FIT3152 Data Analytics
Assignment 2

```
      Day            Month           Year          Location       MinTemp          MaxTemp          Rainfall       Evaporation        Sunshine
10      :  82   11      : 191   2014   :230   Min.   :10.00   Min.   :-7.90   Min.   : 4.10   Min.   : 0.000   Min.   : 0.000   Min.   : 0.00
14      :  75   9       : 182   2018   :220   1st Qu.:17.00   1st Qu.: 7.60   1st Qu.:19.20   1st Qu.: 0.000   1st Qu.: 3.425   1st Qu.: 5.90
15      :  75   3       : 174   2015   :197   Median :37.00   Median :12.50   Median :23.60   Median : 0.000   Median : 5.500   Median : 9.10
13      :  74   10      : 170   2019   :195   Mean   :30.66   Mean   :12.14   Mean   :23.98   Mean   : 2.745   Mean   : 6.193   Mean   : 7.97
12      :  71   12      : 169   2009   :193   3rd Qu.:41.00   3rd Qu.:17.20   3rd Qu.:28.20   3rd Qu.: 0.400   3rd Qu.: 8.000   3rd Qu.:10.70
(Other):1573   (Other):1072   (Other):921   Max.   :47.00   Max.   :27.40   Max.   :43.70   Max.   :167.000   Max.   :38.800   Max.   :13.40
NA's    :  22   NA's    :  14   NA's    : 16                 NA's   :66      NA's   :54      NA's   :99        NA's   :1370     NA's   :1499
 WindGustDir   WindGustSpeed     WindDir9am     WindDir3pm    WindSpeed9am    WindSpeed3pm    Humidity9am      Humidity3pm      Pressure9am
S       : 181   Min.   :  7.0   NW     : 199   NW     : 167   Min.   : 0.00   Min.   : 0.00   Min.   : 10.00   Min.   :  6.00   Min.   : 996.9
SSE     : 175   1st Qu.: 31.0   WNW    : 156   S      : 159   1st Qu.: 9.00   1st Qu.:13.00   1st Qu.: 57.00   1st Qu.: 35.00   1st Qu.:1013.4
NW      : 159   Median : 39.0   S      : 149   SE     : 154   Median :13.00   Median :19.00   Median : 68.00   Median : 50.00   Median :1018.0
WNW     : 153   Mean   : 41.2   NNE    : 133   SSE    : 154   Mean   :14.89   Mean   :20.17   Mean   : 67.82   Mean   : 50.08   Mean   :1018.2
NE      : 134   3rd Qu.: 48.0   SE     : 125   NE     : 146   3rd Qu.:20.00   3rd Qu.:26.00   3rd Qu.: 80.00   3rd Qu.: 65.00   3rd Qu.:1022.7
(Other):1114   Max.   :102.0   (Other):1079   (Other):1143   Max.   :63.00   Max.   :61.00   Max.   :100.00   Max.   :100.00   Max.   :1036.6
NA's    :  56   NA's   :57      NA's   : 131   NA's   :  49   NA's   :41      NA's   :38      NA's   :77       NA's   :119      NA's   :234
 Pressure3pm       Temp9am         Temp3pm       RainToday
Min.   : 996.5   Min.   :-1.80   Min.   : 3.70   No  :1472
1st Qu.:1010.6   1st Qu.:13.40   1st Qu.:17.80   Yes : 399
Median :1015.4   Median :18.00   Median :21.80   NA's: 101
Mean   :1015.4   Mean   :17.67   Mean   :22.17
3rd Qu.:1020.1   3rd Qu.:22.00   3rd Qu.:26.00
Max.   :1033.7   Max.   :33.20   Max.   :42.10
NA's   :236      NA's   :51      NA's   :88
```

Diagram 1: Summary After Modifying Class of Variables

Question 1

For CloudTomorrow, there are N/As, 1 indicating it will be cloudy tomorrow, and 0 indicating it will be a clear day (not cloudy) tomorrow. Since there are 1654 rows are 0, 318 rows are 1 and 28 rows are N/A in CloudTomorrow, we will get 0.19 for the proportion of cloudy days to clear days. For each independent numeric variable, the minimum, first quartile, median, mean, third quartile, maximum and number of N/As will be show. For independent variables that are factors, the frequency of each factor levels and number of N/As will be shown. Note that there are N/As in every variable except Location. Evaporation and Sunshine have a huge amount of N/A values. We can refer to Diagram 1 that shows the summary of this dataset.

Question 2

Before getting the summary in Question 1, observations with N/A for CloudTomorrow are all removed. Besides that, variables that are in class character such as WindGustDir, WindDir9am, WindDir3pm and RainToday are converted to class factor. Day, Month and Year are also converted to factors as they do not make sense with a decimal number. Considering there are N/As in this dataset, N/A is filled in with the mean of that attribute if it is a numeric attribute whereas N/A is filled with the mode of the attribute if the attribute is of class factor. N/A values are replaced with mean and mode because some classifiers can handle N/A values but some not. I wish to compare all of the classifiers so I will need to replace all of the N/A.

