IS THE STORE A STARGET?

STAR STORES THAT ACHIEVED THE TARGETS

Shagil Chaudhary
shagil.chaudhary@yahoo.in
Computer Science Department
Jamia Hamdard University
New Delhi, India

Shouvik Dasgupta
shouvikdasgupta3125@gmail.com
Computer Science Department
Jamia Hamdard University
New Delhi, India

INTRODUCTION

In the world of retail, store owners face the challenge of how to use the wealth of data they have at their fingertips. After all, the data you're sitting on has the ability to give your small business a competitive advantage. This is especially important because there are many moving parts in your store, from sales inventory to customer experience and everything in between. Retail store analysis and forecasts provides the insight needed to make informed decisions to grow your revenue and profitability.

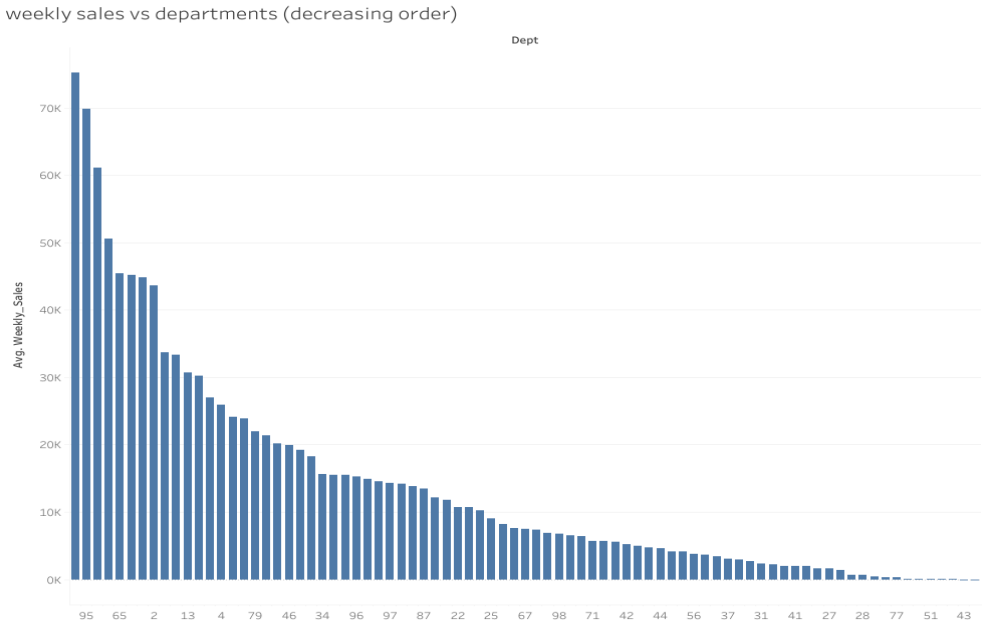
APPROACH

In this project, we have tried to forecast the sales of various stores and several departments and will try to predict which stores are performing well and which are falling behind. This will give the business owners a glimpse of what's actually happening on the ground.

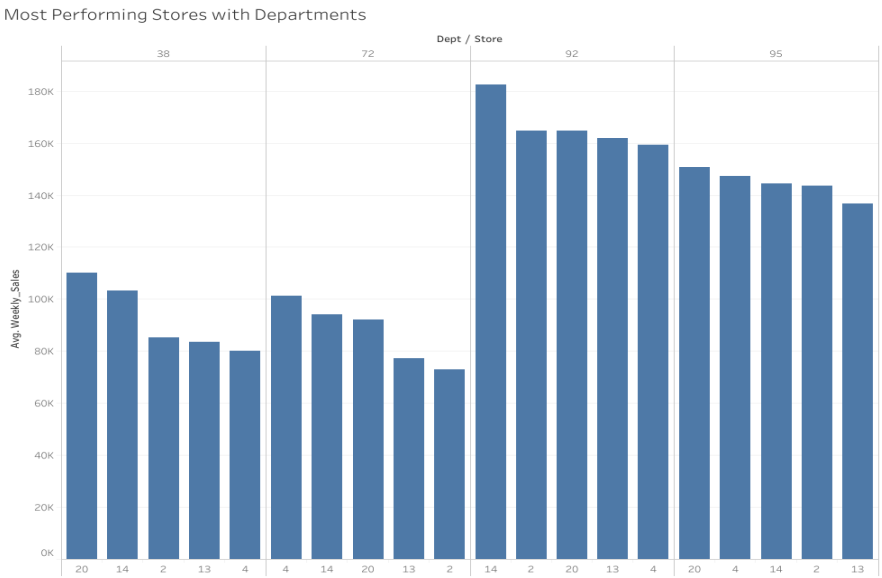
We have used Walmart Recruiting - Store Sales Forecasting dataset (https://www.kaggle.com/c/walmart-recruiting-store-sales-forecasting) for the analysis.

Preliminary Analysis

The dataset has many nan values present that must be dealt, also there are 5 irrelevant columns named ‘markdown’ that play no role in the prediction of the weekly sales. The markdown columns contain majority of the nan values so by removing them the data gets approximately 90% clean. To overcome the nan value situation, we drop all the markdown column. After this we can begin our plotting on tableau.

