IEEE
UNIVERSITY OF COLOMBO SCHOOL OF COMPUTING
STUDENT BRANCH
UCSC

IntelliHack 5.0

TEAM HYPER TUNERS

TASK 01

Weather Forecasting

intelli Hack 5.0
Artificial . But Intelligent

TABLE OF CONTENT

INTRODUCTION

Weather forecasting is an important aspect of smart agriculture, as it allows farmers to make better decisions related to irrigation, planting, and harvesting. However, conventional weather forecasting methods often lack precision with respect to local conditions. To address this issue, Our Team, **Hyper Tuners**, designed a weather forecasting model that applies machine learning to forecast the probability of rainfall over the next 21 days by using past weather data.

Recently, technology has been used as a very important resource that enables forecasters to predict the weather more accurately by using a technique called weather satellite imaging and data analytics. Farmers can make the most of their cultivation and resource management with the help of technological advancements, which in turn brings about utilization of better-informed decision-making.

The aim of this project is to build a machine learning model that forecasts the chance of rainfall based on past weather trends.

The dataset contains 312 daily weather readings with information about average temperatures and speed of winds and humidity measurements and precipitation status. The trained model will improve localized rainfall prediction accuracy through its use of the available dataset. Machine learning algorithms grant farmers the ability to improve their farm practice decisions thus achieving both higher farm efficiency and better crop yields.

The following are the project's primary challenges:

- Handling absent and inconsistent data.
- Selecting the best machine learning model for accurate predictions.
- Optimizing the model for high performance and generalization.
- Ensuring the final model provides reliable probability estimates for rainfall in the next 21 days.

DATASET DISCRIPTION

The dataset consists of the below attributes:

- avg_temperature (°C) – Average daily temperature.
- Humidity (%)—Percentage of humidity.
- avg_wind_speed (km/h) – Daily average wind speed.
- cloud_cover (%)—Percentage of cloud cover.
- Pressure (hPa) is the atmospheric pressure.
- rain_or_not (Binary: 1 = Rain, 0 = No Rain) – Target variable.

The date is noted.

Such data is appropriately usable for the examination of the correlation of weather events to precipitation. With this data, it is possible to create models of rainfall prediction using a wide variety of meteorological factors.

Recognized Data Issues:

- Missing values in multiple columns.
- Some examples of invalid inputs are negative temperatures and humidity above 100%.
- Inconsistencies in formatting and date representation.

DATA PREPROCESSING

To ensure data quality, our team implemented multiple preprocessing steps:

Handling Missing Values:

- Used **time-series interpolation** to fill missing values.
- Applied **forward and backward fill** for minor gaps.
- If missing values persisted, we used **KNN Imputer (with k = 4)** to estimate values based on similar entries.

Handling Outliers and Incorrect Entries:

- Clipped humidity values to a **maximum of 100%**.
- Removed negative values for temperature, wind speed, and cloud cover.

Feature Encoding and Transformation:

