

Faculty of Engineering and Computer Science

Department of Computer science

First semester 2022/2023

Comp4388 – Machine Learning

Project 2 Report

Prepared by:

Salem Mufarreh 1190416

Laith Isbaitan 1190628

Prepared for:

Dr. Radi Jarar

# Table of Contents

[Table of Contents 2](#_Toc126758076)

[Table of figures 3](#_Toc126758077)

[1. Introduction 4](#_Toc126758078)

[2. EDA 4](#_Toc126758079)

[2.1. Summery Statics 4](#_Toc126758080)

[2.2. Missing Values Check 5](#_Toc126758081)

[2.1. Detecting outliers 5](#_Toc126758082)

[2.2. Removing outliers 9](#_Toc126758083)

[2.3. Correlation Matrix 11](#_Toc126758084)

[3. Algorithms 12](#_Toc126758085)

[4. Results 12](#_Toc126758086)

[4.1. Decision Tree Result 13](#_Toc126758087)

# Table of figures

[Figure 1 Outliers Detected 6](#_Toc126500605)

[Figure 2 MaxTemp outliers (1-4) 7](file:///D:\Birzeit\Machine%20Learning\Project%202\Project%202%20Report.docx#_Toc126500606)

[Figure 3 MaxTemp outliers (5-8) 8](file:///D:\Birzeit\Machine%20Learning\Project%202\Project%202%20Report.docx#_Toc126500607)

[Figure 4 MaxTemp outliers (9-12) 9](#_Toc126500608)

[Figure 5 MaxTemp for Region (1-4) without outliers 10](#_Toc126500609)

[Figure 6 Decision Tree plot 13](file:///D:\Birzeit\Machine%20Learning\Project%202\Project%202%20Report.docx#_Toc126500610)

# Introduction

In this report, we will be examining a weather forecasting scenario where the goal is to predict whether it will rain tomorrow or not. To achieve this, we will be using various machine learning classifiers such as Naive Bayes, KNN, Logistic Regression, Decision Tree and SVM. Our aim is to determine the best classifier for this problem based on its performance metrics such as accuracy, precision, recall and F1-score.

In order to carry out this analysis, we will start by exploring the dataset and identifying any outliers or missing values. We will then preprocess the data and evaluate the performance of each classifier. Finally, we will compare the results of each classifier and draw conclusions on which one performs best for the weather forecasting scenario.

The report will provide a comprehensive understanding of the problem and the various methods used to solve it. It will also present the results of our analysis and highlight the strengths and limitations of each classifier. The conclusion will provide insights into the best classifier for the weather forecasting scenario and make recommendations for future work.

# EDA

## Summery Statics

Data columns (total 21 columns):  
 # Column Non-Null Count Dtype  
--- ------ -------------- -----  
 0 Date 36529 non-null datetime64[ns]  
 1 Location 36529 non-null object  
 2 MinTemp 36529 non-null float64  
 3 MaxTemp 36529 non-null float64  
 4 Rainfall 36529 non-null float64  
 5 WindGustDir 36529 non-null object  
 6 WindGustSpeed 36529 non-null float64  
 7 WindDir9am 36529 non-null object  
 8 WindDir3pm 36529 non-null object  
 9 WindSpeed9am 36529 non-null float64  
 10 WindSpeed3pm 36529 non-null float64  
 11 Humidity9am 36529 non-null float64  
 12 Humidity3pm 36529 non-null float64  
 13 Pressure9am 36529 non-null float64  
 14 Pressure3pm 36529 non-null float64  
 15 Cloud9am 36529 non-null float64  
 16 Cloud3pm 36529 non-null float64  
 17 Temp9am 36529 non-null float64  
 18 Temp3pm 36529 non-null float64  
 19 RainToday 36529 non-null object  
 20 RainTomorrow 36529 non-null object  
dtypes: datetime64[ns](1), float64(14), object(6)  
memory usage: 5.9+ MB

## Missing Values Check

Date 0  
Location 0  
MinTemp 500  
MaxTemp 369  
Rainfall 690  
WindGustDir 4936  
WindGustSpeed 4932  
WindDir9am 4477  
WindDir3pm 2103  
WindSpeed9am 831  
WindSpeed3pm 1475  
Humidity9am 667  
Humidity3pm 1333  
Pressure9am 6692  
Pressure3pm 6683  
Cloud9am 15817  
Cloud3pm 16139  
Temp9am 437  
Temp3pm 1105  
RainToday 690  
RainTomorrow 690  
dtype: int64

Two steps were made to replace the missing values:

1. Replacing the messing values in numerical columns, we used the median to replace any missing value.
2. Replacing the messing values in categorical columns we used the most used category in the column to replace it.

