

Capstone Project – Walmart Project

Table of Contents

1	Problem Statement
2	Project Objective
3	Data Description
4	Data Pre-processing Steps and Inspiration
5	Choosing the Algorithm for the Project
6	Motivation and Reasons For Choosing the Algorithm
7	Model Evaluation and Techniques
8	Inferences from the Same
9	Future Possibilities of the Project
10	Conclusion
11	References

Problem Statement

Our problem statement is that we have got the dataset of a retail store which has multiple outlets across country and they are facing issues in managing the inventory. They are facing problem to match the demand with respect to supply. So as a data scientist, let's look into data and find some interesting insights to get over with this problem and make prediction models to forecast the sales for next 12 weeks.

Project Objective

From the data we want draw some insights so that the problem retail stores are facing about inventory management will solve. We need to find is there any connection between unemployment and sales. Due to temperature there is any effect on sales or not. Consumer price index (CPI) is affecting on sales or not. Then we need to find which stores are top performing and which are low performing stores and significant difference between them. And we want to forecast sales of stores so that we can manage our inventory and predict about sales in future.

Data Description.

Here we have secondary dataset of different Walmart stores. It consists of features like store which has store numbers, date which is weekly date of sales, weekly sales which is total income of that store in a week, holiday flag which indicates about holiday in that particular week, temperature is about temperature in that area, fuel price, CPI is consumer price index and unemployment is unemployment in that area. So by using above features we can find some insights to get over with their problem.

Data Pre-processing.

Data pre-process step consist of Data cleaning which includes checking missing values, checking outliers and taking proper action on them if they are present in data. Data transformation which includes standardizing or normalization of data. Then data reduction in which we can remove features which are totally useless for our model.

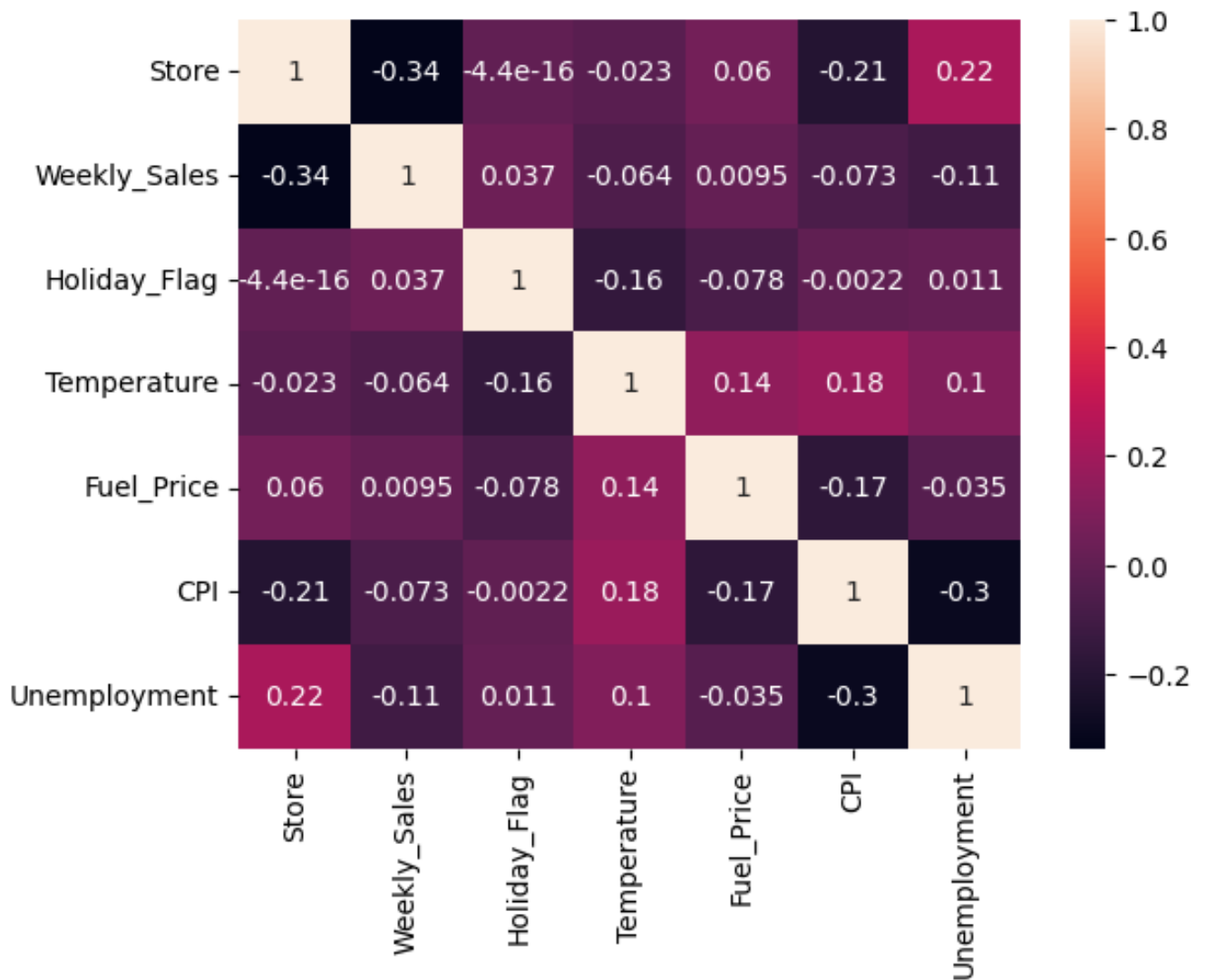
In our Walmart data there are not any missing values. Then there are some outliers in temperature column which we can remove them. Then we are doing standardization to our data. And we don't need to reduce any feature in dataset because our data set is not that large and because we are forecasting sales so we mostly focused on date and weekly sales column.

Choosing the algorithm for project

Here for getting over the problem of management the inventory of stores we are finding some insights using graphical representation. And for forecasting I have used SARIMAX algorithm which is also known as seasonal ARIMA. I have tested ARIMA model at first store but it is not giving me good results because in the data I can see seasonality in plots. So I have followed same steps for all stores and selected best p , q , d values which are seasonality components for SARIMAX model so that I can fit best model for the respective store dataset. In the very beginning I have plotted some nice graphs so that I can get insight of data what are actually important features. And after this management issue can be solved.

