INDIAN INSTITUTE OF INFORMATION TECHNOLOGY RANCHI

3RD SEMESTER PYTHON PROJECT

A PYTHON BASED WEATHER ANALYZER

Team members-

KUMAR HARSH(2023UG2023)

PRABHAT KUMAR(2023UG2011)

SEC-C

PROJECT GIVEN BY-

SHIVANG TRIPATHI SIR

ABOUT THE PROJECT

A PYTHON BASED WEATHER ANALYZER

This Weather Analyzer project uses Python to analyze temperature data for a week. Key functionalities include:

- Simulation of daily temperatures (°C)

- Calculation of basic statistics: mean, median, min, max, and standard deviation

- Identification of the hottest and coldest days

- Visualization of the weekly temperature trend using Matplotlib.

- Common libraries are used such as numpy and Matplotlib.

EXPLANATION OF THE PROJECT.

This Python code analyzes and visualizes simulated weekly temperature data using NumPy and Matplotlib libraries.

1. NumPy: Used for statistical computations and efficient array manipulations, such as calculating mean, median, minimum, maximum, and standard deviation.
2. Matplotlib: Used to create detailed visualizations, including a line chart, pie chart, and heatmap.

The program generates random temperatures for a week, computes basic statistics, identifies the hottest and coldest days, and displays these in a tabular format.

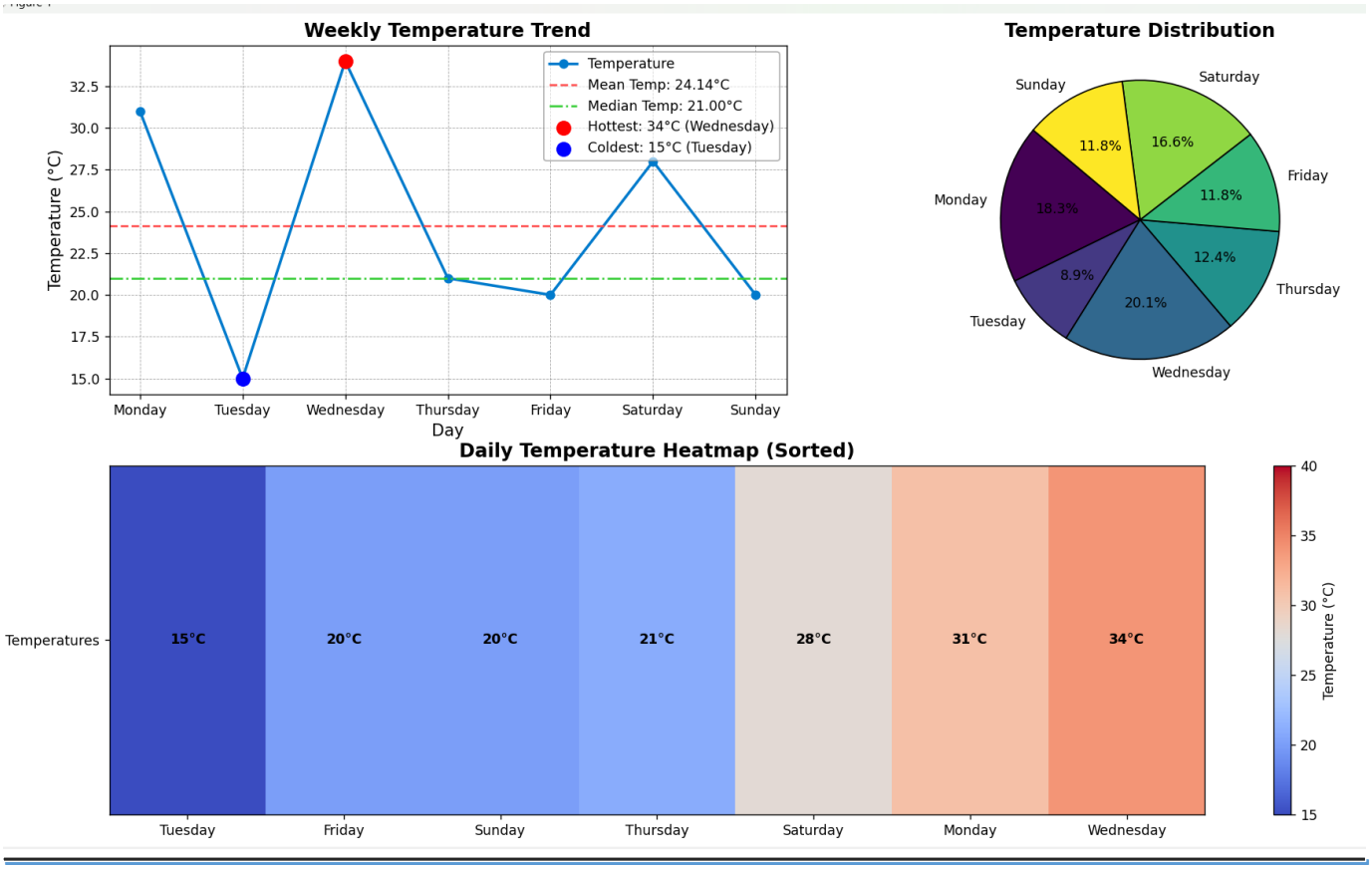
The visualizations include:

- Line Chart: Shows the weekly temperature trend with annotations for the mean, median, hottest, and coldest days.
- Pie Chart: Represents the proportion of weekly temperatures by day.
- Heatmap: Displays sorted daily temperatures using a gradient to highlight variations.

The code employs advanced styling (e.g., colors, markers, grids) and annotations to make insights more comprehensible. It also ensures aesthetic and functional layout adjustments for clarity.

RESULTS

The project simulates and analyzes weekly temperature data, providing insights through statistical summaries and visualizations. Key outcomes include identifying the hottest and coldest days, calculating temperature trends (mean, median, and standard deviation), and displaying data in an interactive table, pie chart, line graph, and heatmap. These results enhance understanding of temperature patterns and their daily distribution.



CONCLUSION AND LEARNINGS

This project provided valuable insights into analyzing and visualizing temperature data. It enhanced understanding of statistical metrics (mean, median, etc.) and showcased effective use of visualizations for trend analysis. Skills in data handling with NumPy and Matplotlib were strengthened, especially in combining statistical insights with aesthetic plots.

Future Improvements:

1. Data Source Integration: Replace simulated data with real-time data using APIs (e.g., OpenWeatherMap).
2. Advanced Visualizations: Add 3D plots or interactive charts (e.g., using Plotly or Dash) for better engagement.
3. Forecasting: Implement predictive models (e.g., linear regression) to estimate future temperatures.
4. Customizations: Allow user-defined inputs for dynamic analysis and location-specific weather insights.