10 Academy
Kifiya AI Mastery Training Program 5

Jerusalem F.
(MSc, MSc DS)

June 2025

10 Academy

Building-an-Amharic-E-commerce-Data-Extractor

Data inspection

	Store	DayOfWeek	Sales	Customers	Open	Promo	StateHoliday	SchoolHoliday	Year	Month	...	Dates	StoreTy
0	1	5	5263.0	555.0	1	1	0	1	2015	7	...	2015-07-31	
1	2	5	6064.0	625.0	1	1	0	1	2015	7	...	2015-07-31	
2	3	5	8314.0	821.0	1	1	0	1	2015	7	...	2015-07-31	
3	4	5	13995.0	609.0	1	1	0	1	2015	7	...	2015-07-31	
4	5	5	4822.0	559.0	1	1	0	1	2015	7	...	2015-07-31	
...	...	...	...	...	...	...	...	...	...	...	...	...	
1017204	1111	2	0.0	0.0	0	0	a	1	2013	1	...	2013-	

Checking missing values form the train dataset


```
DayOfWeek                          0
Sales                              0
Customers                          0
Open                               0
Promo                              0
StateHoliday                       0
SchoolHoliday                      0
Year                               0
Month                              0
Day                                0
WeekOfYear                         0
Dates                              0
StoreType                          0
Assortment                         0
CompetitionDistance                0
Promo2                             0
Promo2SinceWeek                    0
Promo2SinceYear                    0
PromoInterval                      0
SalesperCustomer              172869
CompetitionOpenSince               0
dtype: int64
```

For discussion see the next slide

cont…

The df_train.isnull().sum() is used to check for missing values in a pandas DataFrame called df_train. The method isnull() creates a boolean mask of the same shape as the DataFrame, where True indicates a null value and False indicates a non-null value. The sum() function then counts the number of True values in each column.

The output shows the count of null values for each column in the DataFrame. In this case, the output indicates that there are no missing values (all counts are 0) in any of the columns: Store, DayOfWeek, Date, Sales, Customers, Open, Promo, StateHoliday, and SchoolHoliday. This is a good sign for data quality, as it means the dataset is complete without any null entries that might require handling or imputation before further analysis or modeling.

Checking missing values form the tset dataset

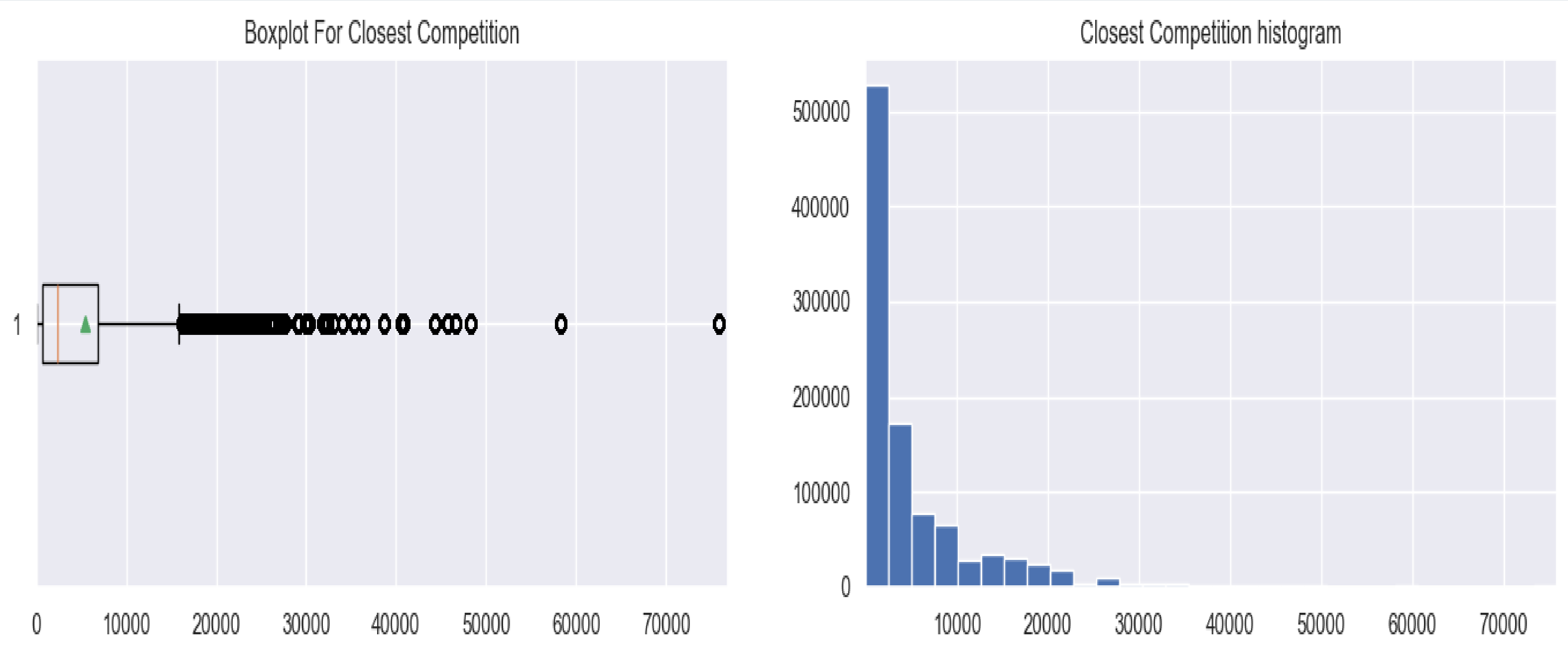

```
#Checking missing values
df_test.isnull().sum()
[12] ✓ 0.0s

Id                0
Store             0
DayOfWeek         0
Date              0
Open             11
Promo             0
StateHoliday      0
SchoolHoliday     0
dtype: int64
```

