



"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

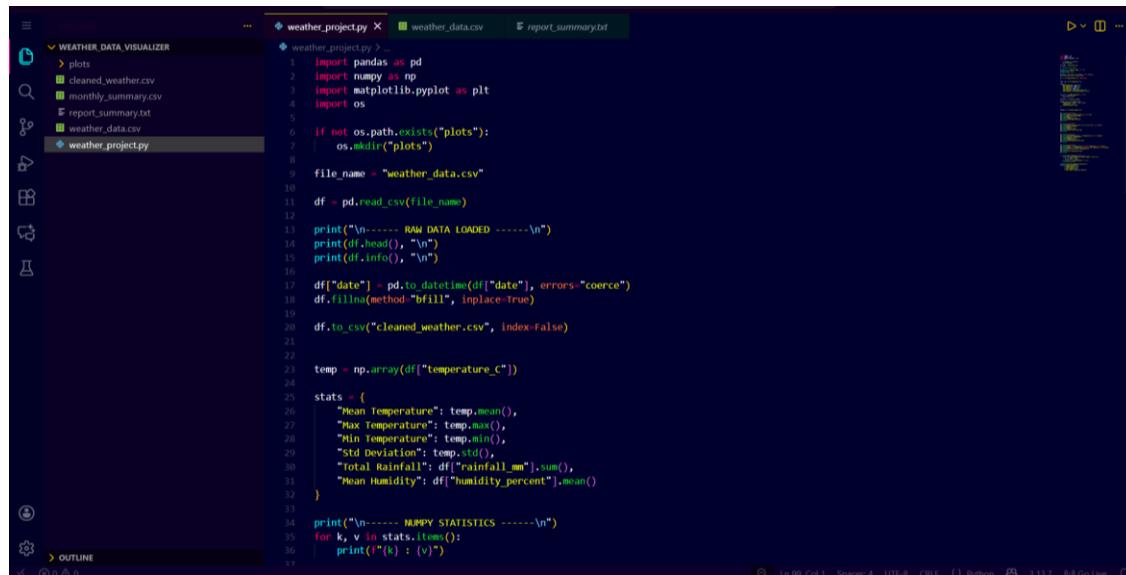
- To clean and preprocess the raw weather dataset by handling missing values and formatting dates.
- To calculate important weather statistics such as mean, maximum, minimum temperature, total rainfall, and humidity patterns.
- To generate monthly weather summaries for comparative analysis.
- To visualise 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

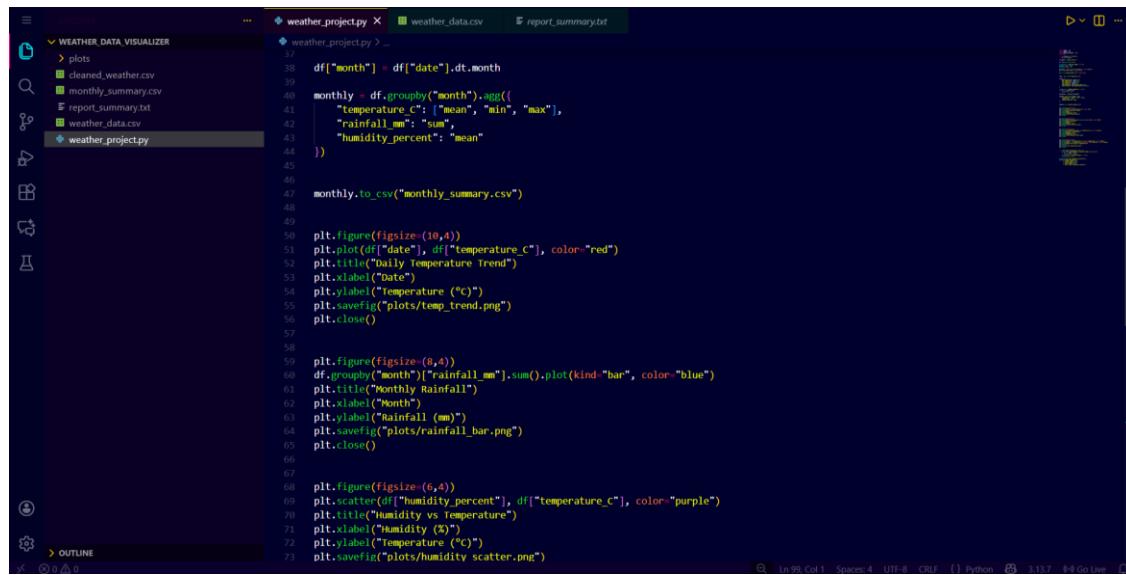


```

WEATHER_DATA_VISUALIZER
plots
cleaned_weather.csv
monthly_summary.csv
report_summary.txt
weather_data.csv
weather_project.py

weather_project.py > ...
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("----- RAW DATA LOADED -----")
14 print(df.head())
15 print(df.info())
16
17 df["date"] = pd.to_datetime(df["date"], errors="coerce")
18 df.fillna(method="ffill", 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("----- NUMPY STATISTICS -----")
35 for k, v in stats.items():
36     print(f"({k}) : ({v})")

