# Final Report

## Walmart Sales Prediction

## Introduction

Walmart operates numerous stores across different locations, selling a variety of products. Accurate sales prediction is vital for effective supply chain management and satisfying customer demand. However, predicting sales accurately at the product level is challenging due to uncertainty caused by various factors, including seasonality, marketing campaigns, holidays, or other contextual information. Another challenge of modeling retail data is making decisions based on limited history. This project aims to overcome these challenges by building a reliable sales forecasting model. This project aims to develop an accurate and robust sales prediction model for Walmart. By accurately predicting future sales, Walmart can optimize inventory management, plan staffing requirements, and make informed business decisions to enhance profitability and customer satisfaction.

## Data Overview

The data is from Kaggle. It is with historical weekly sales data of 81 departments of 45 Walmart stores located in different regions from 2010-02-05 to 2012-11-01. The feature file contains supplementary details for a given date, such as Temperature, Fuel Price, MarkDown, CPI, Unemployment, and Holiday. Additionally, the store file provides information regarding the size and type of each store. Before preprocessing, the merged dataset has 421,570 records and 16 columns.

The list of columns:

- **`Store`**: The ID of the store.
- **`Dept`**: The ID of the department.
- **`Date`**: The date of the record.
- **`Weekly_Sales`**: The amount of sales in the given week. This is our target variable.
- **`IsHoliday`**: A boolean that indicates if the week is a holiday week.
- **`Temperature`**: The average temperature in the region.
- **`Fuel_Price`**: The cost of fuel in the region.
- **`MarkDown1`** to **`MarkDown5`**: Anonymized data related to promotional markdowns.
- **`CPI`**: The consumer price index.
- **`Unemployment`**: The unemployment rate.
- **`Type`**: The type of the store.
- **`Size`**: The size of the store.

## Exploratory Data Analysis

### Store and Department

The total number of available departments in this data is 81. However, the number of departments varies from store to store, with some having as many as 79 and others as few as 61 certain departments are not consistently available in all stores.

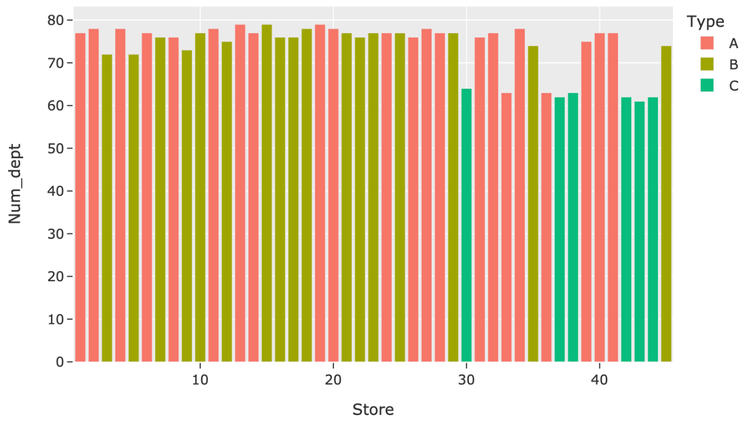


Figure 1. The total number of departments in each store

It appears that the number of departments in a store is related to its type. Stores categorized as Type A and B generally have around 75 to 79 departments, whereas Type C stores typically have 61 to 64 departments, with the exception of stores 33 and 36. These two stores are classified as Type A, but have a similar number of departments as Type C stores. It's possible that store type is an indicator of store size so that larger stores can have more departments. I use a box plot to verify this assumption to compare the size and type.

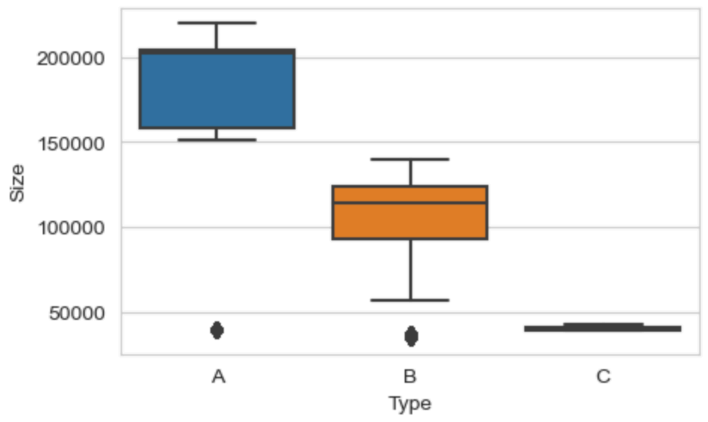


Figure 2. Store size vs type

The store types are differentiated by their size, with Type A being the largest and Type C being the smallest. It’s worth noting that there are only a few outliers where the sizes overlap. Based on figures 1 and 2, stores 33 and 35 show significant deviation from the expected patterns. I conclude that they have been mislabeled as type A. However, stores 3 and 5 do not clearly stand out from other type B stores.

According to Figure 3, Type A stores have the highest median sales, followed by Type B and Type C in that order. Type C stores tend to be clustered around the median. Comparatively, stores 33 and 35 have similar total sales to Type C stores, which further supports the above conclusion. However, there doesn't seem to be a clear relationship between store type and total sales.

