Weather Prediction for 10 locations in Australia

Author: Kuah Jia Chen

Overview

Please refer to the "source_code.R" file for all the R code used in the analysis presented in this report. The report consists of a total of 11 pages.

Question 1

Explore the data: What is the proportion of days when it is warmer than the previous day compared to those where it is cooler? Obtain descriptions of the predictor (independent) variables – mean, standard deviations, etc. for real-valued attributes. Is there anything noteworthy in the data? Are there any attributes you need to consider omitting from your analysis?

Answer:

The analysis reveals that 54.6% of the days were warmer than the previous day, while 44.7% of the days were cooler. This indicates a higher proportion of days being warmer than the previous day, exceeding the cooler days by approximately 10%. Notably, there were 13 instances where the "WarmerTomorrow" attribute had missing values, accounting for approximately 0.7% of the dataset.

For real-valued attributes, I employed the "summary" and "sd" functions to obtain key statistics such as the minimum, 1st quartile, median, mean, 3rd quartile, number of NAs, and standard deviation. It's important to note that these statistics were calculated only for the real-valued attributes, excluding the categorical ones.

```
> summary(WAUS[c(5:9,11,14:19,22:23)]) # use summary function only for those columns that are real-valued
    MinTemp         MaxTemp         Rainfall       Evaporation       Sunshine     WindGustSpeed    WindSpeed9am    WindSpeed3pm    Humidity9am
 Min.   :-7.4   Min.   :-2.3   Min.   :  0.0   Min.   : 0     Min.   : 0     Min.   : 11.0   Min.   : 0.0   Min.   : 0.0   Min.   : 13.0
 1st Qu.: 6.9   1st Qu.:18.0   1st Qu.:  0.0   1st Qu.: 3     1st Qu.: 5     1st Qu.: 31.0   1st Qu.: 7.0   1st Qu.:13.0   1st Qu.: 57.0
 Median :11.8   Median :23.0   Median :  0.0   Median : 5     Median : 8     Median : 41.0   Median :13.0   Median :19.0   Median : 70.0
 Mean   :11.7   Mean   :22.8   Mean   :  2.8   Mean   : 5     Mean   : 8     Mean   : 42.2   Mean   :14.2   Mean   :19.1   Mean   : 68.9
 3rd Qu.:17.2   3rd Qu.:28.1   3rd Qu.:  0.6   3rd Qu.: 7     3rd Qu.:10     3rd Qu.: 50.0   3rd Qu.:20.0   3rd Qu.:24.0   3rd Qu.: 82.0
 Max.   :28.0   Max.   :45.7   Max.   :115.2   Max.   :35     Max.   :14     Max.   :113.0   Max.   :65.0   Max.   :78.0   Max.   :100.0
 NA's   :36     NA's   :18     NA's   :43      NA's   :1079   NA's   :1186   NA's   :295     NA's   :60     NA's   :131    NA's   :46
  Humidity3pm     Pressure9am     Pressure3pm      Temp9am        Temp3pm
 Min.   :  5.0   Min.   : 995   Min.   : 986   Min.   :-4.5   Min.   :-2.7
 1st Qu.: 38.0   1st Qu.:1013   1st Qu.:1010   1st Qu.:12.1   1st Qu.:16.6
 Median : 52.0   Median :1018   Median :1015   Median :17.0   Median :21.4
 Mean   : 52.5   Mean   :1018   Mean   :1015   Mean   :16.8   Mean   :21.2
 3rd Qu.: 66.0   3rd Qu.:1023   3rd Qu.:1020   3rd Qu.:21.8   3rd Qu.:26.2
 Max.   :100.0   Max.   :1039   Max.   :1037   Max.   :35.7   Max.   :43.8
 NA's   :117     NA's   :608    NA's   :615    NA's   :46     NA's   :114
```

Figure 1 shows the description of the predictor variables with real-valued attributes

```
> standard_deviation_dataframe
       Variables Standard Deviation
1        MinTemp                6.95
2        MaxTemp                7.45
3       Rainfall                9.84
4    Evaporation                3.58
5       Sunshine                3.74
6  WindGustSpeed                14.3
7   WindSpeed9am                9.43
8   WindSpeed3pm                9.35
9    Humidity9am                18.4
10   Humidity3pm                20.3
11   Pressure9am                6.92
12   Pressure3pm                6.93
13       Temp9am                6.93
14       Temp3pm                 7.3
```

Figure 2 shows the standard deviation for predictor variables for real-valued attributes

Figure 1 and Figure 2 provide a detailed description of the real-valued attributes. Since the summary function does not provide the standard deviation, I calculated it separately and stored the results in a separate data frame.