The df_test.isnull().sum() is used to check for missing values in another DataFrame called df_test, likely representing a test dataset. The output shows the count of null values for each column in this DataFrame. Unlike the previous example with df_train, this output reveals that there are 11 missing values in the 'Open' column of df_test. All other columns (Id, Store, DayOfWeek, Date, Promo, StateHoliday, and SchoolHoliday) have no missing values.

This information is crucial for data preprocessing, as it highlights that the 'Open' column in the test dataset may require special handling, such as imputation or exclusion, depending on the specific requirements of the analysis or modeling task. The presence of missing values in the test set, but not in the training set, also suggests that the data collection or preparation process might have been different between the two datasets, which could be worth investigating further.

Distribution graphs closet competion



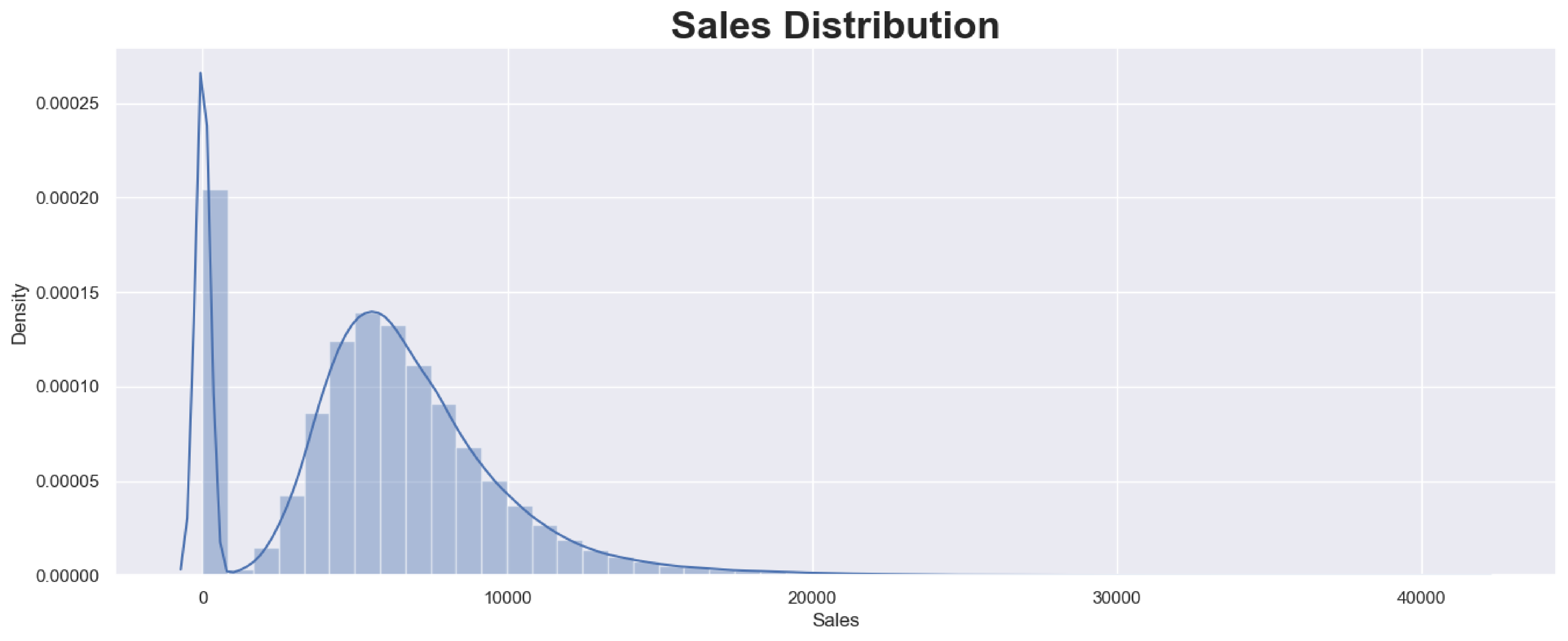
cont…

From the slide 6 figure, the goal is to analyze the distribution of the CompetitionDistance variable from the train_store DataFrame, specifically focusing on stores that have a valid competition distance (i.e., excluding null values). The first line creates a new DataFrame, df_store_check_distribution, by dropping any rows from train_store where CompetitionDistance is null.

Then, two plots are generated: a boxplot and a histogram of the CompetitionDistance. The boxplot provides a visual summary of the distribution, showing outliers and the mean, while the histogram illustrates the frequency distribution of the competition distances across 30 bins. Lastly, the code computes and displays the mean, median, and standard deviation of the CompetitionDistance, revealing significant differences: a mean of approximately 5430.09, a median of 2330.0, and a standard deviation of about 7715.32.

These statistics indicate a right-skewed distribution, where the mean is heavily influenced by a few stores with extremely high competition distances, while the median represents a more typical value.

