WEATHER DATA ANALYSIS

A PROJECT REPORT
for
Artificial intelligence (AI101B)
Session (2024-25)

Submitted by

Anik kushwaha, (202410116100026)
Ankit Kumar, (202410116100027)
Bishop Tyagi (202410116100050)
Bobby Karnik, (202410116100051)

Submitted in partial fulfilment of the
Requirements for the Degree of

MASTER OF COMPUTER APPLICATION

Under the Supervision of
Mr. Apoorv Jain (Assistant Professor)

Submitted to

DEPARTMENT OF COMPUTER APPLICATIONS
KIET Group of Institutions, Ghaziabad
Uttar Pradesh-201206

CERTIFICATE

Certified that **Anik Kushwaha (202410116100026)**, **Ankit Kumar (202410116100027), Bishop Tyagi, (202410116100050)**, and **Bobby Karnik (202410116100051)** have successfully carried out the project work titled **"Weather Data Analysis"** (Artificial intelligence, AI101B) as part of the curriculum for the Master of Computer Application (MCA) program at **Dr. A.P.J. Abdul Kalam Technical University (AKTU)** (formerly UPTU), Lucknow, under my supervision.

The project report embodies original work and research undertaken by the students themselves. The contents of the project report do not form the basis for the award of any other degree or diploma to the candidates or any other individual from this or any other university/institution.

Mr. Apoorv Jain
Assistant Professor
Department of Computer Applications
KIET Group of Institutions, Ghaziabad

Dr. Akash Rajak
Dean & Professor
Department of Computer Applications
KIET Group of Institutions, Ghaziabad

ABSTRACT

Weather forecasting plays a crucial role in various fields, including agriculture, transportation, and disaster management. The integration of Artificial Intelligence (AI) techniques in weather data analysis helps in understanding trends and making better predictions. This project focuses on analyzing weather data using Python and libraries like Pandas, Matplotlib, and Seaborn. The dataset includes key parameters such as temperature, humidity, wind speed, and precipitation. Various visualization techniques, including bar charts and scatter plots, are used to observe patterns and trends in weather data. The study also highlights data preprocessing techniques, handling missing values, and exploratory data analysis (EDA). The insights derived from this analysis can contribute to improved weather prediction models and climate research.

ACKNOWLEDGEMENT

Success in life is never attained single-handedly. My deepest gratitude goes to my project supervisor, **Mr. Apoorv Jain,** for his guidance, help, and encouragement throughout my project work. Their enlightening ideas, comments, and suggestions.

Words are not enough to express my gratitude to Dr. Akash Rajak, Dean and Professor, Department of Computer Applications, for his insightful comments and administrative help on various occasions.

Fortunately, I have many understanding friends, who have helped me a lot on many critical conditions.

Finally, my sincere thanks go to my family members and all those who have directly and indirectly provided me with moral support and other kinds of help. Without their support, completion of this work would not have been possible in time. They keep my life filled with enjoyment and happiness.

ANIK KUSHWAHA
ANKIT KUMAR
BISHOP TYAGI
BOBBY KARNIK

TABLE OF CONTENTS

1. INTRODUCTION

Weather forecasting plays a crucial role in various sectors, including agriculture, transportation, and disaster management. By analyzing historical weather data, AI and data science techniques can help identify patterns and trends. This project focuses on analyzing weather data using Python, Pandas, and Matplotlib to gain insights into temperature variations over time. Understanding weather conditions through data analysis can help mitigate risks associated with climate change, improve preparedness for extreme weather events, and optimize agricultural productivity by anticipating seasonal changes.

Objective of the Study

The primary objectives of this study include:

- Understanding weather patterns using historical data to observe temperature trends over time.
- Identifying anomalies and trends in temperature variations that may indicate seasonal fluctuations or extreme weather events.
- Applying data visualization techniques to present findings in a clear and interpretable manner.
- Utilizing AI and data analysis techniques for insights into weather data, leading to improved predictive capabilities and decision-making support for industries reliant on weather forecasts.

