Capstone Project (Walmart)

Table of Contents

1. Problem Statement

Walmart, a retail giant with multiple outlets across the country, is experiencing challenges in accurately forecasting weekly sales for each store. This challenge is crucial for managing inventory to align supply with demand, thereby preventing overstocking or stockouts. The primary objective is to develop a robust predictive model to forecast sales for the next 12 weeks, aiding better inventory management and overall operational efficiency.

2. Project Objective

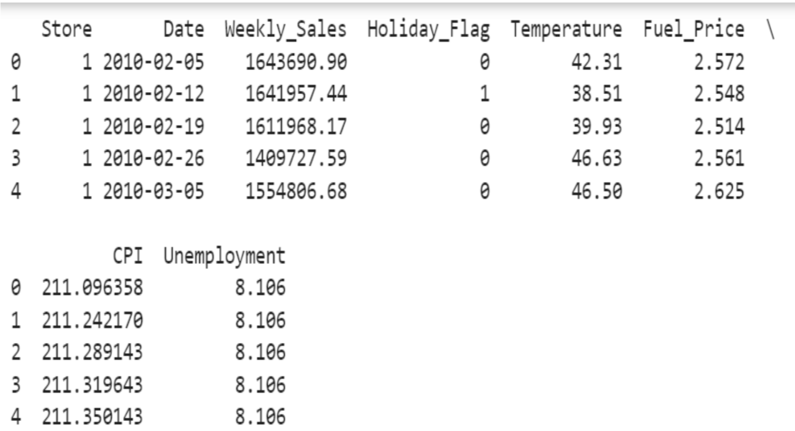
The objective of this project is to analyse Walmart's sales data, derive useful insights for improvement, and develop predictive models to forecast weekly sales for each store for the next 12 weeks.

3. Data Description

The dataset available is Walmart's historical sales data, consisting of the following columns:

- **Store**: Store ID
- **Date**: Date of the observation
- **Weekly_Sales**: Sales for the given store and date
- **Holiday_Flag**: Indicates whether the week includes a major holiday (1) or not (0)
- **Temperature**: Temperature in the region
- **Fuel_Price**: Cost of fuel in the region
- **CPI**: Consumer Price Index
- **Unemployment**: Unemployment rate

```
print(df.head())
```

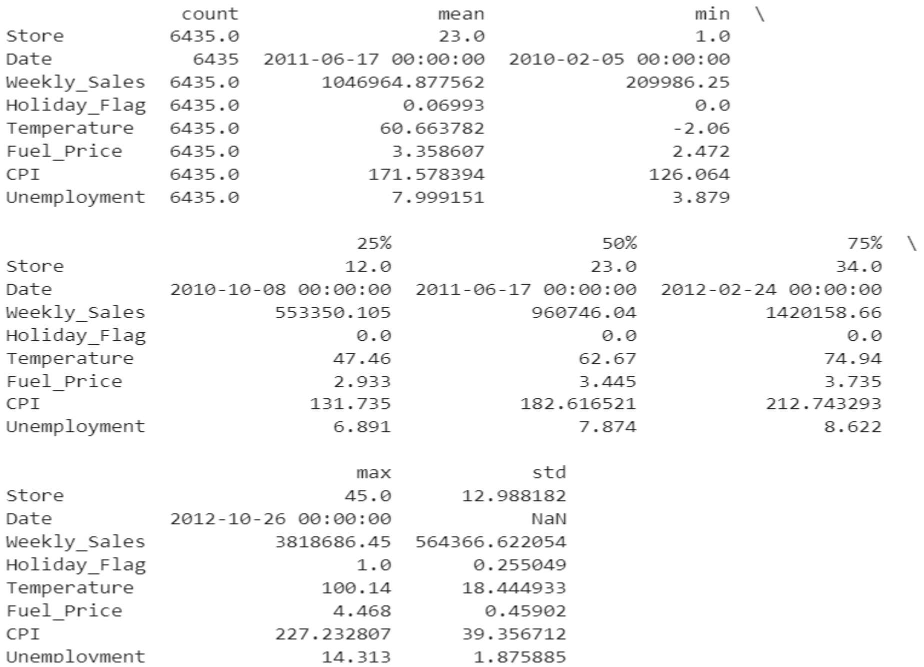


```
   Store       Date  Weekly_Sales  Holiday_Flag  Temperature  Fuel_Price  \
0      1 2010-02-05    1643690.90             0        42.31       2.572
1      1 2010-02-12    1641957.44             1        38.51       2.548
2      1 2010-02-19    1611968.17             0        39.93       2.514
3      1 2010-02-26    1409727.59             0        46.63       2.561
4      1 2010-03-05    1554806.68             0        46.50       2.625

          CPI  Unemployment
0  211.096358         8.106
1  211.242170         8.106
2  211.289143         8.106
3  211.319643         8.106
4  211.350143         8.106
```

1. Descriptive Statistics:

```
print(df.describe().T)
```



```
              count                 mean                  min  \
Store        6435.0                 23.0                  1.0
Date           6435  2011-06-17 00:00:00  2010-02-05 00:00:00
Weekly_Sales 6435.0       1046964.877562            209986.25
Holiday_Flag 6435.0              0.06993                  0.0
Temperature  6435.0            60.663782                -2.06
Fuel_Price   6435.0             3.358607                2.472
CPI          6435.0           171.578394              126.064
Unemployment 6435.0             7.999151                3.879

                             25%                  50%                  75%  \
Store                       12.0                 23.0                 34.0
Date         2010-10-08 00:00:00  2011-06-17 00:00:00  2012-02-24 00:00:00
Weekly_Sales          553350.105            960746.04           1420158.66
Holiday_Flag                 0.0                  0.0                  0.0
Temperature                47.46                62.67                74.94
Fuel_Price                 2.933                3.445                3.735
CPI                      131.735           182.616521           212.743293
Unemployment               6.891                7.874                8.622

                             max            std
Store                       45.0      12.988182
Date         2012-10-26 00:00:00            NaN
Weekly_Sales          3818686.45  564366.622054
Holiday_Flag                 1.0       0.255049
Temperature               100.14      18.444933
Fuel_Price                 4.468        0.45902
CPI                   227.232807      39.356712
Unemployment              14.313       1.875885
```

- **Weekly_Sales**: Mean of $1,046,965 with a standard deviation of $564,367, indicating substantial variability in weekly sales.
- **Holiday_Flag**: Most weeks are non-holidays, with holidays representing about 7% of the data.
- **Temperature**: Ranges from -2.06°F to 100.14°F with a mean of 60.66°F.
- **Fuel_Price**: Ranges from $2.47 to $4.47 with a mean of $3.36.
- **CPI**: Ranges from 126.06 to 227.23 with a mean of 171.58.
- **Unemployment**: Ranges from 3.88% to 14.31% with a mean of 8%.

