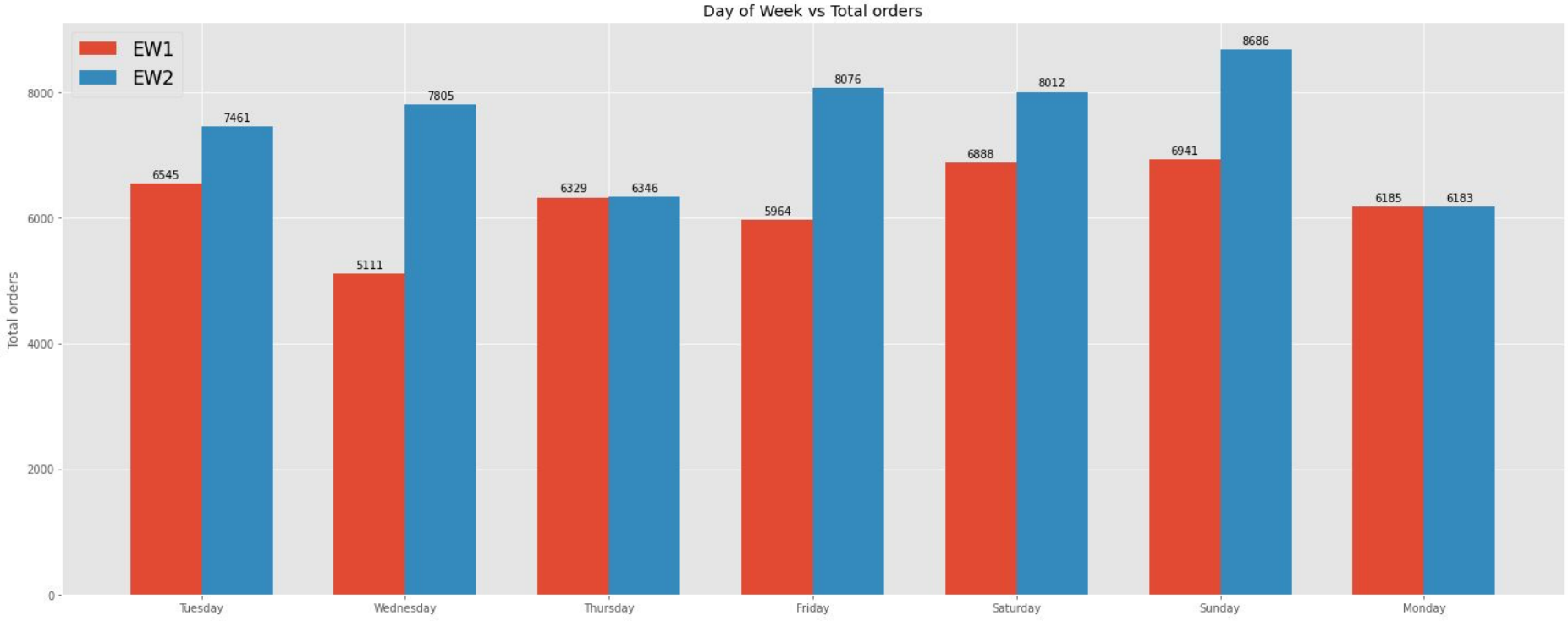
Airlift Case Study

Submitted By:
Muhammad Hamza Naviwala

Exploratory data analysis

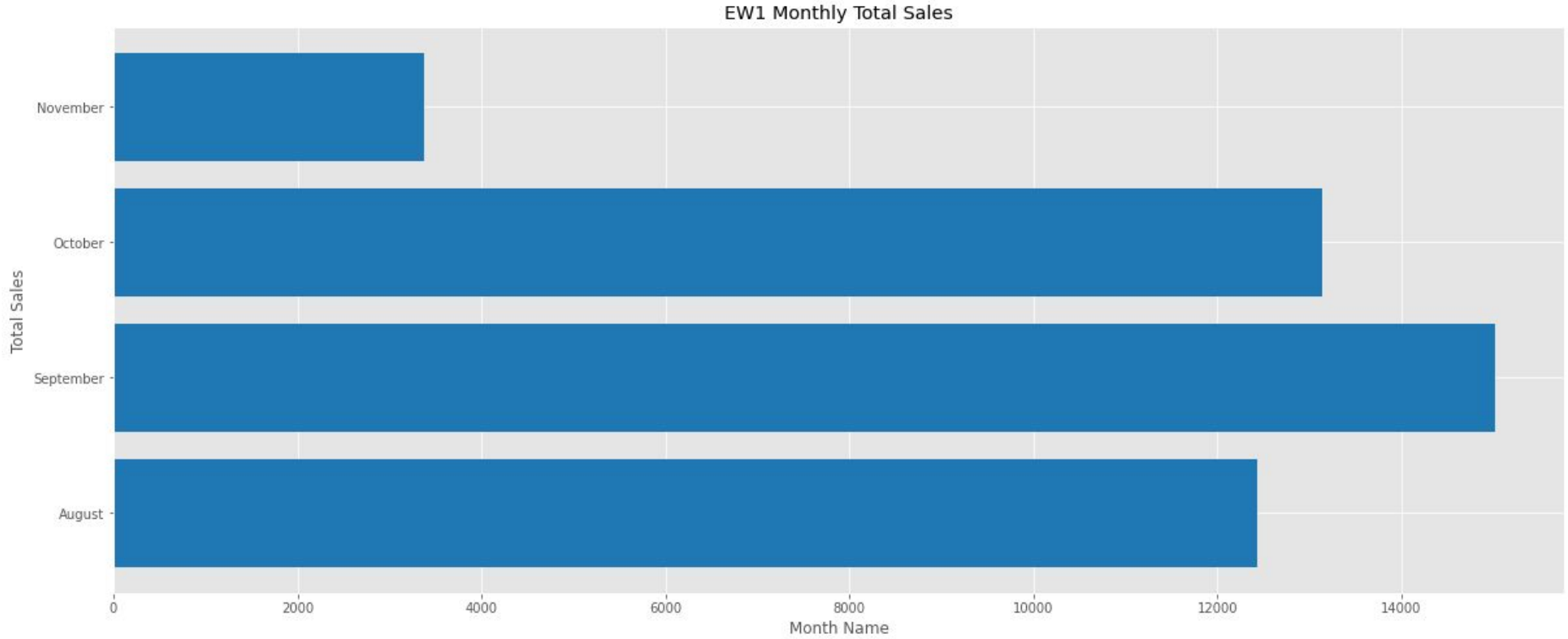
The graph below shows weekdays on x-axis and total sales on y-axis

