# **Milestone 1: Advanced Data Analysis and Feature Engineering**

## **1. Objective:**

After completing data cleaning and merging in Milestone 1, our next step was to dig deeper into the dataset and make it more useful for prediction. In this phase, we focused on two main goals: first, understanding the behavior of the data over time, and second, creating new features that could help the machine learning model make better predictions. We wanted to capture trends, repetitive patterns, special events, and other signals that might influence how much a product sells on a given day.

We used a mix of time series analysis, basic statistics, and domain knowledge to engineer features we believed would be useful for learning demand behavior. By the end of this milestone, we had built a dataset with many new variables that gave the model much more context than just raw sales numbers.

## **2. Questions and Hypothesis:**

### **2.1 Stationarity Test**

Stationarity implies that the statistical properties of the time series, such as mean and variance, remain constant over time. In this case, the ADF test was conducted on the sales data from the merged\_df dataset. To perform the stationarity test, we will use the Augmented Dickey-Fuller (ADF) test commonly used to check for stationarity in a time series.

Null hypothesis (H0): The sales data is non-stationary.

Alternative hypothesis (H1): The sales data is stationary.

Based on the ADF test, the test statistics (-36.6) is significantly lower than the critical values at all confidence levels (1%, 5%, and 10%). Additionally, the p-value is 0.0, which is lower than the significance level of 0.05.

Since the p-value is less than 0.05, we reject the null hypothesis, indicating that the sales data is stationary. The test results suggest that the 'sales' column exhibits stationarity, which means the data has a constant mean and variance over time. This property is essential for time-series analysis and modeling, as it helps to ensure reliable forecasting and prediction of future sales trends.

### **2.2 Hypothesis Testing**

Null Hypothesis (H0): The promotional activities have a significant impact on store sales for Corporation Favorita.

Alternative Hypothesis (H1): The promotional activities have a significant impact on store sales for Corporation Favorita.

Based on the hypothesis test, we obtained a very low p-value of 0.0. This indicates strong evidence to reject the null hypothesis. Therefore, we can conclude that promotional activities have a significant impact on store sales for Corporation Favorita. The test statistic of 68.22 also suggests a substantial difference in sales between promotional and non-promotional periods. These results support the notion that promotional activities play a crucial role in driving store sales.

### **2.3 Answering Questions:**

### 1. Which dates have the lowest and highest sales for each year?

Dates with the lowest sales for each year:

2013 2013-01-01

2014 2014-01-01

2015 2015-01-01

2016 2016-02-08

2017 2017-01-02

Dates with the highest sales for each year:

2013 2013-12-31

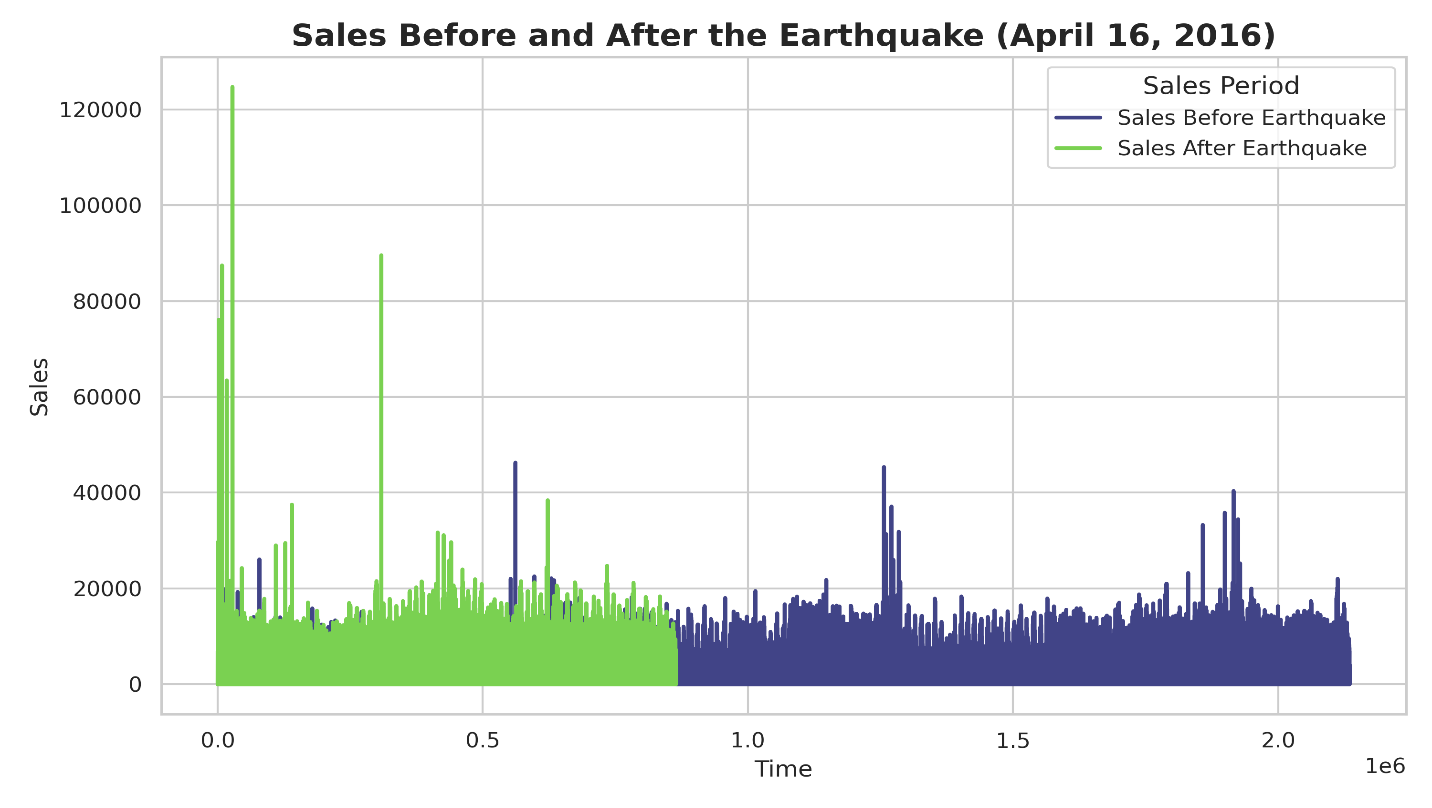
2014 2014-12-31

2015 2015-12-31

2016 2016-12-26

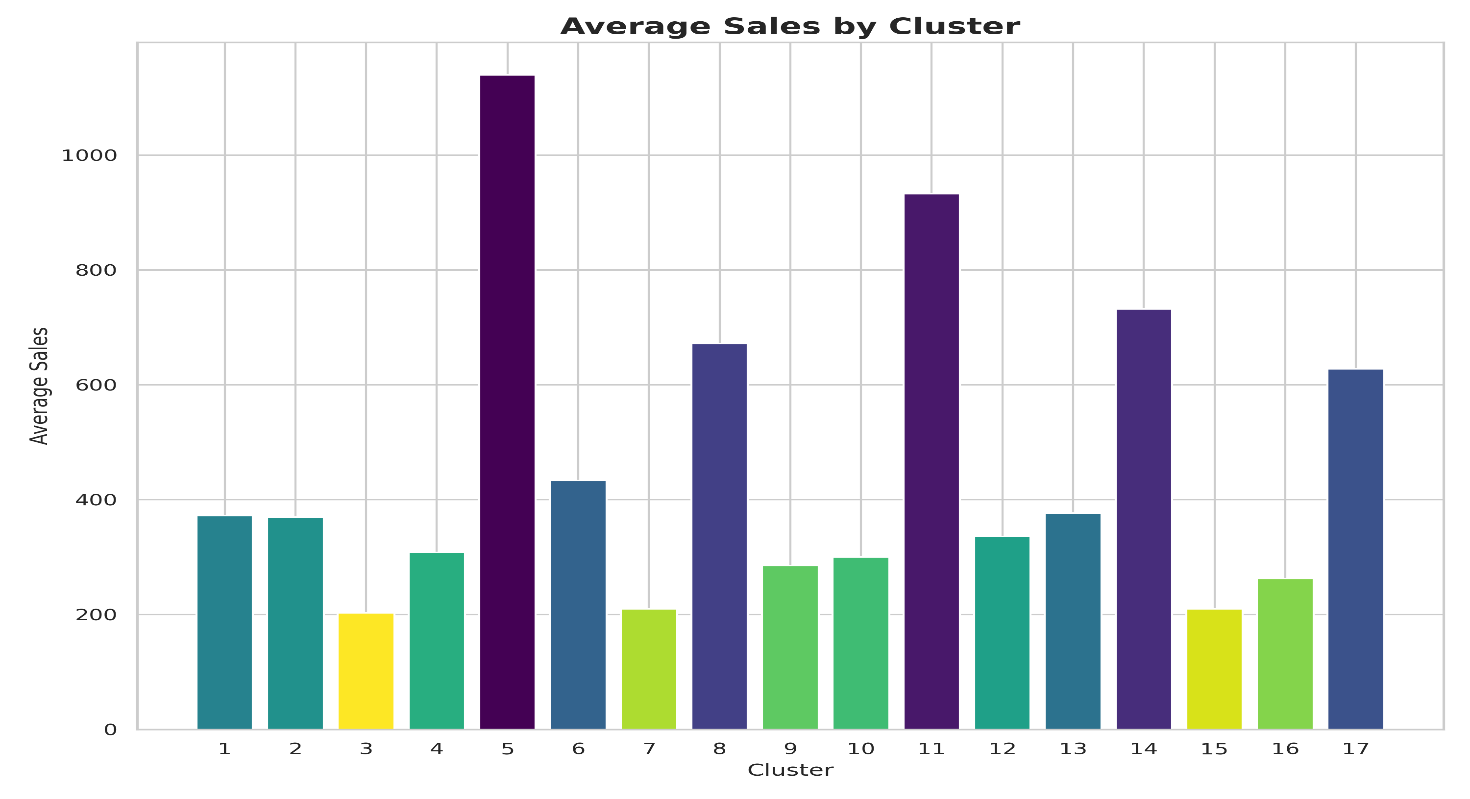
2017 2017-08-15

### 2. Analyze the impact of the earthquake on sales



**There was a surge in sales after the earthquake.**

### 3. Determine if certain groups of stores sell more products



The cluster with the highest number of stores is Cluster 5, followed by Clusters 14, 8, 11 and 12. These clusters have a significantly larger number of stores compared to the others.