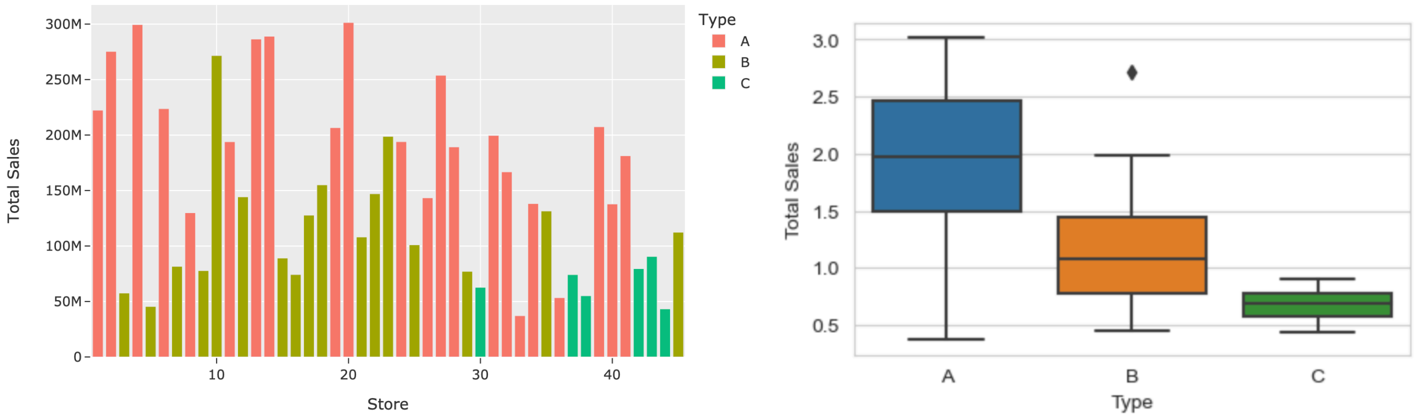


Figure 3. Total sales by store and type

22 departments are sold in all stores at all times. 32 departments are present in all 45 stores, but it is not consistently available. Some departments are exclusively offered in type A and B stores for a limited period. Additionally, department 65 is only sold year-round at store 34.

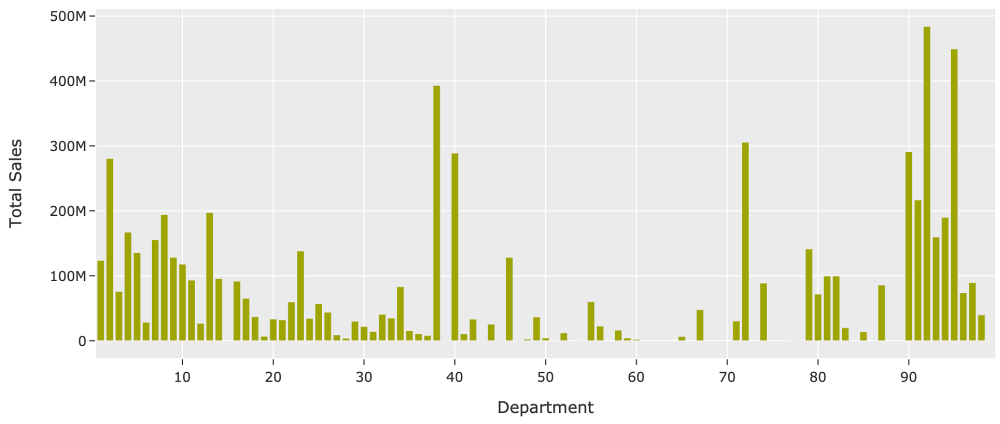
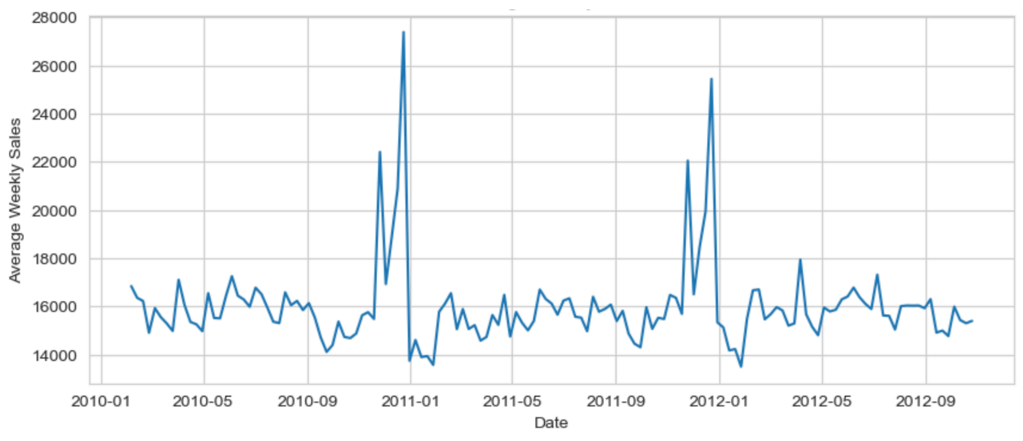


Figure 4. Total sales by department

## Sales Trends

Based on the average weekly sales plot, there seems to be a noticeable peak towards the end of the year that occurs consistently over several years. This indicates that there are recurring seasonal trends. This peak is associated with increased consumer spending during the holiday seasons. The average weekly sales per year plot exhibits a stable and consistent pattern over three years, with little variation or deviation. This suggests that the underlying factors driving sales



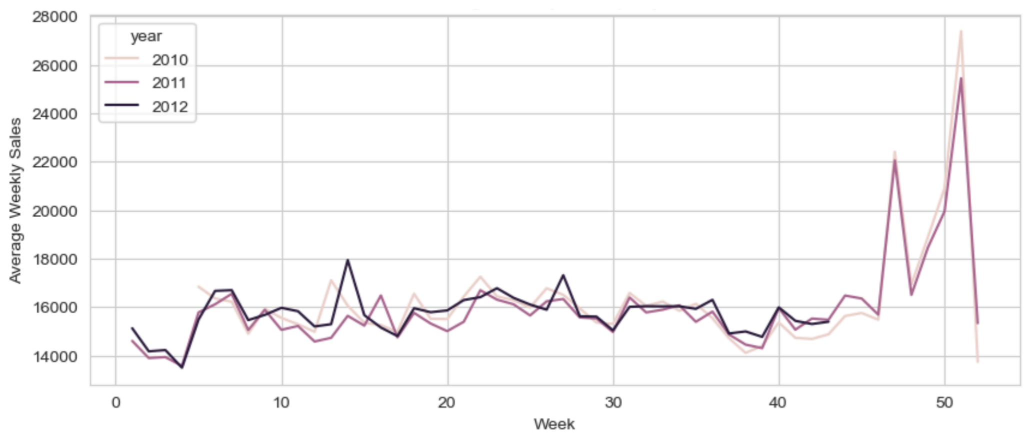


Figure 5. Average weekly sales over time

remain relatively constant over time.

