Introduction to AI

Weather Data Analysis Report

by

Tanmay Kesharwani

CSEAI-D

Class roll no. 41

The Problem

1. **Data Acquisition and Quality:**
   Weather data typically comes from various sources and can be inconsistent, contain missing values, or have varying formats. This poses challenges in reliably ingesting and cleaning the data.
2. **Data Heterogeneity:**
   The dataset might include different types of information—such as temperature, humidity, rainfall, and wind speed—each with its own units and scales. Integrating these varied elements into a coherent analysis requires careful normalization and handling.
3. **Identifying Patterns and Trends:**
   Weather data is inherently time-dependent. The goal is to identify seasonal trends, anomalies, and relationships between different meteorological variables. This involves both statistical summarization and visual exploration.
4. **Interpretability and Actionability:**
   Beyond just processing the data, the analysis must provide insights that are interpretable by decision-makers. This means the system should not only compute statistics but also generate visualizations and summaries that clearly communicate trends and potential anomalies.

The Approach

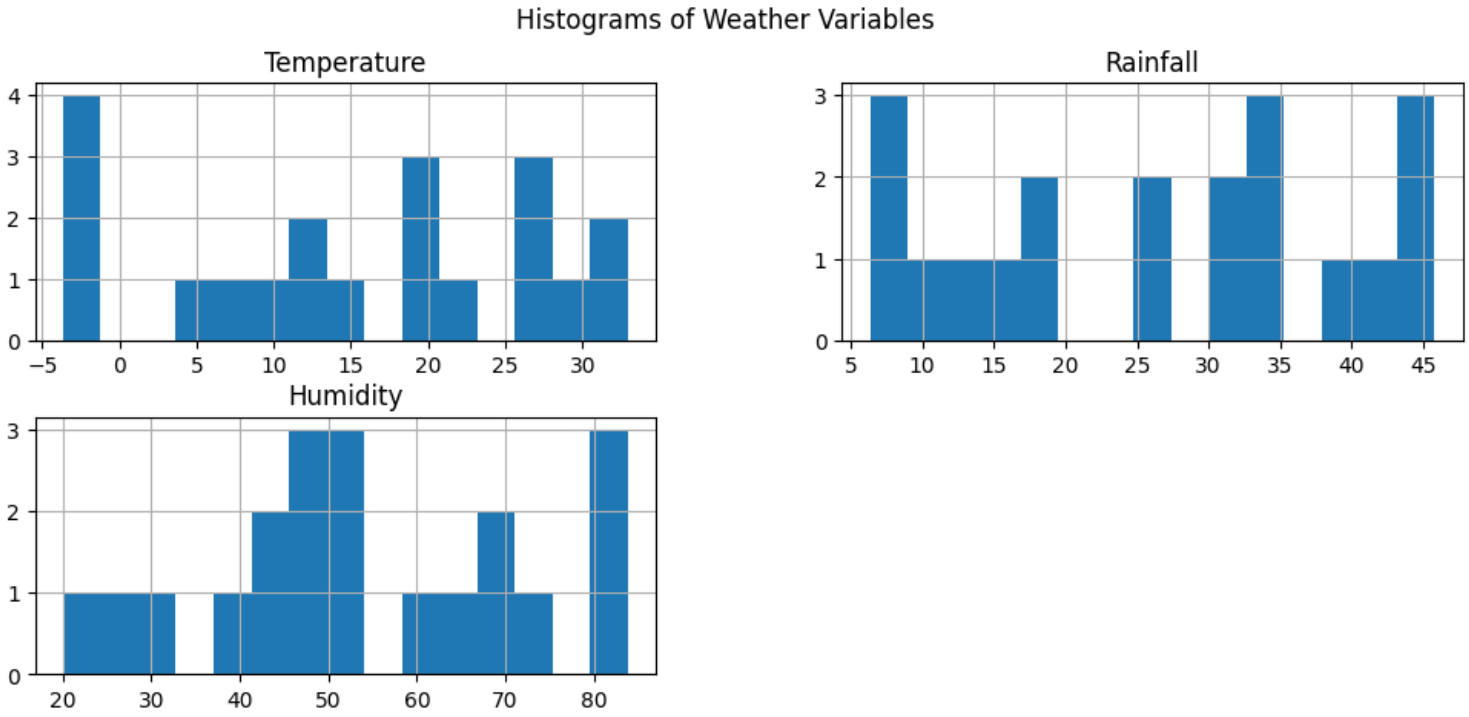
1. **Data Ingestion and Cleaning:**
    - **Collection:** Gather data from reliable sources such as weather stations, satellites, or public datasets.
    - **Cleaning:** Address missing values, outliers, or errors in the dataset to ensure the data is reliable for analysis.
    - **Normalization:** Convert different units and scales to a consistent format to allow for meaningful comparisons.
2. **Exploratory Data Analysis (EDA):**
    - **Descriptive Statistics:** Compute measures like mean, median, standard deviation, and quartiles to understand the distribution of weather variables.
    - **Visualizations:** Use plots such as histograms, line charts, and scatter plots to reveal patterns and trends over time, and to explore relationships between different weather parameters.
    - **Correlation Analysis:** Identify how variables interact, for example, how temperature might correlate with humidity or rainfall.