I tried to find if there was a relationship between day of the week and sales. For EW1 there does not appear to be a relationship but for EW2 orders seem to increase on weekends like Sunday



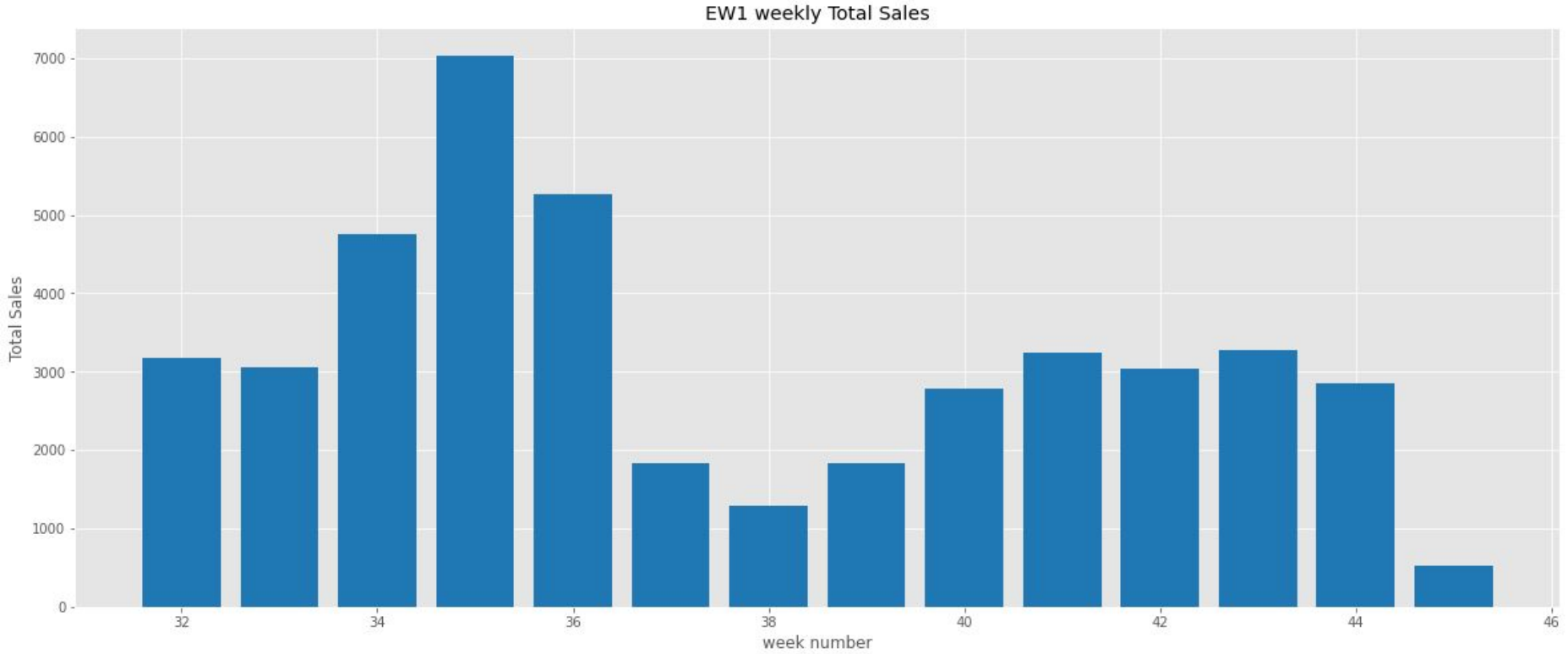
The graph below shows total sales on x-axis and months on y-axis for EW1

I tried to find if there was a relationship between month and sales. Based on three months it shows increasing trend but high sales for September indicates some type of special event, e.g discounts



