



**National University of  
Science and Technology**  
Think in Other Terms



**Faculty of Applied Science**

**Department of Operations Research and Statistics**

---

**Assessing the Effectiveness of Supervised  
Machine Learning Models in Short-Term  
Weather Forecasting**

---

Name	George Amen Kakweza
Student Number	N02019126J
Supervisor	Miss E. Nyakujipa

This document is submitted in partial fulfilment of the requirements of a Bachelor of Science degree in Operations Research and Statistics at the National University of Science and Technology

May 2024

# Research Project Approval Form

The undersigned certifies that he has read and recommends to the Department of Statistics and Operations Research for acceptance, a research project entitled:

**Assessing the Effectiveness of Supervised Machine Learning Models in Short-Term Weather Forecasting**

Submitted by:

Full Name: **George Amen Kakweza**

Student Number: **N02019126J**

in partial fulfilment of the requirements of BSc (Hons) Degree in Operations Research and Statistics

Supervisor Name: Miss E. Nyakujipa

Supervisor's Signature .....

Date .....

## Declaration

I, George Amen Kakweza, hereby declare that this research project titled "Assessing the Effectiveness of Supervised Machine Learning Models in Short-Term Weather Forecasting" is my own work and that all sources used have been acknowledged. This document is submitted in partial fulfilment of the requirements for the Bachelor of Science degree in Operations Research and Statistics at the National University of Science and Technology.

---

Signature

---

Date

## Abstract

Weather forecasting is an important activity used in the fields of agriculture, transportation, emergency management, and so on. Conventional methods use physical knowledge and empirical information to forecast weather. However, a huge opportunity exists to leverage machine learning to enhance weather forecasting. Machine learning tactics may be taught to predict specifics such as temperature, humidity, wind speed, and atmospheric pressure based on weather information from previous years. The research project focused primarily on assessing the effectiveness of supervised machine learning models in short-term weather forecasting. Data was gathered from Visual Crossing, a website containing weather data for various locations, and then pre-processed by choosing only a few significant weather characteristics. Finally, Supervised Machine Learning models namely, Decision Trees, Random Forest, and gradient boosting were utilized to forecast the weather tomorrow, that is, whether it will rain or not. The accuracy of these machine learning models was then evaluated through the use of various performance metrics such as accuracy and Area Under the ROC Curve. This comparative analysis was critical in determining the most efficient and effective algorithm for short-term weather predictions. Ultimately, the research project will enable the improvement of future weather forecasting predictions. This forecast can then be used to make informed decisions by the relevant authorities.

## **Acknowledgements**

I would like to start by thanking the Lord God Almighty, for without Him, nothing is possible. Secondly, I would love to give special mention to the entire Operations Research staff and student body for their unwavering support in assisting me with the research project. Giving special thanks to my supervisor, Miss E. Nyakujipa, for making time to review my overall progress of the research project from the very beginning to completion. Last but not least, I would like to thank myself for putting in the hard work and never giving up.

# Contents

<b>Declaration</b>	<b>ii</b>
<b>Abstract</b>	<b>iii</b>
<b>Acknowledgements</b>	<b>iv</b>
<b>Contents</b>	<b>vii</b>
<b>List of Figures</b>	<b>ix</b>
<b>List of Tables</b>	<b>x</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Background . . . . .	1
1.2 Problem Statement . . . . .	2
1.3 Research Aim . . . . .	2
1.4 Objectives . . . . .	2
1.5 Significance . . . . .	2
1.6 Limitations and Delimitations . . . . .	3
<b>2 Literature Review</b>	<b>4</b>
2.1 Traditional Weather Forecasting Techniques . . . . .	4
2.2 Modern Weather Forecasting Techniques . . . . .	4
2.2.1 Synoptic Method . . . . .	4
2.2.2 Statistical Method . . . . .	4
2.2.3 Numerical Weather Prediction Techniques . . . . .	5
2.3 The Role of Machine Learning (ML) in Weather Forecasting . . . . .	5
2.4 Related Studies . . . . .	7
2.5 Study Area . . . . .	8
<b>3 Methodology</b>	<b>10</b>
3.1 Introduction . . . . .	10
3.2 Data Collection . . . . .	11
3.3 Data Preprocessing . . . . .	12

3.4	Data Transformation . . . . .	15
3.5	Exploratory Data Analysis . . . . .	16
3.6	Machine Learning . . . . .	17
3.6.1	Data Split . . . . .	17
3.6.2	Decision Tree . . . . .	18
3.6.3	Random Forest . . . . .	21
3.6.4	Gradient Boosting . . . . .	23
3.7	Hyperparameter Tuning . . . . .	25
3.8	Model Evaluation and Validation . . . . .	27
3.8.1	Confusion Matrix . . . . .	27
3.8.2	ROC Curve (Receiver Operating Characteristic) . . . . .	29
3.8.3	Area under the ROC Curve (AUC) . . . . .	30
<b>4</b>	<b>Data Analysis</b>	<b>32</b>
4.1	Data Collection . . . . .	32
4.1.1	Importing xlsx data into R . . . . .	34
4.2	Data Preprocessing . . . . .	34
4.2.1	Variable Selection . . . . .	34
4.2.2	Dealing with Missing Data . . . . .	35
4.2.3	Further Variable Selection . . . . .	35
4.3	Data Transformation . . . . .	38
4.4	Exploratory Data Analysis . . . . .	39
4.5	Model Building . . . . .	43
4.5.1	Data Partition . . . . .	43
4.5.2	Conditional Inferencing Trees (CTREE) . . . . .	43
4.5.3	Random Forest . . . . .	49
4.5.4	Gradient Boosting with XGBoost Model . . . . .	53
4.6	Discussion of the results . . . . .	55
4.6.1	Comparative Analysis of Model Performance . . . . .	55
4.6.2	Discussion on other author findings . . . . .	57
<b>5</b>	<b>Conclusions and Recommendations</b>	<b>59</b>
5.1	Conclusions . . . . .	59
5.2	Recommendations . . . . .	59

5.3 Further Research Study . . . . .	60
--------------------------------------	----



## List of Figures

1	Map showing Bulawayo Province . . . . .	9
2	Data Science Process Lau 2019 . . . . .	10
3	Types of Machine Learning . . . . .	17
4	Data Split: Testing and Training . . . . .	18
5	Structure of a Decision Tree. The365team (2024) . . . . .	19
6	Random Forest graphical representation. Koehrsen (n.d.) . . . . .	22
7	Confusion Matrix. Novaes et al. (2021) . . . . .	28
8	ROC Curve. user2149631 (n.d.) . . . . .	29
9	Data Analysis Overview. Turing (n.d.) . . . . .	32
10	Data Head Overview of the Imported Data in R . . . . .	34
11	Correlation Heat Map . . . . .	36
12	Correlation Level . . . . .	37
13	Summary Statistics of Variables . . . . .	39
14	Precipitation vs Rain Tomorrow . . . . .	39
15	Rain Today vs Rain Tomorrow . . . . .	40
16	Precipitation Cover vs Rain Tomorrow . . . . .	40
17	Cloud Cover vs Rain Tomorrow . . . . .	41
18	Humidity vs Rain Tomorrow . . . . .	41
19	Dew vs Rain Tomorrow . . . . .	42
20	Temperature vs Rain Tomorrow . . . . .	42
21	CTREE Decision Tree . . . . .	44
22	Pruned CTREE Decision Tree . . . . .	45
23	CTREE Confusion matrix and Statistics . . . . .	47
24	Area Under the ROC Curve (CTREE) . . . . .	48
25	RF Model Accuracy . . . . .	49
26	Random Forest Model . . . . .	49
27	Importance of each variable in the Random Forest Model . . . . .	50
28	Random Forest Confusion Matrix and Statistics . . . . .	51
29	Random Forest ROC Curve . . . . .	52
30	XGBoost Confusion Matrix . . . . .	54

31	XGBoost ROC Curve . . . . .	55
32	Displays Performance comparison of Model prediction. . . . .	56
33	ROC Curve Comparison . . . . .	57

List of Tables

1	Table Head Overview . . . . .	33
2	Weather Columns . . . . .	33
3	Displays Performance comparison of Model prediction. . . . .	56

# 1 Introduction

According to the Oxford University Press (2010) weather forecasting is an analysis of the state of the weather in an area with an assessment of likely developments. In other words, it is the process of analysing the current and past atmospheric conditions with the aim of predicting the future atmospheric condition. According to Cahir (2024a) weather forecasting includes predictions of changes on Earth’s surface caused by atmospheric conditions—e.g., snow and ice cover, storm tides, and floods. These forecasts play a crucial role in various aspects of human life and society, contributing to safety, planning, and decision-making in numerous sectors. Meteorologists collect weather observations on temperature, air pressure, humidity, precipitation, wind speed and more, from weather stations, weather satellites and weather balloons all over the world. They use computer models to predict the weather, yet they still have trouble correctly predicting the weather over a period of a few days, sometimes they don’t even get it right over a 24-hour period. Why does this happen? Well, their ability to predict the weather according to Let’s Talk Science (2023) is limited by three factors: the amount of available data, the time available to analyse it, and the complexity of weather events. These factors combined together makes it incredibly difficult to forecast the weather. By identifying patterns in historical data, Machine Learning models can predict weather events (like storms, temperature changes, and rainfall) with remarkable precision – even in highly complex and dynamic systems. Reilly (2023)

## 1.1 Background

”The term machine learning means to enable machines to learn without programming them explicitly. There are four general machine learning methods: (1) supervised, (2) unsupervised, (3) semi-supervised, and (4) reinforcement learning methods. The objectives of machine learning are to enable machines to make predictions, perform clustering, extract association rules, or make decisions from a given dataset.” Mohammed et al. (2016). Machine learning (ML) has grown rapidly in recent years in the context of data analysis and computing that typically allows the applications to function in an intelligent manner. ”Supervised Learning is a machine learning paradigm for acquiring the input-output relationship information of a system based on a given set of paired input-output training samples. As the output is regarded as the label of the input data or the supervision, an input-output training sample is also called labelled training data, or supervised data. Occasionally, it is also referred to as Learning with a Teacher (Haykin 1998), Learning from Labelled Data, or Inductive Machine Learning (Kotsiantis, 2007).” Liu & Wu (2012). Supervised machine learning has been and continues to be applied in numerous fields such as customer retention, fraud detection, speech recognition, sperm detection and many more. The application of this model has proven to be a success. In an article

by Chantry et al. (2023) it was observed that Machine learning (ML) has been gradually integrated into weather forecasting over the years, but it saw a significant rise in attention and application from February 2022 through April 2023. A number of studies, mostly from major tech companies like NVIDIA, Huawei, and Google DeepMind, showed how quickly the quality of ML-based weather forecasts was improving. New contributions to the field are currently being produced every few months. But the question is, are these machine learning (ML)-based models generating physically sound and meteorologically significant forecasts? Hence it is apparent that we assess their effectiveness in weather forecasting.

## **1.2 Problem Statement**

”Weather forecasting is a challenging task because it is dependent on a variety of factors such as wind speed, wind direction, global warming, and so on”. Jani et al. (2022). According to Conti (2024) traditional methods that rely on the weather-governing physics equations translated into algorithms are time consuming, laborious and costly, resulting in speed-accuracy trade-offs. Supervised machine learning models offer a promising approach to enhance short-term weather forecasting by leveraging historical data and learning patterns from past weather observations.

## **1.3 Research Aim**

To assess the effectiveness of Supervised Machine Learning Models in short-term weather forecasting.

## **1.4 Objectives**

- To perform short-term weather forecasting using supervised machine learning models.
- To evaluate the performance of supervised machine learning models in short-term weather forecasting.
- To Identify the most effective approaches in short-term weather forecasting.

## **1.5 Significance**

Poor weather forecasts pose a serious risk on various sectors of the economy and individuals. Short-term weather forecasts can lead to better decision-making in agriculture, transportation planning, and disaster preparedness. According to Behrer (2023) more accurate forecasts can help reduce economic losses due to weather-related disruptions and improve overall societal resilience. Making use of Machine Learning to forecast the weather will benefit the the lives of people, businesses and the economy as a whole.

## 1.6 Limitations and Delimitations

### Limitations of the study:

- Data Constraints; The availability, quality and reliability of weather data might be limited impacting the accuracy and applicability of the results.
- Model Assumptions; The forecasting models utilized in the research could have limitations or simplifications that influence their effectiveness and predictive capabilities..
- External Influences; Uncontrollable factors, like shifts in weather patterns or unexpected events may introduce uncertainty or bias into the study outcomes.
- Sample Size Limitations; The size and representativeness of the sample used in the research may be restricted, potentially affecting the reliability and generalizability of the findings.

### Delimitations of the study:

- Geographic Focus; This study is centered on Bulawayo, Zimbabwe so its outcomes may not be universally applicable.
- Forecasting Duration; The research concentrates on short term weather predictions (one day while excluding longer term forecasts.
- Prediction Target; The focus is on forecasting rain probability for tomorrow.
- Methodological Strategy; Specific methodological approaches will be employed in this study while excluding methods due, to feasibility and expertise limitations.

## Chapter Summary

This chapter presented the focus of the study, beginning with introduction and background information regarding the problem under investigation. The main aim and objectives of the study were also outlined.

## 2 Literature Review

### 2.1 Traditional Weather Forecasting Techniques

Traditional weather forecasting techniques are a rich tapestry of knowledge passed down through generations, particularly within indigenous and pastoral communities. These methods often involve keen observations of the natural environment, including animal behaviour, plant growth, and atmospheric conditions. Balehegn et al. (2019)

Traditional weather forecasting techniques have several drawbacks, including limited accuracy due to their dependence on natural observations, which may not always agree with scientific principles. These methods often struggle to handle complicated modern weather systems and provide forecasts beyond a few days or weeks. Moreover, their reliance on local knowledge and subjective interpretation introduces inconsistency and uncertainty into the forecasting process. Moreover, traditional techniques may not fully leverage modern technology and data sources, hindering their ability to provide accurate and timely forecasts. While these methods offer traditional importance and historical insights, combining them with modern scientific approaches is important for improving the reliability and usefulness of weather predictions.

### 2.2 Modern Weather Forecasting Techniques

Weather forecasting methods encompass a range of techniques and approaches used to predict future weather conditions. These methods can vary in complexity, accuracy, and the time frame over which they make predictions.

#### 2.2.1 Synoptic Method

In synoptic meteorology, simultaneous observations for a specific time are plotted on a map for a broad area whereby a general view of the weather in that region is gained. Cahir (2024*b*) A systematic study of recent weather forecasts from a wide area is used in this method of weather forecasting. Present weather conditions are linked to comparable scenarios in the past, and predictions are based on the premise that the current scenario would behave similarly to the analogous situation in the past. Vedantu (2024)

#### 2.2.2 Statistical Method

Regression equations, machine learning algorithms or other advanced relationships are formed between various weather elements and the subsequent climate in this method of weather forecasting. Predictions or weather criteria are usually chosen based on a potential physical interaction with the predictions. Vedantu

(2024) These methods rely on the relationship between various weather variables and use statistical models to forecast weather phenomena.

### **2.2.3 Numerical Weather Prediction Techniques**

Numerical Weather Prediction (NWP) models are sophisticated tools that use mathematical representations of the atmosphere and oceans to forecast the weather. These models apply systems of differential equations based on physical principles, such as fluid motion, thermodynamics, radiative transfer, and chemistry. They operate on a three-dimensional grid that covers the globe, calculating various atmospheric properties like winds, heat transfer, solar radiation, relative humidity, and phase changes of water within each grid cell. for Environmental Information (2023)

## **2.3 The Role of Machine Learning (ML) in Weather Forecasting**

The accuracy of weather forecasting models can be greatly improved with the use of machine learning. Even in extremely complex and dynamic systems, machine learning (ML) models can predict weather occurrences (such as storms, temperature fluctuations, and rainfall) with surprising precision by finding patterns in past data. The ability of ML to be trained on a variety of data sources, including as radar, satellite, and weather station data, is a major factor in its efficacy. These models can also include additional data sources such as environmental sensors, crowdsourcing observations, and social media. By providing models with this data, we may help them produce more accurate predictions by helping them comprehend the relationship between various weather factors. . These data sources can also be used to validate and improve the accuracy of the models by comparing the model's predictions to the actual weather conditions observed in the real world. ML models can analyze vast amounts of data in real-time, allowing for more frequent and precise forecasts. They also update quickly when new information is received. Reilly (2023)

### **Decision Tree**

When it comes to predicting short term weather decision tree classifiers are used to forecast weather conditions, like sunny, cloudy or rainy based on factors such as temperature, humidity, wind speed, atmospheric pressure and cloud cover. These classifiers are valued for their simplicity and clarity in handling both categorical data. Their easy to understand rules make them beneficial for meteorologists in interpreting weather forecasts

Additionally combining decision tree classifiers with techniques like Random Forest can enhance classification accuracy and reliability in short term weather forecasting. By harnessing the strengths of decision trees



through methods these models can provide more dependable weather predictions by mitigating individual biases and uncertainties.

In essence decision tree classifiers offer an understandable method for predicting short term weather patterns. They play a role in offering insights, into weather trends and aiding decision making across different industries.

## **Random Forest**

Random Forest is a machine learning technique known for its flexibility and efficiency, in sorting tasks (Breiman, 2001). Acting as a learning approach Random Forest merges the abilities of numerous decision trees to yield reliable sorting outcomes. Each decision tree within the Random Forest group is trained separately on a subset of training data and characteristics to prevent overfitting (Breiman, 2001).

In the domain of short term weather prediction Random Forest can also be used for sorting assignments. For instance it can anticipate weather events like rain, snow or clear skies based on factors such as temperature, humidity, wind speed and air pressure. By utilizing weather data and ensemble strategies Random Forest models can effectively grasp connections between input factors and weather conditions to ensure accurate sorting results (Wang et al., 2016).

Moreover incorporating techniques like Random Forest in weather forecasting plays a role in enhancing the dependability of forecasts essential for decision making across sectors, like agriculture, transportation and crisis management (Wang et al. 2016). Random Forest is a tool that can improve the precision and effectiveness of categorizing tasks, in short term weather prediction scenarios.

## **Gradient Boosting**

Gradient boosting is a method, in machine learning that is commonly used for tasks involving classification and regression especially when the data involves nonlinear relationships. The technique operates by adding learners, decision trees to the model one after the other. Each new learner focuses on correcting the mistakes made by its predecessors leading to a reduction in prediction errors and the development of precise models (Friedman, 2001).

In classification scenarios gradient boosting constructs a group of decision trees where each tree aims to rectify the errors of trees. Throughout training the algorithm fine tunes a loss function by trees to the residuals from preceding iterations. The final prediction is generated by combining the predictions of all trees within the group.

For short term weather forecasting purposes gradient boosting can be utilized to forecast outcomes like weather conditions (e.g. sunny cloudy rainy) based on meteorological characteristics. By utilizing weather

data and ensemble methods such as gradient boosting models can effectively capture relationships between input features and weather patterns, for accurate forecasts (El Bouabidi et al., 2020). Using gradient boosting in weather prediction has proven to enhance the accuracy and dependability of forecasts making it a valuable tool, for short term weather predictions (El Bouabidi et al., 2020). By leveraging the strengths of decision trees and refining prediction errors in a step by step manner gradient boosting allows for the creation of models that deliver timely and precise forecasts.

In essence gradient boosting stands out as a machine learning method for classification purposes in short term weather forecasting. Through constructing groups of decision trees and refining prediction errors gradient boosting models can adeptly capture correlations, within weather data to produce reliable forecasts.

## 2.4 Related Studies

The paper titled "A Multiple Linear Regression Based Model for Average Temperature Prediction of a Day" by Gupta, Mittal, Rikhari, and Singh (2019) aimed to predict weather conditions using past meteorological data and features through the Multiple Linear Regression Model. The model's performance was assessed, and conclusions were drawn. The Multiple Linear Regression algorithm was employed for model training, along with a feature selection method to identify decisive features from the initial dataset. The dataset, sourced from Weather Underground's API web service, comprised 997 instances, each storing the mean temperature of a day and weather details for the preceding three days. To select linear features, the Pandas library in Python was utilized to calculate correlation coefficients ( $r$ ), while the Stats Models Library provided P-values for feature significance. Following feature selection, seven features were retained for input into the multiple linear regression model, which was trained using Sklearn, a Python library offering various machine learning algorithms. Model accuracy was assessed by predicting temperatures for test set instances and plotting a scatter plot between predicted and actual values using Matplotlib in Python. The scatter plot closely resembled the line  $y = x$ , indicating good accuracy. Despite providing a foundation for predicting average temperature with linear regression and feature selection, the methodology lacked thoroughness in model evaluation, validation, and consideration of alternative approaches. Addressing these deficiencies could enhance the analysis's robustness and reliability.

In another study by Datta et al. (2019), machine learning algorithms were utilized for weather prediction, yielding positive outcomes and presenting an alternative to traditional meteorological approaches. The study highlighted the effectiveness of machine learning algorithms in forecasting various weather phenomena, such as rain, thunderstorms, snow, and fog. Specifically, gradient boosting classifier and artificial neural network

algorithms were employed. Following a comparison of the results between these two models, the author concluded their suitability for such applications. Furthermore, the study suggested that the Back Propagation Algorithm could also predict weather and be applied to similar weather forecasting tasks. It was noted that further improvement in the results of these models could be achieved by proper preprocessing of the dataset at an early stage.

