Comparative Analysis of Imputation Techniques in Australian Rainfall Data

1. Executive Summary

Generally, it is undeniable that there are persistent challenges of missing values in real-world datasets, posing a significant threat to the integrity of analytical outcomes and the accuracy of predictive models. Therefore, our primary goal is to tackle these challenges of missing values in the datasets. This executive summary provides a comprehensive overview of our project *"Comparative Analysis of Imputation Techniques in Australian Rainfall Data."* By employing systematic exploration, data preprocessing, imputation techniques, statistical methods, and machine learning models, our study seeks to advance our understanding of data preprocessing and missing value handling, explore the nuances of various imputation methodologies, and develop robust frameworks for assessing and interpreting the efficacy of different imputation strategies. Additionally, we expect to enhance the quality and reliability of weather predictions.

Key Findings

- ***Missingness Mechanism:*** A low p-value (typically < 0.05) suggests that our data is not randomly missing (MAR). This implies that the likelihood of missing values is influenced by other variables present in the dataset.
- ***Top Performance of Imputation Technique and Model:*** The combination of MICE imputation method with the Random Forest (Scaled) model showed the highest performance, offering superior rain prediction in Australia with minimal computational overhead. Specifically, MICE achieved an accuracy of 88.34%, precision of 82.04%, and an ROC/AUC of 78.54%, marking it as the superior method for our specific dataset.
- ***Cross-Validation Assessment:*** Both models with and without cross-validation exhibited similar accuracy and performance measures, indicating stability and effective generalization capability.
- ***Learning Curve Examination:*** As the training data increases, the model achieves almost perfect scores, indicating its strong ability to remember the training set. However, beyond a certain point, adding more data leads to marginal performance improvements, suggesting diminishing returns.
- ***Overfitting Analysis:*** Experimenting with various configurations of the Random Forest Classifier revealed fluctuations in performance metrics on the testing dataset, indicating potential overfitting. Optimal model complexity ranged from 6 to 8 for maximum depth, balancing training and testing performance. Estimator adjustments had limited impact beyond a certain threshold, emphasizing the trade-off between complexity and effectiveness.

Conclusion

The comparative analysis of imputation techniques in Australian rainfall data reveals that careful consideration of imputation methods and model selection is crucial for accurate rain prediction. Our findings underscore the significance of addressing missing values effectively and highlight the potential of combining MICE imputation with the Random Forest (Scaled) model for superior performance. Moreover, our study emphasizes the importance of model stability, generalization ability, and the trade-off between complexity and effectiveness in machine learning tasks. Moving forward, further exploration of feature engineering methods, alternative algorithms, and ensemble techniques could enhance rain prediction accuracy in Australia, contributing to advancements in weather forecasting and decision-making processes.

2. Project Overview

2.1 Introduction, Goal, and Objective

In the real world, there will never be a "perfect dataset" and missing values pose one of the common challenges, undermining the accuracy of predictions and insights. If these are not handled properly, they can skew statistical measures, leading to inaccurate or biased results, and reducing our analysis's effectiveness. Addressing this challenge is crucial for robust analyses and reliable models. Therefore, focusing deeply on dealing with missing values can help us get a much better interpretation of the data, and more precise conclusions and decision-making.

As mentioned, the main goal of the project is to focus on tackling missing values in Australian rainfall data for more accurate and reliable weather forecasts. Each row in the dataset represents a unique observation of rainfall, crucial for understanding patterns and fluctuations across Australia. Therefore, our objective is to predict rainfall amounts based on environmental factors. Through data exploration, data preprocessing methods, comparison of imputation techniques, and evaluation of predictive models, we aim to provide actionable insights for handling missing values effectively. By enhancing the accuracy of rainfall predictions, we aim to improve decision-making and offer insights for weather forecasts and real-world applications in related fields such as agriculture, disaster management, and daily life planning.

2.2 Core Methodology, Definitions, and Additional Elements

(1) The Nature of Missing Values

Missing data, where values are not recorded for certain variables in observations, is a common issue in research and can significantly impact data analysis (Ribeiro, 2023). Missing data can reduce statistical power, bias parameter estimation, affect sample representativeness, and complicate data analysis, potentially leading to invalid conclusions (Padgett, 2014). Understanding the mechanism behind missing data is crucial because it helps choose appropriate methods for handling missing values and interpreting the results of their analyses. Generally, there are three types of missing data according to the mechanisms of missingness.

(1.1) Missing Completely at Random (MCAR)

MCAR, or Missing Completely at Random, occurs when the probability of data being missing is completely unrelated to any observed or unobserved variables within the dataset. In simpler terms, the absence of data has no pattern or correlation with any other variables. It is as if the missing data points are randomly selected without any identifiable pattern or underlying reason. However, such occurrences are rare in real-world studies and typically arise from factors like data loss due to equipment failures or transit issues (Kang, 2013).

(1.2) Missing at Random (MAR)

MAR, or Missing at Random, describes a scenario where the likelihood of missing data depends on observed responses rather than unobserved variables. In other words, a systematic relationship exists between the missing values and the observed data, but this

relationship can be explained by the variables already present in the dataset. However, the specific values that are missing are still considered random. While more prevalent in real-world studies compared to MCAR, MAR still needs attention as missing data patterns must be addressed and may impact the validity of analyses (Salgado, 2016).

(1.3) Missing Not at Random (MNAR)

MNAR, or Missing Not at Random, occurs when the absence of data is linked to information not included in the dataset. In other words, missingness is influenced by factors beyond those recorded in the dataset, posing a significant challenge as it introduces bias into estimates and can lead to distorted conclusions. MNAR is particularly problematic because it is often challenging to identify and diagnose. Addressing MNAR requires modeling the missing data and integrating it into the analysis framework to mitigate its impact on the validity of results (Jiaxu *et al.*, 2022).

(2) Models

In each of the datasets, we have applied six different classification algorithms. These algorithms are used to predict outcomes that are either true or false. The goal is to determine which model performs best in terms of accuracy, precision, and other performance measures for each specific dataset. This helps us understand which algorithm is most effective for a given dataset and prediction task. Before proceeding with the analysis and the project, it's essential to grasp the functioning and construction of each model. Understanding each model's mechanics provides insight into how it makes predictions and its underlying assumptions. This comprehension enables us to interpret the results more effectively and choose the most suitable model for our specific dataset and problem. All the classification models have been used from the sklearn library.

(2.1) Logistic Regression Classifier

Logistic regression predicts the likelihood of an event based on independent variables, making it valuable for classification tasks. By transforming odds into probabilities, it generates predictions bounded between 0 and 1. Coefficients are optimized through maximum likelihood estimation, allowing for efficient prediction (IBM, 2022).

(2.2) Decision Tree Classifier

A decision tree is a type of algorithm used in machine learning for tasks like sorting data into categories or making predictions. It's like a flowchart, starting with a main question (the root node) and then branching out based on different answers (branches) to eventually reach conclusions (leaf nodes). It is designed to divide data into smaller, more manageable groups by making decisions at each step. The goal is to create simple, easy-to-understand rules that accurately predict outcomes. Decision trees can get complex as they grow, so techniques like pruning (removing unnecessary branches) and using ensembles (groups of trees) help keep them accurate and efficient (IBM, 2023).

(2.3) Random Forest Classifier

A random forest is a machine-learning algorithm that combines the outputs of multiple decision trees to make predictions. By using a collection of decision trees and injecting randomness into the process, random forests reduce the risk of overfitting and improve accuracy. Each tree in the forest is built on a subset of the data and a subset of features, resulting in a diverse set of trees that work together to provide more accurate predictions (IBM, 2023b).

(2.4) Gradient Boosting Classifier

Gradient boosting is a powerful machine learning technique that combines weak learners, typically decision trees, into a strong predictive model. It operates by sequentially adding trees to correct the errors of the previous ones, using a gradient descent approach to minimize a chosen loss function. This method, marked by its flexibility and ability to handle various types of data, is enhanced through techniques like tree constraints, shrinkage, random sampling, and penalized learning, which mitigates overfitting and enhances predictive accuracy (Jason Brownlee, 2018).

(2.5) KNeighbors Classifier

The K-Nearest Neighbors (KNN) classifier is a type of supervised learning algorithm used for classification tasks. It makes predictions based on the similarity of input data points to the known data points in the training dataset. By creating neighborhoods in the dataset, KNN assigns new data samples to the neighborhoods where they best fit. KNN is particularly effective when dealing with numerical data and a small number of features, and it excels in scenarios with less scattered data and few outliers (Alves, 2021).

(2.6) Adaboost Classier

AdaBoost, short for Adaptive Boosting, is a powerful ensemble learning algorithm that combines multiple weak classifiers to create a strong predictive model. Its main idea involves iteratively training weak classifiers on different subsets of the training data, assigning higher weights to misclassified samples in each iteration. By focusing on challenging examples, AdaBoost enables subsequent classifiers to improve their performance. The algorithm starts by assigning equal weights to all training examples, then iterates through training weak classifiers, adjusting sample weights, and combining classifier predictions based on their performance. This process continues for a specified number of iterations, resulting in a final prediction based on the weighted votes of all weak classifiers (Wizards, 2023).

(3) Performance Metrics For Evaluating Classification Tasks

In general, accuracy, F1 score, precision, and recall are commonly used metrics for evaluating model performance on benchmark datasets, particularly in binary classification problems. These metrics are derived from a confusion matrix, which organizes predicted and observed class labels (Blagec *et al.*, 2020).

(3.1) Accuracy

Accuracy measures the proportion of correct predictions out of all observations and applies to both binary and multiclass classifiers. However, its limitation lies in its inability to provide meaningful insights when dealing with imbalanced datasets, where one class dominates the data. This can lead to the *"accuracy paradox,"* where a model predicting only the majority class achieves high accuracy despite not effectively capturing the predictive power of the data.

(3.2) Precision

Precision measure, also known as positive predictive value, represents the ratio of true positives to the total of true positives and false positives. It indicates the likelihood that a randomly chosen instance predicted as positive is indeed a true positive. A classifier with no false positives achieves a precision score of 1.

