WALMART EXPLORATORY DATA ANALYSIS (EDA)

Prepared for Walmart

Presented To INVENTORY MANAGERS

Presented By AMALAI RHAM MACHUNGA

WALMART DATA ANALYSIS

Case Study

- I am a Data Analyst recently recruited by Walmart which has multiple retail stores nationwide. My task is to help the inventory managers with insights that could help them manage inventories to match demand and supply.
- I have been assigned to conduct data analysis and develop data-driven recommendations for the inventory manager.
- I have one (1) week to present my analysis and recommendations to the managers.

About Walmart

Walmart is one of the world's leading multinational retail corporations with a huge turnover. It was founded in the year 1962 and was incorporated in the year 1969 with its headquarters in Bentonville, Arkansas.

Besides being a huge firm, Walmart is still controlled by the Walton family. It operates globally with more than 11,700 retail stores in 28+ countries. Also, it serves its e-retailers in eleven countries.

Dataset Description

Dataset: https://www.kaggle.com/datasets/asahu40/walmart-data-analysis-and-forcasting

Initially, the dataset had 8 columns and 6,435 rows, I feature-engineered the columns to 11 by splitting the 'Date' column into 'Day', 'Month', and 'Year'.

Interestingly, I found a strong correlation between the 'Weekly Sales' and the 'Month' columns.

Description of the Columns

Store: *Store number*. **Date**: *Week of sales*.

Weekly Sales: Weekly sales for the given store.

Holiday Flag: Whether the week is a special holiday week or not (1 = holiday week; 0 = non-

holiday week).

Temperature: Average temperature in the region for the given week.

Fuel Price: *Cost of fuel in the region for the given week.*

CPI: Consumer price index for the given week.

Unemployment: *Unemployment rate for the region for the given week.*

Tools Used

Python, Visual Studio Code, GitHub, Microsoft Word

Issues / Difficulties Encountered

1. The 'Date' column was in a String (Object) datatype format; I needed to convert it to a datetime datatype. I used the .to_datetime function of the pandas dataframe to convert the date column into a datetime format.

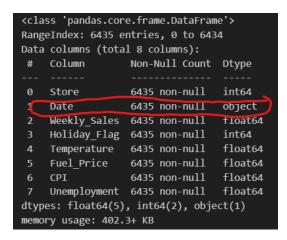


Fig: Object (String) datatype

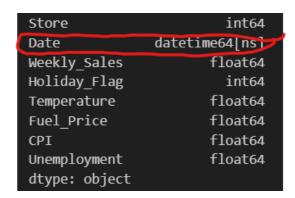


Fig: Datetime format

2. Converting the exponential outputs of the 'Weekly_Sales' column from exponential to integer values by using the .astype(int) function to make it easier to read.

```
Total_weekly_sales = df.groupby('Store')['Weekly_Sales'].sum()
print("Total Weekly Sales For each store: ", Total_weekly_sales)
    0.0s
Total Weekly Sales For each store: Store
       2.224028e+08
       2.753824e+08
       5.758674e+07
       2.995440e+08
       4.547569e+07
       2.237561e+08
      8.159828e+07
       1.299512e+08
8
      7.778922e+07
      2.716177e+08
10
       1.939628e+08
      1.442872e+08
2.865177e+08
      2.889999e+08
8.913368e+07
14
      7.425243e+07
1.277821e+08
16
      1.551147e+08
2.066349e+08
3.013978e+08
1.081179e+08
      1.470756e+08
       1.987506e+08
       1.940160e+08
       9.056544e+07
       4.329309e+07
       1.123953e+08
Name: Weekly_Sales, dtype: float64
```

Fig: Exponential outputs

```
Average_weekly_sales = df.groupby('Store')['Weekly_Sales'].mean().astype(int)
   print("Average Weekly Sales For each store: ", Average_weekly_sales)
Average Weekly Sales For each store: Store
      1555264
      1925751
      402704
      2094712
      318011
      1564728
      570617
      543980
10
     1899424
     1356383
      1009001
      2003620
14
      2020978
      519247
16
      1084718
18
      1444999
      2107676
20
      756069
      1028501
      1389864
      1356755
44
       302748
       785981
Name: Weekly_Sales, dtype: int32
```

Fig: Using the .astype(int) function

3. I encountered 'depreciation' warnings, which I ignored by adding the Python script below.

import warnings
warnings.filterwarnings("ignore")

Fig: Python script to ignore depreciation warnings

Findings

- 1. Store Number '20' has the highest Weekly Sales across all 45 Walmart Stores.
- 2. Store Number '33' has the lowest Weekly Sales across all 45 Walmart Stores.
- 3. The Stores experience more Weekly Sales when it is a non-holiday week (0).
- 4. Q4 (October December) Weekly Sales increases are experienced across all stores.

Recommendations

- 1. Increase inventories during the Q4 fourth quarter to match the increasing sales.
- 2. Increase Inventories on days that are marked as non-holiday weeks as compared to holiday weeks.