In another research conducted by Jakaria et al. (2018), the study focused on predicting continuous numeric values, specifically temperature, using regression techniques. The findings indicated that Random Forest Regression (RFR) outperformed other regression methods due to its ensemble approach, which aggregates multiple decision trees for decision-making. Additionally, the study compared RFR with several other state-of-the-art machine learning (ML) techniques, including Ridge Regression (Ridge), Support Vector Regression (SVR), Multi-layer Perceptron Regression (MLPR), and Extra-Tree Regression (ETR). The research presented a technology aimed at utilizing machine learning techniques for weather forecasting. Unlike traditional physical models, machine learning models offer simplicity and efficiency, requiring fewer computational resources and being adaptable to various computing platforms, including mobile devices. Evaluation results demonstrated that these machine learning models could accurately predict weather features, effectively competing with traditional models. Moreover, the study incorporated historical data from surrounding areas to enhance the accuracy of weather predictions for a specific area, showcasing the effectiveness of utilizing broader datasets

## 2.5 Study Area

The research is going to be conducted on the Bulawayo, Zimbabwe weather dataset obtained from a reliable online source Visual crossing. Bulawayo is the second-largest city in Zimbabwe and the largest city in the country's Matabeleland region. It has a rich history and is known for its significant role in the nation's industrial, cultural, and economic activities. According to the World Population Review, Bulawayo's 2024 population is now estimated at 658,028. In 1950, the population of Bulawayo was 91,635. Bulawayo has grown by 8,519 in the last year, which represents a 1.31 percent annual change. These population estimates and projections come from the latest revision of the UN World Urbanization Prospects. These estimates represent the Urban agglomeration of Bulawayo, which typically includes Bulawayo's population in addition to adjacent suburban areas. Bulawayo covers an area of about 1,707 square kilometres (659 square miles) in the western part of the country, along the Matsheumhlope River.

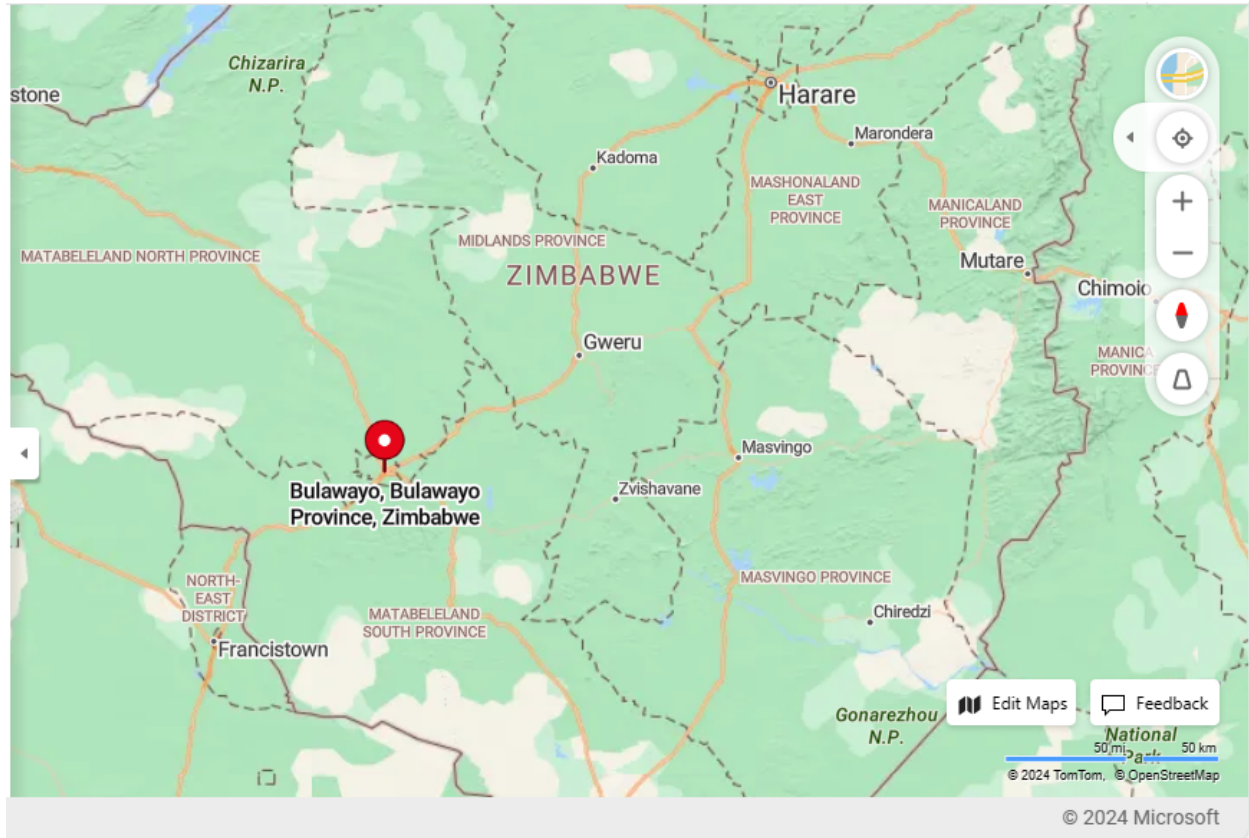


Figure 1: Map showing Bulawayo Province

The climate of Bulawayo is sub-tropical, tempered by altitude, with a hot, rainy season from November to March and a long dry season from April to October, within which there is a cool period from May to August. In the latter, at night the temperature can drop to the freezing point. During the day, it can get very hot from September to April.

## Chapter Summary

It was seen that machine learning was the most efficient way to forecast the weather compared to traditional methods. However, even though machine learning algorithms are a great tool for weather forecasting it is clear that machine combined with other forecasting methods produces more accurate results. Furthermore, domain expertise are also an essential tool in fine tuning the models to meet a particular need as results inevitably vary for different geographical locations.

## 3 Methodology

### 3.1 Introduction

In this section I am going to provide a clear and transparent explanation of how the research was conducted. I shall outline the procedures, techniques, and tools that were employed to answer the research questions.

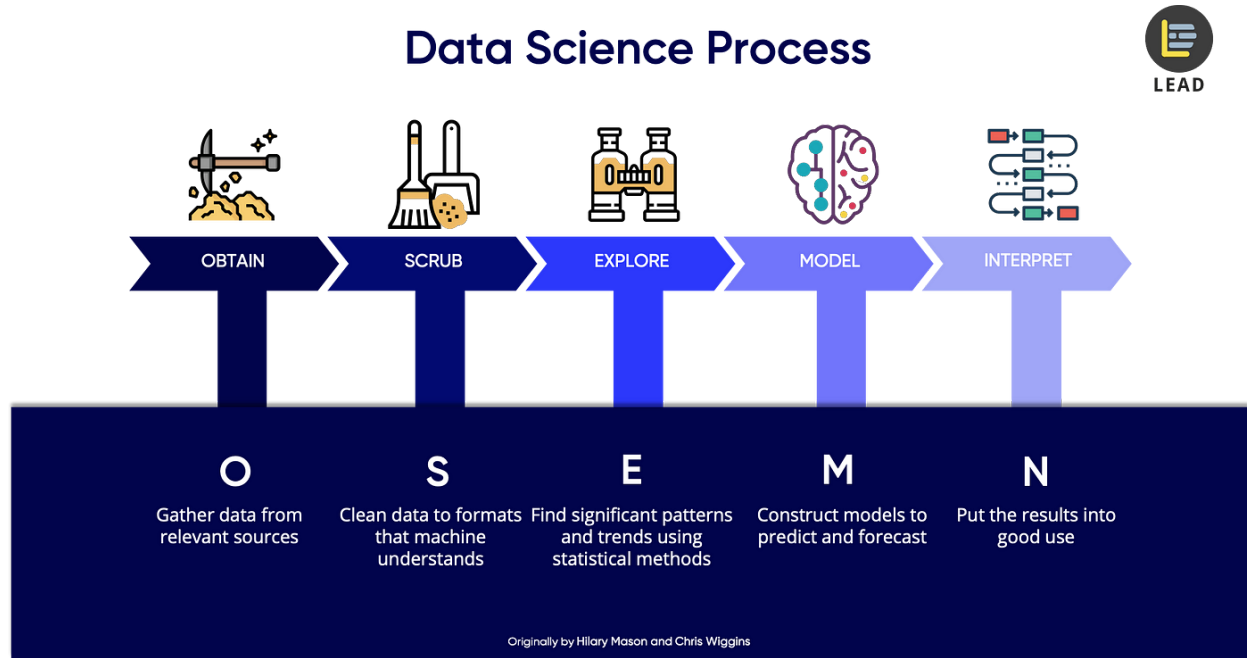


Figure 2: Data Science Process Lau 2019

Predicting weather is a complex task that involves analysing large amounts of data from various sources. The Data science process plays a crucial role in collecting, processing, analysing and modelling this data to make accurate weather forecasts. Presented below is a summary of the process:

1. Data Collection: is a systematic process of gathering observations or measurements for research purposes. Whether you're conducting research for business, academic, or governmental reasons, data collection allows you to gain first hand knowledge and original insights into your research problem.
2. Data preprocessing: is a data mining technique that involves transforming raw data into an understandable format. Real-world data is often incomplete, inconsistent, and/or lacking in certain behaviour or trends, and is likely to contain many errors. Data preprocessing is a proven method of resolving such issues. Demertzis (2019)
3. Feature Engineering/Data transformation: refers to the process of converting or modifying raw data

from its original form into a different format that is more suitable for analysis, modelling, or visualization James et al. (2013)

4. Exploratory Data Analysis: is a way of exploring data sets to find patterns, anomalies, and insights using statistics and visuals.
5. Modelling: Various machine learning and statistical modelling techniques are applied to build predictive models based on the processed data. These models may include: Statistical Models: Linear regression, time series analysis (e.g., ARIMA), and autoregressive models. Machine Learning Models: Decision trees, random forests, gradient boosting, support vector machines (SVM), neural networks.
6. Model Evaluation and Validation: The performance of the predictive models are evaluated using different metrics such as accuracy, precision, recall, mean absolute error (MAE), mean squared error (MSE), Area Under ROC Curve and root mean squared error (RMSE).
7. Operationalization: Once developed and validated, the predictive models are put into use.

To perform the above tasks effectively the following data analysis tools were used:

1. "R Programming Language: R is an open-source programming language that is widely used as a statistical software and data analysis tool. R generally comes with the Command-line interface. R is available across widely used platforms like Windows, Linux, and macOS. Also, the R programming language is the latest cutting-edge tool." GeeksforGeeks (2023)
2. Excel: Excel is an electronic spreadsheet program that is used for storing, organizing, and manipulating data.

## 3.2 Data Collection

For my research I used weather data from Bulawayo, Zimbabwe. The data contained day wise weather attributes from 2011 to April 2024 and was available on Visual Crossing: Weather Data Services — Visual Crossing. The file format was stored in an Excel (.xlsx) file.

Visual Crossing is a top supplier of enterprise analytic tools and meteorological data to academics, professionals, and data scientists. Since its founding in 2003, the company's goal has been to enable analysts and data consumers to make better decisions by providing them with high-quality, easily accessible data. Some of the biggest companies in the world have been using enterprise-class solutions from Visual Crossing for almost 20 years. By offering the most affordable access to all kinds of weather data for apps, code, and web pages globally, their weather data API leads the industry. Visual Crossing offers the information

you require, whether it be for historical weather data, weather forecasts, climatic summaries, or specialised weather metrics like solar radiation, degree days, and weather warnings.

The code snippet below was used to set the working directory to the location of the Excel file, import the data from the Excel file into R using the `read_excel()` function, and then display the imported data in a data viewer window for examination.

```
1 #Importing xlsx data into R
2
3 setwd("C:/Users/Admin/Desktop/Project SORS4010/Project Data Set")
4 getwd()
5 library(readxl)
6 bulawayo_2011_01_01_to_2024_04_14 <- read_excel("bulawayo 2011-01-01 to 2024-04-14.xlsx")
7 View(bulawayo_2011_01_01_to_2024_04_14)
8
```

### 3.3 Data Preprocessing

1. Identifying Unnecessary Features through Initial Data Examination.

```
8
9 ###Data Preprocessing
10
11 ##Feature/Varibale Selection
12 Weather_Data <- subset(bulawayo_2011_01_01_to_2024_04_14,
13                        select = -c(datetime, name, precipprob, preciptype, snow, snowdepth,
14                                   windgust, severerisk, sunrise, sunset, conditions,
15                                   description, icon, stations))
16
```

The code above was used to create a new dataset `Weather Data` that contains all columns from the original dataset, `bulawayo 2011.01.01 to 2024.04.14` except for the specified columns listed within `c(...)`. This allows for the creation of a subset with a reduced set of features for further analysis.

Having a clear understanding of the problem and the data, allowed me to identify features that were irrelevant or known to have no predictive power. Removing these features upfront streamlined the analysis process and improved the efficiency of subsequent modelling tasks.

2. Identify and Handle Missing Values

**Imputation:** This involves filling in missing values with estimated values based on the known values in the dataset. Common imputation techniques include mode, median and mean.

```
##Dealing with missing data
#Identify columns with missing data
missing_data <- colSums(is.na(Weather_Data)) > 0
#Display the columns with missing data
print(names(Weather_Data)[missing_data])
|
#Choosing the appropriate measure to replace missing values
summary(Weather_Data$sealevelpressure)

summary(Weather_Data$visibility)

# Replace missing values with the mean of the column
Weather_Data$sealevelpressure <- ifelse(is.na(Weather_Data$sealevelpressure),
                                       mean(Weather_Data$sealevelpressure, na.rm = TRUE),
                                       Weather_Data$sealevelpressure)

Weather_Data$visibility <- ifelse(is.na(Weather_Data$visibility),
                                 mean(Weather_Data$visibility, na.rm = TRUE),
                                 Weather_Data$visibility)
```

The above code snippet identified columns with missing data, chose an appropriate measure to replace missing values (mean), and then replaced missing values in the specified columns with the mean of each column. This helped to preprocess the data and ensure completeness before further analysis.

3. Identify unnecessary features through correlation analysis. A measure of the relationship between two variables is the correlation coefficient, which measures the strength of the linear association between two variables.

The value of the correlation coefficient always lies in the range  $-1$  to  $1$ ; that is,

$$-1 \leq r \leq 1$$

If  $r = 1$ , it is said to be a perfect positive linear correlation. If  $r = -1$ , the correlation is said to be a perfect negative linear correlation. We do not usually encounter an example with perfect positive or perfect negative correlation. What we observe in real-world problems is either a positive linear correlation with  $0 \leq r \leq 1$  (that is, the correlation coefficient is greater than zero but less than 1) or a negative linear correlation with  $-1 \geq r \geq 0$  (that is, the correlation coefficient is greater than  $-1$  but less than zero). If the correlation between two variables is positive and close to 1, we say that the variables have a strong positive linear correlation. If the correlation between two variables is positive but close to zero, then the variables have a weak positive linear correlation. In contrast, if the correlation between two variables is negative and close to 1, then the variables are said to have a strong negative linear correlation. If the correlation between two variables is negative but close to zero, there exists a weak negative linear correlation between the variables.



The linear correlation coefficient is calculated by using the following formula.

$$r_{xy} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}} \quad (1)$$

$r_{xy}$  represents the correlation coefficient between variables  $x$  and  $y$ .

$x_i$  and  $y_i$  are individual data points for variables  $x$  and  $y$ , respectively.

$\bar{x}$  and  $\bar{y}$  are the means of variables  $x$  and  $y$ , respectively.

$n$  is the number of data points.

$$\text{Correlation Matrix} = \begin{bmatrix} 1 & r_{xy} & r_{xz} & \cdots & r_{xn} \\ r_{yx} & 1 & r_{yz} & \cdots & r_{yn} \\ r_{zx} & r_{zy} & 1 & \cdots & r_{zn} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_{nx} & r_{ny} & r_{nz} & \cdots & 1 \end{bmatrix}$$

Using the R Programming language as seen below we executed a code that calculated the correlation matrix of the Weather Data dataset, converted it to a long format, and created a heatmap visualization using ggplot2.

```
#Correlation Matrix
Correlation_Matrix <- cor(weather_Data)
print(Correlation_Matrix)

install.packages("ggplot2")
library(ggplot2)
install.packages("reshape2")
library(reshape2)
#Convert correlation matrix to long format
Correlation_Df <- melt(Correlation_Matrix)
Correlation_Df
#Create a heatmap
heatmap <- ggplot(Correlation_Df, aes(var1, var2, fill = value)) +
  geom_tile(color = "white") +
  geom_text(aes(label = round(value, 2)), color = "black") + # Add text labels with rounded correlation values
  scale_fill_gradient2(low = "blue", high = "red", mid = "white", midpoint = 0,
    breaks = c(seq(-1, 0, by = 0.2), seq(0.2, 1, by = 0.2)),
    labels = c(seq(-1, 0, by = 0.2), seq(0.2, 1, by = 0.2))) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Correlation Heatmap",
    x = "Variables",
    y = "Variables")

print(heatmap)
```

Overall, the correlation matrix serves as a valuable tool for understanding the interrelationships between variables in a dataset, informing data preprocessing, modelling decisions, and exploratory data analysis.

```

#Correlation between each predictor variable and the independent variable (precip/rainfall)
#Loop through predictor variables and calculate correlation with dependent variable
data <- Weather_Data
correlations <- lapply(names(data)[-1], function(var) {
  cor.test(data[[var]], data$precip)
})

# Extract correlation coefficients and p-values
results <- lapply(correlations, function(correlation) {
  c(correlation$estimate, correlation$p.value)
})

#Combine results into a data frame
results_df <- data.frame(variable = names(data)[-1], do.call(rbind, results))
# Reorder the levels of the Weather Parameter variable based on the value(correlation)
results_df$variable <- factor(results_df$variable, levels = results_df$variable[order(results_df$cor, decreasing = TRUE)])

#Create the bar chart
ggplot(results_df, aes(x = variable, y = cor)) +
  geom_bar(stat = "identity", fill = "skyblue") +
  labs(title = "Vertical Bar Chart",
       x = "Variable", # Replace "Variable" with the actual variable name
       y = "Correlation Coefficient") + # Replace "Correlation Coefficient" with the appropriate label
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

```

The code snippet above was used to visually represent the correlation levels between each predictor and the independent variable using a bar chart.

```

89
90
91 #Now we remove the variables with 0 correlation with our dependent variable
92 #However we shall keep the variables that are correlated to the key predictors
93 weather_Data <- subset(weather_Data, select = -c(winddir, moonphase))
94

```

The above code snippet was used to eliminate the selected variables from the dataset.

### 3.4 Data Transformation

Here I performed several data preprocessing steps related to classifying precipitation as rain or no rain based on a specified threshold, adding columns to the dataset, shifting values to prepare for model building, and changing data types.

```

100 #Define the threshold for classifying rain (in mm)
101 threshold <- 2.5
102
103 #Classify precipitation as rain or no rain. 1 = rain, 0 = no rain
104 rain_classification <- ifelse(weather_Data$precip > threshold, 1, 0)

```

The above code snippet was used to define a threshold for classifying rain. Rainfall was classified as binary based on a precipitation level threshold of 2.5mm: any amount equal to or exceeding this threshold was labeled as "rain," while anything below 2.5mm was categorized as "no rain." To represent this binary classification, a label of 1 was assigned to instances of rain and 0 to instances of no rain. To consider precipitation as rain within a 24-hour period, meteorologists typically use a minimum threshold. While there isn't a universal standard, a common benchmark is 2.5 millimeters (mm) of rainfall. This is because any amount less than this is usually not enough to penetrate the ground or have a significant impact on the

environment.

```
109 #add new column rain_today
110 weather_Data$rain_today <- rain_classification
111
112 #I also added a rain_tomorrow column to prepare for model building
113 #Create a new column rain_tomorrow with the same values as column rain but shifted upward
114 weather_Data$rain_tomorrow <- c(weather_Data$rain_today[-1], NA)
115
116 #Drop the last row using negative indexing
117 weather_Data <- weather_Data[-nrow(weather_Data), ]
118
119 #Changing the data type
120 weather_Data$rain <- as.factor(weather_Data$rain)
121 weather_Data$rain_tomorrow <- as.factor(weather_Data$rain_tomorrow)
122
```

The provided code snippet (as show above) was utilized to augment the dataset by introducing two new columns: "rain today" and "rain tomorrow." This action was taken in line with the core objective of our research project, which focuses on constructing a model to forecast whether it will rain the following day. The "rain today" column was populated using the pre-defined rain classification vector. Subsequently, the "rain tomorrow" column was derived by shifting the "rain today" column upward, thus preparing the data for model development. To ensure compatibility with modeling techniques, the data type of both "rain today" and "rain tomorrow" columns was converted to a categorical factor. Finally, the last row, which lacks a corresponding "rain tomorrow" value, was dropped from the dataset.

