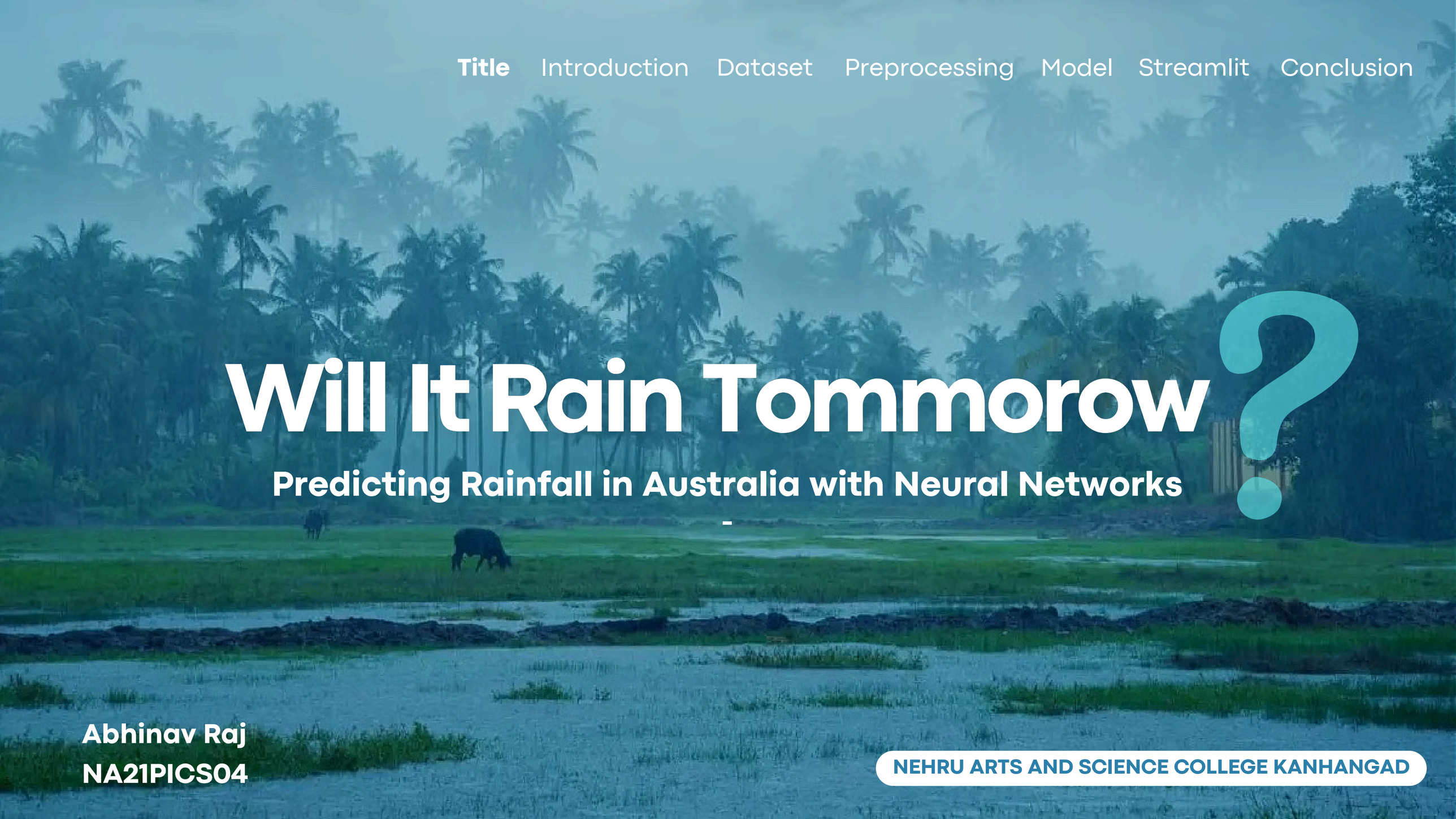


Will It Rain Tommorow ?