Question 5

Note that 0 indicates “not cloudy tomorrow” and 1 indicates “cloudy tomorrow”. The confusion matrix and statistics of each model (Decision Tree, Naïve Bayes, Bagging, Boosting and Random Forest) is in Diagram 2 to Diagram 6 respectively. Accuracy of Decision Tree is 0.8277, Naïve Bayes is 0.8345, Bagging is 0.8598, Boosting is 0.8598 and Random Forest is 0.853.

```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 463  61
         1  41  27

               Accuracy : 0.8277
                 95% CI : (0.7948, 0.8573)
    No Information Rate : 0.8514
    P-Value [Acc > NIR] : 0.95082

                  Kappa : 0.2488

 Mcnemar's Test P-Value : 0.05993

            Sensitivity : 0.9187
            Specificity : 0.3068
         Pos Pred Value : 0.8836
         Neg Pred Value : 0.3971
             Prevalence : 0.8514
         Detection Rate : 0.7821
   Detection Prevalence : 0.8851
      Balanced Accuracy : 0.6127

       'Positive' Class : 0
```

Diagram 2: Statistics of Decision Tree

```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 456  50
         1  48  38

               Accuracy : 0.8345
                 95% CI : (0.802, 0.8635)
    No Information Rate : 0.8514
    P-Value [Acc > NIR] : 0.8864

                  Kappa : 0.3398

 Mcnemar's Test P-Value : 0.9195

            Sensitivity : 0.9048
            Specificity : 0.4318
         Pos Pred Value : 0.9012
         Neg Pred Value : 0.4419
             Prevalence : 0.8514
         Detection Rate : 0.7703
   Detection Prevalence : 0.8547
      Balanced Accuracy : 0.6683

       'Positive' Class : 0
```

Diagram 3: Statistics of Naïve Bayes

```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 488  67
         1  16  21

               Accuracy : 0.8598
                 95% CI : (0.8292, 0.8868)
    No Information Rate : 0.8514
    P-Value [Acc > NIR] : 0.3051

                  Kappa : 0.2719

 Mcnemar's Test P-Value : 4.06e-08

            Sensitivity : 0.9683
            Specificity : 0.2386
         Pos Pred Value : 0.8793
         Neg Pred Value : 0.5676
             Prevalence : 0.8514
         Detection Rate : 0.8243
   Detection Prevalence : 0.9375
      Balanced Accuracy : 0.6034

       'Positive' Class : 0
```

Diagram 4: Statistics of Bagging

```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 490  69
         1  14  19

               Accuracy : 0.8598
                 95% CI : (0.8292, 0.8868)
    No Information Rate : 0.8514
    P-Value [Acc > NIR] : 0.3051

                  Kappa : 0.2535

 Mcnemar's Test P-Value : 3.08e-09

            Sensitivity : 0.9722
            Specificity : 0.2159
         Pos Pred Value : 0.8766
         Neg Pred Value : 0.5758
             Prevalence : 0.8514
         Detection Rate : 0.8277
   Detection Prevalence : 0.9443
      Balanced Accuracy : 0.5941

       'Positive' Class : 0
```

Diagram 5: Statistics of Boosting

```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 478  61
         1  26  27

               Accuracy : 0.853
                 95% CI : (0.8219, 0.8806)
    No Information Rate : 0.8514
    P-Value [Acc > NIR] : 0.4823562

                  Kappa : 0.3054

 Mcnemar's Test P-Value : 0.0002672

            Sensitivity : 0.9484
            Specificity : 0.3068
         Pos Pred Value : 0.8868
         Neg Pred Value : 0.5094
             Prevalence : 0.8514
         Detection Rate : 0.8074
   Detection Prevalence : 0.9105
      Balanced Accuracy : 0.6276

       'Positive' Class : 0
```

Diagram 6: Statistics of Random Forest

Question 6

The confidence of predicting 'cloudy tomorrow' for each case is calculated and an ROC curve for each classifier is constructed and shown in Diagram 7. The AUC for Decision Tree is 0.6932044, Naïve Bayes is 0.7305871, Bagging is 0.8411346, Boosting is 0.8374594 and Random Forest is 0.838598. As we can see, ensemble methods have larger AUC which means better discrimination.

