**GROUP - 8**

**PREDICTING RAIN WITH MACHINE LEARNING**

**CAP 5610 - Final Project Report**

**Problem Statement**

The agricultural industry heavily relies on weather conditions for effective decision-making in farming activities. Predicting weather conditions can be challenging, especially in regions with unpredictable weather patterns, leading to farmers making suboptimal decisions that impact yields and cost them valuable resources.

Traditional methods of weather forecasting can be time-consuming and may not always provide accurate results. In recent years, climate change has made weather patterns even more unpredictable, exacerbating the problem of inaccurate weather predictions. There is a pressing need to develop more accurate and efficient methods for predicting rainfall.

Machine learning algorithms have shown promise in weather prediction, as they can process large amounts of data and identify complex patterns that traditional methods may miss. Developing an accurate machine learning algorithm for rainfall prediction will be beneficial for the agricultural industry, especially for farmers in regions with unpredictable weather patterns.

**Motivation**

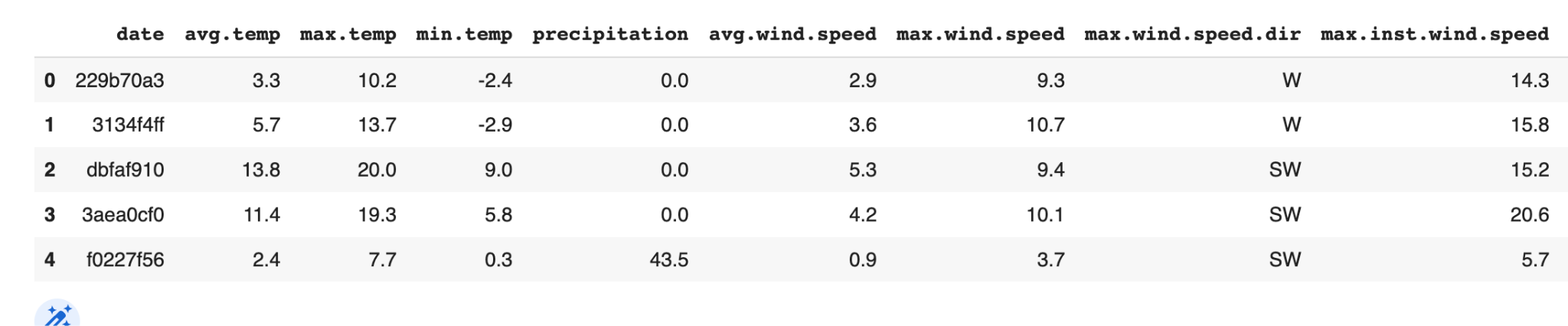
The motivation behind this project is to address the challenges that farmers face in making informed decisions regarding their farming activities due to the unpredictable nature of weather patterns. By developing a machine learning algorithm that accurately predicts rainfall, farmers will have access to more reliable weather forecasts, enabling them to plan their activities more effectively.

This will help them to optimize their production, minimize costs, and maximize growth efficiency. The algorithm will analyze historical weather data to identify complex patterns that traditional methods may miss, enabling it to make more accurate predictions This will lead to increased productivity, better crop yields, and higher profits for farmers, contributing to the overall growth of the agricultural industry.

**Introduction and Description of Data**

The unpredictability of weather patterns has been a significant challenge.This project aims to provide a practical solution to address the challenges of weather prediction in the agricultural industry, contributing to the overall growth and sustainability of the sector.

The data has been conveniently split into train and test datasets. In each train and test, weather data is present which consists of locations named region A through region E, which are all neighbouring regions including an anonymized date column. The further features included in the dataset are as follows:

Figure 1: Glimpse of the dataset

As it is seen from the dataset, the “date” column is anonymized to some random values. There are in total 10 features in the dataset which consists of types and abbreviations of temperature, wind speed, precipitation, wind speed direction and atmospheric pressure.

