



“Python Programming”

Assignment-4

Topic – Data Analysis and Visualisation with Real-World Weather Data

Submitted by – Meghna Kumar

Roll no - 2501730214

Course – B. Tech CSE (AI & ML)

Section – A

Faculty name- Mr Sameer Farooq

Introduction

Weather conditions play a vital role in agriculture, transportation, health, and environmental planning. With data analysis tools like Python, NumPy, and Pandas, as well as visualisation libraries, we can convert raw datasets into meaningful insights. This project focuses on weather data analysis using data cleaning, statistical computation, and visual representation.

Objectives

- To clean and preprocess the raw weather dataset by handling missing values and formatting dates.

- To calculate important weather statistics such as mean, maximum, minimum temperature, total rainfall, and humidity patterns.

- To generate monthly weather summaries for comparative analysis.

- To visualise weather trends using line plots, bar charts, and scatter plots.

- To export analytical results into files for further interpretation (summary CSV and text report).

Program Description

This project processes and analyses weather data stored in a CSV file using Python. It performs data cleaning, calculates essential statistical values, groups data every month, and visualises important weather patterns.

The script reads the dataset, converts date formats, handles missing values, and generates a cleaned file for further use. The project also creates monthly summaries including temperature trends, rainfall distribution, humidity vs. temperature scatter, and combined weather plots. Results are exported as files like CSV summaries, graphs, and a final report.

Program Code

```
1 import pandas as pd
2 import numpy as np
3 import matplotlib.pyplot as plt
4 import os
5
6 if not os.path.exists("plots"):
7     os.mkdir("plots")
8
9 file_name = "weather_data.csv"
10
11 df = pd.read_csv(file_name)
12
13 print("\n----- RAW DATA LOADED -----")
14 print(df.head(), "\n")
15 print(df.info(), "\n")
16
17 df["date"] = pd.to_datetime(df["date"], errors="coerce")
18 df.fillna(method="bfill", inplace=True)
19
20 df.to_csv("cleaned_weather.csv", index=False)
21
22
23 temp = np.array(df["temperature_c"])
24
25 stats = {
26     "Mean Temperature": temp.mean(),
27     "Max Temperature": temp.max(),
28     "Min Temperature": temp.min(),
29     "Std Deviation": temp.std(),
30     "Total Rainfall": df["rainfall_mm"].sum(),
31     "Mean Humidity": df["humidity_percent"].mean()
32 }
33
34 print("\n----- NUMPY STATISTICS -----")
35 for k, v in stats.items():
36     print(f"{k} : {v}")
```

```
37 df["month"] = df["date"].dt.month
38
39 monthly = df.groupby("month").agg({
40     "temperature_c": ["mean", "min", "max"],
41     "rainfall_mm": "sum",
42     "humidity_percent": "mean"
43 })
44
45
46 monthly.to_csv("monthly_summary.csv")
47
48
49 plt.figure(figsize=(10,4))
50 plt.plot(df["date"], df["temperature_c"], color="red")
51 plt.title("Daily Temperature Trend")
52 plt.xlabel("date")
53 plt.ylabel("temperature (°C)")
54 plt.savefig("plots/temp_trend.png")
55 plt.close()
56
57
58 plt.figure(figsize=(9,4))
59 df.groupby("month")["rainfall_mm"].sum().plot(kind="bar", color="blue")
60 plt.title("Monthly Rainfall")
61 plt.xlabel("Month")
62 plt.ylabel("Rainfall (mm)")
63 plt.savefig("plots/rainfall_bar.png")
64 plt.close()
65
66
67 plt.figure(figsize=(6,4))
68 plt.scatter(df["humidity_percent"], df["temperature_c"], color="purple")
69 plt.title("Humidity vs Temperature")
70 plt.xlabel("humidity (%)")
71 plt.ylabel("temperature (°C)")
72 plt.savefig("plots/humidity scatter.png")
73
```

```
weather_project.py X weather_data.csv X report_summary.txt
WEATHER_DATA_VISUALIZER
> plots
  cleaned_weather.csv
  monthly_summary.csv
  report_summary.txt
  weather_data.csv
  weather_project.py
> OUTLINE

70 plt.scatter(x1, humidity_percent, x2, temperature_c, color='purple')
71 plt.title("humidity vs temperature")
72 plt.xlabel("humidity (%)")
73 plt.ylabel("temperature (°C)")
74 plt.savefig("plots/humidity_scatter.png")
75 plt.close()
76
77 plt.figure(figsize=(10,5))
78 plt.plot(df["date"], df["temperature_c"], label="temperature", color="green")
79 plt.bar(df["date"], df["rainfall_mm"], alpha=0.4, label="rainfall", color="orange")
80 plt.title("Temperature & Rainfall Combined Plot")
81 plt.legend()
82 plt.savefig("plots/combined_plot.png")
83 plt.close()
84
85 with open("report_summary.txt", "w") as f:
86     f.write("===== WEATHER DATA ANALYSIS REPORT =====\n\n")
87     for k, v in stats.items():
88         f.write(f"{k} : {v}\n")
89     f.write("\n----- Monthly Summary ----- \n")
90     f.write(str(monthly))
91
92 print("\n✓ All tasks completed successfully!")
93 print("📁 Outputs generated:\n")
94 print("  - cleaned_weather.csv\n")
95 print("  - monthly_summary.csv\n")
96 print("  - report_summary.txt\n")
97 print("  - plots folder containing 4 graphs\n")
98
99
```

```
weather_data.csv
1 date,temperature_c,rainfall_mm,humidity_percent
2 2024-01-01,14,2.5,71
3 2024-01-02,15,0.0,69
4 2024-01-03,14,1.2,73
5 2024-01-04,13,0.0,75
6 2024-01-05,12,4.3,78
7 2024-01-06,16,0.0,65
8 2024-01-07,17,0.0,64
9 2024-01-08,18,0.0,60
10 2024-01-09,19,0.7,58
11 2024-01-10,20,0.0,55
12 2024-01-11,21,0.0,51
13 2024-01-12,22,0.0,50
14 2024-01-13,23,0.0,40
15 2024-01-14,24,0.0,40
16 2024-01-15,25,0.0,45
17 2024-01-16,26,0.0,43
18 2024-01-17,27,0.0,42
19 2024-01-18,28,0.0,40
20 2024-01-19,29,0.0,39
21 2024-01-20,30,0.0,38
22 2024-01-21,31,1.0,41
23 2024-01-22,28,2.1,46
24 2024-01-23,27,4.1,40
25 2024-01-24,24,0.0,50
26 2024-01-25,22,0.5,51
27 2024-01-26,21,0.0,52
28 2024-01-27,18,1.1,55
29 2024-01-28,17,0.5,50
30 2024-01-29,19,1.2,60
31 2024-01-30,20,0.0,58
32 2024-01-31,22,0.0,56
33 2024-02-01,23,0.1,53
34 2024-02-02,24,0.0,50
35 2024-02-03,25,0.0,47
36 2024-02-04,27,0.0,45
37 2024-02-05,28,0.0,44
```

