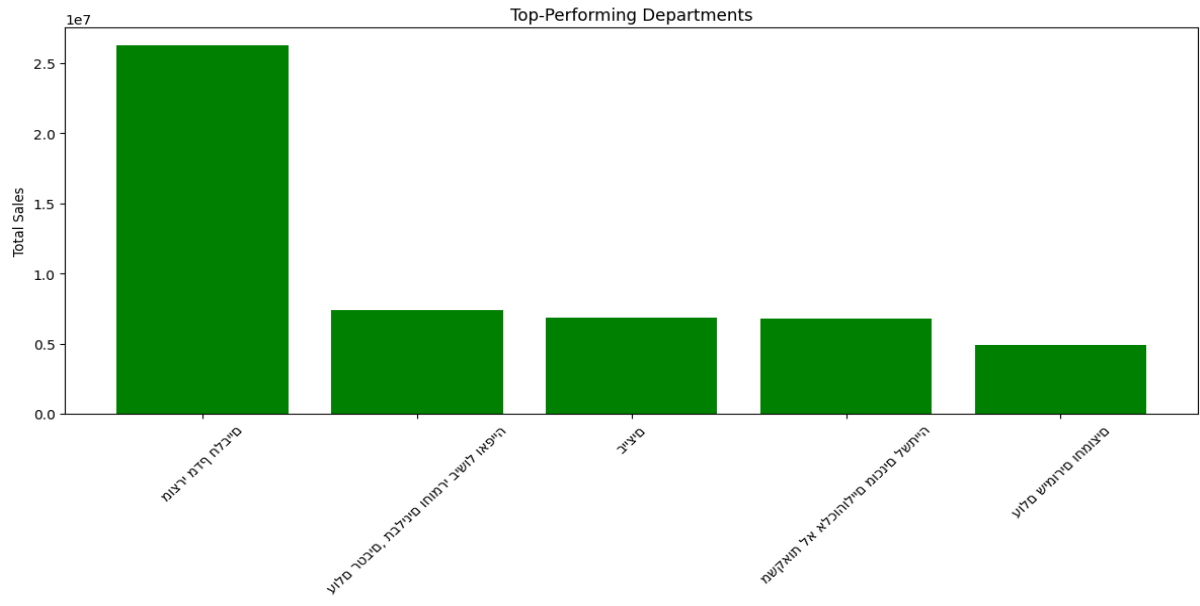
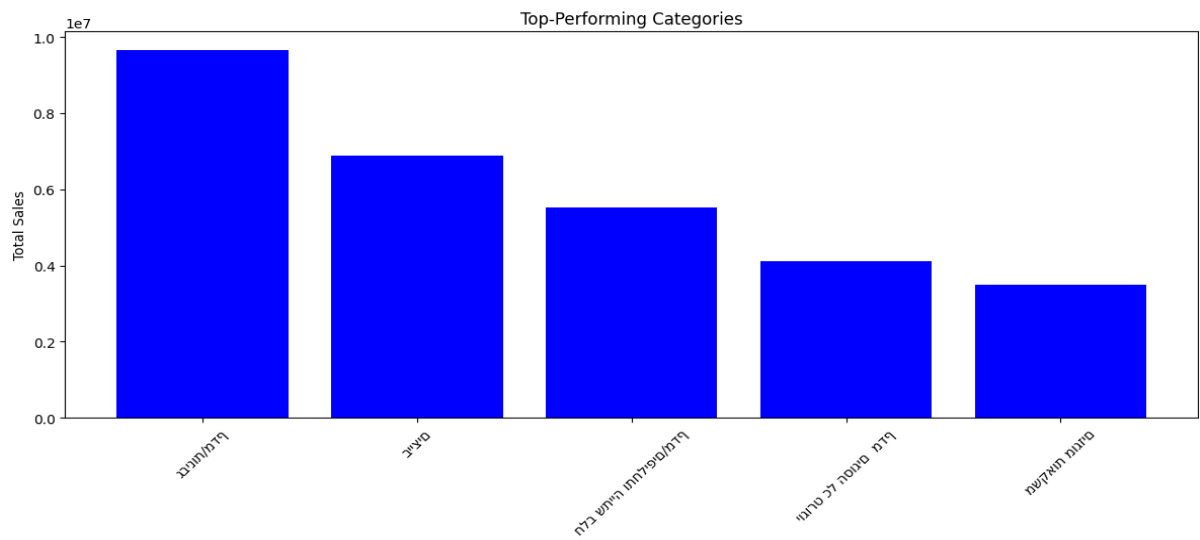
- We removed rows which were having negative values, because the percentage of negative values out of entire dataset values were a small number (0.11%).
- After that we checked for outliers which means the data points, which are having abnormal behaviours. Then we removed outliers.