One notable observation in the data is the presence of numerous NA values in each column, particularly in Evaporation and Sunshine, which have over a thousand NA values (Figure 1). Additionally, Humidity3pm exhibits the highest standard deviation, approximately 20, followed by Humidity9am.

It is important to note that the units for each variable are inconsistent and unrelated, rendering their mean, median, and other descriptive measures (Figure 1) incomparable. However, it is interesting to observe that MinTemp, MaxTemp, Temp9am, and Temp3pm have negative minimum values. Furthermore, Rainfall displays a significant difference between its minimum and maximum values, approximately 115.

Furthermore, there are certain attributes that I am considering omitting from my analysis, namely Day, Month, and Year. These attributes solely represent the date of observation and do not provide any useful information for predicting whether tomorrow will be warmer than today. Including them in the analysis may introduce noise to the data. Initially, I contemplated excluding Location and the attributes related to wind direction as well. However, after conducting online research, I discovered that these attributes do contribute to temperature variations. Consequently, I decided to retain them. To summarize, I have omitted Day, Month, and Year from my analysis, and they will be removed in Part 2.

Question 2

Document any pre-processing required to make the data set suitable for the model fitting that follows.

Answer:

Before proceeding with model fitting, several pre-processing steps are required to ensure the dataset is suitable. Firstly, to address the exclusion of Day, Month, and Year attributes mentioned in Part 1, a new data frame called "new.WAUS" will be created. This data frame is identical to "WAUS," but does not include the aforementioned attributes.

Next, the categorical data needs to be converted to the factor data type. The attributes to be converted to factors are Location, WindGustDir, WindDir9am, WindDir3pm, Cloud9am, Cloud3pm, and WarmerTomorrow. By performing this conversion, the "new.WAUS" data frame will be prepared for model fitting. It is worth noting that the original "WAUS" data frame was not modified directly to adhere to the assignment's specification that "WAUS" should be the default data frame name for the entire dataset. Creating a new data frame to store the pre-processed data maintains consistency with the assignment requirements.

Lastly, the rows containing missing values (NA) will be removed. This step is necessary as not all classification models can handle NAs, such as Random Forest. Excluding rows with missing values only during the construction of the Random Forest model while keeping them

for other models would introduce inconsistency, as the other models would utilize data excluded in Random Forest's construction. To ensure fairness and consistency, all rows with missing values will be removed before building any models. As a result, "new.WAUS" will consist of 554 rows of data.

Question 3

Divide the data into a 70% training and 30% test set by adapting the following code (written for the iris data).

Answer:

Please refer to the code in the source_code.R to generate the training and testing data set.

Question 4

Implement a classification model using each of the following techniques. For this question you may use each of the R functions at their default settings if suitable.

Answer:

```
# Decision Tree
# fit decision tree model to predict WarmerTomorrow
Decision.Tree.new.WAUS.fit = tree(WarmerTomorrow ~ ., data=new.WAUS.train)

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

# Bagging
# fit bagging model to predict WarmerTomorrow
Bagging.new.WAUS.fit = bagging(WarmerTomorrow~., data = new.WAUS.train, mfinal=10)

# Boosting
# fit boosting model to predict WarmerTomorrow
Boosting.new.WAUS.fit = boosting(WarmerTomorrow~., data = new.WAUS.train, mfinal=5)

# Random Forest
# fit random forest model to predict WarmerTomorrow
Random.Forest.new.WAUS.fit = randomForest(WarmerTomorrow ~., data = new.WAUS.train)
```

Figure 3 shows the code I used to implement the five classification models

For the bagging and boosting models, I experimented with different values of mfinal, specifically 5 and 10, to determine the setting that yields higher accuracy. I deviated from the default setting of mfinal=100 because I found that such a large value was not suitable for my analysis.

Question 5

Using the test data, classify each of the test cases as 'warmer tomorrow' or 'not warmer tomorrow'. Create a confusion matrix and report the accuracy of each model.

Answer:

Please refer to the source_code.R for the code I used to perform classification and create the confusion matrix

1) Decision Tree

- The confusion matrix for the decision tree classifier

```
          actual
predicted  0  1
        0 53 45
        1 26 43
```

Figure 4 shows the confusion matrix for decision tree model

- Accuracy = (53+43) / (53+26+45+43) ≈ 0.575 ≈ 57.5%

2) Naïve Bayes

- The confusion matrix for the naive bayes classifier

```
          predicted
predicted  0  1
        0 57 22
        1 37 51
```

Figure 5 shows the confusion matrix for naive bayes classifier

- Accuracy = (57+51) / (57+37+22+51) ≈ 0.647 ≈ 64.7%

3) Bagging

- The confusion matrix for the bagging classifier

```
                 Observed Class
Predicted Class  0  1
              0 39 20
              1 40 68
```