Predicting Rainfall in Australia with Neural Networks

-

Abhinav Raj
NA21PICS04

NEHRU ARTS AND SCIENCE COLLEGE KANHANGAD

Rainfall prediction is a critical task in meteorology with significant implications for **agriculture**, **water resource management**, and **disaster preparedness**. Traditional forecasting methods often struggle to handle the complexity and variability of weather patterns. In this case study, we explore a **data-driven approach** to **predict whether it will rain the next day** (**RainTomorrow**) using a **neural network model** trained on historical weather data from Australia. By leveraging modern machine learning techniques and extensive data preprocessing, this project aims to build a reliable and efficient rainfall prediction system that can support real-world decision-making.

Dataset Overview

📍 Source: Australian Bureau of Meteorology

🗓️ Records: ~145,000 daily observations across various locations

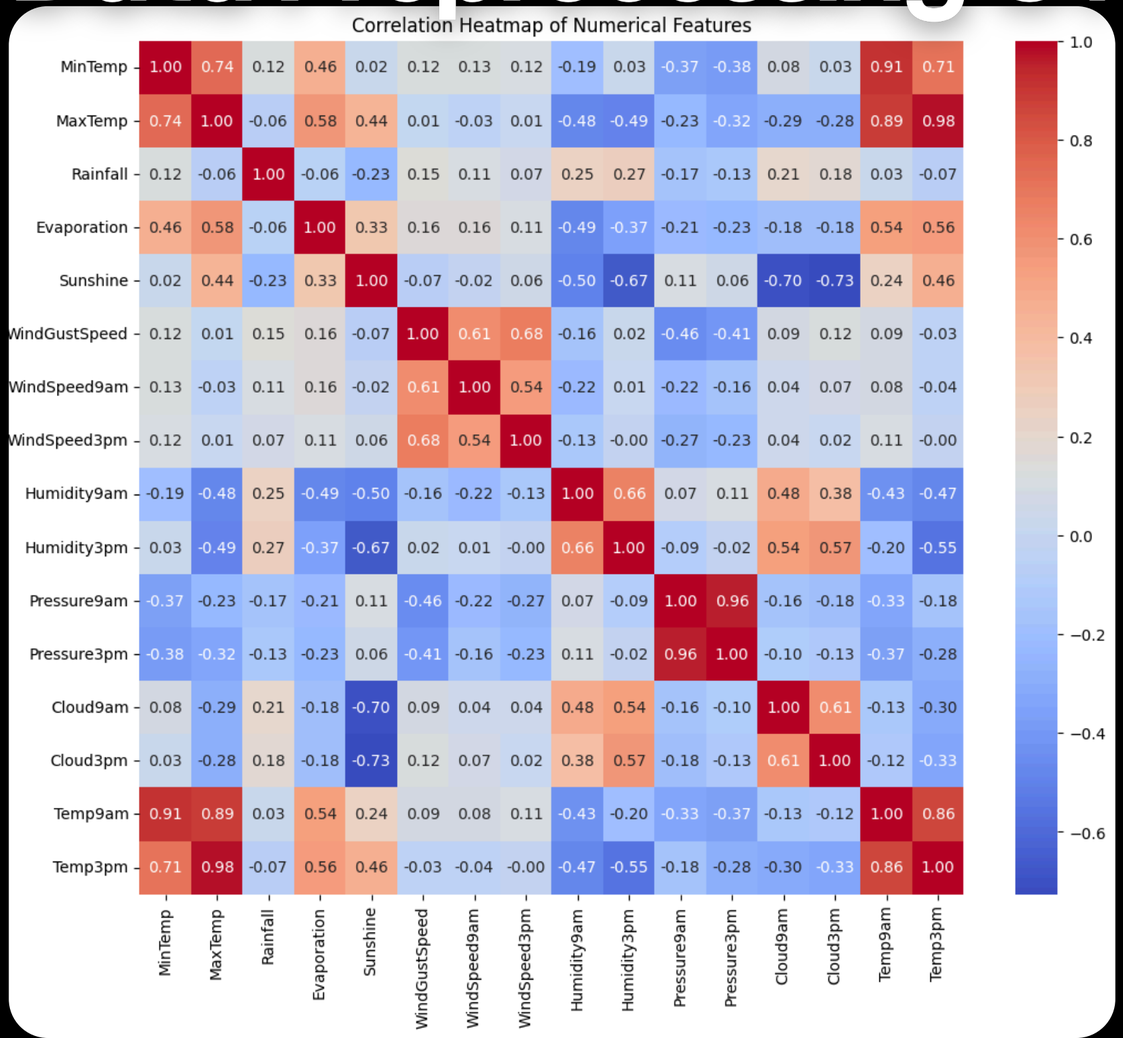
🎯 Target Variable: RainTomorrow (Yes / No)

