# ****PROBLEM STATEMENT:****

The main problem that I am assigned with is that I have to predict the sales given the data-set. As I can understand from the problem itself is that it is a regression problem. That we have to use regression models in-order to predict the sales from the data-set.

# ****DATA DESCRIPTION:****

The first step towards solution of any problem is to understand the data throughly like what the data is about and understand the throughly.if data is not understood then its just impossible to solve the data.

***FORMAL DESCRIPTION OF DATA:***

The data contains historical sales data for 45 Retail stores located in different regions. Each store contains a number of departments, and I am tasked with predicting the department-wide sales for each store.

In addition, Retail stores runs several promotional markdown events throughout the year. These markdowns precede prominent holidays, the four largest of which are the Super Bowl, Labor Day, Thanksgiving, and Christmas. The weeks including these holidays are weighted five times higher in the evaluation than non-holiday weeks. Part of the challenge presented by this competition is modeling the effects of markdowns on these holiday weeks in the absence of complete/ideal historical data.

**stores.csv**

This file contains anonymized information about the 45 stores, indicating the type and size of store.

**train.csv**

This is the historical training data, which covers to 2010–02–05 to 2012–11–01. Within this file you will find the following fields:

* Store — the store number
* Dept — the department number
* Date — the week
* Weekly\_Sales — sales for the given department in the given store
* IsHoliday — whether the week is a special holiday week

**test.csv**

This file is identical to train.csv, except we have withheld the weekly sales. You must predict the sales for each triplet of store, department, and date in this file.

**features.csv**

This file contains additional data related to the store, department, and regional activity for the given dates. It contains the following fields:

* Store — the store number
* Date — the week
* Temperature — average temperature in the region
* Fuel\_Price — cost of fuel in the region
* MarkDown1–5 — anonymized data related to promotional markdowns that Walmart is running. MarkDown data is only available after Nov 2011, and is not available for all stores all the time. Any missing value is marked with an NA.
* CPI — the consumer price index
* Unemployment — the unemployment rate
* IsHoliday — whether the week is a special holiday week

For convenience, the four holidays fall within the following weeks in the dataset (not all holidays are in the data):

Super Bowl: 12-Feb-10, 11-Feb-11, 10-Feb-12, 8-Feb-13  
Labor Day: 10-Sep-10, 9-Sep-11, 7-Sep-12, 6-Sep-13  
Thanksgiving: 26-Nov-10, 25-Nov-11, 23-Nov-12, 29-Nov-13  
Christmas: 31-Dec-10, 30-Dec-11, 28-Dec-12, 27-Dec-13

— — — — — — — — — — — — — — — — — — — — — — — -

## ****SOLUTION:****

My task here is to solve this assignment and i will try every luke and holes to solve the problem at the greatest extent.

Starting with the solution part What i have done at first is that i have downloaded the entire data and import the google drive

**from** google.colab **import** drive

drive**.**mount('/content/drive')

## ****Import the necessary libraries****

**Iimport** numpy **as** np

**import** pandas **as** pd

**import** scipy.stats **as** stats

**import** matplotlib.pyplot **as** plt

**import** seaborn **as** snssns**.**set\_style("whitegrid")

**from** sklearn **import** datasets, linear\_model

**import** plotly.express **as** px

**from** sklearn.preprocessing **import** StandardScaler

I have read the individual files from Retail stores, the data and i have printed the coulumns. I have done this to understand the type of data that was supplied to me.As my core problem in this solution is to design an approach where I am supposed to predict the sales of Retail stores given the above dataset. So, it is really very important to understand each feature with which i am supplied.

t**rain=pd.read\_csv("/content/drive/MyDrive/train.csv")**

**test=pd.read\_csv("/content/drive/MyDrive/test.csv")**

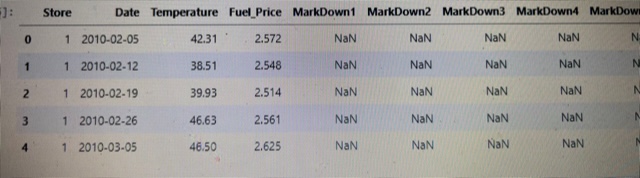
**store=pd.read\_csv("/content/drive/MyDrive/stores.csv")**

