A Project Report on

ANALYZE WEATHER DATA

A report submitted in partial fulfillment of the requriments of the

Industry internship program

By

Mr. Pranav Patil

Mr. Mantesh Swami

Mr. Shreyas Kolekar

Mr. Shreyas Mahangade

Under Guidance of

Ms. Nikita Pawar

NEUAI LABS

Software training institute in Pune, Maharashtra

OBJECTIVE

The objective of this project is to conduct a comprehensive analysis of historical weather data to identify and understand trends in key meteorological parameters, including temperature, rainfall, and humidity. The project aims to explore how these factors fluctuate over various time periods and geographic regions, providing a detailed perspective on long-term climate patterns and short-term weather variations. By utilizing advanced statistical techniques, data visualization tools, and machine learning models, the analysis seeks to uncover hidden correlations, seasonal patterns, and potential anomalies that may impact diverse sectors such as agriculture, urban planning, disaster management, and climate change mitigation. The insights derived from this study can assist stakeholders in making informed decisions, optimizing resource allocation, and developing strategies to adapt to changing weather conditions. Additionally, the project aims to contribute to the broader scientific understanding of climatic trends by offering data-driven evidence to support policy-making and future research initiatives.

SOURCES

Data Source:

- This project successfully conducted an in-depth analysis of historical weather data to better understand temperature trends and identify potential extreme weather events in the cities of Pune and Mumbai. The comprehensive study revealed a noticeable warming trend in both cities, indicating potential climate change effects that could have significant implications for urban living and environmental sustainability.
- The weather data used for this analysis was sourced from reliable platforms such as Kaggle (https://www.kaggle.com) and public/local weather websites. These sources provide extensive datasets covering historical metrics for temperature, rainfall, and humidity, ensuring a robust foundation for analysis.
- By leveraging a powerful suite of Python tools—including Pandas for efficient data manipulation, Matplotlib and Seaborn for in-depth visualizations, and Plotly for interactive data exploration—the study yielded valuable insights, including:
  - Long-term temperature trends – A consistent rise in temperatures overthe years, suggesting the influence of climate change and emphasizing the need for adaptive measures.
  - Humidity variations – Fluctuations that can provide critical information for health monitoring, agricultural planning, and environmental research.
- These insights serve as a valuable resource for researchers, climate scientists, and urban planners, offering data-driven evidence to support further research in climate science. They can aid policymakers and stakeholders in making informed decisions to promote environmental sustainability, enhance climate resilience, and optimize resource management.
- Future work could expand on this study by integrating additional meteorological parameters, employing predictive modeling techniques, and broadening the geographic scope to gain deeper insights into climate dynamics at a regional or national level.

Tools Used:

- To analyze and visualize historical weather data effectively, several Python-based tools were utilized, each serving a specific role in the data processing pipeline:
  - Python: The core programming language used for data processing, statistical analysis, and visualization due to its flexibility and extensive libraries.
  - Pandas: Used for data manipulation and cleaning, enabling efficient handling of large datasets through operations like filtering, aggregation, and transformation.
  - Matplotlib: Employed to create static visualizations such as line and bar charts, providing clear representations of temperature trends and humidity variations.
  - Plotly: Utilized for interactive visualizations, allowing users to explore data dynamically by zooming, panning, and hovering over data points to uncover deeper insights.
  - Seaborn: Built on Matplotlib, it was used to enhance visualization aesthetics and generate advanced statistical plots, making data interpretation easier and more insightful.
- These tools collectively enabled efficient data processing, insightful analysis, and visually compelling representations, helping to identify key weather trends and patterns.

STEPS TAKEN

Step 1: Data Collection

- Downloaded weather datasets from Kaggle, ensuring they contained metrics such as temperature, rainfall, and humidity over multiple years.
- verified the credibility of the sources to ensure data accuracy.

Step 2: Data Cleaning

- Removed missing values or filled them with appropriate statistical methods such as mean or median imputation.
- Standardized date formats to ensure proper time-based analysis.
- Handled outliers using statistical techniques to maintain data reliability.

Step 3: Data Exploration

- Explored the dataset to understand the structure and key attributes.
- Performed descriptive statistics to calculate initial insights, such as average temperature, total rainfall, and maximum humidity levels.

Step 4: Data Visualization

- Created line plots to visualize year-over-year changes in temperature trends.
- Developed bar charts to represent monthly or seasonal rainfall distribution.
- Used box plots to showcase the variability and distribution of humidity levels.

