MONASH University

FIT3152 Data Analytics: Assignment 2 Report

Student ID:28280016
Full Name: Zhiyue Li

Introduction

The purpose of this assignment is to become familiar with various classification models. The dataset consists of various locations' daily weather observations ranging from 2007 to 2019 in Australia. Given the modified version of the Kaggle competition data, the main task is to predict whether the following day will become cloudy or not. In other word, a model will be created to predicted if tomorrow will become cloudy for 10 random sample locations in Australia.

Data Exploration

Firstly, it is important to explore the data types in the dataset. It is easy notice that majority data type is numeric, and the small proportion of data type is character. Secondly, after browsing the data, it is easy to notice that there is a few NAs which stands for not available in the dataset. As a result, to conduct data cleaning, then any row concludes NA is removed. Thirdly, when it comes to considering proportion of cloudy days to clear days, it is around 29.73% overall. Furthermore, if the mean and standard derivation are taken into consideration for each numerical attribute, there are one attribute whose standard derivation is much higher than mean value, such as Rainfall. Furthermore, if looking into summary of each attribute (the plot/information is attached in the appendix—Figure 2), rank-ordered statistics could be viewed. Among rank-ordered statistics, the Inter Quartile Range (IQR), which is the difference between 75th and 25th percentile value, indicates the variation in the data, which is not affected by the outliers. As a result, both attribute Humidity3pm and Humidity9am have a high variation.

	MinTemp	MaxTemp	Rainfall	Evaporation	Sunshine	WindGustSpeed	WindSpeed9am
Mean	13.03	23.81	1.99	5.43	7.70	41.05	15.42
Std	6.31	7.00	6.76	3.69	3.76	13.55	8.32
	WindSpeed3pm	Humidity9am	Humidity3pm	Pressure9am	Pressure3pm	Temp9am	Temp3pm
Mean	19.68	66.04	49.60	1017.37	1014.99	17.72	22.30
Std	8.49	18.66	20.11	7.04	6.979	6.52	6.86

Table 1: Mean and Standard deviations for Attributes

Moreover, when considering the cloudy day or not, there is no point taking Day, Month and Year attribute into considerations. As a result, these attributes could be omitted before analysis to simplify the analysis. In addition, original RainToday attribute is factorised into 0 and 1. In this case, 0 stands for No raining and 1 stand for Raining. Furthermore, speaking of CloudTomorrow attribute, its 0 stands for not cloudy and 1 stand for cloudy tomorrow.

10 locations are randomly selected and predicted whether it is going to be cloudy tomorrow. 2000 rows data are also randomly selected from dataset after pre-processing. In addition, these 2000 rows data are randomly divided into 70% proportion for training and 30% proportion for testing. By the way, the training data is used to train the model and testing data is used to evaluate the performance of the model.

Classification Model

Various techniques are used to build up different classification model including Decision Tree, Naïve Bayes, Bagging, Boosting and Random Forests. After training data is applied by different techniques, and then using the previous testing data is used to evaluate how these techniques applied on classification model. To view the performance of each technique, then confusion matrix and ROC plot are good for visualisation. Confusion matrix table for each technique are appended in the appendix. The accuracy and AUC of each model are listed in the table below.

	Decision Tree	Naïve Bayes	Bagging	Boosting	Random Forest
Accuracy (%)	59.33	59.83	60.33	62.17	61.33
AUC	57%	59%	60%	62%	61%

Table 2: Accuracy of each model

Among the model, it is easy to notice that the Boosting and Random Forest can provide the highest accuracy and best AUC (Area Under the Curve) among all models. The Higher AUC value, the better the model be able to distinguish between classes (CloudTomorrow). In addition, ROC (Receiver Operating characteristics) curve is also a better visualization for multi-class classification technique, which is attached below as Figure 1. It is plotted with True Positive Rate (TPR) against False Positive Rate (FPR) where TPR is on the y-axis and FPR is on the x-axis. From the Figure 1, all classification models have an increasing trend for TPR when FPR increases. While ROC curve shows the trade-off between TPR and (1 – FPR) as well. If the classifiers can give the curves closer to the top-left corner and it indicates a better performance. It is ideal to maximize the TPR while minimizing the FPR. When FPR is lower than 0.6, Decision trees perform the worst among the model, while when FPR is higher than 0.6, the Random Forest and Boosting perform better than others.

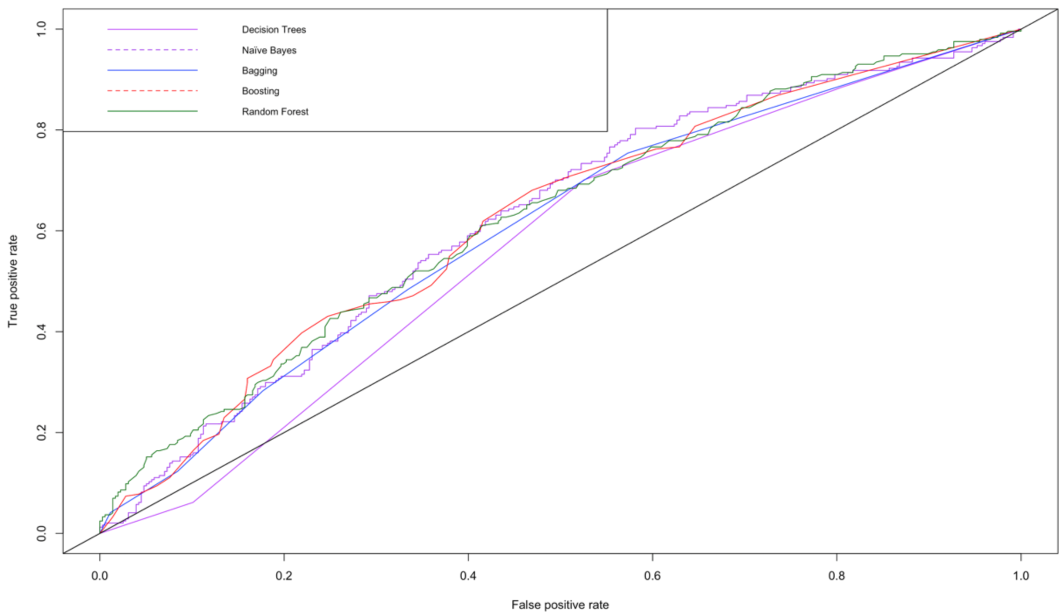


Figure 1: ROC curve for each classifier

Thirdly, it is good to determine what most important attributes affects in prediction. The importance of each variable for prediction can be read via checking the importance from model prediction's importance. Speaking of Decision Tree, the attribute importance can be read from the plot. Speaking of other classification technique, the top 7 important variables for Boosting, bagging and random Forest are listed in a Table 3.

