Pharmaceutical Sales prediction across multiple stores

BY MEHRUNISA NAIK

Points for Discussion

Instructions

The task is divided into the following objectives

- Exploration of customer purchasing behavior
- Prediction of store sales
- Machine learning approach
- Deep Learning approach
- Serving predictions on a web interface

Data Summary

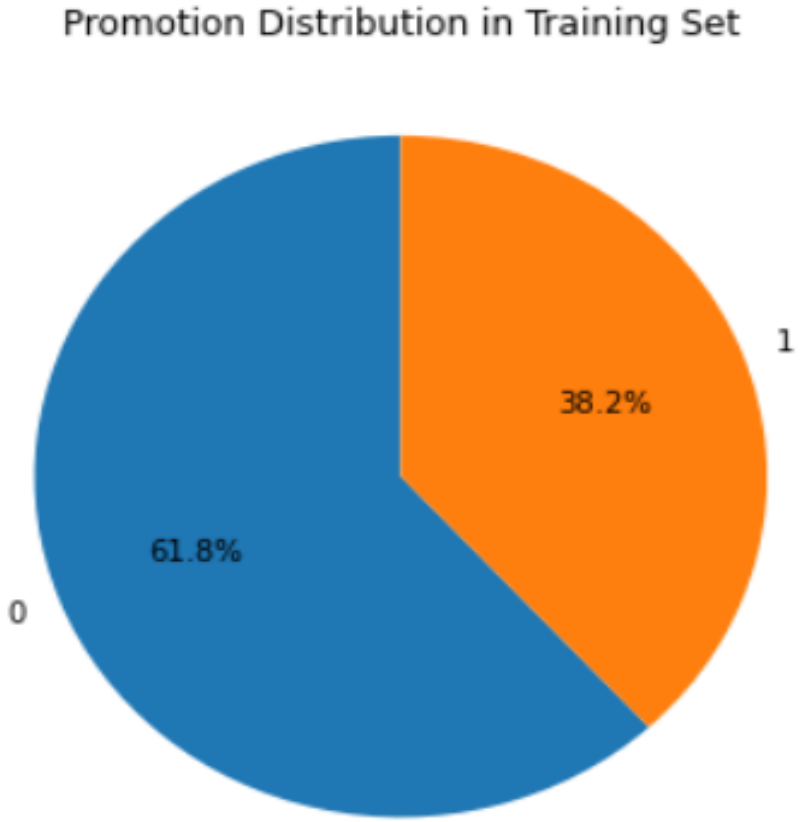
- **Id** - an Id that represents a (Store, Date) duple within the test set
- **Store** - a unique Id for each store
- **Sales** - the turnover for any given day (this is what you are predicting)
- **Customers** - the number of customers on a given day
- **Open** - an indicator for whether the store was open: 0 = closed, 1 = open
- **StateHoliday** - indicates a state holiday. Normally all stores, with few exceptions, are closed on state holidays. Note that all schools are closed on public holidays and weekends. a = public holiday, b = Easter holiday, c = Christmas, 0 = None
- **SchoolHoliday** - indicates if the (Store, Date) was affected by the closure of public schools
- **StoreType** - differentiates between 4 different store models: a, b, c, d
- **Assortment** - describes an assortment level: a = basic, b = extra, c = extended. Read more about assortment here

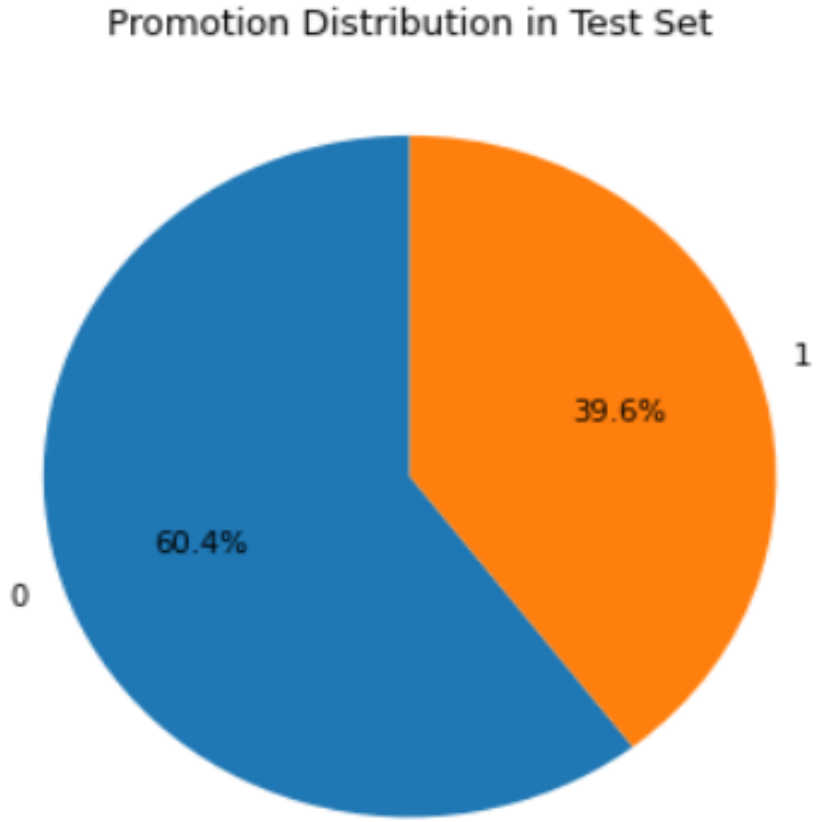
Data Summary

- **CompetitionDistance** - distance in meters to the nearest competitor store
- **CompetitionOpenSince[Month/Year]** - gives the approximate year and month of the time the nearest competitor was opened
- **Promo** - indicates whether a store is running a promo on that day
- **Promo2** - Promo2 is a continuing and consecutive promotion for some stores: 0 = store is not participating, 1 = store is participating
- **Promo2Since[Year/Week]** - describes the year and calendar week when the store started participating in Promo2
- **PromoInterval** - describes the consecutive intervals Promo2 is started, naming the months the promotion is started anew. E.g. "Feb,May,Aug,Nov" means each round starts in February, May, August, November of any given year for that store

Exploration of customer purchasing behavior

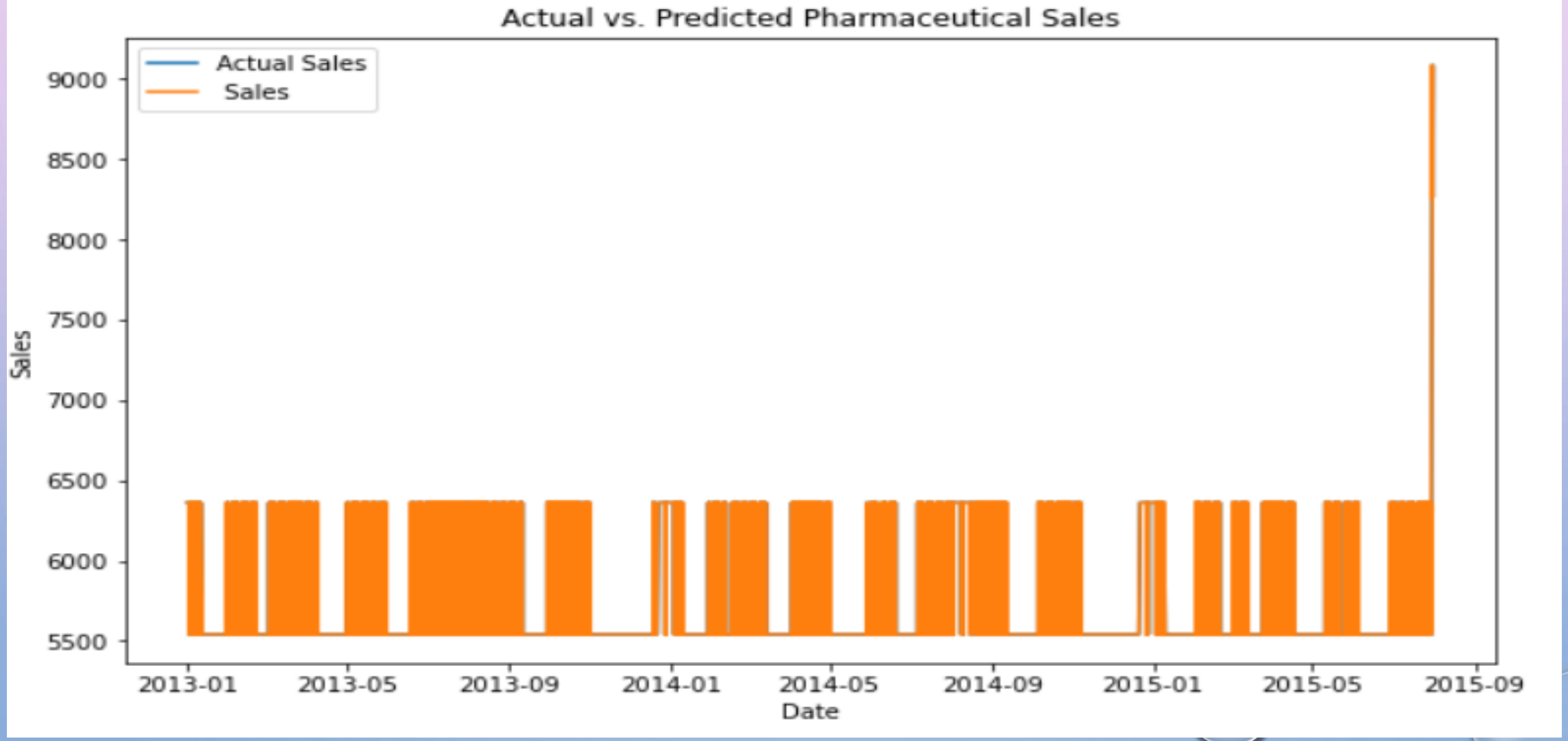
Distribution in both training and test sets





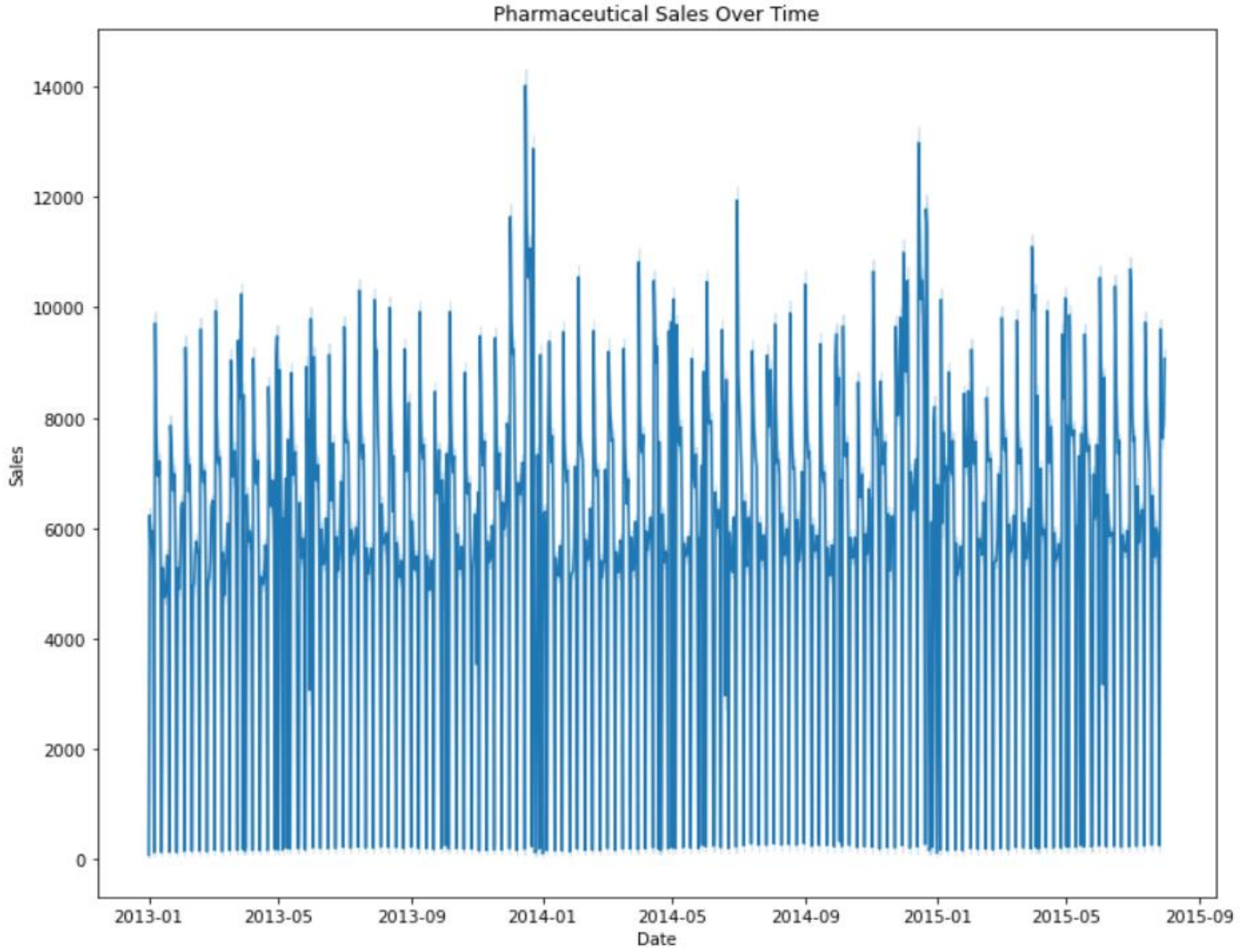
Predicted Pharmaceutical Sales

Predicted pharmaceutical sale using sales and date columns

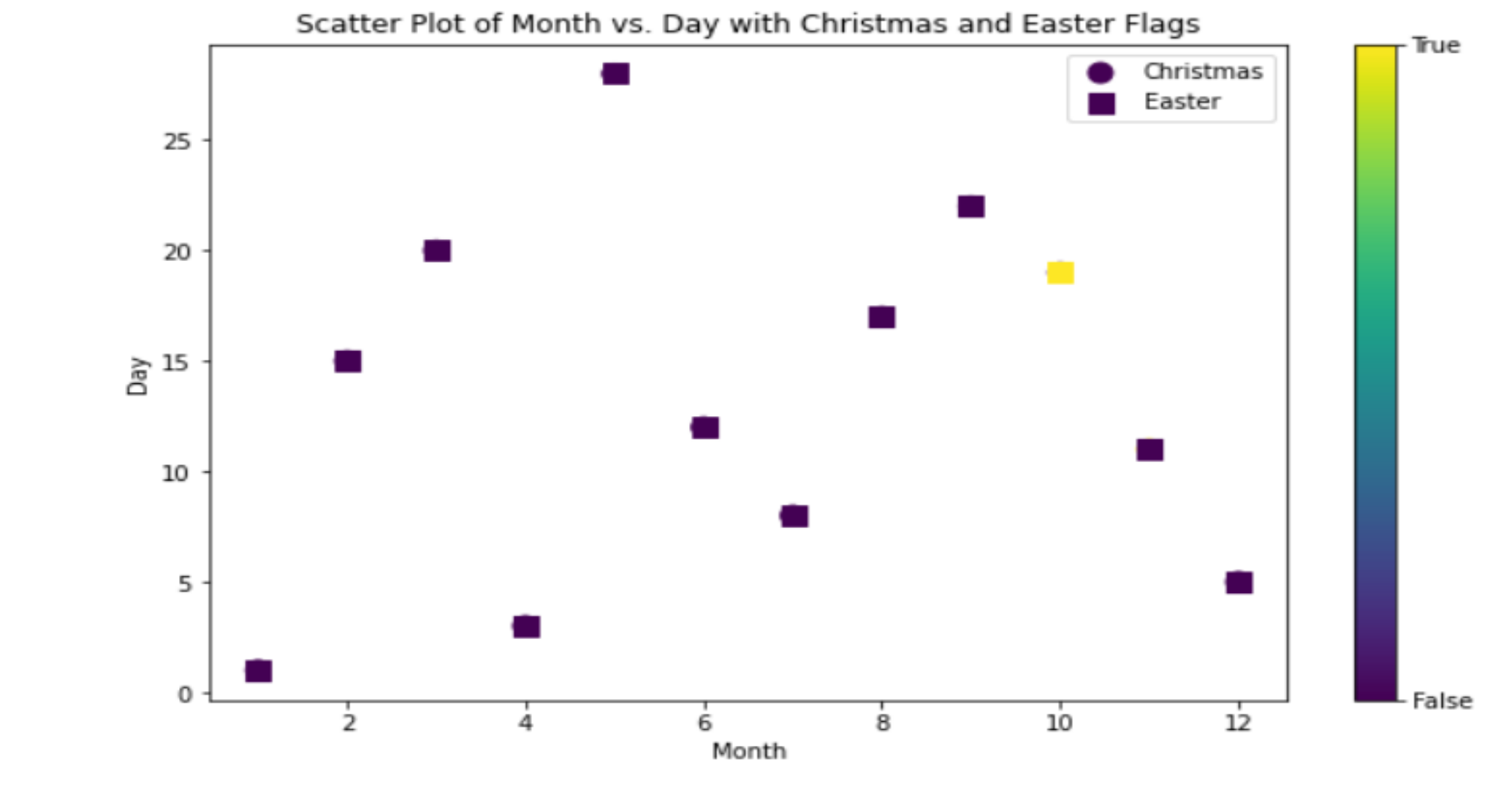


Seasonal (Christmas, Easter etc.) purchase behaviors

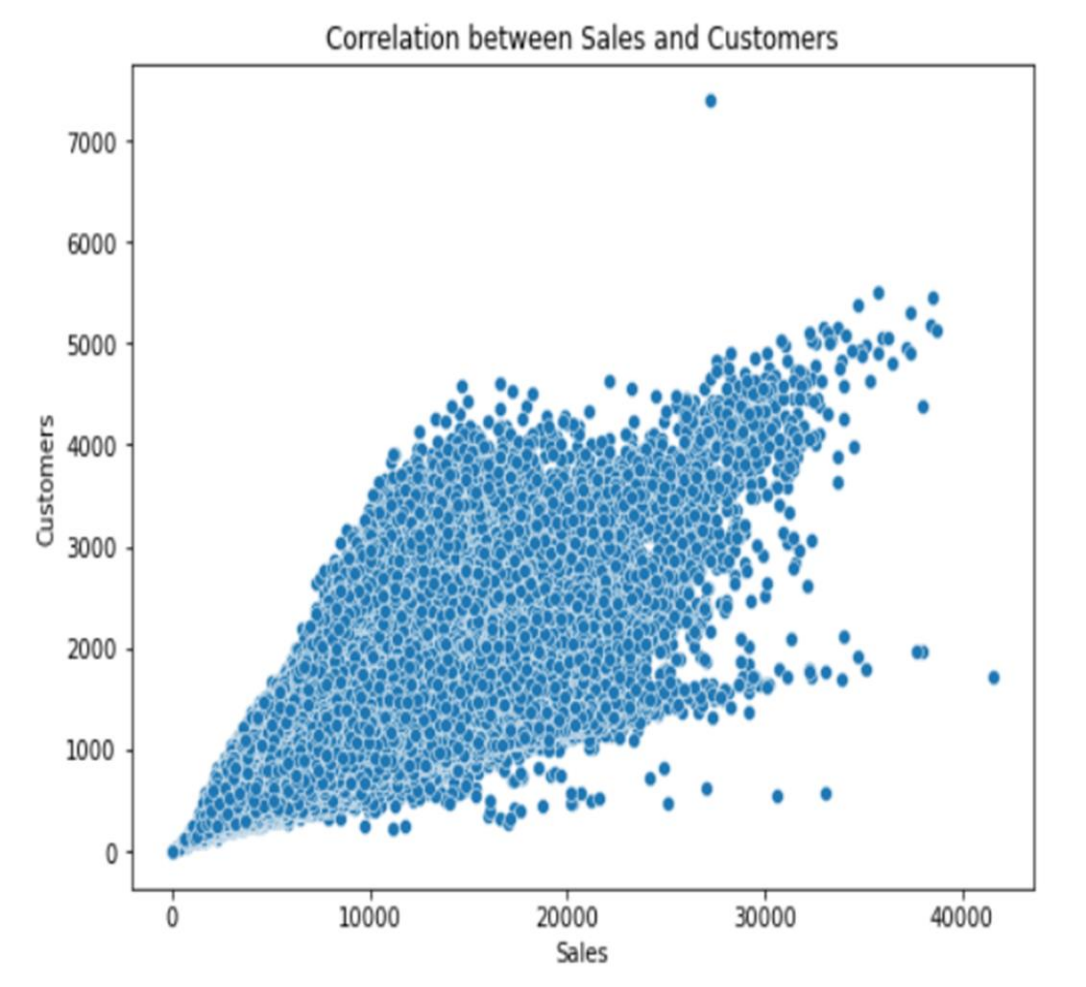
Pharmaceutical Sales Over Time



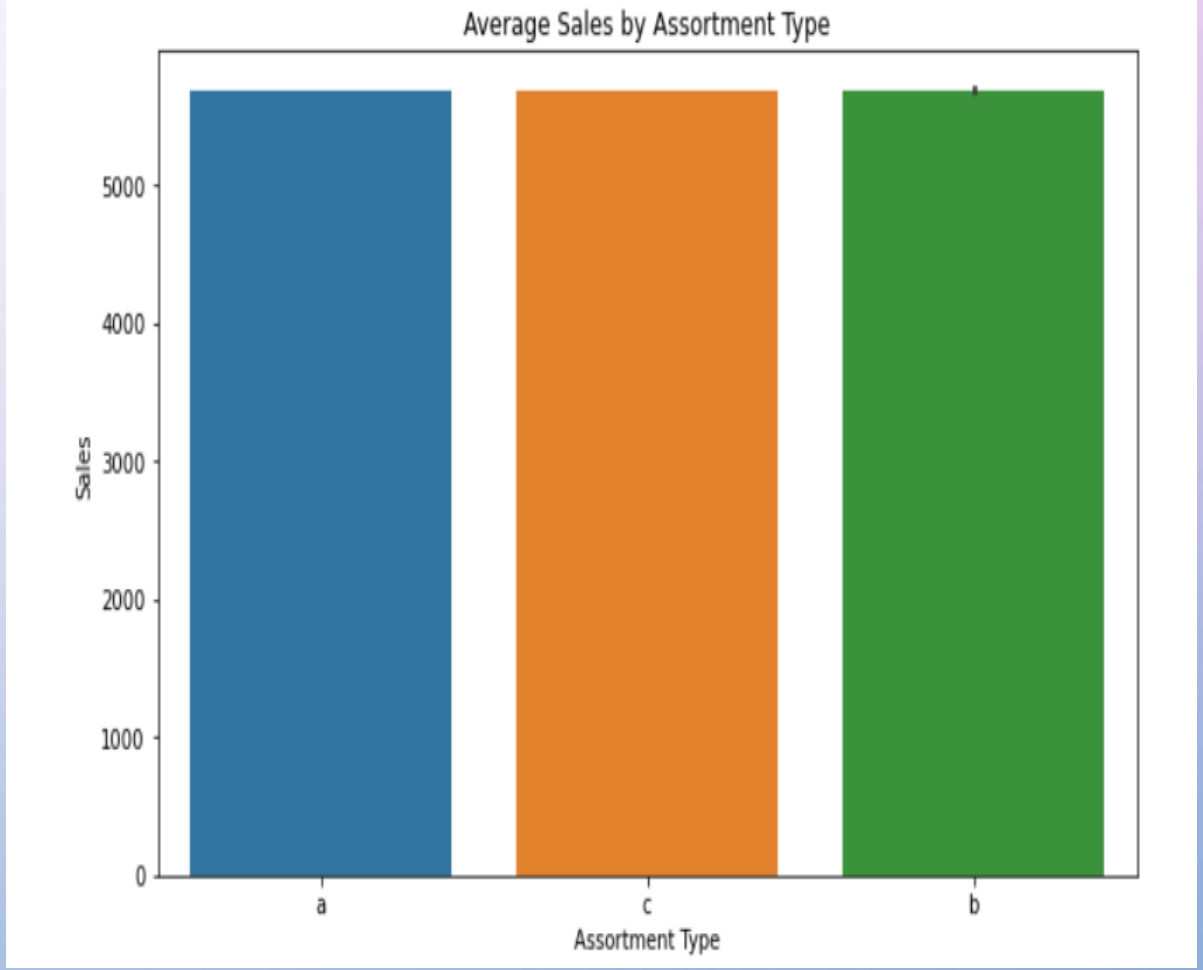
Scatter Plot of Month vs. Day with Christmas and Easter Flags



