SCREENSHOT – RETAIL ANALYSIS WITH WALMART DATA

DATASET:

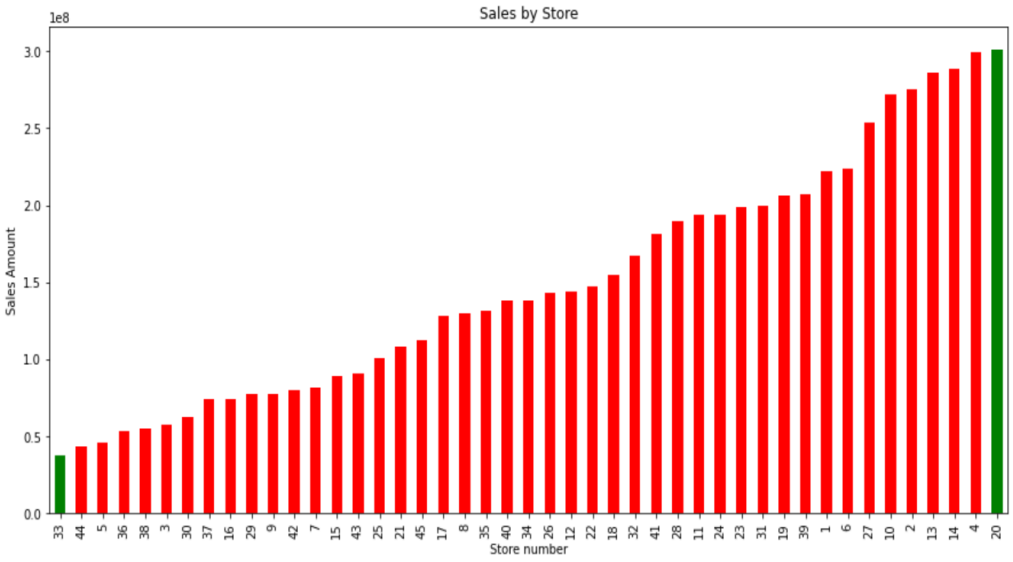
	Store	Date	Weekly_Sales	Holiday_Flag	Temperature	Fuel_Price	CPI	Unemployment
0	1	05-02-2010	1643690.90	0	42.31	2.572	211.096358	8.106
1	1	12-02-2010	1641957.44	1	38.51	2.548	211.242170	8.106
2	1	19-02-2010	1611968.17	0	39.93	2.514	211.289143	8.106
3	1	26-02-2010	1409727.59	0	46.63	2.561	211.319643	8.106
4	1	05-03-2010	1554806.68	0	46.50	2.625	211.350143	8.106
5	1	12-03-2010	1439541.59	0	57.79	2.667	211.380643	8.106
6	1	19-03-2010	1472515.79	0	54.58	2.720	211.215635	8.106
7	1	26-03-2010	1404429.92	0	51.45	2.732	211.018042	8.106
8	1	02-04-2010	1594968.28	0	62.27	2.719	210.820450	7.808
9	1	09-04-2010	1545418.53	0	65.86	2.770	210.622857	7.808
10	1	16-04-2010	1466058.28	0	66.32	2.808	210.488700	7.808
11	1	23-04-2010	1391256.12	0	64.84	2.795	210.439123	7.808
12	1	30-04-2010	1425100.71	0	67.41	2.780	210.389546	7.808
13	1	07-05-2010	1603955.12	0	72.55	2.835	210.339968	7.808
14	1	14-05-2010	1494251.50	0	74.78	2.854	210.337426	7.808
15	1	21-05-2010	1399662.07	0	76.44	2.826	210.617093	7.808
16	1	28-05-2010	1432069.95	0	80.44	2.759	210.896761	7.808
17	1	04-06-2010	1615524.71	0	80.69	2.705	211.176428	7.808

Adding Day, Month, Year:

Out[5]:

	Store	Date	Weekly_Sales	Holiday_Flag	Temperature	Fuel_Price	CPI	Unemployment	Day	Month	Year
0	1	2010-05-02	1643690.90	0	42.31	2.572	211.096358	8.106	2	5	2010
1	1	2010-12-02	1641957.44	1	38.51	2.548	211.242170	8.106	2	12	2010
2	1	2010-02-19	1611968.17	0	39.93	2.514	211.289143	8.106	19	2	2010
3	1	2010-02-26	1409727.59	0	46.63	2.561	211.319643	8.106	26	2	2010
4	1	2010-05-03	1554806.68	0	46.50	2.625	211.350143	8.106	3	5	2010

1) Store having maximum sales:



2) Store with maximum standard deviation:

```
In [8]:  #Store having maximum standard deviation

         sales_std = sales.groupby('Store')['Weekly_Sales'].std().sort_values(ascending=False)
         print("The Store with maximum std deviation is"+ str(sales_std.head(1)))
```

```
The Store with maximum std deviation isStore
14    317569.949476
Name: Weekly_Sales, dtype: float64
```

3) store/s has good quarterly growth rate in Q3'2012

```
In [11]:  #Store having good quarterly growth rate in Q3'2012

          sales_Q3 = sales[(sales['Date']>'2012-07-01')&
                           (sales['Date']< '2012-09-30')].groupby('Store')['Weekly_Sales'].sum().sort_values(ascending=False)
          sales_Q3.head()
```

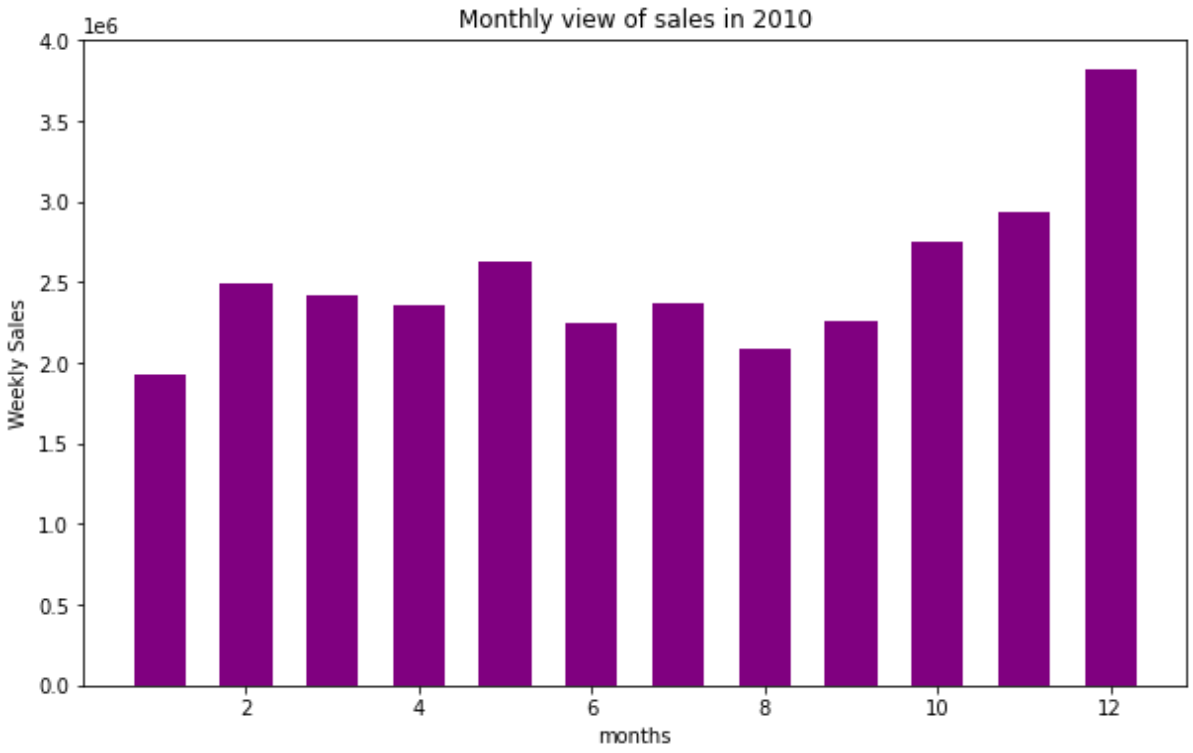
```
Out[11]: Store
         4     25652119.35
         20    24665938.11
         13    24319994.35
         2     22396867.61
         10    21169356.45
         Name: Weekly_Sales, dtype: float64
```