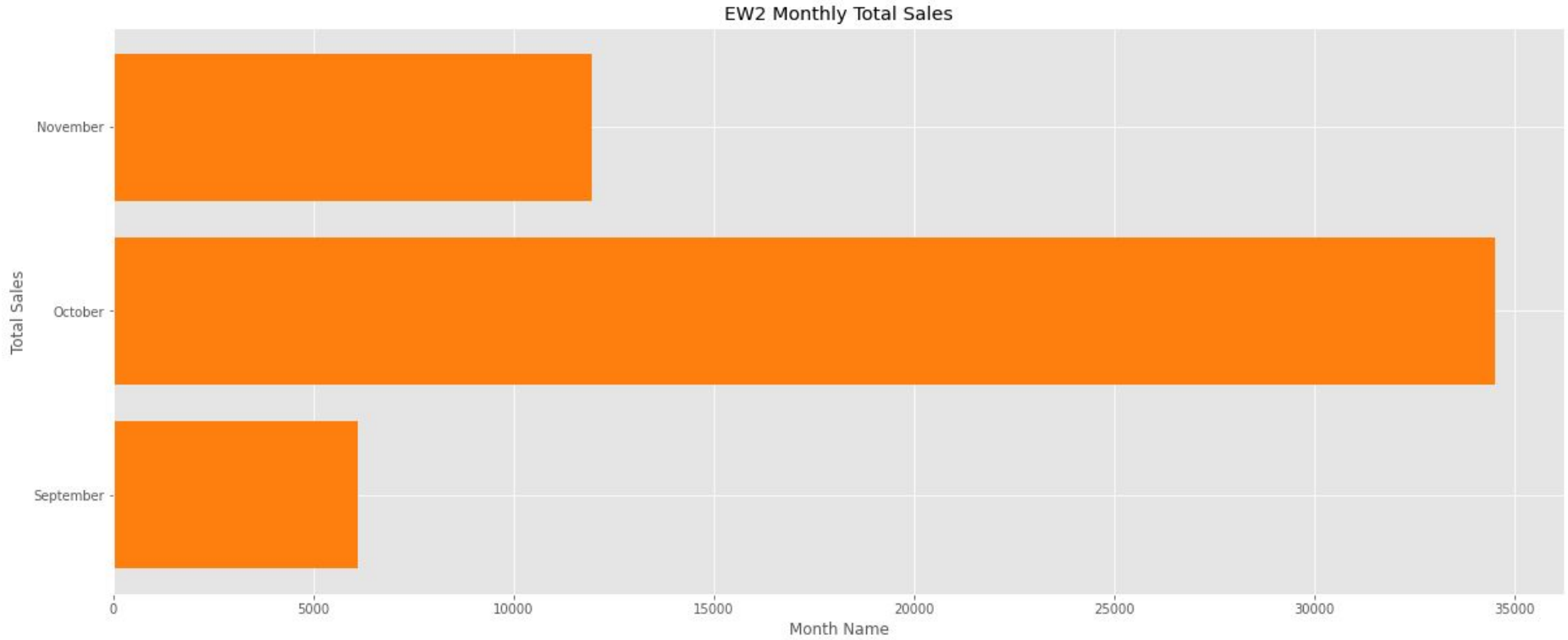
The graph below shows total sales on x-axis and months on y-axis for EW1

I tried to find if there was a relationship between weeks and sales. This graph give a bit more information that the special event happened in end of August and start of September



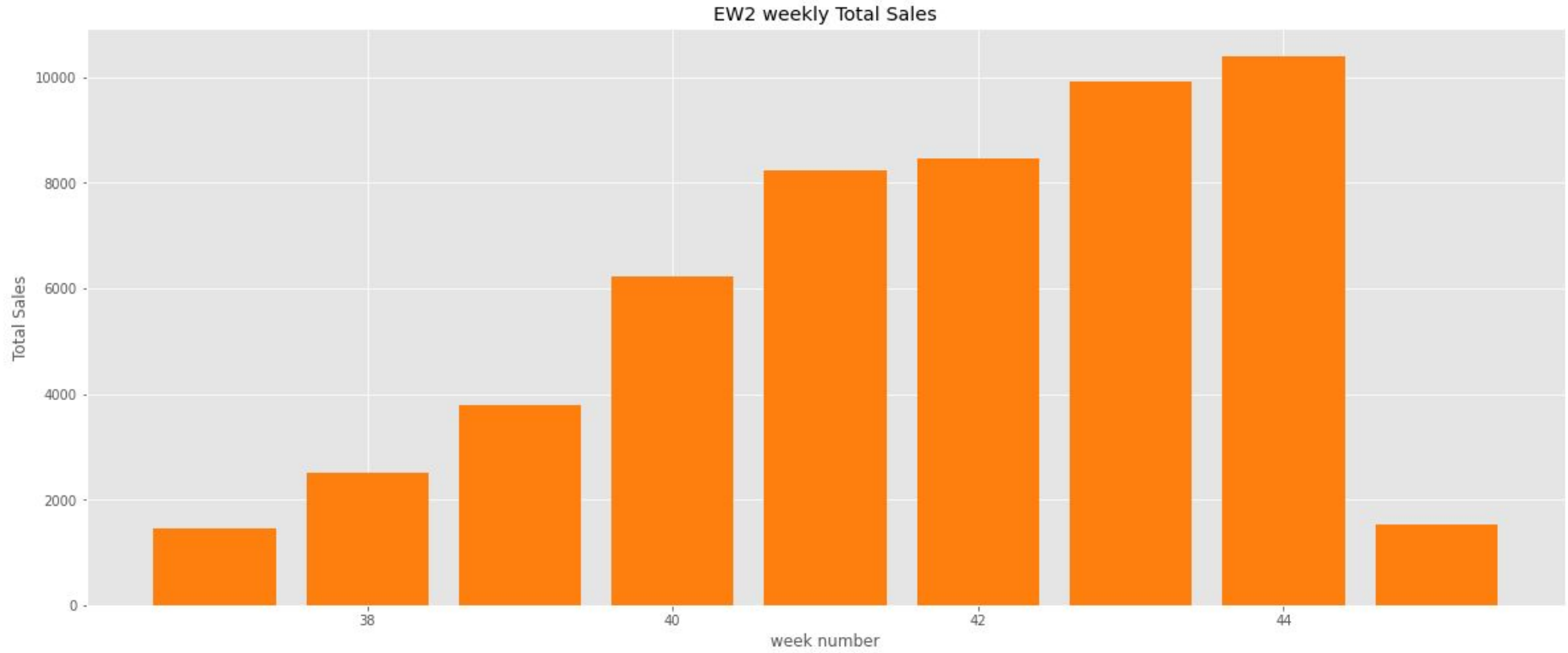
The graph below shows total sales on x-axis and months on y-axis for EW2

I tried to find if there was a relationship between month and sales. No sales for August indicates that this warehouse may have opened in September. Enormous sales in October might mean store opening discount or awareness amongst people living in that area(could be due to great marketing).