### 3.5 Exploratory Data Analysis

Here I performed exploratory data analysis (EDA) on the Weather Data dataset. Histograms and bar plots were used to visualize the relationship between various weather features and the occurrence of rain tomorrow. Below is a breakdown of how the EDA is going was conducted in R:

1. Summary Statistics: Provides summary statistics for the Weather Data dataset.
2. Histograms and Bar Plots: Several ggplot commands were used to create histograms and bar plots for different weather features such as precipitation, precipitation cover, humidity, cloud cover, dew, temperature, and minimum temperature. Each plot visualizes the distribution of the respective weather feature and its relationship with the occurrence of rain tomorrow. The bars were coloured based on whether rain is predicted for tomorrow (blue for no rain, red for rain). The labs function was used to add titles and axis labels to each plot. coord cartesian was used to adjust the limits of the x and y axes to ensure better visualization.

Overall, these visualizations help in exploring the relationships between various weather features and the occurrence of rain tomorrow, providing insights into potential patterns and correlations in the data.

## 3.6 Machine Learning

”Machine learning is a subfield of artificial intelligence (AI) that uses algorithms trained on data sets to create self-learning models that are capable of predicting outcomes and classifying information without human intervention.” Staff (2024).

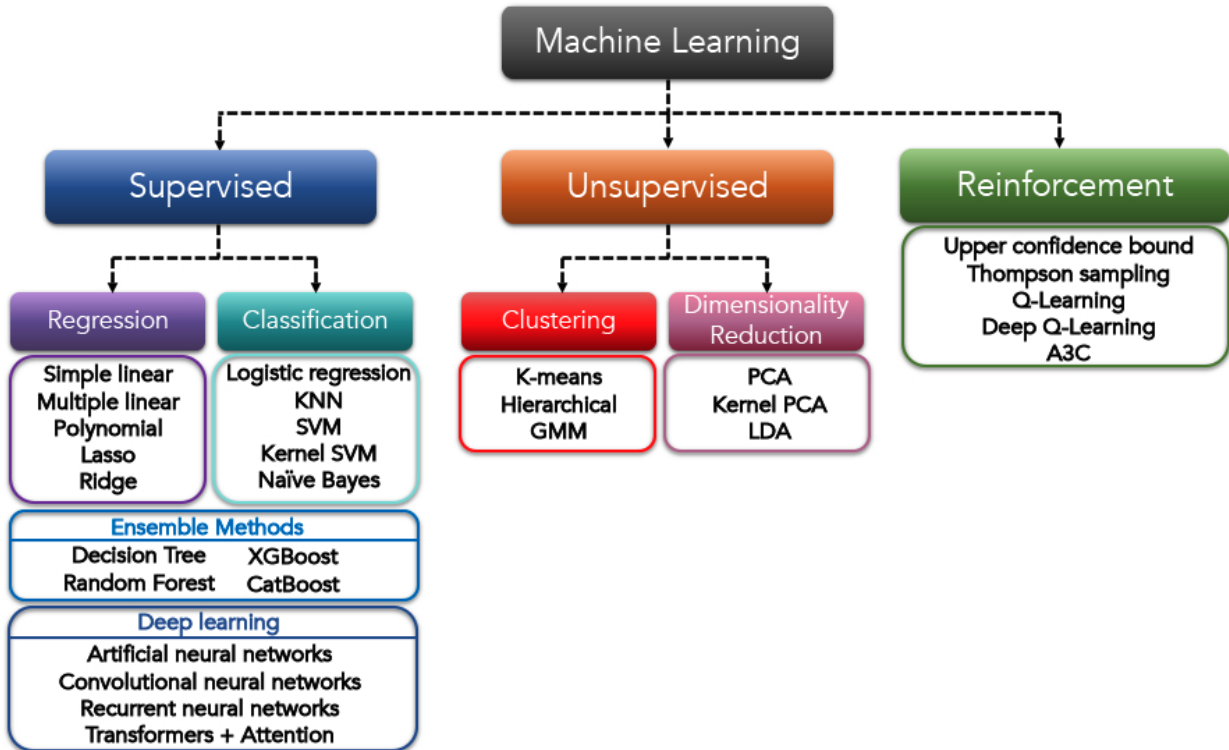


Figure 3: Types of Machine Learning

For my research project, I constructed, compared, and evaluated the predictive accuracy of the following models: Decision tree classifiers and two ensemble methods—Random Forest, which uses Bagging (Bootstrap Aggregating), and Gradient Boosting, which involves Boosted Trees. These models were used to predict whether it will rain tomorrow.

First, we partitioned our data into training and testing sets before proceeding with model construction.

### 3.6.1 Data Split

Splitting data refers to dividing a dataset into two or more subsets for different purposes, typically for training and testing machine learning models.

In R, you can split data using various functions from packages such as caret as shown below.

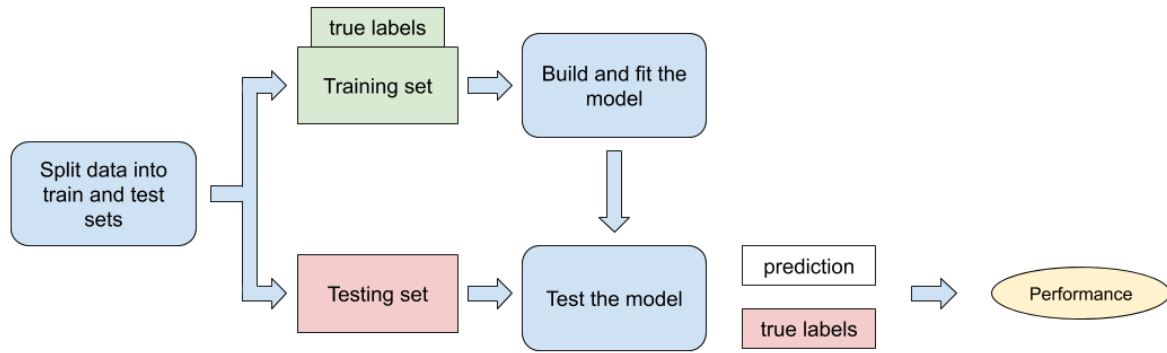


Figure 4: Data Split: Testing and Training

```

##Data Split: Training and Testing Data sets
#install and load the caret package
install.packages("caret")
library(caret)
#Set a random number generator seed so that the partition is always made at the same
#point, choosing 1503.
set.seed(1503)
#Now we create the partition vector using the createDataPartition function. In this
#specific case we will use 80% of the data to train the model and 20% to test it.
#the list = FALSE avoids returning the data as a list
partition <- createDataPartition(y = weather_Data$rain_tomorrow, p = 0.8, list = FALSE)
#Now let's split the weather_Data into training and testing datasets using the
#partition vector
training <- weather_Data[partition,]
testing <- weather_Data[-partition,]
  
```

### 3.6.2 Decision Tree

"A decision tree is a supervised machine learning algorithm that creates a series of sequential decisions to reach a specific result." Corbo (2023).

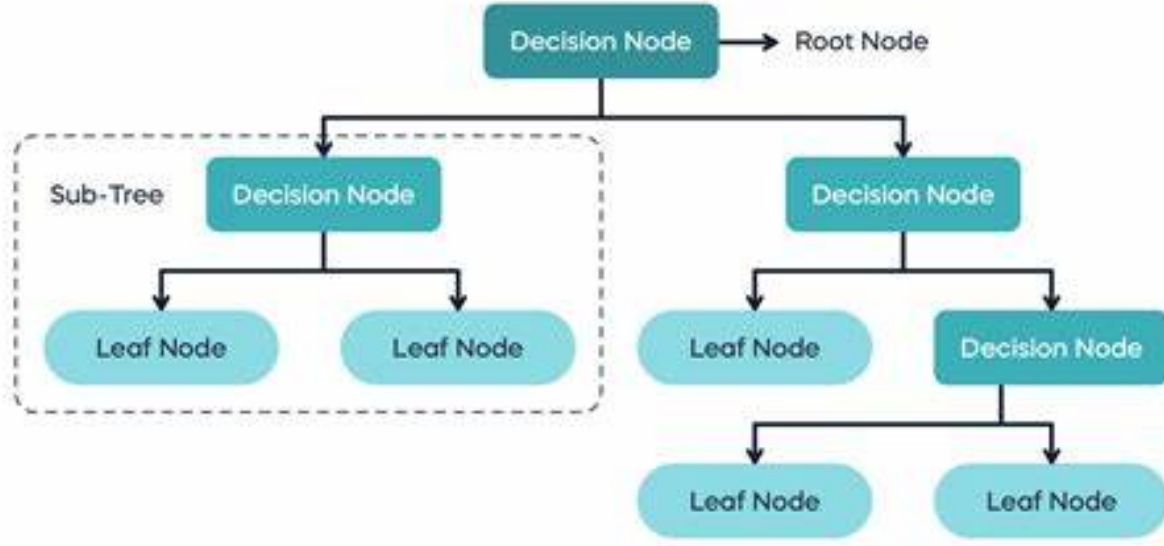


Figure 5: Structure of a Decision Tree. The365team (2024)

In a decision tree, decision nodes represent choices to be made, while leaf nodes signify the final outcomes of decision paths. I delved into the realm of Classification Decision Trees—a type of decision tree that iteratively partitions the dataset until reaching pure leaf nodes, In the context of predicting rain or no rain, this process continues until each leaf node exclusively corresponds to a single class, providing a clear classification outcome for each instance in the dataset.

### Algorithms for constructing Decision Trees

Algorithms for constructing decision trees usually work top-down, choosing a variable at each step that best splits the set of items. Metrics measure the homogeneity of the target variable within the subsets. Examples of metrics include; Gini Impurity and Information Gain.

1. Gini Impurity: This is a measure of how often a randomly chosen element from the set would be incorrectly labelled if it was randomly labelled according to the distribution of labels in the subset.

$$Gini(S) = 1 - \sum_{j=1}^J (p_j)^2 \quad (2)$$

where  $p_j$  is the probability of picking a class  $j$  from the set  $S$

2. Information Gain: This is a measure of reduction in entropy (a measure of disorder or impurity in a

node) It is used to select the best splitting attribute when constructing a decision tree.

$$Entropy(S) = - \sum_{j=1}^J p_j \log_2(p_j) \quad (3)$$

$$\text{Information Gain} = Entropy(S) - \sum_{i=1}^n \frac{|S_i|}{|S|} \cdot Entropy(S_i) \quad (4)$$

where  $|S_i|$  represents the number of instances in the subset  $S_i$  after splitting the dataset.

$|S|$  represents the total number of instances in the original dataset  $S$

The higher the information gain the more significant the feature is.

### Conditional Inferencing Trees (CTREE)

CTREE uses statistical tests, such as chi-squared test or permutation tests, to determine the significance of splits based on conditional inference rather than relying solely on impurity measures.

```
###Decision trees with ctree
#install packages that contain ctree
install.packages("partykit")
library(partykit)
#Now we make a tree with rain_tomorrow as the dependent variable and all other variables
#as predictors
tree.ctree <- ctree(training$rain_tomorrow ~ ., data = training)
plot(tree.ctree, gp = gpar(fontsize = 7))
```

The code snippet above was used to construct the decision tree model using the CTREE algorithm with the training dataset. An end user can easily interpret the resulting tree structure and visualization to understand the relationships between variables and the predicted outcomes.

We then made predictions as shown below:

```
##Prediction with ctree model
#Once we have created our conditional inferencing model with ctree, it is time
#to use such model to predict the dependent variable on the testing dataset. For this
#we will use the predict function. First, lets estimate the rain tomorrow prob
#on the testing dataset.
prob.ctree <- predict(tree.ctree, newdata = testing, type = "prob")
#prob.ctree is a table with 969 rows of the testing dataset and two columns for classes 0 and 1.
#class 0 contains the prob of no rain and class 1 contains the prob of rain, that is, the prob that
#it will rain tomorrow

#Based on the rain prob, lets classify the conditions for that, we need a cutoff value. We shall
#estimate the cutoff value as the average prob of rain
mean(as.integer(Weather_Data$rain_tomorrow))
#We can observe that the average is 1.143858 which should call our attention because it is greater than 1.
#when we take a look at the dependant we notice R saves with values 1 and 2 instead of 0 and 1. So all we
#have to do is subtract 1 from the mean, and we obtain that the mean prob of rain_tomorrow is
mean(as.integer(Weather_Data$rain_tomorrow))-1
#0.143852, which we will use as the cut-off value

#First lets create a vector of 969 obs, the size of the testing base
classification.ctree <- rep("1",969)
#And now we assign zero to those obs with a rain prob less than 0.143852
classification.ctree[prob.ctree[,2]<0.143852] = "0"
classification.ctree
```

Determining the Cutoff Value:

- The code snippet above calculates a cutoff value for classifying the predictions into binary outcomes (rain or no rain).
- It computes the mean of the dependent variable (`rain_tomorrow`) in the entire dataset (`Weather_Data`).
- However, it seems there's a misunderstanding regarding the values of `rain_tomorrow`, as the mean turns out to be greater than 1.
- The script corrects this by subtracting 1, resulting in a corrected mean that is used as the cutoff value.

Classification:

- After determining the cutoff value, the script initializes a vector `classification.ctree` with all elements set to "1".
- Then, it assigns "0" to the elements where the rain probability (from `prob.ctree`) is less than the calculated cutoff value.

This script ultimately generates a binary classification vector `classification.ctree`, where "1" indicates predicted rain and "0" indicates predicted no rain for each observation in the testing dataset.

### 3.6.3 Random Forest

The fundamental idea behind a random forest is to combine many decision trees into a single model. Individually predictions made by decision trees may not be accurate, but combined together the predictors will be closer to the mark on average.



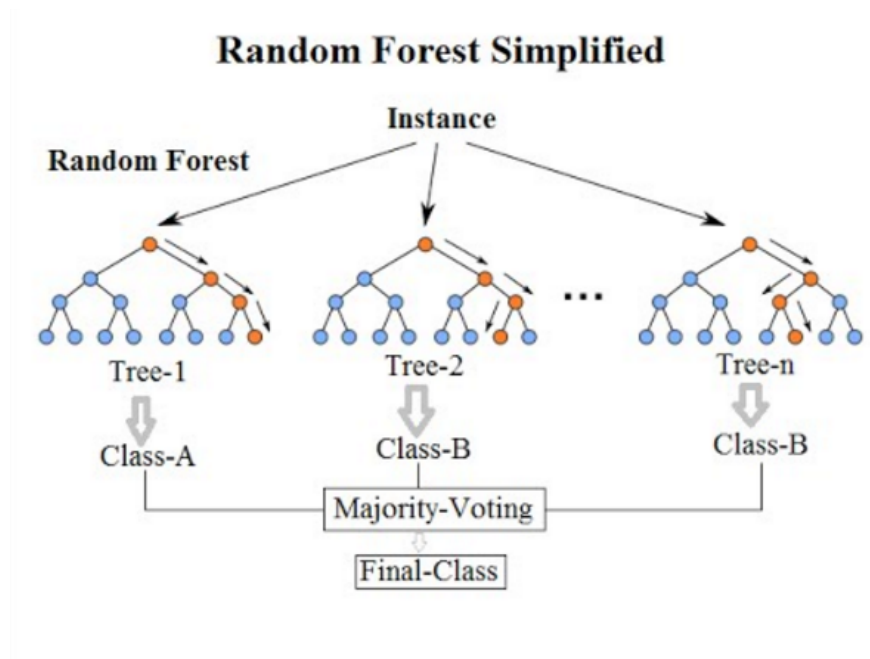


Figure 6: Random Forest graphical representation. Koehrsen (n.d.)

The code snippet below allowed me to build a Random Forest model, visualize the model, and assess the importance of different variables in predicting the target variable (rain tomorrow).

```
##Random Forest
install.packages("randomForest")
library(randomForest)
RF.model <- randomForest(rain_tomorrow ~., data = training, importance = TRUE,
                        proximity = TRUE, cutoff = c(0.6,0.4), type = "classification")
#Now lets print and plot the model
print(RF.model)
plot(RF.model)

#Now lets observe the importance of each of the variable in the model, both due
#accuracy and the impurity associated with them
importance(RF.model)
#Now lets plot it
varImpPlot(RF.model)
```

Here's a breakdown of each parameter and what it does:

1. Formula Specification: `rain tomorrow`; This specifies the formula for the model. It indicates that you want to predict the variable `rain tomorrow` based on all other variables in the dataset (training).
2. Data: `data = training`; This specifies the dataset (training) from which the model will be trained.
3. Importance Calculation: `importance = TRUE`; This parameter tells the `randomForest` function to calculate the importance of predictor variables. Variable importance is a measure of how useful each variable is in predicting the target variable.

4. Proximity Calculation: `proximity = TRUE`: This parameter instructs the function to calculate proximity measures between pairs of observations. Proximity measures quantify how close two observations are in the feature space. They can be useful for clustering or outlier detection.
5. Cutoff Probability: `cutoff = c(0.6, 0.4)`: This parameter sets the cutoff probabilities for classification. In binary classification problems like this one, probabilities above 0.6 are classified as one class (in this case, "rain tomorrow"), and probabilities below 0.4 are classified as the other class (in this case, "no rain tomorrow").
6. Type of Model: `type = "classification"`: This parameter specifies the type of model to build. In this case, it's a classification model because you're predicting a categorical outcome ("rain tomorrow" or "no rain tomorrow"). When you run this line of code, the `randomForest()` function will construct a Random Forest model using the specified parameters. This model will then be stored in the variable `RF.model`, which you can use for prediction, evaluation, and further analysis.

We then made predictions as shown below:

```
#lets compute the probabilities
prob.rf <- predict(RF.model, newdata = testing, type = "prob")
#Vector of ones
classification.RF <- rep("1",969)
#Now lets classify based on the prob, and set the cutoff value at 0.143852 to be
#consistent with the one we used with decision trees
classification.RF[prob.rf[,2]<0.143852] = "0"
#change to factor for use in confusion matrix
classification.RF <- as.factor(classification.RF)
```

### 3.6.4 Gradient Boosting

Gradient boosting is a popular machine learning technique used for both regression and classification tasks. It's an ensemble learning method that combines the predictions of several base estimators (typically decision trees) sequentially, with each new model correcting errors made by previous ones.

The code snippet below was used to build a gradient boosting model using the specified parameters and the training data. The resulting model was then stored in the variable `model.XGB` and used for prediction and evaluation.

```

###Gradient boosting with xgboost
install.packages("xgboost")
library(xgboost)
#Divide data into training and testing for xgboost
training.X <- model.matrix(rain_tomorrow ~., data = training)
testing.X <- model.matrix(rain_tomorrow ~., data = testing)
#creating design matrices using the formula interface ensures that the
#predictors are correctly represented in a format suitable for modeling,
#and applying this process consistently to both training and testing datasets
#ensures consistency and accuracy in model building and evaluation.

#Now we can build the model
model.XGB <- xgboost(data = data.matrix(training.X[,-1]), #remove 1st column (intercept)
                    label = as.numeric(as.character(training$rain_tomorrow)),
                    eta = 0.1,
                    max_depth = 20,
                    nround = 50,
                    objective = "binary:logistic")

```

1. **Label:** This parameter specifies the response variable (i.e., the target variable to be predicted). It is typically a numeric vector. In this case, `as.numeric(as.character(training$rain_tomorrow))` is used to convert the response variable `rain_tomorrow` from the training dataset to a numeric vector. The `as.character()` function is used to ensure that the response variable is treated as a character vector before conversion to numeric.
2. **eta:** [default = 0.3][range:(0,1)]. It controls the learning rate, i.e., the rate at which our model learns patterns in data. After every round, it shrinks the feature weights to reach the next optimum. A lower **eta** leads to slower computations.
3. **max\_depth** [default = 6][range:(0,inf)]: It controls the depth of the tree. The larger the depth, the more complex the model; higher chances of overfitting. There is no standard value for **max\_depth**. Larger datasets require deep trees to learn the rules from the data.
4. **nround** [default = 100]. It controls the maximum number of iterations. For classification, it is similar to the number of trees to grow.
5. **Objective** [default = reg:linear]:
  - (a) **reg:linear** for linear regression
  - (b) **binary:logistic** - logistic regression for binary classification
  - (c) **multi:softmax** - multiclassification using softmax objective. It returns predicted class probabilities.

```
#Prediction
prediction.XGB <- predict(model.XGB, newdata = testing.X[,-1], type = "prob")
#Confusion Matrix
```

`newdata = testing.X[,-1]`: This specifies the new data (testing dataset) on which predictions are to be made. `testing.X` contains the testing dataset, and `[, -1]` removes the first column (an intercept column) since it's not needed for prediction.

### 3.7 Hyperparameter Tuning

Hyperparameter tuning is a critical step in the machine learning model development process aimed at finding the optimal values for the hyperparameters, which are parameters that govern the learning process and model complexity. Unlike model parameters, which are learned from data during training, hyperparameters must be set before training and affect how the model learns and generalizes.

