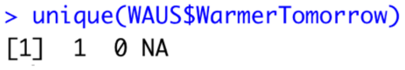
Introduction

Assignment 2 asked to predict whether tomorrow will be warmer than today for 10 locations in Australia by implementing classification model using these techniques, such as: Decision Tree, Naïve Bayes, Bagging, Boosting, Random Forest and Artificial Neural Network. Moreover, from the insights that is gotten from performance. 2000 sample of data rows from WarmerTomorrow2022.csv file is being used as the source of the data. For every question, all data are being pre-processed based on their needs. It has been explained in detail throughout the questions in this report. In addition, all calculations provided in this report are calculated in R Script.

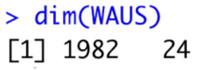
Question 1 – Proportion of Warmer Days and Observation of Predictor Variables

Before finding the proportion of days when it is warmer than previous day compared to those where it is cooler, find the unique values of the WarmerTomorrow column first. Thus, it is shown that there are 3 unique values, that are: 1, 0 and NA.

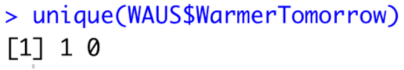


```
> unique(WAUS$WarmerTomorrow)
[1]  1  0 NA
```

Therefore, delete all the rows that has null values of WarmerTomorrow. As a result, 12 rows have been omitted from observed dataset. After that, the unique values of WAUS$WarmerTomorrow will be 1 and 0 only. 1 represents tomorrow will be warmer than today and 0 vice versa.

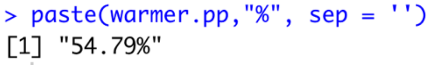


```
> dim(WAUS)
[1] 1982   24
```



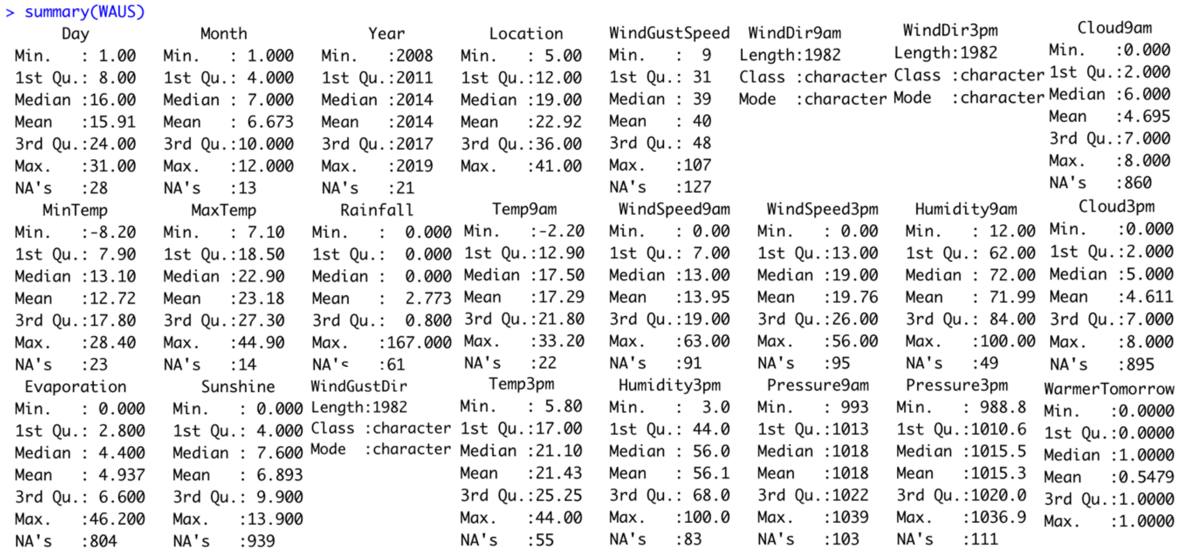
```
> unique(WAUS$WarmerTomorrow)
[1] 1 0
```

Through calculation, the proportion of days when it is warmer than previous day compared to those where it is cooler is 54.79%.

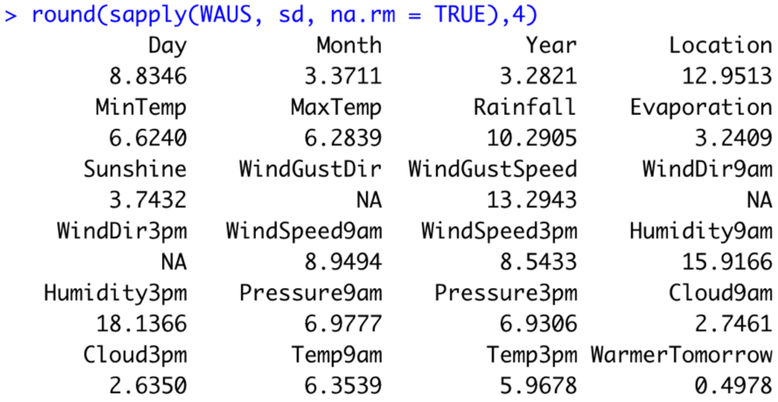


```
> paste(warmer.pp,"%", sep = '')
[1] "54.79%"
```

Based on the observation of each column's minimum, first quartile, median, mean, third quartile, maximum, number of null values, and standard deviation values.



```
> summary(WAUS)
      Day            Month             Year         Location       WindGustSpeed  WindDir9am         WindDir3pm           Cloud9am    
 Min.   : 1.00   Min.   : 1.000   Min.   :2008   Min.   : 5.00   Min.   :  9   Length:1982        Length:1982        Min.   :0.000  
 1st Qu.: 8.00   1st Qu.: 4.000   1st Qu.:2011   1st Qu.:12.00   1st Qu.: 31   Class :character   Class :character   1st Qu.:2.000  
 Median :16.00   Median : 7.000   Median :2014   Median :19.00   Median : 39   Mode  :character   Mode  :character   Median :6.000  
 Mean   :15.91   Mean   : 6.673   Mean   :2014   Mean   :22.92   Mean   : 40                                         Mean   :4.695  
 3rd Qu.:24.00   3rd Qu.:10.000   3rd Qu.:2017   3rd Qu.:36.00   3rd Qu.: 48                                         3rd Qu.:7.000  
 Max.   :31.00   Max.   :12.000   Max.   :2019   Max.   :41.00   Max.   :107                                         Max.   :8.000  
 NA's   :28      NA's   :13       NA's   :21                     NA's   :127                                         NA's   :860    
    MinTemp          MaxTemp          Rainfall          Temp9am       WindSpeed9am    WindSpeed3pm    Humidity9am        Cloud3pm    
 Min.   :-8.20   Min.   : 7.10   Min.   :  0.000   Min.   :-2.20   Min.   : 0.00   Min.   : 0.00   Min.   : 12.00   Min.   :0.000  
 1st Qu.: 7.90   1st Qu.:18.50   1st Qu.:  0.000   1st Qu.:12.90   1st Qu.: 7.00   1st Qu.:13.00   1st Qu.: 62.00   1st Qu.:2.000  
 Median :13.10   Median :22.90   Median :  0.000   Median :17.50   Median :13.00   Median :19.00   Median : 72.00   Median :5.000  
 Mean   :12.72   Mean   :23.18   Mean   :  2.773   Mean   :17.29   Mean   :13.95   Mean   :19.76   Mean   : 71.99   Mean   :4.611  
 3rd Qu.:17.80   3rd Qu.:27.30   3rd Qu.:  0.800   3rd Qu.:21.80   3rd Qu.:19.00   3rd Qu.:26.00   3rd Qu.: 84.00   3rd Qu.:7.000  
 Max.   :28.40   Max.   :44.90   Max.   :167.000   Max.   :33.20   Max.   :63.00   Max.   :56.00   Max.   :100.00   Max.   :8.000  
 NA's   :23      NA's   :14      NA's   :61        NA's   :22      NA's   :91      NA's   :95      NA's   :49       NA's   :895    
  Evaporation        Sunshine      WindGustDir           Temp3pm       Humidity3pm     Pressure9am     Pressure3pm     WarmerTomorrow  
 Min.   : 0.000   Min.   : 0.000   Length:1982        Min.   : 5.80   Min.   :  3.0   Min.   : 993   Min.   : 988.8   Min.   :0.0000  
 1st Qu.: 2.800   1st Qu.: 4.000   Class :character   1st Qu.:17.00   1st Qu.: 44.0   1st Qu.:1013   1st Qu.:1010.6   1st Qu.:0.0000  
 Median : 4.400   Median : 7.600   Mode  :character   Median :21.10   Median : 56.0   Median :1018   Median :1015.5   Median :1.0000  
 Mean   : 4.937   Mean   : 6.893                      Mean   :21.43   Mean   : 56.1   Mean   :1018   Mean   :1015.3   Mean   :0.5479  
 3rd Qu.: 6.600   3rd Qu.: 9.900                      3rd Qu.:25.25   3rd Qu.: 68.0   3rd Qu.:1022   3rd Qu.:1020.0   3rd Qu.:1.0000  
 Max.   :46.200   Max.   :13.900                      Max.   :44.00   Max.   :100.0   Max.   :1039   Max.   :1036.9   Max.   :1.0000  
 NA's   :804      NA's   :939                         NA's   :55      NA's   :83      NA's   :103    NA's   :111                      
```

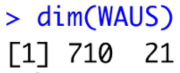


```
> round(sapply(WAUS, sd, na.rm = TRUE),4)
          Day         Month          Year      Location 
       8.8346        3.3711        3.2821       12.9513 
      MinTemp       MaxTemp      Rainfall   Evaporation 
       6.6240        6.2839       10.2905        3.2409 
     Sunshine   WindGustDir WindGustSpeed    WindDir9am 
       3.7432            NA       13.2943            NA 
   WindDir3pm  WindSpeed9am  WindSpeed3pm   Humidity9am 
           NA        8.9494        8.5433       15.9166 
  Humidity3pm   Pressure9am   Pressure3pm      Cloud9am 
      18.1366        6.9777        6.9306        2.7461 
     Cloud3pm       Temp9am       Temp3pm WarmerTomorrow 
       2.6350        6.3539        5.9678        0.4978 
```

Day, Month, and Year columns are not worthy to the WarmerTomorrow prediction as they only give information about the date and will not affect anything to the target variable. Thus, they will be omitted from the analysis. Although, the proportion number of null values of Cloud9am, Cloud3pm, Evaporation and Sunshine columns are almost half of the observed WAUS data. The columns are not chosen to be dropped as they are assumed to have effect on the prediction of WarmerTomorrow column. In addition, all the columns besides the all the Date related columns will be used as the data in WAUS variable for further analysis.

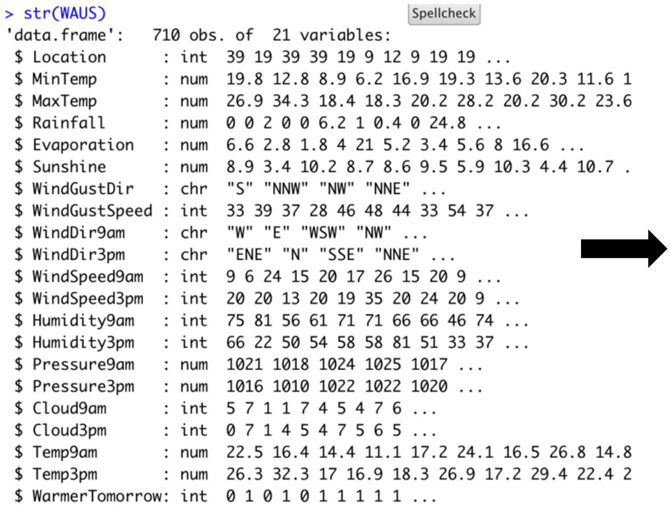
Question 2 – Pre-Processing Data

From the analysis in Question 1, Day, Month, Year columns needed to be dropped as they are not relevant to the analysis. Furthermore, complete.cases(WAUS) function was used to drop all the rows that contain null values. Hence, the number of rows present in WAUS variable is now 710 rows with 21 columns.

