Name: Jin En Tan

Student ID:31336574

FIT3152 Assignment 2

Report

Question 1:

```
r<=WAUS %>% count(WarmerTomorrow)   #ignoring those NA
  WarmerTomorrow    n
1              0  887
2              1 1094
3           <NA>   19
```

Based on the above screenshot from R-studio console, we can see the number of days which is warmer than previous day indicated by 1 and number of days which is cooler than previous day indicated by 0. We will ignore the number of days with NA as only 19 rows are with NA so it will be less likely to affect the analysis.

```
r<=887/(887+1094)*100  #this is percentage of cooler than previous day
[1] 44.8
r<=1094/(887+1094)*100  #this is percentage of warmer than previous day
[1] 55.2
```

Based on the above screenshot, the proportion of days when it is warmer than the previous day is 55.2% whereas the proportion of days when it is cooler than the previous day is 44.8%. We can see that the percentage of days warmer than previous day is higher than that of cooler than previous day.

```
r<=summary(WAUS)
      Day             Month             Year          Location        MinTemp          MaxTemp          Rainfall        Evaporation        Sunshine
 Min.   : 1.0   Min.   : 1.00   Min.   :2008   Min.   : 7.0   Min.   :-2.4   Min.   : 9.4   Min.   :  0.0   Min.   : 0      Min.   : 0
 1st Qu.: 8.0   1st Qu.: 4.00   1st Qu.:2011   1st Qu.:19.0   1st Qu.: 7.3   1st Qu.:17.1   1st Qu.:  0.0   1st Qu.: 2      1st Qu.: 5
 Median :16.0   Median : 7.00   Median :2014   Median :23.0   Median :11.2   Median :21.9   Median :  0.0   Median : 4      Median : 8
 Mean   :15.7   Mean   : 6.73   Mean   :2014   Mean   :23.6   Mean   :11.4   Mean   :22.6   Mean   :  1.9   Mean   : 5      Mean   : 8
 3rd Qu.:23.0   3rd Qu.:10.00   3rd Qu.:2017   3rd Qu.:31.0   3rd Qu.:15.5   3rd Qu.:27.0   3rd Qu.:  0.6   3rd Qu.: 7      3rd Qu.:11
 Max.   :31.0   Max.   :12.00   Max.   :2019   Max.   :34.0   Max.   :26.6   Max.   :44.1   Max.   :143.8   Max.   :21      Max.   :14
 NA's   :19     NA's   :19      NA's   :19                    NA's   :21     NA's   :14     NA's   :47      NA's   :1025    NA's   :1095
  WindGustDir   WindGustSpeed     WindDir9am      WindDir3pm    WindSpeed9am   WindSpeed3pm    Humidity9am     Humidity3pm      Pressure9am
 SSE    : 186   Min.   :  9.0   N      : 219   S      : 182   Min.   : 0.0   Min.   : 0.0   Min.   : 10.0   Min.   :  4.0   Min.   : 994
 W      : 185   1st Qu.: 30.0   SSE    : 166   W      : 180   1st Qu.: 9.0   1st Qu.:13.0   1st Qu.: 57.0   1st Qu.: 36.0   1st Qu.:1013
 N      : 168   Median : 39.0   S      : 164   WSW    : 159   Median :13.0   Median :19.0   Median : 71.0   Median : 53.0   Median :1018
 S      : 152   Mean   : 40.7   NNE    : 136   N      : 151   Mean   :15.1   Mean   :19.7   Mean   : 69.8   Mean   : 51.8   Mean   :1018
 SW     : 149   3rd Qu.: 50.0   SSW    : 135   SW     : 149   3rd Qu.:20.0   3rd Qu.:24.0   3rd Qu.: 84.0   3rd Qu.: 67.0   3rd Qu.:1023
 (Other):1115   Max.   :111.0   (Other):1081   (Other):1147   Max.   :63.0   Max.   :61.0   Max.   :100.0   Max.   :100.0   Max.   :1037
 NA's   :  45   NA's   :37      NA's   :  99   NA's   :  32   NA's   :27     NA's   :27     NA's   :24      NA's   :30      NA's   :243
  Pressure3pm      Cloud9am       Cloud3pm       Temp9am         Temp3pm      WarmerTomorrow
 Min.   : 992   Min.   :0      Min.   :0      Min.   : 0.9   Min.   : 5.0   0   : 887
 1st Qu.:1012   1st Qu.:1      1st Qu.:2      1st Qu.:11.7   1st Qu.:16.0   1   :1094
 Median :1016   Median :6      Median :6      Median :15.6   Median :20.4   NA's:  19
 Mean   :1016   Mean   :5      Mean   :5      Mean   :16.0   Mean   :21.0
 3rd Qu.:1021   3rd Qu.:7      3rd Qu.:7      3rd Qu.:20.1   3rd Qu.:25.2
 Max.   :1035   Max.   :8      Max.   :8      Max.   :35.3   Max.   :43.0
 NA's   :246    NA's   :934    NA's   :927    NA's   :21     NA's   :23
```

