GROUP: ITBDA-1901
NAME: **Dossymbekuly Iliyas**

Walmart Sales Forecasting

PYTHON ANALYSIS

Introduction

- Walmart is an American company that operates the world's largest wholesale and retail chain

- For analysis and research, I chose the Wal Mart project. In the dataset, I researched the coverage of goods for sale.

About Data Set

What you need to know:

```
Data columns (total 8 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   Store         6435 non-null   int64
 1   Date          6435 non-null   object
 2   Weekly_Sales  6435 non-null   float64
 3   Holiday_Flag  6435 non-null   int64
 4   Temperature   6435 non-null   float64
 5   Fuel_Price    6435 non-null   float64
 6   CPI           6435 non-null   float64
 7   Unemployment  6435 non-null   float64
dtypes: float64(5), int64(2), object(1)
memory usage: 402.3+ KB
```

Store

- Store - the store number

Date

- Date - the week of sales

Unemployment

- Prevailing unemployment rate

Weekly_Sales

- Sales for the given store

Holiday_Flag

- Whether the week is a special holiday week

1 – Holiday week

0 – Non-holiday week

Temperature

- Temperature on the day of sale

Fuel_Price

- Cost of fuel in the region

CPI

- Prevailing consumer price index

Methodology

The libraries I use for data analysis.

1. pandas
2. numpy
3. matplotlib
4. seaborn
5. sklearn

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import mean_squared_error, r2_score
```

Data output

What our data looks like?

```
dataSet = pd.read_csv('Walmart.csv')
```

```
dataSet.dtypes
```

```
Store                    int64
Date            datetime64[ns]
Weekly_Sales           float64
Holiday_Flag             int64
Temperature            float64
Fuel_Price             float64
CPI                    float64
Unemployment           float64
dtype: object
```

```
dataSet.nunique()
```

```
Store            45
Date            143
Weekly_Sales   6435
Holiday_Flag      2
Temperature    3528
Fuel_Price      892
CPI            2145
Unemployment    349
dtype: int64
```

```
dataSet.head(5)
```

	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

```
dataSet.shape
```

```
(6435, 8)
```

Data Type and visualization

Changing "object" to "dateTime" for further research

```
dataSet['Date'] = pd.to_datetime(dataSet['Date'])
```

```
dataSet.dtypes
```

```
Store                    int64
Date            datetime64[ns]
Weekly_Sales           float64
Holiday_Flag             int64
Temperature            float64
Fuel_Price             float64
CPI                    float64
Unemployment           float64
dtype: object
```

Visualization

```
sales = dataSet.groupby('Date')['Weekly_Sales'].agg(sales=sum).reset_index()
```

```
plt.figure(figsize=(18,7))
plt.plot(sales['Date'], sales['sales'])
```

```
[<matplotlib.lines.Line2D at 0x132c938f370>]
```


```
print(sales);
```

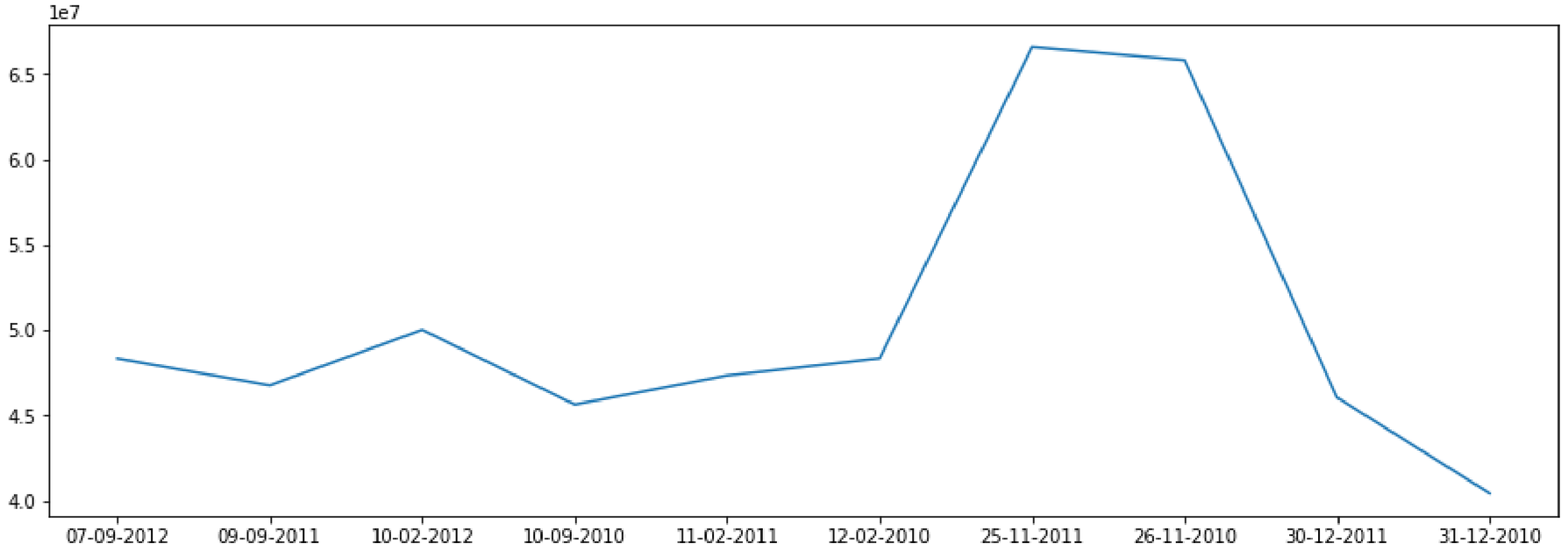
```
           Date        sales
0    2010-01-10  42239875.87
1    2010-02-04  50423831.26
2    2010-02-07  48917484.50
3    2010-02-19  48276993.78
4    2010-02-26  43968571.13
..          ...          ...
138  2012-10-08  47403451.04
```

Visualization

