"Python Programming"

Assignment-4

Topic – Data Analysis and Visualisation with RealWorld Weather Data

Submitted by – Priyam Sharma

Roll no - 2501730184

Course – B. Tech CSE (AI & ML)

Section – B

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.

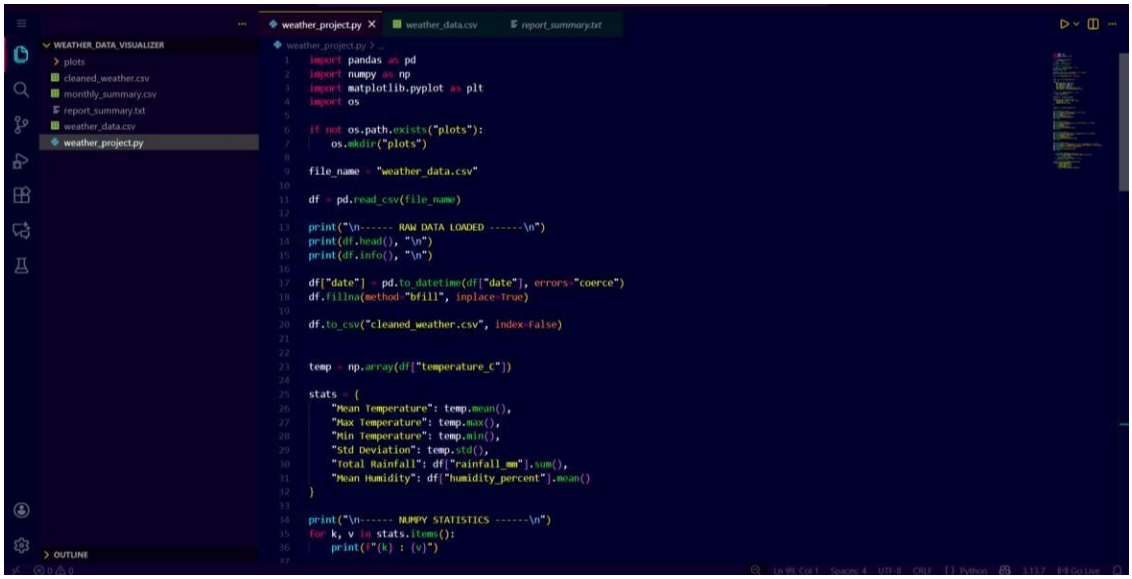
To calculate important weather statistics such as mean, maximum, minimum temperature, total rainfall, and humidity patterns.
-To generate monthly weather summaries for comparative analysis.
-To visualise weather trends using line plots, bar charts, and scatter plots.
-To export analytical results into files for further interpretation (summary CSV and text report).

Program Description

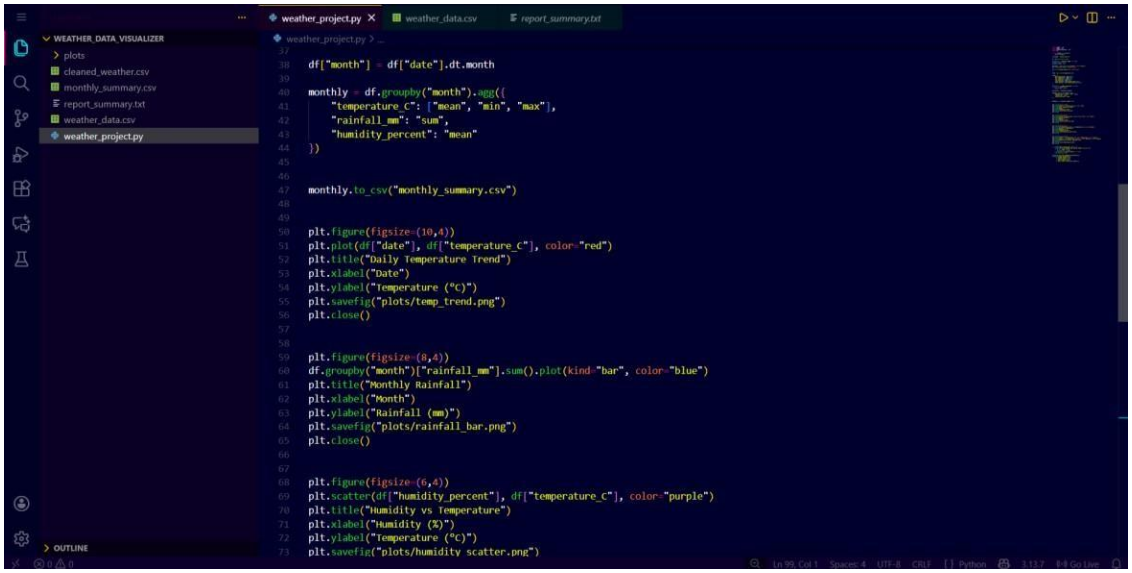
This project processes and analyses weather data stored in a CSV file using Python. It performs data cleaning, calculates essential statistical values, groups data every month, and visualises important weather patterns.
The script reads the dataset, converts date formats, handles missing values, and 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.