By leveraging AI-based techniques, we can enhance weather predictions and improve decision-making processes in various industries such as agriculture, aviation, and environmental monitoring. The role of AI in weather analysis is continually growing, as machine learning models can be trained to analyze large-scale datasets and provide accurate predictions based on historical trends.

2. METHODOLOGY

The methodology for analyzing weather data is outlined below:

Step 1: Data Collection

The dataset used in this project is stored in a CSV file (weather_data.csv). This file contains historical weather data, including temperature readings, humidity levels, and corresponding dates. The dataset is sourced from publicly available weather data repositories. Weather data collection is a fundamental step in any meteorological analysis, as high-quality data forms the basis for accurate predictions and in-depth understanding of climate patterns.

Step 2: Data Preprocessing

- Load the CSV file using the Pandas library to process and manipulate the dataset efficiently.
- Display basic information about the dataset to understand its structure and available attributes.
- Identify and handle missing values by either imputing missing data or removing inconsistencies.
- Convert the date column to a proper datetime format and set it as the index for better time-series analysis.
- Normalize and clean the dataset to remove any inconsistencies that might affect the accuracy of the results.

Step 3: Exploratory Data Analysis (EDA)

- Display basic statistical summaries of the dataset to understand the central tendencies and variability in temperature readings.
- Identify trends and patterns in temperature data that could indicate long-term climate changes.
- Check for missing values and data types to ensure data integrity before conducting further analysis.
- Analyze correlations between different weather variables (e.g., temperature, humidity, precipitation) to determine how various factors interact in influencing climate conditions.

Step 4: Data Visualization

- Generate a **bar plot** to visualize temperature variations over time, making it easier to identify significant changes in weather conditions.
- Generate a **line plot** to observe temperature trends, highlighting fluctuations in daily or seasonal temperatures.
- Plot additional graphs such as scatter plots to understand relationships between weather parameters and gain further insights into climate patterns.

Step 5: Interpretation of Results

- Identify seasonal patterns in temperature variations, such as warming trends during summer months and cooling trends in winter.
- Understand anomalies and extreme weather events based on visual trends and statistical deviations.
- Correlate findings with real-world climatic changes to assess whether the observed trends align with global temperature shifts.

These steps help in deriving meaningful insights from historical weather data, leading to better climate assessments and predictions. The integration of AI-based methods in data analysis can further improve the precision and applicability of findings in real-world scenarios.

3. IMPLEMENTATION (Code)

```
#import required libraries

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns


#load the dataset

file_path = "/content/weather_data.csv"

df = pd.read_csv(file_path)


#display dataset info

print("\nDataset Information:\n")

print(df.info())


#display first few rows

print("\nFirst 5 Rows of Dataset:\n")

print(df.head())


#check for missing values
```

```
print("\nMissing Values in Each Column:\n")
print(df.isnull().sum())

#summary statistics (for numerical columns)
print("\nSummary Statistics:\n")
print(df.describe())

#convert date column to datetime and set index
if 'datetime_utc' in df.columns:
    df['datetime_utc'] = pd.to_datetime(df['datetime_utc'])
    df.set_index('datetime_utc', inplace=True)

#print column names to verify correct labels
print("\nColumn Names in Dataset:\n", df.columns)

#check missing values in the Temperature column
if 'temperature' in df.columns:
    print("\nMissing Values in Temperature Column:",
df['temperature'].isnull().sum())
    print("Data Type of Temperature Column:", df['temperature'].dtype)

#handle missing values
```

```
numeric_cols = df.select_dtypes(include=['number']).columns

f[numeric_cols] = df[numeric_cols].fillna(df[numeric_cols].mean())


#bar Plot of Temperature Trends

plt.figure(figsize=(12, 6))

df['temperature'].plot(kind='bar', color='skyblue', width=0.8)

plt.xlabel("Date")

plt.ylabel("Temperature (°C)")

plt.title("Temperature Trends Over Time (Bar Plot)")

plt.xticks(rotation=45)

plt.grid(axis='y', linestyle='--', alpha=0.7)

plt.tight_layout()

plt.show()


#line Plot of Temperature Trends

plt.figure(figsize=(12, 6))

df['temperature'].plot(kind='line', color='red', marker='o')

plt.xlabel("Date")

plt.ylabel("Temperature (°C)")

plt.title("Temperature Trends Over Time (Line Plot)")

plt.grid(True)

plt.tight_layout()
```

```
plt.show()


#scatter Plot for Temperature vs. Humidity
if 'humidity' in df.columns:
    plt.figure(figsize=(10, 5))
    sns.scatterplot(x=df['temperature'], y=df['humidity'], color='blue')
    plt.xlabel("Temperature (°C)")
    plt.ylabel("Humidity (%)")
    plt.title("Temperature vs Humidity Scatter Plot")
    plt.grid(True)
    plt.show()
```