Sample Output

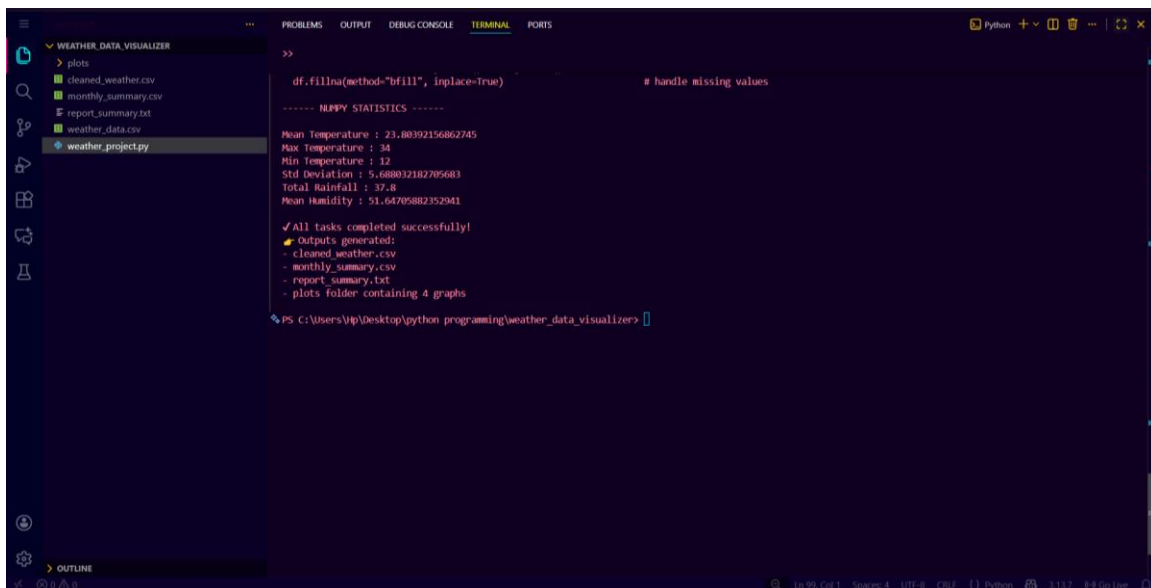
```
PS C:\Users\vip\Desktop\python programming\weather_data_visualizer> python weather_project.py
>>
----- RAW DATA LOADED -----
   date  temperature_c  rainfall_mm  humidity_percent
0 2024-01-01           14          2.5             71
1 2024-01-02           15           0.0             69
2 2024-01-03           14          1.2             73
3 2024-01-04           13           0.0             75
4 2024-01-05           12          4.3             78

c:\class 'pandas.core.frame.DataFrame'
RangeIndex: 51 entries, 0 to 50
Data columns (total 4 columns):
 #   Column                Non-Null Count  Dtype
---  --
 0   date                  51 non-null    object
 1   temperature_c         51 non-null    int64
 2   rainfall_mm           51 non-null    float64
 3   humidity_percent      51 non-null    int64
dtypes: float64(1), int64(2), object(1)
memory usage: 1.7+ KB
None

C:\Users\vip\Desktop\python programming\weather_data_visualizer\weather_project.py:31: FutureWarning: DataFrame.fillna with 'method' is deprecated and will raise
in a future version, use obj.fillna() or obj.bfill() instead.
  df.fillna(method="bfill", inplace=True) # handle missing values

----- Numpy Statistics -----
Mean Temperature : 23.80392156862745
Max Temperature : 34
Min Temperature : 12
Std Deviation : 5.688032182795683
Total Rainfall : 37.8
Mean Humidity : 51.64785882252941

✓ All tasks completed successfully!
```

