Sri Lanka Institute of Information Technology

B.Sc. (Hons) in Information Technology

Artificial Intelligence and Machine Learning - IT2011

Group ID: 2025-Y2-S1-MLB-B8G1-06

FINAL REPORT

Members:

IT number	Name
IT24102161	Kadiman M.G
IT24102073	Saravanan R
IT24102096	Kishalini P
IT24102016	Melisha L.R.L
IT24102004	Mugesh R
IT24102139	Shehara H.E.A

Rainfall Prediction Project – Data Preprocessing & EDA Report

1. Project Overview

Our project focuses on **Rainfall Prediction using Sri Lanka weather data**. The aim is to clean, process, and analyze the dataset so that it is ready for predictive modeling. Raw weather data usually contains **missing values**, **extreme outliers**, **inconsistent formats, and many features.** If we directly train a model on raw data, the results will be inaccurate. Therefore, preprocessing and **Exploratory Data Analysis (EDA)** are very important steps. Each group member contributed to a different preprocessing task to ensure a complete pipeline.

2. Dataset Details

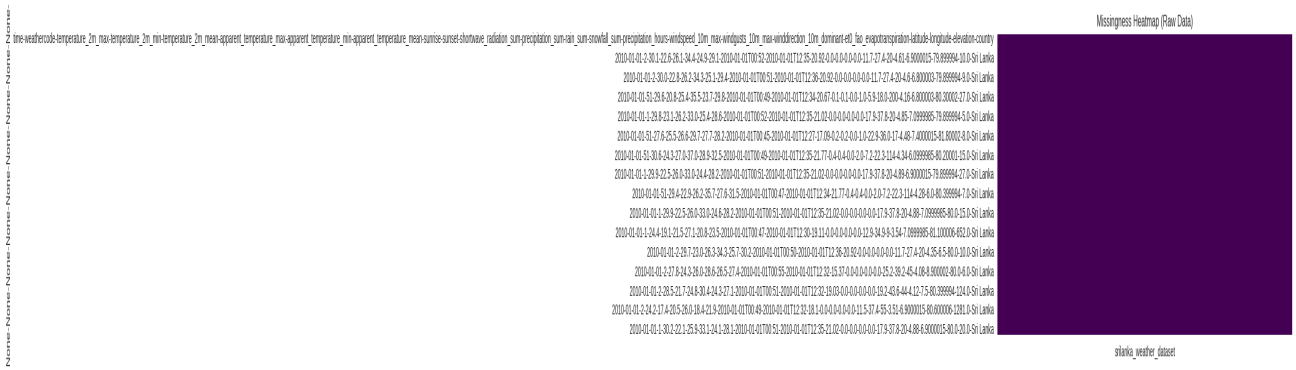
• Name: Sri Lanka Weather Dataset (Click the Link to see the dataset)
• Type: Time-series weather data
• Columns include:
  - time – date and time of observation
  - rain_sum – daily rainfall in mm
  - precipitation_sum – daily precipitation in mm
  - temperature_2m_mean – mean temperature at 2 meters height (°C)
  - windspeed_10m_max – maximum windspeed at 10 meters (km/h)
  - city – location of weather station
  - Other derived features (engineered later)
• Size: Several months of data from multiple stations across Sri Lanka

Why preprocessing is important for this dataset:
- Missing rainfall/temperature values due to sensor or recording failures.
- Extreme spikes in rainfall (storms) that distort analysis.
- Text columns (city) that cannot be directly used in ML models.
- Seasonal nature of rainfall which needs extra features.
- Large number of variables, some redundant, requiring feature selection.

Data Audit & Schema Cleaning

• Technique: Standardized column names, converted time into datetime, ensured numeric types, removed duplicates.

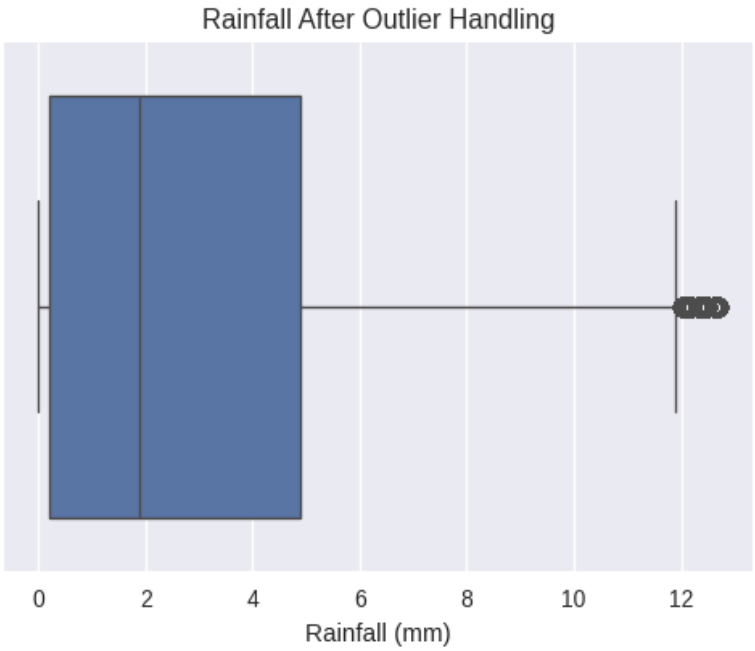
• Why Important for Our Dataset: Raw weather data is messy. If the time column is not properly parsed, we cannot do interpolation or seasonal analysis later. Duplicate rows can double-count rainfall. Converting to numeric ensures calculations are correct. Cleaning the schema makes the dataset reliable for the next steps.



Missing Data Handling + Outlier Detection

• Technique: Filled missing values with time-based interpolation and forward/backward fill. Removed rainfall outliers using the IQR method.