Visualization

Heatmap -:



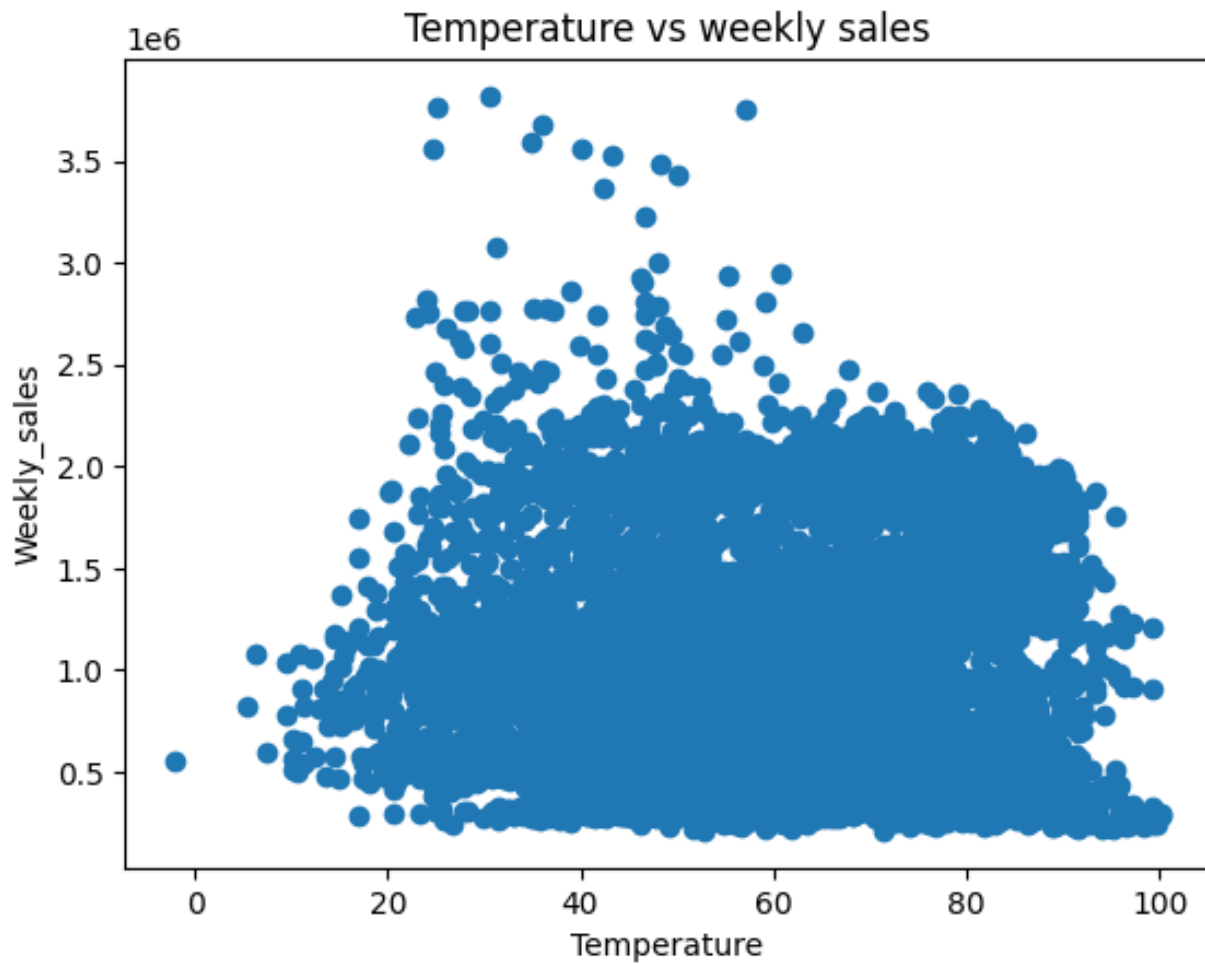
Conclusion- Here we are looking mostly for weekly sales feature correlated with others, so from above plot we can't see much correlation of weekly sales with others. Mostly negative correlation is there. Let's find out with other plots.

Bar Plot -:



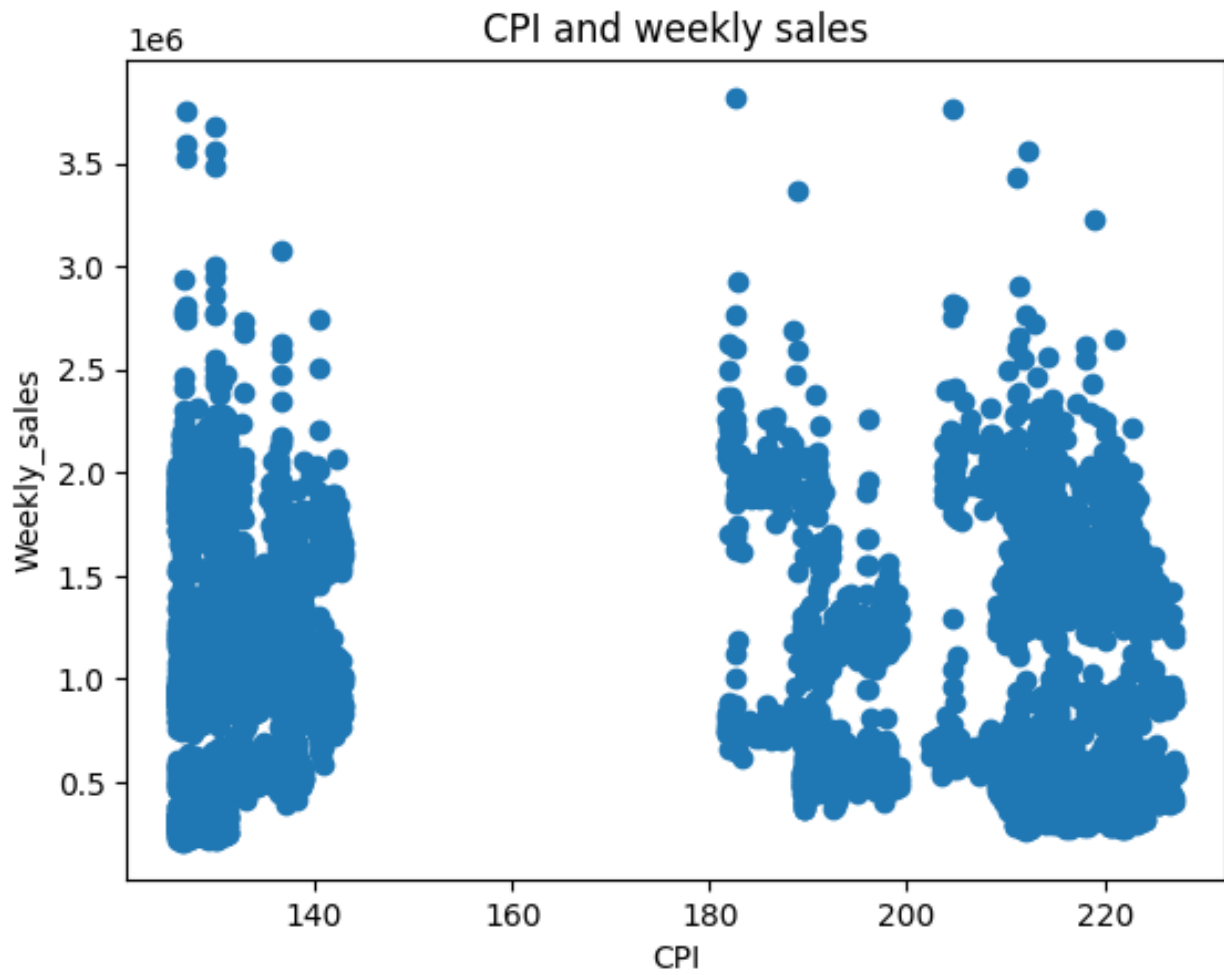
Conclusion- From above plot we can clearly see that when unemployment rate is below 10 sales from store is very high. And as unemployment rate increases sales is getting lower.