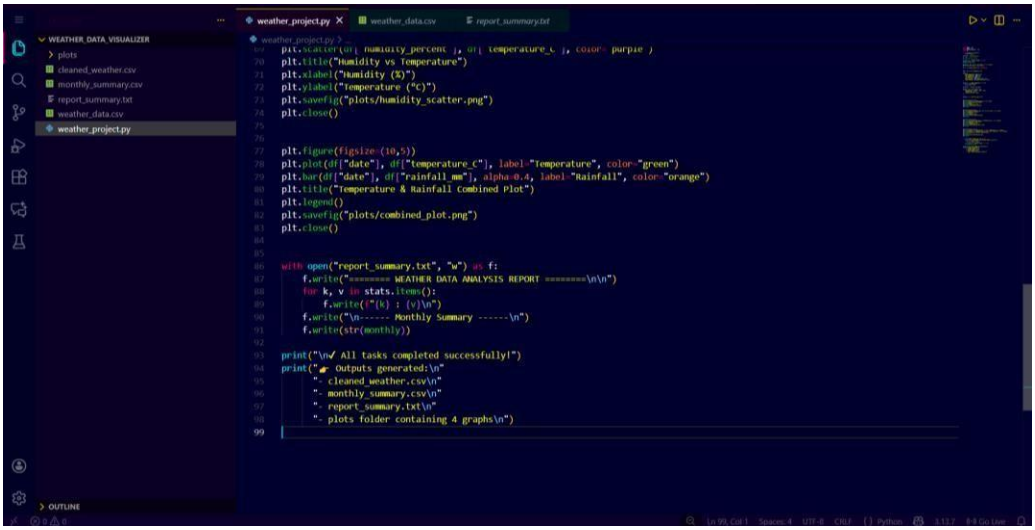
Program Code



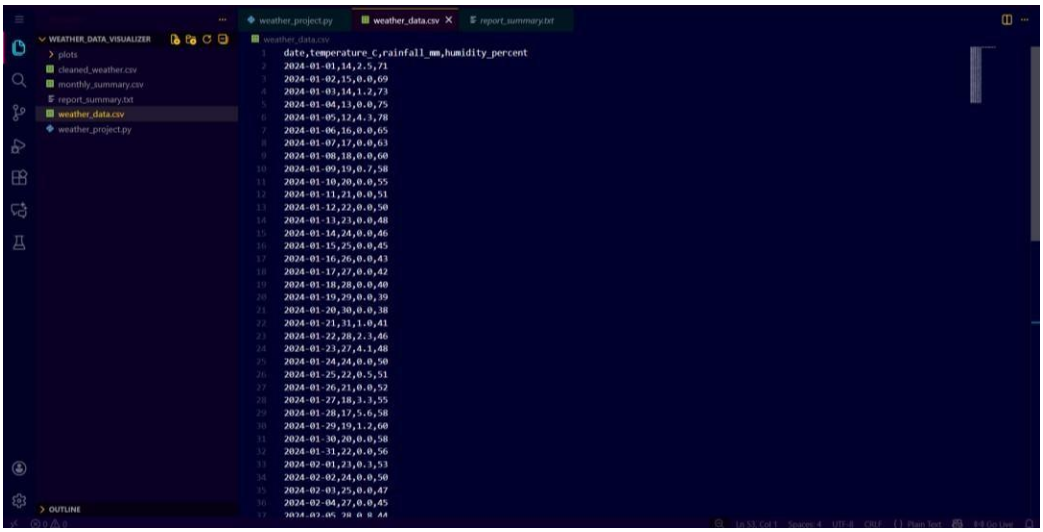
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import os

if not os.path.exists("plots"):
    os.mkdir("plots")

file_name = "weather_data.csv"

df = pd.read_csv(file_name)

print("\n------ RAW DATA LOADED ------\n")
print(df.head(), "\n")
print(df.info(), "\n")

df["date"] = pd.to_datetime(df["date"], errors="coerce")
df.fillna(method="bfill", inplace=True)

df.to_csv("cleaned_weather.csv", index=False)


temp = np.array(df["temperature_C"])

stats = {
    "Mean Temperature": temp.mean(),
    "Max Temperature": temp.max(),
    "Min Temperature": temp.min(),
    "Std Deviation": temp.std(),
    "Total Rainfall": df["rainfall_mm"].sum(),
    "Mean Humidity": df["humidity_percent"].mean()
}

print("\n------ NUMPY STATISTICS ------\n")
for k, v in stats.items():
    print(f"{k} : {v}")
```