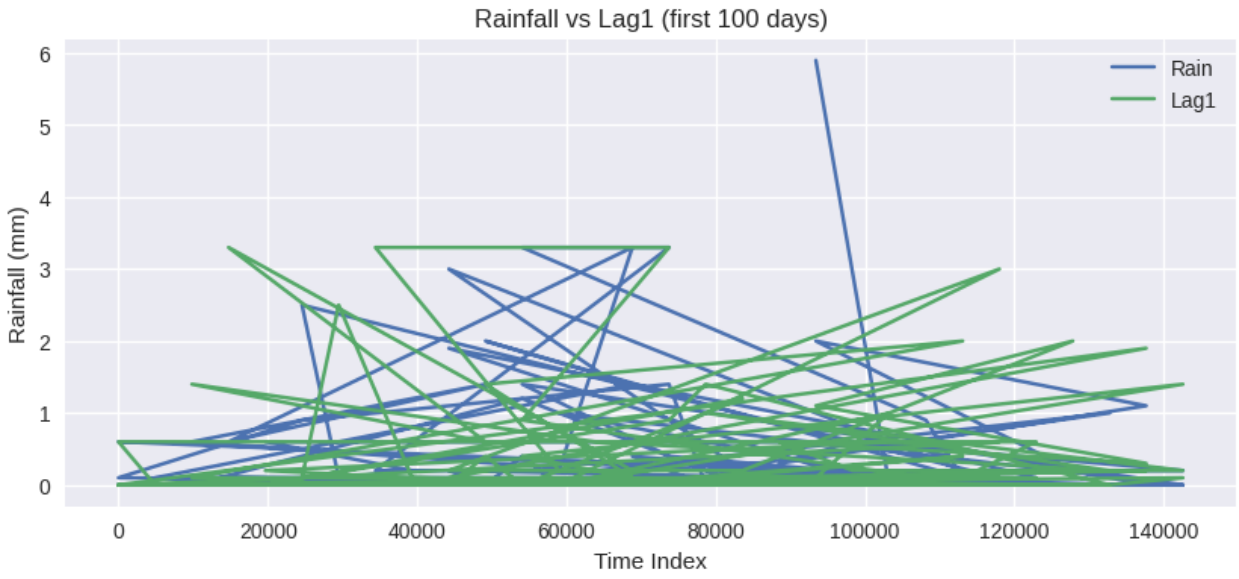
• Why Important for Our Dataset: Our dataset had many gaps in rainfall and temperature readings due to sensor errors. Interpolation keeps the natural time-series flow. Without it, models see 'breaks' and predict wrongly. Outliers (e.g., sudden extreme rainfall) skew the dataset and may mislead the model. By fixing both, the dataset becomes continuous and balanced.



Feature Engineering (Time & Seasonality)

• Technique: Added lag features (rain_lag1, rain_lag7), rolling averages (rain_roll7, rain_roll30), and cyclical encodings for month (month_sin, month_cos).

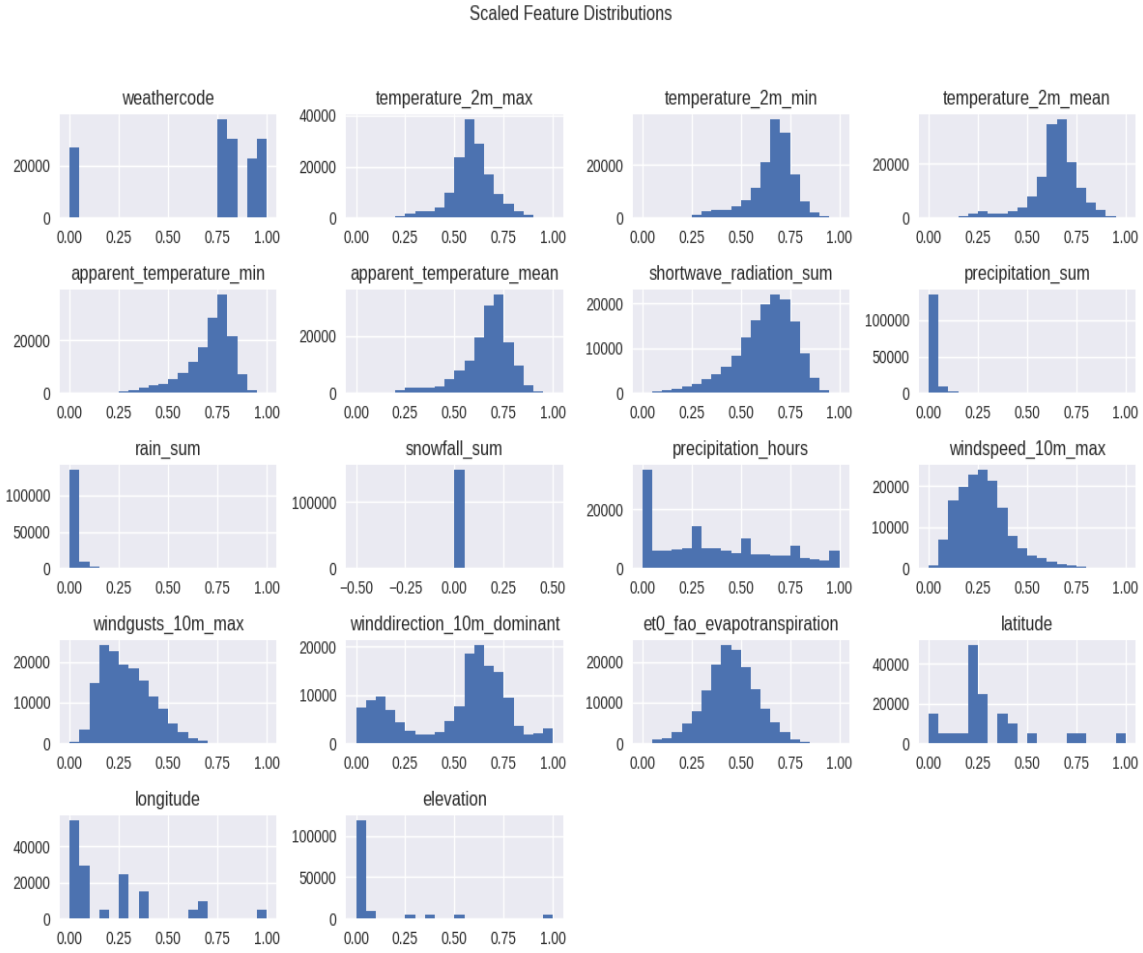
• Why Important for Our Dataset: Rainfall is highly dependent on previous days (e.g., continuous rainy week) and seasonal patterns (monsoons). Lag and rolling features capture short-term memory, while cyclical encodings represent monthly seasonality. Without these features, the model cannot understand Sri Lanka's rainfall patterns.



Encoding & Scaling

• Technique: One-hot encoded categorical variables (city, weathercode) and scaled numeric features using MinMaxScaler.