If I break down the overall sales patterns into individual stores, I can observe the following. Sores of Type A and B have similar sales trends. Also, as they make up the majority of stores in this category, they reflect the overall trend. On the other hand, stores of Type C exhibit fluctuating weekly sales patterns without any clear seasonality.

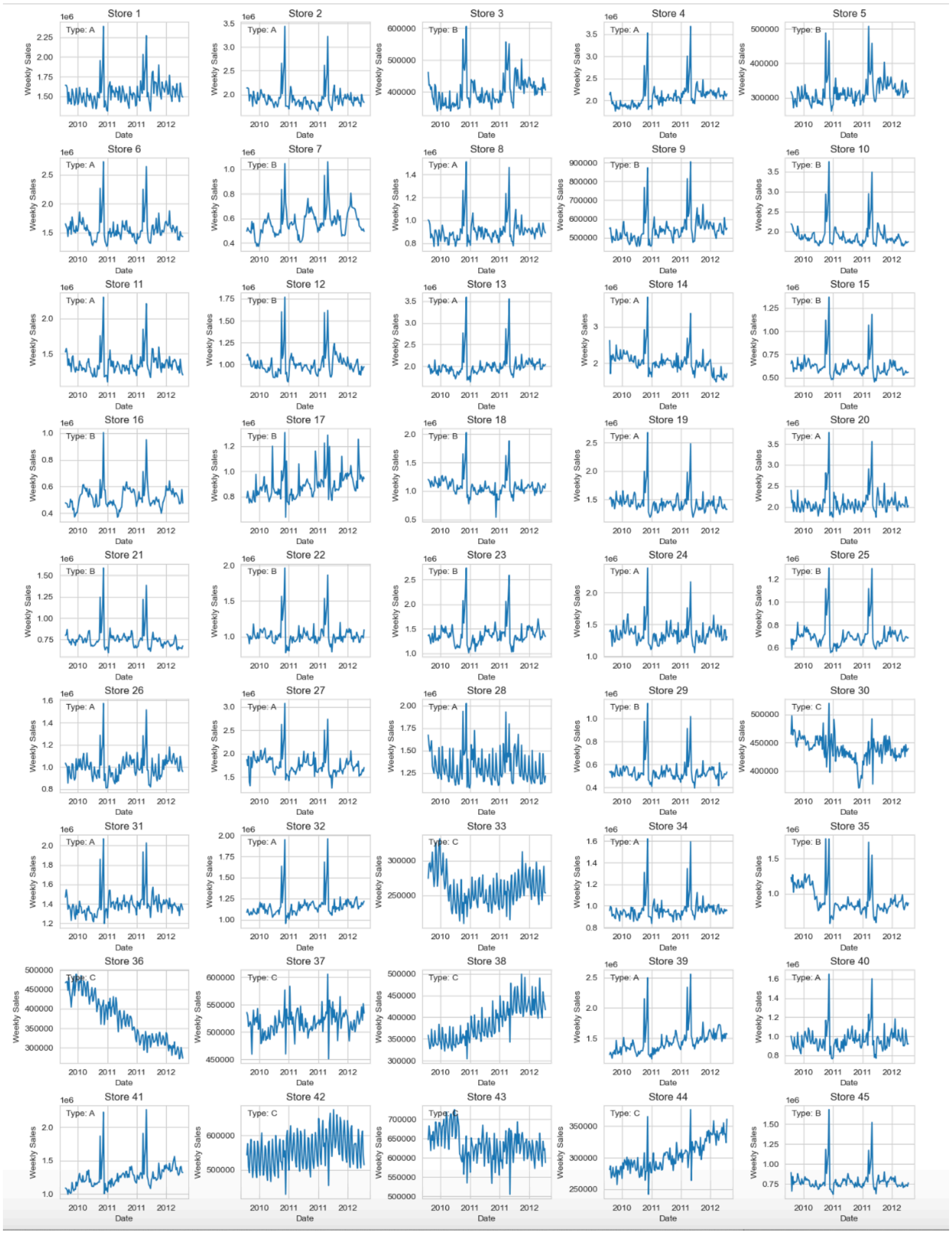


Figure 6. Average weekly sales by store

## Holiday

The sales patterns during peak periods can be largely influenced by holidays. Within the dataset, four holidays were identified: Super Bowl, Labor Day, Thanksgiving, and Christmas. Among these holidays, Thanksgiving has the highest average weekly sales, while Christmas surprisingly has the least impact on sales. However, it's important to note that December has the highest peak in sales throughout the year.  It makes sense that the sales effect of Christmas happened before Christmas since people tend to shop for Christmas gifts in advance. To investigate further, I added an extra column to analyze the sales impact of the week before and after each holiday. It is confirmed that the week leading up to Christmas sees the highest average weekly sales of the year.

