FIT3152 Assignment
Name: Farayha Zaheer Alam

**1. Explore the data:**
The proportion of data when it is warmer than the previous days:
Warmer: 1095 days  Not Warmer: 905 days
Total proportion: 1095/1095+905= 0.452

Following is the description of independent predictors(mean, median, q1, standard deviation, etc):

**Predictors involving Temperature:**

```
    MinTemp          MaxTemp
Min.   :-3.80   Min.   : 9.30
1st Qu.: 8.50   1st Qu.:17.85
Median :11.90   Median :21.70
Mean   :12.23   Mean   :22.50
3rd Qu.:16.20   3rd Qu.:26.10
Max.   :29.70   Max.   :45.80
NA's   :47      NA's   :17
```

```
    Temp9am          Temp3pm
 Min.   : 3.10   Min.   : 7.40
 1st Qu.:12.70   1st Qu.:16.60
 Median :16.20   Median :20.20
 Mean   :16.57   Mean   :20.92
 3rd Qu.:20.20   3rd Qu.:24.40
 Max.   :38.60   Max.   :44.70
 NA's   :26      NA's   :140
```

MinTemp's Standard Deviation: 5.30
MaxTemp's Standard Deviation: 6.90
Temp3am's Standard Deviation: 5.27
Temp3pm's Standard Deviation: 5.98

**Predictors involving Wind:**

```
 WindGustSpeed     WindSpeed9am     WindSpeed3pm
 Min.   : 9.00   Min.   : 0.00   Min.   : 0.00
 1st Qu.:31.00   1st Qu.: 7.00   1st Qu.:11.00
 Median :39.00   Median :13.00   Median :17.00
 Mean   :39.79   Mean   :13.19   Mean   :17.63
 3rd Qu.:46.00   3rd Qu.:19.00   3rd Qu.:22.00
 Max.   :94.00   Max.   :74.00   Max.   :56.00
 NA's   :491     NA's   :44      NA's   :160
```

WindGustSpeed's Standard Deviation: 12.8
WindSpeed3am's Standard Deviation: 8.46

WindSpeed3pm's Standard Deviation:  8.64

**Predictors involving Humidity:**

| Humidity9am | | Humidity3pm | |
|---|---|---|---|
| Min. | : 5.00 | Min. | : 4.00 |
| 1st Qu. | : 57.00 | 1st Qu. | : 37.00 |
| Median | : 71.00 | Median | : 55.00 |
| Mean | : 69.34 | Mean | : 53.18 |
| 3rd Qu. | : 84.00 | 3rd Qu. | : 69.00 |
| Max. | :100.00 | Max. | :100.00 |
| NA's | :45 | NA's | :156 |

Humidity9am's Standard Deviation: 19.4
Humidity3pm's Standard Deviation: 21.4

**Predictors involving Pressure:**

| Pressure9am | | Pressure3pm | |
|---|---|---|---|
| Min. | : 995.5 | Min. | : 986.1 |
| 1st Qu. | :1013.5 | 1st Qu. | :1011.6 |
| Median | :1018.5 | Median | :1016.4 |
| Mean | :1018.4 | Mean | :1016.4 |
| 3rd Qu. | :1023.4 | 3rd Qu. | :1021.4 |
| Max. | :1038.8 | Max. | :1036.0 |
| NA's | :393 | NA's | :404 |

Pressure9am's Standard Deviation:7.11
Pressure3pm's Standard Deviation: 6.99

**List of other Predictors Along with Descriptors:**

| Rainfall | | Evaporation | | Sunshine | |
|---|---|---|---|---|---|
| Min. | : 0.000 | Min. | : 0.000 | Min. | : 0.000 |
| 1st Qu. | : 0.000 | 1st Qu. | : 2.400 | 1st Qu. | : 4.000 |
| Median | : 0.000 | Median | : 4.200 | Median | : 7.700 |
| Mean | : 2.085 | Mean | : 5.172 | Mean | : 7.101 |
| 3rd Qu. | : 0.800 | 3rd Qu. | : 7.000 | 3rd Qu. | :10.300 |
| Max. | :126.400 | Max. | :32.800 | Max. | :14.300 |
| NA's | :35 | NA's | :1190 | NA's | :1339 |

Rainfall's Standard Deviation: 6.79
Evaporation's Standard Deviation: 4.00
Sunshine's Standard Deviation:3.83

The noteworthy part of the data is the essential predictors mainly rainfall, evaporation, and sunshine, and other notables are wind, humidity, pressure, and temperature whose descriptions are provided above. Moreover, these descriptors give us an idea(based on standard deviation) about how the value of the respective variable varies over the course of the respective dataset extracted.