```
df["month"] = df["date"].dt.month

monthly = df.groupby("month").agg({
    "temperature_C": ["mean", "min", "max"],
    "rainfall_mm": "sum",
    "humidity_percent": "mean"
})


monthly.to_csv("monthly_summary.csv")


plt.figure(figsize=(10,4))
plt.plot(df["date"], df["temperature_C"], color="red")
plt.title("Daily Temperature Trend")
plt.xlabel("Date")
plt.ylabel("Temperature (°C)")
plt.savefig("plots/temp_trend.png")
plt.close()


plt.figure(figsize=(8,4))
df.groupby("month")["rainfall_mm"].sum().plot(kind="bar", color="blue")
plt.title("Monthly Rainfall")
plt.xlabel("Month")
plt.ylabel("Rainfall (mm)")
plt.savefig("plots/rainfall_bar.png")
plt.close()


plt.figure(figsize=(6,4))
plt.scatter(df["humidity_percent"], df["temperature_C"], color="purple")
plt.title("Humidity vs Temperature")
plt.xlabel("Humidity (%)")
plt.ylabel("Temperature (°C)")
plt.savefig("plots/humidity scatter.png")
```

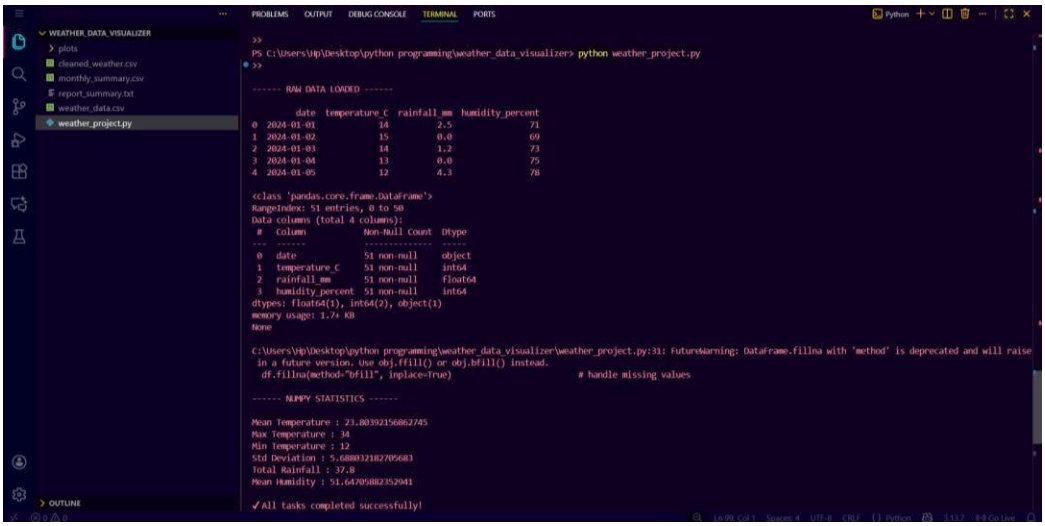


```
plt.scatter(df["humidity_percent"], df["temperature_C"], color="purple")
plt.title("Humidity vs Temperature")
plt.xlabel("Humidity (%)")
plt.ylabel("Temperature (°C)")
plt.savefig("plots/humidity_scatter.png")
plt.close()


plt.figure(figsize=(10,5))
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.legend()
plt.savefig("plots/combined_plot.png")
plt.close()


with open("report_summary.txt", "w") as f:
    f.write("======== WEATHER DATA ANALYSIS REPORT ========\n\n")
    for k, v in stats.items():
        f.write(f"{k} : {v}\n")
    f.write("\n------ Monthly Summary ------\n")
    f.write(str(monthly))

