Classification Tree - Perth

Kathryn Weissman

1/4/2022

Classification Tree: Perth

The goal is to predict if there will be rain the following day.

```
set.seed(1234) # for reproducibility of results
```

Load Train & Test Data

I am loading the same data that was used for the LDA modelling.

```
# Load the data
Ptrain <- read.csv("Train_Test_CSVs/df_Perth_train.csv", stringsAsFactors = T)
Ptest <- read.csv("Train_Test_CSVs/df_Perth_test.csv", stringsAsFactors = T)
Ptrain$Date <- as.Date(Ptrain$Date)
Ptest$Date <- as.Date(Ptest$Date)
```

Summarize Train Data

```
str(Ptrain)
```

```
## 'data.frame':    1431 obs. of  31 variables:
##  $ Date         : Date, format: "2008-07-01" "2008-07-02" ...
##  $ ID           : int  120639 120640 120641 120642 120643 120644 120645 120646 120647 120648 ...
##  $ Year         : int  2008 2008 2008 2008 2008 2008 2008 2008 2008 2008 ...
##  $ Month        : Factor w/ 12 levels "abril","agosto",..: 6 6 6 6 6 6 6 6 6 6 ...
##  $ Day          : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Location     : Factor w/ 1 level "Perth": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Evaporation  : num  0.8 1.8 2.2 1.2 1.4 2.4 0.8 1.4 1.2 2.8 ...
##  $ Sunshine     : num  9.1 7 7.3 4.7 4.9 9.3 9.3 6.9 2.5 1.7 ...
##  $ WindGustDir  : Factor w/ 17 levels "E","ENE","ESE",..: 2 5 5 15 17 6 4 7 3 7 ...
##  $ WindGustSpeed: int  20 22 31 26 44 24 37 24 31 46 ...
##  $ WindDir9am   : Factor w/ 18 levels "calm","E","ENE",..: 1 4 1 7 16 3 6 6 1 6 ...
##  $ WindDir3pm   : Factor w/ 18 levels "calm","E","ENE",..: 2 3 17 8 14 6 7 5 4 7 ...
##  $ WindSpeed9am : int  0 6 0 11 13 4 15 9 0 19 ...
##  $ WindSpeed3pm : int  7 9 4 6 17 7 13 13 9 11 ...
##  $ Humidity9am  : int  97 80 84 93 69 86 72 58 97 79 ...
##  $ Humidity3pm  : int  53 39 71 73 57 41 36 42 64 50 ...
##  $ Pressure9am  : num  1028 1024 1017 1019 1020 ...
##  $ Pressure3pm  : num  1024 1019 1016 1018 1022 ...
##  $ Cloud9am     : int  2 0 1 6 7 0 1 6 7 7 ...
##  $ Cloud3pm     : int  3 6 3 6 5 1 5 5 6 7 ...
##  $ Temp9am      : num  8.5 11.1 12.1 13.2 15.9 6.9 8.7 10.2 12.1 13.4 ...
##  $ Temp3pm      : num  18.1 19.7 17.7 17.7 16 15.5 17.9 19.3 18.7 19 ...
```

```
##  $ RainToday    : Factor w/ 2 levels "No","Yes": 1 1 1 2 2 2 1 1 2 2 ...
##  $ RainTomorrow : Factor w/ 2 levels "No","Yes": 1 1 2 2 2 1 1 2 2 2 ...
##  $ TempRange    : num  16.1 14.3 13.4 9.7 6.9 15.2 17.6 17.2 9.7 9.2 ...
##  $ MaxTemp      : num  18.8 20.7 19.9 19.2 16.4 15.9 18.3 20.4 19.5 20.4 ...
##  $ MinTemp      : num  2.7 6.4 6.5 9.5 9.5 0.7 0.7 3.2 9.8 11.2 ...
##  $ Rainfall     : num  0 0 0.4 1.8 1.8 6.8 0 0 8 4.6 ...
##  $ monthID      : Factor w/ 47 levels "2008-agosto",..: 3 3 3 3 3 3 3 3 3 3 ...
##  $ Season       : Factor w/ 4 levels "autumn","spring",..: 4 4 4 4 4 4 4 4 4 4 ...
##  $ accuRain     : Factor w/ 4 levels "HeavyRain","Mist",..: 3 2 4 4 4 3 3 4 4 4 ...
```

```
summary(Ptrain)
```

```
##       Date                  ID              Year           Month    
##  Min.   :2008-07-01   Min.   :120639   Min.   :2008   agosto   :124  
##  1st Qu.:2009-06-23   1st Qu.:120989   1st Qu.:2009   diciembre:124  
##  Median :2010-06-16   Median :121339   Median :2010   enero    :124  
##  Mean   :2010-06-16   Mean   :121339   Mean   :2010   julio    :124  
##  3rd Qu.:2011-06-08   3rd Qu.:121689   3rd Qu.:2011   marzo    :124  
##  Max.   :2012-05-31   Max.   :122039   Max.   :2012   mayo     :124  
##                       NA's   :30                      (Other)  :687  
##       Day         Location     Evaporation       Sunshine       WindGustDir 
##  Min.   : 1.00   Perth:1431   Min.   : 0.000   Min.   : 0.000   SW     :330  
##  1st Qu.: 8.00                1st Qu.: 2.800   1st Qu.: 6.700   SSW    :211  
##  Median :16.00                Median : 5.000   Median : 9.600   NE     :128  
##  Mean   :15.73                Mean   : 5.761   Mean   : 8.903   WSW    :122  
##  3rd Qu.:23.00                3rd Qu.: 8.400   3rd Qu.:11.500   ENE    : 83  
##  Max.   :31.00                Max.   :17.000   Max.   :13.700   SE     : 63  
##                                                                 (Other):494  
##  WindGustSpeed     WindDir9am    WindDir3pm   WindSpeed9am    WindSpeed3pm  
##  Min.   :15.00   E      :243   SW     :307   Min.   : 0.00   Min.   : 0.00  
##  1st Qu.:30.00   NE     :183   SSW    :181   1st Qu.: 7.00   1st Qu.:11.00  
##  Median :35.00   ENE    :171   WSW    :163   Median :11.00   Median :15.00  
##  Mean   :35.62   SSE    :104   W      :134   Mean   :11.04   Mean   :14.85  
##  3rd Qu.:41.00   ESE    :100   SE     : 76   3rd Qu.:15.00   3rd Qu.:19.00  
##  Max.   :76.00   SE     : 94   ESE    : 74   Max.   :28.00   Max.   :31.00  
##                  (Other):536   (Other):496                                  
##   Humidity9am     Humidity3pm     Pressure9am      Pressure3pm    
##  Min.   :13.00   Min.   : 6.00   Min.   : 996.4   Min.   : 996.8  
##  1st Qu.:49.00   1st Qu.:34.00   1st Qu.:1012.3   1st Qu.:1010.4  
##  Median :59.00   Median :44.00   Median :1016.8   Median :1014.4  
##  Mean   :60.78   Mean   :44.52   Mean   :1017.2   Mean   :1014.8  
##  3rd Qu.:73.00   3rd Qu.:54.00   3rd Qu.:1022.0   3rd Qu.:1019.1  
##  Max.   :99.00   Max.   :97.00   Max.   :1038.8   Max.   :1034.3  
##                                                                   
##     Cloud9am        Cloud3pm       Temp9am         Temp3pm      RainToday 
##  Min.   :0.000   Min.   :0.00   Min.   : 5.60   Min.   : 9.60   No :1167  
##  1st Qu.:1.000   1st Qu.:1.00   1st Qu.:14.45   1st Qu.:19.10   Yes: 264  
##  Median :3.000   Median :3.00   Median :18.30   Median :22.90             
##  Mean   :3.365   Mean   :3.53   Mean   :18.62   Mean   :23.81             
##  3rd Qu.:6.000   3rd Qu.:6.00   3rd Qu.:22.40   3rd Qu.:28.00             
##  Max.   :8.000   Max.   :8.00   Max.   :36.40   Max.   :42.20             
##                                                                           
##  RainTomorrow   TempRange        MaxTemp         MinTemp         Rainfall     
##  No :1166     Min.   : 1.00   Min.   :12.80   Min.   :-0.60   Min.   : 0.000  
##  Yes: 265     1st Qu.: 9.20   1st Qu.:20.30   1st Qu.: 9.10   1st Qu.: 0.000  
```

```
##               Median :12.60   Median :24.20   Median :13.10   Median : 0.000
##               Mean   :12.33   Mean   :25.31   Mean   :12.98   Mean   : 1.764
##               3rd Qu.:15.20   3rd Qu.:29.70   3rd Qu.:17.00   3rd Qu.: 0.200
##               Max.   :24.80   Max.   :42.90   Max.   :28.10   Max.   :57.000
##
##            monthID        Season         accuRain
##  2008-agosto   :  31   autumn:368   HeavyRain:  49
##  2008-diciembre:  31   spring:364   Mist     : 104
##  2008-julio    :  31   summer:361   NoRain   :1054
##  2008-octubre  :  31   winter:338   Rain     : 224
##  2009-agosto   :  31
##  2009-diciembre:  31
##  (Other)       :1245
```