## Detecting outliers

To detect the outliers, we used Interquartile range (IQR) method: The IQR method calculates the range between the first and third quartiles of the data (Q1 and Q3), and any data point outside of 1.5 times the IQR is considered an outlier.

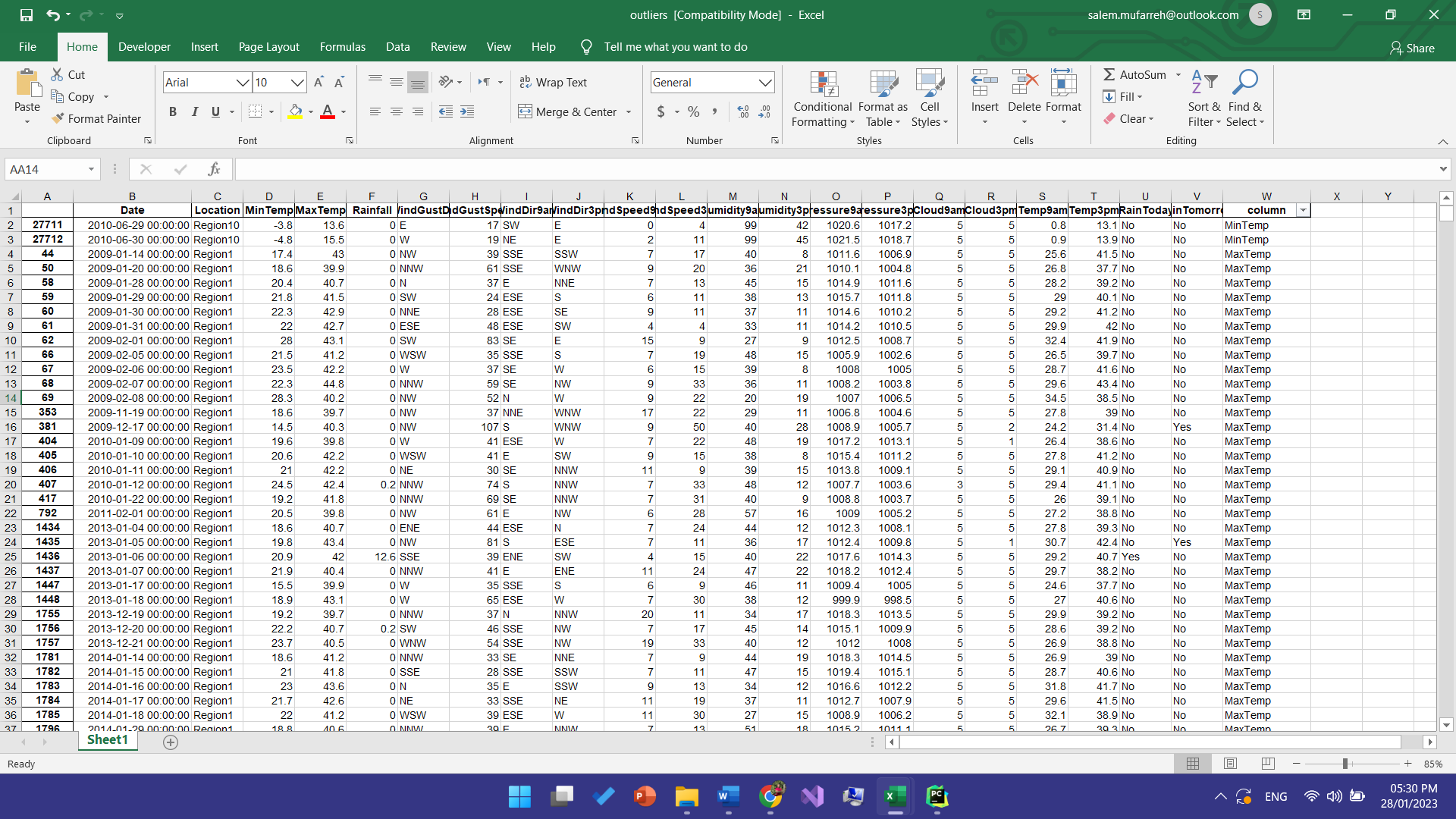


Figure 1 Outliers Detected

Looking at the outliers for region 10 we noticed that there are two outliers for minimum temperature which we can exclude from our data analysis.

