```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
from patsy import dmatrices
import sklearn
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
walmart data =
pd.read csv("C:/Users/Lenovo/Downloads/Compressed/Walmart Store sales.
csv")
walmart data.head()
                Date Weekly Sales Holiday Flag Temperature
   Store
Fuel Price \
       1 05-02-2010
                        1643690.90
                                               0
                                                        42.31
2.572
       1 12-02-2010
                        1641957.44
                                               1
                                                        38.51
2.548
       1 19-02-2010
                        1611968.17
                                               0
                                                        39.93
2.514
       1 26-02-2010
                        1409727.59
                                               0
                                                        46.63
2.561
                                               0
                                                        46.50
4
       1 05-03-2010
                        1554806.68
2.625
               Unemployment
          CPI
  211.096358
                      8.106
1
  211.242170
                      8.106
2 211.289143
                      8.106
  211.319643
                      8.106
  211.350143
                      8.106
walmart data groupby = walmart data.groupby('Store')
['Weekly_Sales'].sum()
print("Store Number {} has maximum Sales. Sum of Total Sales
{}".format(walmart data groupby.idxmax
(),walmart_data_groupby.max()))
Store Number 20 has maximum Sales. Sum of Total Sales 301397792.46
walmart data std =
walmart_data.groupby('Store').agg({'Weekly_Sales':'std'})
print("Store Number {} has maximum Standard Deviation. STD
{}".format(walmart_data_std['Weekly_Sales'].idxmax(),walmart_data_std[
'Weekly Sales'].max()))
```

Store Number 14 has maximum Standard Deviation, STD 317569,9494755081

```
walmart data std = walmart data.groupby('Store').agg({'Weekly Sales':
['mean','std']})
walmart_data_std.head()
       Weekly Sales
               mean
                               std
Store
       1.555264e+06
                     155980.767761
1
2
       1.925751e+06
                     237683.694682
3
       4.027044e+05
                     46319.631557
       2.094713e+06
                     266201.442297
5
       3.180118e+05
                      37737.965745
walmart data Q32012 =
walmart data[(pd.to datetime(walmart data['Date']) >=
pd.to datetime('07-01-2012')) & (pd.to datetime(walmart data['Date'])
<= pd.to datetime('09-30-2012'))]</pre>
walmart_data_growth = walmart_data_Q32012.groupby(['Store'])
['Weekly Sales'].sum()
print("Store Number {} has Good Quartely Growth in 03'2012
{}".format(walmart data growth.idxmax(),walmart data growth.max()))
Store Number 4 has Good Quartely Growth in Q3'2012 25652119.35
# Stores Holiday Sales
stores holiday sales = walmart data[walmart data['Holiday Flag'] == 1]
# Stores Weekday Sales
stores nonholiday sales = walmart data[walmart data['Holiday Flag'] ==
#Stores Sales in Super Bowl Day
#Super Bowl: 12-Feb-10, 11-Feb-11, 10-Feb-12, 8-Feb-13
stores holiday sales superBowl =
stores_holiday_sales[(pd.to_datetime(stores_holiday_sales['Date']) ==
pd.to datetime('12-02-2010'))
(pd.to_datetime(stores_holiday_sales['Date']) == pd.to_datetime('11-
02-2011'))|(pd.to datetime(stores holiday sales['Date']) ==
pd.to datetime('10-02-2012'))|
(pd.to datetime(stores holiday sales['Date']) == pd.to datetime('08-
02-2013'))]
#Stores Sales in Labour Day
#Labour Day: 10-Sep-10, 9-Sep-11, 7-Sep-12, 6-Sep-13
stores holiday sales labourDay =
stores holiday sales[(pd.to datetime(stores holiday sales['Date']) ==