The dataset contains 5 csv files in each training and testing set along with a separate csv file named “solution\_format.csv” containing target rain predictions for each of the dates, which allows us to use supervised learning when building the model.

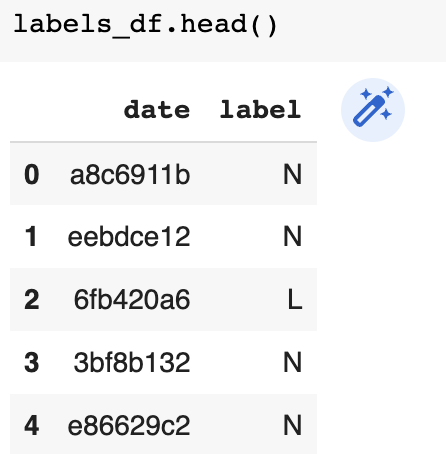


Figure 2: Solution\_format.csv

The goal currently is to predict the weather for the next day based on three labels:

* N - No rain
* L - Light Rain
* H - Heavy Rain

In order to perform weather forecasting for rain occurence, the following tools and technology were utilised:-

1. Python - To implement weather forecasting using machine learning
2. Jupyter- Notebook - To implement Python Code
3. Matplotlib and Seaborn Library in python - To perform EDA

**Modelling Approach**

In order to develop an accurate machine learning model for predicting rainfall, a systematic approach is followed, starting by gathering and preprocessing weather data from the region of interest, including variables such as temperature, humidity, and wind speed. Based on that most relevant variables that influence rainfall are selected and perform feature selection and engineering is performed to reduce noise in the dataset.

Next, a variety of algorithms such as Random Forest, XGBoosting , LightGBM and Stacking have been applied to train and evaluate our model. Finally, hyperparameter tuning is performed to evaluate its performance on a test dataset to ensure it meets the desired accuracy and reliability criteria. By following this systematic approach, the aim to develop a machine learning model that accurately predicts rainfall is implemented. Following are the steps taken for model building:-

1. **Data Preprocessing -** Data preprocessing refers to a series of techniques used to prepare raw data for analysis. The following are the steps taken to perform data preprocessing:-

* Concatenated the segregated data files of training dataset based on regions.
* Created a missing values function to find missing values in each feature in training and testing dataset and segregate it.
* Plotted distribution of each feature.
* As the minimum atmospheric pressure column had the majority of data missing from Regions B,C,D and E, feature elimination was performed and the column was dropped.
* Based on distribution mean values of features were imputed in remaining respective features based on the percentage of values missing in training and testing dataframes.

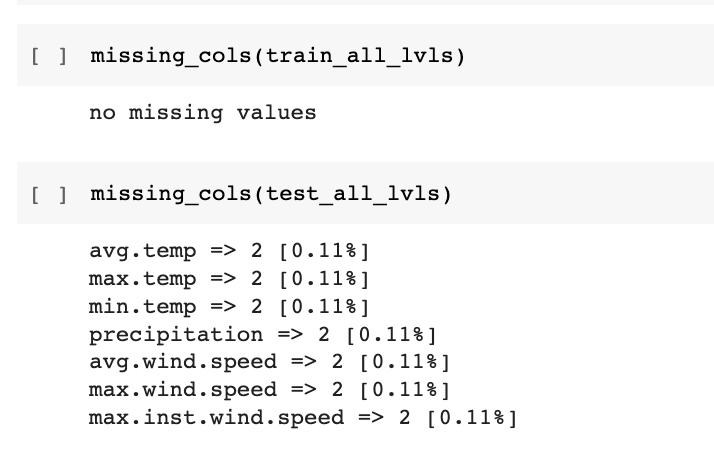


Figure 1. Missing Values in combined train and test dataset

1. **Feature Engineering -** Feature engineering is the process of transforming raw data into features that better represent the underlying problem to improve the performance of machine learning models. The following are the steps taken to perform feature engineering:-

* Merged Labels data file with the main training data based on date feature.
* Converted categorical variables into numeric using LabelEncoder().
* Pivoted data and gave each feature name based on region.(eg: avg\_temp\_A).
* Created a new feature called “Beaufort Scale” and applied on training and testing datasets.