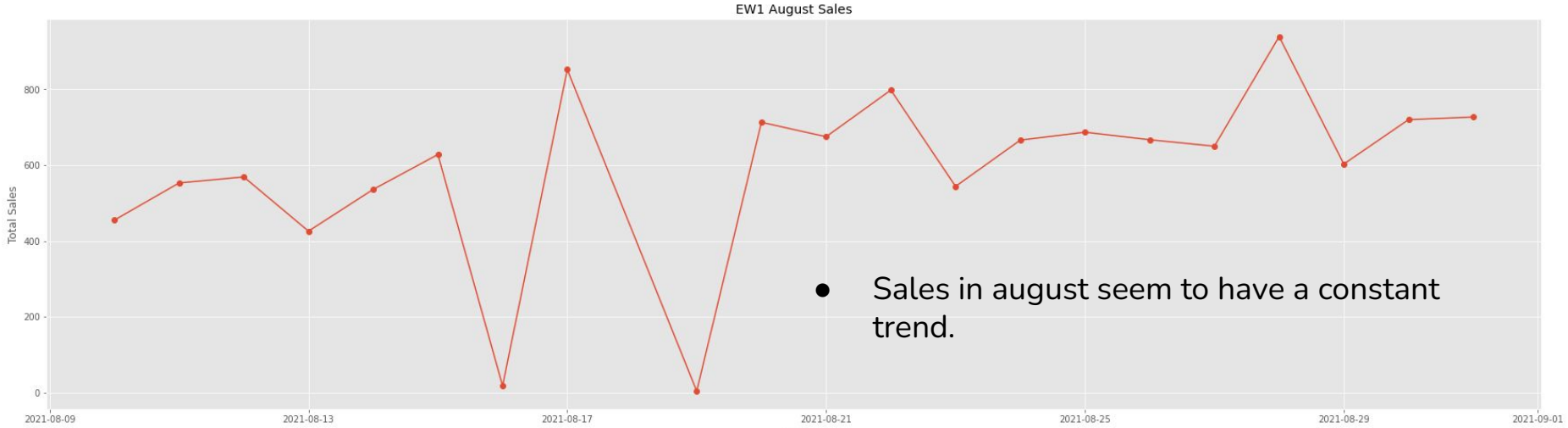


The graph below shows total sales on x-axis and months on y-axis for EW2

I tried to find if there was a relationship between weeks and sales. This graph shows an upward trend for EW2, indicating a good location to open the store.

