Weather Prediction Analysis

1.
Before grabbing and analyzing the data, I ensured to clear the data ensuring that it only has data with completed cases to allow for a better reading and analysis. Upon doing so, It was discovered that 53% of the days were Warmer than the next day.

```
> isWarmer
[1] 47 53
```

	WarmerTomorrow != is.na(WarmerTomorrow)	n
1	FALSE	288
2	TRUE	329

```
> summary(Warmer)
    MinTemp         MaxTemp
 Min.   :-0.4   Min.   :12.0
 1st Qu.: 7.5   1st Qu.:19.5
 Median :11.7   Median :23.5
 Mean   :11.8   Mean   :24.8
 3rd Qu.:16.1   3rd Qu.:29.9
 Max.   :24.9   Max.   :43.9
```

```
> summary(Cooler)
    MinTemp         MaxTemp
 Min.   :-0.1   Min.   : 9.1
 1st Qu.: 9.1   1st Qu.:17.7
 Median :13.9   Median :21.1
 Mean   :13.4   Mean   :22.0
 3rd Qu.:17.7   3rd Qu.:25.9
 Max.   :25.0   Max.   :44.0
```

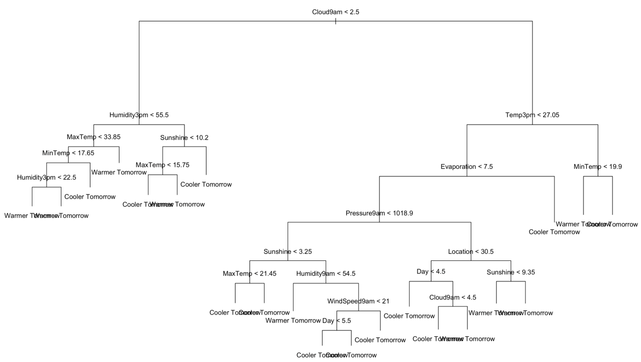
As a base, whenever 1 thinks of the warmth/weather, they would immediately think of looking at the the temperature of the day. I grouped the data to whether it would be warmer the next day and from there I grabbed their summary in order to be able to analyze. Looking at the data when we can see that the data stating that it is warmer the next day tend to have lower MinTemps and higher MaxTemps

2.
Upon grabbing the original data, the initial step was creating a new data frame and omitting rows that are not complete cases. Upon doing so, the initial analysis using all the datas would use the data frame which is fully clean. This would allow the initial analysis to be as accurate as possible.

Later on when certain variables were selected to be used for analysis, I set up a new data frame with the specific columns from the original data and upon doing so, I omitted the rows with any NA values. This would allow for a better analysis as there would be more datas to analyze as compared to using the fully clean data frame.

4.

- Decision Tree




```
#Decision Tree Confusion
> print(t1)
                   Actual_Class
Predicted_Class     0  1
  Cooler Tomorrow 41 36
  Warmer Tomorrow 41 68
> DecTreeAccuracy
[1] 0.586
```

5. Confusion Matrix

- Naive Bayes

```
#Naives Bayes Confusion
> print(tn)
                 actual
predicted          0  1
  Cooler Tomorrow 52 26
  Warmer Tomorrow 30 78
```

```
> NaiveBayesAccuracy
[1] 0.6452
```

- Bagging

```
#Bagging Confusion
> print(WAUSpred.bag$confusion)
                 Observed Class
Predicted Class    0  1
  Cooler Tomorrow 47 27
  Warmer Tomorrow 35 77
```

```
> BaggingAccuracy
[1] 0.6667
```

- Boosting

```
#Boosting Confusion
> print(WAUSpred.boost$confusion)
                 Observed Class
Predicted Class    0  1
  Cooler Tomorrow 43 30
  Warmer Tomorrow 39 74
```