**The attributes to be omitted:**
Location was initially used to extract the respective observations from 10 different locations and won't facilitate our analysis in any way for the respective tasks so it will be omitted from

the data frame. Moreover, Day, Month, and Year don't play a pivotal role in determining how warm the following day will be in comparison to the previous one so after arranging the current 2000 observation by a concatenated attribute of Day, Month, and Year
they will be removed from our data frame.

**2. Preprocessing**

1. Removal of all the unnecessary attributes (as done above already)
2. Removal of all the attributes with NA>1000
3. Removal of all the observations containing incomplete data (attributes) as they can cause inconsistencies in analyzing process

**3. Partition of Data along with comments added in Appendix**
**4-6. Classification Model, Confusion Matrix, ROC Curve, and AUC (**Code is provided at the Appendix)

**Decision Tree:**
**Confusion Matrix:**

| actual \ predicted | 0 | 1 |
|---|---|---|
| 0 | 64 | 72 |
| 1 | 52 | 138 |

**Accuracy:** ((64+138)/(326))*100= 62.0
**AUC:** 0.651
**The Decision Tree is plotted below:**


**Naïve Bayes:**
**Confusion Matrix:**

| predicted \ actual | 0 | 1 |
|---|---|---|
| 0 | 75 | 51 |
| 1 | 61 | 139 |

**Accuracy:-** ((75+139)/(75+139+51+61))*100=65.6%
**AUC:-** 0.71

**Bagging:**
**Confusion Matrix:**

| Predicted Class | Observed Class 0 | 1 |
|---|---|---|
| 0 | 73 | 43 |
| 1 | 63 | 147 |

**Accuracy:-** ((73+147)/(73+147+63+43))*100= 67.5%
**AUC:-** 0.736

**Boosting**
**Confusion Matrix:**

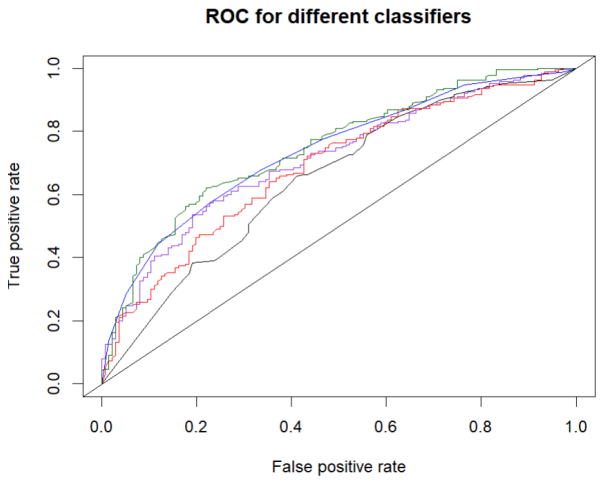
| Predicted Class | Observed Class 0 | 1 |
|---|---|---|
| 0 | 78 | 62 |
| 1 | 58 | 128 |

**Accuracy:-** ((78+128)/(78+128+62+58))*100=63.2%
**AUC:-** 0.687

**Random Forest**
**Confusion Matrix:**

| Predicted_Class | Actual_Class 0 | 1 |
|---|---|---|
| 0 | 61 | 32 |
| 1 | 75 | 158 |

