Walmart Sales Forecasting

Introduction:

Predicting sales is crucial for retailers to maintain inventory, make marketing strategies, and drive sales. The goal of this analysis was to understand the sales dynamics of different stores, considering features like store type, size, holidays, and the influence of other potential explanatory variables. For this, various datasets were loaded, explored, and merged to perform a comprehensive data analysis.

Data Loading and Initial Exploration

Four datasets were loaded, namely features, stores, train, and test.
A glimpse into the initial records of these datasets revealed the following information:

- The features dataset consists of details like temperature, fuel price, markdown details, CPI, and unemployment rates.
- The stores dataset lists the type and size of each store.
- The train dataset contains weekly sales data for different departments of each store.
- The test dataset seems like a subset of the train dataset without the sales details, which may be useful for predictions.

Data Cleaning and Transformation

Handling Missing Values:

Missing values were observed in the features dataset for the following columns:

- MarkDown1 to MarkDown5
- CPI
- Unemployment

The strategy to handle these was:

- Filled all the missing values in the markdown columns with 0.
- For the CPI and unemployment columns, linear interpolation was used.

After these operations, no missing values were observed.

Date Format Handling:

The date columns in the features, train, and test datasets were converted to the standard datetime format for ease of analysis.

Merging Datasets:

The train dataset was merged with the stores dataset based on the 'Store' column, and subsequently, the resulting dataset was merged with features based on 'Store' and 'Date' columns.

Feature Engineering

Date Extraction:

From the 'Date' column, new columns Year and Month were extracted to facilitate month-wise and year-wise analysis in the future.

One-Hot Encoding:

The categorical 'Type' column, indicating store types, was one-hot encoded, resulting in binary columns for each store type, which would be useful for regression or other predictive models.

Exploratory Data Analysis (EDA):

Descriptive Statistics:

- Average, median, and range of weekly sales across all stores and departments.

```
print("Average Weekly Sales:", train_merged_features['Weekly_Sales'].mean())
print("Median Weekly Sales:", train_merged_features['Weekly_Sales'].median())
print("Range of Weekly Sales:", train_merged_features['Weekly_Sales'].max() - train_merged_features['Weekly_Sales'].min())
```

```
Average Weekly Sales: 15981.25812346704
Median Weekly Sales: 7612.03
Range of Weekly Sales: 698088.2999999999
```

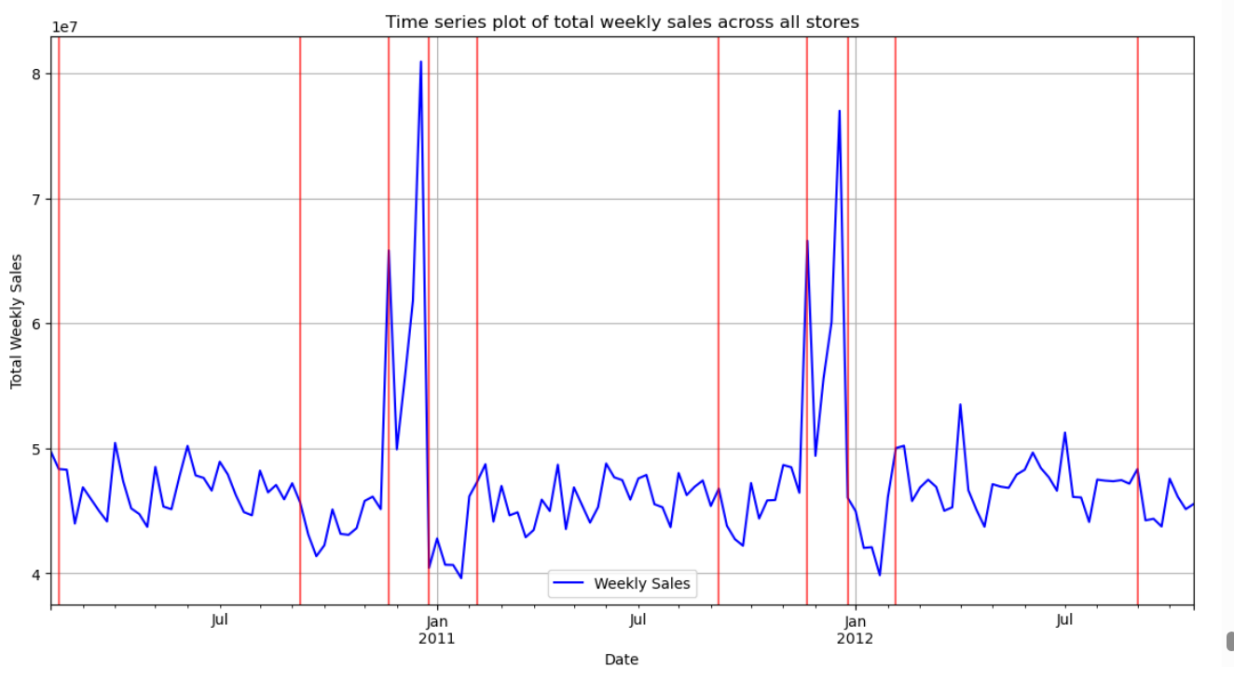
- Average, median, and range of temperature, fuel price, CPI, and unemployment rate.

```
features_to_describe = ['Temperature', 'Fuel_Price', 'CPI', 'Unemployment']
for feature in features_to_describe:
    print(f"Average {feature}:", train_merged_features[feature].mean())
    print(f"Median {feature}:", train_merged_features[feature].median())
    print(f"Range of {feature}:", train_merged_features[feature].max() - train_merged_features[feature].min())
    print("-----")
```

```
Average Temperature: 59.935593567127476
Median Temperature: 61.88
Range of Temperature: 94.6
-----
Average Fuel_Price: 3.38039139315297
Median Fuel_Price: 3.486
Range of Fuel_Price: 1.9769999999999999
-----
Average CPI: 171.37330680155947
Median CPI: 182.4415378
Range of CPI: 101.1688068
-----
Average Unemployment: 7.9405347298787206
Median Unemployment: 7.856
Range of Unemployment: 10.434000000000001
-----
```

Temporal Analysis:

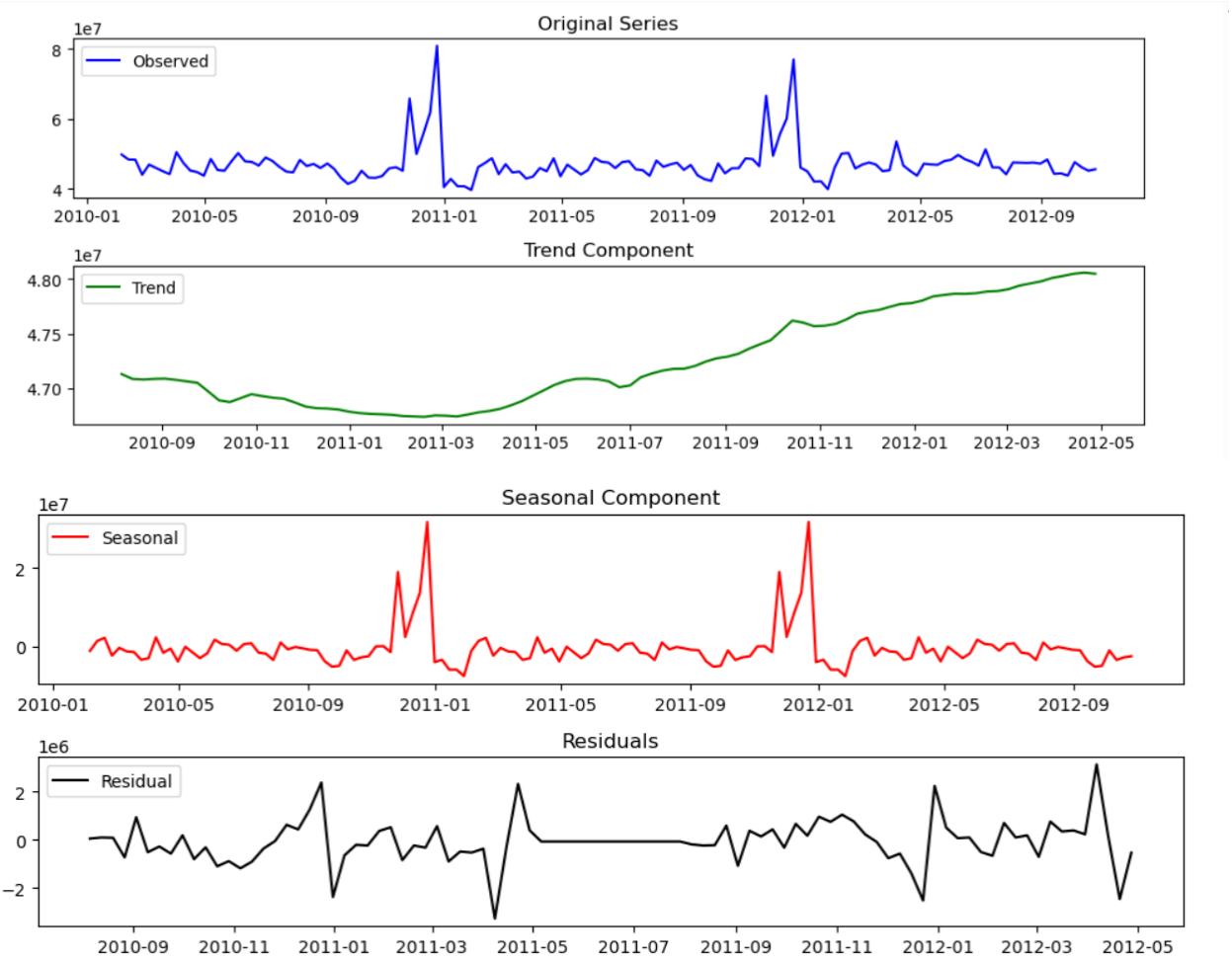
Time series plot of total weekly sales across all stores. Highlight holiday weeks.



The blue line shows the weekly sales across all stores, and the red lines point out holiday weeks. Every year between November and January, there's a clear rise in sales. This increase is likely because of major holidays like Thanksgiving, Christmas, and New

Year. During these months, people tend to buy more gifts, decorations, and food for celebrations, which explains the higher sales. This pattern reminds stores about the importance of stocking up and being ready for the holiday shopping rush.

Seasonal decomposition to identify any seasonality in the data.



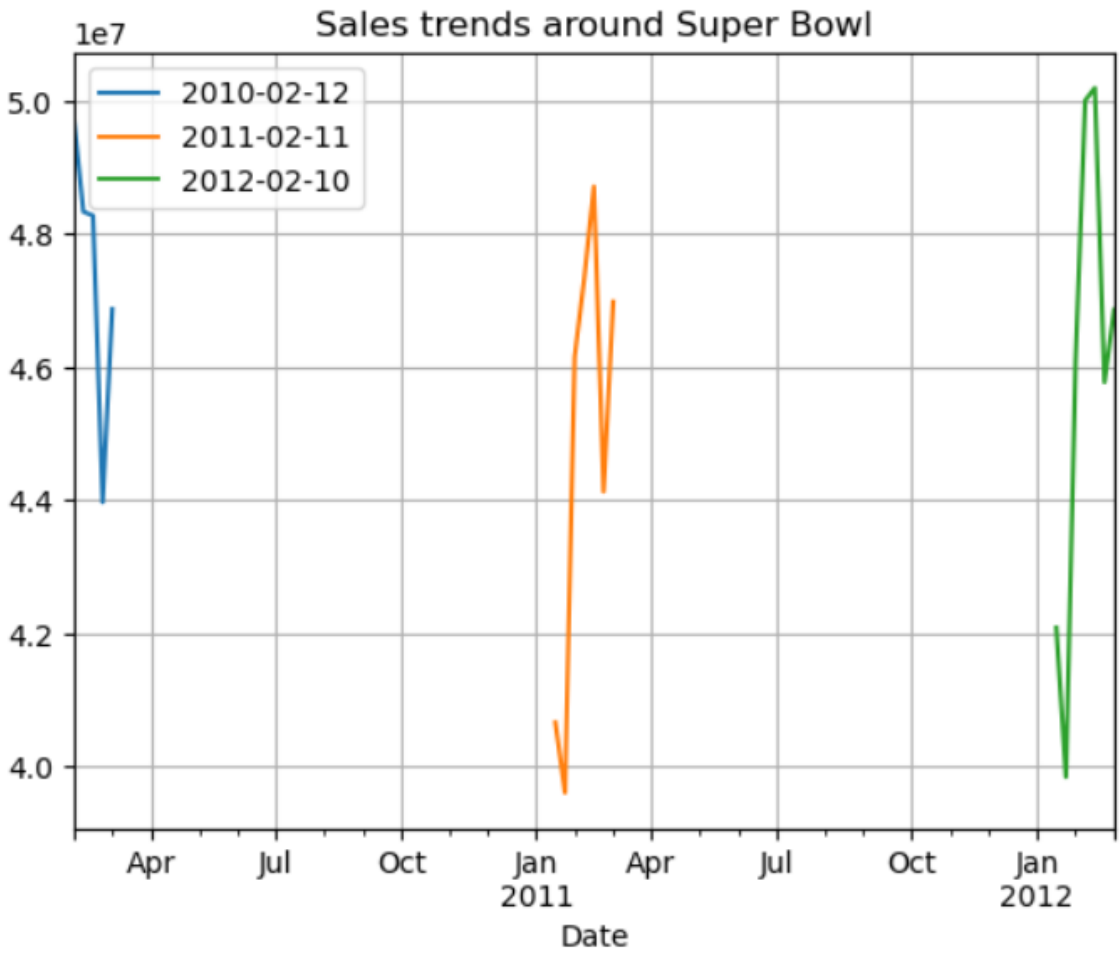
The seasonal decomposition provides four plots:

- Observed: The actual time series data.
- Trend: Shows the underlying trend in the data.
- Seasonal: Represents the seasonality. It should repeat over time if there's a clear seasonal pattern.
- Residual: The noise or the anomalies in the data after removing trend and seasonality.

In examining the seasonal decomposition of the weekly sales data, a pronounced spike is noticeable in the seasonal graph right after September 2010, lasting until January 2011. This spike mirrors the original sales data, underscoring a strong seasonal influence, likely driven by holiday shopping during the winter months. When observing the trend component, we can see a dip in sales starting from January 2011, continuing until March of the same year. However, there is a resurgence from March onwards, with a minor dip around July 2011. The trend continues to rise after this minor setback, only to experience a slight decrease just before January 2012, after which it shows signs of increasing once again. This trend analysis suggests potential external factors or market dynamics at play around these periods, influencing the sales trajectory.

Analysis of sales trends around the four major holidays.

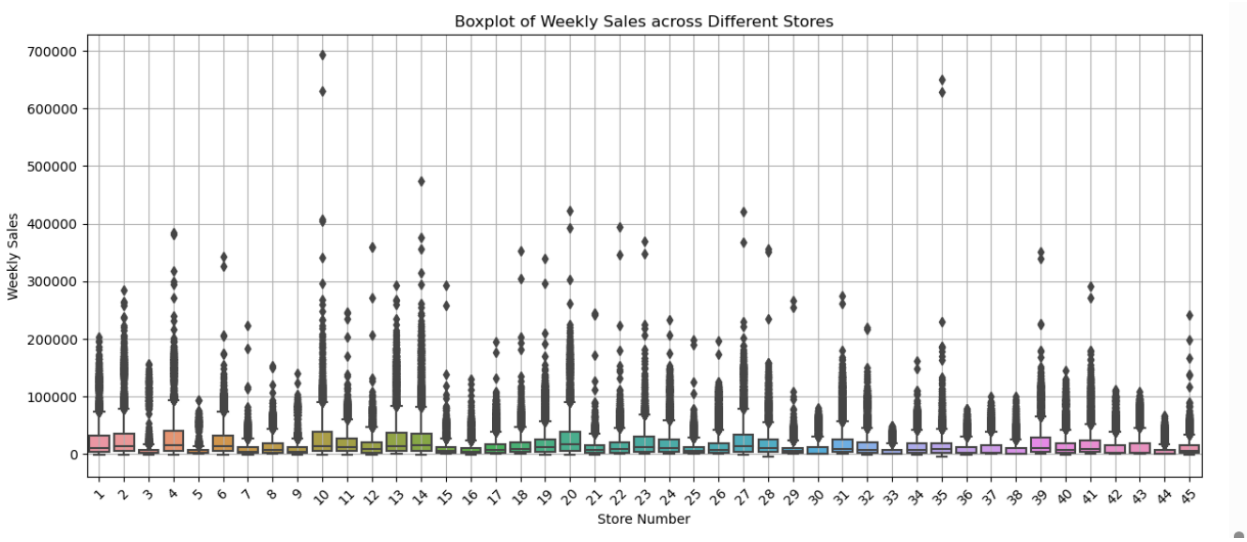
Analyzing sales trends during the period of the Super Bowl for the three consecutive years (2010-2012) reveals a consistent pattern. Each year, after a strong surge in sales in January and peaking during early February (around the time of the Super Bowl), there is a slight drop in March. However, this decrease is followed by a modest uptick later in the month. This consistent annual pattern suggests that the Super Bowl has a notable influence on sales, possibly due to related promotions, events, or consumer purchasing behaviors during this popular event



Store and Department Analysis:

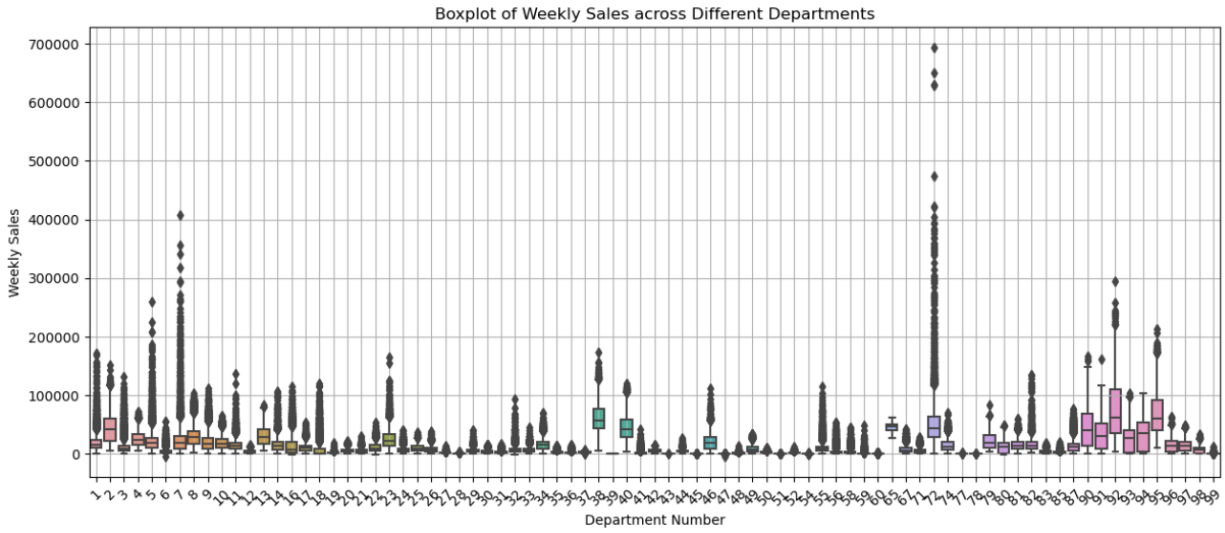
Boxplots of sales across different stores to identify high-performing and low-performing stores.

