Capstone Project

Walmart Weekly Sales Forecasting

Submitted By
Chetan D. Chaudhari

Contents

Introduction:

Sales forecasting may sound like a technical term; however, its significance is beyond the domain of numbers and data. It's a pivotal tool that guides businesses towards sustainable growth and success. Imagine it as a crystal ball that helps organizations make informed decisions, allocate resources wisely, and adapt to a constantly changing marketplace.

Some of the applications of sales forecasting are as:

Anticipating the Future

Sales forecasting is essentially a way to peek into the future. By analyzing historical sales data, market trends, and other relevant factors, businesses can make educated guesses about what lies ahead. This allows companies to prepare for both sunny days and storms, ensuring they are never caught off guard.

Resource Allocation

In business, resources like time, money, and manpower are precious commodities. Sales forecasting helps in allocating these resources efficiently. For instance, if a surge in demand is predicted for a particular product, a company can ramp up production and marketing efforts to meet that demand without overextending itself in areas where demand might be stagnant.

Inventory Management

For retailers, keeping the right amount of stock on hand is a constant challenge. Too much inventory ties up capital, while too little can result in missed sales opportunities. Sales forecasting helps strike that balance by estimating how much stock will be needed in the coming weeks or months.

Financial Planning

For a non-technical person, this might be the most compelling reason to care about sales forecasting. It's about dollars and cents. Accurate sales forecasts allow companies to plan their finances better. They can budget for expenses, set revenue targets, and secure financing if needed.

Strategic Decision Making

Sales forecasts inform strategic decisions. For instance, if a forecast indicates a slowdown in a particular market segment, a company might choose to pivot its strategy or explore new markets. On the other hand, if a product is predicted to be a hit, a business can invest more resources in its development and promotion.

Customer Satisfaction

While it may not be immediately obvious, sales forecasting can benefit customers too. When a business accurately predicts demand, it's better equipped to meet customer needs. This leads to fewer stockouts, faster delivery times, and improved customer satisfaction.

Competing Effectively

In the cutthroat world of business, staying ahead of the competition is crucial. Accurate sales forecasting can give a business a competitive edge by allowing it to respond quickly to market changes, adapt to consumer preferences, and outmaneuver rivals.

Long-Term Viability

Finally, sales forecasting is about securing the long-term future of a company. It helps businesses make smart decisions today to ensure they thrive tomorrow. It's about not just surviving but thriving in a world where change is the only constant.

In a nutshell, sales forecasting isn't just about numbers and graphs; it's about the heartbeat of a business. It's about making sure that the products and services you love are available when you want them, that your favorite companies stay in business, and that the economy keeps humming along.

In today's fast-paced and competitive retail landscape, the ability to forecast sales accurately is paramount to a retailer's success. This project report addresses the critical challenge faced by Walmart, one of the world's largest retail giants, in predicting weekly sales for its vast network of stores. By leveraging advanced data analysis and machine learning techniques, this report aims to develop a robust forecasting model that takes into account the unique characteristics of each store, seasonal variations, and the influence of external factors. The ultimate goal is to equip Walmart with a powerful tool for informed decision-making, resource optimization, and enhanced customer satisfaction.

Tools Used

- In tackling the complicated challenge of forecasting sales for Walmart stores and addressing the associated questions, I leveraged a powerful toolkit of data science and analytics tools.
- Python3 served as our primary programming language, providing versatility and an extensive ecosystem of libraries.
- Pandas was instrumental for data manipulation and preprocessing, enabling us to efficiently clean, organize, and structure the dataset for analysis.
- For data visualization, I utilized the capabilities of Matplotlib and Seaborn, crafting informative charts and graphs to gain insights and communicate findings effectively.
- Statsmodels played a pivotal role in conducting statistical analyses and building time series models to capture sales trends.

Together, these tools empowered us to explore, model, and forecast weekly sales data, ultimately equipping us to provide valuable insights and predictions to Walmart for data-driven decision-making.

Problem Statement

The primary objective of this project is to create a robust and accurate forecasting model that predicts the weekly sales for Walmart stores. This model will enable Walmart to make informed decisions regarding inventory stocking, workforce management, and sales strategies, ultimately improving operational efficiency and customer experience.

Key Challenges:

- **Data Volume and Variety:** Walmart collects a vast amount of data, including historical sales data, promotional activities, economic indicators, and store-specific information. Managing and integrating diverse data sources pose a challenge.
- **Seasonality and Trends:** Retail sales exhibit seasonality and trends influenced by holidays, weather, and consumer behavior. Capturing these patterns is essential for precise forecasting.
- **Store Heterogeneity:** Each Walmart store operates in a unique environment, serving diverse customer demographics. Store-specific characteristics, such as location, size, and assortment, must be considered in the forecasting model.
- **External Factors:** External variables like economic indicators, local events, and competitive factors can impact sales. Integrating these external variables into the model is necessary for accuracy.
- **Data Quality and Missing Values:** Ensuring data quality and handling missing values are critical to prevent bias in the forecasting model.

Project Goals:

The project aims to achieve the following objectives:

- Explore and analyze the data to identify patterns, seasonality, and trends specific to each store.
- Develop and train machine learning models capable of accurately forecasting weekly sales for Walmart stores.
- Evaluate the model's performance using appropriate metrics and validate it against historical data.
- Incorporate external factors that influence sales into the forecasting model to enhance accuracy.
- Provide actionable insights and visualizations to assist Walmart's decision-makers in using the forecasts for inventory management and sales strategies.

Deliverables:

The project's final deliverables will include:

A well-documented dataset with cleaned and preprocessed data.

- Trained machine learning models capable of generating weekly sales forecasts for each Walmart store.

- Detailed performance metrics and validation results for the forecasting model.
- Visualizations and insights to aid decision-making.

Success Criteria:

The project will be considered successful if it produces a forecasting model that demonstrates:

- High accuracy in predicting weekly sales for Walmart stores.
- The ability to capture seasonality, trends, and store-specific factors.
- Robustness in handling variations caused by external factors.

By addressing these challenges and achieving the stated objectives, this project will empower Walmart to make data-driven decisions, optimize resources, and enhance the overall shopping experience for its customers while maintaining its position as a leader in the retail industry.

Project Objectives

Identify Unemployment Rate Impact on Sales:

- Determine if the weekly sales of Walmart stores are significantly affected by the unemployment rate.
- Identify the specific stores that are most impacted by changes in the unemployment rate.

Analyze Seasonal Trends in Sales:

- Investigate the presence of seasonal patterns in weekly sales data.
- Determine when these seasonal trends occur and investigate the reasons behind them, such as holidays or other factors.

Evaluate Temperature's Influence on Sales:

- Assess the relationship between temperature and weekly sales for Walmart stores.
- Determine if temperature fluctuations have a significant impact on sales.

Study Consumer Price Index (CPI) Impact:

- Analyze how variations in the Consumer Price Index (CPI) affect the weekly sales of different stores.
- Identify stores that are particularly sensitive to CPI changes.

Identify Top-Performing Stores:

- Identify and rank the top-performing Walmart stores based on historical sales data.
- Determine which stores consistently outperform others.

Identify the Worst-Performing Store:

- Identify the store with the lowest historical sales performance.
- Evaluate the significance of the difference in sales between the worst-performing store and the highest-performing store.

Develop Sales Forecasting Models:

- Utilize predictive modeling techniques to develop sales forecasting models for each of the 45 Walmart stores.
- Forecast sales for the next 12 weeks for each store using these models.

Validate Forecast Accuracy:

- Validate the accuracy of the developed sales forecasting models by comparing forecasted sales with actual sales data.
- Ensure that the models perform well in predicting sales for each store.

Provide Actionable Insights:

- Offer actionable insights and recommendations based on the analysis of unemployment, seasonal trends, temperature, CPI, and store performance.
- Assist Walmart in making data-driven decisions to improve sales and operations.

Documentation and Reporting:

- Document the entire project, including data analysis, model development, and forecasting.
- Prepare a comprehensive report and presentation that communicate the findings, insights, and sales forecasts effectively.

These project objectives guide the analysis and modeling efforts to address the specific questions raised and provide valuable insights to support Walmart's sales forecasting and decision-making processes.

Project Methodology:

The project methodology serves as the roadmap for our investigation and analysis. It outlines the systematic approach I followed to answer critical questions related to sales forecasting, seasonal trends, and the impact of various factors on Walmart stores' weekly sales. The methodology involved data exploration, statistical analysis, predictive modeling, and validation to ensure robust and actionable results. By adhering to this structured approach, I aimed to provide Walmart with informed insights and accurate sales forecasts, empowering data-driven decision-making for the future.

Data Collection and Understanding:

- Gathered and consolidated the historical sales data for the 45 stores.
- Familiarized with the dataset's structure and variables.
- Understood the context and significance of each variable, including Weekly_Sales, Store Number, Whether Holiday or Not, Temperature, Fuel_Price, Consumer Price Index, and Unemployment.

Data Preprocessing:

- Handled missing values in the dataset using appropriate techniques such as imputation.
- Checked for outliers and considered whether to treat or remove them based on domain knowledge.
- Explored the dataset's statistical properties and distributions.

Exploratory Data Analysis (EDA):

- Performed EDA to gain insights into the data and identified patterns and correlations.
- Visualized the data using charts and graphs and understood relationships between variables.
- Analyzed the impact of holidays, temperature, fuel prices, CPI, and unemployment on weekly sales.

Feature Engineering:

- Create relevant features or transformations that improved the forecasting process.

Data Splitting:

- Data splitting was carried out, since the time series is ordered sequence splitting was carried out without shuffling.

Model Selection and Building:

- Based on exploratory data analysis I selected time series models for the given dataset:
- Time series models included Naïve Historical Mean forecasting, Triple Exponential Smoothing and SARIMA.
- The decision for selecting these particular models was based on the time series decomposition.

Model Evaluation:

- The trained models were evaluated using Mean Absolute Percentage Error (MAPE), and Root Mean Squared Error (RMSE).
- MAPE is a measure of the accuracy of a forecasting model and represents the average percentage difference between predicted and actual values. It quantifies the magnitude of forecasting errors as a percentage of the actual values, making it a valuable metric for assessing the overall accuracy of a model.
- RMSE is a metric that quantifies the average magnitude of errors between predicted and actual values, providing a measure of the model's precision. RMSE calculates the square root of the average squared differences between predicted and actual values, making it sensitive to large errors. It is commonly used to assess the goodness of fit of a forecasting model.

Hyperparameter Tuning:

- Hyperparameters of Triple Exponential Smoothing (Holt-Winters) included alpha (smoothing of level), beta (smoothing of trend), gamma (smoothing of seasonal component), seasonal period (m), and an optional damping parameter (φ).
- These hyperparameters for Triple Exponential Smoothing method were optimized setting inbuilt parameter (optimized = True) for **'ExponentialSmoothing'** function of **'statsmodels.tsa.api'** library.

```
TES_model = ExponentialSmoothing(train, trend = 'additive',seasonal= 'additive').fit(optimized= True)
```

- Hyperparameters of SARIMA (Seasonal AutoRegressive Integrated Moving Average) included p (order of autoregressive component), d (degree of differencing), q (order of moving average component), P (seasonal autoregressive component), D (seasonal degree of differencing), Q (seasonal order of moving average component), and s (seasonal period).
- The degree of differencing (d) and seasonal degree of differencing were tunned by carrying adfuller statistical test. Whereas, other parameters were tuned by exhaustive searching and AIC was used as a selection criterion for the searched parameters.

Validation on Testing Data:

- All three models were validated on the testing dataset to assess their generalized performance.
- The model forecasts were compared against actual weekly sales.

Interpretation and Insights:

- Interpret the results and provide actionable insights to Walmart. Understand the impact of different factors on sales and how they can be leveraged for business decisions.

Documentation and Reporting:

- The whole process I followed for the project is documented in a report.

This methodology provided a structured approach to tackle the problem of forecasting weekly sales for Walmart stores.

Data Description

The provided dataset consists of historical sales data for 45 stores located in various regions across the country. This dataset was collected as part of an assignment from Intellipaat and contains information on several key variables. Here is a brief description of each variable:

1. **Store:** This variable represents the unique identifier for each of the 45 stores included in the dataset. Each store is assigned a distinct number to differentiate it from the others.
2. **Weekly_Sales:** This is the target variable and represents the weekly sales figure for each store. It indicates the total sales revenue generated by each store in a given week. This variable is of primary interest for analysis and forecasting.
3. **Holiday_Flag:** This binary variable indicates whether a given week includes a holiday (1) or not (0). Holidays can significantly impact consumer behavior and sales, so this variable is crucial for understanding sales patterns.
4. **Temperature:** This variable represents the average temperature during the week. Temperature can influence sales, especially for seasonal products or items like clothing and groceries.
5. **Fuel_Price:** Fuel prices can affect transportation costs, which, in turn, can impact product pricing and consumer spending. This variable provides information on the weekly fuel prices.
6. **Consumer Price Index (CPI):** CPI is an economic indicator that measures changes in the average prices paid by consumers for a basket of goods and services. It helps in understanding inflation and its potential impact on consumer behavior.
7. **Unemployment:** This variable represents the unemployment rate during the week. High unemployment rates can impact consumer confidence and spending patterns.

A snapshot of data info:

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 6435 entries, 2010-02-05 to 2012-10-26
Data columns (total 7 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   Store         6435 non-null   int64
 1   Weekly_Sales  6435 non-null   float64
 2   Holiday_Flag  6435 non-null   int64
 3   Temperature   6435 non-null   float64
 4   Fuel_Price    6435 non-null   float64
 5   CPI           6435 non-null   float64
 6   Unemployment  6435 non-null   float64
dtypes: float64(5), int64(2)
memory usage: 402.2 KB
```

Figure 1 Data info

Data was collected from '2010-02-05' to '2012-10-26', spanning almost three years, and with 143 observations corresponding to each week of the year for 2010, 2011, and 2012, we have a comprehensive dataset that captures seasonal variations, trends, and potential year-over-year changes in sales patterns for each store.

The dataset appears to be designed to analyze the relationships between these variables and the weekly sales figures of the 45 stores. It can be used for various data analysis and machine learning tasks, such as sales forecasting, understanding the impact of holidays and economic factors on sales, and identifying trends and patterns that can inform business strategies for each store.

Before conducting any analysis or modeling, it is essential to perform data preprocessing, including handling missing values, scaling, and potentially feature engineering, to prepare the data for meaningful insights and predictions.

Data Preprocessing

Before we proceed further to Data Preprocessing step, data must be imported and stored in suitable data structure. So that data exploration will be easy.

Data importing in a Pandas DataFrame was the initial step, where I efficiently loaded the provided historical sales data, incorporating features like store number, weekly sales, and economic indicators into a structured tabular format. This DataFrame served as the foundation for subsequent data manipulation and analysis.