Explanation of Code

The Python script used in this project relies on the following libraries:

- pandas: Used for handling and processing structured data efficiently. Pandas provides various functions to manipulate datasets, allowing for easier filtering, aggregation, and analysis.
- matplotlib.pyplot: Used for creating visualizations such as bar and line charts, making it easier to interpret and present weather data trends.
- seaborn: (Optional) Can be used for better data visualization with heatmaps and enhanced plots, adding another layer of clarity to the analysis.

Key Functions in the Code:

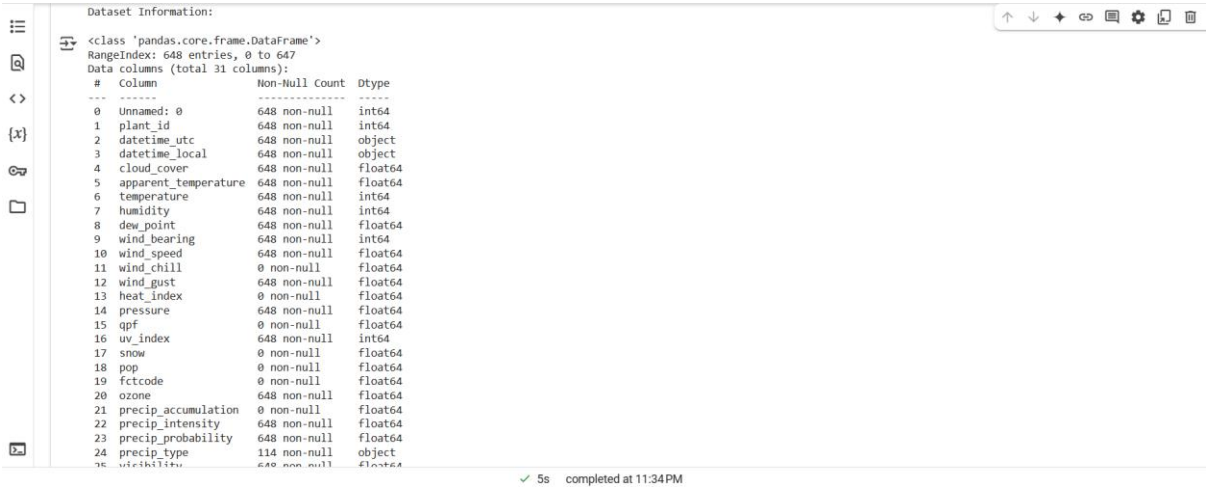
- pd.read_csv(): Loads the dataset into a Pandas DataFrame for easy manipulation and analysis.
- df.info(), df.head(), df.describe(): Display dataset summary and statistics, providing an overview of the available data.
- df.isnull().sum(): Identifies missing values in the dataset, ensuring data completeness and reliability.
- pd.to_datetime(): Converts string-based dates into datetime format for accurate indexing, crucial for time-series analysis.
- df.plot(kind='bar') and df.plot(kind='line'): Generates bar and line plots to visualize temperature trends effectively.