```



```

WEATHER_DATA_VISUALIZER
plots
cleaned_weather.csv
monthly_summary.csv
report_summary.txt
weather_data.csv
weather_project.py

weather_project.py > ...
37
38 df["month"] = df["date"].dt.month
39
40 monthly = df.groupby("month").agg({
41     "temperature_C": ["mean", "min", "max"],
42     "rainfall_mm": "sum",
43     "humidity_percent": "mean"
44 })
45
46
47 monthly.to_csv("monthly_summary.csv")
48
49
50 plt.figure(figsize=(10,4))
51 plt.plot(df["date"], df["temperature_C"], color="red")
52 plt.title("Daily Temperature Trend")
53 plt.xlabel("Date")
54 plt.ylabel("Temperature (°C)")
55 plt.savefig("plots/temp_trend.png")
56 plt.close()
57
58
59 plt.figure(figsize=(8,4))
60 df.groupby("month")["rainfall_mm"].sum().plot(kind="bar", color="blue")
61 plt.title("Monthly Rainfall")
62 plt.xlabel("Month")
63 plt.ylabel("Rainfall (mm)")
64 plt.savefig("plots/rainfall_bar.png")
65 plt.close()
66
67
68 plt.figure(figsize=(6,4))
69 plt.scatter(df["humidity_percent"], df["temperature_C"], color="purple")
70 plt.title("Humidity vs Temperature")
71 plt.xlabel("Humidity (%)")
72 plt.ylabel("Temperature (°C)")
73 plt.savefig("plots/humidity_scatter.png")

```

```

// weather_project.py
# Importing required libraries
import pandas as pd
import matplotlib.pyplot as plt
from datetime import datetime

# Load data
df = pd.read_csv('cleaned_weather.csv')
df['date'] = pd.to_datetime(df['date'])

# Create scatter plot for humidity vs temperature
plt.figure(figsize=(10, 6))
plt.scatter(df['date'], df['humidity_percent'], color='purple')
plt.title("Humidity vs Temperature")
plt.xlabel("Date")
plt.ylabel("Humidity (%)")
plt.savefig("plots/humidity_scatter.png")
plt.close()

# Create combined plot for temperature & rainfall
plt.figure(figsize=(10, 6))
plt.plot(df['date'], df['temperature_C'], label="Temperature", color="green")
plt.bar(df['date'], df['rainfall_mm'], alpha=0.4, label="Rainfall", color="orange")
plt.title("Temperature & Rainfall Combined Plot")
plt.xlabel("Date")
plt.legend()
plt.savefig("plots/combined_plot.png")
plt.close()

# Generate report summary
with open("report_summary.txt", "w") as f:
    f.write("----- DATA ANALYSIS REPORT -----\n")
    for row in stats.items():
        f.write(f"{row[0]} : {row[1]}\n")
    f.write("\n----- Monthly Summary ----- \n")
    f.write(str(monthly))

print("\n All tasks completed successfully!")
print("Outputs generated:")
print("- cleaned_weather.csv")
print("- monthly_summary.csv")
print("- report_summary.txt")
print("- plots folder containing 4 graphs\n")

```

Date	Temperature (C)	Rainfall (mm)	Humidity (%)
2024-01-01	14	2.5	71
2024-01-02	15	0.0	69
2024-01-03	14	1.2	73
2024-01-04	13	0.0	75
2024-01-05	12	4.3	78
2024-01-06	16	0.0	65
2024-01-07	17	0.0	63
2024-01-08	18	0.0	60
2024-01-09	19	0.7	58
2024-01-10	20	0.0	55
2024-01-11	21	0.0	51
2024-01-12	22	0.0	50
2024-01-13	23	0.0	48
2024-01-14	24	0.0	46
2024-01-15	25	0.0	45
2024-01-16	26	0.0	43
2024-01-17	27	0.0	42
2024-01-18	28	0.0	40
2024-01-19	29	0.0	39
2024-01-20	30	0.0	38
2024-01-21	31	1.0	41
2024-01-22	28	2.3	46
2024-01-23	27	4.1	48
2024-01-24	24	0.0	50
2024-01-25	21	0.0	51
2024-01-26	21	0.0	52
2024-01-27	18	3.3	55
2024-01-28	17	5.6	58
2024-01-29	19	1.2	60
2024-01-30	20	0.0	58
2024-01-31	22	0.0	56
2024-02-01	23	0.3	53
2024-02-02	24	0.0	50
2024-02-03	25	0.0	47
2024-02-04	27	0.6	45
2024-02-05	28	0.8	48

Sample Output

```

// Terminal Output
PS C:\Users\lip\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

<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: 1.74 KB
None

C:\Users\lip\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="ffill", inplace=True)          # handle missing values

----- NUMPY STATISTICS -----
Mean Temperature : 23.80392156862745
Max Temperature : 34
Min Temperature : 12
Std Deviation : 5.688032182709681
Total Rainfall : 37.8
Mean Humidity : 51.64705882352941

All tasks completed successfully!

```

The screenshot shows the VS Code interface with the 'TERMINAL' tab selected. The terminal window displays the following output:

```

>>>
df.fillna(method="ffill", inplace=True) # handle missing values
----- NUMPY STATISTICS -----
Mean Temperature : 23.88392156862745
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\hp\Desktop\python programming\weather_data_visualizer>

```

The status bar at the bottom indicates the file is open in Python mode, with 3137 lines of code.

The screenshot shows the VS Code interface with the 'report_summary.txt' file open in the editor. The content of the file is as follows:

```

=====
WEATHER DATA ANALYSIS REPORT =====
1 Mean Temperature : 23.88392156862745
2 Max Temperature : 34
3 Min Temperature : 12
4 Std Deviation : 5.688032182705683
5 Total Rainfall : 37.8
6 Mean Humidity : 51.64705882352941
7
8 ----- Monthly Summary -----
9
10      temperature_c      rainfall_mm humidity_percent
11      mean min max      sum      mean
12      month
13
14      1      21.354839 12 31      26.7      54.322581
15      2      27.600000 28 34      11.1      47.500000

```

The status bar at the bottom indicates the file is open in Plain Text mode, with 1011 lines of code.

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.