WALMART DATA ANALYSIS

Submitted to: Jose Alvarez, BScEE, MBA

April 2024

YOGESH MUPPIRI (YXM220000)

PROBLEM STATEMENT

Context of the issue

For large corporations, expanding their business is crucial, and they continuously seek new opportunities for growth. While our focus is on Walmart for this project, the methods discussed can be applied to stores across various industries. The data cleaning and normalization procedures may need adjustments depending on the specific industry of the selected company.

In this project, we aim to analyze the data collected by Walmart from 45 of its stores over certain time intervals. Our primary goal is to identify the key factors contributing to their sales. We'll investigate whether holidays, fluctuations in fuel prices, or changes in temperature have a significant impact on sales. Based on our findings, Walmart can explore alternative strategies to enhance their sales performance.

What makes this important, and why is it essential? And Contribution

Typically, the number of employees in each Walmart store averages between 300 to 400 individuals. However, this figure can fluctuate based on factors such as the store's size and location. Larger stores may employ over 500 workers, whereas smaller ones may have fewer than 200 employees. This is just one of the several reasons why these stores are vital contributors to the community. Despite any reservations we may have, their presence generates significant tax revenue. All the funds for operating these establishments stem from their sales. If their sales decline, it could lead to widespread unemployment and a cascade of other consequences. Therefore, it's crucial to ensure the stability of these companies, which serve as pillars of our communities. To support them in this endeavour, it's imperative to assist them in analysing strategies to boost their sales.

As previously mentioned, our approach involves refining the data provided by Walmart and utilizing machine learning models to extract insights. Our objective is to identify the factors influencing store sales. For instance, we will examine whether sales increase during holiday periods such as Labor Day or Christmas. We'll also investigate if rising fuel prices deter customers from visiting the store and whether cold temperatures during snowfall impact store traffic. Additionally, we'll explore the relationship between unemployment rates and store sales.

The hypothesis that we are finding in this project:
- Does the rise in fuel prices discourage customers from visiting the store?
- Do the weather changes prevent the customer from visiting the store?
- Does the store sales increases during holidays sessions? (ex. Black Friday, labour day, snowfall, and Christmas etc)
- How does the level of unemployment affect store sales?

Source of evidence

Here We're utilizing Walmart's data collected from 45 distinct stores on a weekly basis. Initially, we have three types of data: stores, features, and training data. Each dataset provides unique aspects of the data. The stores dataset offers store-specific information, while the features dataset contains factors under analysis, with some feature names anonymized for security. Lastly, the training data comprises historical training data.

Feature dataset has 8190 records and 12 fields.

```
# first 5 rows of dataset(features)
features.head()
```

	Store	Date	Temperature	Fuel_Price	MarkDown1	MarkDown2	MarkDown3	MarkDown4	MarkDown5	CPI	Unemployment	IsHoliday
0	1	2010-02-05	42.31	2.572	NaN	NaN	NaN	NaN	NaN	211.096358	8.106	False
1	1	2010-02-12	38.51	2.548	NaN	NaN	NaN	NaN	NaN	211.242170	8.106	True
2	1	2010-02-19	39.93	2.514	NaN	NaN	NaN	NaN	NaN	211.289143	8.106	False
3	1	2010-02-26	46.63	2.561	NaN	NaN	NaN	NaN	NaN	211.319643	8.106	False
4	1	2010-03-05	46.50	2.625	NaN	NaN	NaN	NaN	NaN	211.350143	8.106	False

Figure 1: Features Dataset Store

dataset has 45 records and 3 fields.

```
# first 5 rows of dataset(store)
store.head()
```

	Store	Type	Size
0	1	A	151315
1	2	A	202307
2	3	B	37392
3	4	A	205863
4	5	B	34875

Figure 2: Store dataset

Train dataset has 421570 records and 5 fields.

```
# first 5 rows of dataset(train)
train.head()
```

	Store	Dept	Date	Weekly_Sales	IsHoliday
0	1	1	2010-02-05	24924.50	False
1	1	1	2010-02-12	46039.49	True
2	1	1	2010-02-19	41595.55	False
3	1	1	2010-02-26	19403.54	False
4	1	1	2010-03-05	21827.90	False

Figure 3: train dataset

Exploratory data analysis And Data Cleaning

The dataset might include null values, or certain columns could be empty. It's essential to understand the various types of data present and their respective data types. To accomplish this, we utilized the info () method, which provides information such as the total number of entries, column names, and their corresponding data types. The screenshots below illustrate the nature of the data under analysis.

```
# Dataset info (features)
features.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8190 entries, 0 to 8189
Data columns (total 12 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Store          8190 non-null   int64
 1   Date           8190 non-null   object
 2   Temperature    8190 non-null   float64
 3   Fuel_Price     8190 non-null   float64
 4   MarkDown1      4032 non-null   float64
 5   MarkDown2      2921 non-null   float64
 6   MarkDown3      3613 non-null   float64
 7   MarkDown4      3464 non-null   float64
 8   MarkDown5      4050 non-null   float64
 9   CPI            7605 non-null   float64
 10  Unemployment   7605 non-null   float64
 11  IsHoliday      8190 non-null   bool
dtypes: bool(1), float64(9), int64(1), object(1)
memory usage: 712.0+ KB
```

Figure 4: Features dataset information (info**()**)

The upper figure shows some information about the dataset that we have. This dataset has 8190 entries and 12 columns (fields or attributes). Additionally, it indicates the number of null values present in each attribute, alongside the data type associated with each column.

```
# Dataset info (store)
store.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 45 entries, 0 to 44
Data columns (total 3 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   Store   45 non-null     int64
 1   Type    45 non-null     object
 2   Size    45 non-null     int64
dtypes: int64(2), object(1)
memory usage: 1.2+ KB
```

Figure 5: store dataset info

```
# Dataset info (train)
train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 421570 entries, 0 to 421569
Data columns (total 5 columns):
 #   Column        Non-Null Count   Dtype
---  ------        --------------   -----
 0   Store         421570 non-null  int64
 1   Dept          421570 non-null  int64
 2   Date          421570 non-null  object
 3   Weekly_Sales  421570 non-null  float64
 4   IsHoliday     421570 non-null  bool
dtypes: bool(1), float64(1), int64(2), object(1)
memory usage: 13.3+ MB
```

Figure 6: train dataset info

The figure 6 shows the information about train dataset that having 421570 records and 5 columns (attributes or fields). There are no null values present in any of the attributes. Two attributes are of integer datatype, while the others consist of an object, a float, and a Boolean datatype each.

The figure 5 demonstrates the information about store dataset where it has 45 records and 3 columns (fields or attributes). None of the attributes contain null values. Among the three attributes, two are of integer data type, while the remaining one is of object data type.

Additionally, we utilize the **describe ()** function, which provides an overview of the dataset's statistics, including measures such as mean, median, count, maximum, etc.

```
features.describe()
```

	Store	Temperature	Fuel_Price	MarkDown1	MarkDown2	MarkDown3	MarkDown4	MarkDown5	CPI	Unemployment
count	8190.000000	8190.000000	8190.000000	4032.000000	2921.000000	3613.000000	3464.000000	4050.000000	7605.000000	7605.000000
mean	23.000000	59.356198	3.405992	7032.371786	3384.176594	1760.100180	3292.935886	4132.216422	172.460809	7.826821
std	12.987966	18.678607	0.431337	9262.747448	8793.583016	11276.462208	6792.329861	13086.690278	39.738346	1.877259
min	1.000000	-7.290000	2.472000	-2781.450000	-265.760000	-179.260000	0.220000	-185.170000	126.064000	3.684000
25%	12.000000	45.902500	3.041000	1577.532500	68.880000	6.600000	304.687500	1440.827500	132.364839	6.634000
50%	23.000000	60.710000	3.513000	4743.580000	364.570000	36.260000	1176.425000	2727.135000	182.764003	7.806000
75%	34.000000	73.880000	3.743000	8923.310000	2153.350000	163.150000	3310.007500	4832.555000	213.932412	8.567000
max	45.000000	101.950000	4.468000	103184.980000	104519.540000	149483.310000	67474.850000	771448.100000	228.976456	14.313000

Figure 7: statistical overview of features dataset

Similarly, we perform describe () function for other datasets to find overall statistics of stores.