Summarize Test Data

```
summary(Ptest)
```

```
##       Date                  ID              Year           Month
##  Min.   :2012-06-01   Min.   :122040   Min.   :2012   agosto   : 31
##  1st Qu.:2012-08-31   1st Qu.:122116   1st Qu.:2012   diciembre: 31
##  Median :2012-11-30   Median :122193   Median :2012   enero    : 31
##  Mean   :2012-11-30   Mean   :122193   Mean   :2012   julio    : 31
##  3rd Qu.:2013-03-01   3rd Qu.:122270   3rd Qu.:2013   junio    : 31
##  Max.   :2013-06-01   Max.   :122346   Max.   :2013   marzo    : 31
##                       NA's   :59                      (Other)  :180
##       Day         Location    Evaporation       Sunshine       WindGustDir
##  Min.   : 1.00   Perth:366   Min.   : 0.000   Min.   : 0.000   unkn   : 59
##  1st Qu.: 8.00               1st Qu.: 2.600   1st Qu.: 6.825   SW     : 46
##  Median :16.00               Median : 4.800   Median : 9.400   SSW    : 43
##  Mean   :15.68               Mean   : 5.573   Mean   : 8.731   WSW    : 36
##  3rd Qu.:23.00               3rd Qu.: 8.000   3rd Qu.:11.200   NE     : 35
##  Max.   :31.00               Max.   :16.000   Max.   :13.400   NW     : 23
##                                                                (Other):124
##  WindGustSpeed     WindDir9am     WindDir3pm    WindSpeed9am    WindSpeed3pm
##  Min.   :13.00   unkn   : 59   SW     : 62   Min.   : 0.00   Min.   : 2.00
##  1st Qu.:28.00   E      : 41   unkn   : 59   1st Qu.: 7.00   1st Qu.:11.00
##  Median :35.00   NE     : 40   SSW    : 32   Median :11.00   Median :15.00
##  Mean   :34.84   NNE    : 33   WSW    : 32   Mean   :10.68   Mean   :14.41
##  3rd Qu.:41.00   ENE    : 30   W      : 31   3rd Qu.:13.00   3rd Qu.:19.00
##  Max.   :74.00   ESE    : 22   WNW    : 22   Max.   :30.00   Max.   :30.00
##                  (Other):141   (Other):128
##   Humidity9am     Humidity3pm      Pressure9am      Pressure3pm
##  Min.   :15.00   Min.   :11.00   Min.   : 996.2   Min.   : 991.9
##  1st Qu.:49.25   1st Qu.:38.00   1st Qu.:1012.5   1st Qu.:1010.6
##  Median :61.50   Median :49.00   Median :1016.8   Median :1014.5
##  Mean   :62.65   Mean   :48.25   Mean   :1017.1   Mean   :1014.8
##  3rd Qu.:77.00   3rd Qu.:58.00   3rd Qu.:1021.5   3rd Qu.:1018.7
##  Max.   :99.00   Max.   :93.00   Max.   :1034.5   Max.   :1031.2
##
##     Cloud9am        Cloud3pm        Temp9am         Temp3pm      RainToday
##  Min.   :0.000   Min.   :0.000   Min.   : 8.00   Min.   :11.20   No :288
##  1st Qu.:1.000   1st Qu.:1.000   1st Qu.:14.80   1st Qu.:19.30   Yes: 78
##  Median :5.000   Median :5.000   Median :18.80   Median :22.85
```

```
##  Mean   :4.249   Mean   :4.418   Mean   :18.99   Mean   :23.62
##  3rd Qu.:7.000   3rd Qu.:7.000   3rd Qu.:22.90   3rd Qu.:27.30
##  Max.   :8.000   Max.   :8.000   Max.   :33.90   Max.   :41.80
##
##  RainTomorrow   TempRange         MaxTemp         MinTemp
##  No :289      Min.   : 2.900   Min.   :14.60   Min.   : 0.400
##  Yes: 77      1st Qu.: 8.625   1st Qu.:20.70   1st Qu.: 9.575
##               Median :12.100   Median :24.15   Median :13.200
##               Mean   :12.185   Mean   :25.46   Mean   :13.274
##               3rd Qu.:15.400   3rd Qu.:29.55   3rd Qu.:17.200
##               Max.   :24.100   Max.   :42.10   Max.   :27.300
##
##     Rainfall                 monthID       Season        accuRain
##  Min.   : 0.000   2012-agosto   : 31   autumn:92   HeavyRain: 11
##  1st Qu.: 0.000   2012-diciembre: 31   spring:91   Mist     : 31
##  Median : 0.000   2012-julio    : 31   summer:90   NoRain   :254
##  Mean   : 1.869   2012-octubre  : 31   winter:93   Rain     : 70
##  3rd Qu.: 0.400   2013-enero    : 31
##  Max.   :43.600   2013-marzo    : 31
##                   (Other)       :180
```

Compare Target Variables for Train and Test Data

It is important that our training data and testing data have similar characteristics to check the accuracy of our model.

```r
print ("Percentage of Days with Rain Tomorrow in Train Data")
```

```
## [1] "Percentage of Days with Rain Tomorrow in Train Data"
```

```r
round(prop.table(table(Ptrain$RainTomorrow))*100,1)
```

```
##
##   No  Yes
## 81.5 18.5
```

```r
print ("Percentage of Days with Rain Tomorrow in Test Data")
```

```
## [1] "Percentage of Days with Rain Tomorrow in Test Data"
```

```r
round(prop.table(table(Ptest$RainTomorrow))*100,1)
```

```
##
##  No Yes
##  79  21
```

The seasons are mostly balanced between the training and testing data. The testing data has a slightly larger proportion of winter days. This is due to the span of dates in the training data not including one of the full years. The training data spans from July 1, 2008 to May 31, 2012. We don't have data for the month of June 2008 to include in the training set.

```r
print ("Percentage of Days in each Season in Train Data")
```

```
## [1] "Percentage of Days in each Season in Train Data"
```

```r
round(prop.table(table(Ptrain$Season))*100,1)
```

```
##
## autumn spring summer winter
```

```
##   25.7   25.4   25.2   23.6
```

```r
print ("Percentage of Days in each Season in Test Data")
```

```
## [1] "Percentage of Days in each Season in Test Data"
```

```r
round(prop.table(table(Ptest$Season))*100,1)
```

```
##
## autumn spring summer winter
##   25.1   24.9   24.6   25.4
```

Classification Tree

https://cran.r-project.org/web/packages/rpart/vignettes/longintro.pdf

"The rpart programs build classification or regression models of a very general structure using a two stage procedure; the resulting models can be represented as binary trees."