Bar Plot -:



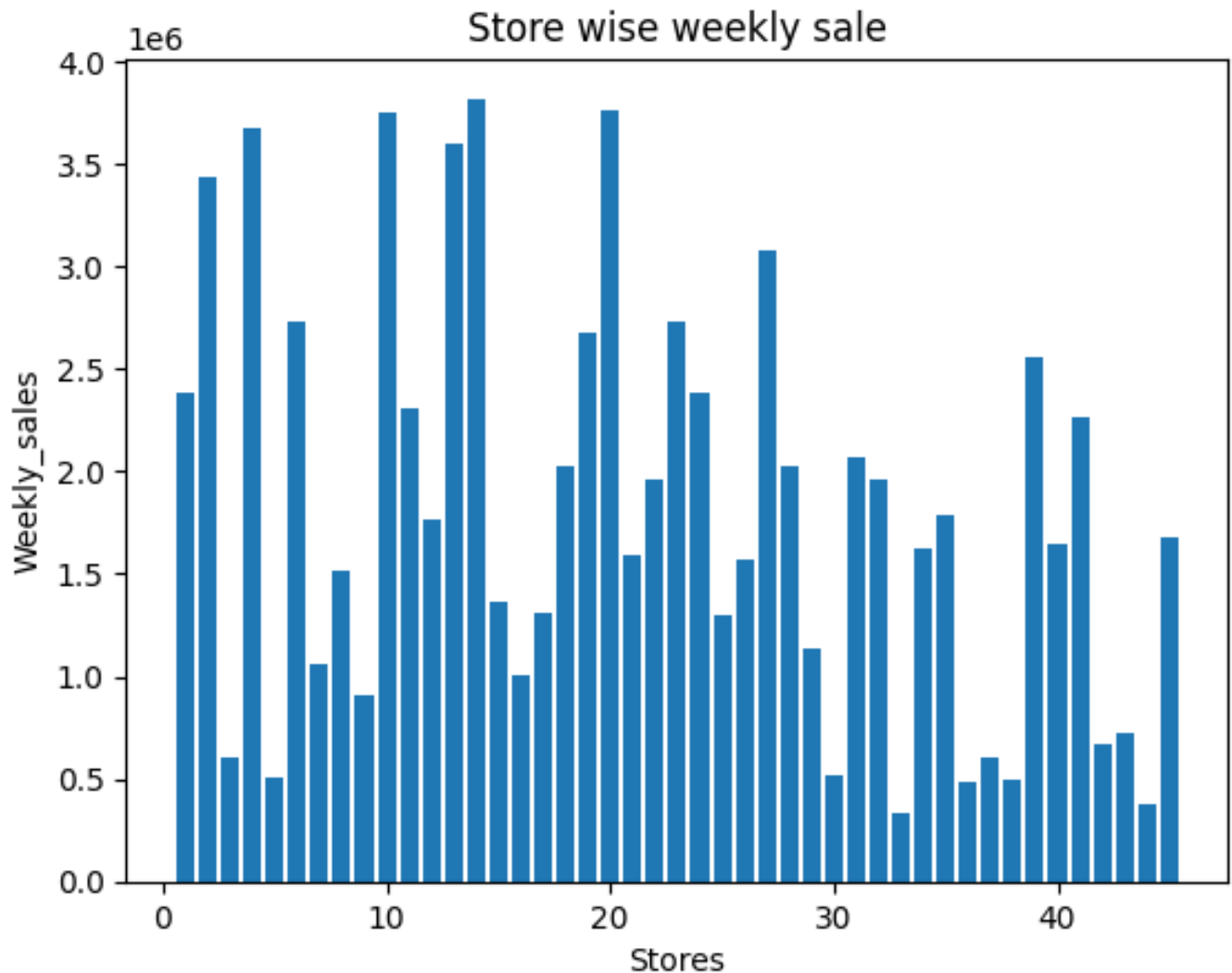
Conclusion- Here we can see that when temperature is low then sale is at its lowest. And as we can see when temperature is just above 20 sale from store is at highest

Bar Plot -:



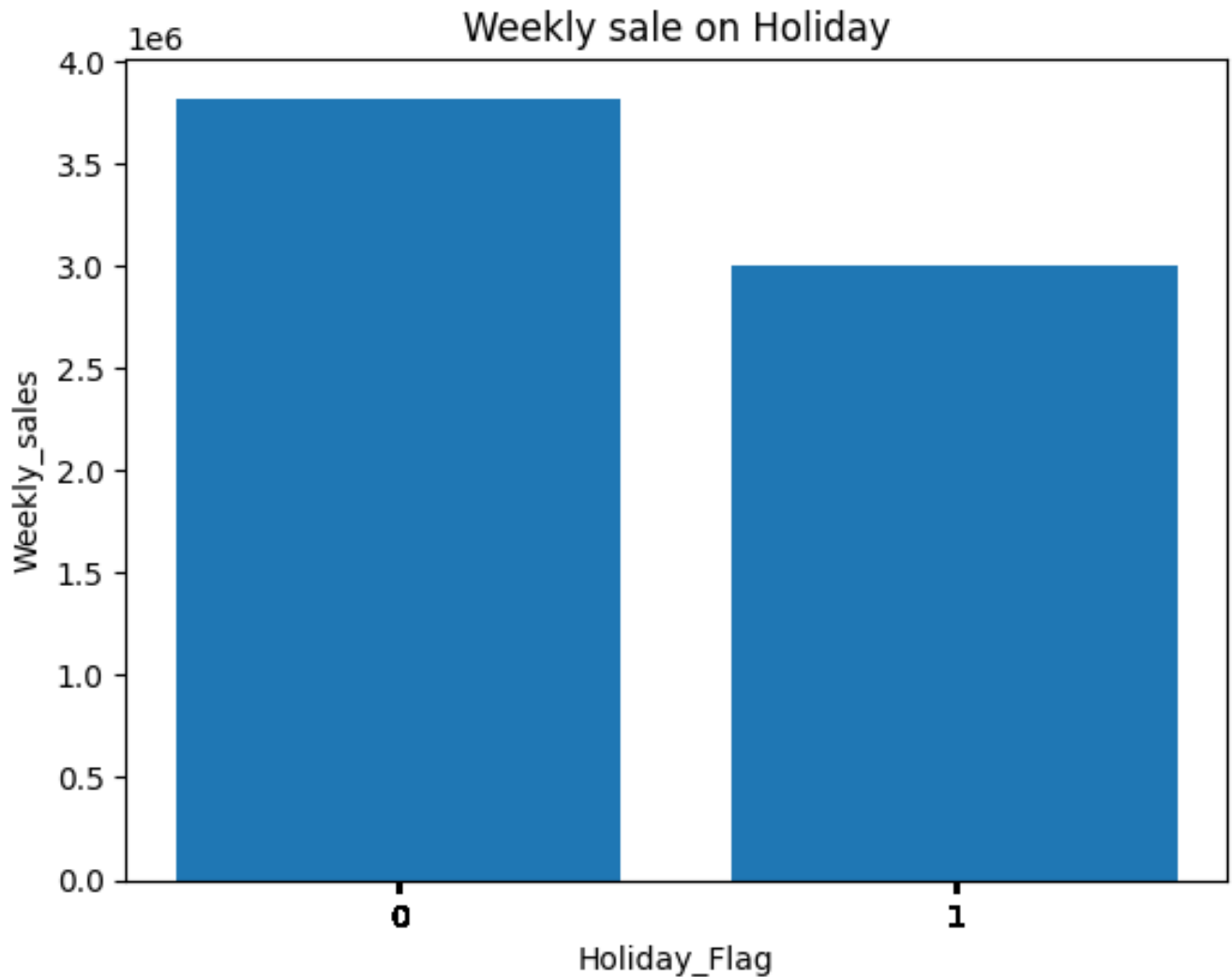
Conclusion- When consumer price index is between 180-200, sale is higher.