```
df = pd.read_csv("Walmart DataSet.csv",   # file_name
                 parse_dates= ['Date'],   # parse 'Date' column as a Datetime datatype
                 dayfirst= True,          # format of date has 1st value day
                 index_col='Date')        # set index as 'Date'
df.head()
```

Date	Store	Weekly_Sales	Holiday_Flag	Temperature	Fuel_Price	CPI	Unemployment
2010-02-05	1	1643690.90	0	42.31	2.572	211.096358	8.106
2010-02-12	1	1641957.44	1	38.51	2.548	211.242170	8.106
2010-02-19	1	1611968.17	0	39.93	2.514	211.289143	8.106
2010-02-26	1	1409727.59	0	46.63	2.561	211.319643	8.106
2010-03-05	1	1554806.68	0	46.50	2.625	211.350143	8.106

Figure 2 Data Importing code snippet

As mentioned in project methodology section data preprocessing steps deals with

- Handling missing values, if any, in the dataset using appropriate techniques such as imputation.
- Checking for outliers and considering whether to treat or remove them based on domain knowledge.
- Exploring the dataset's statistical properties and distributions.

Handling Missing Values:

Pandas' library provides functions to detect and impute the missing values. This makes the task if handling missing values easier. Figure 3 and Figure 4 provides code snippet of these functions.

As we can observe from figure 3 and figure 4, the data doesn't contain any missing values or the duplicate records.

The absence of missing values and duplicated records in the dataset is a positive sign, indicating the dataset's cleanliness and readiness for analysis.

This means that each record is complete, with no critical information missing, and there are no redundant entries, ensuring data integrity.

```
##checking for null value count
pd.DataFrame(df.isnull().sum(), columns = ['Count'])
```

	Count
Store	0
Weekly_Sales	0
Holiday_Flag	0
Temperature	0
Fuel_Price	0
CPI	0
Unemployment	0

Figure 3 missing value check code snippet

```
##checking for duplicated records
print(f"Do any Duplicated Records are present? --> {df.duplicated().any()}")
```

```
Do any Duplicated Records are present? --> False
```

Figure 4 duplicate record check code snippet

This cleanliness simplifies subsequent data preprocessing steps, reducing the need for complex imputation or deduplication processes, and allows us to focus more directly on exploring relationships and patterns in the data, ultimately leading to more robust and reliable insights and forecasts.

Exploring the Statistical Properties

Pandas'. describe () function gives the summary statistics of the data. Using this function, a summary statistic was obtained for weekly sales across all stores and all observations as shown in figure 5.

```
df['Weekly_Sales'].describe()
```

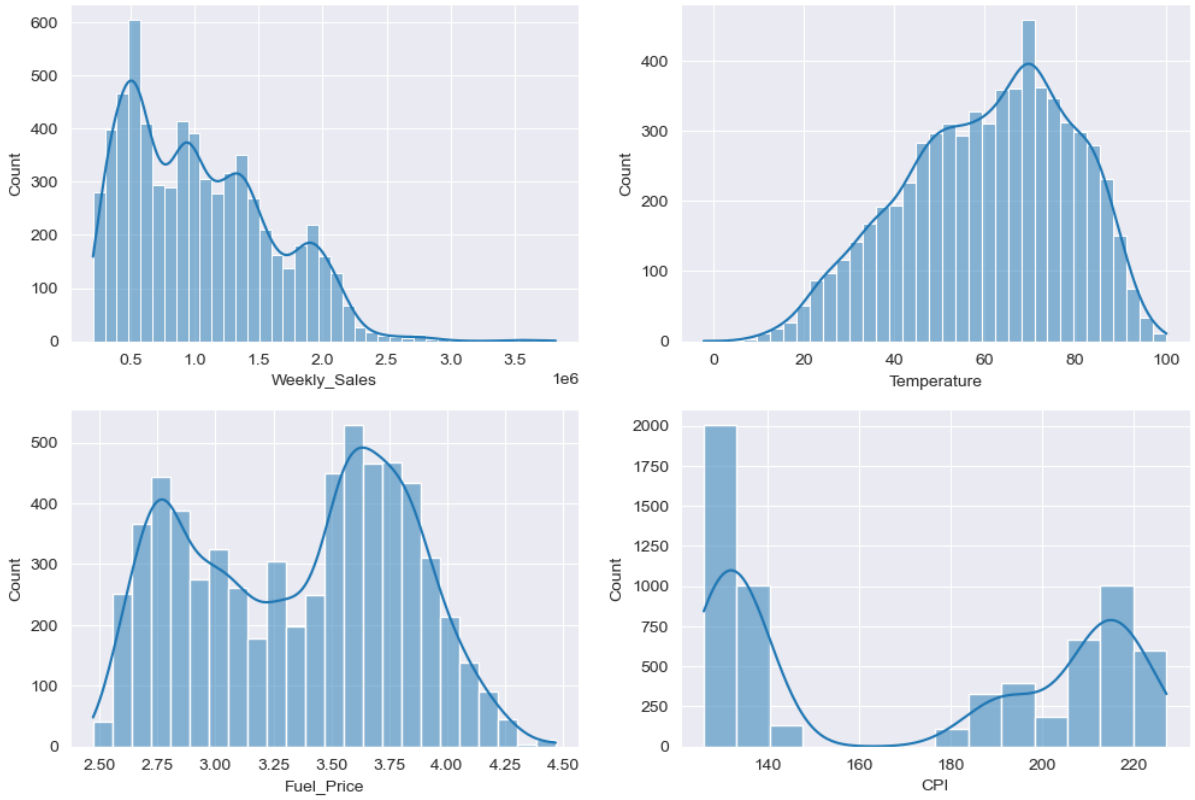
```
count    6.435000e+03
mean     1.046965e+06
std      5.643666e+05
min      2.099862e+05
25%      5.533501e+05
50%      9.607460e+05
75%      1.420159e+06
max      3.818686e+06
Name: Weekly_Sales, dtype: float64
```

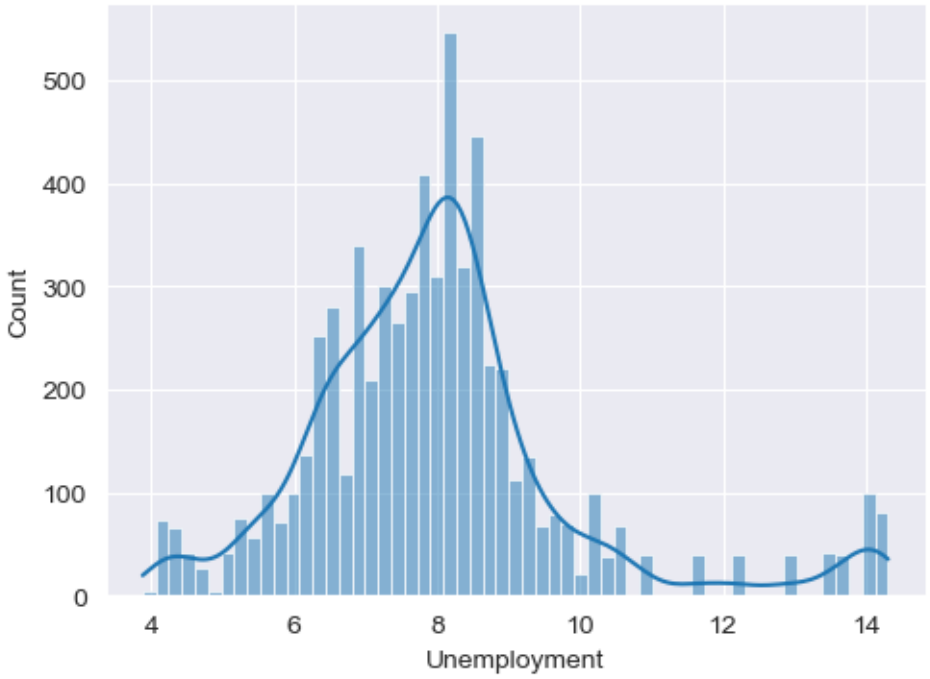
Figure 5 Summary statistic for Weekly Sales

The provided statistics summarize the distribution of weekly sales across all stores and observations combined as:

- **Count:** There are 6,435 observations in the dataset (143 observations for each of the 45 stores), indicating the total number of recorded weekly sales data points.
- **Mean:** The mean (average) weekly sales across all stores is approximately 1,046,965 dollars, offering an insight into the typical sales level.
- **Standard Deviation:** With a standard deviation of approximately 564,366 dollars, there is notable variability in weekly sales, indicating fluctuations and differences between individual store performances.
- **Minimum and Maximum:** The minimum recorded weekly sales value is approximately 209,862 dollars, while the maximum is around 3,818,686 dollars, showcasing the range of sales, with some weeks having exceptionally high or low sales compared to the mean.
- **Quartiles (25%, 50%, 75%):** These quartiles provide insights into the distribution's shape. For instance, the median (50th percentile) weekly sales value is approximately 960,746 dollars, indicating that half of the observations have sales above this value and half below. The 25th percentile (Q1) and 75th percentile (Q3) are approximately 553,501 dollars and 1,420,159 dollars, respectively, helping to assess the spread and skewness of the data distribution.

To explore other numerical variables, I plotted the histograms for respective variables as shown in figure below.





- The histograms offer a visual representation of the distribution of key numerical variables, including 'Weekly_Sales,' 'Temperature,' 'Fuel_Price,' 'CPI' (Consumer Price Index), and 'Unemployment.'
- These histograms effectively illustrate the range of values for each variable within the dataset. However, it's important to note that while histograms showcase the distribution characteristics, they do not inherently reveal the relationships or impacts of these variables on 'Weekly_Sales.'
- To derive insights into how each of these variables influences weekly sales, we will delve deeper into subsequent sections of our analysis.

Feature Engineering

In preparation for our comprehensive data exploration, I recognized the importance of crafting and extracting relevant features essential for our subsequent Exploratory Data Analysis (EDA). Consequently, I initiated the feature engineering process before delving into the depths of the dataset. This proactive step enabled us to construct a solid foundation, ensuring that our EDA would be conducted with the requisite insights and tools for a more meaningful analysis.

To achieve this, I introduced three crucial additional columns: 'Year,' 'Month,' and 'Week Number.' These columns were thoughtfully crafted by extracting temporal details, such as the year, month, and week number, from the recorded dates.

A code snippet with outcome for crafting these features is shown is figure 6.

```
import calendar
df['Week'] = pd.DatetimeIndex(df.index).week       #Extract week number of the date and store in column 'Week'
df['Month'] = pd.DatetimeIndex(df.index).month     #Extract month number of the date and store in column 'month'
df['Year'] = pd.DatetimeIndex(df.index).year       #Extract year of the recorded date and store in column'year'
df.head()
```

Date	Store	Weekly_Sales	Holiday_Flag	Temperature	Fuel_Price	CPI	Unemployment	Week	Month	Year
2010-02-05	1	1643690.90	0	42.31	2.572	211.096358	8.106	5	2	2010
2010-02-12	1	1641957.44	1	38.51	2.548	211.242170	8.106	6	2	2010
2010-02-19	1	1611968.17	0	39.93	2.514	211.289143	8.106	7	2	2010
2010-02-26	1	1409727.59	0	46.63	2.561	211.319643	8.106	8	2	2010
2010-03-05	1	1554806.68	0	46.50	2.625	211.350143	8.106	9	3	2010

Figure 6 feature creation code snippet

Python's calendar module was instrumental in this process. By incorporating these columns, we not only enriched the dataset but also streamlined our EDA, enabling more insightful and precise investigations into sales trends and patterns over time.

Exploratory Data Analysis (EDA)

In the pursuit of our Exploratory Data Analysis (EDA), I employed two distinct approaches. The first approach offered a comprehensive overview of the dataset, while the second approach focused on individual store-level analysis.

Comprehensive Overview

To establish the first comprehensive overview, I amalgamated the weekly sales figures across all stores. This was accomplished by aggregating the weekly sales values for all stores on each recorded date. As a result, we obtained a consolidated dataset comprising 143 records, enabling a holistic examination of the collective sales dynamics across the entire dataset. The consolidated dataset further stored in new DataFrame named- total_df.

The processes of aggregating weekly sales figure across all stores and storing the resulting consolidated dataset in a new DataFrame is displayed in figure 7.

```
## droping irrelevant columns (since we are summing up across each store store number,weekly sale, ... of store is rrelevant)
total_df = df[0:143].drop(columns= ['Weekly_Sales','Store', 'Temperature', 'Fuel_Price', 'CPI', 'Unemployment'])
##compute total weekly sales across each store on each date and storing in new column 'Total_Sales'
total_df['Total_Sales'] = df.groupby(by = df.index)['Weekly_Sales'].sum()
total_df.head()
```

Date	Holiday_Flag	Week	Month	Year	Total_Sales
2010-02-05	0	5	2	2010	49750740.50
2010-02-12	1	6	2	2010	48336677.63
2010-02-19	0	7	2	2010	48276993.78
2010-02-26	0	8	2	2010	43968571.13
2010-03-05	0	9	3	2010	46871470.30

Figure 7 Weekly Sales Aggregation

Analyzing aggregated weekly sales:

The aggregated weekly sales statistics for the combined dataset figure 8, which includes weekly sales across all stores, provide insights into the overall sales distribution.