We use two different sets of modeling variables to see if there is a difference in the performance of the model for classifying whether or not there will be rain tomorrow.

```r
# We use two different sets of variables for the model to consider

# Set 1 includes "RainToday" and "TempRange"
modeling_vars1 <- c("Evaporation", "Sunshine", "WindGustSpeed", "WindSpeed9am",
                    "WindSpeed3pm", "Humidity9am", "Humidity3pm", "Pressure9am",
                    "Pressure3pm", "Cloud9am", "Cloud3pm", "TempRange",
                    "RainToday", "Season", "RainTomorrow")

# Set 2 includes all temperature variables and "Rainfall" instead of "RainToday"
modeling_vars2 <- c("Evaporation", "Sunshine", "WindGustSpeed", "WindSpeed9am",
                    "WindSpeed3pm", "Humidity9am", "Humidity3pm", "Pressure9am",
                    "Pressure3pm",  "Cloud9am", "Cloud3pm", "Temp9am", "Temp3pm",
                    "TempRange", "MaxTemp", "MinTemp", "Rainfall", "Season",
                    "RainTomorrow")

train1 <- Ptrain[,modeling_vars1]
test1 <- Ptest[,modeling_vars1]

train2 <- Ptrain[,modeling_vars2]
test2 <- Ptest[,modeling_vars2]
```

SMOTE algorithm for unbalanced classification problems

From the library {performanceEstimation}

"This function handles unbalanced classification problems using the SMOTE method. Namely, it can generate a new"SMOTEd" data set that addresses the class unbalance problem."

Balanced Training Sets 1 and 2 have different observations due to the nearest neighbors defined by the subset of variables contained in each training data set.

```r
set.seed(1234) # for reproducibility of results
# Create balanced training data sets
trainBal1 <- smote(RainTomorrow ~., train1, perc.over = 2, k = 5, perc.under = 2)
trainBal2 <- smote(RainTomorrow ~., train2, perc.over = 2, k = 5, perc.under = 2)
```

```
print("Training Data: Count of Rain Tomorrow")
```

```
## [1] "Training Data: Count of Rain Tomorrow"
```

```
(table(Ptrain$RainTomorrow))
```

```
##
##   No  Yes
## 1166  265
```

```
print("Balanced Training 1 Data: Count of Rain Tomorrow")
```

```
## [1] "Balanced Training 1 Data: Count of Rain Tomorrow"
```

```
(table(trainBal1$RainTomorrow))
```

```
##
##   No  Yes
## 1060  795
```

```
print("Balanced Training 1 Data: Percent of Days with Rain Tomorrow")
```

```
## [1] "Balanced Training 1 Data: Percent of Days with Rain Tomorrow"
```

```
round(prop.table((table(trainBal1$RainTomorrow)))*100,2)
```

```
##
##    No   Yes
## 57.14 42.86
```

```
print("Balanced Training 2 Data: Count of Rain Tomorrow")
```

```
## [1] "Balanced Training 2 Data: Count of Rain Tomorrow"
```

```
(table(trainBal2$RainTomorrow))
```

```
##
##   No  Yes
## 1060  795
```

```
print("Balanced Training 2 Data: Percent of Days with Rain Tomorrow")
```

```
## [1] "Balanced Training 2 Data: Percent of Days with Rain Tomorrow"
```

```
round(prop.table(table(trainBal2$RainTomorrow))*100,2)
```

```
##
##    No   Yes
## 57.14 42.86
```

```
print("Balanced Training 1 Data: Percent of Days in each Season")
```

```
## [1] "Balanced Training 1 Data: Percent of Days in each Season"
```

```
round(prop.table(table(trainBal1$Season))*100,1)
```

```
##
## autumn spring summer winter
##   21.2   27.5   20.8   30.5
```

```
print("Balanced Training 2 Data: Percent of Days in each Season")
```

```
## [1] "Balanced Training 2 Data: Percent of Days in each Season"
```

```r
round(prop.table(table(trainBal2$Season))*100,1)
```

```
##
## autumn spring summer winter
##   22.9   25.7   19.6   31.9
```

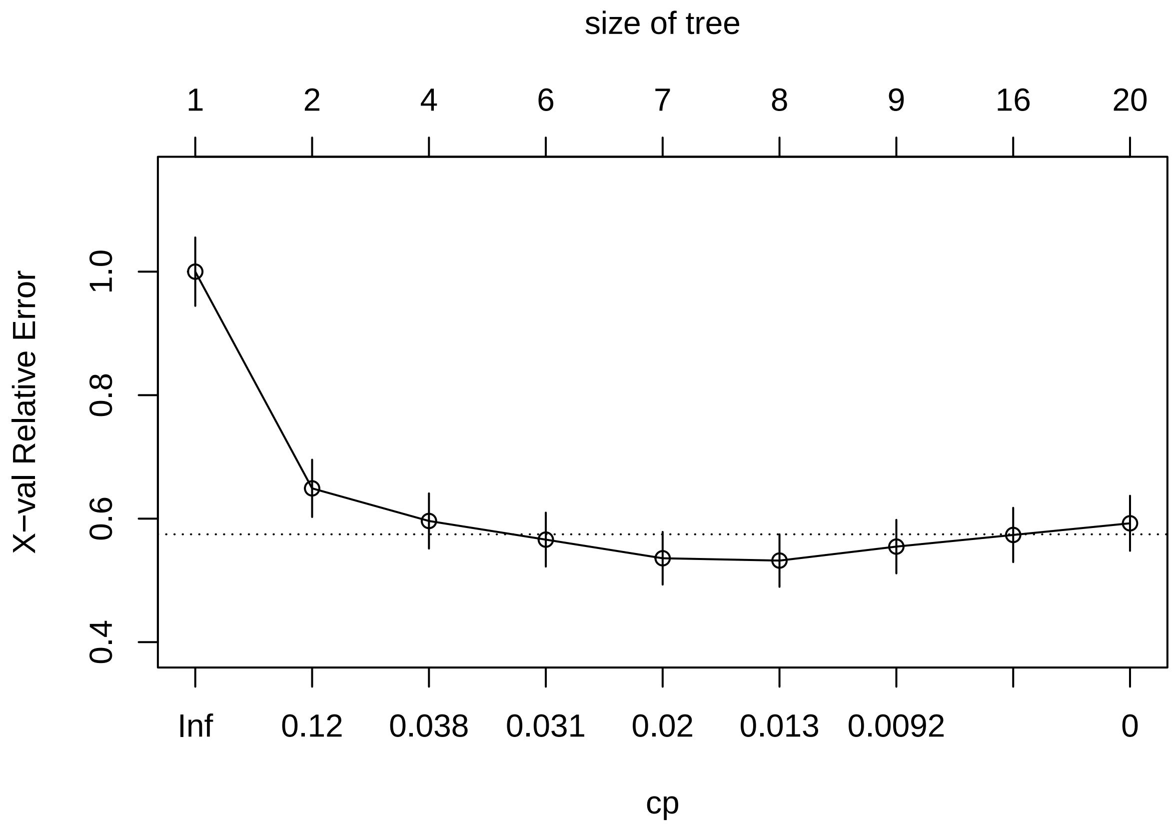
Using Fitting & Pruning Strategy shown in Lab

```r
# Best strategy for tree fitting, cp = 0
set.seed(1234) # for reproducibility of results
treeFit1 <- rpart(RainTomorrow ~., data = train1, method = "class", cp = 0)
printcp(treeFit1)
```