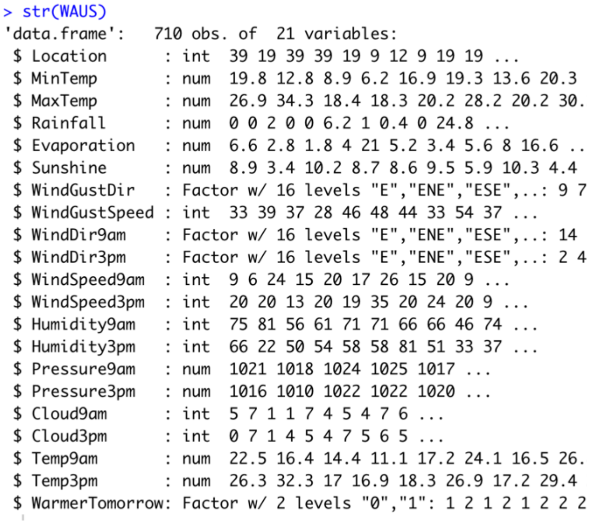


```
> dim(WAUS)
[1] 710  21
```

Moreover, the variable types of WindGustDir, WindDir9am, and WindDir3pm column are in character. Thus, they need to be changed to a factor. It goes the same as the target variable which is WarmerTomorrow that will also be changed into a factor as the techniques use to implement classification model require the character types to be converted into a factor as well as its target variable.

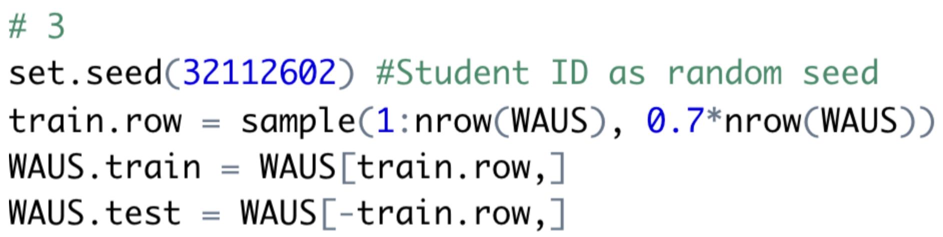


```
> str(WAUS)
'data.frame':	710 obs. of  21 variables:
 $ Location      : int  39 19 39 39 19 9 12 9 19 19 ...
 $ MinTemp       : num  19.8 12.8 8.9 6.2 16.9 19.3 13.6 20.3 11.6 1
 $ MaxTemp       : num  26.9 34.3 18.4 18.3 20.2 28.2 20.2 30.2 23.6
 $ Rainfall      : num  0 0 2 0 0 6.2 1 0.4 0 24.8 ...
 $ Evaporation   : num  6.6 2.8 1.8 4 21 5.2 3.4 5.6 8 16.6 ...
 $ Sunshine      : num  8.9 3.4 10.2 8.7 8.6 9.5 5.9 10.3 4.4 10.7 .
 $ WindGustDir   : chr  "S" "NNW" "NW" "NNE" ...
 $ WindGustSpeed : int  33 39 37 28 46 48 44 33 54 37 ...
 $ WindDir9am    : chr  "W" "E" "WSW" "NW" ...
 $ WindDir3pm    : chr  "ENE" "N" "SSE" "NNE" ...
 $ WindSpeed9am  : int  9 6 24 15 20 17 26 15 20 9 ...
 $ WindSpeed3pm  : int  20 20 13 20 19 35 20 24 20 9 ...
 $ Humidity9am   : int  75 81 56 61 71 71 66 66 46 74 ...
 $ Humidity3pm   : int  66 22 50 54 58 58 81 51 33 37 ...
 $ Pressure9am   : num  1021 1018 1024 1025 1017 ...
 $ Pressure3pm   : num  1016 1010 1022 1022 1020 ...
 $ Cloud9am      : int  5 7 1 1 7 4 5 4 7 6 ...
 $ Cloud3pm      : int  0 7 1 4 5 4 7 5 6 5 ...
 $ Temp9am       : num  22.5 16.4 14.4 11.1 17.2 24.1 16.5 26.8 14.8
 $ Temp3pm       : num  26.3 32.3 17 16.9 18.3 26.9 17.2 29.4 22.4 2
 $ WarmerTomorrow: int  0 1 0 1 0 1 1 1 1 1 ...
```



```
> str(WAUS)
'data.frame':	710 obs. of  21 variables:
 $ Location      : int  39 19 39 39 19 9 12 9 19 19 ...
 $ MinTemp       : num  19.8 12.8 8.9 6.2 16.9 19.3 13.6 20.3
 $ MaxTemp       : num  26.9 34.3 18.4 18.3 20.2 28.2 20.2 30.
 $ Rainfall      : num  0 0 2 0 0 6.2 1 0.4 0 24.8 ...
 $ Evaporation   : num  6.6 2.8 1.8 4 21 5.2 3.4 5.6 8 16.6 ..
 $ Sunshine      : num  8.9 3.4 10.2 8.7 8.6 9.5 5.9 10.3 4.4
 $ WindGustDir   : Factor w/ 16 levels "E","ENE","ESE",..: 9 7
 $ WindGustSpeed : int  33 39 37 28 46 48 44 33 54 37 ...
 $ WindDir9am    : Factor w/ 16 levels "E","ENE","ESE",..: 14
 $ WindDir3pm    : Factor w/ 16 levels "E","ENE","ESE",..: 2 4
 $ WindSpeed9am  : int  9 6 24 15 20 17 26 15 20 9 ...
 $ WindSpeed3pm  : int  20 20 13 20 19 35 20 24 20 9 ...
 $ Humidity9am   : int  75 81 56 61 71 71 66 66 46 74 ...
 $ Humidity3pm   : int  66 22 50 54 58 58 81 51 33 37 ...
 $ Pressure9am   : num  1021 1018 1024 1025 1017 ...
 $ Pressure3pm   : num  1016 1010 1022 1022 1020 ...
 $ Cloud9am      : int  5 7 1 1 7 4 5 4 7 6 ...
 $ Cloud3pm      : int  0 7 1 4 5 4 7 5 6 5 ...
 $ Temp9am       : num  22.5 16.4 14.4 11.1 17.2 24.1 16.5 26.
 $ Temp3pm       : num  26.3 32.3 17 16.9 18.3 26.9 17.2 29.4
 $ WarmerTomorrow: Factor w/ 2 levels "0","1": 1 2 1 2 1 2 2 2
```

Question 3 – Divide 70% Train Data, 30% Test Data



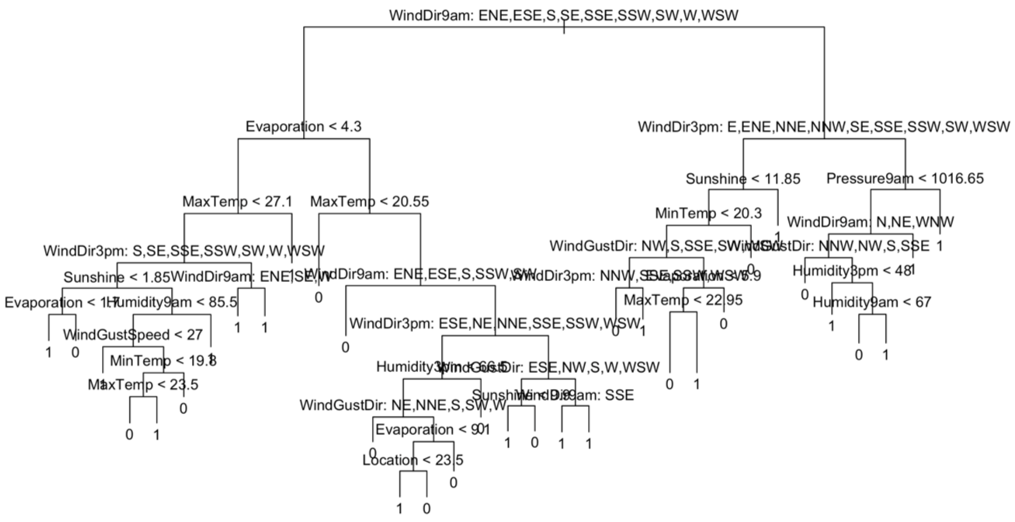
```
# 3
set.seed(32112602) #Student ID as random seed
train.row = sample(1:nrow(WAUS), 0.7*nrow(WAUS))
WAUS.train = WAUS[train.row,]
WAUS.test = WAUS[-train.row,]
```

For the analysis, divide 70% of WAUS data into train data and 30% into test data. The train data will be called as WAUS.train. Meanwhile, WAUS.test variable will be the test data.

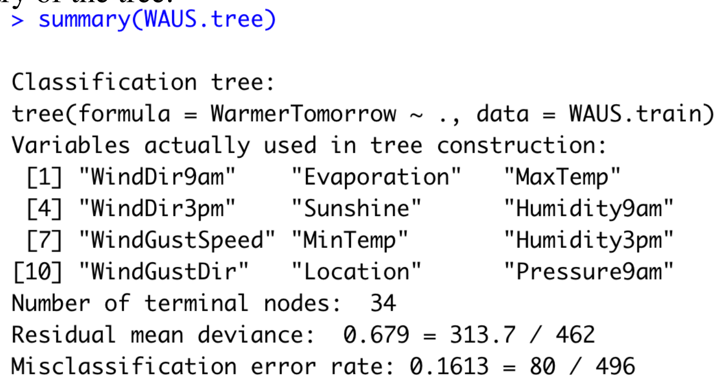
Question 4 – Implement Classification Model by Using Decision Tree, Naïve Bayes, Bagging, Boosting and Random Forest

Decision Tree

To construct the decision tree, all the predictors' variables are used by using WAUS.train as the data. Below is the decision tree constructed.



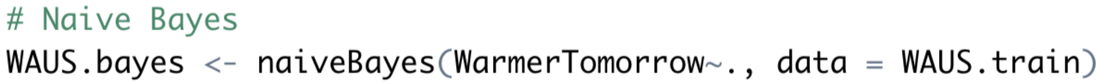
By looking at the summary of the tree:



```
> summary(WAUS.tree)

Classification tree:
tree(formula = WarmerTomorrow ~ ., data = WAUS.train)
Variables actually used in tree construction:
 [1] "WindDir9am"    "Evaporation"   "MaxTemp"
 [4] "WindDir3pm"    "Sunshine"      "Humidity9am"
 [7] "WindGustSpeed" "MinTemp"       "Humidity3pm"
[10] "WindGustDir"   "Location"      "Pressure9am"
Number of terminal nodes:  34
Residual mean deviance:  0.679 = 313.7 / 462
Misclassification error rate: 0.1613 = 80 / 496
```

It is observed that only Windir9am, Evaporation, MaxTemp, WindDir3pm, Sunshine, Humidity9am, WindGustSpeed, MinTemp, Humididty3pm, WindGustDir, Location and Pressure9am columns are used in the tree construction.