```
##summary statistics for 'Total Sales'
total_df['Total_Sales'].describe()
```

```
count    1.430000e+02
mean     4.711342e+07
std      5.444206e+06
min      3.959985e+07
25%      4.488059e+07
50%      4.624390e+07
75%      4.779202e+07
max      8.093142e+07
Name: Total_Sales, dtype: float64
```

Figure 8 Aggregated Weekly Sales Summary Statistics

- **Count:** The count remains the same at 143 records, indicating that we still have data for the same 143 weeks.
- **Mean:** The mean weekly sales amount has substantially increased to approximately 47,113,420 dollars, which is significantly higher than the individual store's mean of approximately 1,046,965 dollars. This jump in the mean reflects the cumulative sales across all stores, illustrating the overall sales magnitude.
- **Standard Deviation:** The standard deviation, at approximately 5,444,206 dollars, is relatively low compared to the individual store's standard deviation of about 564,366 dollars. This suggests that the combined dataset exhibits less variability in weekly sales, likely because it aggregates data from multiple stores.
- **Minimum and Maximum:** The minimum weekly sales value in the aggregated dataset is approximately 39,599,850 dollars, while the maximum is approximately 80,931,420 dollars. These values represent a broader range of sales compared to individual stores, with the highest weekly sales being more than 8 times the lowest.
- **Quartiles (25%, 50%, 75%):** The quartiles in the aggregated dataset indicate that 50% of the weeks have sales between approximately 44,880,590 dollars and 47,792,020 dollars. These quartiles further emphasize the broader distribution of weekly sales when considering all stores together.

In summary, the statistics for the aggregated total sales show significantly higher mean sales and a broader range compared to the individual store-level statistics. This is expected as it reflects the cumulative sales of all stores and provides a perspective on the overall sales performance across the dataset.

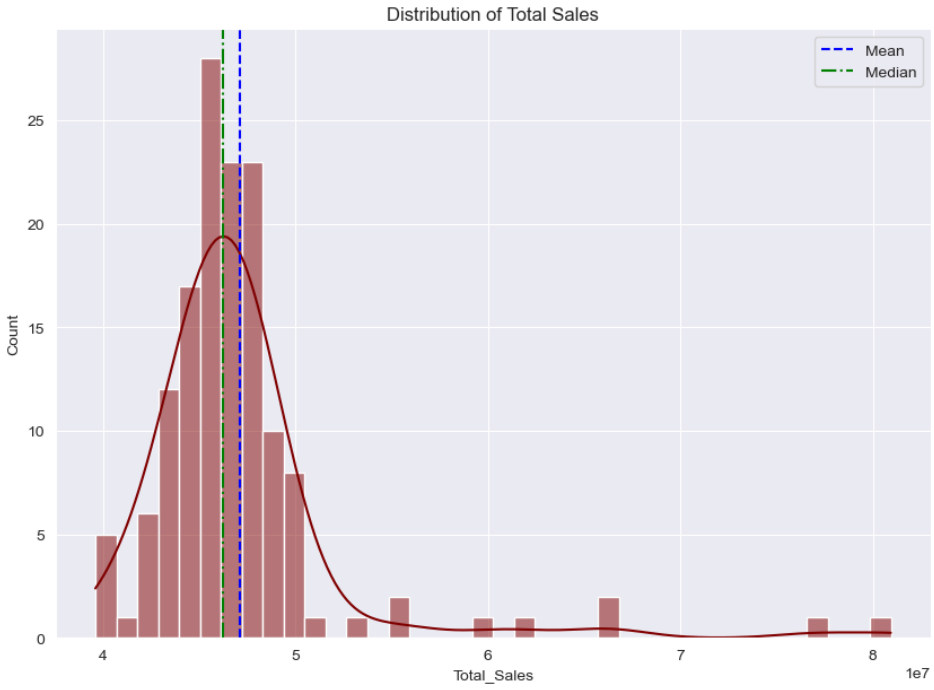


Figure 9 Total Sales Distribution

The distribution of the aggregated weekly sales is shown in figure 9. The slight difference between the mean and median in your histogram suggests that there are some weeks with notably high sales that are pulling the mean slightly to the right.

Analyzing aggregated weekly sales per week:

The aggregated weekly sales plot figure 10, when examined against the week number, reveals several significant insights.

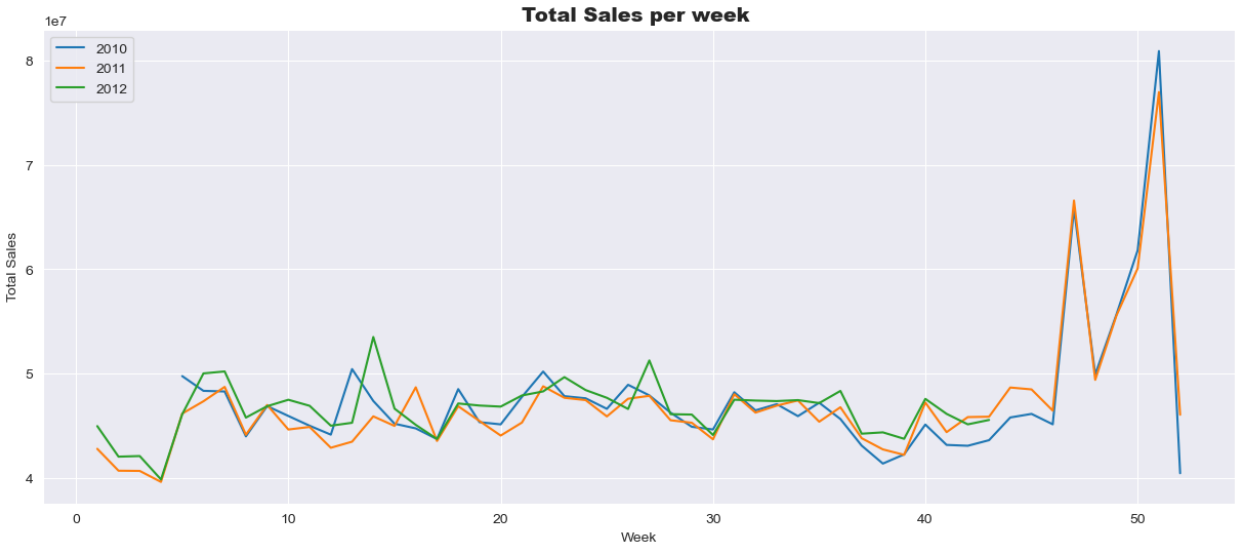


Figure 10 Total Sales vs Week

- **Holiday Sales Increase:** The analysis underscores a consistent increase in total sales during holiday weeks each year. This observation aligns with typical consumer behavior, as holidays often lead to higher spending, whether for gift shopping, holiday meals, or other festive expenses.
- **Thanksgiving and Christmas Peaks:** Notably, there are conspicuous spikes in sales during weeks 47 and 51, which correspond to Thanksgiving and Christmas, respectively. These peaks are indicative of the substantial boost in sales during these major holiday periods, where consumers tend to increase their purchases significantly.
- **Seasonal Consideration for 2012 Forecasting:** Recognizing the seasonality associated with holiday sales is crucial when forecasting sales for the year 2012. Incorporating this seasonality factor into the forecasting model can help accurately predict sales during these holiday periods.
- **Post-New Year Decline**: The graph implies a consistent pattern of sales decline immediately following the New Year. This decline could be attributed to a post-holiday lull, where consumer spending tends to taper off after the holiday season.

Overall, the plot provides valuable insights into the seasonality and holiday-related sales patterns. These observations will be instrumental in developing a robust sales forecasting model that captures these trends and fluctuations, ultimately aiding in more accurate predictions and informed decision-making.

Analyzing aggregated weekly sales per month:

The aggregated weekly sales plot against the month figure 11 offers valuable insights into the recurring sales patterns throughout the year.

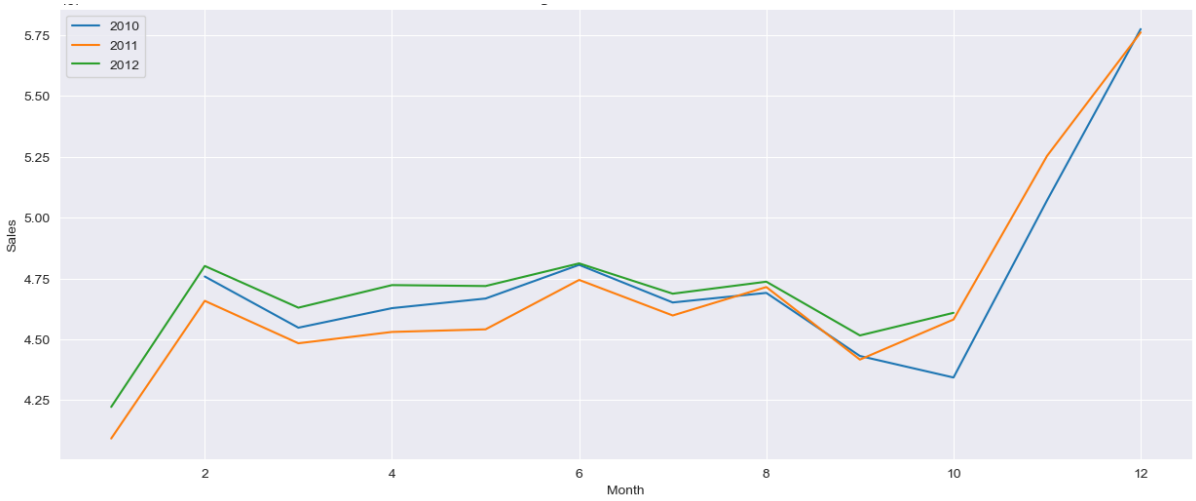


Figure 11 Sales vs Month

- **Monthly Sales Peaks:** The plot reveals distinct sales peaks occurring in the 2nd, 6th, and 8th months. These peaks likely correspond to specific events or factors that drive increased consumer spending during those months. It's essential to identify the underlying reasons for these peaks, which could include seasonal promotions, back-to-school shopping, or other regional factors.
- **Year-End Sales Surge:** Another noteworthy observation is the consistent pattern of sales trending upward after the 10th month (October) each year. This upward trend coincides with the approach of the year-end holiday season, which typically includes Thanksgiving, Christmas, and New Year's. The surge in consumer spending during these holidays justifies the observed trend.
- **December Sales Peak:** The highest sales recorded in December every year align with the holiday shopping season. December is marked by various major holidays, including Christmas, which are associated with heightened consumer activity, gift-buying, and festive spending.
- **Consistent Yearly Pattern:** A particularly significant observation is the repetition of this sales pattern each year. The fact that sales follow a similar trajectory annually suggests a level of seasonality in the data. This consistency will be instrumental in modeling time series data, allowing for the development of forecasting models that can capture and predict these recurring sales patterns effectively.

In summary, the plot illustrates the cyclicality and seasonality in the sales data, highlighting specific months with significant sales peaks and a consistent yearly pattern. These insights will be invaluable when constructing time series models for sales forecasting, as they provide a basis for understanding and predicting future sales trends.

Store-Level Insights

In the second part of our Exploratory Data Analysis, we delve into advanced visualizations to address critical questions pertaining to individual store data and the factors influencing weekly sales. This section's primary focus encompasses the following aspects of the Walmart dataset:

- Identifying the extent of sales increase during holidays.
- Assessing the impact of the unemployment rate on weekly sales and identifying stores most affected.
- Analyzing whether weekly sales exhibit seasonal trends and pinpointing the reasons behind these trends.
- Investigating the potential influence of temperature on weekly sales.
- Examining the relationship between the Consumer Price Index (CPI) and weekly sales for different stores.
- Identifying the top-performing stores based on historical data.
- Identifying the worst-performing store and quantifying the significance of the gap between the highest and lowest performing stores.

Individual Store Performance Evaluation

The bar plot in figure 12 depicting the average weekly sales for each store during the period from February 2010 to October 2012 offers valuable insights into store performance within the Walmart dataset. By arranging the bars in descending order, we can readily identify the highest and lowest performing stores based solely on their sales data.

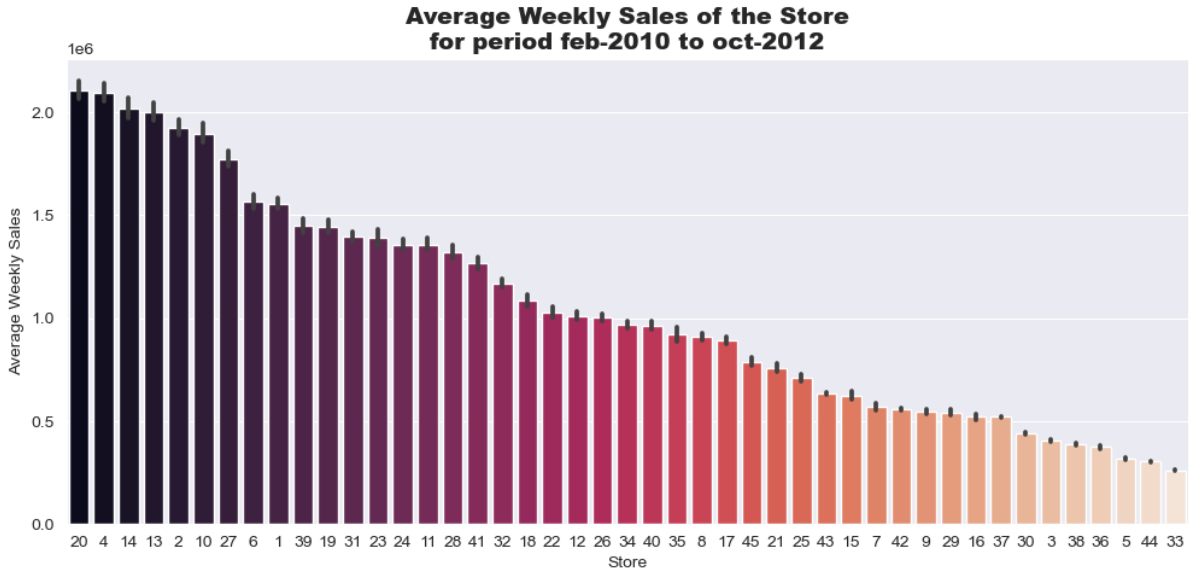


Figure 12 Individual Store Sales

Observations from the plot indicate that:

- **Top Performing Stores:** Stores 20, 4, 14, 13, and 2 consistently stand out as the top revenue-generating stores for Walmart. These stores have maintained average weekly sales figures on an average 2 million dollars throughout the specified period.

- **Low Performing Stores:** Conversely, stores 5, 44, and 33 are identified as the lowest performing stores, exhibiting significantly lower average weekly sales compared to their counterparts.

While it's important to note that factors such as store size and location can indeed impact store performance, this analysis focuses solely on the available sales data. These top-performing stores have consistently demonstrated their ability to generate substantial revenue for Walmart, reflecting their strong sales performance within the dataset.

In the plot depicting the "average weekly sales of the store for the period from February 2010 to October 2012," figure 12, we can distinguish distinct clusters of stores based on their sales performance. To facilitate further analysis and categorization, I encoded these stores into four categories denoted as 'A,' 'B,' 'C,' and 'D' according to their average weekly sales:

- **Category 'A' (Average weekly sales > 1.5 million):** This category represents stores with exceptionally high weekly sales, exceeding 1.5 million dollars. These stores exhibit top-tier performance and contribute significantly to Walmart's revenue.
- **Category 'B' (1 million < Average weekly sales < 1.5 million):** Stores falling within this category have solid weekly sales, ranging between 1 million and 1.5 million dollars. They perform well and contribute substantially to Walmart's revenue but may not be the absolute top performers.
- **Category 'C' (500k < Average weekly sales < 1 million):** This category encompasses stores with moderate weekly sales, ranging from 500,000 to 1 million dollars. These stores make a valuable contribution to Walmart's sales but may have room for growth.
- **Category 'D' (Average weekly sales < 500k):** Stores in this category have relatively lower weekly sales, less than 500,000 dollars. While they are essential components of the Walmart network, they may represent lower-performing or smaller stores within the dataset.

The analysis of each of the 45 stores individually can be a time-consuming task. Therefore, the introduction of categorization based on average weekly sales serves as a practical solution to simplify and streamline the analysis. By grouping stores into categories 'A,' 'B,' 'C,' and 'D,' we not only ease the analysis process but also gain a high-level understanding of store performance, allowing for more efficient and strategic decision-making within the Walmart dataset.

Code for this categorization is given in snippet below:

```
grp_df = df.groupby(by = 'Store').mean().sort_values(by = 'Weekly_Sales', ascending= False)
grp_df['type'] = 0
grp_df['type'].loc[grp_df['Weekly_Sales']> 1.5e6] = 'A'
grp_df['type'].loc[(grp_df['Weekly_Sales']> 1e6) & (grp_df['Weekly_Sales']< 1.5e6)] = 'B'
grp_df['type'].loc[(grp_df['Weekly_Sales']> 0.5e6) & (grp_df['Weekly_Sales']< 1e6)] = 'C'
grp_df['type'].loc[grp_df['Weekly_Sales']< 0.5e6] = 'D'
grp_df = grp_df.sort_index()

df['type'] = 0
for i in df['Store'].unique():
    df['type'].loc[df['Store'] == i] = grp_df['type'].loc[(grp_df.index == i)].values[0]
df.head()
```

Impact of Holidays on Sales

The comparative analysis I've conducted between sales during holiday periods and normal weeks is truly insightful. The two separate bar plots shown in figure 13, one depicting the average sales for holidays and non-holidays from 2010 to 2012 and the other illustrating the count of holidays and non-holidays over the same period, you've revealed a compelling observation.

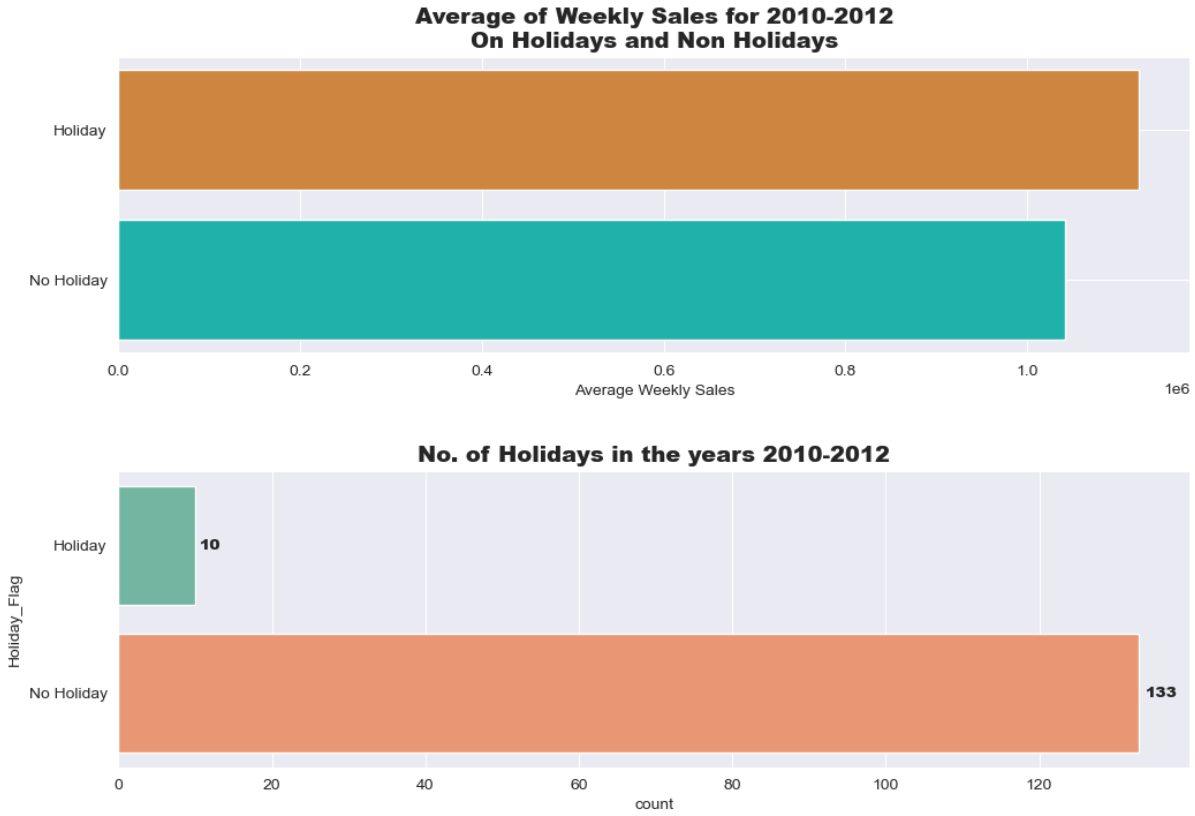


Figure 13 Impact of Holidays on Sales

The surprising finding here is that despite the relatively small number of holidays (only 10 over the period), the average sales during these holiday weeks are either roughly the same or slightly higher compared to sales during non-holiday weeks. This observation strongly suggests a robust and positive relationship between holidays and sales.

Impact of Temperature on Sales

The scatterplot (figure 14) analysis of temperature versus weekly sales, segmented into the four store categories (A, B, C, and D), revealed interesting insights.

The stores falling within categories A, B, and C, a distinctive pattern appeared where higher sales were observed within a specific temperature range of 25-50 degrees Fahrenheit. This temperature range appeared to be particularly favorable for these store categories, likely indicative of consumer preferences and buying patterns during milder weather conditions.

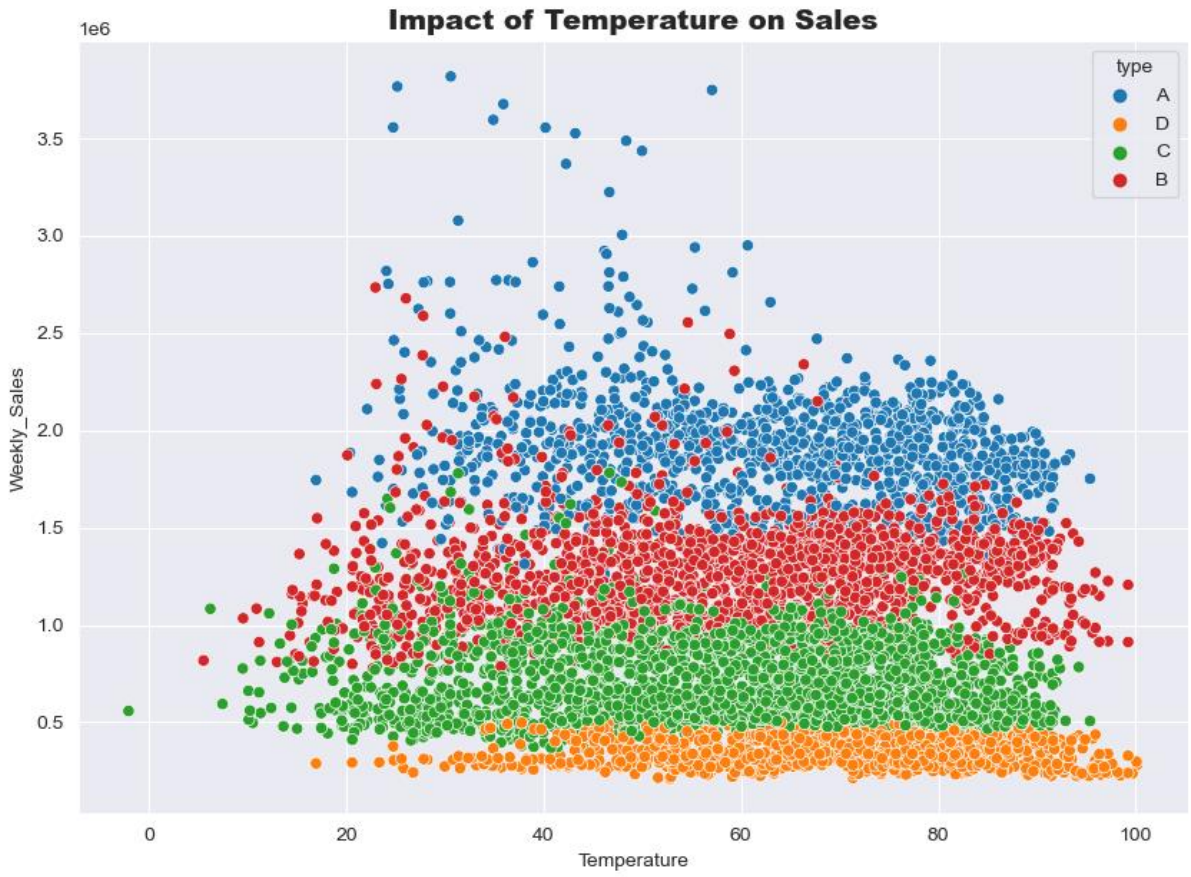


Figure 14 Impact of temperature on sales

Outside of this temperature range, the performance of all store categories exhibited a relatively uniform pattern, suggesting that temperature's impact on sales was more pronounced for these higher-performing stores. This observation underscores the importance of considering temperature.

Impact of Fuel Price on Sales

In the analysis of fuel price's impact on sales, several observations were made, shedding light on the relationship between these variables within the Walmart dataset.

- A scatterplot (figure 15) was constructed with weekly sales on the y-axis and fuel price on the x-axis. The plot indicated a decrease in sales as fuel prices exceeded 4.00, compared to sales within the 2.75 to 3.75 fuel price range.
- The highest sales figures across all store types were observed when fuel prices fell within the range of 2.75 to 3.75.
- Notably, the scatterplot did not reveal a clear and consistent pattern to strongly support a negative impact of fuel price on sales within the narrow range of fuel prices observed in the dataset.

It's important to acknowledge that the dataset's limited fuel price range may contribute to the lack of a definitive pattern. For more conclusive insights, it may be necessary to analyze a larger period and a broader range of fuel prices.

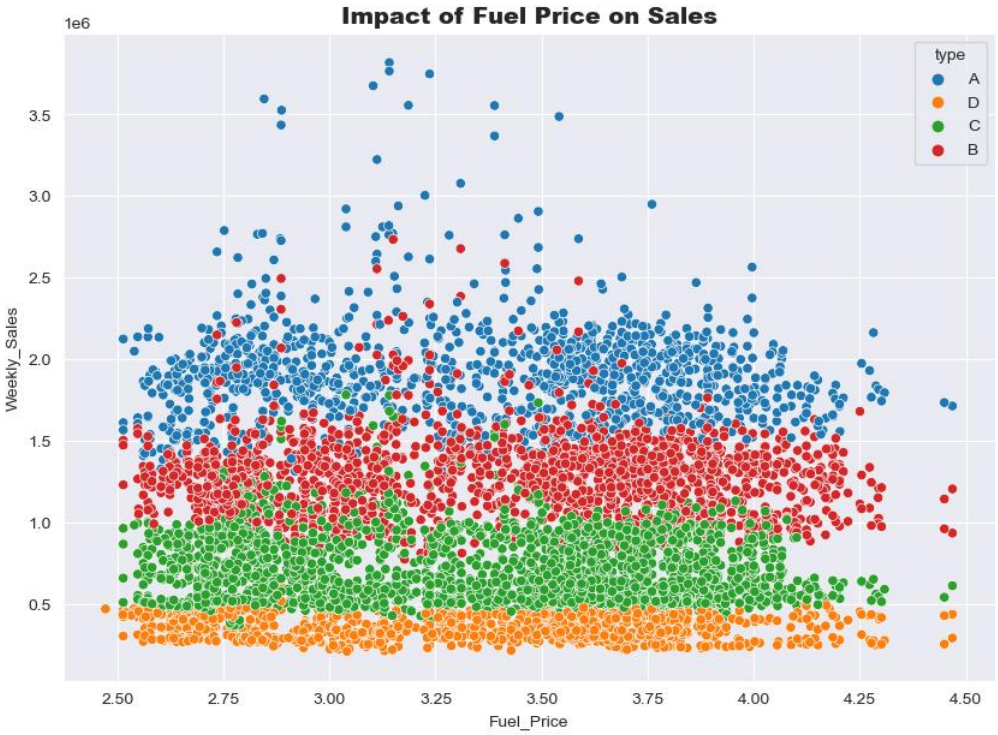


Figure 15 Impact of Fuel Price on Sales

For the current analysis of the Walmart dataset, we will consider the impact of fuel price to be relatively minimal or inconclusive.

Impact of CPI on Sales

In the analysis of the impact of the Consumer Price Index (CPI) on weekly sales (figure16), general observation can be highlighted.

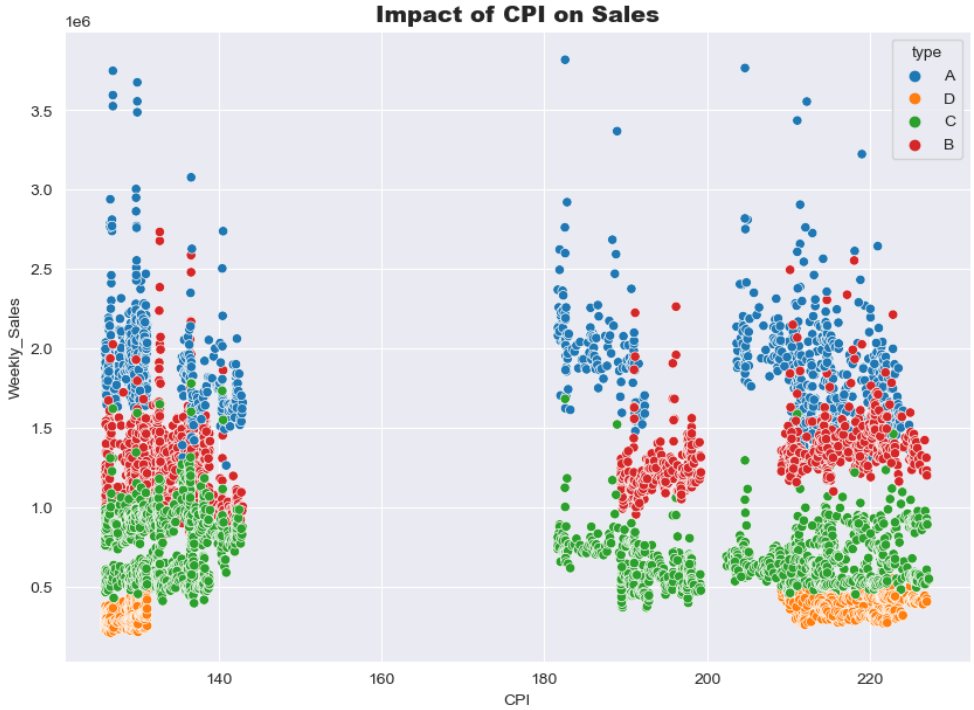


Figure 16 impact of CPI on Sales

A general trend in the scatter plot (figure 16) suggests a negative correlation between CPI and weekly sales. As the CPI increases, there is a tendency for weekly sales to decrease, indicating that economic conditions, reflected by the CPI, may influence consumer spending.

Impact of Unemployment on Sales

In the analysis of the impact of unemployment on weekly sales, several key insights were derived from the scatterplot (figure 17).

Figure 17 Impact of Unemployment on Sales

- A negative correlation is evident between unemployment and weekly sales. As unemployment rates increase, there is a tendency for sales to decline. This suggests that economic conditions, as represented by unemployment, have a notable influence on consumer spending behavior.
- The impact of unemployment on sales varies based on store performance. Low-performing stores appear to be less affected by unemployment, while high-performing stores experience a more significant drop in sales as unemployment rates rise.
- The scatterplot indicates that the highest recorded sales tend to occur within a specific unemployment index range, approximately between 7 to 9. This range suggests an optimal economic condition for sales performance.

These insights underscore the importance of considering unemployment as a critical economic factor.

Summary

As of now the analysis is totally based on graphical visualizations as summarized below.

Part 1 – Comprehensive Overview:

- Analyzed total sales and examined sales behavior by month and week of the year.

Part 2 – Store-Level Insights:

- Explored the impact of dataset factors on weekly sales through visualizations.
- Investigated critical questions regarding holidays, unemployment, seasonality, temperature, CPI, top-performing and worst-performing stores.

With this comprehensive exploration, the foundation is ready to move forward and make informed assumptions about the dataset. The next section is dedicated to Quantitative Data Analysis and hypothesis testing, where the statistical significance of each factor in predicting sales is accessed.

Quantitative Data Analysis

Quantitative data analysis serves as the basis of this project, enabling an examination of numerical information to extract meaningful patterns, correlations, and insights. Through statistical methods and mathematical techniques, I explored the dataset to derive evidence-based conclusions.

Correlation

The correlation (figure 18) values between various variables and weekly sales provide valuable insights into their relationships within the dataset.

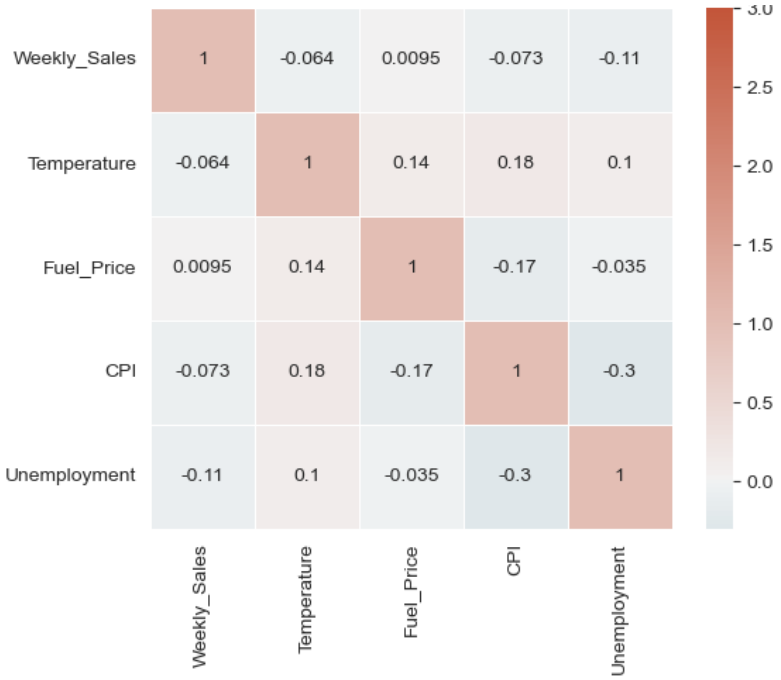


Figure 18 Correlation Plot

- **Temperature (-0.063810):** The negative correlation with temperature implies that as temperatures increase, weekly sales tend to decrease slightly. This could be due to consumers being less inclined to shop during very hot weather or the seasonality of certain products that sell better during cooler seasons.
- **Fuel Price (0.009464):** The positive but weak correlation with fuel price suggests that there is a minor association between higher fuel prices and slightly higher weekly sales. It contradicts our graphical analysis of fuel price.
- **CPI (-0.072634):** The negative correlation with the Consumer Price Index (CPI) indicates that as the CPI increases, weekly sales tend to decrease. This suggests that as the general cost of living rises, consumer spending may be impacted, leading to lower sales.
- **Unemployment (-0.106176):** The strongest negative correlation is observed with unemployment, indicating that as unemployment rates rise, weekly sales tend to decrease significantly.

Significance Testing - Holiday

From the insights gained in the second part of exploratory data analysis (EDA), it is reasonable to assume that there exists a correlation between the "Holiday Flag" and the sales values. To rigorously test and validate this assumption, we will employ a one-way analysis of variance (ANOVA) test on the "Holiday Flag" column.

- **Null Hypothesis (Ho):** The variables are not correlated, and the mean of all groups is the same. In other words, there is no significant difference in sales between holidays and non-holidays.
- **Alternative Hypothesis (H1):** The variables are indeed correlated, implying that there is a significant difference in sales between holidays and non-holidays.

I have conducted an ANOVA assumption test to evaluate the validity of our hypothesis (figure 19) Stats module from scipy library is utilized for the hypothesis testing.