**Accuracy:-** ((61+158)/(61+158+32+75))*100=67.2%
**AUC:-** 0.747




**7. Create a table for comparison:**

| Classifier | Confusion Matrix(in Percentage) | AUC |
|---|---|---|
| **Decision Tree** | 62.0 | 0.651 |
| **Naïve Bayes** | 65.6 | 0.71 |
| **Bagging** | 67.5 | 0.736 |
| **Boosting** | 63.2 | 0.687 |
| **Random Forest** | 67.2 | 0.747 |

If we closely observe the confusion matrix percentage values all of them observe a percentage between 60-70% but Random Forest can be considered the best of all as it has the second-highest percentage along with a 0.3 difference with the highest accurate classifier model (Bagging) and also covers the largest AUC (area under curve). Thus, based on these observations, we can conclude that Random Forest is the most suitable classifier with respect to our dataset.

### 8. Most Important Attributes:

### 1. Decision Tree Attributes:

```
Classification tree:
tree(formula = as.factor(WarmerTomorrow) ~ ., data = WAUS.train)
Variables actually used in tree construction:
[1] "MaxTemp"     "MinTemp"     "Temp3pm"     "WindDir9am"  "WindDir3pm"  "Pressure9am" "Humidity9am"
[8] "WindGustDir"
Number of terminal nodes:  28
Residual mean deviance:  0.7372 = 538.2 / 730
Misclassification error rate: 0.1755 = 133 / 758
```

### 2. Bagging Attributes:

```
  Humidity3pm   Humidity9am       MaxTemp       MinTemp    Pressure3pm    Pressure9am      Rainfall       Temp3pm
    7.0579800     1.5117787     6.8696986    11.0256892     0.7836289     3.5939921     2.3128659     5.6796710
     Temp9am    WindDir3pm    WindDir9am   WindGustDir WindGustSpeed  WindSpeed3pm  WindSpeed9am
    1.9599324    13.8449159    18.0667880    16.1202156     4.1746260     0.3661670     6.6320506
```

### 3. Boosting  Attributes:

```
  Humidity3pm   Humidity9am       MaxTemp       MinTemp    Pressure3pm    Pressure9am      Rainfall       Temp3pm
     1.217578      4.035278     10.840538     12.145289      4.088702      3.306296      1.479210      3.879714
     Temp9am    WindDir3pm    WindDir9am   WindGustDir WindGustSpeed  WindSpeed3pm  WindSpeed9am
     1.444084     12.422865     19.512658     16.327470      3.336691      2.336887      3.626740
```

### 4. Random Forest Attributes:

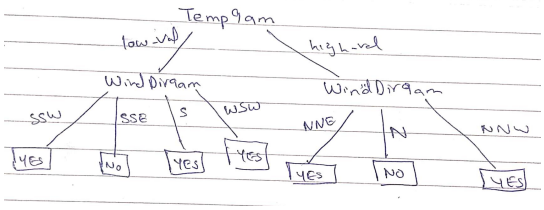
| | MeanDecreaseGini |
|---|---|
| MinTemp | 30.27106 |
| MaxTemp | 30.58905 |
| Rainfall | 10.19674 |
| WindGustDir | 39.71661 |
| WindGustSpeed | 18.23666 |
| WindDir9am | 49.77011 |
| WindDir3pm | 40.33942 |
| WindSpeed9am | 17.74696 |
| WindSpeed3pm | 12.91934 |
| Humidity9am | 18.96924 |
| Humidity3pm | 22.65967 |
| Pressure9am | 19.86570 |
| Pressure3pm | 17.36597 |
| Temp9am | 17.91223 |
| Temp3pm | 26.75834 |

Overall, the four most important variables which are common across all of the classifiers are WindGustDir, WindDir9am, WindDit3pm, and MinTemp. The variable which can be omitted is Rainfall as it shares quite a low value across all the classifiers.

**9. Classifier**

All the attributes with numeric numbers WAUS were initially factorized by assigning them different labels and then the important attributes were determined using information.gain(). Afterward, the top three with the highest gains were selected and used in the building of the decision tree. The main reason for choosing this model over other ones is because of the complexity of the rest of them and doing them by hand can be a tedious task. Moreover, the mathematics involved in the decision tree is easier to implement and is easily readable too. Meanwhile, if we look at the naive bayes model it mostly assigns a zero probability to the ones not available in the data set and since we're already dealing with a small dataset it wasn't ideal to go with such a model which doesn't work well with small chunks of data.