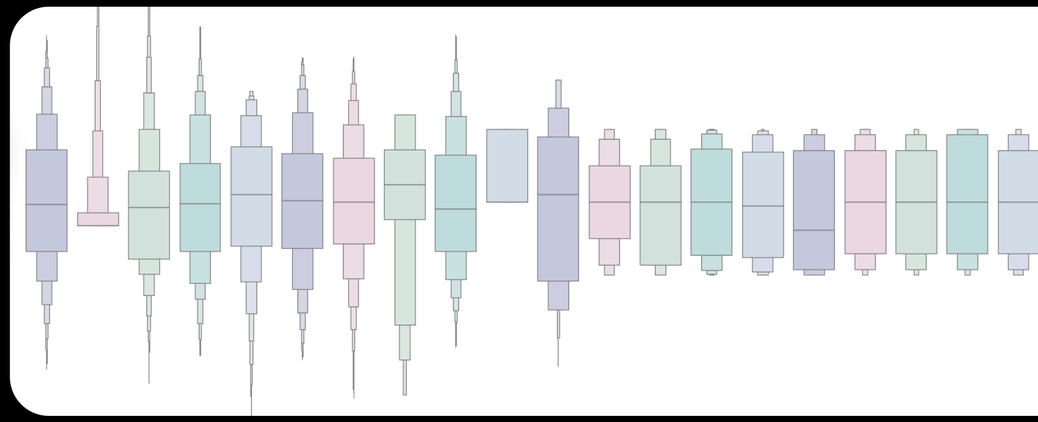
📊 Features Include:

- Temperature (Min, Max)
- Rainfall
- Humidity (9am, 3pm)
- Wind Speed & Direction
- Atmospheric Pressure
- Date & Location

Dataset Link

Data Preprocessing Overview





- **Removed rows** with **missing RainTomorrow**; **encoded as 0/1**
- Balanced classes using **bootstrapped oversampling**
- **Parsed Date → cyclic day, month** (sine/cosine)
- **Imputed missing values** (median for numeric, mode for categorical)
- **Removed outliers using IQR**
- **Encoded**:
  - **Location: label / one-hot**
  - **Wind directions: degree → sine/cosine**
- **Scaled** numeric features (z-score)
- **Created two datasets** (22 & 69 features); **stratified 80:20 split**

Model Architecture & Training Overview

```
model = Sequential()
model.add(Dense(64, kernel_initializer='he_normal', activation='relu', input_dim=22))
model.add(BatchNormalization())

model.add(Dense(32, kernel_initializer='he_normal', activation='relu'))
model.add(BatchNormalization())

model.add(Dense(16, kernel_initializer='he_normal', activation='relu'))
model.add(BatchNormalization())
model.add(Dropout(0.1))
```