Figure 6 shows the confusion matrix for bagging model

- Accuracy = (39+68) / (39+40+20+68) ≈ 0.641 ≈ 64.1%

4) Boosting

- The confusion matrix for the Boosting classifier

	Observed Class	
Predicted Class	0	1
0	44	30
1	35	58

Figure 7 shows the confusion matrix for boosting model

- Accuracy = (44+58) / (44+30+35+58) ≈ 0.611 ≈ 61.1%

5) Random Forest

- The confusion matrix for the Random Forest classifier

	actual	
predicted	0	1
0	40	25
1	39	63

Figure 8 shows the confusion matrix for random forest model

- Accuracy = (40+63) / (40+25+39+63) ≈ 0.617 ≈ 61.7%

Question 6

Using the test data, calculate the confidence of predicting 'warmer tomorrow' for each case and construct an ROC curve for each classifier. You should be able to plot all the curves on the same axis. Use a different colour for each classifier. Calculate the AUC for each classifier.

Answer:

To construct the ROC curve, the first step involves calculating the confidence level for predicting the target attribute for each case. Subsequently, the ROC curve is constructed, representing the True Positive Rate and False Positive Rate across all confidence levels. The ultimate goal is to calculate the AUC (Area Under the Curve) value for each classifier.

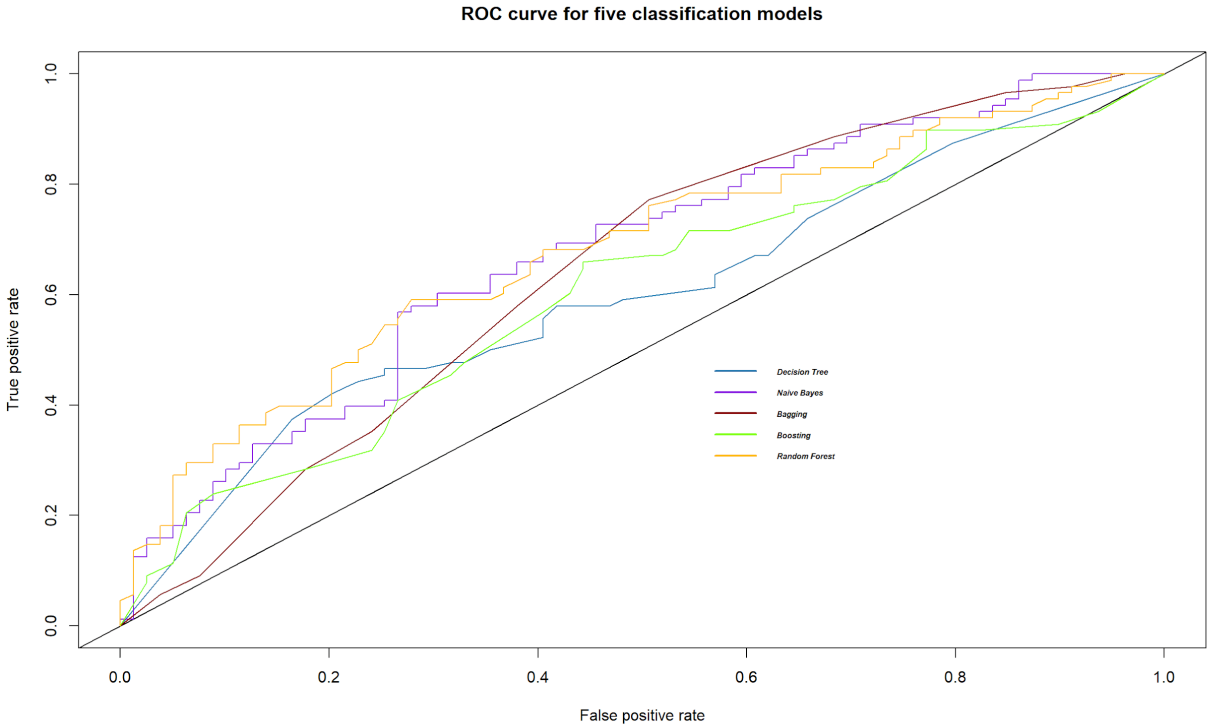

Figure 9 shows the ROC curve for each classifier

The AUC for decision tree: 0.602

The AUC for naïve bayes: 0.676

The AUC for Bagging: 0.642

The AUC for Boosting: 0.605

The AUC for Random Forest: 0.68

Question 7

Create a table comparing the results in parts 5 and 6 for all classifiers. Is there a single "best" classifier?