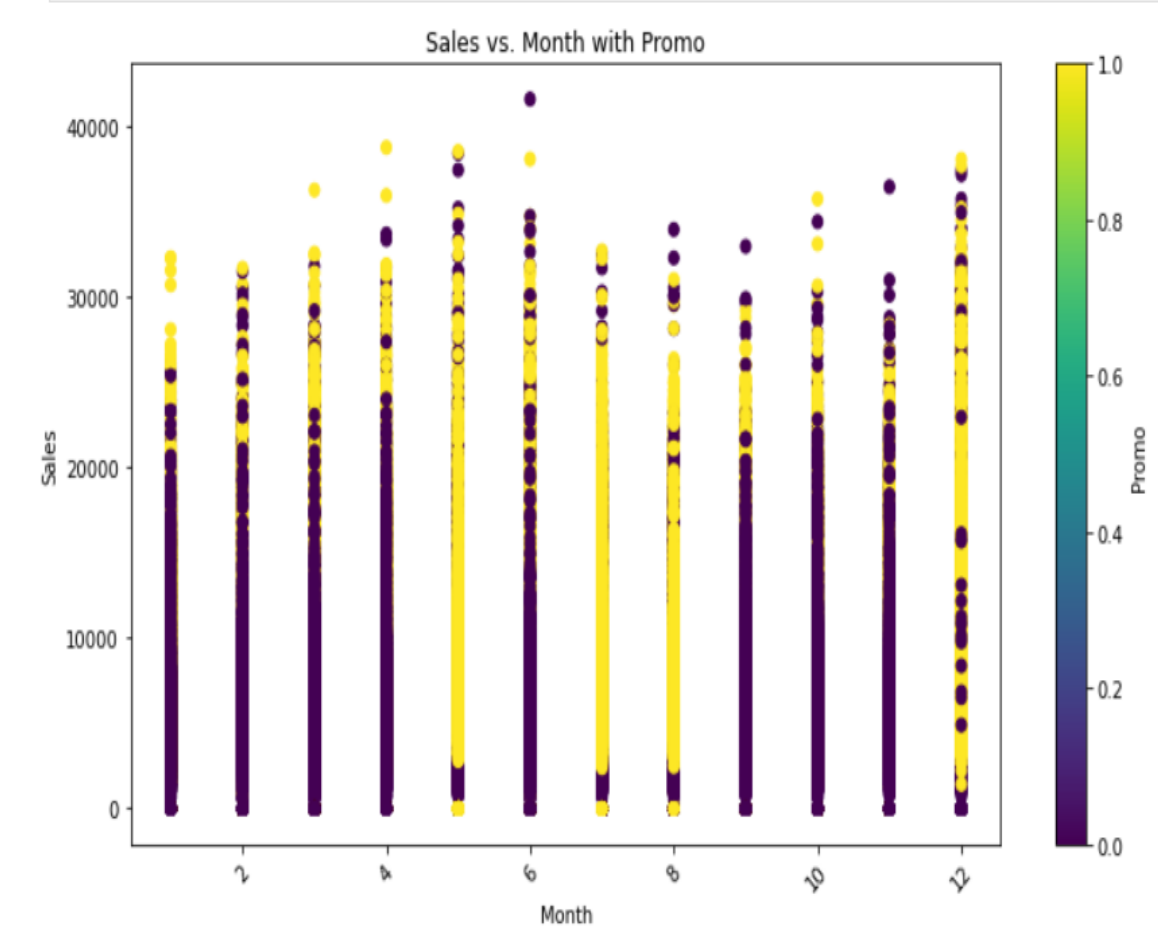
Correlation between Sales and Customers



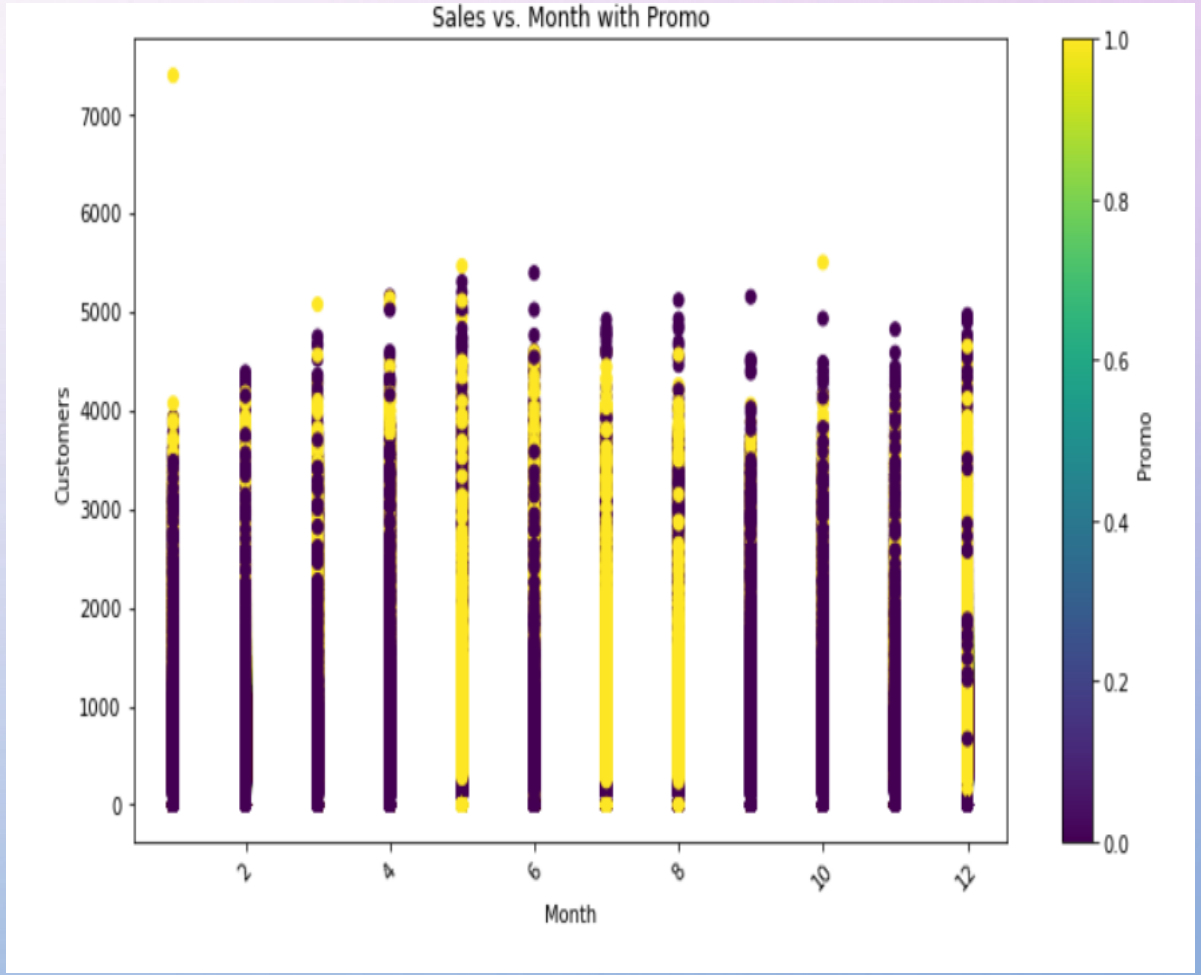
Average Sales by Assortment Type



Sales vs. Month with Promo



Month vs. Customer with Promo

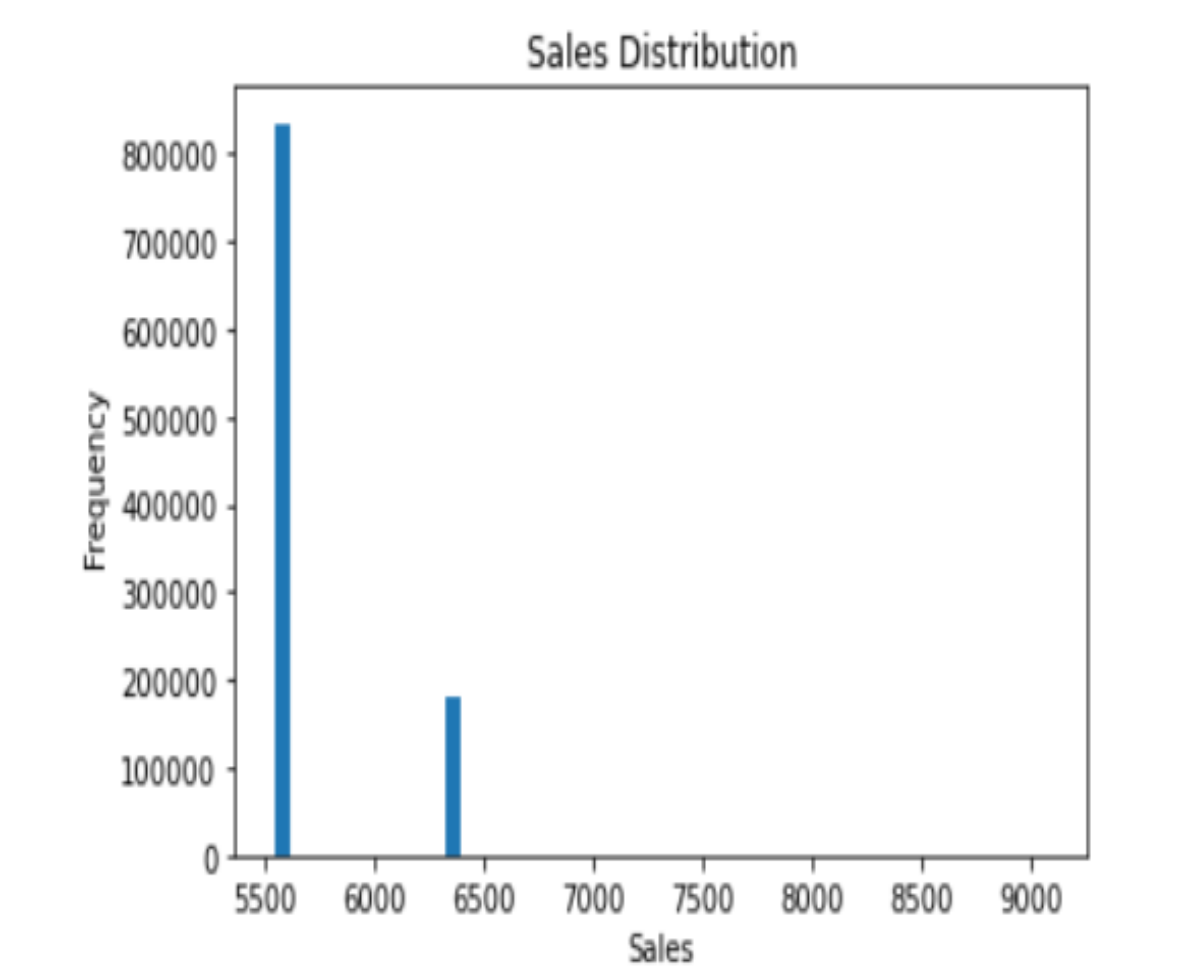


Trends of Customer Behavior during Store Open and Closing Times on Different Days of the Week

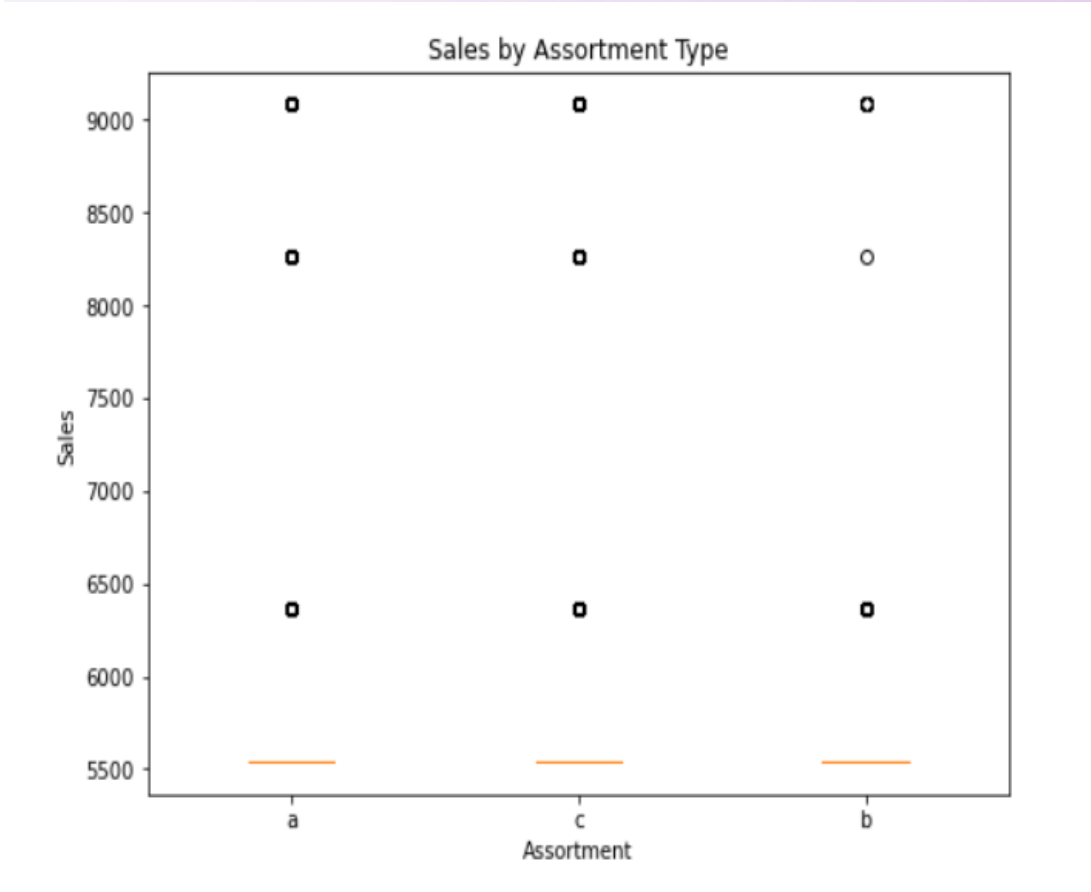
Average Sales on Weekends vs. Weekdays for Stores Open on All Weekdays

Check how the assortment type affects sales

Sales Distribution



Sales by Assortment Type



Preprocessing

it is important to process the data into a format where it can be fed to a machine learning model. This typically means converting all non-numeric columns to numeric, handling NaN values and generating new features from already existing features. In our case, you have a few datetime columns to preprocess. you can extract the following from them

1. Merge csv file test data and store data use this columns and merge data successfully.
2. This train_merge_data ,csv file use and merge useful columns .

```
train_merged_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1017209 entries, 0 to 1017208
Data columns (total 27 columns):
 #    Column                     Non-Null Count      Dtype
---   ------                     --------------      -----
 0    Store                      1017209 non-null    int64
 1    DayOfWeek                  1017209 non-null    int64
 2    Date                       1017209 non-null    datetime64[ns]
 3    Sales                      1017209 non-null    int64
 4    Customers                  1017209 non-null    int64
 5    Open                       1017209 non-null    int64
 6    Promo                      1017209 non-null    int64
 7    StateHoliday               1017209 non-null    int64
 8    SchoolHoliday              1017209 non-null    int64
 9    StoreType                  1017209 non-null    int64
 10   Assortment                 1017209 non-null    int64
 11   CompetitionDistance        1014567 non-null    float64
 12   CompetitionOpenSinceMonth  693861 non-null     float64
 13   CompetitionOpenSinceYear   693861 non-null     float64
 14   Promo2                     1017209 non-null    int64
 15   Promo2SinceWeek            509178 non-null     float64
 16   Promo2SinceYear            509178 non-null     float64
 17   PromoInterval              1017209 non-null    int64
 18   Year                       1017209 non-null    int64
 19   Month                      1017209 non-null    int64
 20   weekDay                    1017209 non-null    int64
 21   IsWeekday                  1017209 non-null    int64
 22   Quarter                    1017209 non-null    int64
 23   SalesPerCustomer           844340 non-null     float64
 24   IsMonthStart               1017209 non-null    int64
 25   IsMonthMiddle              1017209 non-null    int64
 26   IsMonthEnd                 1017209 non-null    int64
dtypes: datetime64[ns](1), float64(6), int64(20)
memory usage: 217.3 MB
```

Building models with sklearn pipelines

```
from sklearn.impute import SimpleImputer
imputer = SimpleImputer(strategy='mean')
regressor = LinearRegression()
```

```
pipeline = Pipeline([
    ('imputer', StandardScaler()),  # Feature scaling (optional)
    ('refressor', regressor)])
```

```
# Fit the pipeline to the training data
pipeline.fit(X_train, y_train)
```

```
Pipeline(steps=[('imputer', StandardScaler()),
                ('refressor', LinearRegression())])
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

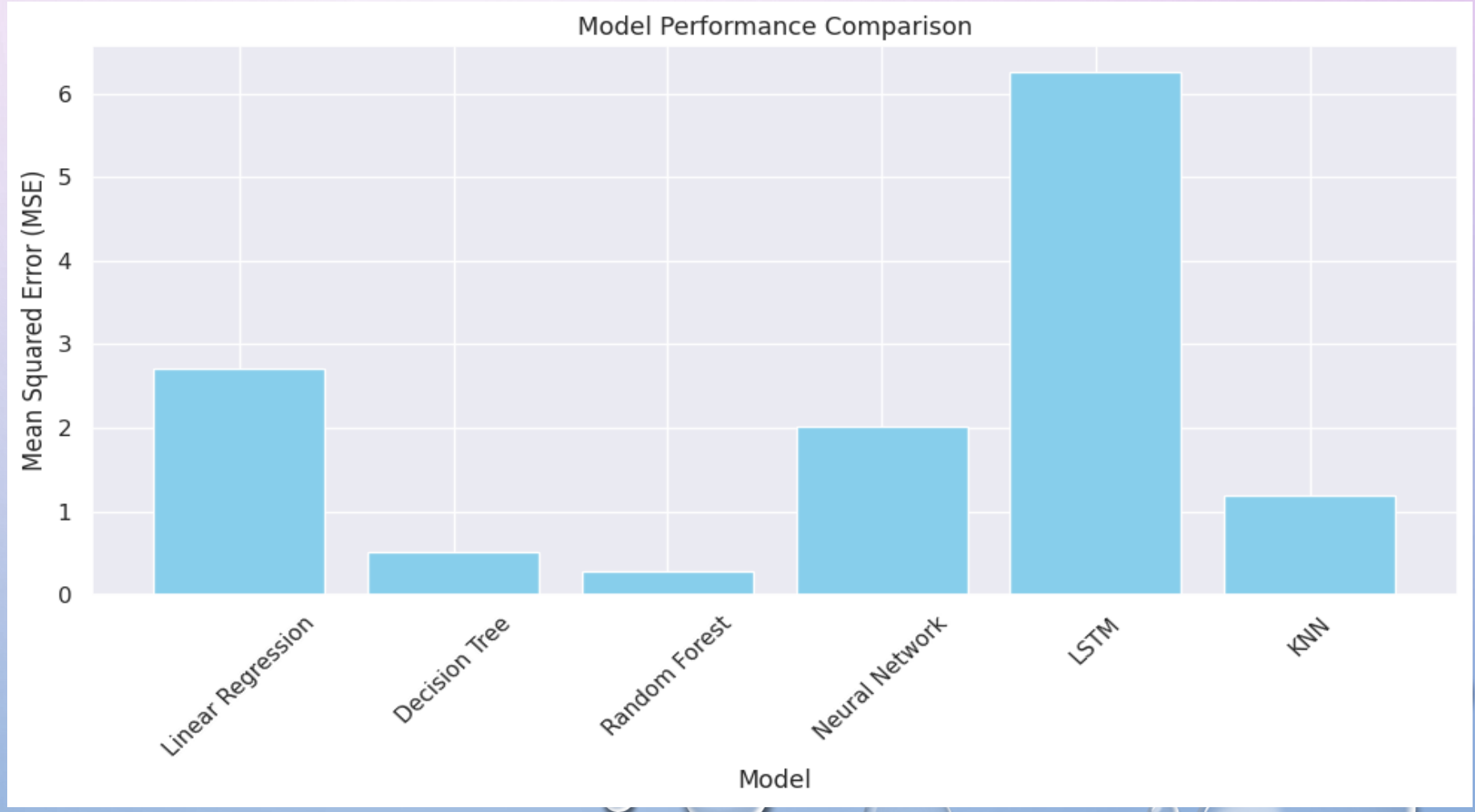
```
# Make predictions on the test data
y_pred = pipeline.predict(X_test)
```