Step 5: Statistical Analysis

- Calculated critical metrics like:
  - Average Temperature: To understand overall climate trends.
  - Maximum and Minimum Values: For identifying extreme weather conditions.
- Conducted correlation analysis to identify relationships between temperature, rainfall, and humidity.

Step 6: Insights and Trends

- Interpreted the visualizations and metrics to identify:
  - Periods of significant temperature increase or decrease.
  - Regions or years showing consistent patterns in humidity levels.

CODE

- **Importing necessary Libraries and dataset:**

```
#Import all necessary libraries

import numpy as np
import pandas as pd
import datetime
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
import plotly.express as px
```

```
# Read the csv file
pune_data=pd.read_csv('pune.csv')
pune_data.head()
```

- **Check the total number null Data:**

```
pune_data.isnull().sum()
```

- **Concise summary of Data:**

```
pune_data.info()
```

- **Statistical summery of Data:**

```
pune_data.describe()
```

➢ Convert Datatype of date_time Column:

```
# Convert 'date_time' column to datetime format

pune_data['date_time'] = pd.to_datetime(pune_data['date_time'])

#seperate year,date and time columns

pune_data['year'] = pune_data['date_time'].dt.year
pune_data['Date']=pune_data['date_time'].dt.date
pune_data['Time']=pune_data['date_time'].dt.time


# setting date as index

pune_data.set_index('date_time',inplace=True)
```

➢ Temperature Trend Over Time:

Temperature Trend Over Time

```
# Line Plot for Trend Over Time:
plt.figure(figsize=(12,6))
sns.lineplot(x='year',y='tempC',data=pune_data)

plt.xticks(rotation=45,horizontalalignment='right')
plt.title("Temperature Trend Over Time In Pune")
plt.grid(True)
plt.show()
```

[11]

➢ Humidity Trend Over Time:

```
# Bar Plot For humidity trend over time

plt.figure(figsize=(10,6))
sns.barplot(x='year',y='humidity',data=pune_data)

plt.xticks(rotation=45,horizontalalignment='right')
plt.title("Humidity Trend Over Time In Pune")
plt.show()
```

➢ Temperature vs Humidity:

```
## Temperature VS Humidity

sns.scatterplot(x='tempC',y='humidity',data=pune_data)
plt.xlabel("Temperature (C)")
plt.ylabel("Humidity (%)")
plt.title("Temperature VS Humidity")
plt.show()
```

- **Load Another City Data (e.g, Mumbai):**

load another city's data (e.g., Mumbai):

```
mumbai_data=pd.read_csv('Mumbai.csv')
```

```
## Add City Labels To Each Dataset
pune_data['City'] = 'pune'
mumbai_data['City'] = 'Mumbai'
```

- **Convert into Datetime Format and Seprate into Year, date, and time Column:**

```
# Convert 'date_time' column to datetime format

mumbai_data['date_time'] = pd.to_datetime(mumbai_data['date_time'])

#seperate hour,quarter,month and year column

mumbai_data['year'] = mumbai_data['date_time'].dt.year
mumbai_data['Date']=mumbai_data['date_time'].dt.date
mumbai_data['Time']=mumbai_data['date_time'].dt.time
# setting date as index

mumbai_data.set_index('date_time',inplace=True)
```

- **Comibine The Dataset:**

```
# Combined Datasets
combined_data=pd.concat([pune_data,mumbai_data])
combined_data.head()
```

➢ Compare Pune And Mumbai Temperature Trends:

```python
## Compare Temperature Trends
plt.figure(figsize=(12,6))
sns.lineplot(x='year',y='tempC',hue='City',data=combined_data)
plt.title('Temperature Trends: Pune vs Mumbai')
plt.xlabel('Year')
plt.ylabel('Temperature')
plt.grid(True)
plt.show()
```

➢ Extream Weather Event Detection:

Extream Weather Event Detection

Identify Heatwaves

```python
# Detect Heatwaves ( temperature > 40 C)
heatwaves=pune_data[pune_data['tempC'] > 40]
print('Heatwaves Events:')
heatwaves.head()
```

➢ Interactive Temperature Trend In Pune:

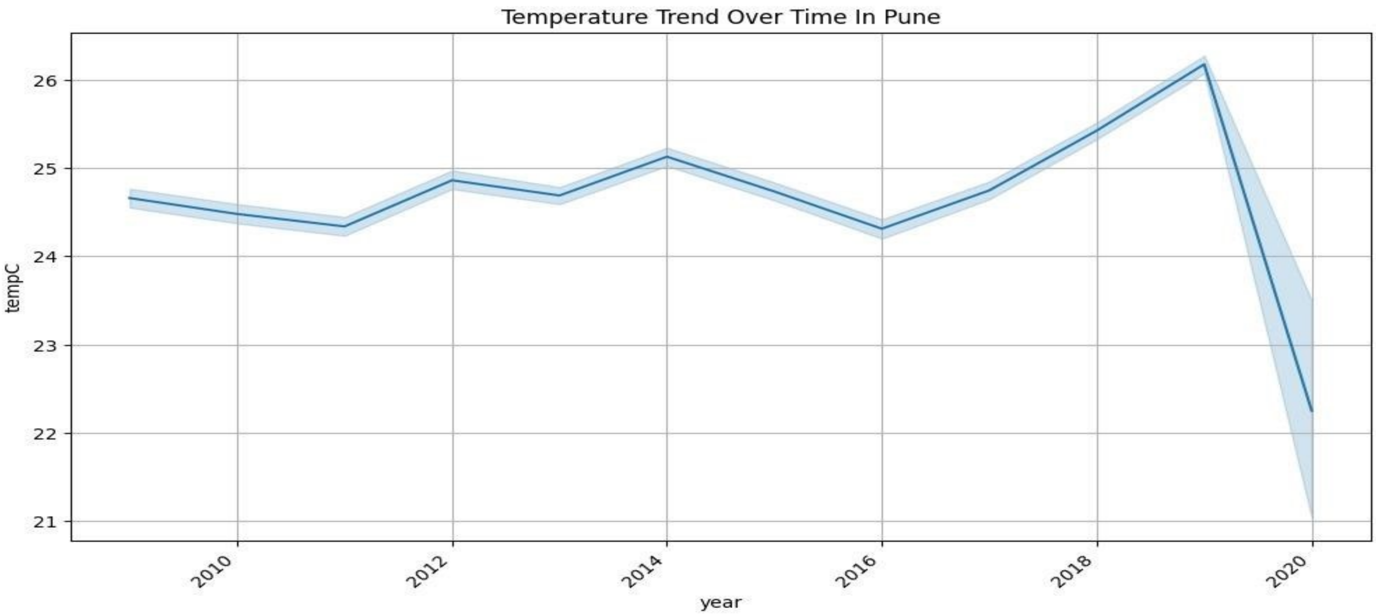
```python
fig=px.line(pune_data,x='Date',y='tempC',title='Interactive Temperature Trend In Pune')
fig.show()
```

➢ Interactive Temperature Trends: Pune Vs Mumbai:

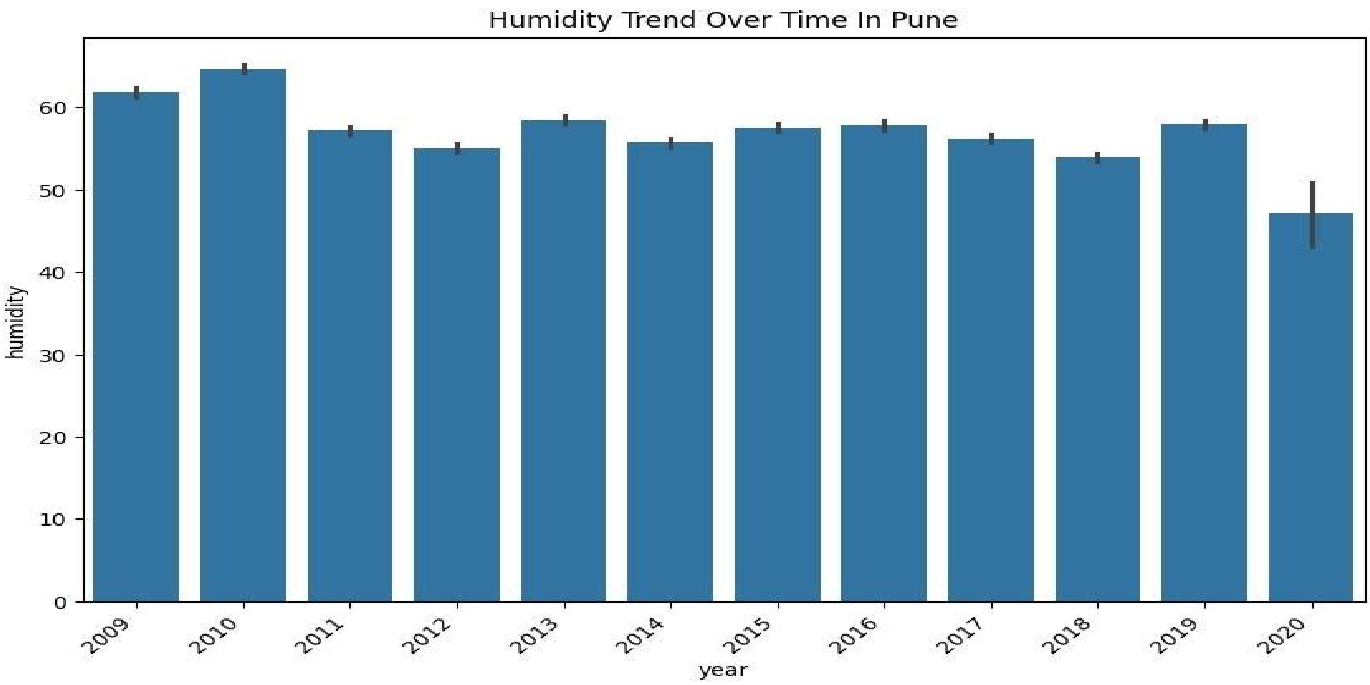
```python
## Compare Cities Interactively
fig= px.line(combined_data,x='Date',y='tempC',color='City',title='Interactive Temperature Trends: Pune VS Mumbai')
fig.show()
```

OUTPUT:

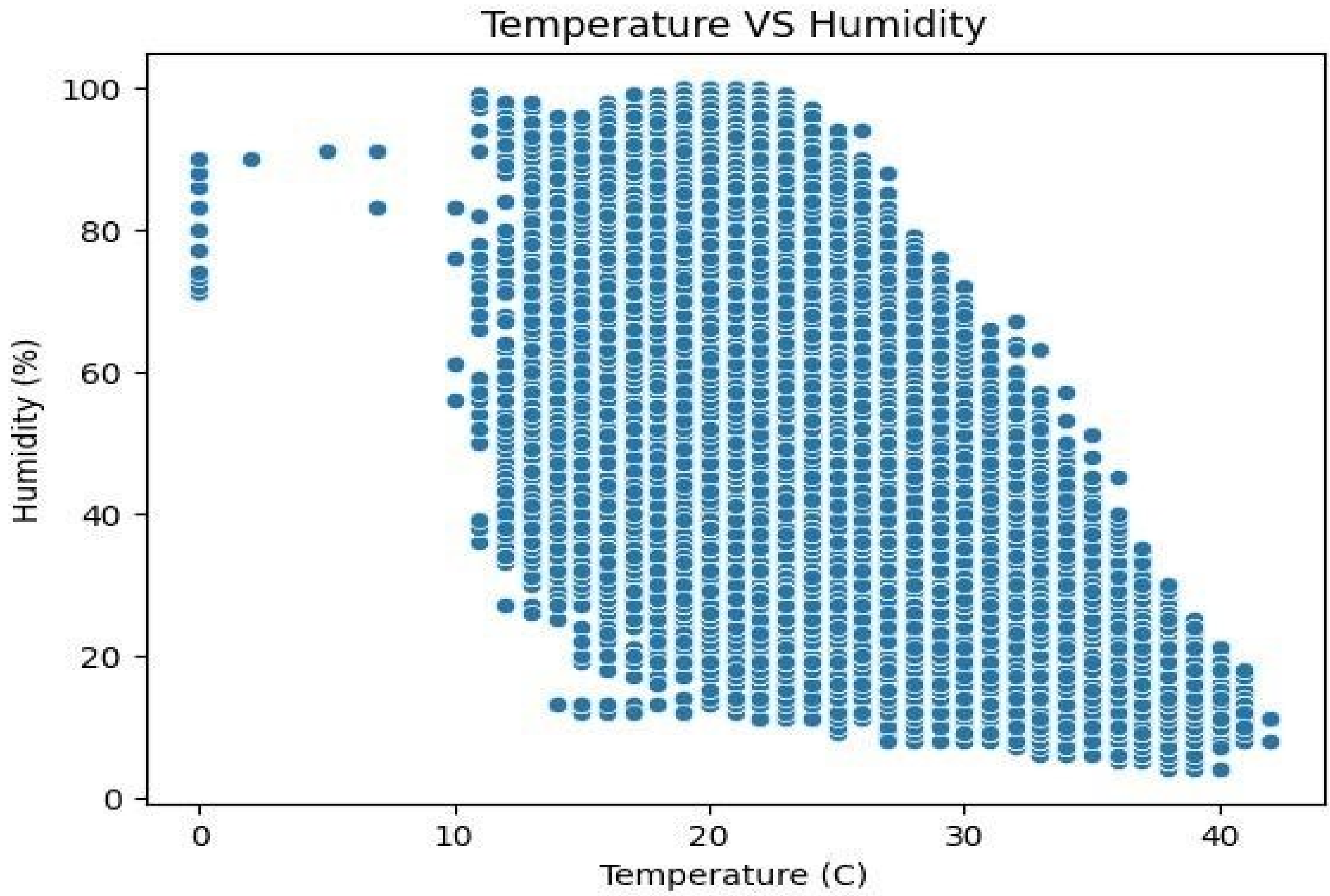
1. **Temperature Trend Over Time In Pune:**