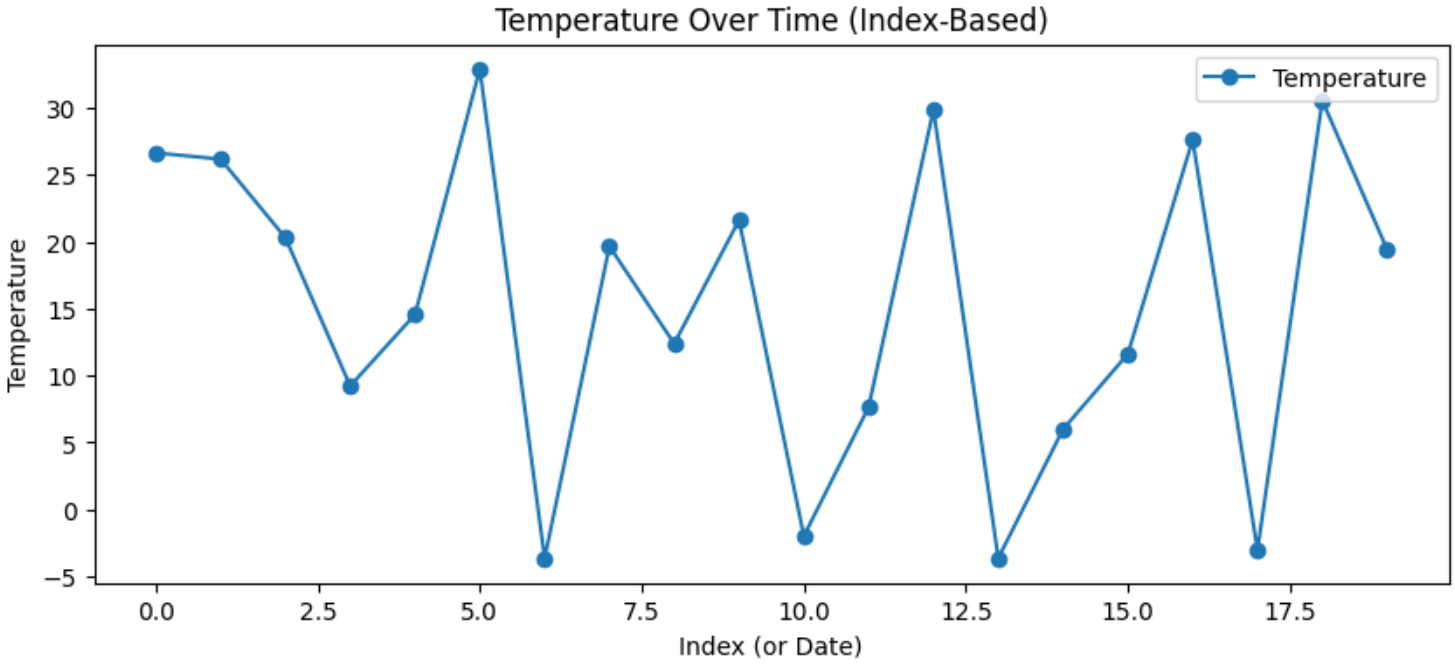
3. **Pattern Recognition and Trend Analysis:**
    - **Time-Series Analysis:** Examine the data over time to identify seasonal patterns, trends, or cyclic behaviours.
    - **Anomaly Detection:** Highlight unusual weather events or outliers that deviate significantly from typical patterns. These could indicate rare events or potential data quality issues.
4. **Interpretation and Communication:**
    - **Summary Reports:** Generate clear summaries and visual reports that convey key insights—such as typical weather patterns, unexpected anomalies, or the relationship between different variables—to help inform decision-making.
    - **Actionable Insights:** Ensure that the analysis provides value, whether for planning, forecasting, or understanding climate trends, by highlighting significant observations that can lead to further investigation or action.