	Random Forests	Bagging	Boosting	Decision Tree
1	Sunshine	Sunshine	Sunshine	Sunshine
2	Pressure3pm	Pressure3pm	Humidity3pm	Pressure3pm
3	Pressure9am	Humidity3pm	MinTemp	Evaporation
4	Humidity3pm	MinTemp	Temp9am	
5	MinTemp	Evaporation	Pressure3pm	
6	Temp9am	Pressure9am	Pressure9am	
7	Humidity9am	Temp9am	Humidity9am	

Table 3: Top 7 most important variables

From the table, overall, Sunshine, Pressure3pm, Humidity3pm, Pressure9am and MinTemp are top 5 important attributes. When it comes to least 5 important attributes, by checking the smallest importance for each attribute, they are RainToday, WindSpeed9am, WindGustSpeed, Temp3pm and WindSpeed3pm.

Next, after identifying most important variables and least important variables, a simpler classifier could be created to identify by hand. Firstly, as it is made by hand, 10 samples are randomly selected from the data set. Secondly, these 10 samples will be randomly divided into training and testing data set with a proportion of 70% and 30% separately. In addition, only top 5 important attributes are selected to analysis for simplicity, which are Sunshine, Pressure3pm, Humidity3pm, Pressure9am and MinTemp. (The selected data sample are listed in the Appendix Part D). In addition, as they are all numeric value for selected attribute so that they are hard to be categorised. As a result, to simplify the process, then each attribute's median value will be set as threshold value. In other word, if attribute is higher than its original median value, then it will be converted into 1 and if the attribute value is lower than its original median value, then it will be converted into 0.

MinTemp	Sunshine	Pressure9am	Pressure3pm	Humidity3pm	CloudTomorrow
0	1	1	1	0	1
0	1	0	0	0	0
0	1	1	1	0	0
1	1	0	0	1	1
0	1	1	1	0	0
0	0	1	0	0	1
1	1	0	0	0	0

Table 4: Simplified training data set

MinTemp	Sunshine	Pressure9am	Pressure3pm	Humidity 3pm	CloudTomorrow
0	1	1	1	0	Unknown
0	1	1	0	0	Unknown
0	0	0	0	1	Unknown

Table 5: Simplified testing data set

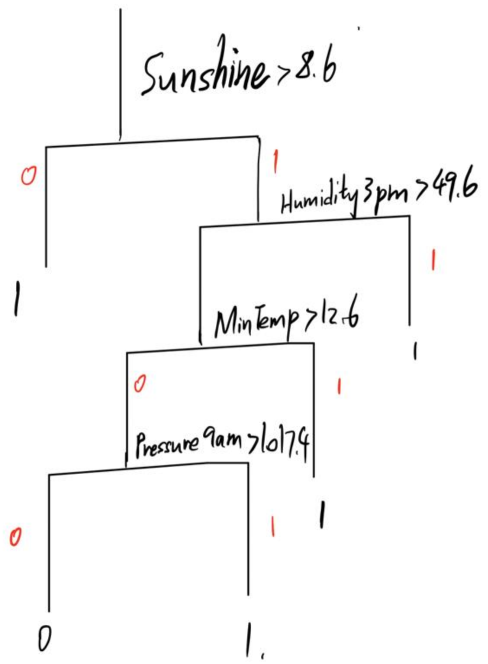


Figure 5: Simple classifier

The simple classifier is a decision tree, which repeatedly find the 'best' attribute and use this attribute to partition the data. The way how to identify the 'best' attribute is calculating the information gain and entropy. For every split, attribute with the highest information gain is selected as 'best' attribute to partition the data. After splitting the data once, it will keep selecting a new 'best' attribute and partitioning the training dataset. This process will be repeated for each non-leaf node. In the end, the simple classifier's diagram is attached above. To illustrate this simple classifier, if the Sunshine is lower than 8.6, then tomorrow is cloudy. If the Sunshine is higher than 8.6, then Humidity3pm will be checked. When inspecting the Humidity3pm, if Humidity3pm is higher than 49.6 and Sunshine is higher than 8.6, tomorrow will be cloudy. If else, then Mintemp is will be inspected again. If the Mintemp is higher than 12.6, Humidity3pm is higher than 49.6 and Sunshine is higher than 8.6, then tomorrow will be cloudy. If else, it comes to inspecting Pressure9am, if pressure9am is higher than 1017.4 Pa, then tomorrow will be cloudy otherwise it is not cloudy. When simple classifier is evaluated via using the simplified testing dataset from Table 5, both first sample's Sunshine and the third sample's Sunshine is higher than 8.6, then they are predicted to cloudy tomorrow, which is the same as original label. However, the second sample sample's Sunshine is lower than 8.6, Humidity3pm is lower than 49.6, MinTemp is lower than 12.6 and Pressure9am is lower than 1017.4, which is predicted to be not cloudy. The prediction result does not fit with original label. In conclusion, the simple classifier's accuracy is around 66.7%.

Furthermore, after removing least important variables and selecting the most 5 attributes, to build up a 'best' classifier, it is good to apply the cross-validation to improve the performance. Boosting is selected as it has the highest accuracy and AUC. After applying cross-validation, then the new accuracy is 60%, which is lower than previous boosting result. There are some potential issues causing such as result such as the original quality of dataset or my student ID. In conclusion, in this case, the best classifier is original boosting model, whose accuracy is 62% and AUC is 62% as well.

Finally, an Artificial Neural Network classifier is built after combining all insights before. Before feeding the data into the model, all selected important attributes are normalised to avoid overfitting. The 1st Artificial Neural Network plot is attached below.