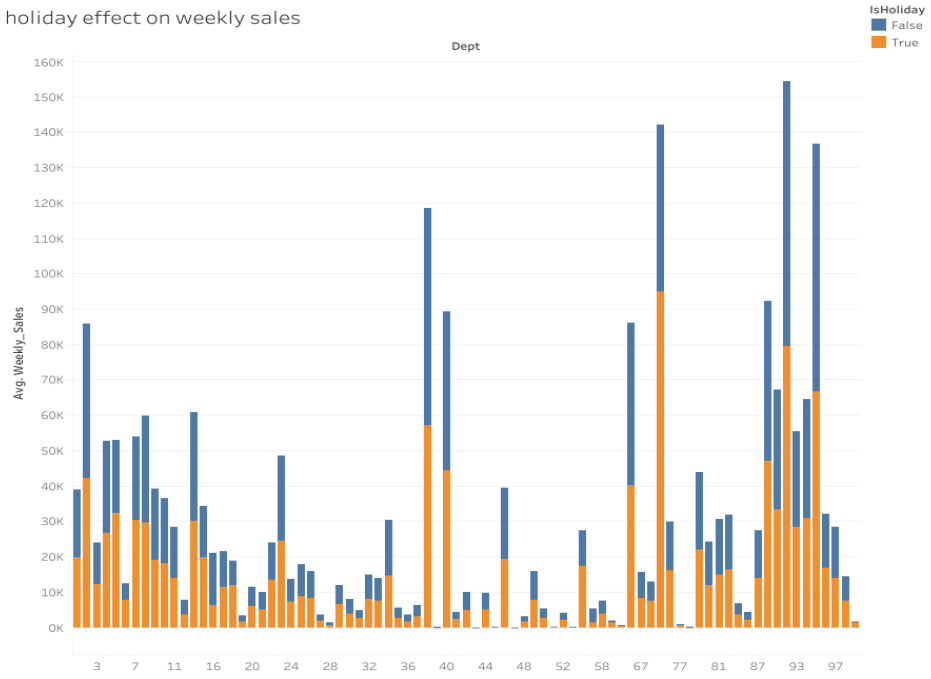


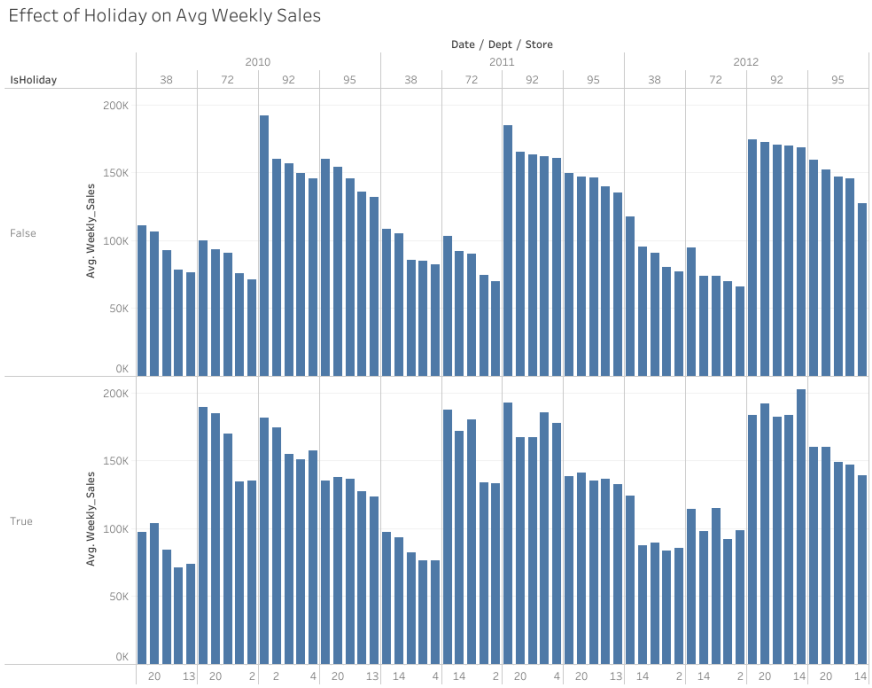
By this plot we can see that the 95th department have the highest weekly sales. The above plot is in descending order so through that we can make an insight about the average weekly sales in relation to the several departments.



Through this plot we can infer which department and store combination performs the best in weekly sales ie having the highest values of average weekly sales.

As the data is spread over a number of years we can safely assume that there must be a timestamp, when date is have a relation with the data a new effect comes into perspective. The effect of holidays can be seen on the weekly sales and this effect is different on different holidays.





As we can see different stores have different average weekly sales when on a particular day there is a holiday or not.

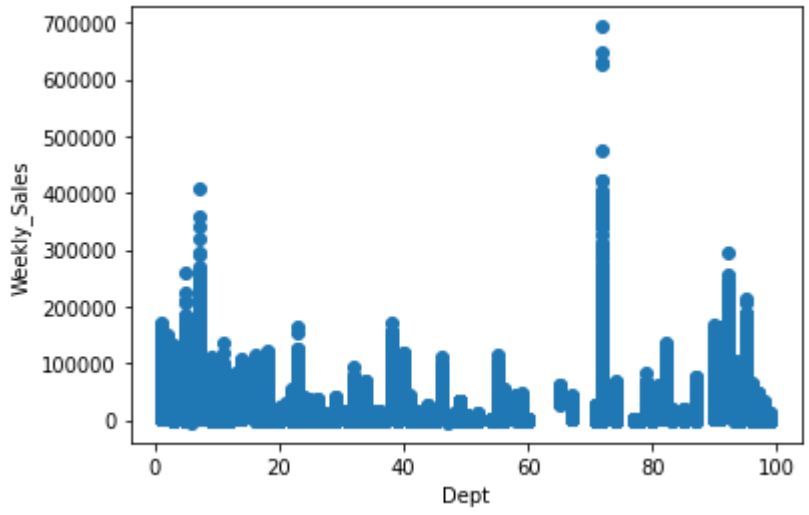
Let's take a look at what our data looks like after data pre-processing.