Sales distribution



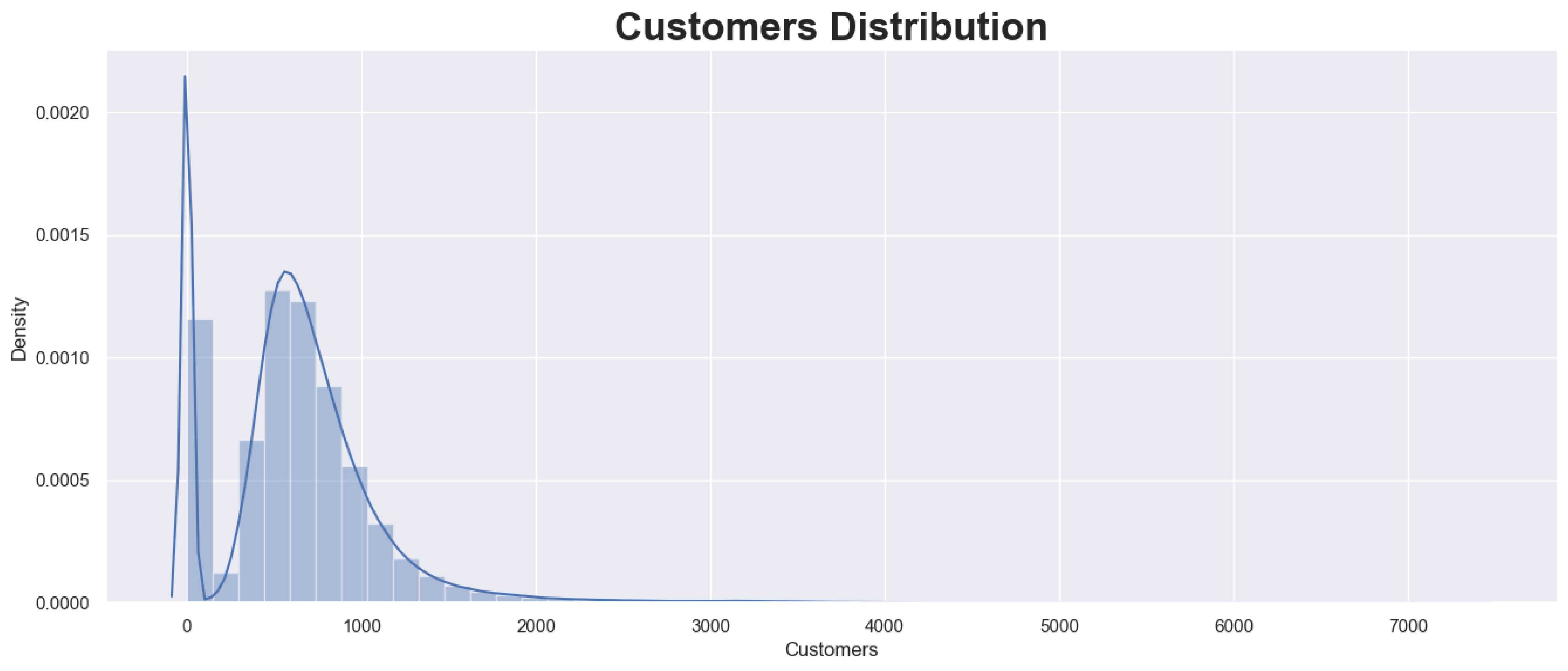
cont…

The slide 8 figure, visualization of the sales distribution, using the Seaborn library in Python sets up a matplotlib figure with a size of 16x6 inches. Then, it uses Seaborn's distplot function to create a histogram with a kernel density estimate (KDE) overlay of the 'Sales' column from the cleaned_df DataFrame.

The output of this code is the image you've shared. It shows a sales distribution graph with sales values on the x-axis ranging from 0 to about 40,000, and density on the y-axis.

The distribution appears to be right-skewed, with a peak around 5,000-7,000 in sales. There's also a smaller, sharp peak near 0, suggesting a significant number of very low or zero sales occurrences. The graph uses a blue color scheme and includes both a histogram and a smoothed density curve.