The below snippets shows the final decision tree(working is attached at the bottom of this document):



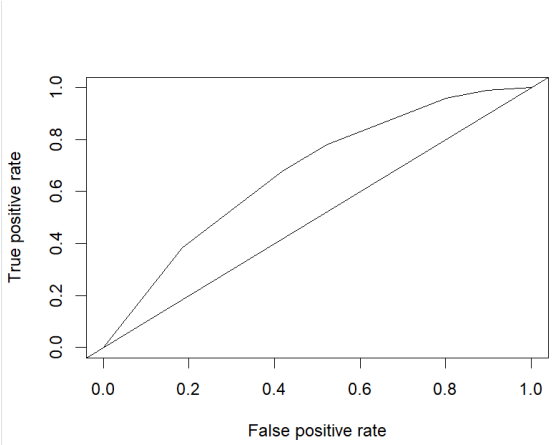

Observing the result above the most important attribute is Temp9am considering the gain value followed by WindDir9am and Pressure9am.

**10. Create the best tree-based classifier**

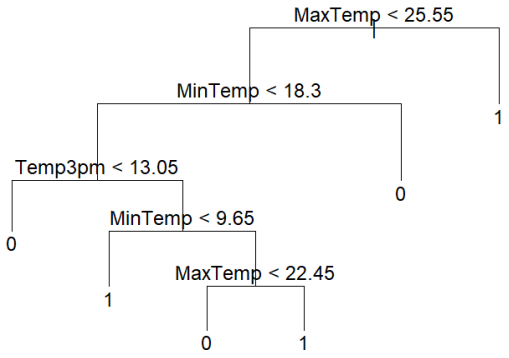
For the creation of the best tree-based classifier, we resort to cross-validation to improve the accuracy of our classifier. The algorithm is only applied to the decision tree-based classifier and then the obtained accuracy is compared with all of the previous models. As it can be seen from the table below, this pruned tree observes a higher accuracy than the decision tree but the percentage is still within the range of other classifiers and not much difference is seen in the AUC value as well.

| Classifier | Confusion Matrix(in Percentage) | AUC |
|---|---|---|
| **Decision Tree** | 62.0 | 0.651 |
| **Naïve Bayes** | 65.6 | 0.71 |
| **Bagging** | 67.5 | 0.736 |
| **Boosting** | 63.2 | 0.687 |
| **Random Forest** | 67.2 | 0.747 |
| **Pruned Tree** | 65.3 | 0.674 |

**ROC (for Pruned Tree):**




**The Pruned Tree is plotted below:**




However if we look at the pruned tree plotted above in comparison to the decision tree it increases the readability of the dataset by reducing the number of pure leaves and makes it easier for readers to reach any sort of conclusion.

**11. Implement an Artificial Neural Network classifier**

To implement the neural network, we use the three most important attributes (MaxTemp, MinTemp, Temp3pm) as shown in the pruned tree above and if we observe the confusion matrix below, it has an accuracy of 69.6% which is extremely good in comparison to the other models. Thus, we can safely conclude that ANN is the most ideal classifier for this respective dataset.

**Confusion Matrix:**

| | predicted | |
|---|---|---|
| observed | 0 | 1 |
| 0 | 74 | 62 |
| 1 | 37 | 153 |

**Accuracy:**(74+153)/(74+153+62+37)*100=69.6%

**Appendix:**
**Imported Libraries:**

```
library(magrittr)
library(dplyr)
library(tree)
library(e1071)
library(ROCR)
library(neuralnet)
library(tree)
library(rminer)
library(randomForest)
library(adabag)
library(FSelector)
```