Based on the descriptions of the predictor variables, we can see that the noteworthy thing in the data is that for real-value attributes, there is a lot of NA's in the data. We can see that evaporation, sunshine, cloud9am and cloud3pm contains a lot of NA's which is almost or already exceeding half the data size of 2000. Hence, I consider to remove these 4 attributes from my analysis as there's too many NA's in the data.

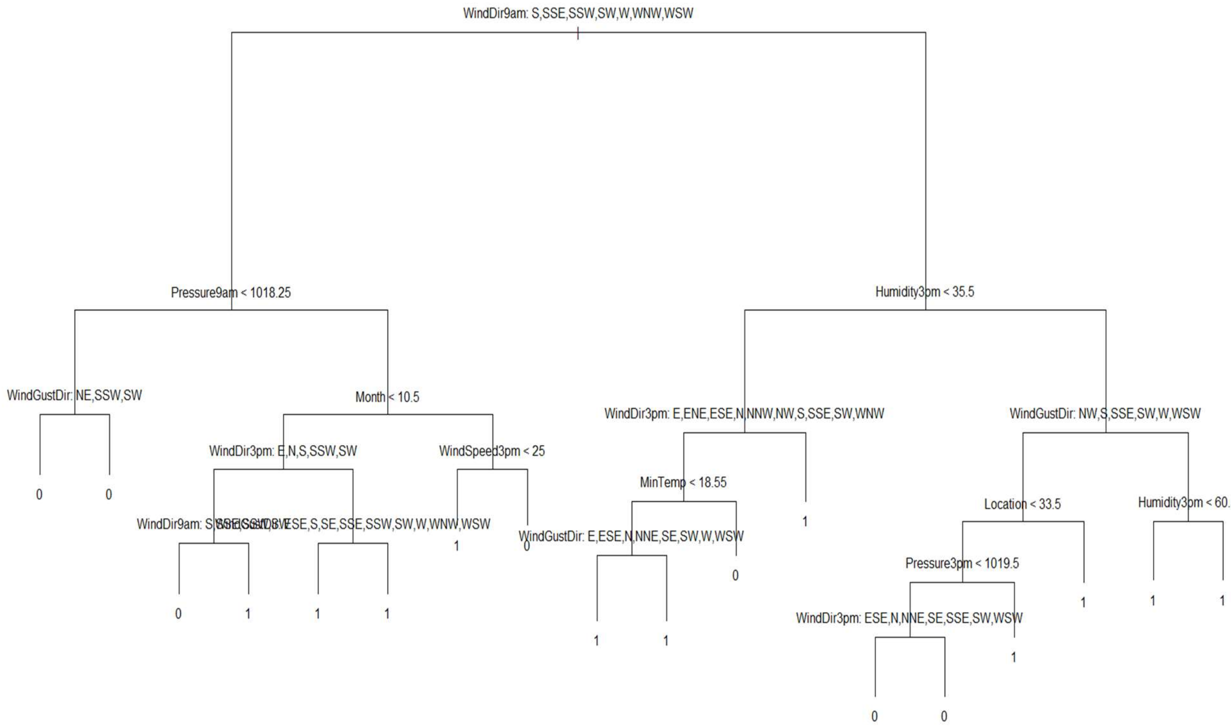
Question 2:

The preprocessing required to make the data set suitable for the model fitting that follows is to firstly remove the 4 attributes with a lot of NA's as mentioned in question 1. After that, since there's still NA's in the remaining attributes, I remove all the rows which contains NA. Screenshot below shows the summary of data after preprocessing and it shows that all remaining attributes has no NA at all.

```
r<=df<-na.omit(df)
r<=summary(df)
      Day            Month            Year         Location        MinTemp          MaxTemp          Rainfall       WindGustDir  WindGustSpeed
 Min.   : 1.0   Min.   : 1.00   Min.   :2008   Min.   : 7.0   Min.   :-2.4   Min.   : 9.4   Min.   :  0.0   W      :144   Min.   : 13.0
 1st Qu.: 8.0   1st Qu.: 4.00   1st Qu.:2011   1st Qu.:19.0   1st Qu.: 7.3   1st Qu.:16.8   1st Qu.:  0.0   SSE    :140   1st Qu.: 31.0
 Median :16.0   Median : 7.00   Median :2014   Median :23.0   Median :11.1   Median :21.7   Median :  0.0   N      :119   Median : 41.0
 Mean   :15.6   Mean   : 6.86   Mean   :2014   Mean   :22.8   Mean   :11.4   Mean   :22.3   Mean   :  1.9   S      :112   Mean   : 42.1
 3rd Qu.:23.0   3rd Qu.:10.00   3rd Qu.:2017   3rd Qu.:27.0   3rd Qu.:15.3   3rd Qu.:26.7   3rd Qu.:  0.8   WSW    :111   3rd Qu.: 50.0
 Max.   :31.0   Max.   :12.00   Max.   :2019   Max.   :34.0   Max.   :26.6   Max.   :44.1   Max.   :128.4   SW     :109   Max.   :111.0
                                                                                                            (Other):671
   WindDir9am     WindDir3pm    WindSpeed9am   WindSpeed3pm  Humidity9am      Humidity3pm      Pressure9am    Pressure3pm       Temp9am
 N      :164   W      :139   Min.   : 2.0   Min.   : 2   Min.   : 10.0   Min.   :  4.0   Min.   : 994   Min.   : 994   Min.   : 0.9
 SSE    :127   S      :134   1st Qu.:11.0   1st Qu.:15   1st Qu.: 57.0   1st Qu.: 37.0   1st Qu.:1013   1st Qu.:1011   1st Qu.:11.7
 S      :119   SW     :112   Median :15.0   Median :20   Median : 70.0   Median : 54.0   Median :1018   Median :1016   Median :15.5
 NNW    :100   WSW    :111   Mean   :16.6   Mean   :21   Mean   : 68.9   Mean   : 52.4   Mean   :1018   Mean   :1016   Mean   :16.0
 NNE    : 96   N      :108   3rd Qu.:20.0   3rd Qu.:26   3rd Qu.: 83.0   3rd Qu.: 67.0   3rd Qu.:1023   3rd Qu.:1021   3rd Qu.:19.8
 SE     : 90   SSE    : 98   Max.   :63.0   Max.   :61   Max.   :100.0   Max.   :100.0   Max.   :1037   Max.   :1034   Max.   :35.3
 (Other):710   (Other):704
    Temp3pm      WarmerTomorrow
 Min.   : 7.9   0:634
 1st Qu.:15.6   1:772
 Median :20.1
 Mean   :20.7
 3rd Qu.:24.8
 Max.   :43.0
```

Question 3 & Question 4:

For these 2 questions, there is nothing to write for the report, refer to the codes in appendix for more details. The image below is the decision tree created that is being visualized.



Question 5:

```
#Decision Tree Confusion
r<=print(t1)
                Actual_Class
Predicted_Class   0   1
              0  89  72
              1  98 163
r<=(89+163)/(89+72+98+163) #accuracy calculation
[1] 0.597
```

Based on the above screenshot which consist of the confusion matrix and accuracy calculation, we can see that the accuracy of decision tree is 0.597.

```
#NaiveBayes Confusion
r<=print(t2)
                Actual_Class
Predicted_Class   0   1
              0  99  58
              1  88 177
r<=(99+177)/(99+58+88+177)
[1] 0.654
```

Based on the above screenshot, we can see that the accuracy for Naïve Bayes is 0.654.

```
r<=dfpred.bag <- predict.bagging(df.bag, df.test)
r<=dfpred.bag$confusion
                Observed Class
Predicted Class   0   1
              0 102  53
              1  85 182
r<=(102+182)/(102+53+85+182)
[1] 0.673
```

Based on the above screenshot, we can see that the accuracy for bagging is 0.673.