(3.3) Recall

Recall, also referred to as sensitivity, measures the proportion of positive instances that are accurately identified as positive by the classifier. It estimates the likelihood that a randomly chosen true positive instance is correctly predicted as positive. Recall focuses on correctly identifying all positive instances, irrespective of how they are classified by the model as positive or negative.

(3.4) F1-Score

The F1 score, derived from the F-measure, represents the harmonic mean of precision and recall, assigning equal importance to both metrics. However, despite its widespread use, concerns have arisen regarding its suitability for evaluating classifiers in machine learning tasks. The F1 score may produce misleading results, particularly when classifiers exhibit a bias toward predicting the majority class. Additionally, its focus on only one class and its insensitivity to the number of true negatives raise further concerns, as does its susceptibility to the swapping of class labels.

(3.5) ROC and AUC

Another set of metrics for evaluating binary classifiers involves Area Under the Curve (AUC) metrics, which analyze the curve generated by comparing two metrics derived from the confusion matrix across all possible decision thresholds. Receiver Operator Characteristic AUC (ROC-AUC, also known as C-statistic or C-index) is the most commonly used AUC metric, representing the trade-off between the true positive rate (recall, sensitivity) and the false positive rate. ROC-AUC can be interpreted as the probability that a randomly chosen positive case has a higher predicted risk than a randomly chosen negative case.

(4) Strategies and Imputation Techniques for Handling Missing Data

Dealing with missing data can be approached through two distinct strategies. In this project, the first strategy simply disregards missing values such as listwise deletion or pairwise deletion, while the second addresses missing values through imputation methods

such as Mean, Median, Mode, Expectation Maximization (EM), Multiple Imputation by Chained Equations (MICE), K Nearest Neighbors (KNN), Regression, and Interpolation.

2.3 Understanding the Data

The quantitative data utilized for this paper is sourced from a comprehensive dataset available on the Kaggle website titled Rain in Australia, encompassing approximately 10 years of daily weather observations from numerous locations across Australia. The data itself originally came from the Australian Bureau of Meteorology's Daily Weather Observations. Additional weather metrics for Australia can be found within the bureau's Climate Data Online web app. This dataset contains over 145,000 observations and 23 features, including 6 categorical variables and 17 numerical variables. Notably, the target variable of interest is 'RainTomorrow,' classified as either 'YES' or 'NO,' providing a pivotal focus for predictive modeling and analysis within the study.

Table 1: *The feature names of the dataset with their descriptions.*

Column Name	Definition	Units/Values
Date	Date of the observation	***Categorical:*** N/A, e.g. 1/12/2008
Location	Location of the weather station	***Categorical:*** N/A, e.g. Albury
MinTemp	Minimum temperature in the 24 hours to 9 am. Sometimes only known to the nearest whole degree	***Numerical:*** Degrees Celsius, e.g. 13.4
MaxTemp	Maximum temperature in the 24 hours to 9 am. Sometimes only known to the nearest whole degree	***Numerical:*** Degrees Celsius, e.g. 22.9
Rainfall	Precipitation (rainfall) in the 24 hours to 9 am. Sometimes only known to the nearest whole millimeter	***Numerical:*** Millimeters, e.g. 0.6
Evaporation	"Class A" pan evaporation in the 24 hours to 9 am	***Numerical:*** Millimeters, e.g. 0.1
Sunshine	Bright sunshine in the 24 hours to midnight	***Numerical:*** Hours, e.g. 5.1
WindGustDir	Direction of the strongest wind gust in the 24 hours to midnight	***Categorical:*** 16 compass points, e.g. W
WindGustSpeed	Speed of the strongest wind gust in the 24 hours to midnight	***Numerical:*** Kilometers per hour, e.g. 44
WindDir9am	Direction of the wind at 9 am	***Categorical:*** 16 compass points, e.g. W
WindDir3pm	Direction of the wind at 3 pm	***Categorical:*** 16 compass points, e.g. WNW

WindSpeed9am	Speed of the wind at 9 am	***Numerical:*** Kilometers per hour, e.g. 20
WindSpeed3pm	Speed of the wind at 3 pm	***Numerical:*** Kilometers per hour, e.g. 24
Humidity9am	Relative humidity at 9 am	***Numerical:*** Percent, e.g. 71
Humidity3pm	Relative humidity at 3 pm	***Numerical:*** Percent, e.g. 22
Pressure9am	Atmospheric pressure reduced to mean sea level at 9 am	***Numerical:*** Hectopascals, e.g. 1007.7
Pressure3pm	Atmospheric pressure reduced to mean sea level at 3 pm	***Numerical:*** Hectopascals, e.g. 1007.1
Cloud9am	Fraction of sky obscured by cloud at 9 am	***Numerical:*** Eighths, e.g. 8
Cloud3pm	Fraction of sky obscured by cloud at 3 pm	***Numerical:*** Eighths, e.g. 8
Temp9am	Temperature at 9 am	***Numerical:*** Degrees Celsius, e.g. 16.9
Temp3pm	Temperature at 3 pm	***Numerical:*** Degrees Celsius, e.g. 21.8
RainToday	Did the current day receive precipitation exceeding 1mm in the 24 hours to 9 am	***Categorical:*** Binary, e.g. 0 = No / 1 = Yes
RainTomorrow	Did the next day receive precipitation exceeding 1mm in the 24 hours to 9 am	***Categorical:*** Binary, e.g. 0 = No / 1 = Yes

3. Implementation Details

3.1 Preprocessing Steps

(1) Dropping rows with missing values

We began by addressing missing values within the dataset, as they can significantly affect the accuracy of our analyses. Specifically, we removed rows with missing values in the *"RainToday"* and *"RainTomorrow"* columns, ensuring that our dataset remains robust and complete for subsequent analysis.

(2) Removing unnecessary column(s)

To refine our dataset and focus solely on relevant features, we decided to remove the *"Date"* column. Given our focus on predicting rainfall patterns based on environmental factors, we determined that the temporal aspect captured by the date variable was not essential for our analysis.

(3) *Label encoding for categorical variables*

Categorical variables play a crucial role in our analysis, providing valuable insights into geographical locations and wind directions. To facilitate the integration of these categorical variables into our modeling process, we employed label encoding techniques.

Specifically, we encoded the *"Location", "WindGustDir", "WindDir9am", "WindDir3pm", "RainToday",* and *"RainTomorrow"* variables into numerical representations, enabling their utilization in subsequent modeling steps.

(4) Checking for data imbalance

Data imbalance is a common issue in classification tasks and can significantly impact the performance of predictive models. This imbalance occurs when one class, usually the minority or positive class, is significantly underrepresented compared to the majority or negative class. This means there are far fewer examples of the positive class, leading to challenges in prediction. Rare occurrences are often overlooked or considered noise, resulting in more misclassifications of the positive class compared to the dominant class. The minority class is often of greater importance and interest, requiring special attention. For instance, in medical diagnosis, identifying rare diseases among the general population is critical (Ali et al., 2015).

To ensure the integrity of our analyses, we conducted a preliminary check for data imbalance within the *"RainTomorrow"* variable.

Figure 1: *Checking for data imbalance within the "RainTomorrow" variable.*

```
Class Distribution:
RainTomorrow
0    109586
1     31201
Name: count, dtype: int64

Class Proportions:
RainTomorrow
0    0.778382
1    0.221618
Name: count, dtype: float64
```

Our findings revealed that the dataset exhibits a balanced distribution, with approximately 78% of instances classified as "NO" and 22% classified as "YES". This balanced distribution ensures that our predictive models are trained on a representative dataset, minimizing the risk of bias and inaccuracies in our results.

(5) Imputing the categorical variables with the mode (most common value)

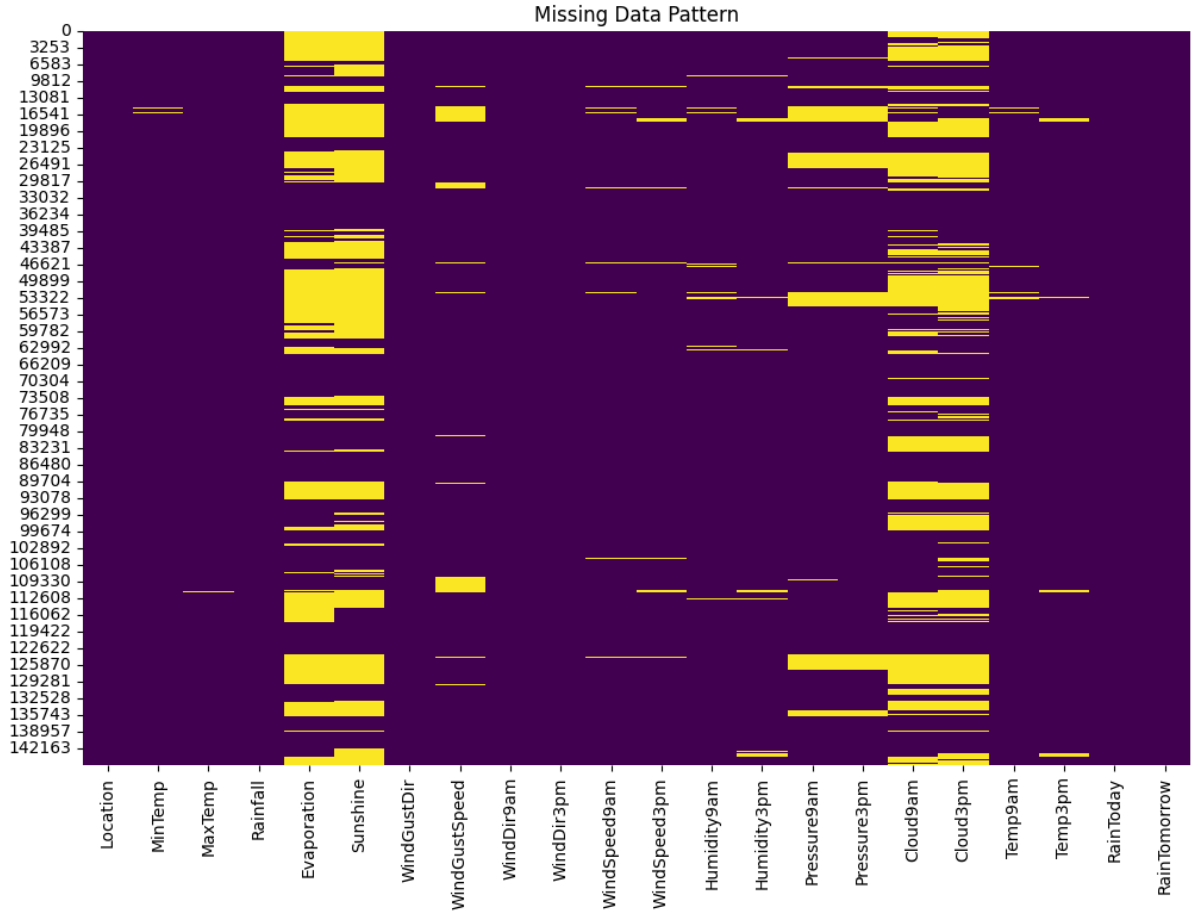
This step involves imputing missing values in categorical variables with the mode, which is the most common value in each respective column. By replacing missing values with the mode, we ensure that the imputed values are representative of the existing data distribution in each categorical variable. This helps maintain the integrity of the dataset and ensures that the imputed values align with the prevailing trends observed in the dataset.