#### **ctree model**

`ctree` (Conditional Inference Trees) is known to overfit by its very nature if not properly controlled, hence it is apparent that we implement hyperparameter tuning to address this challenge.

The code snippet below was used to perform hyperparameter tuning for the `ctree` model in R.

```
library(caret)
train_control <- trainControl(method = "cv", number = 5)
htree.ctree <- train(rain_tomorrow ~ ., data = training, method = "ctree", trControl = train_control)
htree.ctree
plot(htree.ctree, gp = gpar(fontsize = 7))
```

The above code sets up cross-validation with 5-fold cross-validation (`number = 5`) using the `trainControl` function from the `caret` package. Cross-validation is a common technique used for model evaluation and hyperparameter tuning, where the dataset is split into multiple folds, and the model is trained and evaluated multiple times on different subsets of data. The code uses the `train` function from the `caret` package to train a decision tree model (`method = "ctree"`) on the training data (`training`) while utilizing the cross-validation setup defined earlier (`trControl = traincontrol`).

The `train` function will automatically perform hyperparameter tuning by trying different combinations of hyperparameters (such as `mincriterion`, `minsplit`, etc., which are specific to the `ctree` function) and evaluating the model's performance using cross-validation.

Printing `htree.ctree` will display information about the trained model, including the hyperparameters selected during tuning, model performance metrics (e.g., accuracy, AUC), and other details.

```
pruned.ctree <- ctree(training$rain_tomorrow ~ ., data = training, mincriterion = 0.999999, minsplit = 10, minbucket = 5)
plot(pruned.ctree)
```

The mincriterion, minsplit, and minbucket arguments are hyperparameters that control the tree's growth and pruning process. They determine the criteria for splitting nodes (mincriterion) and the minimum number of samples required to split a node (minsplit) or create a terminal node (leaf) (minbucket). In this case, mincriterion is set to 0.999999, minsplit is set to 10, and minbucket is set to 5.

## Random Forest and XGBoost Model

While hyperparameter tuning is a powerful technique to optimize model performance, it is not always necessary for every model, particularly in the case of Random Forests and Gradient Boosting models. Here's why:

### 1. Inherent Robustness

- Random Forest inherently reduces overfitting through its ensemble nature. By averaging the results of many decision trees, it minimizes the impact of any single overfitted tree.
- Gradient Boosting models like XGBoost, LightGBM, and CatBoost come with default hyperparameters that have been fine-tuned through extensive research and practical application, often performing well across a wide range of problems.

### 2. Empirical Performance

- Numerous studies and practical applications have demonstrated that the default settings for both Random Forest and Gradient Boosting models provide competitive results.
- In many cases, the performance gains from extensive hyperparameter tuning are marginal compared to the robust performance achieved with default parameters.

### 3. Overfitting Mitigation by Design

- Random Forest uses bootstrapping and feature randomness to ensure that the trees are less correlated and generalize well to unseen data.
- Gradient Boosting employs a sequential approach where each tree corrects the errors of its predecessors, and techniques like learning rate reduction and early stopping further prevent overfitting.

## 3.8 Model Evaluation and Validation

The evaluation techniques that were used included the confusion matrix and the Area Under the ROC Curve (AUC). These evaluation techniques were used to assess the performance of the model.

### 3.8.1 Confusion Matrix

A confusion matrix is a table that is often used to evaluate the performance of a classification model. The matrix represents a summary of the predictions made by a classifier compared to the actual known class labels in the dataset.

Here is how it is structured:

- True Positive (TP): The case where the classifier correctly predicted the positive class.
- True Negative (TN): The case where the classifier correctly predicted the negative values.
- False Positive (FP): The case where the classifier incorrectly predicted the positive class when the actual class was negative (Type 1 error)
- False Negative (FN): The case where the classifier incorrectly predicted the negative class when the actual class was positive (type 2 error)

		Actual Values	
		0	1
Predicted values	0	TN	FP
	1	FN	TP

Figure 7: Confusion Matrix. Novaes et al. (2021)

Accuracy, sensitivity, and specificity are common performance metrics used to evaluate the effectiveness of classification models, particularly in binary classification tasks.

Accuracy measures the proportion of correct predictions made by the model among all predictions. It is calculated as the ratio of the number of correct predictions to the total number of predictions. Mathematically, accuracy can be expressed as:

$$\text{Accuracy} = \frac{TP + TN}{TN + FP + FN + TP} \quad (5)$$

Sensitivity measures the proportion of actual positive cases that were correctly identified by the model. It is calculated as the ratio of true positive predictions to the total number of actual positive cases. Mathematically, sensitivity can be expressed as:

$$\text{Sensitivity} = \frac{TP}{TP + FN} \quad (6)$$

Specificity measures the proportion of actual negative cases that were correctly identified by the model. It is calculated as the ratio of true negative predictions to the total number of actual negative cases. Mathematically, specificity can be expressed as:

$$\text{Specificity} = \frac{TN}{TN + FP} \quad (7)$$

In R, there are several packages that provide functions for generating confusion matrices. One commonly used package is `caret`, which offers a variety of functions for training and evaluating machine learning models, including generating confusion matrices. Alternatively, you can use other packages like `MLmetrics`, `e1071`, or `caretEnsemble`, which also provide functions for generating confusion matrices and evaluating classification models in R. Then, you can use the `confusionMatrix()` function to create a confusion matrix.

```
confusion_matrix <- confusionMatrix(as.factor(prediction), as.factor(actuals))
```

### 3.8.2 ROC Curve (Receiver Operating Characteristic)

This is a graphical representation commonly used to evaluate the performance of binary classification models. It illustrates the trade off between the True Positive Rate and the False Positive Rate across different threshold values.

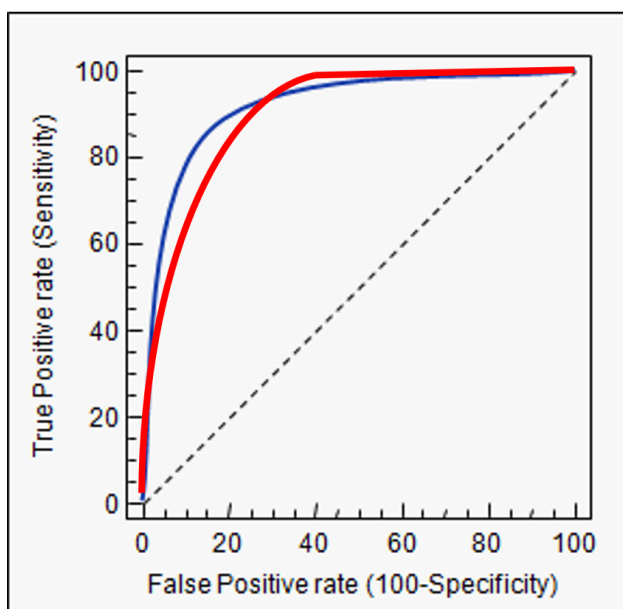


Figure 8: ROC Curve. user2149631 (n.d.)

The ROC Curve shows the trade off between sensitivity (or TPR) and specificity ( $1 - \text{FPR}$ ). Classifiers that give curves closet to the upper left corner indicate better performance. As a baseline, a random classifier is expected to score points along the diagonal ( $\text{FPR} = \text{TPR}$ ). The closer the ROC Curve is to the 45 degree diagonal the less accurate the model will be.

To plot a Receiver Operating Characteristic (ROC) curve in R, I used the `ROCR` package as shown below:



ctree:

```
##ROC Curve: Receiver Operating Characteristic
#This is a graphical representation commonly used to evaluate the performance of binary
#classification models. It illustrates the trade-off between the True Positive Rate(TPR)
#and the False Positive Rate across different threshold values.
install.packages("ROCR")
library(ROCR)
library(caret)
#Let's calculate the errors
prediction.ctree.ROC <- prediction(prob.ctree[,2], testing$rain_tomorrow)
prediction.ctree.ROC
#Let's generate the data for for the ROC curve with the performance function and the parameters
#which are prediction.ctree.ROC, that is the models errors that we computed above, TPR in the
#y-axis, and FPR on the x-axis.
ROC.ctree <- performance(prediction.ctree.ROC, "tpr", "fpr")
#let's plot the ROC curve
plot(ROC.ctree)
```

Random Forest:

```
#ROC curve
#lets generate the data for the ROC curve
library(ROCR)
prediction.RF.ROC <- prediction(prob.rf[,2], testing$rain_tomorrow)
#Now lets build the ROC curve
ROC.RF <- performance(prediction.RF.ROC, "tpr","fpr")
#plot the curve
plot(ROC.RF)
```

Gradient Boosting (XG Boost):

```
##ROC and AUC
#Lets compute the model's errors
errors.XGB <- prediction(prediction.XGB, testing$rain_tomorrow)
#Now lets build the ROC Curve of the XGB model
ROC.XGB <- performance(errors.XGB, "tpr", "fpr")
#lets plot the ROC curve
plot(ROC.XGB)
```

### 3.8.3 Area under the ROC Curve (AUC)

This is a performance metric used to evaluate the ability of a binary classification model to discriminate between positive and negative classes across different thresholds.

How to interpret AUC?

- 1 = Excellent
- 0 = Reciprocating
- 0.5 = No class separation capacity

The AUC (Area Under the Curve) values provide an indication of the discriminatory power of a classification model. Typically, AUC values greater than 0.9 are considered excellent, while values between 0.8 and 0.9 are considered very good. AUC values between 0.7 and 0.8 are deemed acceptable, but represent a more moderate level of accuracy. AUC values less than 0.6 indicate limited discriminatory ability and may not offer much value.

One of the primary utilities of the AUC is its ability to compare the forecasting accuracy of two or more methods. By comparing the AUC values of different models, researchers can determine which model performs better at distinguishing between the classes of interest.

$$\text{AUC} = \sum_{i=1}^{n-1} \frac{(FPR_{i+1} - FPR_i) \cdot (TPR_i + TPR_{i+1})}{2} \quad (8)$$

Here is how I computed the AUC in R

```
# Lets create the data to calculate the AUC
AUC.temp <- performance(prediction.ctree.ROC, "auc")
AUC.temp
#Let's extract and convert the values to a numerical variable
AUC.ctree <- as.numeric(AUC.temp@y.values)
AUC.ctree
```

The code snippet above calculated the Area Under the Curve (AUC) for the ctree classification model using the performance object obtained from the ROC curve. AUC for each model can be determined by making the necessary substitutions into the code snippet shown above.

## 4 Data Analysis

Data analysis in research methods refers to the process of transforming raw data into meaningful and interpretable information to answer research questions, test hypotheses, or conclude. It involves applying various statistical and analytical techniques to understand patterns, relationships, and trends within the data. Data analysis plays a crucial role in the research process and helps researchers make sense of their collected data. It involves several steps, including data cleaning, exploration, transformation, modeling, and interpretation. Hammad (2023)

In this section I am going present and interpret the results I obtained from analysing the data I collected for the study. This section is crucial as it demonstrates how I have addressed the research questions and/or objectives, and provides insights into the implications of my findings.

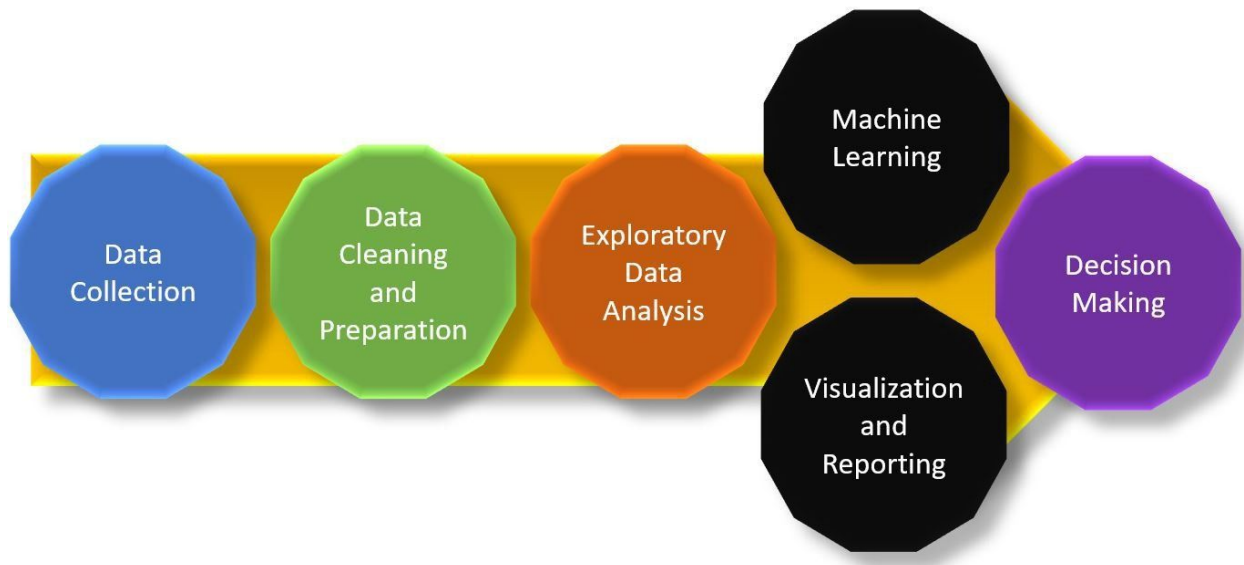


Figure 9: Data Analysis Overview. Turing (n.d.)

### 4.1 Data Collection

This section includes a description of the dataset and its acquisition.

For my research, I will utilize weather data sourced from Bulawayo, Zimbabwe. This dataset encompasses daily weather attributes spanning from January 2011 to March 2024. The data is accessible through Visual Crossing: Visualcrossing (n.d.). The file format was stored in an Excel (.xlsx) file which contains 4854 rows and 33 columns, the columns which were identified as: name, datetime, tempmax, tempmin, temp, feelslikemax, feelslikemin, feelslike, dew, humidity, precip, precipprob, precipcover, preciptype, snow, snowdepth,

windgust, windspeed, winddir, sealevelpressure, cloudcover, visibility, solarradiation, solarenergy, uvindex, severerisk, sunrise, sunset, moonphase, conditions, description, icon and stations. The weather dataset is stored as an Excel (.xlsx), and the R programming language is going to be utilized for preprocessing.

name	datetime	tempmax	tempmin	temp	feelslikemax	feelslikemin	feelslike
bulawayo	01/01/2011	30	17.9	22.5	28.9	17.9	22.4
bulawayo	02/01/2011	29	18	23.3	29.1	18	23.4
bulawayo	03/01/2011	30.1	17.9	23.6	30.1	17.9	23.6
bulawayo	04/01/2011	29.2	17	22.9	29.1	17	22.8
bulawayo	05/01/2011	30.8	16.1	23	29.8	16.1	22.8
bulawayo	06/01/2011	25.6	17.3	20.7	25.6	17.3	20.7
bulawayo	07/01/2011	26	15.6	19.1	26	15.6	19.1
bulawayo	08/01/2011	23.4	15.7	19.2	23.4	15.7	19.2
bulawayo	09/01/2011	27.7	17.2	21.9	28.1	17.2	22
bulawayo	10/01/2011	29.8	16.8	23.2	29.6	16.8	23.1

Table 1: Table Head Overview

NB: Table 1 provides a summary or brief overview of the structure of our dataset. It does not encompass the entire dataset, but rather offers insight into its composition and organization.

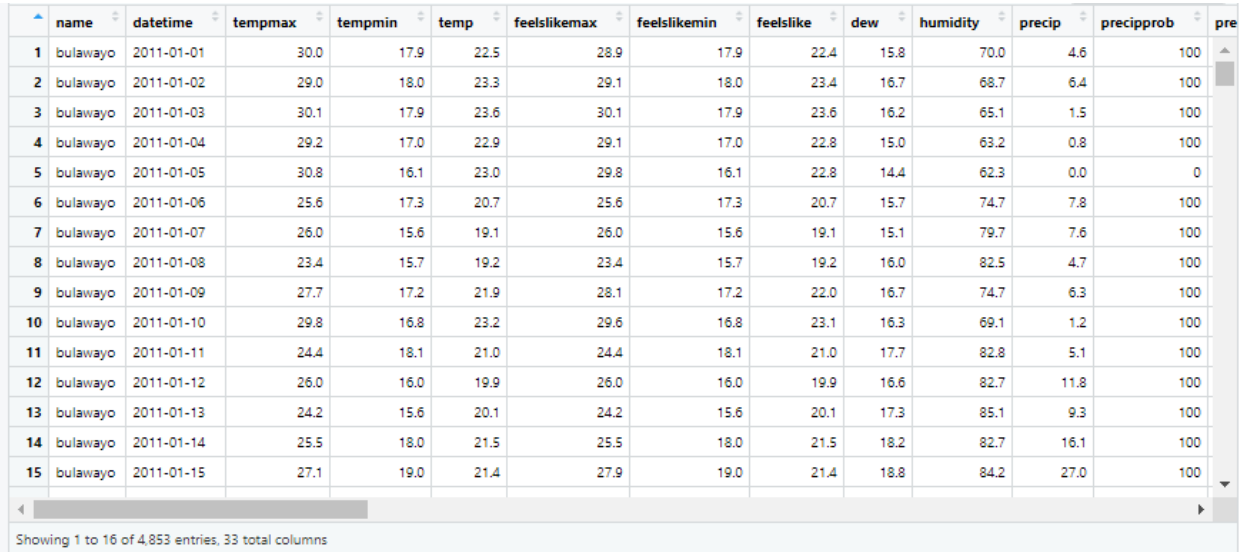
Element	Description	metric
tempmax	Maximum Temperature	C
tmpmin	Minimum Temperature	C
temp	Temperature (or mean temperature)	C
dew	Dew Point	C
feelslike	Feels Like	C
precip	Precipitation	mm
precipprob	Precipitation chance	%
precipcover	Precipitation Cover	%
precipitype	Precipitation type	-
snow	Snow	cm
snowdepth	Snow Depth	cm
windspeed	Wind Speed	kph
windgust	Wind Gust	kph
winddir	Wind Direction	degrees
visibilty	Visisbilty	km
cloudcover	Cloud Cover	%
humidity	Relative Humidity	%
pressure	Sea Level Pressure	mb
solarradiation	Solar Radiation	W/m2
solarenergy	Solar Energy	MJ/m2
uvindex	UV Index	-
severerisk	Severe Risk	-
sunrise	Sunrise time	-
sunset	Sunset time	-
moonphase	Moonphase	-
icon	A weather icon	-
conditions	Short text about the weather	-
description	Description of the weather for the day	-
stations	List of weather stations sources	-

Table 2: Weather Columns

The primary weather feature used to describe rain is 'precipitation'. This term refers to the amount of water that falls to the ground from the atmosphere. Precipitation is typically measured in terms of depth

over a specified period, such as millimeters of rain over an hour. Therefore, precipitation, abbreviated as 'precip', serves as our independent variable, while all other weather features serve as predictors.

#### 4.1.1 Importing xlsx data into R



	name	datetime	tempmax	tempmin	temp	feelslikemax	feelslikemin	feelslike	dew	humidity	precip	precipprob	pre
1	bulawayo	2011-01-01	30.0	17.9	22.5	28.9	17.9	22.4	15.8	70.0	4.6	100	
2	bulawayo	2011-01-02	29.0	18.0	23.3	29.1	18.0	23.4	16.7	68.7	6.4	100	
3	bulawayo	2011-01-03	30.1	17.9	23.6	30.1	17.9	23.6	16.2	65.1	1.5	100	
4	bulawayo	2011-01-04	29.2	17.0	22.9	29.1	17.0	22.8	15.0	63.2	0.8	100	
5	bulawayo	2011-01-05	30.8	16.1	23.0	29.8	16.1	22.8	14.4	62.3	0.0	0	
6	bulawayo	2011-01-06	25.6	17.3	20.7	25.6	17.3	20.7	15.7	74.7	7.8	100	
7	bulawayo	2011-01-07	26.0	15.6	19.1	26.0	15.6	19.1	15.1	79.7	7.6	100	
8	bulawayo	2011-01-08	23.4	15.7	19.2	23.4	15.7	19.2	16.0	82.5	4.7	100	
9	bulawayo	2011-01-09	27.7	17.2	21.9	28.1	17.2	22.0	16.7	74.7	6.3	100	
10	bulawayo	2011-01-10	29.8	16.8	23.2	29.6	16.8	23.1	16.3	69.1	1.2	100	
11	bulawayo	2011-01-11	24.4	18.1	21.0	24.4	18.1	21.0	17.7	82.8	5.1	100	
12	bulawayo	2011-01-12	26.0	16.0	19.9	26.0	16.0	19.9	16.6	82.7	11.8	100	
13	bulawayo	2011-01-13	24.2	15.6	20.1	24.2	15.6	20.1	17.3	85.1	9.3	100	
14	bulawayo	2011-01-14	25.5	18.0	21.5	25.5	18.0	21.5	18.2	82.7	16.1	100	
15	bulawayo	2011-01-15	27.1	19.0	21.4	27.9	19.0	21.4	18.8	84.2	27.0	100	

Showing 1 to 16 of 4,853 entries, 33 total columns

Figure 10: Data Head Overview of the Imported Data in R

## 4.2 Data Preprocessing

In this section I am going to clean, transform, and prepare the raw data for analysis and present the results.