We will determine the number of null values in each dataset to facilitate their replacement. For this purpose, we will use null (). sum() function, which provides the total count of null values, contrary to the info() method.

```
features.isnull().sum()
```

```
Store             0
Date              0
Temperature       0
Fuel_Price        0
MarkDown1      4158
MarkDown2      5269
MarkDown3      4577
MarkDown4      4726
MarkDown5      4140
CPI             585
Unemployment    585
IsHoliday         0
dtype: int64
```

Figure 8: null values

```
features.isnull().sum() #cleared null values
```

```
Store           0
Date            0
Temperature     0
Fuel_Price      0
MarkDown1       0
MarkDown2       0
MarkDown3       0
MarkDown4       0
MarkDown5       0
CPI             0
Unemployment    0
IsHoliday       0
dtype: int64
```

Figure 9: null values are replaced with 0.

It is clear by seeing above figure, The features dataset has null values and similarly when we perform an isnull function for other datasets we get zero null values. so, the null values that we have observed in features dataset are going to replace with zeros, for that we use fillna() function. i.e., features = features.fillna(0)

figure 9 describes the number of null values that we have in features dataset after replacing with zero. Where there are no null values.

Now, we proceed with merging the store and features datasets. This step is necessary because the store dataset exclusively includes information such as the size and type of the store, which is absent in the features dataset. Therefore, we employ an inner join to merge these two datasets into a unified entity. This approach enables us to incorporate these additional features into our analysis without repeating. i.e, features.merge(store, how = 'inner', on = 'Store')

	Store	Date	Temperature	Fuel_Price	MarkDown1	MarkDown2	MarkDown3	MarkDown4	MarkDown5	CPI	Unemployment	IsHoliday	Type	Size
0	1	2010-02-05	42.31	2.572	0.0	0.0	0.0	0.0	0.0	211.096358	8.106	False	A	151315
1	1	2010-02-12	38.51	2.548	0.0	0.0	0.0	0.0	0.0	211.242170	8.106	True	A	151315
2	1	2010-02-19	39.93	2.514	0.0	0.0	0.0	0.0	0.0	211.289143	8.106	False	A	151315
3	1	2010-02-26	46.63	2.561	0.0	0.0	0.0	0.0	0.0	211.319643	8.106	False	A	151315
4	1	2010-03-05	46.50	2.625	0.0	0.0	0.0	0.0	0.0	211.350143	8.106	False	A	151315

Figure 10: Top 5 rows (head) of newly merged data frame

From figure 10, it appears that the 'date' attribute is currently of object datatype. To fully utilize this attribute, we need to convert it into a date datatype. Therefore, we will proceed to convert the 'date' attribute into a date datatype. Additionally, we will create two new columns, 'week' and 'year', to facilitate our analysis. To accomplish this, we will use the. isocalendar() function.

For a clearer understanding of this newly merged table after merging, new datatypes and two new columns.

```
# for better understanding
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 421570 entries, 0 to 421569
Data columns (total 18 columns):
 #   Column         Non-Null Count   Dtype
---  ------         --------------   -----
 0   Store          421570 non-null  int64
 1   Dept           421570 non-null  int64
 2   Date           421570 non-null  datetime64[ns]
 3   Weekly_Sales   421570 non-null  float64
 4   IsHoliday      421570 non-null  bool
 5   Temperature    421570 non-null  float64
 6   Fuel_Price     421570 non-null  float64
 7   MarkDown1      421570 non-null  float64
 8   MarkDown2      421570 non-null  float64
 9   MarkDown3      421570 non-null  float64
 10  MarkDown4      421570 non-null  float64
 11  MarkDown5      421570 non-null  float64
 12  CPI            421570 non-null  float64
 13  Unemployment   421570 non-null  float64
 14  Type           421570 non-null  object
 15  Size           421570 non-null  int64
 16  week           421570 non-null  UInt32
 17  year           421570 non-null  UInt32
dtypes: UInt32(2), bool(1), datetime64[ns](1), float64(10), int64(3), object(1)
memory usage: 52.7+ MB
```

Figure 11: new dataset information **EXPLORATORY**

DATA ANALYSIS (EDA)

In this phase, we will create cross-plots of each attribute against every other plottable attribute to examine the interrelations between them. We are going to utilize histograms for this purpose, as they provide a straightforward visualization method and easy to understand.

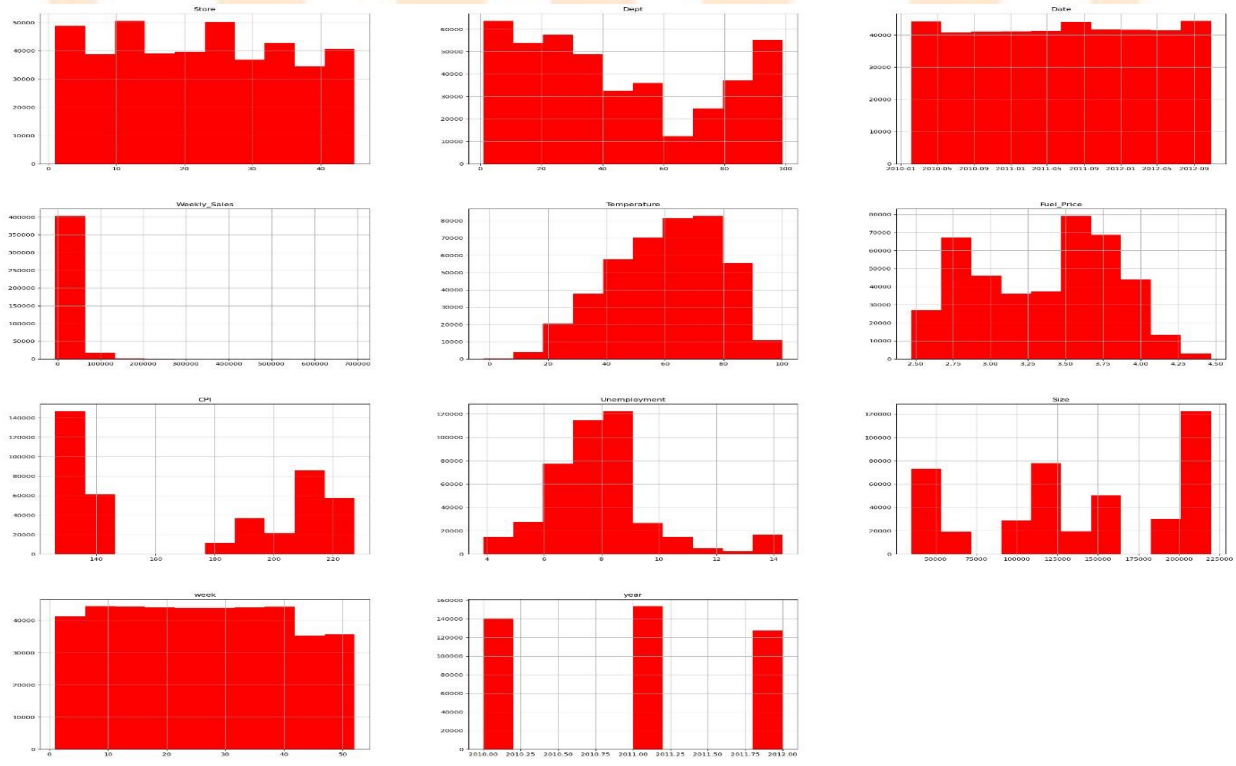


Figure 12: plotting different attributes at a time for better analysis.

Examining the distribution of weekly sales across different stores. To accomplish this, we have employed a histogram plot.

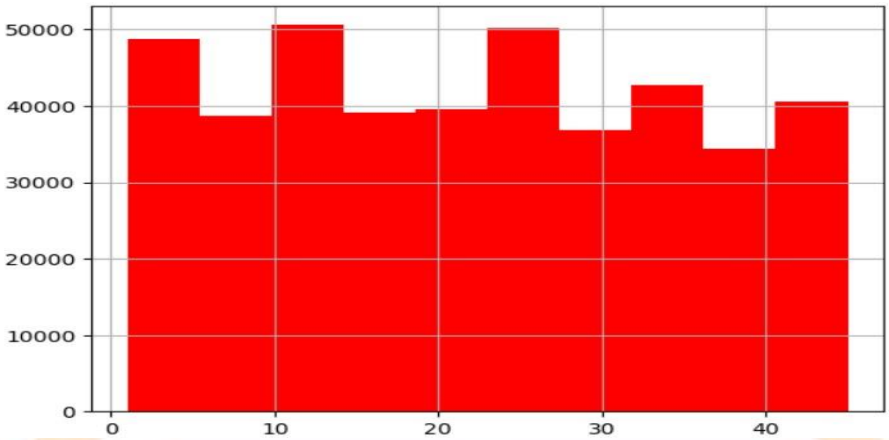


Figure 13: graph shows 45 different store sales.