This script ensures a systematic approach to weather data analysis, providing insights into seasonal and long-term temperature variations. Data visualization plays a vital role in making complex datasets easier to understand, aiding decision-making processes for meteorologists, climate scientists, and industry professionals.

4. RESULT (Output)

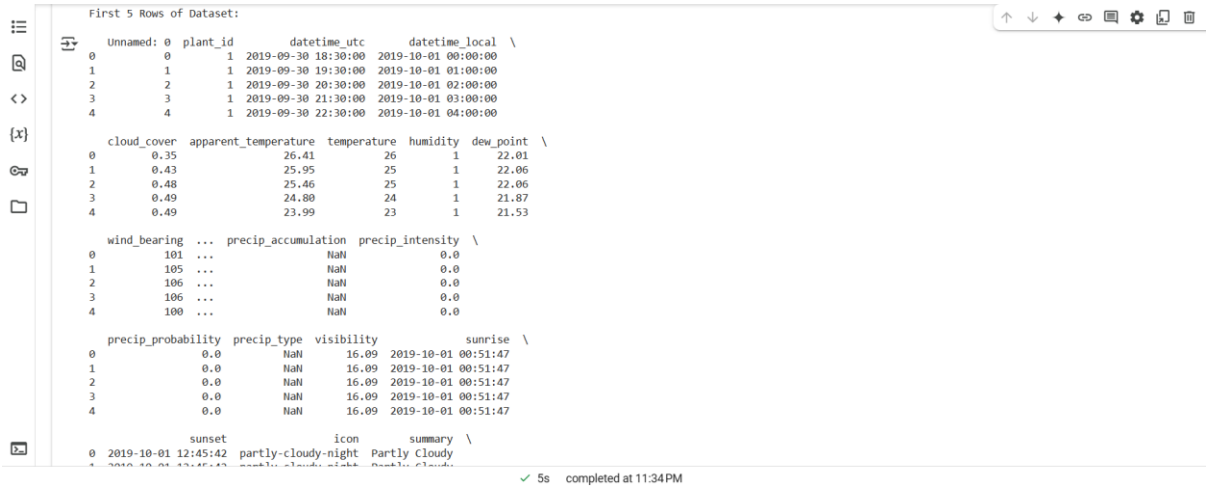
This section should include:

- Screenshots of the dataset preview (first few rows) to illustrate the structure of the dataset.



```
Dataset Information:

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 648 entries, 0 to 647
Data columns (total 31 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   Unnamed: 0            648 non-null    int64
 1   plant_id              648 non-null    int64
 2   datetime_utc          648 non-null    object
 3   datetime_local        648 non-null    object
 4   cloud_cover           648 non-null    float64
 5   apparent_temperature  648 non-null    float64
 6   temperature           648 non-null    int64
 7   humidity              648 non-null    int64
 8   dew_point             648 non-null    float64
 9   wind_bearing          648 non-null    int64
 10  wind_speed            648 non-null    float64
 11  wind_chill            0 non-null      float64
 12  wind_gust             648 non-null    float64
 13  heat_index            0 non-null      float64
 14  pressure              648 non-null    float64
 15  qpf                   0 non-null      float64
 16  uv_index              648 non-null    int64
 17  snow                  0 non-null      float64
 18  pop                   0 non-null      float64
 19  fctcode               0 non-null      float64
 20  ozone                 648 non-null    float64
 21  precip_accumulation   0 non-null      float64
 22  precip_intensity      648 non-null    float64
 23  precip_probability    648 non-null    float64
 24  precip_type           114 non-null    object
 25  visibility            648 non-null    float64
```