### 4.2.1 Variable Selection

Feature selection is also called variable selection or attribute selection. It is the automatic selection of attributes in your data (such as columns in tabular data) that are most relevant to the predictive modeling problem you are working on. Brownlee (2021)

The "datetime" feature was removed from the dataset. This decision was made because weeks, dates, and months are human constructs, and natural phenomena like weather or climate may not have any inherent relationship with them.

The "name" and "station" features were removed from the dataset as they were deemed redundant and unnecessary for model building.

The "snow", "snowdepth", "windgust", and "severerisk" features were removed from the dataset due to their rarity. Including them could have led to skewed model outcomes.

The "sunrise" and "sunset" features were removed from the dataset since Sunrise and sunset themselves do not directly contribute to weather changes. Wilcox & Norris (2008)

The "conditions", "description", "preciptype", "precipprob" and "icon" features were removed from the dataset as they did not appropriately describe the weather element we aim to predict, which is rainfall.

#### 4.2.2 Dealing with Missing Data

```
> ##Dealing with missing data
> #Identify columns with missing data
> missing_data <- colSums(is.na(Weather_Data)) > 0
> #Display the columns with missing data
> print(names(Weather_Data)[missing_data])
[1] "sealevelpressure" "visibility"
>
> #Choosing the appropriate measure to replace missing values
> summary(Weather_Data$sealevelpressure)
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
   996   1018   1021   1021   1024   1034     705
>
> summary(Weather_Data$visibility)
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
   6.10   14.70   17.60   18.21   21.10   30.00     374
> |
```

From the output above, we initially identified columns with missing data, specifically "sealevelpressure" and "visibility". Subsequently, we displayed a summary of each feature to analyse the data, aiming to determine the most appropriate method for replacing the missing values. The mean was found to be the most appropriate measure.

Since there are no longer any missing observations from our Weather Dataset we can now further analyse the data.

#### 4.2.3 Further Variable Selection

We shall construct a correlation matrix for our weather data. The purpose is to measure the relationships between variables and ultimately eliminate insignificant variables based on their correlation with the dependent variable, 'precip'.

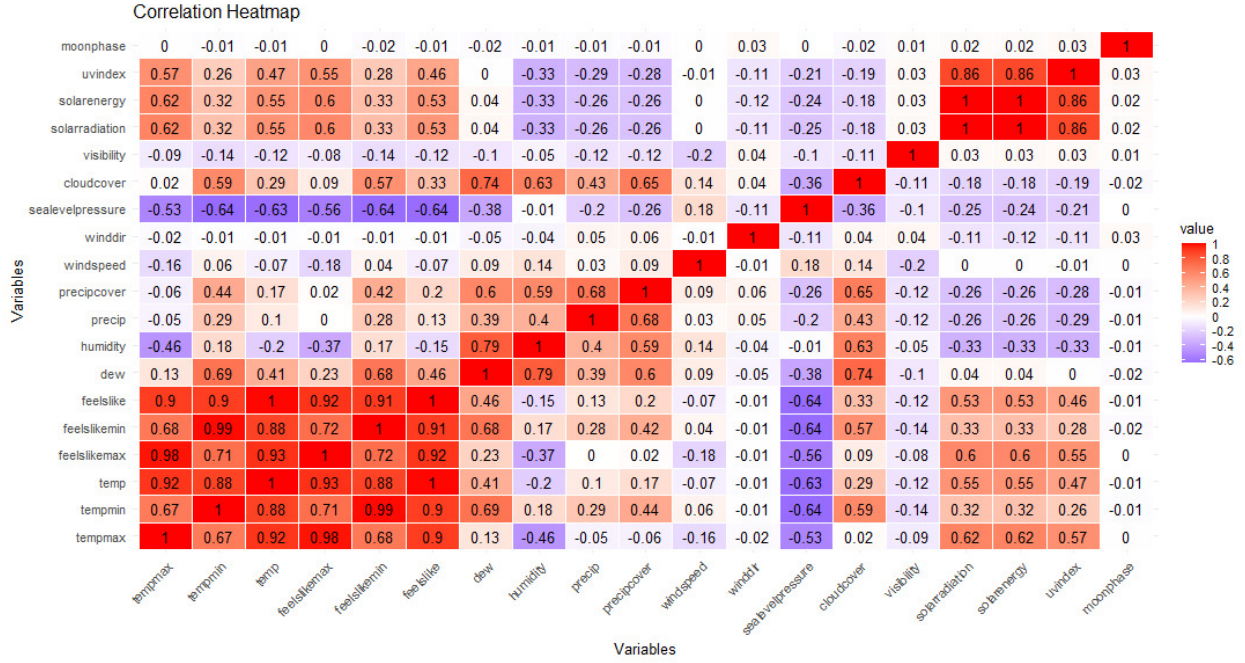


Figure 11: Correlation Heat Map

It can be seen from the above correlation heat map that there is a relatively strong positive correlation between "precipcover", "humidity", "dew", "cloudcover" and the predictor variable "precip" both registering a score above 0.39 with precipcover coming out the strongest with a correlation of 0.68. The other variables have a relatively weak correlation with the dependant variable with "feelslikemax", "windspeed", "moonphase" and "winddir" registering correlations close to zero. However, despite the low correlation of these features with the dependant variable it can be noted that some of these features are strongly related to some of the significant variables.

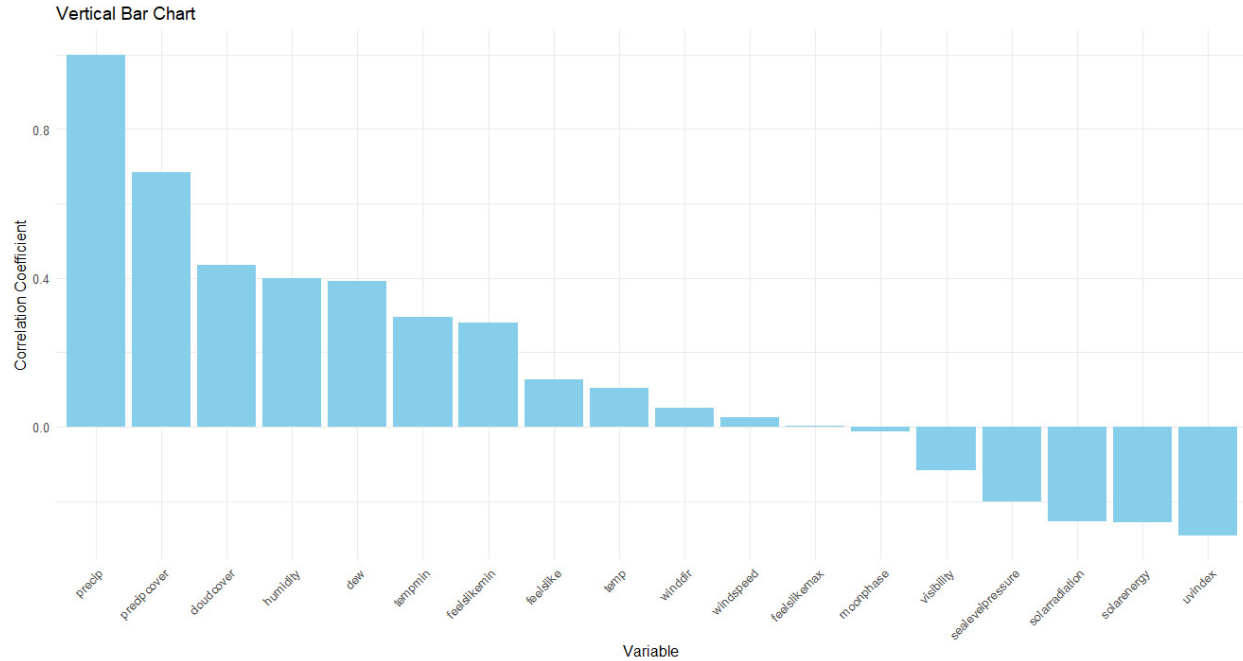


Figure 12: Correlation Level

From figure 12 The features windspeed, winddir, feelslikemax, and moonphase exhibit close to zero correlation with the independent variable precip. However, the correlation heatmap in figure 11 provides conclusive evidence of an existing relationship between windspeed, feelslikemax, and the other key predictors. Therefore, for our final feature selection, we will exclude the variables winddir and moonphase.

Weather_Data		4853 obs. of 17 variables														
\$ tempmax	: num [1:4853]	30	29	30.1	29.2	30.8	25.6	26	23.4	27.7	29.8	.				
\$ tempmin	: num [1:4853]	17.9	18	17.9	17	16.1	17.3	15.6	15.7	17.2	16.8					
\$ temp	: num [1:4853]	22.5	23.3	23.6	22.9	23	20.7	19.1	19.2	21.9	23					
\$ feelslikemax	: num [1:4853]	28.9	29.1	30.1	29.1	29.8	25.6	26	23.4	28.1	29					
\$ feelslikemin	: num [1:4853]	17.9	18	17.9	17	16.1	17.3	15.6	15.7	17.2	16.8					
\$ feelslike	: num [1:4853]	22.4	23.4	23.6	22.8	22.8	20.7	19.1	19.2	22	23					
\$ dew	: num [1:4853]	15.8	16.7	16.2	15	14.4	15.7	15.1	16	16.7	16.3					
\$ humidity	: num [1:4853]	70	68.7	65.1	63.2	62.3	74.7	79.7	82.5	74.7	69					
\$ precip	: num [1:4853]	4.6	6.4	1.5	0.8	0	7.8	7.6	4.7	6.3	1.2	...				
\$ precipcover	: num [1:4853]	62.5	58.3	41.7	33.3	0	...									
\$ windspeed	: num [1:4853]	20.5	14.8	20.2	27.7	27.9	19.8	26.3	19.8	22.3	...					
\$ sealevelpressure	: num [1:4853]	1019	1018	1020	1021	1020	...									
\$ cloudcover	: num [1:4853]	81.2	62.1	41.6	47.5	76.2	72.7	80.3	81.1	83.8	...					
\$ visibility	: num [1:4853]	24.2	21.1	23.6	23.6	24.5	24.7	12.1	21.1	22.3	...					
\$ solarradiation	: num [1:4853]	334	348	319	333	379	...									
\$ solarenergy	: num [1:4853]	28.8	30	27.7	28.9	32.8	16.6	26.2	13.2	25.5	28					
\$ uvindex	: num [1:4853]	10	10	9	10	10	7	10	5	9	10	...				



## 4.3 Data Transformation

Data transformation refers to the process of converting or modifying raw data from its original form into a different format that is more suitable for analysis, modelling, or visualization. James et al. (2013)

New Data Head Overview:

ax	feelslikemin	feelslike	dew	humidity	precip	precipcover	windspeed	sealevelpressure	cloudcover	visibility	solarradiation	solarenergy	uvindex	rain_today	rain_tomorrow
28.9	17.9	22.4	15.8	70.0	4.6	62.50	20.5	1019.000	81.2	24.20000	333.6	28.8	10	1	1
29.1	18.0	23.4	16.7	68.7	6.4	58.33	14.8	1017.500	62.1	21.10000	347.7	30.0	10	1	0
30.1	17.9	23.6	16.2	65.1	1.5	41.67	20.2	1020.000	41.6	23.60000	318.7	27.7	9	0	0
29.1	17.0	22.8	15.0	63.2	0.8	33.33	27.7	1021.000	47.5	23.60000	333.1	28.9	10	0	0
29.8	16.1	22.8	14.4	62.3	0.0	0.00	27.9	1019.500	76.2	24.50000	379.0	32.8	10	0	1
25.6	17.3	20.7	15.7	74.7	7.8	54.17	19.8	1014.900	72.7	24.70000	193.6	16.6	7	1	1
26.0	15.6	19.1	15.1	79.7	7.6	100.00	26.3	1014.300	80.3	12.10000	303.5	26.2	10	1	1
23.4	15.7	19.2	16.0	82.5	4.7	91.67	19.8	1013.900	81.1	21.10000	151.5	13.2	5	1	1
28.1	17.2	22.0	16.7	74.7	6.3	100.00	22.3	1012.600	83.8	22.30000	296.2	25.5	9	1	0
29.6	16.8	23.1	16.3	69.1	1.2	41.67	11.2	1014.800	71.0	25.00000	332.4	28.8	10	0	1
24.4	18.1	21.0	17.7	82.8	5.1	75.00	24.5	1017.800	86.7	22.50000	191.5	16.5	6	1	1
26.0	16.0	19.9	16.6	82.7	11.8	100.00	32.4	1018.500	75.0	17.20000	294.1	25.3	10	1	1
24.2	15.6	20.1	17.3	85.1	9.3	95.83	19.8	1016.900	84.3	16.90000	215.6	18.4	8	1	1
25.5	18.0	21.5	18.2	82.7	16.1	100.00	22.3	1014.700	93.0	21.10000	155.7	13.4	5	1	1
27.9	19.0	21.4	18.8	84.2	27.0	100.00	25.9	1016.400	87.5	18.30000	142.3	12.4	6	1	1
27.9	18.0	21.8	18.5	82.9	7.2	95.83	36.4	1017.500	78.7	18.30000	209.2	18.1	9	1	1
26.4	18.4	21.8	18.8	84.4	12.4	100.00	25.9	1014.900	79.6	27.10000	239.2	20.6	9	1	1
25.1	18.6	20.8	18.7	88.3	17.0	100.00	22.7	1015.900	93.8	20.70000	146.4	12.7	5	1	1
28.6	18.0	20.8	18.3	87.1	6.0	79.17	27.0	1014.700	90.9	20.20000	154.2	13.4	5	1	1
27.9	18.3	21.8	18.7	84.5	6.1	100.00	19.4	1013.400	83.3	23.00000	213.9	16.3	8	1	1

rain today and rain tomorrow columns have been added to the dataset to prepare for model building

New Data Description:

weather_Data		4852 obs. of 19 variables													
\$ tempmax	: num	[1:4852]	30	29	30.1	29.2	30.8	25.6	26	23.4	27.7	29.8	...		
\$ tempmin	: num	[1:4852]	17.9	18	17.9	17	16.1	17.3	15.6	15.7	17.2	16.8	...		
\$ temp	: num	[1:4852]	22.5	23.3	23.6	22.9	23	20.7	19.1	19.2	21.9	23.2	...		
\$ feelslikemax	: num	[1:4852]	28.9	29.1	30.1	29.1	29.8	25.6	26	23.4	28.1	29.6	...		
\$ feelslikemin	: num	[1:4852]	17.9	18	17.9	17	16.1	17.3	15.6	15.7	17.2	16.8	...		
\$ feelslike	: num	[1:4852]	22.4	23.4	23.6	22.8	22.8	20.7	19.1	19.2	22	23.1	...		
\$ dew	: num	[1:4852]	15.8	16.7	16.2	15	14.4	15.7	15.1	16	16.7	16.3	...		
\$ humidity	: num	[1:4852]	70	68.7	65.1	63.2	62.3	74.7	79.7	82.5	74.7	69.1	...		
\$ precip	: num	[1:4852]	4.6	6.4	1.5	0.8	0	7.8	7.6	4.7	6.3	1.2	...		
\$ precipcover	: num	[1:4852]	62.5	58.3	41.7	33.3	0	...							
\$ windspeed	: num	[1:4852]	20.5	14.8	20.2	27.7	27.9	19.8	26.3	19.8	22.3	11.2	...		
\$ sealevelpressure	: num	[1:4852]	1019	1018	1020	1021	1020	...							
\$ cloudcover	: num	[1:4852]	81.2	62.1	41.6	47.5	76.2	72.7	80.3	81.1	83.8	71	...		
\$ visibility	: num	[1:4852]	24.2	21.1	23.6	23.6	24.5	24.7	12.1	21.1	22.3	25	...		
\$ solarradiation	: num	[1:4852]	334	348	319	333	379	...							
\$ solarenergy	: num	[1:4852]	28.8	30	27.7	28.9	32.8	16.6	26.2	13.2	25.5	28.8	...		
\$ uvindex	: num	[1:4852]	10	10	9	10	10	7	10	5	9	10	...		
\$ rain_today	: Factor w/ 2 levels	"0","1":	2	2	1	1	1	2	2	2	2	1	...		
\$ rain_tomorrow	: Factor w/ 2 levels	"0","1":	2	1	1	1	2	2	2	2	1	2	...		

rain today and rain tomorrow were made to factors

## 4.4 Exploratory Data Analysis

Exploratory data analysis (EDA) is a way of exploring data sets to find patterns, anomalies, and insights using statistics and visuals.

tempmax		tempmin		temp		feelslikemax		feelslikemin		feelslike	
Min.	:10.10	Min.	:-0.20	Min.	: 6.40	Min.	:10.10	Min.	:-5.40	Min.	: 2.60
1st Qu.	:25.00	1st Qu.	:10.00	1st Qu.	:16.80	1st Qu.	:25.00	1st Qu.	: 8.90	1st Qu.	:16.70
Median	:28.00	Median	:14.20	Median	:20.50	Median	:27.40	Median	:14.20	Median	:20.40
Mean	:27.89	Mean	:13.49	Mean	:19.99	Mean	:27.23	Mean	:12.94	Mean	:19.68
3rd Qu.	:31.10	3rd Qu.	:17.00	3rd Qu.	:22.90	3rd Qu.	:29.82	3rd Qu.	:17.00	3rd Qu.	:22.70
Max.	:40.40	Max.	:23.20	Max.	:31.30	Max.	:44.40	Max.	:23.20	Max.	:29.90

dew		humidity		precip		precipcover		windspeed		sealevelpressure	
Min.	:-13.400	Min.	:11.60	Min.	: 0.000	Min.	: 0.00	Min.	: 7.90	Min.	: 996
1st Qu.	: 4.600	1st Qu.	:43.20	1st Qu.	: 0.000	1st Qu.	: 0.00	1st Qu.	: 18.40	1st Qu.	:1018
Median	: 9.800	Median	:57.10	Median	: 0.000	Median	: 0.00	Median	: 22.30	Median	:1021
Mean	: 9.126	Mean	:56.24	Mean	: 1.639	Mean	: 15.29	Mean	: 23.29	Mean	:1021
3rd Qu.	:14.200	3rd Qu.	:69.83	3rd Qu.	: 0.600	3rd Qu.	:16.67	3rd Qu.	:27.70	3rd Qu.	:1023
Max.	:20.000	Max.	:95.20	Max.	:99.500	Max.	:100.00	Max.	:111.20	Max.	:1034

cloudcover		visibility		solarradiation		solarenergy		uvindex		rain_today	
Min.	: 0.00	Min.	: 6.10	Min.	: 9.3	Min.	: 0.80	Min.	: 0.000	Min.	:0:4153
1st Qu.	: 3.40	1st Qu.	:14.90	1st Qu.	:217.6	1st Qu.	:18.80	1st Qu.	: 8.000	1st Qu.	: 699
Median	:26.30	Median	:18.20	Median	:255.9	Median	:22.10	Median	: 9.000		
Mean	:34.09	Mean	:18.21	Mean	:262.0	Mean	:22.63	Mean	: 8.757		
3rd Qu.	:61.50	3rd Qu.	:20.70	3rd Qu.	:309.0	3rd Qu.	:26.70	3rd Qu.	:10.000		
Max.	:100.00	Max.	:30.00	Max.	:391.5	Max.	:33.80	Max.	:10.000		

