Australian Weather Prediction Using Machine Learning

1. Objectives

The primary objective of this project is to develop a robust binary classification model to accurately predict whether it will rain on the following day (RainTomorrow) using relevant weather attributes.

1. To build a binary classification model to predict next-day rainfall (RainTomorrow) using historical weather data after proper preprocessing.
2. To compare multiple classifiers (Logistic Regression, Decision Tree, Random Forest, SVM, KNN, Naïve Bayes) using Accuracy, Precision, Recall, and F1-Score to select the best model.

2. Introduction

Weather prediction plays a crucial role in agriculture, transportation, and disaster management. Accurate rainfall prediction helps in planning daily activities and minimizing losses. In this project, we built a machine-learning-based classification system to predict whether it will rain the next day using historical weather data from Australia. The project includes data preprocessing, feature encoding, handling of class imbalance, training multiple classification models, and comparing their performance.

3. Problem Definition and Algorithm

Input: 'Date', 'Location', 'MinTemp', 'MaxTemp', 'Rainfall', 'Evaporation',
'Sunshine', 'WindGustDir', 'WindGustSpeed', 'WindDir9am', 'WindDir3pm',
'WindSpeed9am', 'WindSpeed3pm', 'Humidity9am', 'Humidity3pm',
'Pressure9am', 'Pressure3pm', 'Cloud9am', 'Cloud3pm', 'Temp9am',
'Temp3pm', 'RainToday'

Output: 'RainTomorrow status'

Algorithms Used:

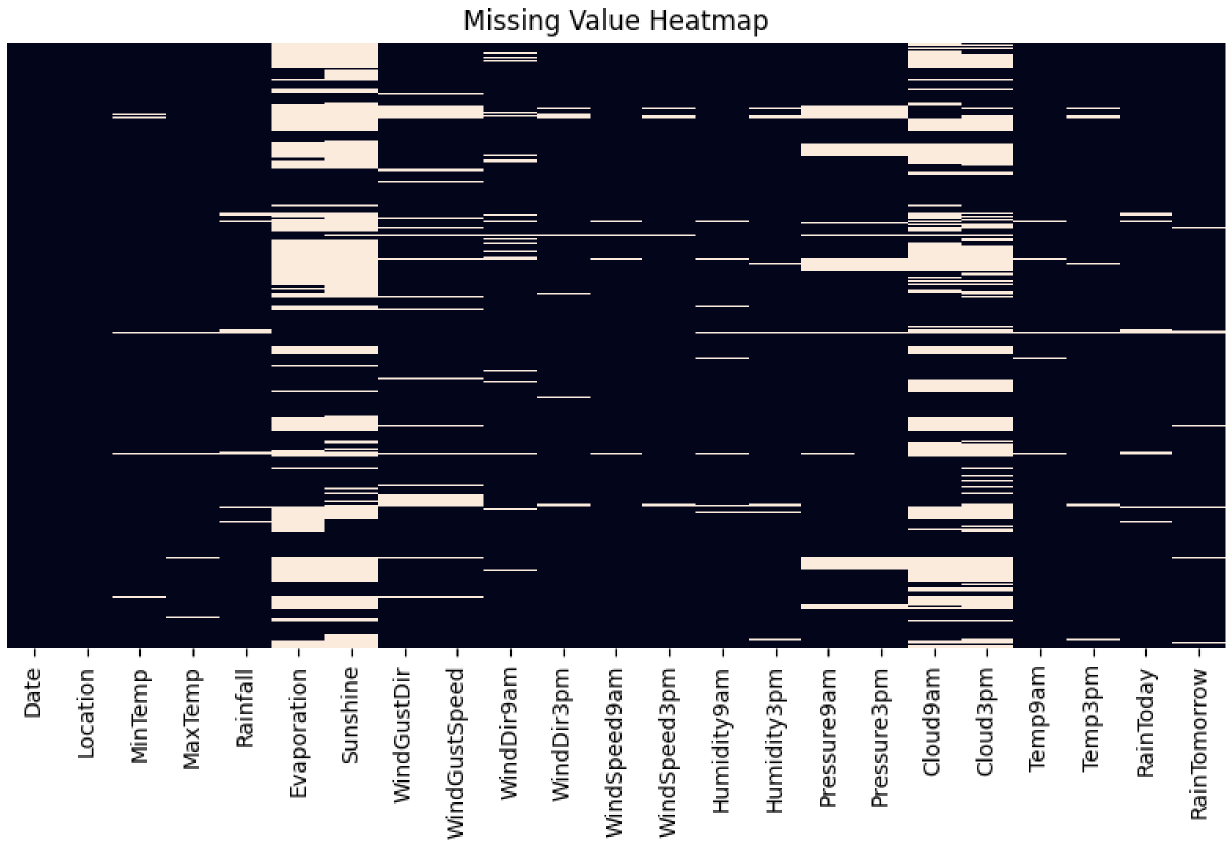
- Logistic Regression
- Support Vector Machine (SVM)
- Decision Tree
- Random Forest
- K-Nearest Neighbors (KNN)
- Gaussian Naïve Bayes