The weekly sales are in uniform except few stores which may be in populated areas, but on average most of the store's sales are uniform.

Overall sales per year, now we are trying to plot histogram for store sales per year (2010, 2011 and 2012)

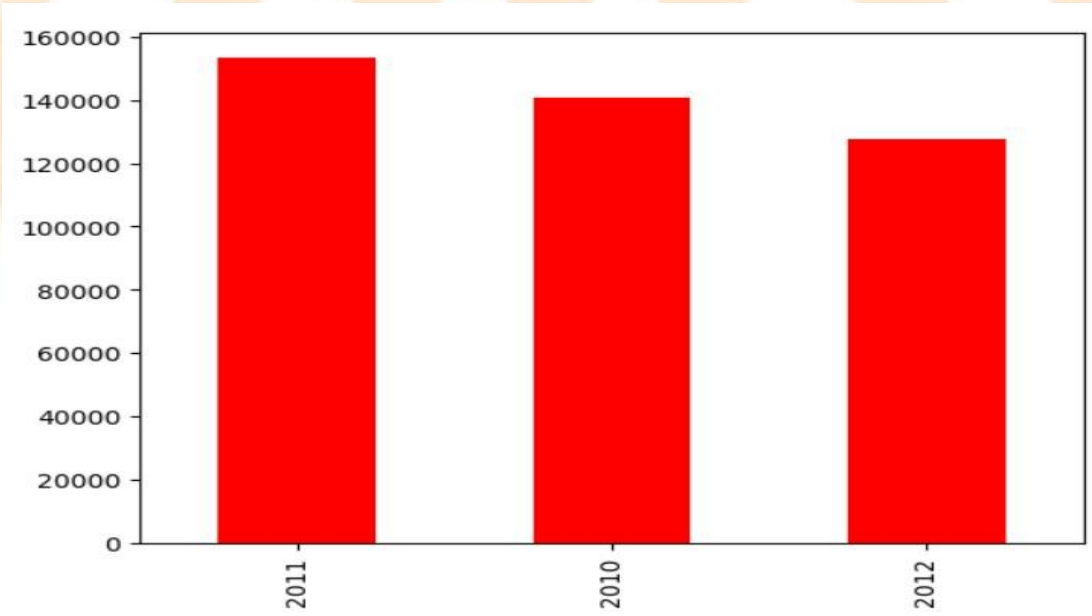


Figure 14: overall sales per year

Figure 14, From the data which we have, the sales for 2011 are the highest among the three years, suggesting that this year had strong performance in terms of revenue generation. Factors contributing to this peak could be seasonal trends, new product releases, or marketing campaigns that were particularly successful. There may be several potential factors that may bring sales down in 2012 like product offerings, pricing strategy, economic condition and etc.

weakly sales over different store, there's a clear variation in sales among the different stores, with some having consistently higher sales and others lower sales. Some stores have consistently high weekly sales, indicating strong customer traffic, good location, or effective marketing. Others with lower sales might require improvement in store management, product range, or location-based strategies. Stores with consistently lower weekly sales could be analysed further to identify root causes, whether it's location, inventory or customer demographics.

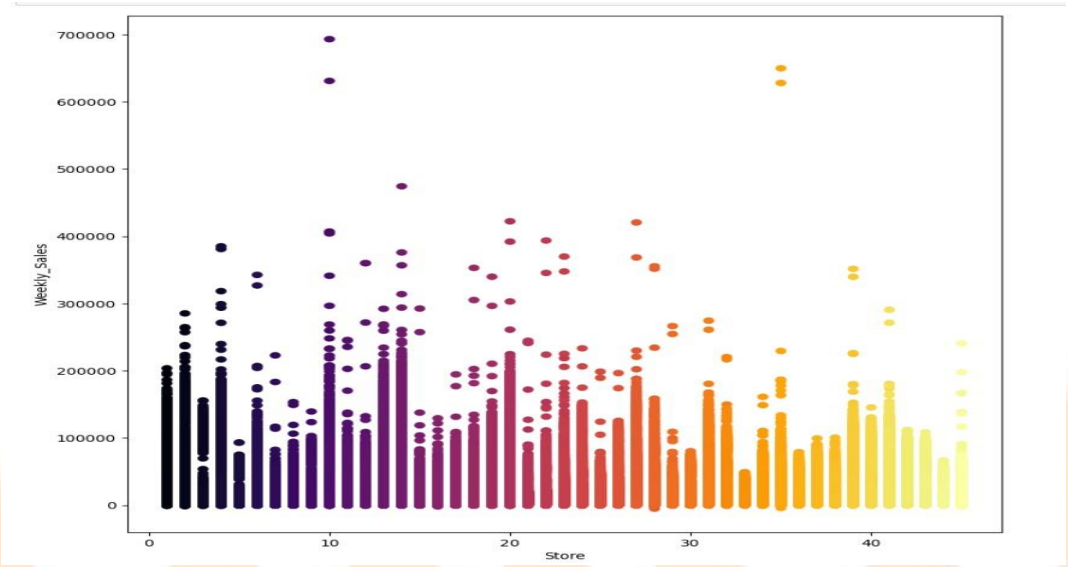


Figure 15: weakly sales over different store

Relationship between weekly sale and each department, this graph appears to be a scatter plot with weekly sales on the y-axis and department numbers on the x-axis. we can say that Some departments show higher sales spikes, indicating high-selling items, special promotions and in addition Some departments consistently show higher sales, suggesting they are popular or have high-demand products. There are departments with considerable fluctuations, indicating variable weekly sales, possibly due to seasonality or promotional events. The pattern and distribution of sales across departments can indicate which ones consistently perform well and which might need strategic adjustments.