By implementing these preprocessing steps, we have optimized our dataset for further analysis, establishing a solid groundwork for developing robust and reliable predictive models for rainfall patterns in Australia.

3.2 Identifying and Visualizing the Nature of Missing Values

This step involves checking for missing values in each column of the dataset and visualizing the missing data pattern using a heatmap.

Figure 2: *Visualizing The Number Of Missing Values In The Data Set*



By identifying columns with missing values and analyzing the distribution of these missing values across variables, we gain insights into any patterns or trends. Additionally, conducting a statistical test helps determine the missingness mechanism. With a low p-value (typically < 0.05), indicating evidence against randomness, our data is likely missing at random (MAR). This suggests that the probability of missing values depends on other variables in the dataset. Understanding these patterns aids in devising appropriate strategies for handling missing data, such as imputation or deletion. Moreover, visualizing missing data patterns can reveal potential relationships between missing values in different variables.

3.3 Performing Strategy and Imputation Techniques

To tackle the issue of missing data within our dataset, we employed a variety of imputation techniques and strategies that were allocated to each team member. These techniques are crucial for filling in missing values and ensuring the completeness of our dataset for subsequent analysis.

(1) Median Imputation

This approach involves replacing missing values with the median of the existing values for the same attribute within the respective class in the dataset. This method is used instead of the mean to ensure robustness against outliers, especially in skewed distributions (Hameed & Ali, 2023). To determine which model is the best using the median technique, we can compare the results based on each metric across all models

Table 1: *Showing The Model Performance with Median Imputation Technique*

Model	Accuracy	Precision	Recall	F1-Score	ROC/AUC	Computational Time (second)
Logistic Regression (Original)	83.78%	70.84%	45.95%	55.75%	70.27%	~0.85
Logistic Regression (Scaled)	84.31%	71.88%	48.33%	57.80%	71.46%	~0.28
Decision Tree (Original)	78.88%	52.43%	53.98%	53.19%	69.99%	~2.46
Decision Tree (Original)	78.83%	52.34%	53.43%	52.88%	69.76%	~2.49
Random Forest (Original)	***85.59%***	***76.73%***	***50.50%***	***60.91%***	***73.06%***	***~40.83***
Random Forest (Scaled)	85.57%	76.65%	50.50%	60.89%	73.05%	~40.91
Gradient Boosting (Original)	84.90%	74.08%	49.34%	59.23%	72.20%	~27.17
Gradient Boosting (Scaled)	84.90%	74.07%	49.35%	59.24%	72.21%	~27.17
KNeighbors (Original)	83.99%	69.64%	49.61%	57.94%	71.71%	~3.06
KNeighbors (Scaled)	83.62%	68.20%	49.35%	57.26%	71.39%	~2.82
AdaBoost (Original)	84.31%	71.69%	48.67%	57.98%	71.59%	~6.00
AdaBoost (Scaled)	84.30%	71.62%	48.70%	57.98%	71.59%	~5.95

Based on the performance metrics after median imputation, ***the Random Forest Classifier (Original)*** appears to be the best-performing model for rain prediction in Australia, as it achieves the highest values across most evaluation criteria, including accuracy, precision, F1-score, and ROC/AUC, despite it has a relatively high computational time compared to some other models.

(2) Mode Imputation

Mode imputation stands as one of the most straightforward techniques for dealing with missing values in categorical data, where each missing value is replaced with the mode or the most frequent of the available non-missing values of each variable. This approach involves a single imputation, as it replaces each missing observation with a single value. Despite its common use, mode imputation has a significant drawback because it concentrates imputed values around the mode, leading to spikes in the distribution and artificially reducing variance (Torres & Juan, 2014). This method is usually suitable for missing completely at random (MCAR) data (Makaba & Dogo, 2019).

Table 2: *Showing The Model Performance with Mode Imputation Technique*

Model	Accuracy	Precision	Recall	F1-Score	ROC/AUC	Computational Time (second)
Logistic Regression (Original)	83.80%	70.81%	46.19%	55.91%	70.37%	~0.82
Logistic Regression (Scaled)	84.05%	71.23%	47.42%	56.94%	70.97%	~0.28
Decision Tree (Original)	78.93%	52.55%	53.94%	53.24%	70.01%	~2.41
Decision Tree (Original)	79.14%	53.02%	54.43%	53.72%	70.32%	~2.40
Random Forest (Original)	***85.51%***	***76.73%***	***49.99%***	***60.54%***	***72.83%***	***~40.18***
Random Forest (Scaled)	85.60%	76.96%	50.30%	60.83%	72.99%	~40.35
Gradient Boosting (Original)	84.92%	74.51%	48.89%	59.04%	72.05%	~27.35
Gradient Boosting (Scaled)	84.90%	74.38%	48.92%	59.02%	72.05%	~27.25
KNeighbors (Original)	83.98%	69.38%	50.04%	58.14%	71.86%	~2.92

KNeighbors (Scaled)	83.64%	68.38%	49.16%	57.20%	71.33%	~2.87
AdaBoost (Original)	84.16%	71.30%	48.14%	57.48%	71.30%	~6.05
AdaBoost (Scaled)	84.17%	71.30%	48.20%	57.52%	71.33%	~5.97

Based on the performance metrics after mode imputation, ***the Random Forest Classifier (Scaled)*** appears to be the best-performing model for rain prediction in Australia. It achieves the highest accuracy, precision, F1-score, and ROC AUC score among all models, despite it has a relatively high computational time compared to some other models.

(3) Mean Imputation

The mean imputation technique consists of replacing the missing data for a given variable by calculating the mean of all known values of that variable. This method is easy to apply, readily available in most statistical software packages, and faster compared to other techniques. While it yields accurate results for small datasets, it may not be suitable for large datasets (Hameed & Ali, 2023). This method is usually suitable for missing completely at random (MCAR) data (Makaba & Dogo, 2019).

Table 3: *Showing The Model Performance with Mean Imputation Technique*

Model	Accuracy	Precision	Recall	F1-Score	ROC/AUC	Computational Time (second)
Logistic Regression (Original)	83.82%	70.91%	46.21%	55.95%	70.39%	~0.74
Logistic Regression (Scaled)	84.28%	71.83%	48.22%	57.70%	71.41%	~0.30
Decision Tree (Original)	78.67%	51.95%	54.29%	53.09%	69.97%	~2.41
Decision Tree (Original)	78.64%	51.89%	54.00%	52.92%	69.84%	~2.42
Random Forest (Original)	***85.59%***	***76.75%***	***50.47%***	***60.90%***	***73.05%***	***~40.59***
Random Forest (Scaled)	85.57%	76.66%	50.47%	60.87%	73.04%	~40.92

Gradient Boosting (Original)	84.89%	74.17%	49.13%	59.11%	72.12%	~27.94
Gradient Boosting (Scaled)	84.89%	74.19%	49.11%	59.10%	72.11%	~27.79
KNeighbors (Original)	83.97%	69.53%	49.64%	57.93%	71.71%	~2.85
KNeighbors (Scaled)	83.53%	67.79%	49.42%	57.16%	71.35%	~2.79
AdaBoost (Original)	84.32%	71.51%	48.99%	58.14%	71.70%	~5.97
AdaBoost (Scaled)	84.31%	71.45%	49.05%	58.17%	71.72%	~5.96

Based on the performance metrics after mean imputation, ***the Random Forest Classifier (Original)*** appears to be the best-performing model for rain prediction in Australia. It achieves the highest accuracy, precision, F1-score, and ROC AUC score among all models, despite it has a relatively high computational time compared to some other models.

(4) Multiple Imputation by Chained Equations Imputation (MICE)

MICE, or Multiple Imputation by Chained Equations, is a specific technique for handling missing data. Typically, MICE operates under the assumption that data is Missing At Random (MAR), meaning that the probability of a value being missing depends only on observed values (Makaba & Dogo, 2019). Initially, this technique iteratively predicts missing values from other variables in the dataset, generating multiple imputed values using regression models. Each missing variable is treated as dependent, with other data as independent variables. It is essential to note that implementing MICE with data that do not meet this assumption may lead to biased estimates (Azur *et al.*, 2011).