Answer:

	Decision Tree	Naïve Bayes	Bagging	Boosting	Random Forest
Accuracy	0.575	0.647	0.641	0.611	0.617
AUC	0.602	0.676	0.642	0.605	0.68

The table above presents the accuracy and AUC values for all the classification models. Among the five models, Naïve Bayes exhibits the highest accuracy at 0.647, followed by Bagging, Random Forest, Boosting, and Decision Tree. On the other hand, Random Forest achieves the highest AUC value at 0.68, with Naïve Bayes, Bagging, Boosting, and Decision Tree following in that order. A higher AUC value indicates a classifier's greater ability to differentiate between positive and negative examples across all confidence levels. In this case, all classifier models demonstrate AUC values between "No discrimination" (AUC = 0.5) and "Acceptable discrimination" (0.7 ≤ AUC < 0.8) ranges.

Based on these comparisons, the Naïve Bayes model emerges as the "best" classifier. It not only boasts the highest accuracy but also ranks second in terms of AUC among the models. Moreover, the difference between its AUC and that of Random Forest is only around 0.004. Thus, considering the overall performance, the Naïve Bayes model is deemed the most favorable classifier in this scenario.

Question 8

Examining each of the models, determine the most important variables in predicting whether it will be warmer tomorrow or not. Which variables could be omitted from the data with very little effect on performance? Give reasons.

Answer:

To address this question, I will initially identify the important or most important variables for each model before determining the most important variables overall. The same approach will be applied to identify variables with minimal impact on performance. However, it should be noted that variable importance cannot be determined for the Naïve Bayes classifier as no model is available for it. Hence, we will solely examine the variable importance for the remaining models.

```
#Decision Tree Attribute Importance
> print(summary(Decision.Tree.new.WAUS.fit))

Classification tree:
tree(formula = WarmerTomorrow ~ ., data = new.WAUS.train)
Variables actually used in tree construction:
 [1] "WindDir9am"   "WindGustDir"  "Sunshine"     "Humidity9am"  "WindDir3pm"
 [6] "Humidity3pm"  "Location"     "Cloud3pm"     "MinTemp"      "Cloud9am"
[11] "Temp9am"      "MaxTemp"      "Evaporation"  "WindSpeed9am" "Rainfall"
[16] "Pressure9am"
Number of terminal nodes:  35
Residual mean deviance:  0.456 = 161 / 352
Misclassification error rate: 0.093 = 36 / 387
```

Figure 10 shows the variables used in decision tree construction

To determine the important variables in the decision tree model, we can refer to the "variables used in tree construction" section in Figure 10. In this section, all the variables listed can be considered important for predicting the target variable using the decision tree model. Consequently, we can infer that WindGustSpeed, WindSpeed3pm, Pressure3pm, and Temp3pm do not contribute to the construction of the decision tree model, as they are not utilized. Moreover, based on the tree plot, the most influential variables in the decision

tree model are WindDir9am, WindGustDir, and MaxTemp, as they appear at the top branch of the decision tree. Although the tree plot itself is not included in this report due to its complexity, the code used to generate the plot is provided in the appendix.

```
#Baging Attribute Importance
> print(Bagging.new.WAUS.fit$importance)
     Cloud3pm      Cloud9am   Evaporation   Humidity3pm   Humidity9am
        3.624         3.674         3.383         3.938         1.036
     Location       MaxTemp       MinTemp   Pressure3pm   Pressure9am
        0.850         7.229         0.749         2.972         3.013
     Rainfall      Sunshine       Temp3pm       Temp9am    WindDir3pm
        1.974         6.179         2.051         2.794        17.187
   WindDir9am   WindGustDir WindGustSpeed  WindSpeed3pm  WindSpeed9am
       19.138        15.701         2.328         1.265         0.915
```

Figure 11 shows the importance of variables in predicting WarmerTomorrow for Bagging model

In Figure 11, the importance of each attribute in the Bagging model is illustrated. The results indicate that WindDir9am holds the highest importance value, making it the most influential variable in the Bagging model. It is closely followed by WindDir3pm and WindGustDir, which also exhibit considerable importance. On the other hand, attributes such as Humidity9am, Location, MinTemp, Pressure3pm, Rainfall, Temp3pm, Temp9am, WindGustSpeed, WindSpeed3pm, and WindSpeed9am display relatively lower importance values, less than 3. These variables can be considered as less important in this model. Meanwhile, the remaining variables demonstrate medium importance.

```
#Boosting Attribute Importance
> print(Boosting.new.WAUS.fit$importance)
     Cloud3pm      Cloud9am   Evaporation   Humidity3pm   Humidity9am
        7.730         6.933         4.113         2.404         3.207
     Location       MaxTemp       MinTemp   Pressure3pm   Pressure9am
        1.037         3.043         0.000         2.139         0.000
     Rainfall      Sunshine       Temp3pm       Temp9am    WindDir3pm
        0.958         3.344         2.169         2.139        10.316
   WindDir9am   WindGustDir WindGustSpeed  WindSpeed3pm  WindSpeed9am
       23.322        22.815         1.989         0.830         1.513
```