**feature=pd.read\_csv("/content/drive/MyDrive/features.csv")**

**train.head()**

****

**feature.head()**

****

**Merging the dataset**

After understanding the columns of the dataset what I did was merging the dataset as the dataset was split into many part. To get proper insight of the data I had to merge the dataset.

merge\_df**=**pd**.**merge(train,feature, on**=**['Store','Date'], how**=**'inner')

merge\_df**.**head()



merge\_df**.**describe()



**from** datetime **import** datetime **as** dt

merge\_df['DateTimeObj']**=**[dt**.**strptime(x,'%Y-%m-%d') **for** x **in** list(merge\_df['Date'])]merge\_df['DateTimeObj']**.**head()

0 2010-02-05

1 2010-02-05

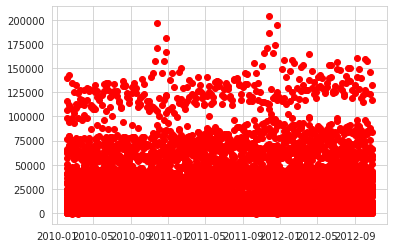
2 2010-02-05

3 2010-02-05

4 2010-02-05

Name: DateTimeObj, dtype: datetime64[ns]

**plt.plot(merge\_df[(merge\_df.Store==1)].DateTimeObj, merge\_df[(merge\_df.Store==1)].Weekly\_Sales, 'ro')plt.show()**



**weeklysales=merge\_df.groupby(['Store','Date'])['Weekly\_Sales'].apply(lambda x:np.sum(x))weeklysales[0:5]**

Store Date

1 2010-02-05 1643690.90

2010-02-12 1641957.44

2010-02-19 1611968.17

2010-02-26 1409727.59

2010-03-05 1554806.68

Name: Weekly\_Sales, dtype: float64

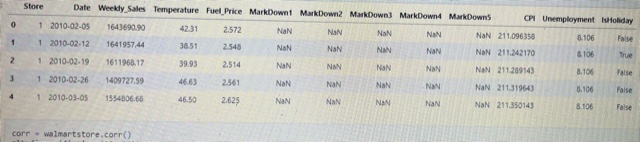
**weeklyscale=weeklysales.reset\_index()weeklyscale[0:5]**

Store Date Weekly\_Sales

0 12010-02-051643690.90112010-02-121641957.44212010-02-191611968.17312010-02-261409727.59412010-03-051554806.68

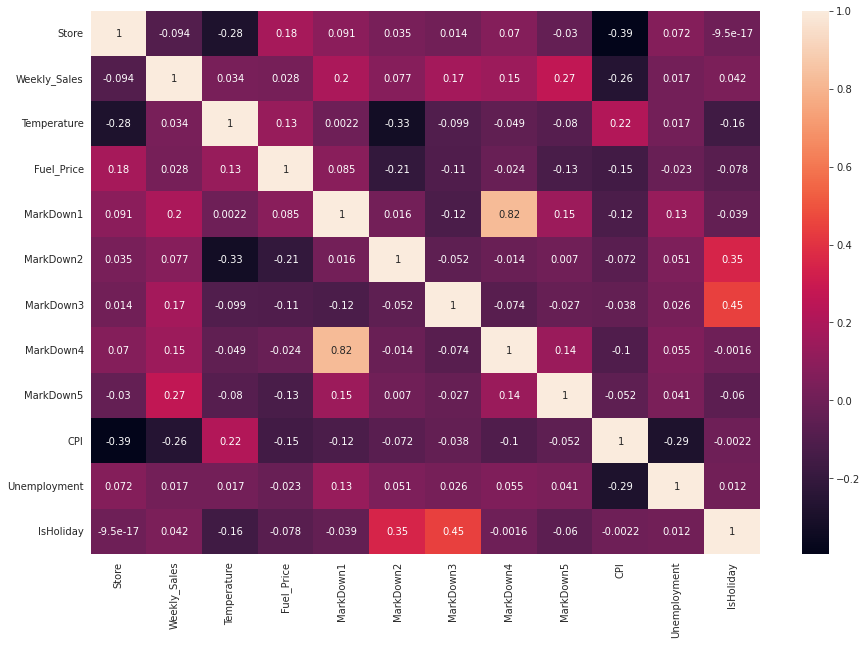
walmartstore**=**pd**.**merge(weeklyscale, feature, on**=**['Store', 'Date'], how**=**'inner')