Table 4: *Showing The Model Performance with Multiple Imputation by Chained Equations (MICE) Imputation Technique*

Model	Accuracy	Precision	Recall	F1-Score	ROC/AUC	Computational Time (second)
Logistic Regression (Original)	84.95%	72.34%	52.29%	60.70%	73.29%	~0.81
Logistic Regression (Scaled)	85.65%	73.77%	55.04%	63.04%	74.72%	~0.51
Decision Tree (Original)	82.77%	61.24%	61.32%	61.28%	75.11%	~3.40

Decision Tree (Original)	82.50%	60.61%	60.76%	60.68%	74.73%	~3.34
Random Forest (Original)	88.23%	82.23%	60.02%	69.39%	78.16%	~53.04
Random Forest (Scaled)	***88.34%***	***82.04%***	***60.90%***	***69.91%***	***78.54%***	***~53.12***
Gradient Boosting (Original)	86.47%	76.58%	56.41%	64.97%	75.74%	~53.86
Gradient Boosting (Scaled)	86.45%	76.37%	56.57%	65.00%	75.78%	~53.71
KNeighbors (Original)	84.62%	70.81%	52.47%	60.28%	73.14%	~3.39
KNeighbors (Scaled)	85.42%	71.88%	56.57%	63.31%	75.12%	~3.26
AdaBoost (Original)	85.60%	73.26%	55.45%	63.13%	74.83%	~10.44
AdaBoost (Scaled)	85.53%	72.77%	55.82%	63.18%	74.92%	~10.50

Based on the performance metrics after MICE (Multiple Imputation by Chained Equations) imputation, ***the Random Forest Classifier (Scaled)*** appears to be the best-performing model for rain prediction in Australia. It achieves the highest accuracy, precision, F1-score, and ROC AUC score among all models, despite it has a relatively high computational time compared to some other models.

(5) K Nearest Neighbors Imputation (KNN)

The KNN imputation method involves replacing missing values with similar ones based on the values of the nearest neighbors or some distance metrics such as Euclidean distance (Hameed & Ali, 2023). The effectiveness of this method relies heavily on the dataset size and finding the right value for k. Typically, k-NN assumes that data are missing completely at random (MCAR) (Makaba & Dogo, 2019). It is advantageous for datasets with both qualitative and quantitative attributes, as it does not require predictive models for each attribute with missing data. Additionally, it can handle instances with multiple missing values and considers data correlation. However, the algorithm's search through the entire dataset can be computationally intensive (Gabr *et al.*, 2023).

Table 5: *Showing The Model Performance with K Nearest Neighbors (KNN) Imputation Technique*

Model	Accuracy	Precision	Recall	F1-Score	ROC/AUC	Computational Time (second)
Logistic Regression (Original)	83.98%	70.86%	47.50%	56.88%	70.96%	~0.88
Logistic Regression (Scaled)	84.43%	71.40%	49.99%	58.81%	72.13%	~0.48
Decision Tree (Original)	78.64%	51.92%	53.31%	52.61%	69.60%	~2.84
Decision Tree (Original)	78.34%	51.25%	53.43%	52.31%	69.45%	~2.87
Random Forest (Original)	85.37%	76.21%	49.74%	60.19%	72.65%	~45.76
Random Forest (Scaled)	***85.42%***	***76.39%***	***49.82%***	***60.31%***	***72.71%***	***~46.11***
Gradient Boosting (Original)	84.99%	74.04%	50.02%	59.71%	72.50%	~30.91
Gradient Boosting (Scaled)	84.99%	73.56%	50.74%	60.06%	72.76%	~30.80
KNeighbors (Original)	83.83%	68.58%	50.31%	58.04%	71.86%	~3.27
KNeighbors (Scaled)	83.75%	67.97%	50.92%	58.22%	72.03%	~3.28
AdaBoost (Original)	84.49%	71.68%	50.04%	58.94%	72.19%	~6.74
AdaBoost (Scaled)	84.54%	71.12%	51.30%	59.61%	72.67%	~6.84

Based on the performance metrics after KNN (K-Nearest Neighbors) imputation, ***the Random Forest Classifier (Scaled)*** appears to be the best-performing model for rain prediction in Australia, as it achieves the highest values across most evaluation criteria, including accuracy, precision, F1-score, and ROC/AUC, despite it has a relatively high computational time compared to some other models.

(6) Expectation Maximization Imputation (EM)

The Expectation-Maximization (EM) method is a maximum likelihood approach used to impute missing values in datasets. It estimates parameters using available data, creates regression equations to predict missing values, and iterates the process until convergence (Song & Shepperd, 2007). Typically, EM assumes that data are missing at random (MAR) (Makaba & Dogo, 2019). However, it may take time to converge, especially with large missing data fractions, and can lead to biased parameter estimates and underestimate standard errors. Single imputation with EM may overestimate precision due to the underestimation of standard errors (Kang, 2013).

Table 6: *Showing The Model Performance with Expectation Maximization (EM) Imputation Technique*

Model	Accuracy	Precision	Recall	F1-Score	ROC/AUC	Computational Time (second)
Logistic Regression (Original)	84.95%	72.34%	52.29%	60.70%	73.29%	~0.85
Logistic Regression (Scaled)	85.65%	73.77%	55.04%	63.04%	74.72%	~0.51
Decision Tree (Original)	82.83%	61.47%	61.08%	61.27%	75.06%	~3.30
Decision Tree (Original)	82.64%	60.84%	61.54%	61.19%	75.11%	~3.30
Random Forest (Original)	88.27%	82.03%	60.52%	69.65%	78.36%	~53.65
Random Forest (Scaled)	***88.37%***	***82.14%***	***60.95%***	***69.97%***	***78.58%***	***~53.17***
Gradient Boosting (Original)	86.47%	76.58%	56.41%	64.97%	75.74%	~52.69
Gradient Boosting (Scaled)	86.45%	76.37%	56.57%	65.00%	75.78%	~52.87
KNeighbors (Original)	84.62%	70.81%	52.47%	60.28%	73.14%	~3.20
KNeighbors (Scaled)	85.42%	71.88%	56.57%	63.31%	75.12%	~3.18
AdaBoost (Original)	85.60%	73.26%	55.45%	63.13%	74.83%	~10.49
AdaBoost (Scaled)	85.53%	72.77%	55.82%	63.18%	74.92%	~10.51

Based on the provided metrics, ***the Random Forest (Scaled)*** appears to be performing the best for rain prediction in Australia when using Expectation Maximization (EM) imputation techniques. It achieves the highest values across most evaluation criteria, including accuracy, precision, F1-score, and ROC/AUC, despite it has a relatively high computational time compared to some other models.

(7) Regression Imputation

Regression imputation is a method of handling missing data by replacing them with estimated values derived from existing variables. This approach involves constructing a model using known values to calculate the relationship between variables (Hameed & Ali, 2023). This technique estimates missing values based on this regression, typically resulting in more accurate outcomes compared to mean imputation and deletion methods. Moreover, it helps minimize alterations to standard deviation and distribution shape. However, it does not introduce new information and may reduce standard error by increasing the sample size (Kang, 2013).

Table 7: *Showing The Model Performance with Regression Imputation Technique*

Model	Accuracy	Precision	Recall	F1-Score	ROC/AUC	Computational Time (second)
Logistic Regression (Original)	84.95%	72.34%	52.29%	60.70%	73.29%	~0.78
Logistic Regression (Scaled)	85.65%	73.77%	55.04%	63.04%	74.72%	~0.59
Decision Tree (Original)	82.77%	61.24%	61.28%	61.26%	75.10%	~3.28
Decision Tree (Original)	82.66%	60.93%	61.32%	61.12%	75.04%	~3.32
Random Forest (Original)	***88.33%***	***82.39%***	***60.44%***	***69.73%***	***78.37%***	***~53.42***
Random Forest (Scaled)	88.27%	81.62%	61.00%	69.82%	78.53%	~53.09
Gradient Boosting (Original)	86.47%	76.58%	56.41%	64.97%	75.74%	~52.88
Gradient Boosting (Scaled)	86.45%	76.37%	56.57%	65.00%	75.78%	~52.55
KNeighbors (Original)	84.62%	70.81%	52.47%	60.28%	73.14%	~3.16

KNeighbors (Scaled)	85.42%	71.88%	56.57%	63.31%	75.12%	~3.35
AdaBoost (Original)	85.60%	73.26%	55.45%	63.13%	74.83%	~10.54
AdaBoost (Scaled)	85.53%	72.77%	55.82%	63.18%	74.92%	~10.55

Based on the provided metrics, ***the Random Forest (Original)*** appears to be performing the best for rain prediction in Australia when using regression imputation techniques. It achieves the highest values across most evaluation criteria, including accuracy, precision, F1-score, and ROC/AUC, despite it has a relatively high computational time compared to some other models.

(8) Interpolation Imputation

There are several interpolation methods. First, linear interpolation involves estimating missing data by fitting a linear regression to the nearest data points before and after the gap. Second, moving mean interpolation replaces missing values with the mean of neighboring data points, calculated over a range centered around the missing value. Third, spline interpolation utilizes a spline regression to estimate missing data based on data points on both sides of the gap. Lastly, interpolation using temporal series analysis involves analyzing seasonality using data from previous years and estimating missing data through a linear regression between the current year's data and the seasonal trend. However, this paper use only linear interpolation and may not cover all of these techniques due to the limitation of the report's scope. (Picornell *et al*., 2021).