rain_tomorrow	
0:	4154
1:	698

Figure 13: Summary Statistics of Variables

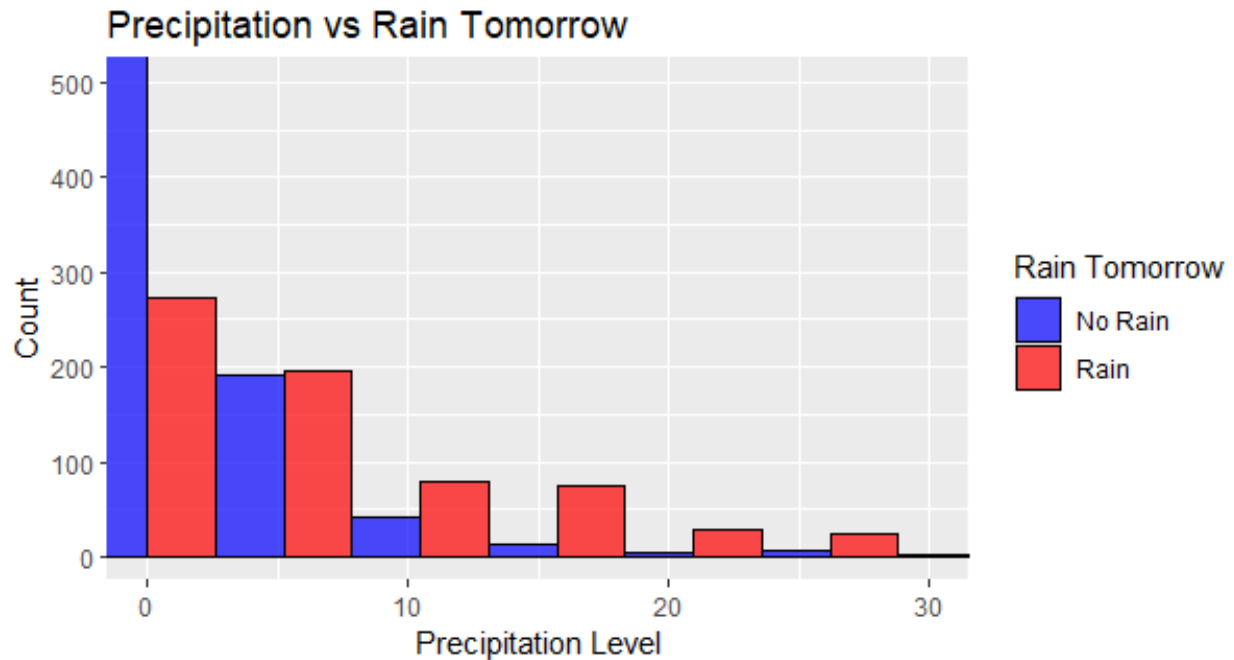


Figure 14: Precipitation vs Rain Tomorrow

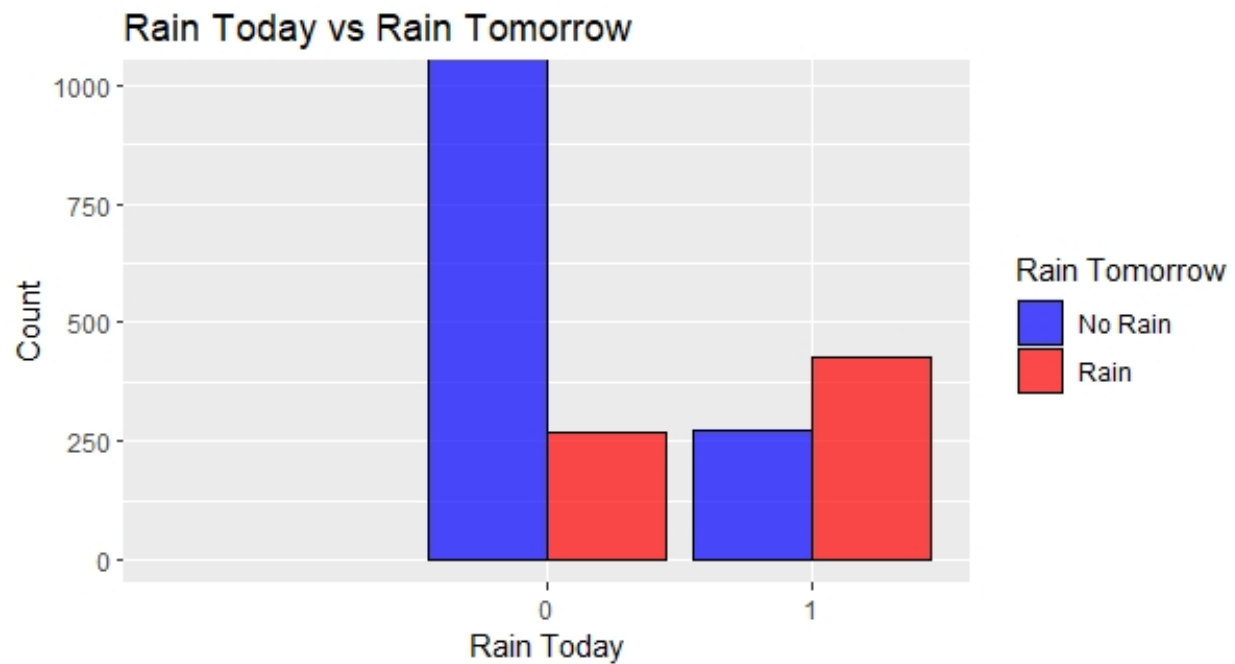


Figure 15: Rain Today vs Rain Tomorrow

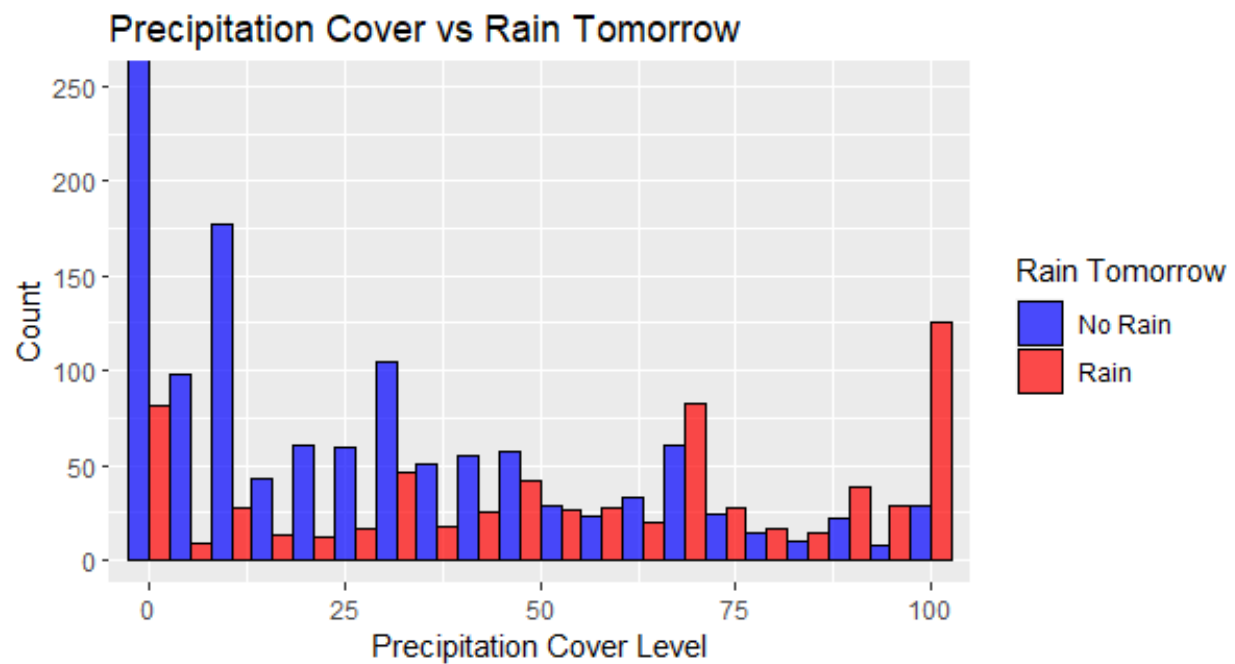


Figure 16: Precipitation Cover vs Rain Tomorrow

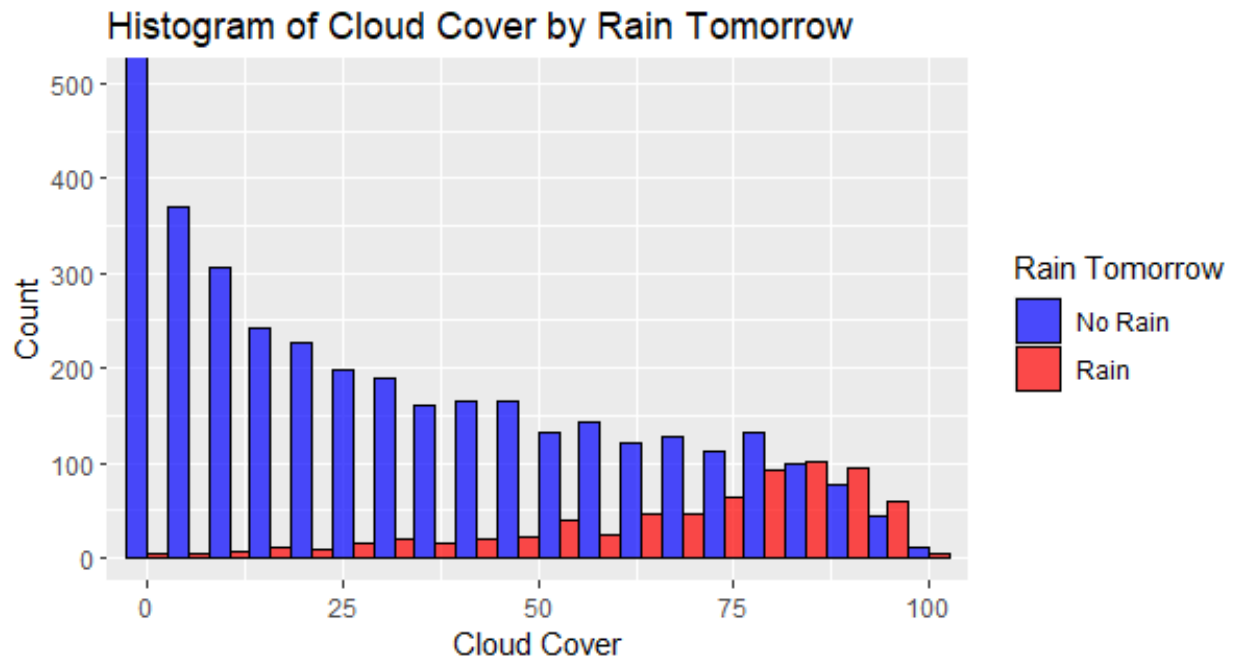


Figure 17: Cloud Cover vs Rain Tomorrow

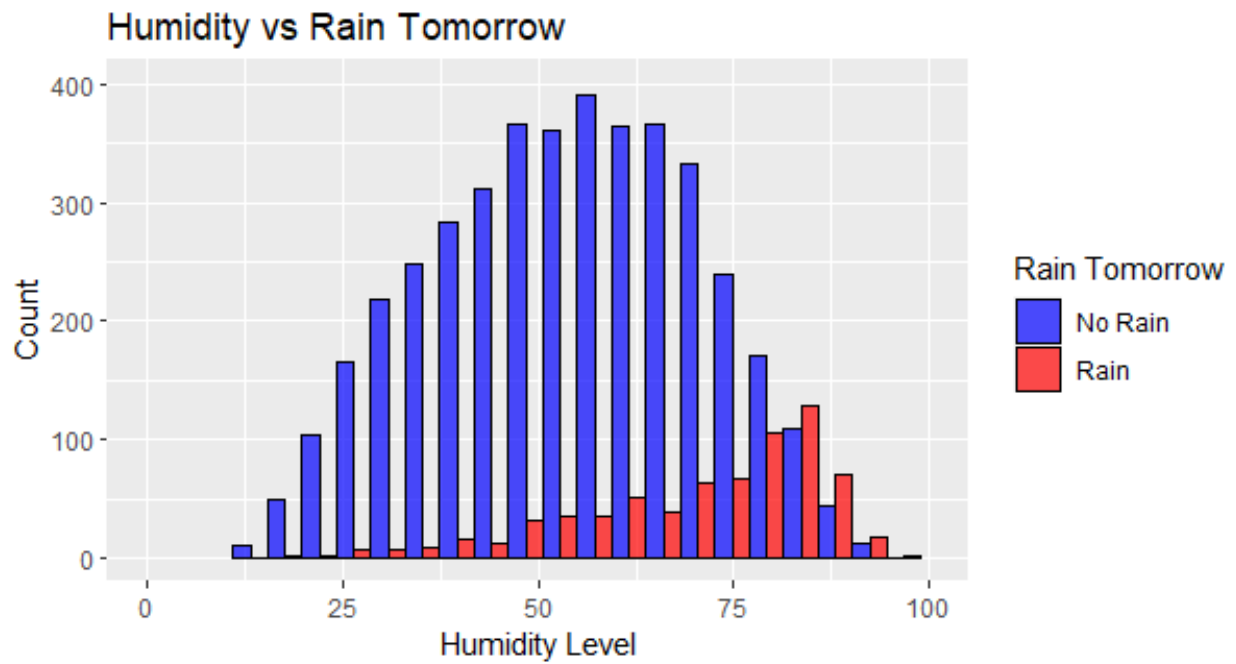


Figure 18: Humidity vs Rain Tomorrow

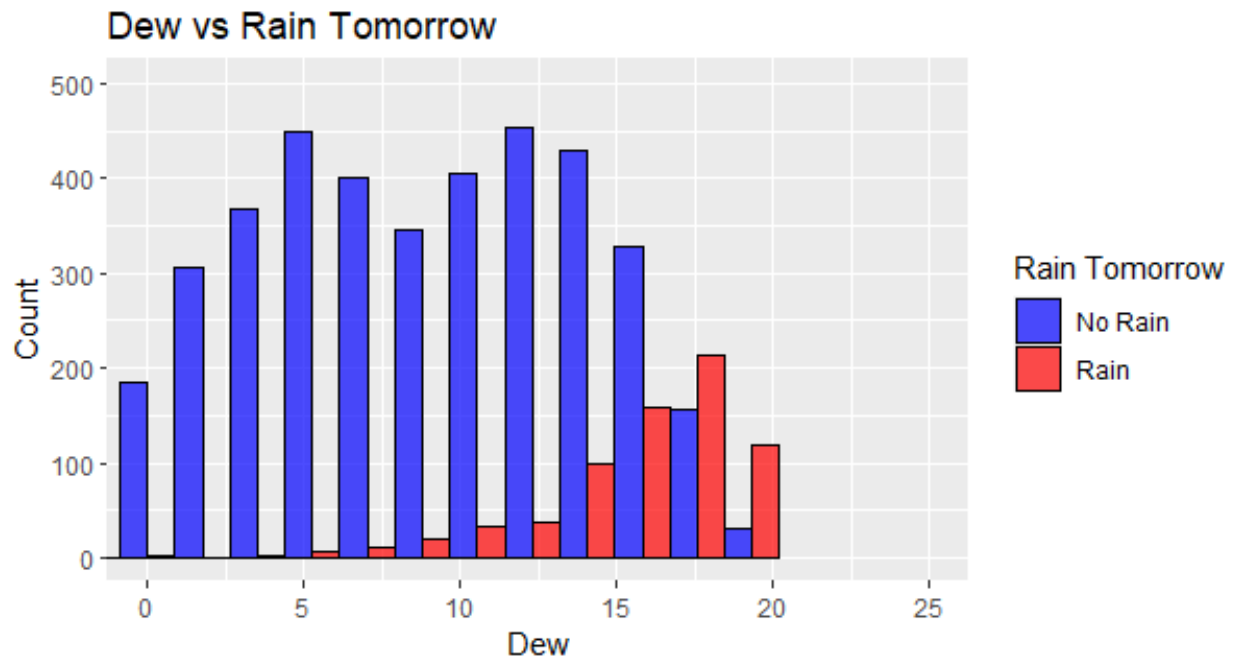


Figure 19: Dew vs Rain Tomorrow

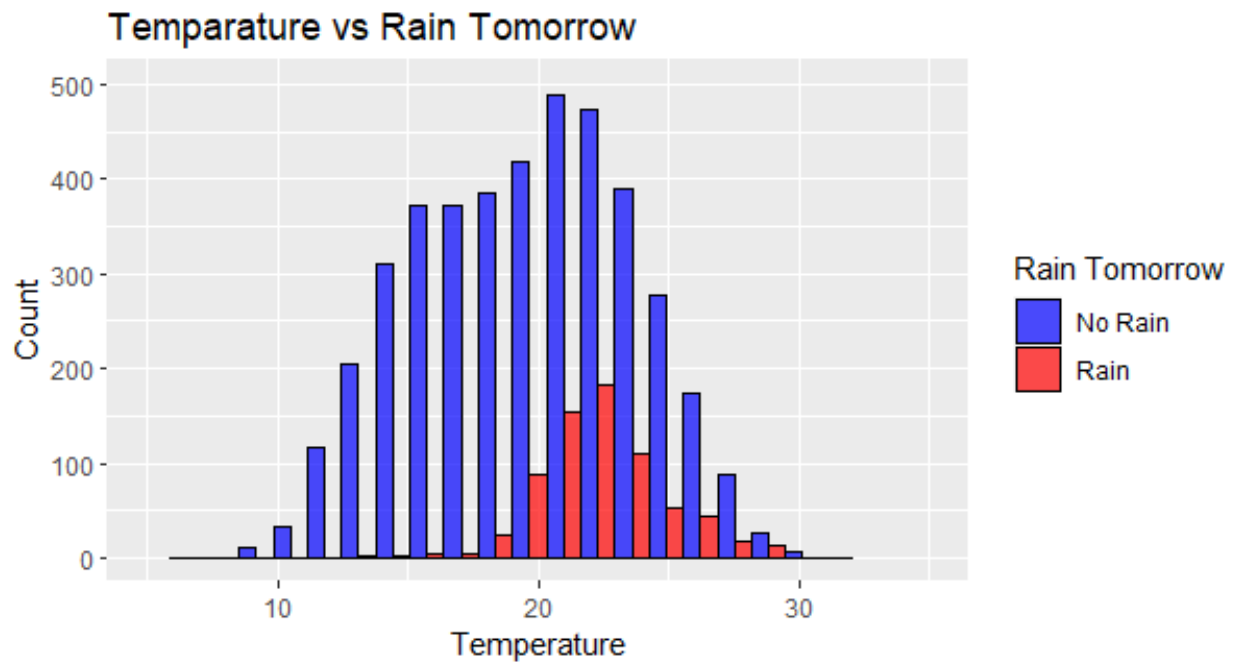


Figure 20: Temperature vs Rain Tomorrow

From the above charts it can be seen that the higher the prevalence of a predictor variable the higher the likelihood of rain tomorrow which is indicative of good feature selection.

## 4.5 Model Building

Model building in machine learning refers to the process of creating a predictive or descriptive model from input data. This process involves selecting a suitable algorithm, training the model on a labelled dataset (for supervised learning), and optimizing its parameters to make accurate predictions or generate insights. Murphy (2012).

In this section, we will construct, compare, and evaluate the predictive accuracy of the following models: Decision tree classifiers and two ensemble methods—Random Forest, which uses Bagging (Bootstrap Aggregating), and Gradient Boosting, which involves Boosted Trees. These models will be used to predict whether it will rain tomorrow.

First, we will partition our data into training and testing sets before proceeding with model construction.

### 4.5.1 Data Partition

In machine learning, data partition refers to the process of dividing a dataset into separate subsets for different purposes, such as training, validation, and testing. The purpose of data partitioning is to ensure that machine learning models are trained, evaluated, and tested on distinct sets of data, which helps to assess their performance and generalization ability accurately.

We split the data into training and testing data sets.

training	3883 obs. of 19 variables
testing	969 obs. of 19 variables

### 4.5.2 Conditional Inferencing Trees (CTREE)

The decision tree model using the CTREE algorithm with the training dataset was constructed as shown below.

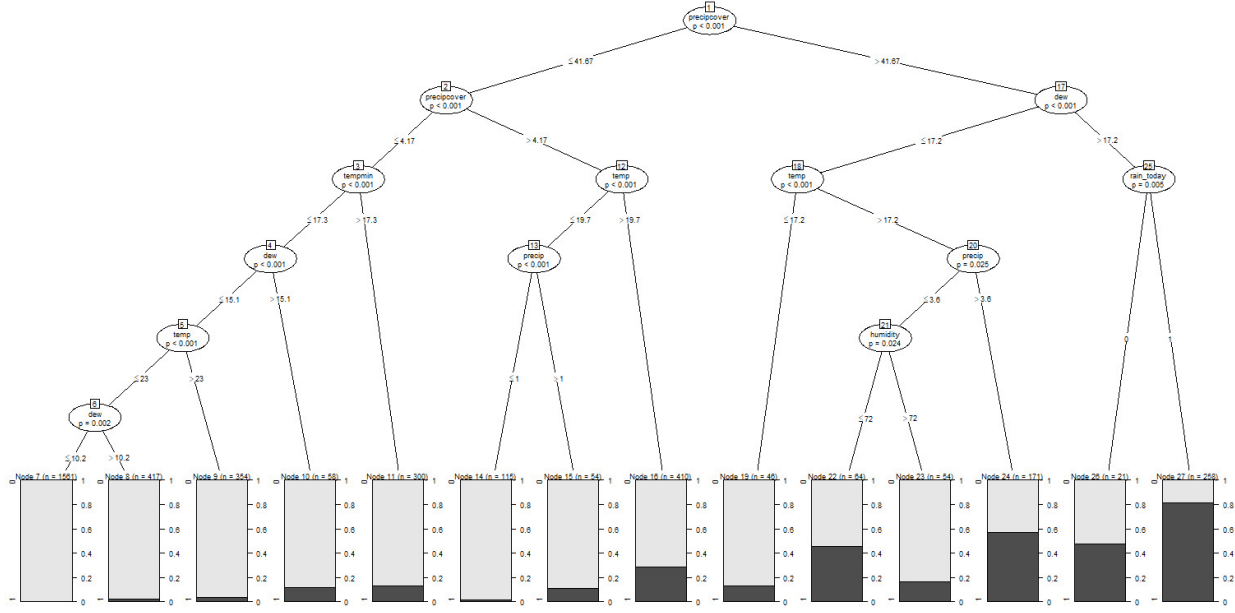


Figure 21: CTREE Decision Tree

### Interpreting the decision tree

From the above decision tree it can be seen that the root node is "precipcover".