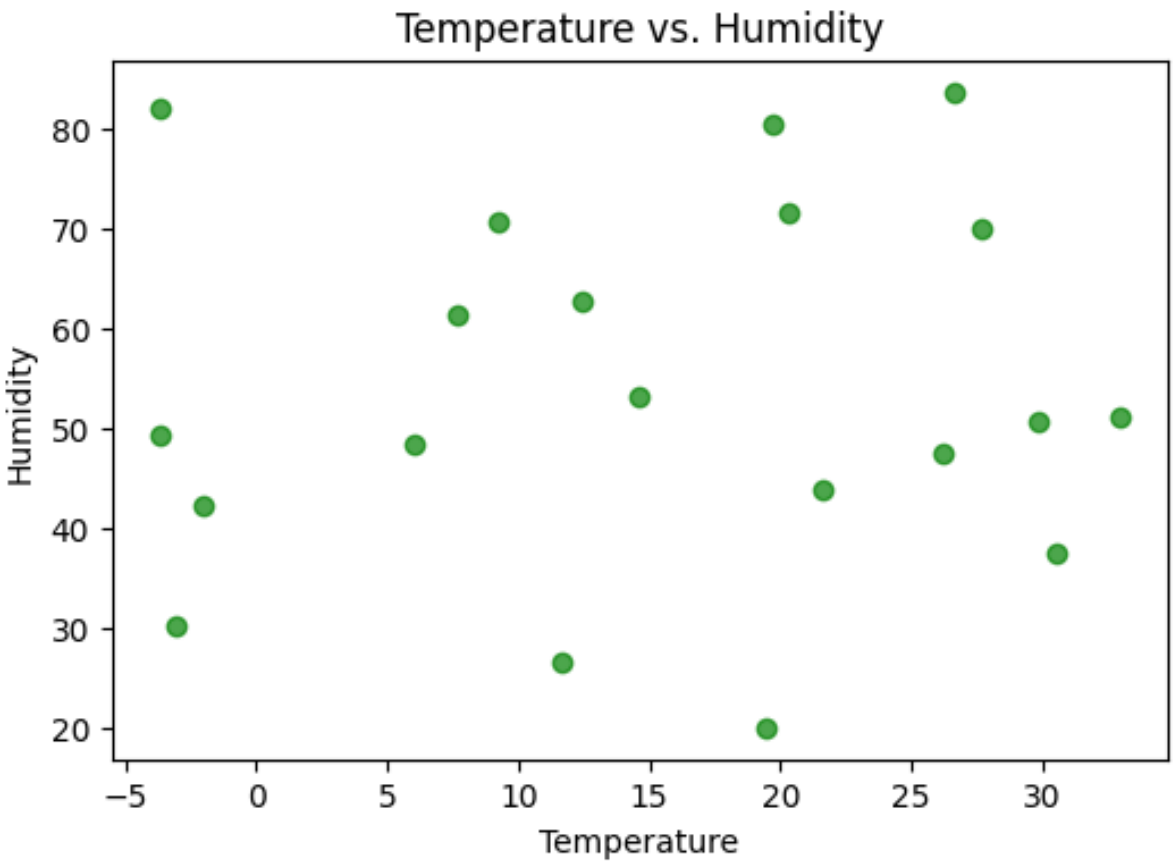
Code

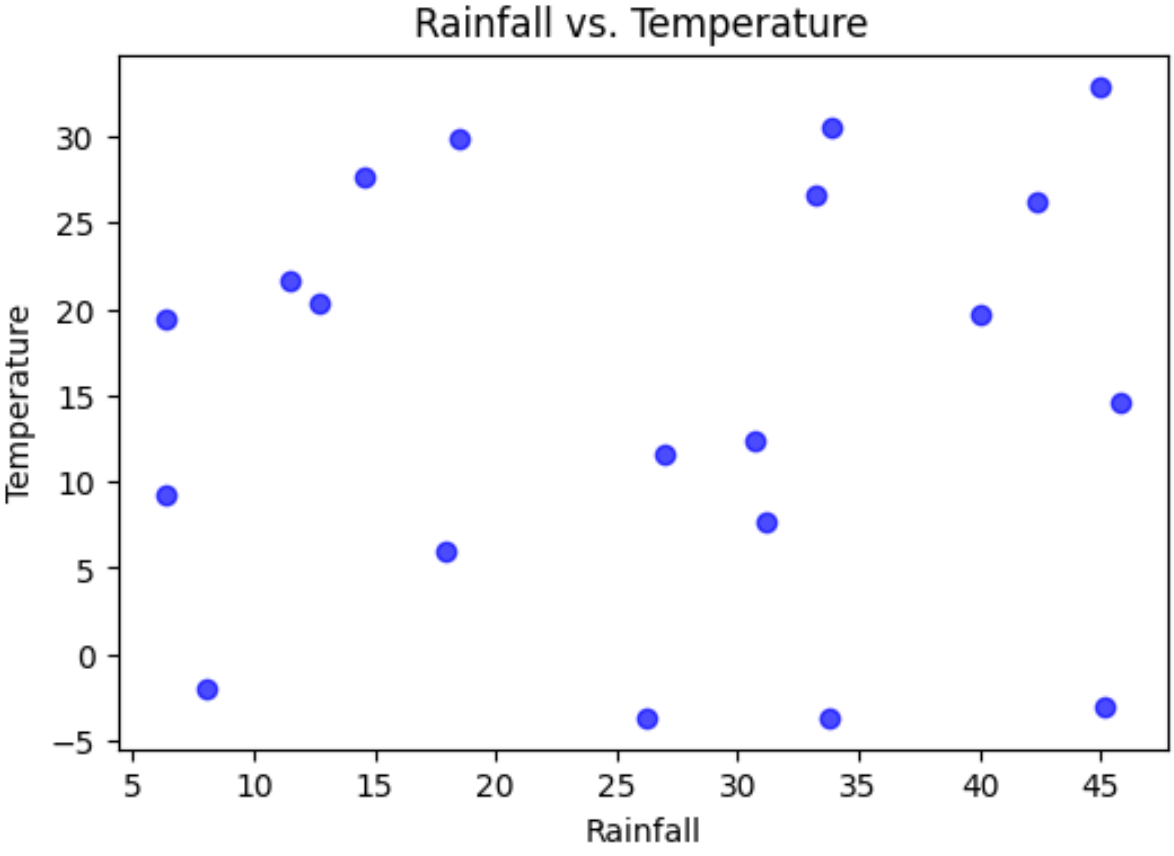
```
# Import necessary libraries

import pandas as pd  # Used for data manipulation and analysis

import numpy as np  # Used for numerical operations

import matplotlib.pyplot as plt  # Used for creating visualizations


# 1. Load the dataset

df = pd.read_csv('weather_data.csv')  # Reads data from a CSV file named 'weather_data.csv' into a pandas DataFrame called 'df'


# 2. Visualization

# 2a. Histogram of each numerical feature

numeric_cols = ['Temperature', 'Rainfall', 'Humidity']  # Define the columns containing numerical weather data

df[numeric_cols].hist(bins=15, figsize=(12, 5))  # Create histograms for each numerical column with 15 bins

plt.suptitle("Histograms of Weather Variables")  # Set the title for the entire histogram plot

plt.show()  # Display the histograms


# 2b. Line plot of temperature over index or date

plt.figure(figsize=(10, 4))  # Create a new figure with a specific size

plt.plot(df['Temperature'], marker='o', label='Temperature')  # Plot temperature data as a line with circular markers

plt.title("Temperature Over Time (Index-Based)")  # Set the plot title

plt.xlabel("Index (or Date)")  # Set the x-axis label

plt.ylabel("Temperature")  # Set the y-axis label

plt.legend()  # Display a legend to identify the data being plotted

plt.show()  # Display the line plot
```