Bar Plot :-



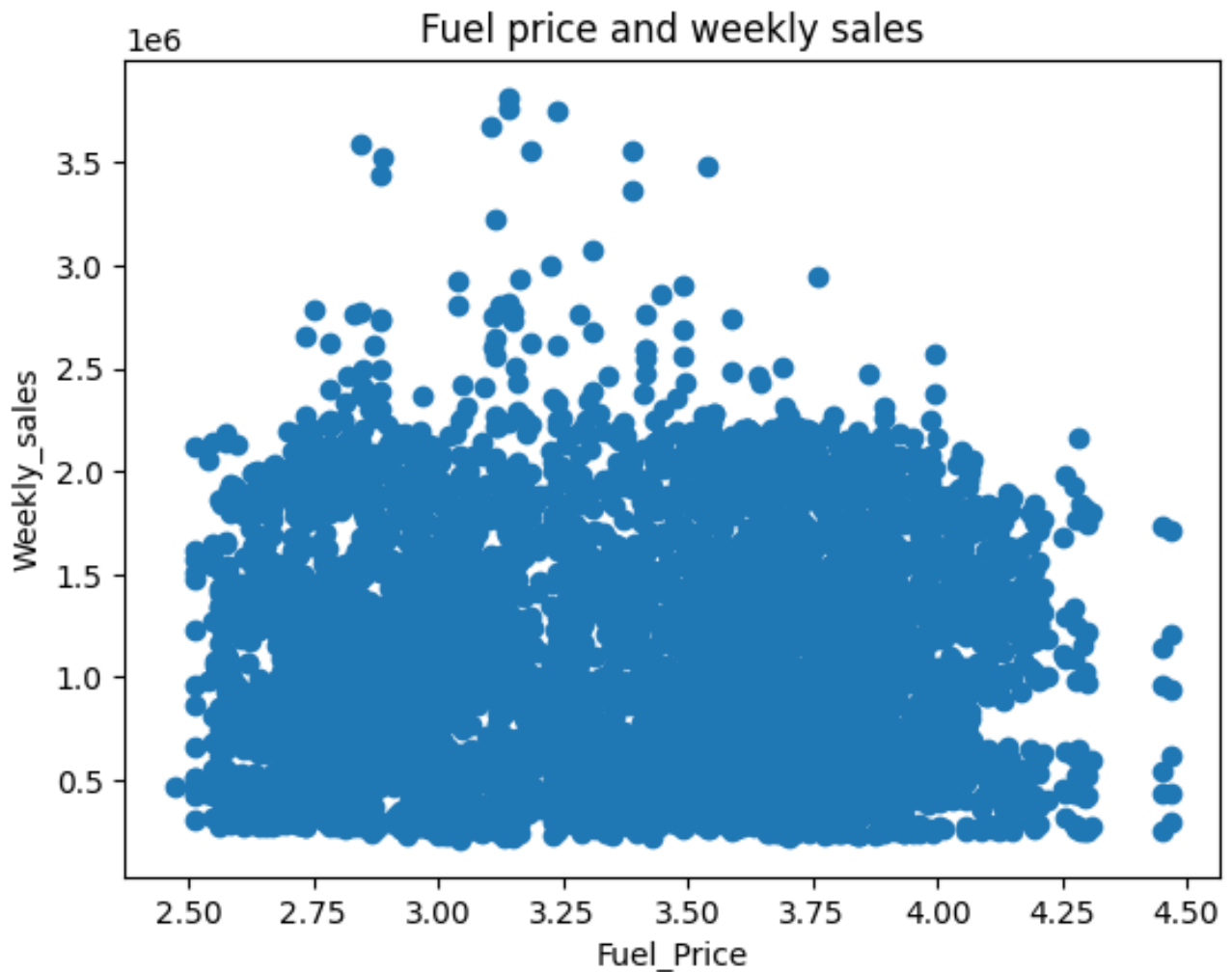
Conclusion- From above bar plot we can see that for each 45 stores how much sales are actually happening. Here store number 14 is making highest sales followed by 10th and 20th store. And 33rd store is making lowest sale followed by store 44th.

Bar Plot -:



Conclusion- Here we can see that how much sales done during other days and holiday where 0 is referred as other days and 1 is for holiday. Surprisingly on holiday, sale is actually low. It seems like families likes to stay at their home only on holiday.