## Exploratory data analysis and visualization

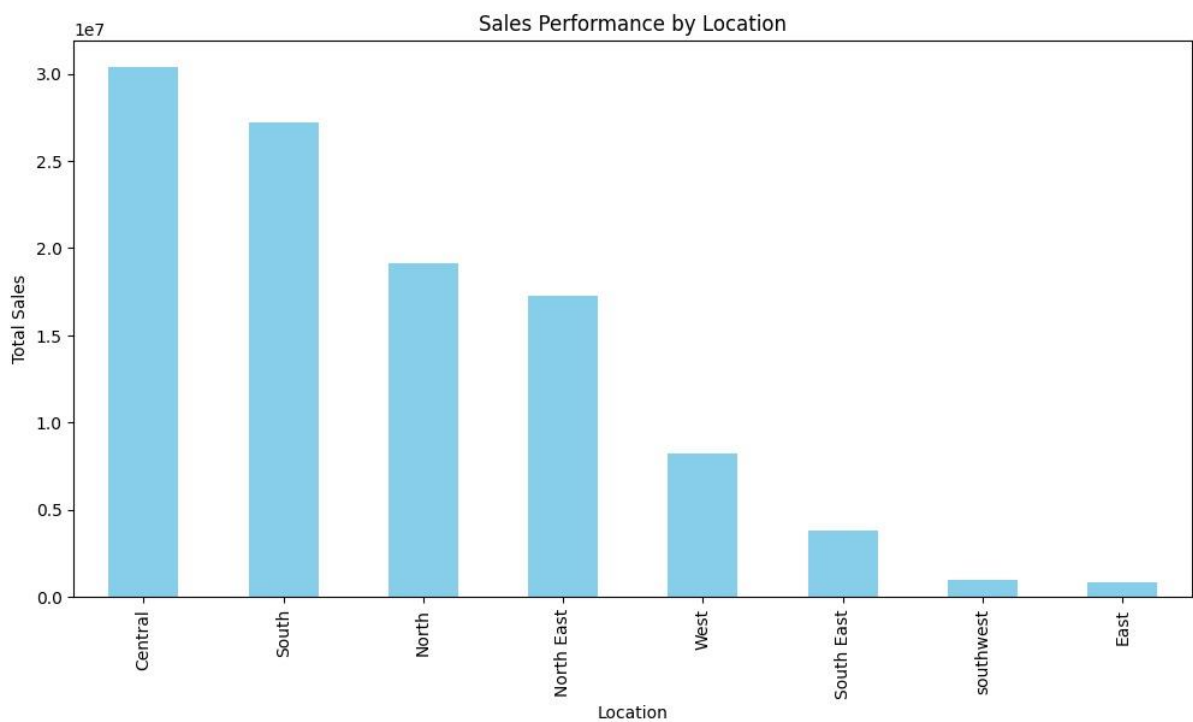


- First, we created a heatmap to find features that are highly correlate with each other, since the features that are highly correlate with each other can be represent only by using a single feature because if 2 features are highly correlate with each other, that means those 2 features are having almost same behaviour.
- Then we plot the top performing stores, categories and departments.
- Further plotted the sales performance by location as well.