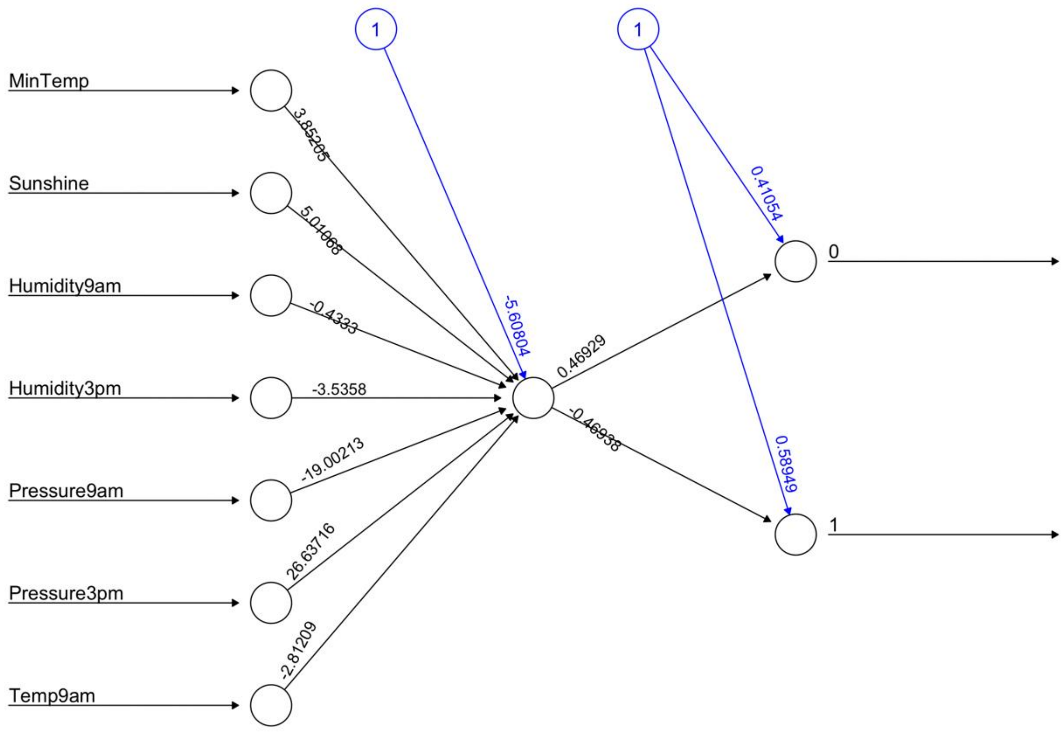


Figure 3: Artificial Neural Network with 7 inputs and 2 outputs along with 1 hidden layer

At the beginning, all the top 7 significant attributes are taken as input and used to train the neural network. Moreover, there is only 1 hidden layer as shown in the Figure 3 above. And it turns out that the accuracy is around 61% when testing. But when the hidden neural has remained the same, and evaporation attribute is added as the input as well, then accuracy remain the same when it comes to testing stage.

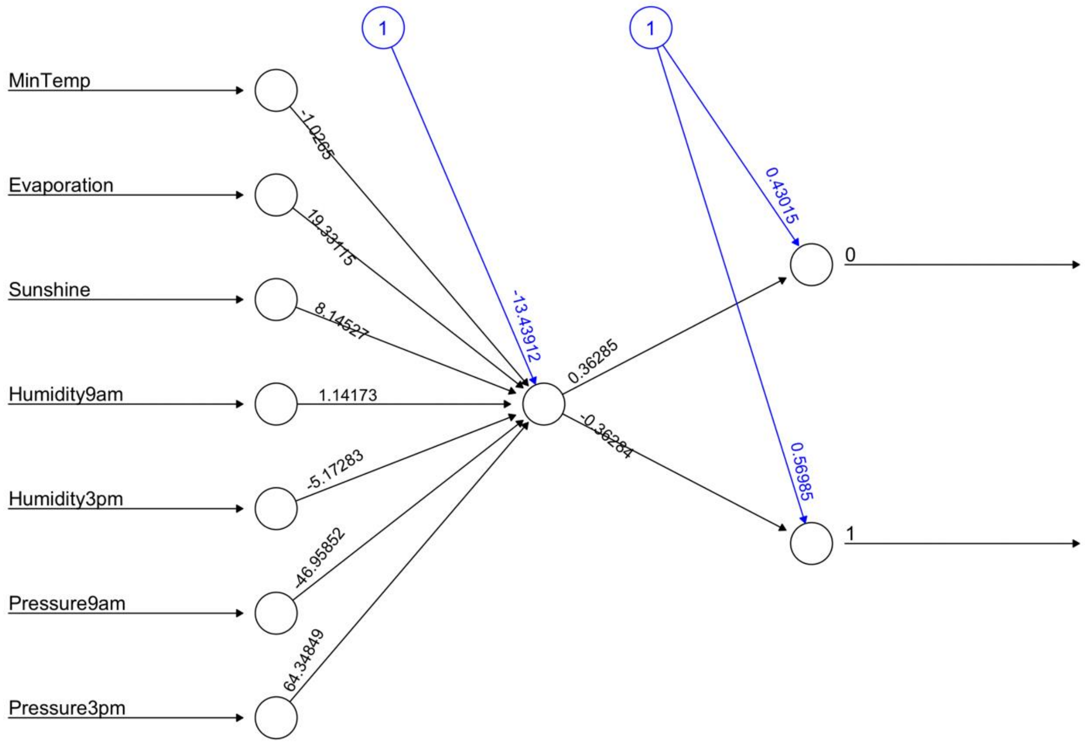


Figure 4: Artificial Neural Network with 8 inputs and 2 outputs with 1 hidden layer.

Summary

In conclusion, given the modified version of Kaggle data, to predict if tomorrow will be cloudy, after applying multiple classification techniques, it is worthy to notice that there are a few significantly domain attributes such as Sunshine, Mintemp, Pressure9am, Pressure3pm and Humidity3pm. As a result, it is easy to use these 5 top attributes to build a simple classifier by partitioning the dataset via ‘best’ attribute that can easily help partition the data. Moreover, the boosting techniques can provide the highest accuracy for prediction for cloudy day compared with other 4 techniques including Decision Tree, Random Forest, Bagging and Naïve Bayes. To improve the classification performance, cross-validation is applied, but unfortunately it seems that accuracy has not improved due to potential problems such as poor quality of data set. In the end, after combing all insight from previous section, then Artificial Neural Network model is created with 7 inputs, 1 hidden layer and 2 outputs, the accuracy is about 61%.