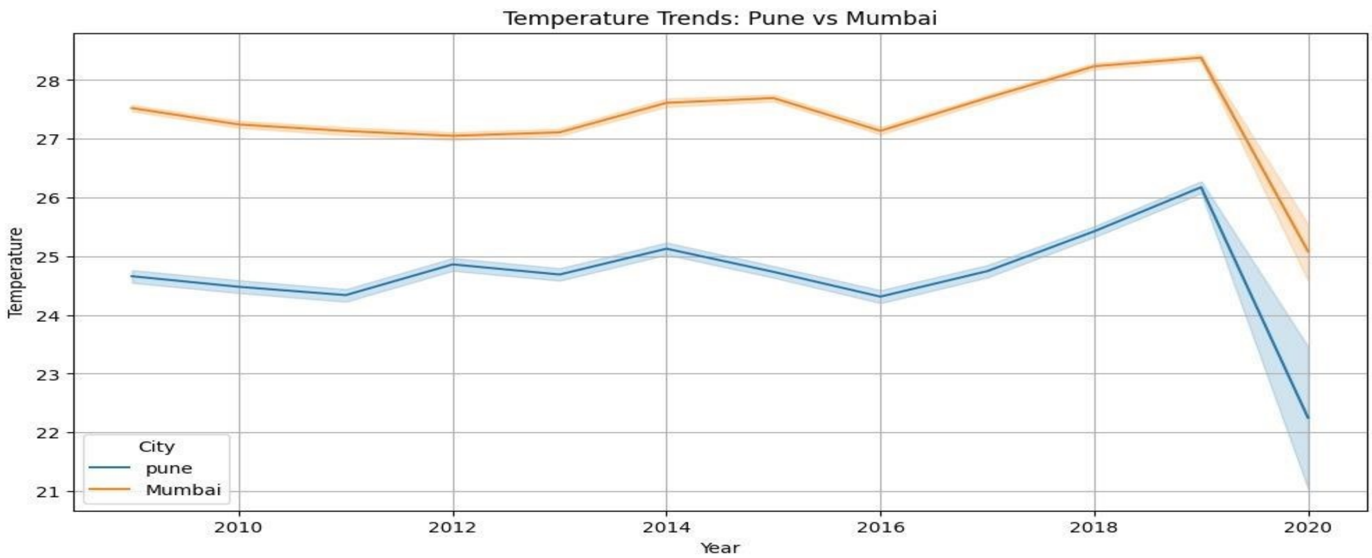
2. **Humidity Trend Over Time In Pune:**