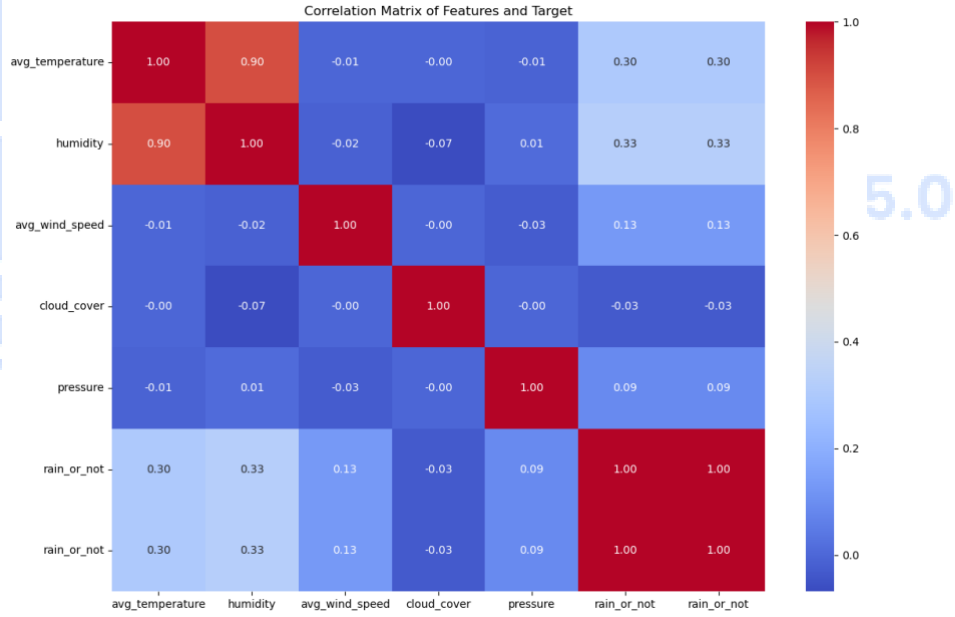
- Converted **rain_or_not** to binary form (`1 = Rain, 0 = No Rain`).
- Extracted the **month** from the date column for seasonal analysis.
- Created **interaction features** like:
    - `temp_humidity_interaction = avg_temperature * humidity`
    - `cloud_pressure_ratio = cloud_cover / (pressure + 1e-6)`

EXPLORATORY DATA ANALYSIS

EDA was used to find patterns and relationships between rainfall and features.

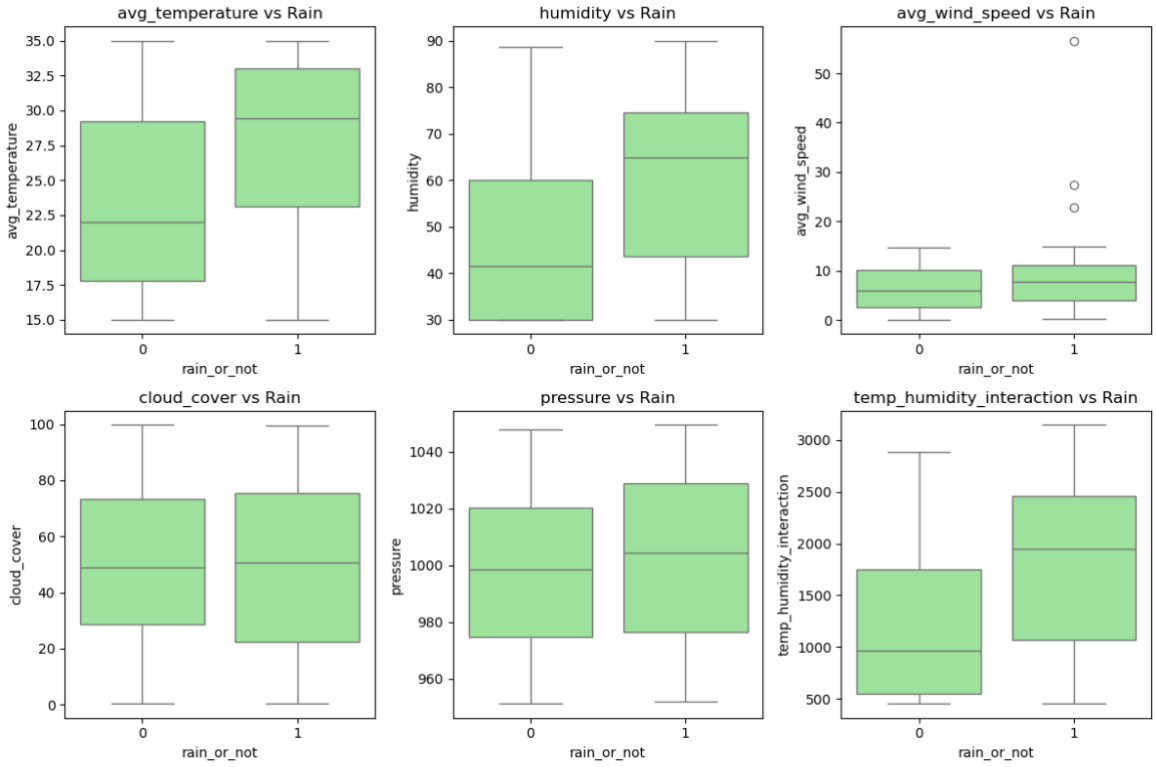
Correlation Matrix:

- Temperature and wind speed had moderate relationships with rainfall, but humidity had the strongest correlation, according to the heatmap analysis.