Scatter plot :-

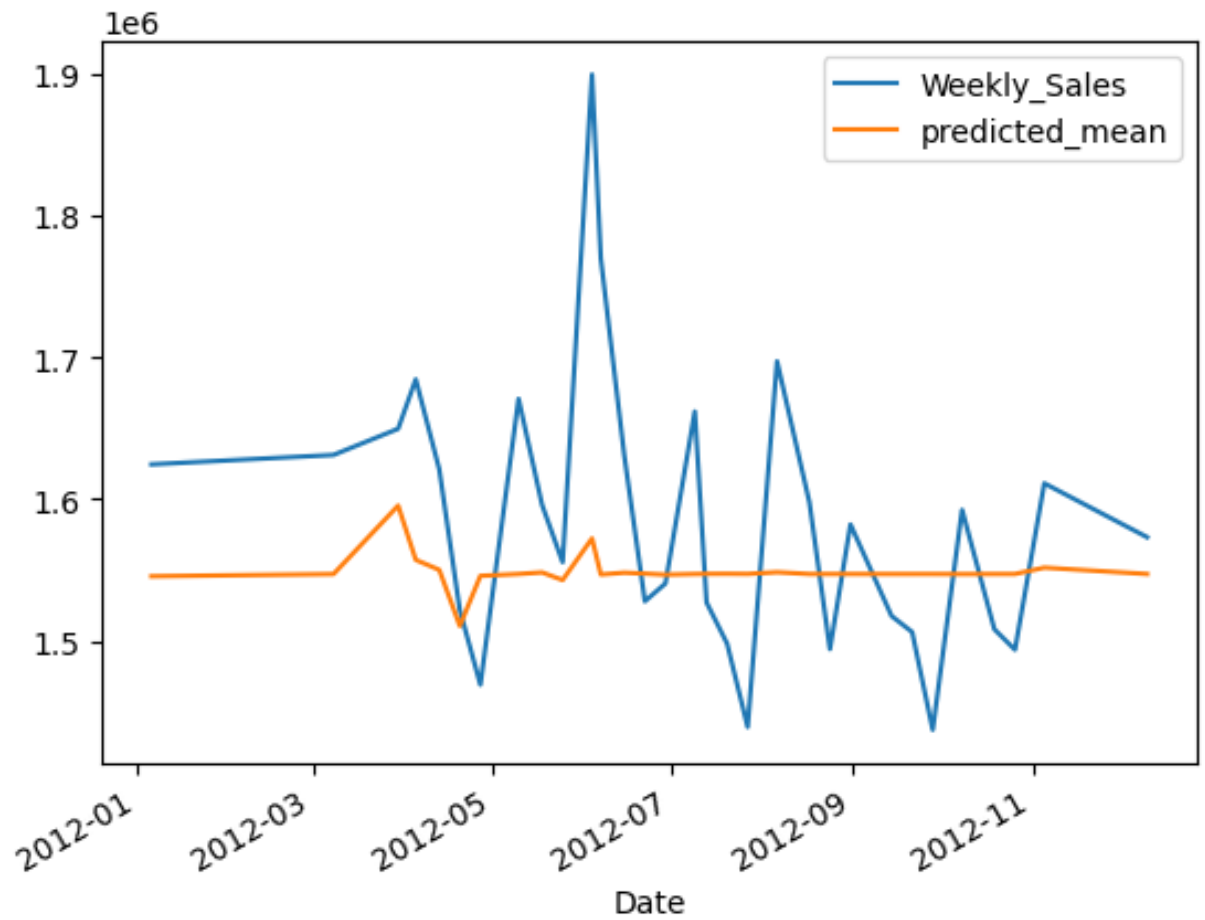


Conclusion- Here I have plotted scatter plot for weekly sales and fuel price to see if there is any connection between them. We can conclude that when fuel price is moderate then sales is at high.

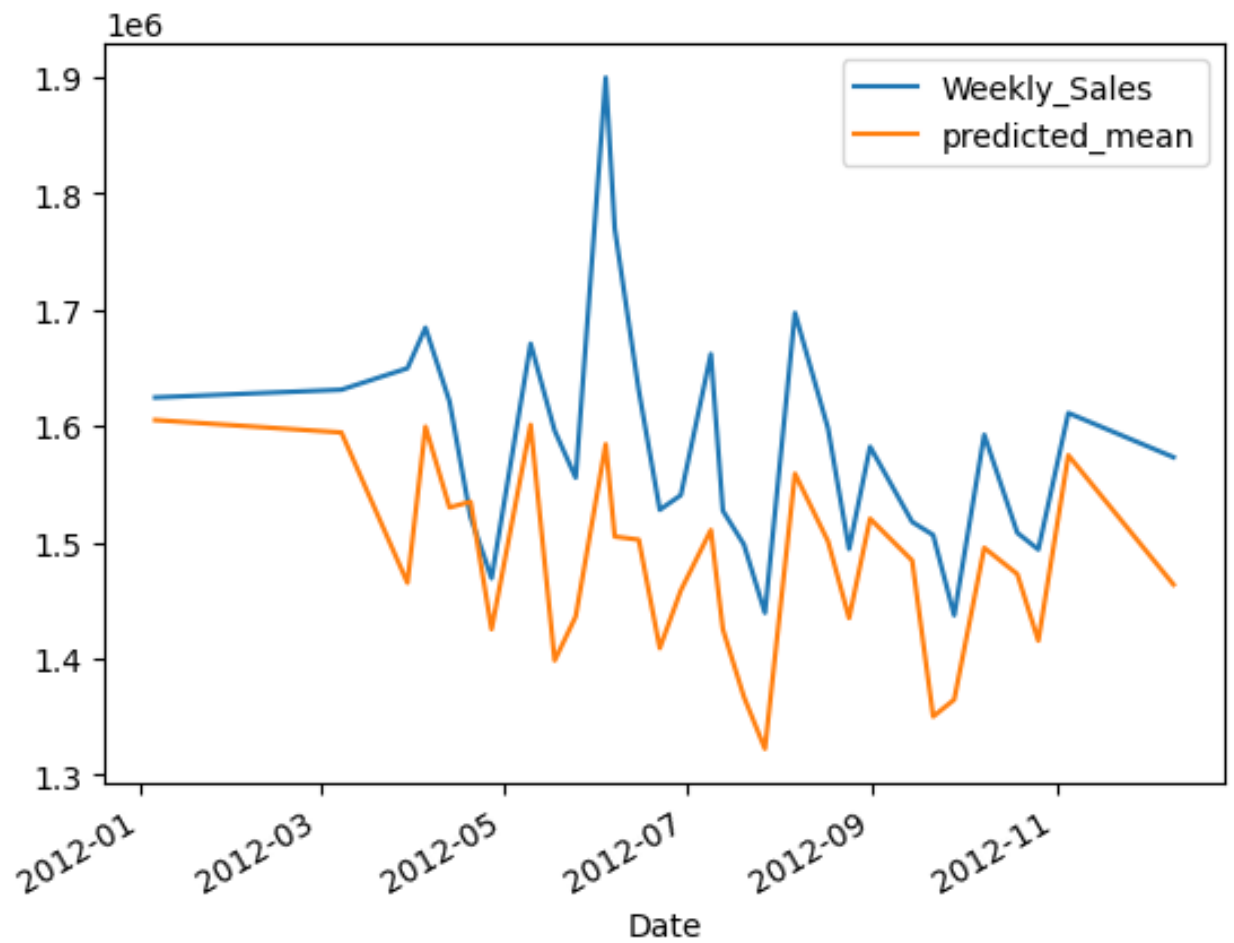
Model building

After checking null, duplicates and outliers values we moved towards actual model building part to forecast sales for next 12 weeks. For that first we need to check data is stationary or not. So we separated 45 stores and for each store we are going to check data is stationary or not using ADF test also known as augmented Dickey Fuller test. It will give p value and if p value is less than 0.05 then data is stationary or vice versa. If data is non stationary then we need to make data stationary using rolling stats method or any other method. Reason behind that are models like arima or sarima models works best on stationary data so need to convert it to stationary data.