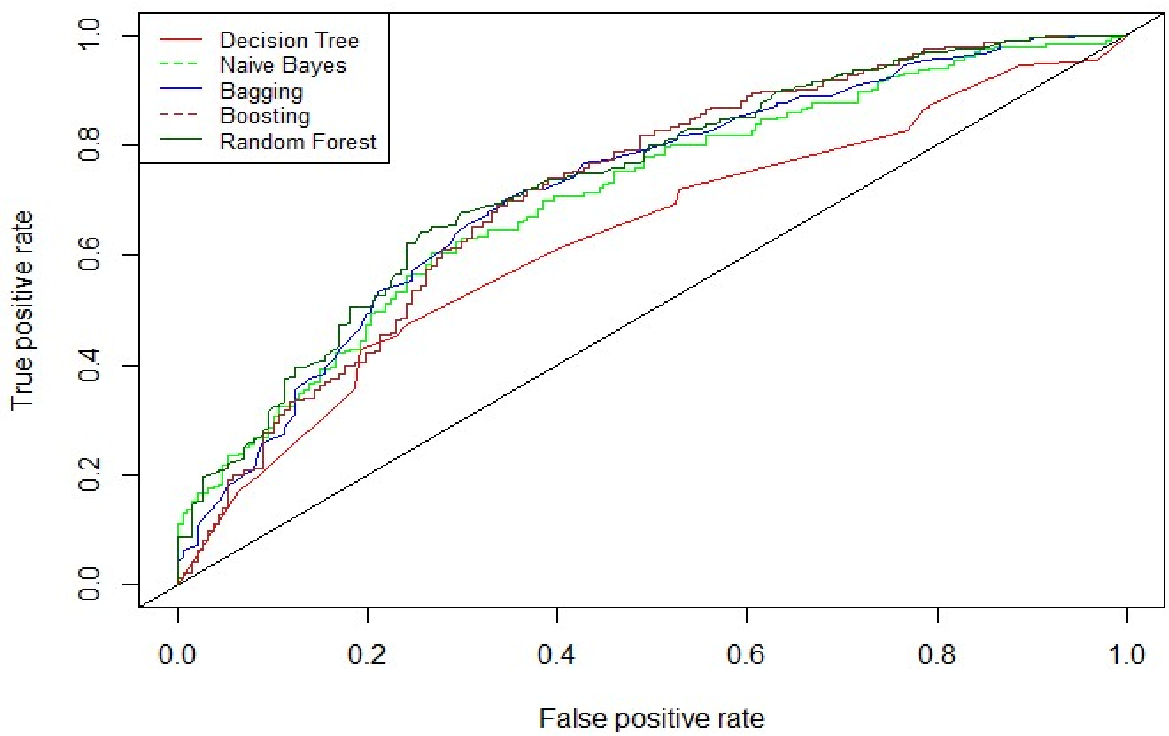
```
r<=dfpred.boost <- predict.boosting(df.boost, newdata=df.test)
r<=dfpred.boost$confusion
               Observed Class
Predicted Class   0   1
              0 111  60
              1  76 175
r<=(111+175)/(111+60+76+175)
[1] 0.678
```

Based on the above screenshot, we can see that the accuracy for boosting is 0.678.

```
#Random Forest Confusion
r<=print(t3)
               Actual_Class
Predicted_Class   0   1
              0  85  40
              1 102 195
r<=(85+195)/(85+40+102+195)
[1] 0.664
```

Based on the above screenshot, we can see that the accuracy for random forest is 0.664.

__Question 6:__



The AUC calculated for each classifier is 0.631, 0.706, 0.718, 0.717 and 0.734 respectively for decision tree, Naïve Bayes, Bagging, Boosting and Random Forest respectively.

Question 7:

```
r<=df2
  Classification Method Area Under Curve Accuracy
1         Decision Tree            0.631    0.597
2           Naive Bayes            0.706    0.654
3               Bagging            0.718    0.673
4              Boosting            0.717    0.678
5         Random Forest            0.734    0.664
```

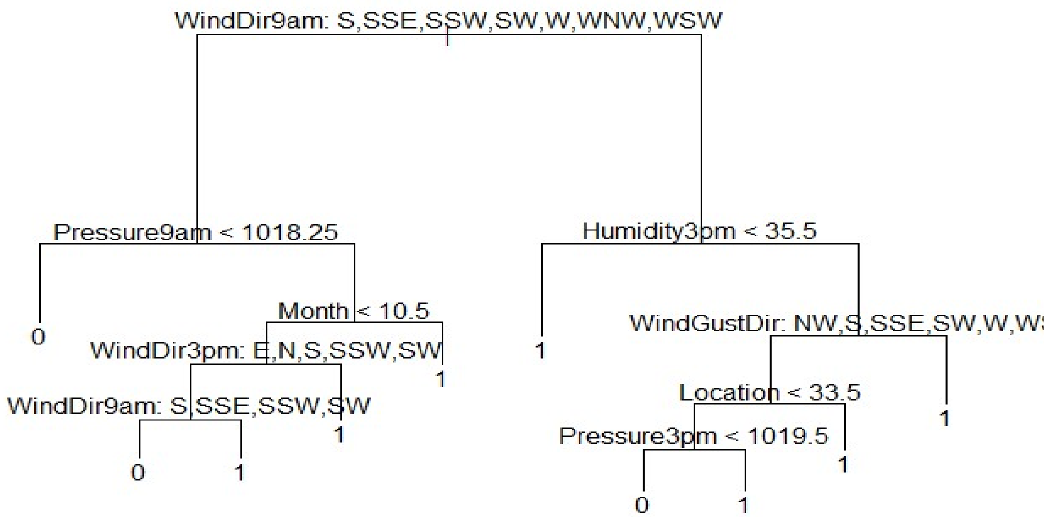
The single best classifier is boosting as it has the higher accuracy and the second highest AUC.