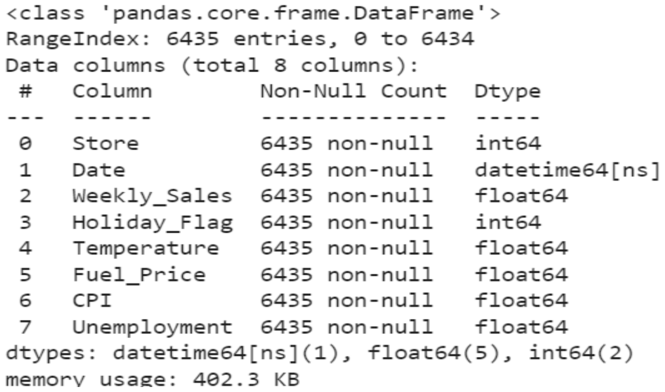
4. Data Pre-processing Steps and Insights

1. Data Types:

- Convert the 'Date' column to a datetime format.
- Confirmed data types: datetime for 'Date', float64 for numerical columns, and int64 for 'Store' and 'Holiday_Flag'.

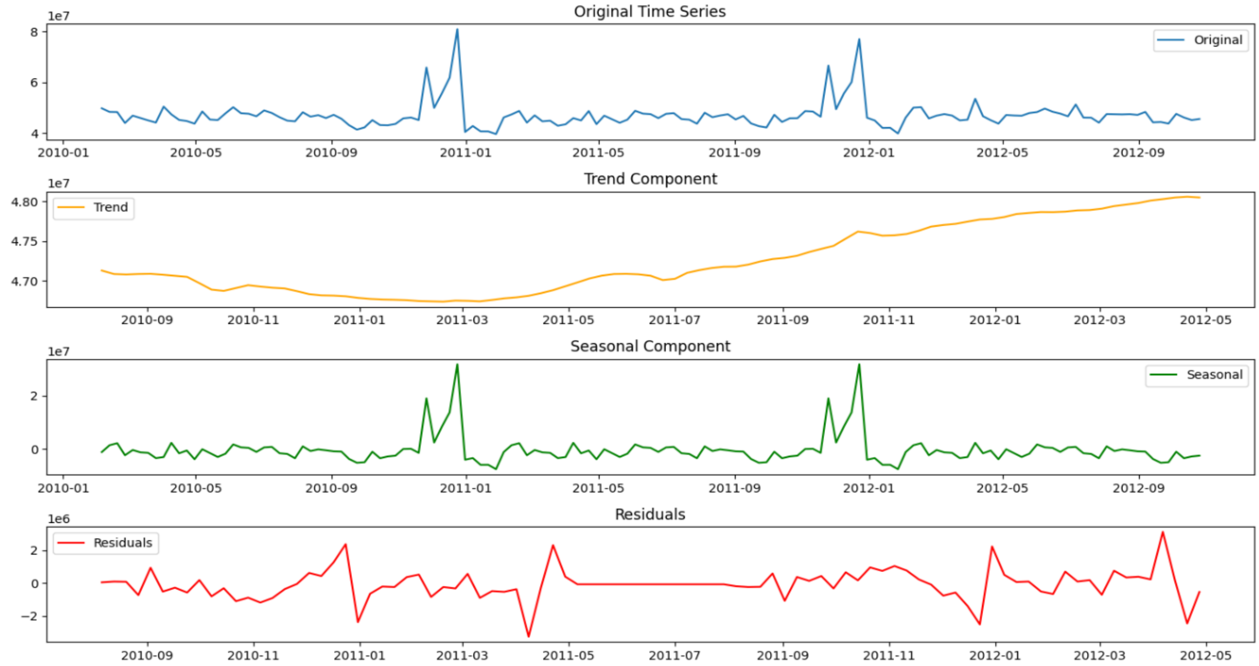
2. Missing Values:

- No missing values found, ensuring data completeness.

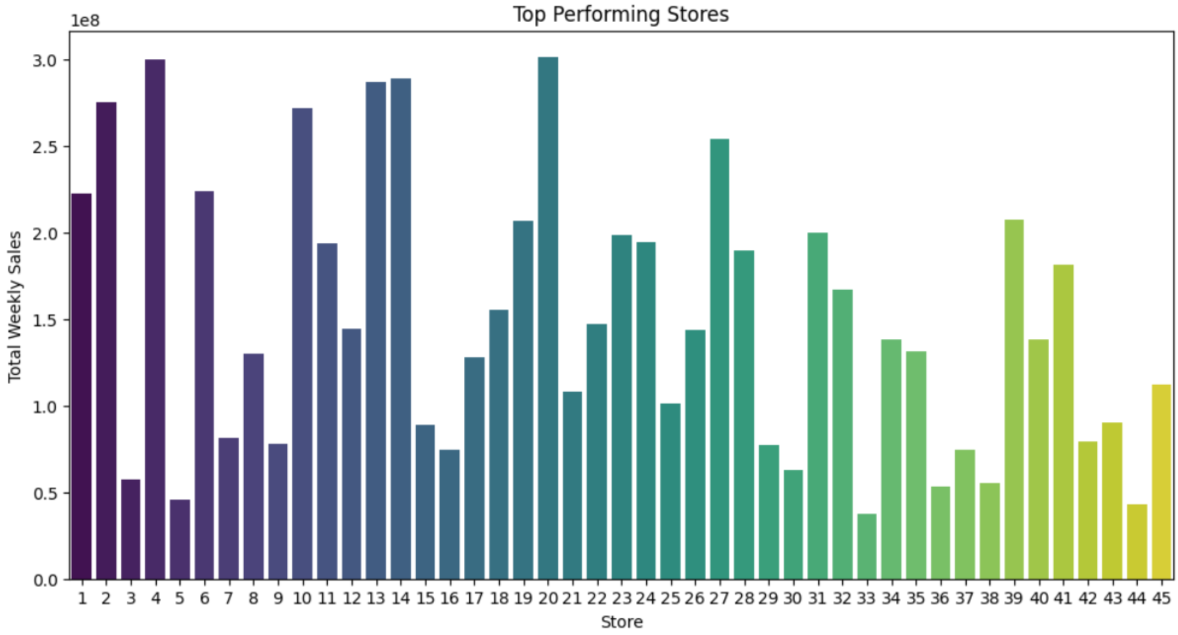


```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6435 entries, 0 to 6434
Data columns (total 8 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   Store         6435 non-null   int64
 1   Date          6435 non-null   datetime64[ns]
 2   Weekly_Sales  6435 non-null   float64
 3   Holiday_Flag  6435 non-null   int64
 4   Temperature   6435 non-null   float64
 5   Fuel_Price    6435 non-null   float64
 6   CPI           6435 non-null   float64
 7   Unemployment  6435 non-null   float64
dtypes: datetime64[ns](1), float64(5), int64(2)
memory usage: 402.3 KB
```