Customers distribution



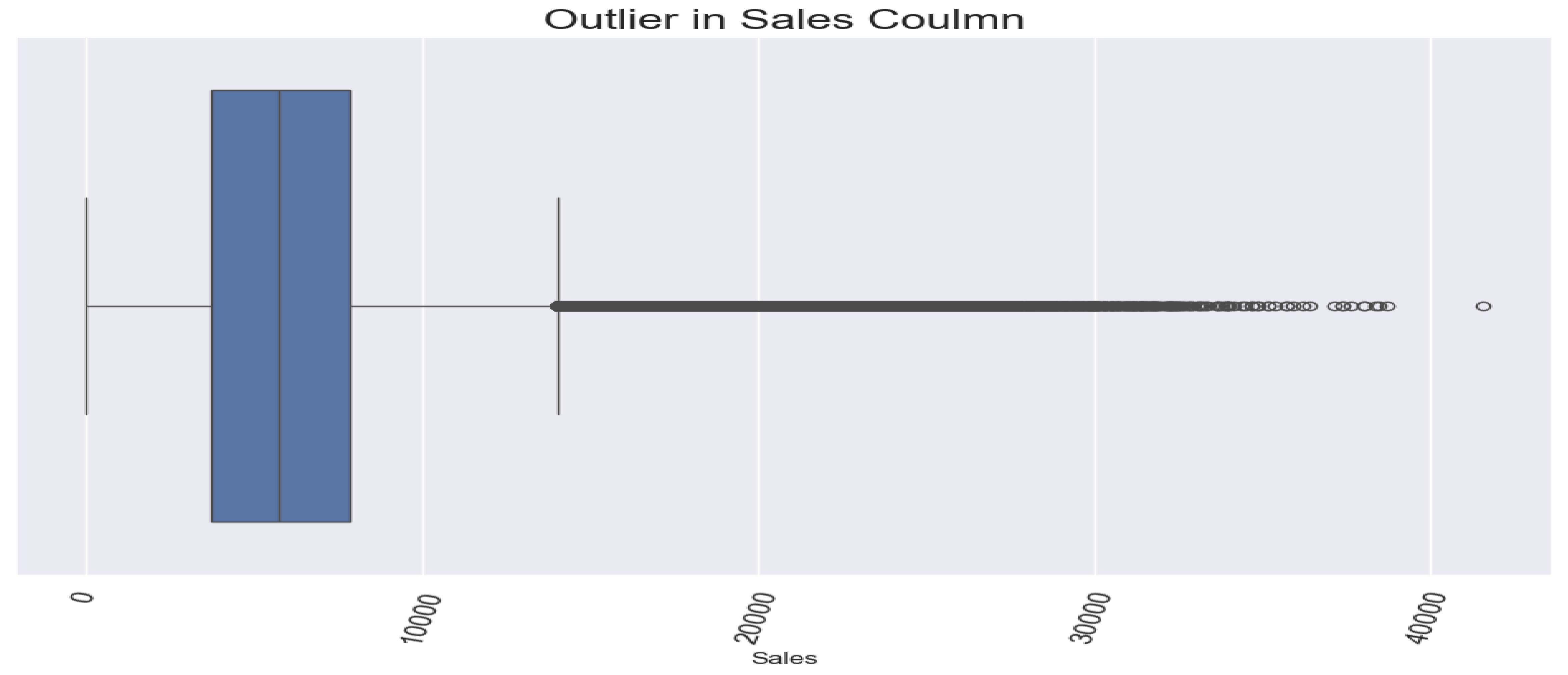
cont…

The sldie 10 figure, visualization of the customer distribution, using Seaborn and Matplotlib libraries in Python sets up a figure with dimensions of 16x6 inches, then uses Seaborn's distplot function to plot a histogram with a kernel density estimate (KDE) of the 'Customers' column from the cleaned_df DataFrame.\

The output is the image you've shared. It displays a right-skewed distribution of customers, with the x-axis representing the number of customers (ranging from 0 to about 7000) and the y-axis showing the density. There's a prominent peak near zero, suggesting many instances with very few customers.