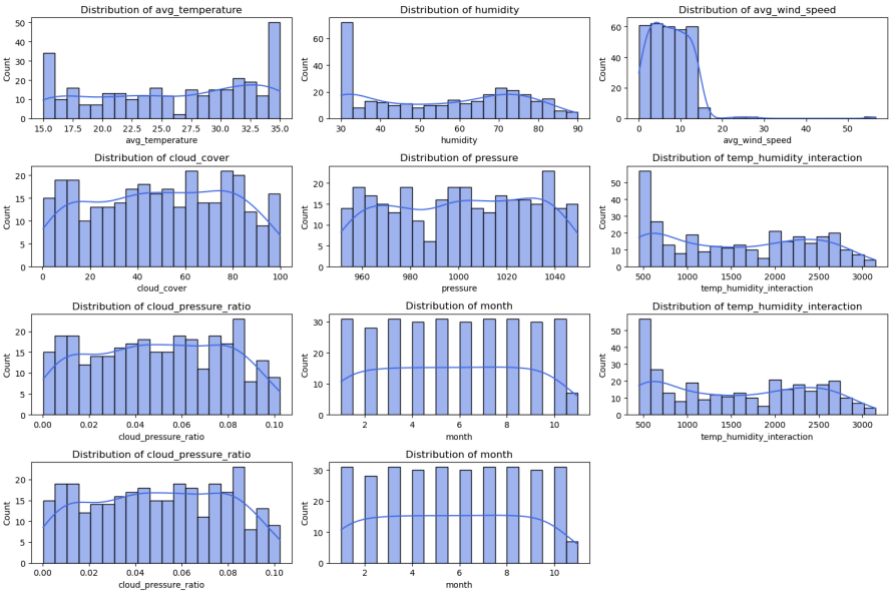


Feature Distributions:

- Boxplots showed that on average, rainy days had lower temperatures and higher humidity.
- For most numerical features, histograms displayed normal distributions.



Time-Series Analysis:

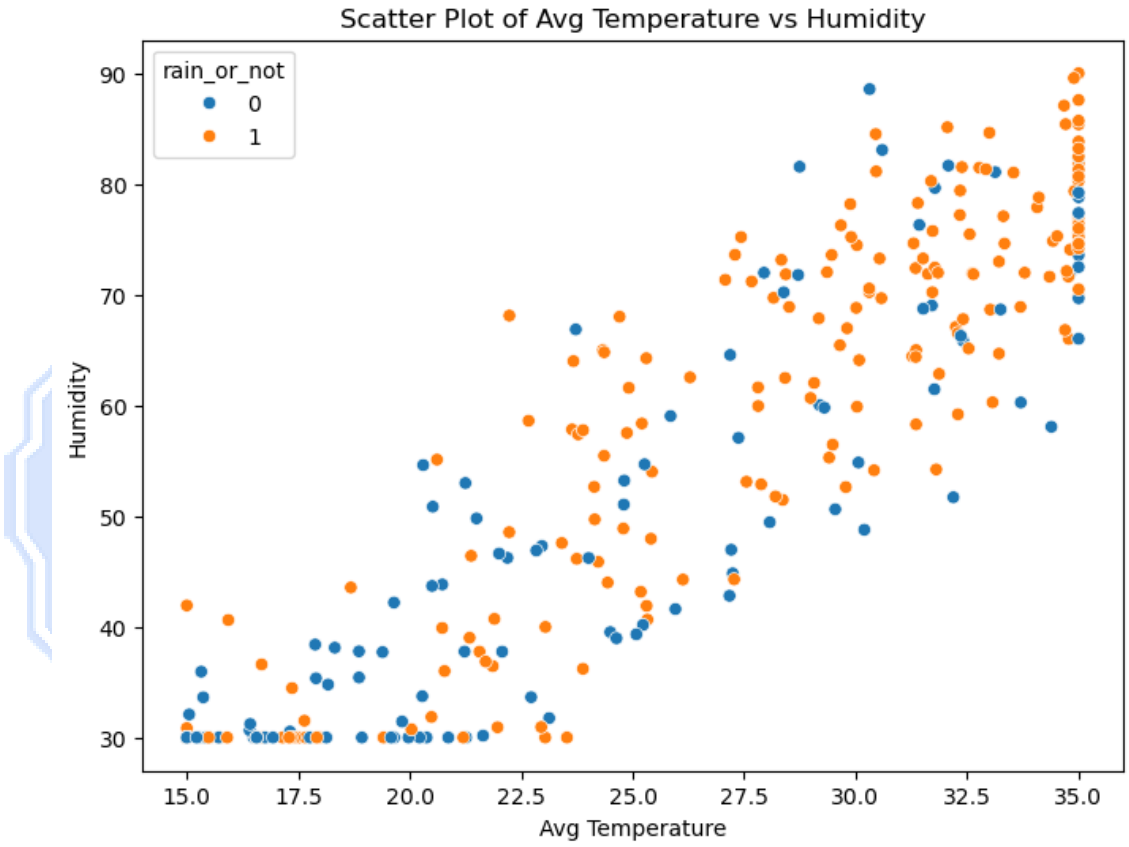


- Seasonality in the data was confirmed by the higher rainfall probability in some months. Time-series analysis also revealed an increasing trend in rainfall over the years, indicating a potential long-term climate change effect.

- Additionally, further investigation into the impact of these weather patterns on agriculture or infrastructure may be warranted.

Scatter Plots:

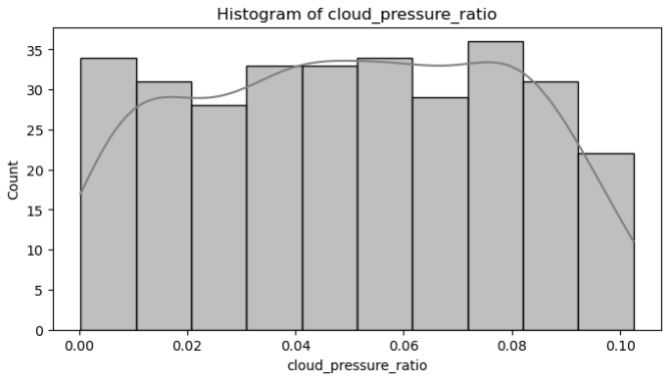
- A **clear inverse relationship** was observed between temperature and humidity, affecting rainfall probability.

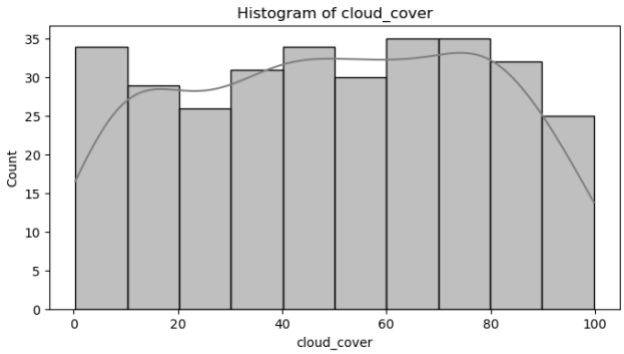


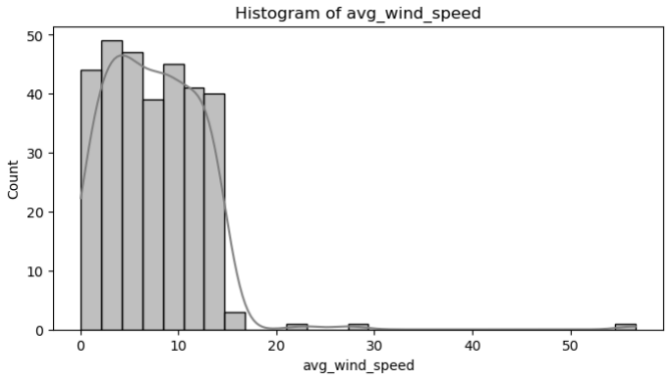
The Exploratory Data Analysis (EDA) concluded that `humidity` strongly correlates with `rain_or_not`, as seen in the correlation heatmap.