```
from scipy import stats

## Normality test
#H0 : Normally Distributed
#H1 : Not normal

stat, pval = stats.shapiro(holiday_sales)
print("Stat:", stat)
print("pval:", pval)
```

```
Stat: 0.9449769258499146
pval: 7.148078136343106e-12
```

```
## Homogenity test
#H0 : All population variances are equal
#H1 : At least two of them differ

stat, pval = stats.levene(holiday_sales, normal_sales)
print("Stat:", stat)
print("pval:", pval)
```

```
Stat: 10.519203649624915
pval: 0.0011874337918717478
```

Figure 19 Assumption test for ANOVA

p-values for both Shapiro-Wilk test for normality and the Levene test for homogeneity of variances yield p-values less than 0.05. It implies that the assumption of normality is violated, and the data is not normally distributed. Also, the assumption of homogeneity of variances is also violated, indicating that the variances across groups are not equal.

Given that the assumptions required for ANOVA testing, such as normality and homogeneity of variances, are not satisfied in our dataset, we will opt for a non-parametric alternative, specifically the Kruskal-Wallis test (figure 20).

```
stat, pval = stats.kruskal(holiday_sales, normal_sales)
print("Stat:", stat)
print("pval:", pval)
```

```
Stat: 4.963273167380976
pval: 0.025891154949924652
```

Figure 20 Kruskal-Wallis Test

- **Statistical Significance:** The Kruskal-Wallis test yielded a test statistic of approximately 4.963, indicating that there is a significant difference in sales values among the categories being compared.
- **p-value:** The associated p-value, which is approximately 0.026, is less than the typical significance level of 0.05. This suggests strong evidence to reject the null hypothesis, indicating that there is a statistically significant difference in sales values between holidays and non-holidays.

The results provide statistical confirmation that the "Holiday" variable has a significant impact on sales values within the Walmart dataset, supporting our initial assumption.

Significance Test – Temperature, Fuel Price, CPI and Unemployment

In our analysis, variables- Temperature, Fuel Price, and CPI as well as Weekly Sales are numeric continuous variables, we will assess the significance of the Pearson correlation coefficient (Pearson's r) to quantify their relationship. The following hypotheses will guide our investigation:

- **Null Hypothesis (Ho):** The correlation between continuous variable and weekly sales is not statistically significant. In other words, there is no significant linear relationship between these variables.
- **Alternative Hypothesis (H1):** The correlation between temperature and weekly sales is statistically significant, suggesting the presence of a significant linear relationship between the two variables.

1. **Temperature:**

   The Pearson correlation test for temperature and sales was conducted as shown in figure 21:

```
stat, pval = stats.pearsonr(x= df['Temperature'], y = df['Weekly_Sales'])
print("Stat:", stat)
print("pval:", pval)
```

```
Stat: -0.06381001317946965
pval: 3.007647625833215e-07
```

Figure 21 Pearsonr test for Temperature and Sales

- The Pearson correlation coefficient (r) calculated to be approximately -0.0638 suggests a weak negative correlation between temperature and sales.
- The associated p-value, which is approximately 3.01e-07 (very close to zero), indicates an extremely high level of statistical significance.
- The statistically significant negative correlation suggests that there is indeed a relationship between temperature and sales. As temperatures increase, weekly sales tend to decrease slightly.

2. **Fuel Price:**

The Pearson correlation test for fuel price and sales was conducted as shown in figure 22:

```
stat, pval = stats.pearsonr(x= df['Fuel_Price'], y = df['Weekly_Sales'])
print("Stat:", stat)
print("pval:", pval)
```

```
Stat: 0.009463786314475139
pval: 0.447828748948578 16
```

Figure 22 Pearsonr test for Fuel Price and Sales

- The Pearson correlation coefficient (r) is approximately 0.0095, indicating an extremely weak positive correlation between fuel price and sales.
- The associated p-value is approximately 0.448, which is notably higher than the typical significance level of 0.05. This suggests that the correlation is not statistically significant.
- The non-significant correlation implies that changes in fuel price are not a reliable predictor of changes in sales for the Walmart dataset.

3. **CPI:**

The Pearson correlation test for CPI and sales was conducted as shown in figure 23:

```
stat, pval = stats.pearsonr(x= df['CPI'], y = df['Weekly_Sales'])
print("Stat:", stat)
print("pval:", pval)
```

```
Stat: -0.07263416204017617
pval: 5.438292612176716e-09
```

Figure 23 Pearsonr test for Fuel Price and Sales

- The Pearson correlation coefficient (r) is approximately -0.0726, indicating a weak negative correlation between CPI and sales.
- The associated p-value is approximately 5.44e-09 (extremely close to zero), signifying an exceptionally high level of statistical significance.
- The statistically significant negative correlation suggests that there is indeed a relationship between CPI and sales. As the Consumer Price Index increases, weekly sales tend to decrease slightly.

4. **Unemployment**

The Pearson correlation test for CPI and sales was conducted as shown in figure 24:

```
stat, pval = stats.pearsonr(x= df['Unemployment'], y = df['Weekly_Sales'])
print("Stat:", stat)
print("pval:", pval)
```