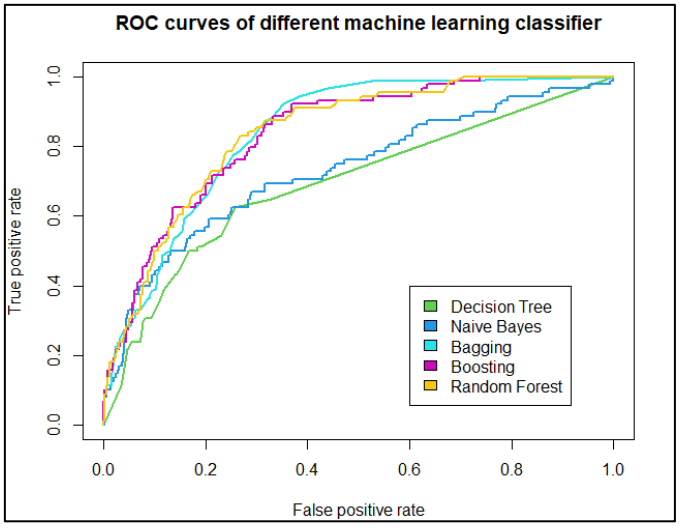

Diagram 7: ROC curves for each classifier

Question 7

From Table 1, we can observe that Bagging has the highest accuracy and AUC; Boosting has the highest accuracy and sensitivity and Naïve Bayes has the highest specificity. Total is a column that sums up all of the measurements to see which of them gives the best performance overall. Although Random Forest does not obtain any highest measurements, it has the best performance considering all measurements. As this is the case, I think that there is no such single best classifier as classifiers such as Naïve Bayes, Bagging, Boosting and Random Forest have their own strength.

Question 8

The importance of each classifiers (excluding Naïve Bayes) can be found in Diagram 8 to Diagram 11. To predict whether or not it is a cloudy day tomorrow, the 10 most important variables for each of the classifiers (except for Decision Tree has 13) are recorded in the table shown in Table 2 and Table 3. For variables that is the important variables of at least three out of four classifiers, we will conclude it to be the important variable to predict whether or not it is a cloudy day tomorrow. For those variables which do not include in any of the important variables of any classifiers, we can omit them.

Hence, important variables include Day, Month, Year, Location, MinTemp, Evaporation, Sunshine, WindGustDir, WindDir9am and WindDir3pm while the variables we can omit are Rainfall, WindGustSpeed, WindSpeed9am, WindSpeed3pm, Humidity9am, Temp9am and RainToday.

Classifiers	Accuracy	Specificity	Sensitivity	AUC	Total
Decision Tree	0.8277	0.3068	0.9187	0.6932	2.7464
Naïve Bayes	0.8345	0.4318	0.9048	0.7306	2.9017
Bagging	0.8598	0.2386	0.9683	0.8411	2.9078
Boosting	0.8598	0.2159	0.9722	0.8375	2.8854
Random Forest	0.8530	0.3068	0.9484	0.8386	2.9468

Table 1: Accuracy, sensitivity, precision, AUC and total for each classifier (red font for best performance)

Classifiers	Day	Month	Year	Location	MinTemp	MaxTemp	Rainfall	Evaporation	Sunshine	WindGustDir	WindGustSpeed	WindDir9am	WindDir3pm
Decision tree	✓	✓	✓	✓	✓	✓			✓	✓		✓	✓
Bagging	✓	✓	✓	✓				✓	✓	✓		✓	✓
Boosting	✓	✓	✓	✓	✓			✓	✓	✓		✓	✓
Random Forest	✓	✓	✓	✓	✓			✓	✓	✓		✓	✓

Table 2: Important variables for each classifier (first 13 variables)

Classifiers	WindSpeed9am	WindSpeed3pm	Humidity9am	Humidity3pm	Pressure9am	Pressure3pm	Temp9am	Temp3pm	RainToday
Decision tree				✓	✓			✓	
Bagging						✓			
Boosting									
Random Forest									

Table 3: Important variables for each classifier (last 9 variables)

```
Classification tree:
tree(formula = CloudTomorrow ~ ., data = WAUS.train)
Variables actually used in tree construction:
 [1] "Sunshine"     "Day"          "WindGustDir" "Month"        "WindDir3pm"   "WindDir9am"   "Location"     "Temp3pm"      "Humidity3pm" "Pressure9am"
[11] "Year"         "MinTemp"      "MaxTemp"
Number of terminal nodes:  30
Residual mean deviance:  0.2678 = 361.6 / 1350
Misclassification error rate: 0.06667 = 92 / 1380
```

Diagram 8: Summary of Decision Tree including variables actually used (important)

Day	Evaporation	Humidity3pm	Humidity9am	Location	MaxTemp	MinTemp	Month	Pressure3pm	Pressure9am
30.0996884	4.2529765	0.6488898	0.4218126	6.0167429	0.1697655	1.0394427	3.7467311	1.5571472	1.0290990
Rainfall	RainToday	Sunshine	Temp3pm	Temp9am	WindDir3pm	WindDir9am	WindGustDir	WindGustSpeed	WindSpeed3pm
0.2640413	0.0000000	26.2106764	1.0626433	0.2944013	5.4181275	5.9494286	8.6889872	0.1000639	0.0000000
WindSpeed9am	Year								
0.1283178	2.9010172								

