**Daily Weather Data Analysis Documentation**

**Overview**

This documentation provides a detailed guide on how to analyze daily weather data using Python. The analysis includes data exploration, visualization, feature engineering, predictive modeling, and deriving insights from the data.

**Table of Contents**

1. [Prerequisites](#prerequisites)
2. [Installation](#installation)
3. [Step-by-Step Analysis](#step-by-step-analysis)
   * [Load the Data](#step-1-load-the-data)
   * [Data Exploration](#step-2-data-exploration)
   * [Data Visualization](#step-3-data-visualization)
   * [Feature Engineering](#step-4-feature-engineering)
   * [Data Analysis](#step-5-data-analysis)
   * [Data Visualization (Part 2)](#step-6-data-visualization-part-2)
   * [Advanced Analysis](#step-7-advanced-analysis)
   * [Conclusions and Insights](#step-8-conclusions-and-insights)
4. [Additional Steps](#additional-steps)
   * [Communication](#step-9-communication)
   * [Future Work](#step-10-future-work)
5. [Frequently Asked Questions (FAQs)](#frequently-asked-questions-faqs)

**Prerequisites**

* Python 3.8 or higher
* Pip (Python package installer)

**Installation**

1. Install required Python packages:

bash

Copy code

pip install pandas matplotlib seaborn scikit-learn

1. Ensure you have the weather data CSV file (weather.csv) in the basic/ directory.

**Step-by-Step Analysis**

**Step 1: Load the Data**

The first step is to load the weather data from a CSV file.

python

Copy code

import pandas as pd

df = pd.read\_csv('basic/weather.csv')

**Step 2: Data Exploration**

Explore the data to understand its structure and basic statistics.

python

Copy code

# Display the first few rows of the dataframe

print(df.head())

# Display information about the dataframe

print(df.info())

# Display basic statistical details of the dataframe

print(df.describe())

**Step 3: Data Visualization**

Visualize relationships between variables using pair plots.

python

Copy code

import seaborn as sns

import matplotlib.pyplot as plt

sns.pairplot(df[['MinTemp', 'MaxTemp', 'Rainfall']])

plt.show()

**Step 4: Feature Engineering**

Create new features if needed. Here, we extract the month from the date.

python

Copy code

df['Date'] = pd.to\_datetime(df['Date'])

df['Month'] = df['Date'].dt.month

**Step 5: Data Analysis**

Analyze the data to derive meaningful insights. For example, calculate the average maximum temperature by month.

python

Copy code

# Calculate average MaxTemp by month

monthly\_avg\_max\_temp = df.groupby('Month')['MaxTemp'].mean()

**Step 6: Data Visualization (Part 2)**

Visualize the monthly average maximum temperature.

python

Copy code

plt.figure(figsize=(10, 5))

plt.plot(monthly\_avg\_max\_temp.index, monthly\_avg\_max\_temp.values, marker='o')

plt.xlabel('Month')

plt.ylabel('Average Max Temperature')

plt.title('Monthly Average Max Temperature')

plt.grid(True)

plt.show()

**Step 7: Advanced Analysis**

Predict rainfall using a linear regression model.

python

Copy code

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import mean\_squared\_error

# Prepare the data for prediction

X = df[['MinTemp', 'MaxTemp']]

y = df['Rainfall']

# Split the data into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Create and train a linear regression model

model = LinearRegression()

model.fit(X\_train, y\_train)

# Make predictions and calculate the Mean Squared Error

y\_pred = model.predict(X\_test)

mse = mean\_squared\_error(y\_test, y\_pred)

print(f'Mean Squared Error for Rainfall Prediction: {mse}')

**Step 8: Conclusions and Insights**

Identify the highest and lowest rainfall months.

python

Copy code

# Identify the highest and lowest rainfall months

highest\_rainfall\_month = monthly\_avg\_max\_temp.idxmax()

lowest\_rainfall\_month = monthly\_avg\_max\_temp.idxmin()