Reference:

1. "FIT3152 Data analytics: Assignment 2", 2021.
2. J,Young, "Rain in Australia",https://www/kaggle.com/jsphyg/weather-dataset-rattle-package, 2020.

Appendix

A. Summary plot for attributes:

```
      Day            Month              Year         Location        MinTemp          MaxTemp
 Min.   : 1.0   Min.   : 1.000   Min.   :2007   Min.   : 1.00   Min.   :-6.90   Min.   : 7.10
 1st Qu.: 8.0   1st Qu.: 4.000   1st Qu.:2010   1st Qu.:14.00   1st Qu.: 8.30   1st Qu.:18.30
 Median :16.0   Median : 7.000   Median :2012   Median :23.00   Median :12.60   Median :23.40
 Mean   :15.7   Mean   : 6.608   Mean   :2013   Mean   :25.55   Mean   :13.03   Mean   :23.81
 3rd Qu.:23.0   3rd Qu.:10.000   3rd Qu.:2015   3rd Qu.:36.00   3rd Qu.:17.80   3rd Qu.:29.20
 Max.   :31.0   Max.   :12.000   Max.   :2019   Max.   :49.00   Max.   :33.90   Max.   :48.10
    Rainfall         Evaporation        Sunshine        WindGustDir        WindGustSpeed     WindDir9am
 Min.   :  0.000   Min.   : 0.000   Min.   : 0.000   Length:31859       Min.   :  9.00   Length:31859
 1st Qu.:  0.000   1st Qu.: 2.800   1st Qu.: 5.000   Class :character   1st Qu.: 31.00   Class :character
 Median :  0.000   Median : 4.800   Median : 8.600   Mode  :character   Median : 39.00   Mode  :character
 Mean   :  1.994   Mean   : 5.433   Mean   : 7.704                      Mean   : 41.05
 3rd Qu.:  0.600   3rd Qu.: 7.400   3rd Qu.:10.700                      3rd Qu.: 48.00
 Max.   :206.200   Max.   :72.200   Max.   :14.500                      Max.   :124.00
  WindDir3pm         WindSpeed9am    WindSpeed3pm     Humidity9am      Humidity3pm      Pressure9am
 Length:31859       Min.   : 2.00   Min.   : 2.00   Min.   :  1.00   Min.   :  0.0   Min.   : 979.1
 Class :character   1st Qu.: 9.00   1st Qu.:13.00   1st Qu.: 55.00   1st Qu.: 35.0   1st Qu.:1012.8
 Mode  :character   Median :15.00   Median :19.00   Median : 67.00   Median : 50.0   Median :1017.3
                    Mean   :15.42   Mean   :19.68   Mean   : 66.04   Mean   : 49.6   Mean   :1017.4
                    3rd Qu.:20.00   3rd Qu.:24.00   3rd Qu.: 79.00   3rd Qu.: 63.0   3rd Qu.:1022.1
                    Max.   :81.00   Max.   :65.00   Max.   :100.00   Max.   :100.0   Max.   :1041.1
  Pressure3pm       Temp9am          Temp3pm       RainToday         CloudTomorrow
 Min.   : 978.9   Min.   :-0.70   Min.   : 3.9   Length:31859       Min.   :0.000
 1st Qu.:1010.3   1st Qu.:12.70   1st Qu.:16.9   Class :character   1st Qu.:0.000
 Median :1014.9   Median :17.10   Median :21.8   Mode  :character   Median :0.000
 Mean   :1015.0   Mean   :17.72   Mean   :22.3                      Mean   :0.459
 3rd Qu.:1019.7   3rd Qu.:22.70   3rd Qu.:27.4                      3rd Qu.:1.000
 Max.   :1040.1   Max.   :39.10   Max.   :46.1                      Max.   :1.000
```

Figure 2: Summary plot for all attributes

B. Matrix Confusion for each technique:

a) Decision Tree

	Actual_class	
Predicted_class	0	1
0	356	244
1	0	0

b) Naïve Bayes

	Actual_class	
Predicted_class	0	1
0	274	159
1	82	85

c) Bagging

	Actual_class	
Predicted_class	0	1
0	293	175
1	63	69

d) Boosting

	Actual_class	
Predicted_class	0	1
0	268	139
1	88	105

e) Random Forest

	Actual_class	
Predicted_class	0	1
0	286	162
1	70	82

C. Importance of attributes for prediction for boosting, bagging, random forest

```
> # Q.8. Examine each models, determine most important variables for prediction
> summary(WAUS.tree)

Classification tree:
tree(formula = CloudTomorrow ~ ., data = WAUS_training)
Variables actually used in tree construction:
[1] "Sunshine"    "Pressure3pm" "Evaporation"
Number of terminal nodes:  4
Residual mean deviance:  1.266 = 1767 / 1396
Misclassification error rate: 0.3871 = 542 / 1400
> plot(WAUS.tree)
> text(WAUS.tree,pretty = 0)
>
> sort(WAUS.bag$importance,decreasing=TRUE)
     Sunshine   Pressure3pm   Humidity3pm       MinTemp   Evaporation   Pressure9am       Temp9am
    20.721541     13.251568     11.256072      9.193332      8.428027      7.558994      6.520033
 WindSpeed3pm   Humidity9am       MaxTemp      Location WindGustSpeed      Rainfall  WindSpeed9am
     6.484383      4.598198      4.040072      2.384280      2.264041      1.728351      1.571108
    RainToday       Temp3pm
     0.000000      0.000000
>
> sort(WAUS.Boost$importance,decreasing =TRUE)
     Sunshine   Humidity3pm       MinTemp       Temp9am   Pressure3pm   Pressure9am   Humidity9am
    18.006671     10.776285     10.763839      9.518432      9.416814      6.859737      6.847930
      MaxTemp   Evaporation      Rainfall      Location WindGustSpeed  WindSpeed9am  WindSpeed3pm
     4.467570      4.106185      4.041084      3.894881      3.277100      3.216011      2.762685
      Temp3pm     RainToday
     2.044777      0.000000
>
> #sort(WAUS.rf$importance,decreasing = TRUE)
> WAUS.rf$importance[order(-WAUS.rf$importance),]
     Sunshine   Pressure3pm   Pressure9am   Humidity3pm       MinTemp       Temp9am   Humidity9am
    67.189498     58.280618     55.454519     55.421918     49.136392     48.137170     48.087760
      Temp3pm   Evaporation       MaxTemp WindGustSpeed  WindSpeed9am  WindSpeed3pm      Rainfall
    47.469300     46.039146     45.667964     37.930757     35.035952     33.202141     18.778739
     Location     RainToday
    14.836268      2.714283
```