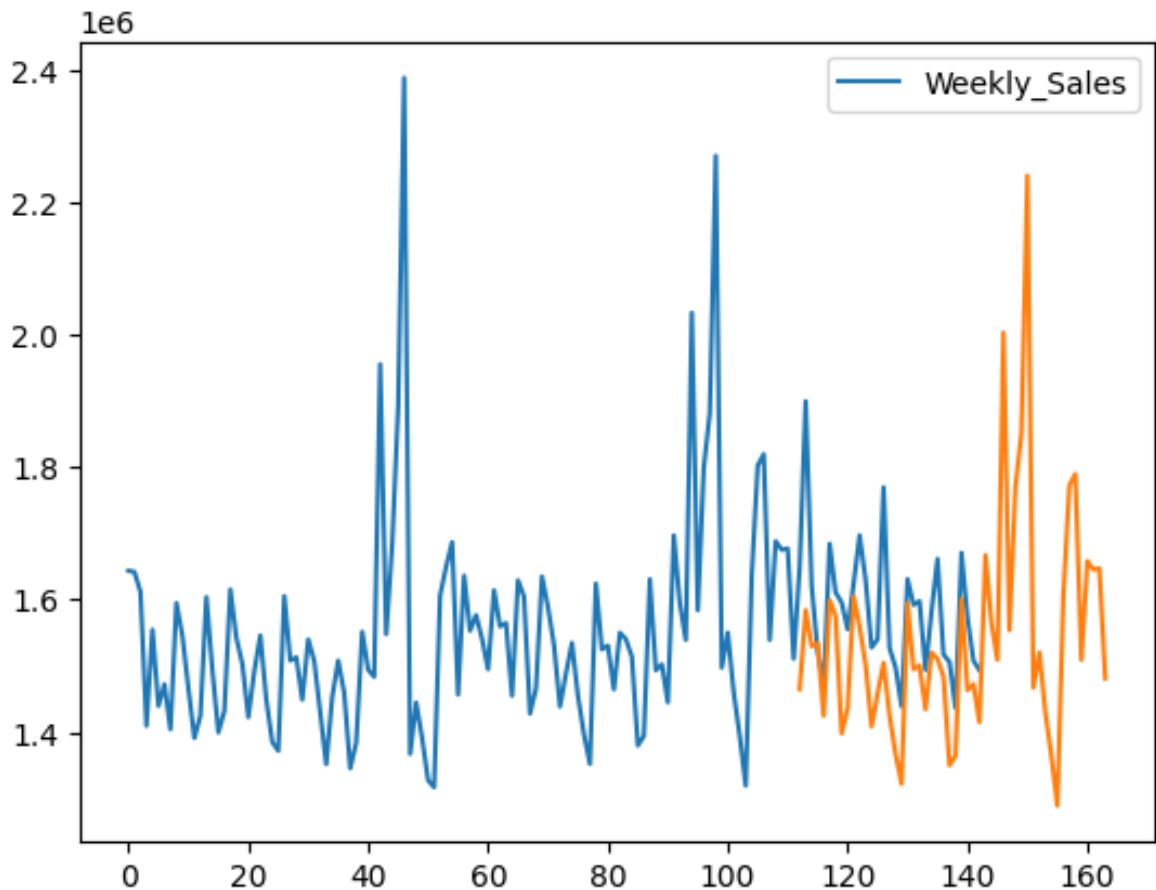
After this we are plotting acf and pacf plots which will gives us parameters like p and q so that we can use it for our model. There are different methods also to get p,d,q values like we can iterate them for range 0 to 8 and check which will fit best to our data. For first I did plot acf and pacf but later I have used iterate method. Then for start we are doing arima model to check the trend of data is matching with forecast or not. If trend is matched with our data then we can forecast for next week's or we can move towards sarima model. So let's build arima model and see if its following trend or not.



As we can see arima is not good for our model. Now we will try sarima model and see if it's getting good trend or not.

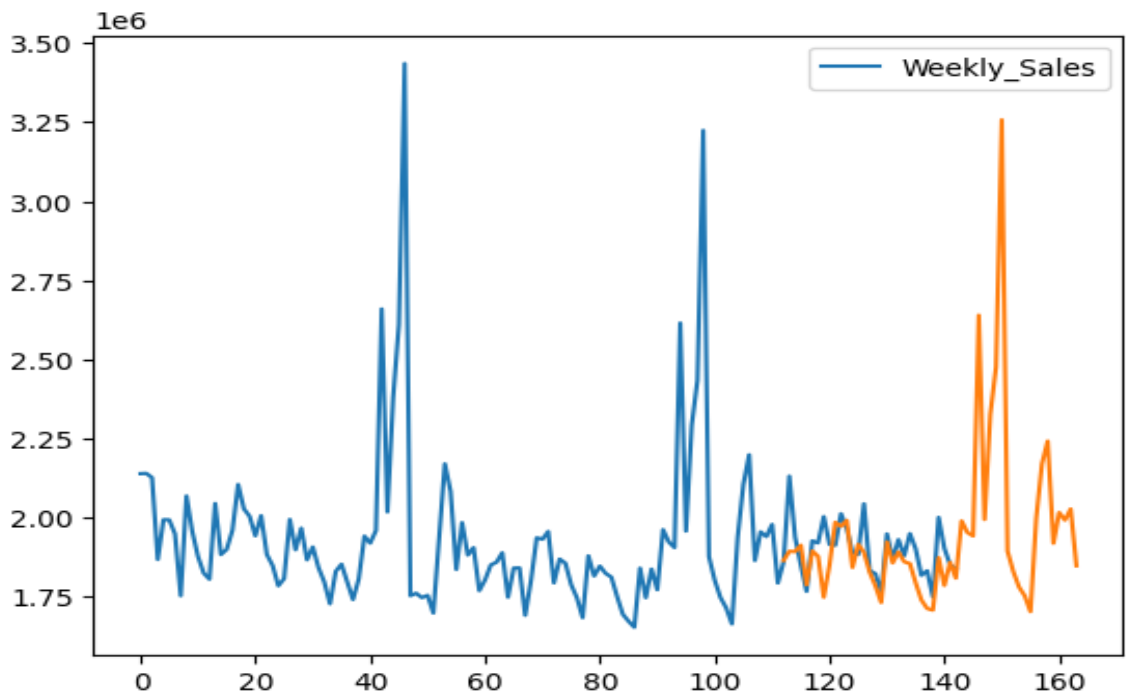


Here we can see very good trend is getting with our actual data.
So here we forecast for next 52 weeks to see better trend.