Diagram 9: Variable importance of Bagging

Day	Evaporation	Humidity3pm	Humidity9am	Location	MaxTemp	MinTemp	Month	Pressure3pm	Pressure9am
24.5021273	4.0118572	1.4962219	1.5014301	4.5364621	0.6743487	1.6866198	8.0124703	1.1526367	1.1319634
Rainfall	RainToday	Sunshine	Temp3pm	Temp9am	WindDir3pm	WindDir9am	WindGustDir	WindGustSpeed	WindSpeed3pm
0.4152046	0.0000000	4.2838296	1.3741232	0.6180397	12.3609315	11.2089531	11.5159167	0.7023295	0.6682988
WindSpeed9am	Year								
0.7022282	7.4440074								

Diagram 10: Variable importance of Boosting

	MeanDecreaseGini
Day	55.3349791
Month	18.6374597
Year	20.4440874
Location	13.9682073
MinTemp	14.3701223
MaxTemp	9.3420960
Rainfall	4.8075942
Evaporation	32.5483833
Sunshine	37.9442971
WindGustDir	29.3650095
WindGustSpeed	8.9621869
WindDir9am	26.7974525
WindDir3pm	26.0082855
WindSpeed9am	7.7123689
WindSpeed3pm	7.6951945
Humidity9am	9.5982180
Humidity3pm	12.3793029
Pressure9am	11.4882259
Pressure3pm	12.2224944
Temp9am	9.8020417
Temp3pm	12.1251079
RainToday	0.6773839

Diagram 11: Variable importance of Random Forest

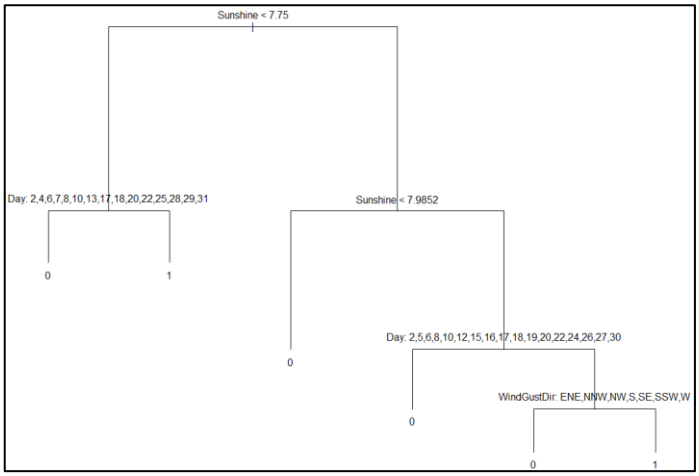

Diagram 12: Model of pruned Decision Tree

Question 9

To create a classifier that is simple enough for a person to be able to classify whether it will be cloudy or not tomorrow by hand, I choose to start with the Decision Tree model as it is simple enough to achieve the aim. To make the decision tree even simpler, I did a 20-folds cross validation to decide the tree complexity. It shows that the error rate it the lowest when tree size is 6. Therefore, I pruned the basic Decision Tree model with the results that I have obtained. Diagram 12 shows the simple pruned model. To predict whether it will be cloudy tomorrow by hand, we will look at the condition starting from top of the model. If the condition is fulfilled, branch to the left else branch to the right. Check the next condition and continue until the leaf is reached, which is 0 or 1 that indicates not cloudy and cloudy tomorrow respectively. These attributes: Sunshine, Day and WindGustDir are the most important attributes as only three of them are actually used in this tree construction. The accuracy is 0.8598, sensitivity is 0.9583, specificity is 0.2955, AUC is 0.7379 and total would be 2.8525 for this simple classifier. Comparing all of the measurements with the classifiers in Question 4, this pruned tree may not be better than Naïve Bayes, Bagging, Boosting or Random Forest, but it is undeniably better than the basic Decision Tree model.

Question 10

Since the Random Forest has the highest performance overall considering accuracy, sensitivity, specificity and AUC among all classifiers, I will like to make improvements on it to create a better Random Forest classifier. By looking at the code, it is clear that I have did a 5-folds cross validation with a 2-times reputation. Since I think accuracy is more important, tuning for mrty based on the accuracy is also done to improve the model. The results of tuning are shown in Diagram 13. The modified Random Tree classifier with mtry as 22 has an accuracy of 0.8598, sensitivity of 0.9722, specificity of 0.2159, AUC of 0.8601 and the sum of all of these measurements is 2.9080. Statistics of this model is shown in Diagram 14 below and ROC curves is shown in Diagram 15. As we can see, this model has the highest accuracy, sensitivity and AUC compared to other classifiers. Therefore, I would say that this modified Random Forest is the best classifier.

```
Random Forest

1380 samples
  22 predictor
   2 classes: '0', '1'

No pre-processing
Resampling: Cross-Validated (5 fold, repeated 2 times)
Summary of sample sizes: 1104, 1104, 1104, 1104, 1104, 1104, ...
Resampling results across tuning parameters:

  mtry  Accuracy   Kappa
   1    0.8333333  0.00000000
   2    0.8333333  0.00000000
   3    0.8351449  0.01775654
   4    0.8380435  0.05067318
   5    0.8456522  0.12744870
   6    0.8478261  0.15537371
   7    0.8507246  0.18506163
   8    0.8525362  0.20263960
   9    0.8539855  0.22907926
  10    0.8554348  0.25367754
  11    0.8547101  0.24964177
  12    0.8565217  0.27050523
  13    0.8554348  0.27297952
  14    0.8572464  0.29075921
  15    0.8597826  0.31233986
  16    0.8554348  0.28562249
  17    0.8536232  0.28405823
  18    0.8568841  0.31142540
  19    0.8594203  0.32889762
  20    0.8568841  0.31533592
  21    0.8568841  0.31920516
  22    0.8601449  0.34086296

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was mtry = 22.
```