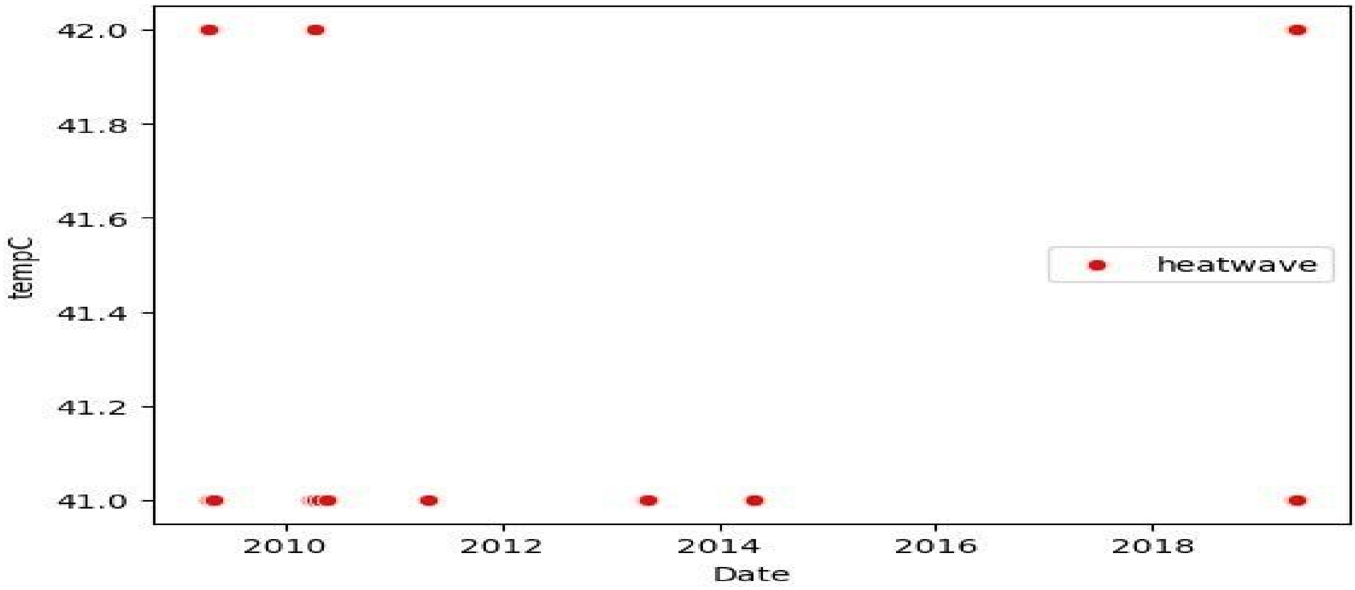
3. Temperature Vs Humidity:



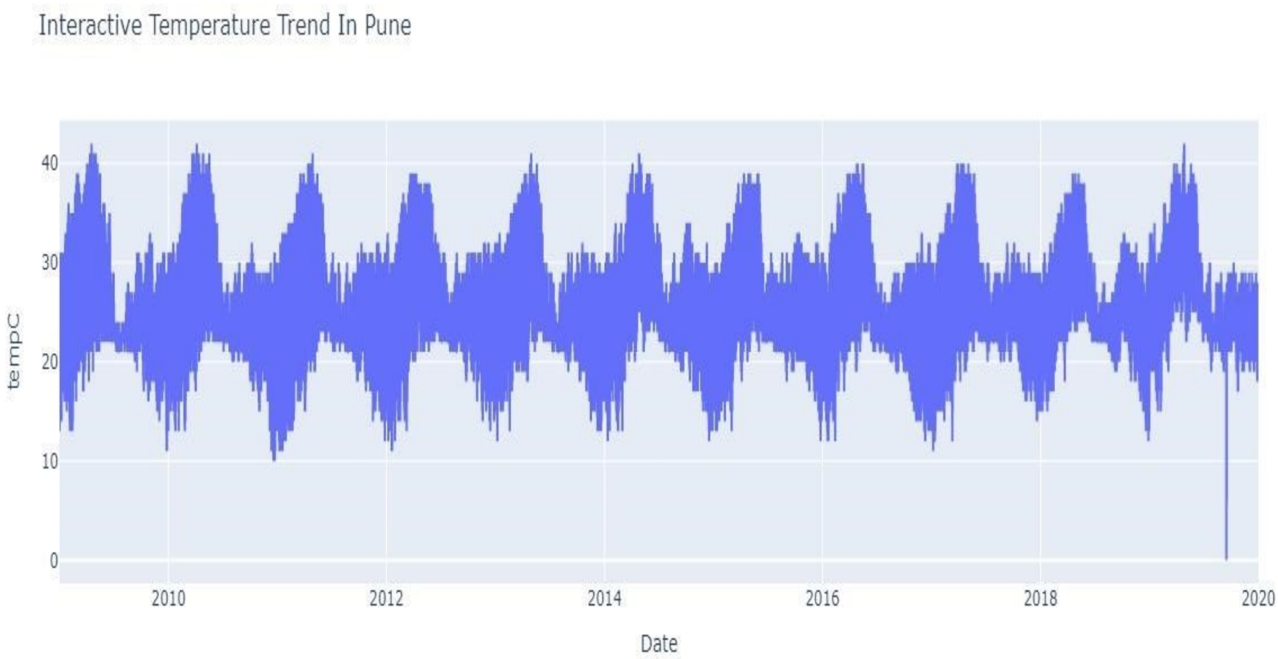
4. Temperature Trend: Pune Vs Mumbai:



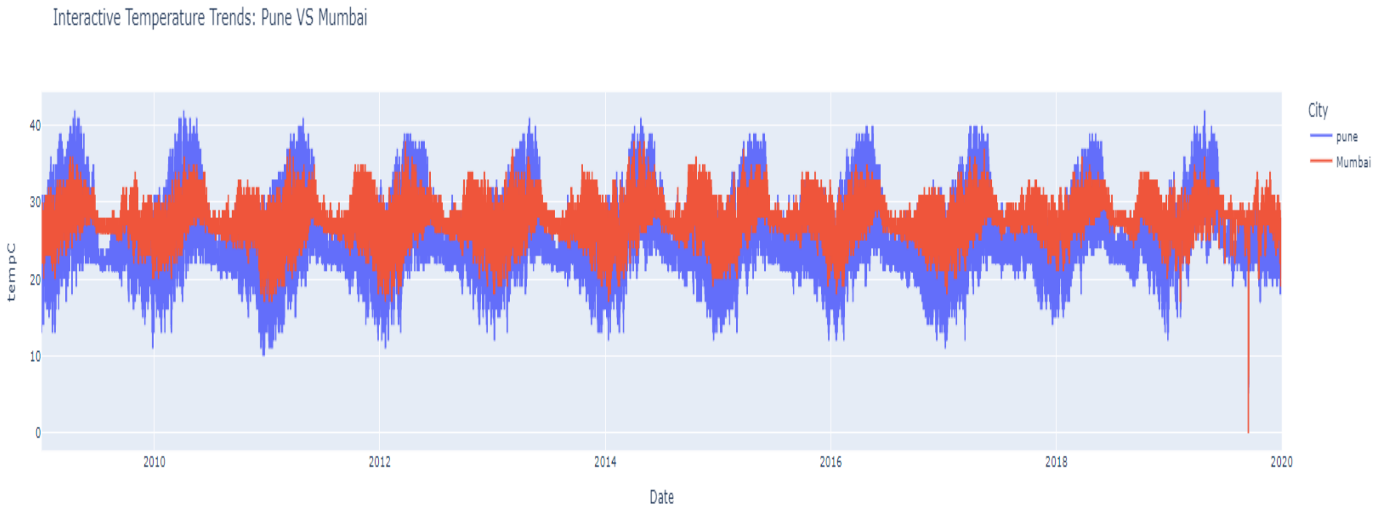
5. **Heatwave Detection:**



6. **Interactive Temperature Trend In Pune :**



7. Interactive Temperature Trend: Pune Vs Mumbai:



Conclusion:

This project successfully conducted an in-depth analysis of historical weather data to better understand temperature trends and identify potential extreme weather events in the cities of Pune and Mumbai. The comprehensive study revealed a noticeable warming trend in both cities, indicating potential climate change effects that could have significant implications for urban living and environmental sustainability.

By leveraging a robust set of Python tools, including Pandas for data manipulation, Matplotlib and Seaborn for detailed visualizations, and Plotly for interactive data exploration, the analysis provided key insights into:

• Long-term temperature trends – A consistent rise in temperatures over the years, suggesting the influence of climate change and the need for adaptive measures.
• Humidity variations – Fluctuations that could have implications for public health, agriculture, and environmental planning.

The insights derived from this analysis can serve as a valuable resource for researchers, climate scientists, and urban planners. They provide a data-driven basis for further studies in climate science, helping policymakers and stakeholders formulate informed strategies to mitigate risks, adapt to evolving climate patterns, and work towards sustainable urban development.

Future work could expand on this study by incorporating additional meteorological parameters, exploring predictive modeling techniques, and analyzing broader geographic regions to gain a more comprehensive understanding of climate dynamics.