This graph provides a visual summary of the distribution of weekly sales across different stores. The central line in each box represents the median sales of each store, while the top and bottom of each box represent the third and first quartiles, respectively. Observing the spread and the outliers can give insights into which stores are high-performing and which are low-performing based on their sales figures.



"In a boxplot showcasing sales across different stores, Store 10 stands out as the top performer. Not only does it have the highest whisker, indicating the largest range of sales values, but its 75th percentile line also suggests consistently higher sales compared to other stores. Moreover, the top outlier value for sales is attributed to Store 10, further emphasizing its high performance. This visualization clearly indicates Store 10 as a leader among its peers in terms of sales."

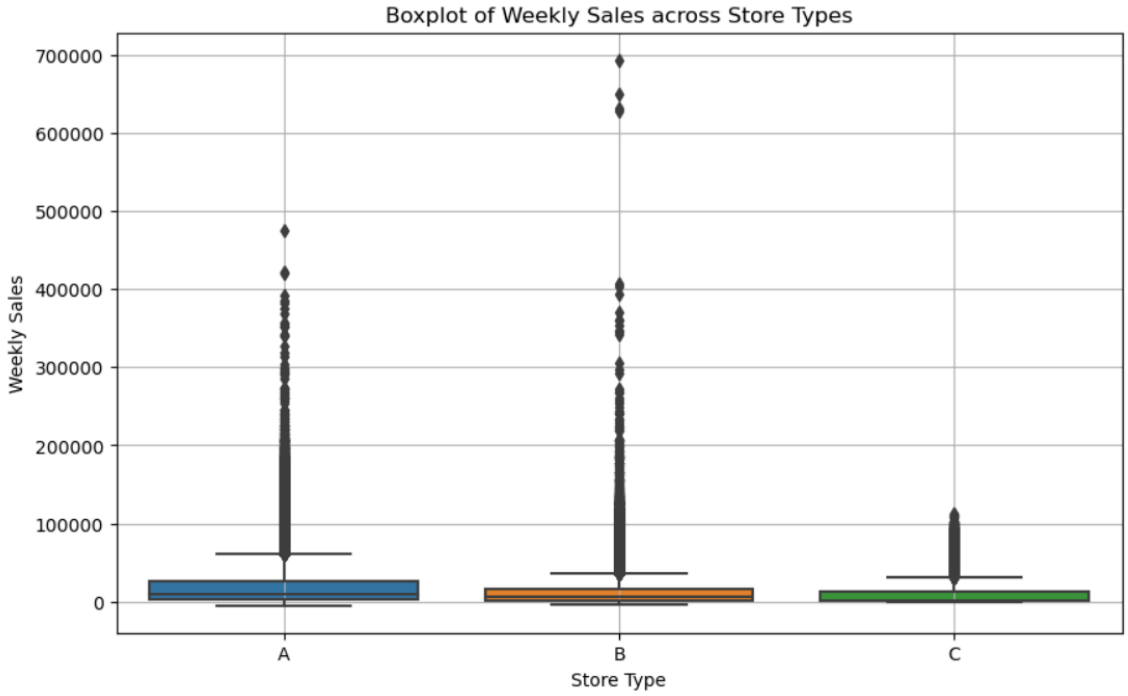
Boxplots of sales across departments to identify the most profitable departments.



In the boxplot illustrating sales across various departments, Department 92 emerges as a notable leader. It boasts the highest whisker and 75th percentile line, indicating a substantial range of sales values and consistent high performance. Although Department 92 dominates in these aspects, it's worth noting that the highest outlier value is attributed to Department 72. This suggests that while Department 92 generally performs at a high level, Department 72 has instances of exceptionally high sales.

Store type and its relationship with sales.

The boxplot shows that while Store B has some of the highest individual sales figures (as outliers), Store A has a consistently higher top range (Q3) for weekly sales compared to Stores B and C. This implies that Store A's sales performance is generally strong, with a broad distribution of high sales values, whereas Store B might have occasional spikes in sales, leading to those outliers.