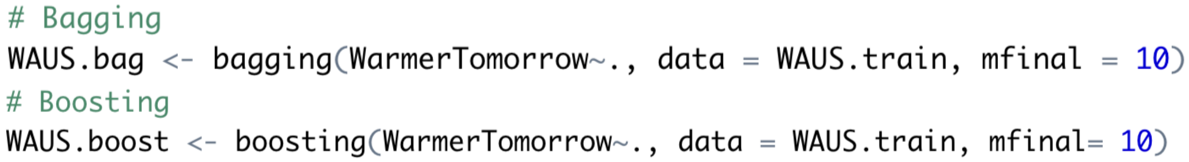
Naïve Bayes



```
# Naive Bayes
WAUS.bayes <- naiveBayes(WarmerTomorrow~., data = WAUS.train)
```

All the predictor variables are also used when implementing Naïve Baye by using WAUS.train as the data. WAUS.bayes is the variable to store the Naïve Bayes techniques that has been implemented

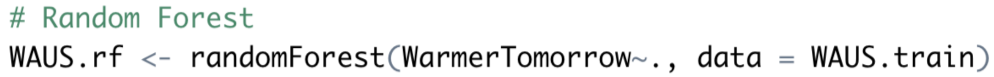
Bagging & Boosting



```
# Bagging
WAUS.bag <- bagging(WarmerTomorrow~., data = WAUS.train, mfinal = 10)
# Boosting
WAUS.boost <- boosting(WarmerTomorrow~., data = WAUS.train, mfinal= 10)
```

For bagging and boosting techniques, all the predictor variables are also used when implementing those techniques by using WAUS.train as the data. The number of mfinal is chosen to be 10 means the number of iterations for which bagging/boosting is run. As the dimension of WAUS data is 710 rows with 21 columns, small value of mfinal can be used which is 10. If the default of mfinal is used which is 100, the number of iterations will be too much and hence will affect the runtime of the analysis.

Random Forest

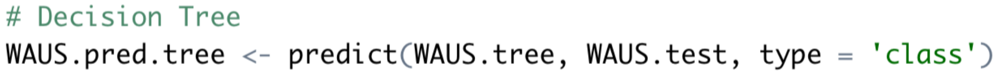


```
# Random Forest
WAUS.rf <- randomForest(WarmerTomorrow~., data = WAUS.train)
```

The last techniques which is random forest also used all the predictor variables by using WAUS.train as the data.

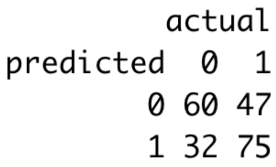
Question 5 – Classification, Confusion Matrix, Accuracy

Decision Tree



```
# Decision Tree
WAUS.pred.tree <- predict(WAUS.tree, WAUS.test, type = 'class')
```

After implementing decision tree techniques, get the prediction of WarmerTomorrow by using predict function with WAUS.tree and WAUS.test data with class type as the arguments. Hence, the values of WAUS.pred.tree has been obtained. From this step, confusion matrix can be created by using WAUS.pred.tree as the prediction data and WAUS.test$WarmerTomorrow as the actual data.

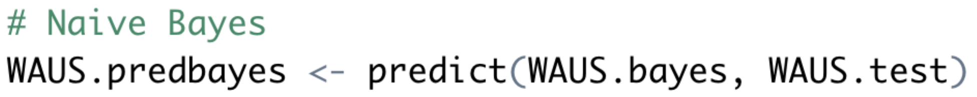


```
         actual
predicted  0  1
        0 60 47
        1 32 75
```

From this confusion matrix, the accuracy of the model can be calculated using the formula (True Positive + True Negative)/ (True Positive + True Negative + False Positive + False Negative). The value of True Negative is when the actual and predicted value are 0 (not warmer tomorrow), while True Positive is when the actual and predicted value are 1 (warmer tomorrow). In addition, False

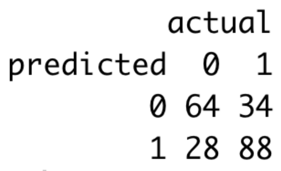
Positive is when the predicted value is 1 and the actual value is 0 and False Negative vice versa. Therefore, the accuracy for this model is (75+60)/(75+60+47+32) = 0.6495.

Naïve Bayes



```
# Naive Bayes
WAUS.predbayes <- predict(WAUS.bayes, WAUS.test)
```

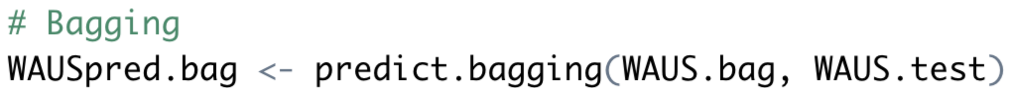
Furthermore, it is also needed to get the prediction of the Naïve Bayes model by using WAUS.bayes and WAUS.test as the arguments. Then, by using the WAUS.predbayes and WAUS.test$WarmerTomorrow create the confusion matrix as below.



```
         actual
predicted  0  1
        0 64 34
        1 28 88
```

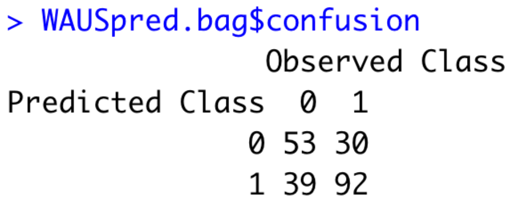
Using the same formula from the decision tree's accuracy, the accuracy of Naïve Bayes model is (88+64)/(88+64+34+28) = 0.7103.

Bagging



```
# Bagging
WAUSpred.bag <- predict.bagging(WAUS.bag, WAUS.test)
```

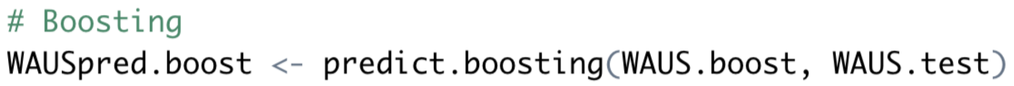
By using the predict.bagging function, the value of WAUSpred.bag is obtained. Next, create the confusion matrix by using the WAUSpred.bag and WAUS.test$WarmerTomorrow as the arguments.



```
> WAUSpred.bag$confusion
               Observed Class
Predicted Class  0  1
              0 53 30
              1 39 92
```

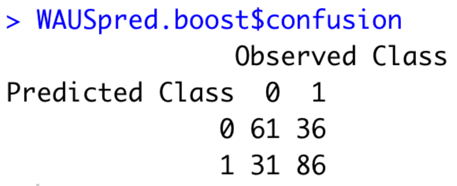
Therefore, the accuracy of Bagging model is (92+53)/(92+53+39+39) = 0.6776.

Boosting



```
# Boosting
WAUSpred.boost <- predict.boosting(WAUS.boost, WAUS.test)
```

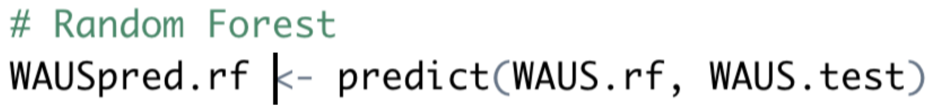
Morover, get the prediction by using WAUS.boost and WAUS.test as the arguments with predict.boosting function. After that, create the confusion matrix by using WAUSpred.boost and WAUS.test$WarmerTomorroe as the arguments.



```
> WAUSpred.boost$confusion
               Observed Class
Predicted Class  0  1
              0 61 36
              1 31 86
```

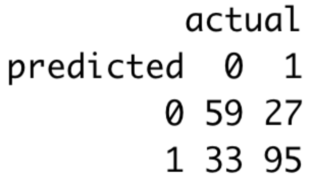
Thus, the accuracy of Boosting model is (86+61)/(86+61+36+31) = 0.6869.

Random Forest



```
# Random Forest
WAUSpred.rf <- predict(WAUS.rf, WAUS.test)
```

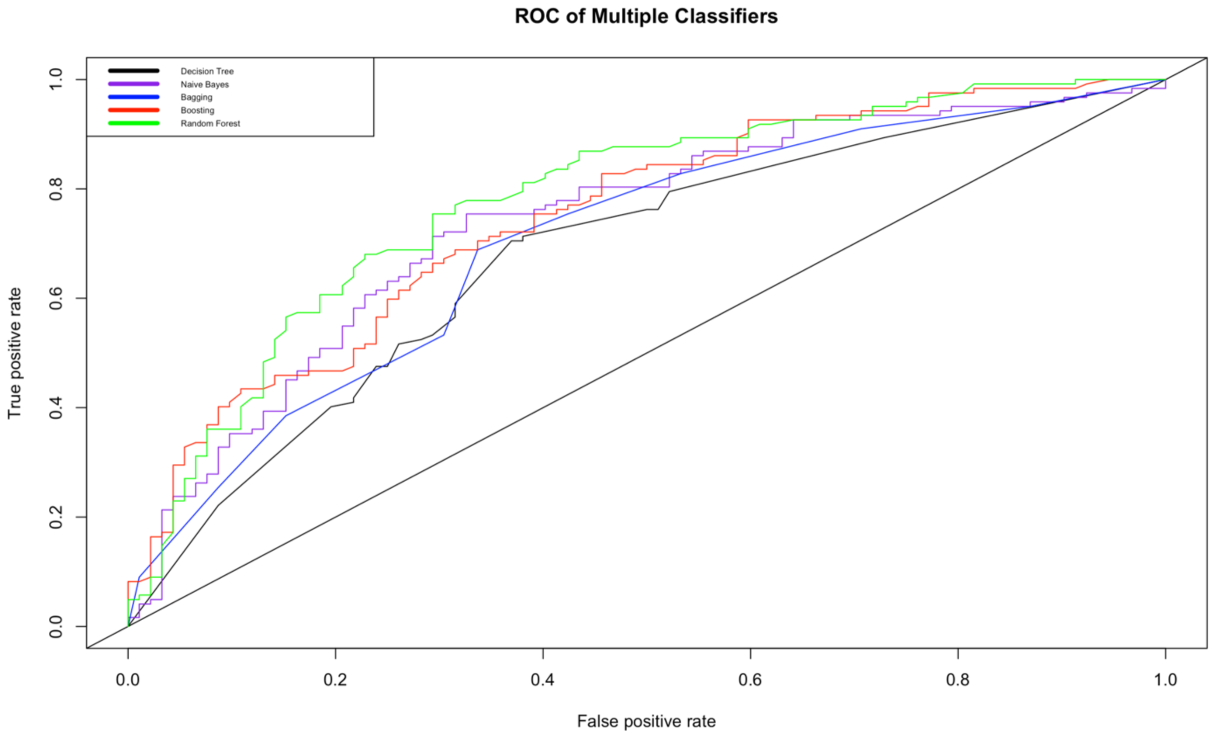
Predict the Random Forest model by using WAUS.rf and WAUS.test as the arguments, then store the values into WAUSpred.rf. Then, create the confusion matrix by using that variable and WAUS.test$WarmerTomorrow.



```
         actual
predicted  0  1
        0 59 27
        1 33 95
```

Hence, the accuracy of Random Forest model is (95+59)/(95+59+27+33) = 0.7196.