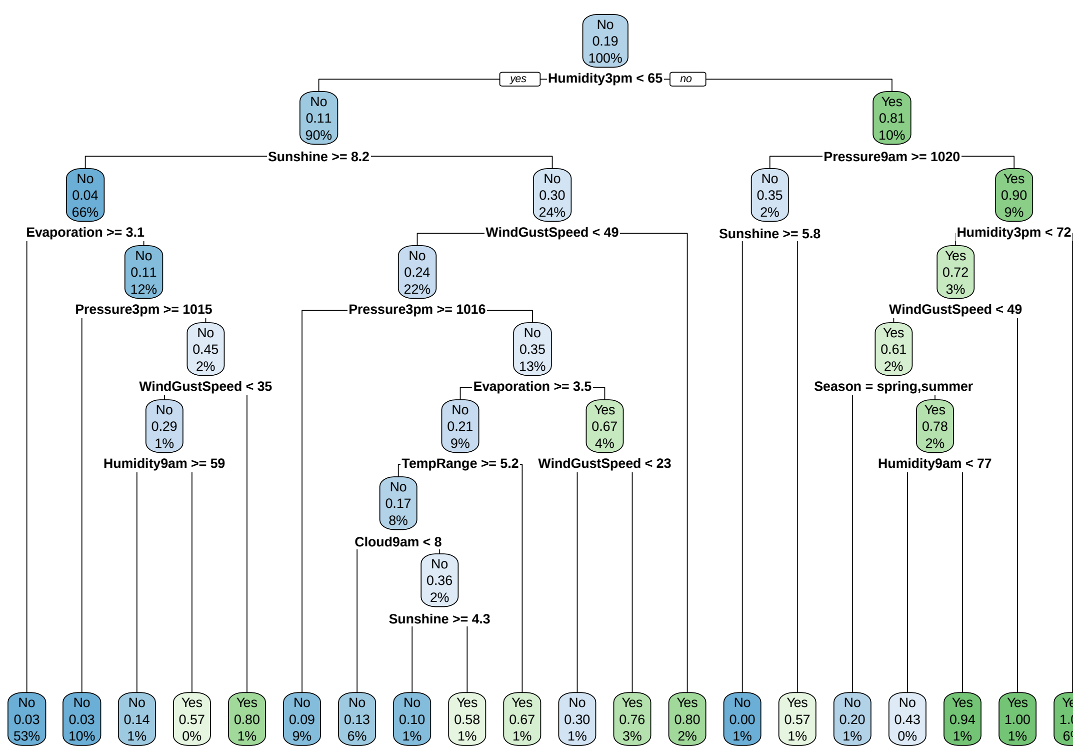
First Set of Variables using Imbalanced Training Data

```
##
## Classification tree:
## rpart(formula = RainTomorrow ~ ., data = train1, method = "class",
##     cp = 0)
##
## Variables actually used in tree construction:
##  [1] Cloud9am      Evaporation   Humidity3pm   Humidity9am   Pressure3pm
##  [6] Pressure9am   Season        Sunshine      TempRange     WindGustSpeed
##
## Root node error: 265/1431 = 0.18519
##
## n= 1431
##
##          CP nsplit rel error  xerror     xstd
## 1 0.3509434      0   1.00000 1.00000 0.055451
## 2 0.0396226      1   0.64906 0.64906 0.046421
## 3 0.0358491      3   0.56981 0.59623 0.044738
## 4 0.0264151      5   0.49811 0.56604 0.043727
## 5 0.0150943      6   0.47170 0.53585 0.042678
## 6 0.0113208      7   0.45660 0.53208 0.042544
## 7 0.0075472      8   0.44528 0.55472 0.043339
## 8 0.0037736     15   0.39245 0.57358 0.043984
## 9 0.0000000     19   0.37736 0.59245 0.044614
```

```r
plotcp(treeFit1)
```




```r
rpart.plot(treeFit1)
```



```r
xerror <- treeFit1$cptable[,"xerror"]
imin.xerror <- which.min(xerror)
```

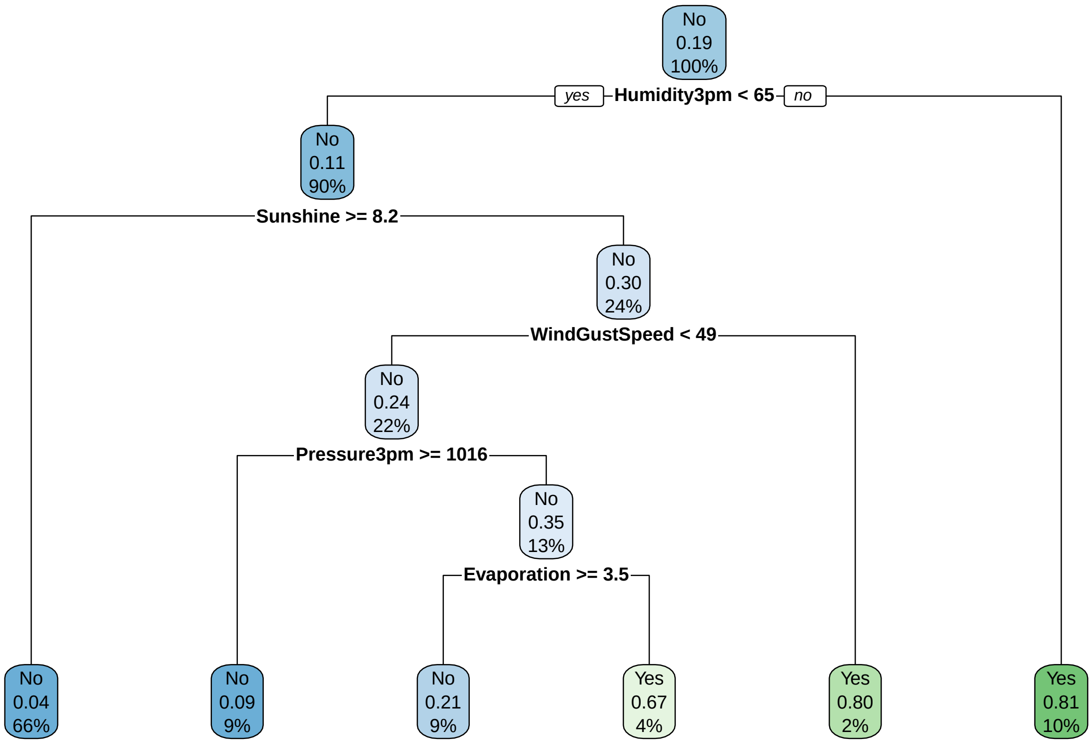
```r
treeFit1$cptable[imin.xerror, ]
```

```
##         CP     nsplit  rel error     xerror       xstd
## 0.01132075 7.00000000 0.45660377 0.53207547 0.04254404
```

```r
upper.xerror <- xerror[imin.xerror] + treeFit1$cptable[imin.xerror, "xstd"]
icp <- min(which(xerror <= upper.xerror))
cp <- treeFit1$cptable[icp, "CP"]
```

The pruned tree using imbalanced data is easy to understand, and uses five variables to make the splits.

```r
tree1 <- prune(treeFit1, cp = cp)
rpart.plot(tree1)
```




```r
#Classification Rules
rpart.rules(tree1, style = "tall")
```

```
## RainTomorrow is 0.04 when
##     Humidity3pm < 65
##     Sunshine >= 8.2
##
## RainTomorrow is 0.09 when
##     Humidity3pm < 65
##     Sunshine < 8.2
##     WindGustSpeed < 49
##     Pressure3pm >= 1016
##
## RainTomorrow is 0.21 when
##     Humidity3pm < 65
##     Sunshine < 8.2
##     WindGustSpeed < 49
##     Pressure3pm < 1016
```

```
##     Evaporation >= 3.5
##
## RainTomorrow is 0.67 when
##     Humidity3pm < 65
##     Sunshine < 8.2
##     WindGustSpeed < 49
##     Pressure3pm < 1016
##     Evaporation < 3.5
##
## RainTomorrow is 0.80 when
##     Humidity3pm < 65
##     Sunshine < 8.2
##     WindGustSpeed >= 49
##
## RainTomorrow is 0.81 when
##     Humidity3pm >= 65
```

```
#Checking important variables
importance1 <- tree1$variable.importance
importance1 <- round(100*importance1/sum(importance1), 1)
importance1[importance1 >= 1]
```

```
##   Humidity3pm      Sunshine WindGustSpeed   Evaporation     TempRange
##          41.9          13.8           7.8           6.0           5.2
##   Pressure3pm   Pressure9am        Season      Cloud9am  WindSpeed3pm
##           4.6           3.7           3.6           3.3           3.2
##      Cloud3pm     RainToday   Humidity9am
##           2.6           2.2           1.2
```

Confusion Matrix

Help for Confusion Matrix: https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62

Recall,Precision and Accuracy should be high as possible

Balanced Accuracy represents area under ROC.