For region1 we have 55 rows of outliers that say the temperature above 40 degrees are outliers, after studying the data we noticed that due to high temperature the humidity was low. Referring to the relation between humidity and temperature formula simply says they are inversely proportional. If temperature increases it will lead to a decrease in relative humidity, thus the air will become drier whereas when temperature decreases, the air will become wet means the relative humidity will increase.

We decided on eliminating the outliers that are higher than the average of these outliers. To keep as much data as possible and try to eliminate any possible outliers

Using the figures below we will notice each region and its Max Temperature outliers

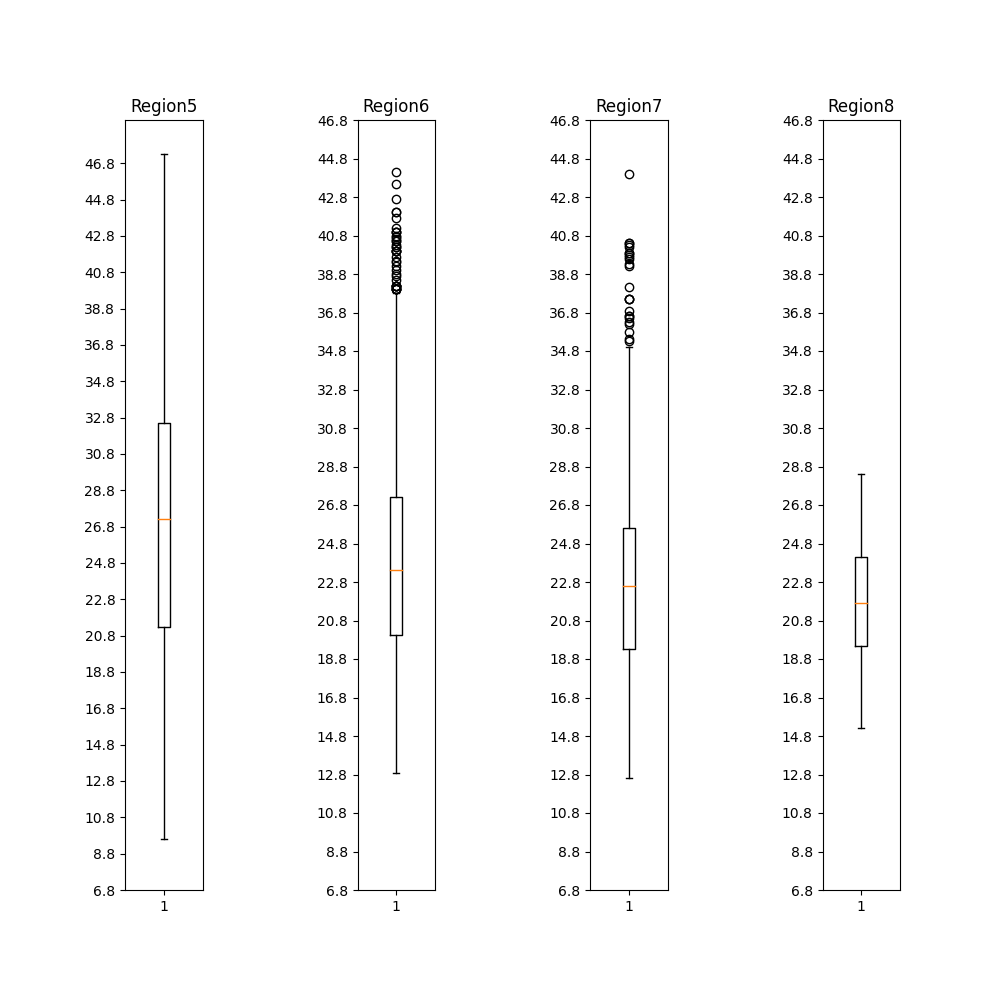
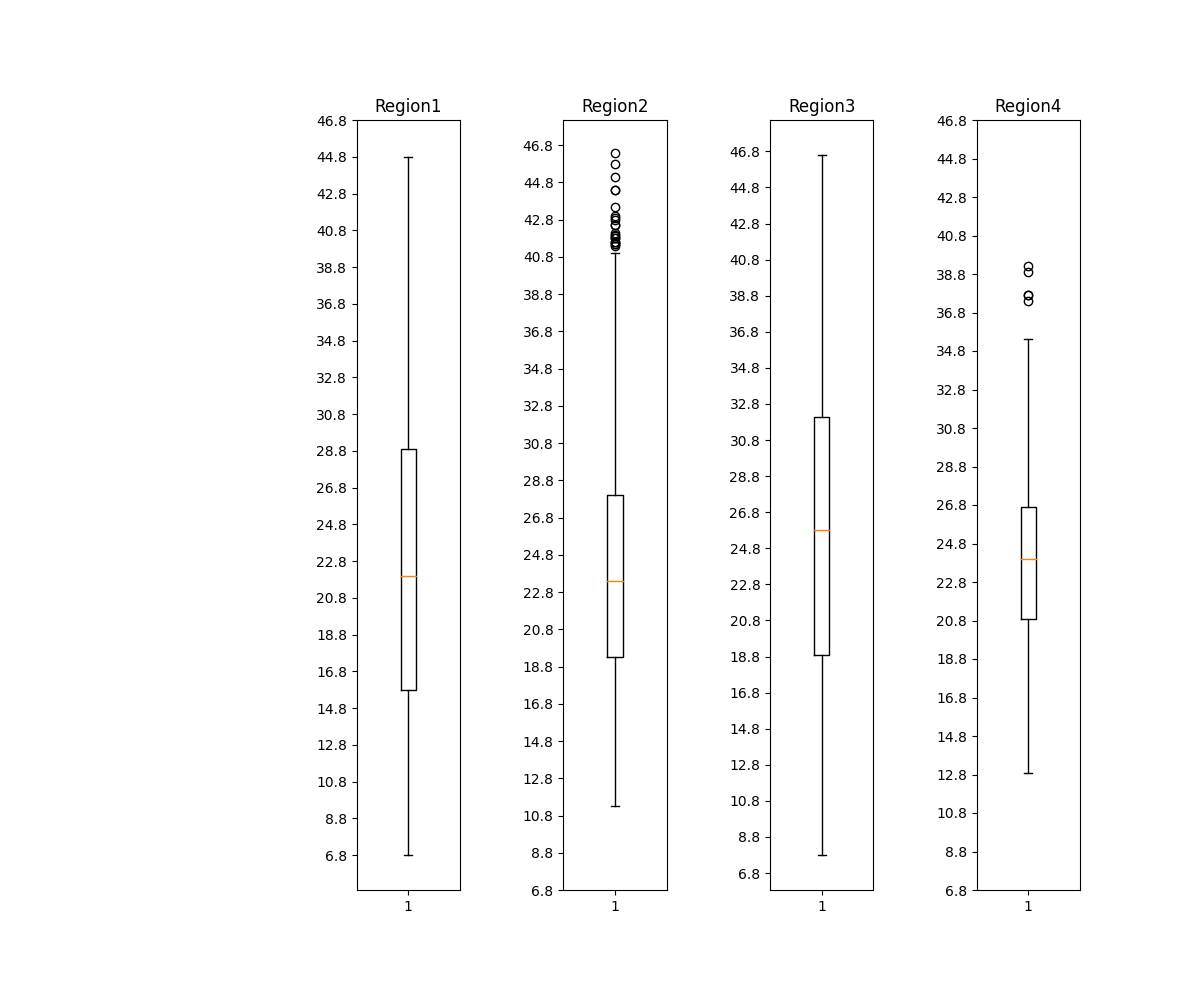