**R-Code:**

```
rm(list = ls())
WAUS <- read.csv("WarmerTomorrow2022.csv", stringsAsFactors = TRUE)
attach(WAUS)
L <- as.data.frame(c(1:49))
set.seed(31164943) # Your Student ID is the random seed
WAUS<-WAUS %>% filter(WarmerTomorrow!="NA")# removal of all the rows with
WarmerTomorrow variable empty
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. Explore data
str(WAUS)
table(as.factor(WAUS$WarmerTomorrow)) # gives the table where 0 represents not warmer
tomorrow and 1 represents warmer tomorrow and i manually calculate the proportion
through it
summary(WAUS[5:23]) # description of the predictors
# standard deviation values are individually calculated using sd() function
# Arranging by Day, Month, Year, Time (concatenated attribute)
# converting attributes to string for concatenation
WAUS$Day<-as.character(WAUS$Day)
WAUS$Month<-as.character(WAUS$Month)
WAUS$Year<-as.character(WAUS$Year)

# ordering by the attribute Time
WAUS$Time <- paste(WAUS$Day, WAUS$Month, WAUS$Year, sep="/")
WAUS <- WAUS[order(as.Date(WAUS$Time, format="%d/%m/%Y")),]
# removal of the attributes (Day, Month, Year, Location)
WAUS <- WAUS[,5:24]

# 2. Preprocessing:
# Removal of all the attributes with NA>1000
WAUS$Evaporation<-NULL
WAUS$Cloud9am<-NULL
```

```
WAUS$Cloud3pm<-NULL
WAUS$Sunshine<-NULL
# Removal of all the observations containing incomplete data (attributes) as they can cause inconsistencies in analyzing process
WAUS<-na.omit(WAUS)
```

**# 3. Partitioning dataset into 70% for training and 30% for testing dataset**

```
WAUS$WarmerTomorrow <- as.factor(WAUS$WarmerTomorrow)
set.seed(31164943) # Your Student ID is the random seed
train.row = sample(1:nrow(WAUS), 0.7*nrow(WAUS))
WAUS.train = WAUS[train.row,]
WAUS.test = WAUS[-train.row,]
```

**#4-6: Classification Model, Confusion Matrix, ROC Curve, and AUC:**

**# Decision Tree**

```
# # fit decision tree model to predict WarmerTomorrow
summary(WAUS.train)
WAUS.fit=tree(as.factor(WarmerTomorrow) ~ . , data=WAUS.train)
summary(WAUS.fit)
plot(WAUS.fit)
text(WAUS.fit, pretty=0, cex=0.3)
WAUS.predict = predict(WAUS.fit, WAUS.test, type="class")
t1=table(actual=WAUS.test$WarmerTomorrow, predicted=WAUS.predict)
cat("\n#Decision Tree Confusion\n")
```

**# Accuracy of the model**

```
print(t1)
((64+138)/(326))*100
```

**"Model accuracy is 61.96319"**

**# calculate confidence and construct ROC curve**

```
# do predictions as probabilities and draw ROC
WAUS.pred.tree = predict(WAUS.fit, WAUS.test, type = "vector")
# computing a simple ROC curve (x-axis: fpr, y-axis: tpr)
# labels are actual values, predictors are probability of class
WAUSpred <- ROCR::prediction( WAUS.pred.tree[,2], WAUS.test$WarmerTomorrow)
WAUS.perf <- performance(WAUSpred,"tpr","fpr")
plot(WAUS.perf, main="ROC for different classifiers")
abline(0,1)
```

**# calculate AUC**

```
waus.aus.tree <- performance(WAUSpred,"auc");
waus.aus.tree <- as.numeric(waus.aus.tree@y.values)
waus.aus.tree
"value is 0.6506385"
```