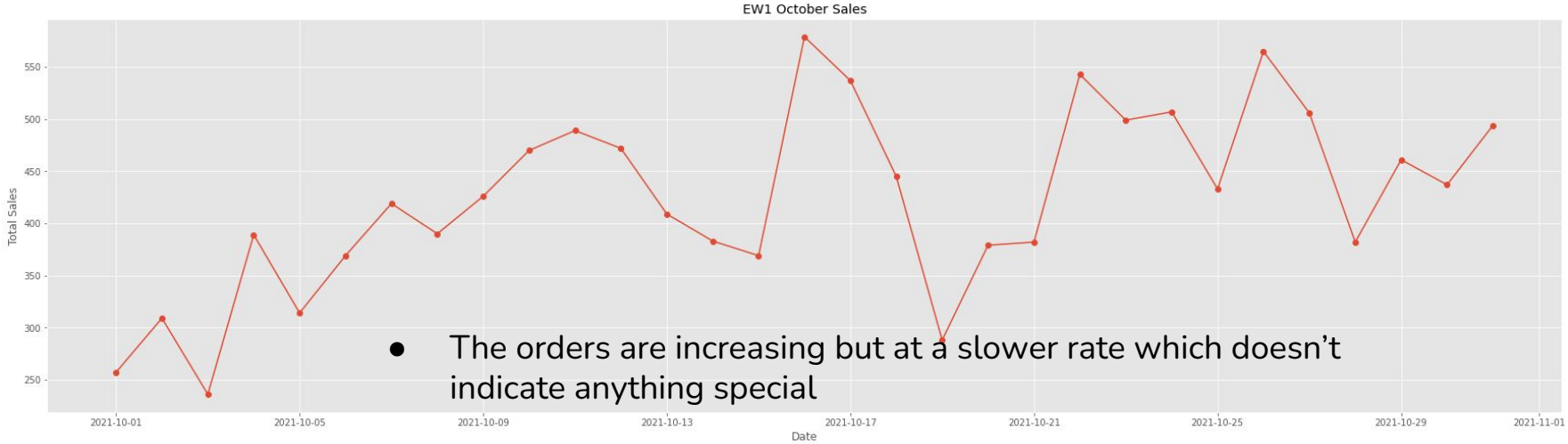


Visualization of EW1 daily sales for each month

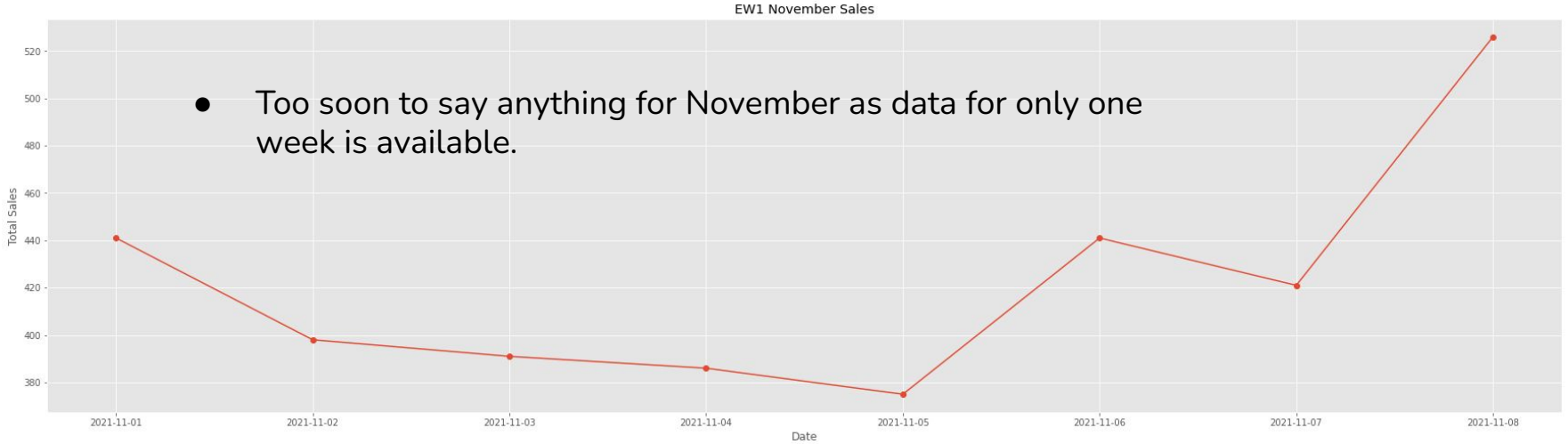


- Sales in august seem to have a constant trend.

- Sales in September can be seen to have a decreasing trend.
- Initial rise in orders could be due to a discount on orders.
- Sudden decrease in orders could be due to shortage of items.



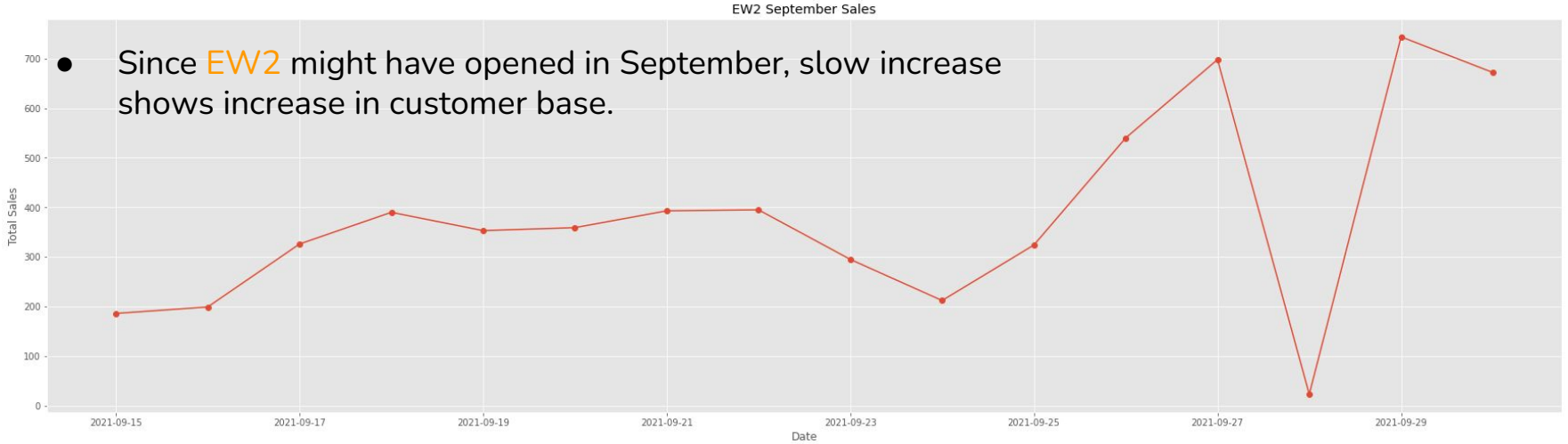
- The orders are increasing but at a slower rate which doesn't indicate anything special



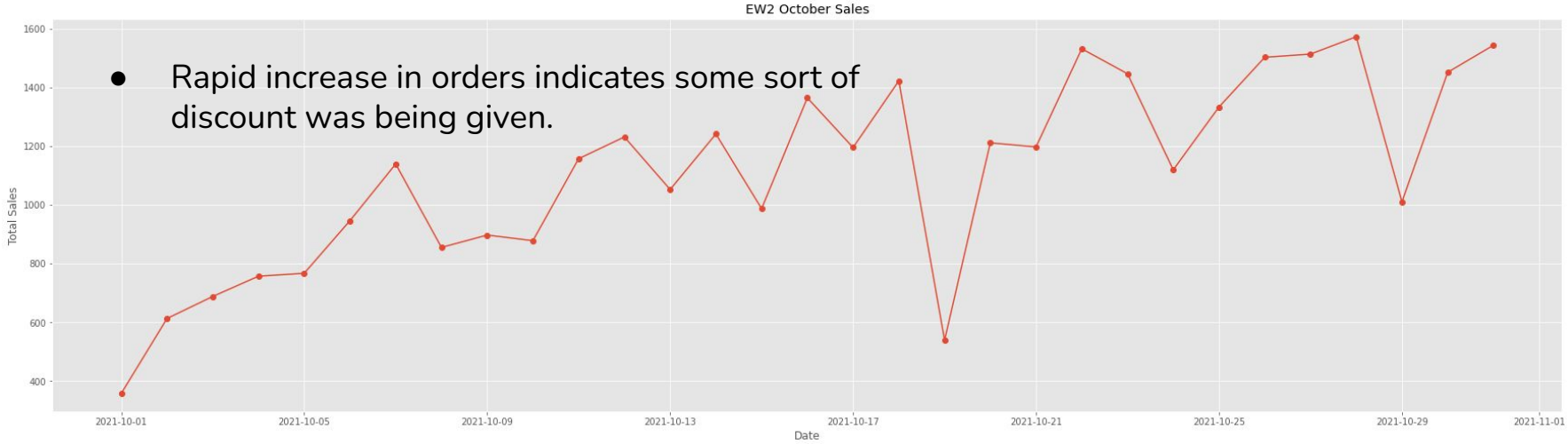
- Too soon to say anything for November as data for only one week is available.