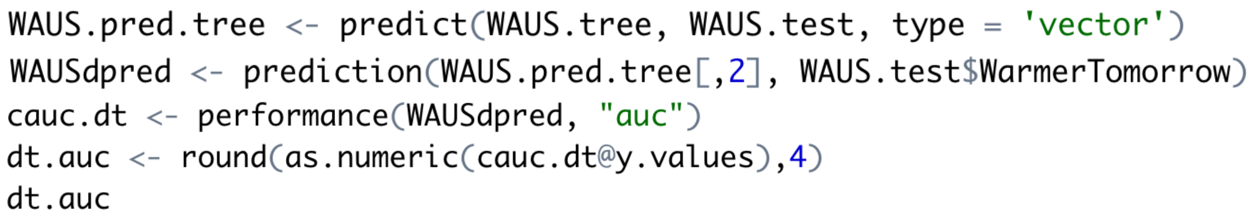
Question 6 – ROC and AUC



ROC curve is plotting the true positive rate against the false positive rate. TPR is the observation that were correctly predicted to be positive out of all positive observation. Meanwhile, FPR is the observation that are incorrectly predicted to be positive out of all negative observation. If ROC curve is closer to the (1,1) which is the top left corner of the plot, the better the model does at classifying the WAUS data to predict whether tomorrow will be warmer than today or not. The diagonal line of the ROC curve is the random guess that means if one of the ROC curves lies on or under this diagonal line that means the model is a bad classifier. Meanwhile, if the ROC curves are above the diagonal line, then it can be said as a better or good classifier.

Based on the plot, Random Forest is the best classifier out of the other models as its ROC curve is the closest to (1,1). On the other hand, Decision tree is the worst classifier as its ROC curve is the closest to the diagonal line. To quantify, let's calculate its Area Under Curve (AUC) which tells the area under the curve of each model. The closer AUC to 1, the better the model. In addition, the model with 0.5 AUC means a perfectly diagonal line and that means the model makes random classifications.

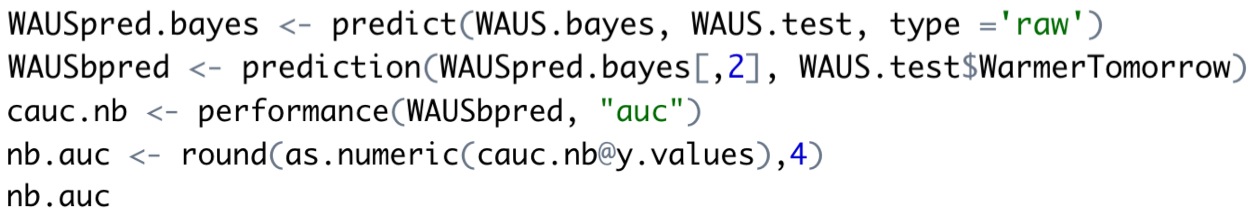
Decision Tree



```
WAUS.pred.tree <- predict(WAUS.tree, WAUS.test, type = 'vector')
WAUSdpred <- prediction(WAUS.pred.tree[,2], WAUS.test$WarmerTomorrow)
cauc.dt <- performance(WAUSdpred, "auc")
dt.auc <- round(as.numeric(cauc.dt@y.values),4)
dt.auc
```

After calculating the AUC by using WAUSdpred as the arguments, where WAUSdpred is the prediction by using the second column of WAUS.pred.tree and WAUS.test$WarmerTomorrow. It is obtained that the AUC of decision tree is 0.681

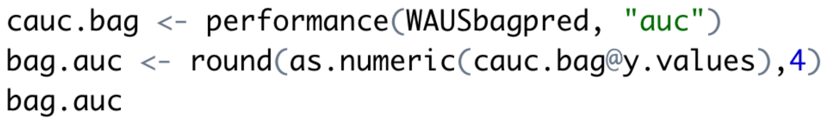
Naïve Bayes



```
WAUSpred.bayes <- predict(WAUS.bayes, WAUS.test, type ='raw')
WAUSbpred <- prediction(WAUSpred.bayes[,2], WAUS.test$WarmerTomorrow)
cauc.nb <- performance(WAUSbpred, "auc")
nb.auc <- round(as.numeric(cauc.nb@y.values),4)
nb.auc
```

Using the same techniques as decision tree, where WAUSbpred is the prediction by using the second column of WAUSpred.bayes and WAUS.test$WarmerTomorrow. Thus, the AUC of Naïve Bayes is 0.739

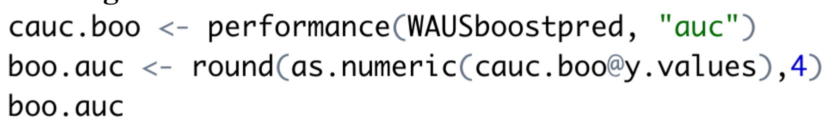
Bagging



```
cauc.bag <- performance(WAUSbagpred, "auc")
bag.auc <- round(as.numeric(cauc.bag@y.values),4)
bag.auc
```

Moreover, WAUSbagpred is the prediction by using WAUSpred.bag second column and WAUS.test$WarmerTomorrow. Therefore, the Bagging model AUC is 0.7029

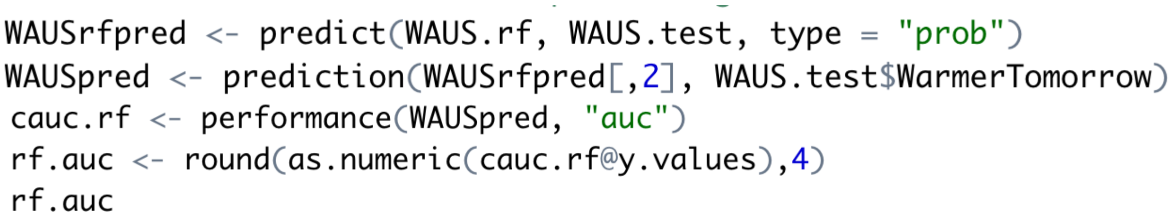
Boosting



```
cauc.boo <- performance(WAUSboostpred, "auc")
boo.auc <- round(as.numeric(cauc.boo@y.values),4)
boo.auc
```

Furthermore, WAUSboostpred is the prediction by using WAUSpred.boost second column and WAUS.test$WarmerTomorrow. Hence, the Boosting model AUC is 0.7491

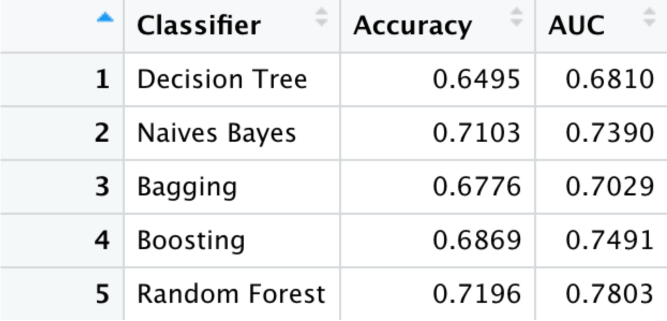
Random Forest



```
WAUSrfpred <- predict(WAUS.rf, WAUS.test, type = "prob")
WAUSpred <- prediction(WAUSrfpred[,2], WAUS.test$WarmerTomorrow)
cauc.rf <- performance(WAUSpred, "auc")
rf.auc <- round(as.numeric(cauc.rf@y.values),4)
rf.auc
```

For Random Forest, WAUSrfpred second column and WAUS.test$WarmerTomorrow are used to do the prediction. Thus, by using the performance function the calculated AUC of this model is 0.7803.

Question 7 – Comparison of Accuracy and AUC of Each Model



	Classifier	Accuracy	AUC
1	Decision Tree	0.6495	0.6810
2	Naives Bayes	0.7103	0.7390
3	Bagging	0.6776	0.7029
4	Boosting	0.6869	0.7491
5	Random Forest	0.7196	0.7803

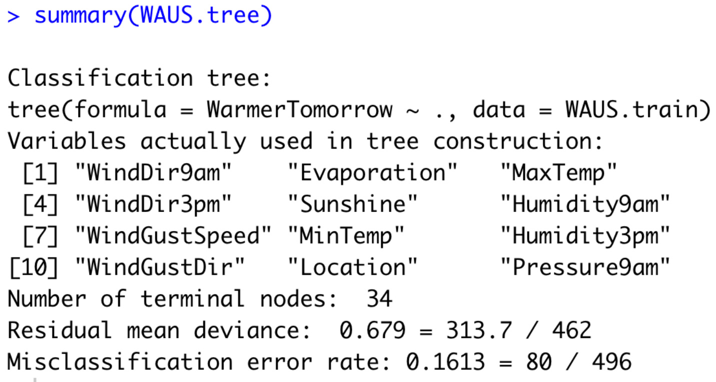
From the previous analysis Question 5 and 6, a table to compare the results of Accuracy and AUC for each model are created. If the accuracy is used to order which model is the best classifier, Random Forest is the best model out of all classifiers followed by Naïve Bayes and Boosting. In addition, Decision Tree model has the worst classifier since it has the lowest amount of accuracy.

Furthermore, if AUC is used as the measure the best classifier, Random Forest still wins as it has the highest number of AUC which is 0.7803, followed by Boosting and Naïve Bayes. Moreover, Decision Tree has the lowest AUC that means it is the worst classifier which is the same as the accuracy measurement.

By using both accuracy and AUC measurement, Random Forest has the best model out of all other models as its Accuracy and AUC are the highest than the others and Decision Tree has the worst model as it has the lowest amount of accuracy and AUC.

Question 8 – Variables Importance

Decision Tree



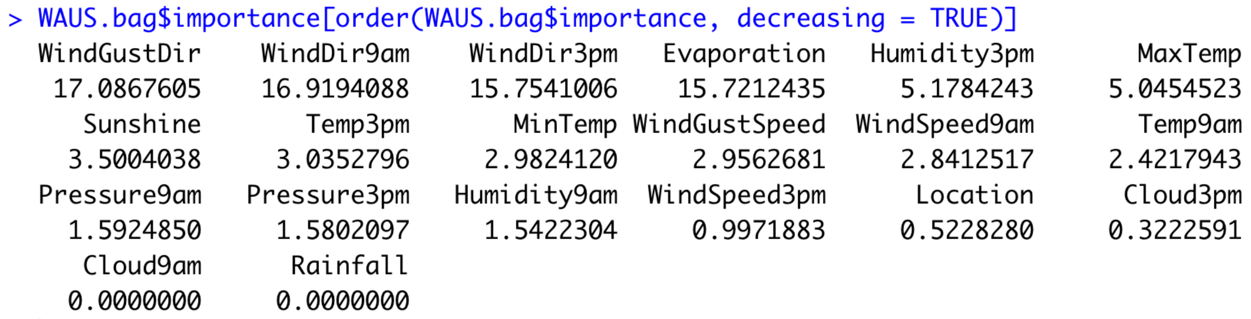
```
> summary(WAUS.tree)

Classification tree:
tree(formula = WarmerTomorrow ~ ., data = WAUS.train)
Variables actually used in tree construction:
 [1] "WindDir9am"    "Evaporation"   "MaxTemp"
 [4] "WindDir3pm"    "Sunshine"      "Humidity9am"
 [7] "WindGustSpeed" "MinTemp"       "Humidity3pm"
[10] "WindGustDir"   "Location"      "Pressure9am"
Number of terminal nodes:  34
Residual mean deviance:  0.679 = 313.7 / 462
Misclassification error rate: 0.1613 = 80 / 496
```

From the summary of WAUS.tree, the most important variables to predict whether it will be warmer tomorrow or not is WindDir9am column (Wind Direction at 9 am). In addition, the columns that are not present in the summary of WAUS.tree which are Rainfall, WindSpeed9am, WindSpeed3pm, Pressure3pm, Cloud9am, Cloud3pm, Temp9am and Temp3pm can be omitted as they are not used in the tree construction. If one or two of the variables in the tree construction are chosen to be omitted, Pressure9am and Location can be omitted which will result in little effect on the performance of the model as they appeared in the last two of the summary of the decision tree. The order of the variables used in the tree constructions represents the order of the most to the least important variables used in the tree construction.