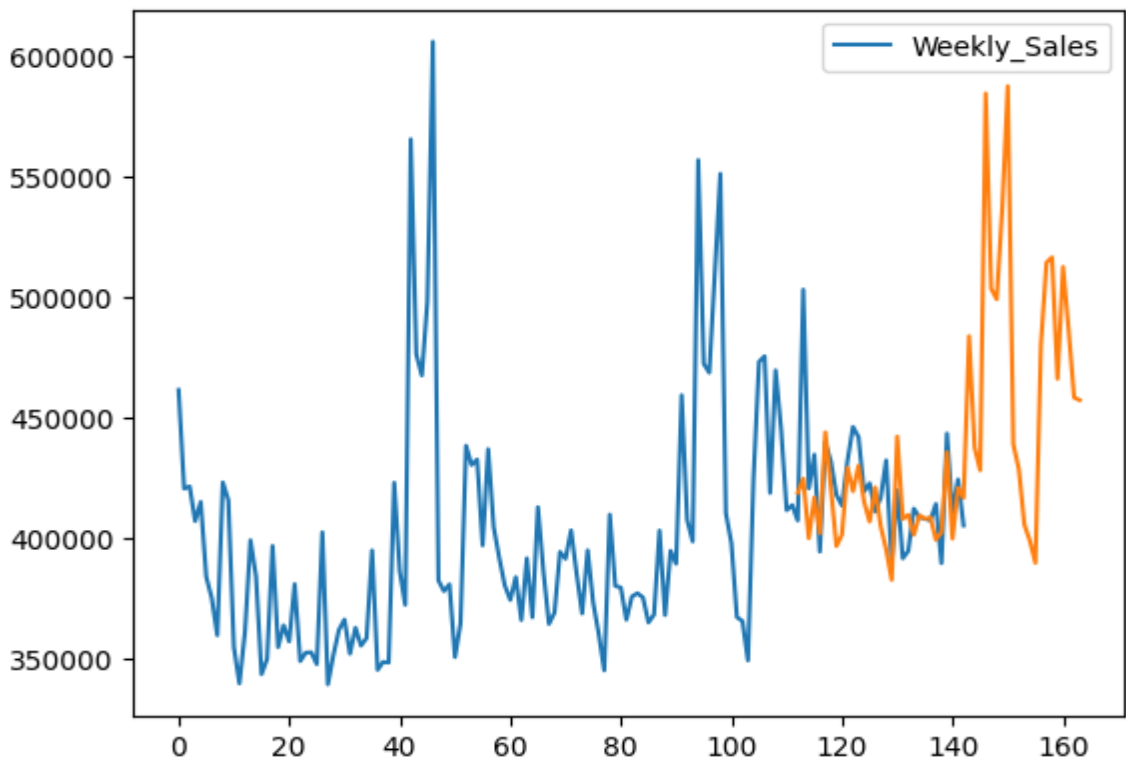


We can see that trend is actually good and forecast is also pretty good. Sarima model worked best for our data as compared to arima model. It's because we can see seasonality in our data so seasonal arima is better option for us now. So for next stores also we are doing sarima model only. So for next stores we will follow same steps and let's see the trend.