```
mse = mean_squared_error(y_test, y_pred)
print(f'Mean Squared Error: {mse}')
```

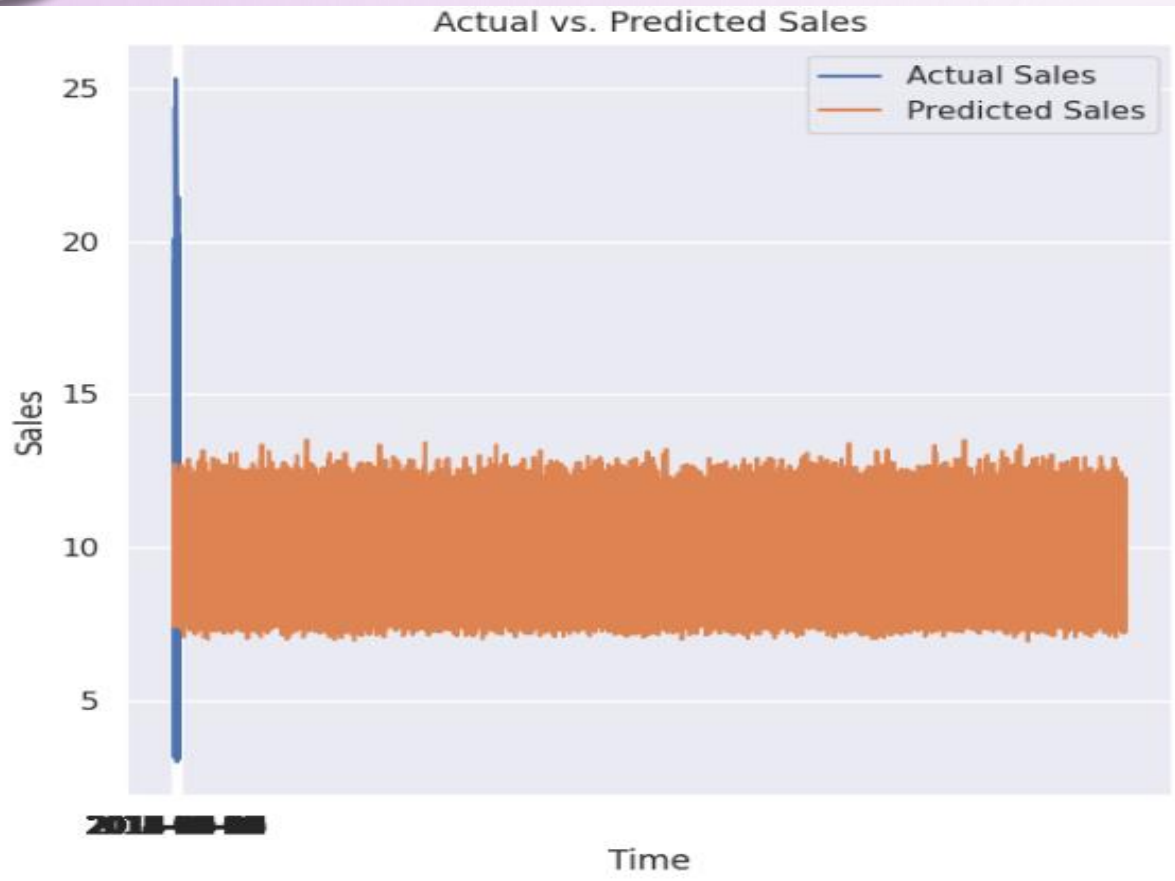
```
Mean Squared Error: 2.7168604478654887
```

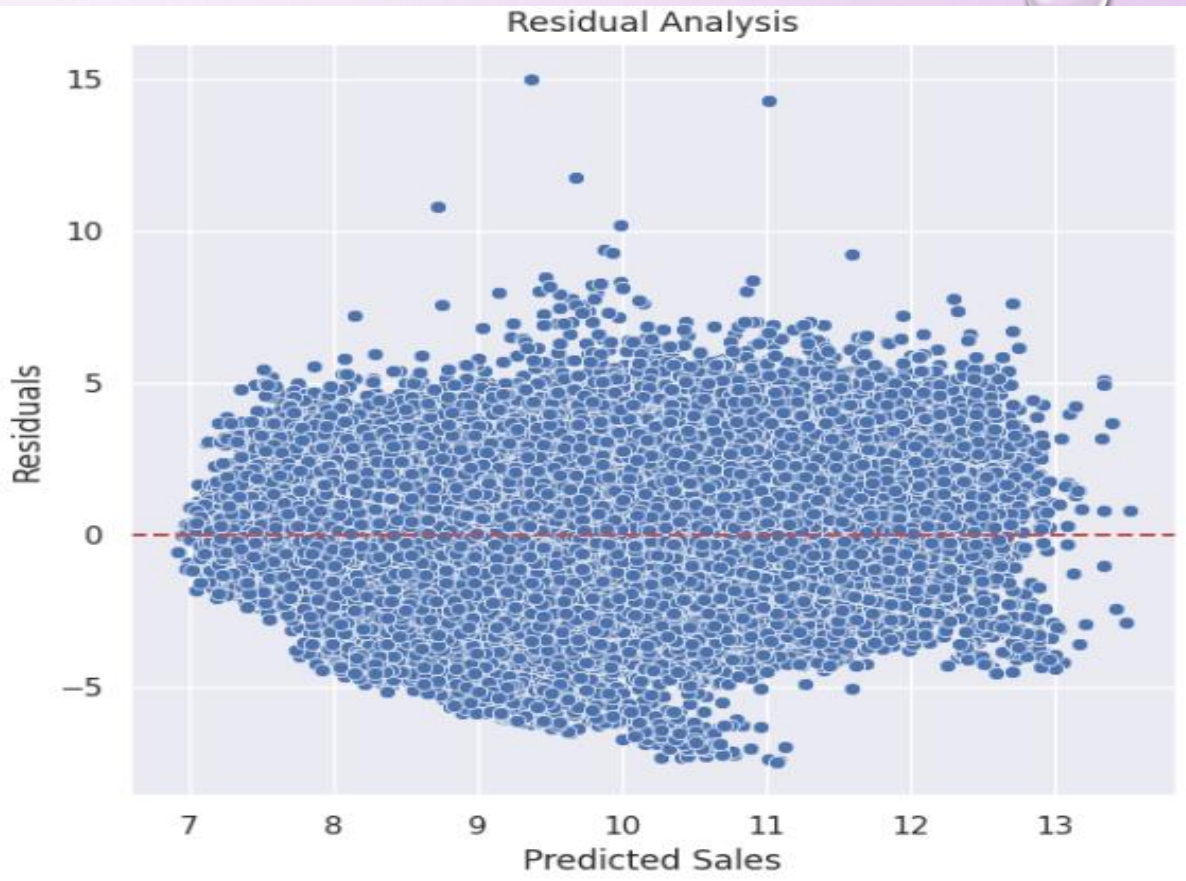
Models

	Model	MSE
0	Linear Regression	2.701617
1	Decision Tree	0.510121
2	Random Forest	0.282171
3	Neural Network	2.023631
4	LSTM	6.254384
5	KNN	1.189246



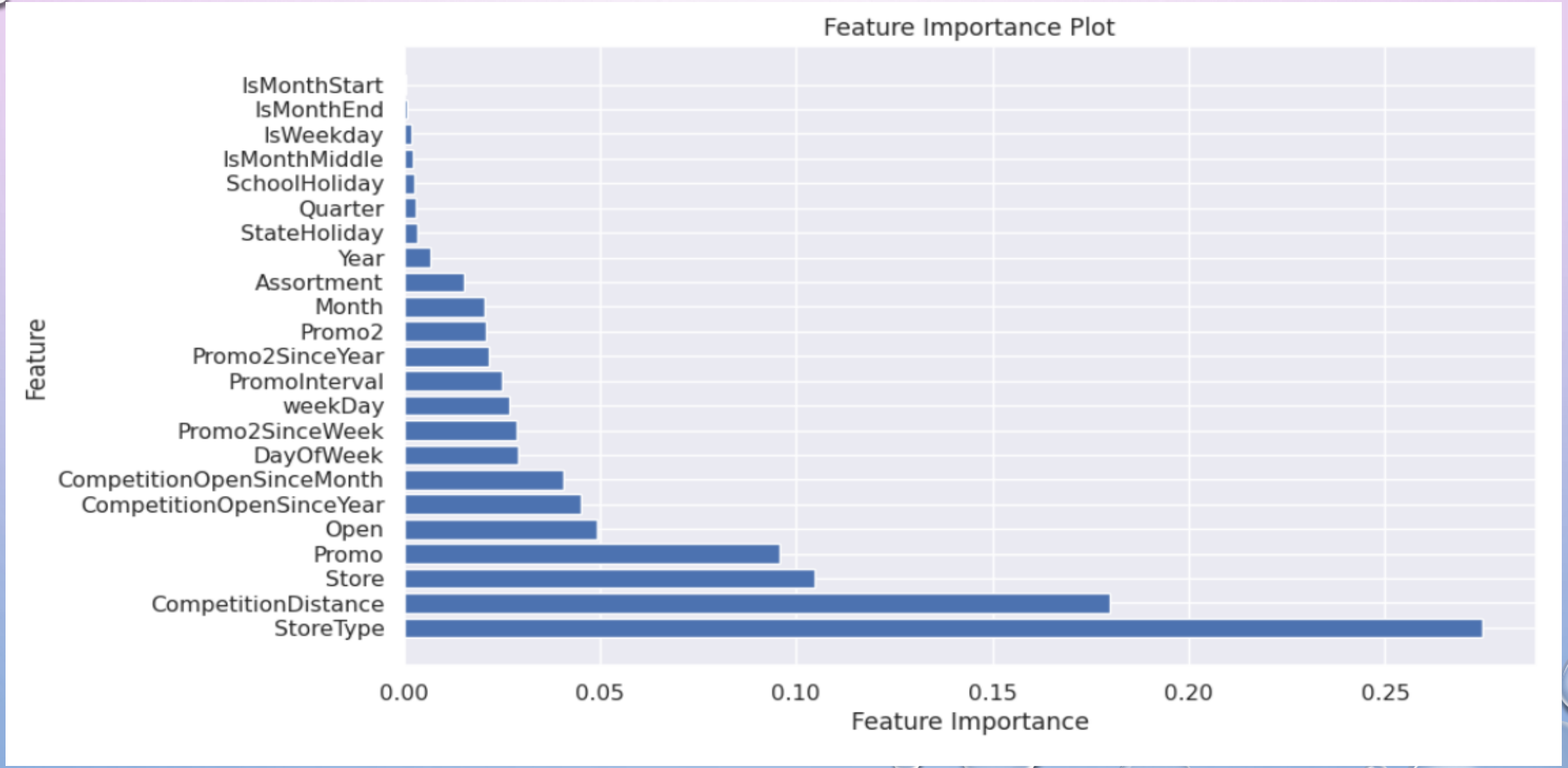
Post Prediction analysis



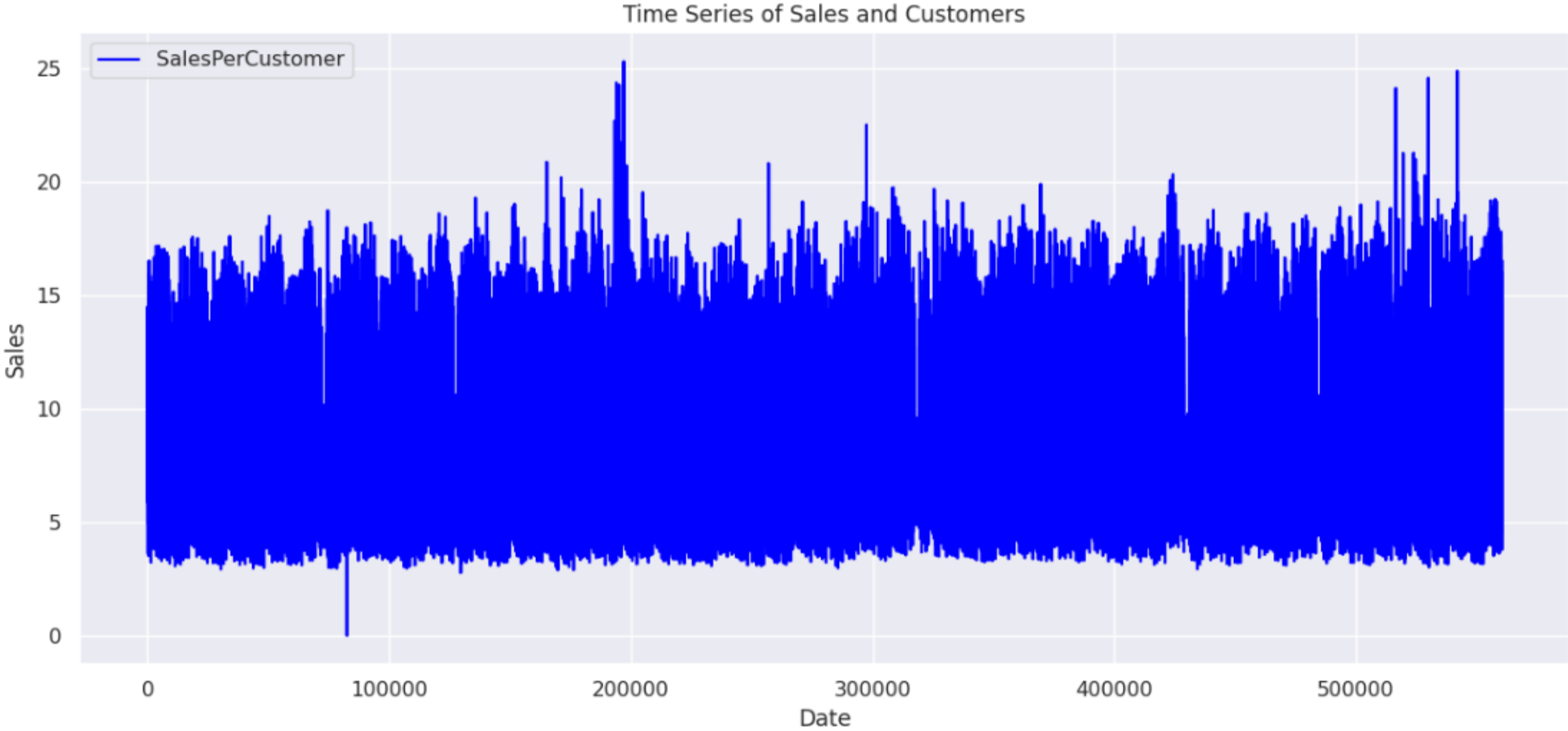