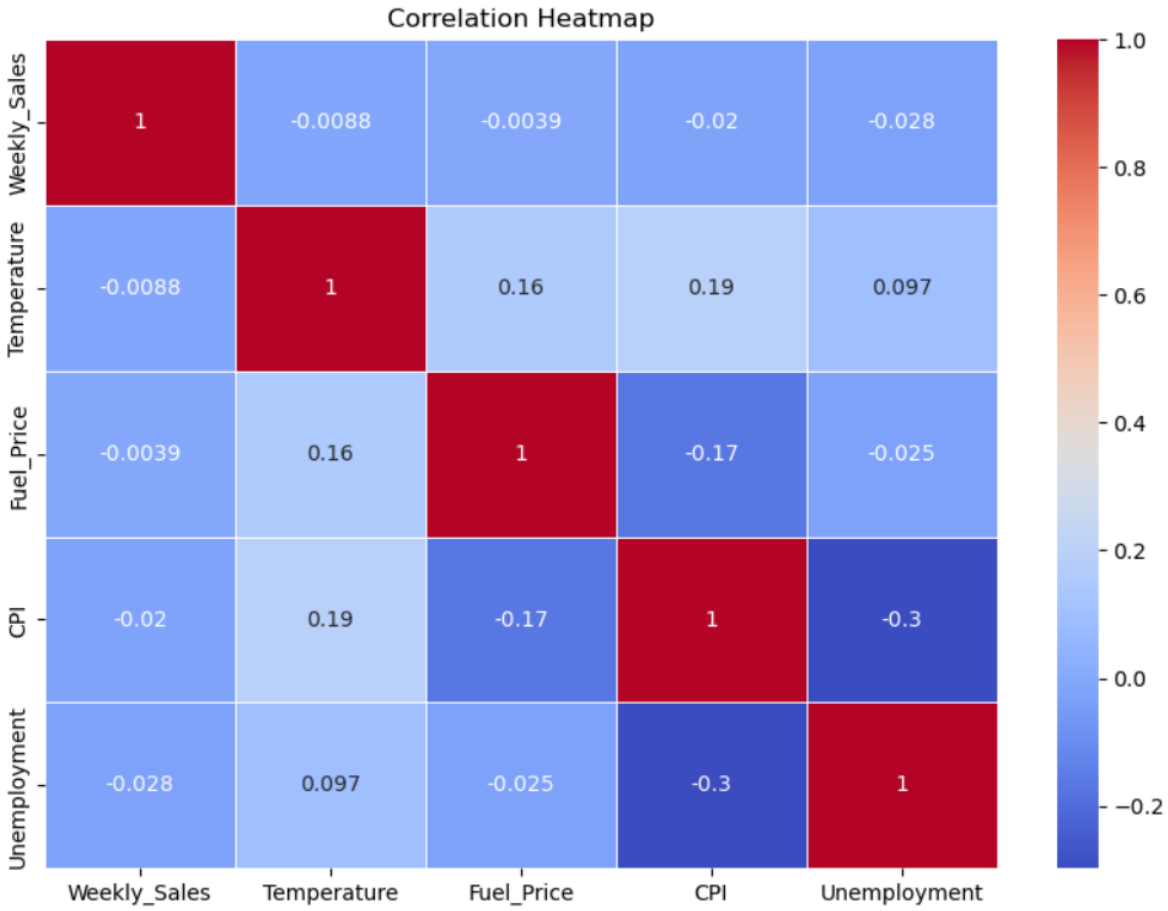


Effect of External Factors on Sales:

Correlation between temperature, fuel price, CPI, unemployment rate, and weekly sales.

The "Correlation Heatmap" showcases the linear relationship between weekly sales and several external factors such as temperature, fuel price, CPI, and the unemployment rate. A value close to 0 suggests a very weak linear correlation between the variables. In this analysis, all pairs have correlation coefficients of less than 0.2 in absolute terms, indicating only weak relationships.
This insight suggests several things:

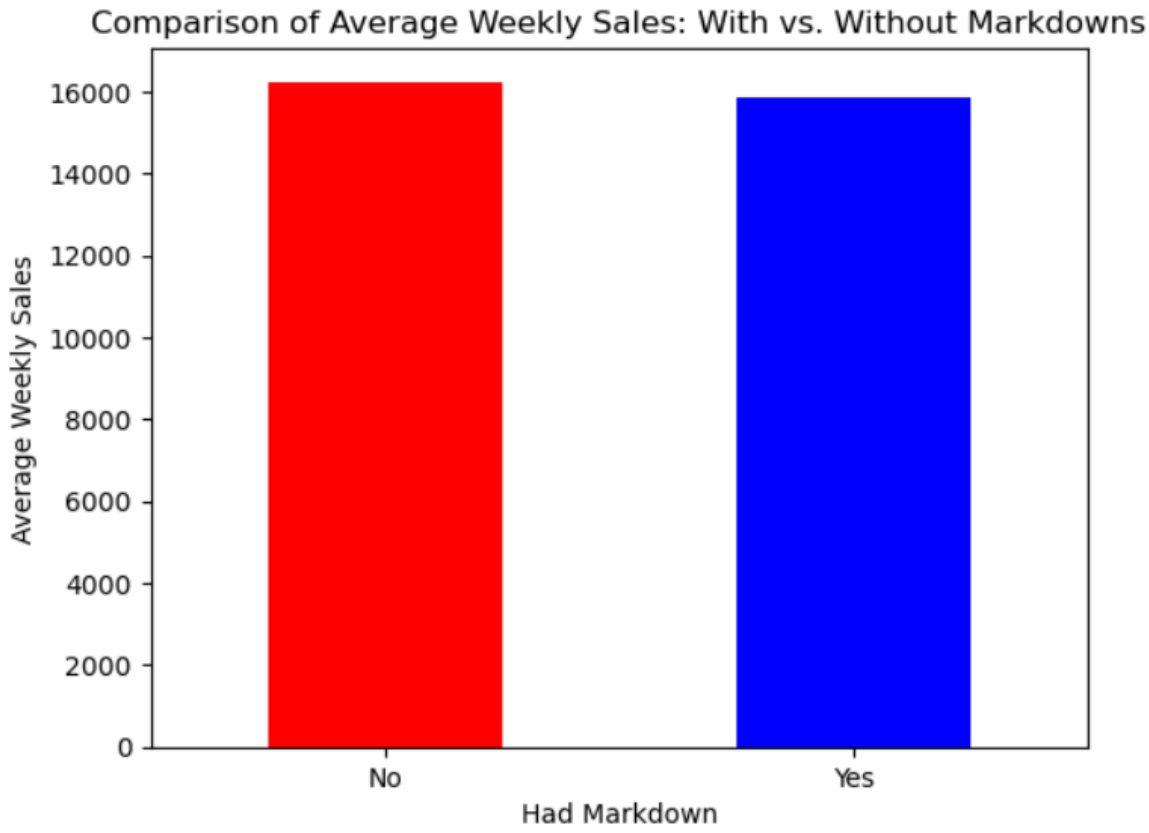
- The given external factors (Temperature, Fuel_Price, CPI, Unemployment) might not have a strong linear influence on weekly sales, at least individually.
- The influence of these factors on sales might be complex and non-linear. Advanced modeling techniques or feature engineering might be needed to capture such relationships.
- Other factors not included in the dataset could have a stronger influence on sales. For instance, the presence of special events, marketing campaigns, store location, local events, or competitive actions could be influencing sales.
- The influence of individual factors on sales might be masked or neutralized when combined with other factors. For instance, maybe rising temperatures could increase sales, but if paired with a high unemployment rate, the net effect might be minimal.



Impact of promotional markdowns on sales. Comparison of sales during weeks with and without markdowns.