	Store	Dept	Date	Weekly_Sales	IsHoliday	Type	Size	Temperature	Fuel_Price	CPI	Unemployment
0	1	1	05-02-2010	24924.50	False	A	151315	42.31	2.572	211.096358	8.106
1	1	1	12-02-2010	46039.49	True	A	151315	38.51	2.548	211.242170	8.106
2	1	1	19-02-2010	41595.55	False	A	151315	39.93	2.514	211.289143	8.106
3	1	1	26-02-2010	19403.54	False	A	151315	46.63	2.561	211.319643	8.106
4	1	1	05-03-2010	21827.90	False	A	151315	46.50	2.625	211.350143	8.106
...	...	...	...	...	...	...	...	...	...	...	...
421565	45	98	28-09-2012	508.37	False	B	118221	64.88	3.997	192.013558	8.684
421566	45	98	05-10-2012	628.10	False	B	118221	64.89	3.985	192.170412	8.667
421567	45	98	12-10-2012	1061.02	False	B	118221	54.47	4.000	192.327265	8.667
421568	45	98	19-10-2012	760.01	False	B	118221	56.47	3.969	192.330854	8.667
421569	45	98	26-10-2012	1076.80	False	B	118221	58.85	3.882	192.308899	8.667

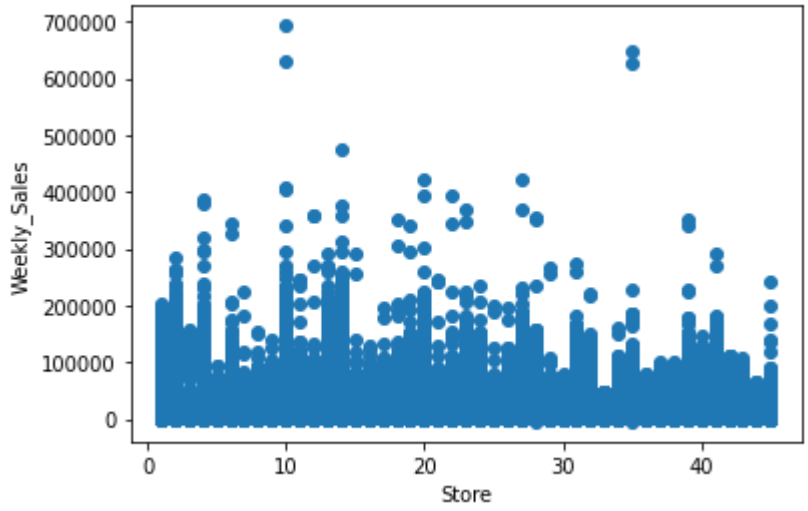
We can see that there are total 11 columns present out of which one is a dependent variable and the rest are independent variables.

For further analysis of the impact of independent variables on the dependent variables and to get a better picture of the dataset we do some basic plotting with matplotlib.

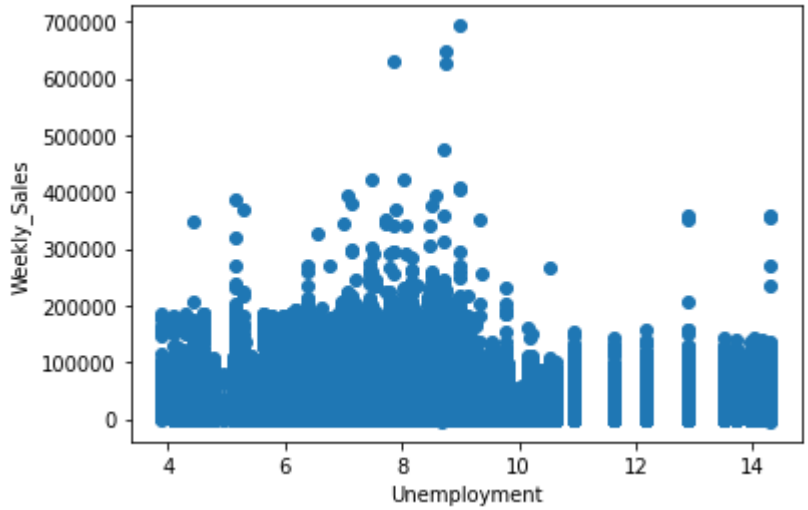
First plot: department vs weekly sales:



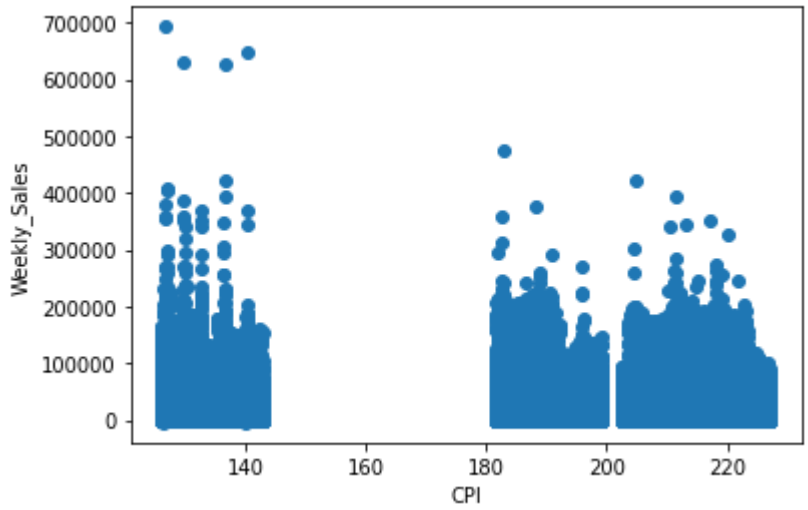
Second plot: store vs weekly sales:



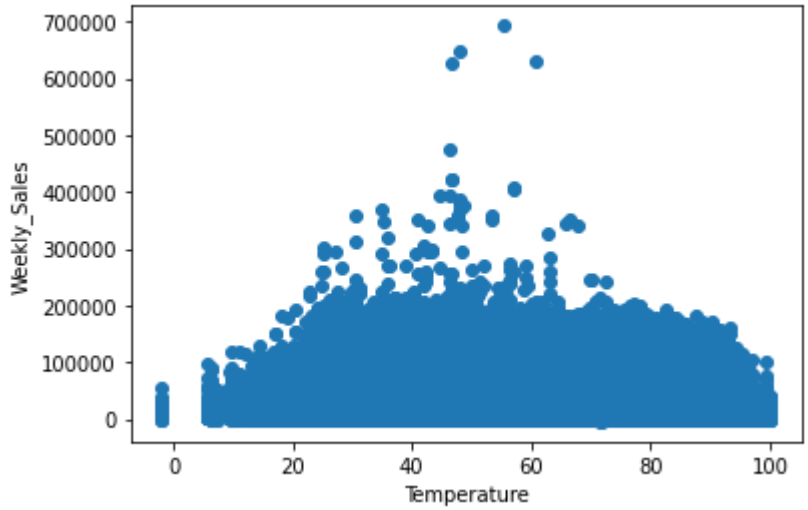
Third plot: unemployment vs weekly sales:



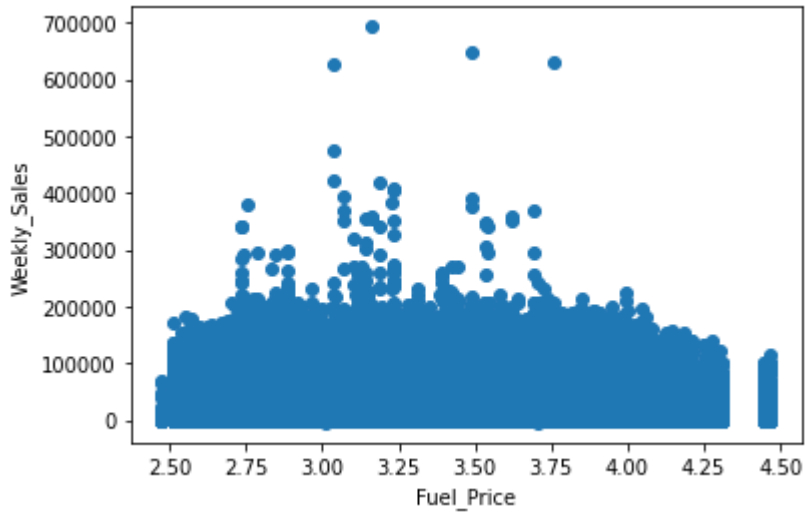
Fourth plot:: cpi vs weekly sales:



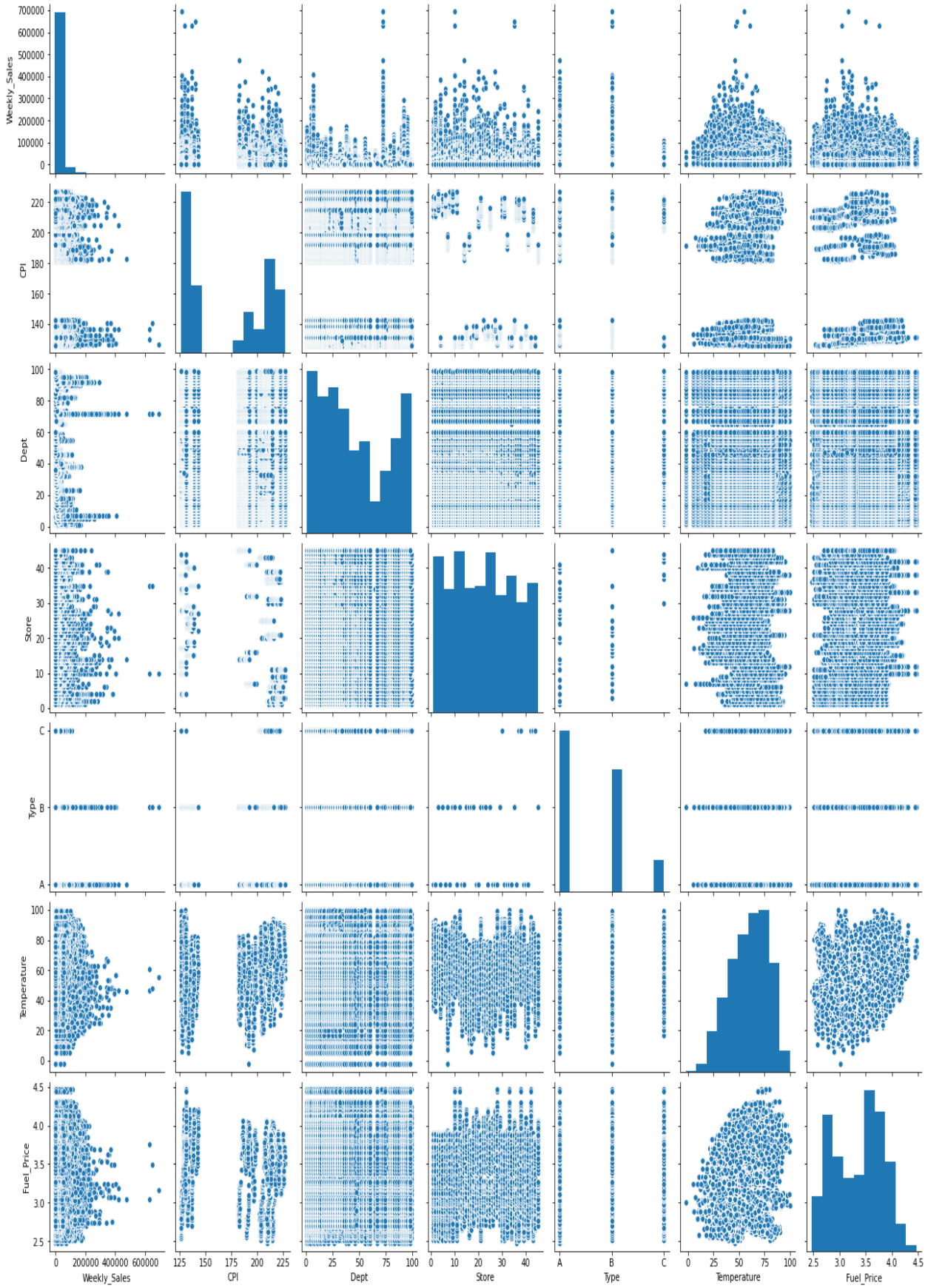
Fifth plot : temperature vs weekly sales:



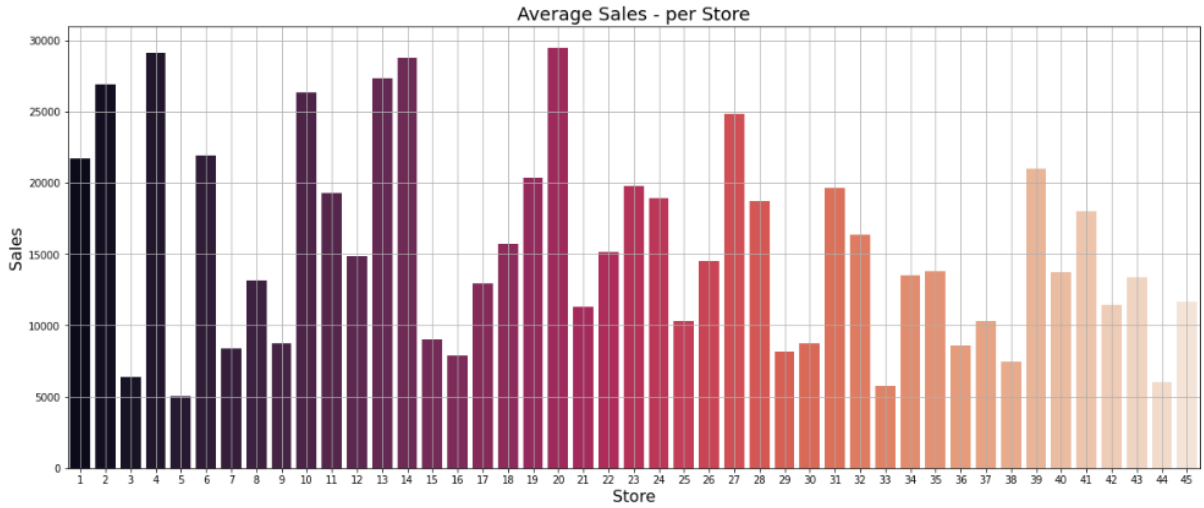
Sixth plot: fuel price vs weekly sales:



Now to better understand the inter-relation between the different variables we build a pair plot which will show us a better picture of the variables.



The average sales per store can be a step towards developing a predictive model as it will help us cross reference the results that we get.



We will now move onto the next section which is building the predictive models. These models will allow us to forecast future sales values for each store/department. We have tried various state of the art models which are optimised to get the best results. Some models are repeated based on the variation of datasets we have used.

First predictive model

The first predictive model that we are using is the KNearest Regressor in this we will be defining the model parameters with 20 neighbours and then evaluating the model's performance on the basis of various metrics.