Although the accuracy is high, 90%, the sensitivity is lower, at 70%, which is how well the model predicts it will rain on a rainy day. Since the data is imbalanced, we should try using SMOTE sampling for the training data to see if it improves the performance of the model.

```
#Evaluation
#Confusion matrix-train
pred_train1 <- predict(tree1, train1, type = 'class') # using train data
#Make sure to state positive class in the confusion matrix.
confusionMatrix(pred_train1, train1$RainTomorrow, positive="Yes")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   No  Yes
##        No  1113   79
##        Yes   53  186
##
##                Accuracy : 0.9078
##                  95% CI : (0.8916, 0.9223)
##     No Information Rate : 0.8148
```

```
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.6823
##
##  Mcnemar's Test P-Value : 0.02956
##
##             Sensitivity : 0.7019
##             Specificity : 0.9545
##          Pos Pred Value : 0.7782
##          Neg Pred Value : 0.9337
##              Prevalence : 0.1852
##          Detection Rate : 0.1300
##    Detection Prevalence : 0.1670
##       Balanced Accuracy : 0.8282
##
##        'Positive' Class : Yes
##
```

The sensitivity is very low, which is how accurate the predictions are for rainy days. Since the data is imbalanced, we should try using SMOTE sampling for the training data to see if it improves the performance of the model.

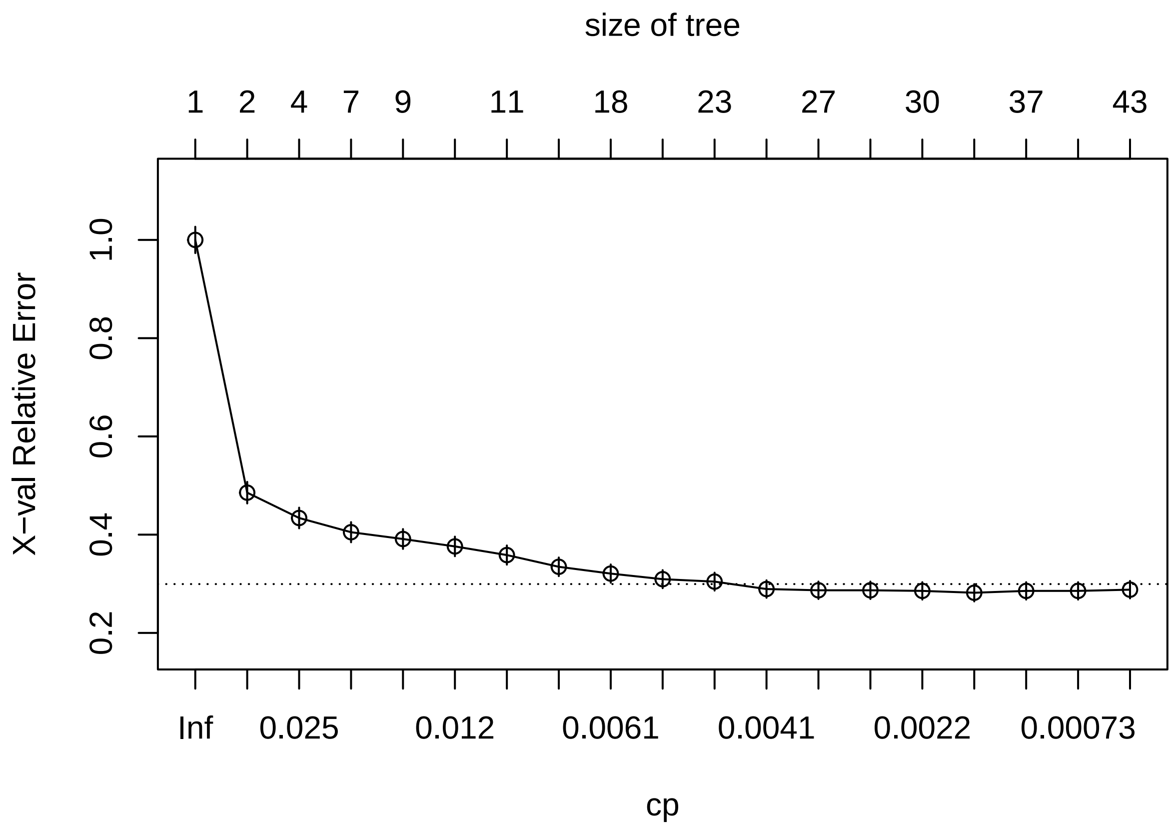
First Set of Variables on Balnced Training Data using SMOTE This is the model that performs the best when evaluating it on the test set.

```r
# Best strategy for tree fitting, start with cp = 0, then prune.
set.seed(1234) # for reproducibility of results
treeFitBal1 <- rpart(RainTomorrow ~., data = trainBal1, method = "class", cp = 0)
printcp(treeFitBal1)
```

```
##
## Classification tree:
## rpart(formula = RainTomorrow ~ ., data = trainBal1, method = "class",
##     cp = 0)
##
## Variables actually used in tree construction:
##  [1] Cloud3pm      Cloud9am      Evaporation   Humidity3pm   Pressure3pm
##  [6] Pressure9am   Season        Sunshine      TempRange     WindGustSpeed
## [11] WindSpeed3pm  WindSpeed9am
##
## Root node error: 795/1855 = 0.42857
##
## n= 1855
##
##            CP nsplit rel error  xerror     xstd
## 1  0.52452830      0   1.00000 1.00000 0.026810
## 2  0.04150943      1   0.47547 0.48553 0.021992
## 3  0.01509434      3   0.39245 0.43396 0.021079
## 4  0.01320755      6   0.34717 0.40503 0.020519
## 5  0.01257862      8   0.32075 0.39119 0.020238
## 6  0.01132075      9   0.30818 0.37610 0.019921
## 7  0.00796646     10   0.29686 0.35849 0.019536
## 8  0.00628931     13   0.27296 0.33459 0.018987
## 9  0.00587002     17   0.24528 0.32075 0.018655
## 10 0.00503145     21   0.21761 0.30943 0.018374
```

```
## 11 0.00440252     22    0.21258 0.30440 0.018247
## 12 0.00377358     24    0.20377 0.28931 0.017855
## 13 0.00314465     26    0.19623 0.28679 0.017788
## 14 0.00251572     28    0.18994 0.28679 0.017788
## 15 0.00188679     29    0.18742 0.28553 0.017754
## 16 0.00150943     31    0.18365 0.28176 0.017653
## 17 0.00125786     36    0.17610 0.28553 0.017754
## 18 0.00041929     39    0.17233 0.28553 0.017754
## 19 0.00000000     42    0.17107 0.28805 0.017821
```

```
plotcp(treeFitBal1)
```