⚙️ Training Configuration

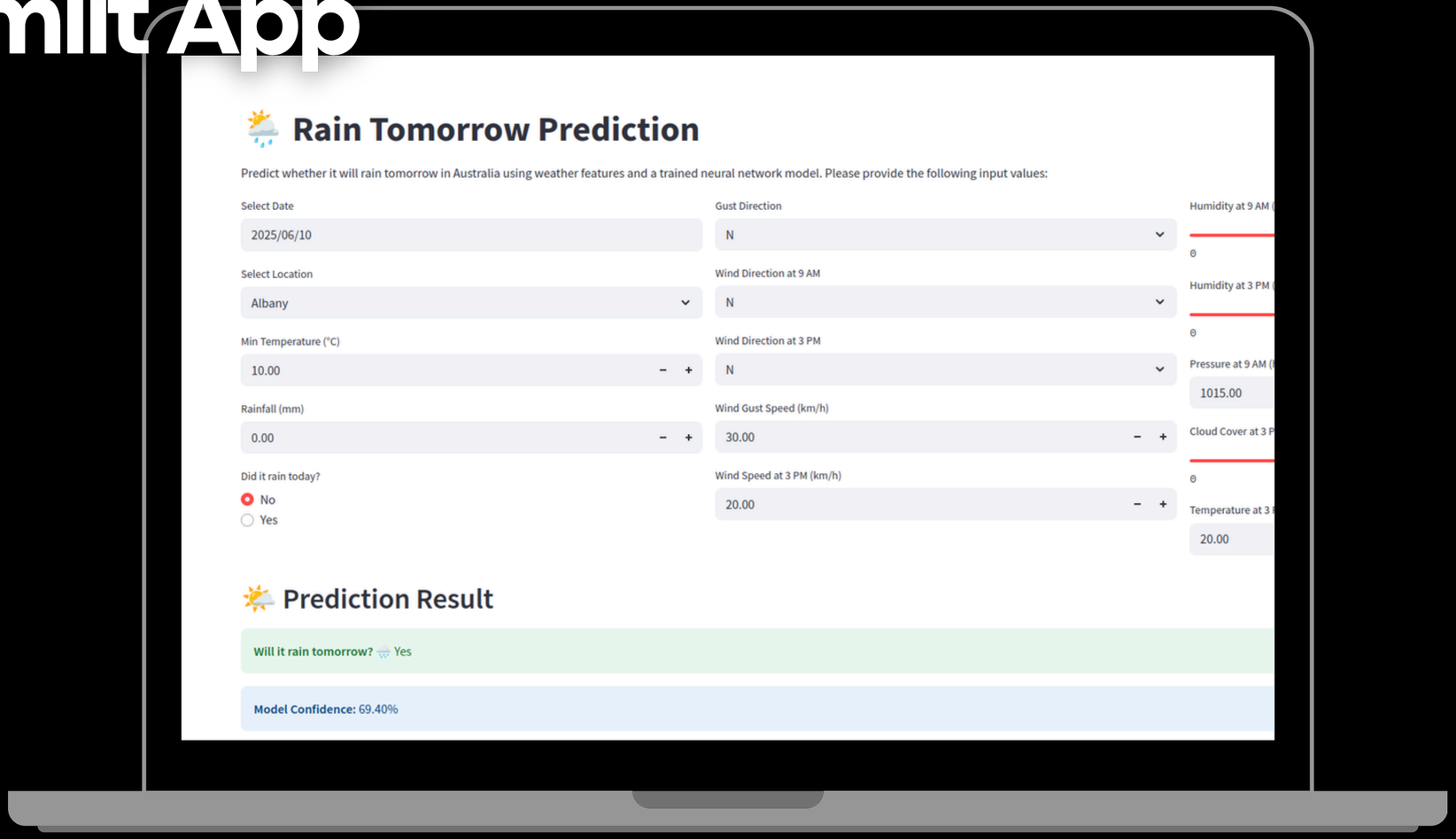
- Optimizer: Adam (LR = 0.00009)
- Loss: Binary Crossentropy
- Metrics: Accuracy, AUC
- Validation Split: 20%
- Early Stopping: Patience = 20, Δ = 0.001
- Max Epochs: 150
- Batch Size: 32

Used BatchNormalization and Dropout to improve generalization.

Trained both versions: Label Encoded & One-Hot Encoded.

Visualized training metrics (accuracy & AUC) to monitor performance.

Streamlit App



A simple web app built with Streamlit lets users input weather data and instantly predicts whether it will rain tomorrow using the trained ANN model. It features a clean UI, real-time inference, and visual feedback.

Conclusion

✅ Developed a Neural Network model to predict RainTomorrow using Australian weather data.

🔍 Applied robust preprocessing:

- Handled missing values
- Normalized numerical features
- Categorical and cyclical encoding

⚖️ Tackled class imbalance using balanced class weights

🧪 Compared encoding methods:

- Label Encoding: Moderate performance
- One-Hot Encoding: Best performance with 82% accuracy

📈 Achieved high recall (≈89%) for predicting rainy days — crucial for real-world utility

💡 Highlighted the importance of feature representation and data balancing in classification tasks

Thank You