3. Aggregated weekly sales to observe the overall trend.
4. Set the frequency of the date column to weekly.



5. **Top Performing Stores:**
   - Identified stores with the highest total weekly sales.
   - Store 20 and Store 4 are notable examples based on the analysis.
6. **Worst Performing Stores:**
   - Stores with the lowest total weekly sales.
   - Store 33 was identified as the worst-performing store in the analysis.



```
Worst Performing Store: 33.0 with total sales: 37160221.96
Difference between highest and lowest performing stores: 264237570.49999997
```

5. Choosing the Algorithm for the Project

The SARIMAX (Seasonal Autoregressive Integrated Moving Average with Exogenous regressors) model is chosen for forecasting future sales. This model is well-suited for time series data with seasonality and trends, and it allows for incorporating external factors (exogenous variables).

6. Motivation and Reasons for Choosing the Algorithm

The SARIMAX model is chosen due to:

- Its ability to handle seasonal patterns in the data.
- Its capability to include external variables that could impact sales.
- Its proven effectiveness in time series forecasting in retail and other domains.

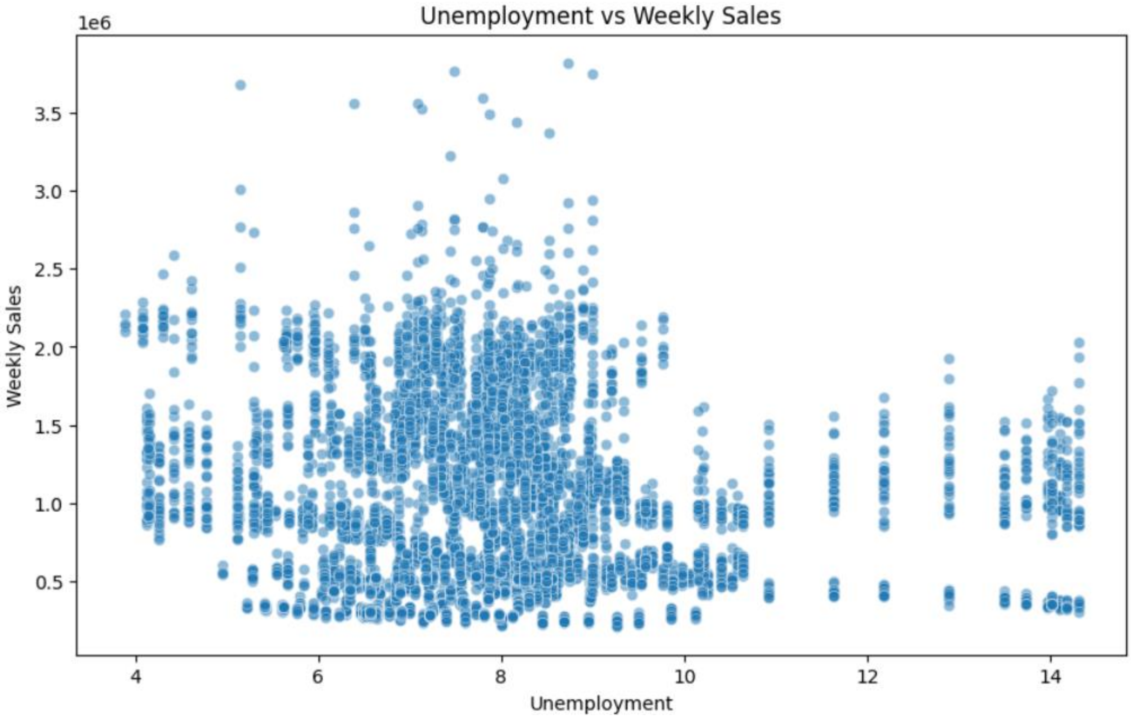
7. Assumptions

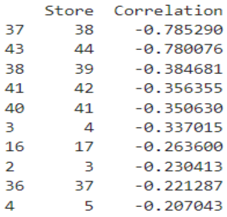
- The dataset's weekly sales follow a seasonal pattern, particularly on a yearly basis.
- The exogenous variables (Temperature, Fuel_Price, CPI, Unemployment) impact the sales trends.
- SARIMAX is well-suited for time series forecasting and has a track record of accuracy in similar applications.

8. Model Evaluation and Techniques

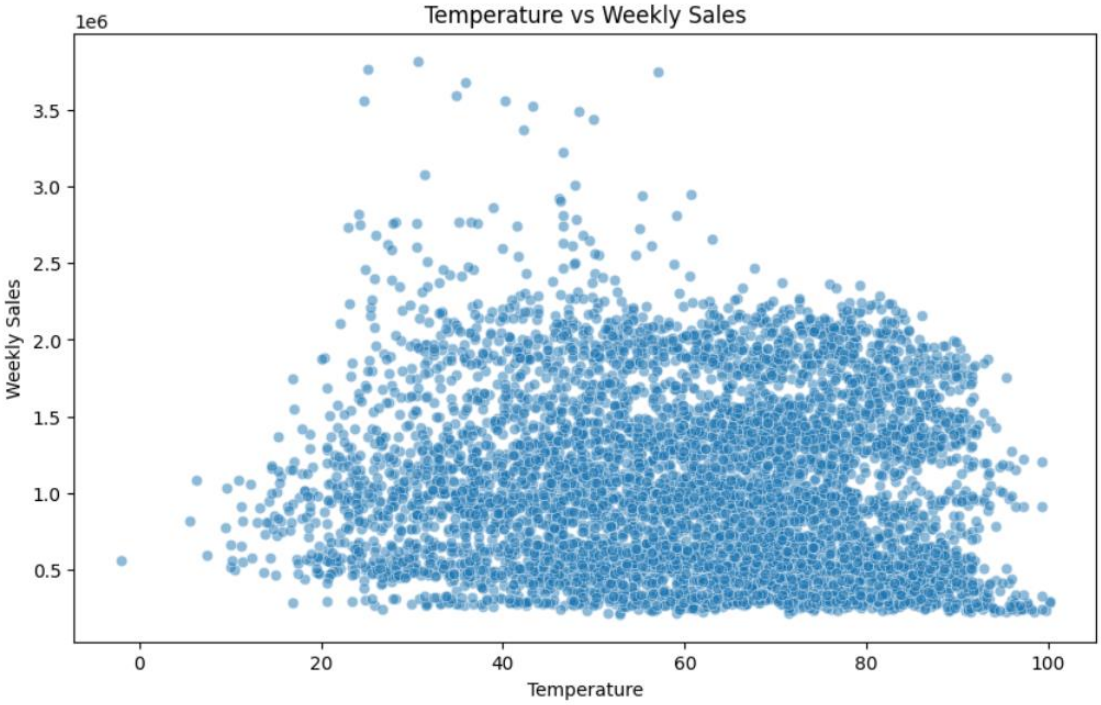
1. **Model Fitting**:
    - The SARIMAX model is fitted to the weekly sales data for each store using the specified ARIMA and seasonal orders.
    - This involves using the SARIMAX class from the statsmodels.tsa.statespace.sarimax module with the specified parameters.
2. **Saving the Fitted Model**:
    - The fitted model is saved using the joblib library for future use and validation.
3. **Forecasting**:
    - The fitted model is used to forecast the next 12 weeks of sales.
    - The get_forecast method is used to generate these forecasts, and confidence intervals are calculated to understand the uncertainty of the forecasts.
4. **Visualization of Forecasts**:
    - The observed sales data along with the forecasted values and their confidence intervals are plotted to visually inspect the model's performance.