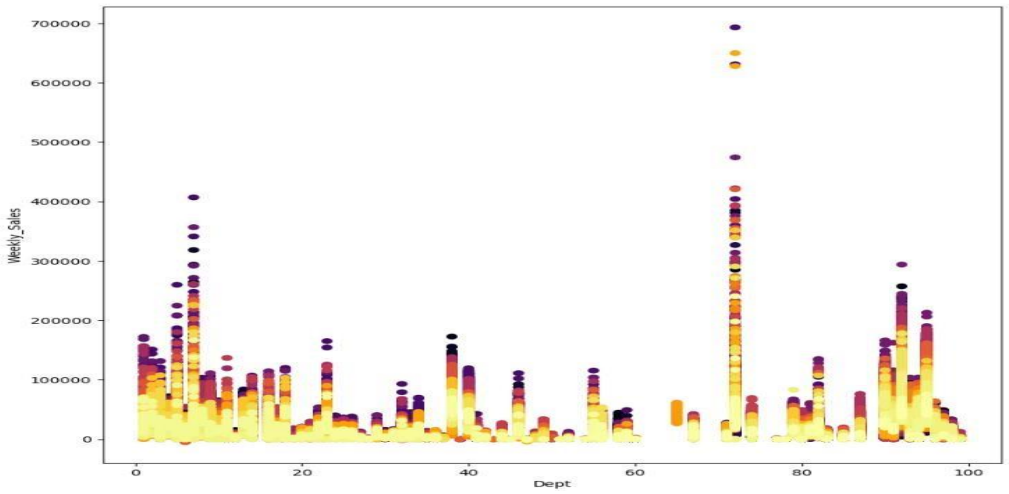
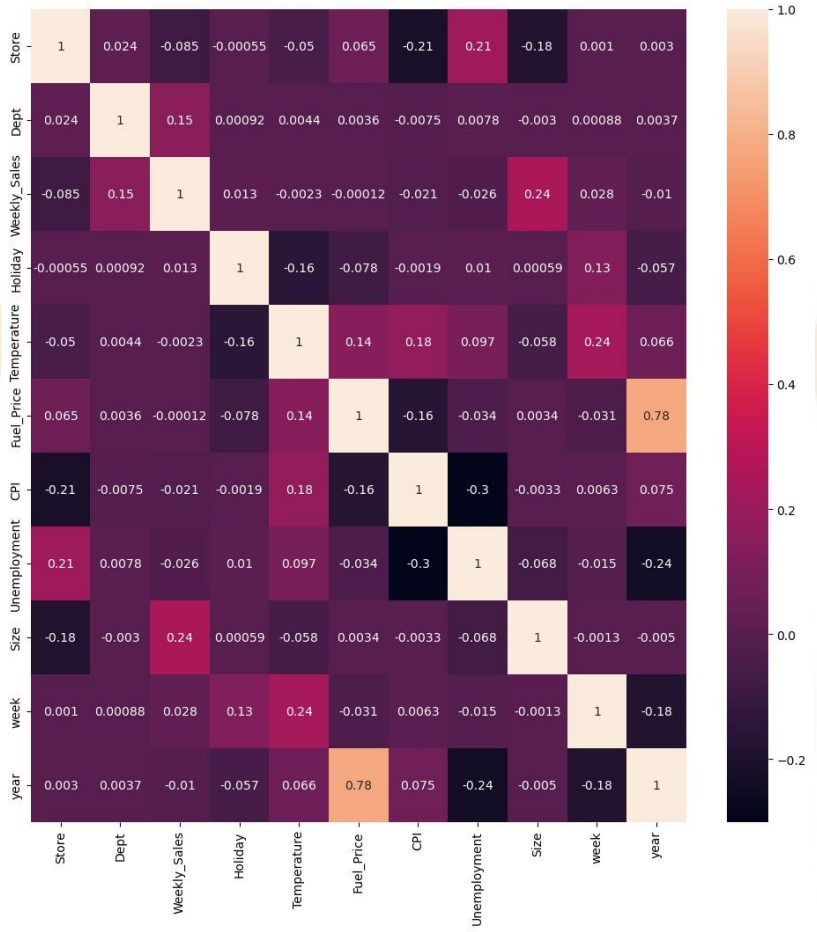


Figure 16: Relationship between weekly sale and each department

Correlation between each of the attributes, instead of examining the relationship between two columns at a time, a heatmap allows us to visualize the correlations between all the attributes in the dataset in a single plot. This comprehensive view helps identify which attributes have strong or weak correlations with each other.



The heatmap visualizes the correlation matrix, showing the relationships between various attributes in a dataset. The heatmap uses colours to represent the strength and direction of correlations between attributes. Darker shades typically indicate stronger correlations, while lighter shades represent weaker correlations. The diagonal elements of the heatmap (top-left to bottom-right) represent correlations of attributes with themselves, which is always 1. This acts as a reference point for other correlations.

Findings from heatmaps:

Fuel Price and Year: The heatmap indicates a strong positive correlation between fuel price and year, suggesting a trend of increasing fuel prices over time.

Temperature and Weekly Sales: The correlation between temperature and weekly sales is relatively weak, indicating that temperature might not significantly influence weekly sales.

Size and Weekly Sales: The positive correlation between size and weekly sales suggests that larger stores tend to have higher sales. This can guide strategies for store expansion or inventory management.

Several attributes have weak or negligible correlations, suggesting that they may not significantly influence each other. For example, the correlation between holiday and other attributes is generally weak, indicating that holidays may not strongly affect other metrics.

Weekly sales during holiday weeks and non-holiday weeks, the pie chart shows that holiday weeks account for 51.7% of total sales, while non-holiday weeks represent 48.3%. This indicates that holiday weeks have a slightly higher impact on sales compared to nonholiday weeks where holidays drive more customer spending. Retailers might need to plan for increased demand, larger inventories, and additional staff during holiday weeks to capitalize on the sales boost. Although the proportion of sales is lower during non-holiday weeks, they still contribute a significant share of revenue (48.3%). Retailers should maintain a consistent sales strategy during non-holiday periods to ensure a stable revenue stream.

Retailers could explore new approaches to increase sales during traditionally slower periods. A balanced strategy, focusing on both holiday and non-holiday weeks, can help retailers maximize their sales potential.

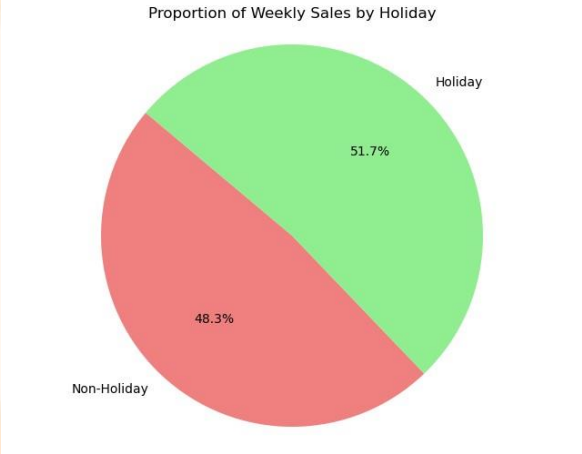
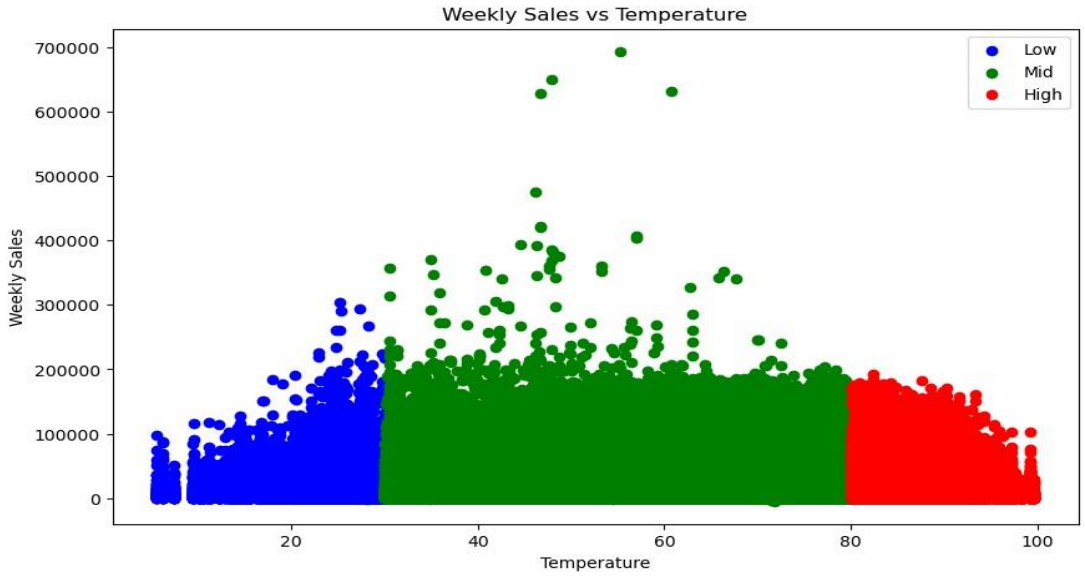


Figure 17: The pie chart illustrates the proportion of weekly sales during holiday weeks and non-holiday weeks.