1. **Model Building and Algorithm Application-** Model building and machine learning algorithm application refer to the process of selecting and training a machine learning model on a dataset to make predictions or classifications. The following algorithms were implemented to predict rainfall on respective dates and regions:-
   1. **XGBoosting -** Gradient boosting is an ensemble learning technique that combines multiple weak models (decision trees in the case of XGBoost) to create a single strong model. The following steps were taken to build the model using XGBoosting:-
      1. RFE function was implemented with Random Forest Classifier to select and fit features within the training data.
      2. Performed oversampling to deal with imbalance data using RandomOverSampler() function.
      3. Data was standardized using StandardScaler()
      4. Scaled data was passed through XGBClassifier and fitted using score() function.
      5. Accuracy - 83.56%
   2. **RandomForest Classification -** Random Forest Classification is a type of machine learning algorithm that uses an ensemble of decision trees to classify data. The following steps were taken to build the model using Random Forest Classification:-
      1. RFE function was implemented with Random Forest Classifier to select and fit features within the training data.
      2. Performed oversampling to deal with imbalance data using RandomOverSampler() function.
      3. Data was standardized using StandardScaler().
      4. Data was splitted using train\_test\_split() function to fit using RandomForestClassifer() and tested for accuracy.
      5. Hyperparameter tuning was performed by defining an objective function using a single parameter which takes into account search space for each hyperparameter. The hyperparameters used are:- criterion,bootstrap,max\_depth,max\_features,max\_leaf\_nodes, n\_estimators and random state.
      6. A random forest classifier is then instantiated again using these hyperparameters by passing them to the RandomForestClassifier constructor and fitted.
      7. The cross\_val\_score function is then used to compute the accuracy. This function is then passed through optimize() function of optuna stacking to obtain final accuracy.
      8. Accuracy - 87.32%
   3. **LightGBM (Gradient Boosting Framework) -** LightGBM is an open-source gradient boosting framework that is designed to be efficient, scalable, and high-performance.The following steps were taken to build the model using LightGBM:-
      1. Data types were converted into categorical.
      2. Target class was dropped and training data was split using train\_test\_split() function (75%:25%).
      3. Data was fitted using the LGBMClassifier() and fitted to test for accuracy.
      4. Hyperparameter tuning was performed by defining an objective function using a single parameter which takes into account search space for each hyperparameter. The hyperparameters used are:- criterion,bootstrap,max\_depth,max\_features,max\_leaf\_nodes, n\_estimators and random state.
      5. A LGBMClassifier is then instantiated again using these hyperparameters by passing them to the LGBMClassifier constructor and fitted.
      6. The cross\_val\_score function is then used to compute the accuracy. This function is then passed through optimize() function of optuna stacking to obtain final accuracy.
      7. Confusion matrix was plotted to know occurrence of data and evaluating model performance.
      8. Classification report was implemented to analyze overall model results.
      9. Accuracy - 83.09%
   4. **Stacking Classification -** Stacking is an ensemble machine learning technique that involves training multiple models, using their predictions as input, and then combining them to make a final prediction. Stacking can be applied to classification problems. The following are the steps taken to implement the stacking classification:-
      1. A stacked classification model is created using the StackingClassifier and the final estimator of Logistic Regression is passed to compute classification.
      2. The stacked classification model is trained on the training data using the fit method.
      3. Predictions are made on the test data using the predict method of the trained stacked classification model.
      4. The accuracy score of the stacked classification model on the test data is calculated using the accuracy\_score function.
      5. Accuracy - 90.14%

**Results and Interpretation**

The machine learning models implemented performed reasonably well providing a decent accuracy to go forward with predicting rain in different regions based on the features available.Glancing at the overview of models and how they performed for the respective data available to them.