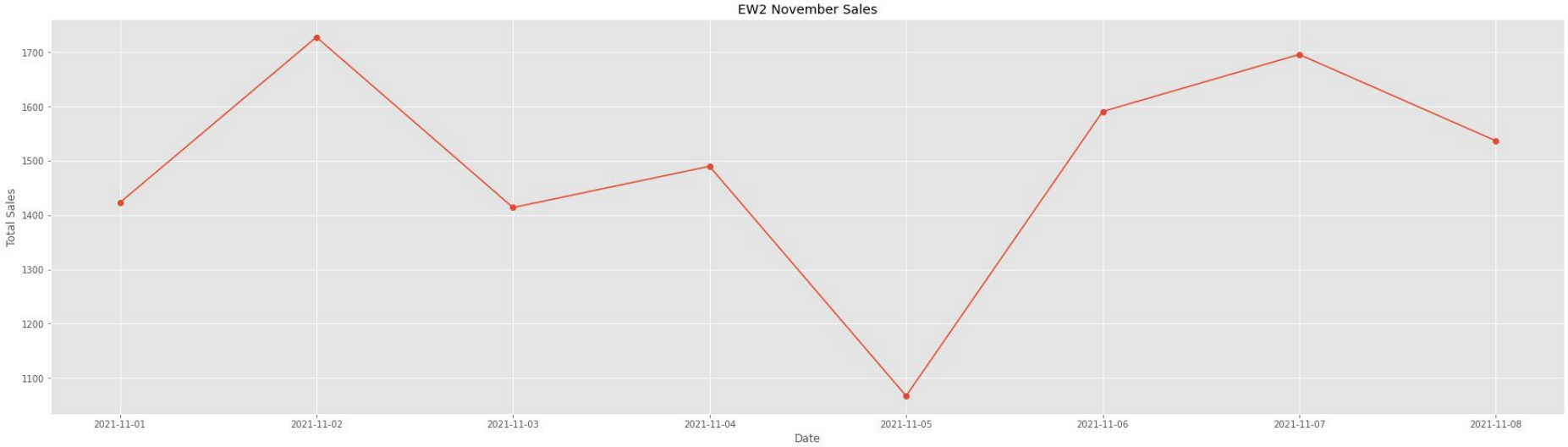
Visualization of EW2 daily sales for each month

- Since EW2 might have opened in September, slow increase shows increase in customer base.



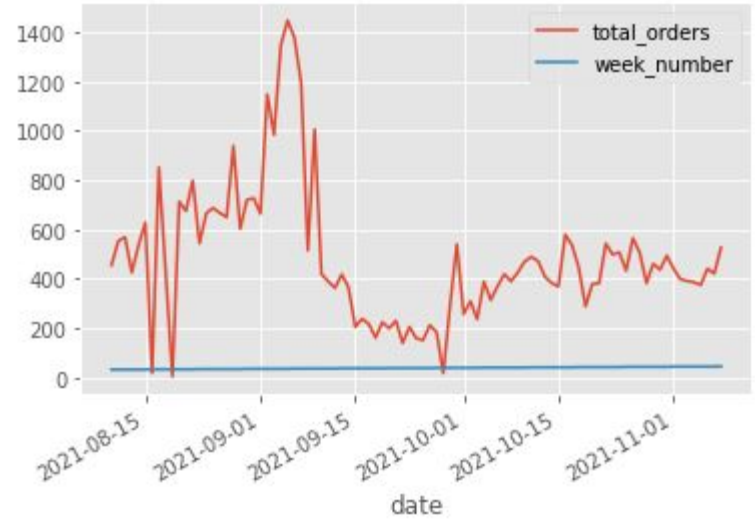
- Rapid increase in orders indicates some sort of discount was being given.