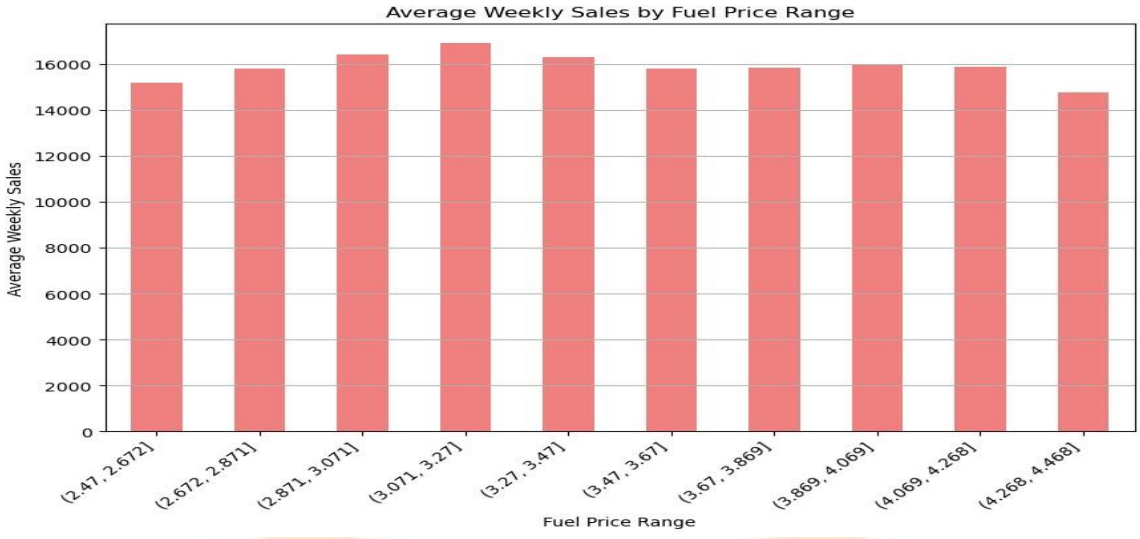
Weekly sales vs Temperature, the scatter plot uses colour to indicate different temperature ranges: blue for low, green for mid, and red for high temperatures.

In general, sales appear to increase with temperature, peaking around the mid-temperature range. Sales tend to decrease as temperature reaches higher levels, suggesting that there might be an optimal temperature range for high sales. The highest sales are seen in the green zone, corresponding to mid-range temperatures (around 40 to 60 degrees). This could indicate that moderate weather conditions encourage more customer activity and higher sales.

In the lower temperature range (blue), sales are relatively lower, suggesting that colder weather may deter shopping activity. Similarly, in the high-temperature range (red), sales tend to decrease, possibly due to extreme heat impacting shopping behaviour.



Weekly sales with fuel price, The weekly sales show a relatively stable pattern across the different fuel price ranges. There might be slight fluctuations, but overall, the average weekly sales seem constant regardless of fuel prices. This stability suggests that fuel price fluctuations don't significantly impact weekly sales, indicating that consumer purchasing behaviour remains consistent. If specific fuel price ranges show noticeable peaks or dips, it could point to isolated events or external factors influencing sales.

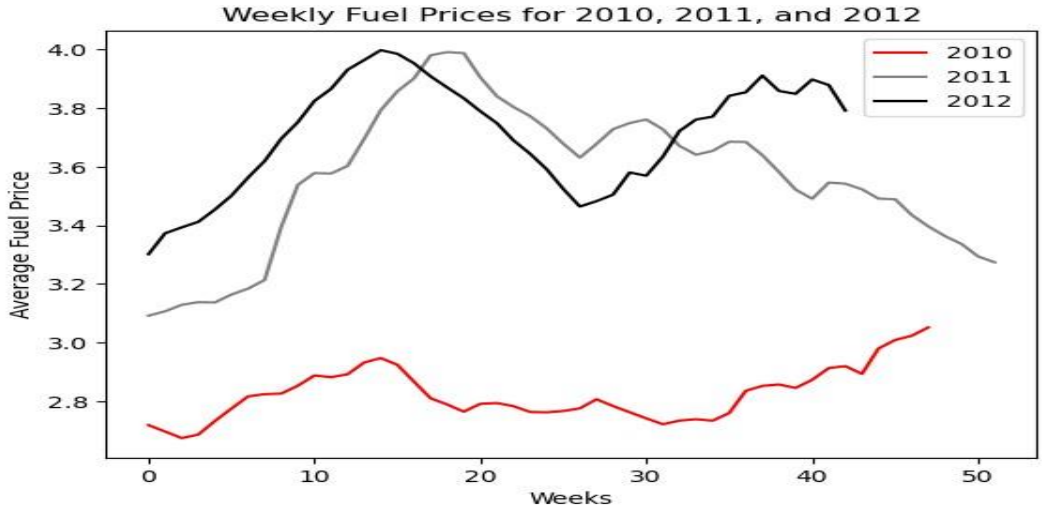


Weekly Fuel Price over years (2010,2011,2012), below line plot depicts average weekly fuel prices over 52 weeks, with lines representing the years 2010 (red), 2011 (grey), and 2012 (black).

In 2010, Fuel prices start lower and remain relatively stable throughout the year, with slight fluctuations. The overall trend is fairly consistent with a small upward slope towards the year's end.

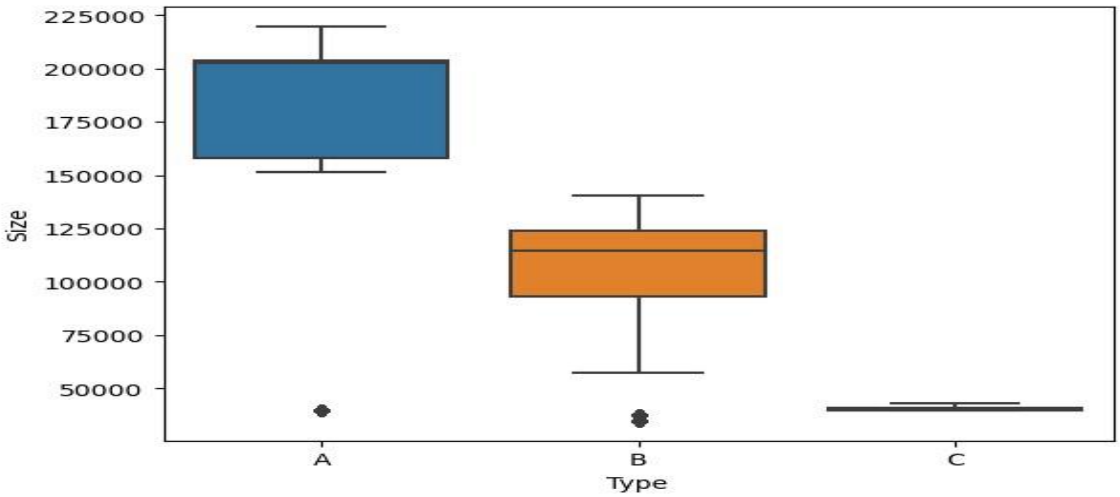
In 2011, The prices rise steeply in the first half of the year, reaching a peak around midyear, then declining. This indicates that 2011 experienced a significant spike in fuel prices compared to 2010.

In 2012, Fuel prices follow a similar pattern to 2011, with an initial rise, a peak, and then a decline, but the peak is somewhat lower. The year ends with a slight increase in prices.



Fuel prices generally increase year by year due to a combination of factors such as inflation, changes in global oil prices, and geopolitical events. Inflation drives up the cost of goods and services over time, including fuel. Global oil prices are influenced by supply and demand dynamics leading to a gradual increase year by year.

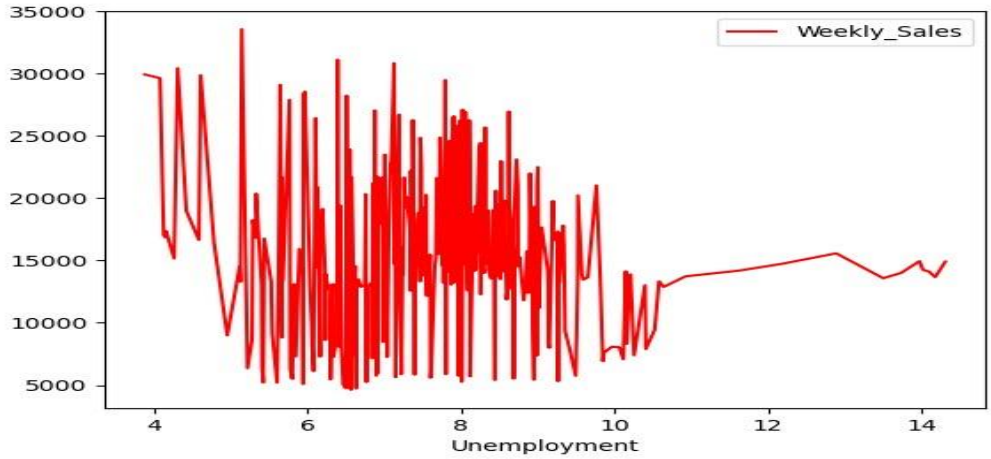
Relationship between store size and store type We're using a boxplot to analyse the relationship between store size and store type. Type A generally has larger stores, with Type B following. This could indicate that Type A represents a different business model, such as larger retail spaces or more extensive product offerings. Type C has the smallest stores, suggesting a more focused or good business approach.