Figure 2 MaxTemp outliers (1-4)

Figure 3 MaxTemp outliers (5-8)

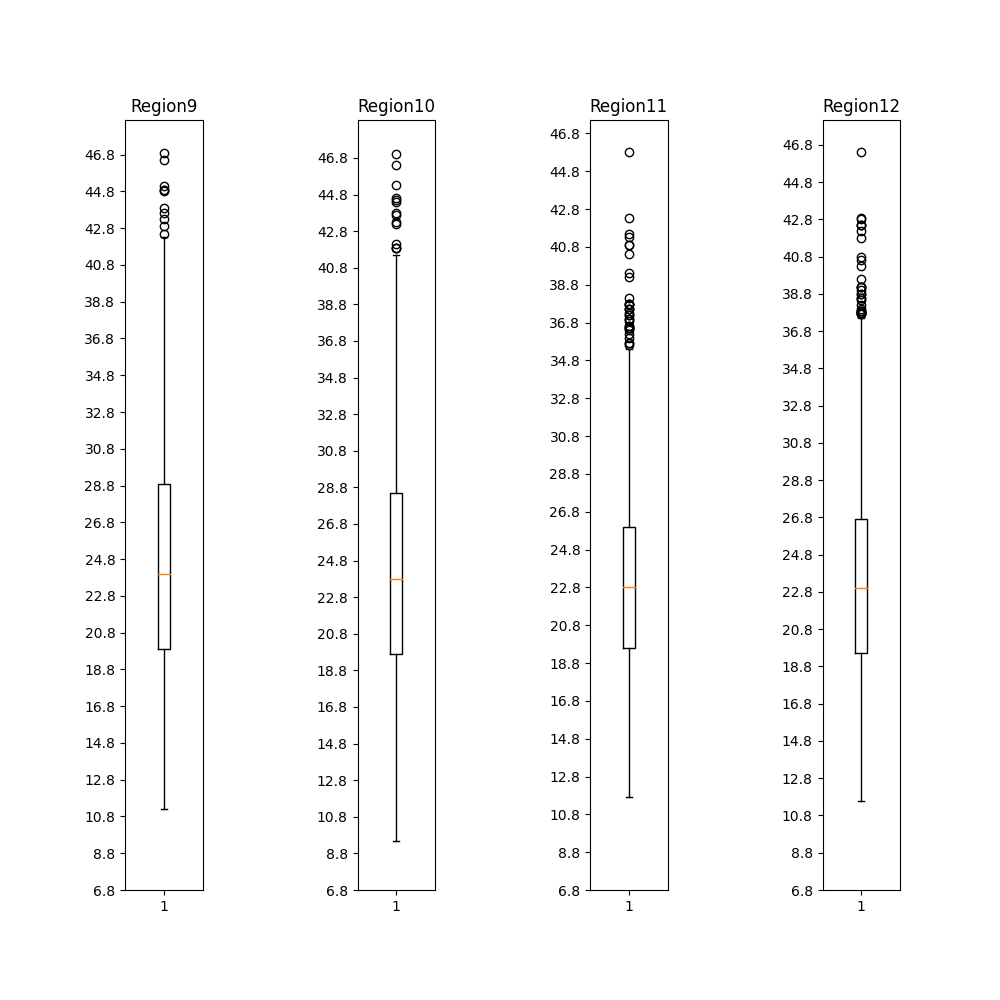


Figure 4 MaxTemp outliers (9-12)

## Removing outliers

Initially, we noticed that the presence of outliers in our data could potentially affect the results of our experiment. To better understand their impact, we first visualized the distribution of the data using box plots. After observation, it was clear that the outliers were largely due to high temperature spikes in various regions. These temperature fluctuations are not uncommon and can occur periodically in different regions.

As a first step to mitigate the impact of outliers, we decided to remove them from the dataset and ran our first set of tests, the results of which are shown in Table 1: Classifiers Results -- Removing Outliers. However, eliminating outliers also meant losing valuable information from the affected rows. To balance this, we replaced all outliers with the mean value for each respective column. This approach allowed us to retain meaningful information while still addressing the issue caused by outliers. The results of the second test are shown in Table 2: Classifiers Results -- Replacing Outliers.