pd.to datetime('10-09-2010')) |
(pd.to datetime(stores holiday sales['Date']) == pd.to datetime('09-
09-2011'))|(pd.to datetime(stores holiday sales['Date']) ==
pd.to datetime('07-09-2012'))|
(pd.to datetime(stores holiday sales['Date']) == pd.to datetime('06-
09-2013'))]
```

```
#Stores Sales in Thanks Giving
#Thanksgiving: 26-Nov-10, 25-Nov-11, 23-Nov-12, 29-Nov-13
stores_holiday_sales_thanksgiving =
stores holiday sales[(pd.to datetime(stores holiday sales['Date']) ==
pd.to datetime('26-11-2010')) |
(pd.to_datetime(stores_holiday_sales['Date']) == pd.to_datetime('25-
11-2011'))|(pd.to datetime(stores holiday sales['Date']) ==
pd.to datetime('23-11-2012'))|
(pd.to datetime(stores holiday sales['Date']) == pd.to datetime('29-
11-2013'))]
#Stores Sales in Christmas
# Christmas: 31-Dec-10, 30-Dec-11, 28-Dec-12, 27-Dec-13
stores_holiday_sales_Christmas =
stores holiday sales[(pd.to datetime(stores holiday sales['Date']) ==
pd.to datetime('31-12-2010')) |
(pd.to datetime(stores holiday sales['Date']) == pd.to datetime('30-
12-2011'))|(pd.to datetime(stores holiday sales['Date']) ==
pd.to datetime('28-12-2012'))|
(pd.to datetime(stores holiday sales['Date']) == pd.to datetime('27-
12-2013'))]
stores nonholiday sales mean =
stores nonholiday sales.groupby(['Date']).agg({'Weekly Sales':'mean'})
.reset index()
stores holiday sales sum =
stores_holiday_sales.groupby(['Date']).agg({'Weekly_Sales':'sum'}).res
et index()
for row in stores holiday sales sum.itertuples():
    for row1 in stores nonholiday sales mean.itertuples():
        if row.Weekly Sales > row1.Weekly Sales:
            print("On this Date {} Holiday Sales is greater than Non
Holiday Sales and the Sales :- {}".format(row.Date,row.Weekly Sales))
            break;
On this Date 07-09-2012 Holiday Sales is greater than Non Holiday
Sales and the Sales :- 48330059.31
On this Date 09-09-2011 Holiday Sales is greater than Non Holiday
Sales and the Sales :- 46763227.53
On this Date 10-02-2012 Holiday Sales is greater than Non Holiday
Sales and the Sales :- 50009407.92
On this Date 10-09-2010 Holiday Sales is greater than Non Holiday
Sales and the Sales :- 45634397.839999996
On this Date 11-02-2011 Holiday Sales is greater than Non Holiday
Sales and the Sales :- 47336192.79
On this Date 12-02-2010 Holiday Sales is greater than Non Holiday
Sales and the Sales :- 48336677.63
On this Date 25-11-2011 Holiday Sales is greater than Non Holiday
Sales and the Sales :- 66593605.26
On this Date 26-11-2010 Holiday Sales is greater than Non Holiday
Sales and the Sales :- 65821003.24
```

```
On this Date 30-12-2011 Holiday Sales is greater than Non Holiday
Sales and the Sales :- 46042461.04
On this Date 31-12-2010 Holiday Sales is greater than Non Holiday
Sales and the Sales :- 40432519.0
print("Super Bowl Day
Sale",stores holiday sales superBowl['Weekly Sales'].sum())
print("Labour Day
Sale",stores holiday sales labourDay['Weekly Sales'].sum())
print("Thanksgiving Day
Sale",stores_holiday_sales_thanksgiving['Weekly Sales'].sum())
print("Christmas Day
Sale",stores holiday sales Christmas['Weekly Sales'].sum())
Super Bowl Day Sale 145682278.34
Labour Day Sale 140727684.68
Thanksgiving Day Sale 132414608.5
Christmas Day Sale 86474980.03999999
x features object = walmart data[walmart data['Store'] ==1]
[['Store','Date']]
date obj = walmart data[walmart data['Store'] ==1][['Date']]
date obj.index +=1
x features object.Date = date obj.index
x features object.head()
   Store Date
0
       1
             1
             2
1
       1
             3
2
       1
3
       1
             4
4
       1
             5
y target = walmart data[walmart data['Store'] ==1]['Weekly Sales']
y target.head()
0
     1643690.90
1
     1641957.44
2
     1611968.17
3
     1409727.59
4
     1554806.68
Name: Weekly Sales, dtype: float64