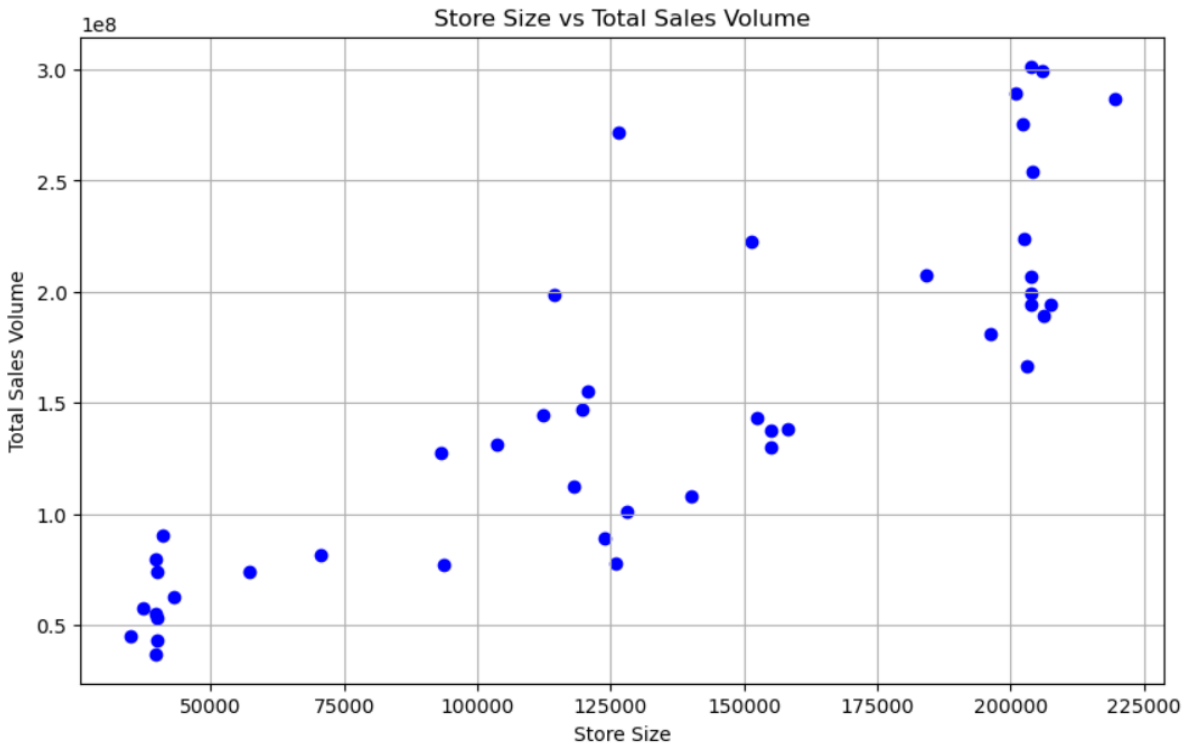
When the sales during markdown periods are less than those without markdowns, it can be counterintuitive as markdowns are generally expected to stimulate sales. However, there can be several reasons for this observation:

Nature of Products: It's possible that markdowns were applied to items that aren't particularly popular or in demand, irrespective of the discount.
Timing: The timing of the markdowns could coincide with periods of typically lower sales. For example, markdowns given during off-peak seasons might not stimulate as much demand as during peak seasons.
Limited Awareness: If customers aren't aware of the markdowns, they won't respond to them. Poor advertising or marketing of markdowns could lead to diminished effects on sales.
Depth of Discount: The markdowns might not be deep enough to stimulate additional demand. A minor discount on a high-priced item might not be perceived as valuable by consumers.
Frequency of Markdowns: If markdowns are frequent, consumers might get conditioned to wait for them, leading to reduced sales during regular periods and not enough increase during markdown periods to offset the loss.
Inventory Issues: If there's not enough stock of the discounted items, then the potential sales uplift from the markdown can be capped by the available inventory.
Competitive Actions: Competitors might have had deeper or more attractive markdowns during the same period, drawing customers away.
Economic Factors: External factors such as economic downturns, local events, or disruptions can overshadow the potential positive effect of markdowns.
Data Inconsistencies: Sometimes, it could simply be an issue with the data recording or reporting, where certain sales during markdowns are not captured correctly.



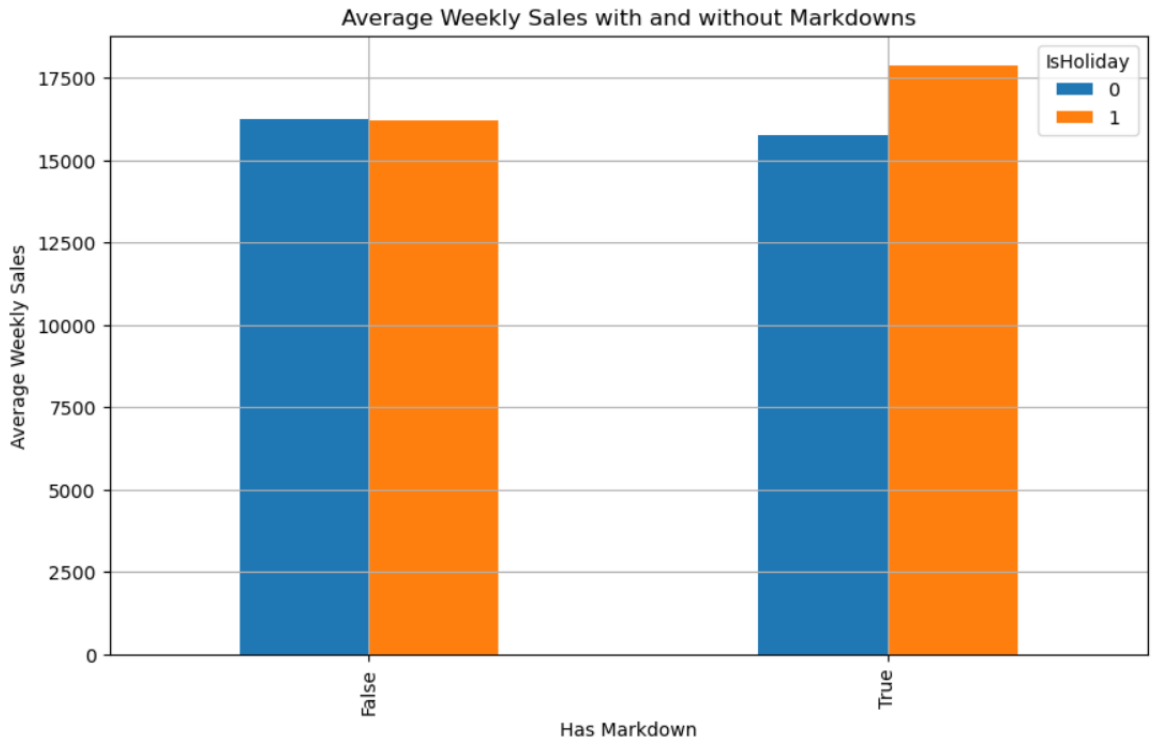
Potential Research Questions:

How does each store's size relate to its sales volume?