Diagram 13: Results of Cross Validation and Tuning

```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 490  69
         1  14  19

               Accuracy : 0.8598
                 95% CI : (0.8292, 0.8868)
    No Information Rate : 0.8514
    P-Value [Acc > NIR] : 0.3051

                  Kappa : 0.2535

 Mcnemar's Test P-Value : 3.08e-09

            Sensitivity : 0.9722
            Specificity : 0.2159
         Pos Pred Value : 0.8766
         Neg Pred Value : 0.5758
             Prevalence : 0.8514
         Detection Rate : 0.8277
   Detection Prevalence : 0.9443
      Balanced Accuracy : 0.5941

       'Positive' Class : 0
```

Diagram 14: Confusion Matrix and Statistics

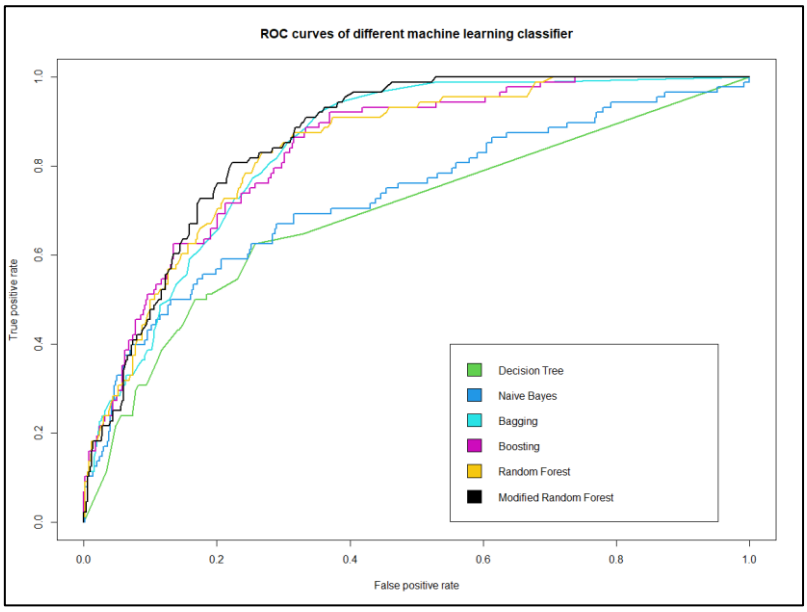

Diagram 15: ROC curves of all classifiers

Question 11

Attributes are all used in implementing this Artificial Neural Network. Since it can only handle numeric attributes, attributes that are in class factor are converted to numeric. WindGustDir, WindDir9am and WindDir3pm are converted to their degree based on the 16-wind compass. If RainToday is Yes, the value of it will be 1 else 2. After converting all attributes to numeric attributes except CloudTomorrow, data normalisation is done. Values of the attributes are shifted and rescaled so that they end up ranging between 0 and 1. The pre-processing is done. After training the ANN model, the confusion matrix and statistics of this ANN model is shown in Diagram 16 while the plot of this ANN model is shown in Diagram 17. We can retrieve statistic such as accuracy, sensitivity and precision. By comparing the accuracy (0.8429), sensitivity (0.9286) and specificity (0.3523) with other classifiers created above, this model is not considered as the best classifier as the modified Random Forest in Question 10 performs better than this. I think that the cause of this is the insufficient dataset. If the training dataset is very huge, ANN will have a better performance.

```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 468  57
         1  36  31

               Accuracy : 0.8429
                 95% CI : (0.811, 0.8713)
    No Information Rate : 0.8514
    P-Value [Acc > NIR] : 0.74004

                  Kappa : 0.3115

 Mcnemar's Test P-Value : 0.03809

            Sensitivity : 0.9286
            Specificity : 0.3523
         Pos Pred Value : 0.8914
         Neg Pred Value : 0.4627
             Prevalence : 0.8514
         Detection Rate : 0.7905
   Detection Prevalence : 0.8868
      Balanced Accuracy : 0.6404

       'Positive' Class : 0
```

