Department of CSE(AI) and CSE(AIML)

A
Project Report

On

Weather Data Analysis –
Visualize temperature trends, rainfall, and seasonal patterns from a small weather dataset.

By

KESHAV AGARWAL

(202401100400107)

Date: 10th March 2025

1. **Introduction**

I have been assigned to create an AI using python, that can analyse data and display a structured graph to visualize the given data set. It is accomplished using inbuilt libraries like "Pandas", "Matplotlib", "Seaborn" and "time"

2. **Dataset Overview**

Current data set is retrieved from "Kaggle" ; it is openly available as "Seattle-weather.csv" on Kaggle. This data set contains daily weather data of Seattle city.

- Its key features include:
  1. Date
  2. Precipitation
  3. Max. Temperature
  4. Min. Temperature
  5. Wind
  6. Weather condition

3. **Data Preprocessing**

Using the inbuilt library "Pandas", we would be reading the dataset . Also using "Pandas" date column is converted into datetime format. Checked for any missing value, if there any how to handle that. Finally generating a basic summary of the data and printing that before the actual plots.

4. **Data Visualization and analysis**

Data visualization is an important part, it helps understand the trends and patterns in the weather. It makes the data interpretation easier for the user.

5. **Implementation in python**

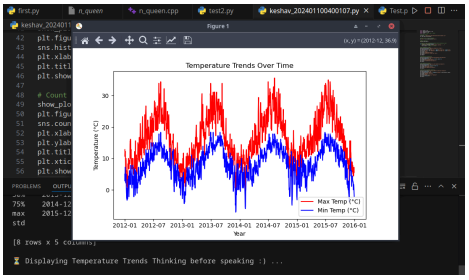
```
import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

import time # Import time for delay

# Load the dataset

file_path =
"/home/keshav-0211/Downloads/seattle-weather.csv"

df = pd.read_csv(file_path)

# Convert date column to datetime

df['date'] = pd.to_datetime(df['date'])

# Basic statistics

print(df.describe())

# Function to show a message and delay before
displaying a plot

def show_plot_with_delay(message):

    print(f"\n ⏳ {message} Thinking before
speaking :) ...")

    time.sleep(2) # Delay of 2 seconds

# Plot temperature trends

show_plot_with_delay("Displaying Temperature
Trends")

plt.figure(figsize=(12, 5))

plt.plot(df['date'], df['temp_max'], label="Max Temp
(°C)", color="red")
```

```
plt.plot(df['date'], df['temp_min'], label="Min Temp (°C)", color="blue")

plt.xlabel("Year")

plt.ylabel("Temperature (°C)")

plt.title("Temperature Trends Over Time")

plt.legend()

plt.show()

# Histogram of precipitation

show_plot_with_delay("Displaying Precipitation Distribution")

plt.figure(figsize=(8, 5))

sns.histplot(df['precipitation'], bins=30, kde=True)

plt.xlabel("Precipitation (mm)")

plt.title("Precipitation Distribution")

plt.show()

# Wind speed distribution

show_plot_with_delay("Displaying Wind Speed Distribution")

plt.figure(figsize=(8, 5))

sns.histplot(df['wind'], bins=20, kde=True)

plt.xlabel("Wind Speed (m/s)")

plt.title("Wind Speed Distribution")

plt.show()

# Count plot for weather conditions

show_plot_with_delay("Displaying Weather Condition Frequency")

plt.figure(figsize=(8, 5))

sns.countplot(x=df['weather'], order=df['weather'].value_counts().index)

plt.xlabel("Weather Condition")

plt.ylabel("Count")

plt.title("Weather Condition Frequency")

plt.xticks(rotation=45)

plt.show()
```

6. Key Observations and insights

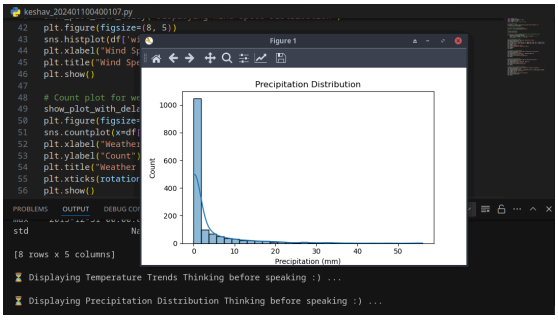
6.1 Temperature

We have used a line chat to present the data over the years. Multiple colors being used to determine the different settings. We could easily determine the variation in temperature over time, see seasonal patterns of temperature changes and the gap between maximum and minimum temperature.



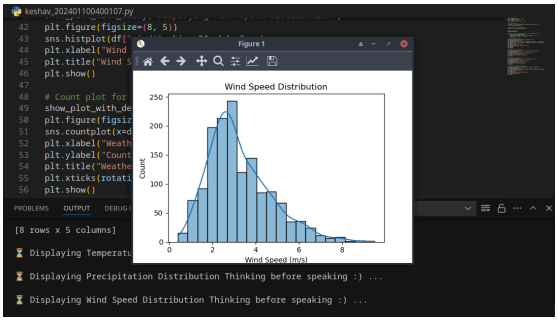
6.2 Precipitation

We have used a histogram to depict precipitation from the data. We could find from the given data that most days got little to no precipitation. A few days show extremely high rainfall.



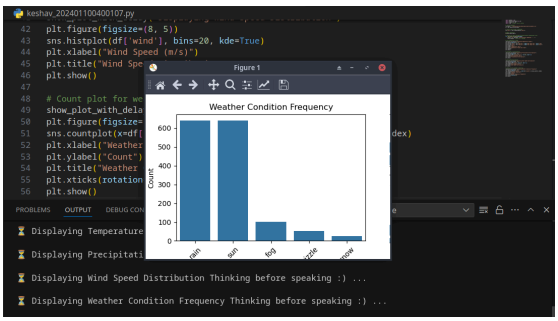
6.3 Wind Speed

We have used a histogram to depict precipitation from the data. We could find that most wind speeds are within a certain range. A very rare occurrence of high wind speed



6.4 Weather Conditions

We have used a bar chart to depict weather conditions from the data. We can conclude that some conditions are more common than others.



7. Challenges and improvement

Major challenges faced were data handling and ensuring accurate visualization.
Improvement can be made by making the graph attractive, a web-based dashboard can be made. Machine learning can be used to predict future weather.

8. Conclusion

Weather analysis can be helpful in every aspect, be it farming, construction or any other field. Using this weather data analysis, it could be helpful for all

9. References

Data - https://www.kaggle.com/datasets/ananthr1/weather-prediction/data

Code - https://www.chatgpt.com

Pandas - https://pandas.pydata.org/

Matplotlib - https://matplotlib.org/

Seaborn - https://seaborn.pydata.org/