FIT3152 Assignment 2 Report

Jason Nathaniel Setiawan

The data for this assignment is taken from "WarmerTomorrow2022.csv". The libraries that are used are tree, e1071, ROCR, randomForest, adabag, rprart, car, dplyr, and neuralnet.

Part 1

The proportion of days when it is warmer than the previous day is 0.521 while the proportion of days when it is colder than the previous day is 0.479. I use the summary function to get the description of the real value variables. From there we can see some interesting thing about the data including:

- The MinTemp data has a range from -2.3 until 28 degree Celsius
- The MaxTemp data has a range from 12 until 43.5 degree Celsius
- The mean rainfall is 1.748 but the maximum value is 62 which indicates that there are days where it rains heavily
- The Cloud9am and Cloud3pm has almost similar values in minimal, 1st quantile, mean, 3rd quantile, and maximum. This means that in most days the fractions of sky obscured by cloud in 9am and 3pm is not much different.

If I were to consider any attributes to omit from the data, it would be the attribute Year because I think the value of this attribute does not make any sense on how it would predict if tomorrow is warmer or not.

Part 2

The data needs some additional changes to make it suitable to building the models. Here are the data pre-processing that was taken:

- Changing data that has character type and the target value into factors
- Omitting the data observation that contains NA value

There were 353 observations left in the data after pre-processing.

Part 3

In this part, use the code that was given in the assignment brief to divide the data into the training and testing dataset.

Part 4

Implement decision tree, Naïve Bayes, bagging, boosting, and random forest classification model and set the seed every time building the model to get a consistent output when evaluating the accuracy and area under the curve later.

Part 5

Decision Tree

	Predicted 0	Predicted 1
Actual 0	32	18
Actual 1	13	43

Accuracy = 0.707

Naïve Bayes

	Predicted 0	Predicted 1
Actual 0	27	23
Actual 1	7	49

Accuracy = 0.717

Bagging

	Predicted 0	Predicted 1
Actual 0	31	19
Actual 1	16	40

Accuracy = 0.670

Boosting

	Predicted 0	Predicted 1
Actual 0	35	15
Actual 1	16	40

Accuracy = 0.707

Random Forest

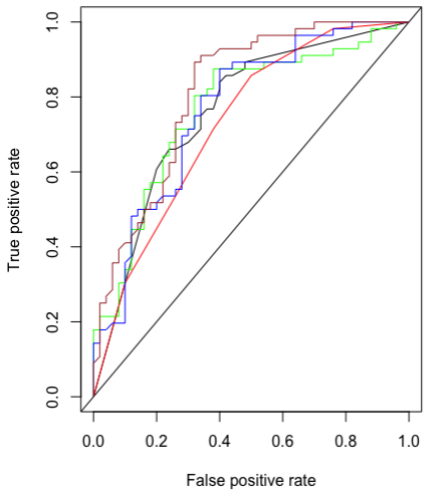
	Predicted 0	Predicted 1
Actual 0	33	17
Actual 1	6	50

Accuracy = 0.783

The above tables are the confusion matrix and accuracy of each of the classification model. The value “0” indicates it’s not warmer tomorrow and the value “1” indicates its warmer tomorrow. The model with the highest accuracy is random forest with 0.783.

Part 6

Figure 1. ROC curve for each classifier



Model	AUC
Decision Tree	0.761
Naïve Bayes	0.768
Bagging	0.728
Boosting	0.768
Random Forest	0.814

In this section, ROC curve was calculated for each classifier and the plot can be seen in figure 1. Each different line colour represents different classification model, black represents decision tree, green represents Naïve Bayes, red represents bagging, blue represents boosting, and brown represents random forest. The Area under the curve was also calculated and is presented in the table on the right above. The model with the highest AUC is random forest model.

Part 7

Model	Accuracy	AUC
Decision Tree	0.707	0.761
Naïve Bayes	0.717	0.768
Bagging	0.670	0.728
Boosting	0.707	0.768
Random Forest	0.783	0.814

The table above is to compare the accuracy and area under the curve from all classification models. Looking at the result, we can see that the random forest performs the best in terms of accuracy and area under the curve while bagging is the worst model in both measurements as well.