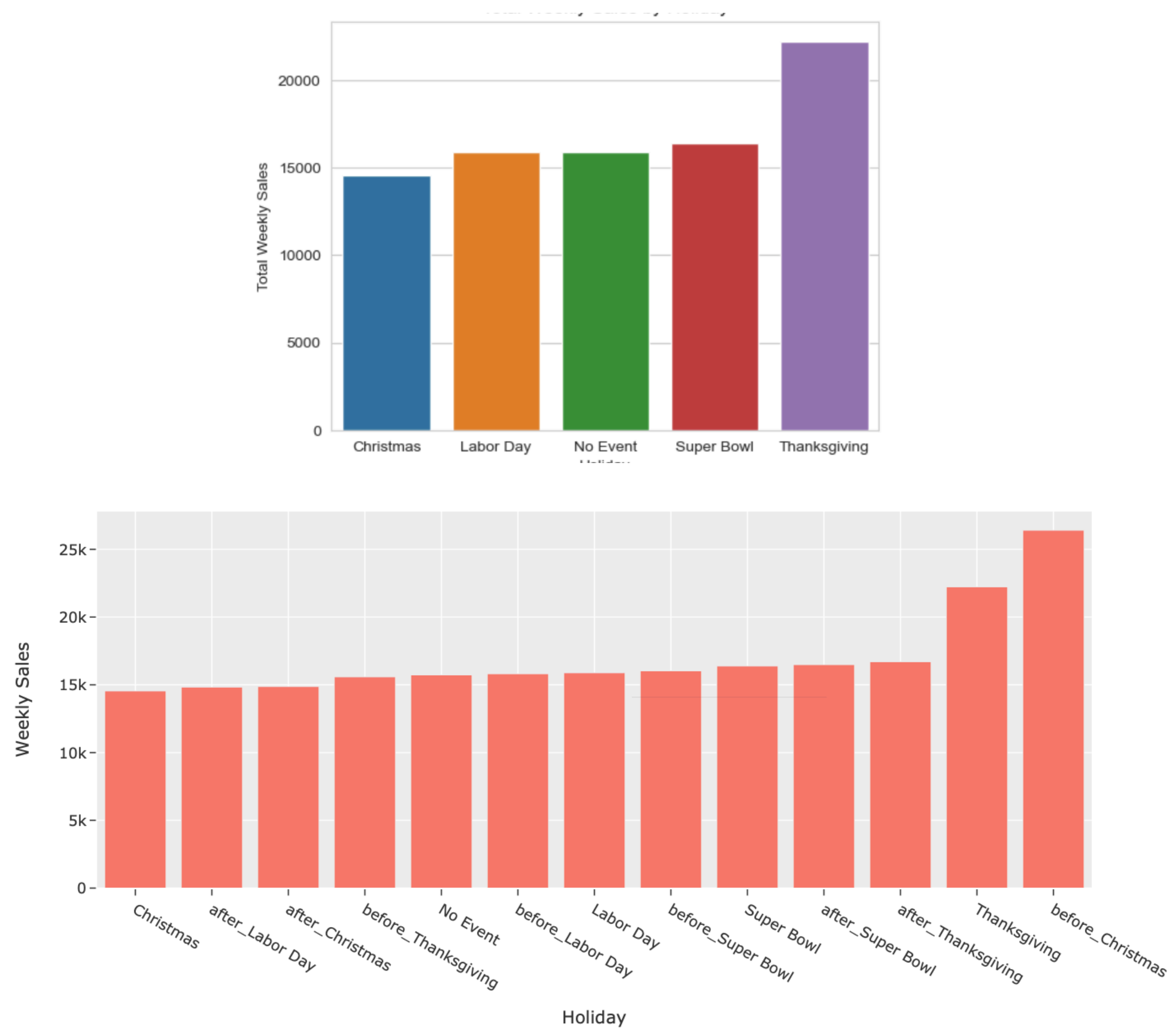


Figure 7. Average weekly sales by holiday

## Markdown

The Markdown columns contain anonymized data that relates to the promotional markdowns that Walmart is running. These columns are only available from November 2011 onward and were not present before that time. Although markdown promotions for holidays are known to affect sales, a significant number of missing values present a challenge when building a model to predict weekly sales with markdowns. To gain a better understanding of Markdown columns, a careful examination is required. To start, examine the markdown patterns and investigate if there are any meaningful relationships between Markdown columns and weekly sales.

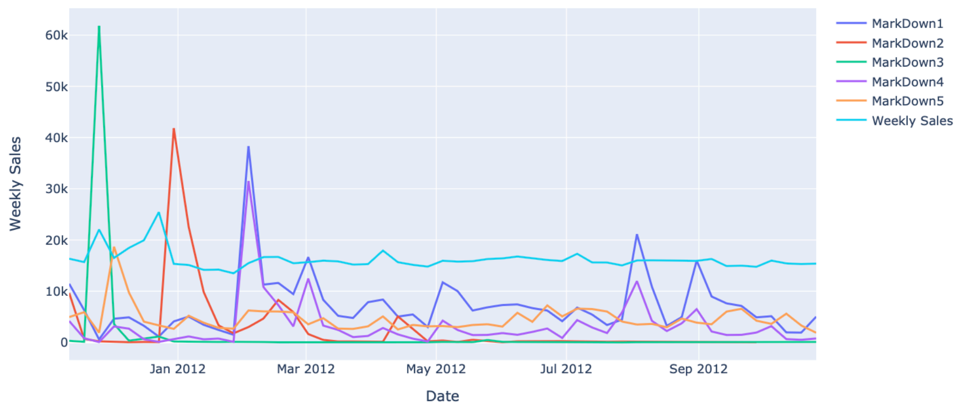


Figure 8. Markdowns and weeklys sales over time

In general, Markdowns exhibit a consistent pattern throughout the year, with one notable exception: a significant spike in sales leading up to the holidays. This peak is particularly extreme, as evidenced by the large difference between the maximum and mean values and a high standard deviation. Each promotional markdown event seems to precede this major holiday surge. Markdown 3, for example, experiences a dramatic increase of 61817.05 during Thanksgiving but remains stable during the remainder of the year with a median of 56 and a mean of 1348. Markdowns 1 and 4 experience peaks during the Superbowl, while Markdown 4 is specifically geared towards Christmas. Markdown 5 falls between Thanksgiving and Christmas.