Question 8:

```
#Decision Tree Attribute Importance
r<=print(summary(df.fit))

Classification tree:
tree(formula = WarmerTomorrow ~ ., data = df.train)
Variables actually used in tree construction:
 [1] "WindDir9am"   "Pressure9am"  "WindGustDir"  "Month"        "WindDir3pm"   "WindSpeed3pm" "Humidity3pm"  "MinTemp"      "Location"
[10] "Pressure3pm"
Number of terminal nodes:  18
Residual mean deviance:  1.02 = 989 / 966
Misclassification error rate: 0.263 = 259 / 984
r<=plot(df.fit)
r<=text(df.fit,pretty=0)
r<=cat("\n#Bagging Attribute Importance\n")

#Bagging Attribute Importance
r<=print(df.bag$importance)
         Day  Humidity3pm  Humidity9am     Location      MaxTemp      MinTemp        Month  Pressure3pm  Pressure9am     Rainfall
       2.028        6.334        2.119        1.800        4.870        5.912        2.022        1.976        9.352        1.510
     Temp3pm      Temp9am   WindDir3pm   WindDir9am  WindGustDir WindGustSpeed WindSpeed3pm WindSpeed9am         Year
       2.812        1.723       16.170       19.660       16.289        1.759        1.182        0.866        1.615
r<=cat("\n#Boosting Attribute Importance\n")

#Boosting Attribute Importance
r<=print(df.boost$importance)
         Day  Humidity3pm  Humidity9am     Location      MaxTemp      MinTemp        Month  Pressure3pm  Pressure9am     Rainfall
        3.62         3.90         4.81         2.11         6.29         6.02         2.64         2.61         4.64         1.80
     Temp3pm      Temp9am   WindDir3pm   WindDir9am  WindGustDir WindGustSpeed WindSpeed3pm WindSpeed9am         Year
        2.60         3.30        14.49        16.11        13.34         2.90         2.81         2.95         3.07
r<=cat("\n#Random Forest Attribute Importance\n")

#Random Forest Attribute Importance
r<=print(df.rf$importance)
              MeanDecreaseGini
Day                       17.8
Month                     13.3
Year                      13.3
Location                  12.7
MinTemp                   30.6
MaxTemp                   27.7
Rainfall                  13.4
WindGustDir               51.4
WindGustSpeed             18.6
WindDir9am                56.4
WindDir3pm                52.9
WindSpeed9am              14.6
WindSpeed3pm              17.4
Humidity9am               20.5
Humidity3pm               26.5
Pressure9am               29.9
Pressure3pm               21.4
Temp9am                   23.1
Temp3pm                   25.7
```

For decision tree, the most important attribute is the WindDir9am as it is the root of the tree, the attributes that can be omitted are those that are not used for tree construction. We can see attributes used for construction of tree through the summary. For bagging, the most important attribute is the WindDir9am as it has the highest value of 19.660. The attribute that can be omitted with least effect on performance will be WindSpeed9am with value of 0.866. For boosting, the most important attribute is the WindDir9am as it has the highest value of 16.11. The attribute that can be omitted with least effect on performance will be Rainfall with value of 1.80. For random forest, the most important attribute is the WindDir9am as it has the highest value of 56.4. The attribute that can be omitted with least effect on performance will be Location with value of 12.7.

Question 9:



```
r<=print(t4)
          actual
predicted   0   1
        0  87  65
        1 100 170
r<=print(t1)
                 Actual_Class
Predicted_Class   0   1
              0  89  72
              1  98 163
r<=#accuracy for t4
r<=(87+170)/(87+65+100+170)
[1] 0.609
r<=#accuracy for t1
r<=(89+163)/(89+72+98+163)
[1] 0.597
```

This model of decision tree performs better than the one in question 4 as it has higher accuracy which is 0.609. However, it is still not performing better than the other 4 classification model in question 4. The important factor in my decision is the relationship between the attributes and WarmerTomorrow. The attributes I used are chosen as they have the highest value of importance level as compare to other attributes.