**walmartstore.head()**



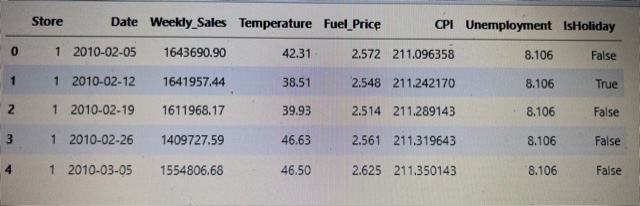
## Data Visualizations

corr **=** walmartstore**.**corr()plt**.**figure(figsize**=**(15, 10))sns**.**heatmap(corr, annot**=True**)plt**.**plot()



walmartstoredf **=** walmartstore**.**iloc[:, list(range(5)) **+** list(range(10,13))]

walmartstoredf**.**head()

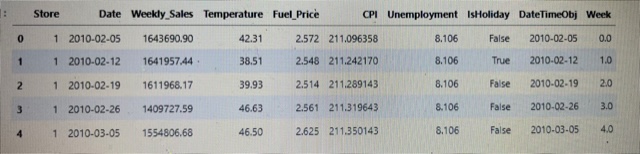


walmartstoredf['DateTimeObj'] **=** [dt**.**strptime(x, '%Y-%m-%d') **for** x **in** list(walmartstoredf['Date'])]weekNo**=**walmartstoredf**.**reset\_index()

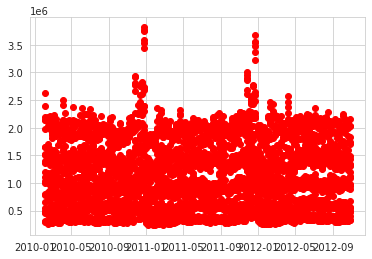
weekNo **=** [(x **-** walmartstoredf['DateTimeObj'][0]) **for** x **in** list(walmartstoredf['DateTimeObj'])]

walmartstoredf['Week'] **=** [np**.**timedelta64(x, 'D')**.**astype(int)**/**7 **for** x **in** weekNo]

walmartstoredf**.**head()

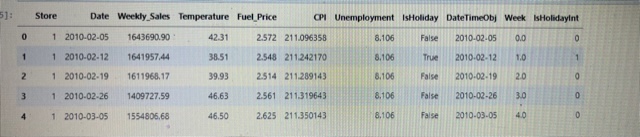


plt**.**plot(walmartstoredf**.**DateTimeObj, walmartstoredf**.**Weekly\_Sales, 'ro')plt**.**show()



walmartstoredf['IsHolidayInt'] **=** [int(x) **for** x **in** list(walmartstoredf**.**IsHoliday)]

walmartstoredf**.**head()



walmartstoredf**.**Store**.**unique()

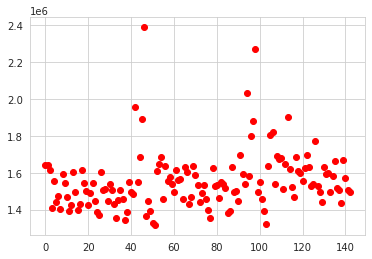
array([ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,

17,18, 19, 20, 21, 22, 23, 24, 25])