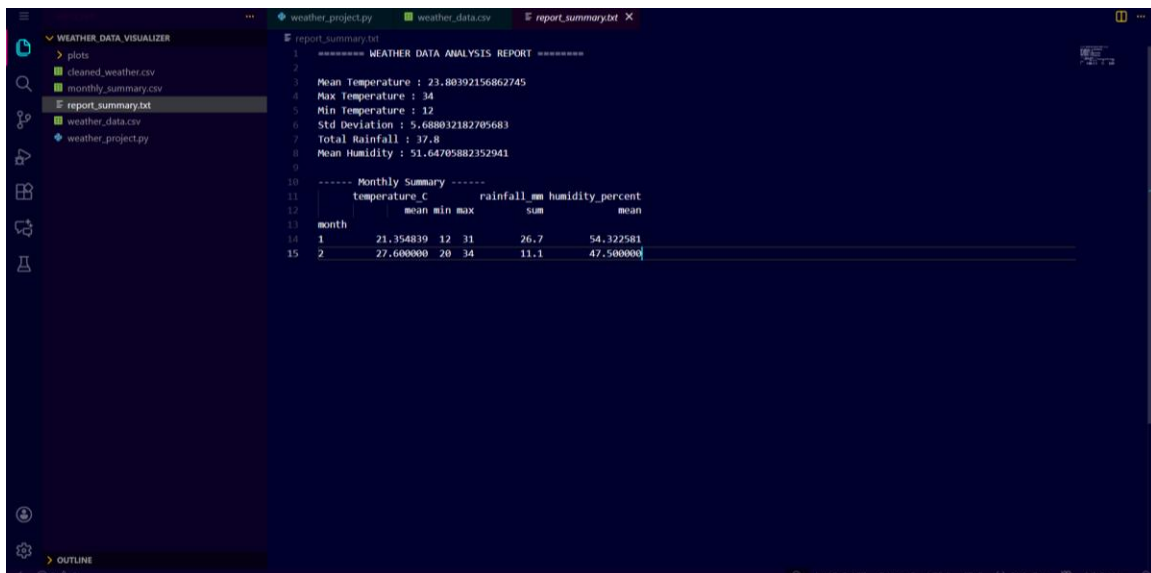


```
>>
df.fillna(method="bfill", inplace=True) # handle missing values

----- NUPPY STATISTICS -----
Mean Temperature : 23.80392156862745
Max Temperature : 34
Min Temperature : 12
Std Deviation : 5.688032182705683
Total Rainfall : 37.8
Mean Humidity : 51.64705882352941

✓ All tasks completed successfully!
✦ Outputs generated:
- cleaned_weather.csv
- monthly_summary.csv
- report_summary.txt
- plots folder containing 4 graphs

PS C:\Users\vip\Desktop\python programming\weather_data_visualizer>
```



```
1 ===== WEATHER DATA ANALYSIS REPORT =====
2
3 Mean Temperature : 23.80392156862745
4 Max Temperature : 34
5 Min Temperature : 12
6 Std Deviation : 5.688032182705683
7 Total Rainfall : 37.8
8 Mean Humidity : 51.64705882352941
9
10 ----- Monthly Summary -----
11 temperature_C      rainfall_mm      humidity_percent
12    mean min max      sum      mean
13 month
14 1      21.354839  12  31      26.7      54.322581
15 2      27.600000  20  34      11.1      47.500000
```

Conclusion

This project successfully analyses weather data by cleaning the dataset, calculating key statistics, and visualising temperature, rainfall, and humidity trends. It shows how Python can effectively convert raw climate data into useful insights for weather monitoring and decision-making.