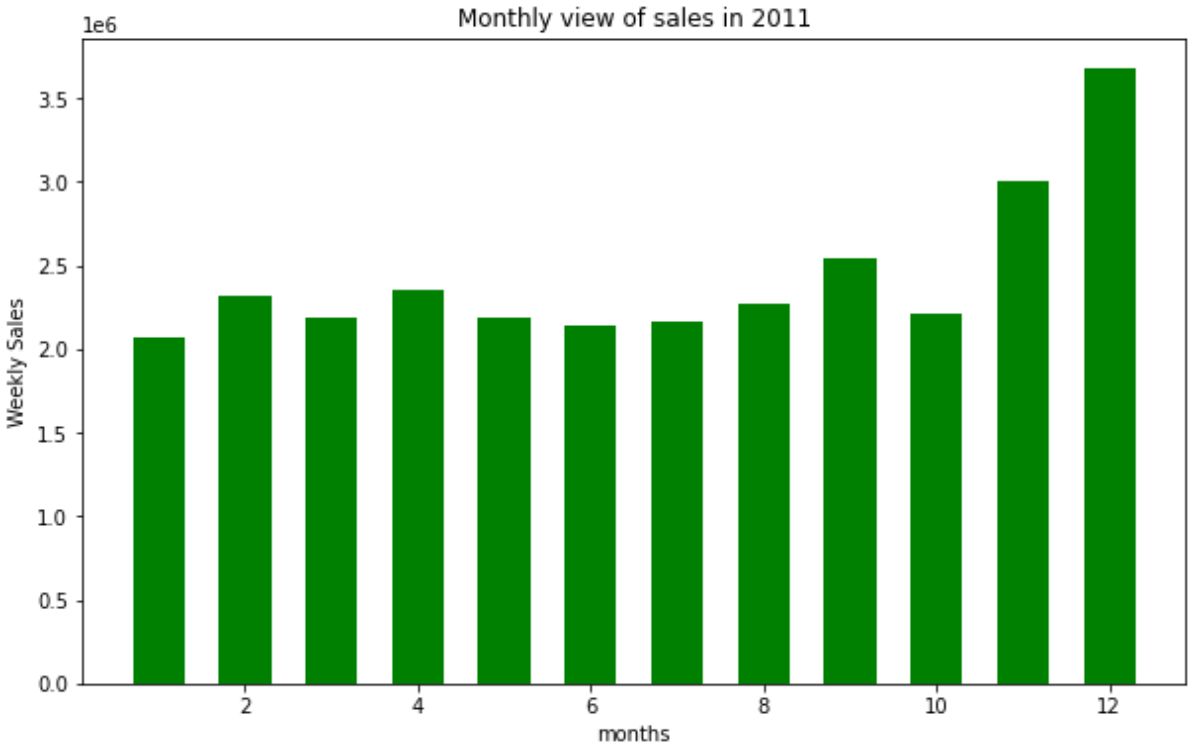
4) Holidays having higher sales than the mean sales in non-holiday season:

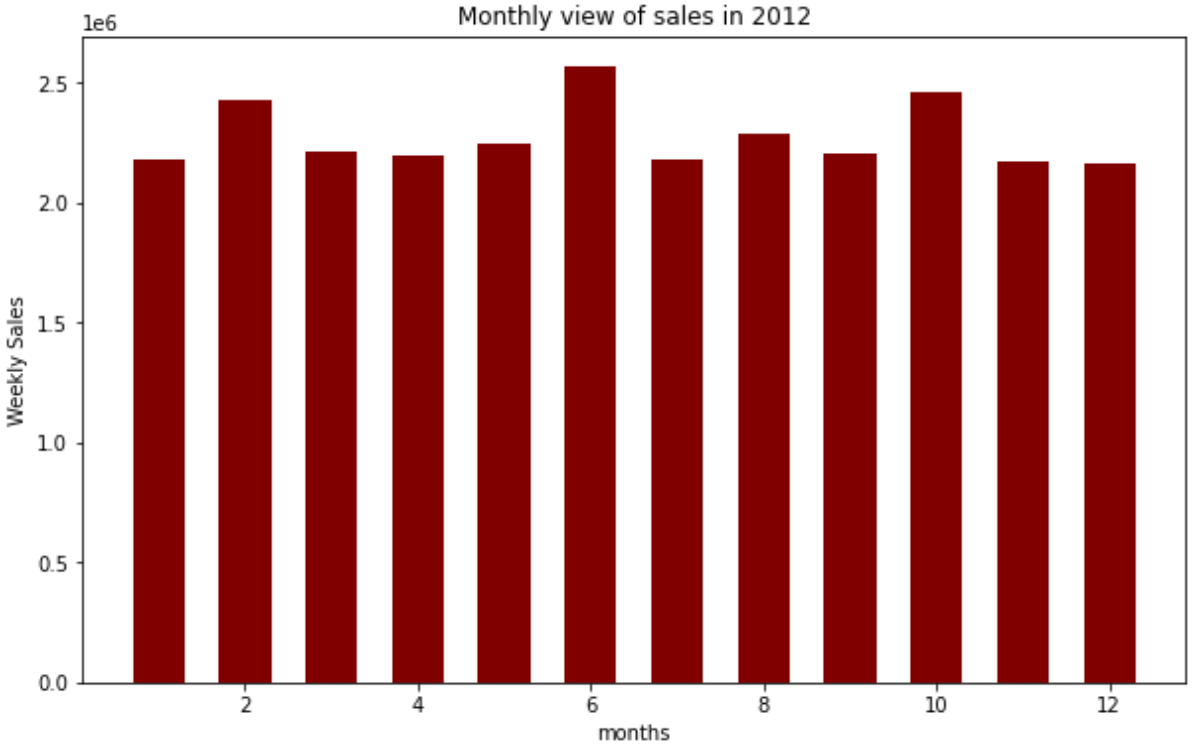
```
In [15]: print("Holidays whose sales are higher than non-holiday sales")
         for X in holidayweek_sale_sum.itertuples():
             for X1 in mean_non_holiday.itertuples():
                 if X.Weekly_Sales > X1.Weekly_Sales:
                     print("On Date {} , sale {}"
                           .format(X.Date,X.Weekly_Sales))
                     break;

Holidays whose sales are higher than non-holiday sales
On Date 2010-10-09 00:00:00 , sale 45634397.839999996
On Date 2010-11-26 00:00:00 , sale 65821003.24
On Date 2010-12-02 00:00:00 , sale 48336677.63
On Date 2010-12-31 00:00:00 , sale 40432519.0
On Date 2011-09-09 00:00:00 , sale 46763227.53
On Date 2011-11-02 00:00:00 , sale 47336192.79
On Date 2011-11-25 00:00:00 , sale 66593605.26
On Date 2011-12-30 00:00:00 , sale 46042461.04
On Date 2012-07-09 00:00:00 , sale 48330059.31
On Date 2012-10-02 00:00:00 , sale 50009407.92
```