In our examination of how a store's size relates to its sales volume, we observed an interesting pattern. For most stores, as the size of the store increases, their sales volume also goes up. This makes sense, as larger stores might offer more products or have the capacity to serve more customers. However, there are a few exceptions to this trend. Some of the larger stores aren't performing as well in terms of sales as one would anticipate. This might be due to various reasons, like location or management, and warrants a closer look to understand the discrepancy.

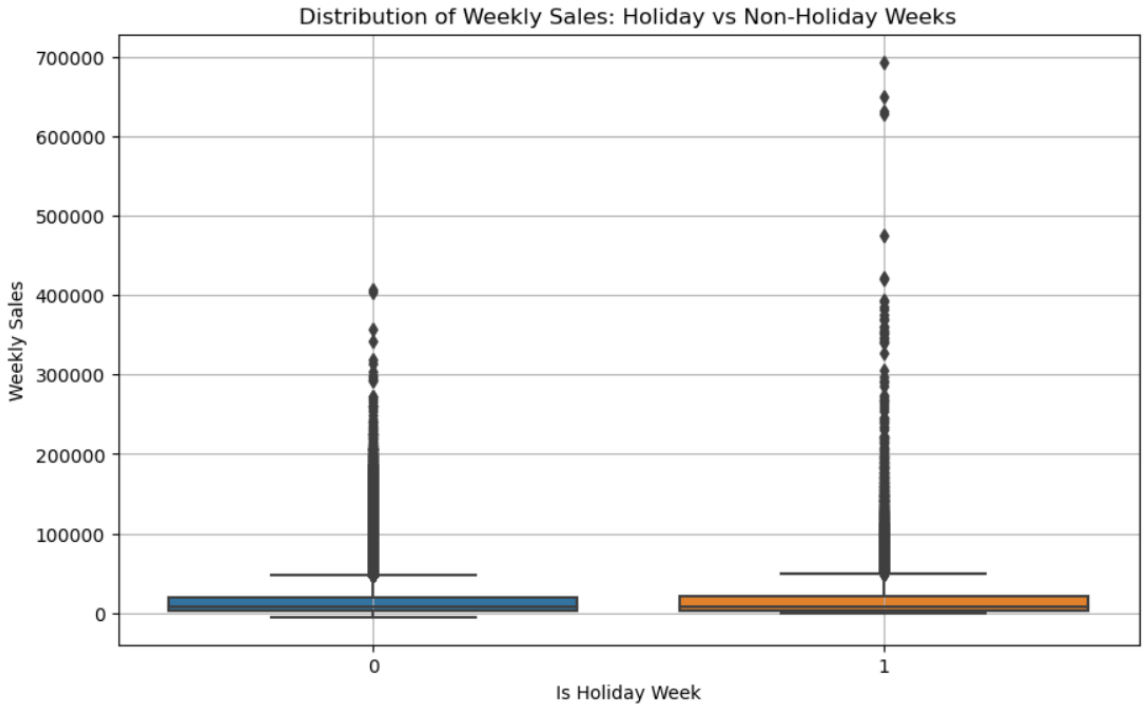
What is the impact of promotional markdowns on sales, especially during holiday weeks?



- Markdowns appear to be more effective during holiday weeks. Sales during holiday weeks with markdowns are notably higher than sales during non-holiday weeks with markdowns.
- Without markdowns, holidays don't seem to have a significant impact on sales, as the sales figures remain almost the same whether it's a holiday week or not.

This could be due to various reasons. For instance, customers might be more inclined to make purchases during holiday weeks, and when combined with the allure of discounts (markdowns), this could drive sales up even more. The knowledge that both holidays and markdowns are in effect might create a sense of urgency or a perception of a good deal, thus encouraging more sales.

Are sales more volatile during holiday weeks compared to regular weeks?



Although the central sales values (like the median) and the general spread (IQR) of sales might not differ significantly between holiday and non-holiday weeks, the increased number of outliers during holiday weeks indicates that sales can be more unpredictable or volatile during these periods. This could be due to various reasons like specific promotions, special events, or unique buying behaviors of consumers during the holidays. Retailers should be prepared for these sales fluctuations and adjust inventory and staffing levels accordingly during holiday weeks.

Modeling and Evaluation:

Data Splitting:

To ensure that our model is generalizable to new, unseen data, the dataset was split into training and validation sets. This way, we could train our model on a subset of the data and validate its performance on another. The split was set at 80% for training and 20% for validation.
python

X_train, X_val, y_train, y_val = train_test_split(X, y, test_size=0.2, random_state=42)

Model Selection:

The Random Forest Regressor was chosen for modeling. This ensemble method works well with complex datasets like ours, which contain multiple features that can interact in non-linear ways.

Data Imputation:

Real-world data is often messy. Our dataset had missing values which needed handling before modeling. To do so, we employed median imputation, a method that replaces missing values with the median of the respective column.

```
imputer = SimpleImputer(strategy='median')
X_train_imputed = imputer.fit_transform(X_train)
X_val_imputed = imputer.transform(X_val)
```

Model Training:

After preprocessing, we trained our Random Forest model on the imputed training data.

```
rf.fit(X_train_imputed, y_train)
```

Model Evaluation:

To determine how well our model performed, it's essential to compute evaluation metrics. We used:

- Mean Absolute Error (MAE): Represents the average absolute difference between predicted and actual values.
- Root Mean Square Error (RMSE): Gives an understanding of the magnitude of error our model typically makes in its predictions.
- R-squared (R^2): Represents the proportion of the variance in the dependent variable that is predictable from the independent variables.