```
MAE for Store Type '0': 1.2874098159015392
MAE for Store Type '1': 1.2874098159015392
MAE for Store Type '2': 1.2874098159015392
MAE for Store Type '3': 1.2874098159015392
```

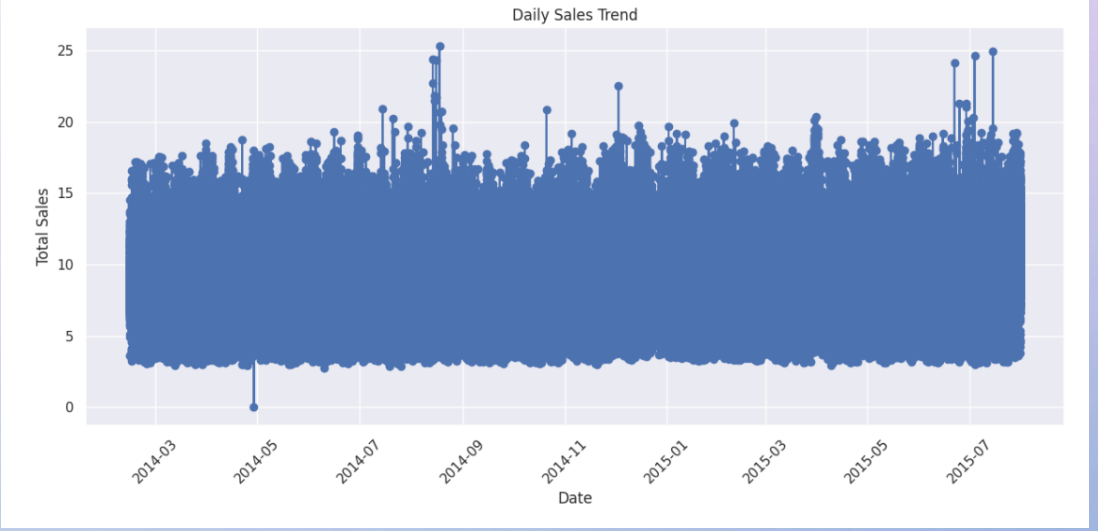
Feature Importance Plot



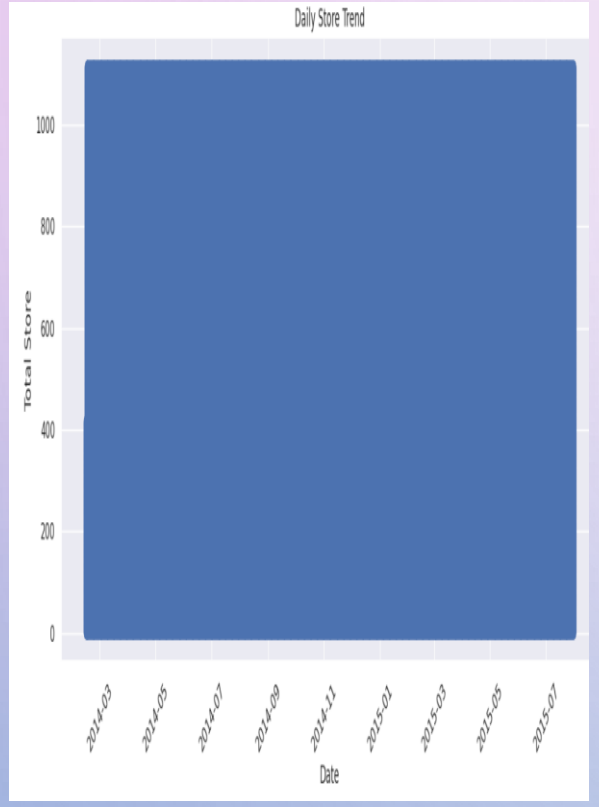
TIME SERIES ANALYSIS

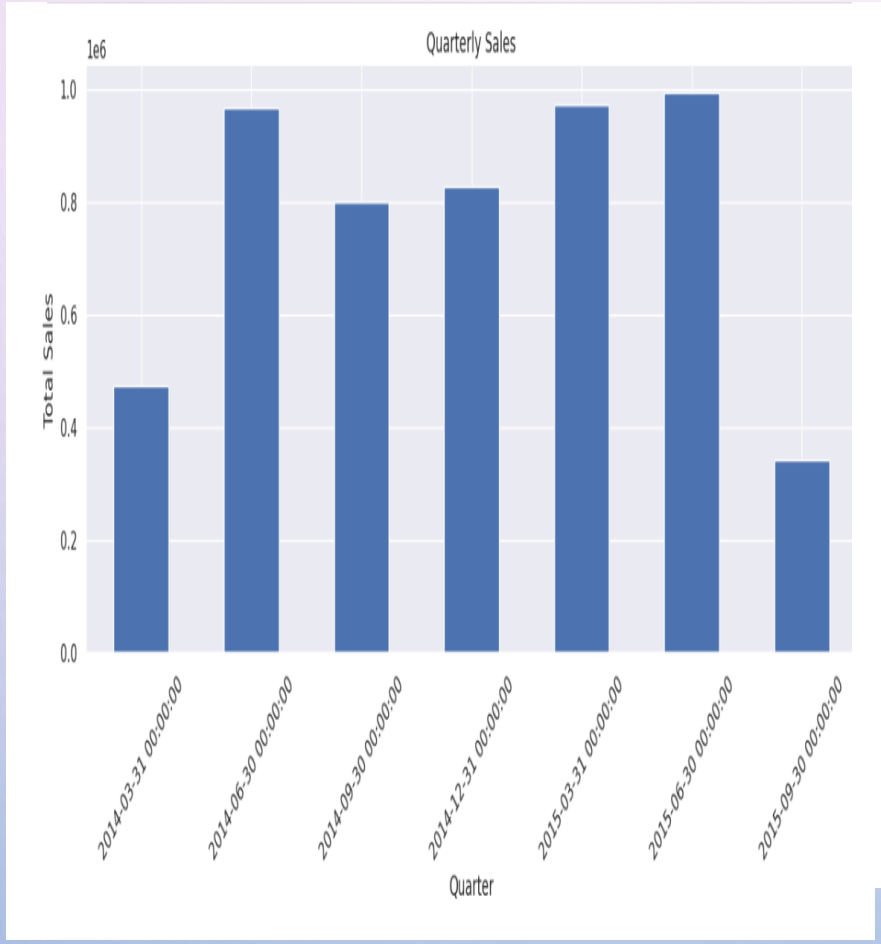


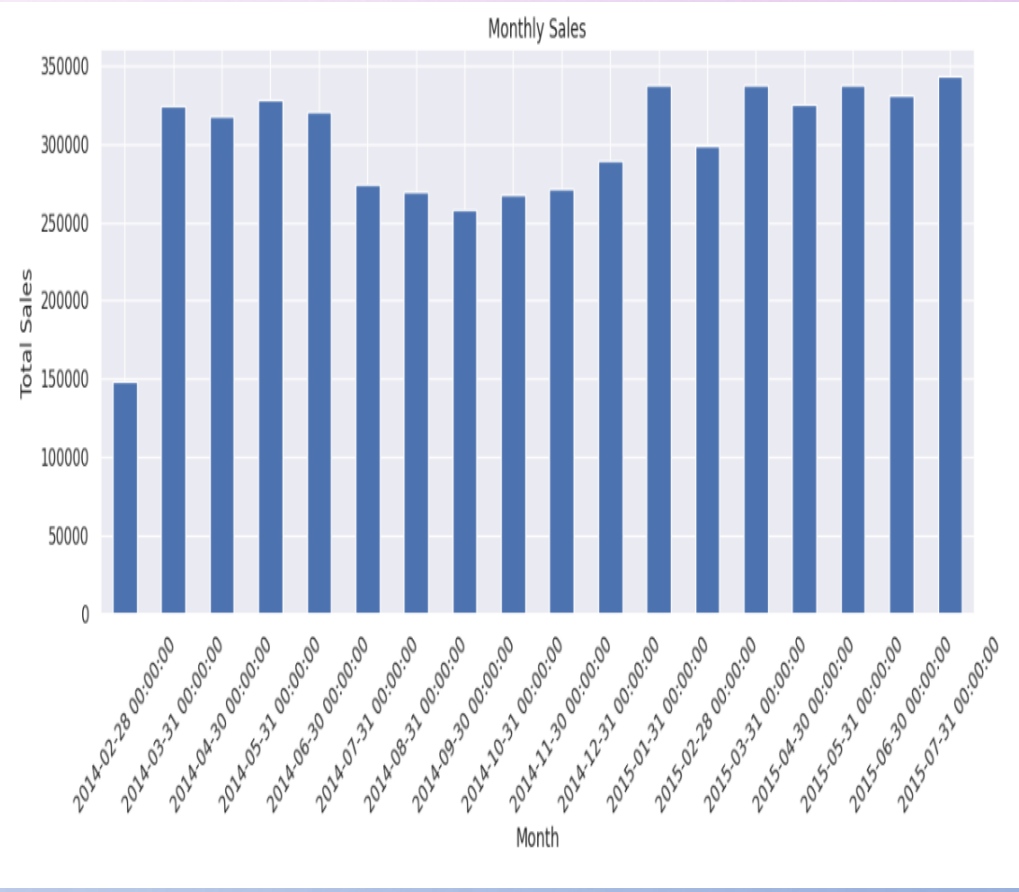

Time Series Plot of Sales
StateHoliday
Sales
Date
0
1
2
3
4
5
0
100000
200000
300000
400000
500000


Daily Sales Trend
Total Sales
Date
0
5
10
15
20
25
2014-03
2014-05
2014-07
2014-09
2014-11
2015-01
2015-03
2015-05
2015-07

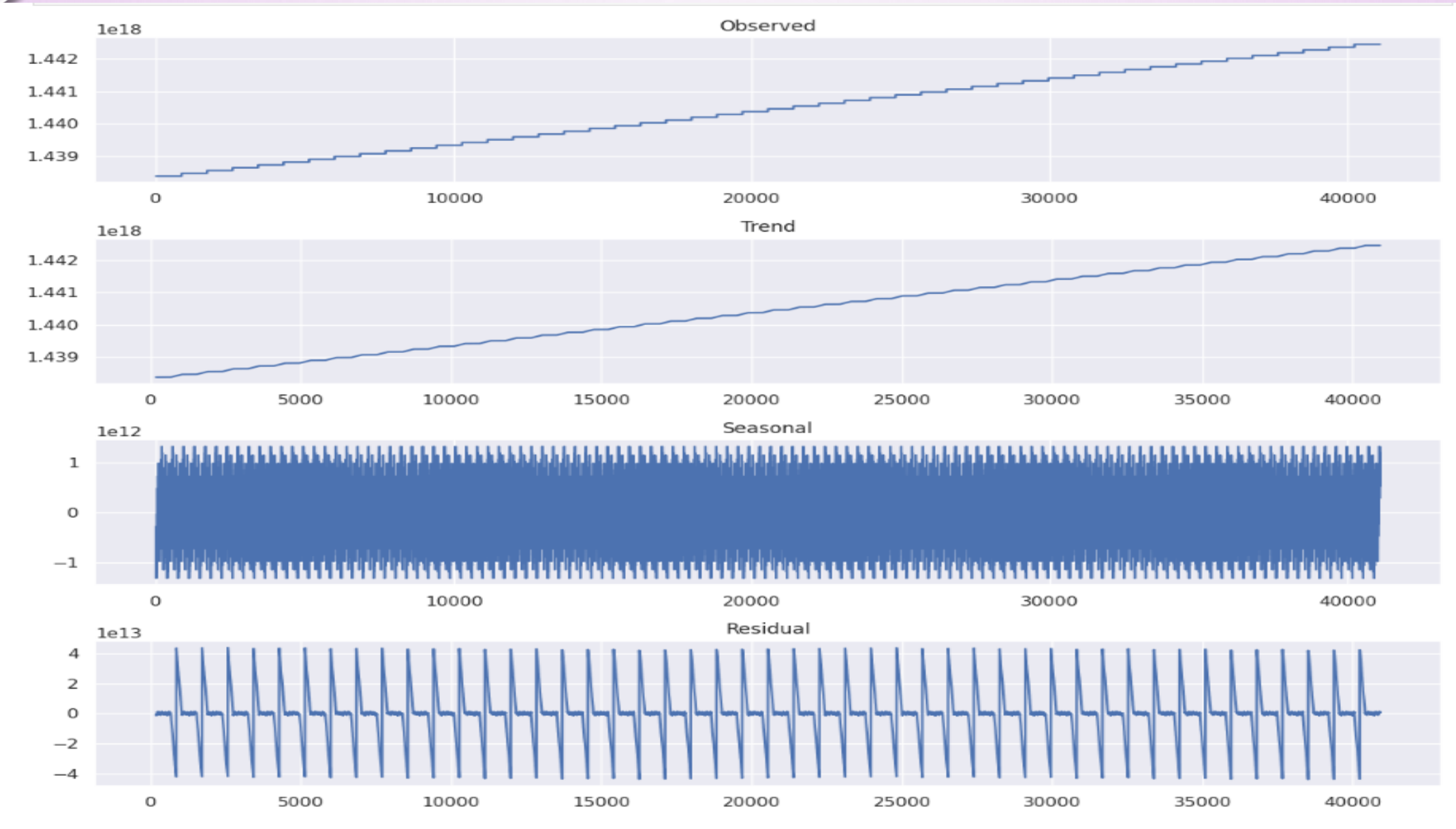
TREND ANALYSIS





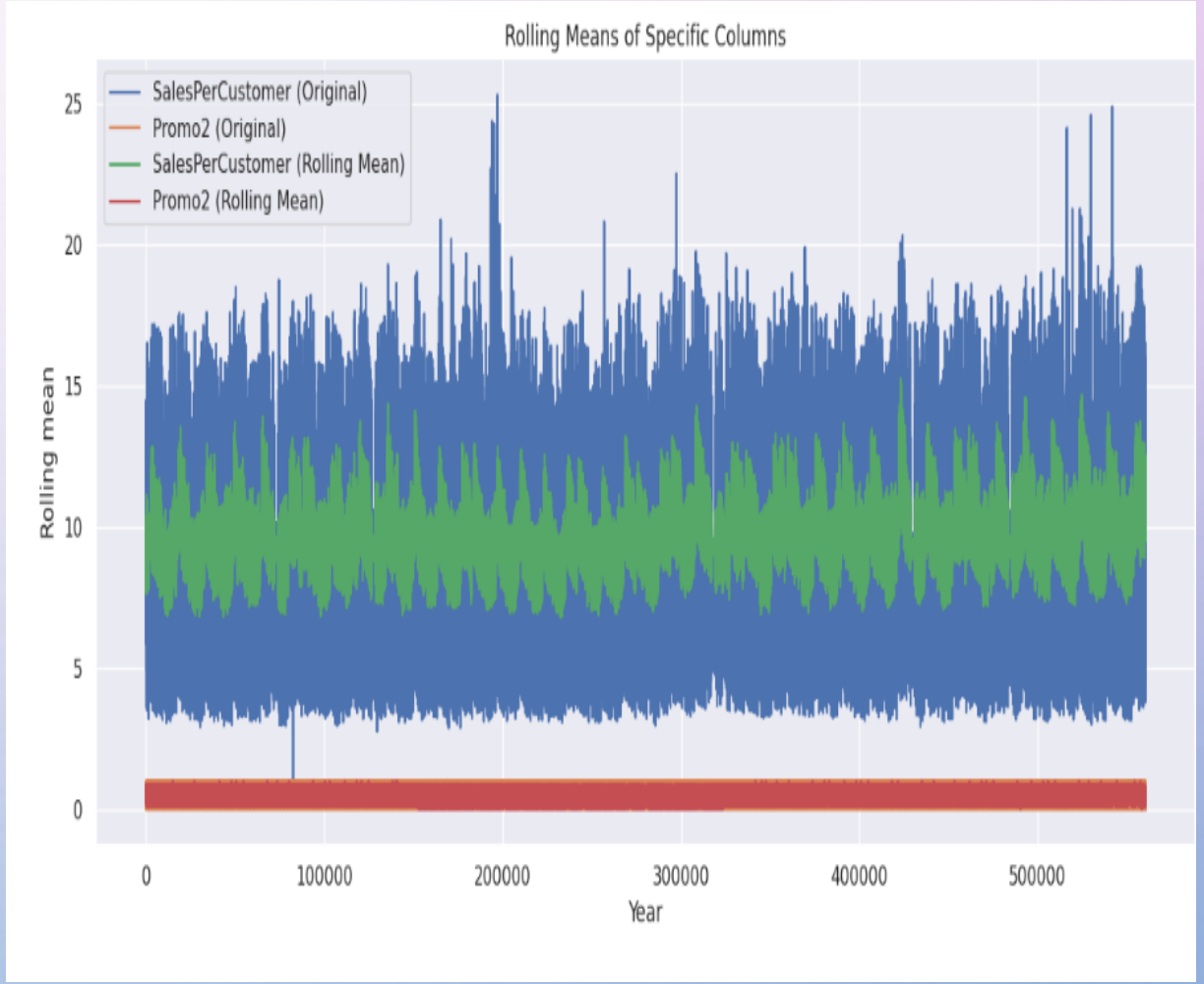


Trend Analysis

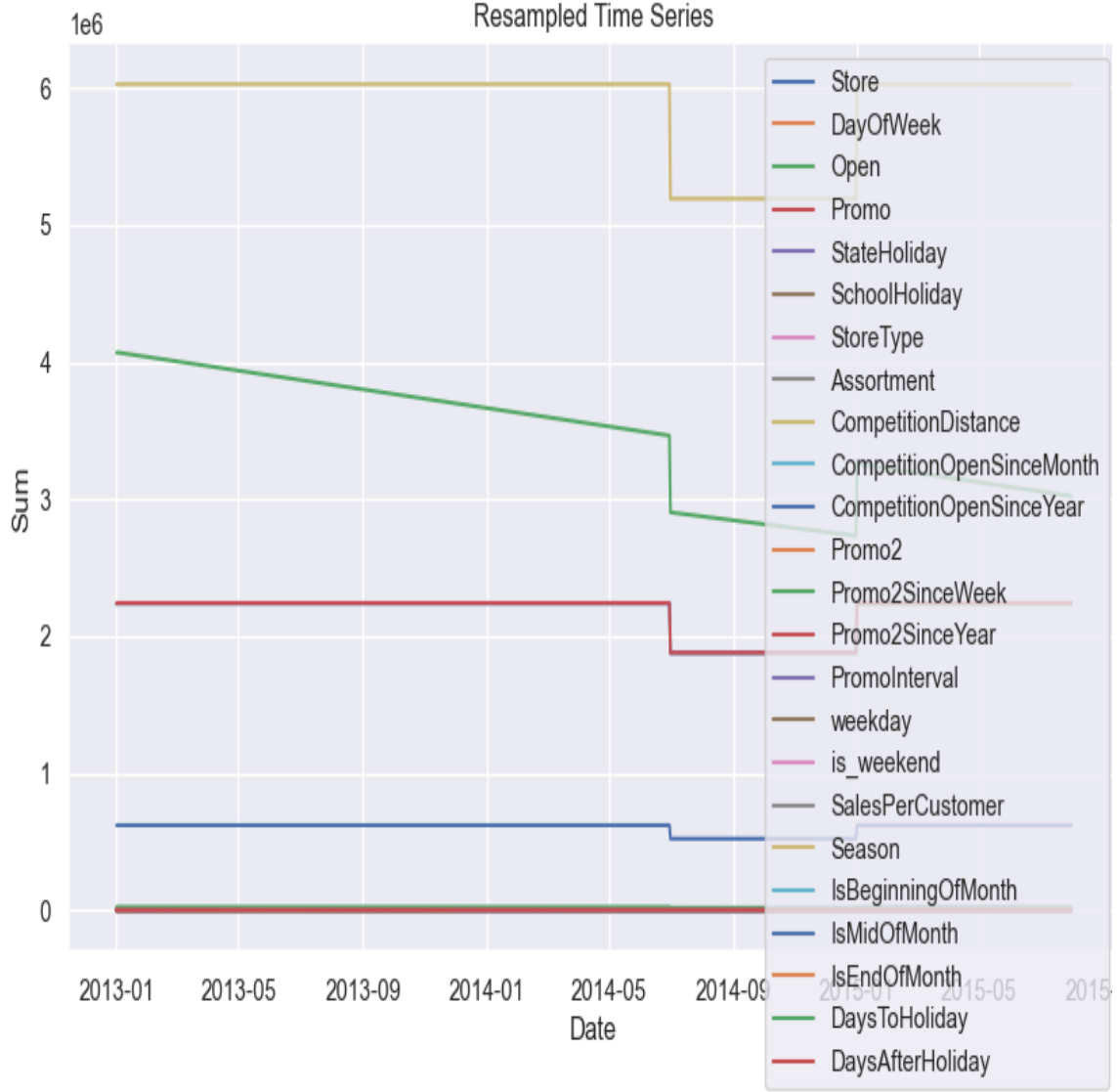


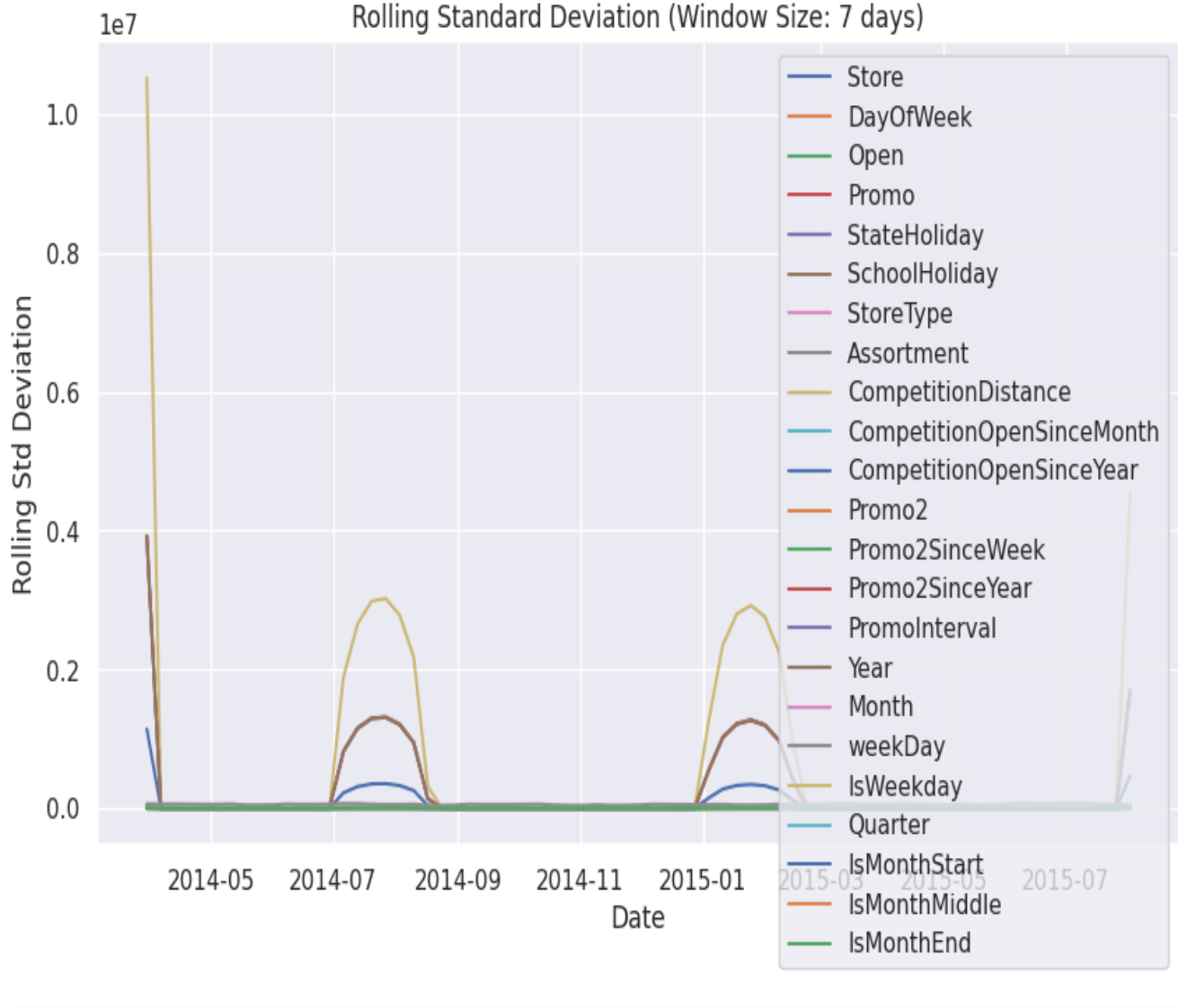
ROLLING MEANS OF SPECIFIC COLUMNS

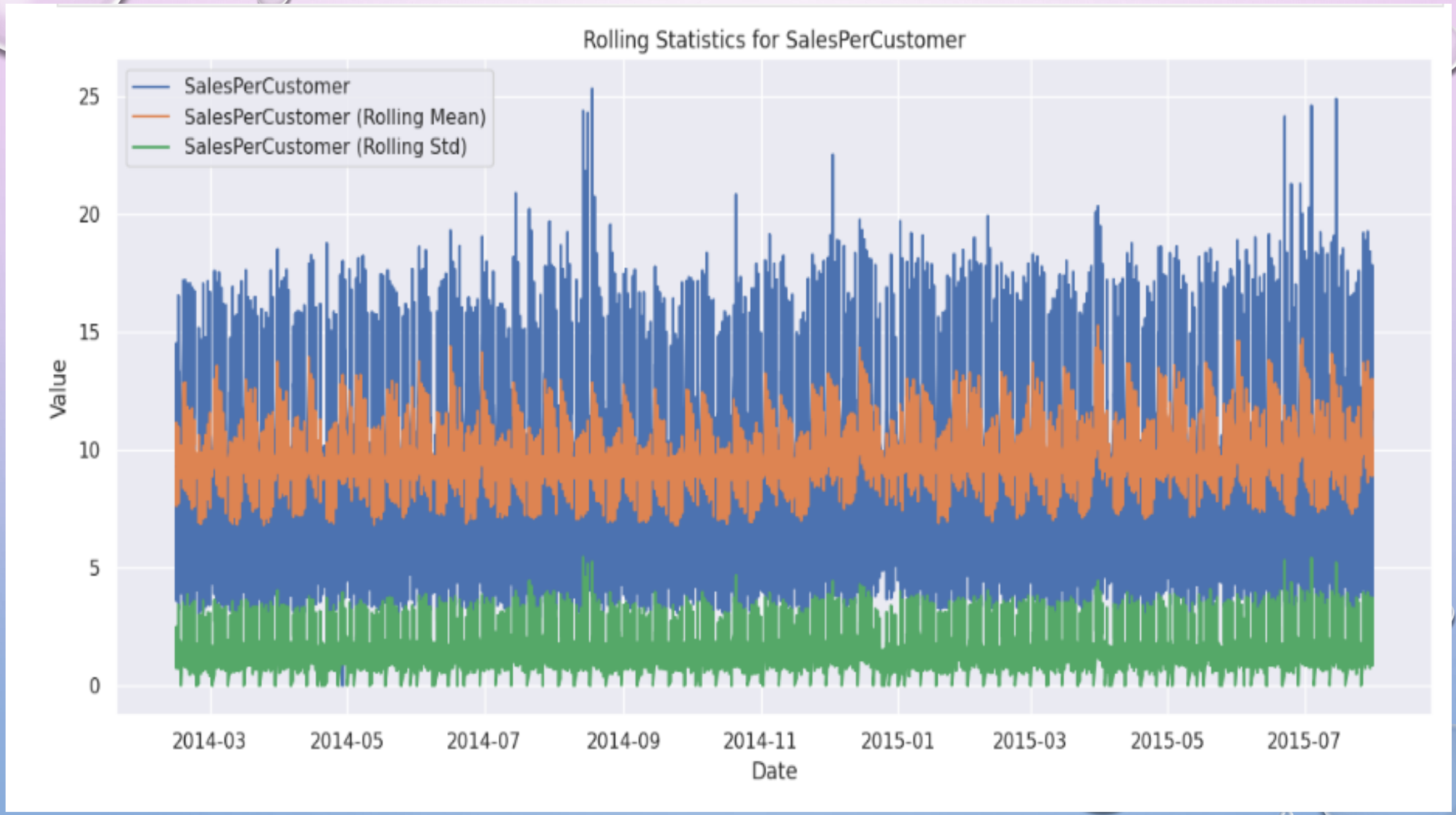
- Analyzing the rolling mean of sales within the range of 7-14 while considering promotional activity levels at a constant range of 0 to 1 can provide insights into how promotions may impact sales trends over a medium-term period.
- This analysis helps assess whether the presence or absence of promotions within this range is associated with specific patterns or fluctuations in sales behavior.



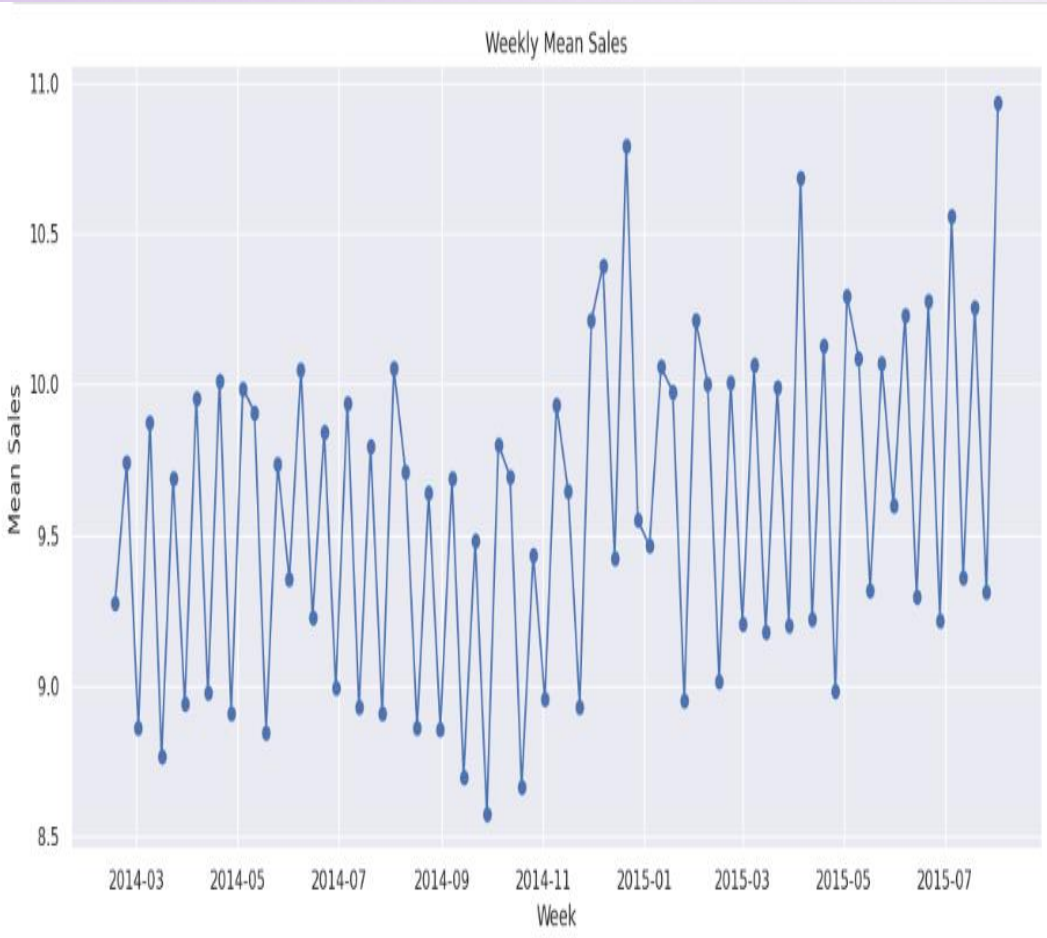
Both in Resampled Time series and Rolling Standard Deviation Plot we can see there is high fluctuation in the mid-year 2014

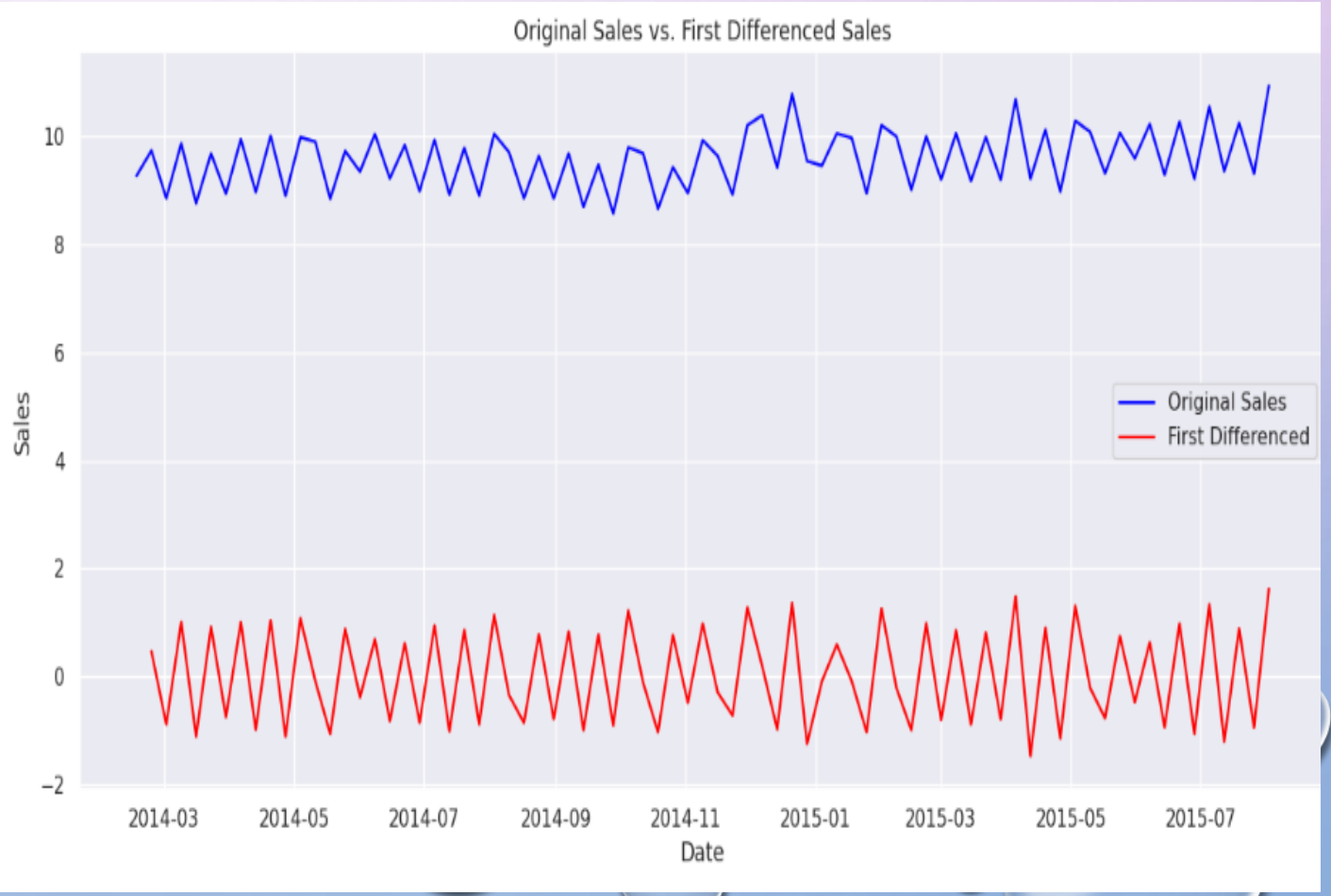




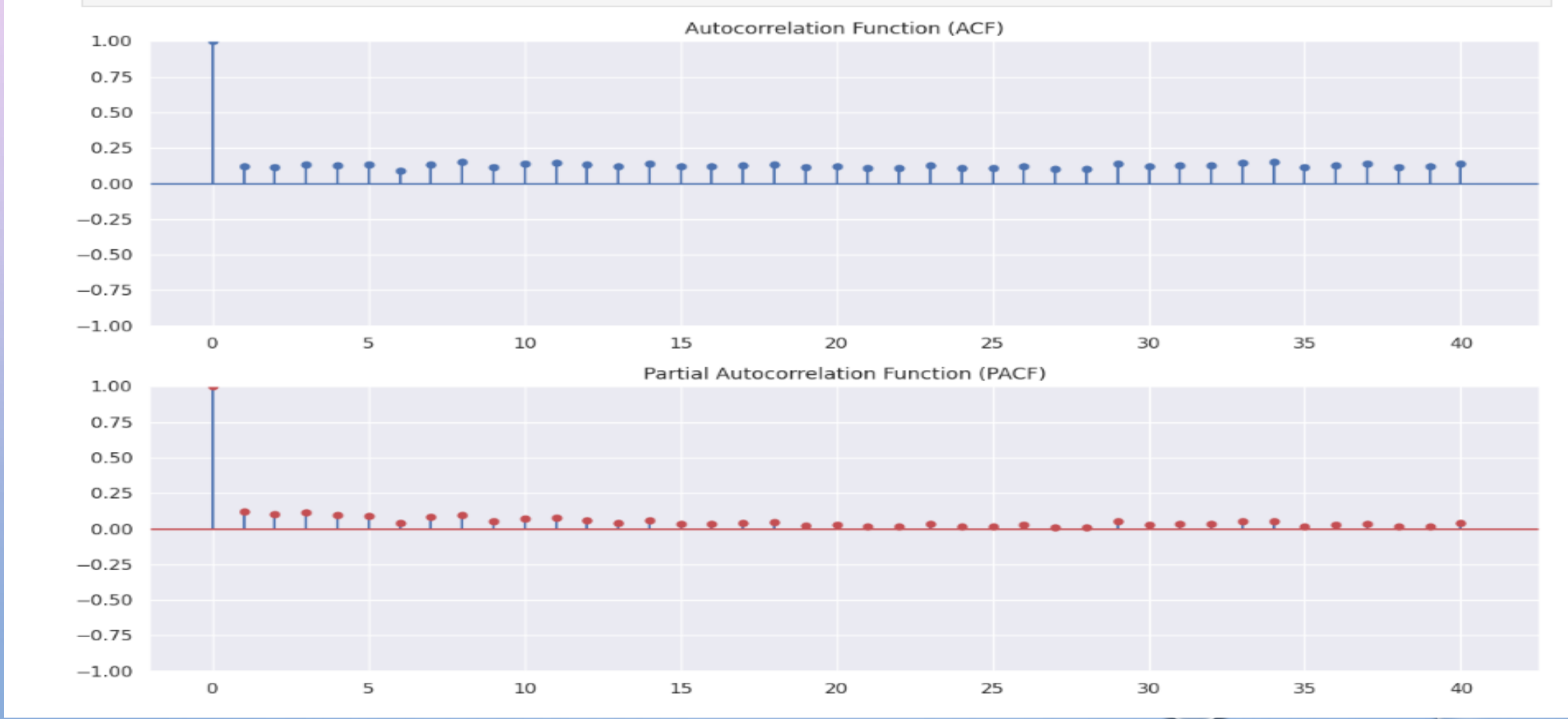

Rolling Statistics for SalesPerCustomer
SalesPerCustomer
SalesPerCustomer (Rolling Mean)
SalesPerCustomer (Rolling Std)
Value
0
5
10
15
20
25
2014-03
2014-05
2014-07
2014-09
2014-11
2015-01
2015-03
2015-05
2015-07
Date

- First Differenced Sales represents the sales data after applying a differencing operation, specifically taking the difference between each data point and the previous one.