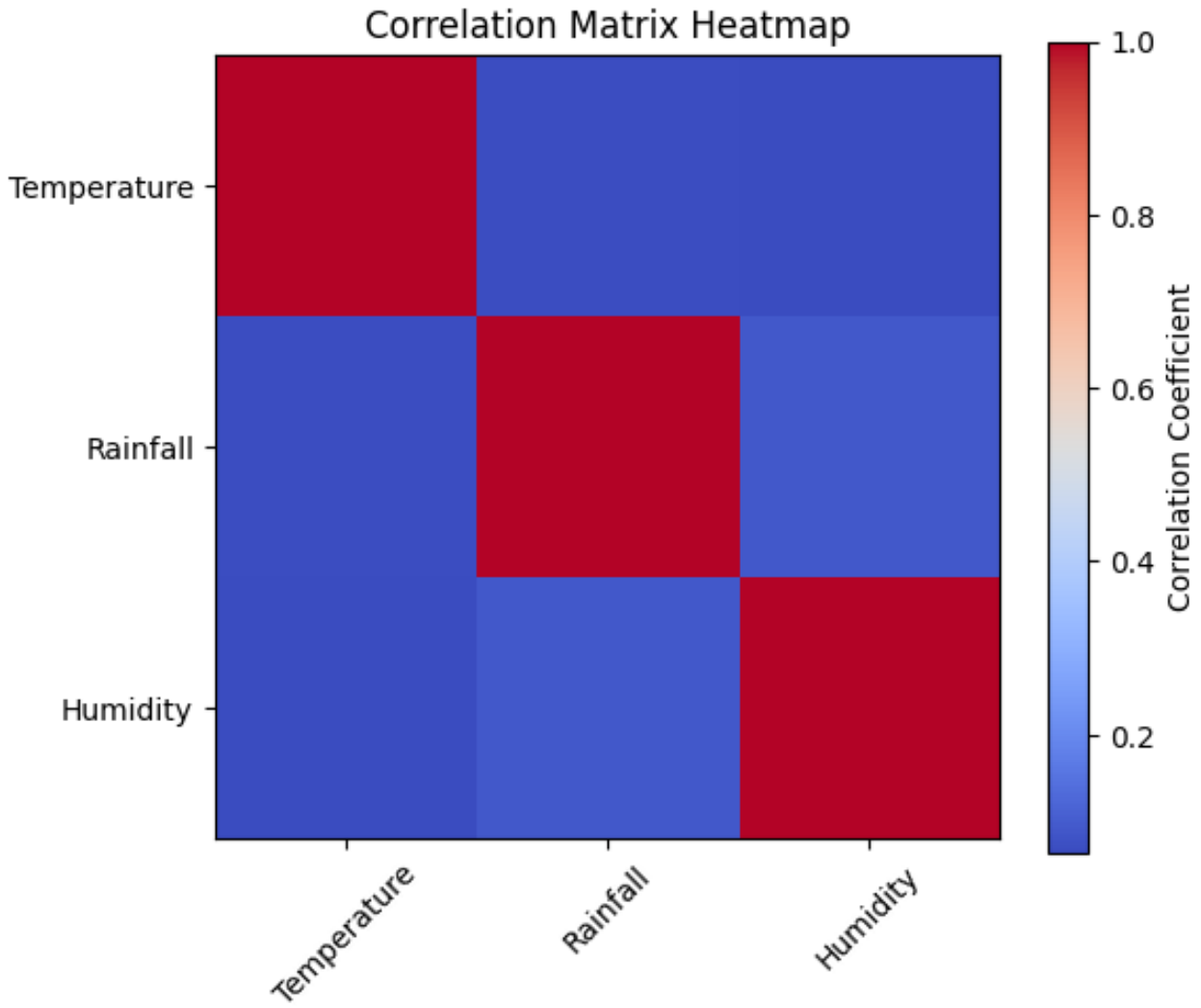
```
# 2c. Scatter plot: Temperature vs. Humidity
plt.figure(figsize=(6, 4))  # Create a new figure with a specific size
plt.scatter(df['Temperature'], df['Humidity'], alpha=0.7, c='green')  # Create a scatter plot of Temperature vs. Humidity
plt.title("Temperature vs. Humidity")  # Set the plot title
plt.xlabel("Temperature")  # Set the x-axis label
plt.ylabel("Humidity")  # Set the y-axis label
plt.show()  # Display the scatter plot

# 2d. Scatter plot: Rainfall vs. Temperature
plt.figure(figsize=(6, 4))  # Create a new figure with a specific size
plt.scatter(df['Rainfall'], df['Temperature'], alpha=0.7, c='blue')  # Create a scatter plot of Rainfall vs. Temperature
plt.title("Rainfall vs. Temperature")  # Set the plot title
plt.xlabel("Rainfall")  # Set the x-axis label
plt.ylabel("Temperature")  # Set the y-axis label
plt.show()  # Display the scatter plot

# 3. Correlation matrix (purely for numeric insight)
correlation_matrix = df[numeric_cols].corr()  # Calculate the correlation between numerical columns