D. Selected data for simple classifier

```
> WAUS_simple_training
      MinTemp Sunshine Pressure9am Pressure3pm Humidity3pm CloudTomorrow
41310    10.0     11.1      1019.2      1017.6          31             1
81915    12.4     11.6      1014.9      1013.9          45             0
14261     2.4      9.7      1020.9      1017.7          32             0
33766    17.6     11.5      1016.2      1014.6          55             1
24267     6.2      9.9      1019.4      1015.7          33             0
16278     9.1      4.0      1018.2      1011.9          41             1
50550    19.3     11.7      1011.3      1012.3          47             0
```

```
> WAUS_simple_testing
      MinTemp Sunshine Pressure9am Pressure3pm Humidity3pm CloudTomorrow
84424     1.5      9.2      1030.4      1025.0          32             1
29759     3.4     10.9      1017.0      1012.8          16             1
26367    12.0      8.1      1015.6      1014.1          62             1
```

E. Process to calculate the simple classifier

1. Calculate the entropy of Cloud Tomorrow from training data-set.

Yes: 3, No: 4.

$$E(S) = -\frac{3}{7}\log_2\left(\frac{3}{7}\right) - \frac{4}{7}\log_2\left(\frac{4}{7}\right)$$

$$= 0.9852$$

2. For Sunshine

sunshine	Cloud Tomorrow Yes	No	Entropy
1	2	4	$-\frac{2}{6}\times\log_2\left(\frac{2}{6}\right) - \frac{4}{6}\log_2\left(\frac{4}{6}\right) = 0.9182$
0	1	0	$-1\times\log_2(1) - 0 = 0$

$$Gain(S, sunshine) = 0.9852 - \left(0.9182\times\frac{6}{7}\right) = 0.1982$$

3. For MinTemp

MinTemp	Cloud Tomorrow Yes	No	
1	1	1	$-\frac{1}{2}\times\log_2\left(\frac{1}{2}\right) - \frac{1}{2}\times\log_2\left(\frac{1}{2}\right) = 1$
0	2	3	$-\frac{2}{5}\times\log_2\left(\frac{2}{5}\right) - \frac{3}{5}\times\log_2\left(\frac{3}{5}\right) = 0.97$

$$Gain(MinTemp) = 0.9852 - \left(\frac{5}{7}\times 0.97 + \frac{2}{7}\times 1\right) = 0.006$$

4. For Humidity 3 pm

Humidity 3pm	Cloud Tomorrow Yes	No	
1	1	0	$-1 \times \log_2(1) - 0 = 0$
0	2	4	$-\frac{2}{6} \times \log_2\left(\frac{2}{6}\right) - \frac{4}{6} \times \log_2\left(\frac{4}{6}\right) = 0.9813$

$$Gain(Humidity\ 3pm) = 0.9852 - \left(\frac{1}{7} \times 0 + \frac{6}{7} \times 0.9813\right) = 0.144$$

5. For Pressure 3 pm

Pressure 3pm	Cloud Tomorrow Yes	No	
1	1	2	$-\frac{1}{3} \times \log_2\left(\frac{1}{3}\right) - \frac{2}{3} \times \log_2\left(\frac{2}{3}\right) = 0.9183$
0	2	2	$-\frac{2}{4} \times \log_2\left(\frac{1}{2}\right) - \frac{1}{2} \times \log_2\left(\frac{1}{2}\right) = 1$

$$Gain(Pressure\ 3pm) = 0.9852 - \left(\frac{3}{7} \times 0.9183 + \frac{4}{7} \times 1\right) = 0.02$$

6. For Pressure 9am

Pressure 9 am	Cloud Tomorrow Yes	No	
1	2	1	$-\frac{2}{3} \log_2\left(\frac{2}{3}\right) - \frac{1}{3} \times \log_2\left(\frac{1}{3}\right) = 0.9183$
0	1	3	$-\frac{1}{4} \times \log_2\left(\frac{1}{4}\right) - \frac{3}{4} \times \log_2\left(\frac{3}{4}\right) = 0.8113$

$$Gain(Pressure\ 9am) = 0.9852 - \left(\frac{3}{7} \times 0.9183 + \frac{4}{7} \times 0.8113\right) = 0.128$$

4. For Humidity 3pm

	Cloud Tomorrow		
Humidity 3pm	Yes	No.	
1	1	0	$-1\times\log_2(1) - 0 = 0$
0	2	4.	$-\frac{2}{6}\times\log_2\left(\frac{2}{6}\right) - \frac{4}{6}\times\log_2\left(\frac{4}{6}\right) = 0.9813$

$$\text{Gain}(\text{Humidity 3pm}) = 0.9852 - \left(\frac{1}{7}\times 0 + \frac{6}{7}\times 0.9813\right) = 0.144$$

5. For Pressure 3pm

	Cloud Tomorrow		
Pressure 3pm	Yes	No	
1	1	2	$-\frac{1}{3}\times\log_2\left(\frac{1}{3}\right) - \frac{2}{3}\times\log_2\left(\frac{2}{3}\right) = 0.9183$
0	2	2	$-\frac{2}{4}\times\log_2\left(\frac{1}{2}\right) - \frac{1}{2}\times\log_2\left(\frac{1}{2}\right) = 1$

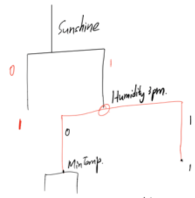
$$\text{Gain}(\text{Pressure 3pm}) = 0.9852 - \left(\frac{3}{7}\times 0.9183 + \frac{4}{7}\times 1\right) = 0.02.$$

6. For Pressure 9am

	Cloud Tomorrow		
Pressure 9 am	Yes	No	
1	2	1	$-\frac{2}{3}\log_2\left(\frac{2}{3}\right) - \frac{1}{3}\times\log_2\left(\frac{1}{3}\right) = 0.9183$
0	1	3	$-\frac{1}{4}\times\log_2\left(\frac{1}{4}\right) - \frac{3}{4}\times\log_2\left(\frac{3}{4}\right) = 0.8113$

$$\text{Gain}(\text{Pressure 9am}) = 0.9852 - \left(\frac{3}{7}\times 0.9183 + \frac{4}{7}\times 0.8113\right) = 0.128$$

As a result, sunshine's gain is the highest, then sunshin is selected as branch.