Question 10:

```
r<=df.boost2<-boosting(WarmerTomorrow~.,df.train,mfinal=500,coeflearn = 'Freund')
r<=dfpred.boost2 <- predict.boosting(df.boost2, newdata=df.test)
r<=dfpred.boost2$confusion
               Observed Class
Predicted Class   0   1
              0 111  57
              1  76 178
r<=(111+178)/(111+57+76+178)
[1] 0.685
```

As shown above, the accuracy of this boosting model is higher than those in question 4. I chose this model because boosting method was the best in question 4 and hence I would like to improve it. Basically I created this improved model by increasing the mfinal to 500, the default was 100 and I changed the coeflearn to Freund so that it uses different weight updating coefficient method. I chose to use all attributes as I tried to use only a few highest important attributes to do it but it had even lower accuracy, so I stick to use all attributes after preprocessing the data.

Question 11:

The attributes I used are WindDir9am , Humidity3pm , Pressure9am and WindGustDir as these 4 attributes are the more important attributes based on my improved decision tree in question 9. As for the preprocessing required to implement ANN, basically I set the attributes with string values into numeric with values from 0 to 15. The WarmerTomorrow is also turned into numeric as well because it was set into binary.

```
r<=#confusion matrix
r<=table(observed = df.test$WarmerTomorrow, predicted = df.pred < 0.5)
        predicted
observed FALSE TRUE
       1   105   82
       2    65  170
r<=(105+170)/(105+82+65+170)
[1] 0.652
```

Based on the screenshot above, the accuracy is 0.652. It is only better than decision tree in terms of performance but it's performance is almost similar to the remaining 4 classification models in question 4. It is probably because the dataset is too small for the neural network to perform better, generally the ANN should perform better than all the 5 classification models in question 4.