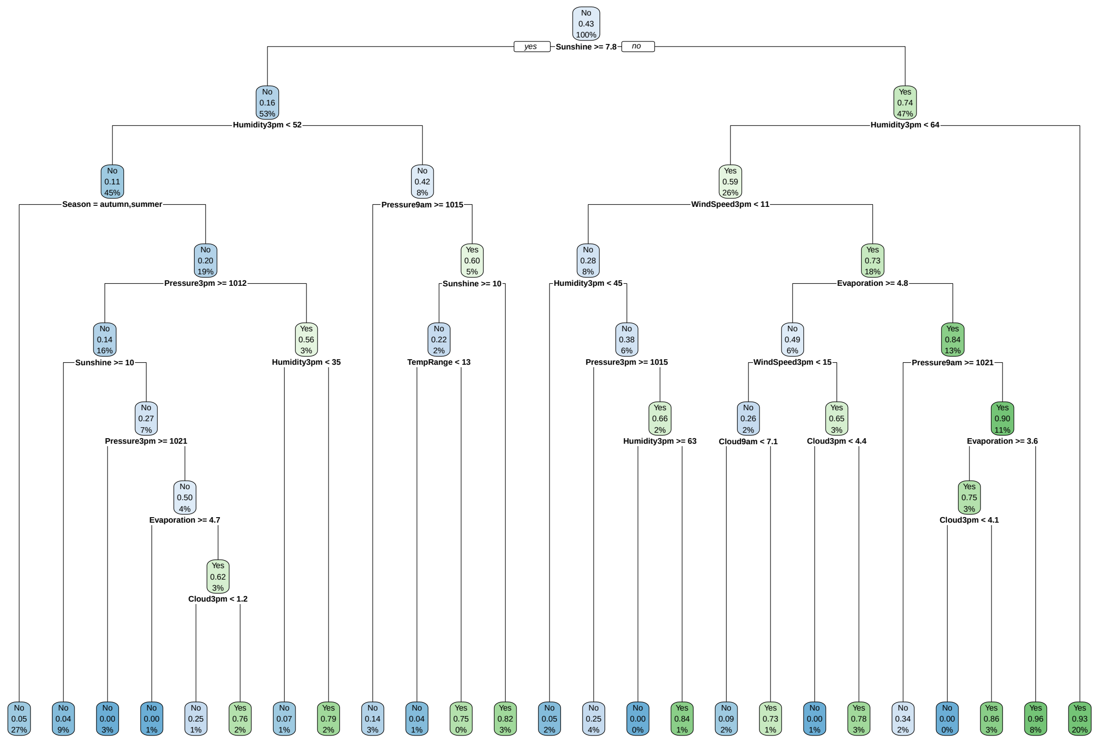

```
#rpart.plot(treeFitBal1)
```

```
# Find the cp with lowest error, then prune.
xerror <- treeFitBal1$cptable[,"xerror"]
imin.xerror <- which.min(xerror)
treeFitBal1$cptable[imin.xerror, ]
```

```
##           CP      nsplit    rel error       xerror         xstd
##  0.001509434 31.000000000  0.183647799  0.281761006  0.017652731
```

```
upper.xerror <- xerror[imin.xerror] + treeFitBal1$cptable[imin.xerror, "xstd"]
icp <- min(which(xerror <= upper.xerror))
cp <- treeFitBal1$cptable[icp, "CP"]
```

```
# prune using cp
treeBal1 <- prune(treeFitBal1, cp = cp)
rpart.plot(treeBal1)
```




```r
#Classification Rules
rpart.rules(treeBal1, style = "tall")
```

```
## RainTomorrow is 0.00 when
##     Sunshine is 7.8 to 10.0
##     Humidity3pm < 52
##     Pressure3pm >= 1021
##     Season is spring or winter
##
## RainTomorrow is 0.00 when
##     Sunshine is 7.8 to 10.0
##     Humidity3pm < 52
##     Evaporation >= 4.7
##     Pressure3pm is 1012 to 1021
##     Season is spring or winter
##
## RainTomorrow is 0.00 when
##     Sunshine < 7.8
##     Humidity3pm is 63 to 64
##     WindSpeed3pm < 11
##     Pressure3pm < 1015
##
## RainTomorrow is 0.00 when
##     Sunshine < 7.8
##     Humidity3pm < 64
##     WindSpeed3pm >= 15
##     Evaporation >= 4.8
##     Cloud3pm < 4.4
##
## RainTomorrow is 0.00 when
##     Sunshine < 7.8
```

```
##     Humidity3pm < 64
##     WindSpeed3pm >= 11
##     Evaporation is 3.6 to 4.8
##     Pressure9am < 1021
##     Cloud3pm < 4.1
##
## RainTomorrow is 0.04 when
##     Sunshine >= 10.0
##     Humidity3pm < 52
##     Pressure3pm >= 1012
##     Season is spring or winter
##
## RainTomorrow is 0.04 when
##     Sunshine >= 10.2
##     Humidity3pm >= 52
##     Pressure9am < 1015
##     TempRange < 13
##
## RainTomorrow is 0.05 when
##     Sunshine < 7.8
##     Humidity3pm < 45
##     WindSpeed3pm < 11
##
## RainTomorrow is 0.05 when
##     Sunshine >= 7.8
##     Humidity3pm < 52
##     Season is autumn or summer
##
## RainTomorrow is 0.07 when
##     Sunshine >= 7.8
##     Humidity3pm < 35
##     Pressure3pm < 1012
##     Season is spring or winter
##
## RainTomorrow is 0.09 when
##     Sunshine < 7.8
##     Humidity3pm < 64
##     WindSpeed3pm is 11 to 15
##     Evaporation >= 4.8
##     Cloud9am < 7.1
##
## RainTomorrow is 0.14 when
##     Sunshine >= 7.8
##     Humidity3pm >= 52
##     Pressure9am >= 1015
##
## RainTomorrow is 0.25 when
##     Sunshine is 7.8 to 10.0
##     Humidity3pm < 52
##     Evaporation < 4.7
##     Pressure3pm is 1012 to 1021
##     Season is spring or winter
##     Cloud3pm < 1.2
##
```

```
## RainTomorrow is 0.25 when
##     Sunshine < 7.8
##     Humidity3pm is 45 to 64
##     WindSpeed3pm < 11
##     Pressure3pm >= 1015
##
## RainTomorrow is 0.34 when
##     Sunshine < 7.8
##     Humidity3pm < 64
##     WindSpeed3pm >= 11
##     Evaporation < 4.8
##     Pressure9am >= 1021
##
## RainTomorrow is 0.73 when
##     Sunshine < 7.8
##     Humidity3pm < 64
##     WindSpeed3pm is 11 to 15
##     Evaporation >= 4.8
##     Cloud9am >= 7.1
##
## RainTomorrow is 0.75 when
##     Sunshine >= 10.2
##     Humidity3pm >= 52
##     Pressure9am < 1015
##     TempRange >= 13
##
## RainTomorrow is 0.76 when
##     Sunshine is 7.8 to 10.0
##     Humidity3pm < 52
##     Evaporation < 4.7
##     Pressure3pm is 1012 to 1021
##     Season is spring or winter
##     Cloud3pm >= 1.2
##
## RainTomorrow is 0.78 when
##     Sunshine < 7.8
##     Humidity3pm < 64
##     WindSpeed3pm >= 15
##     Evaporation >= 4.8
##     Cloud3pm >= 4.4
##
## RainTomorrow is 0.79 when
##     Sunshine >= 7.8
##     Humidity3pm is 35 to 52
##     Pressure3pm < 1012
##     Season is spring or winter
##
## RainTomorrow is 0.82 when
##     Sunshine is 7.8 to 10.2
##     Humidity3pm >= 52
##     Pressure9am < 1015
##
## RainTomorrow is 0.84 when
##     Sunshine < 7.8
```

```
##     Humidity3pm is 45 to 63
##     WindSpeed3pm < 11
##     Pressure3pm < 1015
##
## RainTomorrow is 0.86 when
##     Sunshine < 7.8
##     Humidity3pm < 64
##     WindSpeed3pm >= 11
##     Evaporation is 3.6 to 4.8
##     Pressure9am < 1021
##     Cloud3pm >= 4.1
##
## RainTomorrow is 0.93 when
##     Sunshine < 7.8
##     Humidity3pm >= 64
##
## RainTomorrow is 0.96 when
##     Sunshine < 7.8
##     Humidity3pm < 64
##     WindSpeed3pm >= 11
##     Evaporation < 3.6
##     Pressure9am < 1021
```

In the Imbalanced Training Data for the first set of variables, Humidity3pm, Sunshine, WindGustSpeed, Evaporation, and TempRange were the 5 most important variables. For the balanced training data, Cloud3pm, and Cloud9am are more important than Evaporation and WindGustSpeed.

```r
#Checking important variables
importanceBal1 <- treeBal1$variable.importance
importanceBal1 <- round(100*importanceBal1/sum(importanceBal1), 1)
importanceBal1[importanceBal1 >= 1]
```

```
##      Sunshine   Humidity3pm      Cloud3pm     TempRange       Cloud9am
##          19.1          15.2          13.1          11.1           11.0
##   Humidity9am   Pressure9am   Pressure3pm  WindSpeed3pm   Evaporation
##           9.5           4.6           4.4           3.6            3.5
## WindGustSpeed        Season  WindSpeed9am
##           2.1           1.5           1.5
```