```
Stat: -0.10617608965795447
pval: 1.3448365210232518e-17
```

Figure 24 Pearsonr test for Unemployment and Sales

- The Pearson correlation coefficient (r) is approximately -0.1062, indicating a weak to moderate negative correlation between Unemployment and sales.
- The associated p-value is approximately 1.34e-17 (extremely close to zero), demonstrating an exceptionally high level of statistical significance.
- The statistically significant negative correlation suggests that there is indeed a relationship between Unemployment and sales. As Unemployment rates increase, weekly sales tend to decrease, albeit moderately.

Summary:

- **Kruskal-Wallis Test for Holiday Impact on Sales:** The Kruskal-Wallis test confirmed that holidays significantly affect sales, supporting our hypothesis.
- **Pearson Correlation Test for Temperature and Sales:** Temperature has a statistically significant but weak negative correlation with sales, suggesting a slight decrease in sales as temperature rises.
- **Pearson Correlation Test for Fuel Price and Sales:** There is no statistically significant correlation between fuel price and sales, indicating that changes in fuel price do not predict changes in sales.
- **Pearson Correlation Test for CPI and Sales:** The Consumer Price Index (CPI) has a statistically significant weak negative correlation with sales, indicating a slight decrease in sales as CPI increases.
- **Pearson Correlation Test for Unemployment and Sales:** Unemployment rates have a statistically significant moderate negative correlation with sales, suggesting that higher unemployment corresponds to lower sales.

EDA Conclusion:

Based on the analysis conducted on the Walmart dataset and the observations made from significance tests, it's possible to justify that regression models may not yield exceptionally accurate results for sales forecasting.

Model Building

Based on the comprehensive data analysis conducted, it has become evident that the nature of the dataset, particularly the weekly sales data, lends itself well to time series modeling. Time series models are specifically designed to capture and leverage temporal patterns and trends, making them highly suitable for forecasting in this context. Recognizing this, I decided to proceed with the general modeling procedure for time series analysis.

The development of a model of this kind to describe the dependence structure in an observed time series is usually best achieved by a three-stage iterative procedure based on identification, estimation, and diagnostic checking. *(ref: Granville Tunnicliffe Wilson, 2016)*

1. By identification we mean the use of the data, and of any information on how the series was generated, to suggest a subclass of parsimonious models worthy to be entertained.
2. By estimation we mean efficient use of the data to make inferences about the parameters conditional on the adequacy of the model entertained.
3. By diagnostic checking we mean checking the fitted model in its relation to the data with intent to reveal model inadequacies and so to achieve model improvement.

I followed the model-building process with two distinct approaches:

- **Total Sales Forecasting:**
  The first approach was centered on creating a model that forecasts the total sales of Walmart, aggregating data from all its stores. This approach provides an overview of the company's performance.
- **Individual Store Sales Forecasting:**
  The second approach focused on constructing models that predict the weekly sales of each individual store separately. This approach allows for a more granular analysis, helping identify specific store performance and trends.

Total Sales Forecasting

Train Test Split

Before model building, I configured the training and testing set. Since time series is an ordered set of observations splitting was done without shuffling the original dataset.

- **Dataset Splitting:** To facilitate model building and evaluation, the dataset was divided into two distinct sets: the training set and the test set. This division is essential to assess the model's predictive capabilities accurately.
- **Test Set Configuration:** The test set was carefully configured to achieve the project's primary objective: forecasting sales for the subsequent 12 weeks. As a result, the test set consists of a specific set of data points corresponding to the 12-week forecasting horizon.
- **Training Data Exclusivity:** The training set exclusively served as the dataset for model development. This partitioning helped ensuring that the model learns

from historical data and past patterns while remaining uninfluenced by future observations.

- **Model Evaluation:** Model performance was evaluated by comparing the forecasts generated by the model against the test set data. This evaluation served a critical validation step, measuring the accuracy and reliability of the model's predictions for the 12-week sales horizon

The code for train-test split is given in figure 25 below:

```
train = total_df.iloc[: -12, -1]
test = total_df.iloc[-12:, -1]

train = pd.DataFrame(data = train)
test = pd.DataFrame(data = test)
```

Figure 25 Train-Test Split

Data Decomposition

The decomposition method was employed to dissect the time series data into its constituent components, which are the trend, seasonality, and irregular (or residual) components. Code snippet and output is shown in figure 26 below.

```
decompose = seasonal_decompose(total_df['Total_Sales'], model = 'additive', period = 52)
fig = decompose.plot()
fig.set_figwidth(12)
fig.set_figheight(8)
fig.suptitle('Decomposition of Weekly Sales time series')
plt.show()
```

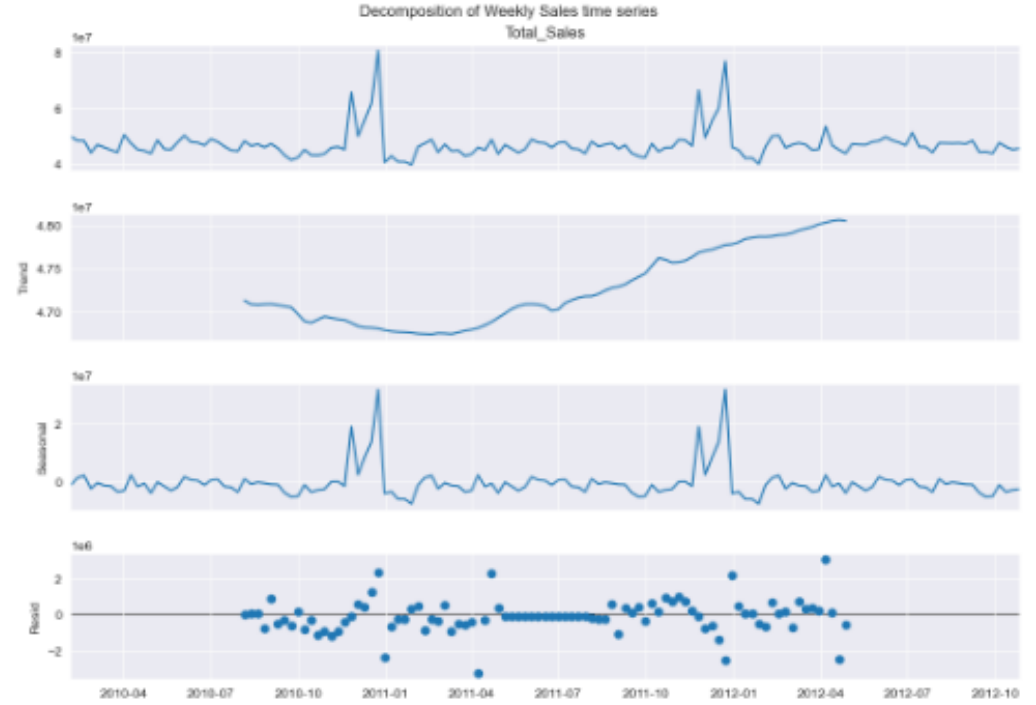


Figure 26 Decomposition of Total Sales

Through the decomposition process, I successfully identified and separated these three critical components from the given time series data.

The analysis revealed that the trend component follows a clear linear upward trajectory over time. This implies that there is a consistent and continuous increase in sales, indicating potential growth.

The decomposition method highlighted the presence of seasonality within the time series data. Seasonality suggests that there are recurring patterns or cycles in the data

Helper Functions

Two helper functions were developed by to avoid repetitive coding. These helper functions are 'print_err' and 'plt_forecast':

print_err Helper Function:

```
from sklearn.metrics import mean_absolute_percentage_error
error_df = pd.DataFrame(index= ['MAPE','RMSE'])

def print_err(col, Name):

    MAPE = mean_absolute_percentage_error(test['Total_Sales'], test[col])
    RMSE = np.sqrt(mean_squared_error(test['Total_Sales'], test[col]))

    error_df.loc['MAPE', col] = MAPE
    error_df.loc['RMSE', col] = RMSE

    print(f"""--------------------------------------------------------------------------------
    Mean Absolue Percentage Error for {Name} forecast is : {MAPE:.4f}
    Root Mean Squared Error for {Name} forecast is : {RMSE:.4f}
    --------------------------------------------------------------------------------""")
```

Figure 27 Helper Function - print_err

- **Purpose:** The print_err helper function was created to streamline the evaluation process for time series models.
- **Functionality:** It calculates and prints two crucial performance metrics, the Mean Absolute Percentage Error (MAPE) and Root Mean Square Error (RMSE), for each trained time series model.
- **Error Logging:** Additionally, it records these MAPE and RMSE values in an error_df dataframe. This logging of errors aids in the systematic comparison and assessment of model performance.
- **Efficiency:** By automating error calculation and recording, print_err eliminates repetitive coding and ensures consistent evaluation across different models.

plt_forecast Helper Function:

- **Purpose:** The plt_forecast helper function serves as a visual aid for understanding time series model forecasts.

```
def plt_forecast(col):
    fig, ax = plt.subplots(figsize = (10,8))

    ax.plot(train.index, train['Total_Sales'], 'b-', label = 'Train')
    ax.plot(test.index, test['Total_Sales'], 'g-', label = 'Test')
    ax.plot(test.index, test[col], 'r--', label = 'Predicted')
    ax.set_xlabel("Date")
    ax.set_ylabel("Sales")
    ax.legend()
    fig.autofmt_xdate()
```

Figure 28 helper Function plt_forecast

- **Functionality:** It generates a single plot that displays the following components together: the training dataset, the test dataset, and the forecasted values produced by the model.
- **Visual Insight:** This function offers a clear and concise visual representation of how the model's forecasts align with both historical and unseen data points.
- **Interpretability**: By plotting these elements in a single graph, plt_forecast facilitates the interpretation of model performance and helps stakeholders gain insights from the forecasted values.

Naive Model

The development of a naive models served as a fundamental step in setting a performance baseline for the sales forecasting problem. It provided a simple yet essential point of reference against which the performance of more complex models was assessed.

Naive Model: Historical Mean

Forecasting the historical mean serves as a simplistic yet informative baseline for sales prediction. It operates under the assumption that, on average, sales will remain consistent with past performance. This approach is useful for establishing a basic reference point against which more complex models can be compared.

Figure 29 shows the implementation and output of a historical mean forecasting.

- The calculated MAPE for the historical mean forecast is approximately 0.0301. a MAPE of 0.0301 indicates that, on average, the historical mean forecast deviates from the actual sales values by about 3.01%.
- The calculated RMSE for the historical mean forecast is approximately 1,794,183.4473. An RMSE of approximately 1,794,183.4473 provides insight into the magnitude of errors in the historical mean forecast. It quantifies the typical discrepancy between the forecasted and actual sales values in absolute terms.

These error values provide a baseline for assessing the accuracy of sales forecasts and serve as a point of comparison for evaluating the performance of more advanced forecasting models developed for Walmart's weekly sales prediction.

```
## implementing historical mean forecast.
historical_mean = np.mean(train)
test.loc[:, 'pred_mean'] = historical_mean.values[0]  ## seting a historical mean to pred_mean column
print_err('pred_mean', 'Historical mean')
plt_forecast('pred_mean')
```

```
--------------------------------------------------------------------------------
    Mean Absolue Percentage Error for Historical mean forecast is : 0.0301
    Root Mean Squared Error for Historical mean forecast is : 1794183.4473
    --------------------------------------------------------------------------------
```

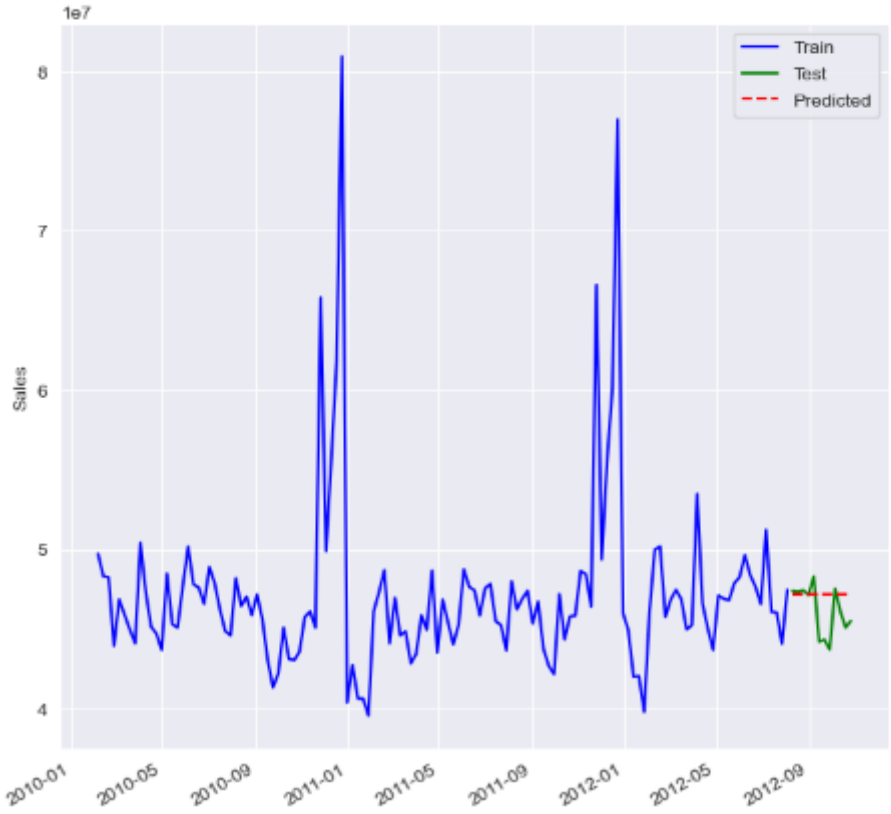


Figure 29 Historical Mean Forecast

Exponential Smoothing

Exponential smoothing is a widely used time series forecasting technique that relies on weighted averages of past observations to predict future values. It assigns exponentially decreasing weights to older data points, with the most recent data points receiving the highest weights.

The choice of the specific exponential smoothing method (e.g., simple exponential smoothing, Holt's linear exponential smoothing, or Holt-Winters exponential smoothing) depends on the characteristics of the time series and the desired level of complexity.

- **Simple Exponential Smoothing:** Simple exponential smoothing is a basic time series forecasting method that assigns exponentially decreasing weights to past observations and uses a single smoothing parameter (alpha) to predict future values.
- **Holt's Linear Exponential Smoothing:** Holt's linear exponential smoothing, also known as double exponential smoothing, extends simple exponential

smoothing by incorporating a trend component in addition to the level component, making it suitable for time series data with linear trends.

- **Holt-Winters Exponential Smoothing:** Holt-Winters exponential smoothing is an advanced forecasting technique that includes both level and trend components, as well as a seasonal component, making it effective for time series data with trends and seasonality.

Since our data for total sales shows both trend and seasonal component as shown in figure 26, I opted for Holt-Winters Exponential Smoothing (Triple Exponential Smoothing) model for forecasting.

Exponential Smoothing: Triple Exponential Smoothing

The **'ExponentialSmoothing'** function from the **'statsmodels.tsa.api'** library was utilized to implement the triple exponential smoothing model as shown in figure 30. This function provides a convenient and powerful way to set up and configure time series forecasting models.