# Sales Performance by Category, Department, and Store



# Sales Performance by Location

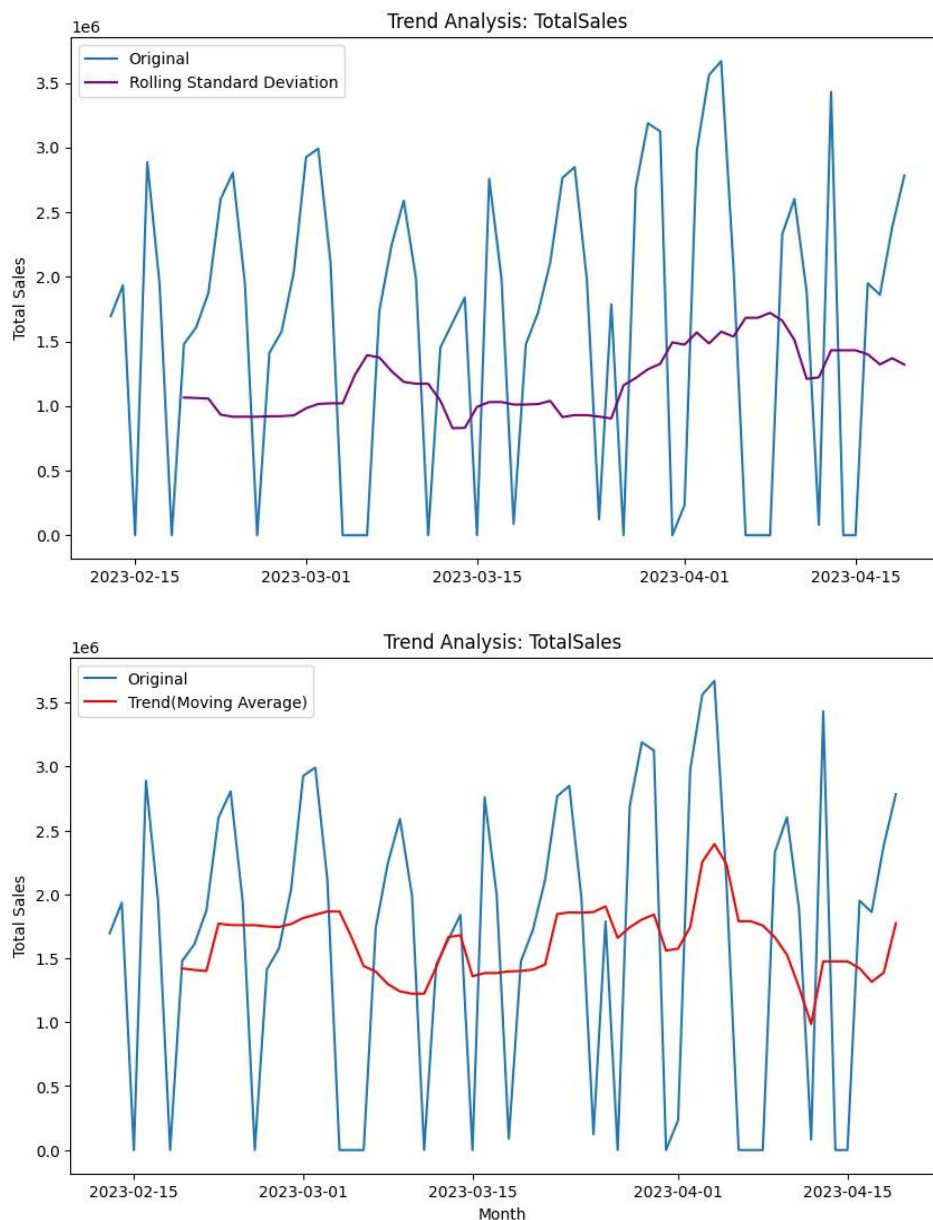


## Sales forecasting

Time series forecasting is a technique used to predict future values based on historical observations ordered chronologically. This method is particularly effective for datasets where the data points are associated with specific time intervals. Since our dataset also having sales data related to 2-month duration, we chose time series forecasting technique to proceed with the data analysis part.