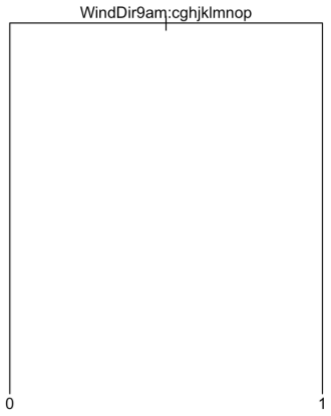
Part 8

From the R output, we can see the variable importance of each model. The most important variables in decision tree is WindDir9am as it is the attribute to make the first branch of the tree. The most important variable in bagging and random forest model is also WindDir9am while the most important variable in boosting model is WindGustDir. Overall, the most important variable is WinDir9am.

The variables that can be omitted from the data because it has a little effect in the performance are location, rainfall, and year as it constantly scores on of the lowest value in the importance level in multiple models.

Part 9

I choose to create a decision tree because it is simple enough for a person to be able to compute it by hand. Based on the cross-validation test, the size of 2 is chosen for tree as it is as simple as it gets when it comes to decision tree and still has a low misclassification rate.



Above is the diagram of the pruned decision tree. This tree model has the accuracy of 0.679 and area under the curve of 0.672 which still performs under the models that are created in part 4. However, there isn't a big difference compared to the unpruned tree and because the main purpose of the tree is the simplicity where we would be able to compute by hand, this accuracy is acceptable. The variable that is used in this decision tree is WindDir9am and it make sense as it is also the most important variable in the decision tree. The person can determine whether tomorrow is warmer (output 1) or not (output 0)by the direction of the wind at 9 am.

Part 10

To create the best tree-based classifier, the random forest is chosen to be the model that is improved as it is the best model in part 7. After many trials and error trying to improve the model, the best classifier can was obtained with an accuracy of 0.820 and area under the curve of 0.870 which is the best out of all the classifier that has been created so far as can be seen in the table below.

Model	Accuracy	AUC
Decision Tree	0.707	0.761
Naïve Bayes	0.717	0.768
Bagging	0.670	0.728
Boosting	0.707	0.768
Random Forest	0.783	0.814
Improved random Forest	0.820	0.870

The model was created by removing the column Cloud3pm and Cloud9am from the dataset before removing the NA values. This column has a lot of NA values in it which leads to more observations being remove in WAUS dataset. Because we have removed the 2 columns before removing the NA values, we ended up having more observations in our data to be able to train the random forest model better.

The attributes that were chosen to build the model is still the same as the previous random forest model (all attributes) because there are no improvements in the accuracy when I tried to modify the parameters.

Part 11

In this section, an artificial neural network is implemented. The requirement for artificial neural network is that this model only accept numerical value. Therefore, there are some data pre-processing that needs to be done which is to turn the target variable WarmerTomorrow into a numerical value.

When building the model itself, the attributes that were chosen was Sunshine, MaxTemp, Humidity9am, and Humidity3pm as these attributes are numerical value and gives the best result from modifying the attributes that are used. The model has an accuracy of 0.704 which is not the best but still comparable with the other models such as decision tree, Naïve Bayes, bagging, and boosting.

Part 12

Brief Report (This document)