5. **Correlation Analysis**:
    - The correlation between unemployment and weekly sales, temperature and weekly sales, and CPI and weekly sales is calculated to understand the impact of these factors on sales.

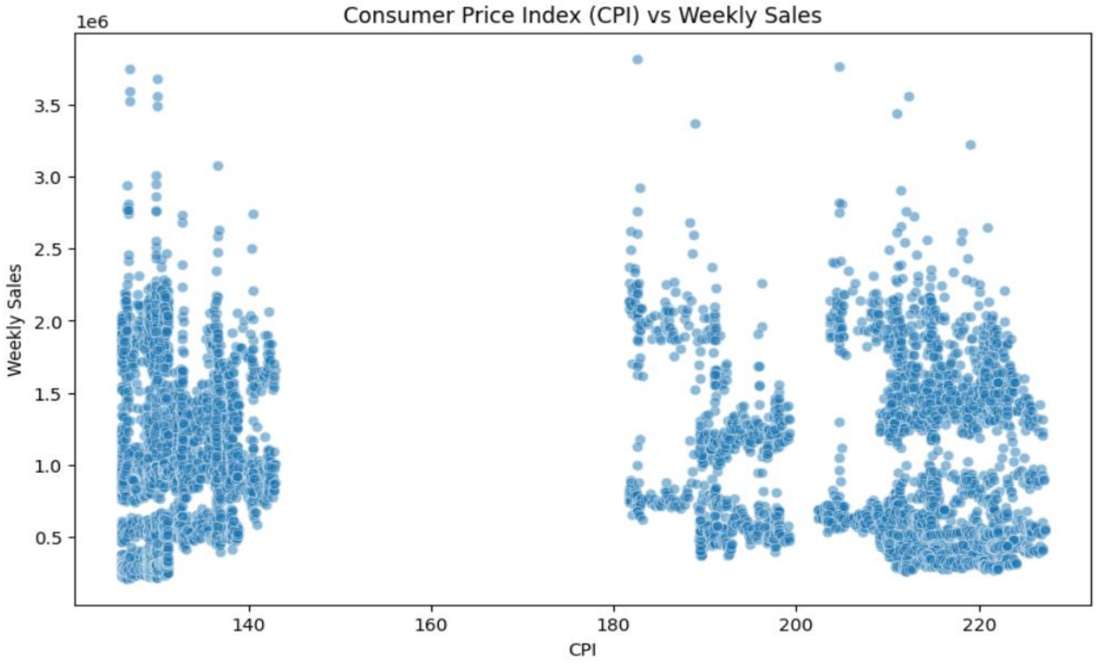




```
    Store  Correlation
37     38    -0.785290
43     44    -0.780076
38     39    -0.384681
41     42    -0.356355
40     41    -0.350630
3       4    -0.337015
16     17    -0.263600
2       3    -0.230413
36     37    -0.221287
4       5    -0.207043
```



```
Correlation between Temperature and Weekly Sales: -0.06381001317946962
```



```
Correlation between CPI and Weekly Sales: -0.07263416204017631
```

6. **Trend and Seasonality Analysis:**

- Seasonal decomposition of the total weekly sales data is performed to identify trends and seasonal patterns.

9. Inferences from the EDA

1. Unemployment vs. Weekly Sales:

- A significant negative correlation between unemployment rates and weekly sales was observed for some stores, indicating that higher unemployment leads to lower sales. Stores most affected include Store 38 and Store 44.

2. Temperature vs. Weekly Sales:

- The correlation between temperature and weekly sales was weakly negative (-0.064), suggesting that temperature has a minor impact on sales.

3. CPI vs. Weekly Sales:

- The correlation between CPI and weekly sales was also weakly negative (-0.073), implying limited direct influence on sales.

4. Trend and Seasonality Analysis:

- Seasonal decomposition revealed clear trends and seasonal patterns in the sales data, with strong seasonal peaks indicating periods of high sales.

7. **Top Performing and Worst Performing Stores:**

- The analysis identified the top-performing stores by total weekly sales, with a significant difference in sales between the highest and lowest-performing stores. Store 33 was the worst-performing store.

