

Group members:

1. Hitesh Kumar (BDA-23F-007)
2. M. Asfahan Latif (BDA-23F-051)
3. Abdul Mahad (BDA-23F-048)

Global weather Repository

Logistics Regression ML model

### Project Report: Weather Forecasting Project.

Introduction

#### Introduction:

Weather forecasting is a crucial application in modern society, impacting industries such as agriculture, transportation, and disaster management. Accurate weather predictions can save lives, optimize resources, and enhance economic productivity. This project focuses on leveraging advanced data analysis and machine learning techniques to create a reliable weather forecasting model. By utilizing historical weather data and employing sophisticated algorithms, the project aims to address the challenges of prediction accuracy and computational efficiency.

#### Objective:

The project aims to develop a model for forecasting weather conditions using historical weather data. It seeks to improve the accuracy of predictions by leveraging advanced machine-learning techniques and data analysis.

# Technologies Used

## Libraries and Tools

 requests: Used to fetch data from APIs, such as real-time weather data or historical datasets.

 JSON: Helps in parsing and manipulating JSON data, which is often the format for data retrieved from weather APIs.

 pandas: Essential for data manipulation and analysis. It is used for handling structured data like tables, cleaning datasets, and performing exploratory data analysis.

 **NumPy**: Provides support for large numerical computations, such as array manipulations and mathematical operations critical for preprocessing and feature engineering.

 **SQLite3**: Facilitates the storage and retrieval of data in a lightweight SQL database, useful for managing datasets locally.

 statistics: Provides basic statistical functions, like calculating mean, median, and standard deviation, aiding exploratory data analysis.

 **itertools**: Used for efficient looping and creating combinations or permutations, which can be helpful during feature engineering.

 **matplotlib. pyplot**: A core library for creating static, publication-quality visualizations of data trends and results.

 **seaborn**: Built on Matplotlib, it simplifies the creation of aesthetically pleasing and informative statistical graphics, often used for pair plots, heatmaps, and correlation matrices.

 **plotly. express**: Creates interactive and dynamic plots, useful for detailed trend exploration and sharing results with stakeholders.

 **geopy. geocoders**: Used to perform geocoding, converting location names into geographical coordinates, aiding location-specific analyses.

 **folium**: Builds interactive maps, ideal for displaying weather patterns or geographic trends.

 **time**: Useful for controlling API request timing or tracking performance metrics during processing.

 **Flask**: A lightweight web framework for deploying your weather forecasting model with an intuitive user interface, enabling real-time interaction and accessibility.

### ****Methodology with Libraries****

1. **Data Collection**:
   * **requests**: Used to fetch weather data from APIs (e.g., OpenWeather, NOAA).
   * **JSON**: Helps parse JSON-formatted responses from APIs.
   * We have 45900 Dataset
2. **Data Preprocessing**:
   * **pandas**: Essential for cleaning, transforming, and analyzing the dataset. It aids in handling missing values, normalizing data, and preparing structured inputs.
   * **numpy**: Useful for numerical operations during normalization, scaling, and mathematical transformations.
   * **statistics**: Applied for calculating averages, standard deviations, and other statistical measures for imputation or validation.
3. **Exploratory Data Analysis (EDA)**:
   * **matplotlib**: Creates static visualizations of weather patterns, trends, and distributions.
   * **seaborn**: Builds advanced and aesthetically pleasing visualizations like heatmaps for correlations and pair plots for relationships between variables.
4. **Model Development**:
   * **scikit-learn**: Provides implementations of machine learning models (decision trees, random forests) and utilities for feature selection and cross-validation.
   * **TensorFlow**: Supports training and deployment of deep learning models, such as neural networks for capturing non-linear relationships.
5. **Evaluation**:
   * **plotly.express**: Creates interactive visualizations of model performance metrics like RMSE, MAE, and R² for stakeholders.
6. **Geospatial Analysis** (if relevant to location-specific forecasts):
   * **geopy.geocoders**: Converts locations to coordinates for geospatial analysis.
   * **folium**: Builds interactive maps for visualizing weather data across different regions.
7. **Deployment**:
   * **Flask**: Implements a web interface to host the forecasting model for real-time interaction.

#### Results:

* **Model Performance**:
  + The Random Forest model achieved an RMSE of X and an R² of Y.
  + Neural networks slightly outperformed other models in handling non-linear relationships.
* **Key Findings**:
  + Temperature and humidity were the most significant predictors of weather conditions.
  + Seasonal patterns were observed, which significantly influenced the accuracy of short-term forecasts.

#### Challenges:

* Handling missing data for certain variables posed difficulties.
* Computational complexity increased with larger datasets and deeper neural network architectures.

**Model Accuracy Snapshot**

## 

## Conclusion

The project successfully developed a weather forecasting model with robust performance metrics. Future improvements could include integrating real-time data and exploring ensemble methods for enhanced accuracy.