Appendix

```
# Assignment 2

# install and import the required packages
install.packages("tree")
library(tree)
install.packages("e1071")
library(e1071)
install.packages(("ROCR"))
library(ROCR)
install.packages("randomForest")
library(randomForest)
install.packages("adabag")
library(adabag)
install.packages("rpart")
library(rpart)
install.packages("car")
library(car)
library(dplyr)

detach("package:neuralnet", unload = TRUE)

# creating the data set
rm(list = ls())
WAUS <- read.csv("WarmerTomorrow2022.csv")
L <- as.data.frame(c(1:49))
set.seed(30899559) # 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


# ---------------------- Question 1 -------------------------

# get the number of days when it is warmer than the previous day
warmerCount <- nrow(WAUS[WAUS$WarmerTomorrow == 1,])
# proportion of days when it is warmer than the previous day
propWarmer <- warmerCount/nrow(WAUS)
# get the number of days when it is colder than the previous day
colderCount <- nrow(WAUS[WAUS$WarmerTomorrow == 0,])
# proportion of days when it is colder than the previous day
propColder <- colderCount/nrow(WAUS)

# see the results
propWarmer
propColder
```

```
# description of the predictors
summary(WAUS)


# ---------------------- Question 2 -------------------------

# make the attributes as factor
WAUS <- as.data.frame(unclass(WAUS), stringsAsFactors = TRUE)
WAUS$WarmerTomorrow <- factor(WAUS$WarmerTomorrow)

# remove NA values from the data
WAUS <- WAUS[complete.cases(WAUS),]


# ---------------------- Question 3 -------------------------

# partition data into training and testing data
set.seed(30899559) #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,]

# ---------------------- Question 4 -------------------------

# fit decision tree model
set.seed(30899559)
WAUS.tree <- tree(WarmerTomorrow ~ . , data = WAUS.train)
plot(WAUS.tree)
text(WAUS.tree)

#fit Naive Bayes model
set.seed(30899559)
WAUS.nb <- naiveBayes(WarmerTomorrow ~ . , data = WAUS.train)

# fit a bagging model
set.seed(30899559)
WAUS.bag <- bagging(WarmerTomorrow ~ . , data = WAUS.train, mfinal = 5)

# fit a boosting model
set.seed(30899559)
WAUS.boost <- boosting(WarmerTomorrow ~ . , data = WAUS.train, mfinal = 10)

# fit a random forest model
set.seed(30899559)
WAUS.rf <- randomForest(WarmerTomorrow ~ ., data = WAUS.train)
```

```
# ---------------------- Question 5 -------------------------

# make the prediction using decision tree model
WAUS.tree.pred <- predict(WAUS.tree, WAUS.test, type = "class")
# create confusion matrix and calculate accuracy
WAUS.tree.cf <- table(predicted = WAUS.tree.pred, actual = WAUS.test$WarmerTomorrow)
WAUS.tree.cf
WAUS.tree.acc <- (WAUS.tree.cf[1,1] + WAUS.tree.cf[2,2]) /sum(WAUS.tree.cf)

# make the prediction for Naive Bayes model
WAUS.nb.pred <- predict(WAUS.nb, WAUS.test, type = "raw")
WAUS.nb.cpred <- predict(WAUS.nb, WAUS.test, type = "class")
# create confusion matrix and calculate accuracy
WAUS.nb.cf <-table(predicted = WAUS.nb.cpred, actual = WAUS.test$WarmerTomorrow)
WAUS.nb.cf
WAUS.nb.acc <- (WAUS.nb.cf[1,1] + WAUS.nb.cf[2,2]) /sum(WAUS.nb.cf)

# make the prediction for bagging model
WAUS.bag.pred <- predict(WAUS.bag, WAUS.test, type = "raw")
# create confusion matrix and calculate accuracy
WAUS.bag.cf <-WAUS.bag.pred$confusion
WAUS.bag.cf
WAUS.bag.acc <- (WAUS.bag.cf[1,1] + WAUS.bag.cf[2,2]) /sum(WAUS.bag.cf)