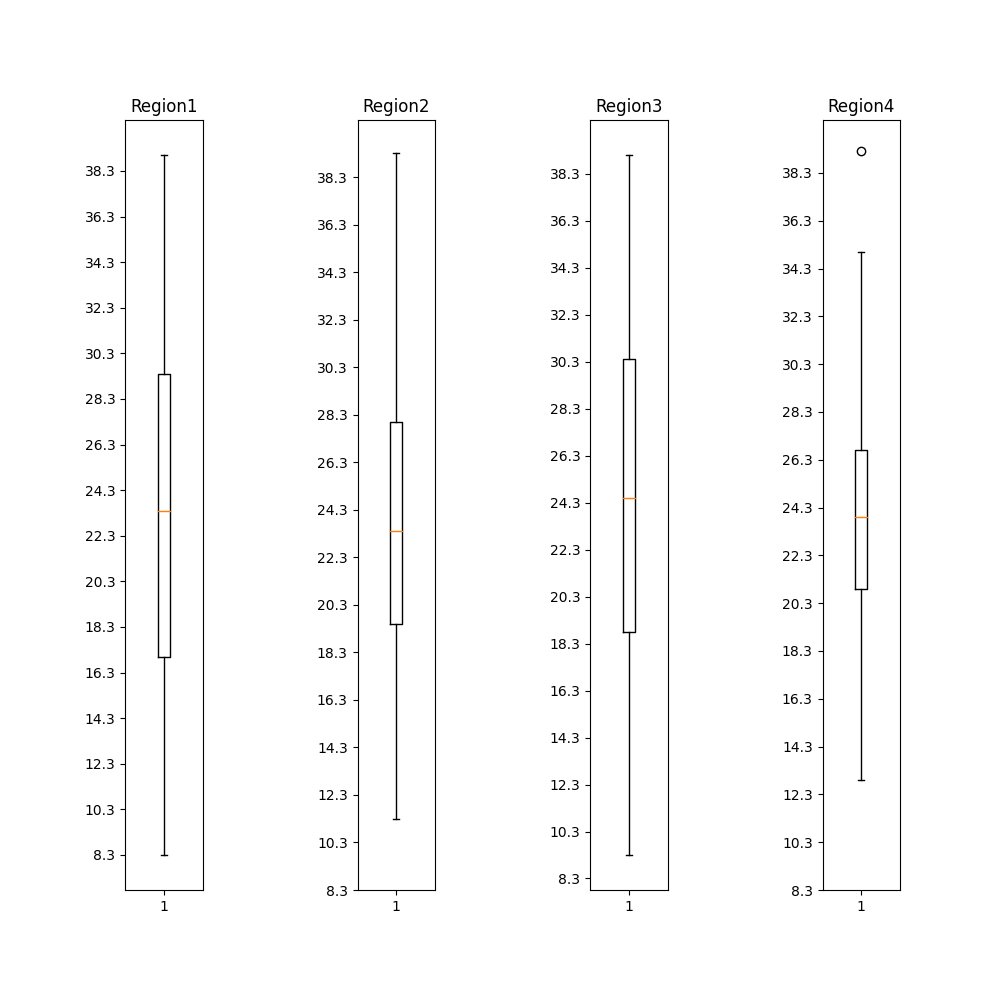
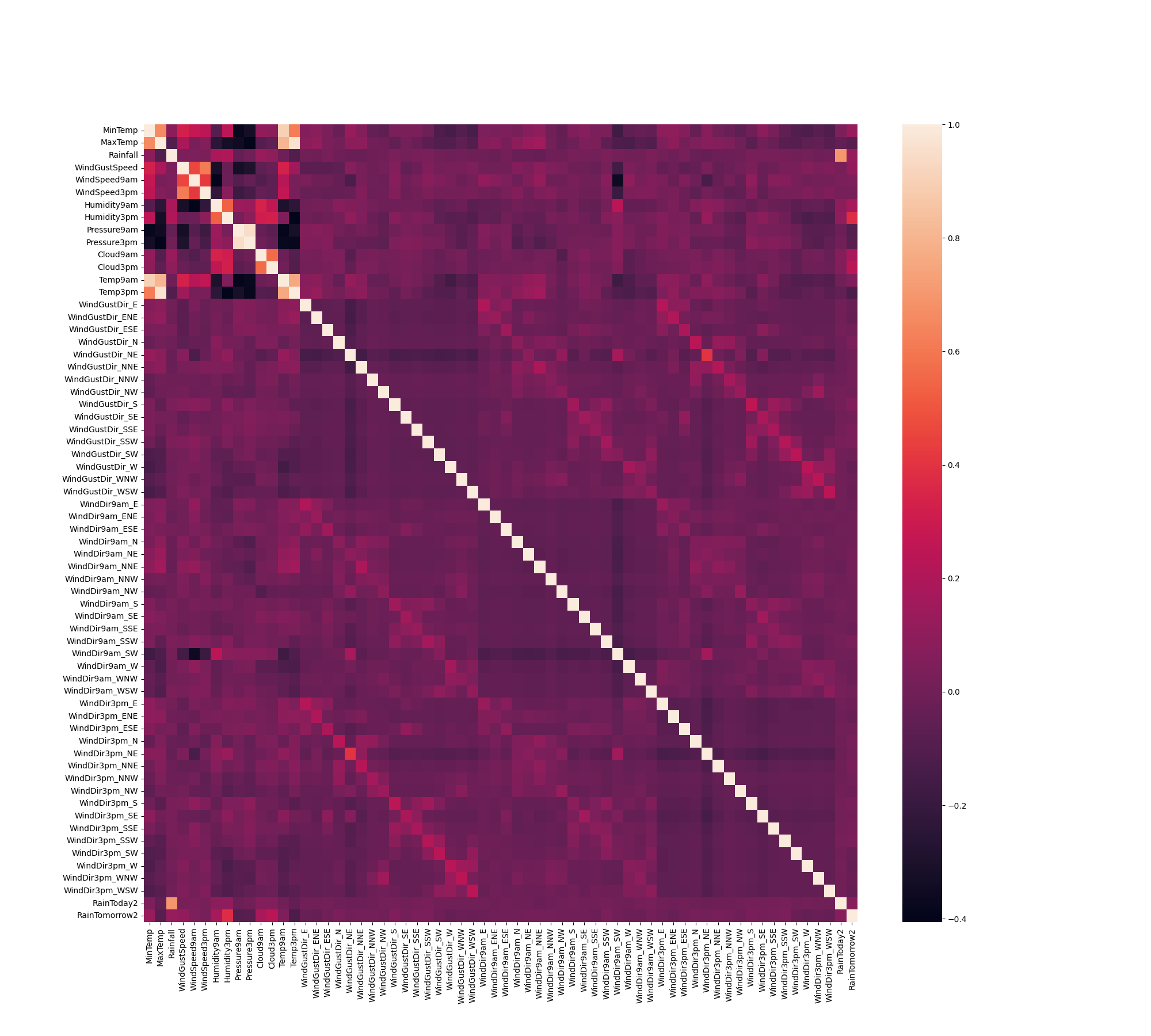