# Heatmap of correlation matrix
plt.figure(figsize=(6, 5))  # Create a new figure with a specific size
plt.imshow(correlation_matrix, cmap='coolwarm', interpolation='none')  # Create a heatmap of the correlation matrix
plt.colorbar(label='Correlation Coefficient')  # Add a colorbar to the heatmap
plt.xticks(range(len(numeric_cols)), numeric_cols, rotation=45)  # Set x-axis ticks and labels
plt.yticks(range(len(numeric_cols)), numeric_cols)  # Set y-axis ticks and labels
plt.title("Correlation Matrix Heatmap")  # Set the plot title
plt.show()  # Display the heatmap
```


Histograms of Weather Variables
Temperature
Rainfall
Humidity


Temperature Over Time (Index-Based)
Temperature
Temperature
Index (or Date)


Temperature vs. Humidity
Humidity
Temperature
80
70
60
50
40
30
20
−5
0
5
10
15
20
25
30


Rainfall vs. Temperature
Temperature
Rainfall
30
25
20
15
10
5
0
−5
5
10
15
20
25
30
35
40
45



References:

- ***OpenAI ChatGPT. (2025, March 11).*** *Explanation of the problem and approach used to create a weather analysis system.*
- ***Sample Weather Dataset.*** *Embedded weather data used for analysis. (Provided by the researcher).*