A second, broader peak occurs around 400-600 customers, indicating this as a common range. The distribution has a long tail extending to the right, implying some cases with very high customer numbers, though these are less frequent. The graph uses a blue color scheme for both the histogram bars and the smoothed density curve.

Outliers in sales



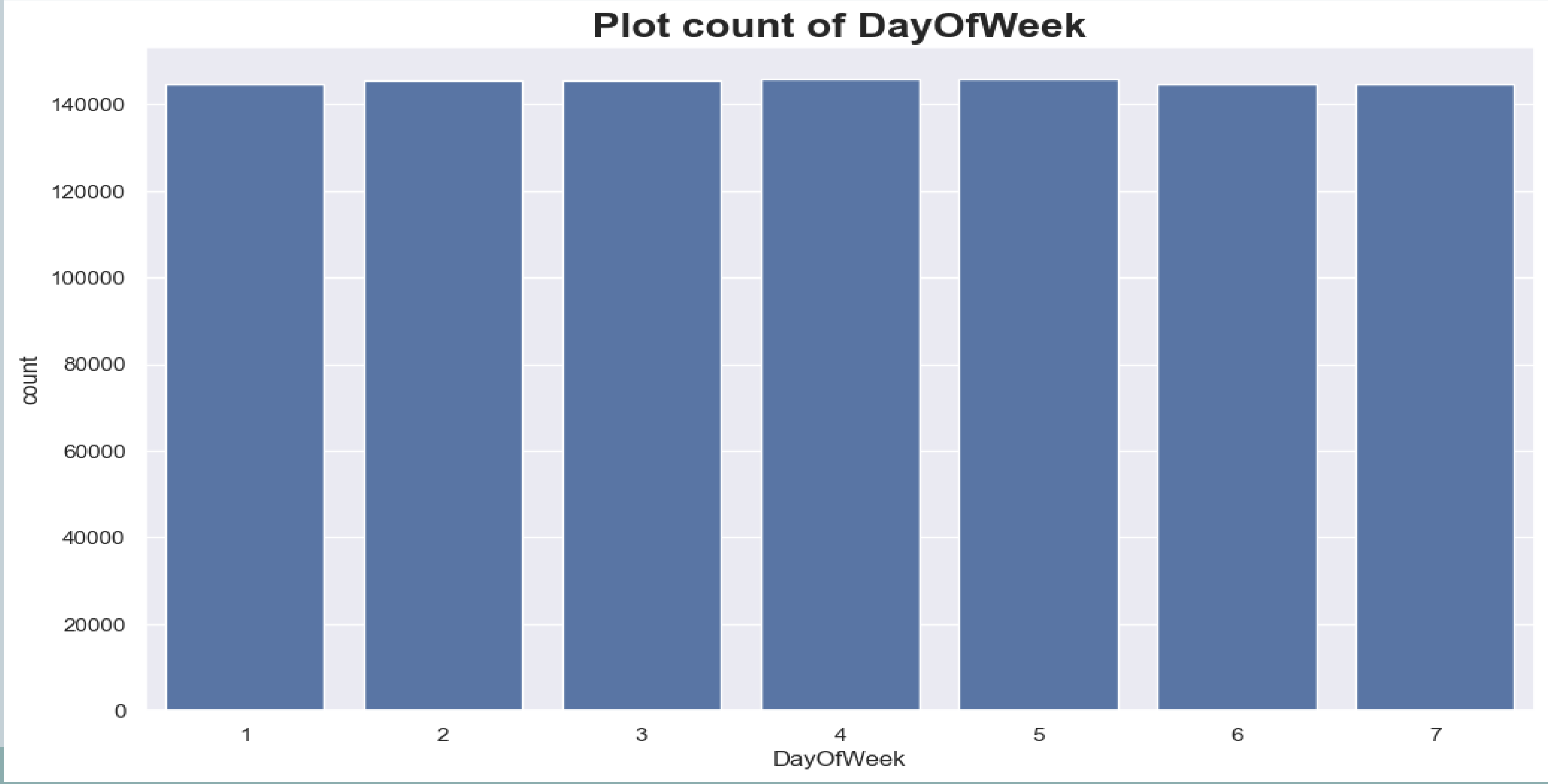
cont…

The slide 12 boxplot displayed highlights the presence of outliers in the "Sales" column of the dataset. The box represents the interquartile range (IQR), which encompasses the middle 50% of the data, while the line inside the box indicates the median sales value.

