***PROBLEM STATEMENT****:*

**How can we use temperature, humidity, and wind speed data to predict weather conditions accurately, and what are the most influential meteorological features in this prediction?**

**TITLE:**

**Weather Condition Prediction Using Machine Learning Techniques**

**CODE:**

**import pandas as pd**

**import numpy as np**

**import seaborn as sns**

**import matplotlib.pyplot as plt**

**from sklearn.preprocessing import LabelEncoder**

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

**from sklearn.ensemble import RandomForestClassifier**

**from sklearn.metrics import classification\_report, accuracy\_score, confusion\_matrix**

**# Data Generation**

**np.random.seed(42)**

**n = 10000**

**date = pd.date\_range(start='2025-01-01', periods=n, freq='D')**

**temperature = np.round(np.random.normal(loc=25, scale=5, size=n), 1)  # Mean 25°C, std 5**

**humidity = np.round(np.random.uniform(30, 90, size=n), 1)**

**wind\_speed = np.round(np.random.uniform(0, 30, size=n), 1)**

**weather = np.random.choice(['Sunny', 'Cloudy', 'Rainy', 'Stormy', 'Foggy'], size=n, p=[0.4, 0.3, 0.2, 0.05, 0.05])**

**# DataFrame**

**df\_weather = pd.DataFrame({**

**'date': date,**

**'temperature': temperature,**

**'humidity': humidity,**

**'wind\_speed': wind\_speed,**

**'weather': weather**

**})**

**# First data overview**

**print("First 5 rows of weather data:")**

**print(df\_weather.head())**

**# Missing Values**

**print("\nMissing Values:")**

**print(df\_weather.isnull().sum())**

**# Summary Statistics**

**print("\nSummary Statistics:")**

**print(df\_weather.describe())**

**# Weather Distribution**

**print("\nWeather Condition Distribution:")**

**print(df\_weather['weather'].value\_counts())**

**sns.countplot(x='weather', data=df\_weather, palette='pastel')**

**plt.title('Weather Condition Distribution')**

**plt.show()**

**# Histograms**

**sns.histplot(df\_weather['temperature'], kde=True, color='orange')**

**plt.title('Temperature Distribution')**

**plt.show()**

**sns.histplot(df\_weather['humidity'], kde=True, color='blue')**

**plt.title('Humidity Distribution')**

**plt.show()**

**sns.histplot(df\_weather['wind\_speed'], kde=True, color='green')**

**plt.title('Wind Speed Distribution')**

**plt.show()**

**# Boxplots**

**sns.boxplot(x='weather', y='temperature', data=df\_weather, palette='pastel')**

**plt.title('Temperature by Weather Condition')**

**plt.show()**

**# Encode Weather for ML**

**le\_weather = LabelEncoder()**

**df\_weather['weather\_encoded'] = le\_weather.fit\_transform(df\_weather['weather'])**

**# Features and Target**

**X = df\_weather[['temperature', 'humidity', 'wind\_speed']]**

**y = df\_weather['weather\_encoded']**

**# Train-Test Split**

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

**# Random Forest Classifier**

**rf\_model = RandomForestClassifier(n\_estimators=100, random\_state=42)**

**rf\_model.fit(X\_train, y\_train)**

**rf\_pred = rf\_model.predict(X\_test)**

**# Results**

**print("\n🔷 Random Forest Weather Prediction Results:")**

**print("Accuracy:", accuracy\_score(y\_test, rf\_pred))**

**print("Classification Report:\n", classification\_report(y\_test, rf\_pred))**

**print("Confusion Matrix:\n", confusion\_matrix(y\_test, rf\_pred))**

**# Feature Importances**

**importances = rf\_model.feature\_importances\_**

**feature\_names = X.columns**

**sns.barplot(x=importances, y=feature\_names, palette='pastel')**

**plt.title('Feature Importances (Weather Prediction)')**

**plt.show()**