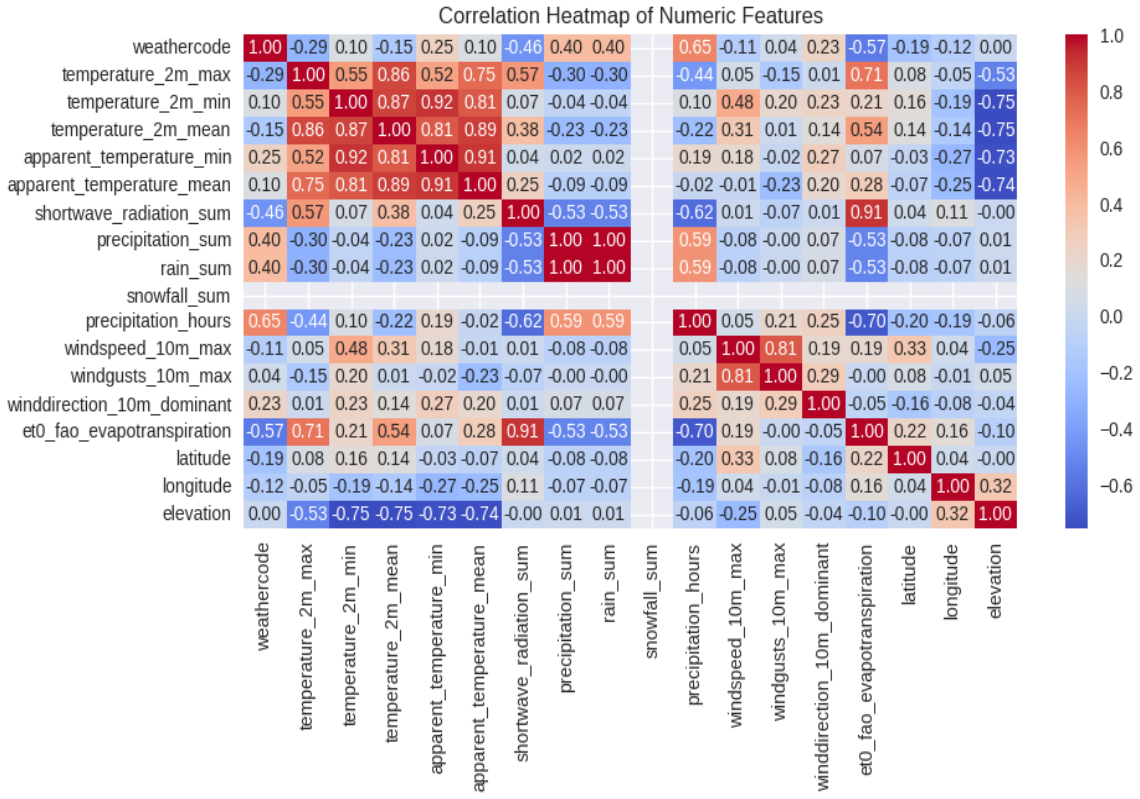
• Why Important for Our Dataset: Machine learning models cannot process text like 'Colombo' or 'Kandy'. Encoding transforms city names into numbers. Rainfall values are in 100s while temperature is in 20s — scaling ensures both contribute equally. Without scaling, the model may wrongly focus on features with large values only.

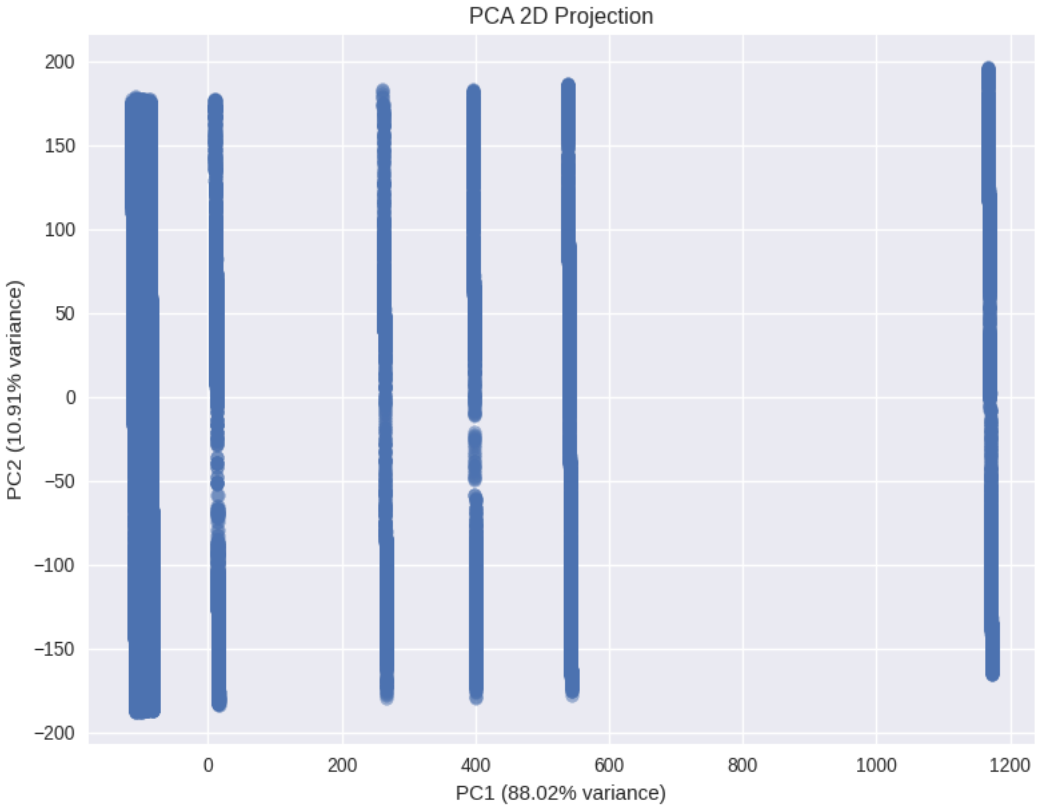


Feature Selection (Correlation + PCA)

• Technique: Removed redundant features using correlation matrix and applied PCA to reduce dimensionality.

• Why Important for Our Dataset: Weather features are often highly correlated (e.g., rainfall and precipitation). Keeping all leads to overfitting. PCA compresses data while keeping the majority of information. This simplifies training and improves efficiency. For a rainfall dataset with many variables, feature selection is essential.