**Overview of Models:**

The following are the machine learning techniques used to implement classification model for predicting rain and their accuracies are shown:

1. XGBoosting - 83.46%
2. Random Forest Classification - 87.32%
3. LightGBM - 83.09%
4. Stacking - 90.14%

Based on the results obtained in the respective machine learning models, Stacking classification is used to proceed with predicting rain using machine learning

**Confusion Matrix:**

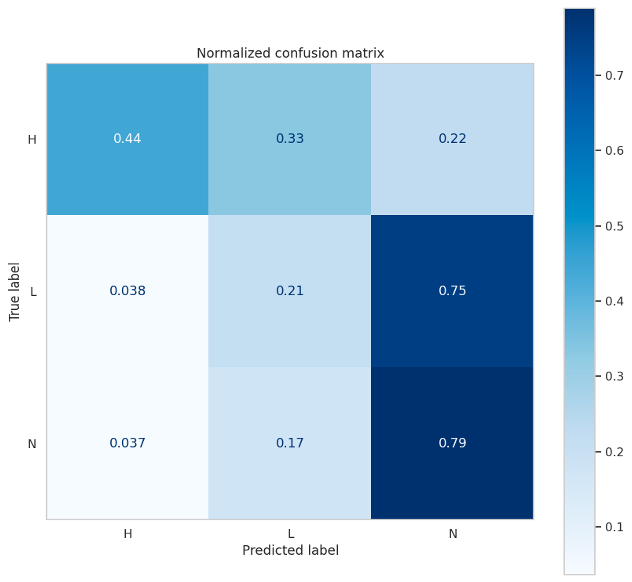
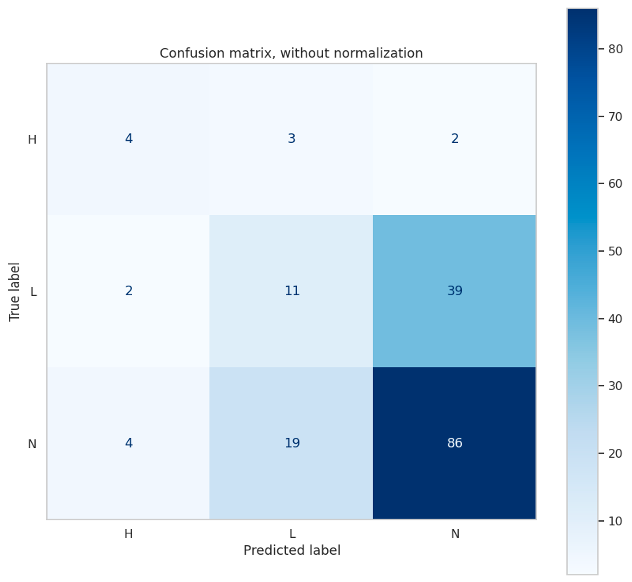
****

Figure 2 Non-normalized Vs. normalized confusion matrix

A non-normalized confusion matrix is simply a count of the number of test samples that were correctly or incorrectly classified by the model, without any further normalization or scaling of the data. On the other hand, a normalized confusion matrix scales the counts so that they are expressed as proportions or percentages of the total number of test samples.

Based on Figure 2, it is quite visible from the shade of the plot that label “N” which indicates “No-rain‘ is predicted more than the others.