4. Working Steps

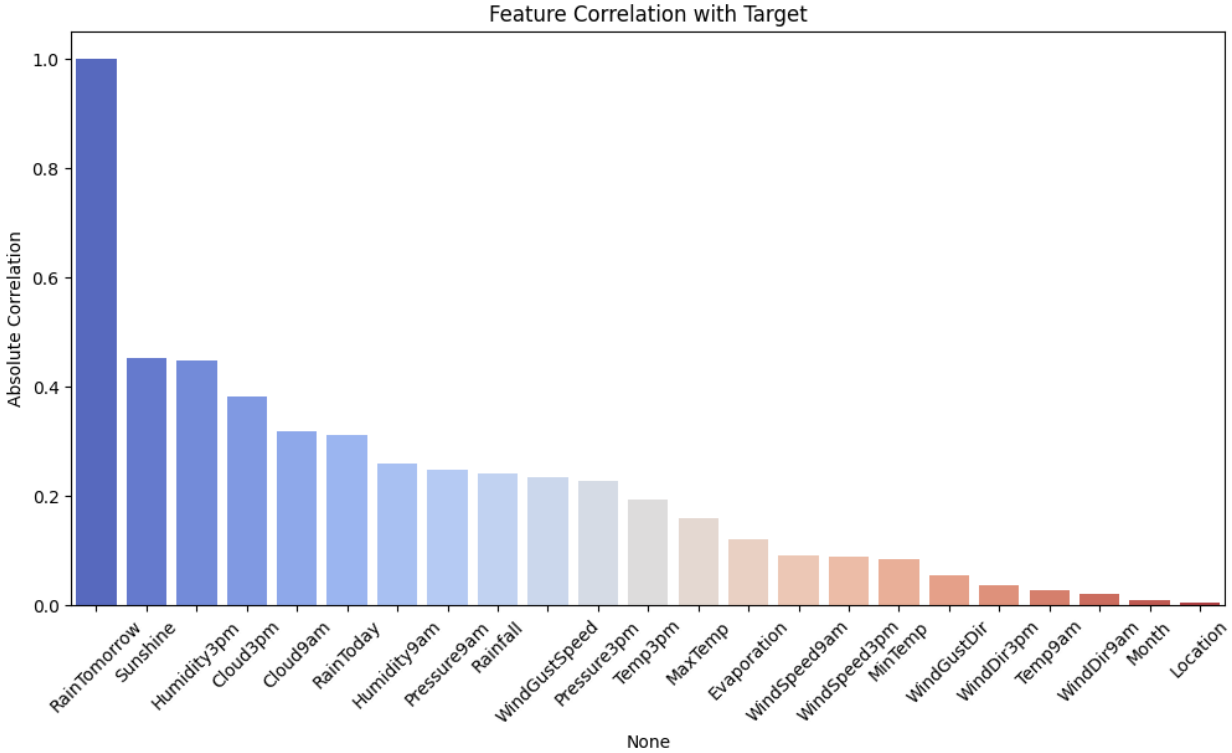
1. Import required Python libraries
2. Load dataset and perform initial inspection
3. Check dataset shape
4. Visualize missing values
5. Remove rows with missing target values
6. Encode categorical variables using Label Encoder
7. Handle missing values with Median, Mode, and Forward Fill
8. Apply SMOTE to balance classes
9. Split the dataset into training and testing sets
10. Apply StandardScaler
11. Train multiple machine learning models
12. Evaluate model performance using Accuracy, Precision, Recall, and F1-Score
13. Compare all models

Visualization:

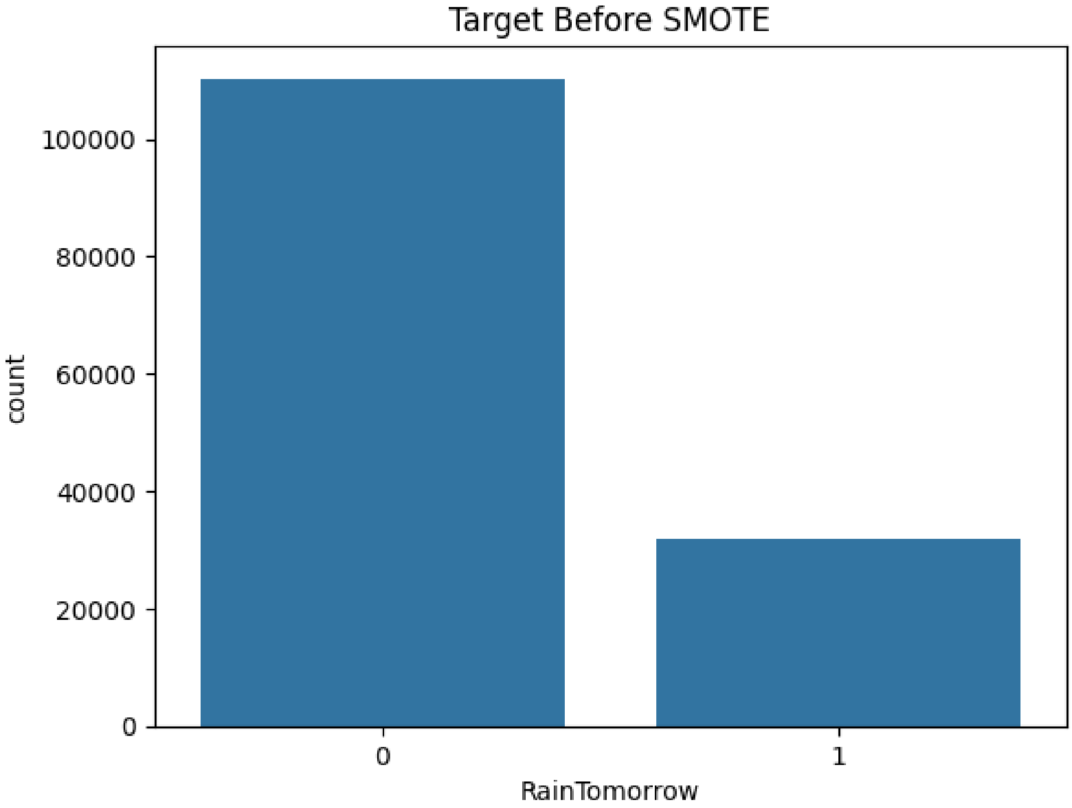
(i) Null Value:

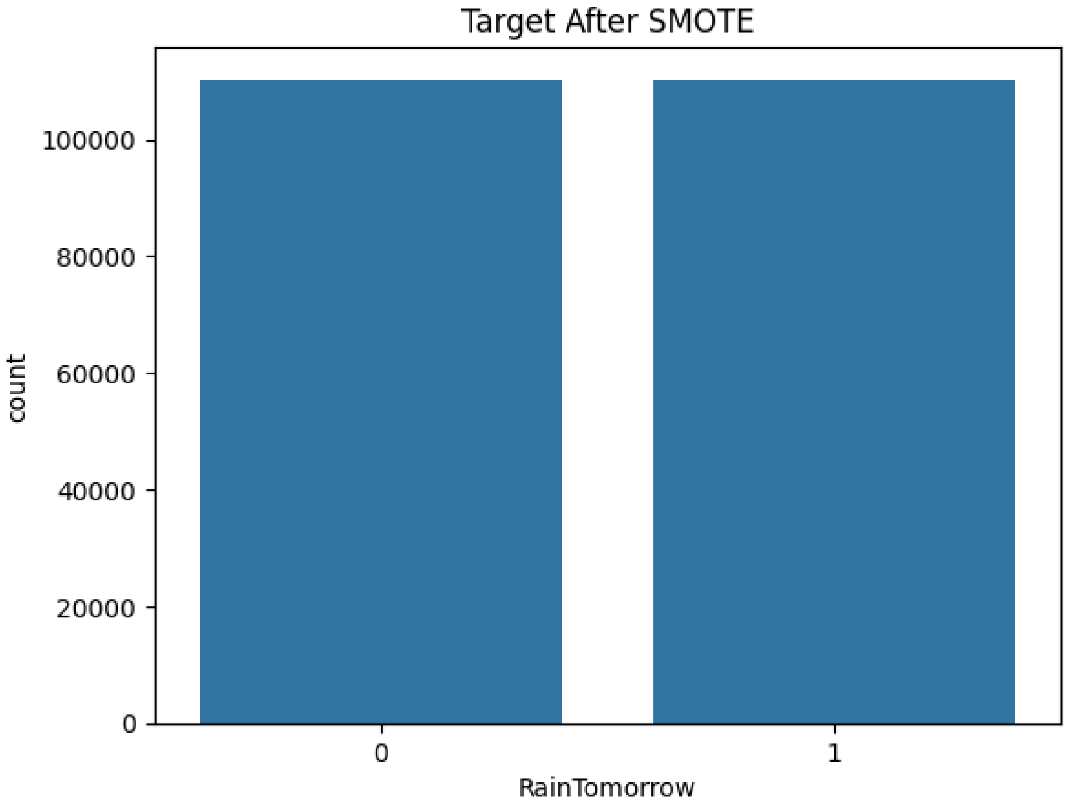


(ii) Correlation With Target Feature:

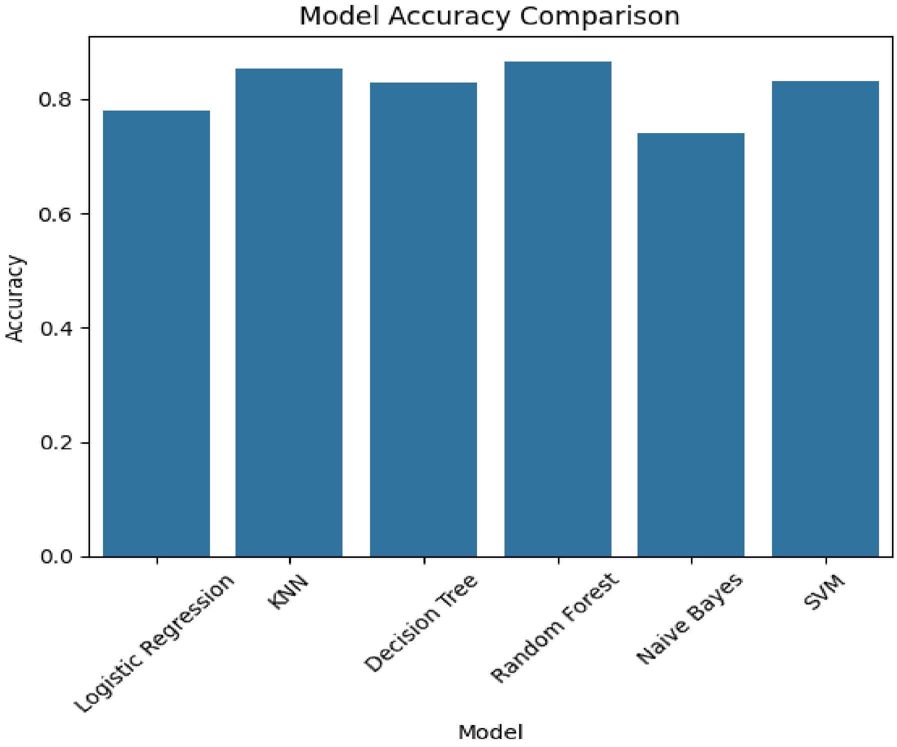


(iii) Handling Class Imbalance Using SMOTE





(iv) Model Performance Comparison Using Accuracy:



5. Results

Classification:

Random Forest performed best among all classifiers with the highest accuracy and balanced scores.

Model	Accuracy	Precision	Recall	F1 Score
Random Forest	0.866	0.867	0.866	0.866
Decision Tree	0.836	0.838	0.832	0.835
KNN	0.832	0.796	0.893	0.842
Logistic Regression	0.790	0.781	0.770	0.775
Naïve Bayes	0.748	0.740	0.735	0.737
SVM	0.805	0.798	0.790	0.794

6. Discussion

Among all the evaluated models, Random Forest achieved the highest accuracy and balanced metrics, demonstrating strong predictive performance for rainfall. Decision Tree also showed competitive results but slightly lagged due to its tendency to overfit. KNN delivered the highest recall, indicating strong sensitivity to rainy instances, but its lower precision suggests an increased rate of false positives. Logistic Regression and Naïve Bayes provided moderate results, making them suitable for simpler, linear relationships, though Naïve Bayes struggled with feature dependencies. SVM performed fairly well and could potentially improve with further hyperparameter tuning. Overall, ensemble-based learning (Random Forest) proved most effective by combining robustness, generalization, and reliable predictive capability across different evaluation metrics.