The whiskers extend to the minimum and maximum values within 1.5 times the IQR from the quartiles, illustrating the range of typical sales. However, several points beyond the whiskers are identified as outliers, indicated by individual dots, which may represent unusually high sales figures.

These outliers could significantly impact analysis and modeling, suggesting the need for further investigation to understand their causes, whether they are due to data entry errors, exceptional circumstances, or legitimate high-value transactions.

DayOfWeek


Plot count of DayOfWeek
count
140000
120000
100000
80000
60000
40000
20000
0
1
2
3
4
5
6
7
DayOfWeek

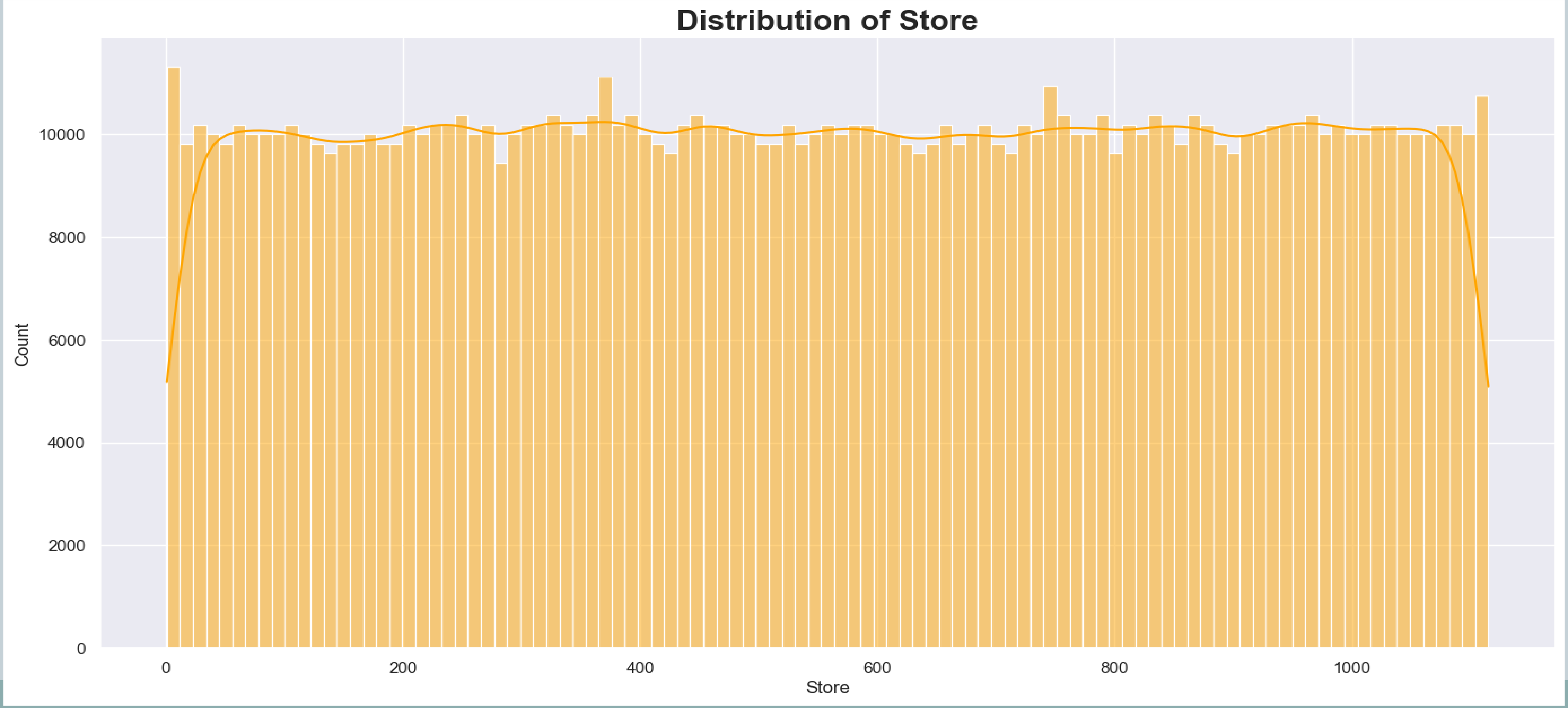
cont…

The bar plot illustrates the distribution of sales across the days of the week, represented by the "DayOfWeek" variable, where each bar corresponds to a specific day numbered from 1 to 7.