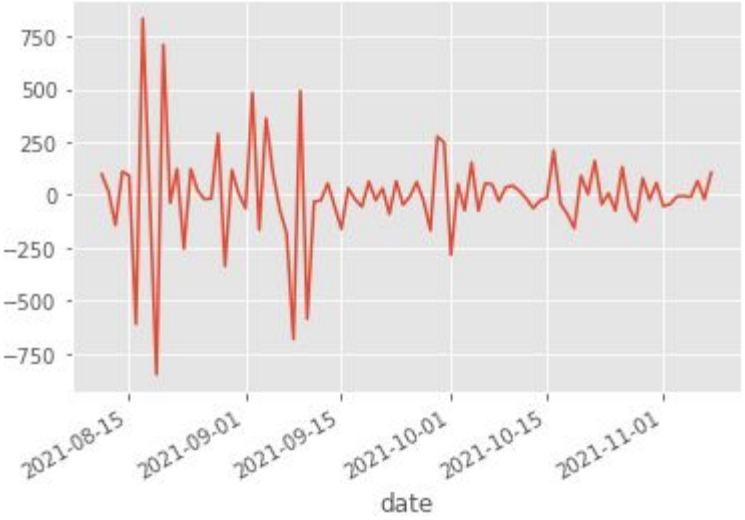


- Too soon to say anything for November as data for only one week is available

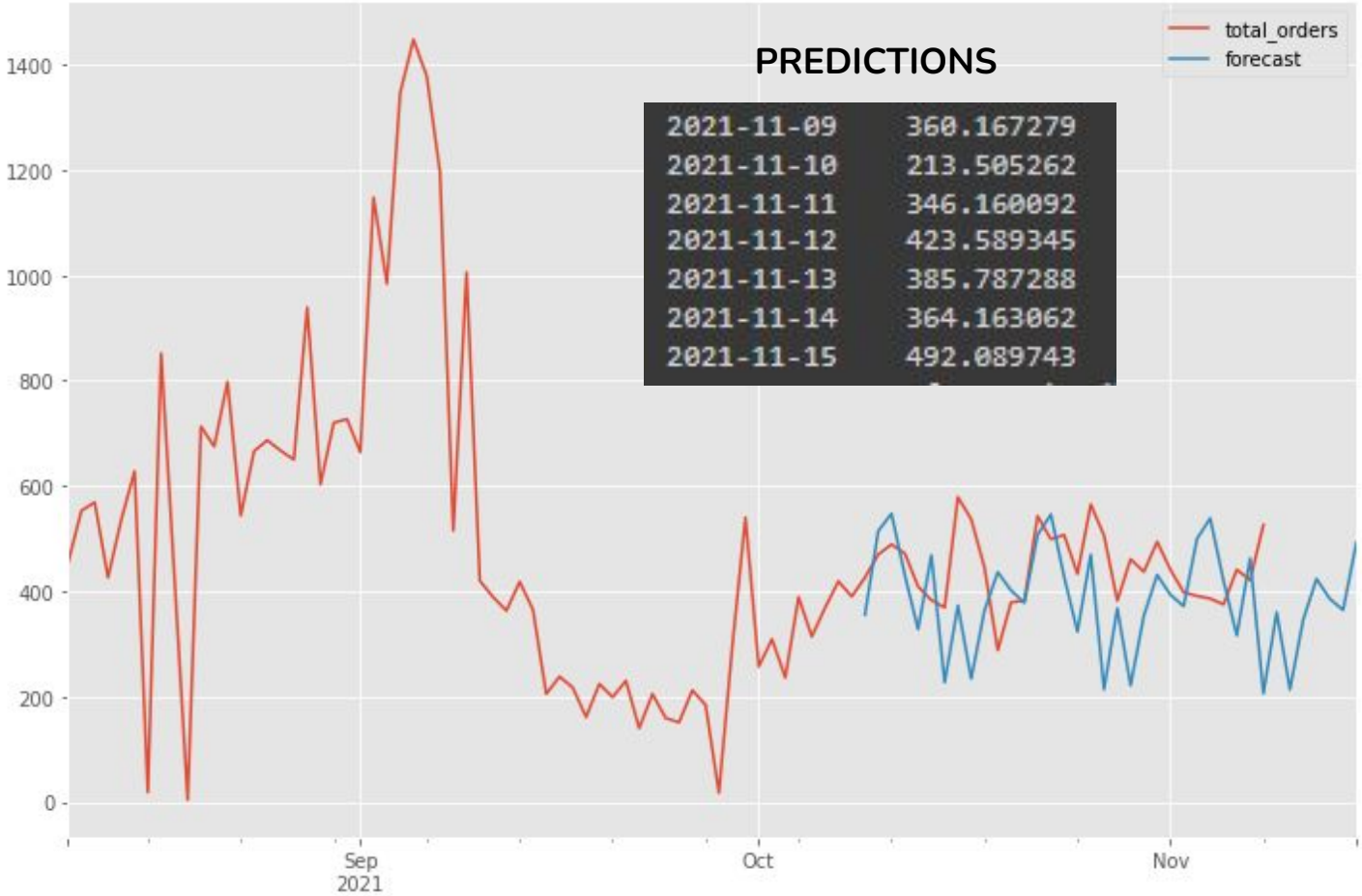
Forecasting EW1 Sales using Sarimax



- Visualising the data of store **EW1**

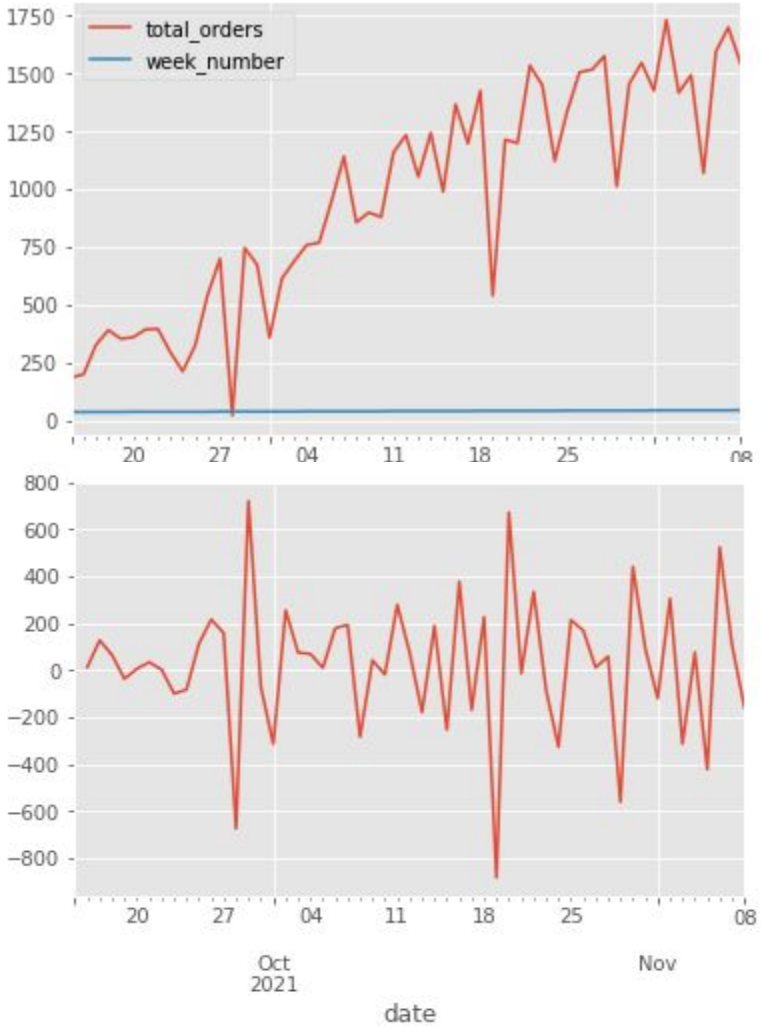


- Since the data in top left image is non-stationary **differencing** technique is used to make it stationary