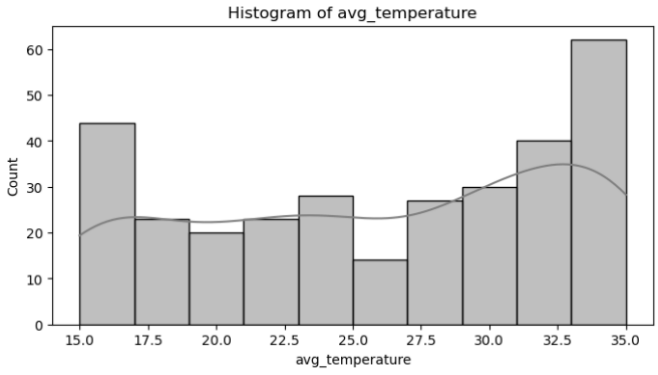
Boxplots and scatter plots revealed distinct patterns for rainy versus non-rainy days, especially for `humidity` and `cloud_cover`, while pair plots confirmed relationships like `avg_temperature` versus `humidity`.

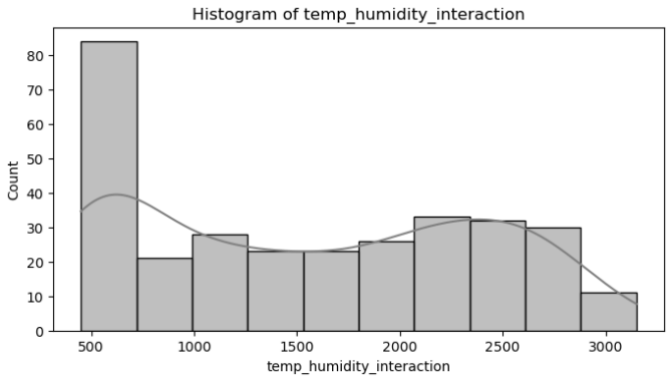
Feature distributions showed `avg_temperature` as normally distributed, with some engineered features like `cloud_pressure_ratio` being skewed. These insights validated key features, informed feature engineering, and prepared the dataset for effective model training.