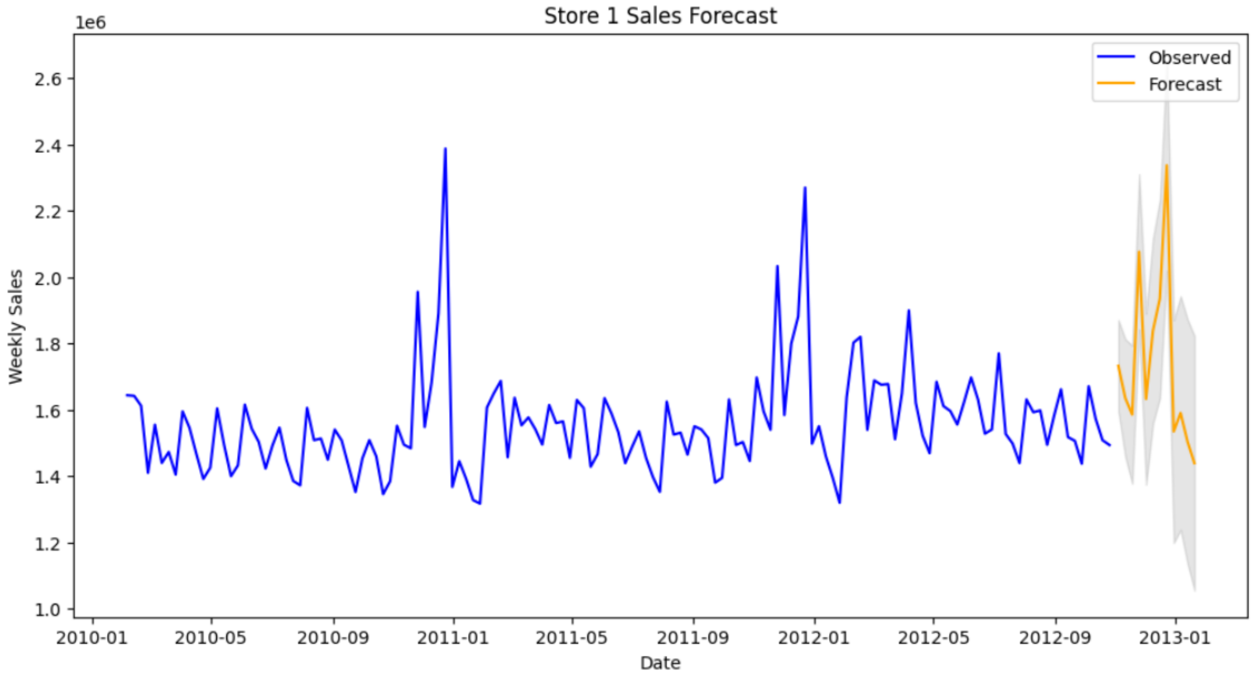
10. Future Possibilities of the Project

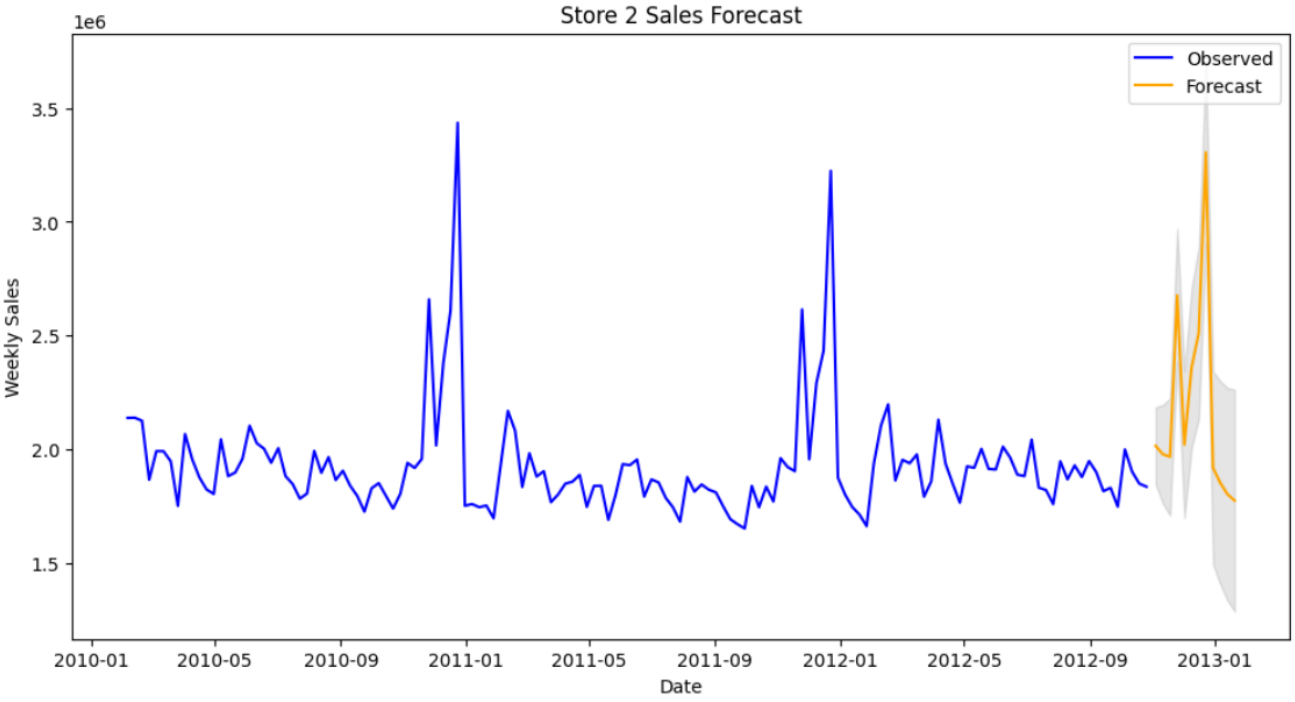
1. **Incorporate Additional Variables:**
    - Integrate more variables such as promotions, holidays, and local economic factors to improve the forecasting model’s accuracy.
2. **Store-specific Strategies:**
    - Develop tailored strategies for each store based on their unique sales patterns and correlations with external factors.
3. **Advanced Machine Learning Models:**
    - Explore more sophisticated machine learning techniques, such as ensemble methods or deep learning, to enhance forecasting accuracy.