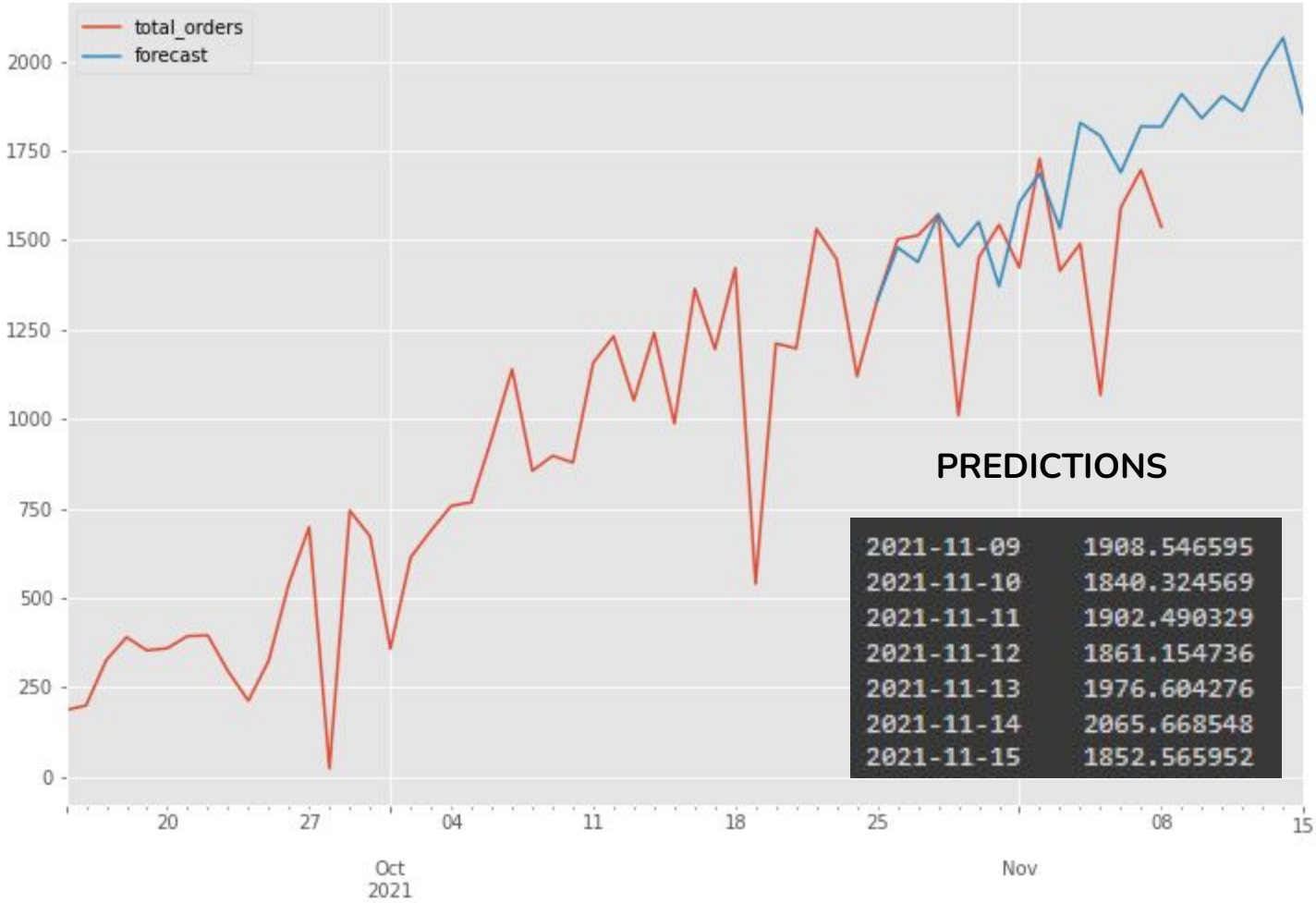
The blue line on the left shows predictions made by the model. The black box displays the results for the second week of November for store **EW1**

Forecasting EW2 Sales using Sarimax



- Visualising the data of store **EW2**

- Since the data in top left image is non-stationary **differencing** technique is used to make it stationary



PREDICTIONS

2021-11-09	1908.546595
2021-11-10	1840.324569
2021-11-11	1902.490329
2021-11-12	1861.154736
2021-11-13	1976.604276
2021-11-14	2065.668548
2021-11-15	1852.565952

The blue line on the left shows predictions made by the model. The black box displays the results for the second week of November for store EW2

Forecasting Accuracy of the model used for predictions

Forecast accuracy of the model

```
forecast_errors = (future_df['total_orders'] - future_df['forecast']).dropna()
forecast_errors = forecast_errors.values.tolist()
forecast_errors_squared = [i ** 2 for i in forecast_errors]
```

```
[265] from statistics import mean
from math import sqrt
mean_squared_error = mean(forecast_errors_squared)
rmse = sqrt(mean_squared_error)
rmse
# mse = mean_squared_error()
```

```
133.75695540277383
```

Forecast accuracy of the model

```
forecast_errors = (future_df['total_orders'] - future_df['forecast']).dropna()
forecast_errors = forecast_errors.values.tolist()
forecast_errors_squared = [i ** 2 for i in forecast_errors]
```

```
[297] from statistics import mean
from math import sqrt
mean_squared_error = mean(forecast_errors_squared)
rmse = sqrt(mean_squared_error)
rmse
# mse = mean_squared_error()
```

```
265.56382883410873
```

Root Mean Squared Error(RSME) has been used for accuracy

The top left image indicates RSME for store **EW1** as the bottom left image indicates RSME for store **EW2**

Some of the dates whose orders were already given were also predicted to calculate accuracy

There are a variety of measures used for calculating the accuracy but RSME is used here in which difference of expected and predicted value is taken which is then squared. After taking out the mean, you square root the result.

Thank you.