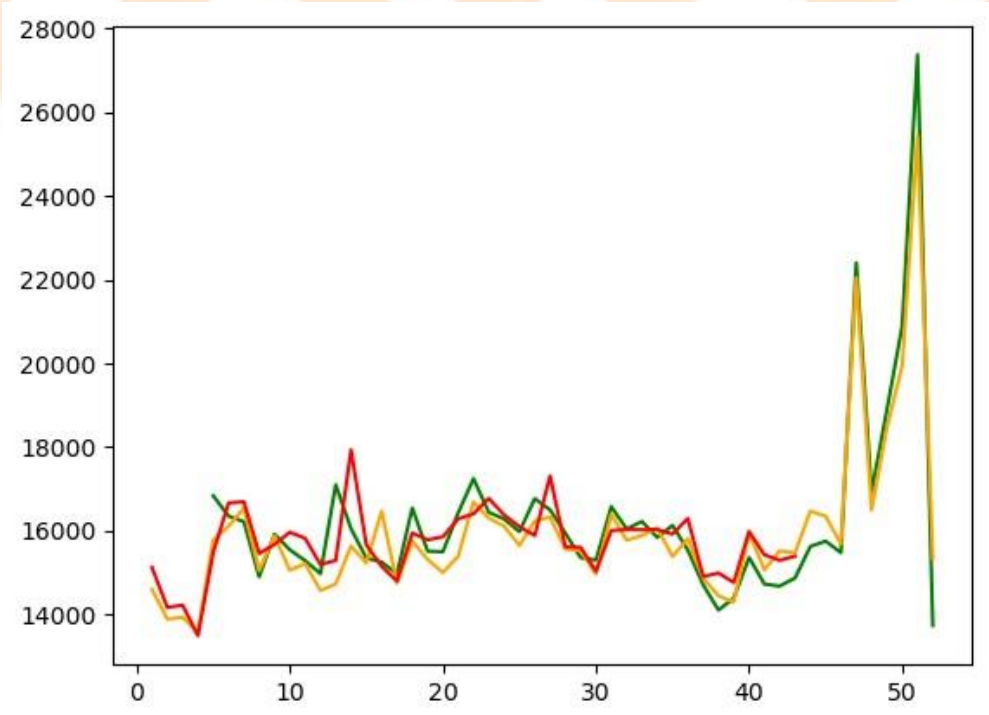
Relationship between unemployment rates and weekly sales, the line plot indicates the relationship between unemployment rates and weekly sales in a retail company. There's noticeable fluctuation in weekly sales across different levels of unemployment. At lower unemployment rates (around 4 to 8), weekly sales are more variable, indicating inconsistency in retail performance.

As unemployment rates rise (above 8), weekly sales show a downward trend. This aligns with general economic theory, where higher unemployment typically leads to reduced

consumer spending, impacting retail sales. External economic conditions, such as inflation or market trends, might also play a role in affecting weekly sales. Companie might need to adjust their marketing strategies or product offerings during periods of high unemployment to maintain sales performance.



Weekly sales over the year, the below chat tells quarter four holds higher sales, Higher sales towards the end of the year often reflect seasonal trends, such as holiday shopping. Events like Black Friday, Cyber Monday, and Christmas contribute to increased consumer spending. This pattern is common in retail, where the fourth quarter typically sees the most sales activity. Businesses often run major marketing campaigns and promotions in the final months of the year, aiming to boost sales during the holiday season. This could include discounts, special offers, and holiday-themed marketing, which drive increased customer traffic. Consumers may have much interest to spend during the holiday season, leading to increased sales towards the year's end. Gift-giving and end-of-year bonuses can contribute to higher discretionary spending.



,

PART 2:

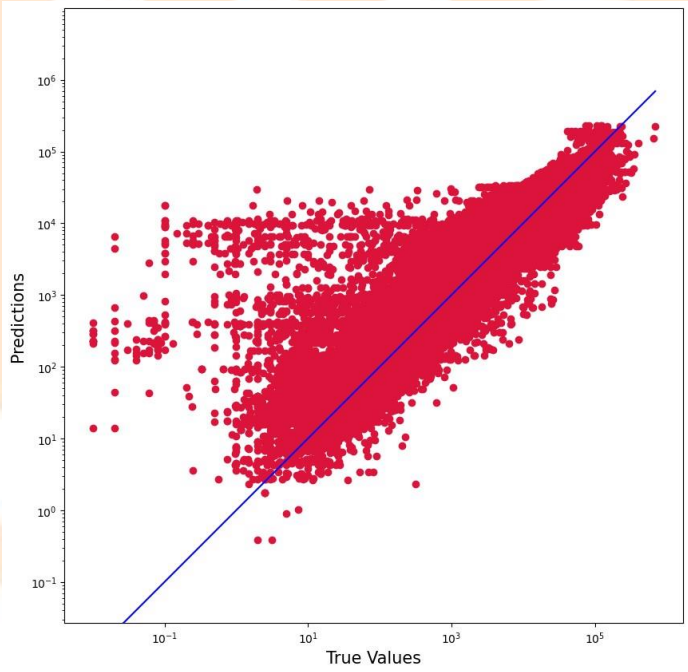
The following models were selected to address our problem:

- Random Forest
- Decision Tree
- Logistic Regression
- Linear Regression

These models were chosen based on the characteristics of our data and the specific tasks at hand. Given the structure and complexity of our data, tree-based classifiers like Random Forest and Decision Tree were considered ideal. For predicting and classifying outcomes, we opted for traditional classifiers such as Linear Regression, Logistic Regression, and Support Vector Machines. This diverse selection of models allows us to approach the problem from multiple angles and achieve robust results.

Random forest:

Random Forest is an ensemble learning method that constructs multiple decision trees during training and merges their outputs to improve prediction accuracy and reduce the risk of overfitting.



This scatter visualizes the predictions of a Random Forest model against the true values, with a blue diagonal line representing perfect predictions (where predicted values equal true values). The blue diagonal line represents the ideal outcome, indicating a perfect match between predicted and true values. The scatter of red dots indicates the distribution of predictions relative to the true values.

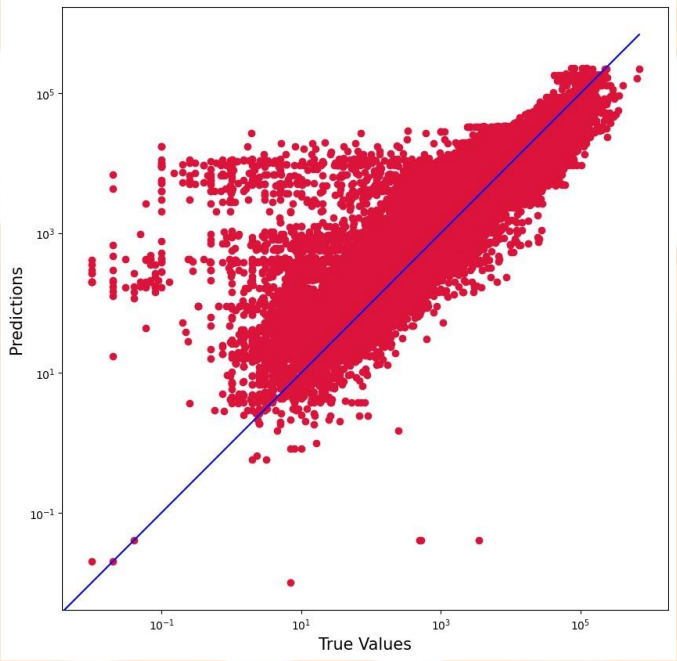
We're predicting the store's weekly sales and comparing these predictions to true values of predictor. With a 92% accuracy rate, the plot suggests that the Random Forest model performs well overall. Most predictions are concentrated near the blue diagonal line, indicating a good fit to the data. Wider dispersion of points from the line suggests areas

where the model's predictions deviate from the true values. Outliers (points far from the diagonal) represent predictions with high errors.

Decision Tree

A Decision Tree is a supervised learning algorithm that makes predictions by splitting data into branches based on feature conditions, leading to a series of decisions until reaching an outcome at the terminal nodes. It is straightforward to interpret but can be overfit without proper pruning.

