# WEATHER DATA ANALYSIS

**Name: Krish Gupta**

**Class: CSE-AI (B)**

**Roll No: 202401100300137**

**Course: ARTIFICIAL INTELLIGENCE**

# Introduction:

Weather plays a crucial role in various aspects of life, influencing agriculture, transportation, disaster management, and daily activities. Weather patterns impact human health, economic activities, and even national security. Accurate weather forecasting helps in disaster preparedness, reducing the risk of damage from severe weather events such as hurricanes, storms, and heatwaves.

With advancements in artificial intelligence (AI) and machine learning (ML), analysing weather data has become more efficient and accurate. AI-based models can process vast amounts of historical and real-time weather data, identify patterns, and provide forecasts with improved precision. These AI-driven approaches help in predicting temperature fluctuations, humidity levels, wind patterns, and atmospheric pressure changes, which are crucial for various industries, including agriculture and aviation.

# Methodology:

## 1. Data Collection

The dataset used for this analysis contains historical weather information, including temperature, humidity, pressure, and wind speed. The data was obtained from publicly available weather databases and stored in CSV format.

## 2. Data Preprocessing

- Missing values were identified and removed to maintain data integrity.
- The date column was converted into a datetime format to facilitate time-series analysis.
- Outliers and anomalies in temperature, humidity, and pressure readings were detected and handled accordingly.

## 3. Data Visualization

- Line plots were generated using Matplotlib and Seaborn to observe temperature trends over time.
- Correlation analysis was performed to identify relationships between temperature and other weather variables.

## 4. Model Selection & Training

- A Linear Regression model was used to predict temperature based on humidity, pressure, and wind speed.
- The dataset was split into 80% training and 20% testing sets to evaluate model performance.
- The model was trained on the training data and tested using real-world weather conditions.

## 5. Model Evaluation

To assess the model's accuracy, the following metrics were used:

- **Mean Absolute Error (MAE)** – Measures the average absolute differences between actual and predicted temperatures.
- **Mean Squared Error (MSE)** – Measures the average squared difference between actual and predicted values.
- **Root Mean Squared Error (RMSE)** – Represents the standard deviation of prediction errors.

**6. Prediction & Future Applications**

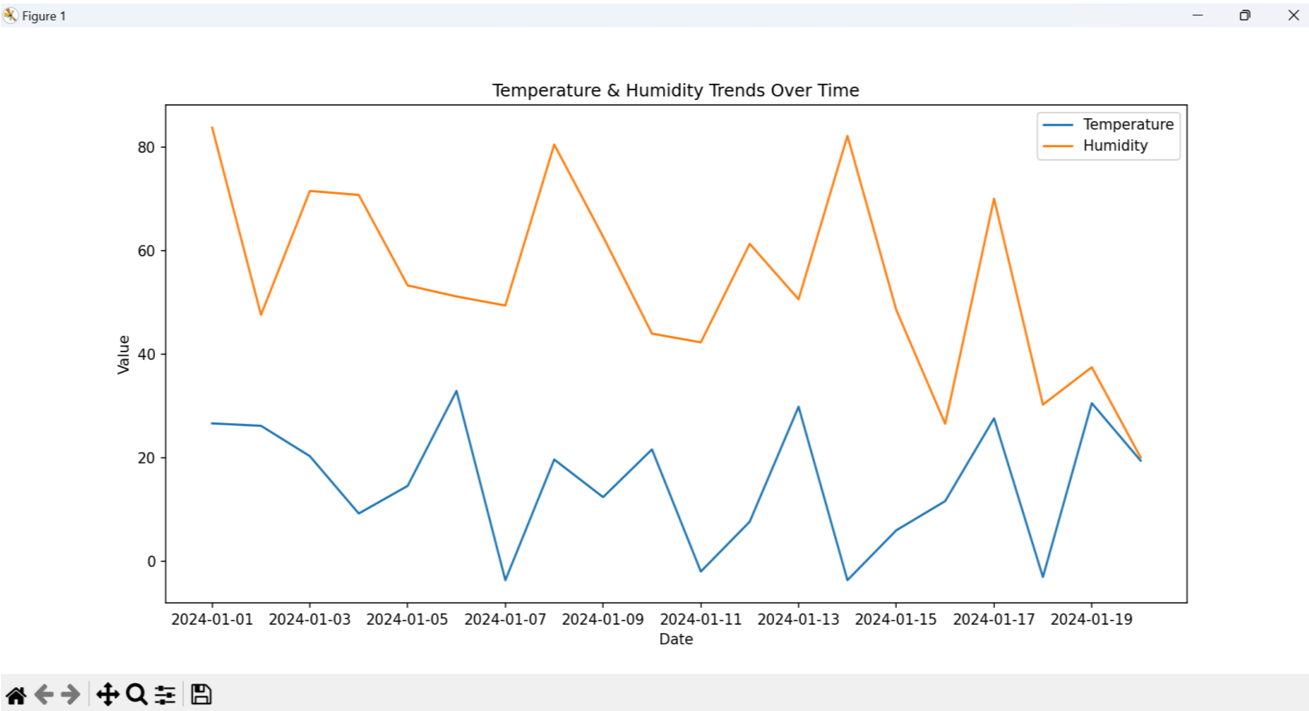
- A function was developed to predict future temperature values based on input weather conditions.
- AI-driven weather prediction models can be further improved with more complex deep learning techniques, such as recurrent neural networks (RNNs) and long short-term memory (LSTM) networks.