Naïve Bayes

As discussed from the lecture, there are no variable importance measured for Naïve Bayes model

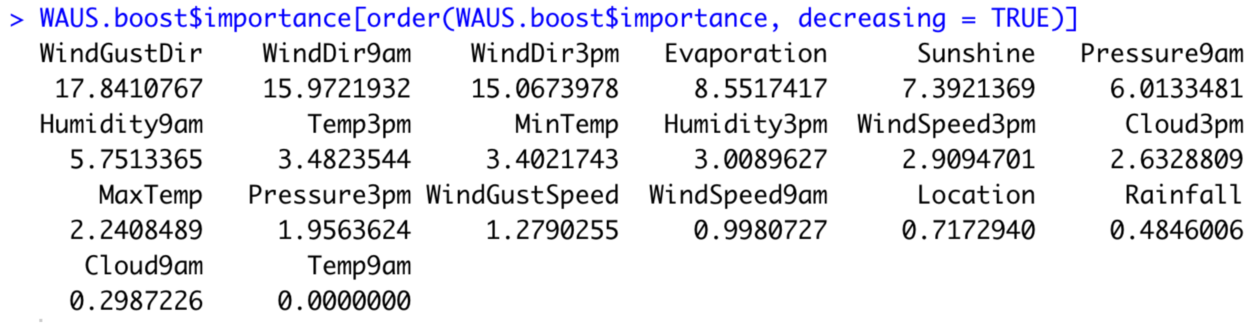
Bagging



```
> WAUS.bag$importance[order(WAUS.bag$importance, decreasing = TRUE)]
  WindGustDir    WindDir9am    WindDir3pm   Evaporation   Humidity3pm       MaxTemp 
   17.0867605    16.9194088    15.7541006    15.7212435     5.1784243     5.0454523 
     Sunshine       Temp3pm       MinTemp WindGustSpeed  WindSpeed9am       Temp9am 
    3.5004038     3.0352796     2.9824120     2.9562681     2.8412517     2.4217943 
  Pressure9am   Pressure3pm   Humidity9am  WindSpeed3pm      Location      Cloud3pm 
    1.5924850     1.5802097     1.5422304     0.9971883     0.5228280     0.3222591 
     Cloud9am      Rainfall 
    0.0000000     0.0000000 
```

Based on the variable importance measured, the most important attributes when using Bagging techniques is WindGustDir (Direction of Strongest wind Gust Over the Day) column. Moreover, the variables that can be omitted that will have less effect on the performance of the model are Rainfall and Cloud9am as they are not related to the performance of the model. In addition, Cloud3pm and Location column can also be omitted as the value of its importance are less than 5% of the importance value of most important attribute which is WindGustDir.

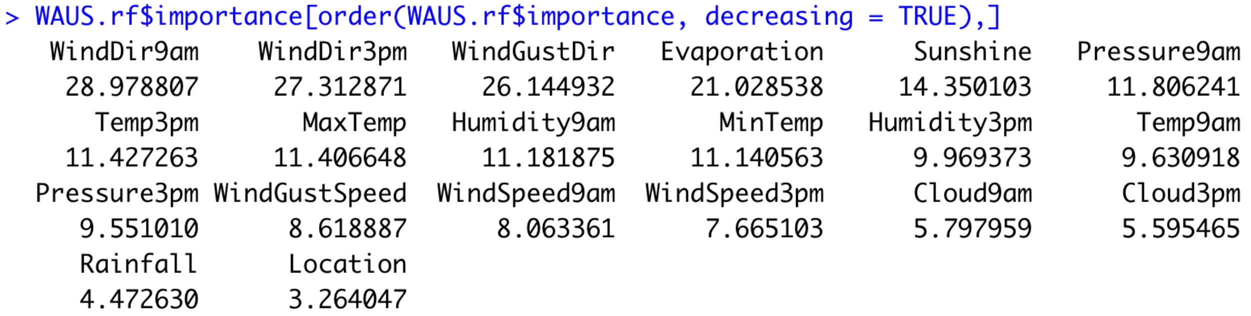
Boosting



```
> WAUS.boost$importance[order(WAUS.boost$importance, decreasing = TRUE)]
  WindGustDir    WindDir9am    WindDir3pm   Evaporation      Sunshine   Pressure9am 
   17.8410767    15.9721932    15.0673978     8.5517417     7.3921369     6.0133481 
  Humidity9am       Temp3pm       MinTemp   Humidity3pm  WindSpeed3pm      Cloud3pm 
    5.7513365     3.4823544     3.4021743     3.0089627     2.9094701     2.6328809 
      MaxTemp   Pressure3pm WindGustSpeed  WindSpeed9am      Location      Rainfall 
    2.2408489     1.9563624     1.2790255     0.9980727     0.7172940     0.4846006 
     Cloud9am       Temp9am 
    0.2987226     0.0000000 
```

Furthermore, the most important attributes when using Boosting techniques is also WindGustDir column. In addition, Temp9am column can be omitted as it is not relevant to the performance of the model. Cloud9am, Rainfall, and Location columns can also be removed from the model as their values are less than 5% of the importance value of the most important variable in the model that is WindGustDir.

Random Forest

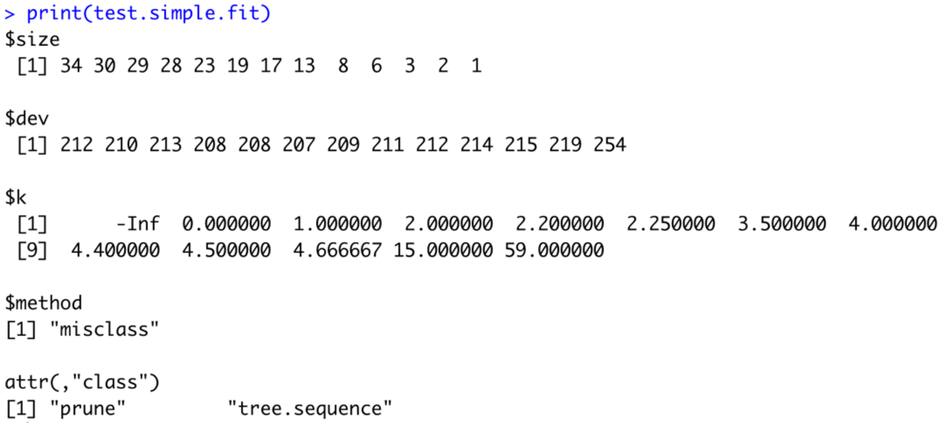


```
> WAUS.rf$importance[order(WAUS.rf$importance, decreasing = TRUE),]
   WindDir9am    WindDir3pm   WindGustDir   Evaporation      Sunshine   Pressure9am 
    28.978807     27.312871     26.144932     21.028538     14.350103     11.806241 
      Temp3pm       MaxTemp   Humidity9am       MinTemp   Humidity3pm       Temp9am 
    11.427263     11.406648     11.181875     11.140563      9.969373      9.630918 
  Pressure3pm WindGustSpeed  WindSpeed9am  WindSpeed3pm      Cloud9am      Cloud3pm 
     9.551010      8.618887      8.063361      7.665103      5.797959      5.595465 
     Rainfall      Location 
     4.472630      3.264047 
```

The most important variable to the performance of random forest model is WinDir9am. Unlike the other techniques which have several columns not related to the model, all predictors in random forest model are being used. If one of the attributes need to be omitted, it is the Location column as all the importance value in each column are more than 5% of the WindDir9am importance value.

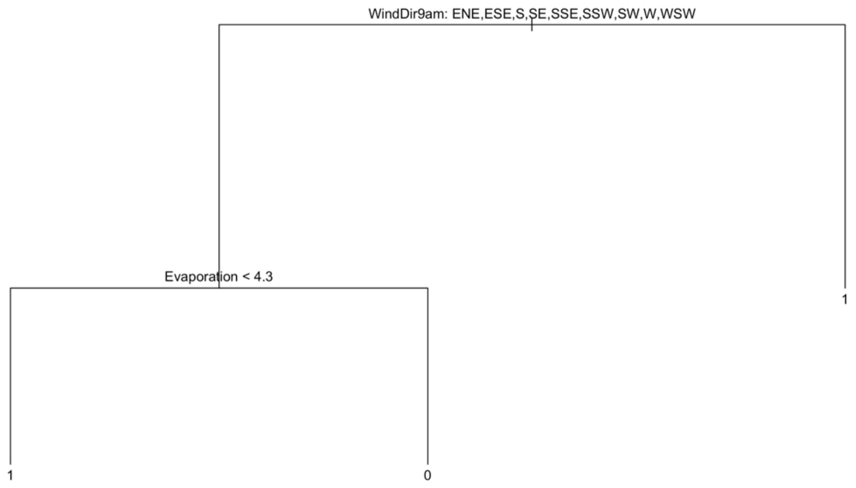
Question 9 – Simple Classifier

The decision tree technique is being used to create a simple classifier for people to understand whether tomorrow will be warmer or not by hand as they provide a tree graph that can be easily understood. Firstly, do a cross validation of the previous decision tree to see what size suit best for the simple classifier.



```
> print(test.simple.fit)
$size
 [1] 34 30 29 28 23 19 17 13  8  6  3  2  1

$dev
 [1] 212 210 213 208 208 207 209 211 212 214 215 219 254

$k
 [1]      -Inf  0.000000  1.000000  2.000000  2.200000  2.250000  3.500000  4.000000
 [9]  4.400000  4.500000  4.666667 15.000000 59.000000

$method
[1] "misclass"

attr(,"class")
[1] "prune"         "tree.sequence"
```

To have a simple classifier, the smaller size should be chosen. Based on the cross-validation test, size 30 has the least misclassification error. If a size of 30 is chosen, then the classifier would not be a simple one. Therefore, it is needed to choose the smaller size ranging from 1-5, by looking in its misclassification error, size 3 will be chosen as it has the smallest value of misclassification error rather than the other sizes that are 1 and 2. Then, prune the previous decision tree with its best/size equals to 3.



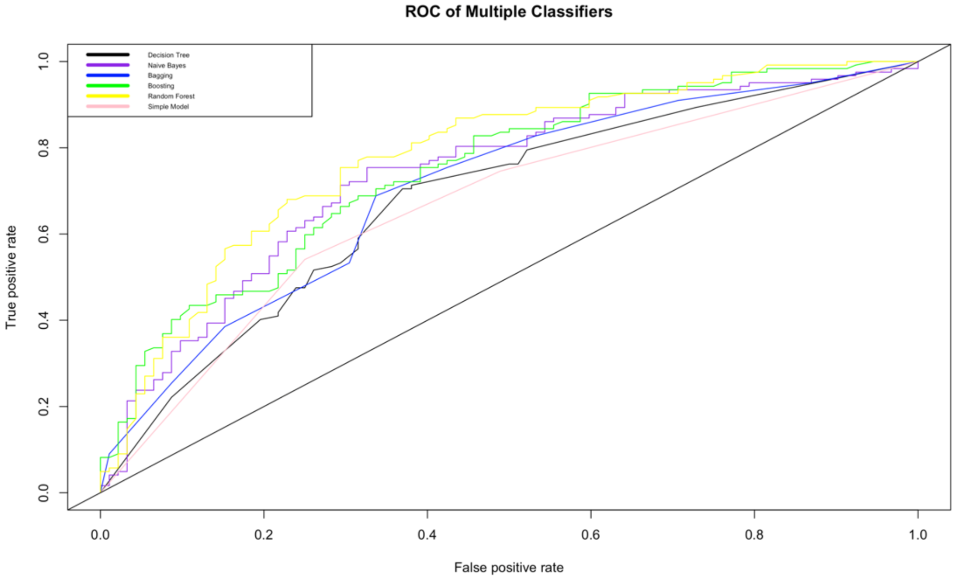
Due to the pruning, the simple tree graph provided above is obtained with only WindDir9am and Evaporation attribute used in the tree construction. From the previous analysis, it is known that both attributes are two of the most important variables used in the previous decision tree model. To interpret this simple model, when WindDir9am value equals to one of the mentioned directions in the graph then go to the left node if not then tomorrow will be warmer than today. Next, if the evaporation is less than 4.3 then tomorrow will be warmer than today and vice versa. Therefore, this simple decision tree graph will be easily implemented to classify whether it will be warmer or not by hand.