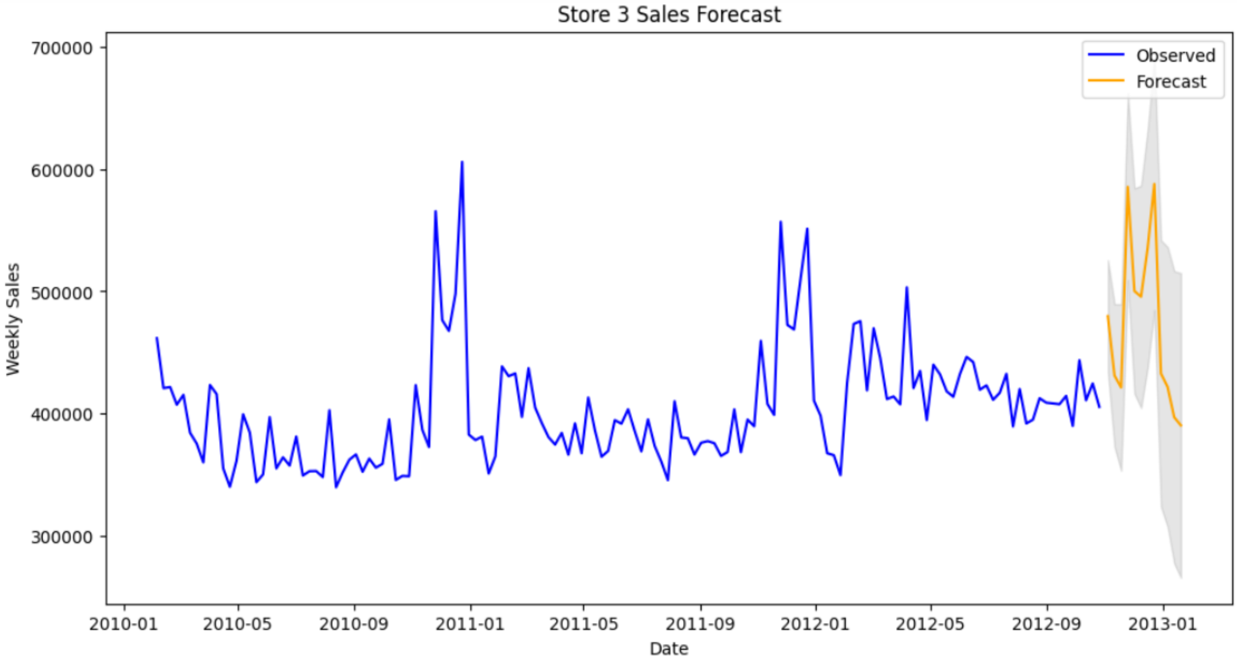
Sales Forecasting for Next 12 Weeks

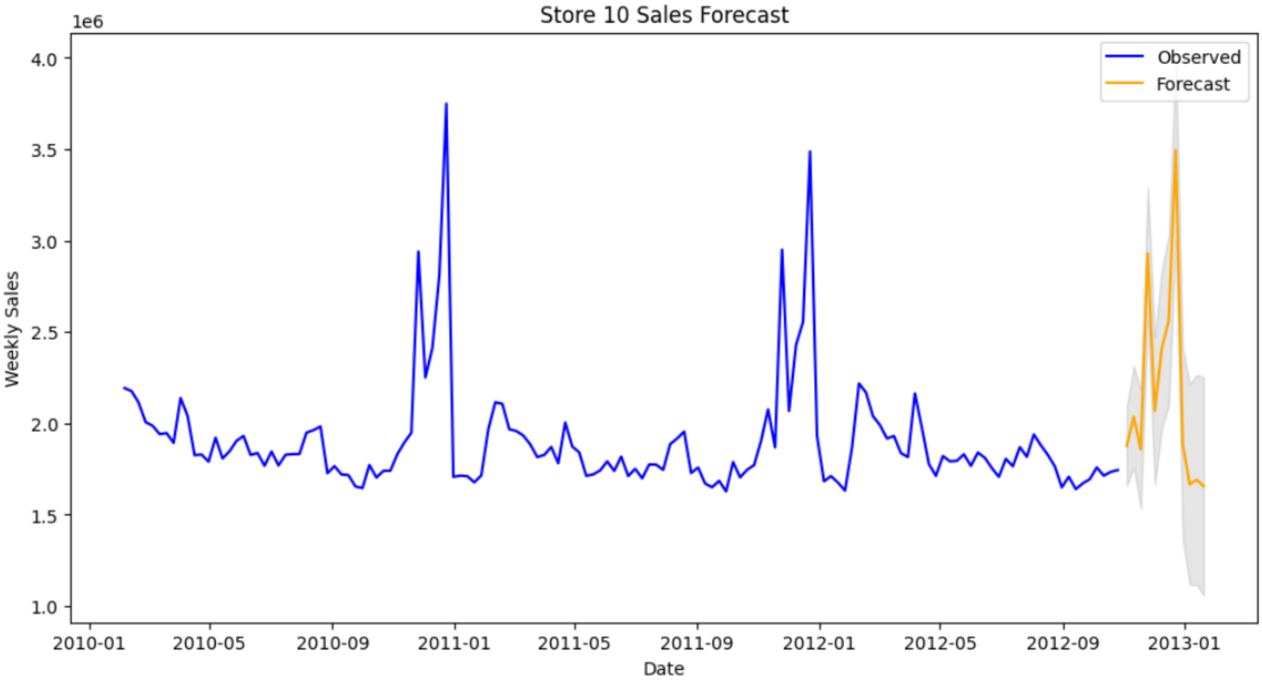
- Using the SARIMAX model, sales for each store were forecasted for the next 12 weeks. The model was fitted to the historical sales data, incorporating seasonal patterns and external variables. The forecasts provide valuable insights for inventory management, helping stores plan inventory, staffing, and marketing efforts effectively.

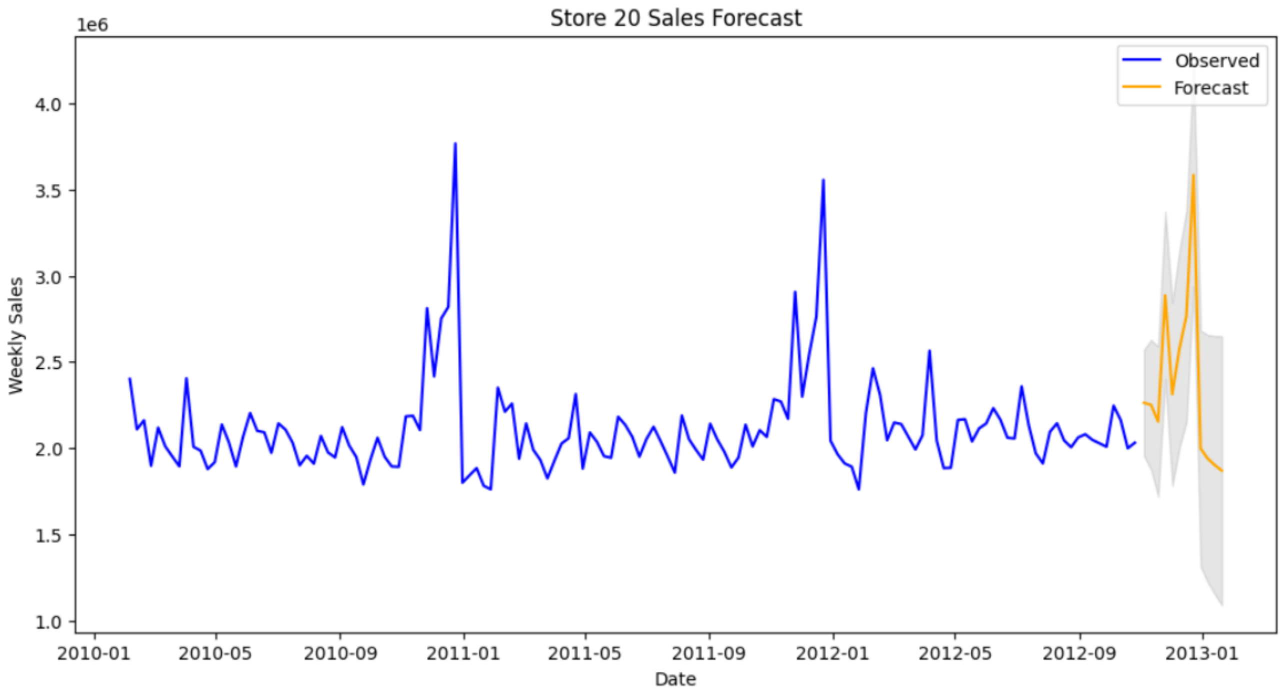
Attaching Sales Forecast of few Stores: -

