Rainfall Prediction – Weather Forecasting for Agriculture

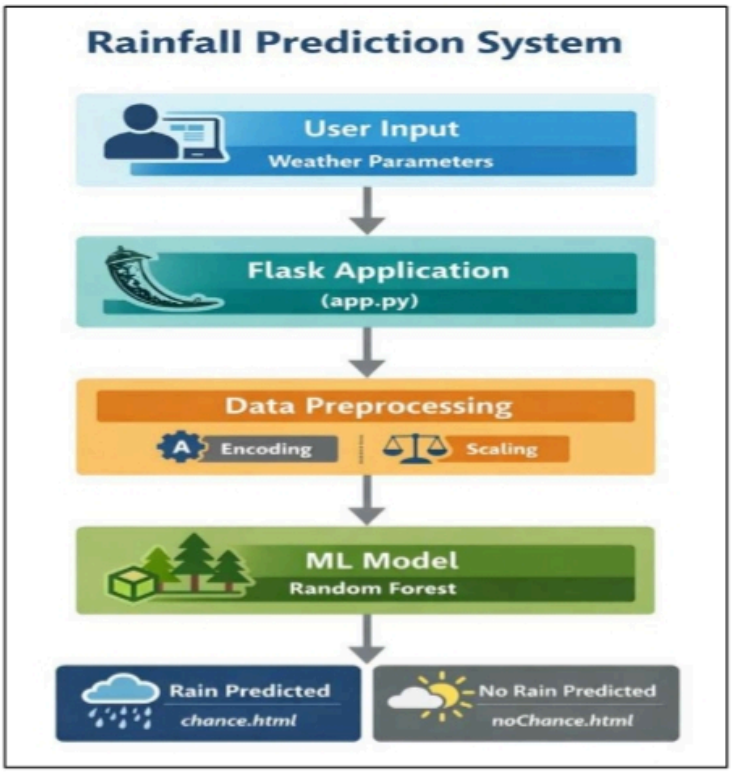
Project Description:

Rainfall plays a crucial role in agriculture, water resource management, and climate planning. Predicting rainfall accurately helps farmers, agricultural experts, and policymakers make informed decisions regarding crop planning, irrigation scheduling, and disaster management.

Rainfall prediction is a complex task due to the dynamic nature of weather conditions. By leveraging Machine Learning techniques, we can analyze historical weather data and predict whether it will rain tomorrow.

In this project, we use classification algorithms such as Decision Tree, Random Forest, Support Vector Machine (SVM), Gradient Boosting, and XGBoost. The best-performing model is selected based on accuracy and evaluation metrics and saved in .pkl format. The model is integrated into a Flask web application and deployed for real-time prediction.

Technical Architecture:



Pre requisites:

To complete this project, you must required following software's, concepts and packages

- **Anaconda navigator and pycharm:**
  - Refer the link below to download anaconda navigator
  - Link : https://youtu.be/1ra4zH2G4o0
- **Python packages:**
  - Open anaconda prompt as administrator
  - Type "pip install numpy" and click enter.
  - Type "pip install pandas" and click enter.
  - Type "pip install scikit-learn" and click enter.
  - Type "pip install matplotlib" and click enter.
  - Type "pip install scipy" and click enter.
  - Type "pip install pickle-mixin" and click enter.
  - Type "pip install seaborn" and click enter.
  - Type "pip install Flask" and click enter.
  - Type "pip install xgboost" and click enter.

Prior Knowledge:

You must have prior knowledge of following topics to complete this project.

- **ML Concepts**
  - Supervised learning: https://www.javatpoint.com/supervised-machine-learning
  - Unsupervised learning: https://www.javatpoint.com/unsupervised-machine-learning
  - Regression and classification
  - Decision tree: https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm
  - Random forest: https://www.javatpoint.com/machine-learning-random-forest-algorithm
  - KNN: https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning
  - Xgboost: https://www.analyticsvidhya.com/blog/2018/09/an-end-to-end-guide-to-understand-the-math-behind-xgboost/
  - Evaluation metrics: https://www.analyticsvidhya.com/blog/2019/08/11-important-model-evaluation-error-metrics/
- **Flask Basics** : https://www.youtube.com/watch?v=lj4I_CvBnt0

Project Objectives:

By the end of this project you will:

- Understand data preprocessing techniques
- Perform exploratory data analysis
- Apply classification algorithms
- Compare multiple ML models

- Deploy ML model using Flask
- Understand real-world ML deployment workflow

Project Flow:

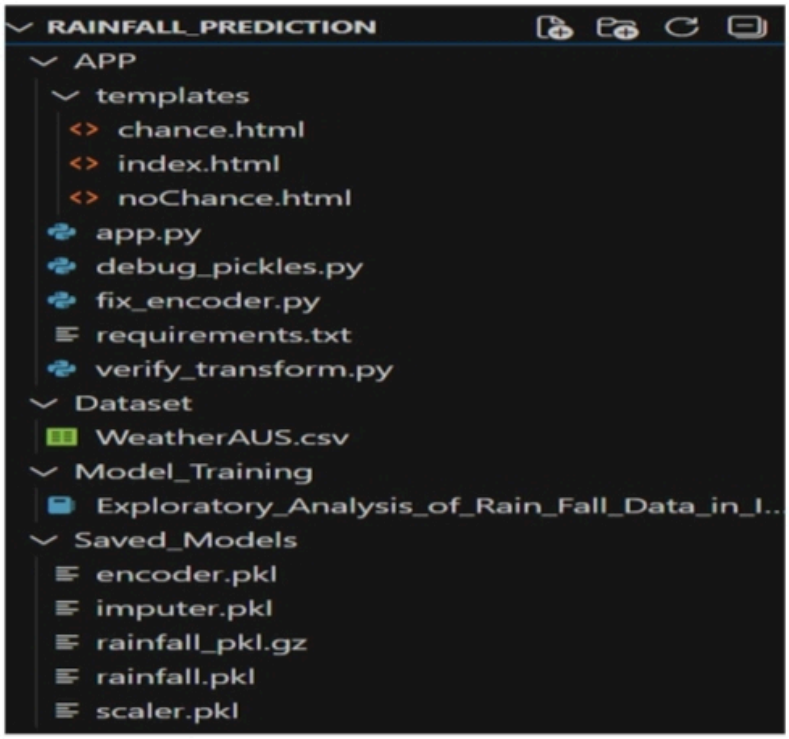
- User interacts with the UI and enters weather parameters.
- The input values are sent to the Flask backend.
- The backend preprocesses the input (encoding + scaling).
- The trained model predicts rainfall.
- Prediction result is displayed on UI.

To accomplish this, we have to complete all the activities listed below,

- Data collection
    - Collect the dataset or create the dataset
- Visualizing and analyzing data
    - Univariate analysis
    - Bivariate analysis
    - Multivariate analysis
    - Descriptive analysis
- Data pre-processing
    - Checking for null values
    - Handling outlier
    - Handling categorical data
    - Splitting data into train and test
- Model building
    - Import the model building libraries
    - Initializing the model
    - Training and testing the model
    - Evaluating performance of model
    - Save the model
- Application Building
    - Create an HTML file
    - Build python code

Project Structure:

Create the Project folder which contains files as shown below



- We are building a flask application which needs HTML pages stored in the templates folder and a python script app.py for scripting.
- rainfall.pkl is our saved model. Further we will use this model for flask integration.
- Training folder contains model training files and templates folder contains IBM deployment files.

Milestone 1: Data Collection

ML depends heavily on data, It is most crucial aspect that makes algorithm training possible. So this section allows you to download the required dataset.

Activity 1: Download the dataset

There are many popular open sources for collecting the data. Eg: kaggle.com, UCI repository, etc.

In this project we have used WeatherAUS.csv data. This data is downloaded from kaggle.com. Please refer to the link given below to download the dataset.

Link:

https://docs.google.com/spreadsheets/d/1RA2OO0LZTeQykI_mvnensAjp6LM4YzWI1Tz0SUG5-Ao/edit?gid=121883362#gid=121883362

The dataset contains historical weather observations including:

- Location
- Temperature
- Rainfall
- Humidity
- Pressure
- Wind Speed
- Rain Tomorrow (Target Variable)

Milestone 2: Data Visualizing and analysis

As the dataset is downloaded. Let us read and understand the data properly with the help of some visualization techniques and some analysing techniques.

Note: There is n number of techniques for understanding the data. But here we have used some of it. In an additional way, you can use multiple techniques.

Activity 1: Importing the libraries

Import the necessary libraries as shown in the image. (optional) Here we have used visualization style as fivethirtyeight.

```
import pandas as pd
import numpy as np
import pickle
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
import sklearn
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import GradientBoostingClassifier, RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import RandomizedSearchCV
import imblearn
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix, f1_score
```