Diagram 16: Confusion Matrix and Statistics of ANN model

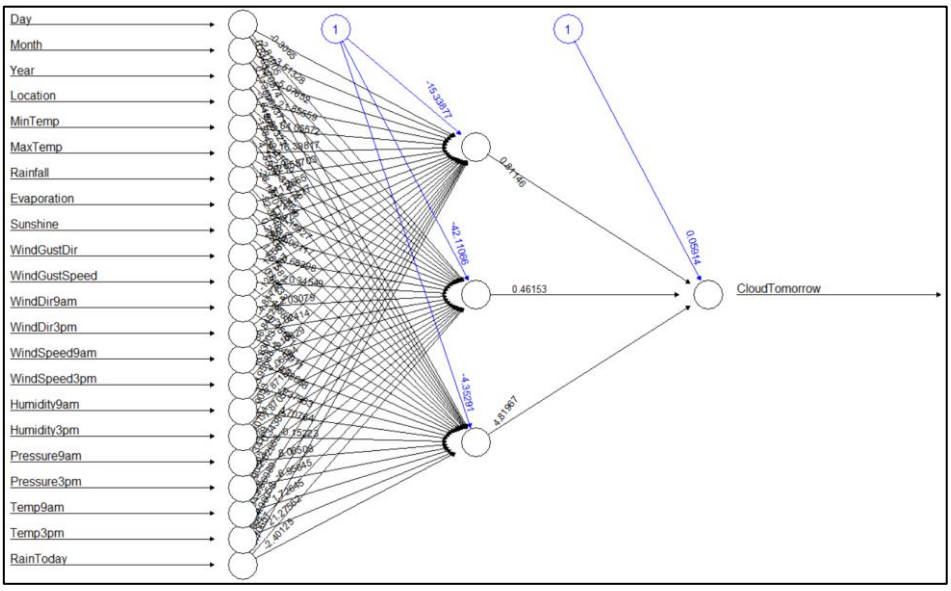

Diagram 17: Plot of ANN model

Appendix

```
library(tree)
library(e1071)
library(adabag)
library(randomForest)
library(ROCR)
library(caret)

rm(list = ls())
WAUS <- read.csv("CloudPredict2021.csv")
L <- as.data.frame(c(1:49))

set.seed(29797918)

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 & 2 ------------------------------ #

# Look at structure of WAUS
str(WAUS)

# Look at the summary of each independent variables before making any changes
summary(WAUS[,-23])

# Remove rows that are N/A in CloudTomorrow and modify class of variables below
WAUS <- WAUS[-which(is.na(WAUS$CloudTomorrow)),]
WAUS$Day <- as.factor(WAUS$Day)
WAUS$Month <- as.factor(WAUS$Month)
WAUS$Year <- as.factor(WAUS$Year)
WAUS$WindGustDir <- as.factor(WAUS$WindGustDir)
WAUS$WindDir9am <- as.factor(WAUS$WindDir9am)
WAUS$WindDir3pm <- as.factor(WAUS$WindDir3pm)
WAUS$RainToday <- as.factor(WAUS$RainToday)
WAUS$CloudTomorrow <- as.factor(WAUS$CloudTomorrow)

# Find proportion of cloudy to not cloudy days
sum(WAUS$CloudTomorrow==0)
sum(WAUS$CloudTomorrow==1)
sum(WAUS$CloudTomorrow==1)/sum(WAUS$CloudTomorrow==0)

# Look at the summary of current independent variables
summary(WAUS[,-23])

# Standard deviation and variation of each independent variables (only numeric
attriutes)
lapply(WAUS[complete.cases(WAUS[,c(5:9, 11, 14:21)]), c(5:9, 11, 14:21)], sd)
lapply(WAUS[complete.cases(WAUS[,c(5:9, 11, 14:21)]), c(5:9, 11, 14:21)], var)
```

```
# Function to return mode of the attribute (column)
getMode <- function(column) {
  freq <- as.data.frame(table(column))
  freq <- freq[order(freq$Freq, decreasing = TRUE),]
  return(as.factor(freq[1,1]))
}

# If numeric, fill in N/As with mean; if factor, fill in N/As with mode
for (i in 1:22){
  if (class(WAUS[,i]) == "factor"){
    WAUS[is.na(WAUS[,i]), i] <- getMode(WAUS[,i])
  } else {
    WAUS[is.na(WAUS[,i]), i] <- mean(WAUS[,i], na.rm=TRUE)
  }
}

# Look at the summary of independent variables after replacing N/A
summary(WAUS[,-23])


# ---------------------------------- QUESTION 3 ---------------------------------- #

# Split data to 70% training set and 30% test set
train.row = sample(1:nrow(WAUS), 0.7*nrow(WAUS))
WAUS.train = WAUS[train.row,]
WAUS.test = WAUS[-train.row,]


# ---------------------------------- QUESTION 4 ---------------------------------- #

# Fitting the DECISION TREE model
tree.fit <- tree(CloudTomorrow~., data = WAUS.train)

# Fitting the NAIVE BAYES model
nb.fit <- naiveBayes(CloudTomorrow~., data = WAUS.train)

# Fitting the BAGGING model
bag.fit <- bagging(CloudTomorrow~., data = WAUS.train)

# Fitting the BOOSTING model
boost.fit <- boosting(CloudTomorrow~., data = WAUS.train)

# Fitting the RANDOM FOREST model
rf.fit <- randomForest(CloudTomorrow~., data = WAUS.train)


# ---------------------------------- QUESTION 5 ---------------------------------- #

# Making predictions for DECISION TREE from test data
tree.predict <- predict(tree.fit, WAUS.test, type="class")
```

```
# Making predictions for NAIVE BAYES from test data
nb.predict <- predict(nb.fit, WAUS.test)

# Making predictions for BAGGING from test data
bag.predict <- predict.bagging(bag.fit, WAUS.test)

# Making predictions for BOOSTING from test data
boost.predict <- predict.boosting(boost.fit, WAUS.test)

# Making predictions for RANDOM FOREST from test data
rf.predict <- predict(rf.fit, WAUS.test)

