Report On Exploratory Analysis Using Weather Dataset

Student Details:

- Kuldip Rameshbhai Savaliya - 1001832000
- Shivani Manojkumar Panchiwala - 1001982478
- Meghaben Ghanshyambhai Patel - 1002006777

Introduction:

In order to perform this assignment, I used the free, open-source, interactive web tool known as a Jupyter notebook, which allows researchers to combine software code, computational output, explanatory text, and multimedia resources in a single document, to work on the "weather_dataset.csv" file. Using python libraries that deal with data preprocessing, data visualization, and data exploratory analysis, I have extrapolated several insights from the provided data. Following are specifics of the exploratory analysis done on the provided dataset:

1. Analyzed Dataset: “weather_dataset.csv” - Each record in the dataset has information on weather given as below:
   Date, Time, Temp Humidity Index, Outside Temperature, WindChill, Hi Temperature, Low Temperature, Outside Humidity, Dew Point, Windspeed, Hi, Wind Direction, Rain, Barometer, Inside Temperature, Inside Humidity, ArchivePeriod.
   From this information, one may determine different weather-related indices.
2. Tools used: Jupyter Notebook in Anaconda for Python
3. Aim: To analyze the data using python libraries and visualize them

For clarification and understanding panda’s user guide and Stackoverflow were referred.

Retrieving the Data:

The ‘read_csv’ function offered by the panda’s library is used to obtain the data in comma-separated value format into DataFrame.

```
[11] #read the csv file into a Pandas data frame
     df_data = pd.read_csv('weather_dataset.csv', encoding='latin1')

     #return the first 5 rows of the dataset
     df_data.head(5)
```

Time	Temp Humidity Index	Outside Temperature	WindChill	Hi Temperature	Low Temperature	Outside Humidity	DewPoint	WindSpeed	Hi	Wind Direction
09:00	9.3	9.3	9.3	9.7	9.1	55	0.8	1	7	NNW
09:10	10.1	10.1	10.1	10.4	9.7	53	0.9	2	5	NE
09:20	10.7	10.7	10.7	11.0	10.4	52	1.3	2	5	NE
09:30	11.2	11.2	11.2	11.3	10.9	52	1.7	1	3	NNW
09:40	11.4	11.4	11.4	11.6	11.3	51	1.7	2	6	E

Task 1: Statistical Exploratory Data Analysis

```
#Task 1-a: Print the details of the df_data data frame (information such as number of rows,columns, name of columns, etc)
print ("Task 1-a: Details of df_data data frame are: \n", df_data.info())
```

Output:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4463 entries, 0 to 4462
Data columns (total 17 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   Date                    4463 non-null   datetime64[ns]
 1   Time                    4463 non-null   object
 2   Temp Humidity Index     4463 non-null   float64
 3   Outside Temperature     4463 non-null   float64
 4   WindChill               4463 non-null   float64
 5   Hi Temperature          4463 non-null   float64
 6   Low Temperature         4463 non-null   float64
 7   Outside Humidity        4463 non-null   int64
 8   DewPoint                4463 non-null   float64
 9   WindSpeed               4463 non-null   int64
 10  Hi                      4463 non-null   int64
 11  Wind Direction          4463 non-null   object
 12  Rain                    4463 non-null   float64
 13  Barometer               4463 non-null   float64
 14  Inside  Temperature     4463 non-null   float64
 15  Inside  Humidity        4463 non-null   int64
 16  ArchivePeriod           4463 non-null   int64
dtypes: datetime64[ns](1), float64(9), int64(5), object(2)
memory usage: 592.9+ KB
```

```
#Task 1-b: Find the number of rows and columns in the df_data data frame.
a = (len(df_data))
b = (len(df_data.columns))
print ("Task 1-b: Number of rows: %s and number of columns: %s" %(a,b))
```

Output:

```
Task 1-b: Number of rows: 4463 and number of columns: 17
```

```
#Task 1-c: Print the descriptive detail (count, unique, top, freq etc) for 'Wind Direction' column of the df_data
print ("Task 1-c: Descriptive details of 'Wind Direction' column are\n", df_data['Wind Direction'].describe())
```

Output:

```
Task 1-c: Descriptive details of 'Wind Direction' column are
 count     4463
unique      17
top        SSW
freq      1276
Name: Wind Direction, dtype: object
```

```
#Task 1-d: Print the average Outside Temperature for each day
print("Task 1-d: The average temp for each day:", df_data[["Date","Outside Temperature"]].groupby(['Date']).mean())
```

Output:

```
Task 1-d: The average temp for each day:             Outside Temperature
Date
2006-01-06                          12.520139
2006-01-07                          11.605556
2006-02-06                          12.960417
2006-03-06                          12.840278
2006-04-06                          12.013889
2006-05-06                          12.186806
2006-05-31                          11.950000
2006-06-06                          16.304167
2006-06-13                          13.838889
2006-06-14                          12.810417
2006-06-15                          15.224306
2006-06-16                          15.126389
2006-06-17                          15.547222
2006-06-18                          14.267361
2006-06-19                          13.454167
2006-06-20                          11.600000
2006-06-21                          10.246528
2006-06-22                          12.647917
2006-06-23                          12.529167
2006-06-24                          12.902797
2006-06-25                          11.951389
2006-06-26                          11.690278
2006-06-27                          13.687500
2006-06-28                          14.739583
2006-06-29                          15.716667
2006-06-30                         14.709722
2006-07-06                         14.219444
2006-08-06                         15.315278
2006-09-06                         12.748611
2006-10-06                         13.595139
2006-11-06                         16.884722
2006-12-06                         15.029861
```

```python
#Task 1-e: Print the average Outside Temperature for each week
print("Task 1-e: The average temp for each week:")
df_data['Date'] = pd.to_datetime(df_data['Date'])
print(df_data.resample('W', on='Date')['Outside Temperature'].mean())
```

Output:

```
Task 1-e: The average temp for each week:
Date
2006-01-08    12.270707
2006-01-15          NaN
2006-01-22          NaN
2006-01-29          NaN
2006-02-05          NaN
2006-02-12    12.960417
2006-02-19          NaN
2006-02-26          NaN
2006-03-05          NaN
2006-03-12    12.840278
2006-03-19          NaN
2006-03-26          NaN
2006-04-02          NaN
2006-04-09    12.013889
2006-04-16          NaN
2006-04-23          NaN
2006-04-30          NaN
2006-05-07    12.186806
2006-05-14          NaN
2006-05-21          NaN
2006-05-28          NaN
2006-06-04    11.950000
2006-06-11    16.304167
2006-06-18    14.469097
2006-06-25    12.189573
2006-07-02    14.108750
2006-07-09    14.219444
2006-07-16          NaN
2006-07-23          NaN
2006-07-30          NaN
2006-08-06    15.315278
2006-08-13          NaN
2006-08-20          NaN
2006-08-27          NaN
2006-09-03          NaN
2006-09-10    12.748611
2006-09-17          NaN
2006-09-24          NaN
2006-10-01          NaN
2006-10-08    13.595139
2006-10-15          NaN
2006-10-22          NaN
2006-10-29          NaN
2006-11-05          NaN
2006-11-12    16.884722
2006-11-19          NaN
2006-11-26          NaN
2006-12-03          NaN
2006-12-10    15.029861
Freq: W-SUN, Name: Outside Temperature, dtype: float64
```

```
#Task 1-f: What is the maximum Outside Temperature difference each day for all the days of the months?
print("Task 1-f: Maximum temperature difference each day for all the days of the months:", df_data.loc[df_data["Outside Temperature"] ==
                                                                                                df_data["Outside Temperature"].max()])
df1 = (df_data.set_index('Date').resample('D')['Outside Temperature'].max() - df_data.set_index('Date').resample('D')['Outside Temperature'].min())
print(df1)
```

Output:

```
Task 1-f: Maximum temperature difference each day for all the days of the months:          Date   Time  Temp Humidity Index     Outside Temperature
896 2006-06-06  14:20                   23.3                23.2       23.2

     Hi Temperature  Low Temperature  Outside Humidity  DewPoint  WindSpeed  \
896            23.2             23.1                55      13.6          3

     Hi Wind Direction  Rain  Barometer  Inside  Temperature  \
896   8             SW   0.0     1015.5                24.2

     Inside  Humidity  ArchivePeriod
896               45             10
Date
2006-01-06    8.6
2006-01-07    6.4
2006-01-08    NaN
2006-01-09    NaN
2006-01-10    NaN
              ...
2006-12-02    NaN
2006-12-03    NaN
2006-12-04    NaN
2006-12-05    NaN
2006-12-06    7.6
Freq: D, Name: Outside Temperature, Length: 335, dtype: float64
```

```
#Task 1-h: What is the minium Outside Temperature difference each day for all the days of the months?
print("Task 1-h: Minium temperature difference each day for all the days of the months:", df_data.loc[df_data["Outside Temperature"] ==
                                                                                                df_data["Outside Temperature"].min()])
```

Output:

```
Task 1-h: Minium temperature difference each day for all the days of the months:          Date   Time  Temp Humidity Index     Outside Temperature
4005 2006-06-28  04:40                   6.4                 6.4

      WindChill  Hi Temperature  Low Temperature  Outside Humidity  DewPoint  \
4005        6.4             6.5              6.3                86       4.3

      WindSpeed  Hi Wind Direction  Rain  Barometer  Inside  Temperature  \
4005          0   0             ---   0.0     1008.3                20.4

      Inside  Humidity  ArchivePeriod
4005                48             10
```

```
#Task 1-i: Display all the unique values for each column.
#for col in df_data:
    #print(df_data[col].unique())
print("Task 1-i: The unique values for each column are:", df_data.nunique())
```

Output:

```
  Task 1-i: The unique values for each column are: Date                          32
  Time                        144
  Temp Humidity Index         151
  Outside Temperature         166
  WindChill                   176
  Hi Temperature              166
  Low Temperature             166
  Outside Humidity             56
  DewPoint                    144
  WindSpeed                    13
  Hi                           30
  Wind Direction               17
  Rain                          7
  Barometer                   369
  Inside  Temperature          75
  Inside  Humidity             27
  ArchivePeriod                 1
  dtype: int64
```

Task 2: Aggregation & Filtering & Rank

```
# Task 2-A: Generate a Outside_Temperature.txt file containing the answers to the following questions:
# Using the "Outside Temperature" values:
# a. What is the average time of hottest daily temperature (over month);

import pandas as pd
df_data = pd.read_csv('weather_dataset.csv', encoding='latin1')

print("TASK 2-A.a")
d=df_data[['Date','Outside Temperature', 'Time']].groupby(['Date']).max()
print(d["Time"])
```

Output:

```
TASK 2-A.a
Date
01/06/2006    23:50
01/07/2006    08:50
02/06/2006    23:50
03/06/2006    23:50
04/06/2006    23:50
05/06/2006    23:50
06/06/2006    23:50
07/06/2006    23:50
08/06/2006    23:50
09/06/2006    23:50
10/06/2006    23:50
11/06/2006    23:50
12/06/2006    23:50
13/06/2006    23:50
14/06/2006    23:50
15/06/2006    23:50
16/06/2006    23:50
17/06/2006    23:50
18/06/2006    23:50
19/06/2006    23:50
20/06/2006    23:50
21/06/2006    23:50
22/06/2006    23:50
23/06/2006    23:50
24/06/2006    23:50
25/06/2006    23:50
26/06/2006    23:50
27/06/2006    23:50
28/06/2006    23:50
29/06/2006    23:50
30/06/2006    23:50
31/05/2006    23:50
Name: Time, dtype: object
```

```
# b. What time of the day is the most commonly occurring hottest time;
print("TASK 2-A.b")
d_t =df_data[["Date","Time","Outside Temperature"]].groupby(['Date']).max()
print("The most commonly occurring hottest time is " + d_t["Time"].mode())
```

Output:

```
TASK 2-A.b
0    The most commonly occurring hottest time is 23:50
dtype: object
```

```
# # c. Which are the Top Ten hottest times on distinct days, preferably sorted by date order.
print("TASK 2-A.c")
print("Top Ten hottest times on distinct days where Date is in descending order are as below : ")
print(d_t.sort_values(by = ["Date"] , ascending = False)['Time'].head(10))
```

Output:

```
TASK 2-A.c
Top Ten hottest times on distinct days where Date is in descending order are as below :
Date
31/05/2006    23:50
30/06/2006    23:50
29/06/2006    23:50
28/06/2006    23:50
27/06/2006    23:50
26/06/2006    23:50
25/06/2006    23:50
24/06/2006    23:50
23/06/2006    23:50
22/06/2006    23:50
Name: Time, dtype: object
```