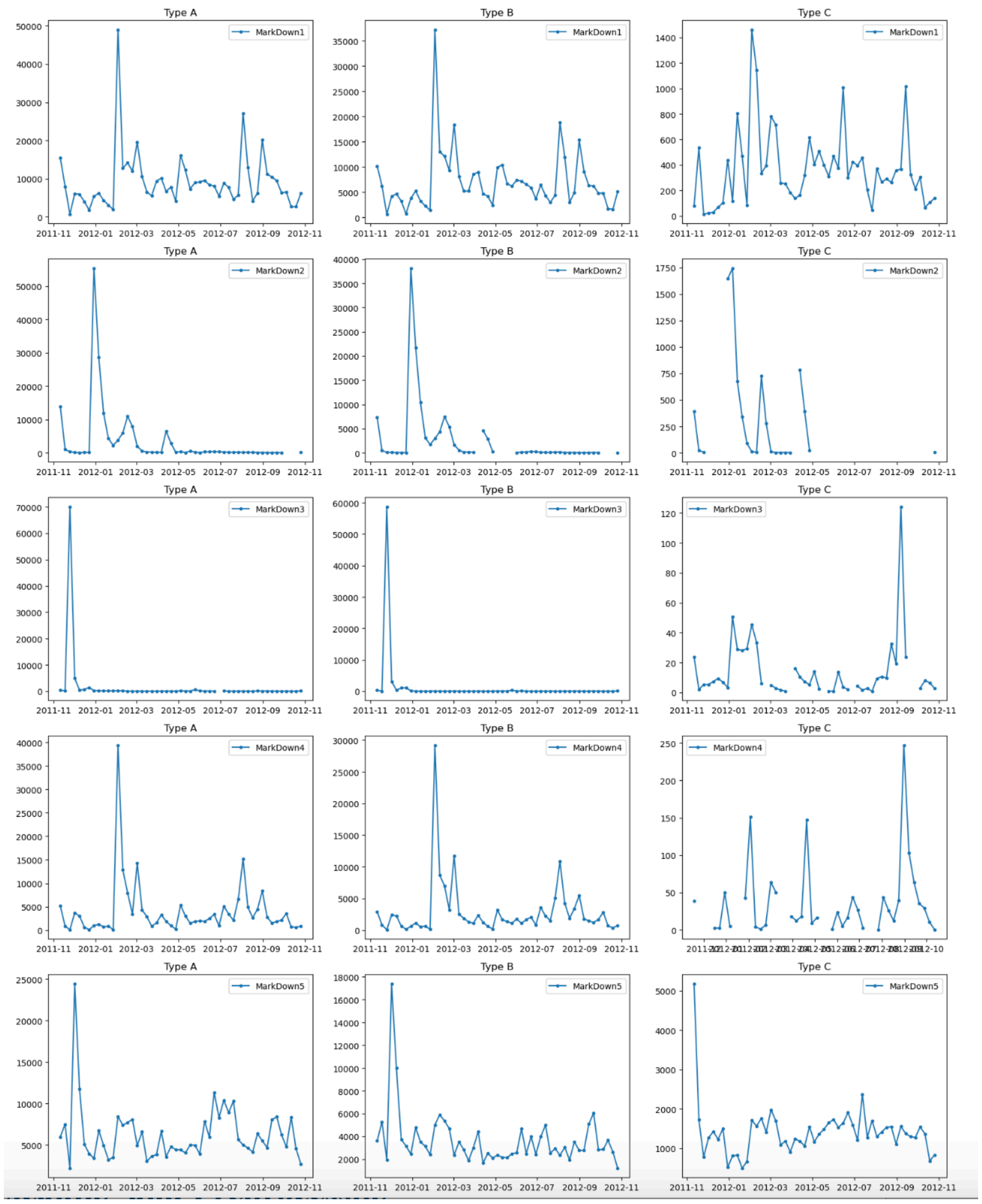


Figure 9. Markdowns by type

By categorizing the markdown patterns by type, we can see that stores classified as Type A and B have a similar pattern over time. In these stores, the majority of markdown columns have low percentages of missing values. On the other hand, Type C stores, despite being the smallest in size, have the highest percentage of missing values, Also, the plot indicates a different pattern compared to the other store types. Additionally, there are differences in the distribution of markdown values among store types, with A and B stores having wider ranges and higher mean and median values than C stores.

### Other Features

Weekly sales do not have clear correlations with temperature, fuel prices, CPI, or unemployment.

## Data Cleaning and Preprocessing

### Feature Engineering

This project aims to examine retail data with limited history and identify the factors that influence sales. While this analysis does not forecast future sales, examining the temporal aspect of the data can provide insights into the dynamics of sales and the relationships between sales and other variables. To achieve this, I generate date and time features, lag features, and a window feature to take these factors into account.

Based on the ACF and PACF plot, I created two Lag features, Lag1 and Lag4. Adding Lag1 as a feature enables the model to consider in the previous week's sales. This can capture the dependencies and incorporate the recent sales history as a predictor. The peak at lag 4 suggests that there may be a cyclical pattern occurring every four weeks. If there is a consistent fluctuation in sales every four weeks due to external factors like monthly promotions, Lag 4 helps the model capture this repeating pattern and its potential impact on the current week's sales. Note that data is organized hierarchically on a weekly basis. But not every department is consistently available in all stores at all times. Instead of creating lag features at the individual department level, I generated lag features for the store as a whole. This accounts for the temporal dependencies within each store.