The uniform height of the bars indicates that each day has a similar count of transactions, suggesting that sales are consistently distributed throughout the week without significant fluctuations. This pattern may imply that customer behavior remains stable across different days, which can inform inventory management and staffing decisions.

However, further analysis might be needed to explore any underlying factors that could influence sales on specific days, such as promotions or seasonal trends.

Distribution of store



cont…

The figure presents the distribution of sales data across different stores, indicated by the "Store" variable on the x-axis, with the corresponding counts on the y-axis. The histogram bars, along with the overlaid line graph, reveal that the sales are relatively evenly distributed among the stores, with most stores showing a consistent count of transactions around the 5,000 to 10,000 range.

This uniformity suggests that no specific store significantly outperforms or underperforms compared to others, implying a balanced sales environment across the locations.

However, the slight fluctuations in counts may indicate variations that could be explored further, potentially revealing insights into individual store performance or customer preferences.

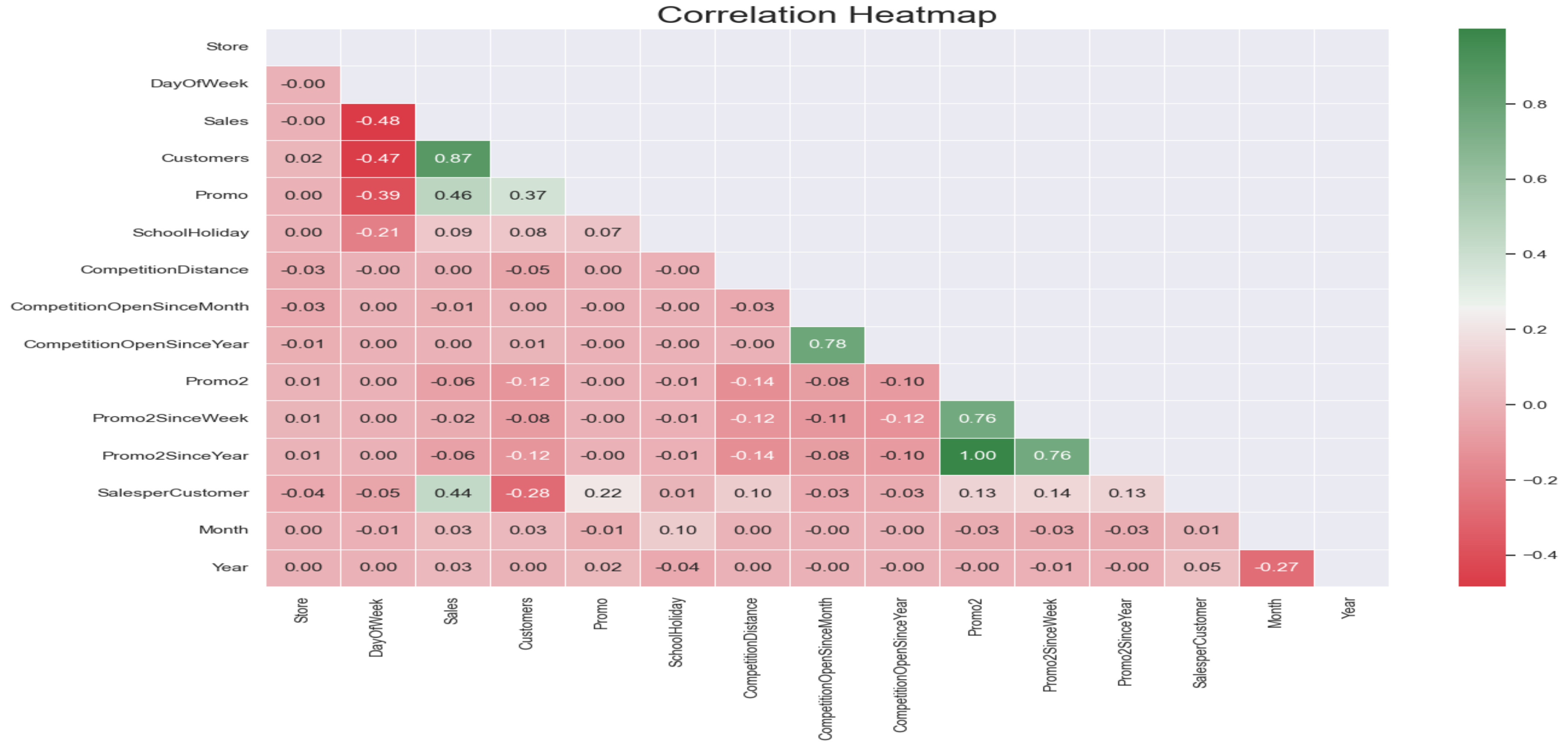
Feature Engineering

Feature engineering is the process of creating new variables or modifying existing ones in a dataset to improve machine learning model performance. It involves leveraging domain knowledge and data insights to extract more meaningful information from raw data.

Common techniques include creating ratios (e.g., sales per customer), binning continuous variables into categories, applying mathematical transformations (like logarithms for skewed distributions), encoding categorical variables, generating interaction terms between features, and developing time-based features if applicable.

The goal is to capture relevant patterns, reduce noise, and provide the model with more informative inputs. Effective feature engineering can significantly enhance a model's predictive power by highlighting underlying relationships in the data that may not be immediately apparent in the original features.

Correlation Analysis



cont…

This correlation heatmap for numeric variables in a dataset selects only numeric columns from the original DataFrame to avoid issues with non-numeric data. Then, it calculates the correlation matrix for these numeric columns. A mask for the upper triangle of the correlation matrix is created to avoid redundancy in the visualization.

The resulting plot provides a clear visual representation of the relationships between different variables in the dataset.

The output is the image provided, which shows a correlation heatmap. Key observations include: a strong positive correlation (0.87) between Sales and Customers; moderate positive correlation (0.46) between Sales and Promo; strong negative correlation (-0.48) between DayOfWeek and Sales; perfect correlation (1.00) between Promo2 and Promo2SinceYear; and several variables showing very weak or no correlation with others.