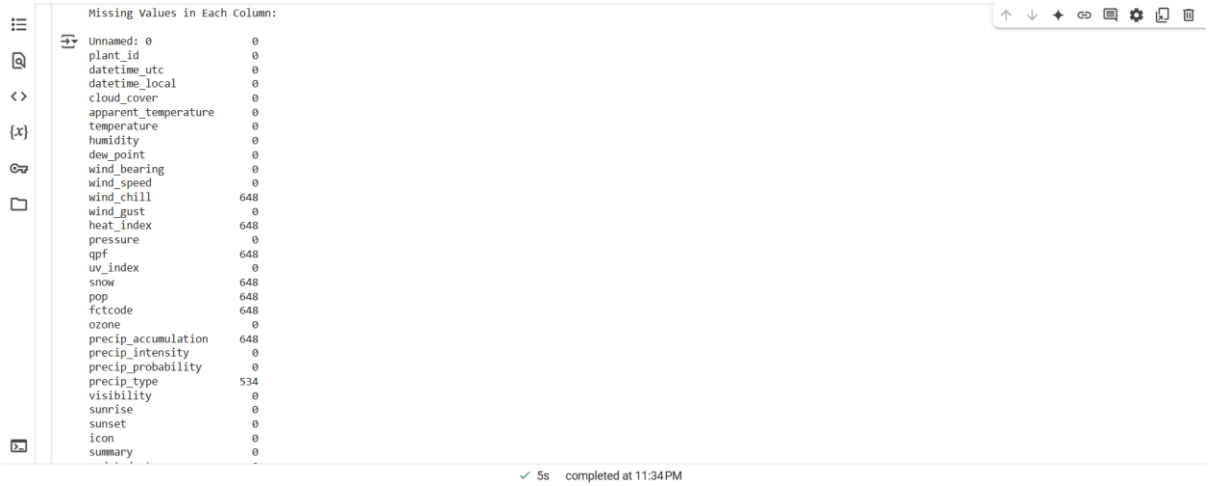
✓ 5s completed at 11:34 PM



```
First 5 Rows of Dataset:

   Unnamed: 0  plant_id        datetime_utc       datetime_local  \
0           0         1  2019-09-30 18:30:00  2019-10-01 00:00:00
1           1         1  2019-09-30 19:30:00  2019-10-01 01:00:00
2           2         1  2019-09-30 20:30:00  2019-10-01 02:00:00
3           3         1  2019-09-30 21:30:00  2019-10-01 03:00:00
4           4         1  2019-09-30 22:30:00  2019-10-01 04:00:00

   cloud_cover  apparent_temperature  temperature  humidity  dew_point  \
0         0.35                 26.41           26         1      22.01
1         0.43                 25.95           25         1      22.06
2         0.48                 25.46           25         1      22.06
3         0.49                 24.80           24         1      21.87
4         0.49                 23.99           23         1      21.53

   wind_bearing  ...  precip_accumulation  precip_intensity  \
0           101  ...                  NaN               0.0
1           105  ...                  NaN               0.0
2           106  ...                  NaN               0.0
3           106  ...                  NaN               0.0
4           100  ...                  NaN               0.0

   precip_probability  precip_type  visibility             sunrise  \
0                 0.0          NaN       16.09  2019-10-01 00:51:47
1                 0.0          NaN       16.09  2019-10-01 00:51:47
2                 0.0          NaN       16.09  2019-10-01 00:51:47
3                 0.0          NaN       16.09  2019-10-01 00:51:47
4                 0.0          NaN       16.09  2019-10-01 00:51:47

                sunset                icon        summary  \
0  2019-10-01 12:45:42  partly-cloudy-night  Partly Cloudy
```