**# Naïve Bayes**

```
#fit naive bayes model to predict WarmerTomorrow
WAUS.model=naiveBayes(WarmerTomorrow~., data = WAUS.train)
```

```
# prediction
WAUS.predict = predict(WAUS.model, WAUS.test)
t2=table(predicted = WAUS.predict, actual = WAUS.test$WarmerTomorrow)
 # Accuracy of the model
print(t2)
((75+139)/(75+139+51+61))*100
"Model accuracy is 65.64417"
# calculate confidence and construct ROC curve
WAUS.bayes = predict(WAUS.model, WAUS.test, type = 'raw')
WAUS.bayes.pred <- ROCR::prediction( WAUS.bayes[,2], WAUS.test$WarmerTomorrow)
WAUS.perf <- performance(WAUS.bayes.pred,"tpr","fpr")
plot(WAUS.perf, add=TRUE, col = "blueviolet")
# calculate AUC
waus.aus.bayes <- performance(WAUS.bayes.pred,"auc");
waus.aus.bayes <- as.numeric(waus.aus.bayes@y.values)
waus.aus.bayes
"value is 0.7109133"

# Bagging
WAUS.bag <- bagging(WarmerTomorrow ~. , data = WAUS.train, mfinal=10)
WAUS.pred.bag <- predict.bagging(WAUS.bag, WAUS.test)
cat("\n#Bagging Confusion\n")

# Model Accuracy
print(WAUS.pred.bag$confusion)
((73+147)/(73+147+63+43))*100
"Model accuracy is 67.48466"

# calculate confidence and construct the ROC curve
WAUS.Bagpred <- ROCR::prediction( WAUS.pred.bag$prob[,2],
WAUS.test$WarmerTomorrow)
WAUS.Bagperf <- performance(WAUS.Bagpred,"tpr","fpr")
plot(WAUS.Bagperf, add=TRUE, col = "blue")
# calculate AUC
waus.aus.bag <- performance(WAUS.Bagpred,"auc");
waus.aus.bag <- as.numeric(waus.aus.bag@y.values)
waus.aus.bag
"value is 0.7358746"

# Boosting
# fit boosting model to predict WarmerTomorrow
WAUS.Boost <- boosting(WarmerTomorrow ~. , data = WAUS.train, mfinal=10)
# prediction
WAUS.boost <- predict.boosting(WAUS.Boost, newdata=WAUS.test)
# Boosting
cat("\n#Boosting Confusion\n")
print(WAUS.boost$confusion)
((78+128)/(78+128+62+58))*100
```

"Model accuracy is 63.19018"

**# calculate confidence and construct the ROC curve**

```
WAUSBoostpred <- ROCR::prediction( WAUS.boost$prob[,2],
WAUS.test$WarmerTomorrow)
WAUSBoostperf <- performance(WAUSBoostpred,"tpr","fpr")
plot(WAUSBoostperf, add=TRUE, col = "red")
```

**# calculate AUC**

```
waus.aus.boost <- performance(WAUSBoostpred,"auc");
waus.aus.boost <- as.numeric(waus.aus.boost@y.values)
waus.aus.boost
```

"value is 0.6866873"

**# Random Forest**

```
WAUS.test <- na.omit(WAUS.test)
WAUS.train <- na.omit(WAUS.train)
# fit random forest model to predict WarmerTomorrow
WAUS.rf <- randomForest(WarmerTomorrow ~. , data = WAUS.train, na.action = na.exclude)
# prediction
WAUS.predrf <- predict(WAUS.rf, WAUS.test)

t5=table(Predicted_Class = WAUS.predrf, Actual_Class = WAUS.test$WarmerTomorrow)
cat("\n#Random Forest Confusion\n")
```

**# Model Accuracy**

```
print(t5)
((61+158)/(61+158+32+75))*100
```

"Model accuracy is 67.17791"

**# calculate confidence and construct the ROC curve**

```
WAUSpred.rf <- predict(WAUS.rf, WAUS.test, type="prob")
WAUSpred <- ROCR::prediction( WAUSpred.rf[,2], WAUS.test$WarmerTomorrow)
WAUSperf <- performance(WAUSpred,"tpr","fpr")
plot(WAUSperf, add=TRUE, col = "darkgreen")
```

**# calculate AUC**

```
waus.aus.forest <- performance(WAUSpred,"auc");
waus.aus.forest  <- as.numeric(waus.aus.forest @y.values)
waus.aus.forest
```

"the value is 0.7467105"

**# 8 Attribute importance**

```
cat("\n#Decision Tree Attribute Importance\n")
print(summary(WAUS.fit))
cat("\n#Bagging Attribute Importance\n")
print(WAUS.bag$importance)
cat("\n#Boosting Attribute Importance\n")
print(WAUS.Boost$importance)
cat("\n#Random Forest Attribute Importance\n")
print(WAUS.rf$importance)
```