## CODE:

```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.model_selection import train_test_split

from sklearn.linear_model import LinearRegression

from sklearn.metrics import mean_absolute_error, mean_squared_error


# Load dataset

file_path = "weather_data.csv"  # Update path if necessary

df = pd.read_csv(file_path)


# Convert 'Date' column to datetime format

df['Date'] = pd.to_datetime(df['Date'])


# Display basic info and statistics

print(df.info())

print(df.describe())


# Check for missing values

df = df.dropna()


# Visualizing data trends

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

sns.lineplot(x=df['Date'], y=df['Temperature'], label='Temperature')

sns.lineplot(x=df['Date'], y=df['Humidity'], label='Humidity')

plt.xlabel('Date')
```

```python
plt.ylabel('Value')
plt.title('Temperature & Humidity Trends Over Time')
plt.legend()
plt.show()

# Correlation heatmap
plt.figure(figsize=(6,4))
sns.heatmap(df[['Temperature', 'Rainfall', 'Humidity']].corr(), annot=True,
cmap='coolwarm', fmt='.2f')
plt.title("Correlation Matrix")
plt.show()

# Splitting data into training and testing sets
X = df[['Rainfall', 'Humidity']]
y = df['Temperature']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Train Linear Regression model
model = LinearRegression()
model.fit(X_train, y_train)

# Make predictions
y_pred = model.predict(X_test)

# Evaluate model performance
mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)
```

```
print(f"Mean Absolute Error: {mae}")

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

print(f"Root Mean Squared Error: {rmse}")
```



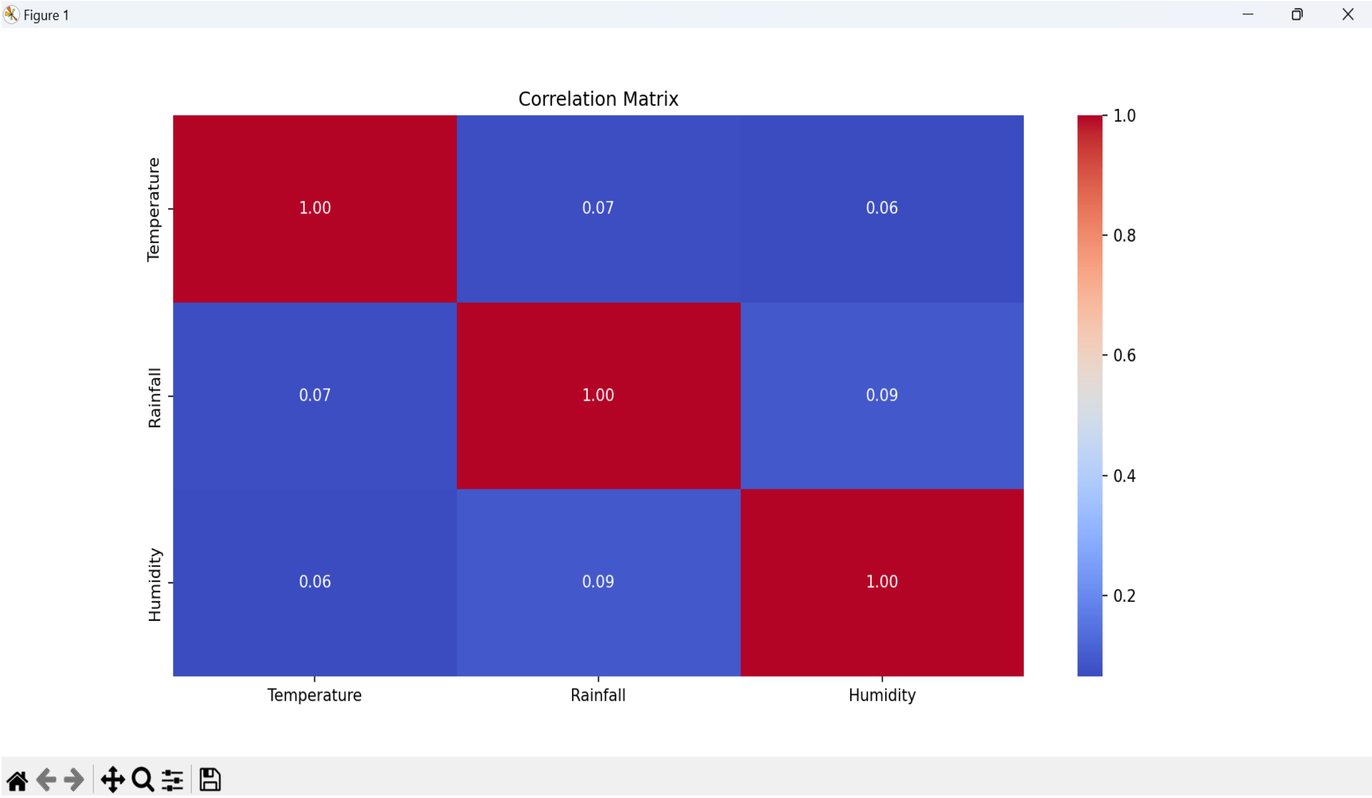


Figure 1
Correlation Matrix
1.00
0.07
0.06
0.07
1.00
0.09
0.06
0.09
1.00
Temperature
Rainfall
Humidity
Temperature
Rainfall
Humidity
1.0
0.8
0.6
0.4
0.2

## Credits:-

This project was developed using Python, leveraging libraries such as:

- **Pandas** for data manipulation
- **NumPy** for numerical computations
- **Matplotlib & Seaborn** for data visualization
- **Scikit-learn** for machine learning model development and evaluation

## References:-


- National Weather Service Data (www.weather.gov)
- Kaggle Weather Datasets (www.kaggle.com)
- Scikit-learn Documentation (https://scikit-learn.org)
- Python Data Science Handbook by Jake VanderPlas