**from** sklearn.model\_selection **import** train\_test\_split

train\_WM, test\_WM **=** train\_test\_split(walmartstoredf, test\_size**=**0.3,random\_state**=**42)

plt**.**plot(walmartstoredf[(walmartstoredf**.**Store**==**1)]**.**Week, walmartstoredf[(walmartstoredf**.**Store**==**1)]**.**Weekly\_Sales, 'ro')plt**.**show()



XTrain **=** train\_WM[['Temperature', 'Fuel\_Price', 'CPI', 'Unemployment', 'Week', 'IsHolidayInt']]YTrain **=** train\_WM['Weekly\_Sales']

XTest **=** test\_WM[['Temperature', 'Fuel\_Price', 'CPI', 'Unemployment', 'Week', 'IsHolidayInt']]YTest **=** test\_WM['Weekly\_Sales']wmLinear **=** linear\_model**.**LinearRegression(normalize**=True**)wmLinear**.**fit(XTrain, YTrain)

wmLinear**.**coef\_

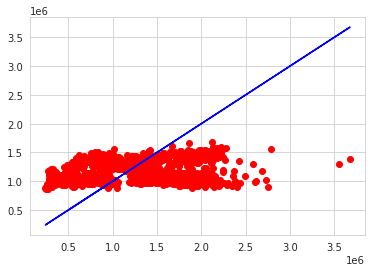
array([ 4231.2662823 , -95338.07560815, -4842.29207362, -19459.8121

3995, 486.40980057, 130397.89425663])

***Performance on the test data***

*sets*YHatTest **=** wmLinear**.**predict(XTest)

plt**.**plot(YTest, YHatTest,'ro')plt**.**plot(YTest, YTest,'b-')plt**.**show()



walmartstoredf['Store']**.**unique()

array([ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17,18, 19, 20, 21, 22, 23, 24, 25])

**Creating dummy variables for categorical data**

Store\_Dummies **=** pd**.**get\_dummies(walmartstoredf**.**Store, prefix**=**'Store')**.**iloc[:,1:]walmartstoredf **=** pd**.**concat([walmartstoredf, Store\_Dummies], axis**=**1)

walmartstoredf**.**head()



## 

## training and testing set

## train\_WM, test\_WM **=** train\_test\_split(walmartstoredf, test\_size**=**0.3,random\_state**=**42)XTrain **=** train\_WM**.**iloc[:,([3,4,5,6] **+** [9,10]) **+** list(range(11,walmartstoredf**.**shape[1]))]yTrain **=** train\_WM**.**Weekly\_Sales

XTest **=** test\_WM**.**iloc[:,([3,4,5,6] **+** [9,10]) **+** list(range(11,walmartstoredf**.**shape[1]))]yTest**=**test\_WM**.**Weekly\_Sales

XTrain**.**head()

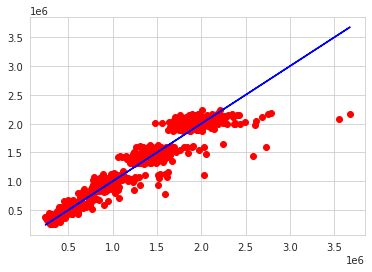


wmLinear **=** linear\_model**.**LinearRegression(normalize**=True**)

wmLinear**.**fit(XTrain, YTrain)

***Performance on the test data sets***

YHatTest **=** wmLinear**.**predict(XTest)plt**.**plot(YTest, YHatTest,'ro')plt**.**plot(YTest, YTest,'b-')plt**.**show()



# Machine Learning Models

****the accuray of the model by sum of Square and mean absolute prediction error****

# MAPE = np.mean(abs((YTest - YHatTest)/YTest))

# MSSE = np.mean(np.square(YHatTest - YTest))

print(MAPE, MSSE)

0.09021536920201166 26889099516.83717

accuracy is 90.21536920201166

## Conclusion

So our dataset was labelled and our problem statement was of prediction, hence we have used different supervised learning algorithms used for prediction.

All the algorithms used in this project are :

* Linear Regression

the best model that we obtained was Linear Regression with Accuracy is 90.21%