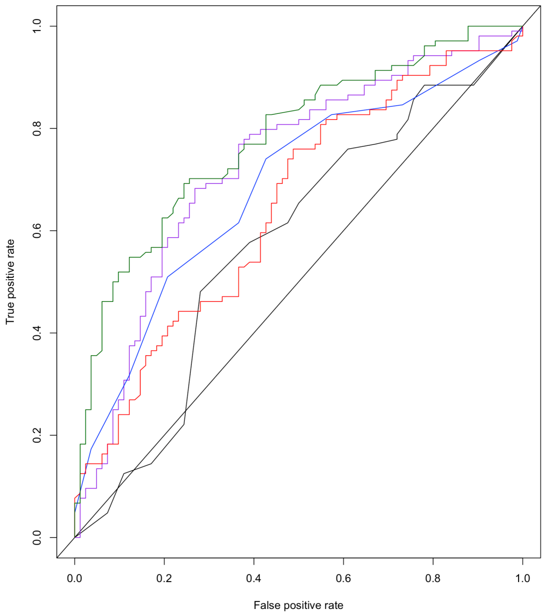
```
> BoostingAccuracy
[1] 0.629
```

- Random Forest

```
#Random Forest Confusion
> print(t3)
                 Actual_Class
Predicted_Class    0  1
  Cooler Tomorrow 55 31
  Warmer Tomorrow 27 73
```

```
> RandomForestAccuracy
[1] 0.6882
```

6.



- Black Line: Decision Tree
- Violet Line: Naive Bayes
- Blue Line: Bagging
- Red Line: Bagging
- Green Line: Random Forest

```
#Decision Tree AUC
> print(as.numeric(DecTreeAUC@y.values))
[1] 0.5772
```

```
#Naive Bayes AUC
> print(as.numeric(NaiveBayesAUC@y.values))
[1] 0.7276
```

```
#Bagging AUC
> print(as.numeric(BaggingAUC@y.values))
[1] 0.6832
```

```
#Boosting AUC
> print(as.numeric(BoostingAUC@y.values))
[1] 0.6468
```

```
#Random Forest AUC
> print(as.numeric(RandomForestAUC@y.values))
[1] 0.7786
```

7.

	Accuracy	AUC
Decision Tree	0.586	0.5772
Naive Bayes	0.6452	0.7276
Bagging	0.6667	0.6832
Boosting	0.629	0.6468
Random Forest	0.6882	0.7786

Upon Looking at the table, it can be seen that the Random Forest model did the best as it has the highest accuracy and the highest AUC from its ROC

8.

```
#Decision Tree Attribute Importance
> print(summary(DecTreeFit))

Classification tree:
tree(formula = WarmerTomorrow ~ ., data = WAUS.train)
Variables actually used in tree construction:
 [1] "Cloud9am"     "Humidity3pm"  "MaxTemp"      "MinTemp"      "Sunshine"     "Temp3pm"      "Evaporation"  "Pressure9am"  "Humidity9am"
[10] "WindSpeed9am" "Day"          "Location"
Number of terminal nodes:  21
Residual mean deviance:  0.863 = 354 / 410
Misclassification error rate: 0.202 = 87 / 431
> cat("\n#Baging Attribute Importance\n")

#Baging Attribute Importance
> print(WAUS.bag$importance)
     Cloud3pm      Cloud9am           Day   Evaporation   Humidity3pm   Humidity9am      Location       MaxTemp       MinTemp
       0.4774        4.3769        2.4777        0.8032       11.7398        2.1396        0.7008        2.4471        3.0365
        Month   Pressure3pm   Pressure9am      Rainfall      Sunshine       Temp3pm       Temp9am    WindDir3pm    WindDir9am
       2.6633        0.3165        9.7440        3.0283        0.4963        2.8825        0.0000       16.0084       16.8340
  WindGustDir WindGustSpeed  WindSpeed3pm  WindSpeed9am          Year
      13.7536        2.0665        1.9239        1.2636        0.8203
> cat("\n#Boosting Attribute Importance\n")

#Boosting Attribute Importance
> print(WAUS.boost$importance)
     Cloud3pm      Cloud9am           Day   Evaporation   Humidity3pm   Humidity9am      Location       MaxTemp       MinTemp
       1.7579        3.4531        1.9654        2.7039        4.7799        1.3394        1.2227        3.8968        3.0356
        Month   Pressure3pm   Pressure9am      Rainfall      Sunshine       Temp3pm       Temp9am    WindDir3pm    WindDir9am
       2.1336        2.2291        4.0906        0.4512        6.2716        2.8372        0.3372       19.7717       15.9521
  WindGustDir WindGustSpeed  WindSpeed3pm  WindSpeed9am          Year
      17.8585        1.8420        0.5534        1.5173        0.0000
```

```
#Random Forest Attribute Importance
> print(WAUS.rf$importance)
              MeanDecreaseGini
Day                      8.817
Month                    6.180
Year                     5.815
Location                 3.121
MinTemp                 12.779
MaxTemp                 13.271
Rainfall                 4.553
Evaporation             10.379
Sunshine                13.992
WindGustDir              6.406
WindGustSpeed            8.471
WindDir9am               9.445
WindDir3pm               7.182
WindSpeed9am             8.123
WindSpeed3pm             7.749
Humidity9am              8.952
Humidity3pm             14.866
Pressure9am             14.217
Pressure3pm             11.094
Cloud9am                 9.733
Cloud3pm                 5.820
Temp9am                  9.422
Temp3pm                 14.155
```