```
from statsmodels.tsa.api import ExponentialSmoothing
TES_model = ExponentialSmoothing(train, trend = 'additive',seasonal= 'additive').fit(optimized= True)
TES_pred = TES_model.forecast(steps = len(test))
test.loc[:, 'TES_pred'] = TES_pred
print_err('TES_pred', 'Triple Exponential Smoothing')
plt_forecast('TES_pred')
```

```
-------------------------------------------------------------------------------
    Mean Absolue Percentage Error for Triple Exponential Smoothing forecast is : 0.0156
    Root Mean Squared Error for Triple Exponential Smoothing forecast is : 899961.8380
    -------------------------------------------------------------------------------
```

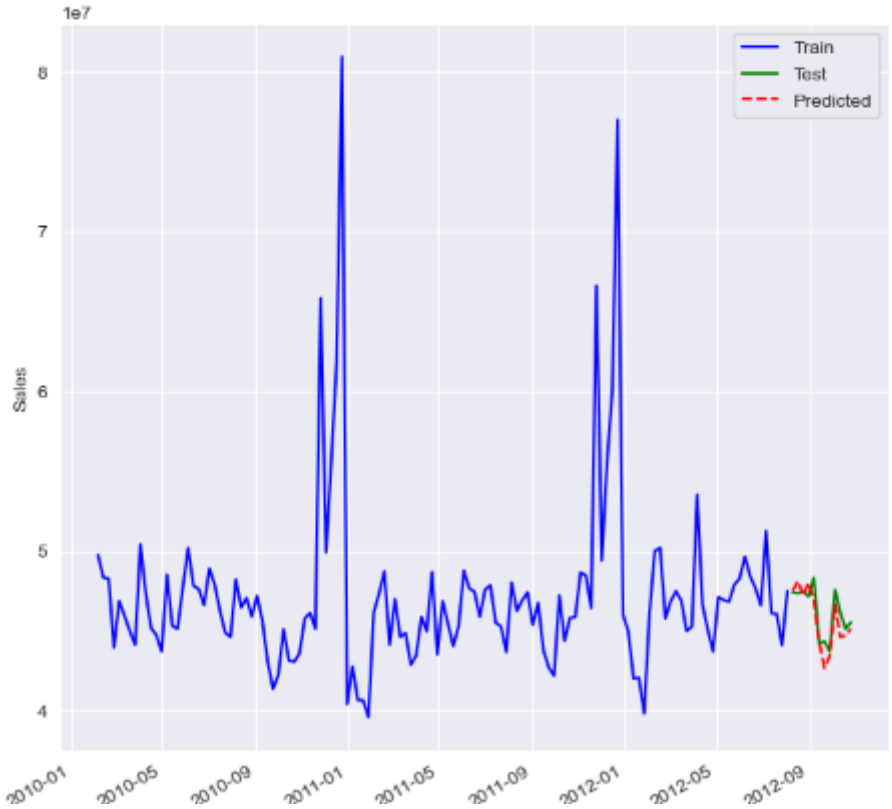


Figure 30 Triple Exponential Smoothing

The decision to use an 'additive' seasonal component was based on the decomposition analysis of the time series data, which indicated that the seasonal fluctuations could be adequately represented through additive adjustments.

Observations:

- The Triple Exponential Smoothing model significantly outperformed the Historical Mean forecasting in terms of MAPE. It achieved a MAPE of approximately 0.0156, which is about half the MAPE obtained from the Historical Mean forecast (0.0301). This indicates a substantially higher level of accuracy in predicting sales.
- Similarly, the Triple Exponential Smoothing model exhibits a substantially lower RMSE of approximately 899,961.8380 compared to the Historical Mean forecast's RMSE of about 1,794,183.4473. This indicates that the model's forecasts are closer to the actual sales values, resulting in smaller prediction errors.
- These improved error metrics suggested that the Triple Exponential Smoothing model is a valid and robust choice for predicting Walmart's weekly sales, capturing both trend and seasonality in the data effectively.
- Also, the plot of train test and predicted sales values (figure 30), gives an visual evidence of model's accuracy. The forecasted data (dashed red line) approximately follows the patterns in sales values in test set (solid green line).

Regression Methods

Autoregressive Integrated Moving Average (ARIMA):

An autoregressive integrated moving average (ARIMA) process is the combination of the AR(p) and MA(q) processes, but in terms of the differenced series. It is denoted as ARIMA(p,d,q), where p is the order of the AR(p) process, d is the order of integration, and q is the order of the MA(q) process. Integration is the reverse of differencing, and the order of integration d is equal to the number of times the series has been differenced to be rendered stationary.

Seasonal autoregressive integrated moving average (SARIMA):

The seasonal autoregressive integrated moving average (SARIMA) model adds seasonal parameters to the ARIMA(p,d,q) model. It is denoted as SARIMA(p,d,q)(P,D,Q)m, where P is the order of the seasonal AR(P) process, D is the seasonal order of integration, Q is the order of the seasonal MA(Q) process, and m is the frequency, or the number of observations per seasonal cycle. Note that a SARIMA(p,d,q)(0,0,0)m model is equivalent to an ARIMA(p,d,q) model

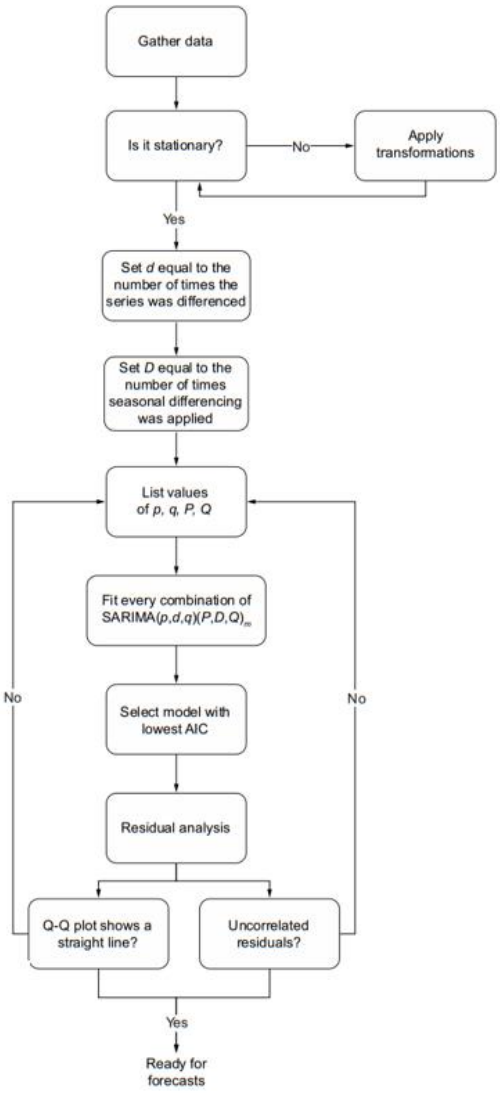
As our dataset shows seasonality (see figure 26) I decided to proceed with SARIMA(p,d,q)(P,D,Q)m model.

Seasonal Autoregressive Integrated Moving Average:

The flowchart below shows the general modeling procedure for SARIMA process. Following the steps general modeling procedure (flowchart), I first checked for stationarity.

The Augmented Dickey-Fuller (ADF) test is a statistical hypothesis test used to check the stationarity of a time series.

Augmented Dickey-Fuller test was implemented using **'adfuller'** function from **'statsmodels.tsa.stattools'** to check the stationarity. The code snippet to check the stationarity of the process is shown in figure 31.




```
from statsmodels.tsa.stattools import adfuller

ADF_result = adfuller(train)

print(f'ADF Statistic {ADF_result[0]}')
print(f'p-value {ADF_result[1]}')
```

```
ADF Statistic -5.667865856980574
p-value 9.063916855404198e-07
```

Figure 31 ADF test for stationarity

- The ADF statistic is negative, indicating that the data is moving closer to stationarity.
- The extremely low p-value suggests, it supports the alternative hypothesis (H1) that the time series is stationary.

The seasonal component of the data figure 26 also tested using Augmented Dickey-Fuller (ADF) test as shown in figure 32.

```
ADF_result = adfuller(decompose.seasonal)

print(f'ADF Statistic {ADF_result[0]}')
print(f'p-value {ADF_result[1]}')
```

```
ADF Statistic -5.913398012001533
p-value 2.6066955770100113e-07
```

Figure 32 Test for stationarity on Seasonal Component

The extremely low p-value suggests that the seasonal component is also stationarity.

Observation:

- Both the original time series and the seasonal component pass the ADF test (low p-values).
- It suggests that no differencing (d=0) and no seasonal differencing (D=0) are required for stationarity.

With the Augmented Dickey-Fuller testing step was done, I defined the range of possible values for p, q, P, and Q, fit each unique SARIMA(p,d,q)(P,D,Q)m model, and selected the one with the lowest AIC figure 33 .

```
ps1 = range(0,4) #try the values [0,1,2,3] for p
qs1 = range(0,4) #try the values [0,1,2,3] for q
Ps1 = range(0,4) #try the values [0,1,2,3] for P
Qs1 = range(0,4) #try the values [0,1,2,3] for Q

d = 0
D = 0
s = 52

order_ls1 = list(product(ps1, qs1, Ps1, Qs1)) #generate the unique combination of order


results1 = []
for order in order_ls1:    #for each unique combination fit a model
    model = SARIMAX(train,
                    order = (order[0], d, order[1]),
                     seasonal_order= (order[2], D, order[3], s),
                    simple_differencing = False).fit()
    aic = model.aic
    #print([order, aic])
    results1.append([order, aic])


results_df1 = pd.DataFrame(results1, columns= ['(p,q)', 'AIC']).sort_values(by = 'AIC').reset_index()
results_df1
```

	index	(p,q)	AIC
0	244	(3, 3, 1, 0)	4406.710593
1	228	(3, 2, 1, 0)	4406.848865
2	232	(3, 2, 2, 0)	4408.864597
3	229	(3, 2, 1, 1)	4408.883812
4	248	(3, 3, 2, 0)	4409.212178
...	...	...	...
251	16	(0, 1, 0, 0)	4921.363873
252	31	(0, 1, 3, 3)	4922.634193
253	0	(0, 0, 0, 0)	5005.121150
254	2	(0, 0, 0, 2)	5462.321853
255	1	(0, 0, 0, 1)	7536.276637

Figure 33 Order Search for SARIMA

Note: Results in figure 33 shows that the SARIMA(3,0,3)(1,0,0)52 model has the lowest AIC, which is a value of 4406.71 however the second lowest model has minimum RMSE I selected SARIMA(3,0,2)(1,0,0)52 .

The AIC (Akaike Information Criterion) is a model selection criterion that balances goodness of fit and model complexity. The SARIMA(3,0,2)(1,0,0)52 model is chosen based on AIC as it provides the most suitable representation of the time series data, effectively capturing its seasonal and temporal patterns.

Once the optimal order of SARIMA was found, The model was retrained on the training set to perform residual analysis as shown in figure 34 below:

To assess the model quality, diagnostic plots figure 34, played a crucial role in qualitatively evaluating the residuals. These plots provided valuable insights into the behavior of the residuals:

```
model = SARIMAX(train,  #model retraining
                order = (3,0,2),
                seasonal_order=(1,0,0,52),
                simple_differencing = False).fit()

plt.style.use('seaborn-pastel') #residual plot
model.plot_diagnostics()
plt.tight_layout()
plt.show()
```

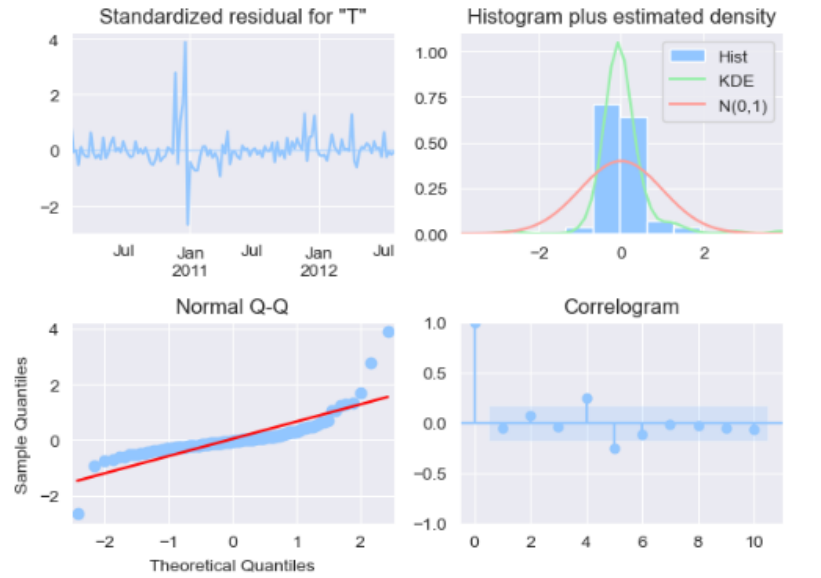


Figure 34 Residual Diagnostics

- **Top-Left Plot:** This plot displays the residuals across the entire dataset. The absence of any visible trend in the plot, along with a stable mean over time, strongly suggests that the residuals exhibit characteristics similar to white noise, indicating a random stationary process.
- **Top-Right Plot:** The histogram of the residuals is presented in this plot. Its similarity in shape to a normal distribution is indicative that the residuals closely follow a normal distribution, reinforcing the notion that they resemble white noise.
- **Bottom-Left Plot (Q-Q Plot):** This plot compares the distribution of residuals against a theoretical normal distribution. The resemblance of the residual distribution to a normal distribution further validates the model's goodness of fit.
- **Bottom-Right Plot (Autocorrelation Function):** The autocorrelation function of residuals is depicted in this plot. The presence of only a significant peak at lag 0 is a strong indicator that the residuals are not correlated with each other, affirming their similarity to white noise.

These diagnostic plots collectively support the conclusion that the model is well-fitted, with residuals demonstrating properties consistent with white noise. This reinforces the model's reliability for forecasting purposes.

The final test to determine whether to use this model for forecasting or not was the Ljung-Box test. The Ljung-Box test was implemented using ‘acorr_ljungbox’ function of a ‘statsmodels.stats.diagnostic’ library. The implantation is shown in figure 35.

```
## quantitative analysis for residual correlation

from statsmodels.stats.diagnostic import acorr_ljungbox

acorr_ljungbox(model.resid, np.arange(1,11,1))
```

	lb_stat	lb_pvalue
1	0.126854	0.721716
2	0.288137	0.865829
3	0.755166	0.860158
4	3.043230	0.550617
5	6.024058	0.303885
6	6.764709	0.343156
7	6.916071	0.437671
8	6.966835	0.540215
9	7.154105	0.621078
10	7.598453	0.667994

Figure 35 Ljung-Box Test

The returned p-values were all greater than 0.05. Therefore, I failed to reject the null hypothesis, and concluded that the residuals are independent and uncorrelated, just like white noise.

The SRIMA model passed all the tests from the residuals analysis, and was ready to use it for forecasting. Again, 12 period sales values were forecasted to compare the predicted values to the observed values in the test set.

- The SARIMA model demonstrated a lower MAPE (0.0148) (figure 36) compared to the previous Triple Exponential Smoothing model (0.0156), signifying improved accuracy in forecasts.
- Similarly, the SARIMA model yielded a lower RMSE (846,488.4748) compared to the Triple Exponential Smoothing model (899,961.8380), indicating better precision in predicting values.