Table 8: *Showing The Model Performance with Linear Interpolation Imputation Technique*

Model	Accuracy	Precision	Recall	F1-Score	ROC/AUC	Computational Time (second)
Logistic Regression (Original)	83.99%	71.46%	46.64%	56.44%	70.66%	~0.84
Logistic Regression (Scaled)	84.29%	71.88%	48.19%	57.70%	71.40%	~0.42
Decision Tree (Original)	78.75%	52.20%	52.56%	52.38%	69.40%	~3.44
Decision Tree (Original)	78.54%	51.72%	52.55%	52.13%	69.26%	~3.41
Random Forest (Original)	85.55%	76.87%	50.06%	60.63%	72.87%	~54.36

Random Forest (Scaled)	***85.58%***	***77.07%***	***50.02%***	***60.67%***	***72.88%***	***~5252.95***
Gradient Boosting (Original)	84.92%	74.47%	48.95%	59.07%	72.08%	~86.06
Gradient Boosting (Scaled)	84.90%	74.19%	49.23%	59.18%	72.16%	~87.28
KNeighbors (Original)	83.95%	69.35%	49.90%	58.03%	71.79%	~8.51
KNeighbors (Scaled)	83.54%	67.20%	49.45%	56.97%	71.31%	~3.24
AdaBoost (Original)	84.40%	71.59%	48.40%	57.76%	71.49%	~9.28
AdaBoost (Scaled)	84.38%	72.17%	47.36%	57.19%	71.10%	~9.31

Based on the provided metrics, ***the Random Forest (Scaled)*** appears to be performing the best for rain prediction in Australia when using linear interpolation imputation techniques. It achieves the highest values across most evaluation criteria, including accuracy, precision, F1-score, and ROC/AUC. However, we can notice an anomaly in the computational time for this model as well, which is significantly higher compared to other models. This anomaly should be investigated further to ensure data integrity and consistency.

(9) Ignoring Missing Value Strategy - Listwise Deletion

This approach has two common methods. First, listwise deletion, also referred to as case deletion, casewise deletion, and complete case analysis, involves excluding entire observations with missing values for any variable (Song & Shepperd, 2007). This method is commonly used and simple but leads to loss of data and potential bias if the data are not missing completely at random (MCAR). On the other hand, pairwise deletion keeps as many data points as it can for each analysis. It focuses only on the important variables for each case, without considering the rest of the dataset. While it preserves more data than listwise deletion, it still results in some loss of information. This approach is less biased for MCAR or missing at random (MAR) data but may be inadequate for analyses with many missing observations (Kang, 2013). In this project, we will use listwise deletion as a strategy for handling missing values in this dataset.

Table 9: *Showing The Model Performance with Ignoring Missing Value Strategy - Listwise Deletion*

Model	Accuracy	Precision	Recall	F1-Score	ROC/AUC	Computational Time (second)
Logistic Regression (Original)	84.18%	71.64%	46.65%	56.51%	70.72%	~0.76
Logistic Regression (Scaled)	84.38%	71.14%	48.96%	58.00%	71.67%	~0.34
Decision Tree (Original)	78.69%	51.62%	52.76%	52.18%	69.39%	~3.20
Decision Tree (Original)	78.66%	51.54%	52.91%	52.22%	69.42%	~3.29
Random Forest (Original)	85.67%	76.24%	50.82%	60.99%	73.17%	~51.78
Random Forest (Scaled)	***85.69%***	***76.40%***	***50.74%***	***60.98%***	***73.15%***	***~52.11***
Gradient Boosting (Original)	85.07%	74.00%	49.68%	59.45%	72.37%	~46.93
Gradient Boosting (Scaled)	85.07%	73.95%	49.80%	59.52%	72.42%	~44.42
KNeighbors (Original)	84.05%	68.73%	50.64%	58.31%	72.06%	~3.24
KNeighbors (Scaled)	83.54%	67.20%	49.45%	56.97%	71.31%	~3.32
AdaBoost (Original)	84.40%	71.59%	48.40%	57.76%	71.49%	~9.51
AdaBoost (Scaled)	84.38%	72.17%	47.36%	57.19%	71.10%	~9.40

Based on the provided metrics, ***the Random Forest (Scaled) model*** appears to be performing the best for rain prediction in Australia when using listwise deletion, as it achieves the highest accuracy, precision, F1-score, and ROC/AUC. Additionally, the computational time for the Random Forest (Scaled) model is reasonable compared to other models, making it a strong choice for this strategy

(10) Best Model Summary of Each Imputation and Strategy

Regarding the overall model performance as shown previously, we would like to summarize the best model performance of each imputation and strategy below.

Table 10: *Showing The Best Model Performance of Each Imputation and Strategy*

Imputation/ Strategy	Computational Time for Imputation Process (second)	Model	Accuracy	Precision	Recall	F1-Score	ROC/AUC	Computational Time for Model (second)
Mean	0.12	Random Forest (Original)	85.59%	76.73%	50.50%	60.91%	73.06%	~40.83
Median	0.17	Random Forest (Original)	85.51%	76.73%	49.99%	60.54%	72.83%	~40.18
Mode	0.12	Random Forest (Original)	85.59%	76.75%	50.47%	60.90%	73.05%	~40.59
MICE	***10.06***	***Random Forest (Scaled)***	***88.34%***	***82.04%***	***60.90%***	***69.91%***	***78.54%***	***~53.12***
KNN	77.42	Random Forest (Scaled)	85.42%	76.39%	49.82%	60.31%	72.71%	~46.11
EM	11.03	Random Forest (Scaled)	88.37%	82.14%	60.95%	69.97%	78.58%	~53.17
Regression	10.47	Random Forest (Original)	88.33%	82.39%	60.44%	69.73%	78.37%	~53.42
Linear Interpolation	10.41	Random Forest (Scaled)	85.58%	77.07%	50.02%	60.67%	72.88%	~5252.95
Listwise Deletion	0.02	Random Forest (Scaled)	85.69%	76.40%	50.74%	60.98%	73.15%	~52.11

As the table shown above, ***the combination of MICE imputation technique with the Random Forest (Scaled) model*** yields the highest accuracy, precision, recall, F1-score, and ROC/AUC among the presented options. Additionally, it achieves these metrics with a reasonable computational time, making it the best choice for rain prediction in Australia among the listed techniques and models.

4. Analysis and Results

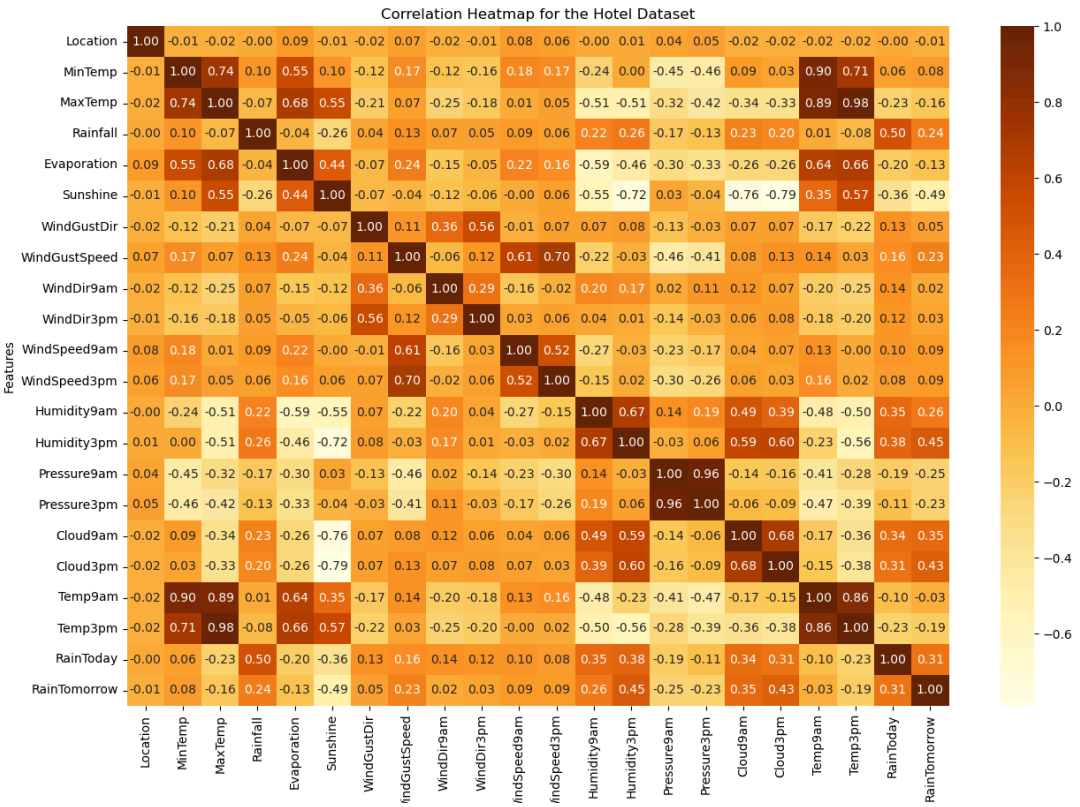
4.1 Exploratory Data Analysis

Having identified the MICE Imputation Technique with Random Forest Scaling as the optimal performer within this dataset, we want to focus on delving deeper into our analysis, using this technique and model.

(1) Correlation Heatmap for the Numerical Features

Now, we would like to generate a correlation heatmap, visualizing the relationships between different features. A correlation heatmap is a graphical representation of the correlation matrix, where each cell's color indicates the strength and direction of the correlation between two variables. Moreover, the correlation matrix shows the linear relationship between pairs of features in the dataset. The colormap 'YlOrBr' is applied to represent the strength of correlations, with darker shades indicating stronger correlations.