The dataset included two categorical features, Type and IsHodilay. Through exploratory data analysis, it was discovered that store type is related to store size, so Ordinal Encoding was utilized for Type. IsHoliday denotes whether a holiday is observed or not. Each holiday has a distinct impact on sales. A new Holiday column was created to indicate the name of the holiday on each date. One-Hot encoding was used to generate binary columns for each holiday. During the Christmas season, there are significant sales in the weeks leading up to the holiday. In order to account for this, the WeeksBeforeChristmas column assigns corresponding values up until four weeks prior to Christmas.

## Imputing Missing Value

In markdown columns, missing values are not random, so it may not be appropriate to use imputation techniques such as mean, median, or mode. To address this issue, here are some possible strategies:

1. Fill missing values with zero and flag missingness with a binary column

It's important to note that a missing 'MarkDown' value doesn't necessarily mean that there was no markdown event during that week. To account for this, additional features were added to indicate whether the 'MarkDown' values were originally missing. This would enable the model to consider the possibility of missing data that isn't random.

2. Impute missing values

During the exploratory data analysis, it was observed that the average markdown pattern over time was related to specific holidays. Based on this insight, an effective strategy for imputing missing MarkDown values for each store could be to utilize MarkDown during the corresponding week. This approach would preserve the temporal patterns in the MarkDown data. Assuming that the patterns in 'MarkDown' remain consistent, the patterns observed after November 2011 can also be applied to the period before November 2011. However, if the 'MarkDown' strategies of the store have significantly changed over time, the imputed values may not be reliable. In this case, it would be more reasonable to follow the first strategy employed prior to November 2011.

3. Drop 'MarkDown' features

## Modeling

When dealing with time series data, it is crucial to maintain the chronological order while splitting it. Rather than dividing the data using a ratio that could potentially cause train-test contamination, I opted for a specific time point to include all observations in the same set before that point. The training set consists of 80% of the total weeks in the data.

To start, I tested different imputation methods which included simple linear regression as well as five different tree-based regression algorithms: Decision Tree, Random Forest, XGBoost, CatBoost, and Gradient Boosting. Without any hyperparameter tuning, using default parameters, Random Forest produced the lowest RMSE (Root Mean Square Error) compared to all other models. After testing all possible scenarios, I used the second method - Impute missing values. This involved using interpolation at the store level to fill in missing values for Type A and Type B stores. For Type C stores, I aggregated the values for each MarkDown column and then performed interpolation. This pattern was then used to impute missing values before November 2011.

I started with a Random Forest model as a baseline. The RMSE for the Random Forest model was approximately 3554.79. Figure 10 shows the predicted weekly sales from the Random Forest model. The mean of Weekly_Sales is $15,982.77, and the standard deviation is $22,714.76. Given that the RMSE of the Random Forest model is significantly lower than the standard deviation of Weekly_Sales, we could say that the model's predictions have a reasonable amount of error compared to the natural variability in the target variable; therefore, the model is generally performing well in terms of reducing the overall prediction error. However, even though, on average, the model's predictions are relatively close to the actual sales values, the model does not capture the spikes in sales.

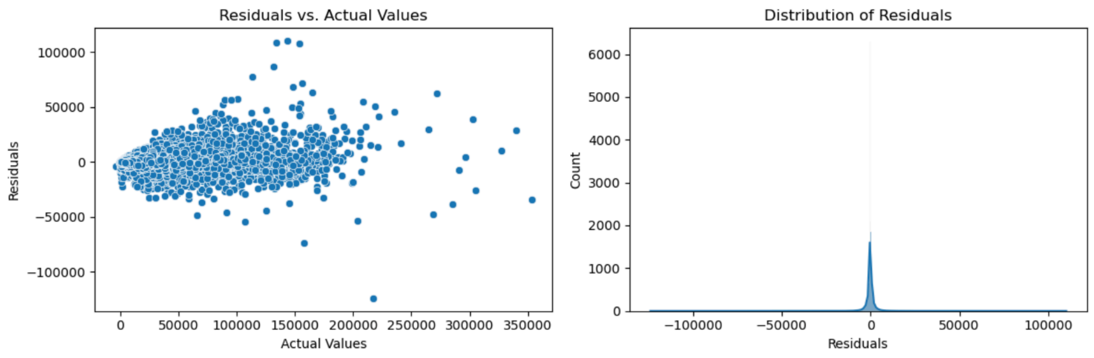


Figure 10. Residual analysis - Random Forest

In Figure 11, the first plot displays a scatter plot of residuals compared to actual values. It is ideal for the residuals to be randomly distributed around 0, indicating that the model's predictions are equally likely to be slightly too high or too low, regardless of the actual value. This suggests that the model's performance may vary depending on the actual sales values. The second plot shows a histogram of residuals with a long tail to the right, which suggests that there are instances where the model significantly underpredicts the sales.

Moreover, there exists a considerable difference between the RMSE of the training set and the test set. This indicates that the model is overfitting to the training data. To improve the model's performance, I experimented with the XGBoost model, which demonstrated the second-highest level of performance during the initial test. Hyperparameters of the XGboost model were tuned to improve the model's performance further. After finding the optimal parameters, the RMSE of the XGboost model improved to 2747.75.