```
pred_f = model.get_forecast(steps = len(test))            #forecast for the 12 period
SARIMA_pred = pred_f.summary_frame(alpha = 0.05)['mean']  #capture the forecasted values
test.loc[:, 'SARIMA_pred'] = SARIMA_pred                  #add forecasted values to test dataframe

print_err('SARIMA_pred', 'SARIMA')                        #print errors based on forecasted values
plt_forecast('SARIMA_pred')                               #plot train test and forecasted values
```

```
---------------------------------------------------------------------------
    Mean Absolue Percentage Error for SARIMA forecast is : 0.0148
    Root Mean Squared Error for SARIMA forecast is : 846488.4748
    ---------------------------------------------------------------------------
```

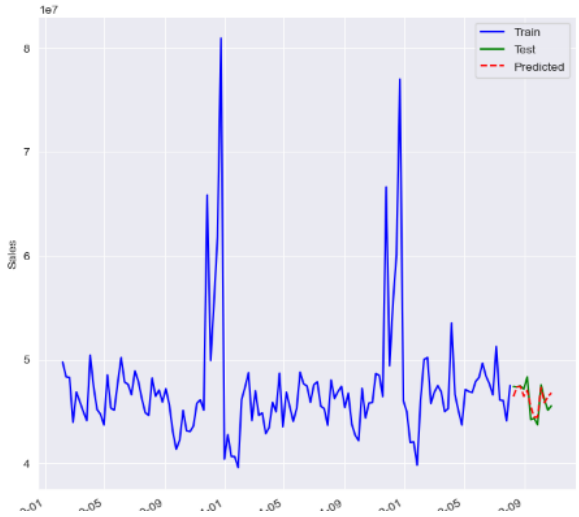
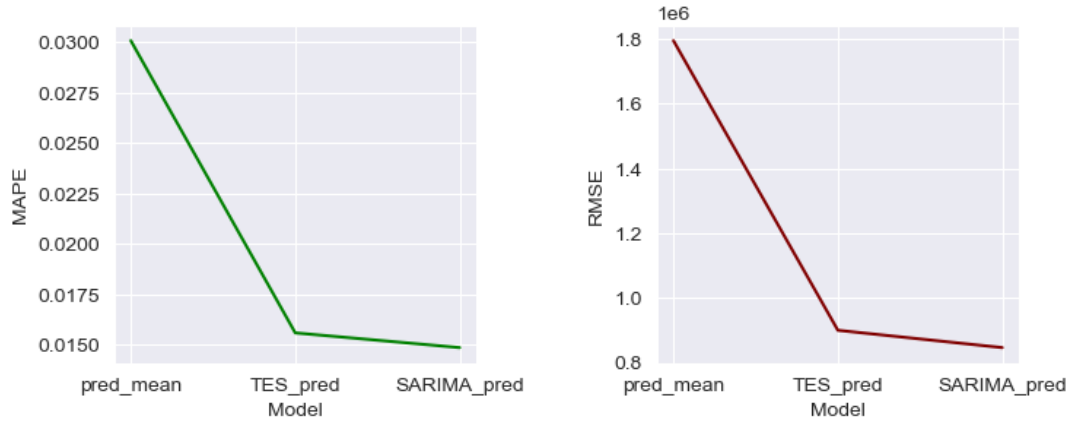


Figure 36 SARIMA Forecast

The SARIMA model outperforms the Triple Exponential Smoothing model, showcasing superior forecasting accuracy and precision. These results affirm the suitability of SARIMA for this dataset

Summary:



- **Historical Mean Forecasting:**
  - Mean Absolute Percentage Error (MAPE): 0.0301
  - Root Mean Squared Error (RMSE): 1,794,183.4473
  - Summary: Historical mean forecasting provides a straightforward baseline for predictions. However, it exhibits relatively high errors and lacks accuracy in capturing fluctuations in the data.
- **Triple Exponential Smoothing:**
  - Mean Absolute Percentage Error (MAPE): 0.0156
  - Root Mean Squared Error (RMSE): 899,961.8380
  - Summary: Triple Exponential Smoothing improves accuracy compared to the historical mean, offering lower MAPE and RMSE. It captures trends and seasonality in the data, making it a better choice for forecasting.
- **SARIMA Model (Seasonal Autoregressive Integrated Moving Average):**
  - Mean Absolute Percentage Error (MAPE): 0.0148
  - Root Mean Squared Error (RMSE): 846,488.4748
  - Summary: The SARIMA model further enhances forecasting accuracy, achieving the lowest MAPE and RMSE. It effectively captures seasonality, trends, and autocorrelation in the data, making it a robust choice for accurate and reliable forecasts.

In summary, the historical mean served as a benchmark, the triple exponential smoothing and SARIMA models significantly improved forecasting accuracy. Among the two, the SARIMA model stands out as the most reliable choice for forecasting Walmart stores Total Sales.

Individual Store Sales Forecasting

As mentioned at the beginning of the chapter, The second approach focused on constructing models that predict the weekly sales of each store separately. This approach allows for a more granular analysis, helping identify specific store performance and trends.

Train Test Split

Before model building, I configured the training and testing set. Configuration was similar to the Total Sales forecasting train test split configuration. The only difference in this approach was instead of Total aggregated weekly sales, I made a separate DataFrame for store no 1. And then applied a train test split.

The same methodology can be applied to the remaining 44 stores. However, for this project, I limit myself to the analysis of store no 1 only.

```
store_1 = df.loc[df['Store'] == 1] # creating seperate DataFrame for store 1

train = store_1.iloc[: -12, 1] #spliting stor 1 data into train
test = store_1.iloc[-12:, 1] #spliting stor 1 data into test

train = pd.DataFrame(data = train) #saving data back to the train Dataframe
test = pd.DataFrame(data = test) #saving data back to the test Dataframe
```

Figure 37 Store 1 train-test split

Data Decomposition

Similar to the Total Sales decomposition, Weekly sales decomposition was carried out for store 1 and is shown in figure 38

Figure 38 Weekly Sales Decomposition for Store 1

Forecasting Models

As part of modeling process, I conducted a dedicated analysis for Store 1 to fit forecasting approach to its specific sales patterns. I evaluated the performance of three forecasting methods: historical mean, triple exponential smoothing, and SARIMA models.

The procedure for model building using these methods was similar to the model building procedure followed for total sales forecasting and is mentioned in the section 'Total Sales Forecasting'.

The results of the forecasting models for the store 1 is summarized below:

Historical Mean Forecasting (Store 1):

- Mean Absolute Percentage Error (MAPE): 0.0387
- Root Mean Squared Error (RMSE): 68,624.5866
- The historical mean forecasting approach resulted in a relatively high MAPE and RMSE, indicating limited accuracy in predicting Store 1's sales.

Triple Exponential Smoothing (Store 1):

- Mean Absolute Percentage Error (MAPE): 0.0285
- Root Mean Squared Error (RMSE): 54,118.5502
- Implementing triple exponential smoothing significantly improved forecasting accuracy for Store 1 compared to historical mean forecasting, with a notable reduction in both MAPE and RMSE.

SARIMA Model (Store 1):

- Mean Absolute Percentage Error (MAPE): 0.0246
- Root Mean Squared Error (RMSE): 47,149.4948
- The SARIMA model further enhanced forecasting precision, achieving the lowest MAPE and RMSE among the three methods. It is poised to provide highly accurate sales predictions tailored to Store 1's unique sales patterns.

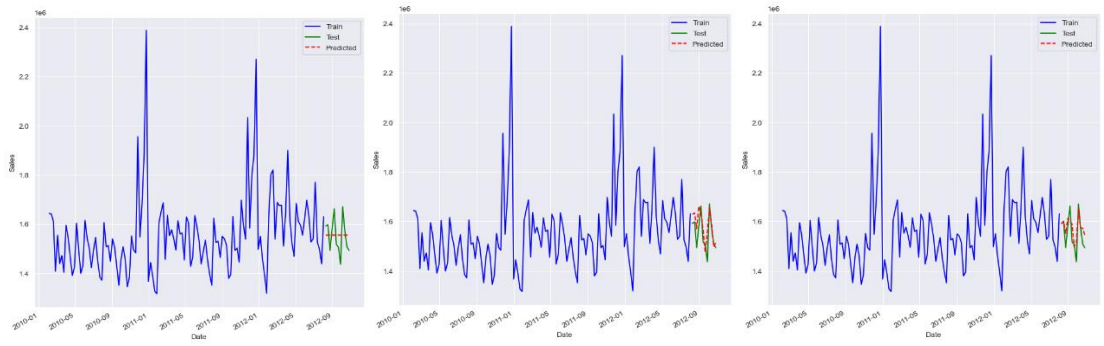


Figure 39 Forecasts for Store 1

In summary, analysis for Store 1 revealed substantial improvements in forecasting accuracy compared to historical mean forecasting. Both triple exponential smoothing and the SARIMA model demonstrated their effectiveness in capturing the underlying

trends and seasonality of Store 1's sales data, with the SARIMA model offering the highest level of accuracy.

Inferences

- **Impactful Factors on Walmart Sales:**
  - Through comprehensive data analysis, I identified several key factors influencing Walmart's weekly sales. These factors include holidays, temperature, consumer price index (CPI), and unemployment rate.
  - I observed that holiday weeks consistently led to higher sales, with Thanksgiving and Christmas showing significant spikes.
  - Temperature had varying effects on different store categories, indicating that weather plays a role in shopping behavior.
- **Store Performance:**
  - I ranked stores based on historical data, revealing the top-performing stores generating the highest sales. Store size and location likely play a role in performance, with stores 20, 4, 14, 13, and 2 consistently leading in revenue generation.
- **Time Series Forecasting:**
  - Utilizing advanced time series forecasting techniques, I built and evaluated models for total Walmart sales and individual store sales.
  - I compared various methods, including historical mean, triple exponential smoothing, and SARIMA models, to identify the most accurate forecasting approach.
- **Model Performance:**
  - The evaluation of forecasting models revealed that the SARIMA model consistently outperformed other methods in terms of accuracy.
  - It demonstrated lower Mean Absolute Percentage Error (MAPE) and Root Mean Squared Error (RMSE), making it a suitable choice for predicting Walmart sales.
- **White Noise Residuals:**
  - To ensure the quality of our forecasting models, I conducted residual analysis.
  - I observed that the residuals displayed white noise properties, indicating that the models effectively captured the underlying patterns in the data.
- **Store-Level Insights:**
  - I applied forecasting models to individual store, allowing us to tailor predictions for specific locations.
  - This granularity provides Walmart with valuable insights for inventory management, staffing, and sales optimization at the store level.
- **Seasonal Trends:**
  - The analysis unveiled consistent seasonal patterns in Walmart's weekly sales, with peaks occurring in the second, sixth, and eighth months, particularly in December. These patterns are influenced by major holidays, highlighting the importance of seasonality in sales forecasting.
- **Model Selection:**

- The project demonstrated that choosing an appropriate forecasting model is crucial for accurate predictions.

- While simpler models like historical mean and triple exponential smoothing provided reasonable results, the SARIMA model, with its ability to capture seasonality and trends, consistently outperformed other methods.
- This highlights the importance of utilizing advanced time series models for sales forecasting tasks, especially in complex retail environments like Walmart.

In conclusion, the comprehensive analysis of Walmart's weekly sales dataset has provided valuable insights into the factors influencing sales patterns and the effectiveness of various forecasting models. I found that seasonality, especially related to major holidays, plays a significant role in shaping sales trends. Variables such as unemployment, temperature, fuel price, and CPI impact sales differently across different store categories. The exploration of forecasting models revealed that while simpler methods like historical mean and triple exponential smoothing yield reasonable results, the SARIMA model, with its ability to capture seasonality and trends, proved to be the most reliable for predicting sales accurately. This project underscores the importance of data-driven decision-making in retail, emphasizing the significance of advanced time series models for sales forecasting and the potential for optimizing strategies to maximize sales and profitability.

Future Possibilities

Looking ahead, there are several exciting possibilities for expanding upon this project.

One promising direction is the application of deep learning methods to enhance sales forecasting accuracy. Deep neural networks, such as recurrent neural networks (RNNs) and long short-term memory networks (LSTMs), can capture intricate temporal patterns and dependencies within the data, potentially leading to even more precise predictions.

Additionally, incorporating external data sources, such as social media trends, economic indicators, and competitor data, can further enrich the forecasting models.

Overall, the future of this project holds great potential for leveraging advanced machine learning techniques to continually improve sales forecasting and support data-driven decision-making in the dynamic world of retail.

Conclusion

In this comprehensive project, I started a journey to tackle the complex task of sales forecasting for Walmart Dataset, Walmart is one of the world's largest and most dynamic retail giants. My analysis began with a thorough exploration of historical sales data, encompassing 45 stores located across diverse regions. I delved into the impact of various factors, from holidays and temperature to economic indicators like unemployment and consumer price index, unveiling their intricate relationships with weekly sales. Through exploratory data analysis, I gained insights into seasonality, trends, and store performance, shedding light on critical aspects of Walmart's sales patterns.

As I entered into modeling phase, I explored a range of forecasting techniques, from traditional methods like historical mean and triple exponential smoothing to advanced models like SARIMA. My rigorous evaluation process revealed that while simpler models provided reasonable results, the SARIMA model stood out as the most effective choice, showcasing its ability to capture the intricate seasonality and trends in the sales data. This project underscores the value of data-driven decision-making in the retail sector and highlights the potential for leveraging sophisticated machine learning approaches to enhance sales forecasting accuracy. With an eye toward the future, the possibilities for further improvements, including the integration of deep learning techniques and additional data sources, promise to make this project an ongoing asset for Walmart's optimization of inventory management and sales strategies.

References

1. Peixeiro, M. (2022) Time Series Forecasting in Python. Manning Publications Co., Shelter Island, 51.
2. Box, George EP, et al. Time series analysis: forecasting and control. John Wiley & Sons, 2015.
3. https://towardsdatascience.com/understanding-the-seasonal-order-of-the-sarima-model-ebef613e40fa