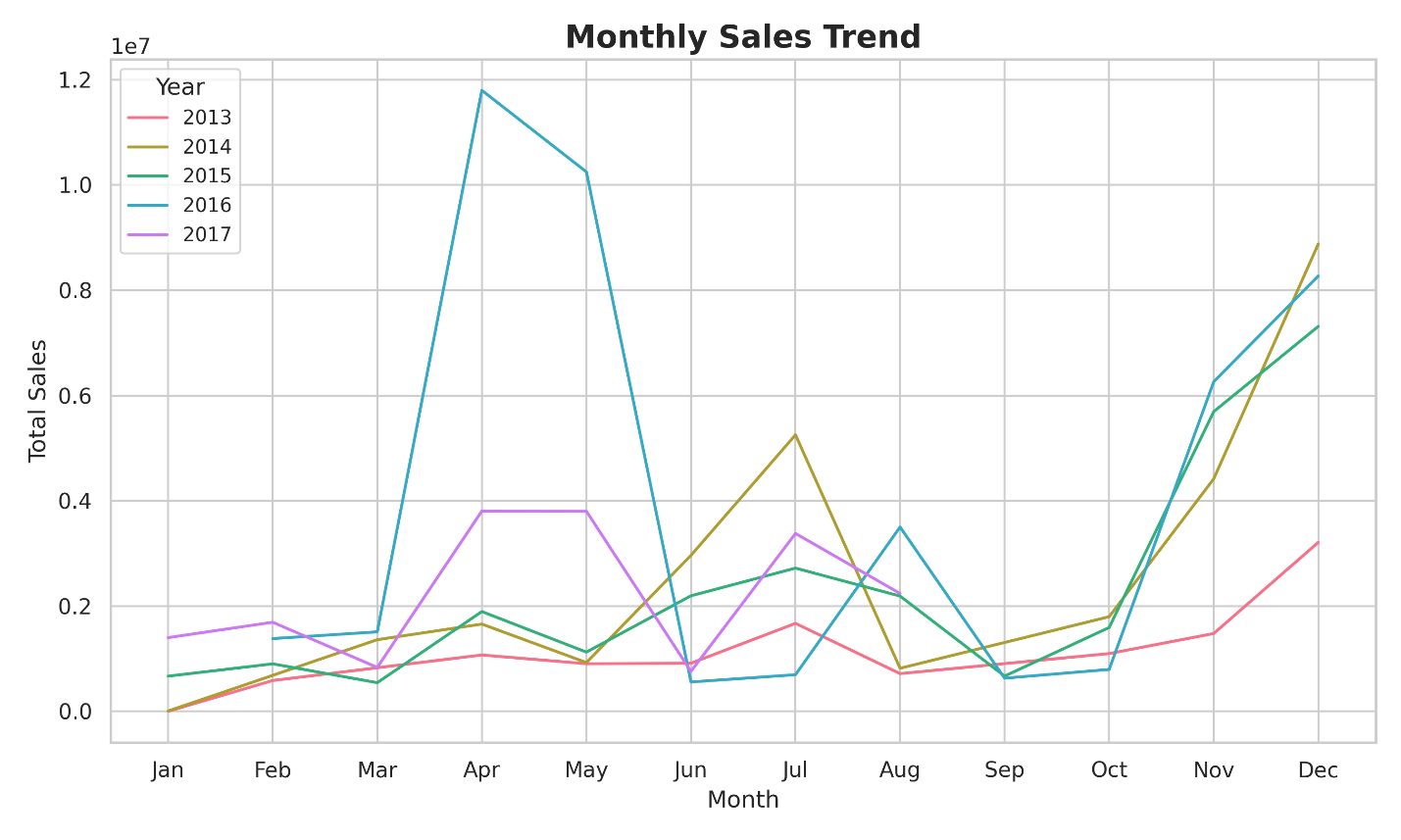
### 4. Are sales affected by promotions, oil prices and holidays?

Correlation between Sales and Promotions: 0.41802891996712443