- The differencing process have reduced the overall variability or patterns in the data, making it appear more stable or stationary.



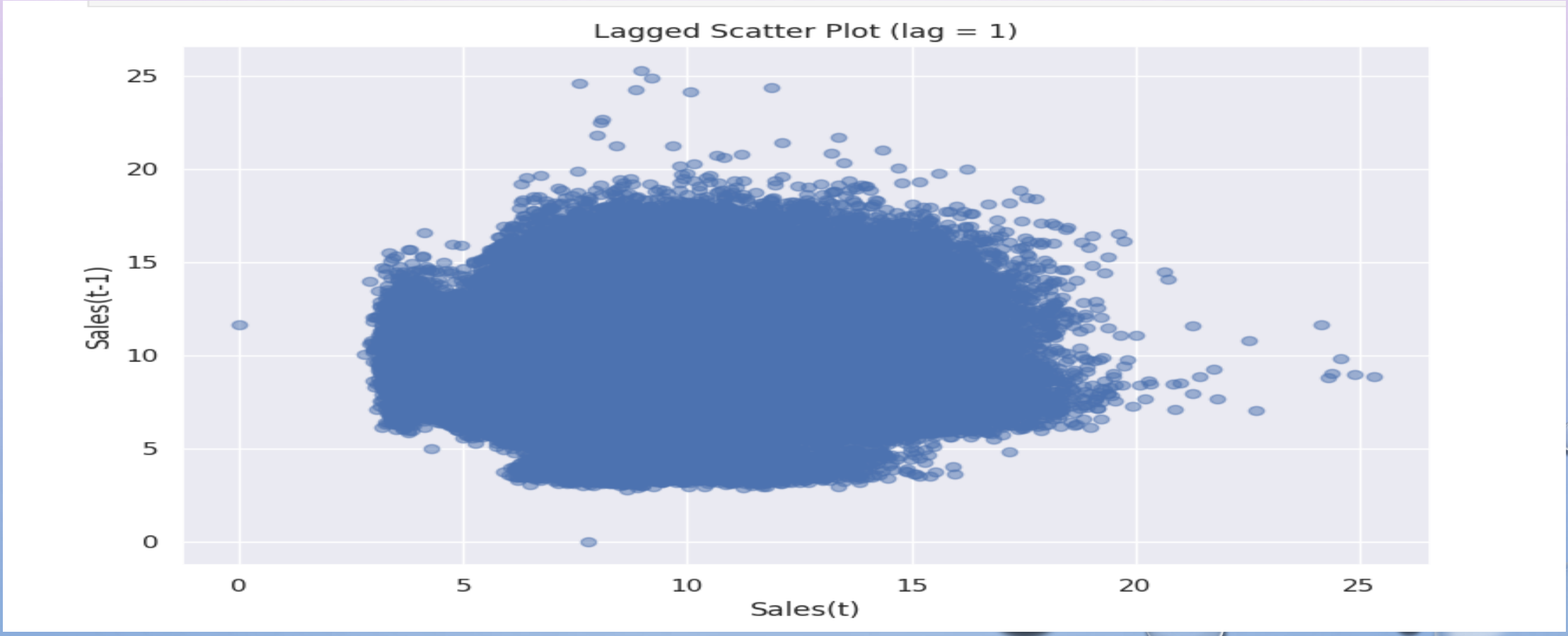


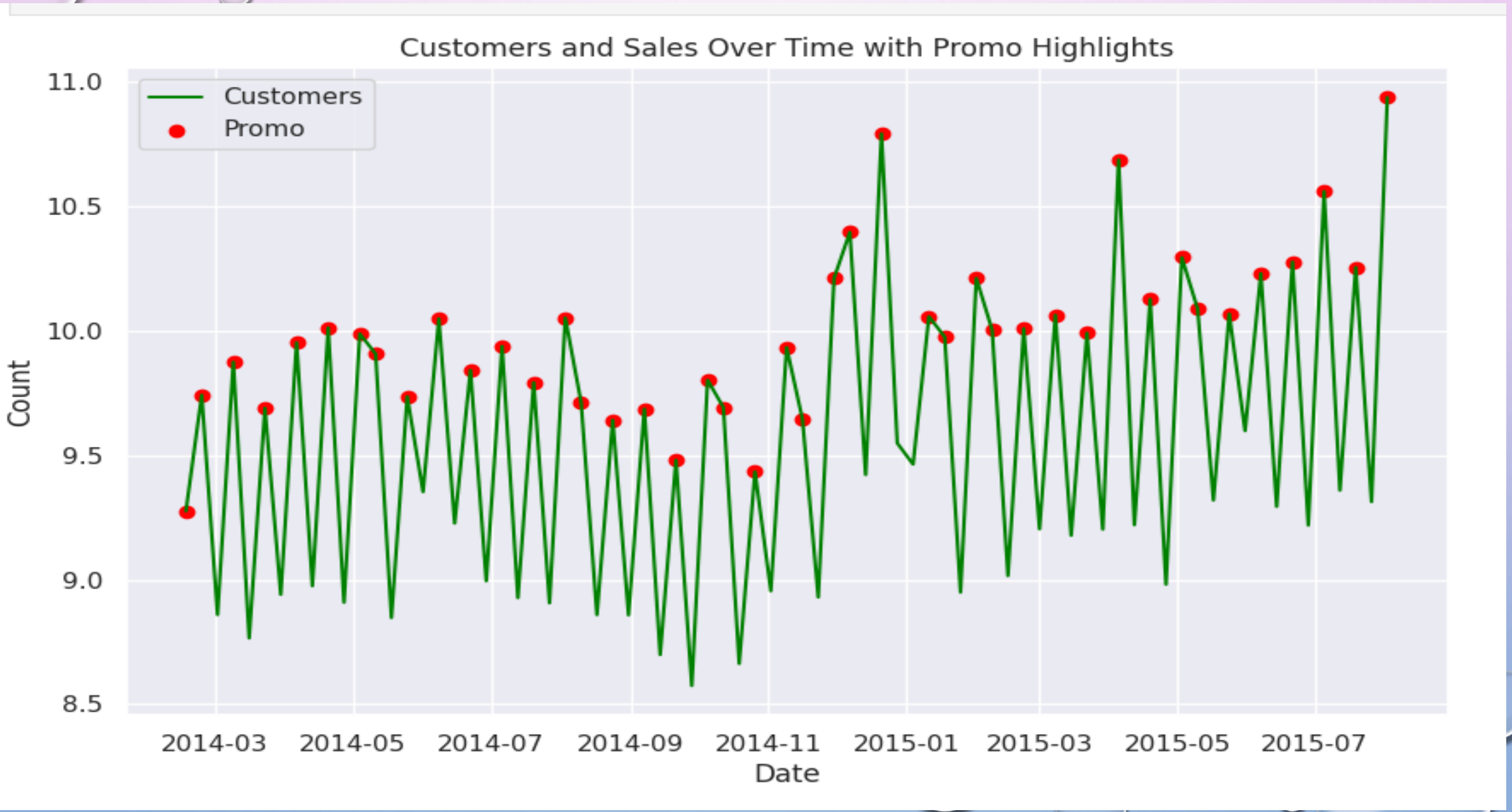
- A fluctuating ACF pattern with both falling and rising values typically signifies seasonality within the data. It suggests regular, repeating cycles at specific time intervals.
- On the other hand, when PACF values are notably lower than ACF values, it indicates that the direct correlation between the current observation and its past lags is relatively weak compared to the influence of other lags or variables.

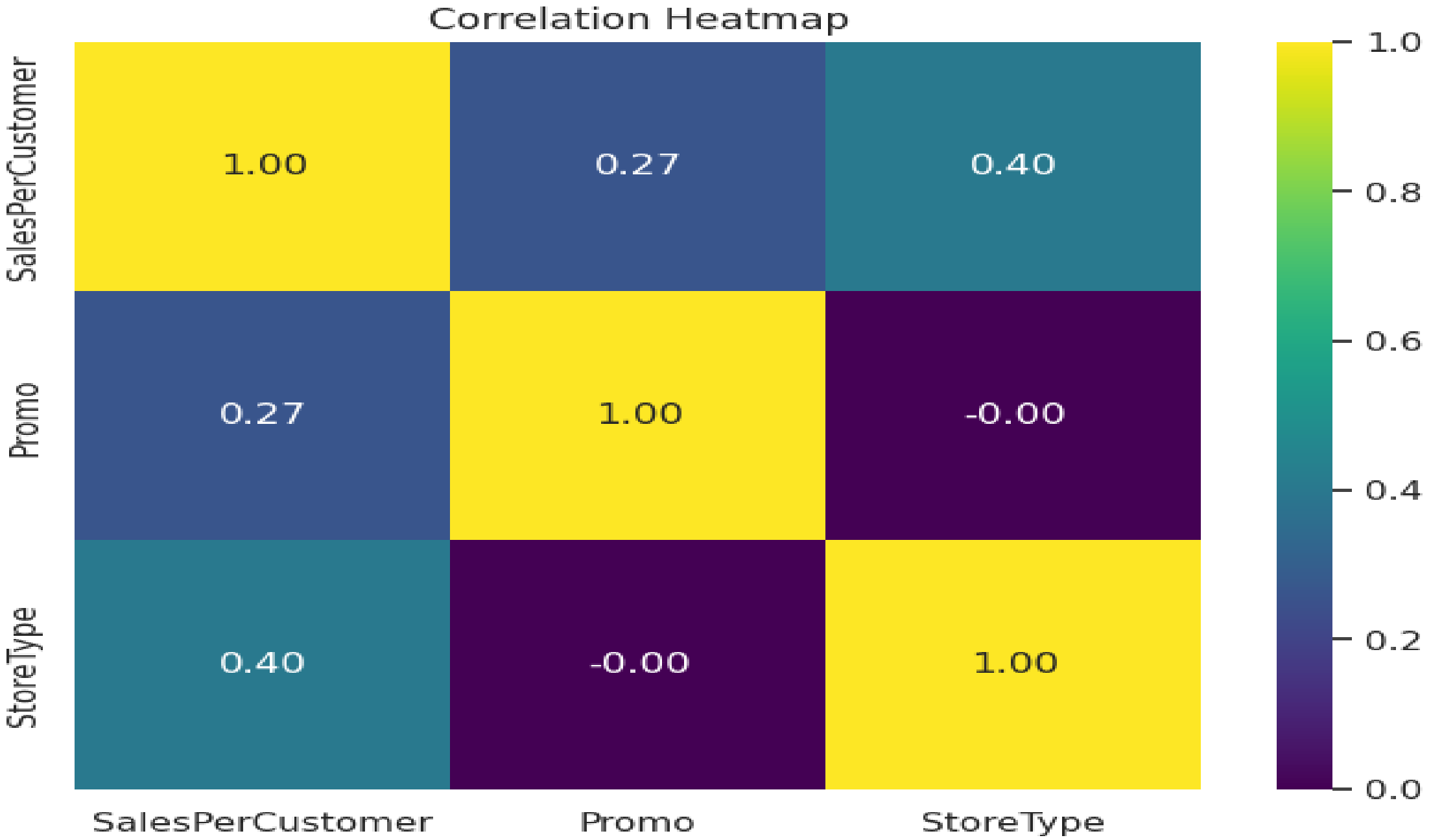


- An increase in the current time period, denoted as 't,' leading to an increase in the future time period 't+lag' indicates a positive correlation or dependency between the two time points.

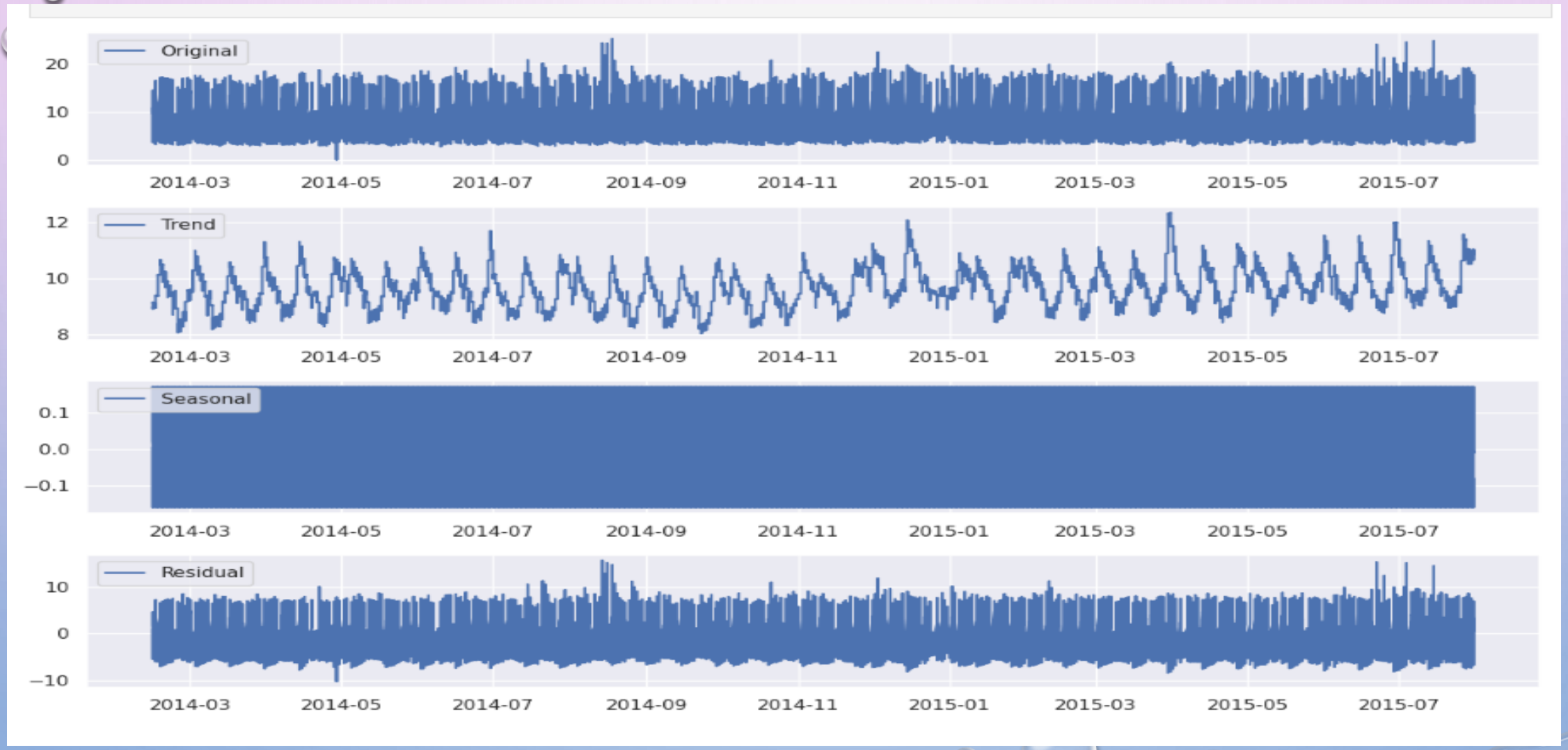
- In other words, as time progresses from 't' to 't+lag,' any changes, improvements, or variations occurring at 't' are expected to have a corresponding impact on the future time period 't+lag.



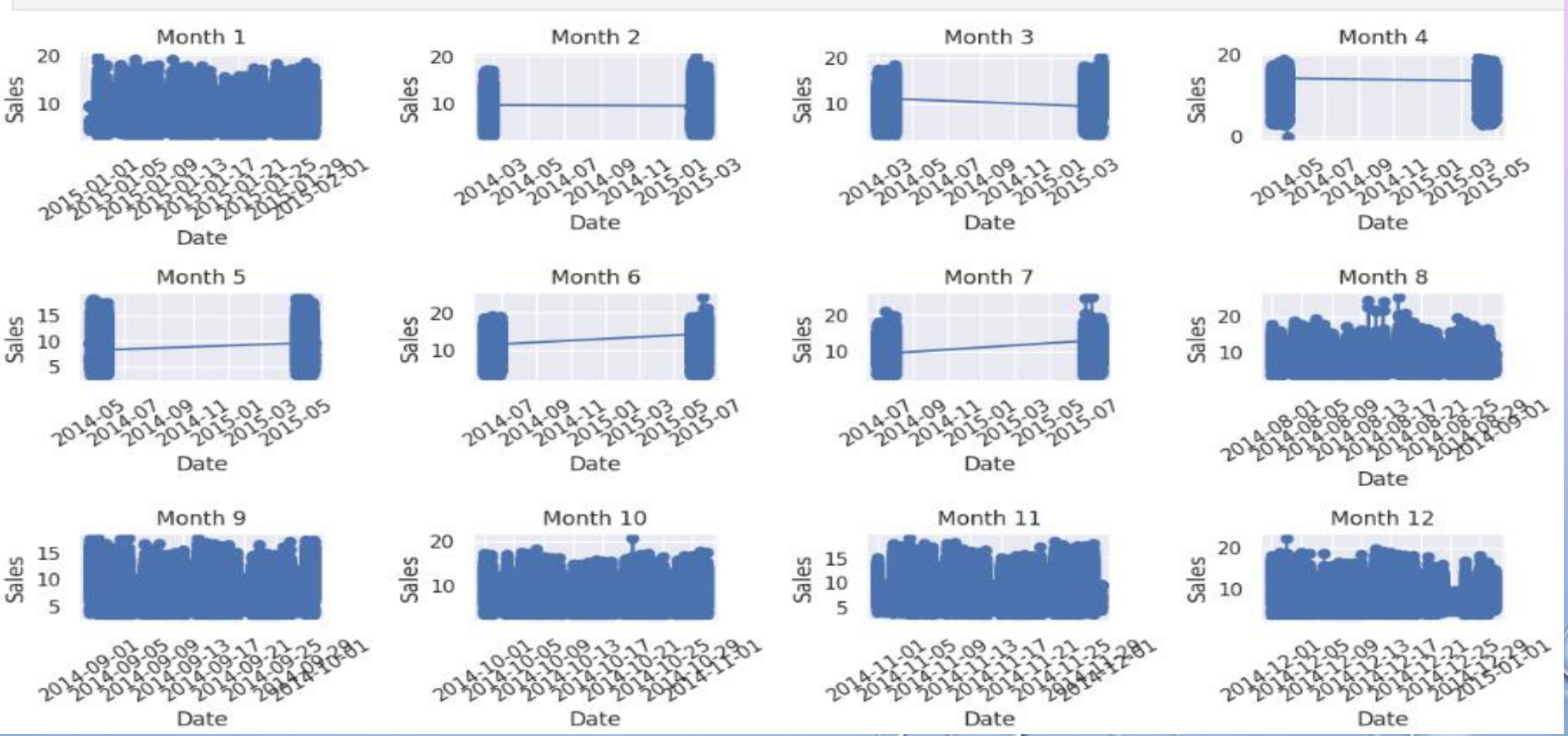

Customers and Sales Over Time with Promo Highlights
Customers
Promo
Count
11.0
10.5
10.0
9.5
9.0
8.5
2014-03
2014-05
2014-07
2014-09
2014-11
2015-01
2015-03
2015-05
2015-07
Date


Correlation Heatmap
SalesPerCustomer
Promo
StoreType
1.00
0.27
0.40
0.27
1.00
-0.00
0.40
-0.00
1.00
SalesPerCustomer
Promo
StoreType
1.0
0.8
0.6
0.4
0.2
0.0

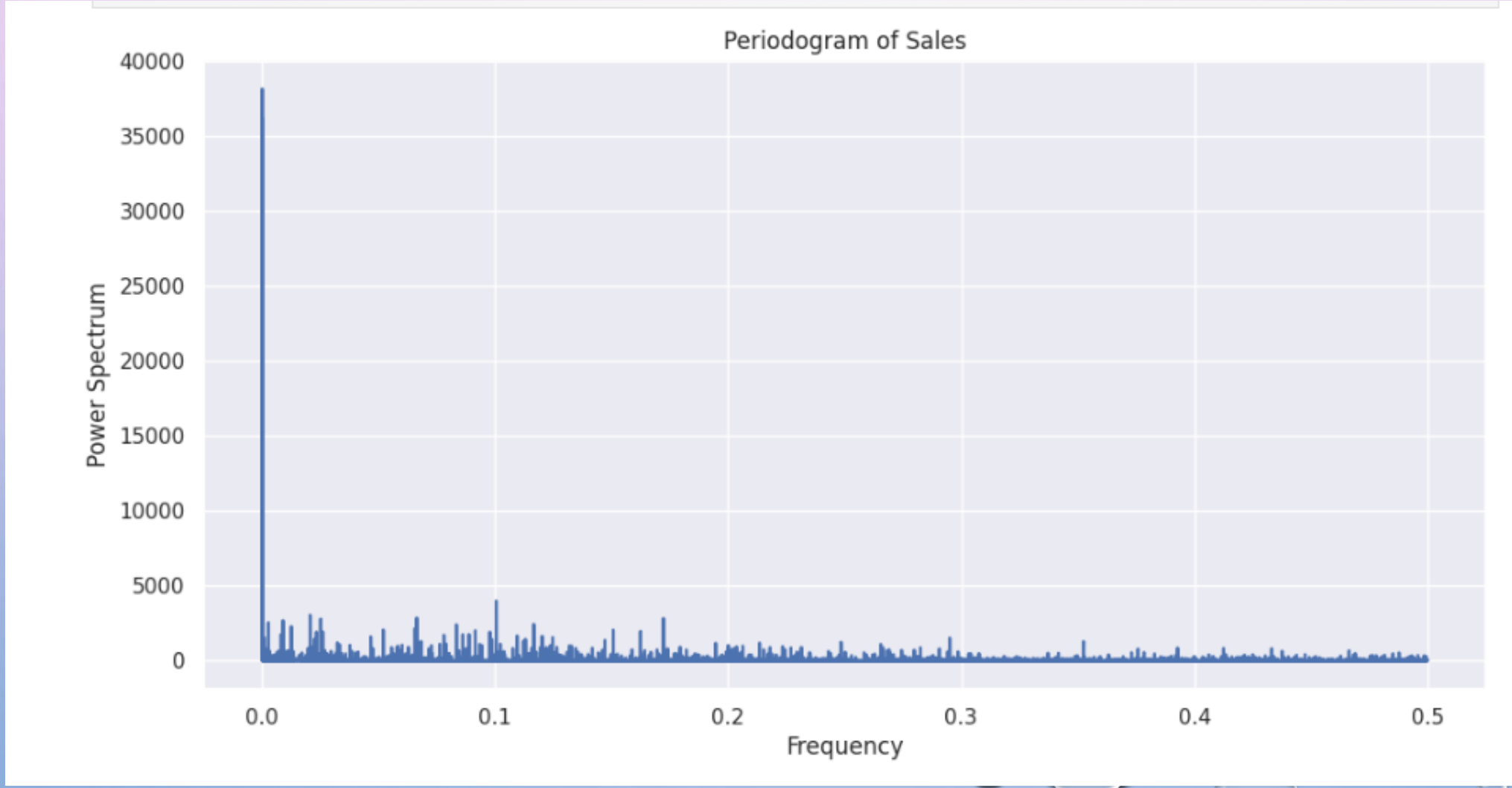
Plot the decomposed components

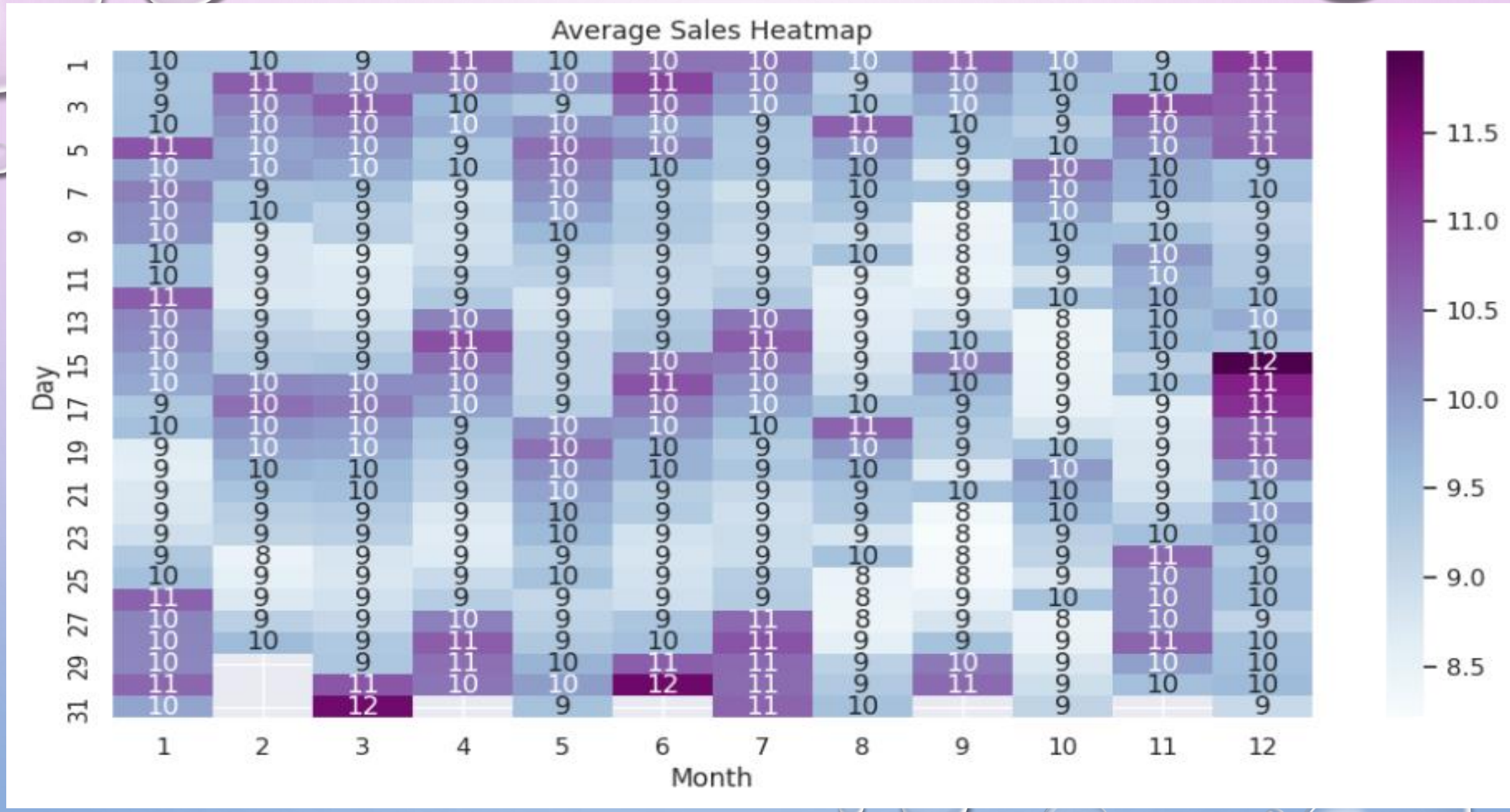


Iterate through each month and create subplots

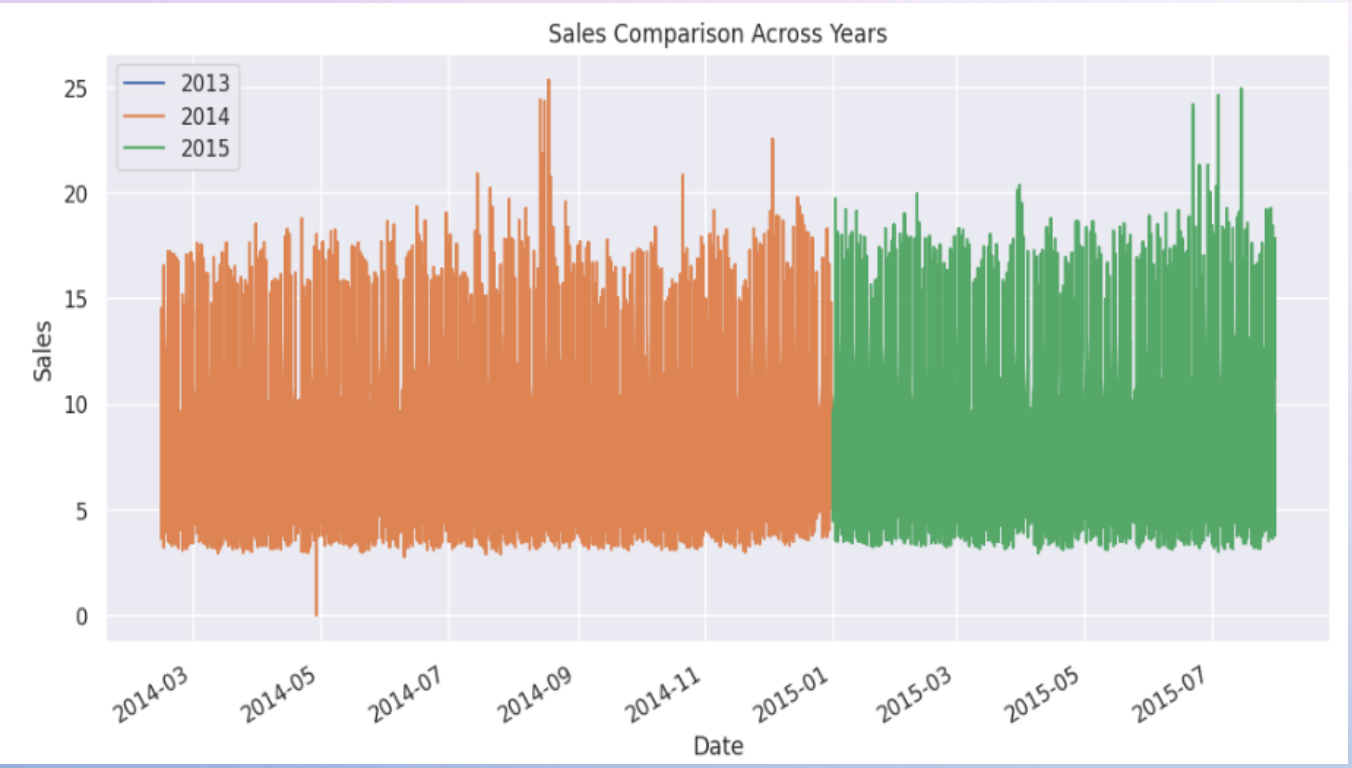


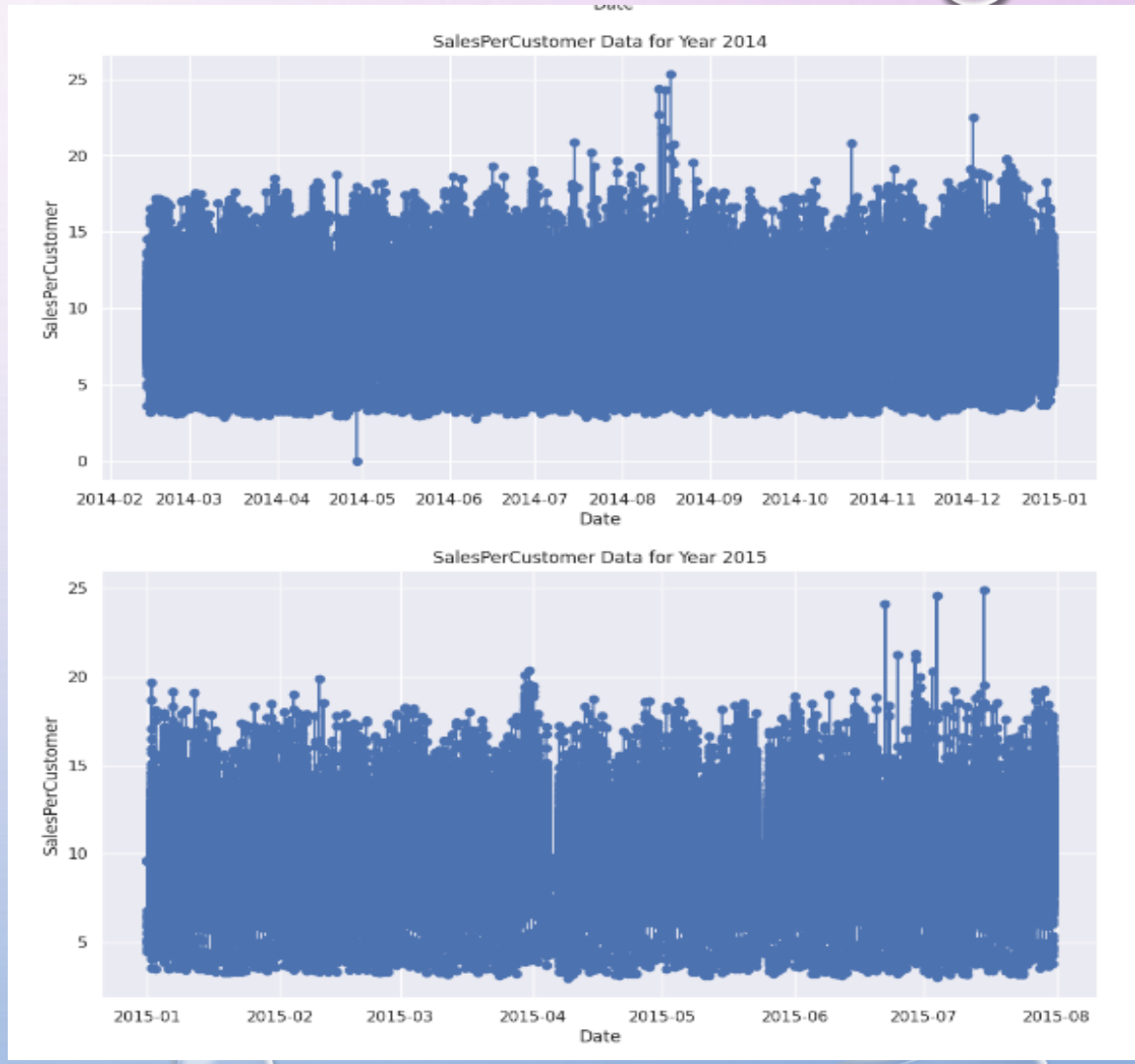