1st leaf node:

If "precipcover" is less than 41.57, "precipcover" is also less than 4.17, "tempmin" is less than 17.3, "dew" is less than 15.1, temp is less than 23 and "dew" is less than 10.2 then the probability of rain tomorrow will be zero i.e. it will not rain tomorrow with certainty.

Last leaf Node:

However, given "precipcover" is greater than 41.57, dew is greater than 17.2 and given that it rained today then the probability that it will rain tomorrow is approximately 0.82, indicating that there is a very strong chance that it will rain tomorrow.

All the other leaf nodes are interpreted a similar manner as above.

After performing hyperparameter optimization, the decision tree was pruned. This process involved selecting the most effective combination of hyperparameters, which led to the reduction of unnecessary branches in the tree. As a result, the pruned decision tree is more efficient and generalizes better to new data by minimizing overfitting.

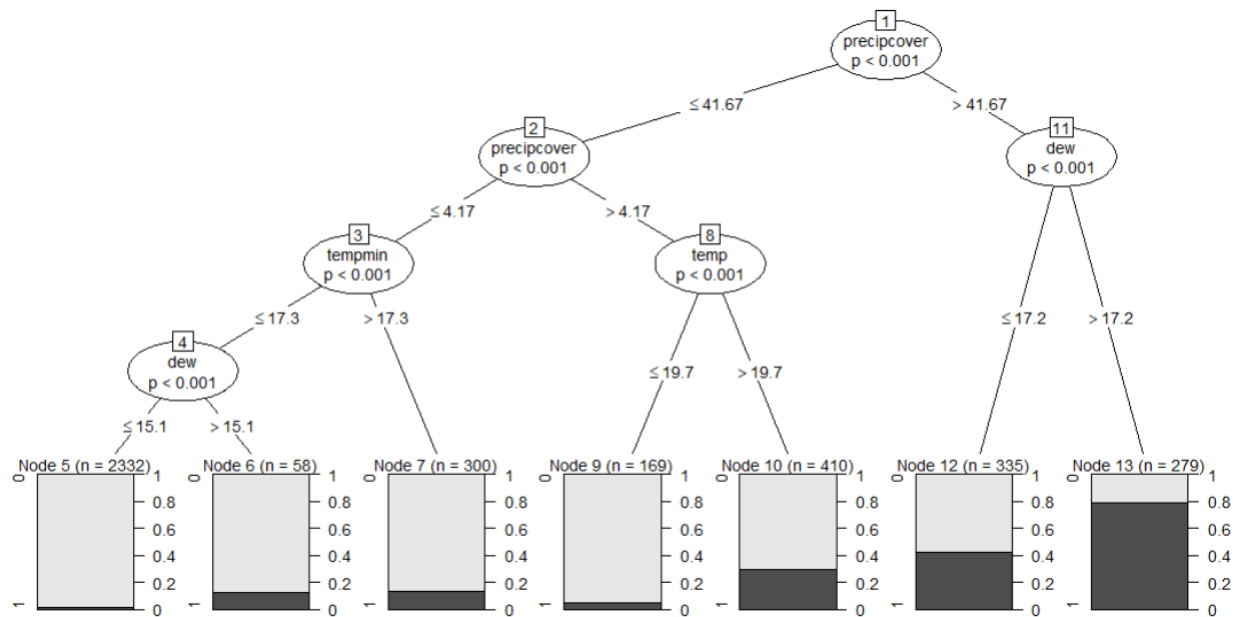


Figure 22: Pruned CTREE Decision Tree

The pruned tree offers a more succinct visualization that is easier to interpret and understand compared to the unpruned tree, which tends to overfit. Pruning removes unnecessary branches, resulting in a cleaner structure that highlights the most important splits and decisions within the data. This not only makes the model more efficient but also enhances its ability to generalize to new, unseen data by avoiding overfitting.

## Predictions

Once we have created our conditional inferencing model with `ctree`, it is time to use such a model to predict the dependent variable on the testing dataset. First, we estimated the rain tomorrow probability resulting in the vector shown below.

Resulting vector:



	0	1
1	0.9760192	0.02398082
2	0.1860465	0.81395349
3	0.1860465	0.81395349
4	0.9760192	0.02398082
5	0.9760192	0.02398082
6	0.9760192	0.02398082
7	0.9760192	0.02398082
8	0.9760192	0.02398082
9	0.9760192	0.02398082
10	0.9760192	0.02398082
11	0.9760192	0.02398082
12	0.9760192	0.02398082
13	0.9760192	0.02398082
14	0.9760192	0.02398082
15	0.9987188	0.00128123
16	0.7097561	0.29024390
17	0.9661017	0.03389831
18	0.7097561	0.29024390
19	0.5468750	0.45312500
20	0.7097561	0.29024390

Showing 1 to 20 of 969 entries, 2 total columns

The above table contains 969 rows of the testing dataset and two columns for classes 0 and 1. Class zero contains the probability of no rain and class 1 contains the probability of rain, that is, the probability that it will rain tomorrow.

Based on the rain probability, we classified the conditions for that, we needed a cutoff value. We estimated the cutoff value as the average probability of rain which was 1.143858. We then assigned zero to those observations with a rain probability less than 0.143852 and 1 to those observations with probability greater than 0.143852.

[illegible]

Now we evaluate the performance of our ctree model

```

Confusion Matrix and Statistics

          Reference
Prediction  0    1
          0 708  20
          1 122 119

          Accuracy : 0.8535
          95% CI   : (0.8296, 0.8751)
    No Information Rate : 0.8566
    P-Value [Acc > NIR] : 0.6295

          Kappa : 0.5432

  McNemar's Test P-Value : <2e-16

          Sensitivity : 0.8561
          Specificity : 0.8530
         Pos Pred Value : 0.4938
         Neg Pred Value : 0.9725
          Prevalence : 0.1434
          Detection Rate : 0.1228
    Detection Prevalence : 0.2487
         Balanced Accuracy : 0.8546

          'Positive' Class : 1

```

The proportion of correctly classified instances among all instances was found to be 0.8535 which intern is the measure of the accuracy of the decision tree model.

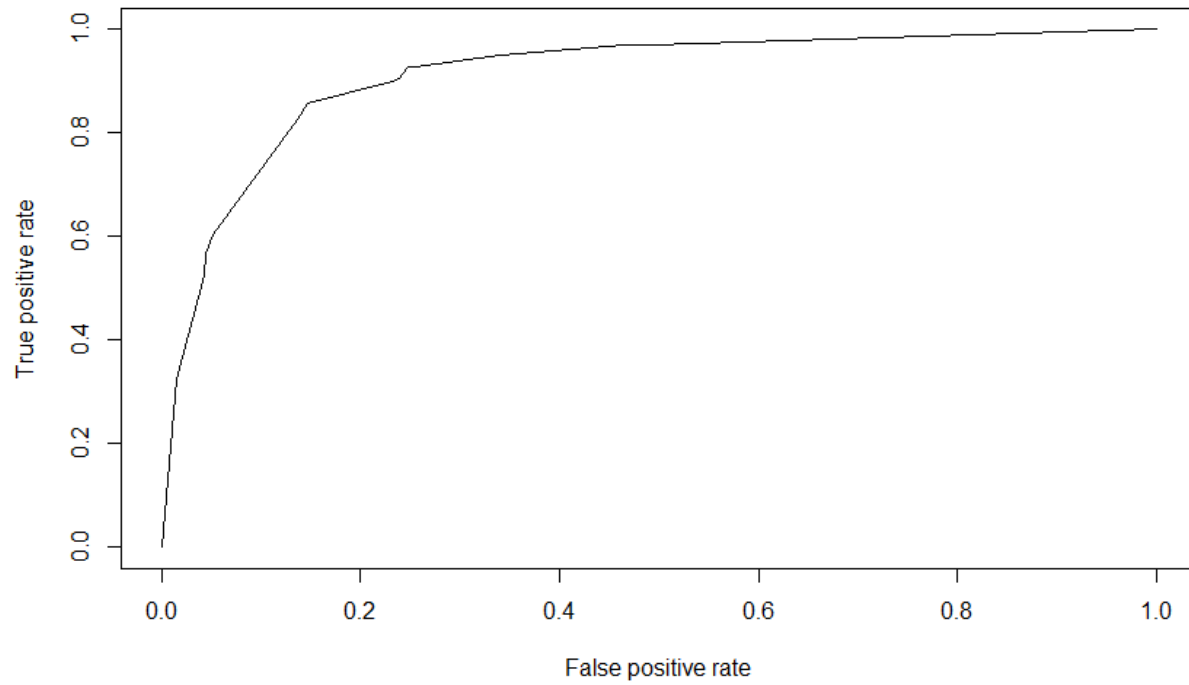


Figure 24: Area Under the ROC Curve (CTREE)  
Area Under the ROC Curve for the CTREE Model was found to be 0.9109474

From the above figure we see that the curve is close to the left corner of the plot which intern is a clear indication of good classification as was discussed in the methodology section.

### 4.5.3 Random Forest

Here we built the Random Forest model, visualized the model, and assessed the importance of different variables in predicting the target variable (rain tomorrow).

```
OOB estimate of error rate: 10.61%
Confusion matrix:
      0   1 class.error
0 3114 210  0.0631769
1  202 357  0.3613596
```

Figure 25: RF Model Accuracy

The random forest model had as seen from figure 25 has estimate error rate of 0.1061 which relatively low.

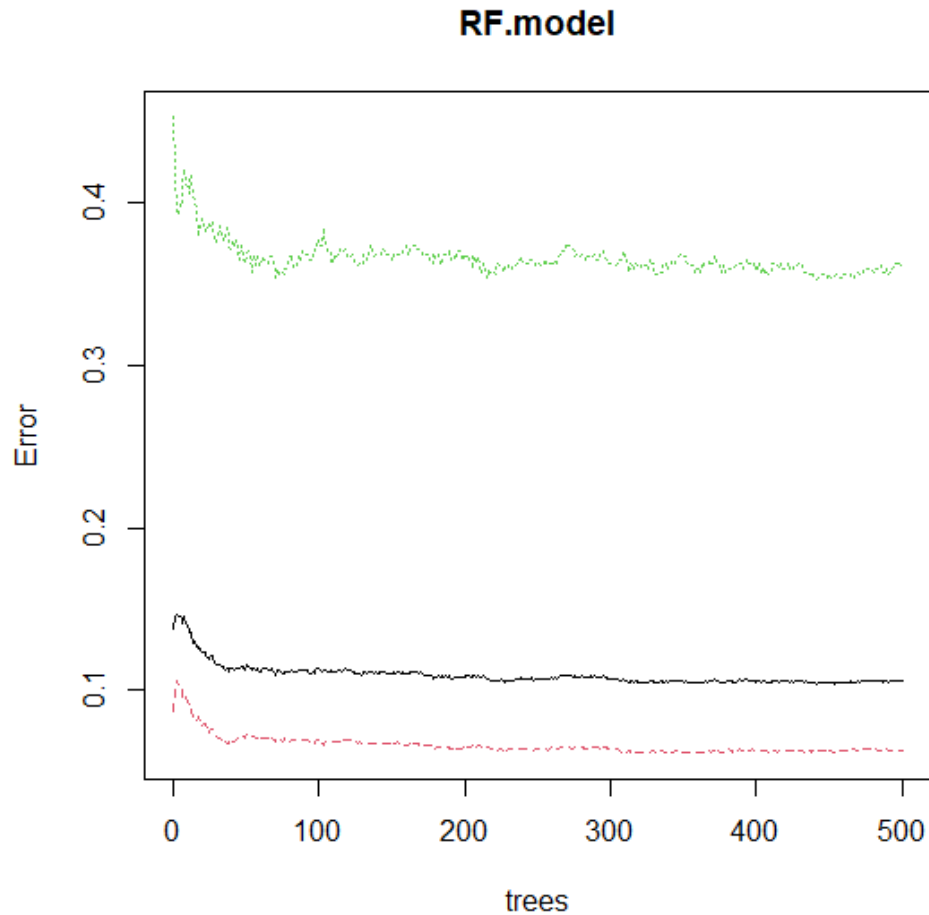


Figure 26: Random Forest Model

We can see how after 250 trees, the models have an error of approximately 0.1061. This implies that no

matter how many trees we use in the forest, we will no longer reduce the error.

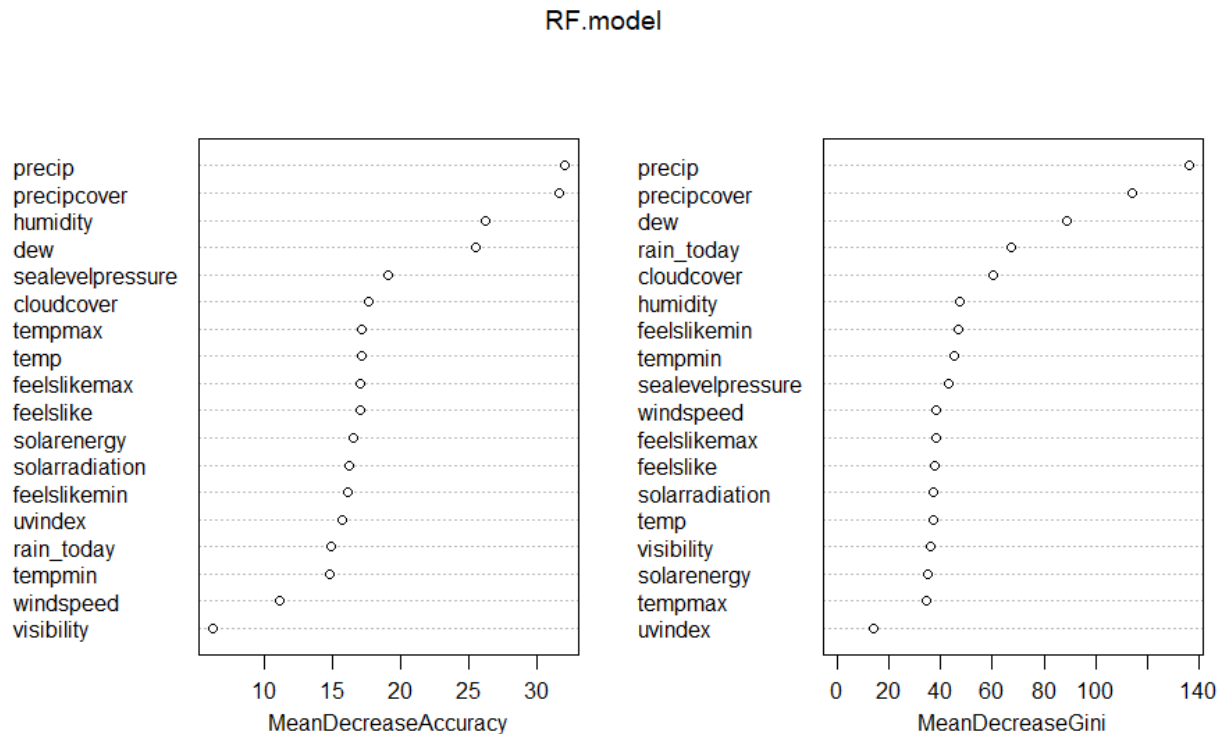


Figure 27: Importance of each variable in the Random Forest Model

### Mean Decrease Accuracy:

The importance of each feature in maintaining the overall predictive accuracy of the model is shown in figure 27. precip, precipcover, humidity and dew came out as the most important features in this case. To compute this, the Random Forest algorithm randomly permutes the values of each feature across all the out-of-bag (OOB) samples and measures the decrease in accuracy. The greater the decrease in accuracy, the more important the feature is considered to be

### Mean Decrease Gini:

This metric is based on the Gini impurity index, which is used by the Random Forest algorithm to make decisions at each split of the trees. The Gini index measures the inequality among values of a frequency distribution (e.g., probability distribution of a target variable). The Mean Decrease Gini measures the total decrease in node impurity (Gini impurity) that a feature contributes to across all the trees in the forest. Features that lead to larger decreases in Gini impurity are considered more important. In our case precip, precipcover, dew and raintoday turned out to be the most important features.

## Predictions

The vector above shows the classification of whether it rained or not, with 1 indicating rain and 0 indicating no rain.

Here we applied the confusion matrix and Area Under the ROC Curve as metrics to evaluate the performance of the model.

Figure 28: Random Forest Confusion Matrix and Statistics

51

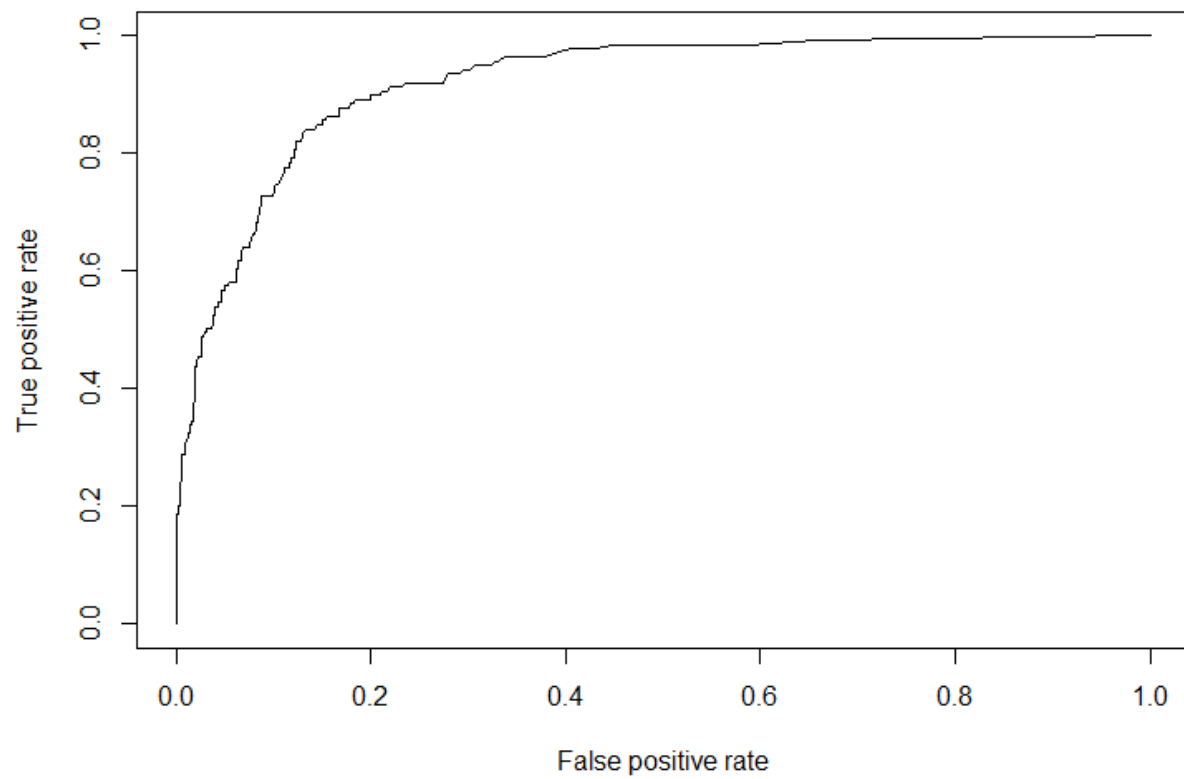


Figure 29: Random Forest ROC Curve  
Area Under the ROC Curve for the Random Forest Model was found to be 0.9212447

#### 4.5.4 Gradient Boosting with XGBoost Model

In this section, the gradient boosting model was constructed with the XGBoost package.

```
> model.XGB
##### xgb.Booster
Handle is invalid! Suggest using xgb.Booster.complete
raw: 429.8 Kb
call:
  xgb.train(params = params, data = dtrain, nrounds = nrounds,
    watchlist = watchlist, verbose = verbose, print_every_n = print_every_n,
    early_stopping_rounds = early_stopping_rounds, maximize = maximize,
    save_period = save_period, save_name = save_name, xgb_model = xgb_model,
    callbacks = callbacks, eta = 0.1, max_depth = 20, objective = "binary:logistic")
params (as set within xgb.train):
  eta = "0.1", max_depth = "20", objective = "binary:logistic", validate_parameters = "TRUE"
callbacks:
  cb.print.evaluation(period = print_every_n)
  cb.evaluation.log()
# of features: 18
niter: 50
nfeatures : 18
evaluation_log:
  iter train_logloss
    1    0.61553256
    2    0.55198645
---
    49    0.04664212
    50    0.04541589
```

As shown from the above output we trained an XGBoost model for binary classification (objective = "binary:logistic") with specific hyperparameters (eta = 0.1, maxdepth = 20) and logged the training loss over 50 iterations. The model appears to be trained successfully, with the training log showing a significant decrease in logloss from the first to the last iteration.

Now let us make the prediction using our testing dataset, classify the probabilities and assess the accuracy of our predictions using the confusion matrix.

## Predictions

[illegible]



The vector above shows the classification of whether it rained or not, with 1 indicating rain and 0 indicating no rain.