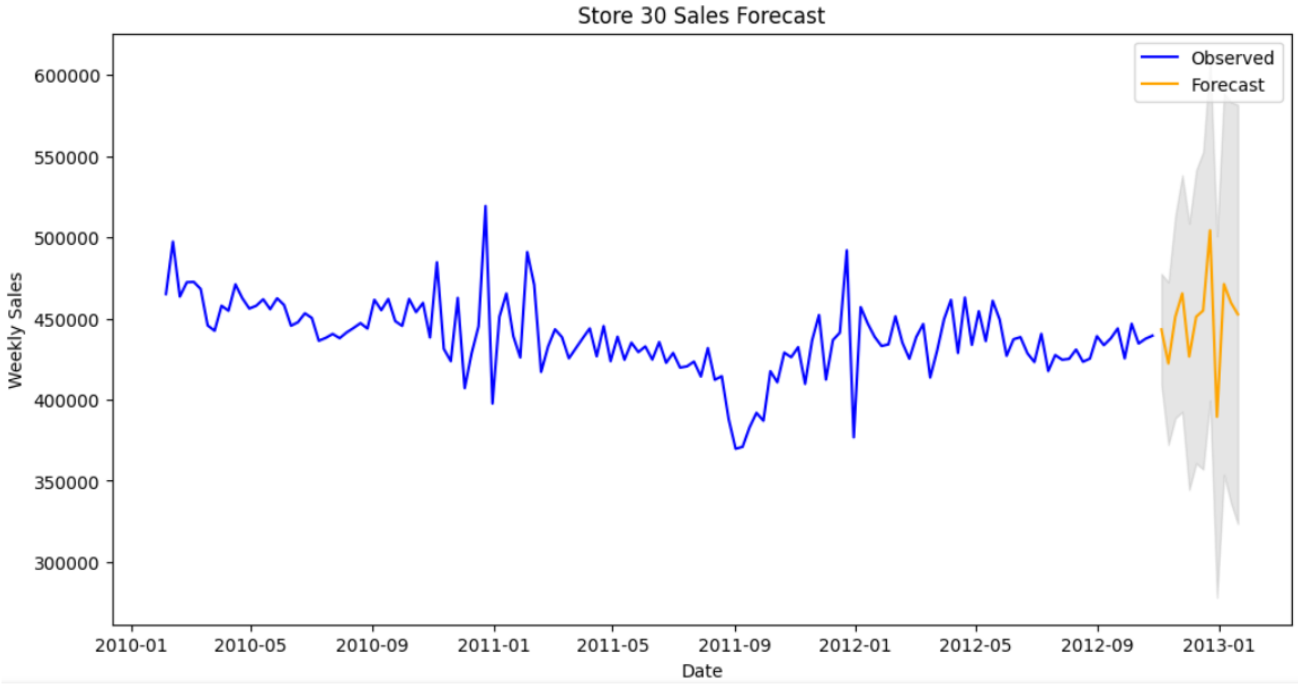


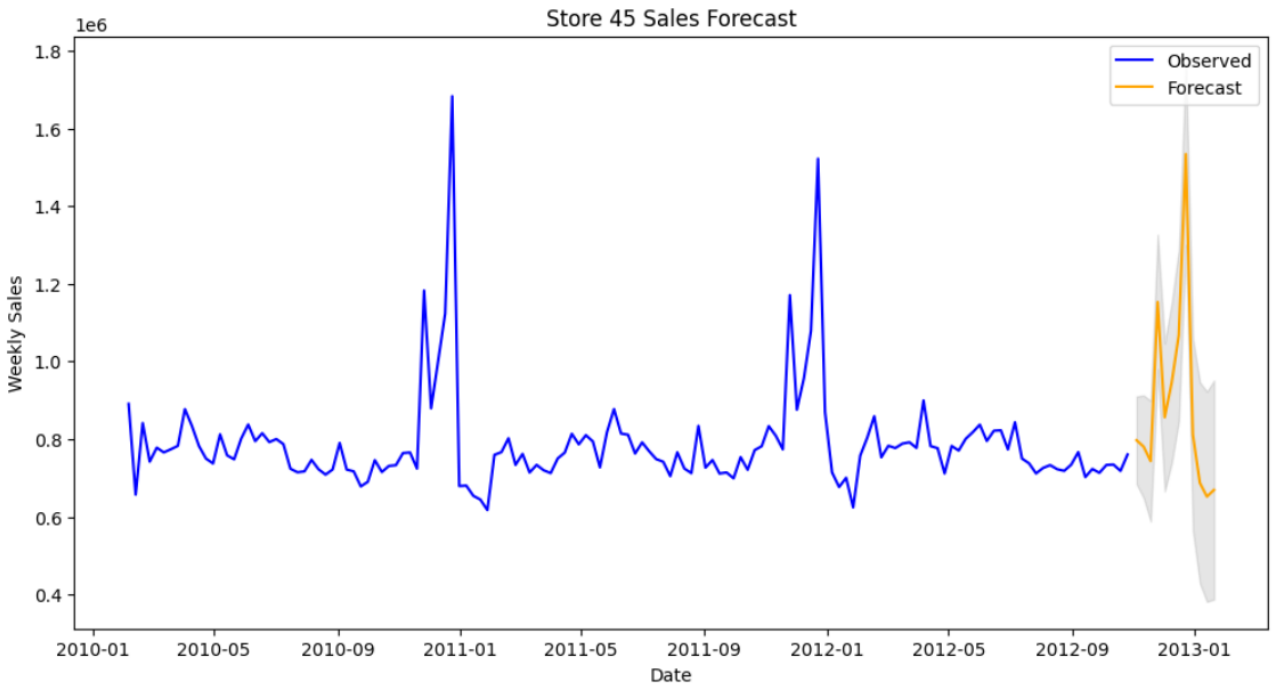


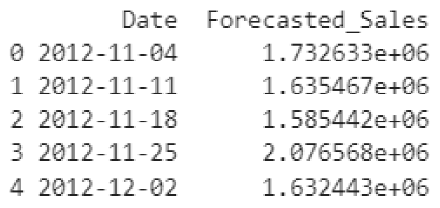

Store 3 Sales Forecast
Observed
Forecast
700000
600000
500000
400000
300000
Weekly Sales
2010-01
2010-05
2010-09
2011-01
2011-05
2011-09
2012-01
2012-05
2012-09
2013-01
Date