from sklearn.model selection import train test split
x train,x test,y train,y test =
train test split(x features object,y target,random state=1)
from sklearn.linear model import LinearRegression
linreg = LinearRegression()
linreq.fit(x train,y train)
feature dataset = walmart data[walmart data['Store'] ==1]
```

```
[['Store','CPI','Unemployment','Fuel Price']]
feature dataset.head()
  Store
               CPI
                    Unemployment Fuel Price
      1 211.096358
0
                           8.106
                                      2.572
1
      1 211.242170
                           8.106
                                      2.548
2
      1 211.289143
                                      2.514
                           8.106
3
      1 211.319643
                           8.106
                                      2.561
      1 211.350143
                           8.106
                                      2.625
response set cpi = walmart data[walmart data['Store'] ==1]
['CPI'].astype('int64')
response set unemployment = walmart data[walmart data['Store'] ==1]
['Unemployment'].astype('int64')
from sklearn.model selection import train test split
x_train_cpi,x_test_cpi,y_train_cpi,y_test_cpi =
train test split(feature dataset, response set cpi, random state=1)
x_train_unemp, x_test_unemp, y_train_unemp, y_test_unemp =
train_test_split(feature_dataset,response_set_unemployment,random_stat
e=1)
from sklearn.linear model import LogisticRegression
logreg = LogisticRegression(max iter=10000)
logreg.fit(x_train_cpi,y_train_cpi)
y pred = logreg.predict(x test cpi)
logreg.fit(x_train_unemp,y_train_unemp)
#y pred unemp = logreg.predict(x test unemp)
LogisticRegression(max iter=10000)
y pred unemp = logreg.predict(x test unemp)
from sklearn import metrics
print(metrics.accuracy_score(y_test_cpi,y_pred))
print(metrics.accuracy score(y test unemp,y pred unemp))
0.72222222222222
0.944444444444444
print('cpi actual :', y_test_cpi.values[0:30])
print('cpi Predicted :', y pred[0:30])
print('actual Unemployment :', y_test_unemp.values[0:30])
print('Predicted Unemployment :', y_pred_unemp[0:30])
cpi actual : [215 221 211 211 221 211 210 211 215 217 221 212 216 218
211 210 211 217
 215 211 212 217 221 219 214 211 211 219 215 219]
218 211 211 211 217
 215 211 211 217 221 220 215 211 211 221 215 220]
```

walmart\_data['Day'] =
pd.to\_datetime(walmart\_data['Date']).dt.day\_name()
walmart\_data.head()

| Sto    | Store Date |            | Weekly_Sales | Holiday_Flag | Temperature |
|--------|------------|------------|--------------|--------------|-------------|
| Fuel_P | ric        | e \        |              |              | ·           |
| 0      | 1          | 05-02-2010 | 1643690.90   | 0            | 42.31       |
| 2.572  |            |            |              |              |             |
| 1      | 1          | 12-02-2010 | 1641957.44   | 1            | 38.51       |
| 2.548  | _          |            |              | _            |             |
| 2      | 1          | 19-02-2010 | 1611968.17   | 0            | 39.93       |
| 2.514  | _          |            |              | _            |             |
| 3      | 1          | 26-02-2010 | 1409727.59   | 0            | 46.63       |
| 2.561  | _          |            |              |              |             |
| 4      | 1          | 05-03-2010 | 1554806.68   | 0            | 46.50       |
| 2.625  |            |            |              |              |             |

|   | CPI        | Unemployment | Day      |
|---|------------|--------------|----------|
| 0 | 211.096358 | 8.106        | Sunday   |
| 1 | 211.242170 | 8.106        | Thursday |
| 2 | 211.289143 | 8.106        | Friday   |
| 3 | 211.319643 | 8.106        | Friday   |
| 4 | 211.350143 | 8.106        | Monday   |