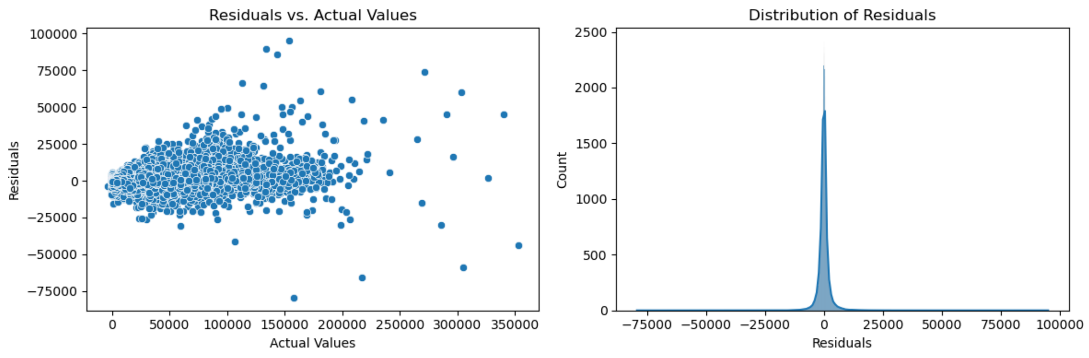


Figure 11. Residual analysis - XGBoost

Figure 12 shows the weekly sales predicted by the XGBoost model compared to the actual sales. The data is divided into five folds, maintaining the temporal order. The predicted sales values are obtained for each fold from the trained XGBoost model.

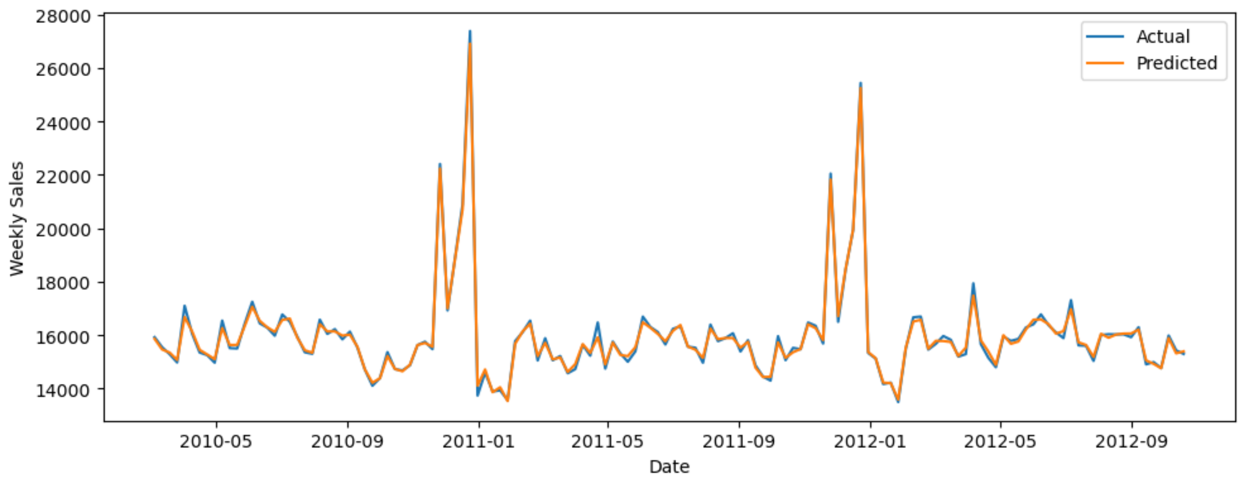


Figure 12. Actual vs Predicted weekly sales

I have noticed that the weekly sales predictions during the peak period are underestimated. Upon further analysis, I have found that certain departments have high residuals, with some exceeding 50,000. Almost half of these high residuals are found in departments 72 and 7. Department 72 has an average weekly sale of $50010.49, which is among the top 4 highest, and its peak sale of $693099.36 is the highest among all departments. Similarly, Department 7 has an average weekly sale of 24387, but its top sale of $206988.62 is the second highest among all departments. Unfortunately, this model cannot accurately capture sudden jumps in sales like these.

## Feature Importance

To explain the contribution of each feature to a specific prediction compared to the average price, I utilize SHAP (Shapley Additive Explanations) analysis. In Figure 13, the feature names are displayed on the Y-axis in descending order of