Accuracy, ROC and AUC

Let’s count the simple model accuracy by using the same technique as the previous decision tree model.

```
         actual
predicted  0  1
        0 47 31
        1 45 91
```

Hence, the accuracy of this simple model is (91+47)/(91+47+31+45) = 0.6449.

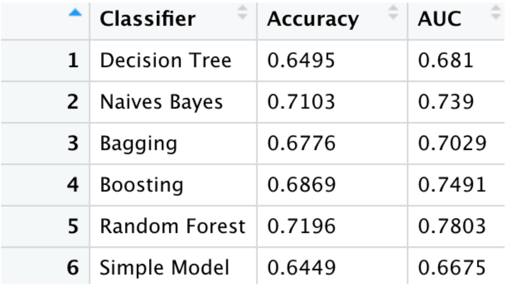


From its ROC, the performance of the simple model can be categorized as a good model as its curve is not closer to the diagonal line. The closer the curve to the diagonal line, the worse the model will be. Next, let’s find it Area Under Curve (AUC) value.

```
cauc.simple.dt <- performance(WAUS.simple.dpred, "auc")
dt.simple.auc <- round(as.numeric(cauc.simple.dt@y.values),4)
dt.simple.auc
```

Using the same techniques as the previous decision tree, WAUS.simple.dpred use the second column of WAUS.simple.pred.tree and WAUS.tree$WarmerTomorrow column to get its predicted values. After that, the AUC can be calculated by using performance function and that is 0.6675.

Comparison of Simple Model Performance to Other Classifiers

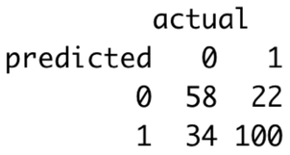


	Classifier	Accuracy	AUC
1	Decision Tree	0.6495	0.681
2	Naives Bayes	0.7103	0.739
3	Bagging	0.6776	0.7029
4	Boosting	0.6869	0.7491
5	Random Forest	0.7196	0.7803
6	Simple Model	0.6449	0.6675

As expected from the simple model, the accuracy and AUC of the model is the lowest from all the other classifiers. This happens because only 2 predictor variables out of 20 variables are used from WAUS data. Although it is said as the worst model out of the other classifiers, the simple model is not as bad as the others especially decision tree because its accuracy and AUC values are relatively the same to the decision tree's values.

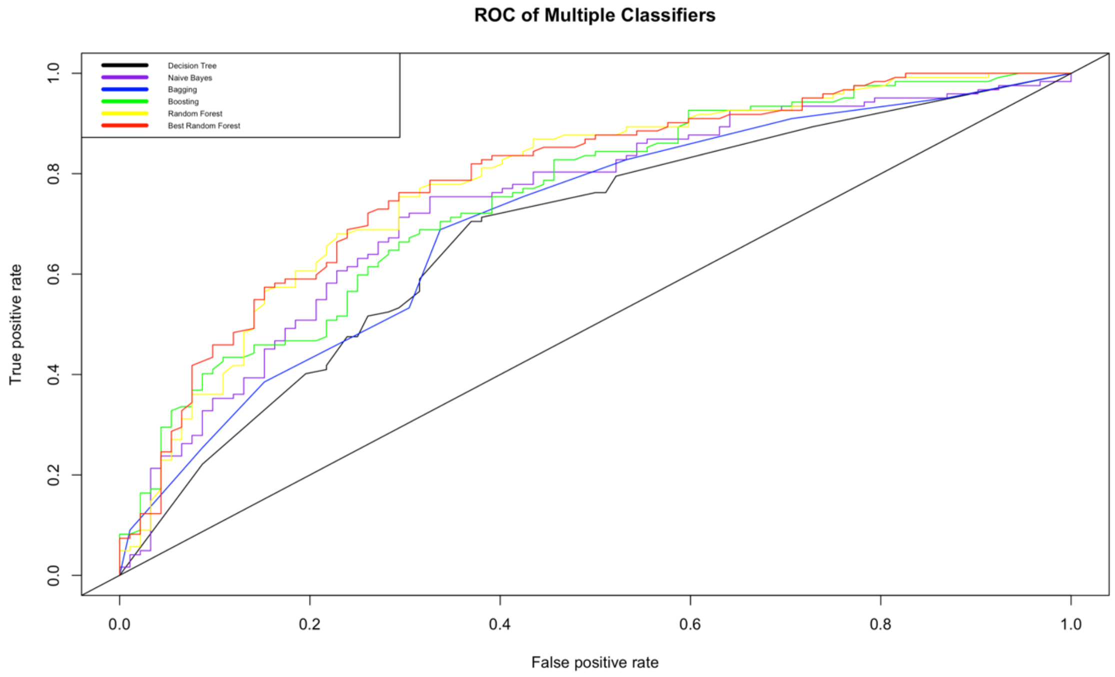
Question 10 – Best Classifier

From the previous analysis, it is observed that Random Forest is the best classifier. Thus, let's try to improve that model to obtain the best classifier. Furthermore, Location column is dropped as it has the least effect on previous random forest model. In addition, cross-validation is not done because in random forest cross-validation has been done internally during the run of the randomForest function. Hence, after creating random forest model by using previous techniques, the accuracy of its model can be obtained.



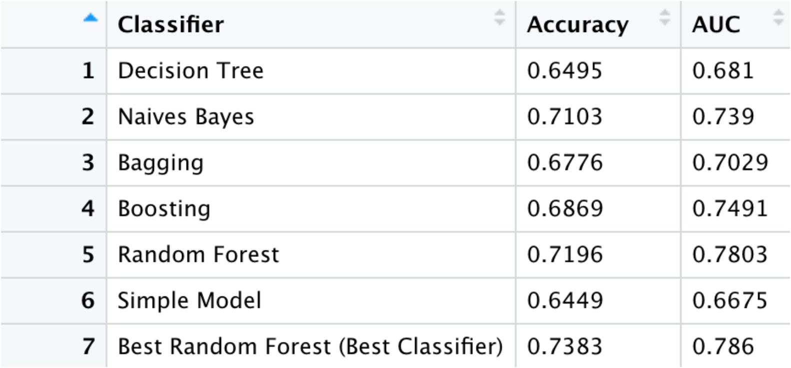
```
         actual
predicted   0   1
        0  58  22
        1  34 100
```

The accuracy of this model is (100+58)/(100+58+22+34) = 0.7383. To compare whether this classifier is the best out of the other classifiers or not, let's try to plot its ROC curve and calculate its AUC value.



From the above ROC graph, Best Random Forest curve is the closest to (1,1) which means it is the best classifier. In addition, using the same technique from the previous calculated AUC of random forest, the calculated AUC of the best random forest classifier is 0.786.

Comparison of Best Classifier Performance to Other Classifiers



	Classifier	Accuracy	AUC
1	Decision Tree	0.6495	0.681
2	Naives Bayes	0.7103	0.739
3	Bagging	0.6776	0.7029
4	Boosting	0.6869	0.7491
5	Random Forest	0.7196	0.7803
6	Simple Model	0.6449	0.6675
7	Best Random Forest (Best Classifier)	0.7383	0.786

From the comparison table provided, the value of accuracy and AUC values of Best Random Forest (Best Classifier) are the highest among the others. Hence, it can be said that this improved random forest classifier is the best tree-based classifier.

Question 11 – Artificial Neural Network

Data Pre-Processing:

Clear the workspace and create the individual data from the code provided in the assignment with WAUS variable. Then, dropped Day, Month and Year columns as they are not relevant to the analysis. In addition, dropped the Location column as from the previous analysis it increases the accuracy as well as the AUC of the best classifier. Then, use complete.cases function to delete all the rows that have null values.

Neuralnet function only accepts integer values, thus, all character type columns needed to be changed into an integer type by creating a model matrix for WindGustDir, WindDir9am and WindDir3pm column and store it into WAUS.mm variable. Furthermore, bind the column of WAUS and WAUS.mm and store it into WAUS variable. Moreover, as the integer value of the previous character type columns have been created, those columns can be dropped from WAUS variable.

Analysis:

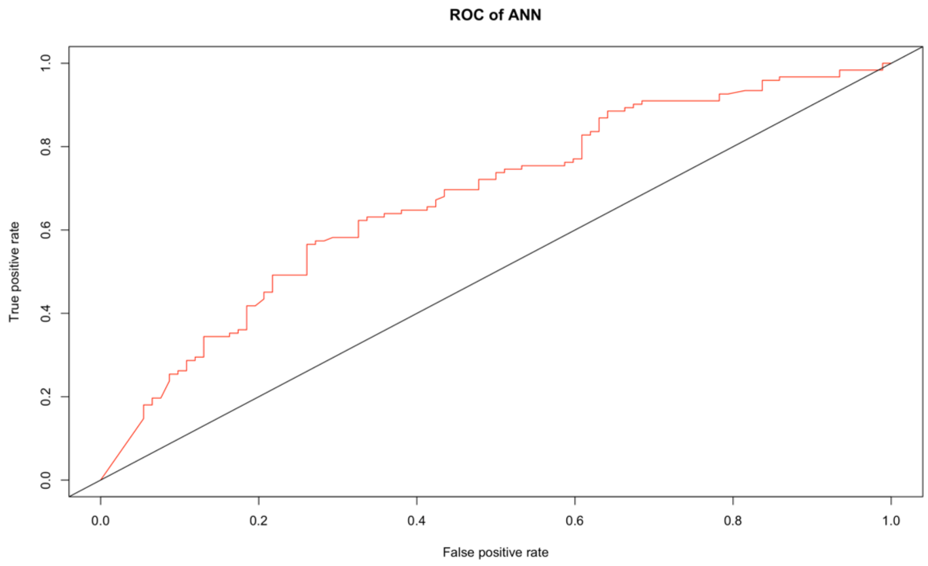
```
> WAUS.nn <- neuralnet(WarmerTomorrow~MinTemp + MaxTemp + Rainfall + Evaporation + Sunshine + WindGustSpeed + WindSpeed9a
m + WindSpeed3pm + Humidity9am + Humidity3pm + Pressure9am+ Pressure3pm + Cloud9am + Cloud3pm + Temp9am + Temp3pm + WindG
ustDirENE + WindGustDirESE + WindGustDirN + WindGustDirNE + WindGustDirNNE + WindGustDirNNW+WindGustDirNW + WindGustDirS
 + WindGustDirSE + WindGustDirSSE + WindGustDirSSW + WindGustDirSW + WindGustDirW + WindGustDirWNW + WindGustDirWSW + Win
dDir9amENE + WindDir9amESE + WindDir9amN + WindDir9amNE + WindDir9amNNE + WindDir9amNNW + WindDir9amNW + WindDir9amS + Wi
ndDir9amSE + WindDir9amSSE + WindDir9amSSW + WindDir9amSW + WindDir9amW + WindDir9amWNW + WindDir9amWSW + WindDir3pmENE +
 WindDir3pmESE + WindDir3pmN + WindDir3pmNE + WindDir3pmNNE + WindDir3pmNNW + WindDir3pmNW + WindDir3pmS + WindDir3pmSE +
 WindDir3pmSSE + WindDir3pmSSW + WindDir3pmSW + WindDir3pmW + WindDir3pmWNW + WindDir3pmWSW , data= WAUS.nn.train,hidden
 = 25, linear.output = FALSE)
```

To create a neural net, it is needed to type all the predictor variables into the argument. In addition, to calculate number of hidden values is to divide the number of rows which is 496 by 10 then divide by two. Therefore, 24.8 is obtained from the calculation and rounded up to 25. Thus, 25 is used as the hidden value.