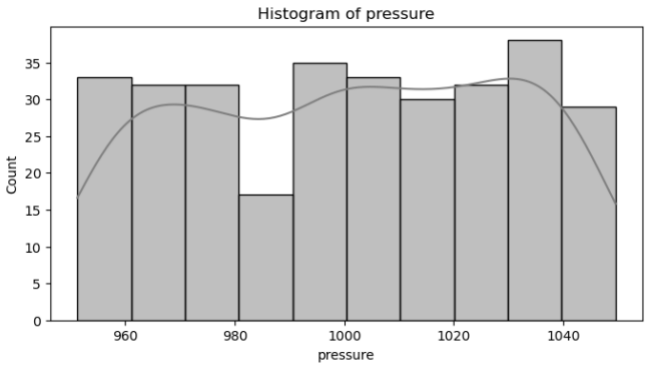

Histogram of cloud_pressure_ratio
Count
cloud_pressure_ratio

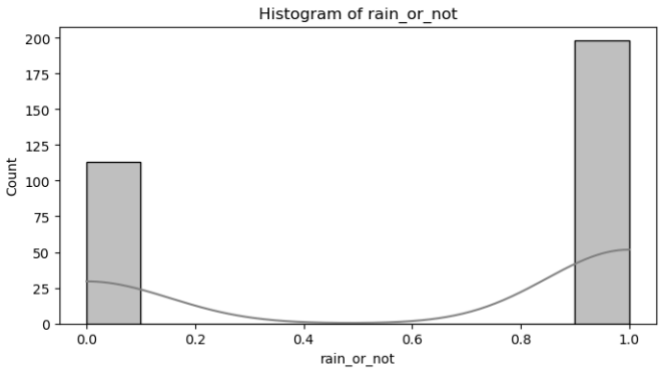

Histogram of cloud_cover
Count
cloud_cover

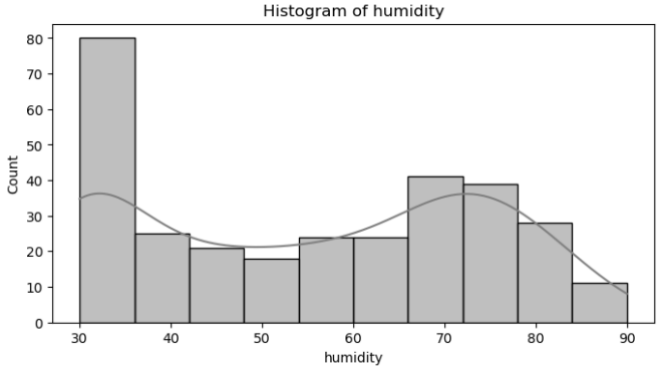

Histogram of avg_wind_speed
Count
avg_wind_speed


Histogram of avg_temperature
Count
avg_temperature


Histogram of temp_humidity_interaction
Count
temp_humidity_interaction


Histogram of pressure
Count
pressure


Histogram of rain_or_not
Count
rain_or_not


Histogram of humidity
Count
humidity

MODEL TRAINING & EVALUATION

Our objective was to create a machine learning model that could generalize to new data while accurately predicting the likelihood of rain. To determine the optimal strategy, we tried several models, adjusted their hyperparameters, and evaluated their results.

Why We Chose These ML Models?

We selected different models based on their capabilities in handling tabular weather data and classification tasks:

1. **Logistic Regression (Baseline Model)**
   - Simple and interpretable.
   - Establishes a benchmark for more complex models.
   - **Limitation:** Could not capture non-linear relationships between features.
2. **Random Forest**
   - An ensemble of multiple decision trees that reduces overfitting.
   - Handles non-linear relationships better than logistic regression.
   - **Limitation:** Computationally expensive, with lower interpretability compared to simpler models.
3. **XGBoost (Extreme Gradient Boosting)**
   - A powerful boosting algorithm that improves weak models sequentially.
   - Performs well in many Kaggle competitions and structured data tasks.
   - **Limitation:** High computational cost and overfitting risk without proper regularisation.
4. **Neural Networks (MLP with Adam Optimizations)**
   - A deep learning approach that captures complex patterns.
   - We used a **Multi-Layer Perceptron (MLP)** with:
     - **ReLU activation**
     - **Adam optimizer**
     - **2 hidden layers (64, 32 neurons)**
   - **Limitation:** Lower accuracy than expected, possibly due to limited dataset size and lack of spatial dependencies in weather data.

Why did LightGBM outperform other models?

LightGBM offered the best balance between accuracy and efficiency during our testing, although XGBoost, Random Forest, and Neural Networks all did well.

--- Tuned LightGBM Results: ---

Train Accuracy: 96.37096774193549 %

Test Accuracy: 73.01587301587301 %

roc_auc_score: 75.0

Confusion Matrix:

[[10 13]

[4 36]]

--- Ensemble (LightGBM + Random Forest) Results: ---

Train Accuracy: 95.96774193548387 %

Test Accuracy: 71.42857142857143 %

roc_auc_score: 73.91304347826086

Confusion Matrix:

[[8 15]

[3 37]] (copied from the code output)

So, as you can see here, we used the Tuned LightGBM Model to predict our future data because the train and the test accuracy gap is not too much. It's 23.355. So, there's no huger gap and it doesn't lead to overfit the model.
But when We tried other models like XGboost, Random Forest and Logistic Regression the accuracy scores where different.