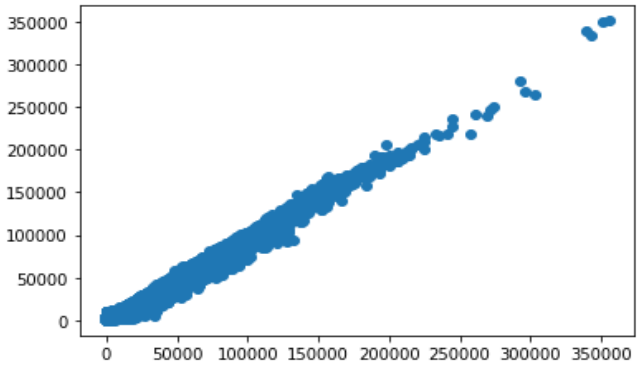
```
RMSE value for k=  1 is: 2934.539057342966
RMSE value for k=  2 is: 2595.5398992439896
RMSE value for k=  3 is: 2543.661798764371
RMSE value for k=  4 is: 2560.45121038921
RMSE value for k=  5 is: 2603.8027677184837
RMSE value for k=  6 is: 2655.2659845856665
RMSE value for k=  7 is: 2716.623656900424
RMSE value for k=  8 is: 2767.4064180999367
RMSE value for k=  9 is: 2826.877713962906
RMSE value for k=  10 is: 2879.6701656966757
RMSE value for k=  11 is: 2928.0456893914757
RMSE value for k=  12 is: 2973.45485091137
RMSE value for k=  13 is: 3018.6834589562377
RMSE value for k=  14 is: 3062.5738351188156
RMSE value for k=  15 is: 3101.8174524622495
RMSE value for k=  16 is: 3141.390636190269
RMSE value for k=  17 is: 3175.404975750531
RMSE value for k=  18 is: 3210.4266579478103
RMSE value for k=  19 is: 3247.3408095500486
RMSE value for k=  20 is: 3283.166753531629
```

Results of the model:

Mean absolute error: 2312.59

Mean squared error: 10779183.93

Root mean squared error: 3283.16

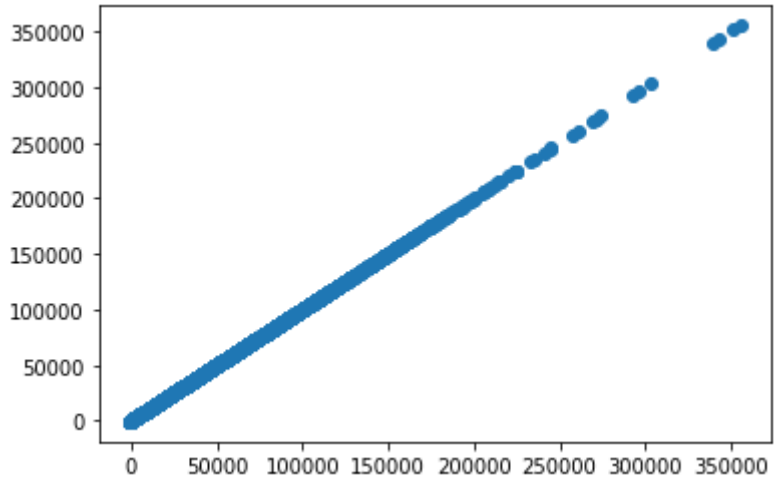


(scatter plot of y_test vs y_pred)

Second predictive model

The second predictive model is the linear regressor model. This is a very basic model and this model's prediction will be the baseline of assessment for best model amongst various models.

(scatter plot of y_test vs y_pred)



The results of the model are:

Mean absolute error: 2.75

Mean squared error: 1.22

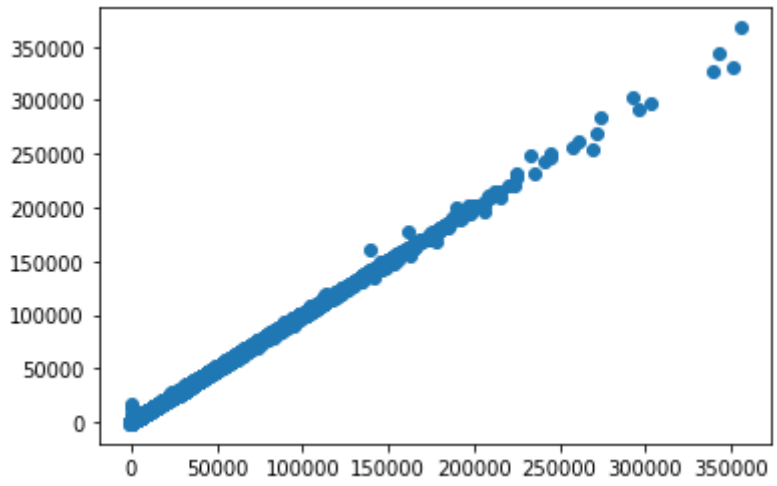
Root mean squared error: 3.49

Third predictive model

The third predictive model is the XGBoost model this model has been hyper tuned so as to produce best results as a predictive model. We will use grid search algorithm in order to find the optimal and the best parameters to use in order to get the best results.

```
Fitting 2 folds for each of 9 candidates, totalling 18 fits
[Parallel(n_jobs=5)]: Using backend LokyBackend with 5 concurrent workers.
[Parallel(n_jobs=5)]: Done  18 out of  18 | elapsed: 23.8min finished
0.9991604315249406
{'colsample_bytree': 0.7, 'learning_rate': 0.07, 'max_depth': 7, 'min_child_weight': 4, 'n_estimators': 500, 'nthread': 4,
'objective': 'reg:linear', 'silent': 1, 'subsample': 0.7}
```