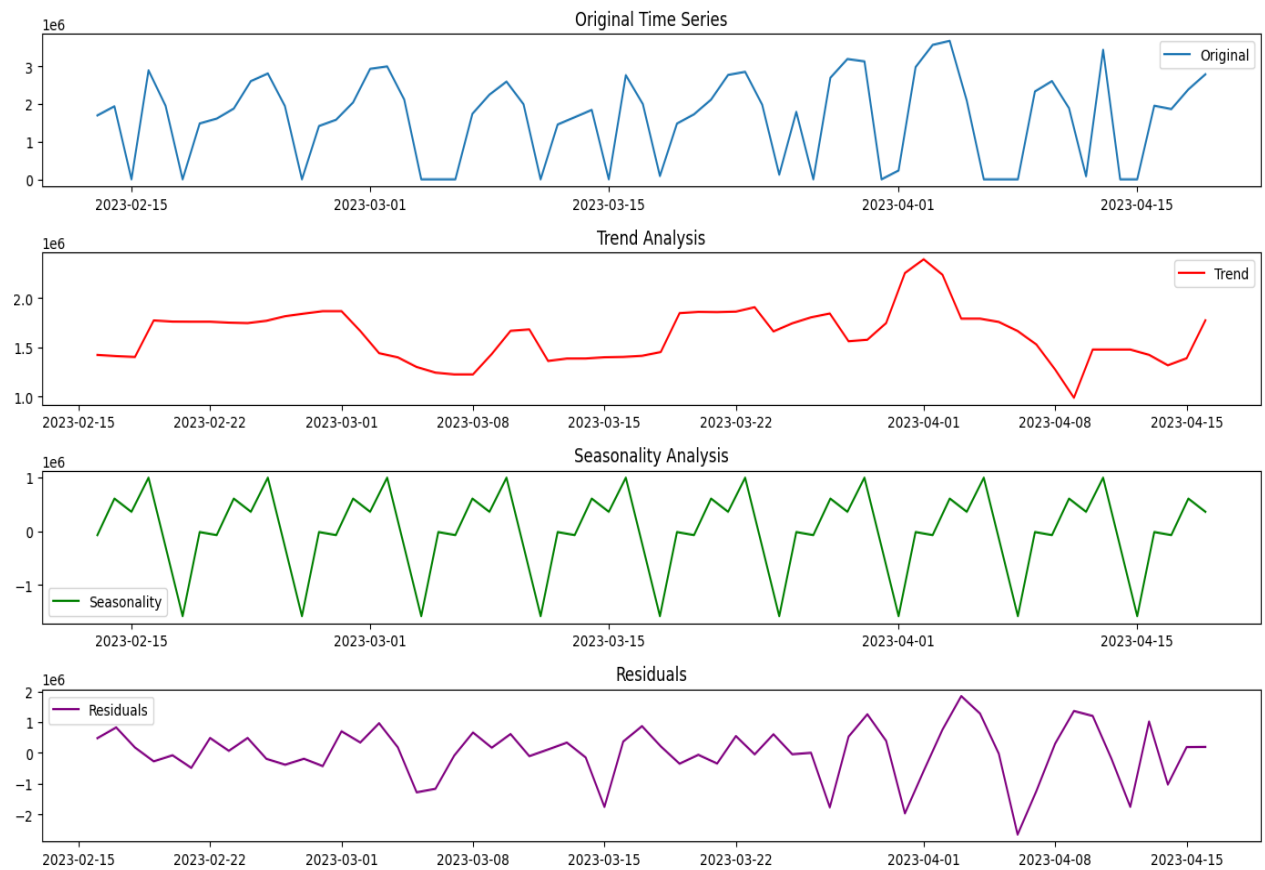
We have chosen the ARIMA (Autoregressive Integrated Moving Average) model as the forecasting model.