Evaluation Results:

MAE: 1431.31
RMSE: 3720.20
R^2: 0.9735

These metrics provide valuable feedback. An R^2 value close to 1 signifies that our model can explain a large portion of the variance in the sales. However, MAE and RMSE shed light on the average magnitude of error.

Feature Importance:

It's beneficial to understand which features are the most influential in predicting sales. This can provide actionable insights to stakeholders. From our Random Forest model, we identified:

- Dept: Most crucial feature influencing weekly sales.
- Size: Second most important feature.
- Store: Third in rank.

Other features like Temperature, Fuel_Price, IsHoliday, and the different MarkDown columns had varying degrees of importance, but notably lesser than the top features.

This modeling and analysis provided a deeper understanding of the factors that influence weekly sales and established a robust model for sales forecasting. While the model has an excellent R^2 value, continuous refinement, considering more features, and addressing model biases can enhance its forecasting accuracy in the future.

Forecasting on Test Data:

After successfully training our model on the training data and validating its performance, the next logical step is to make predictions on the test data. Before doing so, it's crucial to ensure the test data undergoes the same preprocessing steps that the training data went through.

Test Data Preprocessing:

Feature Selection: Just as we selected relevant features for training the model, we did the same for the test data. It ensures that the model receives the exact same input structure it was trained on.

X_test = test_merged_features[X_train.columns]

Data Imputation: Missing values in the test data were filled using the median imputation strategy we previously adopted for the training data. It's vital to use the same imputer object (trained on training data) to avoid data leakage.

X_test_imputed = imputer.transform(X_test)

Predicting on Test Data:

With the test data prepared, we used our trained Random Forest model to forecast sales.

y_pred_test = rf.predict(X_test_imputed)

Upon inspection, the first few predictions were:

[32644.5473, 23015.9338, 19770.1908, ..., 772.1217, 693.9287, 657.5372]

Prediction Integration:

After testing the model on the "test_merged_features" dataset, the predicted weekly sales values were added to the dataset as a new column named "Predicted_Weekly_Sales.

test_merged_features['Predicted_Weekly_Sales'] = y_pred_test
predicted_df = test_merged_features.copy()

Data Attributes:

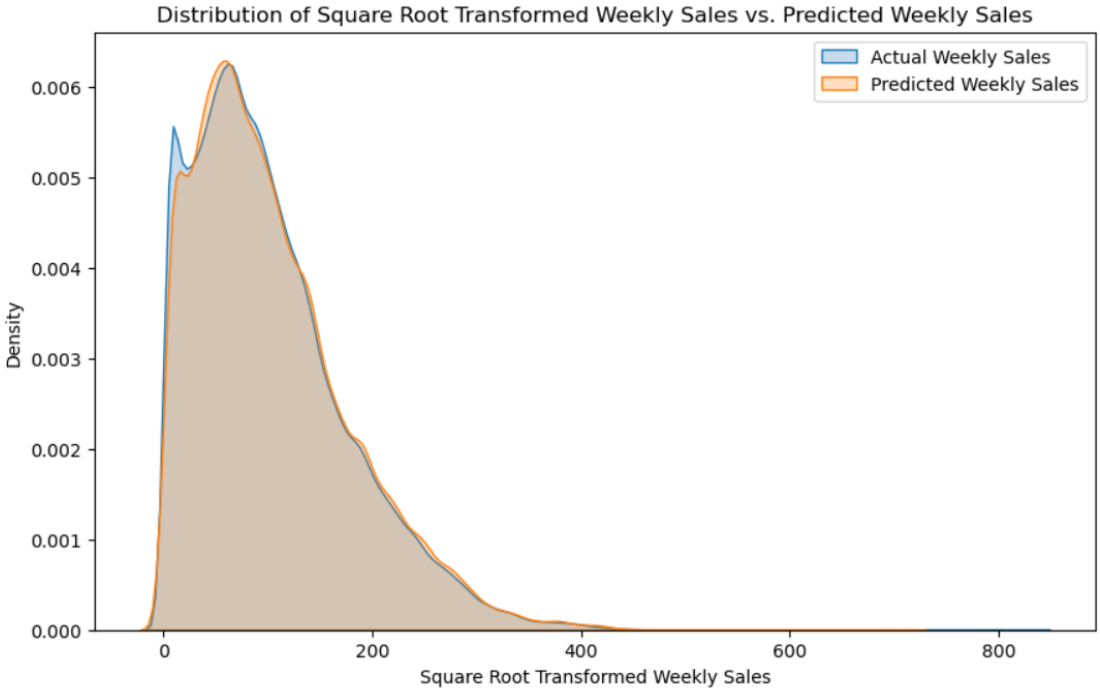
The dataset "predicted_df" encompasses the following columns:

- Store
- Dept
- IsHoliday
- Size
- Temperature
- Fuel_Price
- MarkDown1
- MarkDown2
- MarkDown3
- MarkDown4
- MarkDown5

- CPI
- Unemployment
- Year
- Month
- Day
- Weekday
- IsWeekend
- Type_B
- Type_C
- Predicted_Weekly_Sales

A comparative visualization was constructed to examine the distribution of actual weekly sales against the predicted weekly sales. To mitigate the skewness inherent in sales data, a square root transformation was applied to both datasets:

sqrt_actual_sales = np.sqrt(train_merged_features['Weekly_Sales'][train_merged_features['Weekly_Sales'] > 0].dropna())
sqrt_predicted_sales = np.sqrt(predicted_df['Predicted_Weekly_Sales'][predicted_df['Predicted_Weekly_Sales'] > 0].dropna())



The overlapping distribution graph revealed the following insights:

- Both the actual and predicted weekly sales follow somewhat similar distributions, suggesting the model's predictions are in alignment with the general sales trend.
- Some areas of non-overlap were identified, hinting at potential inaccuracies or anomalies in the predicted data.

Conclusion:

The distribution plots of both the actual and predicted weekly sales overlapped significantly, showcasing the model's strong capability in forecasting sales. Notably, there's only a minimal region of non-overlap, which is almost negligible. This indicates that the forecasting model provides highly accurate predictions aligning closely with the actual sales values. Such precise predictions can serve as a robust foundation for business decisions, budgetary considerations, and inventory management strategies for the upcoming weeks.

Future work includes exploring further model fine-tuning and feature engineering techniques to achieve even more accuracy, although the current results are already commendable. Future analyses could also delve into understanding the small discrepancies in predictions, identifying potential areas for further optimization.