Appendix:

```
#Jin En Tan

#Student ID:31336574

#FIT3152 assignment 2

getwd()

setwd("c:/Users/Vapor-15 Pro/Downloads")


#creating individual data set

rm(list = ls())

options(digits = 3,prompt ="r<=")

WAUS <- read.csv("WarmerTomorrow2022.csv",stringsAsFactors = TRUE)

WAUS$WarmerTomorrow=factor(WAUS$WarmerTomorrow)

L <- as.data.frame(c(1:49))

set.seed(31336574) # 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


#libraries

library(dplyr)

library(tree)

library(e1071)

library(adabag)

library(randomForest)
```

```
#q1

WAUS %>% count(WarmerTomorrow)   #ignoring those NA

887/(887+1094)*100  #this is percentage of cooler than previous day

1094/(887+1094)*100  #this is percentage of warmer than previous day

#Based on this we know that the proportion of day which is warmer than previous day is higher than that of

#cooler than previous day.

summary(WAUS)

#Based on the descriptions of the predictors, we can see that evaporation, sunshine, cloud9am and cloud3pm contains

#a lot of NA.Hence, I consider to remove these 4 attributes from my analysis as there's too many NA in the data.


#q2

#preprocessing, remove those 4 attributes with a lot of NA

df<-WAUS[,c(1,2,3,4,5,6,7,10,11,12,13,14,15,16,17,18,19,22,23,24)]

#remove remaining rows with NA

df<-na.omit(df)

summary(df)


#q3

set.seed(31336574) #Student ID as random seed

train.row = sample(1:nrow(df), 0.7*nrow(df))

df.train = df[train.row,]

df.test = df[-train.row,]


#q4

#decision tree

df.fit=tree(WarmerTomorrow~.,data=df.train)
```

```r
plot(df.fit)
text(df.fit,pretty=0)

#naive bayes
df.naive <- naiveBayes(WarmerTomorrow~., df.train)

#bagging
df.bag <- bagging(WarmerTomorrow~., df.train)

#boosting
df.boost<-boosting(WarmerTomorrow~.,df.train)

#random forest
df.rf<-randomForest(WarmerTomorrow~.,df.train)

#q5
#decision tree
df.predtree = predict(df.fit, df.test, type = "class")
t1=table(Predicted_Class = df.predtree, Actual_Class = df.test$WarmerTomorrow)
cat("\n#Decision Tree Confusion\n")
print(t1)
(89+163)/(89+72+98+163) #accuracy calculation

#naive bayes
df.predbayes = predict(df.naive, df.test)
t2=table(Predicted_Class = df.predbayes, Actual_Class = df.test$WarmerTomorrow)
cat("\n#NaiveBayes Confusion\n")
print(t2)
(99+177)/(99+58+88+177)
```

```
#bagging

dfpred.bag <- predict.bagging(df.bag, df.test)

dfpred.bag$confusion

(102+182)/(102+53+85+182)


#boosting

dfpred.boost <- predict.boosting(df.boost, newdata=df.test)

dfpred.boost$confusion

(111+175)/(111+60+76+175)


#random forest

dfpredrf <- predict(df.rf, df.test)

t3=table(Predicted_Class = dfpredrf, Actual_Class = df.test$WarmerTomorrow)

cat("\n#Random Forest Confusion\n")

print(t3)

(85+195)/(85+40+102+195)


#q6

#decision tree

# do predictions as probabilities and draw ROC

library(ROCR)

df.pred.tree = predict(df.fit, df.test, type = "vector")

# computing a simple ROC curve (x-axis: fpr, y-axis: tpr)

# labels are actual values, predictors are probability of class

dfDpred <- ROCR::prediction(df.pred.tree[,2], df.test$WarmerTomorrow)

dfDperf <- performance(dfDpred,"tpr","fpr")

treeauc = performance(dfDpred, "auc")

print(as.numeric(treeauc@y.values))
```

```
plot(dfDperf,col="red")
abline(0,1)

#naive bayes
#output as confidence level
dfpred.bayes = predict(df.naive, df.test, type = 'raw')
dfBpred <- ROCR::prediction( dfpred.bayes[,2], df.test$WarmerTomorrow)
dfBperf <- performance(dfBpred,"tpr","fpr")
naiveauc = performance(dfBpred, "auc")
print(as.numeric(naiveauc@y.values))
plot(dfBperf, add=TRUE, col = "green")

#bagging
dfBagpred <- ROCR::prediction( dfpred.bag$prob[,2], df.test$WarmerTomorrow)
dfBagperf <- performance(dfBagpred,"tpr","fpr")
bagauc = performance(dfBagpred, "auc")
print(as.numeric(bagauc@y.values))
plot(dfBagperf, add=TRUE, col = "blue")

#boosting
dfBoostpred <- ROCR::prediction( dfpred.boost$prob[,2], df.test$WarmerTomorrow)
dfBoostperf <- performance(dfBoostpred,"tpr","fpr")
boostauc = performance(dfBoostpred, "auc")
print(as.numeric(boostauc@y.values))
plot(dfBoostperf, add=TRUE, col = "brown")

#random forest
dfpred.rf <- predict(df.rf, df.test, type="prob")
dfFpred <- ROCR::prediction( dfpred.rf[,2], df.test$WarmerTomorrow)
```

```
dfFperf <- performance(dfFpred,"tpr","fpr")

rfauc = performance(dfFpred, "auc")

print(as.numeric(rfauc@y.values))

plot(dfFperf, add=TRUE, col = "darkgreen")

legend("topleft",legend=c("Decision Tree","Naive Bayes","Bagging","Boosting","Random Forest"),col=c("red","green","blue","brown","darkgreen"), lty=1:2, cex=0.8)


#q7

comp_auc=c(0.631,0.706,0.718,0.717,0.734)

comp_acc=c(0.597,0.654,0.673,0.678,0.664)

comp_class=c("Decision Tree","Naive Bayes","Bagging","Boosting","Random Forest")

df2<-data.frame(comp_class,comp_auc,comp_acc)

colnames(df2)<-c("Classification Method","Area Under Curve","Accuracy")

df2

#the single best classifier is boosting as it has the highest accuracy and a relatively higher area under curve as well


#q8

#Attribute importance

cat("\n#Decision Tree Attribute Importance\n")

print(summary(df.fit))

plot(df.fit)

text(df.fit,pretty=0)

cat("\n#Bagging Attribute Importance\n")

print(df.bag$importance)

cat("\n#Boosting Attribute Importance\n")

print(df.boost$importance)

cat("\n#Random Forest Attribute Importance\n")

print(df.rf$importance)
```

#for decision tree, the most important attribute is the WindDir9am as it is the root of the tree,

#the attributes that can be omitted are those that are not used for tree construction.We can see attributes used for

#construction of tree through the summary

#for bagging, the most important attribute is the WindDir9am as it has the highest value of 19.660

#the attribute that can be omitted with least effect on performance will be WindSpeed9am with value of 0.866