## Trend Analysis

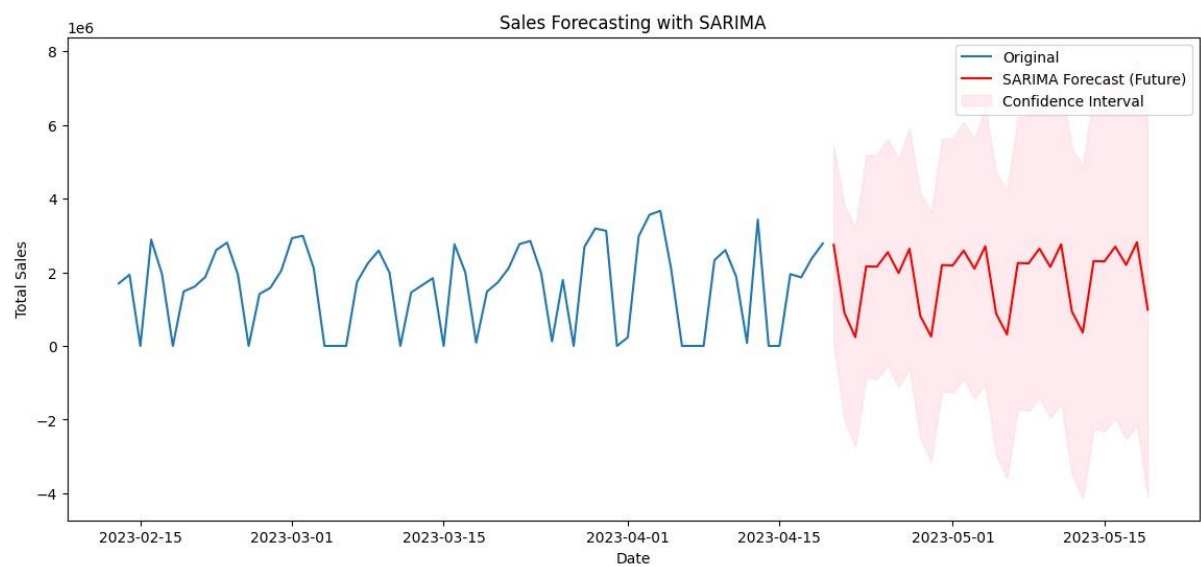
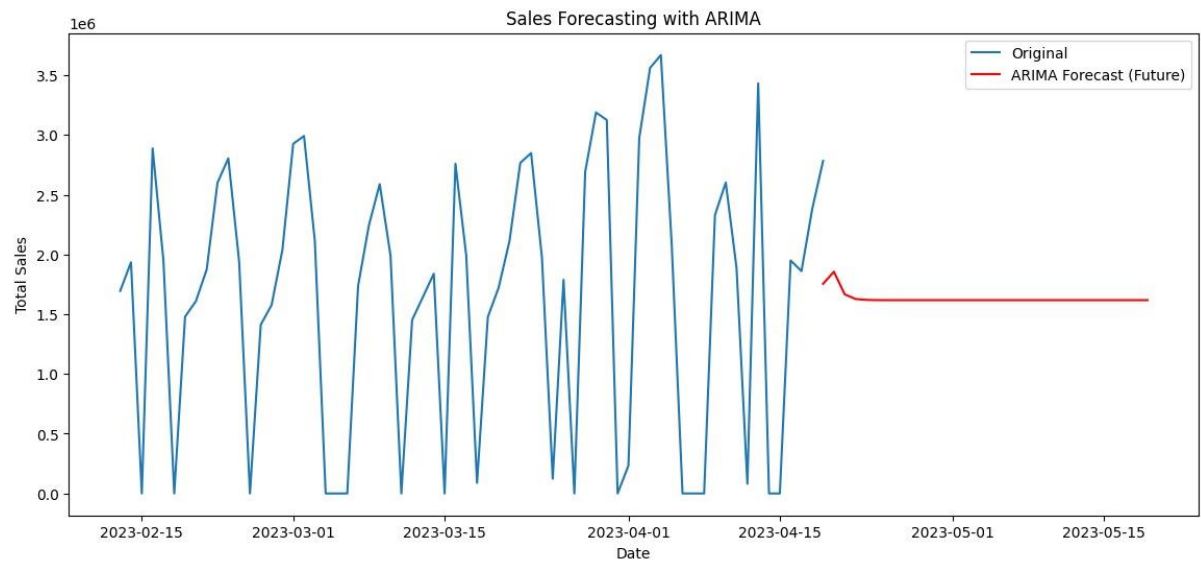




## Seasonal Decomposition of Time Series (STL)



## Sales Forecasting with ARIMA and SARIMA



## Sales Forecasting with Holt-Winters Exponential Smoothing