# make the prediction for boosting model
WAUS.boost.pred <- predict(WAUS.boost, WAUS.test, type = "raw")
# create confusion matrix and calculate accuracy
WAUS.boost.cf <- WAUS.boost.pred$confusion
WAUS.boost.cf
WAUS.boost.acc <- (WAUS.boost.cf[1,1] + WAUS.boost.cf[2,2]) /sum(WAUS.boost.cf)

# make the prediction for random forest model
WAUS.rf.pred <- predict(WAUS.rf, WAUS.test)
# create confusion matrix and calculate accuracy
WAUS.rf.cf <-table(predicted = WAUS.rf.pred, actual = WAUS.test$WarmerTomorrow)
WAUS.rf.cf
WAUS.rf.acc <- (WAUS.rf.cf[1,1] + WAUS.rf.cf[2,2]) /sum(WAUS.rf.cf)


# ---------------------- Question 6 -------------------------

# ROC curve for decision tree model
WAUS.tree.pred.vec <- predict(WAUS.tree, WAUS.test, type = "vector")
WAUSdPred <- prediction(WAUS.tree.pred.vec[,2], WAUS.test$WarmerTomorrow)
WAUSdPerf <- performance(WAUSdPred, "tpr", "fpr")
```

```
plot(WAUSdPerf)
abline(0,1)
# calculate the AUC
WAUS.tree.auc <- as.numeric(performance(WAUSdPred,"auc")@y.values)

# ROC curve for Naive Bayes model
WAUSdPred <- prediction(WAUS.nb.pred[,2], WAUS.test$WarmerTomorrow)
WAUSdPerf <- performance(WAUSdPred, "tpr", "fpr")
plot(WAUSdPerf, add = TRUE, col = "green")
# calculate the AUC
WAUS.nb.auc <- as.numeric(performance(WAUSdPred,"auc")@y.values)

# ROC curve for bagging model
WAUSdPred <- prediction(WAUS.bag.pred$prob[,2], WAUS.test$WarmerTomorrow)
WAUSdPerf <- performance(WAUSdPred, "tpr", "fpr")
plot(WAUSdPerf, add = TRUE, col = "red")
# calculate the AUC
WAUS.bag.auc <- as.numeric(performance(WAUSdPred,"auc")@y.values)

# ROC curve for boosting model
WAUSdPred <- prediction(WAUS.boost.pred$prob[,2], WAUS.test$WarmerTomorrow)
WAUSdPerf <- performance(WAUSdPred, "tpr", "fpr")
plot(WAUSdPerf, add = TRUE, col = "blue")
# calculate the AUC
WAUS.boost.auc <- as.numeric(performance(WAUSdPred,"auc")@y.values)

# ROC curve for random forest model
WAUSpred.rf <- predict(WAUS.rf, WAUS.test, type = "prob")
WAUSdPred <- prediction(WAUSpred.rf[,2], WAUS.test$WarmerTomorrow)
WAUSdPerf <- performance(WAUSdPred, "tpr", "fpr")
plot(WAUSdPerf, add = TRUE, col = "brown")
# calculate the AUC
WAUS.rf.auc <- as.numeric(performance(WAUSdPred,"auc")@y.values)


# ---------------------- Question 7 -------------------------

# create a table to collect the accutacy and area under the curve from all models and
combine it
model <- c("Decision tree", "Naive Bayes", "Bagging", "Boosting", "Random Forest")
accuracy <- c(WAUS.tree.acc, WAUS.nb.acc, WAUS.bag.acc, WAUS.boost.acc, WAUS.rf.acc)
auc <- c(WAUS.tree.auc, WAUS.nb.auc, WAUS.bag.auc, WAUS.boost.auc, WAUS.rf.auc)
results <- data.frame(accuracy, auc)
rownames(results) <- model


# ---------------------- Question 8 -------------------------
```

```
# get variable importance, in decision treel the most important is WindDir9am
print(summary(WAUS.tree))

# get variable importance, in bagging the most important is WindDir9am omit rainfall
print(WAUS.bag$importance)

# get variable importance, in boosting the most important is WindGustDir  and omit rainfall
print(WAUS.boost$importance)

# get variable importance, in random forest the most important is WindDir9am and omit
Location
print(WAUS.rf$importance)
varImpPlot(WAUS.rf)


# ---------------------- Question 9 ------------------------

# perfrom cross validation test
cvtest<- cv.tree(WAUS.tree, FUN = prune.misclass)