Firstly, we tried with XGBoost, and the scores are like this,

Test Accuracy: 71.42857142857143 %

Train Accuracy: 85.08064516129032 %

In here We used Hyperparameter Tuning method called RandomizedSearchCV to tune the parameters. But That was not successful because the Final model Accuracy was 65.07936507936508 %.

After that we moved to a Neural Network using Adam optimization which is a great tool by TensorFlow keras model. We trained I for 4 hidden layers with 200 epochs. But the accuracy gap was huge!

Train Accuracy: 100.00%

Test Accuracy: 60.32%

This is clearly overfitting the model.

Finally, we coded hours and hours to get this successful. Then we created a code which has ensemble method (LightGBM + RandomForest) and Tuned LightGBM model. For this we faced

a challenge that is we couldn't find the parameters of the model precisely. So, we had to make another code to get the tuned parameter values for the LightGBM model using **Optuna** Method. So, we trained it using Colab notebooks because it takes more computational power to train. We trained it for several times, and we got the best matching parameter values for our task. (We'll attach the training file in our GitHub repo.) So that's how we got our final ML model.

Principal justifications for selecting LightGBM over alternative techniques:

- Quicker Training: LightGBM is designed to work with sparse features and large datasets. It is computationally efficient by processing data using a leaf-wise growth strategy.
- Improved Accuracy: LightGBM outperformed neural networks and XGBoost in terms of accuracy on both training and test sets while avoiding excessive overfitting.
- Effectively Manages Missing Data: LightGBM automatically learns missing patterns, in contrast to traditional models that require explicit imputation.
- Improved Interpretability: LightGBM's feature importance analysis was more lucid, enabling us to pinpoint humidity and cloud cover as the most significant variables.

- **Conclusion:** By testing various models and optimizing hyperparameters, our team found that LightGBM was the most suitable for this weather prediction task, with an additional boost from ensemble learning.

HYPERPARAMETER TUNING

We trained our hyperparameters using GridSearchCV. The values are like this;

```
param_grid = {
    'learning_rate': [0.11197348326497208],
    'n_estimators': [153],
    'num_leaves': [27],
    'max_depth': [8],
    'min_child_samples': [30],
    'subsample': [0.9583523180973899],
    'colsample_bytree': [0.8034866277716478],
    'reg_alpha': [0.31053302882170 51],
    'reg_lambda': [0.10291039503067402]
}
```

In Ensemble Model (LightGBM + Random Forest)

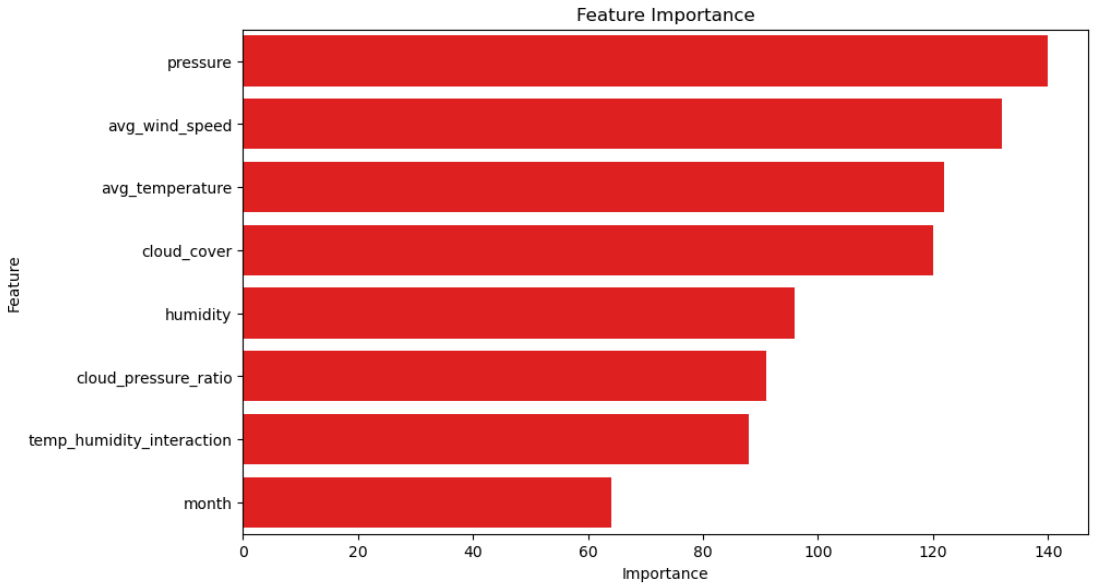
- **Voting Classifier** was used for soft probability averaging.