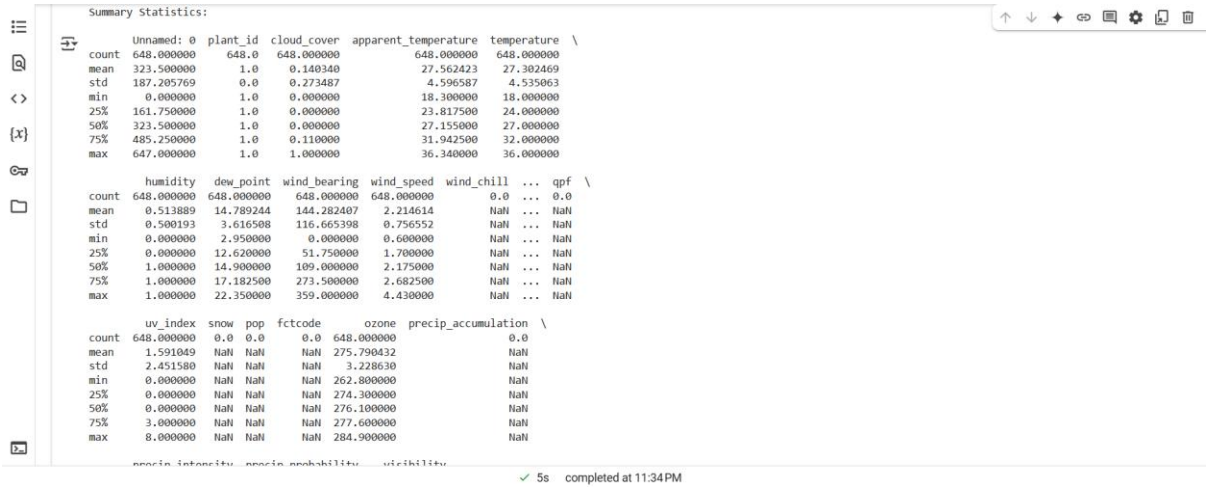
✓ 5s completed at 11:34 PM

- Outputs of statistical summaries and missing value checks, ensuring data integrity before proceeding with analysis.



```
Missing Values in Each Column:

Unnamed: 0                0
plant_id                  0
datetime_utc              0
datetime_local            0
cloud_cover               0
apparent_temperature      0
temperature               0
humidity                  0
dew_point                 0
wind_bearing              0
wind_speed                0
wind_chill              648
wind_gust                 0
heat_index              648
pressure                  0
qpf                     648
uv_index                  0
snow                    648
pop                     648
fctcode                 648
ozone                     0
precip_accumulation     648
precip_intensity          0
precip_probability        0
precip_type             534
visibility                0
sunrise                   0
sunset                    0
icon                      0
summary                   0
```