Correlation between Sales and Oil Prices: -0.06150915236754859

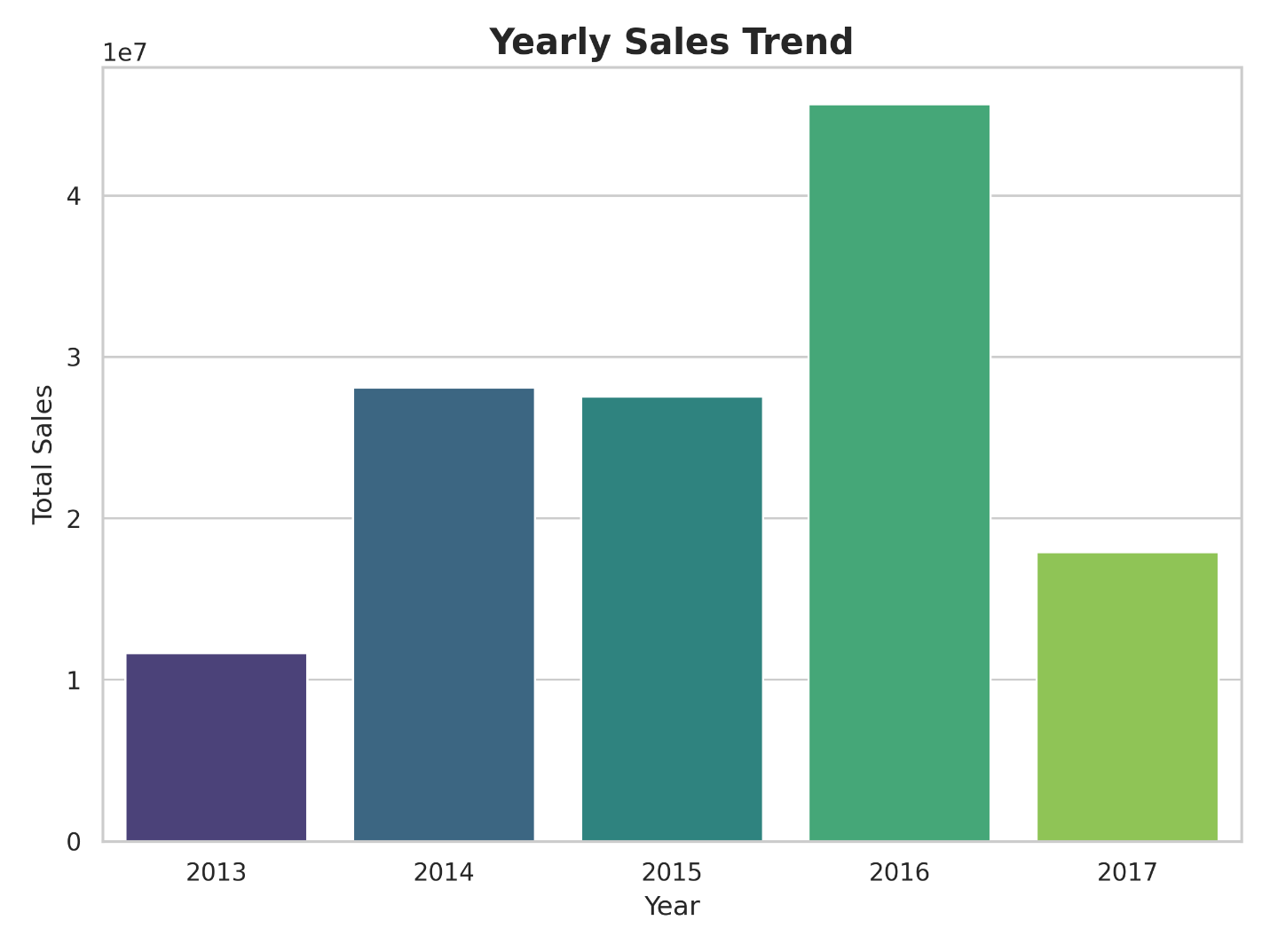
Correlation between Sales and Holidays: -0.037068929312132404