Figure 12 shows the importance of variables in predicting WarmerTomorrow for Boosting model

In the Boosting model, it is evident from Figure 12 that WindDir9am, WindGustDir, and WindDir3pm are the important variables, given their relatively higher importance values compared to others. Conversely, the least important variables include Humidity3pm, Location, MinTemp, Pressure3pm, Pressure9am, Rainfall, Temp3pm, Temp9am, WindGustSpeed, WindSpeed3pm, and WindSpeed9am. This can be attributed to the fact that these variables either do not contribute to the model (with an importance value of 0) or their importance values are below 3. On the other hand, the remaining variables exhibit moderate importance.

```
#Random Forest Attribute Importance
> print(Random.Forest.new.WAUS.fit$importance)
              MeanDecreaseGini
Location                  3.06
MinTemp                   6.51
MaxTemp                   9.47
Rainfall                  3.37
Evaporation              10.11
Sunshine                  8.79
WindGustDir              22.07
WindGustSpeed             5.54
WindDir9am               23.58
WindDir3pm               19.44
WindSpeed9am              4.64
WindSpeed3pm              5.46
Humidity9am               7.45
Humidity3pm               9.39
Pressure9am               8.40
Pressure3pm               7.59
Cloud9am                  8.31
Cloud3pm                 11.44
Temp9am                   7.36
Temp3pm                   9.70
```

Figure 13 shows the importance of variables in predicting WarmerTomorrow for Random Forest model

In the Random Forest model, it is evident from Figure 13 that the important variables are WindGustDir, WindDir9am, and WindDir3pm, as their importance values are comparatively higher than others. Conversely, I would consider Location, Rainfall, WindGustSpeed, WindSpeed9am, and WindSpeed3pm as unimportant variables in constructing the Random Forest model, as their importance values are lower than 6.

In summary, the most important variables overall for predicting WarmerTomorrow would be WindDir9am, WindDir3pm, and WindGustDir. This conclusion is drawn from the fact that these variables are considered important in all or most of the classifier models. On the other hand, the variables that could be omitted from the data with minimal effect on performance overall are WindGustSpeed and WindSpeed3pm, as they are considered unimportant variables in all four classifier models. Additionally, Pressure3pm, Temp3pm, Location, Rainfall, and WindSpeed9am can also be omitted, as they are classified as unimportant variables in most of the classification models. In summary, there are three important variables and seven variables that could be omitted from the data with minimal effect on performance.

Question 9

Starting with one of the classifiers you created in Part 4, create a classifier that is simple enough for a person to be able to classify whether it will be warmer tomorrow or not by hand. Describe your model, either with a diagram or written explanation. How well does your model perform, and how does it compare to those in Part 4? What factors were important in your decision? State why you chose the attributes you used.

Answer:

For this question, I have decided to create a simplified decision tree model instead of using ensemble classifiers like Bagging, Boosting, and Random Forest. The main reason behind this decision is that a single decision tree is much easier to interpret compared to complex models involving multiple classifiers. The goal is to assess whether simplifying the decision tree would improve its performance. Additionally, I did not consider the Naïve Bayes model from Part 4 as it already yielded satisfactory results. Although the decision tree had the lowest accuracy and AUC among the classifiers in Part 4, I believe that by making the decision tree simpler, its performance can potentially be enhanced. Furthermore, if the resulting decision tree model turns out to be complex, I can utilize cross-validation to prune it. Considering these factors, I believe that the advantages of using a decision tree outweigh those of the other classifiers. These considerations guided my decision-making process.

To build the new decision tree model, I will only utilize the variables WindDir9am, WindDir3pm, and WindGustDir to predict WarmerTomorrow. This selection is based on their significance as determined in Part 8. Therefore, I will follow a similar data preprocessing approach as described in Part 2 and proceed to create the model.

```
> table(predicted = Q9.D.predict ,actual = Q9.WAUS.test$WarmerTomorrow)
         actual
predicted   0   1
        0 112 110
        1 101 142
> summary(Decision.Tree.Q9.WAUS.fit)

Classification tree:
tree(formula = WarmerTomorrow ~ ., data = Q9.WAUS.train)
Number of terminal nodes:  4
Residual mean deviance:  1.3 = 1400 / 1080
Misclassification error rate: 0.366 = 397 / 1084
> (112+142)/(112+110+101+142)
[1] 0.546
```

Figure 14 shows the accuracy, confusion matrix and summary of the new decision tree model after preprocessing the data