**Feature Importance**

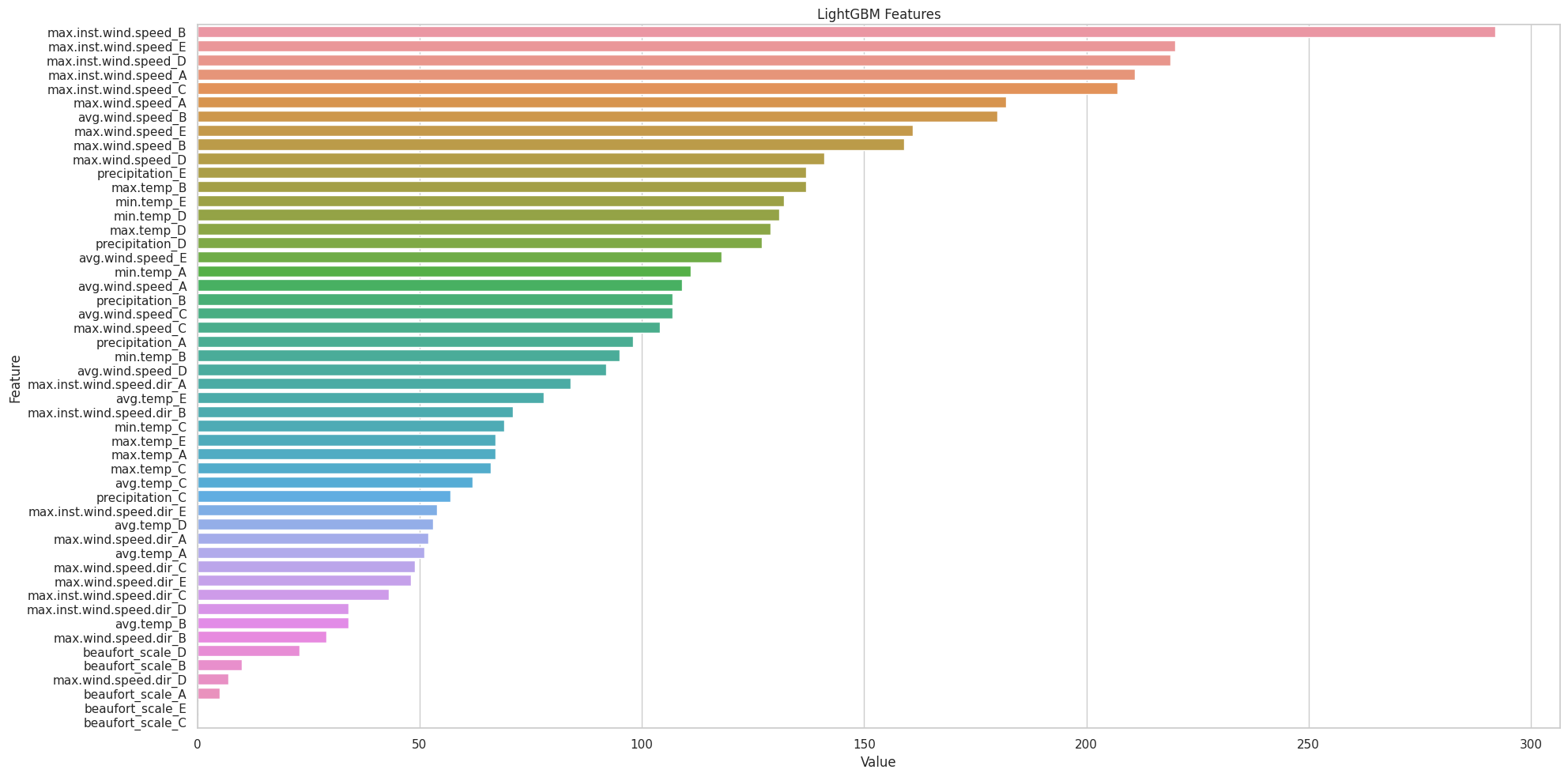
****

Figure 3 Feature Importance

From figure 3 it can be interpreted that the wind speed features and precipitation are the key features that are good predictors. The Beaufort scale feature created seems to have very low importance.

Stacking Classifier is used to predict the labels of the testing data and a submission dataframe is created. The submission dataframe includes the date and the predicted label for each instance in the testing data. The head() function is used to display the first few rows of the submission dataframe.