Figure 3: *Visualizing the Correlation Heatmap for the Numerical Features*

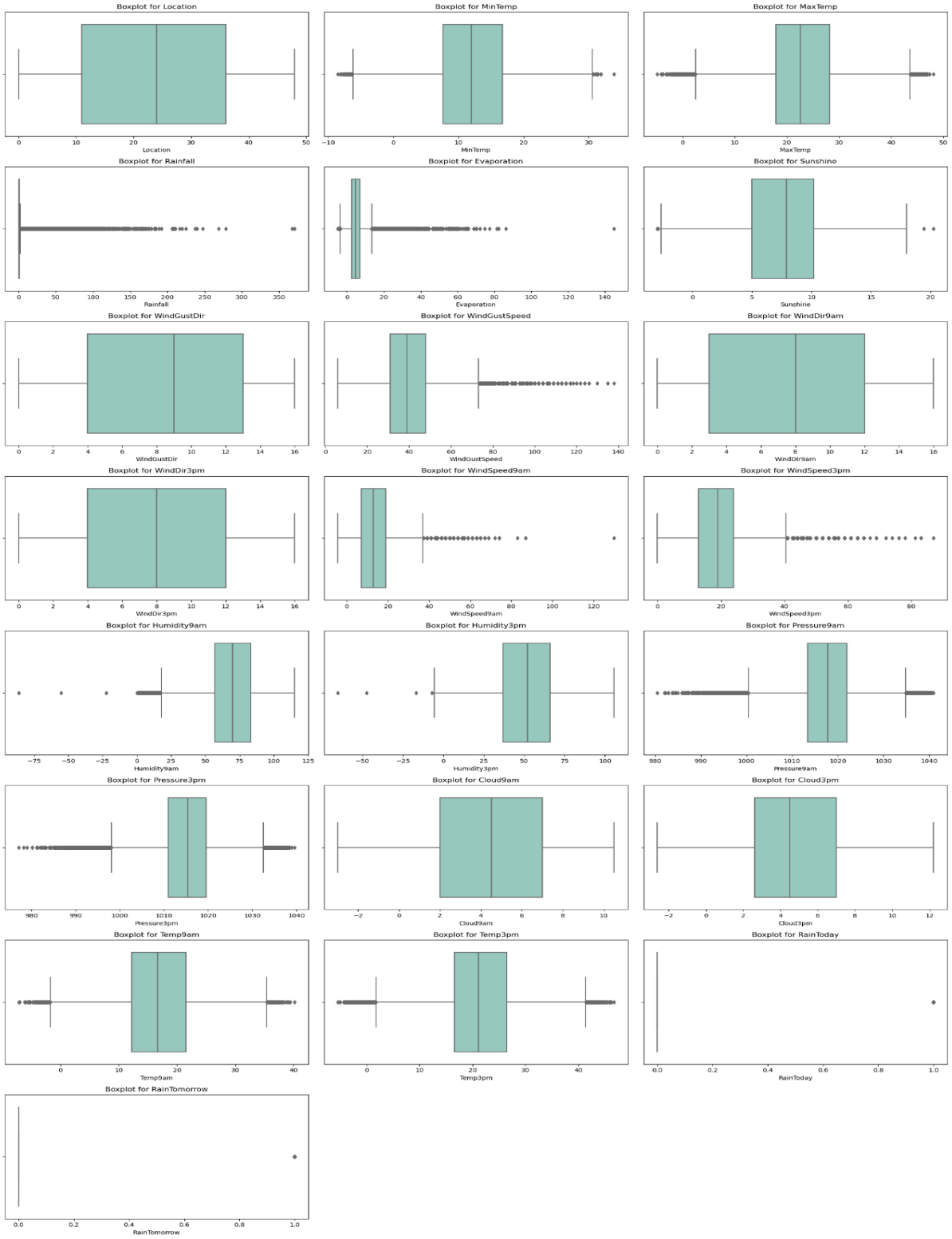


We can see that variables that show the status in different time of the day are strongly correlated with each other, which means that if the temperature is high at 9AM is it also expected to be high at 3PM, and vice versa.

(2) Boxplot of the Numerical Features

We now turn our attention to visualizing the distribution of numerical features within the dataset. This step is crucial, as it allows us to identify any outliers that may impact our overall predictive task. By visualizing boxplots, we want to gain insights into the variability and distribution of these features, paving the way for a more informed analysis.

Figure 4: *Visualizing the Boxplot of the Numerical Features*



We can see that for the rainfall, which is measured in millimeters, there are a lot of outliers but the furthest one is 300 mm, which could indicate a real day when there was that amount of rain in a specific area, and then for the wind speed, we can see that there are outliers until 80 km/h which could be logical. When looking at each of the variables, the outliers make sense, and therefore we do not want them removed.

(3) Histogram for the Numerical Features

Following our examination of boxplots, we now go into further analysis by visualizing the distribution of numerical features through histograms. This step is essential for assessing the skewness of the variables, providing insights into their distribution characteristics.

Figure 5: *Visualizing the Histogram for the Numerical Features*

We can see that most of the numerical features show a normal distribution, only for the three plots in the second row, we can see that they are sightly left-skewed.

4.2 Modelling

(1) Modeling Summary

We will now revisit the Random Forest Scaled model and additionally employ its feature selection variant to conduct a comparative analysis. Therefore, we employ the SelectKBest algorithm to identify the most influential features among the dataset's variables.

SelectKBest acts like a filter that sifts through a heap of data to identify the most crucial pieces of information. Picture having various features such as temperature, humidity, and wind speed, and wanting to pinpoint which ones are most valuable for predicting if it will rain tomorrow. SelectKBest helps in this regard. (*Optimizing Performance: SelectKBest for Efficient Feature Selection in Machine Learning*, 2023)

It evaluates each piece of information, assigning a score based on its relevance to predicting future rainfall. For instance, if high humidity frequently precedes rain, humidity would likely receive a high score. Once all features are scored, SelectKBest cherry-picks the top few (K) with the highest scores. (*Optimizing Performance: SelectKBest for Efficient Feature Selection in Machine Learning*, 2023)

Next, we use these top-ranking features to train a specialized computer program called a Random Forest. This program learns from the data to forecast whether it will rain tomorrow, based on factors like temperature, humidity, and wind speed. (*Optimizing Performance: SelectKBest for Efficient Feature Selection in Machine Learning*, 2023)

Following training with the selected features, we assess the Random Forest's performance in predicting rainfall.

Therefore, using the SelectKBest algorithm to find which are the ten best features of the 19 that the dataset has, we got these variables **'MaxTemp', 'Rainfall', 'WindGustSpeed', 'WindSpeed3pm', 'Humidity9am', 'Humidity3pm', 'Pressure9am', 'Pressure3pm' 'Temp3pm',** and **'RainToday'.**

Therefore, we ran the original model with only these best features, and we got these results of performance metrics:

Table 11: *Showing the Comparison of Model Performance between the Original Model of Random Forest Scaled and the Random Forest with Feature Selection Scaled*

Model	Accuracy	Precision	Recall	F1-Score	ROC/AUC	Computational Time (second)
Random Forest Classifier Scaled	88.41%	82.66%	60.60%	69.93%	78.48%	~56.00
Random Forest with Feature Selection Scaled	88.24%	81.77%	60.61%	69.62%	78.37%	~50.82

According to the results, the model didn't improve its performance by using the best variables. The reason why our model didn’t improve when using SelectKBest depends on these various factors:

- ***Relevance of Features:*** It may have happened due to the importance of the other features that were not used in the model. Those features could have had important information that the model needed to pick, therefore, removing them didn't help the model.

- ***Noise in Data:*** Sometimes, features may contain noise or irrelevant information that can degrade the performance of the model. SelectKBest can help by filtering out these noisy features, leading to an improvement in model performance. Therefore, in our case, this means that our features that we didn't select, didn't contain noise or irrelevant information.

- ***Interaction among Features:*** SelectKBest does not consider interactions among features. In our case, our features are quite correlated with each other, especially the ones that are the same but at a different hour. Consequently, selecting individual features may not capture the full predictive power of the data.

Therefore, we are going to only use the Random Forest Scaled with all the variables for further analysis, since it performs better.

(2) Best Model Performance

Cross-validation is a method used to evaluate the performance of a machine learning model by splitting the dataset into multiple subsets, or folds. In each iteration of cross-validation, one fold is used as the validation set, while the remaining folds are used as the training set. This process is repeated multiple times, with each fold serving as the validation set exactly once. Cross-validation helps to assess how well the model generalizes to unseen data by providing a more robust estimate of its performance compared to a single train-test split. (Alhamid, 2020)

For this reason, we're now interested in seeing how Random Forest with scaled data performs, both with and without cross-validation. Will the results remain consistent, or will there be differences?

Table 12: *Showing The Comparison of Model Performance between The Original Model of Random Forest Scaled and The Random Forest with Cross Validation*

Model	Accuracy	Precision	Recall	F1-Score	ROC/AUC
Random Forest Classifier Scaled	88.41%	82.66%	60.60%	69.93%	78.48%
Random Forest Classifier Scaled with Cross Validation	88.28%	81.81%	60.53%	69.58%	92.50%

Overall, from the table above, we can see that both models have similar accuracy and performance in terms of precision, recall, and F1-score, the second model with cross-validation demonstrates superior discrimination between classes, as evidenced by its higher ROC AUC score.

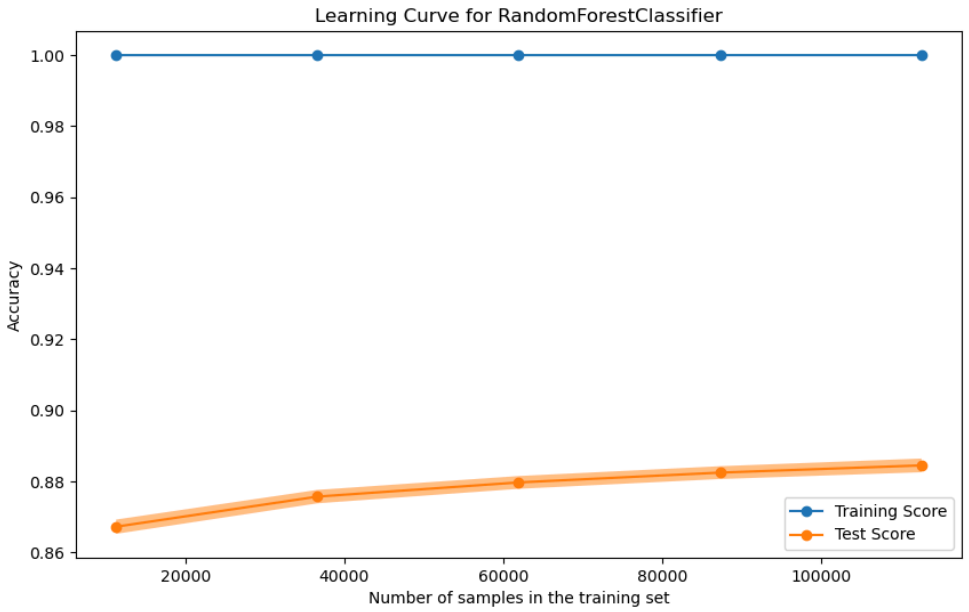
When the results of a model with and without cross-validation are almost the same, it means the model is consistent and doesn't rely heavily on how the data is split for validation. This suggests the model is stable and can generalize well to new data.