```
holiday_week = dataSet[dataSet['Holiday_Flag'] == 1]
holiday_week = holiday_week.groupby('Date')['Weekly_Sales'].agg(sales=sum).reset_index()
```

```
plt.figure(figsize=(15,5))
plt.plot(holiday_week['Date'], holiday_week['sales'])
```

```
[<matplotlib.lines.Line2D at 0x185431d0700>]
```



Holiday weeks

holiday_week

	Date	sales
0	2010-10-09	45634397.84
1	2010-11-26	65821003.24
2	2010-12-02	48336677.63
3	2010-12-31	40432519.00
4	2011-09-09	46763227.53
5	2011-11-02	47336192.79
6	2011-11-25	66593605.26
7	2011-12-30	46042461.04
8	2012-07-09	48330059.31
9	2012-10-02	50009407.92

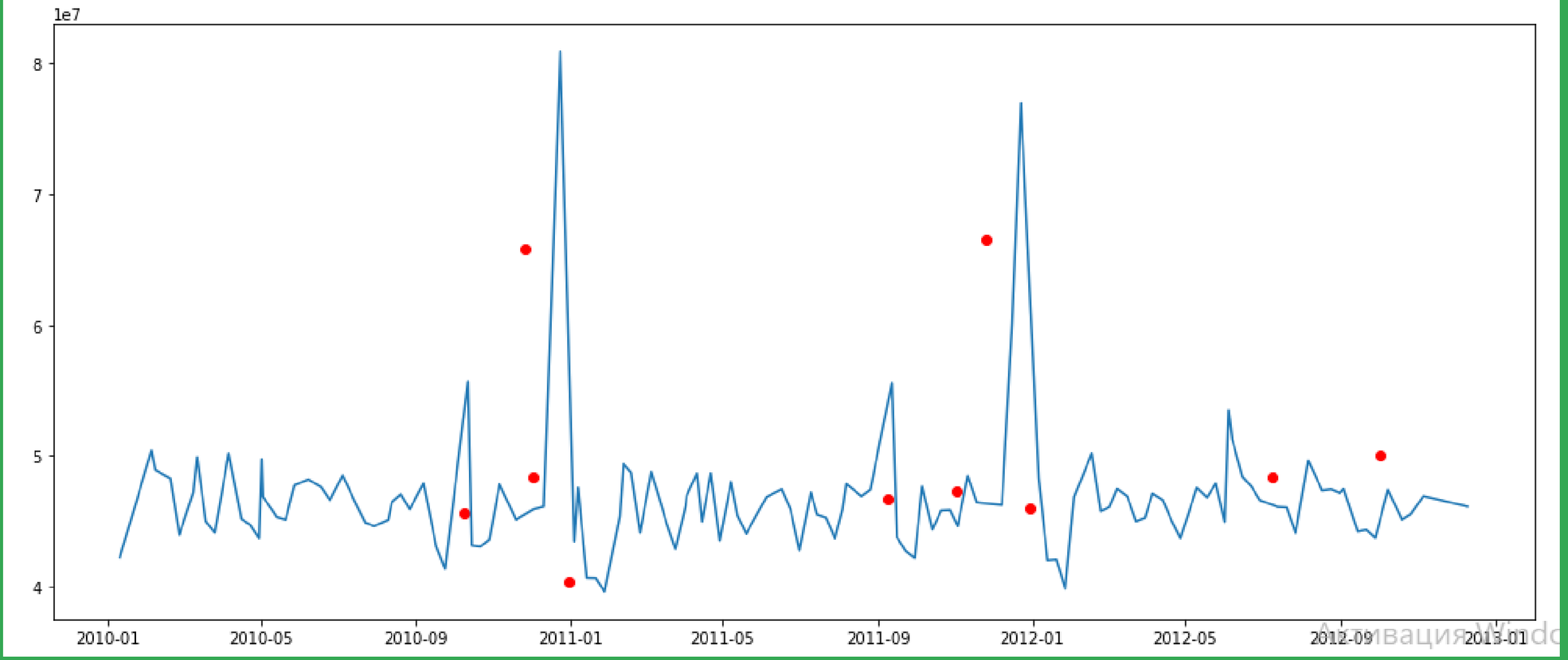
```
no_holiday_week = dataSet[dataSet['Holiday_Flag'] == 0]
no_holiday_week = no_holiday_week.groupby('Date')['Weekly_Sales'].agg(sales=sum).reset_index()
no_holiday_week
```

	Date	sales
0	2010-01-10	42239875.87
1	2010-02-04	50423831.26
2	2010-02-07	48917484.50
3	2010-02-19	48276993.78
4	2010-02-26	43968571.13
...	...	...
128	2012-10-08	47403451.04
129	2012-10-19	45122410.57
130	2012-10-26	45544116.29
131	2012-11-05	46925878.99
132	2012-12-10	46128514.25

133 rows × 2 columns

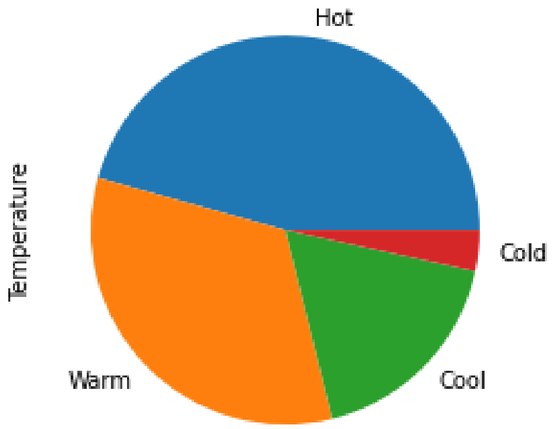
SCATTER PLOT