We're predicting the store's weekly sales and comparing these predictions to true values of predictors. with a 91.5% accuracy rate, the blue diagonal line represents perfect predictions, where predicted values match the true values exactly.



The Decision Tree model achieved an accuracy rate of 91.5%, indicating strong predictive capability. The proximity of the data points to the blue diagonal line suggests that the model generally predicts well. Outliers are visible, these outliers might indicate cases where the model struggles with complex or unusual data patterns, suggesting areas for model improvement or data pre-processing.

The dispersion in predictions could result from overfitting, where the model learns specific patterns in the training data but fails to generalize well to unseen data. The variability and outliers suggest that the model may benefit from additional tuning, pruning, or the use of ensemble techniques to reduce overfitting. Hence Decision Tree model's 91.5% accuracy indicates strong performance, but the scatter plot reveals some variability and outliers.

Logistic Regression

Earlier, we predicted weekly sales with Decision Tree and Random Forest classifiers. Now, we aim to forecast sales using the "Holiday" column as a predictor. The choice of algorithms is somewhat limited when predicting weekly sales, so we are experimenting with different methods and predictors to improve our results. As part of this exploration, we've decided to use Logistic Regression for this task.

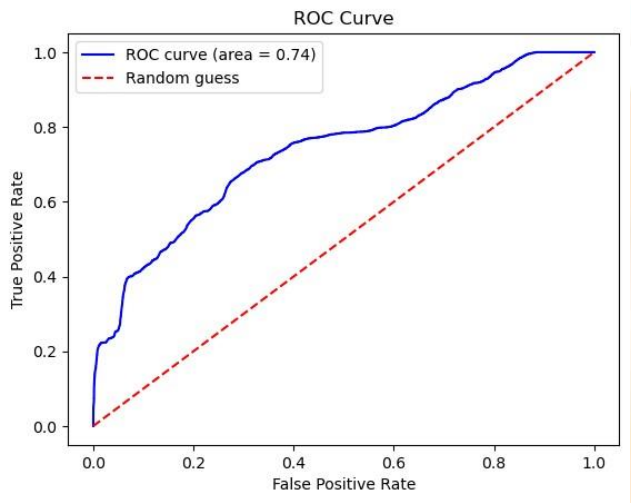
Logistic Regression is a supervised learning algorithm used for binary and multi-class classification. It models the probability that a given input belongs to a specific class by fitting data to a logistic curve, providing a straightforward and interpretable approach to classification tasks.

The ROC (Receiver Operating Characteristic) curve in this plot evaluates the performance of a logistic regression model. It helps visualize how well the classifier distinguishes between classes at varying thresholds, providing insight into the model's overall effectiveness.

This curve plots two parameters: True Positive Rate and False Positive Rate

True Positive Rate: Also known as sensitivity or recall, it represents the proportion of actual positives correctly identified by the classifier.

False Positive Rate: It indicates the proportion of actual negatives incorrectly classified as positives.



The ROC curve's position above the diagonal and the AUC value of 0.74 show that the logistic regression model has predictive value, with a 92% accuracy rate indicating strong overall performance. While the model is effective, the curve's spread and distance from the top-left corner suggest there may be room for improvement through further tuning or additional feature engineering.

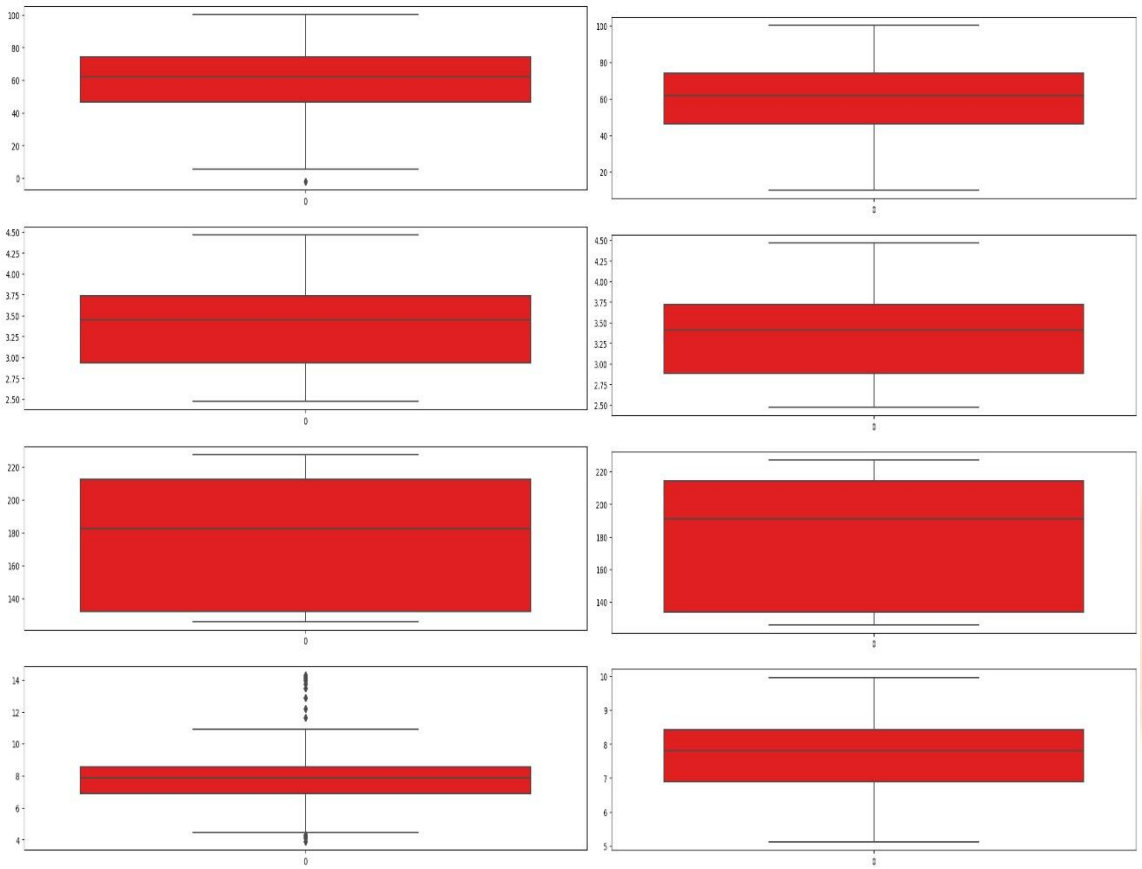
Linear Regression:

The first step is to identify any outliers that could affect the model's accuracy. Once located, these outliers will be removed. The box plot is a visualization tool that displays the distribution of a dataset, highlighting key statistics like the median, quartiles, and outliers.

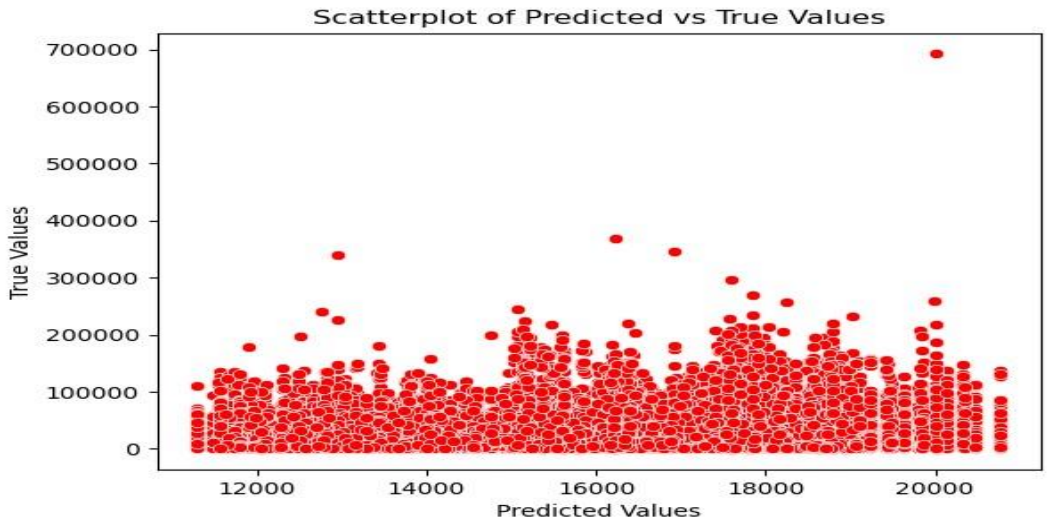
Outliers in the box plot appear as individual points or clusters of points outside the whiskers. These outliers can skew linear regression results, leading to inaccurate predictions and increased error. By identifying outliers with box plots, you can remove them from the dataset before training the linear regression model. This process helps improve the model's accuracy.