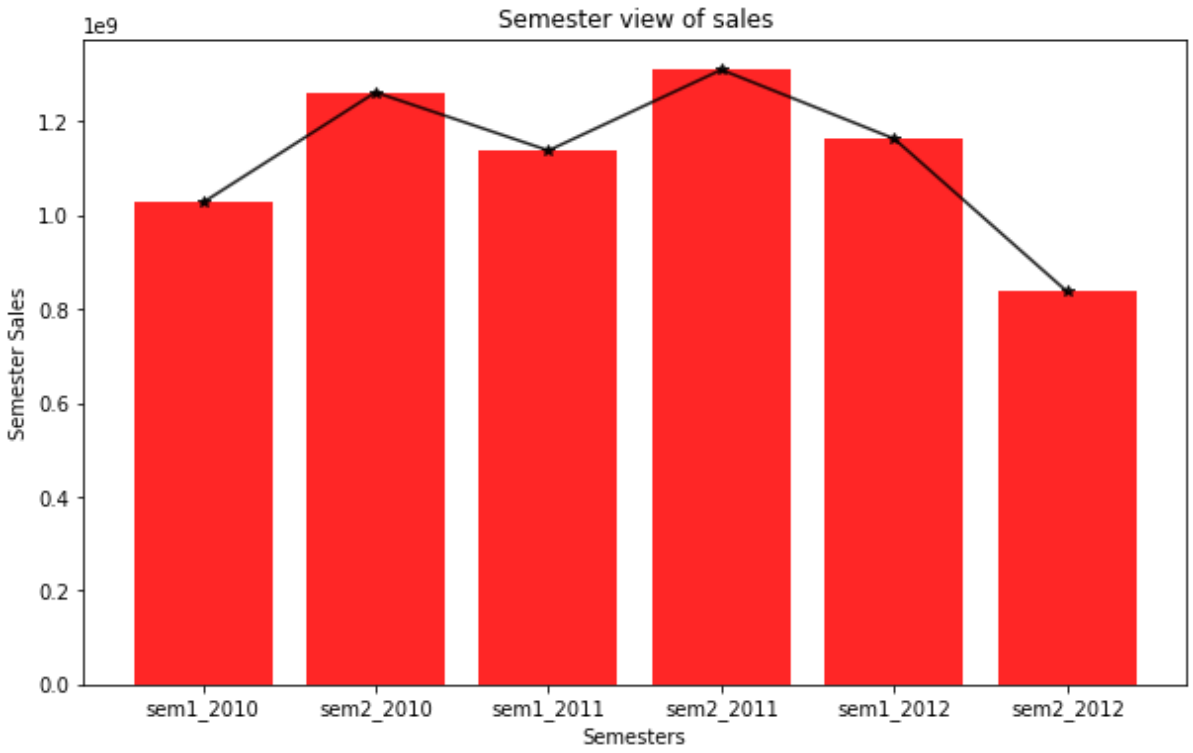
5) Monthly and semester view of sales




Monthly view of sales in 2011
1e6
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Weekly Sales
2
4
6
8
10
12
months