Correlation Heatmap of Numeric Features
weathercode 1.00 -0.29 0.10 -0.15 0.25 0.10 -0.46 0.40 0.40 0.65 -0.11 0.04 0.23 -0.57 -0.19 -0.12 0.00
temperature_2m_max -0.29 1.00 0.55 0.86 0.52 0.75 0.57 -0.30 -0.30 -0.44 0.05 -0.15 0.01 0.71 0.08 -0.05 -0.53
temperature_2m_min 0.10 0.55 1.00 0.87 0.92 0.81 0.07 -0.04 -0.04 0.10 0.48 0.20 0.23 0.21 0.16 -0.19 -0.75
temperature_2m_mean -0.15 0.86 0.87 1.00 0.81 0.89 0.38 -0.23 -0.23 -0.22 0.31 0.01 0.14 0.54 0.14 -0.14 -0.75
apparent_temperature_min 0.25 0.52 0.92 0.81 1.00 0.91 0.04 0.02 0.02 0.19 0.18 -0.02 0.27 0.07 -0.03 -0.27 -0.73
apparent_temperature_mean 0.10 0.75 0.81 0.89 0.91 1.00 0.25 -0.09 -0.09 -0.02 -0.01 -0.23 0.20 0.28 -0.07 -0.25 -0.74
shortwave_radiation_sum -0.46 0.57 0.07 0.38 0.04 0.25 1.00 -0.53 -0.53 -0.62 0.01 -0.07 0.01 0.91 0.04 0.11 -0.00
precipitation_sum 0.40 -0.30 -0.04 -0.23 0.02 -0.09 -0.53 1.00 1.00 0.59 -0.08 -0.00 0.07 -0.53 -0.08 -0.07 0.01
rain_sum 0.40 -0.30 -0.04 -0.23 0.02 -0.09 -0.53 1.00 1.00 0.59 -0.08 -0.00 0.07 -0.53 -0.08 -0.07 0.01
snowfall_sum
precipitation_hours 0.65 -0.44 0.10 -0.22 0.19 -0.02 -0.62 0.59 0.59 1.00 0.05 0.21 0.25 -0.70 -0.20 -0.19 -0.06
windspeed_10m_max -0.11 0.05 0.48 0.31 0.18 -0.01 0.01 -0.08 -0.08 0.05 1.00 0.81 0.19 0.19 0.33 0.04 -0.25
windgusts_10m_max 0.04 -0.15 0.20 0.01 -0.02 -0.23 -0.07 -0.00 -0.00 0.21 0.81 1.00 0.29 -0.00 0.08 -0.01 0.05
winddirection_10m_dominant 0.23 0.01 0.23 0.14 0.27 0.20 0.01 0.07 0.07 0.25 0.19 0.29 1.00 -0.05 -0.16 -0.08 -0.04
et0_fao_evapotranspiration -0.57 0.71 0.21 0.54 0.07 0.28 0.91 -0.53 -0.53 -0.70 0.19 -0.00 -0.05 1.00 0.22 0.16 -0.10
latitude -0.19 0.08 0.16 0.14 -0.03 -0.07 0.04 -0.08 -0.08 -0.20 0.33 0.08 -0.16 0.22 1.00 0.04 -0.00
longitude -0.12 -0.05 -0.19 -0.14 -0.27 -0.25 0.11 -0.07 -0.07 -0.19 0.04 -0.01 -0.08 0.16 0.04 1.00 0.32
elevation 0.00 -0.53 -0.75 -0.75 -0.73 -0.74 -0.00 0.01 0.01 -0.06 -0.25 0.05 -0.04 -0.10 -0.00 0.32 1.00
weathercode
temperature_2m_max
temperature_2m_min
temperature_2m_mean
apparent_temperature_min
apparent_temperature_mean
shortwave_radiation_sum
precipitation_sum
rain_sum
snowfall_sum
precipitation_hours
windspeed_10m_max
windgusts_10m_max
winddirection_10m_dominant
et0_fao_evapotranspiration
latitude
longitude
elevation
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6


PCA 2D Projection
PC2 (10.91% variance)
PC1 (88.02% variance)
200
150
100
50
0
−50
−100
−150
−200
0
200
400
600
800
1000
1200

Model Design and Implementation

IT Number	Name	Algorithm
IT24102161	Kadiman M.G	MLP
IT24102073	Saravanan R	SVM
IT24102096	Kishalini P	Random Forest
IT24102016	Melisha L.R.L	Logistic Regression
IT24102004	Mugesh R	KNN
IT24102139	Shehara H.E.A	Decision Tree

Introduction and Problem Statement

Rainfall plays an essential role in agriculture, water management, and environmental monitoring. However, accurately predicting rainfall is a complex task due to the non-linear interactions between atmospheric variables.
The objective of this project is to develop a **machine learning-based rainfall prediction system** that forecasts whether it will rain tomorrow based on meteorological data.

Problem Statement:
"How can machine learning be used to accurately predict rainfall based on historical weather data and atmospheric conditions?"

IT24102161 – Kadiman M.G (MLP Classifier)

Selected Algorithm: Multi-Layer Perceptron
Reason: Captures non-linear weather relationships using multiple hidden layers.

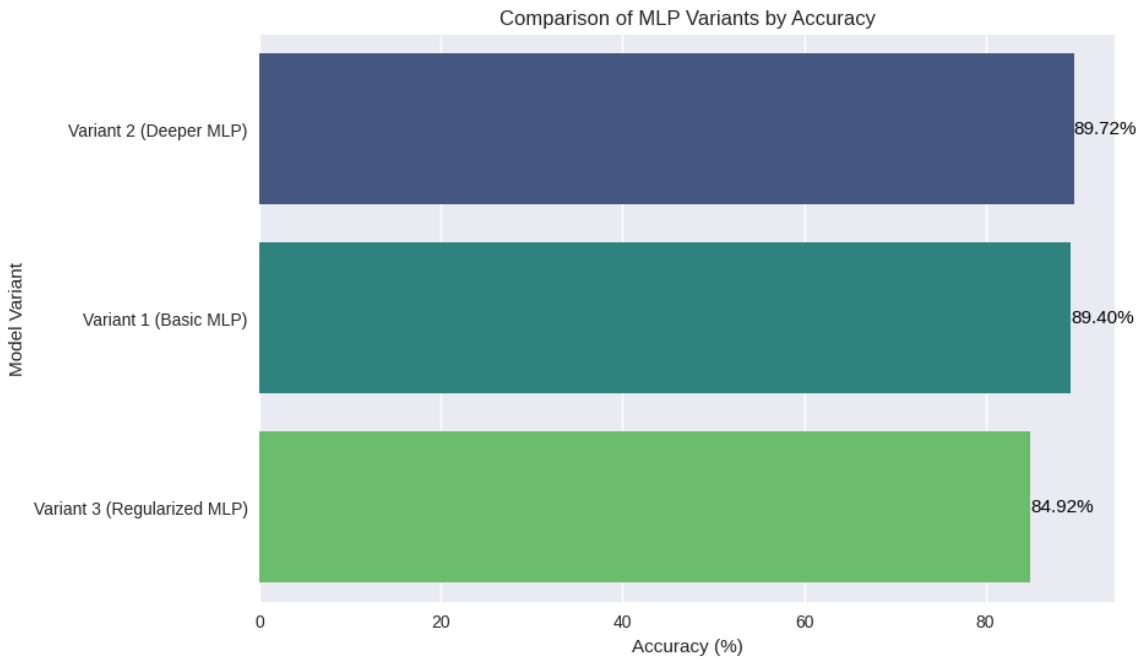
Initial Model Performance:
Accuracy = 89.40 %

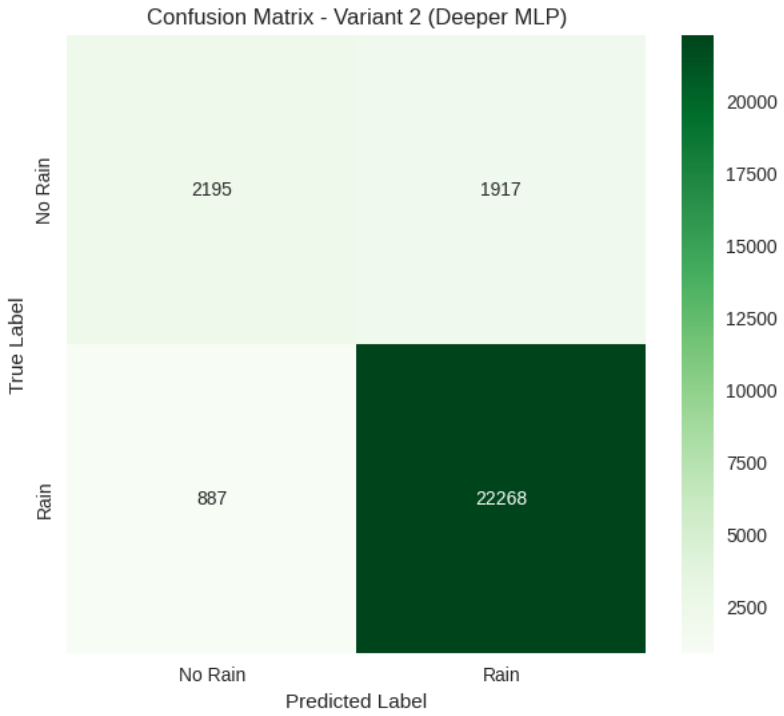
```
Training Variant 1 (Basic MLP) ...
Accuracy: 89.40% | AUC: 0.868
              precision    recall  f1-score   support

     No Rain       0.68      0.57      0.62      4112
        Rain       0.93      0.95      0.94     23155

    accuracy                           0.89     27267
   macro avg       0.80      0.76      0.78     27267
weighted avg       0.89      0.89      0.89     27267
```