(scatter plot of y_test vs y_pred)



The results of the model are:

Mean absolute error: 221.82

Mean squared error: 161993.57

Root mean squared error: 402.48

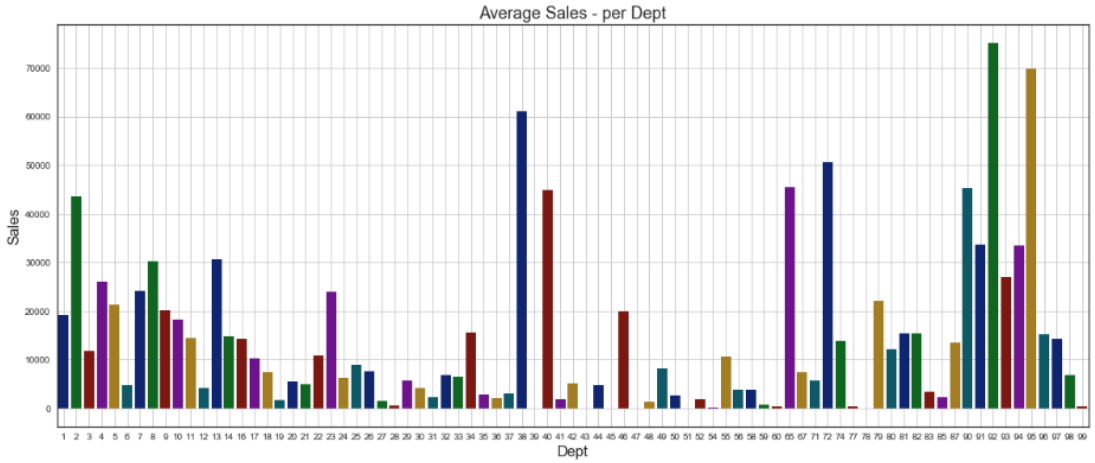
In the end we plot a table containing all the metric values of all the models with their accuracy calculated as well.

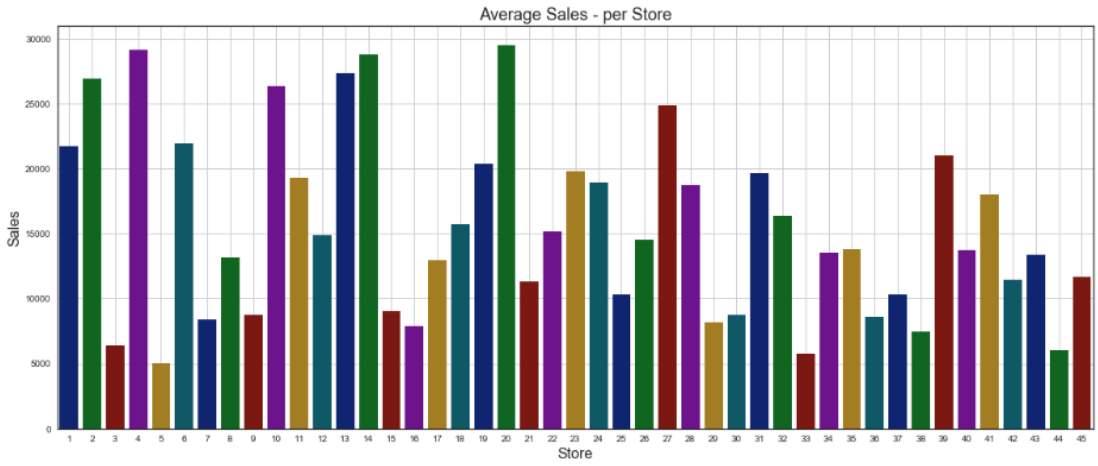
Model	MAE	MSE	RMSE	Accuracy
LinearRegressor(BASELINE)	2.754066694287157e-11	1.2241969281967417e-21	3.498852566480534e-11	1.0
KNNRegressor	2312.5928250705692	10779183.931495419	3283.166753531629	0.9790613650499077
XGBRegressor	221.82097426212192	161993.5770013043	402.4842568365927	0.9996853264222368

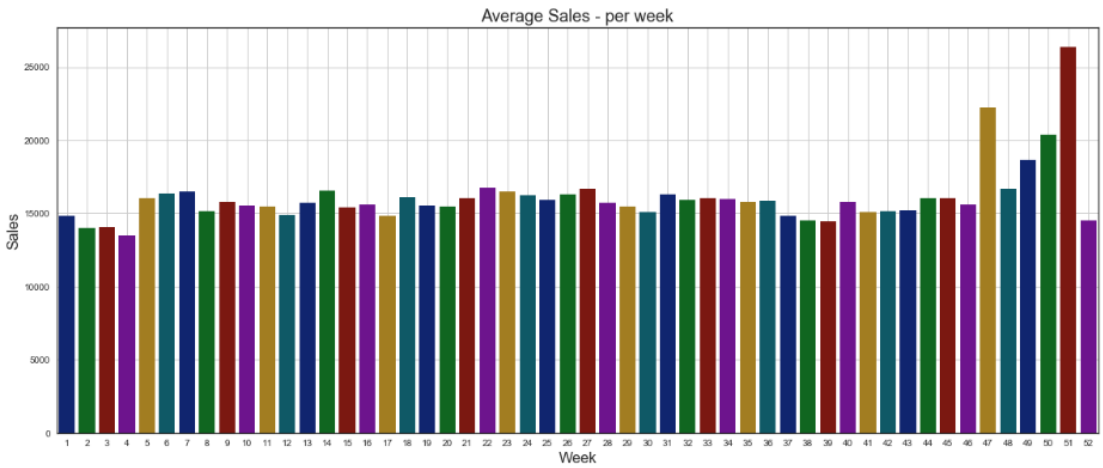
So from the above table it is conclusive that the XGBRegressor model works the best on the test dataset and produces a very high accuracy value.