From the figure above, we can observe that this decision tree model consists of 4 terminal nodes, with an accuracy of approximately 54.6%. To further enhance the interpretability of the model, I will attempt to prune the tree using cross-validation. By employing a cross-validation test based on the misclassification rate, I aim to obtain a simpler tree while maintaining a comparable level of accuracy.

```
> #cross validation test
> cvtest = cv.tree(Decision.Tree.Q9.WAUS.fit, FUN = prune.misclass)
> cvtest
$size
[1] 4 3 2 1

$dev
[1] 444 444 444 499

$k
[1] -Inf    0   28   66

$method
[1] "misclass"

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

Figure 15 shows the details of the cross-validation test

By pruning the decision tree size to 3 (i.e., 3 terminal nodes), we observe that the dev value remains unchanged compared to the original tree, and the cost complexity is 0. Consequently, we can conclude that pruning the decision tree to a size of 3 would result in a simpler tree without compromising the model's performance.

```
> #prune using size 3 considering lowest misclassification rate
> # and lowest cost complexity
> pruned.Dfit = prune.misclass(Decision.Tree.Q9.WAUS.fit, best = 3)
> summary(pruned.Dfit)

Classification tree:
snip.tree(tree = Decision.Tree.Q9.WAUS.fit, nodes = 3L)
Variables actually used in tree construction:
[1] "WindDir9am"  "WindGustDir"
Number of terminal nodes:  3
Residual mean deviance:  1.31 = 1420 / 1080
Misclassification error rate: 0.366 = 397 / 1084
> # check accuracy using the pruned tree
> PD.predict = predict(pruned.Dfit, Q9.WAUS.test, type = "class")
> table(predicted = PD.predict ,actual = Q9.WAUS.test$WarmerTomorrow)
         actual
predicted   0   1
        0 112 110
        1 101 142
> (112+142)/(112+110+101+142)
[1] 0.546
```

Figure 16 shows the details of the pruned decision tree model

Following the pruning process, the pruned decision tree now solely relies on WindDir9am and WindGuestDir as predictors. Interestingly, the accuracy and misclassification error rate remain unchanged when comparing the original and pruned tree. Nevertheless, a noteworthy improvement is achieved as the pruned tree is simpler, consisting of only 3 terminal nodes, compared to the previous tree with 4 nodes.