Store 10 Sales Forecast
1e6
Observed
Forecast
4.0
3.5
3.0
2.5
2.0
1.5
1.0
Weekly Sales
2010-01
2010-05
2010-09
2011-01
2011-05
2011-09
2012-01
2012-05
2012-09
2013-01
Date


Store 20 Sales Forecast
1e6
4.0
3.5
3.0
2.5
2.0
1.5
1.0
Weekly Sales
2010-01
2010-05
2010-09
2011-01
2011-05
2011-09
2012-01
2012-05
2012-09
2013-01
Date
Observed
Forecast


Store 30 Sales Forecast
600000
550000
500000
450000
400000
350000
300000
Weekly Sales
2010-01
2010-05
2010-09
2011-01
2011-05
2011-09
2012-01
2012-05
2012-09
2013-01
Date
Observed
Forecast



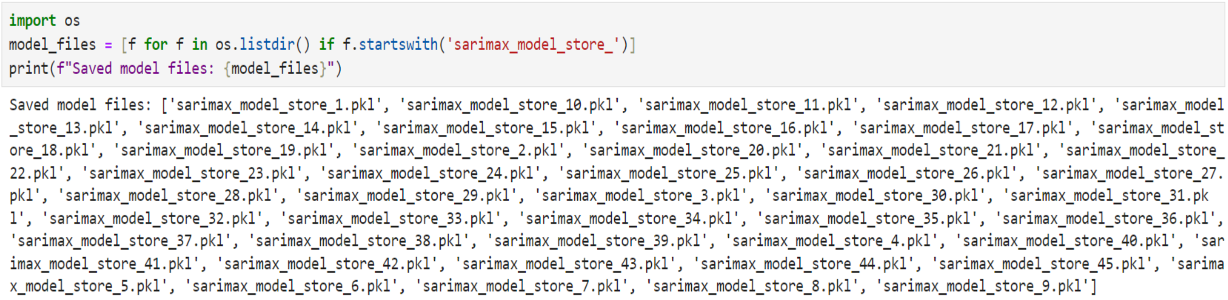


```
        Date  Forecasted_Sales
0 2012-11-04      1.732633e+06
1 2012-11-11      1.635467e+06
2 2012-11-18      1.585442e+06
3 2012-11-25      2.076568e+06
4 2012-12-02      1.632443e+06
```

11. Insights for Store Improvement

1. **Stores with High Negative Correlation to Unemployment:**
   - Stores like Store 38 and Store 44 should focus on strategies to counteract the effects of high unemployment rates, such as localized promotions or community engagement programs.
2. **Seasonal Trends:**
   - Utilize peak sales periods identified from the seasonal decomposition for targeted marketing campaigns and inventory planning.
3. **Temperature and CPI Insights:**
   - Although the correlations with temperature and CPI are weak, these factors should still be monitored as part of a broader strategy for understanding sales drivers.
4. **Holiday Sales:** Weeks with holidays have distinct sales patterns that should be accounted for in planning and forecasting.

Saved Model Files:



```
import os
model_files = [f for f in os.listdir() if f.startswith('sarimax_model_store_')]
print(f"Saved model files: {model_files}")
```

```
Saved model files: ['sarimax_model_store_1.pkl', 'sarimax_model_store_10.pkl', 'sarimax_model_store_11.pkl', 'sarimax_model_store_12.pkl', 'sarimax_model
_store_13.pkl', 'sarimax_model_store_14.pkl', 'sarimax_model_store_15.pkl', 'sarimax_model_store_16.pkl', 'sarimax_model_store_17.pkl', 'sarimax_model_st
ore_18.pkl', 'sarimax_model_store_19.pkl', 'sarimax_model_store_2.pkl', 'sarimax_model_store_20.pkl', 'sarimax_model_store_21.pkl', 'sarimax_model_store_
22.pkl', 'sarimax_model_store_23.pkl', 'sarimax_model_store_24.pkl', 'sarimax_model_store_25.pkl', 'sarimax_model_store_26.pkl', 'sarimax_model_store_27.
pkl', 'sarimax_model_store_28.pkl', 'sarimax_model_store_29.pkl', 'sarimax_model_store_3.pkl', 'sarimax_model_store_30.pkl', 'sarimax_model_store_31.pk
l', 'sarimax_model_store_32.pkl', 'sarimax_model_store_33.pkl', 'sarimax_model_store_34.pkl', 'sarimax_model_store_35.pkl', 'sarimax_model_store_36.pkl',
'sarimax_model_store_37.pkl', 'sarimax_model_store_38.pkl', 'sarimax_model_store_39.pkl', 'sarimax_model_store_4.pkl', 'sarimax_model_store_40.pkl', 'sar
imax_model_store_41.pkl', 'sarimax_model_store_42.pkl', 'sarimax_model_store_43.pkl', 'sarimax_model_store_44.pkl', 'sarimax_model_store_45.pkl', 'sarima
x_model_store_5.pkl', 'sarimax_model_store_6.pkl', 'sarimax_model_store_7.pkl', 'sarimax_model_store_8.pkl', 'sarimax_model_store_9.pkl']
```

12. Conclusion

- The project successfully developed a SARIMAX-based predictive model to forecast Walmart's weekly sales for the next 12 weeks. The analysis provided valuable insights into the relationship between sales and external factors such as unemployment and CPI, aiding in better inventory management and operational efficiency.

13. References

a. ***Walmart Sales Data*** *(provided dataset) - Internal dataset used for analysis and forecasting.*

b. ***Statsmodels contributors****. (2023).* Statsmodels: Statistics in Python - SARIMAX. *Retrieved from https://www.statsmodels.org/stable/generated/statsmodels.tsa.statespace.sarimax.SARIMAX.html*

c. ***joblib****.* Tools for parallel computing with Python *from https://joblib.readthedocs.io/en/latest/*

d. ***Seaborn****.* Seaborn: Statistical data visualization. *Retrieved from https://seaborn.pydata.org/*

e. ***Matplotlib****.* Matplotlib: Visualization with Python. *Retrieved from https://matplotlib.org/*

f. ***McKinney, W. (2017).*** Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython. *O'Reilly Media.*

g. ***OpenAI****.* ChatGPT Documentation. *Retrieved from https://www.openai.com*