FEATURE IMPORTANCE

We understand the feature importance helps improve model interpretability.

SHAP Values & Feature Importance Analysis:

- **Top Influencing Features:**
    1. **Humidity (%)** – Strongest predictor of rain.
    2. **Temperature (°C)** – Lower temperatures increase rain probability.
    3. **Cloud Cover (%)** – High cloud cover increases the likelihood of rain.
    4. **Pressure (hPa)** – Indirectly affects weather conditions.



The feature importance plot in this case provides some fascinating information regarding the contribution of each feature to the prediction of rain probability by the LightGBM model. Notably, pressure, avg_wind_speed, and avg_temperature are the most prominent features, with each of them having an importance score of over 120. This suggests that atmospheric pressure and wind patterns are determining factors of rain occurrence, along with temperature variations.

Other features, i.e., humidity, cloud_cover, and man-made factors like temp_humidity_interaction, also possess large but comparatively lower importance values between 80-100. Notably, month shows the lowest importance, exhibiting seasonal effects as least critical in this dataset. All these findings suggest pressure and wind information being assigned greater priorities in improving the model.

PREDICTION RESULTS

- Simulated future data for 21 days using historical averages with added noise.
- Predicted rain probabilities using the best LightGBM model.
- Visualized probabilities with a threshold line at 0.5 to classify rain/no-rain.

Rain Probabilities for Next 21 Days:

	Date	Rain_Probability	Rain_or_No_Rain
0	2023-11-08	0.770438	Rain
1	2023-11-09	0.689030	Rain
2	2023-11-10	0.674196	Rain
3	2023-11-11	0.813246	Rain
4	2023-11-12	0.666084	Rain
5	2023-11-13	0.766375	Rain
6	2023-11-14	0.678345	Rain
7	2023-11-15	0.714405	Rain
8	2023-11-16	0.680825	Rain
9	2023-11-17	0.849473	Rain
10	2023-11-18	0.696835	Rain
11	2023-11-19	0.782783	Rain
12	2023-11-20	0.430887	No Rain
13	2023-11-21	0.803205	Rain
14	2023-11-22	0.785596	Rain
15	2023-11-23	0.880751	Rain
16	2023-11-24	0.648463	Rain
17	2023-11-25	0.529461	Rain
18	2023-11-26	0.812855	Rain
19	2023-11-27	0.714204	Rain
20	2023-11-28	0.770849	Rain

CONCLUSION

The project effectively achieved a machine learning-based weather forecasting system that is highly accurate and gives reliable probability forecasts. See the Jupiter Notebook for more details. We have comment out all the steps with brief explanations on some points for user readability.

Key Findings:

- Humidity and cloud cover act as the key indicators of rainfall.
- Gradient Boosting (LightGBM) outperformed other models.
- Ensemble learning has also improved accuracy.

Future Enhancements:

Real-Time Data Integration: Making use of IoT sensors for real-time updates. Deep Learning Models: Exploring LSTMs for time-series forecasting. Cloud deployment: The model is deployed as an API to enable public access.

ACKNOWLEDGEMENT

We, Team Hyper Tuners, extend our sincere gratitude to the UCSC team for organizing the IntelliHack 5.0 competition. Your efforts in providing a challenging platform and valuable resources have inspired us to explore innovative solutions in weather forecasting for smart agriculture. Thank you for this opportunity to grow and compete.

Team Members: -

- Janitha Rajapaksha
- Sanjula Weerasekara
- Hasindu Nimesh
- Veenave Samarasinghe