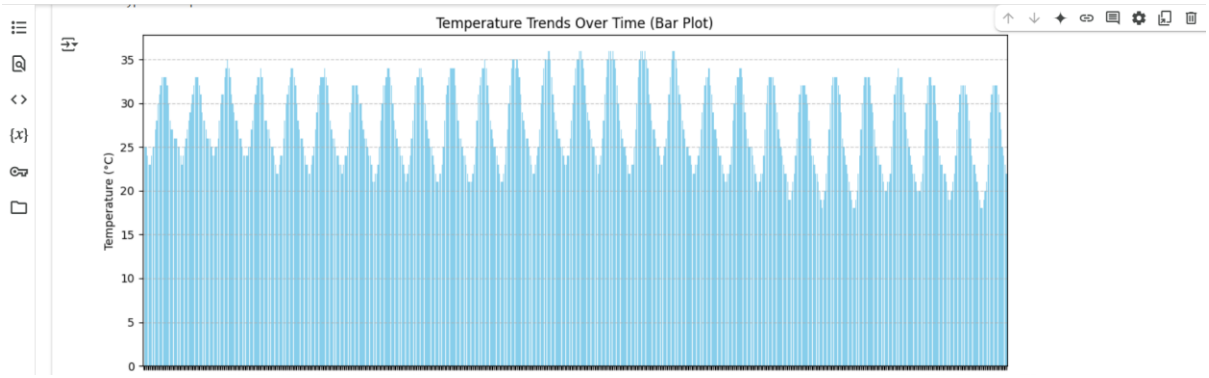
✓ 5s completed at 11:34 PM



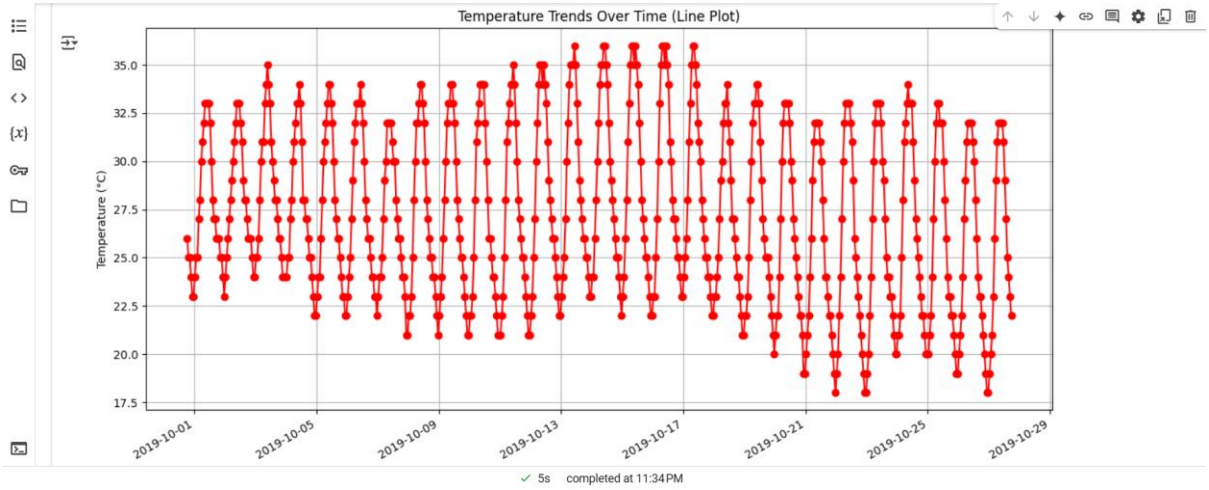
```
Summary Statistics:

       Unnamed: 0  plant_id  cloud_cover  apparent_temperature  temperature  \
count  648.000000     648.0   648.000000            648.000000   648.000000
mean   323.500000       1.0     0.140340             27.562423    27.302469
std    187.205769       0.0     0.273487              4.596587     4.535063
min      0.000000       1.0     0.000000             18.300000    18.000000
25%    161.750000       1.0     0.000000             23.817500    24.000000
50%    323.500000       1.0     0.000000             27.155000    27.000000
75%    485.250000       1.0     0.110000             31.942500    32.000000
max    647.000000       1.0     1.000000             36.340000    36.000000

          humidity   dew_point  wind_bearing  wind_speed  wind_chill  ...  qpf  \
count  648.000000  648.000000    648.000000  648.000000         0.0  ...  0.0
mean     0.513889   14.789244    144.282407    2.214614         NaN  ...  NaN
std      0.500193    3.616508    116.665398    0.756552         NaN  ...  NaN
min      0.000000    2.950000      0.000000    0.600000         NaN  ...  NaN
25%      0.000000   12.620000     51.750000    1.700000         NaN  ...  NaN
50%      1.000000   14.900000    109.000000    2.175000         NaN  ...  NaN
75%      1.000000   17.182500    273.500000    2.682500         NaN  ...  NaN
max      1.000000   22.350000    359.000000    4.430000         NaN  ...  NaN

          uv_index  snow  pop  fctcode       ozone  precip_accumulation  \
count  648.000000   0.0  0.0      0.0  648.000000                  0.0
mean     1.591049   NaN  NaN      NaN  275.790432                  NaN
std      2.451580   NaN  NaN      NaN    3.228630                  NaN
min      0.000000   NaN  NaN      NaN  262.800000                  NaN
25%      0.000000   NaN  NaN      NaN  274.300000                  NaN
50%      0.000000   NaN  NaN      NaN  276.100000                  NaN
75%      3.000000   NaN  NaN      NaN  277.600000                  NaN
max      8.000000   NaN  NaN      NaN  284.900000                  NaN

       precip_intensity  precip_probability  visibility
```