cvtest

# prune using size 2 for simplicity
prune.zfit <- prune.misclass(WAUS.tree, best=2)
print(summary(prune.zfit))
plot(prune.zfit)
text(prune.zfit)

# do prediction and get make confusion matrix to get the accuracy
prune.zfit.pred <- predict(prune.zfit, WAUS.test, type = "class")
prune.zfit.cf <- table(predicted = prune.zfit.pred, actual = WAUS.test$WarmerTomorrow)
prune.zfit.acc <- (prune.zfit.cf[1,1] + prune.zfit.cf[2,2]) /sum(prune.zfit.cf)
prune.zfit.acc

# calculate the ROC curve
WAUS.pruned.tree.pred.vec <- predict(prune.zfit, WAUS.test, type = "vector")
WAUSdPred <- prediction(WAUS.pruned.tree.pred.vec[,2], WAUS.test$WarmerTomorrow)
WAUSdPerf <- performance(WAUSdPred, "tpr", "fpr")
# calculate the AUC
WAUS.pruned.tree.auc <- as.numeric(performance(WAUSdPred,"auc")@y.values)
WAUS.pruned.tree.auc


# ---------------------- Question 10  ------------------------

# read the data again as in the first part of the script
w <- read.csv("WarmerTomorrow2022.csv")
```

```
L <- as.data.frame(c(1:49))
set.seed(30899559) # Your Student ID is the random seed
L <- L[sample(nrow(L), 10, replace = FALSE),] # sample 10 locations
w <- w[(w$Location %in% L),]
w <- w[sample(nrow(w), 2000, replace = FALSE),] # sample 2000 rows

# remove the Cloud3pm and Cloud 9am column
w <- subset(w, select= -c(Cloud3pm, Cloud9am))
# make the attributes as factor
w <- as.data.frame(unclass(w), stringsAsFactors = TRUE)
w$WarmerTomorrow <- factor(w$WarmerTomorrow)
# remove NA values from the data
w <- w[complete.cases(w),]

# partition data into training and testing data
set.seed(30899559) #Student ID as random seed
train.row = sample(1:nrow(w), 0.7*nrow(w))
w.train = w[train.row,]
w.test = w[-train.row,]

# fit an improved random forest model
set.seed(30899559)
w.rf <- randomForest(WarmerTomorrow ~ ., data = w.train, ntree = 500)

# make the prediction for the impoved random forest model
w.rf.pred <- predict(w.rf, w.test)
# create confusion matrix and calculate accuracy
w.rf.cf <-table(predicted = w.rf.pred, actual = w.test$WarmerTomorrow)
w.rf.cf
w.rf.acc <- (w.rf.cf[1,1] + w.rf.cf[2,2]) /sum(w.rf.cf)
w.rf.acc

# ROC curve for improved random forest model
wpred.rf <- predict(w.rf, w.test, type = "prob")
wPred <- prediction(wpred.rf[,2], w.test$WarmerTomorrow)
wPerf <- performance(wPred, "tpr", "fpr")
# calculate the AUC
w.rf.auc <- as.numeric(performance(wPred,"auc")@y.values)
w.rf.auc

# ---------------------- Question 11  ------------------------

library(neuralnet)

# get the data and turn the target variable as numeric
N <- WAUS
N$WarmerTomorrow <- as.numeric(N$WarmerTomorrow)
```

```
# split into training and testing dataset
train.row = sample(1:nrow(N), 0.8*nrow(N))
N.train = N[train.row,]
N.test = N[-train.row,]

# craete the neural network model with selected attributes
set.seed(30899559)
N.net <- neuralnet(WarmerTomorrow == 1 ~ Sunshine +  MaxTemp + Humidity9am +
Humidity3pm, N.train, hidden=3,linear.output = FALSE)

# do the test and get the result of the model
N.pred = compute(N.net, N.test[c("Sunshine", "MaxTemp", "Humidity9am",
"Humidity3pm")])
prob <- N.pred$net.result
pred <- ifelse(prob>0.5, 1, 0)
# create confusion matrix
N.cf <- table(observed = N.test$WarmerTomorrow, predicted = pred)
N.cf
# compute accuracy
N.cf.acc <- (N.cf[1,1] + N.cf[2,2]) /sum(N.cf)
N.cf.acc
```