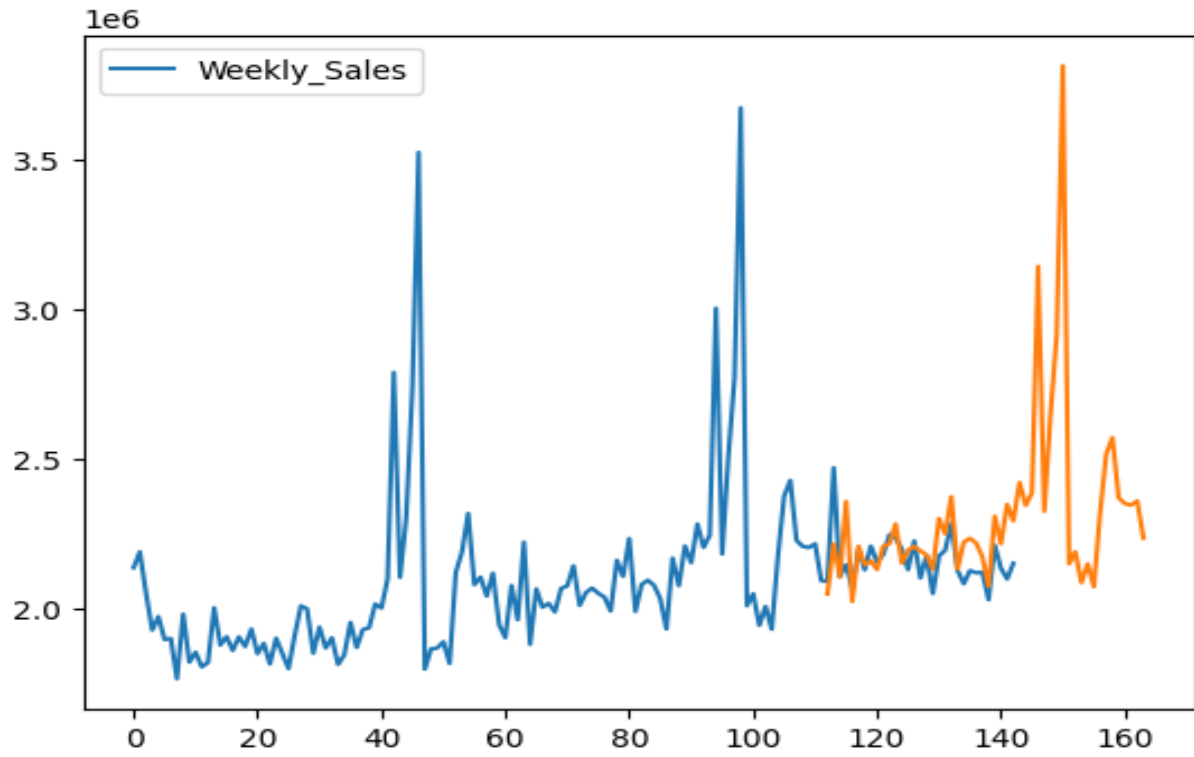
Store 2



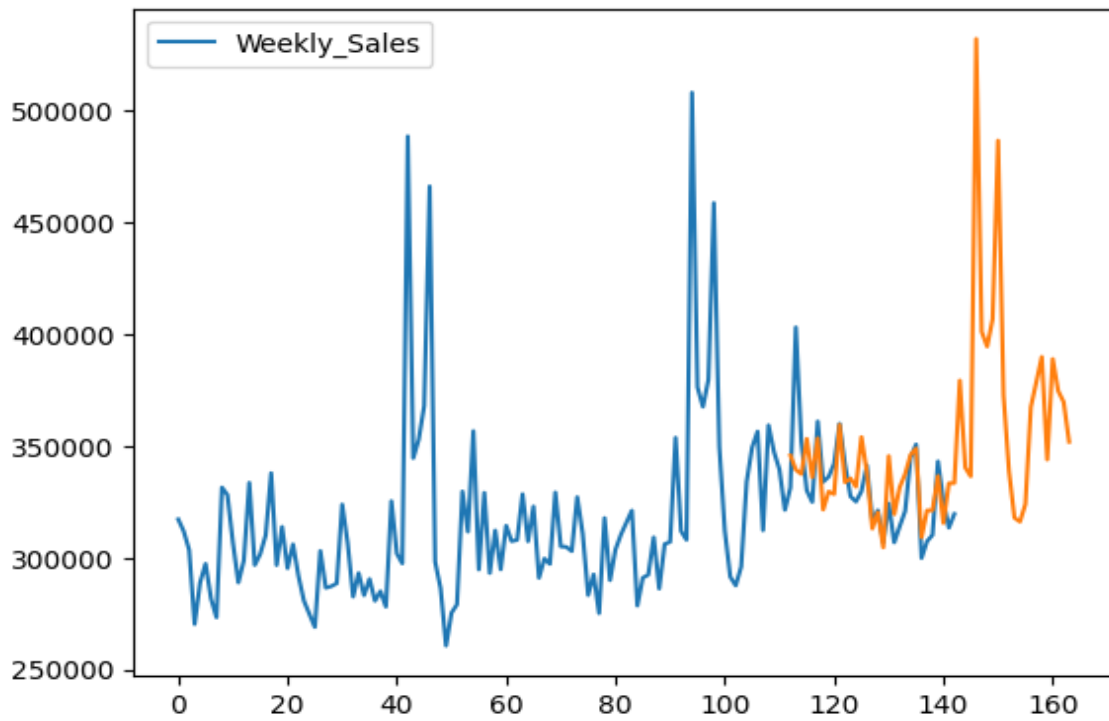
Store 3



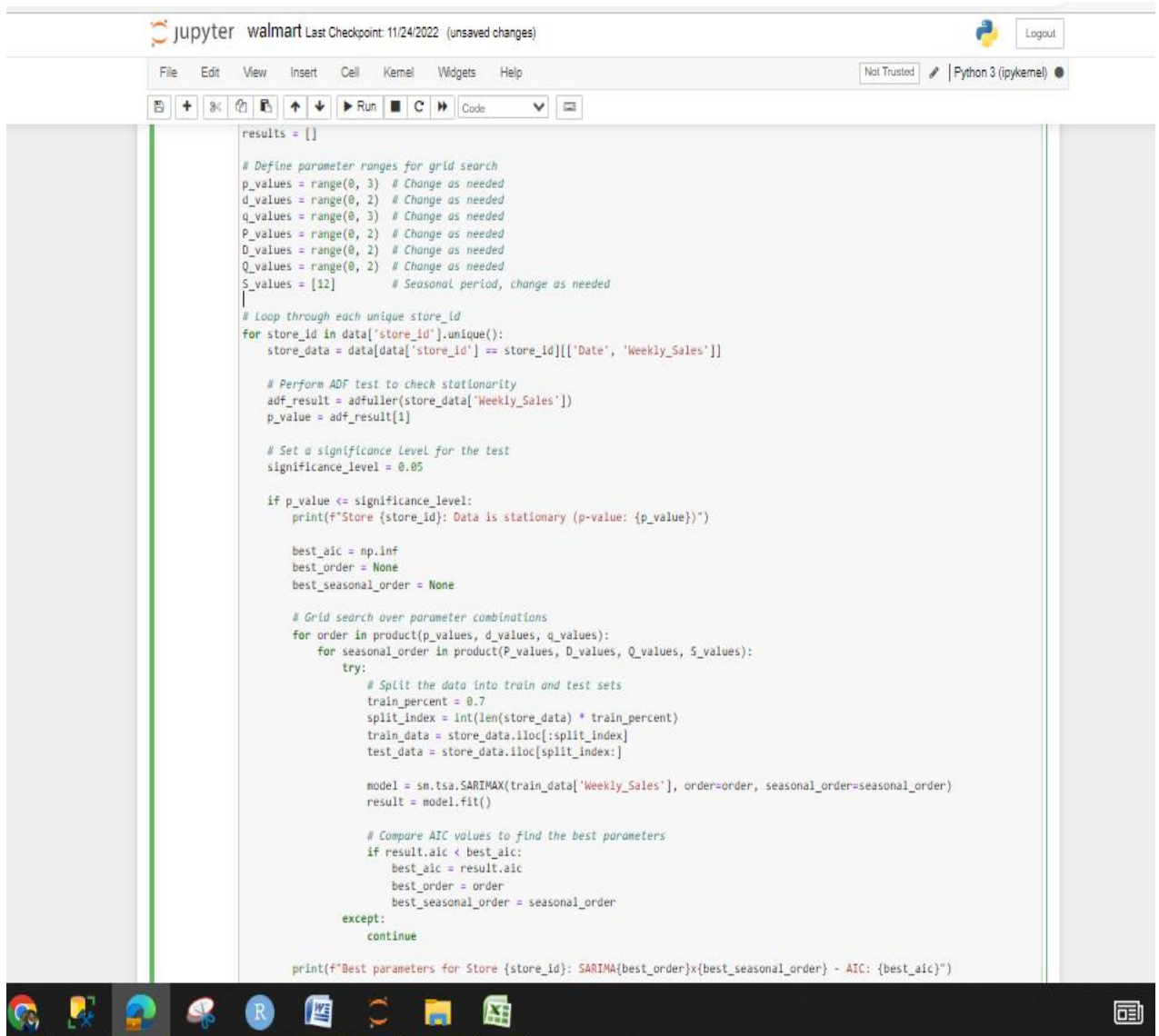
Store 4



Store 5



Loop for best parameters-



```
results = []

# Define parameter ranges for grid search
p_values = range(0, 3) # Change as needed
d_values = range(0, 2) # Change as needed
q_values = range(0, 3) # Change as needed
P_values = range(0, 2) # Change as needed
D_values = range(0, 2) # Change as needed
Q_values = range(0, 2) # Change as needed
S_values = [12] # Seasonal period, change as needed

# Loop through each unique store_id
for store_id in data['store_id'].unique():
    store_data = data[data['store_id'] == store_id][['Date', 'Weekly_Sales']]

    # Perform ADF test to check stationarity
    adf_result = adfuller(store_data['Weekly_Sales'])
    p_value = adf_result[1]

    # Set a significance level for the test
    significance_level = 0.05

    if p_value <= significance_level:
        print(f"Store {store_id}: Data is stationary (p-value: {p_value})")

        best_aic = np.inf
        best_order = None
        best_seasonal_order = None

        # Grid search over parameter combinations
        for order in product(p_values, d_values, q_values):
            for seasonal_order in product(P_values, D_values, Q_values, S_values):
                try:
                    # Split the data into train and test sets
                    train_percent = 0.7
                    split_index = int(len(store_data) * train_percent)
                    train_data = store_data.iloc[:split_index]
                    test_data = store_data.iloc[split_index:]

                    model = sm.tsa.SARIMAX(train_data['Weekly_Sales'], order=order, seasonal_order=seasonal_order)
                    result = model.fit()

                    # Compare AIC values to find the best parameters
                    if result.aic < best_aic:
                        best_aic = result.aic
                        best_order = order
                        best_seasonal_order = seasonal_order
                except:
                    continue

        print(f"Best parameters for Store {store_id}: SARIMA({best_order}x{best_seasonal_order} - AIC: {best_aic})")
```

Here is the loop for finding best parameters of p,d,q for our model

Inference

Here are some major findings from this project as below-

1. When unemployment rate is below 10, sales is higher.
2. Weekly sales shows trend because data is collected in the sequence of weeks. So cycle gets repeated after every week and data shows seasonality.
3. When temperature is just above 20 sales is higher.
4. Store 14 is making highest sales and store 33 is at lowest sale.
5. The significant difference between stores 14th and 33rd sales is approx 251839689.4.
6. On holidays sale from stores is low.
7. When fuel prices are at moderate then sales is at high
8. Sarimax works best when data is seasonal.