5. What analysis can we get from the date and its extractable features?

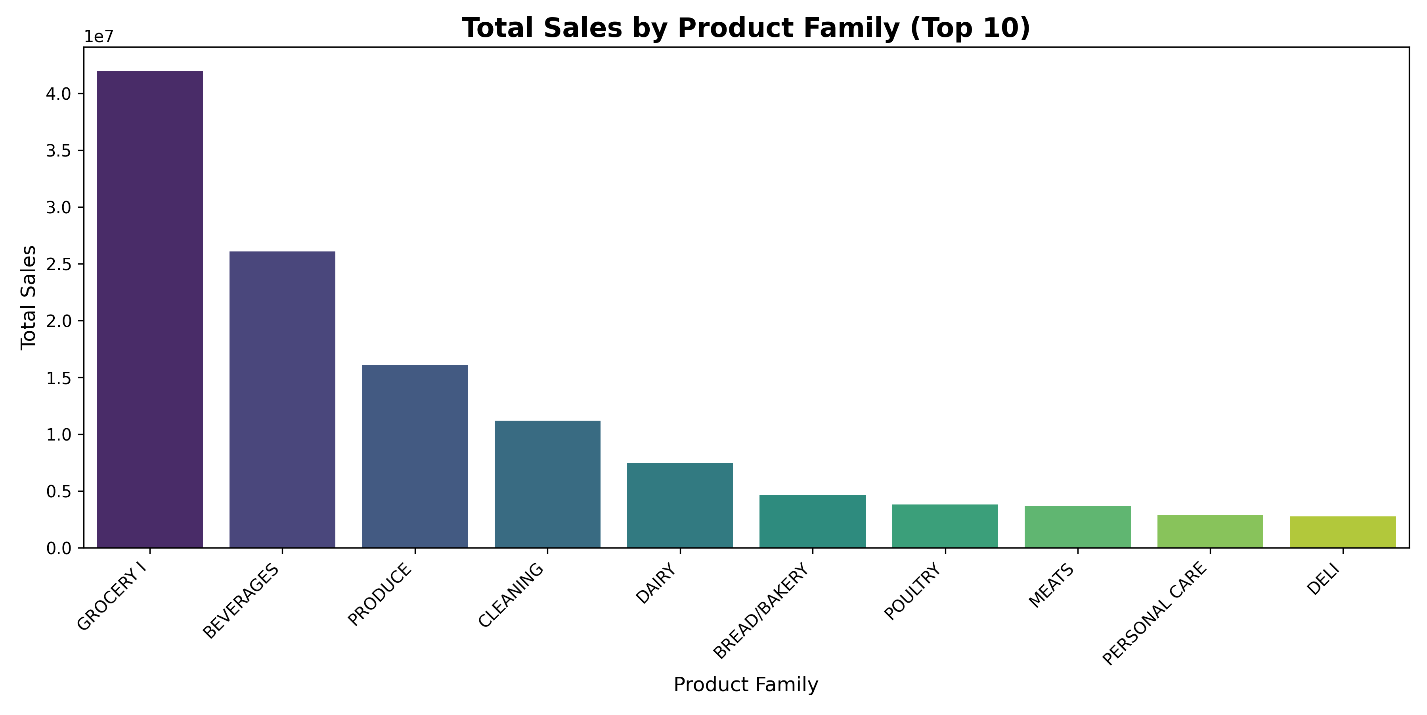


The line plot depicting the monthly sales trend shows variations in sales throughout the year. Sales seem to dip during the middle months of the year (July to September) before rising again in the last quarter (October to December), with the highest sales in December. The year-wise color distinction helps to observe sales patterns for each year.

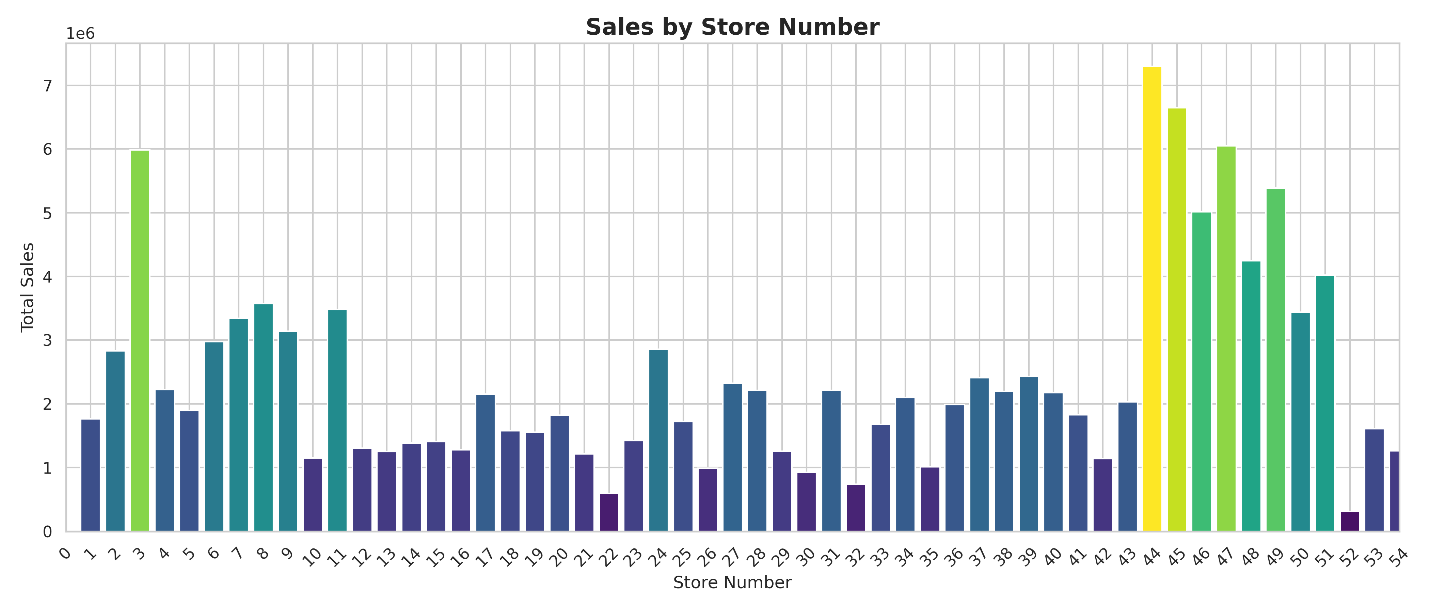
### 



### 6. Are there certain product families types that exhibit higher sales performance?



### 7. How does the sales trend vary across different store numbers?



## **3. Feature Engineering:**

Once we understood the nature of the data, we focused on building new features that would give the model better visibility into trends and cycles. Here's a breakdown of what we added:

### **3.1 Time-Based Feature Engineering:**

These help the model to capture temporal patterns in data and improve overall model performance.