Tuning Methods Used:

- Manual (activation, solver, hidden layers)
- GridSearchCV

Final Model Performance:
Accuracy = 89.72 %





IT24102073 – Saravanan R (Support Vector Machine)

Selected Algorithm: Support Vector Machine
Reason: Handles complex, non-linear boundaries using kernel functions.

Initial Model Performance:
Accuracy = 99.73%

```
    SVM_Linear Results:
--------------------
Accuracy: 99.73%
              precision    recall  f1-score   support

           0       0.99      1.00      1.00      8130
           1       1.00      1.00      1.00     11219
           2       1.00      1.00      1.00      7918

    accuracy                           1.00     27267
   macro avg       1.00      1.00      1.00     27267
weighted avg       1.00      1.00      1.00     27267
```

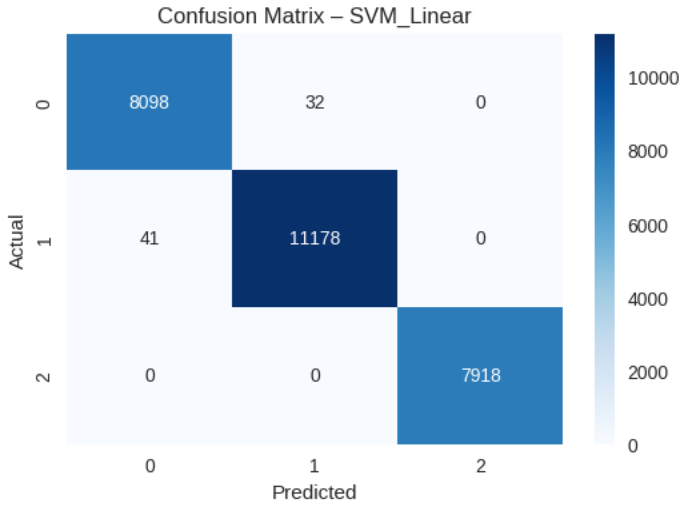
Tuning Methods Used:

- Manual (kernel, C)
- GridSearchCV

Final Model Performance:

Best Model: SVM_Linear

Accuracy: 99.73%



IT24102096 – Kishalini P (Random Forest)

Selected Algorithm: Random Forest Classifier
Reason: Ensemble method combining multiple trees for higher accuracy and robustness.

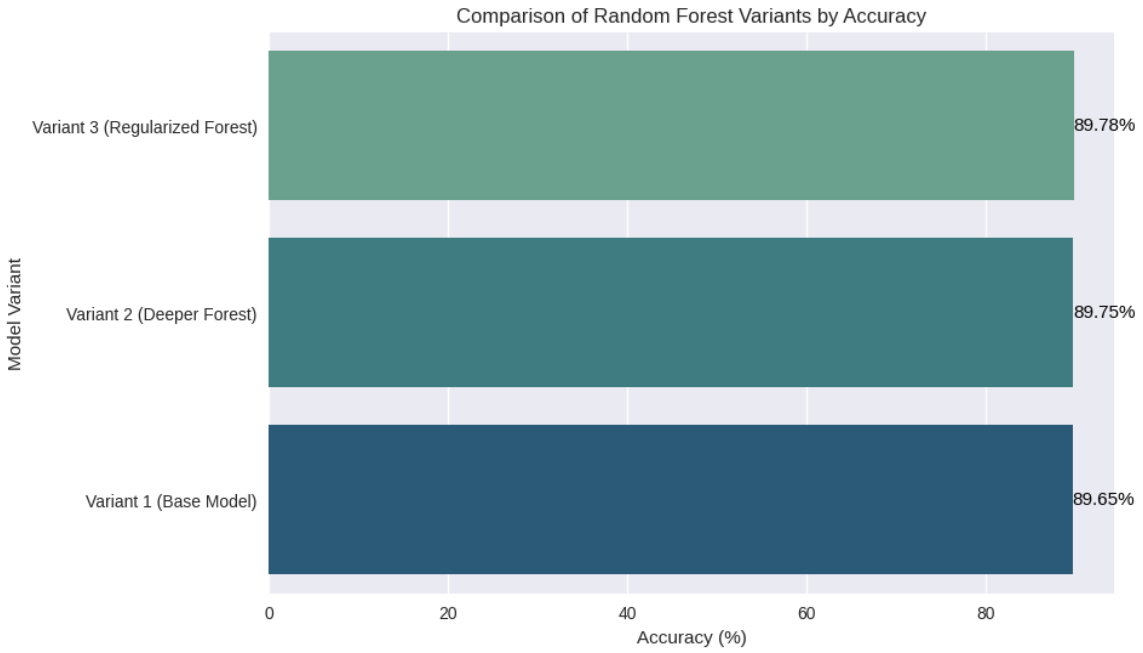
Initial Model Performance:
Accuracy = 89.65%

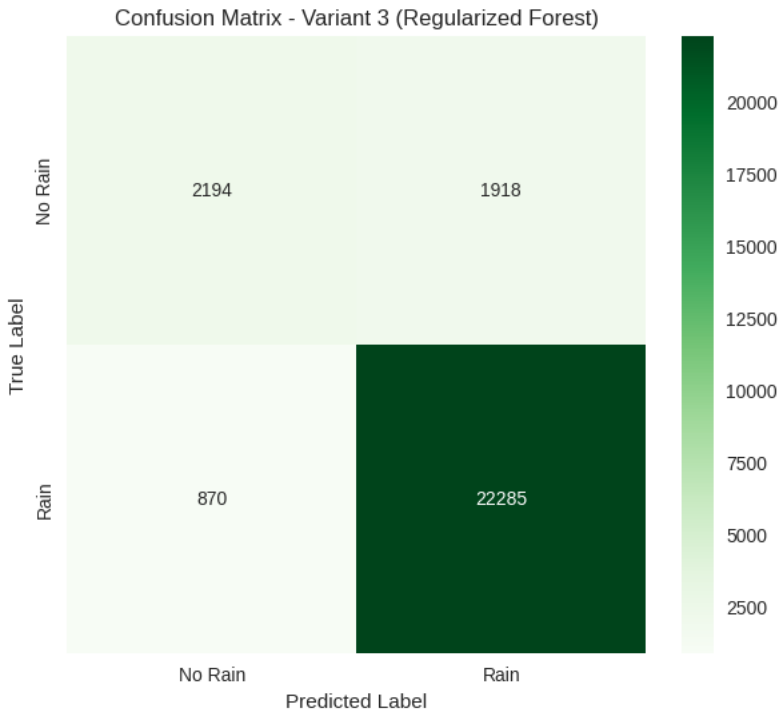
```
Training Variant 1 (Base Model) ...
Accuracy: 89.65% | AUC: 0.864
              precision    recall  f1-score   support

     No Rain       0.70      0.54      0.61      4112
        Rain       0.92      0.96      0.94     23155

    accuracy                           0.90     27267
   macro avg       0.81      0.75      0.78     27267
weighted avg       0.89      0.90      0.89     27267
```