```
> # AUC for Question 9 Pruned Decision Tree model
> Q9.decision.tree.auc = performance(Q9.decision.tree.pred, "auc")
> print(as.numeric(Q9.decision.tree.auc@y.values)) # 0.547
[1] 0.547
```

Figure 17 shows the AUC value for pruned decision tree model

Now, let's evaluate the performance of our final decision model, which achieved an accuracy of 54.6% and an AUC of approximately 0.547. Comparing these results with the performance of the models in Part 4, it is evident that this decision tree model performed the poorest, exhibiting the lowest accuracy and AUC among all the classifiers.

To further understand and explain the model, I will plot the pruned decision tree and provide a detailed explanation of its structure and decision-making process.

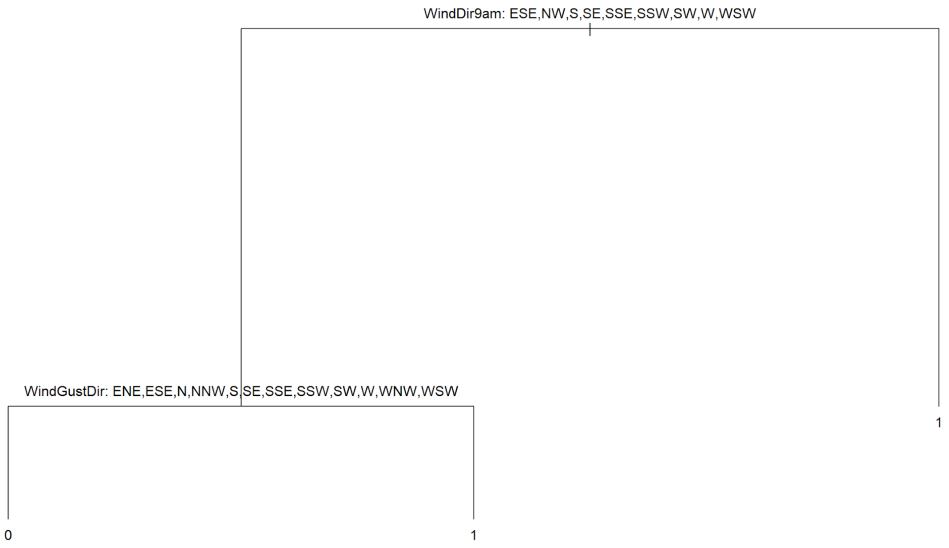

Figure 18 shows the pruned decision tree

The pruned decision tree model can be described as follows:

1. If the direction of the wind at 9 am (WindDir9am) is ESE, NW, S, SE, SSE, SSW, SW, W, or WSW, and the direction of the strongest wind gust over the day (WindGustDir) is ENE, ESE, N, NNW, S, SE, SSE, SSW, SW, W, WNW, or WSW, the model predicts that tomorrow will not be warmer than today (WarmerTomorrow = 0).
2. If the direction of the wind at 9 am (WindDir9am) is ESE, NW, S, SE, SSE, SSW, SW, W, or WSW, and the direction of the strongest wind gust over the day (WindGustDir) is NNE, NE, NW, or E, the model predicts that tomorrow will be warmer than today (WarmerTomorrow = 1).
3. If the direction of the wind at 9 am (WindDir9am) is WNW, NNE, N, ENE, NE, NNW, or E, the model predicts that tomorrow will be warmer than today (WarmerTomorrow = 1).

Question 10

Create the best tree-based classifier you can. You may do this by adjusting the parameters, and/or cross-validation of the basic models in Part 4 or using an alternative tree-based learning algorithm. Show that your model is better than the others using appropriate measures. Describe how you created your improved model, and why you chose that model. What factors were important in your decision? State why you chose the attributes you used.

To create my improved model, I carefully selected the attributes based on the findings from Part 8. I included MaxTemp, Sunshine, WindGustDir, WindDir9am, WindDir3pm, Humidity9am, Cloud9am, Cloud3pm, and Temp9am as they were identified as important variables in previous analyses. These attributes were deemed crucial for predicting WarmerTomorrow.

After selecting the attributes, I performed the necessary pre-processing steps, such as extracting the selected attributes from the dataset, converting categorical attributes to factors, removing rows with missing values, and splitting the data into training and testing sets. These steps ensured that the data was suitable for model fitting.

Next, I considered various classifiers, including Naïve Bayes, Bagging, Boosting, and Random Forest. Since the decision tree models created in Parts 4 and 8 did not yield satisfactory results, I excluded them from consideration. To determine the best classifier, I conducted multiple rounds of model fitting, parameter adjustment, and cross-validation.

After thorough experimentation, I found that the Bagging model with mfinal set to 29 provided the most significant improvement in performance compared to the other classifiers. Hence, I chose the Bagging model as my best model for this question.

It is important to note that the process involved some trial and error, as I explored different models and parameters to identify the one that produced the best results. To ensure clarity, I have removed the code related to other classifiers from my report, focusing solely on the code used to create the Bagging model.

The performance of my improved model demonstrates its superiority compared to the classifiers in Parts 4 and 8. The accuracy of the model is approximately 67.2%, while the AUC is around 0.698.

The performance of my best model (Bagging):

	Observed Class	
Predicted Class	0	1
0	62	36
1	26	65

Figure 19 shows the confusion matrix of the best model using Bagging

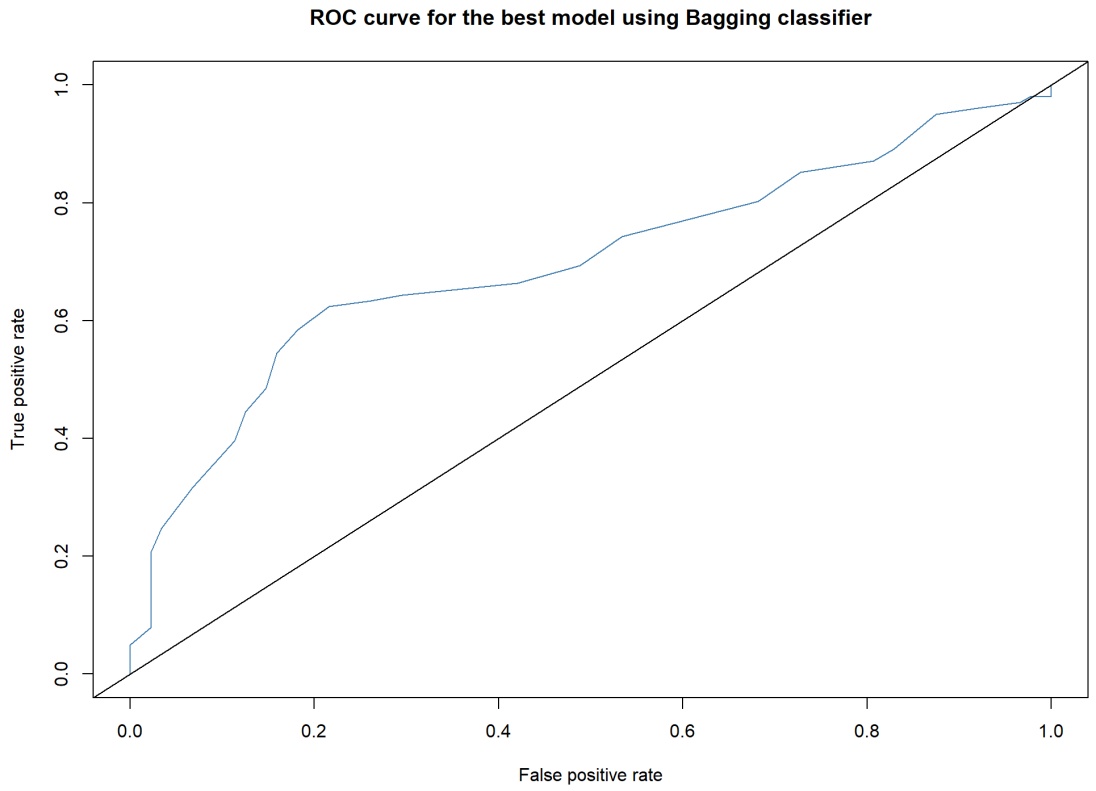

Figure 20 shows the ROC curve for the best model using Bagging classifier