A periodogram showing high spectral power near 0 and decreasing power as it approaches 1 suggests that the sales data is characterized by a dominant low-frequency component, possibly indicating long-term trends or seasonality



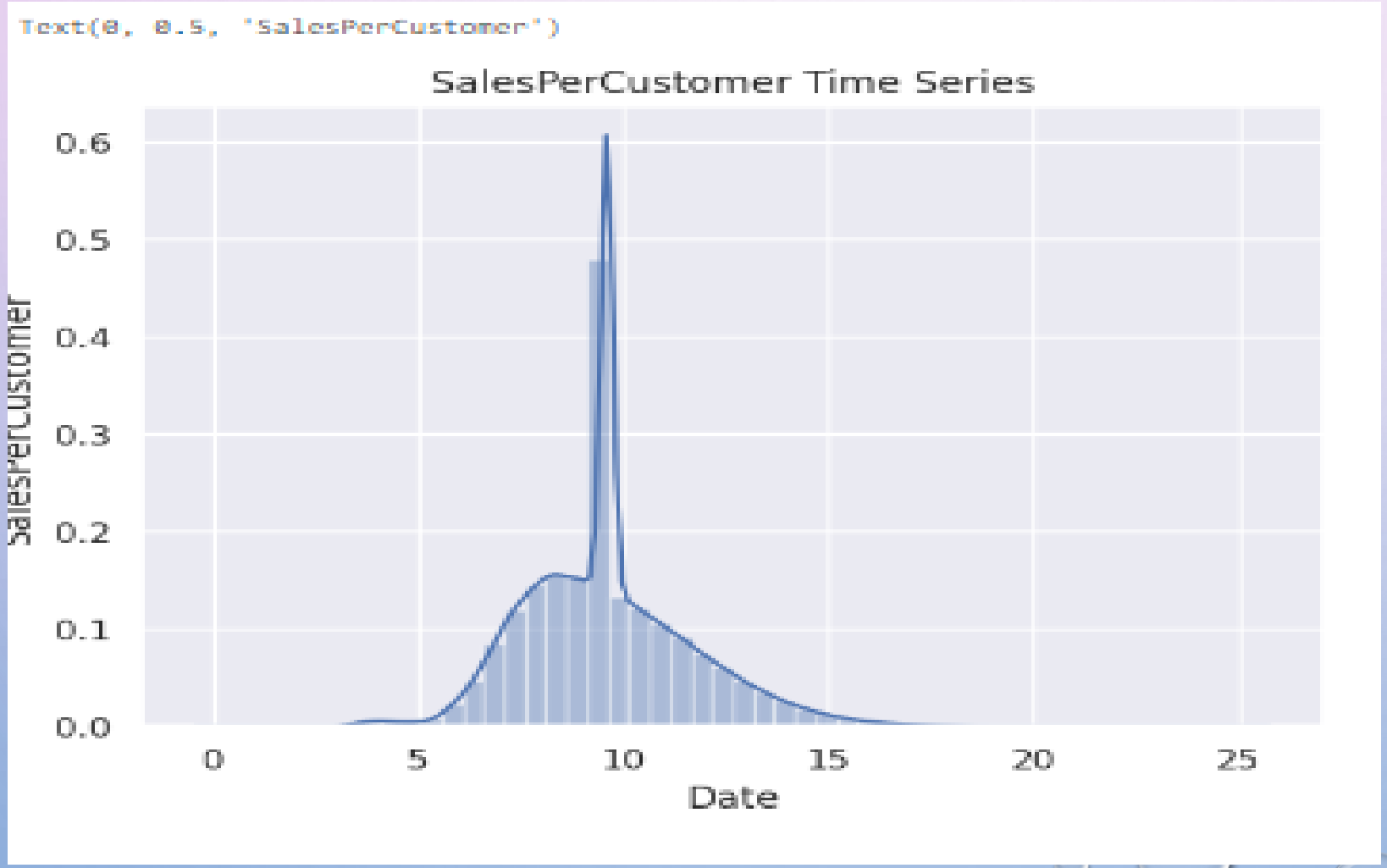

Average Sales Heatmap
Day
Month
1
3
5
7
9
11
13
15
17
19
21
23
25
27
29
31
1
2
3
4
5
6
7
8
9
10
11
12
11.5
11.0
10.5
10.0
9.5
9.0
8.5

The sales data exhibits low variation, with a substantial increase in sales in the year 2013, followed by consistent sales levels in the subsequent years of 2014 and 2015.



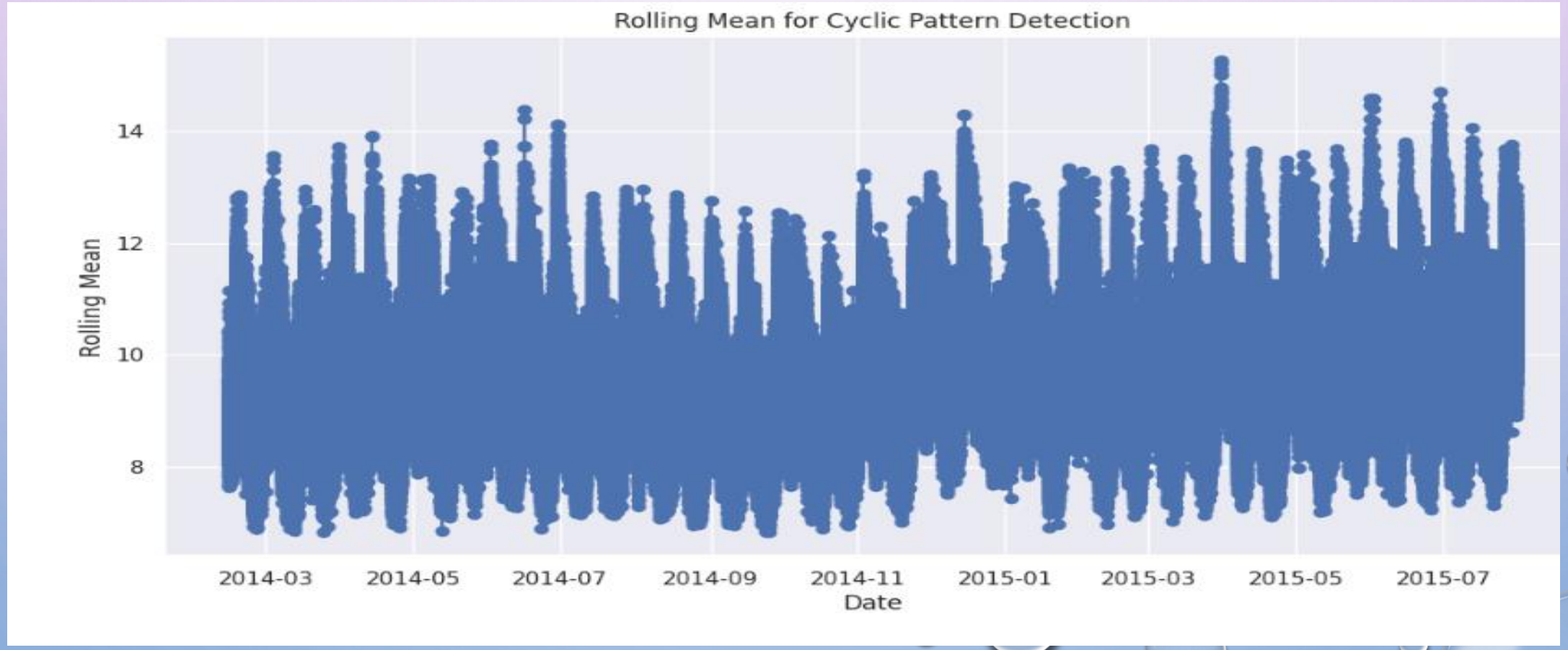


Stationarity Check



Cyclicality:

Cyclicality refers to the presence of long-term periodic patterns in a time series that are not tied to a fixed frequency like seasonality. These cycles usually last longer than a year and are not as regular as seasonal patterns. Cycles can be caused by economic, political, or social factors that influence the data over time. Unlike seasonality, cyclic patterns are not as predictable and can vary in amplitude and duration



ARIMA (AutoRegressive Integrated Moving Average):

ARIMA models are versatile and can handle both trend and seasonality in data.
They consist of three main components: AutoRegressive (AR) terms, Integrated (I) terms for differencing, and Moving Average (MA) terms.
Model selection involves determining the order of these components (p, d, q) based on ACF and PACF plots.
You can use functions like auto_arima from the pmdarima library for automatic ARIMA order selection.
SARIMA (Seasonal ARIMA)

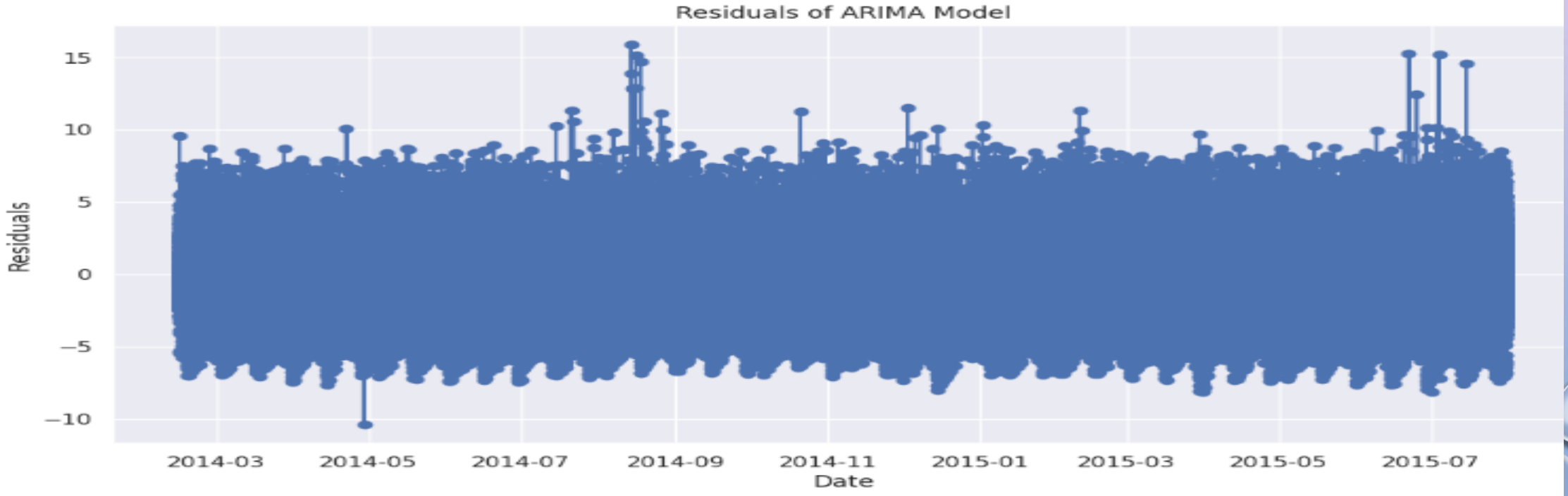
SARIMAX Results

Dep. Variable:	SalesPerCustomer	No. Observations:	560478
Model:	ARIMA(1, 1, 1)	Log Likelihood	-1152617.734
Date:	Tue, 05 Sep 2023	AIC	2305241.468
Time:	10:50:49	BIC	2305275.177
Sample:	0 - 560478	HQIC	2305250.965
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	-0.0055	0.001	-4.392	0.000	-0.008	-0.003
