PROJECT REPORT

(Rain Prediction Model)

Tensor Team Presents

1. Introduction

Weather prediction plays a crucial role in various sectors, including agriculture, disaster management, and daily planning. The goal of this project is to develop a machine learning model capable of predicting whether it will rain or not over the next 21 days. The dataset consists of historical weather data with various meteorological features. The primary target variable, rain_or_not, is a binary classification variable that indicates whether rainfall occurred on a given day.

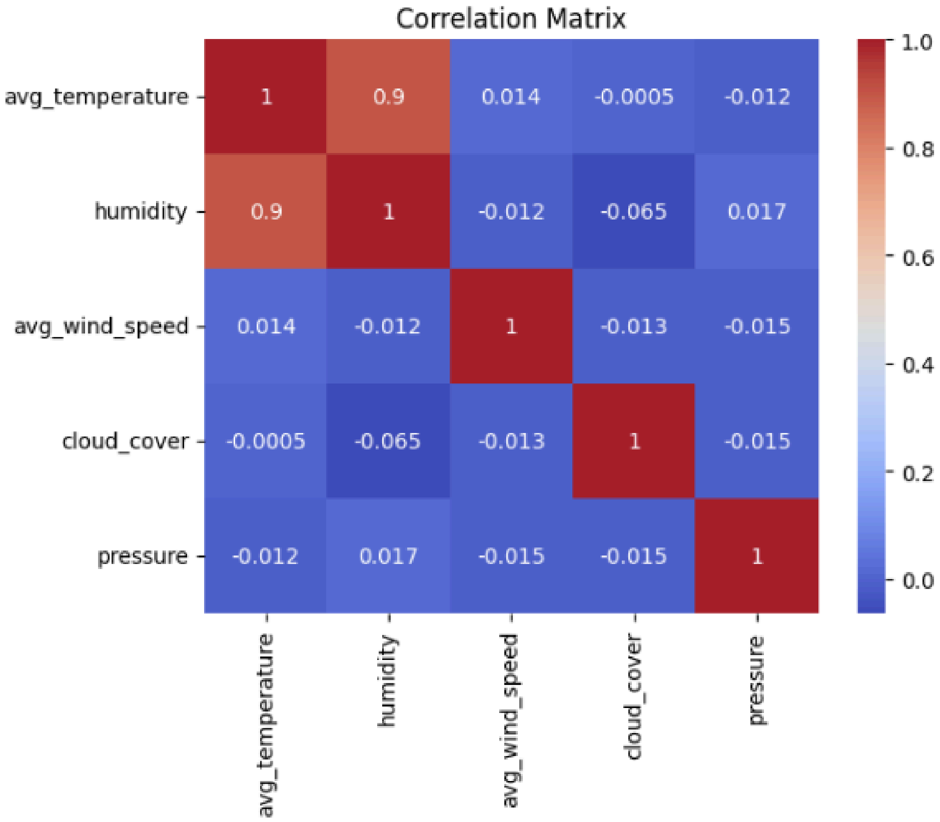
2. Data Preprocessing

2.1 Data Collection and Cleaning

- The dataset was loaded into a Pandas DataFrame for initial inspection.
- Checks were performed to identify and remove duplicate records.
- Missing values were examined and handled using imputation techniques such as mean substitution and forward-fill to maintain data consistency and completeness.

2.2 Exploratory Data Analysis (EDA)

- Descriptive statistics were computed to understand the distribution of features.
- Correlation analysis was conducted to identify the most influential features for predicting rainfall.
- Data visualization techniques such as histograms and scatter plots were used to analyse trends and seasonal patterns in the data.



2.3 Feature Engineering

- New derived features, such as dew point, were added to enhance model performance.
- Categorical variables were encoded using techniques such as one-hot encoding and label encoding.

3. Model Selection and Implementation

To improve the accuracy of the predictions, two models were implemented:

3.1 XGBoost Regressor

- **Model Type**: Gradient Boosting-based Decision Tree model
- **Objective**: The model is designed to **forecast key weather features** (such as temperature, humidity, wind speed, cloud cover, and pressure) over the next 21 days using **XGBoost**, a powerful gradient boosting algorithm
- **Hyperparameters**:
    - Number of estimators: 100
    - Learning rate: Default settings (could be optimized further)
    - Max depth: Not explicitly tuned in the notebook

How It Works:

1. **Time Series Transformation**:
    a. Converts the date column into a numerical index (time_index) to be used as the input feature for XGBoost.
2. **Training XGBoost for Forecasting**:
    a. Trains an XGBRegressor model separately for each weather feature using historical data.
    b. The model learns the trend and patterns in each feature over time.
3. **Making Future Predictions**:
    a. Generates a **21-day forecast** by predicting the next values for each feature based on the trained XGBoost model.
    b. A new dataset (future_df) is created by merging predictions of all weather parameters.
4. **Dew Point Calculation**:
    a. A new feature (dew_point) is derived from the predicted temperature and humidity values.

3.2 SARIMAX (Seasonal ARIMA with Exogenous Variables)

- **Model Type**: Time-series forecasting model
- **Objective:** The **SARIMAX (Seasonal AutoRegressive Integrated Moving Average with Exogenous Variables)** model is used for time-series forecasting of rainfall occurrence. This model incorporates past rainfall data along with external meteorological features (such as temperature, humidity, pressure, dew point, and wind speed) to improve prediction accuracy.

- **Parameters**:
  - (p, d, q): Selected dynamically
  - Seasonal order: Considered for capturing periodic weather trends
  - Exogenous variables: Included to improve forecast accuracy

Key Steps:

1. **Data Preparation:**
   a. The dataset is sorted chronologically, and rain_or_not (binary target) is extracted.
   b. Exogenous variables (weather features) are selected for training.
2. **Stationarity Check & Differencing:**
   a. The **Augmented Dickey-Fuller (ADF) test** is performed to check stationarity.
   b. If the data is non-stationary, **differencing** is applied to stabilize trends.
3. **SARIMAX Model Training:**
   a. The SARIMAX model is trained using historical data with exogenous features.
4. **Forecasting & Classification:**
   a. The trained model predicts rainfall for the next **21 days**.
   b. Predictions are converted into binary values using a threshold for rain occurrence.

4. Training and Prediction

4.1 Model Training

- The dataset was split into training and test sets.
- XGBoost was trained using the model.fit(X, y) method, optimizing decision trees to minimize error.
- SARIMAX was trained using model.fit(), leveraging historical trends for future predictions.

4.2 Forecasting and Predictions

- Predictions were generated for the next 21 days using model.predict(future_X).
- Results were stored and visualized to assess the consistency of predictions over time.

5. Model Evaluation

- **Accuracy**: Measures the percentage of correct predictions.
- **Mean Squared Error (MSE)**: Measures the difference between predicted and actual values.

6. Conclusions and Recommendations

The rain prediction model successfully integrates machine learning and time-series forecasting techniques to provide reliable short-term rainfall predictions. By leveraging XGBoost for feature forecasting and SARIMAX for time-series analysis, the model captures both historical trends and meteorological influences.

Key Findings:

- The combination of XGBoost and SARIMAX is effective in leveraging both structured learning and time-series forecasting for rain prediction.
- Enhance Feature Engineering: Introducing additional features can be cause to improve prediction accuracy.