Figure 5 After Removing the outliers

## Correlation Matrix



The correlation map indicates that the target column "Rain Tomorrow" has strong correlation with columns "humidity9am", "humidity3pm", "cloud9am", and "cloud3pm". On the other hand, the correlation between "Rain Tomorrow" and columns "MaxTemp", "pressure9am", "pressure3pm", "Temp3pm", and "windDir3pm\_2" is weak.

# Algorithms

Based on the task of predicting the class target “Rain tomorrow” we choose to test the following classification algorithms:

* Naïve Bayes
* K nearest neighbor (KNN)
* Logistic Regression
* Decision Tree
* Support Vector Machine (SVM)

# Results

Removing the outliers

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Classifier** | **Accuracy** | **Precision** | **Recall** | **TP** | **FP** | **FN** | **TN** |
| ***Naïve Bayes*** | 0.74151018 | 0.28778625 | 0.5108401 | 3335 | 933 | 361 | 377 |
| ***KNN*** | 0.87195365 | 0.77094972 | 0.18699186 | 4227 | 41 | 600 | 138 |
| ***Logistic Regression*** | 0.87355173 | 0.68041237 | 0.26829268 | 4175 | 93 | 540 | 198 |
| ***Decision Tree*** | 0.87155413 | 0.69230769 | 0.23170731 | 4192 | 76 | 567 | 171 |
| ***SVM*** | 0.87435077 | 0.6925795 | 0.26558265 | 4181 | 87 | 542 | 196 |

Table 1 Classifiers Results -- Removing outliers

Replace the outliers with median

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Classifier** | **Accuracy** | **Precision** | **Recall** | **TP** | **FP** | **FN** | **TN** |
| ***Naïve Bayes*** | 0.72433616 | 0.40881590 | 0.59422111 | 4346 | 1368 | 646 | 946 |
| ***KNN*** | 0.840815767 | 0.768460575 | 0.385678392 | 5529 | 185 | 978 | 614 |
| ***Logistic Regression*** | 0.839857652 | 0.714867618 | 0.440954773 | 5434 | 280 | 890 | 702 |
| ***Decision Tree*** | 0.8362989323 | 0.7757660167 | 0.349874372 | 5553 | 161 | 1035 | 557 |
| ***SVM*** | 0.8402682727 | 0.7118644068 | 0.448492462 | 5425 | 289 | 878 | 714 |

Table 2 Classifiers Results -- Replacing outliers

As we can see from viewing the confusion matrix alone (TP, FP, FN, TN) we can observe that a large number of useful data was discarded when we used the (remove outliers) approach.

The model that had the highest accuracy, the most TP, TN and least FP, FN, after replacing outliers with median values is the KNN model.

The results of the K-Nearest Neighbors (KNN) model have an accuracy of 84.08%, which means that the model correctly classified 84.08% of the samples in the test set. The precision of the model is 76.85%, which refers to the ratio of true positive predictions to the total number of positive predictions made by the model. The recall is 38.57%, which measures the proportion of actual positive instances that were correctly identified by the model.