Prediction of store sales (Task-2)


```
C:\Users\you\AppData\Local\Temp\ipykernel_13572\2234050922.py:2: DtypeWarning: Columns (6) have mix
  train = pd.read_csv(train_PATH)
               Store     DayOfWeek         Sales     Customers          Open  \
count   1.017209e+06  1.017209e+06  1.017209e+06  1.017209e+06  1.017209e+06
unique           NaN           NaN           NaN           NaN           NaN
top              NaN           NaN           NaN           NaN           NaN
freq             NaN           NaN           NaN           NaN           NaN
mean    5.584297e+02  3.998341e+00  5.472856e+03  5.783493e+02  8.301067e-01
std     3.219087e+02  1.997391e+00  3.323989e+03  3.483565e+02  3.755392e-01
min     1.000000e+00  1.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00
25%     2.800000e+02  2.000000e+00  3.727000e+03  4.050000e+02  1.000000e+00
50%     5.580000e+02  4.000000e+00  5.744000e+03  6.090000e+02  1.000000e+00
75%     8.380000e+02  6.000000e+00  7.584000e+03  7.940000e+02  1.000000e+00
max     1.115000e+03  7.000000e+00  1.404900e+04  1.485000e+03  1.000000e+00

               Promo StateHoliday  SchoolHoliday          Year         Month  \
count   1.017209e+06      1017209   1.017209e+06  1.017209e+06  1.017209e+06
unique           NaN            5            NaN           NaN           NaN
top              NaN            0            NaN           NaN           NaN
freq             NaN       592943            NaN           NaN           NaN
```

```
..                 Id         Store     DayOfWeek        Date          Open  \
count    41088.000000  41088.000000  41088.000000       41088  41077.000000
unique            NaN           NaN           NaN          48           NaN
top               NaN           NaN           NaN  2015-09-17           NaN
freq              NaN           NaN           NaN         856           NaN
mean     20544.500000    555.899533      3.979167         NaN      0.854322
std      11861.228267    320.274496      2.015481         NaN      0.352787
min          1.000000      1.000000      1.000000         NaN      0.000000
25%      10272.750000    279.750000      2.000000         NaN      1.000000
50%      20544.500000    553.500000      4.000000         NaN      1.000000
75%      30816.250000    832.250000      6.000000         NaN      1.000000
max      41088.000000   1115.000000      7.000000         NaN      1.000000

              Promo StateHoliday  SchoolHoliday
count  41088.000000        41088   41088.000000
unique          NaN            2            NaN
top             NaN            0            NaN
freq            NaN        40908            NaN
mean       0.395833          NaN       0.443487
std        0.489035          NaN       0.496802
```


Thank You For
Your Attention!
Your questions, please!