# Confusion Matrix of Decision Tree, Naive Bayes, Bagging, Boosting and Random
Forest accordingly
confusionMatrix(data = tree.predict, reference =
as.factor(WAUS.test$CloudTomorrow))
confusionMatrix(data = nb.predict, reference = as.factor(WAUS.test$CloudTomorrow))
confusionMatrix(data = as.factor(bag.predict$class), reference =
as.factor(WAUS.test$CloudTomorrow))
confusionMatrix(data = as.factor(boost.predict$class), reference =
as.factor(WAUS.test$CloudTomorrow))
confusionMatrix(data = rf.predict, reference = as.factor(WAUS.test$CloudTomorrow))


# ------------------------------- QUESTION 6 -------------------------------- #

# ROC curve for DECISION TREE
t.pred <- prediction(predict(tree.fit, WAUS.test, type="vector")[,2],
WAUS.test$CloudTomorrow)
t.perf <- performance(t.pred,"tpr","fpr")
plot(t.perf, col=3, lwd=2, main="ROC curves of different machine learning
classifier")

# ROC curve for NAIVE BAYES
nb.pred <- prediction(predict(nb.fit, WAUS.test, type="raw")[,2],
WAUS.test$CloudTomorrow)
nb.perf <- performance(nb.pred,"tpr","fpr")
plot(nb.perf, col=4, lwd=2, add=TRUE)

# ROC curve for BAGGING
bag.pred <- prediction(predict(bag.fit, WAUS.test)$prob[,2],
WAUS.test$CloudTomorrow)
bag.perf <- performance(bag.pred,"tpr","fpr")
plot(bag.perf, col=5, lwd=2, add=TRUE)

# ROC curve for BOOSTING
boost.pred <- prediction(predict(boost.fit, WAUS.test)$prob[,2],
WAUS.test$CloudTomorrow)
boost.perf <- performance(boost.pred,"tpr","fpr")
plot(boost.perf, col=6, lwd=2, add=TRUE)
```

```
# ROC curve for RANDOM FOREST
rf.pred <- prediction(predict(rf.fit, WAUS.test, type="prob")[,2],
WAUS.test$CloudTomorrow)
rf.perf <- performance(rf.pred,"tpr","fpr")
plot(rf.perf, col=7, lwd=2, add=TRUE)

legend(0.6, 0.4, c("Decision Tree","Naive Bayes","Bagging","Boosting","Random
Forest"), 3:7)

# AUC
cat("Area under curve for Decision Tree:", as.numeric(performance(t.pred,
"auc")@y.values))
cat("Area under curve for Naive Bayes:", as.numeric(performance(nb.pred,
"auc")@y.values))
cat("Area under curve for Bagging:", as.numeric(performance(bag.pred,
"auc")@y.values))
cat("Area under curve for Boosting:", as.numeric(performance(boost.pred,
"auc")@y.values))
cat("Area under curve for Random Forest:", as.numeric(performance(rf.pred,
"auc")@y.values))


# ------------------------------- QUESTION 8 ------------------------------- #

# Importance of variables for each model
summary(tree.fit)
bag.fit$importance
boost.fit$importance
rf.fit$importance


# ------------------------------- QUESTION 9 ------------------------------- #

# Do cross validation to find out the best tree size (least error)
cv <- cv.tree(tree.fit, FUN=prune.misclass, K=20)
plot(cv$size, cv$dev, type="b", xlab="Tree Size", ylab="Error Rate", main="Cross
Validation: Error Vs Size")

# View the size and error
cv$size
cv$dev

# Prune the tree according to results above
new.tree <- prune.misclass(tree.fit, best=6)

# Predict this simple tree
new.tree.predict <- predict(new.tree, WAUS.test, type="class")

# Look at the summary of this pruned simple tree
summary(new.tree)
```

```
# Plot the simple tree
plot(new.tree)
text(new.tree, pretty = 12)

# Confusion matrix and statistics of this tree
confusionMatrix(data = new.tree.predict, reference =
as.factor(WAUS.test$CloudTomorrow))

# AUC of this pruned simple tree
new.pred <- prediction(predict(new.tree, WAUS.test, type="vector")[,2],
WAUS.test$CloudTomorrow)
cat("Area under curve for pruned Decision Tree:", as.numeric(performance(new.pred,