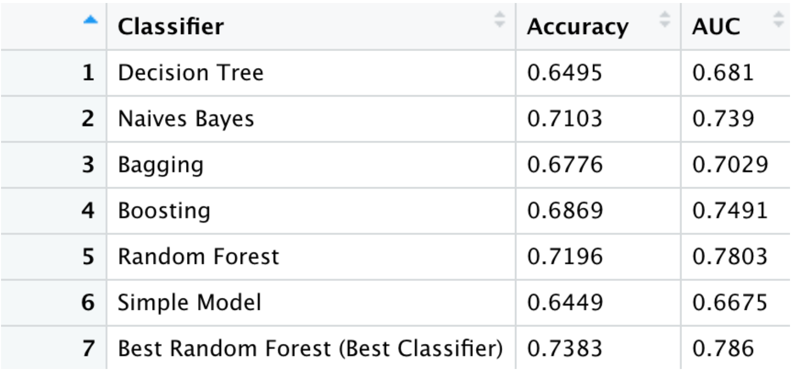
Compare Accuracy, AUC, and ROC with other classifiers
To obtain its performance, let’s count its accuracy.

```
        prediction
actual   0  1
     0 62 30
     1 46 76
```

From the confusion matrix provided above, the accuracy of ANN model is (76+62)/(76+62+30+46) = 0.6449. In addition, its calculated AUC using the previous techniques as the other classifiers is 0.677.



From the ROC curve provided above, ANN technique gives us a good model as its curve does not lie on the diagonal line. Although its curve is not close to (1,1), it can be categorized as a good model.



	Classifier	Accuracy	AUC
1	Decision Tree	0.6495	0.681
2	Naives Bayes	0.7103	0.739
3	Bagging	0.6776	0.7029
4	Boosting	0.6869	0.7491
5	Random Forest	0.7196	0.7803
6	Simple Model	0.6449	0.6675
7	Best Random Forest (Best Classifier)	0.7383	0.786

With the value of accuracy and AUC 0.6449 and 0.677 of ANN model respectively, the performance of ANN is the same as the Simple Model Performance. It can be categorized as the worst classifier before Simple Model. This might happen because lack of data, ANN model will work best if the data is huge. On the other hand, the number of rows of the train data is only 496 rows which can be classified as a small amount of data. Therefore, ANN performs worse than other classifiers.

Conclusion

From this assignment, the objective to gain familiarity by implementing classification model using these techniques, such as: Decision Tree, Naïve Bayes, Bagging, Boosting, Random Forest and Artificial Neural Network to predict whether tomorrow will be warmer than today for 10 locations in Australia has been achieved as well as the analysis of the accuracy, ROC curve, and AUC values.

