import pandas as pd

import matplotlib.pyplot as plt

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

from sklearn.linear\_model import LogisticRegression

# Load the dataset

df = pd.read\_csv('D:\\insurance.csv')

# Display dataset and its shape

print("Dataset preview:")

print(df.head())

print(f"Dataset shape: {df.shape}") # Number of rows and columns in the DataFrame

# Normalize column names to avoid case sensitivity issues

df.columns = df.columns.str.strip().str.upper() # Strip whitespace and convert to uppercase

print("Normalized column names:", df.columns)

# Ensure 'DATE' is converted to datetime with the correct format

df['DATE'] = pd.to\_datetime(df['DATE'], dayfirst=True, errors='coerce')

# Plotting weather data trends

required\_columns = ['DATE', 'PRECIPITATION(LOW)', 'PRECIPITATION(HIGH)', 'WIND(LOW)', 'WIND(HIGH)',

'HUMIDITY(LOW)', 'HUMIDITY(HIGH)', 'TEMPERATURE(LOW)', 'TEMPERATURE(HIGH)']

if all(col in df.columns for col in required\_columns):

plt.figure(figsize=(16, 10))

plt.plot(df['DATE'], df['PRECIPITATION(LOW)'], marker='o', color='red', label='Precipitation (Low)')

plt.plot(df['DATE'], df['PRECIPITATION(HIGH)'], marker='o', color='orange', label='Precipitation (High)')

plt.plot(df['DATE'], df['WIND(LOW)'], marker='o', color='yellow', label='Wind (Low)')

plt.plot(df['DATE'], df['WIND(HIGH)'], marker='o', color='brown', label='Wind (High)')

plt.plot(df['DATE'], df['HUMIDITY(LOW)'], marker='o', color='purple', label='Humidity (Low)')

plt.plot(df['DATE'], df['HUMIDITY(HIGH)'], marker='o', color='pink', label='Humidity (High)')

plt.plot(df['DATE'], df['TEMPERATURE(LOW)'], marker='o', color='blue', label='Temperature (Low)')

plt.plot(df['DATE'], df['TEMPERATURE(HIGH)'], marker='o', color='cyan', label='Temperature (High)')

plt.legend()

plt.xlabel('Date', fontname="times new roman", color="#5F9EA0", size=16)

plt.ylabel('Values', fontname="times new roman", color="#000080", size=16)

plt.title('Weather Data Trends', fontname="times new roman", color="#8A2BE2", size=16)

plt.legend()

plt.show()

else:

print("One or more required columns are missing for plotting.")

# Prepare data for Logistic Regression

if 'PRECIPITATION(HIGH)' in df.columns:

# Create a binary target column

df['PRECIPITATION\_CLASS'] = (df['PRECIPITATION(HIGH)'] < 10).astype(int)

print("Target column PRECIPITATION\_CLASS created.")

else:

print("Error: PRECIPITATION(HIGH) column not found. Cannot create target column.")

df['PRECIPITATION\_CLASS'] = None

# Extract features from 'DATE' for modeling

df['YEAR'] = df['DATE'].dt.year

df['MONTH'] = df['DATE'].dt.month

df['DAY'] = df['DATE'].dt.day

# Check if target column exists and has valid values

if 'PRECIPITATION\_CLASS' in df.columns and df['PRECIPITATION\_CLASS'].notnull().all():

# Splitting the data

X = df[['YEAR', 'MONTH', 'DAY']] # Features

y = df['PRECIPITATION\_CLASS'] # Target variable

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.8, random\_state=0)

# Train a logistic regression model

model = LogisticRegression()

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

# Predictions

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

accuracy = model.score(X\_test, y\_test)

print("\nLogistic Regression Results:")

print(f"Model Accuracy: {accuracy:.2f}")

print("Prediction Probabilities for Test Data:")

print(model.predict\_proba(X\_test))

# Example prediction

example\_date = pd.DataFrame([[2023, 1, 25]], columns=['YEAR', 'MONTH', 'DAY'])

example\_prediction = model.predict(example\_date)

print(f"Prediction for date {example\_date.values.tolist()}: {example\_prediction}")

# Save plot

plt.savefig("project5.jpg")