These were all extracted from date column :

**day**: Day of the month (1–31)

**month**: Month of the year (1–12)

**week**: week number

**year**: Four-digit year

**day\_of\_week**: Day of the week (0 = Monday, 6 = Sunday)

**is\_weekend**: Binary indicator for weekends (1 if Saturday or Sunday, else 0)

**quarter**: Calendar quarter of the year (1 to 4)

**season**: There is a number given for each season

* 0 for late winter or early spring (February–March),
* 1 for spring (April–June),
* 2 for summer (July–August),
* 3 for fall (September–November),
* 4 for winter (December–January)

**Days\_to\_Thanksgiving**: Number of days remaining until November 24 in the same year.

**Days\_to\_Christmas**: Number of days remaining until December 24 in the same year.

### **3.2 External Economic Indicators:**

These account for macro-level factors that influence demand.

**oil\_price**: Proxy for economic health/inflation, which affects consumer spending.

**holiday\_type:**  Holiday, Event, Additional, Transfer, Bridge, and Work Day (if it isn’t a holiday)

**is\_holiday**:

The column is a binary feature indicating whether a specific day is a holiday. A value of 1 means it is a holiday, while 0 means it is a regular day. In the dataset, there are **323,829 holiday records** and **26,730 non-holiday records**. This feature is important for capturing the impact of holidays on trends such as customer behavior, demand, or sales.

### **3.3 Internal Business Factors:**

Directly influenced by the retailer’s operations and strategy.

**onpromotion**: Indicates whether an item is under promotion, often driving short-term sales increases.

**96,020 records** have items that were on promotion

**254,539 records** have items that were not on promotion

### **3.4 Lag Features:**

To give the model access to historical context, we created lag variables:

* lag\_1: Yesterday’s sales
* lag\_7: Sales a week ago
* lag\_14: Sales one month ago
* lag\_promo\_1: Whether the item was on promotion the previous day

### 

### **3.5 Rolling Averages and Trends:**

We wanted to capture short- and medium-term trends by using rolling statistics:

* 7-day and 14-day **rolling median**
* **rolling\_promo\_7**: 7-day rolling median of the onpromotion feature, grouped by store

This helped smooth out noisy data and showed whether a product was currently in a growth or decline trend.

### **3.6 Target Transformation:**

To stabilize variance and prevent the model from predicting negative sales values, we applied a log transformation:

* **log\_sales**: Natural log of sales incremented by 1 (i.e., log(sales + 1))

This transformation helps normalize the distribution of the target variable, which often improves model performance and convergence.

## **4. Dropping unnecessary columns as it is not needed for our analysis:**

To eliminate irrelevant or redundant information, we dropped the following columns that were not needed for analysis or modeling

**date, id**: Already used for feature extraction; original columns no longer needed

**locale, locale\_name, description, transferred, state**: Metadata or duplicate information already captured in other features

**base\_price, daily\_variation, transactions**: Either not useful for our predictive goals or potentially redundant with engineered features

## **5. Encoding Categorical Variables:**

To prepare categorical variables for machine learning models that require numerical input, we applied **Binary Encoding** to selected columns. Binary encoding is efficient for high-cardinality categories, as it reduces dimensionality while preserving information.

#### Columns Encoded:

* family
* city
* holiday\_type
* store\_type

## **6. Conclusion:**

This milestone turned a basic historical sales dataset into a much richer, feature-packed version that gives the forecasting model more context. We engineered a wide range of variables, from lagged sales to smoothed oil prices, and tested their usefulness through visualization and early model feedback.

By the end of this phase, we were confident that the dataset was well-prepared for machine learning, especially for models like Random Forest and XGBoost that benefit from strong, structured inputs. These features laid the groundwork for the training and evaluation work that followed in Milestone 3.