To sum up we can say that for the prediction of the weekly sales data we can use the XGBRegressor for a better and more efficient results.

Here we tried out adding some features and experiment with our models. We noticed some stores, departments and weeks have relatively greater impact on the weekly sales compared to other stores, departments or weeks. We have already plotted their variations in the preliminary analysis section. We will again give you a glimpse of how its changing and how we will get some insights from it.







You can notice that some stores, departments and weeks have higher average weekly sales compared to other stores, dept and weeks.

We will add 3 additional columns -: Store_weight, Dept_weight and week_weight.

Weights will be assigned to Stores, Department and weeks based on the trends of weekly sales across all the Stores, Department and weeks respectively.

The resulting database we got here is –

Store	Dept	IsHoliday	Size	Week	Year	Store_weight	Dept_weight	Week_weight	Type_A	Type_B	Type_C
18	98	False	120653	21	2011	27	56	34	0	1	0
41	32	False	196321	29	2012	29	57	19	1	0	0
22	32	False	119557	38	2011	26	57	5	0	1	0
29	58	True	93638	36	2010	7	43	29	0	1	0
30	52	False	42988	49	2010	10	34	49	0	0	1

We will go ahead with this dataset and will try out some Machine learning models to predict the weekly sales values.

We will try out -

- KNN regression
- Linear Regression is getting us a huge WMAE value. Hence, we rejected it.
- Linear SVM and RBF SVM is running too slow. We chose to skip it.
- Random Forest Regressor
- LightGBM instead of XGBoost as LightGBM is relatively faster
- Ensemble model (Simple average of all the above models)

Fourth predictive model

We tried out KNNRegressor on the dataset. After each iteration and for various values of K, the Weighted Mean Absolute Error (WMAE) value we got were -

```
For K =  1, WMAE we got was 2279.02
For K =  5, WMAE we got was 2912.24
For K =  10, WMAE we got was 3358.03
For K =  15, WMAE we got was 3665.89
For K =  21, WMAE we got was 3950.76
For K =  30, WMAE we got was 4256.38
For K =  41, WMAE we got was 4585.3
```

The final parameter on which we trained our dataset was –

```
KNeighborsRegressor(algorithm='auto', leaf_size=30, metric='minkowski',
                    metric_params=None, n_jobs=None, n_neighbors=1, p=2,
                    weights='uniform')
```

The result of the model is:

Weighted Mean absolute error: 2279.02

Fifth predictive model

We then tried out Random Forest Regressor. An ensemble model consisting of various decision trees.

After hyper parameter tuning several parameters, the best set of parameters we got was -

```
RandomForestRegressor(bootstrap=True, criterion='mse', max_depth=None,
                      max_features='auto', max_leaf_nodes=None,
                      min_impurity_decrease=0.0, min_impurity_split=None,
                      min_samples_leaf=1, min_samples_split=2,
                      min_weight_fraction_leaf=0.0, n_estimators=150,
                      n_jobs=None, oob_score=False, random_state=None,
                      verbose=0, warm_start=False)
```

The result of the model is:

Weighted Mean absolute error: 1444.02

Sixth predictive model

For a change, we tried out LightGBM Regressor. An boosting algorithm much faster and sophisticated than XGBRegressor. After hyper parameter tuning various parameters, the best set of parameters we got was –

```
LGBM = LGBMRegressor(boosting_type='gbdt', class_weight=None, colsample_bytree=1.0,importance_type='split',
                     learning_rate=0.3, max_bin=150, max_depth=-1, min_child_samples=5, min_child_weight=0.001,
                     min_data_in_leaf=3, min_depth=2, min_split_gain=0.0,n_estimators=3000, n_jobs=-1, num_leaves=80,
                     objective='regression',random_state=42, reg_alpha=0.1, reg_lambda=2, silent=True,
                     subsample=1.0, subsample_for_bin=200000, subsample_freq=0,verbose=1)
```

The result of the model is:

Weighted Mean absolute error: 1194.88

Seventh predictive model

In the end, we merged all the models and created an ensembled simple average model. The score wasn't bad compared to the basic KNNRegressor model.

```
pred1=knn.predict(X_test)
pred2=RF.predict(X_test)
pred3=LGBM.predict(X_test)

finalpred=(pred1+pred2+pred3)/3
```

The result of the model is:

Weighted Mean absolute error: 1400.9

CONCLUSION

The final weekly sales predictions are pretty close to the actual sales values. These models can be used to forecasts future sales values to some extent. Hence, the retail store owners can focus more on the underperforming stores and make some changes to improve the sales. They can even promote the stores/departments which are getting high revenues to enhance the overall profit.