**# 9 Classifier**

```
WAUS_2=head(WAUS, 10)
# setting the labels of all the attributes which can't be categorized to 'high_val' or 'low_val'
based
# on their comparison with mean value of that attribute overall
for (i in 1:length(WAUS_2)){
  if(i!=4 & i!=6 & i!=7 & i!=16) # this is to ommit the non-numerical attributes in the data frame
    {
  data<-WAUS_2[,i]
  mean_val<-mean(data)
  for(j in 1:nrow(WAUS_2)){
  if (as.numeric(WAUS_2[j,i])> mean_val){
    WAUS_2[j,i]="high_val"
  }
  else{
    WAUS_2[j,i]="low_val"
  }
  }
  }
}
# using information gain for determining the most important variable
information.gain(formula(WAUS_2), WAUS_2)
```

**# 10 Create the best tree-based classifier**

```
#cross validation test
set.seed(100)
cvtest = cv.tree(WAUS.fit, FUN = prune.misclass)
cvtest
#prune using size 6 considering lowest misclassification rate
tree.pruned.Dfit = prune.misclass(WAUS.fit, best = 6)
summary(tree.pruned.Dfit)
plot(tree.pruned.Dfit)
text(tree.pruned.Dfit, pretty = 0)
```

**# check accuracy using the pruned tree**

```
tree.PD.predict = predict(tree.pruned.Dfit, WAUS.test, type = "class")
table(actual = WAUS.test$WarmerTomorrow, predicted = tree.PD.predict)
"Model accuracy is 65.3%"

tree.PD.predict = predict(tree.pruned.Dfit, WAUS.test, type = "vector")
# computing a simple ROC curve (x-axis: fpr, y-axis: tpr)
# labels are actual values, predictors are probability of class
WAUS.pruned.pred <- ROCR:: prediction(tree.PD.predict[,2], WAUS.test$WarmerTomorrow)
WAUS.pruned.perf <- performance(WAUS.pruned.pred,"tpr","fpr")
plot(WAUS.pruned.perf)
abline(0,1)
```

**# calculate auc**

```
waus.aus.pruned.tree <- performance(WAUS.pruned.pred,"auc");
waus.aus.pruned.tree<- as.numeric(waus.aus.pruned.tree@y.values)
waus.aus.pruned.tree
```

**# 11 implement a ANN**

```
# code taken from tutorial 9
WAUS.nn = neuralnet(WarmerTomorrow == 1 ~ MinTemp+MaxTemp+Temp3pm ,
WAUS.train,
                    hidden=3,linear.output = FALSE)
JC.pred = compute(WAUS.nn, WAUS.test[c(1,2,15)])
prob <- JC.pred$net.result
pred <- ifelse(prob>0.5, 1, 0)
#confusion matrix
table(observed = WAUS.test$WarmerTomorrow, predicted = pred)
```

3 ①

Date:

| ID | Temp9am | WinDir9am | Pressure9am | WarmerTomorrow |
|---|---|---|---|---|
| 1 | low-val | SSW | high-val | 1 |
| 2 | low-val | SSE | high-val | 0 |
| 3 | low-val | S | high-val | 1 |
| 4 | high-val | NNE | low-val | 1 |
| 5 | low-val | E | high-val | 0 |
| 6 | low-val | WSW | low-val | 1 |
| 7 | high-val | NNE | low-val | 1 |
| 8 | high-val | W | low-val | 0 |
| 9 | low-val | SSW | high-val | 1 |
| 10 | high-val | NNW | low-val | 1 |

Initial Entropy $= -\frac{7}{10}\log\left(\frac{7}{10}\right) - \frac{3}{10}\log\left(\frac{3}{10}\right) = 0.88312$

① Information-gain : Temp9am

Gain(S, Temp9am) = 0.673

Information-gain : WinDir9am

Gain(S, WinDir9am) = 0.673

Information-gain : Pressure9am

Gain(S, Pressure9am) = 0.423

① $E(S_{low\text{-}val}) = -\frac{4}{6}\log\left(\frac{4}{6}\right) - \frac{2}{6}\log\left(\frac{2}{6}\right) = 0.276$