#for boosting,the most important attribute is the WindDir9am as it has the highest value of 16.11

#the attribute that can be omitted with least effect on performance will be Rainfall with value of 1.80

#for random forest, the most important attribute is the WindDir9am as it has the highest value of 56.4

#the attribute that can be omitted with least effect on performance will be Location with value of 12.7

```
#q9

#cross validation and pruning

test.fit=cv.tree(df.fit, FUN=prune.misclass)

print(test.fit)

prune.dffit = prune.misclass(df.fit, best=10)

print(summary(prune.dffit))

plot(prune.dffit)

text(prune.dffit, pretty=0)

#test accuracy after pruning

dfp.predict = predict(prune.dffit, df.test, type = "class")

t4=table(predicted = dfp.predict, actual = df.test$WarmerTomorrow)

print(t4)

print(t1)

#accuracy for t4

(87+170)/(87+65+100+170)

#accuracy for t1

(89+163)/(89+72+98+163)
```

```
#this model is better than the previous model in part 4 as the accuracy is higher

#the relationship between attributes and the WarmerTomorrow is important in the decision

#these attributes are chosen because the importance level is among the highest as compare to other attributes


#q10

#boosting

df.boost2<-boosting(WarmerTomorrow~.,df.train,mfinal=500,coeflearn = 'Freund')

dfpred.boost2 <- predict.boosting(df.boost2, newdata=df.test)

dfpred.boost2$confusion

(111+178)/(111+57+76+178)



library(neuralnet)

#q11

#preprocessing

df.train$WindGustDir=recode(df.train$WindGustDir,'E'='0','ENE'='1','ESE'='2','N'='3','NE'='4','NNE'='5','NNW'='6','NW'='7','S'='8','SE'='9','SSE'='10','SSW'='11','SW'='12','W'='13','WNW'='14','WSW'='15')

df.test$WindGustDir=recode(df.test$WindGustDir,'E'='0','ENE'='1','ESE'='2','N'='3','NE'='4','NNE'='5','NNW'='6','NW'='7','S'='8','SE'='9','SSE'='10','SSW'='11','SW'='12','W'='13','WNW'='14','WSW'='15')

df.train$WindDir9am=recode(df.train$WindDir9am,'E'='0','ENE'='1','ESE'='2','N'='3','NE'='4','NNE'='5','NNW'='6','NW'='7','S'='8','SE'='9','SSE'='10','SSW'='11','SW'='12','W'='13','WNW'='14','WSW'='15')

df.test$WindDir9am=recode(df.test$WindDir9am,'E'='0','ENE'='1','ESE'='2','N'='3','NE'='4','NNE'='5','NNW'='6','NW'='7','S'='8','SE'='9','SSE'='10','SSW'='11','SW'='12','W'='13','WNW'='14','WSW'='15')

df.train$WindDir3pm=recode(df.train$WindDir3pm,'E'='0','ENE'='1','ESE'='2','N'='3','NE'='4','NNE'='5','NNW'='6','NW'='7','S'='8','SE'='9','SSE'='10','SSW'='11','SW'='12','W'='13','WNW'='14','WSW'='15')

df.test$WindDir3pm=recode(df.test$WindDir3pm,'E'='0','ENE'='1','ESE'='2','N'='3','NE'='4','NNE'='5','NNW'='6','NW'='7','S'='8','SE'='9','SSE'='10','SSW'='11','SW'='12','W'='13','WNW'='14','WSW'='15')


df.train$WarmerTomorrow <- as.numeric(df.train$WarmerTomorrow)

df.train$WindGustDir <- as.numeric(df.train$WindGustDir)
```

```
df.train$WindDir9am <- as.numeric(df.train$WindDir9am)

df.train$WindDir3pm <- as.numeric(df.train$WindDir3pm)

df.test$WarmerTomorrow <- as.numeric(df.test$WarmerTomorrow)

df.test$WindGustDir <- as.numeric(df.test$WindGustDir)

df.test$WindDir9am <- as.numeric(df.test$WindDir9am)

df.test$WindDir3pm <- as.numeric(df.test$WindDir3pm)



df.nn = neuralnet(WarmerTomorrow == 1 ~ WindDir9am + Humidity3pm + Pressure9am + WindGustDir, df.train,

                hidden=1,linear.output = FALSE)

df.pred = predict(df.nn, df.test)

#confusion matrix

table(observed = df.test$WarmerTomorrow, predicted = df.pred < 0.5)

(105+170)/(105+82+65+170)
```