Dec Tree:
- Important Variables:
  - Cloud9am, Humidity3pm, MaxTemp, MinTemp, Sunshine
- Omit:
  - Dates and Location

Bagging:
- Important Variables:
  - WindGustDir(3), Humidity3pm(4), WindDir3pm(2), WindDir9am(1)
- Omit:
  - Dates and Location
  - Cloud3pm, Evaporation, Pressure3pm, Sunshine, Temp9am

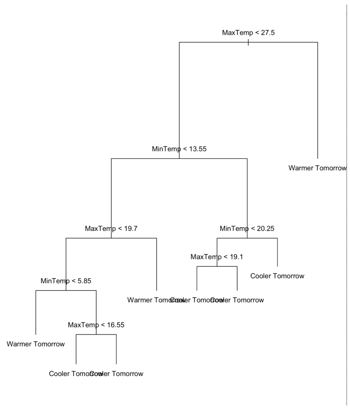
Boosting:
- Important Variables
  - WindGustDir(2), Sunshine, WindDir3pm(1), WindDir9am(3)
- Omit:
  - Dates and Location
  - Rainfall, Temp9am, Windspeed3pm

Random Forest:
- Important Variables
  - MinTemp(5), MaxTemp(3), Sunshine(4), Humidity3pm(1), Pressure9am(2), Pressure3pm(6)
- Omit:
  - Dates and Location
  - Rainfall, Cloud3pm

Dates and Location would need to be omitted in all models as they do not have a relationship with the weather.

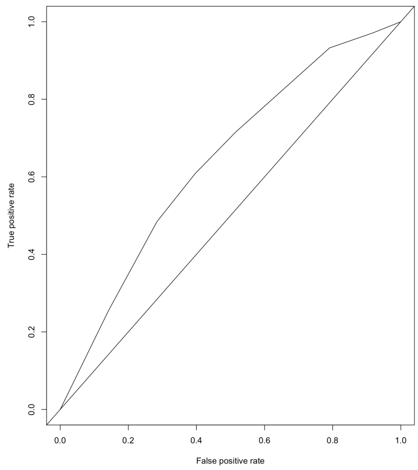
Variables which were stated to Omit will not cause much of an effect on the performance on the model as they have the least contributions. The lower the number is, the lower their relative importance is in the model. Thus proving that the stated variables above being omitted would not cause much effect on the performance if omitted from the model

9.




```
#Decision Tree Confusion
> print(t4)
                 Actual_Class
Predicted_Class    0   1
  Cooler Tomorrow 404 313
  Warmer Tomorrow 265 486

#Decision Tree Accuracy
> SimpleDecTreeAccuracy
[1] 0.6063
```




```
#Decision Tree AUC
> print(as.numeric(SimpleDecTreeAUC@y.values))
[1] 0.6397
```