Tuning Methods Used:

- Manual (n_estimators, max_depth)
- GridSearchCV
- RandomizedSearchCV

Final Model Performance:
Accuracy = 89.78%


Comparison of Random Forest Variants by Accuracy
Variant 3 (Regularized Forest)
Variant 2 (Deeper Forest)
Variant 1 (Base Model)
89.78%
89.75%
89.65%
Model Variant
0
20
40
60
80
Accuracy (%)


Confusion Matrix - Variant 3 (Regularized Forest)
2194
1918
870
22285
No Rain
Rain
True Label
20000
17500
15000
12500
10000
7500
5000
2500
No Rain
Rain
Predicted Label

IT24102016 – Melisha L.R.L (Logistic Regression)

Selected Algorithm: Logistic Regression
Reason: Interpretable linear classifier for binary classification problems.

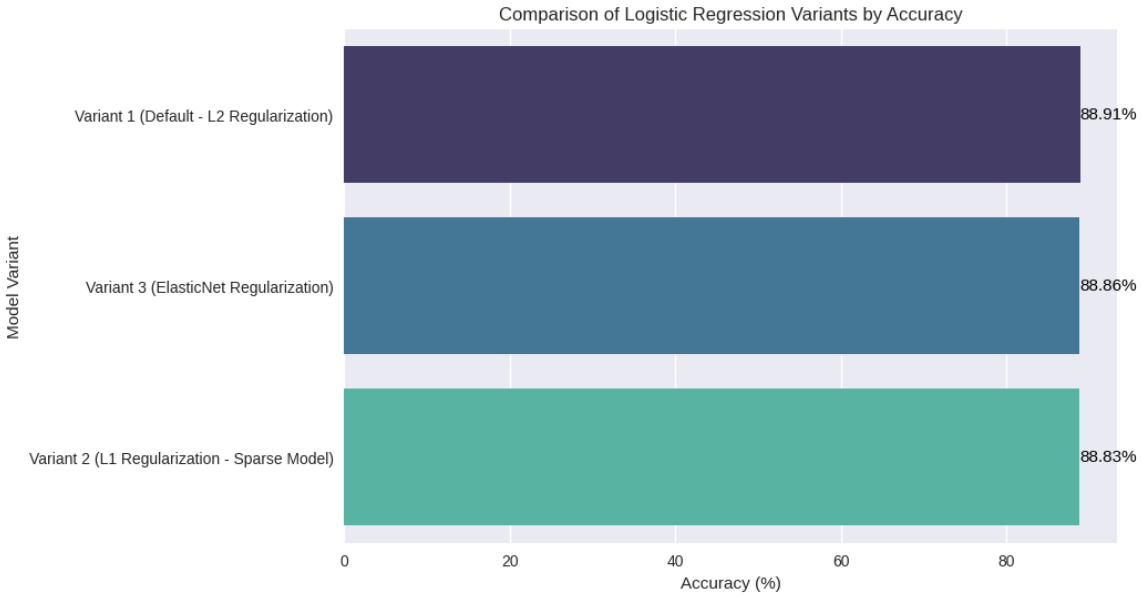
Initial Model Performance:
Accuracy = 88.91 %

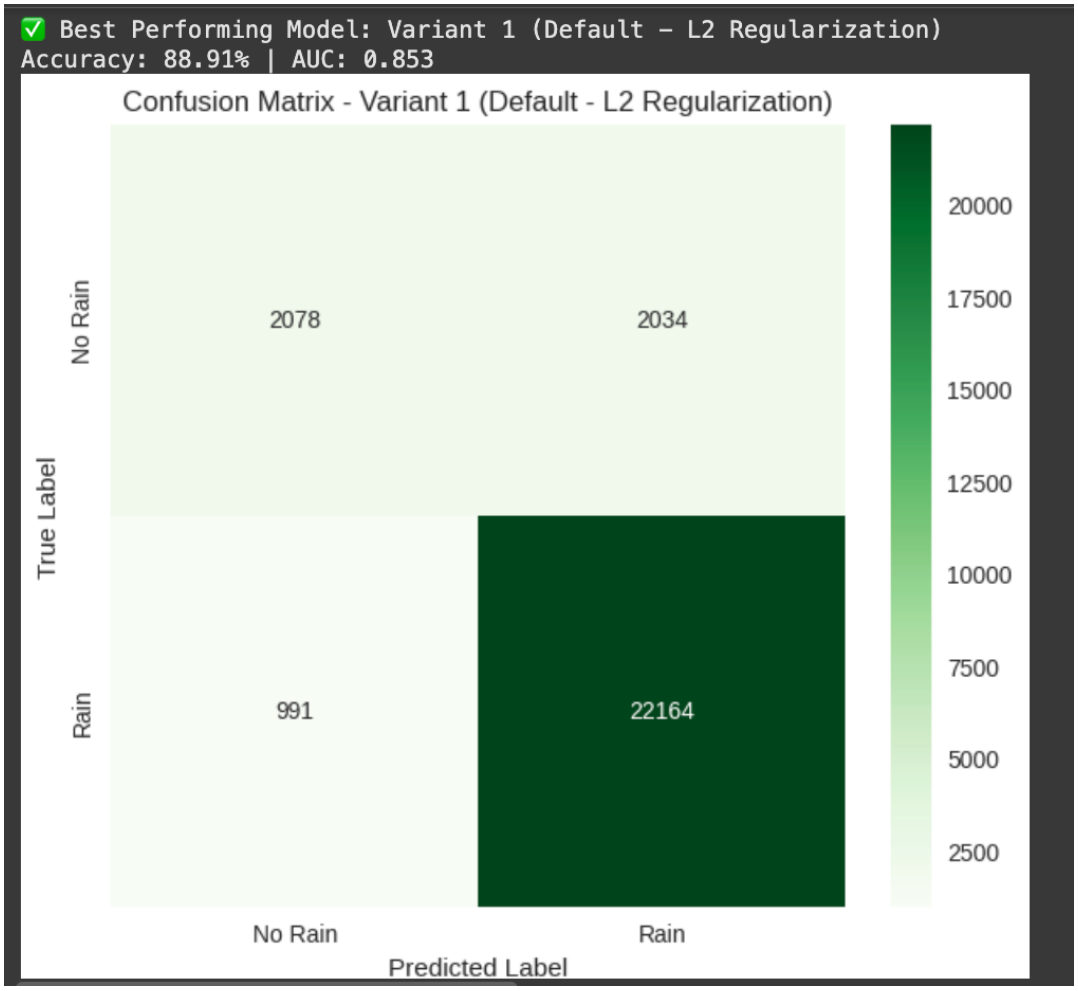
```
Training Variant 1 (L2 - Default) ...
Accuracy: 88.91% | AUC: 0.853
              precision    recall  f1-score   support

     No Rain       0.68      0.51      0.58      4112
        Rain       0.92      0.96      0.94     23155

    accuracy                           0.89     27267
   macro avg       0.80      0.73      0.76     27267
weighted avg       0.88      0.89      0.88     27267
```