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

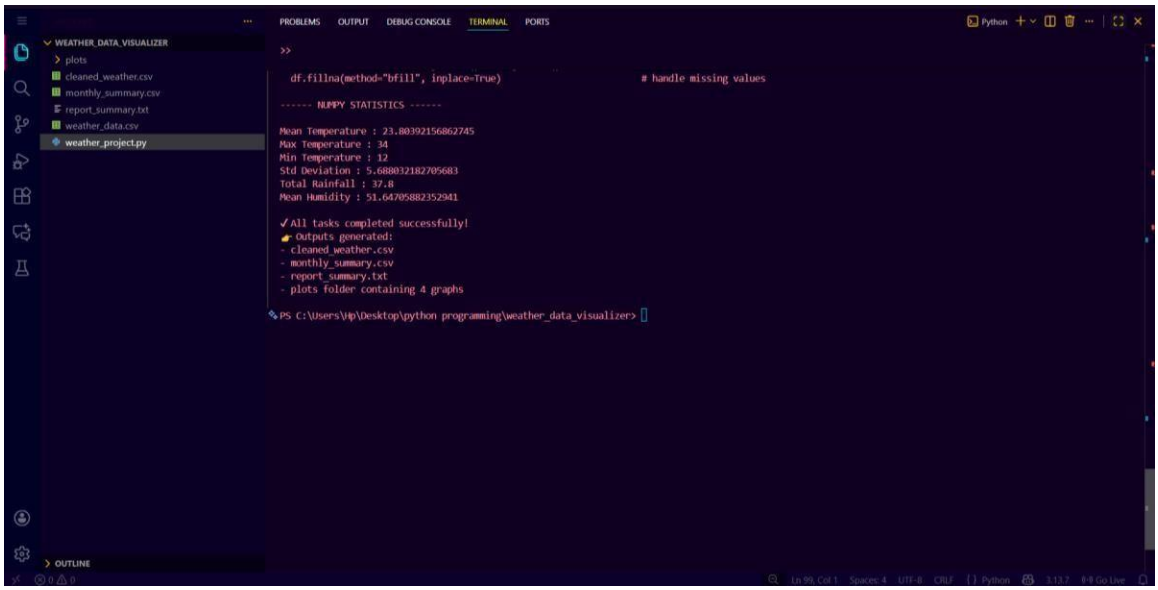


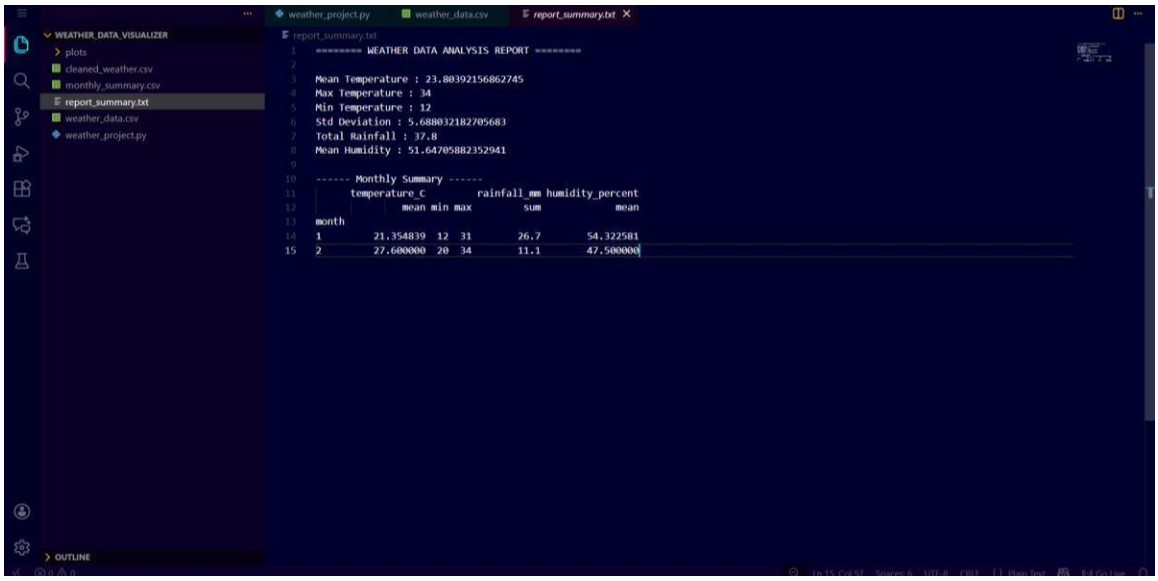
```
date,temperature_C,rainfall_mm,humidity_percent
2024-01-01,14,2.5,71
2024-01-02,15,0.0,69
2024-01-03,14,1.2,73
2024-01-04,13,0.0,75
2024-01-05,12,4.3,78
2024-01-06,16,0.0,65
2024-01-07,17,0.0,63
2024-01-08,18,0.0,60
2024-01-09,19,0.7,58
2024-01-10,20,0.0,55
2024-01-11,21,0.0,51
2024-01-12,22,0.0,50
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
2024-01-25,22,0.5,51
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
2024-02-01,23,0.1,53
2024-02-02,24,0.0,50
2024-02-03,25,0.0,47
2024-02-04,27,0.0,45
```

Sample Output



```
PS C:\Users\Hp\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: 1.7+ KB
None

C:\Users\Hp\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.ffill() or obj.bfill() instead.
  df.fillna(method="bfill", inplace=True)                    # handle missing values

------ NUMPY STATISTICS ------

Mean Temperature : 23.803921568627452
Max Temperature : 34
Min Temperature : 12
Std Deviation : 5.688032182705683
Total Rainfall : 37.8
Mean Humidity : 51.64705882352941

✓All tasks completed successfully!
```





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.