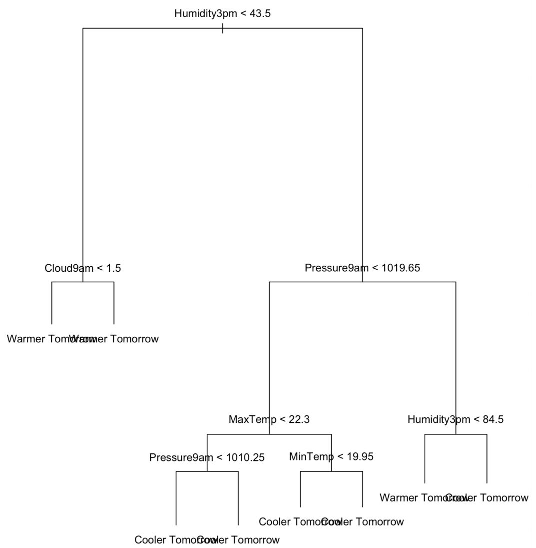
As mentioned above, Temperature tends to be the first thing people think of when in regards to thinking of the warmth and coolness of the day. Thus a simple Decision tree using the MinTemp and MaxTemp variables to determine whether it would be warmer or cooler the following day was used. This would allow anyone to be able to classify the results by looking at the decision tree.

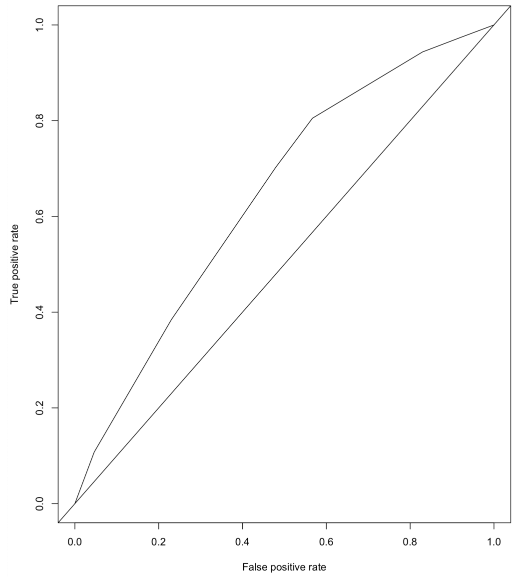
A New data frame was created taking data from the original data frame grabbing the WarmerTomorrow, MaxTemp and MinTemp Variables. Upon doing so it was filtered of all the NA values. Cleaning the new data frame.

As it can be seen, It has an Accuracy of 60.1% and the AUC is 0.6397. This turned out to be even more accurate than the original Decision Tree Created using all the variables. But as predicted, it did not perform better than the other models. The prediction was made upon seeing the initial results of all models seeing that the Decision tree performed the least.

10

- Used the top variables from each confidential model as they impacted their respective models the most.
  - Min and Max temps being taken into account as it is the base on how people would understand if it is warmer the following day or not
  - Top Variables Used were:
    - Decision Tree: Cloud9am, Humidity3pm
    - Bagging: WindGustDir, WindDir9am, WindDir3pm
    - Boosting: WindGustDir, WindDir9am, WindDir3pm
    - Random Forest: Humidity3pm, Pressure9am
- Omitting Dates and the location as it would not be necessary in predictions due to the fact that these variables have no relationships with the weather and its attributes.
- Create new Data frame with Variables mentioned above and from there it was cleaned.






```
#Decision Tree Confusion
> print(t4)
                 Actual_Class
Predicted_Class     0   1
  Cooler Tomorrow 113  58
  Warmer Tomorrow 104 137
```

```
#Decision Tree Accuracy
> ImpDecTreeAccuracy
[1] 0.6068
```

```
#Decision Tree AUC
> print(as.numeric(ImpDecTreeAUC@y.va
[1] 0.6445
```

By using the most important variables from each model, it was assumed that it would create a more accurate analysis as those variables affect their respective models the most and have the highest importance as well.

Upon Looking at the results, the model did better than the simple tree model created above and the initial tree model as well. The model even nearly matches the accuracy and AUC of the Boosting model.

11.

```
        predicted
observed   0   1
       0   1  81
       1   1 103
```

```
#ANN Accuracy
> ANNaccuracy
[1] 0.5591
```

Variables used were similar to the variables used for attempting to create the best Tree-based classifier. Sunshine was used as well as it had a high importance as well. Process of cleaning data was similar to above including the sunshine attribute.

Looking at the results, the ANN has the worst accuracy among all the classification models. It may have performed badly due to the data's not being balanced or it might not have had enough training data to process and analyze.