**PROJECT 2**

**WEATHER ANALYSIS**

**Data preparation with python.**

First Data collection. Obtain the dataset for example ensure we have a weather dataset. This could include parameters like temperature, humidity, wind speed, etc.

Load the dataset: use libraries like ‘pandas’ to load the dataset as below:

\*import pandas as pd

\*data = pd.read\_csv(‘weather\_data.csv’)

check the first few rows, column names, and data types using below codes:

\*print(data.head())

\*print(data.info())

Decide on a strategy to handle missing values (e.g., removal) as below:

\*data.isnull().sum()

\*data = data.fillna(method=’ffill’)

Use statistical methods or visualization to detect and handle outliers:-

\*import numpy as np

\*from scipy import stats

\*z\_scores = np.abs(stats.zscore(data.select\_dtypes(include=[np.number])))

\*data = data[(z\_scores < 3).all(axis=1)]

Normalize or standardize numerical features if required. Based on domain knowledge, create new features if necessary. Save the cleaned dataset for use in power BI or Tableau as

\*data.to\_csv(‘cleaned\_weather\_data.csv’, index=False)

**Advanced analysis with Power BI or Tableau**

First step is to Load the cleaned dataset into Power BI or Tableau. Second step is to create visualizations. Create dashboards that showcase key metrics like temperature trends, humidity levels, etc. Line chart showing temperature changes over time. Scatter plot to see how humidity and temperature are related. Heatmap or geographical map if location data is available. Add interactive elements to allow users to filter data by date range weather parameters, etc. Last is advanced features. Use trendlines or moving averages to highlight long-term trends. If possible, integrate forecasting models or predictions.

**Correlation and Regression analysis.**

Use Pearson or Spearman correlation to identify relationships between variables as below

\*correlation\_matrix = data.corr()

\*print(correlation\_matrix)

Choose a weather parameter to predict for e.g., temperature. Use linear regression or other models to predict the chosen parameter based on others.

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

\*from sklearn.linear\_model import LinearRegression

\*X = data[['humidity', 'wind\_speed']]

\*y = data['temperature']

\*X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=0)

\*model = LinearRegression()

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

\*predictions = model.predict(X\_test)

Evaluate the model using metrics like R-squared, Mean Squared Error(MSE)

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

\*mse = mean\_squared\_error(y\_test, predictions)

\*r2 = r2\_score(y\_test, predictions)

\*print(f'MSE: {mse}, R^2: {r2}')