ma.L1	-0.9903	0.000	-5966.247	0.000	-0.991	-0.990
sigma2	3.5791	0.006	634.342	0.000	3.568	3.590

Ljung-Box (L1) (Q):	0.00	Jarque-Bera (JB):	31355.14
Prob(Q):	0.96	Prob(JB):	0.00
Heteroskedasticity (H):	1.09	Skew:	0.37
Prob(H) (two-sided):	0.00	Kurtosis:	3.90

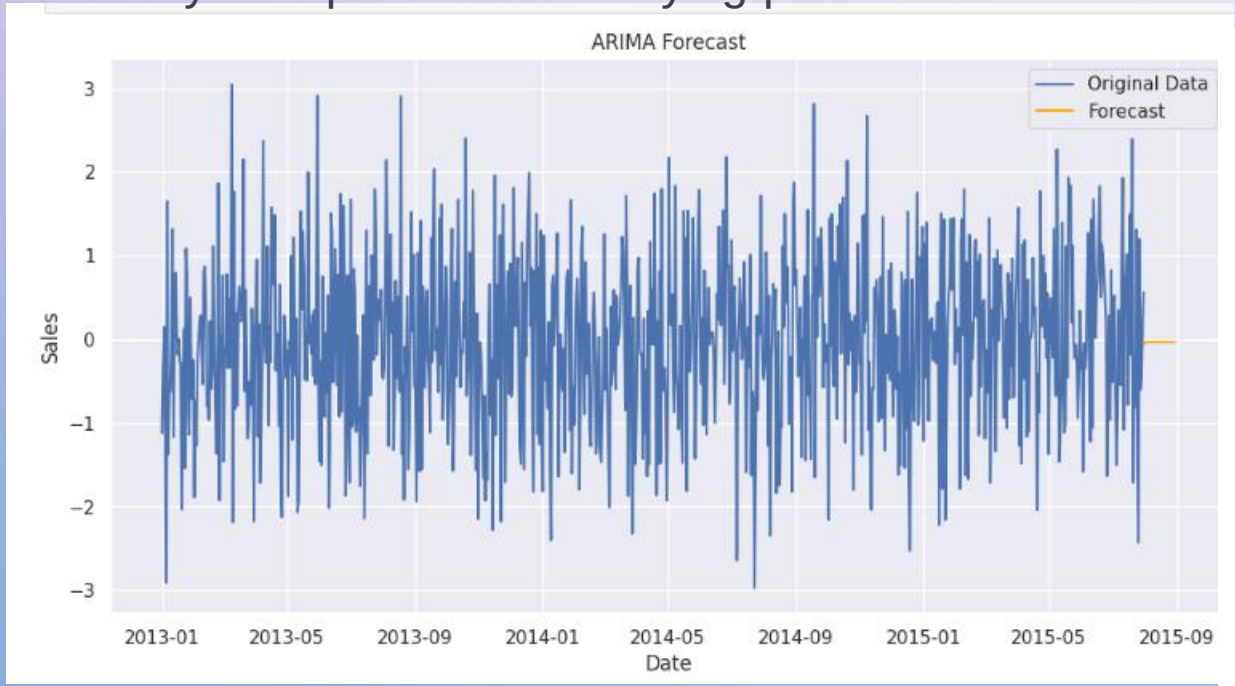
Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).

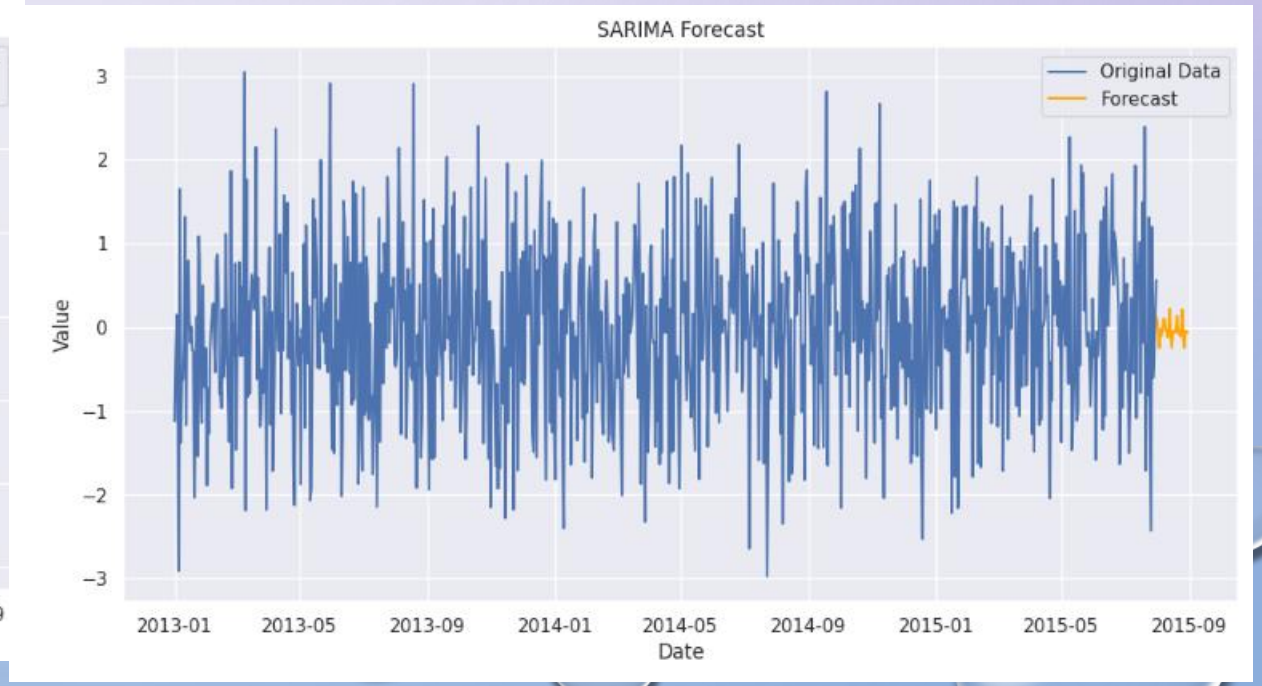


- In this stepwise search for the best ARIMA model, various combinations were tested to minimize the AIC (Akaike Information Criterion), which is a measure of a model's goodness of fit. The presented models represent different combinations of autoregressive (AR), differencing (I), and moving average (MA) components, as well as seasonal components (S).

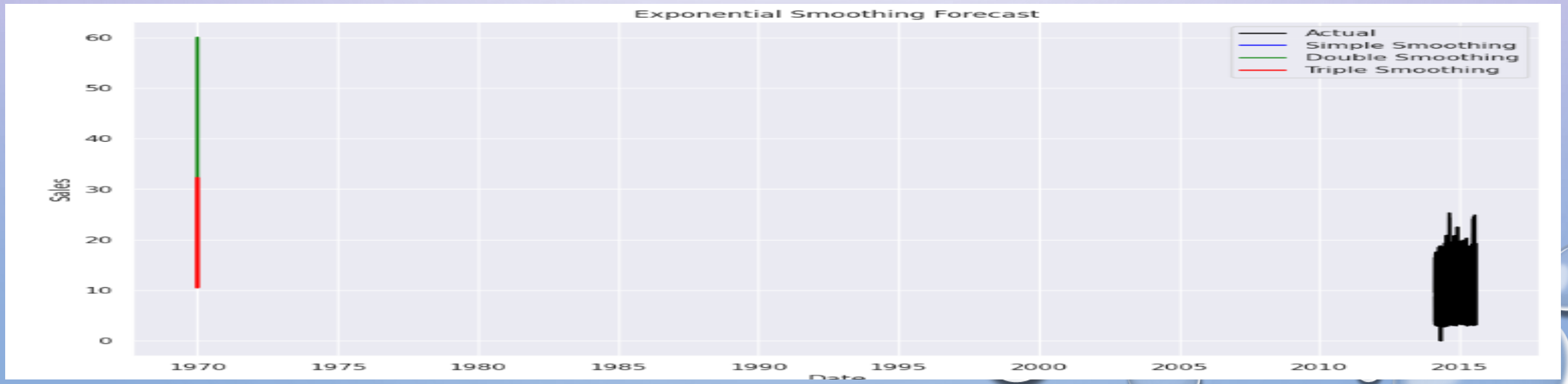
- The best-fitting model identified is ARIMA(0,1,0)(1,1,0)[12], which suggests a non-seasonal differencing order of 1, no autoregressive or moving average components, and a seasonal differencing order of 1 with a seasonal component of 12 (indicating seasonality at a 12-month interval).

- The results of this best-fitting model show that it has the lowest AIC value (408.305), indicating a good fit to the data. The coefficients and diagnostic statistics are also provided, offering insights into the model's performance and its ability to capture the underlying patterns in the time series data.

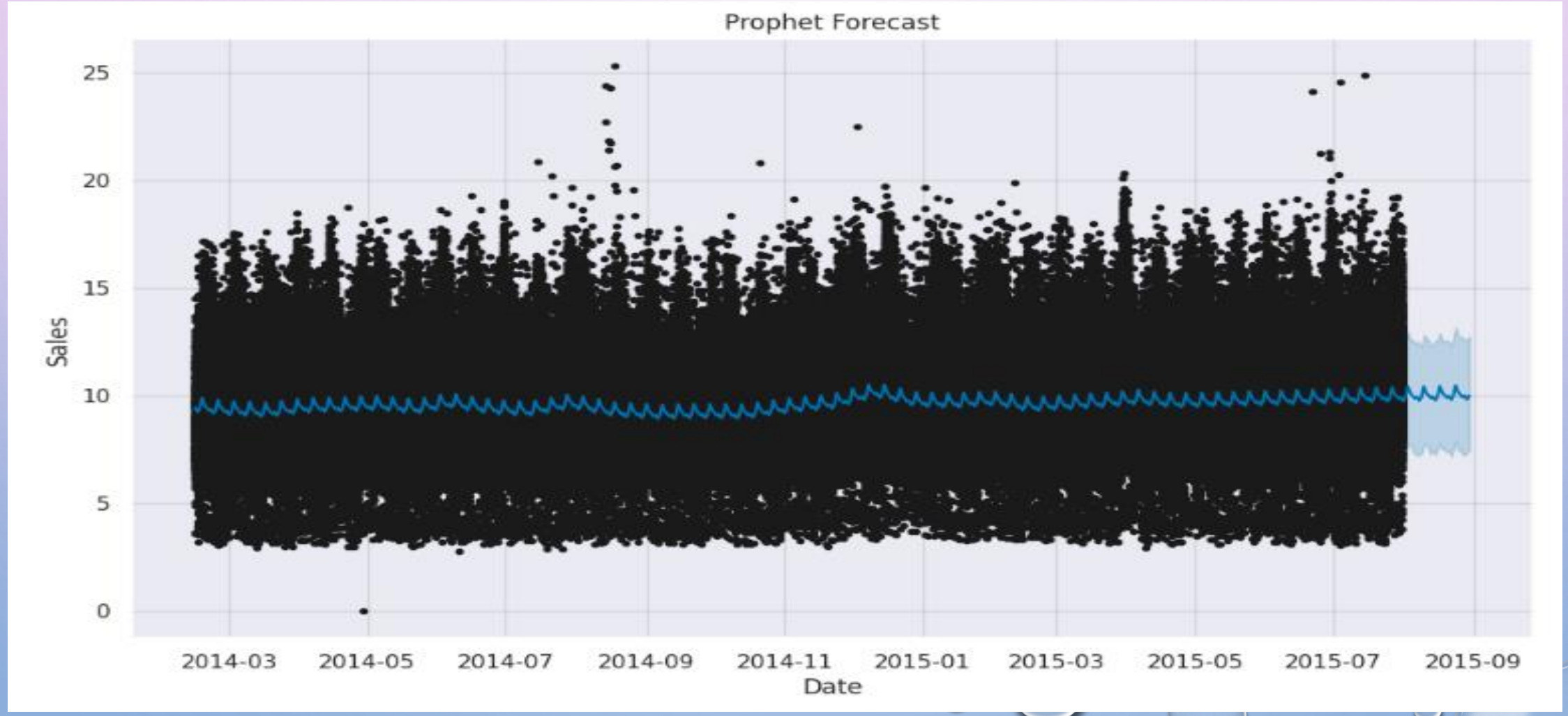




In exponential smoothing, the actual values tend to deviate from the predicted values.



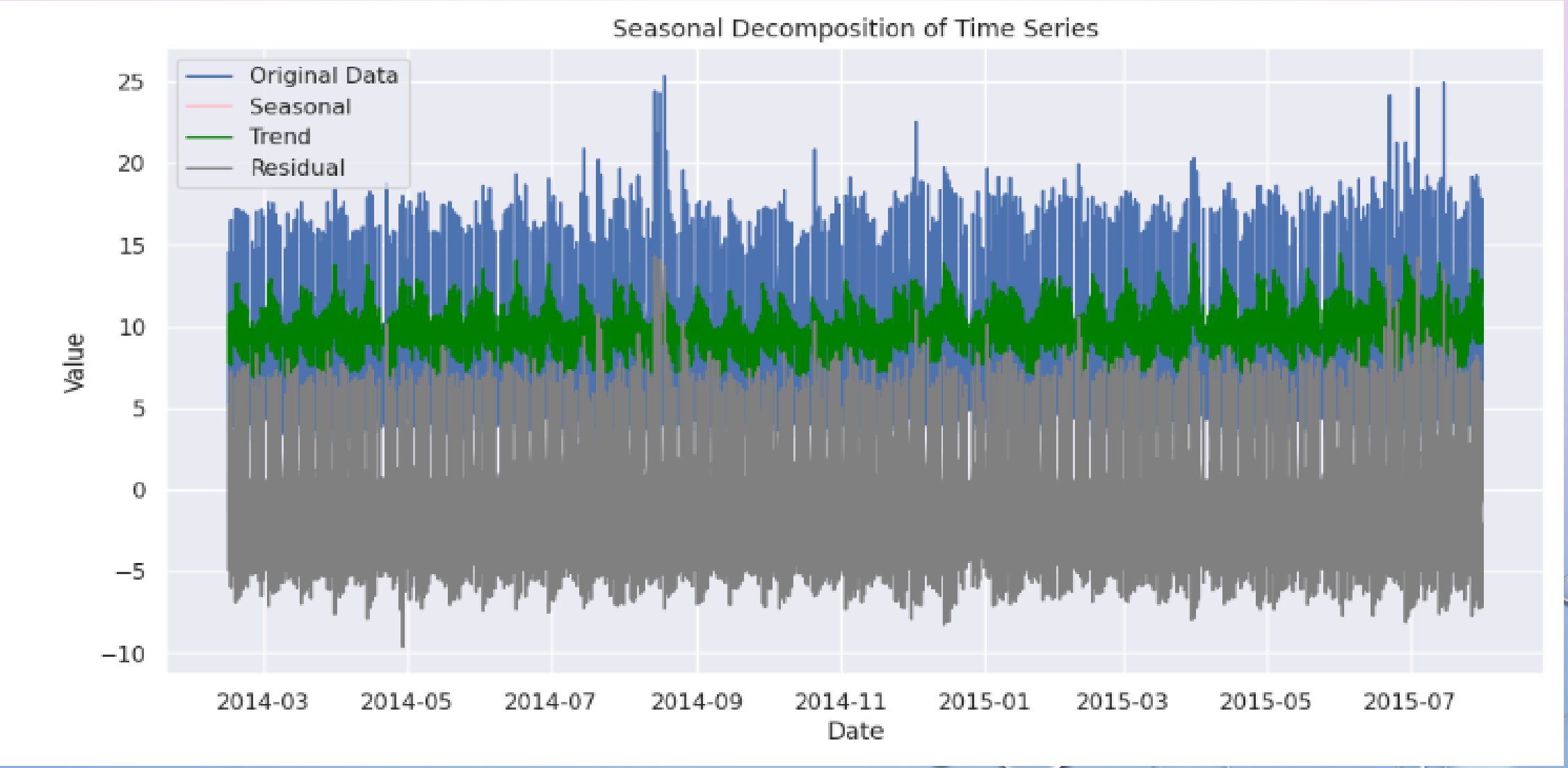
Prophet model



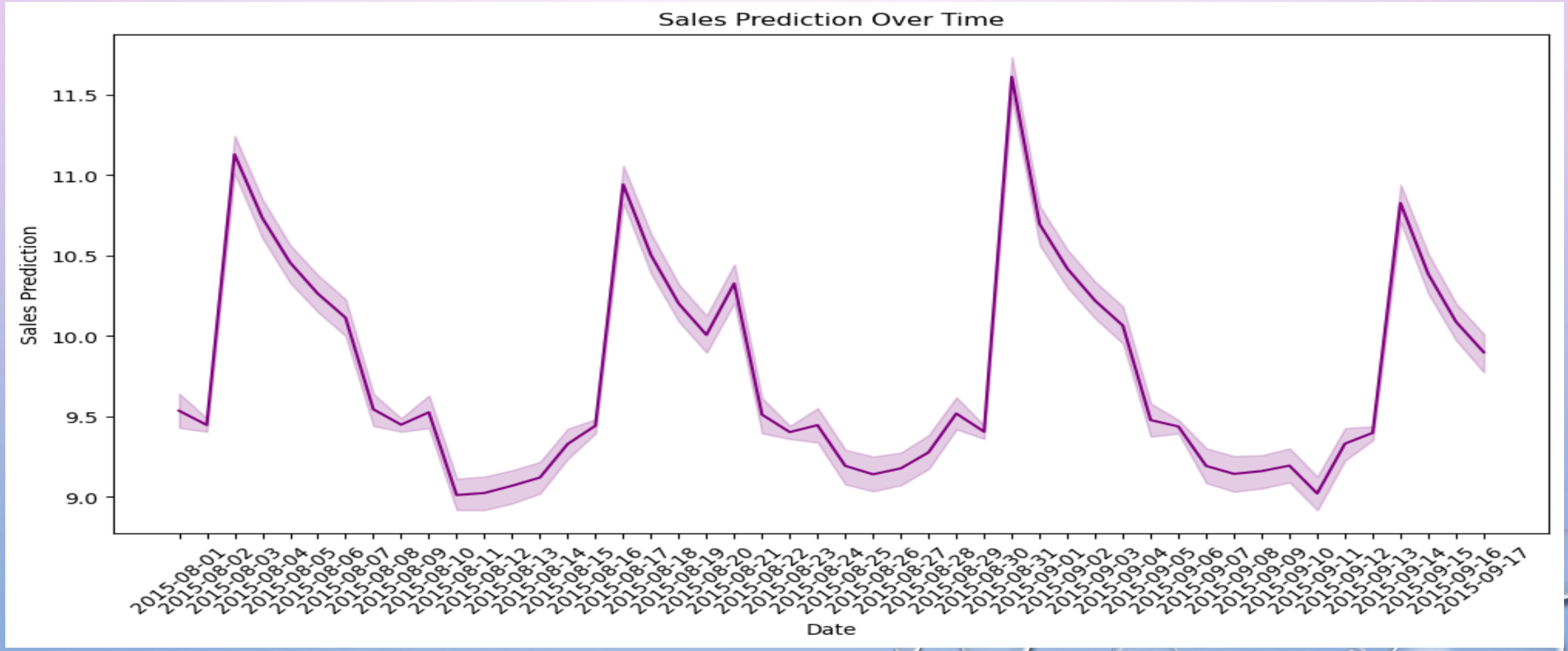
- The training loss of 0.00057 and the validation loss of 0.00064, both sustained over 8 epochs, indicate a model that has achieved a high level of accuracy and generalization.