Monthly view of sales in 2012
1e6
2.5
2.0
1.5
1.0
0.5
0.0
Weekly Sales
2
4
6
8
10
12
months

Semester view of sales



Statistical Model

Linear Regression – Store 1

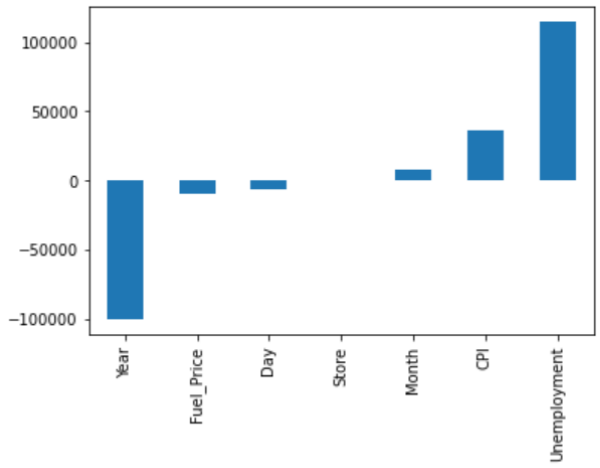
```
In [138]: print('Accuracy:',reg_all.score(X_train, y_train)*100)
          print('Mean Absolute Error:', metrics.mean_absolute_error(y_test, y_pred))
          print('Mean Squared Error:', metrics.mean_squared_error(y_test, y_pred))
          print('Root Mean Squared Error:', np.sqrt(metrics.mean_squared_error(y_test, y_pred)))

          Accuracy: 26.428789286627683
          Mean Absolute Error: 78781.91416420814
          Mean Squared Error: 12771481727.018492
          Root Mean Squared Error: 113010.98055949471
```

Relationship b/w feature and target variables:

```
In [137]: relation=pd.Series(reg_all.coef_,x.columns).sort_values()
          relation.plot(kind="bar")

Out[137]: <AxesSubplot:>
```



Polynomial Regression:

```
#shape of features in Polyreg
print(X_feature.shape)
print(X_feature_PolyReg.shape)


from math import sqrt
from sklearn. metrics import r2_score
from sklearn.metrics import mean_squared_error

#Root mean square error value and score(Accuracy) calculation
r_squared = r2_score(y_target, Ypipehat)
print('R-squared :', r_squared)
print('RMSE:' , sqrt(mean_squared_error(y_target,Ypipehat)))
```

```
Pipeline(steps=[('scale', StandardScaler()),
                ('polynomial', PolynomialFeatures(include_bias=False)),
                ('model', LinearRegression())])
[[1.000000e+00 1.671584e+06]
 [1.000000e+00 1.568736e+06]
 [1.000000e+00 1.635136e+06]
 [1.000000e+00 1.435456e+06]]
(143, 4)
(143, 35)
R-squared : 0.8509254518051932
RMSE: 60013.50032850766
```

Changing Dates into Days:

In [99]:
```
#change of dates to days
sales['Days'] = pd.to_datetime(sales['Date']).dt.day_name()
sales
```

Out[99]:

	Store	Date	Weekly_Sales	Holiday_Flag	Temperature	Fuel_Price	CPI	Unemployment	Day	Month	Year	Days
0	1	2010-05-02	1643690.90	0	42.31	2.572	211.096358	8.106	2	5	2010	Sunday
1	1	2010-12-02	1641957.44	1	38.51	2.548	211.242170	8.106	2	12	2010	Thursday
2	1	2010-02-19	1611968.17	0	39.93	2.514	211.289143	8.106	19	2	2010	Friday
3	1	2010-02-26	1409727.59	0	46.63	2.561	211.319643	8.106	26	2	2010	Friday
4	1	2010-05-03	1554806.68	0	46.50	2.625	211.350143	8.106	3	5	2010	Monday
...	...	...	...	...	...	...	...	...	...	...	...	...
6430	45	2012-09-28	713173.95	0	64.88	3.997	192.013558	8.684	28	9	2012	Friday
6431	45	2012-05-10	733455.07	0	64.89	3.985	192.170412	8.667	10	5	2012	Thursday
6432	45	2012-12-10	734464.36	0	54.47	4.000	192.327265	8.667	10	12	2012	Monday
6433	45	2012-10-19	718125.53	0	56.47	3.969	192.330854	8.667	19	10	2012	Friday
6434	45	2012-10-26	760281.43	0	58.85	3.882	192.308899	8.667	26	10	2012	Friday

6435 rows × 12 columns