The categorical labels which were mapped into numeric and mapped back to categorical labels and appended into the final result dataframe.

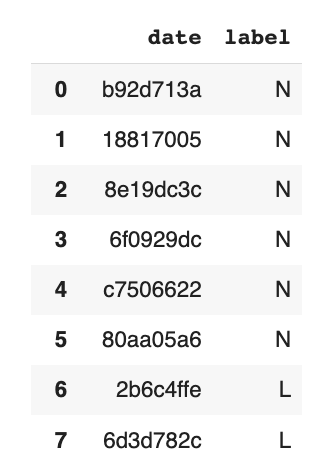


Figure 4 Predicted Rain on respective dates

From figure 4, it is quite visible that labels are appended on the respective dates which had rainfall, based on the data available the date column was anonymized.

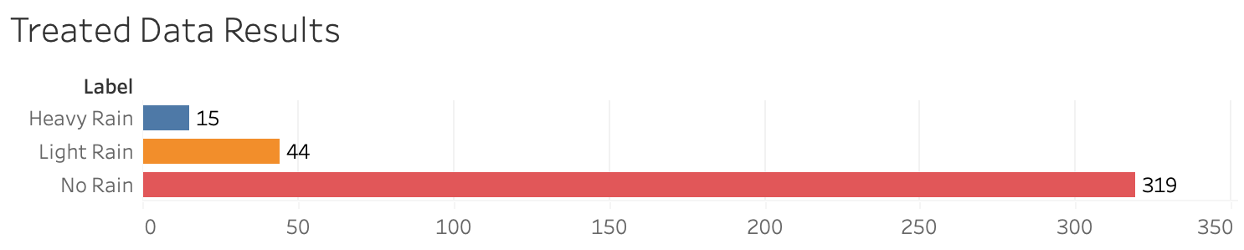


Figure 5 Occurrence of rainfall

From figure 5, it can be interpreted that maximum times there was no rain followed by light rain and heavy rain based on the data available and features used to predict rainfall.

**Future Work**

There is always availability to test the data with more precision and approaching from a different perspective to get more accurate predictions. So, the base model won’t be enough to make a good prediction; here are some additional steps to improve upon the given approach.

* More feature preprocessing and engineering can prove beneficial.
* Using cross-validation to have a better measure of the performance.
* Tuning class\_weight parameter of respective classifier directly to handle imbalance classes.
* Balance the dataset by utilizing undersampling or resampling.
* Test out other algorithms like KNN, SVM, CatBoost etc…
* Moreover, it always depends on the amount of availability of data and features to predict rainfall and forecast weather in a more precise manner.

**Conclusion**

Predicting rainfall with machine learning can provide valuable insights for the agriculture industry, assisting farmers with planning and ensuring food supply and security. However, the data for weather prediction can be vague and require preprocessing to build an accurate predictive model.This is where data preprocessing involves handling missing or irrelevant data, normalizing or scaling the data, and encoding categorical features. The techniques used in building the machine learning model for predicting rain included LightGBM, Random Forest Classification, Optuna Stacking, and XGBoosting. These techniques provide comprehensive results, with each model performing differently and offering unique strengths and weaknesses.

The stacked model that combined these models together also demonstrated promising results, highlighting the importance of using different techniques to improve the performance of the model.However, while machine learning can provide valuable insights, it should not be solely relied upon as the only method for predicting rainfall. Other factors, such as expert knowledge and data from other sources, should also be considered for informed decision making.

In conclusion, predicting rainfall through machine learning is an extensive process that requires careful data preprocessing, model selection, and evaluation. However, the results obtained through the techniques used in this study demonstrate that machine learning can provide valuable insights for predicting rainfall and help the agriculture industry towards food security and ensuring that crops are grown efficiently. By continuing to explore and develop machine learning models, we can build on these results and potentially improve the accuracy and reliability of predicting rainfall with machine learning.