"auc")@y.values))


# ------------------------------- QUESTION 10 -------------------------------- #

# Do 5-folds cross validation (2 times) and hyperparameter tuning (mtry)
control <- trainControl(method='repeatedcv', number=5, repeats=2, search='grid')
tunegrid <- expand.grid(.mtry=c(1:22))
rft.fit <- train(CloudTomorrow ~., data = WAUS.train, method = 'rf',metric =
'Accuracy', tuneGrid = tunegrid, trControl=control)

# Look at the results
print(rft.fit)

# Make predictions using test data
rft.predict <- predict(rft.fit, WAUS.test)

# Confusion Matrix and statistics of this model
confusionMatrix(data = rft.predict, reference =
as.factor(WAUS.test$CloudTomorrow))

# AUC
rft.pred <- prediction(predict(rft.fit, WAUS.test, type="prob")[,2],
WAUS.test$CloudTomorrow)
rft.perf <- performance(rft.pred,"tpr","fpr")
cat("Area under curve for modified Random Forest:",
as.numeric(performance(rft.pred, "auc")@y.values))

# Plot the ROC curves for all classifiers until now to compare
plot(t.perf, col=3, lwd=2, main="ROC curves of different machine learning
classifier")
plot(nb.perf, col=4, lwd=2, add=TRUE)
plot(bag.perf, col=5, lwd=2, add=TRUE)
plot(boost.perf, col=6, lwd=2, add=TRUE)
plot(rf.perf, col=7, lwd=2, add=TRUE)
plot(rft.perf, col=9, lwd=2, add=TRUE)

legend(0.55, 0.4, c("Decision Tree","Naive Bayes","Bagging","Boosting","Random
Forest","Modified Random Forest"), c(3:7,9))
```

```
# ------------------------------- QUESTION 11 -------------------------------- #

library(neuralnet)

# Make a copy for training set and testing set
neural.train <- WAUS.train
neural.test <- WAUS.test

# Convert wind directions to integer, change factors to numeric (Pre-processing)
dir <- setNames(seq(0, 337.5 , by=22.5), c("N","NNE","NE", "ENE", "E", "ESE",
"SE", "SSE", "S", "SSW", "SW", "WSW", "W", "WNW", "NW", "NNW"))
neural.train$Day <- as.numeric(neural.train$Day)
neural.train$Month <- as.numeric(neural.train$Month)
neural.train$Year <- as.numeric(neural.train$Year)
neural.train$WindGustDir <- dir[neural.train$WindGustDir]
neural.train$WindDir9am <- dir[neural.train$WindDir9am]
neural.train$WindDir3pm <- dir[neural.train$WindDir3pm]
neural.train$RainToday <- ifelse(neural.train$RainToday=="Yes", 1, 2)
neural.train$CloudTomorrow <- ifelse(neural.train$CloudTomorrow==0, 0, 1)

neural.test$Day <- as.numeric(neural.test$Day)
neural.test$Month <- as.numeric(neural.test$Month)
neural.test$Year <- as.numeric(neural.test$Year)
neural.test$WindGustDir <- dir[neural.test$WindGustDir]
neural.test$WindDir9am <- dir[neural.test$WindDir9am]
neural.test$WindDir3pm <- dir[neural.test$WindDir3pm]
neural.test$RainToday <- ifelse(neural.test$RainToday=="Yes", 1, 2)
neural.test$CloudTomorrow <- ifelse(neural.test$CloudTomorrow==0, 0, 1)

# Normalize data (min-max scaling)
normalization <- function(x) {
  return ((x - min(x)) / (max(x) - min(x)))
}

for (i in 1:22){
  neural.train[,i] <- normalization(neural.train[,i])
  neural.test[,i] <- normalization(neural.test[,i])
}

# Fit the ANN model
ann <- neuralnet(CloudTomorrow~., data=neural.train, hidden=3, threshold=0.01)

# Plot ANN model
plot(ann)

# Look at the result matrix of ANN model
ann$result.matrix

# Make prediction using test data
prediction <- compute(ann, neural.test)
```

```
prob <- prediction$net.result
pred <- ifelse(prob > 0.5, 1, 0)

# Confusion Matrix of ANN model
confusionMatrix(data = as.factor(pred), reference =
as.factor(WAUS.test$CloudTomorrow))
```