### Now we evaluate the performance of our XGBoost Model

Here we also applied the confusion matrix and Area Under the ROC Curve as metrics to evaluate the performance the model.

```
Confusion Matrix and Statistics

      Reference
Prediction 0    1
0      728    36
1      102   103

      Accuracy : 0.8576
      95% CI : (0.834, 0.879)
No Information Rate : 0.8566
P-Value [Acc > NIR] : 0.4861

      Kappa : 0.5161

McNemar's Test P-Value : 3.145e-08

      Sensitivity : 0.7410
      Specificity : 0.8771
      Pos Pred value : 0.5024
      Neg Pred value : 0.9529
      Prevalence : 0.1434
      Detection Rate : 0.1063
      Detection Prevalence : 0.2116
      Balanced Accuracy : 0.8091

      'Positive' Class : 1
```

Figure 30: XGBoost Confusion Matrix

The accuracy of our XGBoost Model as shown above was 0.8576 which is indicative of good model performance

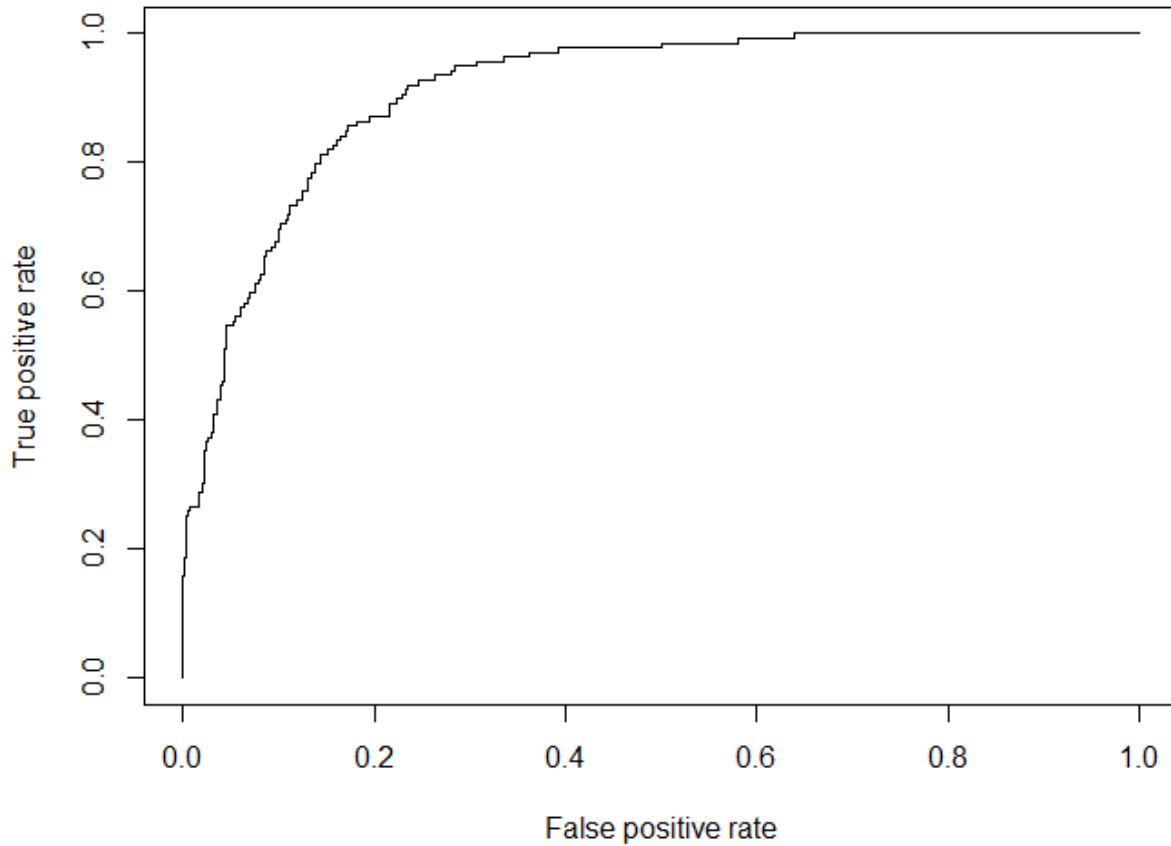


Figure 31: XGBoost ROC Curve  
Area Under the ROC Curve for the XGBoost Model was found to be 0.9131317

## 4.6 Discussion of the results

The primary aim of this research was to assess the effectiveness of Supervised Machine Learning models in short-term weather forecasting. Three models were utilized to investigate the effectiveness, namely decision trees, random forest and gradient boosting. In this section I shall present the results and conduct a comparative analysis of model performance.

### 4.6.1 Comparative Analysis of Model Performance

In my research, I used machine learning metrics for accuracy, sensitivity, specificity and the Area Under the ROC Curve (AUC) to assess model performance.

**Accuracy:** Accuracy measures the overall correctness of predictions, indicating the proportion of correctly

classified instances (both rain and no rain) out of all instances. It is a useful metric if you want to know how often your model correctly predicts whether it will rain or not. AUC (Area Under the ROC Curve): AUC measures the ability of the model to discriminate between positive and negative instances across different threshold settings. It plots the true positive rate (sensitivity) against the false positive rate (1-specificity) at various threshold settings. A higher AUC indicates better discrimination ability, i.e., the model is better at distinguishing between rainy and non-rainy days.

Three algorithms were used in this experiment with the aim of improving rain tomorrow prediction.

Model Name	Sensitivity	Specificity	Accuracy	AUC
Decision Tree (CTREE)	0.8561	0.853	0.8535	0.9109
Random Forest	0.8921	0.806	0.8184	0.9212
Gradient Boosting (XGBoost)	0.741	0.8771	0.8576	0.9131

Table 3: Displays Performance comparison of Model prediction.

From the above table we can see that both models were relatively effective in short term weather forecasting with both models registering a score above 0.8 in terms of accuracy and AUC, which intern is indicative of good model performance. Therefore, we can confidently establish that Supervised Machine Learning Models, that is, Decision Tree, Random Forest and Gradient Boosting are effective models for short term weather forecasting.

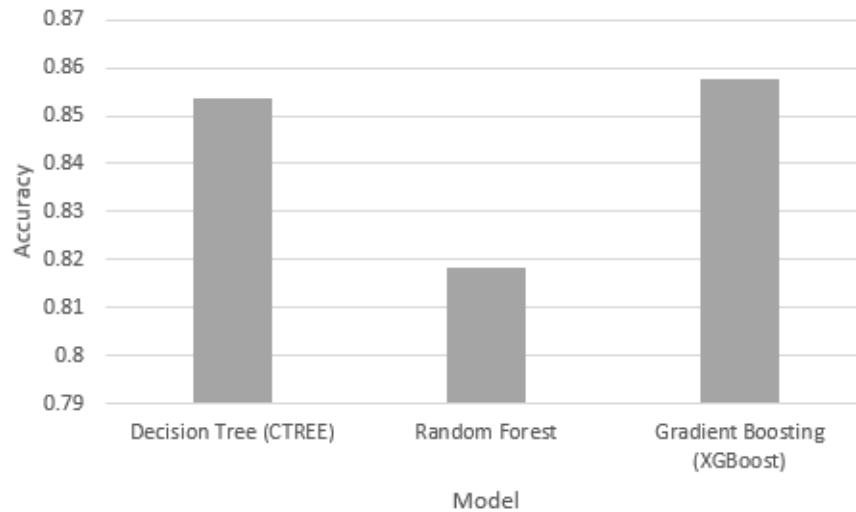


Figure 32: Displays Performance comparison of Model prediction.

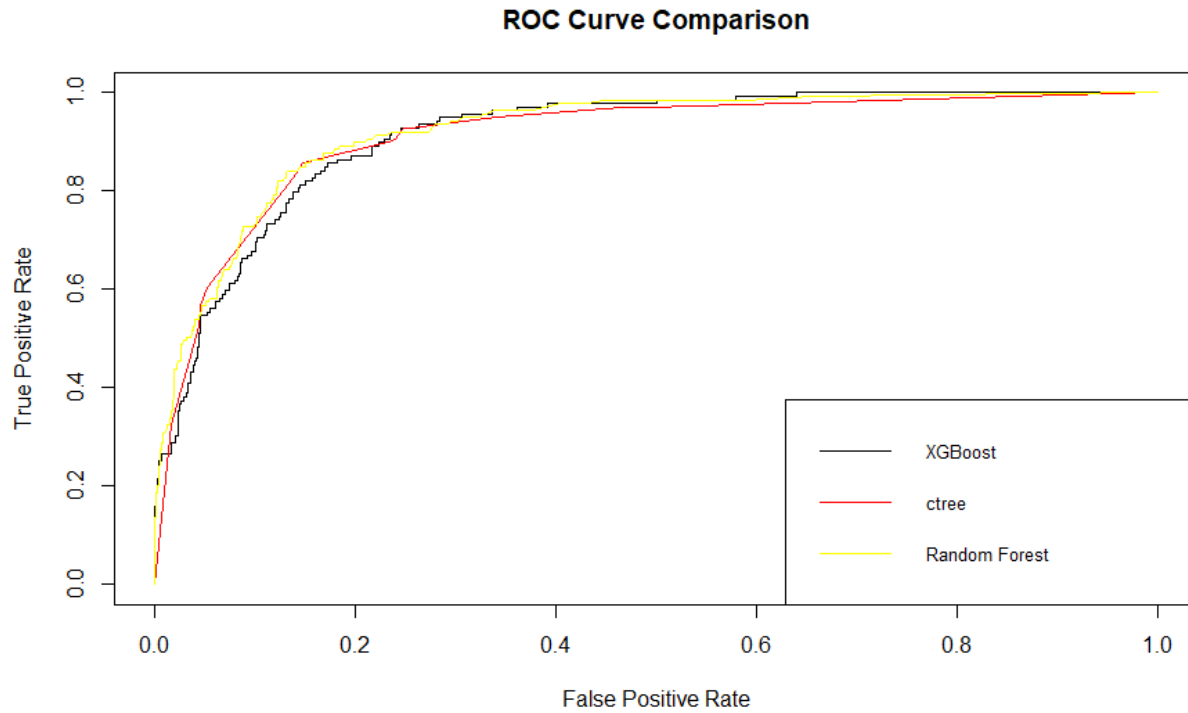


Figure 33: ROC Curve Comparison

AUC provides a more comprehensive assessment of the model's performance in distinguishing between rainy and non-rainy days, particularly when the class distribution is imbalanced as in our dataset or when both false positives and false negatives have implications. A higher AUC indicates better discrimination ability, which is crucial for predicting rain or no rain accurately, especially in real-world scenarios where the consequences of misclassification can be significant. Finally, one can establish that the Random Forest model is better in the sense it yields higher AUC than other models.

#### 4.6.2 Discussion on other author findings

In a research by Hassan et al. (2023) a systematic approach to creating an effective classification system for predicting rainfall for the next day was evaluated. The study found out that the Artificial Neural Network (ANN) classification method achieved a 0.91 accuracy rate. It also highlighted the potential for applying different machine learning methods to predict various outcomes and enhance automation in data analysis by integrating alternative algorithms. Rather than analysing datasets solely based on attributes, the research employed a complex algorithm to identify meaningful patterns, leading to better predictions and more informed decision-making. Although this research marked a significant advance in rainfall prediction, it also recognised limitations such as dependency on historical weather data and variability in data availability

across regions.

In another paper by Liyew & Melese (2021), three machine learning algorithms—Multiple Linear Regression (MLR), Random Forest (RF), and XGBoost—using data from the meteorological station in Bahir Dar City, Ethiopia were assessed. The relevant environmental features for rainfall prediction were identified using the Pearson correlation coefficient and served as input variables for the models. The study found that XGBoost outperformed MLR and RF in predicting daily rainfall amounts using these features.

Tüysüzoğlu et al. (2023) addressed the crucial environmental issue of rainfall estimation using a machine learning approach, utilizing ten years of weather data from Australia. The EK-stars method, an ensemble learning approach with multiple K-star classifiers, was proposed and found to outperform the standard K-star method, achieving a classification accuracy of 87.15 percent. Logistic Regression emerged as the best base learner within EK-stars, and sunshine was identified as the most significant predictive feature. The method demonstrated superior performance over state-of-the-art techniques and could predict drought-affected areas and provide early flood warnings. Limitations included the lack of application development and consideration of seasonal changes.

Overall, it can be concluded that supervised machine learning algorithms are effective for short-term weather forecasting.

## 5 Conclusions and Recommendations

### 5.1 Conclusions

In this research, we evaluated the effectiveness of three machine learning algorithms—Random Forest, Decision Tree, and Gradient Boosting—in predicting the likelihood of rain tomorrow. We found that Decision tree and gradient boosting outperformed random forest in terms of accuracy. However, through comprehensive analysis and comparison, we found that ensemble methods, particularly Random Forest and Gradient Boosting, outperformed the stand alone Decision Tree model in terms of ROC-AUC. Finally, we came to the conclusion that Random Forest was better in the sense that it yielded higher AUC than the others.

### 5.2 Recommendations

Based on the findings of this study, we provide the following recommendations:

**1. Utilize Ensemble Methods for Weather Prediction:**

- Given the superior performance of Random Forest and Gradient Boosting, we recommend adopting these ensemble methods for operational weather prediction tasks. Their ability to handle complex interactions and reduce overfitting makes them ideal for such applications.

**2. Model Optimization and Hyperparameter Tuning:**

- Further optimization of the Random Forest and Gradient Boosting models through hyperparameter tuning can potentially enhance their predictive performance. Techniques such as cross-validation and grid search should be employed to identify the best model parameters.

**3. Incorporate Additional Data Sources:**

- Integrating additional data sources such as real-time weather station data, satellite imagery, and historical weather patterns can provide more context and improve model accuracy. Ensemble methods can effectively handle and integrate these diverse data sources.

**4. Continuous Model Evaluation and Updating:**

- The performance of predictive models can degrade over time due to changing weather patterns and data distributions. Regularly evaluating model performance and updating the models with new data will ensure they remain accurate and reliable.

### 5.3 Further Research Study

While this study has provided valuable insights into the effectiveness of Random Forest, Decision Tree, and Gradient Boosting for predicting rain tomorrow, there are several areas that warrant further investigation:

#### 1. Exploration of Additional Machine Learning Models:

- Future research could explore other machine learning models such as Support Vector Machines (SVM), Neural Networks, and deep learning architectures to compare their performance against the models used in this study.

#### 2. Hybrid and Ensemble Techniques:

- Investigating hybrid approaches that combine multiple models or using advanced ensemble techniques like stacking and blending could potentially improve prediction accuracy and robustness.

#### 3. Long-term Weather Prediction:

- Extending the prediction horizon beyond a single day to forecast weather conditions over multiple days or weeks could be beneficial. This would require models that can handle temporal dependencies and long-term patterns.

#### 4. Incorporation of Climate Change Factors:

- Considering the impacts of climate change on weather patterns, future research should incorporate climate change variables and scenarios to assess their influence on rain prediction models.

#### 5. Real-time Data Integration:

- Integrating real-time data from various sources such as IoT sensors, weather stations, and satellites can enhance the models' predictive capabilities. Developing methods for real-time data processing and assimilation will be crucial.

#### 6. Improving Model Interpretability:

- Developing techniques to enhance the interpretability of complex models, especially ensemble and deep learning models, will help in understanding the decision-making process and gaining trust from end-users.

#### 7. Cross-Regional Model Validation:

- Validating and testing the models across different geographical regions with varying climatic conditions can help generalize the findings and ensure the models' robustness in diverse environments.

By addressing these areas, future research can build on the foundation laid by this study, leading to more accurate, reliable, and user-friendly weather prediction models.



## References

- Balehegn, M., Balehey, S. & Fu, C. (2019), ‘Indigenous weather and climate forecasting knowledge among afar pastoralists of north eastern ethiopia: Role in adaptation to weather and climate variability’, *Journal Name* .
- Behrer, P. (2023), ‘The economic benefits of weather forecasting’, <https://blogs.worldbank.org/en/impactevaluations/economic-benefits-weather-forecasting>.
- Brownlee, J. (2021), ‘An introduction to feature selection’, <https://machinelearningmastery.com/an-introduction-to-feature-selection>. Accessed: 2024.
- Cahir, J. J. (2024a), ‘Weather forecasting’, Available online at: <https://www.britannica.com/science/weather-forecasting>.
- Cahir, J. J. (2024b), ‘Weather forecasting’.
- Chantry, M., Bouallegue, Z. B., Magnusson, L., Maier-Gerber, M. & Dramsch, J. (2023), ‘The rise of machine learning in weather forecasting’.
- Conti, S. (2024), ‘Artificial intelligence for weather forecasting’.
- Corbo, A. (2023), ‘What is a decision tree?’.  
**URL:** <https://builtin.com/machine-learning/decision-tree>
- Datta, A., Si, S. & Biswas, S. (2019), Complete statistical analysis to weather forecasting.
- Demertzis, K. (2019), ‘Data preprocessing’.
- for Environmental Information, N. C. (2023), ‘Numerical weather prediction’, <https://www.ncei.noaa.gov/products/weather-climate-models/numerical-weather-prediction>.
- GeeksforGeeks (2023), ‘R programming language – introduction’.  
**URL:** <https://www.geeksforgeeks.org/r-programming-language-introduction/>
- Hammad, M. (2023), ‘What is data analysis in research methods?’.
- Hassan, M., Rony, A., Khan, M., Hassan, M., Yasmin, F., Nag, A., Zarin, T., Bairagi, A., Alshathri, S. & El-Shafai, W. (2023), ‘Machine learning-based rainfall prediction: Unveiling insights and forecasting for improved preparedness’, *IEEE Access* **11**, 132196–132222.

- Jakaria, A. H. M., Hossain, M. M. & Rahman, M. (2018), Smart weather forecasting using machine learning: A case study in tennessee.
- James, G., Witten, D., Hastie, T. & Tibshirani, R. (2013), ‘An introduction to statistical learning: with applications in r’, *Journal Name* .
- Jani, N., Gupta, R., Bharti, D. & Jain, D. (2022), *Advancements in Weather Forecasting With Deep Learning*, pp. 75–86.
- Koehrsen, W. (n.d.), ‘Random forest simple explanation’, <https://williamkoehrsen.medium.com/random-forest-simple-explanation-377895a60d2d>. Accessed: 2024.
- Lau, D. C. H. (2019), ‘5 steps of a data science project lifecycle’.  
**URL:** <https://towardsdatascience.com/5-steps-of-a-data-science-project-lifecycle-26c50372b492>
- Let’s Talk Science (2023), ‘Why is the weather so hard to predict?’, <https://letstalkscience.ca/educational-resources/stem-in-context/why-weather-so-hard-predict>.
- Liu, Q. & Wu, Y. (2012), ‘Supervised learning’.
- Liyew, C. & Melese, H. (2021), ‘Machine learning techniques to predict daily rainfall amount’, *Journal of Big Data* **8**.
- Mohammed, M., Khan, M. & Bashier, E. (2016), *Machine Learning: Algorithms and Applications*, CRC Press.
- Murphy, K. P. (2012), *Machine Learning: A Probabilistic Perspective*, The MIT Press.
- Novaes, M., de Carvalho, O. L., Ferreira, P., Tiraboschi, T., Silva, C., Zambrano Contreras, J., Gomes, C., Miranda, E., de Carvalho Júnior, O. & De Bessa Junior, J. (2021), ‘Prediction of secondary testosterone deficiency using machine learning: Comparative analysis of ensemble and base classifiers, probability calibration, and sampling strategies in a slightly imbalanced dataset’, *Informatics in Medicine Unlocked* **23**, 100538.
- Oxford University Press (2010), *Oxford English Dictionary*, 3rd edn, Oxford University Press, Oxford, England.
- Reilly, J. (2023), ‘Using machine learning for accurate weather forecasts in 2023’, <https://www.akkio.com/post/weather-prediction-using-machine-learning>.

Staff, C. (2024), ‘What is machine learning?’.

**URL:** <https://www.coursera.org/articles/what-is-machine-learning>

The365team (2024), ‘Introduction to decision trees: Why should you use them?’.

**URL:** <https://365datascience.com/tutorials/machine-learning-tutorials/decision-trees/>

Turing (n.d.), <https://www.turing.com/kb/how-data-collection-and-data-preprocessing-in-python-help-in-machine-learning>. Accessed: 2024.

Tüysüzöglü, G., Birant, K. & Birant, D. (2023), ‘Rainfall prediction using an ensemble machine learning model based on k-stars’, *Sustainability* **15**, 5889.

user2149631 (n.d.), ‘Will roc curve for a model always be symmetric if we have enough training data?’, Cross Validated. URL: <https://stats.stackexchange.com/q/264477> (version: 2018-01-31).

**URL:** <https://stats.stackexchange.com/q/264477>

Vedantu (2024), ‘Weather forecasting’, <https://www.vedantu.com/geography/weather-forecasting>.

Visualcrossing (n.d.), <https://www.visualcrossing.com/weather/weather-data-services/bulawayo/metric/last15days>. Accessed: 2024.

Wilcox, E. M. & Norris, J. R. (2008), ‘Diurnal cloudiness changes and shortwave cloud radiative effects’, *Journal Name* pp. 1677–1689.