Then when Sunshine =1, check on Humidity 3pm.

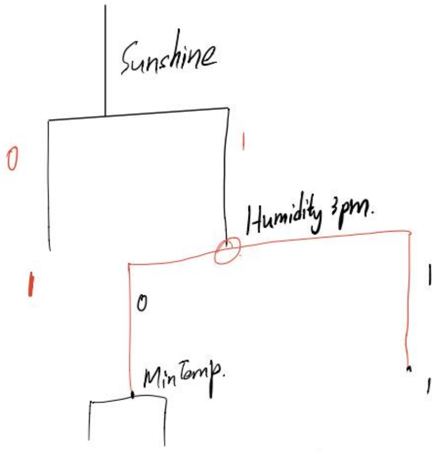
$E = 0.9182$

	Cloud Tomorrow		
Humidity 3pm	Yes	No	
1	1	0	$E = 0$
0	1	4.	$-\frac{1}{5}\times\log_2\left(\frac{1}{5}\right) - \frac{4}{5}\log_2\left(\frac{4}{5}\right) = 0.7219.$

$$\text{Gain} = 0.9182 - \left(0 + \frac{5}{6}\times 0.7219\right) = 0.3166.$$

Then do similiar thing on MinTemp.

As a result, sunshine's gain is the highest, then sunshin is selected as branch.



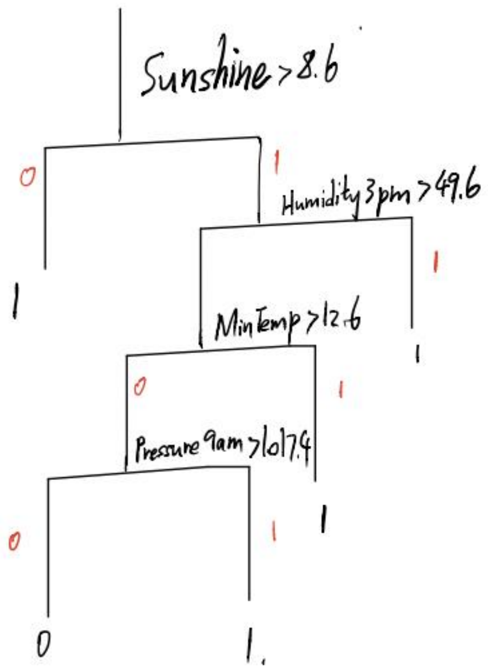
Then when Sunshine =1, check on Humidity 3pm.

$E = 0.9182$

Humidity 3pm	Cloud Tomorrow Yes	No	
1	1	0	$E = 0$
0	1	4	$-\frac{1}{5} \times \log_2\left(\frac{1}{5}\right) - \frac{4}{5}\log_2\left(\frac{4}{5}\right) = 0.7219$

$$Gain = 0.9182 - \left(0 + \frac{5}{6} \times 0.7219\right) = 0.3166$$

Then do similiar thing on MinTemp.