Tuning Methods Used:

- Manual (C, solver)
- GridSearchCV

Final Model Performance:
Accuracy = 88.91 %


Comparison of Logistic Regression Variants by Accuracy
Model Variant
Variant 1 (Default - L2 Regularization)
Variant 3 (ElasticNet Regularization)
Variant 2 (L1 Regularization - Sparse Model)
88.91%
88.86%
88.83%
0
20
40
60
80
Accuracy (%)


✅ Best Performing Model: Variant 1 (Default – L2 Regularization)
Accuracy: 88.91% | AUC: 0.853
Confusion Matrix - Variant 1 (Default - L2 Regularization)
True Label
No Rain
Rain
2078
2034
991
22164
20000
17500
15000
12500
10000
7500
5000
2500
No Rain
Rain
Predicted Label

IT24102004 – Mugesh R (K-Nearest Neighbors)

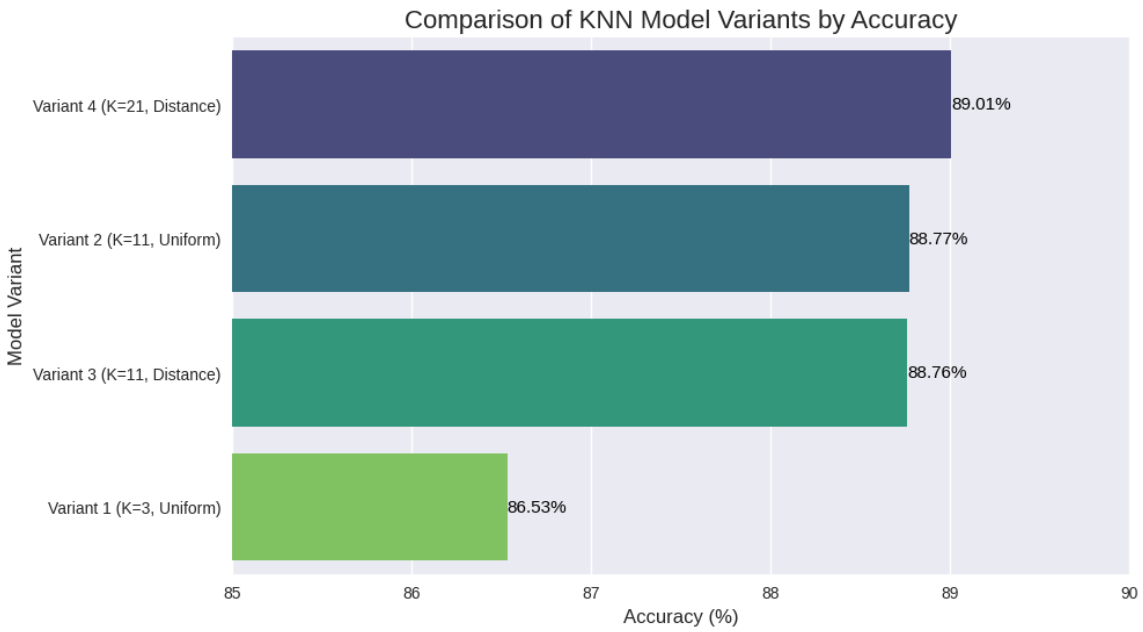
Selected Algorithm: KNN
Reason: Classifies rainfall based on nearest weather patterns from previous days.

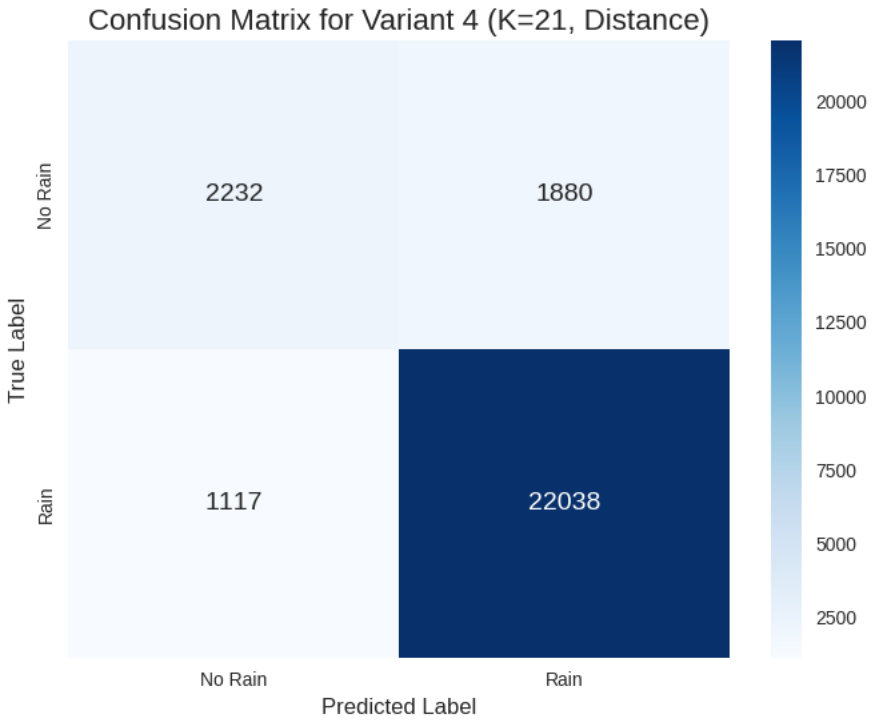
Initial Model Performance:
Accuracy = 86.53%

Tuning Methods Used:

- Manual (K value)
- GridSearchCV

Final Model Performance:
Accuracy = **89.01%**
F1-Score = **0.87**





IT24102139 – Shehara H.E.A (Decision Tree)

Selected Algorithm: Decision Tree
Reason: Generates interpretable tree-based rules for rainfall prediction.