The plot displays the predictions versus true values for a linear regression model. The linear regression achieved an accuracy of 81.7%. Several outliers are visible, suggesting predictions that significantly deviate from true values. An accuracy rate of 81.7% suggests the linear regression model is reasonably accurate but has room for improvement.

Temperature, Fuel_Price, CPI, Unemployment



The linear regression model demonstrates moderate accuracy but also shows significant variability and error. The error metrics highlight areas where the model's assumptions may not align with the underlying data patterns.



PART 3:

After experimenting with four different machine learning models on the dataset, we've chosen Random Forest as the best model to use going forward. Our decision is based on two key factors: the model's accuracy and its handling of input features. Among all the models tested, Random Forest stands out with an accuracy exceeding 90%.

Out of the four models, only a few can take three input features and predict a numerical value for weekly sales. These include Random Forest, Decision Tree, and Linear Regression. Among these models, Random Forest proves to be the most suitable for our data, which is why we selected it for further testing and eventual implementation.

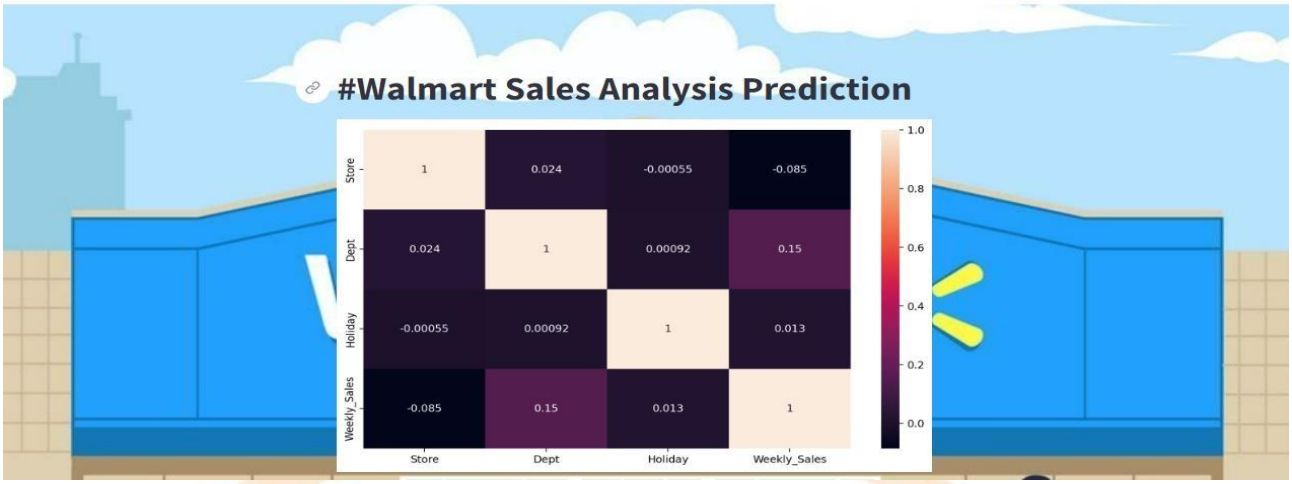
With the model selected, we're shifting focus to fine-tuning it for optimal performance. Our next step involves building an interactive webpage where users can enter their input values to receive an estimate of their expected weekly sales. This feature will allow users to leverage the model's predictive capabilities for their specific scenarios, providing a practical and user-friendly tool for real-time sales estimation.

Our web application has visualizations that shows the correlation matrix which tells correlation between the variables like department, weely sales, temperature and holiday which is user- friendly. It contains three different phases, they are visualization, input, and output.

Upon visiting the website, users can quickly understand how various factors correlate and impact store sales, as well as view the performance of different departments. This provides a comprehensive overview of the elements influencing sales and insights into departmental contributions and other chat shows departmental sales.

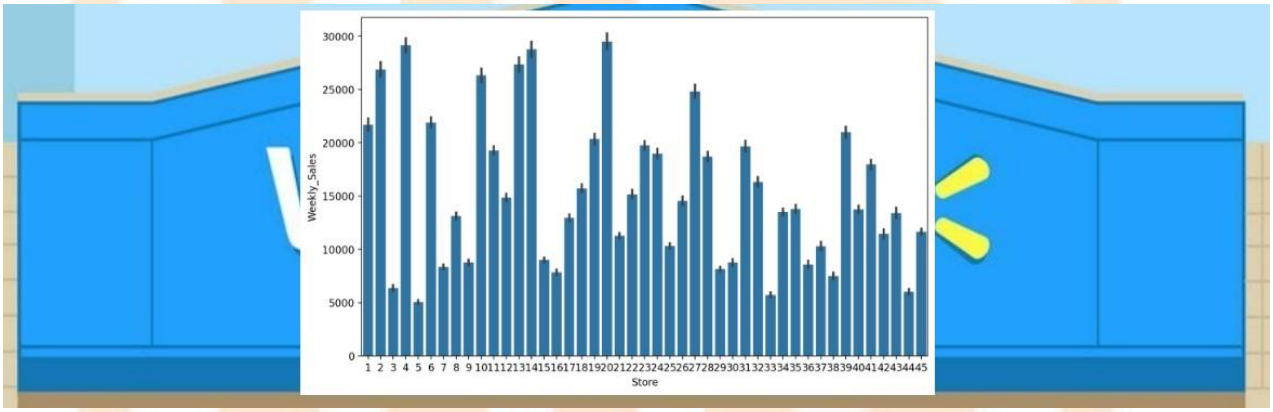
We attempted to build the user interface (UI) using a Streamlit application. This framework enables the creation of interactive and data-driven web applications, allowing users to visualize data and interact with various features in a simple and intuitive way.

Outputs of the web application:



The heatmap depicts the correlation between various factors, showing how they are related to each other. darker shades indicating lower correlation and lighter shades indicating higher correlation.

The strongest correlation among these factors is between Dept and Weekly Sales, suggesting that the type of department might play a minor role in weekly sales.



The bar chart represents weekly sales across various stores. There is a noticeable variation in weekly sales across different stores, indicating that some stores consistently achieve higher sales while others maintain lower sales. These peaks may represent larger or more popular stores, or those located in high-traffic areas.

The user interface (UI) provides input fields for Store ID, Department, and Holiday, allowing users to enter specific information. A "Calculate Weekly Sales" button is available to generate predictions based on these inputs. This interactive feature enables users to get an estimate of expected weekly sales by providing key data points, making it a user-friendly and practical tool for sales estimation.

Input:

The input section includes three fields: Store ID, where you can specify the store whose sales you want to predict or analyze; Department, allowing you to focus on sales from a specific department; and a Holiday field, indicating whether holidays might impact weekly sales. This setup provides flexibility for users to customize their input and determine if these factors influence weekly sales.

Output:

Case 1:

Store number 20 and department number 29 on holiday week sales would be 9913.37.

Case 2:

Similarly, Store number 20 and department number 29 on non-holiday week sales would be 8832.02.

Conclusion:

Holiday weeks contribute to a slight increase in weekly sales, indicating a moderate impact during these periods. Sales tend to dip in extreme weather conditions, with a noticeable improvement when temperatures are moderate. In contrast, fluctuations in fuel prices show no significant effect on sales. Overall, the data suggests that weather and holidays are more reliable indicators of sales trends, while fuel prices play a less decisive role. In addition, from part 3 we can predict weekly sales based on the holiday factors.

References: https://www.kaggle.com/code/gcdatkin/walmart-holiday-sale-prediction

https://streamlit.io/

https://www.infoworld.com/article/3715100/intro-to-streamlit-web-based-python-dataapps-made-easy.html https://scikit-learn.org/stable/supervised_learning.html#supervised-learning

https://www.youtube.com/watch?v=D0D4Pa22iG0

THANK YOU