4.3 Additional Techniques

(1) Learning Curve

A learning curve is applied to illustrate how well a model performs based on the amount of training data. It helps identify learning issues like underfitting or overfitting and assesses dataset representativeness. By comparing training and validation scores across different training set sizes, learning curves reveal how much the model improves with more data and whether its limitations are due to bias or variance errors (Giola *et al.*, 2021). Now, let's delve into exploring the learning curve of our best-performing model.

Figure 6: *Visualizing The Learning Curve for Random Forest*



The learning curve plot, as shown here, reveals that the model achieves a perfect score when trained on 20,000 samples, indicating its ability to memorize the training data entirely. However, the most significant improvement in performance for the testing data occurs when using over 100,000 samples. Beyond this point, additional samples result in minimal enhancements in performance.

(2) Checking for Overfitting

Overfitting occurs when a model accurately captures the intricacies of the training data but struggles to generalize to new data from the same population, often because it learns patterns that are specific to the training data but not representative of the overall population. It can also refer to a model that is excessively complex for the given data and problem. Some define overfitting as the model learning noise present in the training data but not in the population. (Aliferis & Simon, 2024).

Now we want to check overfitting for the Random Forest Classifier Scaled model using different settings: max depth ranging from 1 to 20, and the number of estimators set at 50, 100, and 150. By testing various configurations, we aim to understand how the model's

performance changes with different complexities. This helps us identify the optimal balance between model complexity and generalization ability.

Figure 7: *Visualizing Random Forest Classifier with 50 n_estimators*

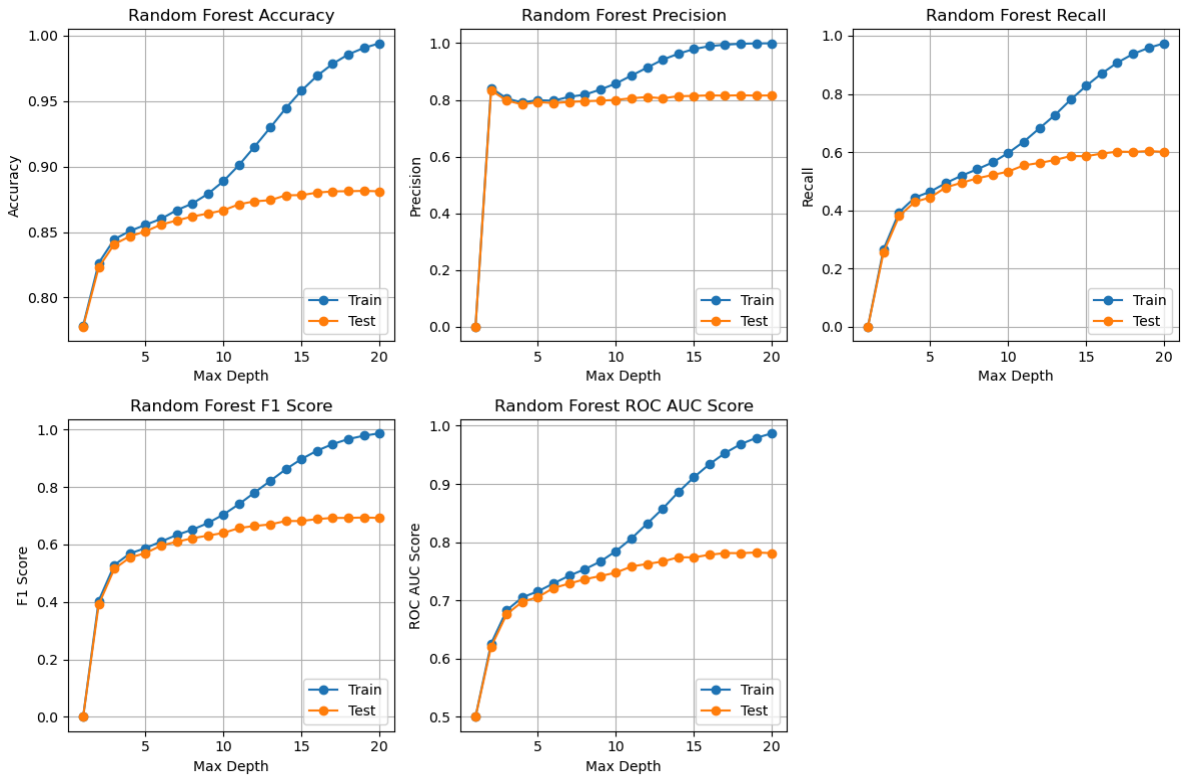


Figure 8: *Visualizing Random Forest Classifier with 100 n_estimators*

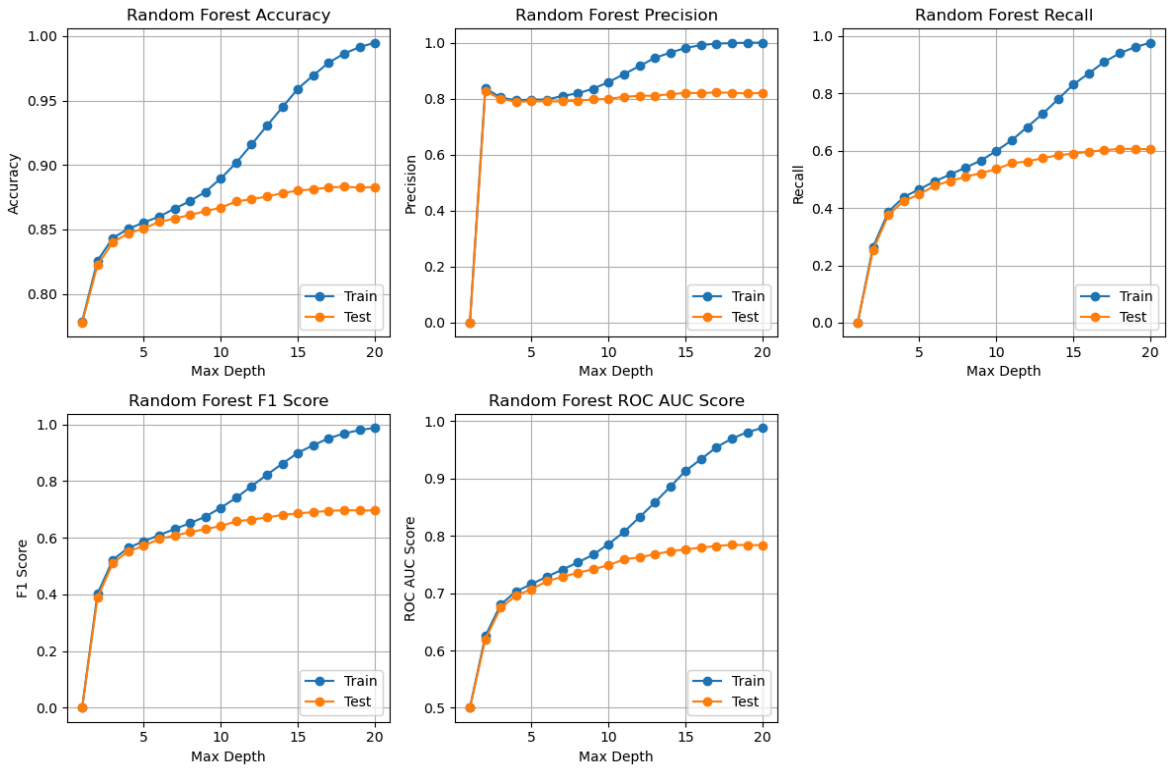
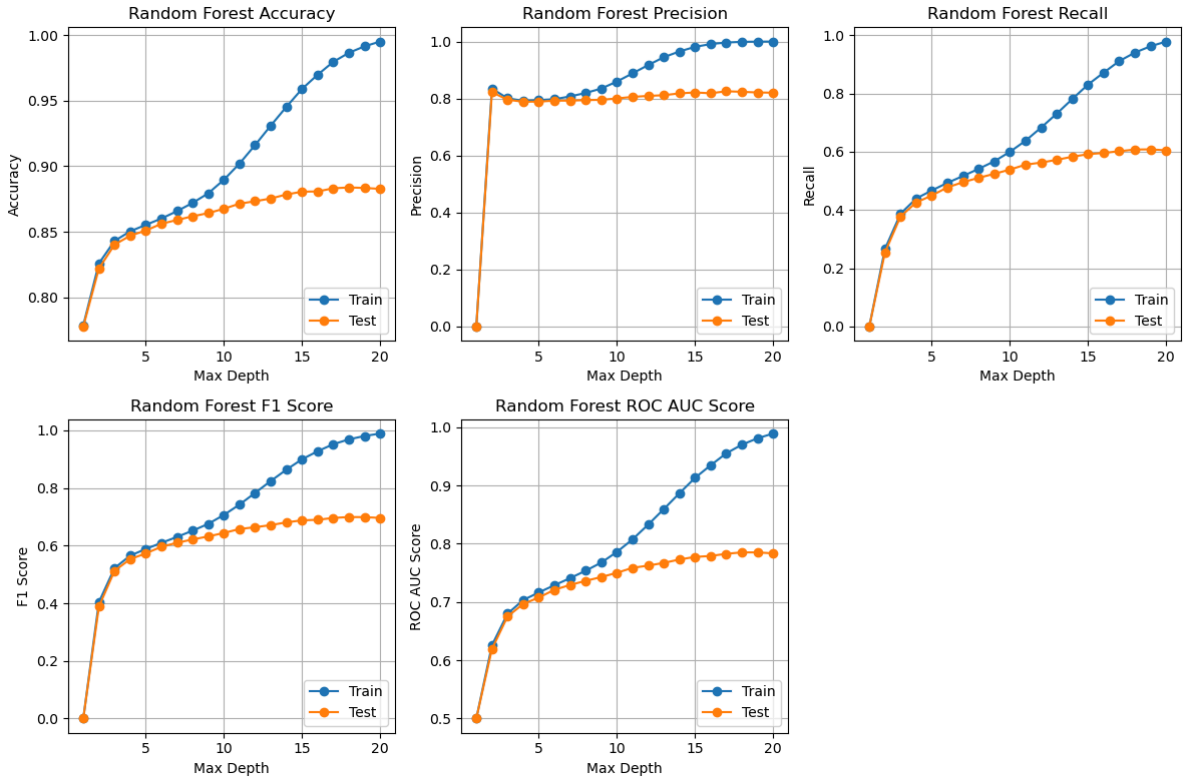


Figure 9: *Visualizing Random Forest Classifier with 150 n_estimators*



When exploring different combinations of max depth and number of estimators for the Random Forest Classifier, increasing the max depth generally led to improved performance metrics on the training set, including accuracy, precision, recall, F1-score, and ROC AUC score. However, the performance on the testing dataset showed fluctuations, with some max depths performing better than others. From the plots, it's evident that the training scores consistently improve with increasing max depth, but the testing scores fluctuate, indicating potential overfitting.