```
plt.figure(figsize=(17,7))
plt.scatter(holiday_week['Date'],holiday_week['sales'],c='red');
plt.plot(no_holiday_week['Date'],no_holiday_week['sales']);
```



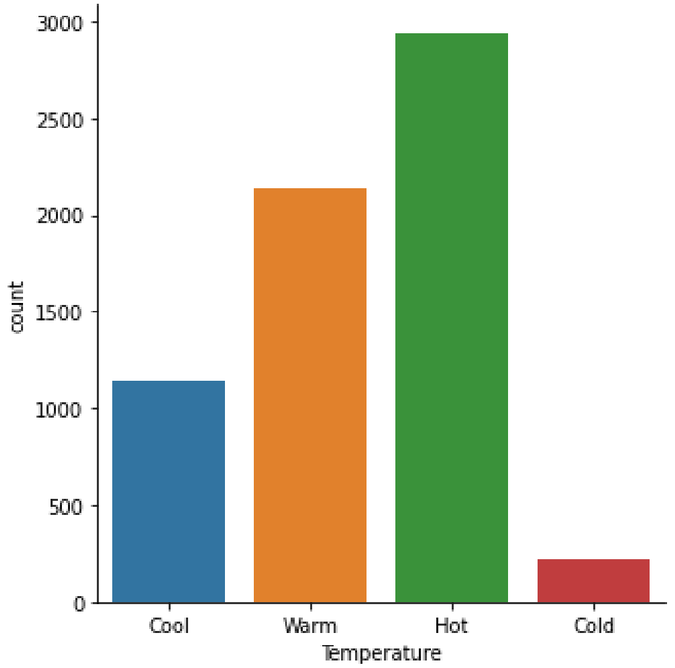
Processes with data:

```
dataSet['Temperature'].value_counts().plot(kind='pie')
alpha_color=1
```




```
sns.catplot(x ="Temperature",
kind ="count", data = dataSet)
```

```
<seaborn.axisgrid.FacetGrid at 0x1f5de782a40>
```




```
print(dataSet.Temperature.value_counts
```

```
Hot     2939
Warm    2137
Cool    1141
Cold     218
Name: Temperature, dtype: int64
```

```
dataSet['Temperature']= np.where((dataSet['Temperature'] <= 65) & (dataSet['Temperature'] > 45) ,'Warm',
                         np.where((dataSet['Temperature'] <= 45) & (dataSet['Temperature'] > 25) ,'Cool'
                         ,
                         np.where((dataSet['Temperature'] <= 25) & (dataSet['Temperature'] >= 10) ,'Cold'
                         ,
                         'Hot')))
```

Processes with data:

```
dataSet['day'] = dataSet['Date'].dt.weekday
dataSet['month'] = dataSet['Date'].dt.month
dataSet['year']  = dataSet['Date'].dt.year
```

```
# The Top 10 stores in total sales over all years
dataSet.groupby(['Store'])['Weekly_Sales'].sum().to_frame()
```

	Store	Weekly_Sales
0	20	3.013978e+08
1	4	2.995440e+08
2	14	2.889999e+08
3	13	2.865177e+08
4	2	2.753824e+08
5	10	2.716177e+08
6	27	2.538559e+08
7	6	2.237561e+08
8	1	2.224028e+08
9	39	2.074455e+08

```
# Total yearly sales
dataSet.groupby('year')['Weekly_Sales'].sum().sort_values(ascending=False).to_frame()
```

	Weekly_Sales
year	
2011	2.448200e+09
2010	2.288886e+09
2012	2.000133e+09

Thank you for
your attention!!!