```
#Task 2-B: Using the 'Hi Temperature' values produce a "Hi_Temperature.txt" file containing all of the Dates and Times
# where the "Hi Temperature" was within +/- 1 degree of 22.3 or
# the "Low Temperature" was within +/- 0.2 degree higher or lower of 10.3 over the first 9 days of June
print("TASK 2-B")
high =df_data[df_data["Hi Temperature"].between(21.3, 23.3) | df_data["Low Temperature"].between(10.1, 10.5)]
low =high[["Date", "Time"]]
temp = low[low["Date"].between('2006-06-01', '2006-06-09')]
if(temp.empty):
    print("No data found between '2006-06-01' and '2006-06-09' where Hi Temp is +/- 1 22.3 or Low temp is +/- 0.2 10.3")
else:
    print(low[low["Date"].between('2006-06-01', '2006-06-09')])
```

Output:

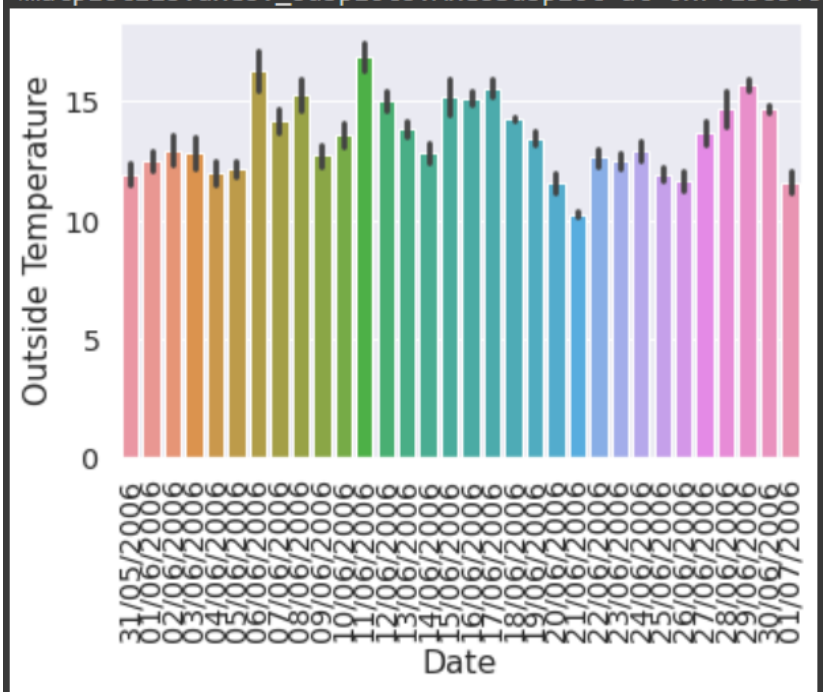
```
TASK 2-B
No data found between '2006-06-01' and '2006-06-09' where Hi Temp is +/- 1 22.3 or Low temp is +/- 0.2 10.3
```

Task 3: Visualization

```
# Task 3-a: Visualize the temperature for each month
# Think of a way to nicely visualize all the temperature and provide detailed explaination
print("Task 3-a")
import seaborn as sns
import matplotlib.pyplot as plt
sns.set_style('whitegrid')
sns.set(font_scale = 1.3)
plt.xticks(rotation=90)
sns.barplot(x=df_data['Date'], y=df_data['Outside Temperature'])
```

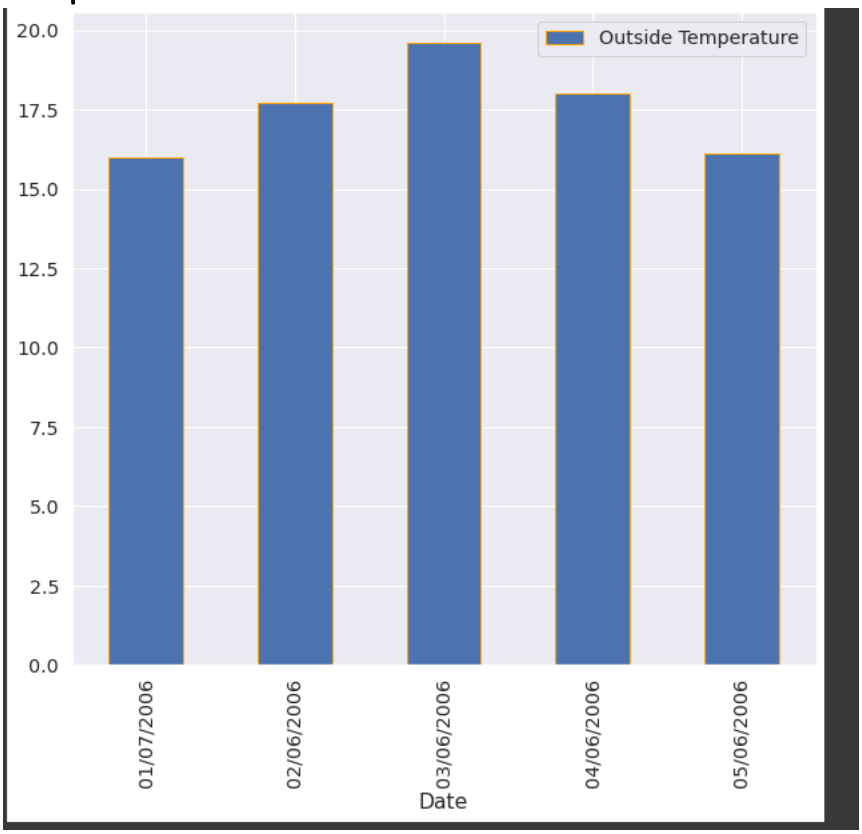
Output:

```
Task 3-b
<matplotlib.axes._subplots.AxesSubplot at 0x7f1583f36bd0>
```



```
# Task 3-b: Display the time period on a bar plot which has highest temperature for the first 5 days of every month
# provide detailed explaination
#########################begin code for Task 3-b
print("Task 3-b")
#For May and July months, data are not available
import seaborn as sns
import matplotlib.pyplot as plt
maxOutsideTemp = df_data.groupby('Date').max()
maxOutsideTemp = maxOutsideTemp.filter(["Date", "Outside Temperature"])
maxOutsideTemp = maxOutsideTemp.reset_index()
maxOutsideTemp = maxOutsideTemp[1:6]
maxOutsideTemp.plot(x="Date", y="Outside Temperature", kind="bar",figsize=(10,9), edgecolor='orange')
```

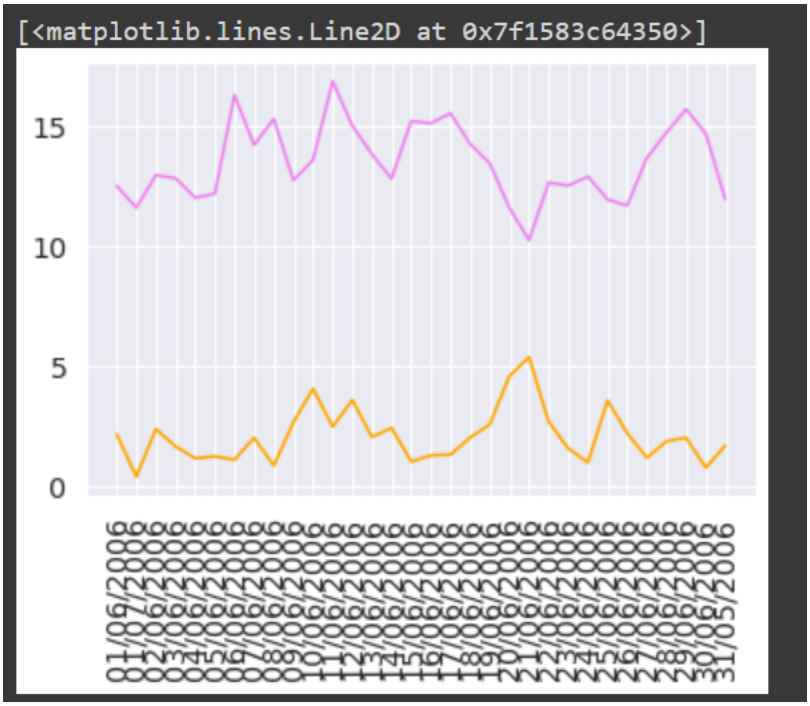
Output:



Task 4:

```
meanWindSpeed=df_data[["Date","WindSpeed"]].groupby(['Date']).mean()
meanOutsideTmperature=df_data[["Date","Outside Temperature"]].groupby(['Date']).mean()
plt.xticks(rotation=90)
plt.plot(meanWindSpeed["WindSpeed"],  color="orange")
plt.plot(meanOutsideTmperature["Outside Temperature"],  color="violet")
```

Output:



Explanation of above visualization:

In the weather dataset.csv line graph above, wind speed and outside temperature are displayed. The information above demonstrates how the wind speed affects the outdoor temperature. It is clear from the graph that as the wind speed, shown in orange, increases, the outdoor temperature decreases. For the 2006 months of May, June, and July, this study is conducted daily. If the statistics are examined for the month of May, it can be shown that the speed and temperature are inversely related, and during June, the same consistency has been seen. On the other hand, the wind speed in July has little effect on the temperature.

Other Two members' observations:

Kuldip Savaliya (1001832000):
She did excellently. She properly described Pandas and its implementation to us. Even she applied some fresh tactics to some assignments. She completed all of the tasks that were given to her in this section, produced a report for them, and provided descriptions for each one.

Shivani Panchiwala(1001982478):
Meghaben Ghanshyambhai Patel worked on Pandas (Python). She did a great job. She gave this assignment everything she had. While carrying out this duty, she worked on every other task and answered every question. Even yet, she considers our viewpoints and explains them to us.