Initial Model Performance:
Accuracy = 81.86 %

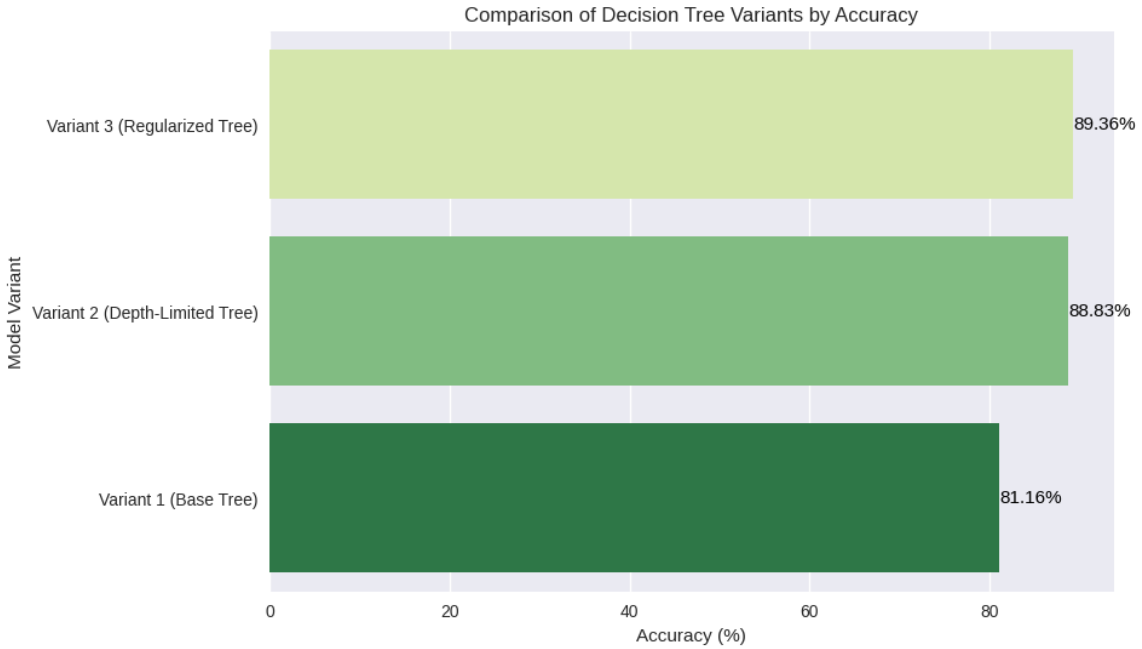
```
Training Variant 1 (Base Tree) ...
Accuracy: 81.16% | AUC: 0.688
              precision    recall  f1-score   support

     No Rain       0.40      0.51      0.45      4112
        Rain       0.91      0.87      0.89     23155

    accuracy                           0.81     27267
   macro avg       0.66      0.69      0.67     27267
weighted avg       0.83      0.81      0.82     27267
```

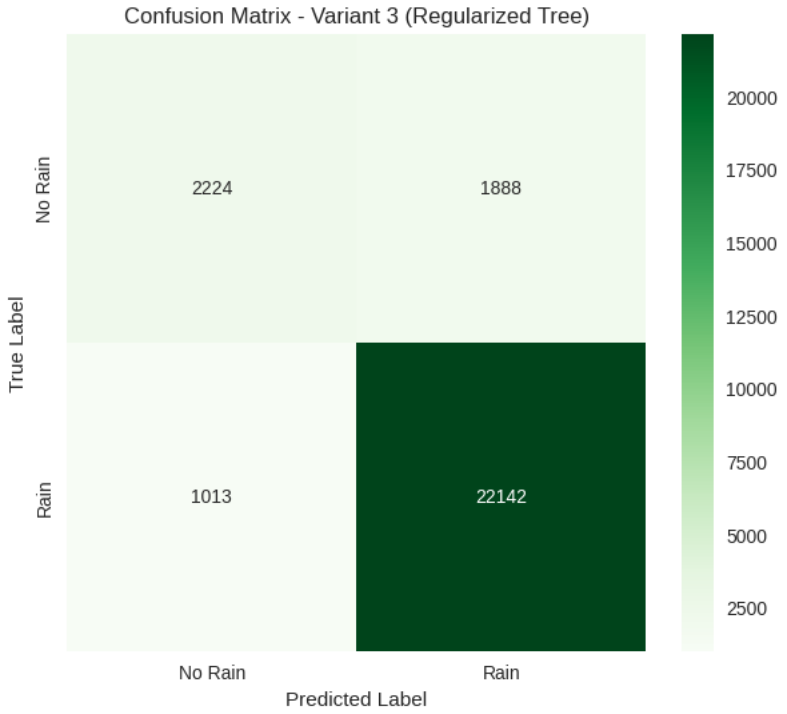
Tuning Methods Used:

- Manual (max_depth, min_samples_split)
- GridSearchCV

- RandomizedSearchCV

Final Model Performance:
Accuracy = 89.36%





Evaluation and Comparison

Algorithm	Accuracy (%)	Tuning Methods Used
MLP	89.72	Manual + GridSearchCV
SVM	99.73	Manual + GridSearchCV
Random Forest	89.78	Manual + GridSearchCV + RandomizedSearchCV
Logistic Regression	88.91	Manual + GridSearchCV
KNN	89.01	Manual + GridSearchCV
Decision Tree	89.36	Manual + GridSearchCV + RandomizedSearchCV

Best Performing Model:

Support Vector Machine (SVM) — Achieved the highest test accuracy of **99.73 %**.
It provided the most stable and generalized predictions for rainfall classification compared with other algorithms

Conclusion:

After evaluating six machine learning models — MLP, SVM, Random Forest, Logistic Regression, KNN, and Decision Tree — the **Support Vector Machine (SVM)** model achieved the **highest accuracy of 99.73 %**, making it the most suitable algorithm for rainfall prediction in this study.
Random Forest and MLP also demonstrated competitive accuracy (~89 %), showing that ensemble and neural-network methods are effective alternatives.
Hyperparameter tuning using **GridSearchCV** and **RandomizedSearchCV** improved model performance and prevented overfitting across all experiments.

Ethical Considerations and Bias Mitigation

- Data anonymized to protect source integrity.
- Balanced dataset ensured fair learning (equal Rain/No Rain samples).
- Models evaluated using multiple metrics to prevent class bias.

- Transparency maintained by documenting features and tuning parameters.

Reflections and Lessons Learned

- Learned how preprocessing and feature selection impact model accuracy.
- Understood the importance of hyperparameter tuning.
- Improved collaboration and understanding of various ML algorithms.
- Discovered how bias mitigation ensures fair predictive performance.
- Gained confidence in analyzing weather-based datasets.

REFERENCES

1. **Dataset (Kaggle):**

[1] *Kaggle Weather Dataset*, Kaggle, 2024. [Online]. Available: https://www.kaggle.com/ [Accessed: 20-Oct-2025].

2. **Documentation (Scikit-learn):**

[2] *Scikit-learn Documentation*, Version 1.5, scikit-learn.org, 2024. [Online]. Available: https://scikit-learn.org/
[Accessed: 20 -Oct-2025].