The true positive (TP) count is 5529, meaning the model correctly predicted 5529 positive instances. The false positive (FP) count is 185, meaning the model incorrectly predicted 185 instances as positive. The false negative (FN) count is 978, meaning the model incorrectly predicted 978 positive instances as negative. The true negative (TN) count is 614, meaning the model correctly predicted 614 negative instances.

## Decision Tree Result

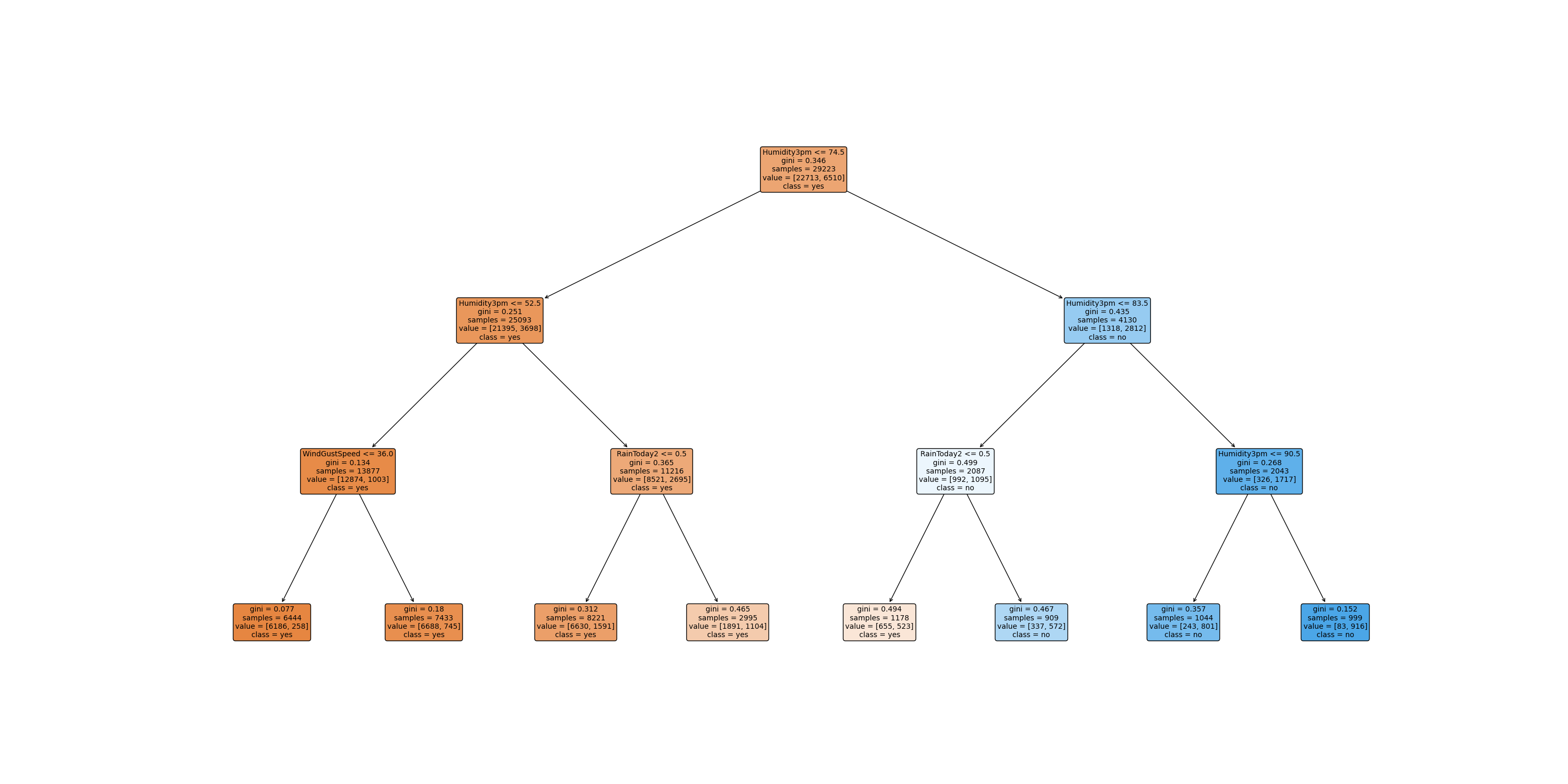


Figure 6 Decision Tree plot