Appendix

```
# Set Working Directory Based on your devices
getwd()
setwd("/Users/Juliet/Documents/Monash_Uni/FIT3152/Assignment/2")

# Install Packages (If you have not have these packages)
install.packages("tree")
install.packages("e1071")
install.packages("ROCR")
install.packages("randomForest")
install.packages("adabag")
install.packages("rpart")

# Load the Packages
library(tree)
library(e1071)
library(ROCR)
library(randomForest)
library(adabag)
library(rpart)

# Create individual data based on requirements
rm(list = ls())
WAUS <- read.csv("WarmerTomorrow2022.csv")
L <- as.data.frame(c(1:49))
set.seed(32112602) # Your Student ID is the random seed
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

# 1
# Find the unique values of Warmer Tomorrow
unique(WAUS$WarmerTomorrow)

# NA values are present. It needs to be omitted
WAUS <- subset(WAUS, is.na(WAUS$WarmerTomorrow) == FALSE)

# Check the unique values once more
# Only 1 and 0 unique values are present. Thus, NA values has successfully omitted
unique(WAUS$WarmerTomorrow)

# Proportion of days when it is warmer than the previous day compared to those where it is cooler
warmer.pp <- round(nrow(WAUS[WAUS$WarmerTomorrow == 1,])/nrow(WAUS)*100, 2) #
54.79%
paste(warmer.pp,"%", sep = '')
```

```
# Description of predictor (independent) variables
round(sapply(WAUS, sd, na.rm = TRUE),4)
summary(WAUS)

# 2
# Drop Day, Month, Year columns as they are not relevant to our model.
WAUS <- subset(WAUS, select = -c(Day, Month, Year))

# Delete all rows that have null values in the column
WAUS <- WAUS[complete.cases(WAUS),]
dim(WAUS)

# Change chr type and WarmerTomorrow column into Factor
str(WAUS)
WAUS[c("WindGustDir", "WindDir9am", "WindDir3pm","WarmerTomorrow")] <-
lapply(WAUS[c("WindGustDir", "WindDir9am", "WindDir3pm","WarmerTomorrow")],
as.factor)
str(WAUS)

# 3
# Divide 70% train data - 30% test data
set.seed(32112602) #Student ID as random seed
train.row = sample(1:nrow(WAUS), 0.7*nrow(WAUS))
WAUS.train = WAUS[train.row,]
WAUS.test = WAUS[-train.row,]

# 4
# Decision Tree
WAUS.tree <- tree(WarmerTomorrow~., data = WAUS.train)
plot(WAUS.tree)
text(WAUS.tree, pretty = 0)
summary(WAUS.tree)

# Naive Bayes
WAUS.bayes <- naiveBayes(WarmerTomorrow~., data = WAUS.train)

# Bagging
WAUS.bag <- bagging(WarmerTomorrow~., data = WAUS.train, mfinal = 10)

# Boosting
WAUS.boost <- boosting(WarmerTomorrow~., data = WAUS.train, mfinal= 10)

# Random Forest
WAUS.rf <- randomForest(WarmerTomorrow~., data = WAUS.train)
```

```
# 5
# Decision Tree
WAUS.pred.tree <- predict(WAUS.tree, WAUS.test, type = 'class')
WAUS.pred.tree
dt.t <- table(predicted = WAUS.pred.tree, actual = WAUS.test$WarmerTomorrow)
dt.t

dt.a <- round((60+75)/nrow(WAUS.test),4)
dt.a
# Accuracy = (60+75)/nrow(WAUS.test) = 0.6308

# Naive Bayes
WAUS.predbayes <- predict(WAUS.bayes, WAUS.test)
nb.t <- table(predicted = WAUS.predbayes, actual = WAUS.test$WarmerTomorrow)
nb.t

nb.a <- round((64+88)/nrow(WAUS.test),4)
nb.a
# Accuracy =(64+88)/nrow(WAUS.test) = 0.7103

# Bagging
WAUSpred.bag <- predict.bagging(WAUS.bag, WAUS.test)
WAUSpred.bag$confusion
bag.a <- round((53+92)/nrow(WAUS.test),4)
bag.a
# Accuracy =(53+92)/nrow(WAUS.test) = 0.6776

# Boosting
WAUSpred.boost <- predict.boosting(WAUS.boost, WAUS.test)
WAUSpred.boost$confusion
boo.a <- round((61+86)/nrow(WAUS.test),4)
boo.a
# Accuracy =(61+86)/nrow(WAUS.test) = 0.6869

# Random Forest
WAUSpred.rf <- predict(WAUS.rf, WAUS.test)
rf.t <- table(predicted = WAUSpred.rf, actual = WAUS.test$WarmerTomorrow)
rf.t

rf.a <- round((59+95)/nrow(WAUS.test),4)
rf.a
# Accuracy = (59+95)/nrow(WAUS.test) = 0.7196

# 6
# Decision Tree
# Calculate the confidence of predicting ‘warmer tomorrow’ and construct an ROC curve
```

```
WAUS.pred.tree <- predict(WAUS.tree, WAUS.test, type = 'vector')
WAUSdpred <- prediction(WAUS.pred.tree[,2], WAUS.test$WarmerTomorrow)
WAUSdperf <- performance(WAUSdpred, 'tpr','fpr')
plot(WAUSdperf)
abline(0,1)

cauc.dt <- performance(WAUSdpred, "auc")
dt.auc <- round(as.numeric(cauc.dt@y.values),4)
dt.auc
# AUC = 0.681

# Naive Bayes
# Calculate the confidence of predicting ‘warmer tomorrow’ and construct an ROC curve
WAUSpred.bayes <- predict(WAUS.bayes, WAUS.test, type ='raw')
WAUSbpred <- prediction(WAUSpred.bayes[,2], WAUS.test$WarmerTomorrow)
WAUSbperf <- performance(WAUSbpred, 'tpr','fpr')
plot(WAUSbperf, add = TRUE, col = 'blueviolet')

cauc.nb <- performance(WAUSbpred, "auc")
nb.auc <- round(as.numeric(cauc.nb@y.values),4)
nb.auc
# AUC = 0.739

# Bagging
# Calculate the confidence of predicting ‘warmer tomorrow’ and construct an ROC curve
WAUSbagpred <- prediction(WAUSpred.bag$prob[,2], WAUS.test$WarmerTomorrow)
WAUSbagperf <- performance(WAUSbagpred, 'tpr','fpr')
plot(WAUSbagperf, add = T, col = 'blue')

cauc.bag <- performance(WAUSbagpred, "auc")
bag.auc <- round(as.numeric(cauc.bag@y.values),4)
bag.auc
# AUC = 0.7029

# Boosting
# Calculate the confidence of predicting ‘warmer tomorrow’ and construct an ROC curve
WAUSboostpred <- prediction(WAUSpred.boost$prob[,2], WAUS.test$WarmerTomorrow)
WAUSboostperf <- performance(WAUSboostpred, 'tpr','fpr')
plot(WAUSboostperf, add = T, col = 'red')

cauc.boo <- performance(WAUSboostpred, "auc")
boo.auc <- round(as.numeric(cauc.boo@y.values),4)
boo.auc
# AUC = 0.7491

# Random Forest
# Calculate the confidence of predicting ‘warmer tomorrow’ and construct an ROC curve
WAUSrfpred <- predict(WAUS.rf, WAUS.test, type = "prob")
WAUSpred <- prediction(WAUSrfpred[,2], WAUS.test$WarmerTomorrow)
WAUSrfperf <- performance(WAUSpred, 'tpr','fpr')
plot(WAUSrfperf, add = T, col = 'green')

cauc.rf <- performance(WAUSpred, "auc")
rf.auc <- round(as.numeric(cauc.rf@y.values),4)
rf.auc
# AUC = 0.7803

# Completing ROC curve graph
title("ROC of Multiple Classifiers")
legend("topleft",
     legend=c("Decision Tree", "Naive Bayes", "Bagging", "Boosting", "Random Forest"),
     col=c("black",  "blueviolet","blue", "red","green"),
     lwd=4, cex =0.5, xpd = FALSE, horiz = FALSE)

# 7
# Create comparison table for Accuracy and AUC for each classifiers
table.compare <- data.frame(Classifier = c("Decision Tree", "Naives Bayes",
"Bagging","Boosting","Random Forest"), Accuracy = c(dt.a, nb.a, bag.a, boo.a, rf.a), AUC =
c(dt.auc, nb.auc, bag.auc, boo.auc, rf.auc))
View(table.compare)

# 8 - Variable Importance
# Decision Tree
summary(WAUS.tree)
WAUS.bag$importance[order(WAUS.bag$importance, decreasing = TRUE)]
WAUS.boost$importance[order(WAUS.boost$importance, decreasing = TRUE)]
WAUS.rf$importance[order(WAUS.rf$importance, decreasing = TRUE),]

# 9  - Create Simple Model
test.simple.fit <- cv.tree(WAUS.tree, FUN = prune.misclass)
print(test.simple.fit)
# Should have choose size 30 due to the lowest misclassification rate . But for simplicity of the
graph, use smaller size number ranging from 2-5.
# The lowest misclassification rate from size 1-5 is 3.
prune.WAUS.fit <- prune.misclass(WAUS.tree, best = 3)
summary(prune.WAUS.fit)
plot(prune.WAUS.fit)
text(prune.WAUS.fit, pretty = 0)

WAUS.simple.predict <- predict(prune.WAUS.fit, WAUS.test, type = 'class')
WAUS.simple.t<- table(predicted = WAUS.simple.predict, actual =
WAUS.test$WarmerTomorrow)
```

```
print(WAUS.simple.t)

dt.simple.a <- round((47 + 91)/nrow(WAUS.test),4)
dt.simple.a
# Accuracy = (47 + 91)/nrow(WAUS.test) = 0.6449

# Calculate the confidence of predicting ‘warmer tomorrow’ and construct an ROC curve of multiple classifiers
WAUS.simple.pred.tree <- predict(prune.WAUS.fit, WAUS.test, type = 'vector')
WAUS.simple.dpred <- prediction(WAUS.simple.pred.tree[,2], WAUS.test$WarmerTomorrow)
WAUS.simple.dperf <- performance(WAUS.simple.dpred, 'tpr','fpr')
plot(WAUSdperf)
plot(WAUSbperf, add = T, col = 'blueviolet')
plot(WAUSbagperf, add = T, col = 'blue')
plot(WAUSboostperf, add = T, col = 'green')
plot(WAUSrfperf, add = T, col = 'yellow')
plot(WAUS.simple.dperf, add = T, col = "pink")
title("ROC of Multiple Classifiers")
legend("topleft",
    legend=c("Decision Tree", "Naive Bayes", "Bagging", "Boosting", "Random Forest", "Simple Model"),
    col=c("black",  "blueviolet","blue", "green","yellow","pink"),
    lwd=4, cex =0.5, xpd = FALSE, horiz = FALSE)
abline(0,1)

cauc.simple.dt <- performance(WAUS.simple.dpred, "auc")
dt.simple.auc <- round(as.numeric(cauc.simple.dt@y.values),4)
dt.simple.auc
# AUC = 0.6675

# Insert the accuracy and AUC simple model's values into table.compare
simple.values <- c("Simple Model", dt.simple.a, dt.simple.auc)
table.compare <- rbind(table.compare, simple.values)
View(table.compare)

# 10 - Best Classifier
# Choose to improve RF since it has the highest Accuracy and AUC out of the other classifiers in Q7
set.seed(32112602)
WAUS.rf.imp <- randomForest(WarmerTomorrow~.-Location, data = WAUS.train)
WAUSpred.rf.imp <- predict(WAUS.rf.imp, WAUS.test)
rf.t.imp <- table(predicted = WAUSpred.rf.imp, actual = WAUS.test$WarmerTomorrow)
rf.t.imp

rf.a.imp <- round((58+100)/nrow(WAUS.test),4)
rf.a.imp
```

```
# Accuracy = (58+100)/nrow(WAUS.test) = 0.7383

WAUSrfpred.imp <- predict(WAUS.rf.imp, WAUS.test, type = "prob")
WAUSpred.imp <- prediction(WAUSrfpred.imp[,c("1")], WAUS.test$WarmerTomorrow)
cauc.rf.imp <- performance(WAUSpred.imp, "auc")
rf.auc.imp <- round(as.numeric(cauc.rf.imp@y.values),4)
rf.auc.imp
# AUC = 0.786

# Calculate the confidence of predicting ‘warmer tomorrow’ and construct an ROC curve of multiple classifiers
WAUS.rf.perf <- performance(WAUSpred.imp, 'tpr','fpr')
plot(WAUSdperf)
plot(WAUSbperf, add = T, col = 'blueviolet')
plot(WAUSbagperf, add = T, col = 'blue')
plot(WAUSboostperf, add = T, col = 'green')
plot(WAUSrfperf, add = T, col = 'yellow')
plot(WAUS.rf.perf, add = T, col="red")
abline(0,1)
title("ROC of Multiple Classifiers")
legend("topleft",
    legend=c("Decision Tree", "Naive Bayes", "Bagging", "Boosting", "Random Forest", "Best Random Forest"),
    col=c("black",  "blueviolet","blue", "green","yellow","red"),
    lwd=4, cex =0.5, xpd = FALSE, horiz = FALSE)

# Insert the accuracy and AUC best classifier's values into table.compare
best.values <- c("Best Random Forest (Best Classifier)", rf.a.imp, rf.auc.imp)
table.compare <- rbind(table.compare, best.values)
View(table.compare)

# 11 - ANN
install.packages('neuralnet')
install.packages('car')

library(neuralnet)
library(car)

# Create individual data based on requirements
rm(list = ls())
WAUS <- read.csv("WarmerTomorrow2022.csv")
L <- as.data.frame(c(1:49))
set.seed(32112602) # Your Student ID is the random seed
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
```

```
# Delete the rows that have null values of WarmerTomorrow
WAUS <- subset(WAUS, is.na(WAUS$WarmerTomorrow) == FALSE)

# Drop Day, Month, Year, Location columns as they are not relevant to our model and deletion of
Location column are proved to improve the accuracy of best classifier random forest
WAUS <- subset(WAUS, select = -c(Day, Month, Year, Location))

# Delete all rows that have null values
WAUS <- WAUS[complete.cases(WAUS),]
dim(WAUS)

# Create a model matrix for chr type columns
WAUS.mm <- model.matrix(~WindGustDir+WindDir9am+WindDir3pm,data= WAUS)
WAUS <- cbind(WAUS ,WAUS.mm)
WAUS <- subset(WAUS, select = -c(WindGustDir, WindDir9am, WindDir3pm, `(Intercept)`))

# Divide 70% Train Data - 30% Test Data
set.seed(32112602) #Student ID as random seed
train.row = sample(1:nrow(WAUS), 0.7*nrow(WAUS))
WAUS.nn.train = WAUS[train.row,]
WAUS.nn.test = WAUS[-train.row,]

# Build ANN model (May take some time)
WAUS.nn <- neuralnet(WarmerTomorrow~MinTemp + MaxTemp + Rainfall + Evaporation +
Sunshine + WindGustSpeed + WindSpeed9am + WindSpeed3pm + Humidity9am + Humidity3pm
+ Pressure9am+ Pressure3pm + Cloud9am + Cloud3pm + Temp9am + Temp3pm +
WindGustDirENE + WindGustDirESE + WindGustDirN + WindGustDirNE + WindGustDirNNE
+ WindGustDirNNW+WindGustDirNW + WindGustDirS + WindGustDirSE + WindGustDirSSE
+ WindGustDirSSW + WindGustDirSW + WindGustDirW + WindGustDirWNW +
WindGustDirWSW + WindDir9amENE + WindDir9amESE + WindDir9amN + WindDir9amNE
+ WindDir9amNNE + WindDir9amNNW + WindDir9amNW + WindDir9amS + WindDir9amSE
+ WindDir9amSSE + WindDir9amSSW + WindDir9amSW + WindDir9amW +
WindDir9amWNW + WindDir9amWSW + WindDir3pmENE + WindDir3pmESE +
WindDir3pmN + WindDir3pmNE + WindDir3pmNNE + WindDir3pmNNW + WindDir3pmNW
+ WindDir3pmS + WindDir3pmSE + WindDir3pmSSE + WindDir3pmSSW + WindDir3pmSW
+ WindDir3pmW + WindDir3pmWNW + WindDir3pmWSW , data= WAUS.nn.train,hidden =
25, linear.output = FALSE)

# Predict
WAUS.nn.pred <-compute(WAUS.nn, WAUS.nn.test[,c("MinTemp","MaxTemp","Rainfall",
"Evaporation" , "Sunshine" , "WindGustSpeed" , "WindSpeed9am" , "WindSpeed3pm" ,
"Humidity9am" , "Humidity3pm" , "Pressure9am", "Pressure3pm" , "Cloud9am" , "Cloud3pm" ,
"Temp9am" , "Temp3pm" , "WindGustDirENE" , "WindGustDirESE" , "WindGustDirN" ,
"WindGustDirNE" , "WindGustDirNNE" , "WindGustDirNNW","WindGustDirNW" ,
"WindGustDirS" , "WindGustDirSE" , "WindGustDirSSE" , "WindGustDirSSW" ,
"WindGustDirSW" , "WindGustDirW" , "WindGustDirWNW" , "WindGustDirWSW",
"WindDir9amENE" , "WindDir9amESE" , "WindDir9amN" , "WindDir9amNE" ,
"WindDir9amNNE" , "WindDir9amNNW" , "WindDir9amNW" , "WindDir9amS" ,
"WindDir9amSE" , "WindDir9amSSE" , "WindDir9amSSW" , "WindDir9amSW" ,
"WindDir9amW" , "WindDir9amWNW" , "WindDir9amWSW" , "WindDir3pmENE" ,
"WindDir3pmESE" , "WindDir3pmN", "WindDir3pmNE" , "WindDir3pmNNE" ,
"WindDir3pmNNW" , "WindDir3pmNW" , "WindDir3pmS" , "WindDir3pmSE" ,
"WindDir3pmSSE" , "WindDir3pmSSW" , "WindDir3pmSW" , "WindDir3pmW" ,
"WindDir3pmWNW" , "WindDir3pmWSW")])
WAUS.nn.pred <- ifelse(WAUS.nn.pred$net.result >= 0.5,1,0)
table(actual = WAUS.nn.test$WarmerTomorrow, prediction = WAUS.nn.pred)

nn.a<-round((62+76)/nrow(WAUS.nn.test),4)
nn.a
# Accuracy (62+76)/nrow(WAUS.nn.test) = 0.6449

WAUSpred.nn <- predict(WAUS.nn, WAUS.nn.test, type = "prob")
detach(package:neuralnet,unload = T)
WAUSpred.nn <- prediction(WAUSpred.nn, WAUS.nn.test$WarmerTomorrow)
cauc.nn <- performance(WAUSpred.nn, "auc")
nn.auc <- round(as.numeric(cauc.nn@y.values),4)
nn.auc
# AUC = 0.677

# Calculate the confidence of predicting ‘warmer tomorrow’ and construct an ROC curve
WAUS.nn.perf <- performance(WAUSpred.nn, 'tpr','fpr')
plot(WAUS.nn.perf, col="red", main="ROC of ANN")
abline(0,1)
```