$E(S_{high\text{-}val}) = -\frac{3}{4}\log\left(\frac{3}{4}\right) - \frac{1}{4}\log\left(\frac{1}{4}\right) = 0.244$

$Gain = 0.88312 - \left(\frac{4}{10} \times 0.244 + \frac{6}{10} \times 0.276\right)$

$Gain = 0.619$

Since we have already know the gain performance value of 0.673 from the respective code.
The information gain in the descendant tree is based on entropy change of each branch. In the case of Temp9am, the highest entropy is

(2) Date:

$E(S_{low-rel}) = 0.276$

Information Gain of WindDir9am

$E(S_{low-rel, SSW}) = -\frac{1}{2}\log(\frac{1}{2}) - \frac{0}{2}\log(\frac{0}{2}) = 0$

$E(S_{low-rel, SSE}) = -\frac{0}{1}\log(\frac{0}{1}) - \frac{1}{1}\log(\frac{1}{1}) = 0$

$E(S_{low-rel, S}) = -\frac{1}{1}\log(\frac{1}{1}) - \frac{0}{1}\log(\frac{0}{1}) = 0$

$E(S_{low-rel, NNE}) = 0$

$E(S_{low-rel, WSW}) = -\frac{1}{1}\log(\frac{1}{1}) - \frac{0}{1}\log(\frac{0}{1}) = 0$

$E(S_{low-rel, W}) = 0$

$E(S_{low-rel, NNW}) = 0$

$Gain(S, low-rel, WindDir9am) = 0.276 - 0 = 0.276$

Information Gain of Pressure9am

$E(S_{low-rel, high-rel}) = -\frac{3}{5}\log(\frac{3}{5}) - \frac{2}{5}\log(\frac{2}{5}) = 0.292$

$E(S_{low-rel, low-rel}) = -\frac{1}{1}\log(\frac{1}{1}) - \frac{0}{1}\log(\frac{1}{1}) = 0$

$Gain(S, low-rel, Pressure9am) = 0.276 - (\frac{5}{6} \times 0.292)$

$= 0.0324$

Since the Gain of WindDir9am is higher so it will be the attribute selected

(3)

Date:

Now we will consider another subset $S_{high-wind}$

Information Gain of WindDir9am :-

$E(S_{high-wind, SSW}) = 0$

$E(S_{high-wind, SSE}) = 0$

$E(S_{high-wind, S}) = 0$

$E(S_{high-wind, NNE}) = -\frac{2}{2}\log\left(\frac{2}{2}\right) - \frac{0}{2}\log\left(\frac{0}{2}\right) = 0$

$E(S_{high-wind, WSW}) = 0$

$E(S_{high-wind, W}) = -\frac{0}{1}\log\left(\frac{0}{1}\right) - \frac{1}{1}\log\left(\frac{1}{1}\right) = 0$

$E(S_{high-wind, NNW}) = -\frac{1}{1}\log\left(\frac{1}{1}\right) - \frac{0}{1}\log\left(\frac{0}{1}\right) = 0$

$Gain(S_{high-wind}, WindDir9am) = 0.244 - 0 = 0.244$

Information Gain of Pressure9am :-

$E(S_{low-wind, high-wind}) =$

$E(S_{high-wind, low-wind}) =$

$Gain(S, low-val, P$

Information Gain of Pressure9am :-

$E(S_{high-wind, low-wind}) = -\frac{3}{4}\log\left(\frac{3}{4}\right) - \frac{1}{4}\log\left(\frac{1}{4}\right) = 0.244$

$E(S_{high-wind, high-wind}) = 0$

$Gain(S, high-val, Pressure9am) = 0$

$= 0.244 - 0.244 = 0$

(4)

Date: ____ ____ ____

Since, the Gain value of WindPar9am is higher so it will be selected.

Thus, it will be Temp9am at the root node with WindPar9am on the descendant trees. (Displayed in the decision tree)