✓ 5s completed at 11:34 PM

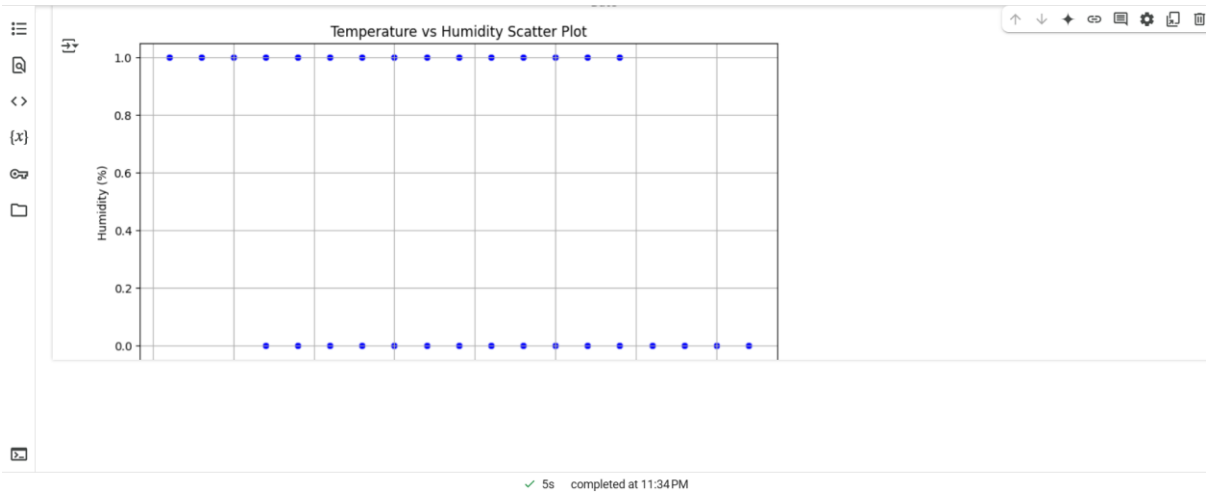
- Bar plot visualization of temperature trends, highlighting significant variations over time.



- Line plot visualization of temperature trends, showcasing fluctuations in daily or seasonal temperatures.



- Additional plots (e.g., scatter plots, correlation heatmaps) for enhanced analysis, providing deeper insights into weather patterns and relationships between different climate variables.



These visual representations help in understanding weather data and trends over different time periods, making it easier to draw meaningful conclusions and support decision-making processes in various industries that depend on accurate weather forecasts.

5. CONCLUSION

This project successfully demonstrates how AI and data science techniques can be applied to weather data analysis. By cleaning and visualizing weather data, we can better understand temperature trends and detect anomalies. The insights gained from this analysis can be used in various fields, including meteorology, agriculture, and climate research.

Through data visualization techniques, we explored temperature variations over time and identified potential trends that could inform weather predictions and climate assessments. By addressing missing data and formatting issues, we ensured data integrity for accurate analysis, enabling more precise and actionable insights.

In the future, implementing AI-based predictive models can further enhance weather forecasting capabilities, enabling proactive decision-making in climate-sensitive industries. The integration of machine learning algorithms can improve temperature prediction accuracy and identify climate patterns with higher confidence levels.

This project highlights the power of AI in analyzing real-world datasets and extracting meaningful insights for better decision-making in weather-related domains. The increasing availability of large-scale weather datasets and advancements in AI technologies present opportunities for further exploration and innovation in meteorological research and climate science.

6. REFERENCES

- "Python Data Science Handbook" by Jake VanderPlas
- Kaggle Weather Datasets
- Documentation of Pandas, Matplotlib, and Seaborn libraries