F. R code

```
## FIT3152 Assignment 2
## Last Modified date: 21th May 2021
## Student ID: 28280016

# Load necessary libraries
library(tree)
library(e1071)
library(ROCR)
library(randomForest)
library(rpart)
library(adabag)
library(ggplot2)
#library(neuralnet)
library(dplyr)
library(pROC)

# Set up data root path
setwd("~/Desktop/FIT3152/Assignment 2")

rm(list = ls())
WAUS <- read.csv("CloudPredict2021.csv")
# nrow(WAUS)
```

```
# Q.1 Data exploration
str(WAUS)
summary(WAUS)
cloudy_freq_table <- as.data.frame(xtabs(~CloudTomorrow,WAUS))
cbind(cloudy_freq_table, Percent = prop.table(cloudy_freq_table$Freq)*100)

# Check if there are any missing and NAs value
any(is.na(WAUS))     # The answer is Yes

count_NAs <- apply(WAUS, MARGIN = 2, function(col) sum(is.na(col)))

# Remove the NAs rows first
data <- WAUS[complete.cases(WAUS),] # Remove missing data for mean,std
description
min(data$Year)
max(data$Year)

# Calculate the std for each column
std_col <- apply(data[,c(5:9,11,14:21)],2,sd)
mean_col <- apply(data[,c(5:9,11,14:21)],2,mean)
summary(data)

## Normalize the data / Regularisation
normalisation = function(col){
  max_value = max(col)
  min_value = min(col)
  result = (col - min_value)/(max_value - min_value)
  return (result)
}


# Remove few columns
WAUS[,c("WindGustDir","WindDir9am","WindDir3pm")] = NULL

# Convert categorical data columns into numeric data type
WAUS$RainToday <- as.factor(WAUS$RainToday)

# observe the effect of independent variables (inputs) on the dependent
variable (output)

## Pre - processing deal with NAs/missing data.
WAUS <- WAUS[complete.cases(WAUS),] # Remove missing data from the
beginning

# Q 2 Document pre-processing required parts
# Random places
L <- as.data.frame(c(1:49))
set.seed(28280016) # My student ID:28280016
L <- L[sample(nrow(L), 10, replace = FALSE),] # sample 10 locations
WAUS <- WAUS[(WAUS$Location %in% L),]
WAUS <- WAUS[sample(nrow(WAUS), 2000, replace = FALSE),] # sample 2000 rows


## Q.3 Partition on to training and test data
set.seed(28280016) #Student ID as random seed
train.row = sample(1:nrow(WAUS), 0.7*nrow(WAUS))
WAUS_training = WAUS[train.row,]
WAUS_testing = WAUS[-train.row,]

nrow(WAUS_training)
nrow(WAUS_testing)
```

```r
# Extract Day, Month, Year of the observation
WAUS_training <- WAUS_training[,c(4:20)]
WAUS_testing <- WAUS_testing[,c(4:20)]

# Resolve the argument of length 0 error
WAUS_training$CloudTomorrow <- as.factor(WAUS_training$CloudTomorrow)
WAUS_testing$CloudTomorrow <- as.factor(WAUS_testing$CloudTomorrow)

## Q.4 Implement a classification model via different techniques
## Decision Tree
WAUS.tree <- tree(CloudTomorrow~.,data = WAUS_training)
summary(WAUS.tree)
# Running some plot
plot(WAUS.tree)
text(WAUS.tree,pretty = 0)

## Naïve Bayes
WAUS.bayes <- naiveBayes(CloudTomorrow~.,data = WAUS_training)
summary(WAUS.bayes)

## Bagging
WAUS.bag <- bagging(CloudTomorrow~.,data = WAUS_training,mfinal = 5)
summary(WAUS.bag)

## Boosting
WAUS.Boost <- boosting(CloudTomorrow~., data = WAUS_training, mfinal = 5)
summary(WAUS.Boost)

## Random Forests
WAUS.rf <- randomForest(CloudTomorrow~., data = WAUS_training)
summary(WAUS.rf)


# Q.5 using test data, classify each of the test cases.
# Q.6.Using test data, calculate confidence of predicting 'cloudy tomorrow'
# Construct ROC for each classifier.
# to compute accuracy
# Step 1: Create confusion matrix for each method
# Step 2 : (TP + TN)/(TP+TN+FP+FN) = Accuracy

# Decision Tree testing
WAUS.predtree <- predict(WAUS.tree,WAUS_testing,type = "class")
tree_table <- table(Predicted_Class = WAUS.predtree, Actual_Class =
WAUS_testing$CloudTomorrow)
cat("\n#Decision Tree Confusion Matrix\n")
tree_table
tree_accuracy <- (sum(diag(tree_table))/sum(tree_table))*100

detach(package:neuralnet)   # To solve some crash errors.

# Do predicitions as classes and draw a table
WAUS.pred.tree <- predict(WAUS.tree,WAUS_testing,type = "vector")
# Computing a simple ROC curve
# labels are actual values, predictors are probability of class
WAUSpred_tree <- prediction(WAUS.pred.tree[,2],WAUS_testing$CloudTomorrow)
WAUSperf_tree <- performance(WAUSpred_tree,"tpr","fpr")

# calculate auc for trees
auc_tree <- performance(WAUSpred_tree,"auc")
print(as.numeric(auc_tree@y.values))
```

```
plot(WAUSperf_tree,col="purple")
abline(0,1)

## Naïve Bayes testing
WAUS.predbayes = predict(WAUS.bayes,WAUS_testing)
naiveBayes_table = table(Predicted_Class = WAUS.predbayes, Actual_Class =
WAUS_testing$CloudTomorrow)
cat("\n#Naïve Bayes Confusion Matrix\n")
naiveBayes_table
naiveBayes_accuracy <-
(sum(diag(naiveBayes_table))/sum(naiveBayes_table))*100

# Outputs as confidence levels
WAUSpred.bayes <- predict(WAUS.bayes,WAUS_testing,type = "raw")
WAUSpred <- prediction(WAUSpred.bayes[,2], WAUS_testing$CloudTomorrow)
WAUSperf_naive <- performance(WAUSpred,"tpr","fpr")
plot(WAUSperf_naive,add = TRUE,col = 'blueviolet')

## Bagging testing
WAUSpred.bag <- predict.bagging(WAUS.bag,WAUS_testing)
WAUSpred.bag$confusion
bagging_accuracy <-
sum((diag(WAUSpred.bag$confusion)))/sum(WAUSpred.bag$confusion)*100

WAUSBagpred <- prediction(WAUSpred.bag$prob[,2],WAUS_testing$CloudTomorrow)
WAUSBagperf <- performance(WAUSBagpred,"tpr","fpr")

# calculate auc for Bagging
auc_bagging <- performance(WAUSBagpred,"auc")
print(as.numeric(auc_bagging@y.values))
plot(WAUSBagperf,add=TRUE,col='blue')
cat("\n#Bagging Confusion\n")
print(WAUSpred.bag$confusion)

## Boosting testing
WAUSpred.boost <- predict.boosting(WAUS.Boost,newdata = WAUS_testing)
WAUSpred.boost$confusion
boosting_accuracy <-
sum((diag(WAUSpred.boost$confusion)))/sum(WAUSpred.boost$confusion)*100
WAUSBoostpred <-
prediction(WAUSpred.boost$prob[,2],WAUS_testing$CloudTomorrow)

auc_boosting <- performance(WAUSBoostpred,"auc")