The optimal max depth appears to be in the range of 6 to 8, where the differences in performance metrics between different depths are minimal, suggesting a balance between model complexity and generalization. This range offers good performance on both the training and testing datasets while reducing the risk of overfitting.

Interestingly, varying the number of estimators 50, 100, and 150 in the Random Forest did not significantly impact the shape or trend of the learning curves. Despite differences in the number of trees in the forest, the overall behavior of the model, as reflected in the learning curves, remained consistent. This suggests that increasing the number of estimators beyond a certain point may not lead to substantial improvements in model performance. Therefore, it is very important to consider the trade-off between computational complexity and performance when selecting the number of estimators.

5. Reflection and Conclusion

5.1 Reflection

Best Model Performance: The combination of MICE imputation technique with the Random Forest (Scaled) model yielded the highest the highest values across most evaluation criteria, including accuracy, precision, F1-score, and ROC/AUC. Moreover, this model achieved superior performance while maintaining reasonable computational time, making it the best choice for rain prediction in Australia among the listed techniques and models.

Feature Selection Analysis: Employing the SelectKBest algorithm to identify the most influential features did not significantly improve model performance. Potential reasons for the lack of improvement include the relevance of excluded features, absence of noise in the data, and the complexity of feature interactions.

Cross-Validation: Models with and without cross-validation demonstrated similar accuracy and performance metrics, indicating stability and good generalization ability.

Learning Curve Analysis: The learning curve depicted the model's performance based on varying amounts of training data. The model achieved near-perfect scores when trained on a large amount of data, suggesting a high capacity to memorize the training set. Additional samples beyond a certain point showed minimal improvement in performance, indicating diminishing returns with increased training data.

Overfitting Analysis: Testing various configurations of the Random Forest Classifier revealed fluctuations in performance metrics on the testing dataset, indicating potential overfitting. Optimal model complexity was observed in the range of 6 to 8 for max depth, balancing between performance on training and testing datasets. On the other hand, the number of estimators did not significantly impact model performance beyond a certain threshold, highlighting the importance of considering the trade-off between computational complexity and performance.

5.2 Conclusion:

In this project, we effectively explored various techniques and models for rain prediction in Australia, focusing on data preprocessing, feature selection, and model evaluation.

Despite efforts to optimize model performance with some strategies, like feature selection, which did not lead to significant improvements, the results emphasize the importance of selecting appropriate techniques and considering model complexity in machine learning tasks.

The analysis provided insights into the importance of model stability, generalization ability, and the balance between complexity and performance in machine learning tasks.

Future research could explore additional feature engineering methods, alternative algorithms, or ensemble techniques to further improve rain prediction accuracy in Australia.

6. References

A. Picornell, J. Oteros, R. Ruiz-Mata, M. Recio, M.M. Trigo, M. Martínez-Bracero, B. Lara, A. Serrano-García, C. Galán, H. García-Mozo, P. Alcázar, R. Pérez-Badia, B. Cabezudo, J. Romero-Morte, J. Rojo. (2021). Methods for interpolating missing data in aerobiological databases. *Environmental Research.* https://doi.org/10.1016/j.envres.2021.111391

Alhamid, M. (2020, December 24). What is Cross-Validation? Testing your machine learning models with cross-validation. *Towards Data Science.* https://towardsdatascience.com/what-is-cross-validation-60c01f9d9e75

Ali, Aida & Shamsuddin, Siti Mariyam & Ralescu, Anca. (2015). Classification with class imbalance problem: A review. 7. 176-204. *UTM Big Data Centre, Ibnu Sina Institute for Scientific and Industrial Research Universiti Teknologi Malaysia* https://www.researchgate.net/publication/288228469_Classification_with_class_imbalance_problem_A_review

Aliferis, C., Simon, G. (2024). Overfitting, Underfitting, and General Model Overconfidence and Under-Performance Pitfalls and Best Practices in Machine Learning and AI. *Artificial Intelligence and Machine Learning in Health Care and Medical Sciences. Health Informatics. Springer, Cham.* https://doi.org/10.1007/978-3-031-39355-6_10

Alves, L. M. (2021, July 2). KNN (K Nearest Neighbors) and KNeighborsClassifier — What it is, how it works, and a practical example! https://luis-miguel-code.medium.com/knn-k-nearest-neighbors-and-kneighborsclassifier-what-it-is-how-it-works-and-a-practical-914ec089e467

Azur, M. J., Stuart, E. A., Frangakis, C., & Leaf, P. J. (2011). Multiple imputation by chained equations: what is it and how does it work? *International journal of methods in psychiatric research, 20(1), 40–49*. https://doi.org/10.1002/mpr.329

Blagec, K., Dorffner, G., Moradi, M., & Samwald, M. (2020). A critical analysis of metrics used for measuring progress in artificial intelligence. *Section for Artificial Intelligence and Decision Support; Center for Medical Statistics, Informatics, and Intelligent Systems; Medical University of Vienna.* https://arxiv.org/pdf/2008.02577

Gabr, M., Ibrahim, Y., Mostafa H., & Doaa S. E. (2023). Effect of Missing Data Types and Imputation Methods on Supervised Classifiers: An Evaluation Study. *Big Data and Cognitive Computing 7, no. 1: 55.* https://doi.org/10.3390/bdcc7010055

Giola, C., Danti, P., & Magnani, S. (2021, July 13). Learning curves: A novel approach for robustness improvement of load forecasting. *MDPI.* https://www.mdpi.com/2673-4591/5/1/38#metrics

Hameed, W. & Ali, N. (2023). Missing value imputation Techniques: A Survey. *UHD Journal of Science and Technology.* https://www.researchgate.net/publication/369674417_Missing_value_imputation_Techniques_A_Survey

IBM. (2022). What Is Logistic Regression? *IBM.* https://www.ibm.com/topics/logistic-regression

IBM. (2023a). What is a Decision Tree | *IBM.* https://www.ibm.com/topics/decision-trees

IBM. (2023b). What is Random Forest? | *IBM*. https://www.ibm.com/topics/random-forest

Jason B. (2018, November 20). A Gentle Introduction to the Gradient Boosting Algorithm for Machine Learning. Machine Learning Mastery. https://machinelearningmastery.com/gentle-introduction-gradient-boosting-algorithm-machine-learning/

Jiaxu P., Jungpil H., Ke-Wei H. (2022) Handling Missing Values in Information Systems Research: A Review of Methods and Assumptions. Information Systems Research 34(1): 5-26. https://doi.org/10.1287/isre.2022.1104

Kang H. (2013). The prevention and handling of the missing data. Korean Journal of Anesthesiology, 64(5), 402–406. https://doi.org/10.4097/kjae.2013.64.5.402

Kavya, D. (2023, February 15). Optimizing Performance: SelectKBest for Efficient Feature Selection in Machine Learning. *Medium.* https://medium.com/@Kavya2099/optimizing-performance-selectkbest-for-efficient-feature-selection-in-machine-learning-3b635905ed48

Makaba, T. & Dogo, E. (2019). A Comparison of Strategies for Missing Values in Data on Machine Learning Classification Algorithms. *International Multidisciplinary Information Technology and Engineering Conference (IMITEC), Vanderbijlpark, South Africa, pp. 1-7.* https://ieeexplore.ieee.org/document/9015889

Ribeiro, D. (2023). Missing values in data analysis: A comprehensive guide. *Medium.* https://medium.com/data-science-as-a-better-idea/missing-values-in-data-analysis-a-comprehensive-guide-2151bc2e8579

Padgett, C. & Skilbeck, C. & Summers, M. (2014). Missing Data: The Importance and Impact of Missing Data from Clinical Research. *Brain Impairment.* https://www.researchgate.net/publication/262036960_Missing_Data_The_Importance_and_Impact_of_Missing_Data_from_Clinical_Research

Salgado, C.M., Azevedo, C., Proença, H., & Vieira, S.M. (2016). Missing Data. In: Secondary Analysis of Electronic Health Records. *Springer, Cham.* https://doi.org/10.1007/978-3-319-43742-2_13

Song, Q. & Shepperd, M. (2007). Missing Data Imputation Techniques. *International Journal of Business Intelligence and Data Mining.* https://www.researchgate.net/publication/220579612_Missing_Data_Imputation_Techniques

Torres, M. & Juan, A. (2014). Comparison of imputation methods for handling missing categorical data with univariate patterns. *Revista de Métodos Cuantitativos para la Economía y la Empresa, ISSN 1886-516X, Universidad Pablo de Olavide, Sevilla, Vol.17, pp. 101-120.* https://hdl.handle.net/10419/113873

Wizards, D. S. (2023, July 7). Understanding the AdaBoost Algorithm. *Medium.* https://medium.com/@datasciencewizards/understanding-the-adaboost-algorithm-2e9344d83d9b