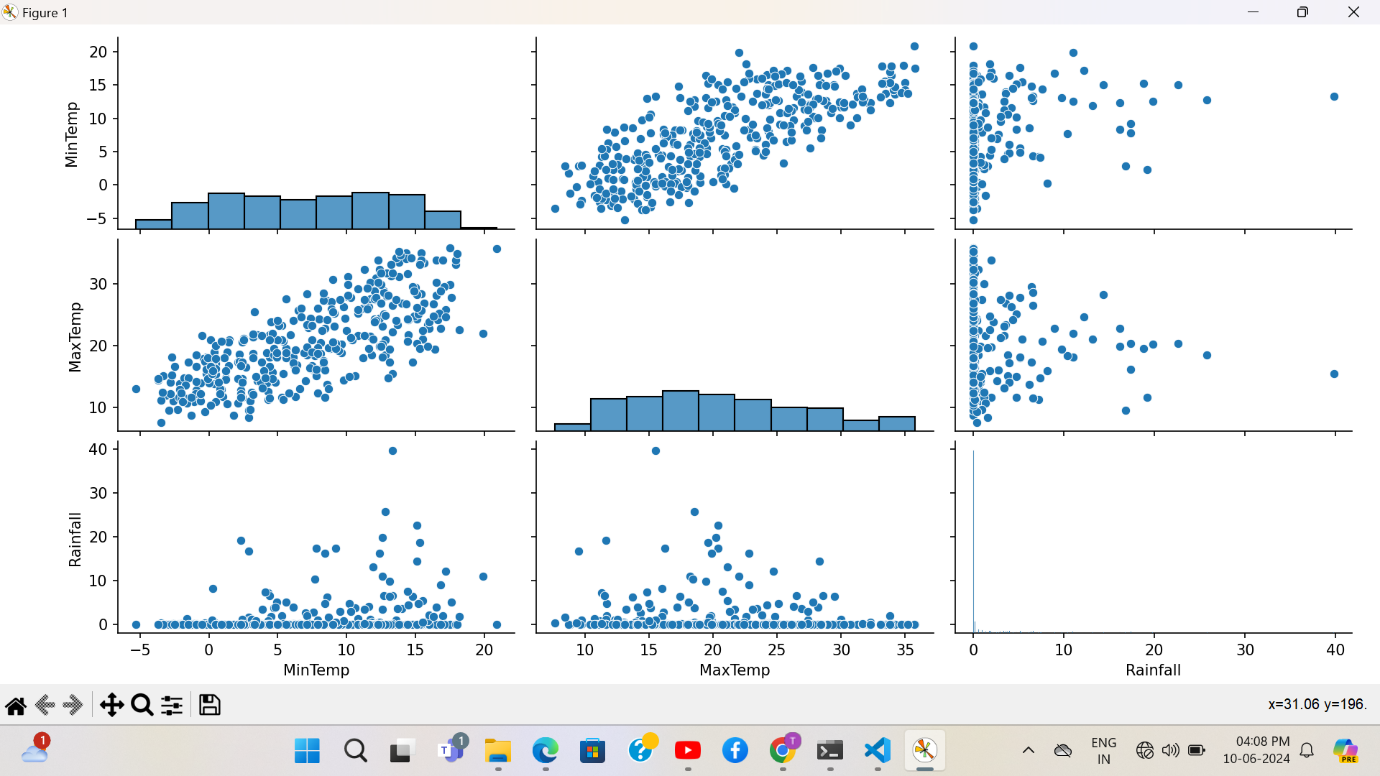
print(f'Highest rainfall month: {highest\_rainfall\_month}, Lowest rainfall month: {lowest\_rainfall\_month}')

**Additional Steps**

**Step 9: Communication**

(Optional) Save or display the results and potentially export to a report or presentation.

**Step 10: Future Work**

(Optional) Plan for further analysis or additional features to be added in future iterations. ****