print(as.numeric(auc_boosting@y.values))

WAUSBoostperf <- performance(WAUSBoostpred,"tpr","fpr")
plot(WAUSBoostperf,add = TRUE,col = 'red')
cat("\n#Boosting Confusion\n")
print(WAUSpred.boost$confusion)

## Random Forest
WAUSpredrf <- predict(WAUS.rf, WAUS_testing)
WAUSRandF <- table(Predicted_Class = WAUSpredrf, Acutual_Class =
WAUS_testing$CloudTomorrow) # Confusion Matrix
cat("\n#Random Forest Confusion Matrix\n")
print(WAUSRandF)
randomForest_accuracy <- sum(diag(WAUSRandF))/sum(WAUSRandF)*100

WAUSpred.rf <- predict(WAUS.rf,WAUS_testing,type = "prob")
```

```
#WAUSpred.rf
WAUSRFpred <- prediction(WAUSpred.rf[,2],WAUS_testing$CloudTomorrow)

auc_randomForest <- performance(WAUSBagpred,"auc")
print(as.numeric(auc_randomForest@y.values))
WAUSRFperf <- performance(WAUSRFpred,"tpr","fpr")
plot(WAUSRFperf,add = TRUE,col = "darkgreen")

legend("topleft",legend=c("Decision Trees","Naïve Bayes","Bagging",
"Boosting","Random Forest"),
       col=c("purple","blueviolet","blue", "red","darkgreen"), lty=1:2,
cex=0.8)

# Add a straight line onto diagram
abline(0,1)

# Q.7 Comparisons between task 5 and task 6 for all classifiers.
# Combine accuracy results from previous tasks
accuracy_list <-
c(tree_accuracy,naiveBayes_accuracy,bagging_accuracy,boosting_accuracy,rand
omForest_accuracy)
summary_models <- rbind(Accuray = accuracy_list)
colnames(summary_models) <- c("Decision Trees","Naïve Bayes","Bagging",
"Boosting","Random Forest")
summary_models


# Q.8. Examine each models, determine most important variables for
prediction
summary(WAUS.tree)
plot(WAUS.tree)
text(WAUS.tree,pretty = 0)

sort(WAUS.bag$importance,decreasing=TRUE)

sort(WAUS.Boost$importance,decreasing =TRUE)

#sort(WAUS.rf$importance,decreasing = TRUE)
WAUS.rf$importance[order(-WAUS.rf$importance),]

## Based on these check, WindSpeed9am,
RainToday,WindGustSpeed,WindSpeed3pm,Temp3pm are top 5 least important
attributes
# remove these attributes from data-set
#WAUS_training[,c("RainToday","WindSpeed9am","WindGustSpeed","Temp3pm","Win
dSpeed3pm")] = NULL

#WAUS_testing[,c("RainToday","WindSpeed9am","WindGustSpeed","Temp3pm","Wind
Speed3pm")] <- NULL

# Q.9.Create a simpler classifier
set.seed(28280016)
WAUS_simple <- WAUS[sample(nrow(WAUS), 10, replace = FALSE),] # sample 10
rows
simpler_data <- WAUS_simple[,c(5,9,15,16,14,20)]

train_simple.row = sample(1:nrow(simpler_data), 0.7*nrow(simpler_data))
WAUS_simple_training = simpler_data[train_simple.row,]
WAUS_simple_testing = simpler_data[-train_simple.row,]
```

```
# Q.10 improve the performance
# K-fold cross validation on decision tree
WAUS_new.tree <- cv.tree(WAUS.tree,FUN = prune.misclass)
# Punning trees
WAUS_prunned.tree <- prune.misclass(WAUS.tree,best = 2)
summary(WAUS_prunned.tree)
plot(WAUS_prunned.tree)
text(WAUS_prunned.tree,pretty = 0)

# Conduct prediction
WAUS_prunned.predtree <- predict(WAUS_prunned.tree,WAUS_testing,type =
"class")
treeprunned_table <- table(Predicted_Class = WAUS_prunned.predtree,
Actual_Class = WAUS_testing$CloudTomorrow)
cat("\n#Decision Tree after Prunning Confusion Matrix\n")
treeprunned_table
treeprunned_accuracy <-
(sum(diag(treeprunned_table))/sum(treeprunned_table))*100


# Q. 10  Create the best tree-based classifier
# Based on accuracy table, Boosting and Random Forest are chosen as the
highest accuracy.
set.seed(28280016)
data_best_training <- WAUS_training[,c(6,13,17,11,2,12)]
data_best_testing <- WAUS_testing[,c(6,13,17,11,2,12)]

WAUS_best.Boost <- boosting(CloudTomorrow~., data = data_best_training,
mfinal = 10)
# Run the boosting model
## Boosting testing
WAUSpred_best.boost <- predict.boosting(WAUS_best.Boost,newdata =
data_best_testing)
WAUSpred_best.boost$confusion
boosting_best_accuracy <-
sum((diag(WAUSpred_best.boost$confusion)))/sum(WAUSpred_best.boost$confusio
n)*100
WAUSBoostpred_best <-
prediction(WAUSpred_best.boost$prob[,2],data_best_testing$CloudTomorrow)
auc_boosting_best <- performance(WAUSBoostpred_best,"auc")
print(as.numeric(auc_boosting_best@y.values))
WAUSBoostperf_best <- performance(WAUSBoostpred_best,"tpr","fpr")

# Run cross validation for boosting
WAUS.boostingcv <- boosting.cv(CloudTomorrow~.,data =
data_best_training,boos = TRUE,mfinal = 100,coeflearn =
"Breiman",control=rpart.control(cp=0.01))
WAUS.boostingcv$confusion
WAUS.boostingcv$error


# Q.11 Build ANN
WAUS_new <- WAUS
WAUS_new$CloudTomorrow <- as.factor(WAUS_new$CloudTomorrow)

# Select some important variables manually
WAUS_new <- WAUS_new[,c(5,8,9,13,14,15,16,17,20)]

# Normalise data
WAUS_new_norm <- as.data.frame(lapply(WAUS_new[1:7], normalisation))
WAUS_new_norm <-cbind(WAUS_new_norm,WAUS_new$CloudTomorrow)
```

```r
colnames(WAUS_new_norm)[8] <-"CloudTomorrow"

# Split data into train and test
set.seed(28280016)
train.row <-sample(1:nrow(WAUS_new_norm),0.7*nrow(WAUS_new_norm))
WAUS_new.train <- WAUS_new_norm[train.row,]
WAUS_new.test <- WAUS_new_norm[-train.row,]

# Build up an ANN
library(neuralnet)
WAUS.nn <- neuralnet(CloudTomorrow~.,WAUS_new.train,hidden = 1)
# Visualize Artificial Neural Network
plot(WAUS.nn)
WAUS.nn$result.matrix
WAUS.nn$net.result

WAUS_nn.predict <- compute(WAUS.nn,WAUS_new.test)
WAUS_nn.predr <- round(WAUS_nn.predict$net.result,0)
WAUS_nn.predrdf <- as.data.frame(as.table(WAUS_nn.predr))
WAUS_nn.predrdf <- WAUS_nn.predrdf[!WAUS_nn.predrdf$Freq == 0,]
WAUS_nn.predrdf$Freq = NULL
colnames(WAUS_nn.predrdf) <- c("Obs","CloudTomorrow")
WAUS_nn.predrdf <- WAUS_nn.predrdf[order(WAUS_nn.predrdf$Obs),]

# Create the confusion matrix and calculate the accuracy
WAUS_nn_cm <- table(observed = WAUS_new.test$CloudTomorrow,predicted =
WAUS_nn.predrdf$CloudTomorrow)
WAUS_nn_cm
WAUS_nn_accuracy <- round(sum(diag(WAUS_nn_cm))/sum(WAUS_nn_cm)*100,3)
WAUS_nn_accuracy
```