Project Documentation: Weather Analysis

1. Objectives:

1.1. Prepare and clean a weather dataset using Python.

1.2. Conduct advanced data analysis using Power BI.

1.3. Identify correlations between different weather parameters.

1.4. Perform regression analysis to predict one weather parameter based on others.

2. Data Preparation with Python

Step 1: Load the Dataset

We used Python's pandas library to load the weather dataset from a CSV file.

#

import pandas as pd

weather\_df = pd.read\_csv('weather\_data.csv')

#

Step 2: Inspect and Clean the Data

. Checked for missing values and handled them appropriately.

. Identified and removed any outliers.

. Ensured data consistency and corrected any errors.

#

# Inspect the dataset

print(weather\_df.info())

print(weather\_df.describe())

# Handle missing values

weather\_df = weather\_df.dropna()

# Handle any outliers if necessary

# Example: Removing rows where temperature is outside a reasonable range

weather\_df = weather\_df[(weather\_df['Temperature'] > -50) & (weather\_df['Temperature'] < 50)]

Step 3: Save the Cleaned Data

. Saved the cleaned dataset for further analysis.

#

weather\_df.to\_csv('cleaned\_weather\_data.csv', index=False)

#

3. Advanced Analysis with Power BI

Step 1: Import Cleaned Data into Power BI

3.1. Imported the cleaned dataset (cleaned\_weather\_data.csv) into Power BI.

Step 2: Create Visualizations

3.2. Created various visualizations to highlight trends and patterns:

Line Charts:

Displayed trends over time for temperature, humidity, and other parameters.

Used the date column on the X-axis and weather parameters on the Y-axis.

Bar Charts:

Compared different weather parameters.

Used the date column on the X-axis and weather parameters on the Y-axis.

Scatter Plots:

Explored relationships between different weather parameters.

Used one weather parameter on the X-axis and another on the Y-axis.

Custom Visuals:

Utilized Power BI’s custom visuals for advanced and interactive analysis.

Key Insights:

Observed seasonal trends in temperature and humidity.

Identified periods with unusually high or low rainfall.

Noted the correlation between temperature and humidity.

4. Correlation and Regression Analysis

Correlation Analysis:

Step 1: Calculate the Correlation Matrix in Python

Calculated the correlation coefficients between different weather parameters.

#

correlation\_matrix = weather\_df.corr()

correlation\_matrix.to\_csv('correlation\_matrix.csv')

#

Step 2: Visualize the Correlation Matrix in Power BI

4.1. Imported the correlation matrix (correlation\_matrix.csv) into Power BI.

4.2. Used a Matrix visual or custom heatmap to display the correlation coefficients.

Key Insights:

4.3. Identified strong positive correlations between temperature and humidity.

4.4. Noted negative correlations between wind speed and rainfall.

Regression Analysis:

Step 1: Perform Regression Analysis in Python

Used linear regression to predict one weather parameter based on others.

#

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

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import mean\_squared\_error, r2\_score

X = weather\_df[['Temperature', 'Humidity', 'WindSpeed']] # Example features

y = weather\_df['Rainfall'] # Example target

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

model = LinearRegression()

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

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

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

r2 = r2\_score(y\_test, y\_pred)

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

print(f'R^2 Score: {r2}')

#

Step 2: Visualize Regression Results in Power BI

1. Exported the predictions and actual values to a CSV file.

2. Imported the CSV file into Power BI.

3. Created scatter plots to compare the predicted values against the actual values.

Key Insights:

1. The regression model provided reasonable predictions for rainfall based on temperature, humidity, and wind speed.

2. Identified areas for potential model improvement, such as including additional features or using more complex models.

5. Conclusion

This weather analysis project involved comprehensive data preparation, advanced visualization, and statistical analysis techniques. By leveraging Python and Power BI, we gained valuable insights into weather patterns and their interrelationships. The correlation and regression analyses further helped in understanding the predictive relationships between different weather parameters, enabling more informed decision-making.