Using the model created by balancing the data produces better results when checking predictions on the training data. Accuracy decreased from 90.8% to 88%, however Sensitivity improved from 70.2% to 87.2%. Specificity decreased from 95.5% to 88.3%, but Balanced Accuracy (Area under ROC) improved from 82.8% to 87.7%

```r
#Evaluation of model created with balanced data
#Confusion matrix-train
pred_trainBal1 <- predict(treeBal1, train1, type = 'class') # using original train data
#Make sure to state positive class in the confusion matrix.
confusionMatrix(pred_trainBal1, train1$RainTomorrow, positive="Yes")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   No  Yes
##        No  1029   34
##        Yes  137  231
##
```

```
##                Accuracy : 0.8805
##                  95% CI : (0.8626, 0.8969)
##     No Information Rate : 0.8148
##     P-Value [Acc > NIR] : 1.014e-11
##
##                   Kappa : 0.6557
##
##  Mcnemar's Test P-Value : 6.184e-15
##
##             Sensitivity : 0.8717
##             Specificity : 0.8825
##          Pos Pred Value : 0.6277
##          Neg Pred Value : 0.9680
##              Prevalence : 0.1852
##          Detection Rate : 0.1614
##    Detection Prevalence : 0.2572
##       Balanced Accuracy : 0.8771
##
##        'Positive' Class : Yes
##
```

Some of our key metrics decrease slightly when expanded to the test set, which could be an indicator of overfitting to the training data, but it is not too different.

Accuracy decreased from 88% to 83.6%, Sensitivity decreased from 87.2% to 75.3%, Specificity decreased from 88 to 85.8%, and Balanced Accuracy decreased from 87.7 to 80.6%.

```
#Test Set Evaluation of Balanced Model 1
#Confusion matrix-test
pred_testBal1 <- predict(treeBal1, test1, type = 'class') # using testing data
confusionMatrix(pred_testBal1, test1$RainTomorrow, positive="Yes")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  No Yes
##        No  248  19
##        Yes  41  58
##
##                Accuracy : 0.8361
##                  95% CI : (0.7941, 0.8725)
##     No Information Rate : 0.7896
##     P-Value [Acc > NIR] : 0.015202
##
##                   Kappa : 0.5534
##
##  Mcnemar's Test P-Value : 0.006706
##
##             Sensitivity : 0.7532
##             Specificity : 0.8581
##          Pos Pred Value : 0.5859
##          Neg Pred Value : 0.9288
##              Prevalence : 0.2104
##          Detection Rate : 0.1585
##    Detection Prevalence : 0.2705
##       Balanced Accuracy : 0.8057
```

```
##
##        'Positive' Class : Yes
##
```

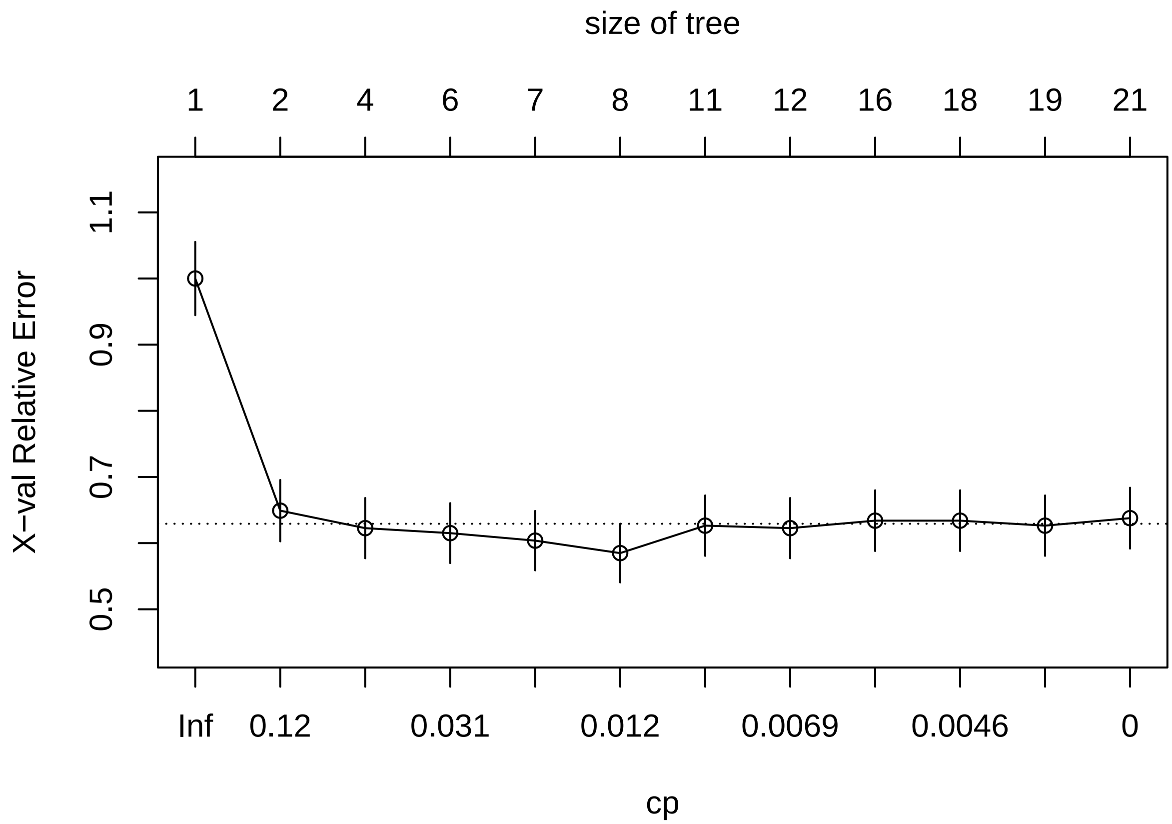
Second Set of Variables

Imbalanced Data This set includes more variables than the first set. Set 1 included "RainToday", but set 2 includes "Rainfall". Set 1 included "TempRange", but set 2 includes all temperature related variables including TempRange.