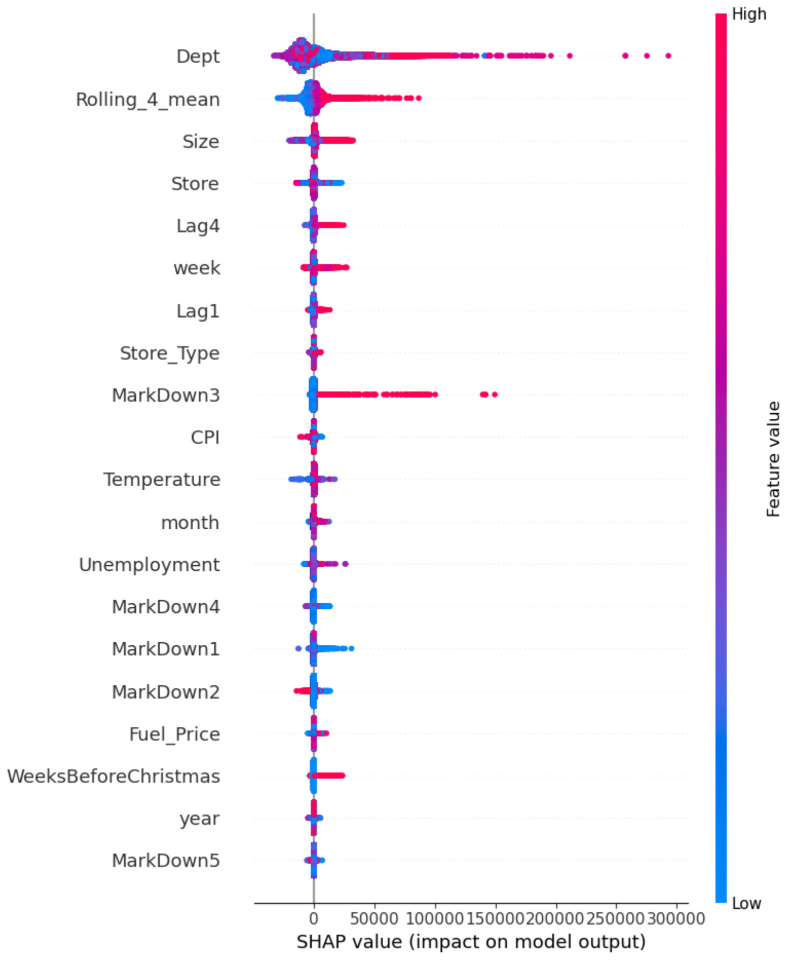


Figure 13. SHAP summary plot

importance, determined by mean absolute SHAP values from top to bottom. Departemnt is the most important feature on average. Some department with high values has large positive SHAP values. This is mainly due to the Department 72.

Rolling_4_mean calculates the rolling average of the data with a window size of 4, and it significantly impacts the model's weekly sales predictions. Using rolling features in time series forecasting is useful because it allows models to analyze historical patterns and dependencies in the data. This helps the models adapt to changing dynamics and improve the accuracy of their predictions by incorporating relevant context and information from the recent past. The SHAP values that are positive and associated with higher rolling averages indicate that when there are high recent sales trends as captured by the rolling average, the model tends to predict higher weekly sales. The importance of the rolling average suggests that recent sales trends are valuable for forecasting.  It might be capturing seasonality or other important patterns for predicting future sales.

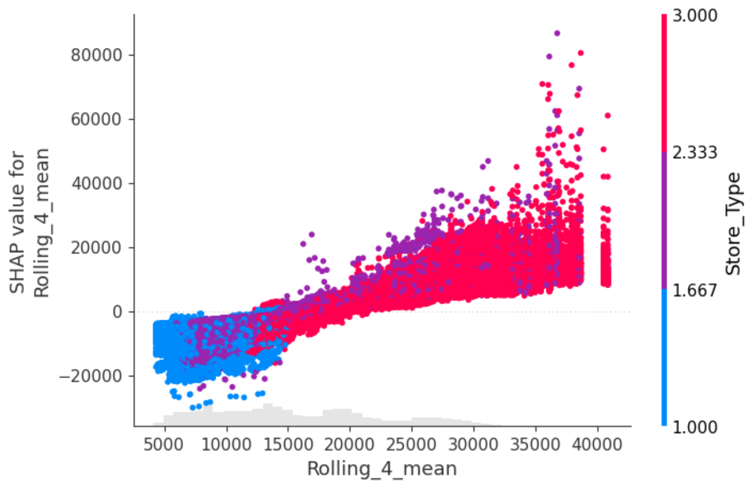


Figure 14. Dependency plot of Rolling_4_mean

Figure 14 shows a dependency plot of SHAP values against the rolling average. The points are colored based on store type. Since the red color represents store type A, this indicates that larger stores tend to have higher rolling averages. This is also consistent with previous EDA findings that larger generally have higher weekly sales. The SHAP analysis supports the insight into store type influence that the larger stores drive the positive correlation between rolling average and weekly sales.

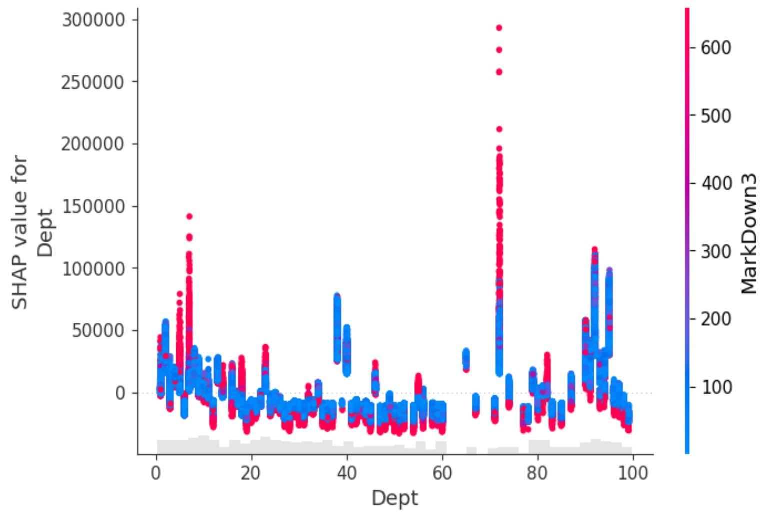


Figure 15. Dependency plot of Dept

I pointed out that large positive SHAP values of some departments with high values were due to Department 72, which also caused the model to underestimate prediction points. Figure 15 visualizes the interaction between Dept and markdown 3.

## Conclusion and Future Work

In this project, I developed an XGBoost model that predicts weekly sales with better performance in terms of RMSE compared to the initial baseline model. There are some suggestions for further improvements to enhance sales prediction accuracy. Firstly, additional features such as holidays and more detailed department categories could be included to capture more nuanced factors that affect sales. Secondly, the XGBoost model could be further optimized by conducting extensive hyperparameter tuning. Third, it has been observed that the XGBoost model's accuracy is limited in predicting certain departments during holiday seasons. Therefore, it would be beneficial to experiment with ensemble methods such as stacking multiple models to achieve even more accurate sales predictions. Lastly, it is recommended to consider time series forecasting techniques to capture seasonality and trend patterns more precisely.