- The training loss reflects how well the model is fitting the training data, and the validation loss gauges the model's performance on unseen data.

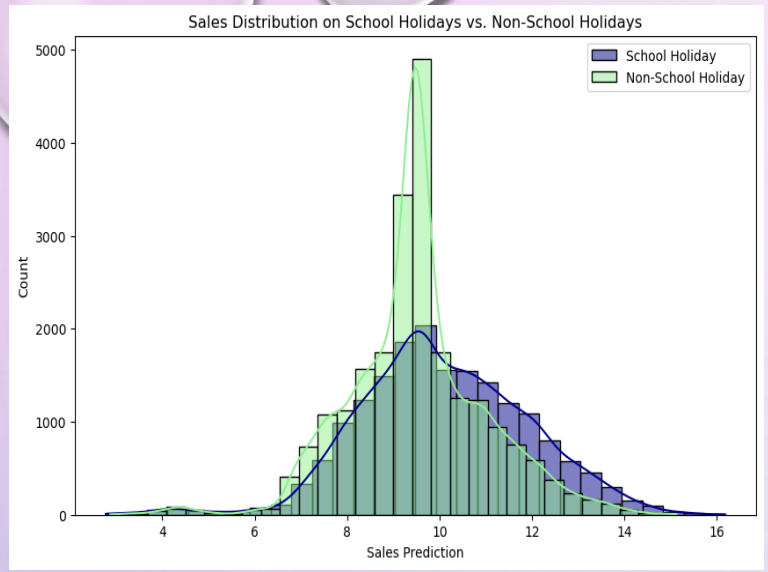
```
Epoch 1/5
14012/14012 - 135s - loss: 0.0059 - val_loss: 0.0064 - 135s/epoch - 10ms/step
Epoch 2/5
14012/14012 - 97s - loss: 0.0058 - val_loss: 0.0064 - 97s/epoch - 7ms/step
Epoch 3/5
14012/14012 - 93s - loss: 0.0057 - val_loss: 0.0063 - 93s/epoch - 7ms/step
Epoch 4/5
14012/14012 - 92s - loss: 0.0057 - val_loss: 0.0064 - 92s/epoch - 7ms/step
Epoch 5/5
14012/14012 - 93s - loss: 0.0057 - val_loss: 0.0063 - 93s/epoch - 7ms/step
```

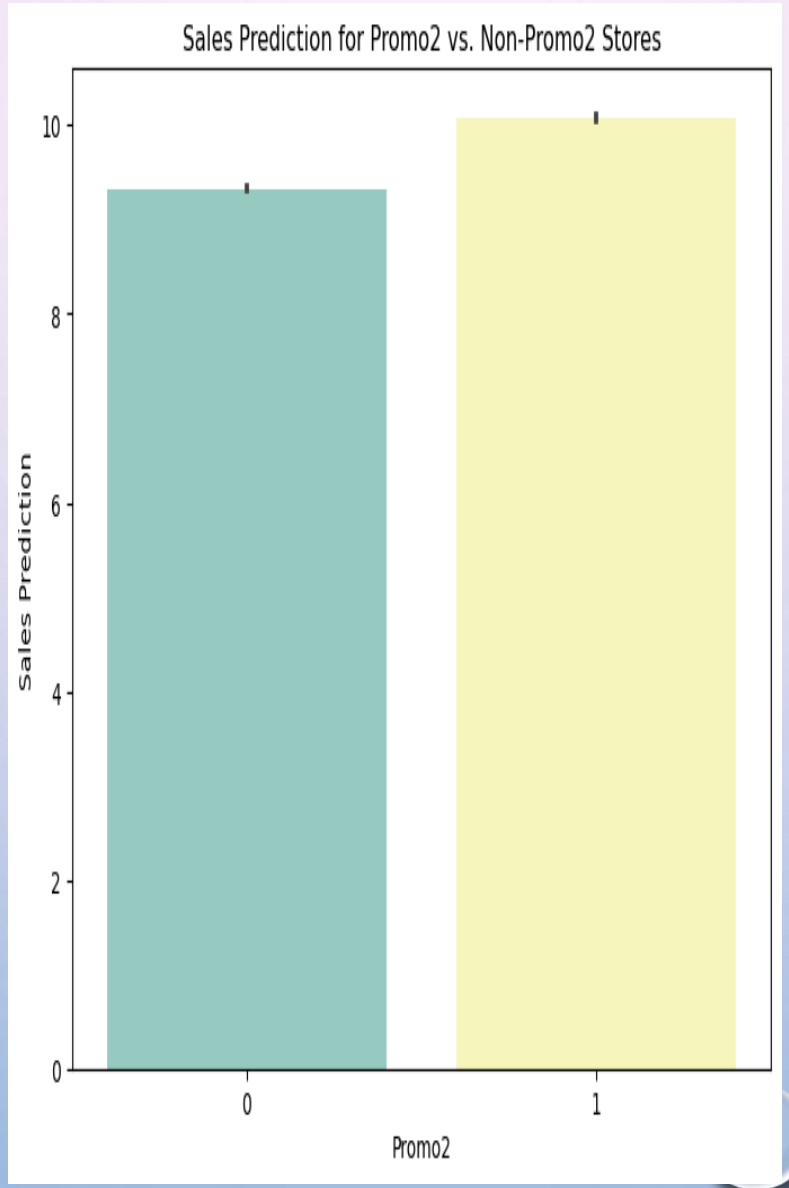
Seasonal Decomposition of Time Series

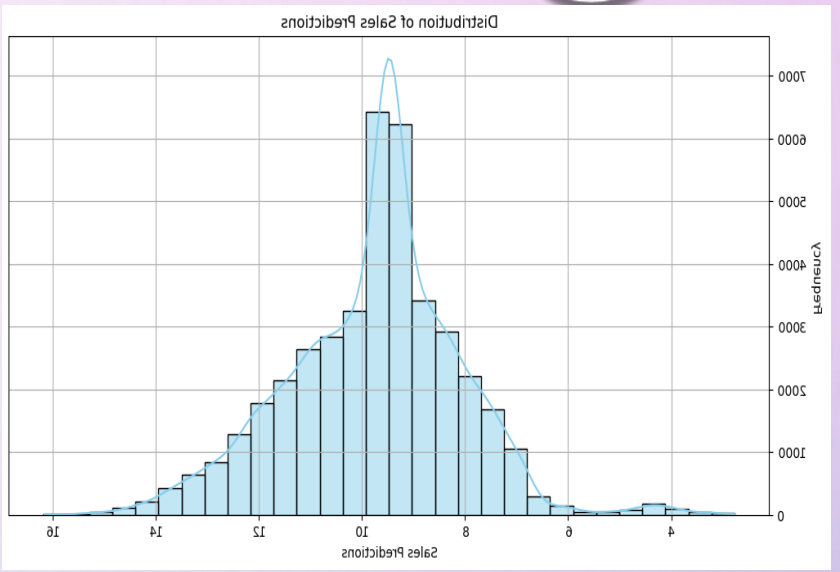


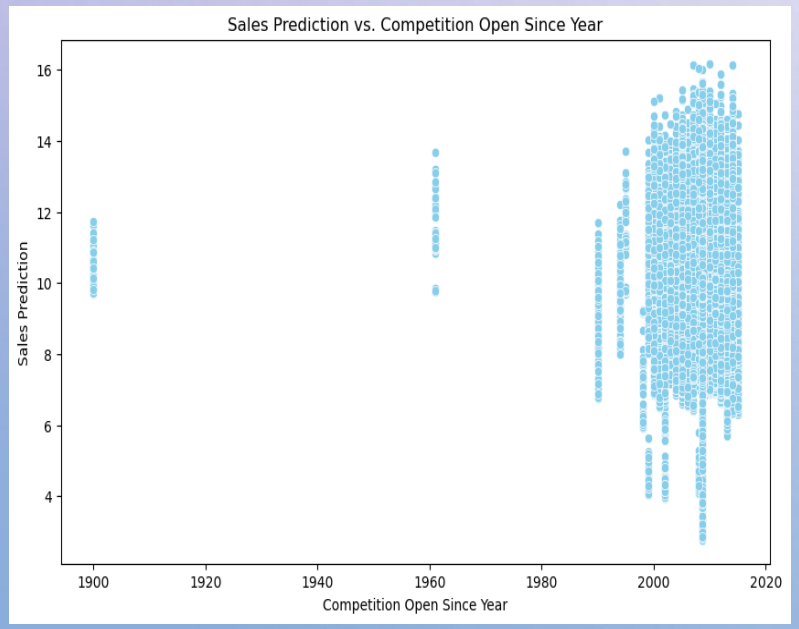
Sales Predicted for Test Data

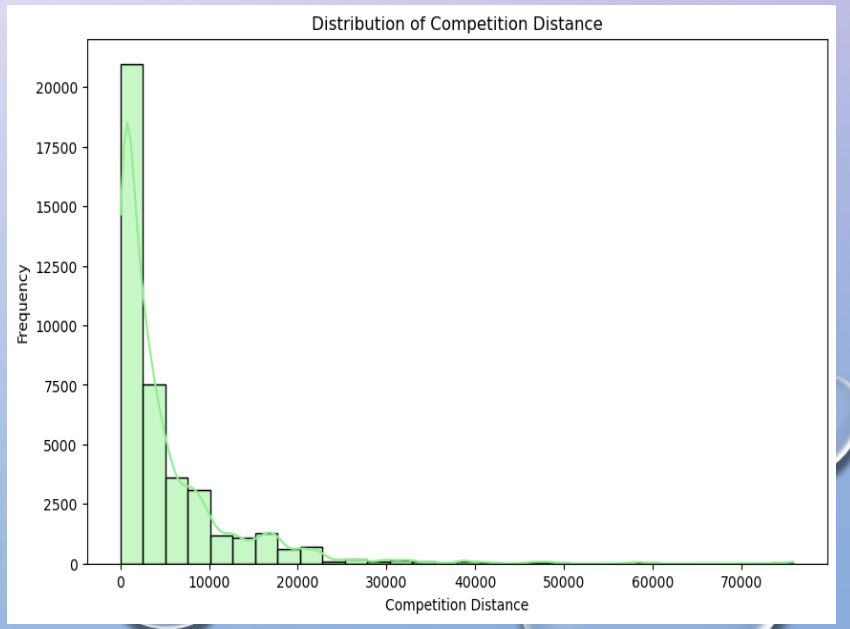


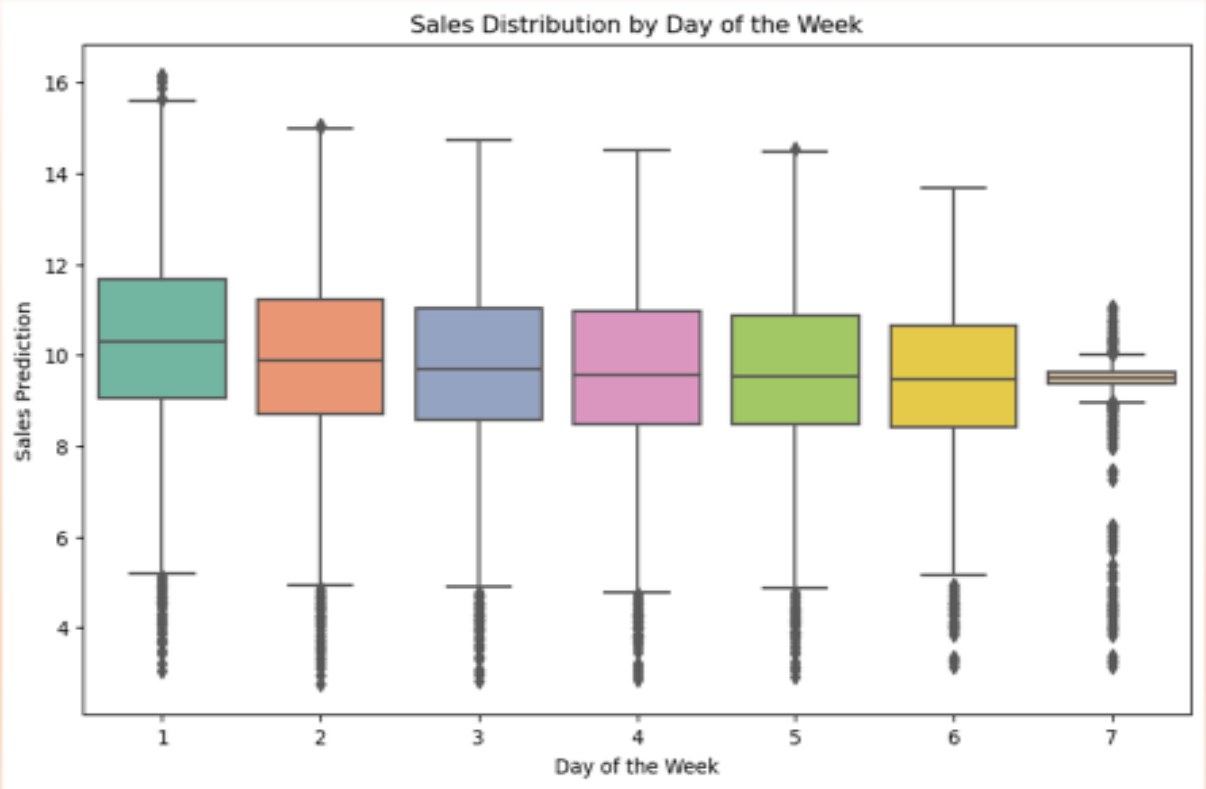

Sales Distribution on School Holidays vs. Non-School Holidays
School Holiday
Non-School Holiday
Count
Sales Prediction

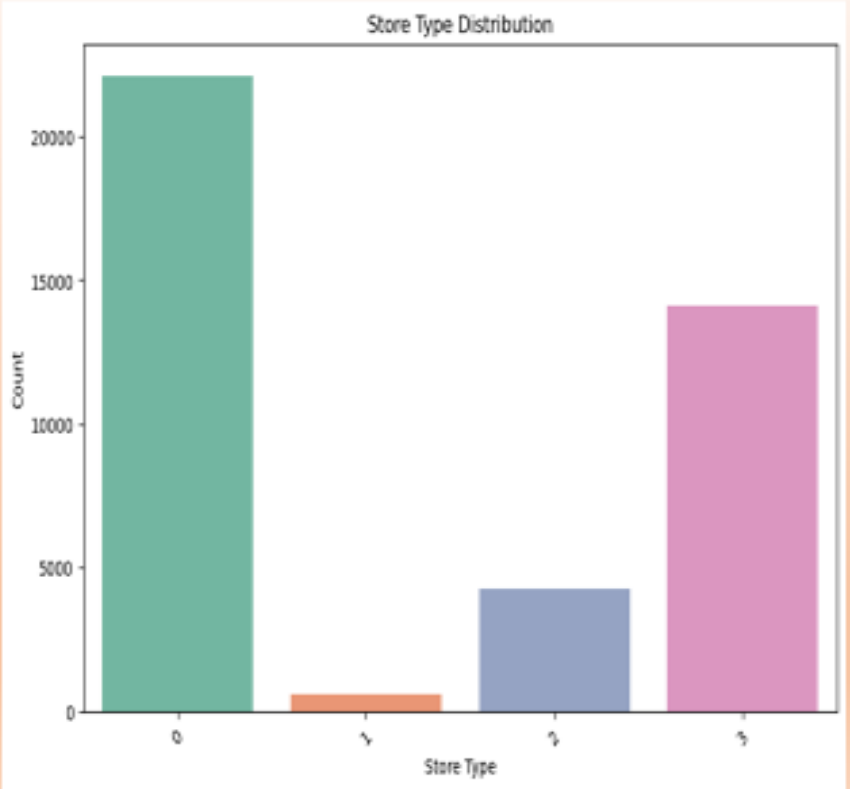

Sales Prediction for Promo2 vs. Non-Promo2 Stores
Sales Prediction
Promo2

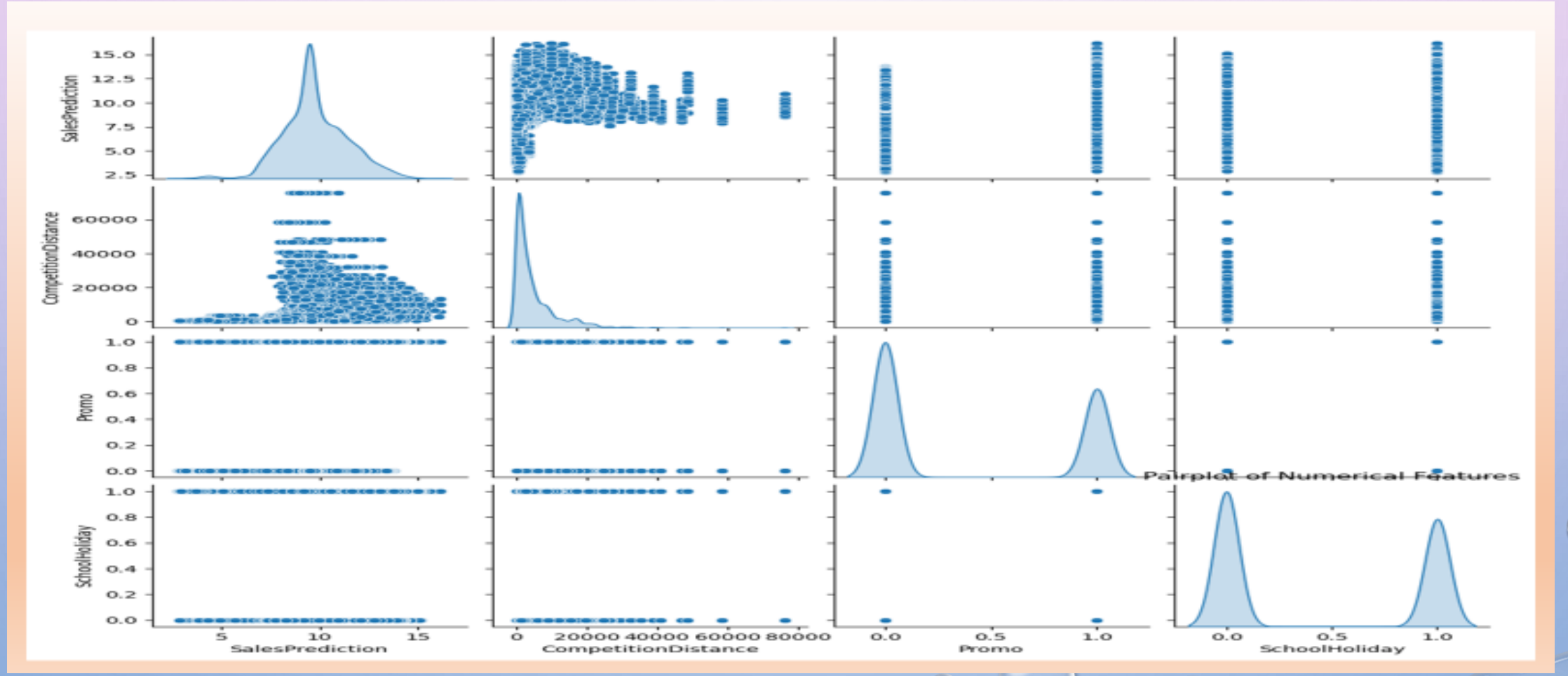

Distribution of Sales Predictions


Sales Prediction vs. Competition Open Since Year
Sales Prediction
Competition Open Since Year


Distribution of Competition Distance
Frequency
Competition Distance


Sales Distribution by Day of the Week
Sales Prediction
16
14
12
10
8
6
4
1
2
3
4
5
6
7
Day of the Week


Store Type Distribution
Count
20000
15000
10000
5000
0
Store Type


SalesPrediction
CompetitionDistance
Promo
SchoolHoliday
Pairplot of Numerical Features

Sales Prediction for next 6 Weeks

Sales Predictions for Next 6 Weeks

Date	Week	PredictedSales
2015-09-17	1	8.794497
2015-09-24	2	8.663102
2015-10-01	3	8.658852
2015-10-08	4	8.658711
2015-10-15	5	8.658706
2015-10-22	6	8.658706

Thank You