```
> # AUC for Bagging model
> bagging.Q10.auc = performance(Bagging.predict.Q10.pred, "auc")
> print(as.numeric(bagging.Q10.auc@y.values)) # 0.698
[1] 0.698
```

Figure 21 shows the AUC value for the best model using Bagging classifier

The improved Bagging model achieved an accuracy of approximately 67.2% (i.e., (62+65) / (62+36+26+65)) and an AUC of 0.698. Comparing these results with the classifiers in Parts 4 and 8, we observe that this best model outperforms all of them. Although the improvement may appear modest, with only approximately 2.5% higher accuracy and a roughly 0.02 higher AUC compared to the best model in Parts 4 and 8 (which was the Naïve Bayes model in Part 4), even minor enhancements can have a significant impact when predicting the target variable.

Question 11

Using the insights from your analysis so far, implement an Artificial Neural Network classifier and report its performance. Comment on attributes used and your data preprocessing required. How does this classifier compare with the others? Can you give any reasons?

Answer:

I selected the following attributes to construct the Artificial Neural Network (ANN) classifier: MaxTemp, Sunshine, WindGustDir, WindDir9am, WindDir3pm, Humidity9am, Cloud9am, Cloud3pm, and Temp9am. The rationale behind choosing these attributes is consistent with the explanation provided in Part 10. Based on the analysis in Part 8, I believe these attributes can make a moderate contribution to predicting the value of WarmerTomorrow.

Regarding data pre-processing for the ANN classifier, it differs from other models. ANN models typically require additional steps such as normalization or scaling of input features, handling missing values, and potentially encoding categorical variables. These procedures ensure that the data is appropriately prepared for training the neural network.

Here are the steps/order of the data pre-processing:

1. Extract the selected attributes from the "WAUS" dataset and store them in a new data frame called "Question.11.WAUS".
2. Remove rows from "Question.11.WAUS" that contain missing values.
3. Change the data type of the categorical attributes Cloud9am and Cloud3pm to factor. This step is necessary to use the model.matrix function.
4. Use the model.matrix function to create indicator columns for each categorical attribute.
5. Merge the output of model.matrix with the "Question.11.WAUS" data frame.
6. Tidy up the merged data frame by removing unnecessary columns, retaining only the necessary attributes.
7. Split the tidy data frame into separate training and testing datasets.
8. Fit the neural network model using the training dataset.

The performance of this Artificial Neural Network classifier:

```
> table(observed = Q11.WAUS.test$WarmerTomorrow, predicted = Q11.WAUS.nn.predr)
        predicted
observed  0  1
       0 15 73
       1  5 96
> # Accuracy
> (15+96)/(15+73+5+96)
[1] 0.587
```

Figure 22 shows the confusion matrix and accuracy for Artificial Neural Network classifier

Based on the provided information, the Artificial Neural Network classifier achieved an accuracy of approximately 58.7%, which is the third-lowest among all the classifiers compared in Parts 4, 9, and 10. Therefore, it is evident that this classifier did not perform as well as desired in predicting the attribute WarmerTomorrow.

Comparing it with the model created in Part 10, the performance of the Artificial Neural Network classifier falls short. The model in Part 10 achieved an accuracy that was approximately 8% higher than the accuracy of the Artificial Neural Network classifier. Therefore, the model in Part 10 can be considered superior in terms of predictive performance.

In summary, while the Artificial Neural Network classifier was implemented with selected attributes and adjusted parameters, it did not demonstrate satisfactory accuracy compared to other classifiers, particularly the model in Part 10.