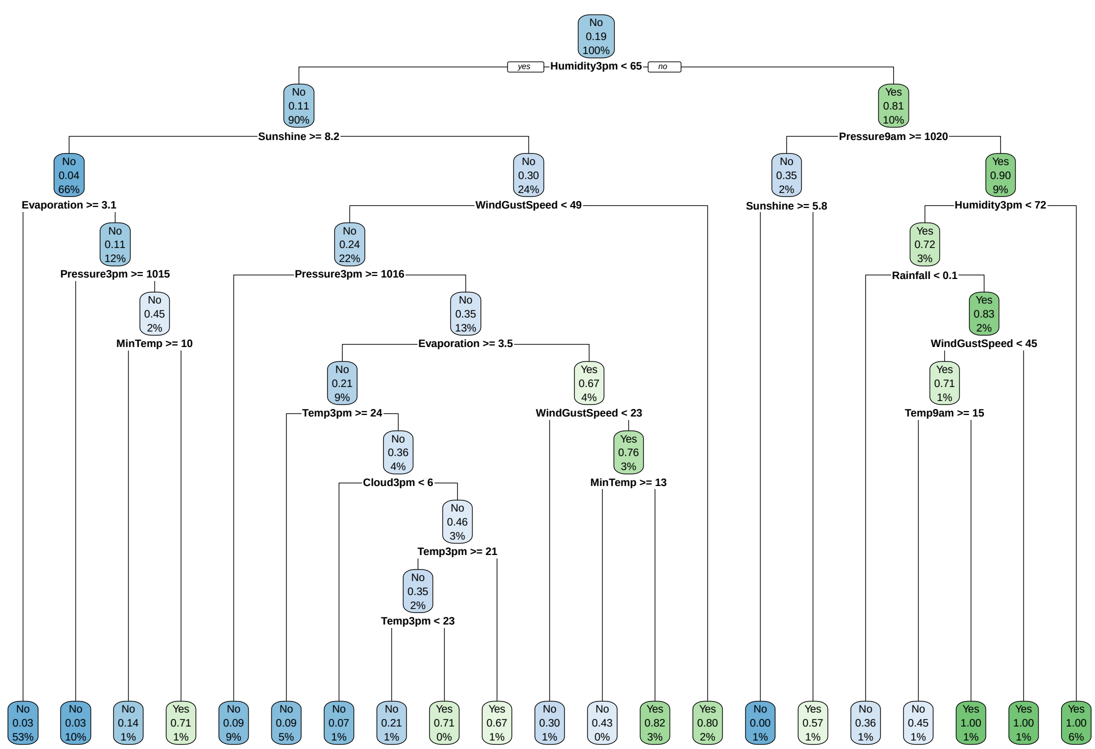
```r
# Best strategy for tree fitting, cp = 0
set.seed(1234) # for reproducibility of results
treeFit2 <- rpart(RainTomorrow ~., data = train2, method = "class", cp = 0)
printcp(treeFit2)
```

```
##
## Classification tree:
## rpart(formula = RainTomorrow ~ ., data = train2, method = "class",
##     cp = 0)
##
## Variables actually used in tree construction:
##  [1] Cloud3pm      Evaporation   Humidity3pm   MinTemp       Pressure3pm
##  [6] Pressure9am   Rainfall      Sunshine      Temp3pm       Temp9am
## [11] WindGustSpeed
##
## Root node error: 265/1431 = 0.18519
##
## n= 1431
##
##           CP nsplit rel error  xerror     xstd
## 1  0.3509434      0   1.00000 1.00000 0.055451
## 2  0.0396226      1   0.64906 0.64906 0.046421
## 3  0.0358491      3   0.56981 0.62264 0.045592
## 4  0.0264151      5   0.49811 0.61509 0.045351
## 5  0.0150943      6   0.47170 0.60377 0.044985
## 6  0.0088050      7   0.45660 0.58491 0.044363
## 7  0.0075472     10   0.43019 0.62642 0.045712
## 8  0.0062893     11   0.42264 0.62264 0.045592
## 9  0.0056604     15   0.39245 0.63396 0.045951
## 10 0.0037736     17   0.38113 0.63396 0.045951
## 11 0.0018868     18   0.37736 0.62642 0.045712
## 12 0.0000000     20   0.37358 0.63774 0.046069
```

```r
plotcp(treeFit2)
```




```r
rpart.plot(treeFit2)
```




```r
xerror <- treeFit2$cptable[,"xerror"]
imin.xerror <- which.min(xerror)
```

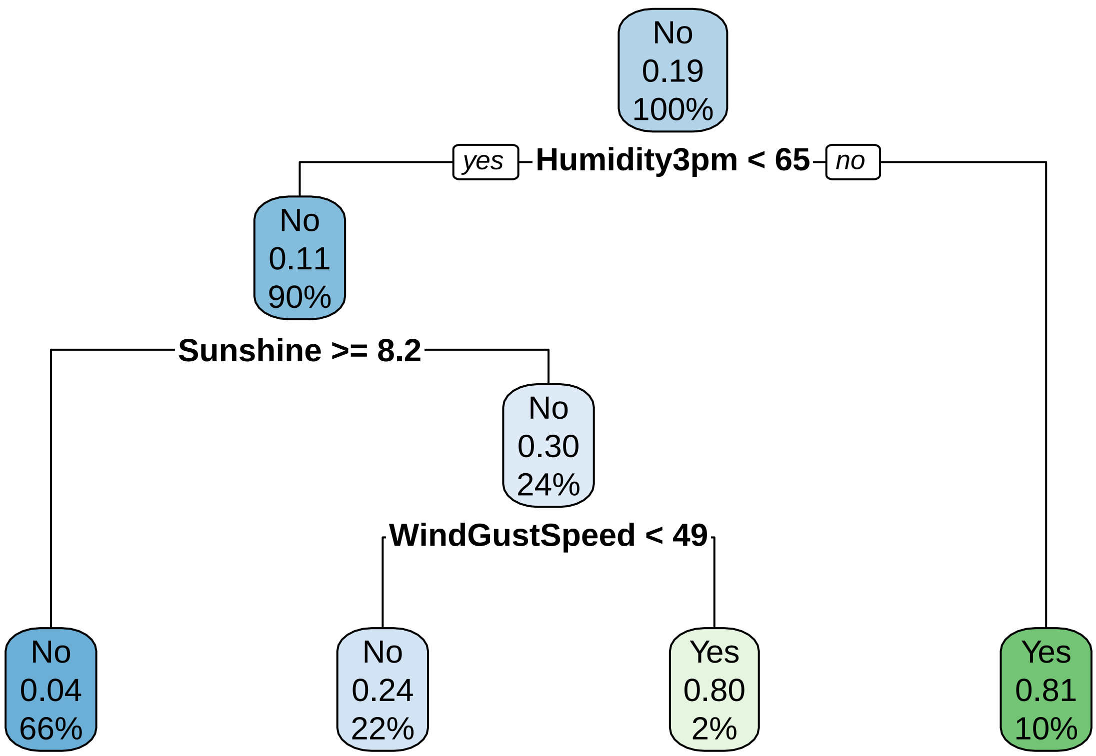
```r
treeFit2$cptable[imin.xerror, ]
```

```
##          CP      nsplit   rel error      xerror        xstd
## 0.008805031 7.000000000 0.456603774 0.584905660 0.044363469
```

```r
upper.xerror <- xerror[imin.xerror] + treeFit2$cptable[imin.xerror, "xstd"]
icp <- min(which(xerror <= upper.xerror))
cp <- treeFit2$cptable[icp, "CP"]
```

After pruning, the tree for the second set of variables is extremely simple, using just 3 variables: Humidity3pm, Sunshine, and WindGustSpeed.

```r
tree2 <- prune(treeFit2, cp = cp)
rpart.plot(tree2)
```




```r
#Classification Rules
rpart.rules(tree2, style = "tall")
```

```
## RainTomorrow is 0.04 when
##     Humidity3pm < 65
##     Sunshine >= 8.2
##
## RainTomorrow is 0.24 when
##     Humidity3pm < 65
##     Sunshine < 8.2
##     WindGustSpeed < 49
##
## RainTomorrow is 0.80 when
##     Humidity3pm < 65
##     Sunshine < 8.2
##     WindGustSpeed >= 49
```

```
##
## RainTomorrow is 0.81 when
##     Humidity3pm >= 65
```

For imbalanced training data, 4 of the 5 most important variables are the same in the second set of variables. The difference is that Temp3pm is considered more important than Evaporation in the second set.

```
#Checking important variables
importance2 <- tree2$variable.importance
importance2 <- round(100*importance2/sum(importance2), 1)
importance2[importance2 >= 1]
```

```
##   Humidity3pm      Sunshine WindGustSpeed     TempRange       Temp3pm
##          49.9          16.4           7.1           6.2           4.0
##      Cloud9am  WindSpeed3pm      Cloud3pm      Rainfall       MaxTemp
##           3.9           3.2           3.1           2.1           1.6
##   Pressure9am  WindSpeed9am
##           1.3           1.0
```

Training the model with the second set of imbalanced training data had worse results than the first set of variables. Specificity was the only metric that was better, increasing from 95.5% to 87%. Sensitivity decreased from 70% to 56% and Balanced Accuracy decreased from 82.8% to 76.6%.

Next, we will check if the SMOTE'd data set performs better with the second set of variables than the first set.

```
#Train Set Evaluation
#Confusion matrix-train
pred_train2 <- predict(tree2, train2, type = 'class') # using train data
#Make sure to state positive class in the confusion matrix.
confusionMatrix(pred_train2, train2$RainTomorrow, positive="Yes")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   No  Yes
##        No  1131  116
##        Yes   35  149
##
##                Accuracy : 0.8945
##                  95% CI : (0.8774, 0.9099)
##     No Information Rate : 0.8148
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.6035
##
##  Mcnemar's Test P-Value : 7.5e-11
##
##             Sensitivity : 0.5623
##             Specificity : 0.9700
##          Pos Pred Value : 0.8098
##          Neg Pred Value : 0.9070
##              Prevalence : 0.1852
##          Detection Rate : 0.1041
##    Detection Prevalence : 0.1286
##       Balanced Accuracy : 0.7661
##
```

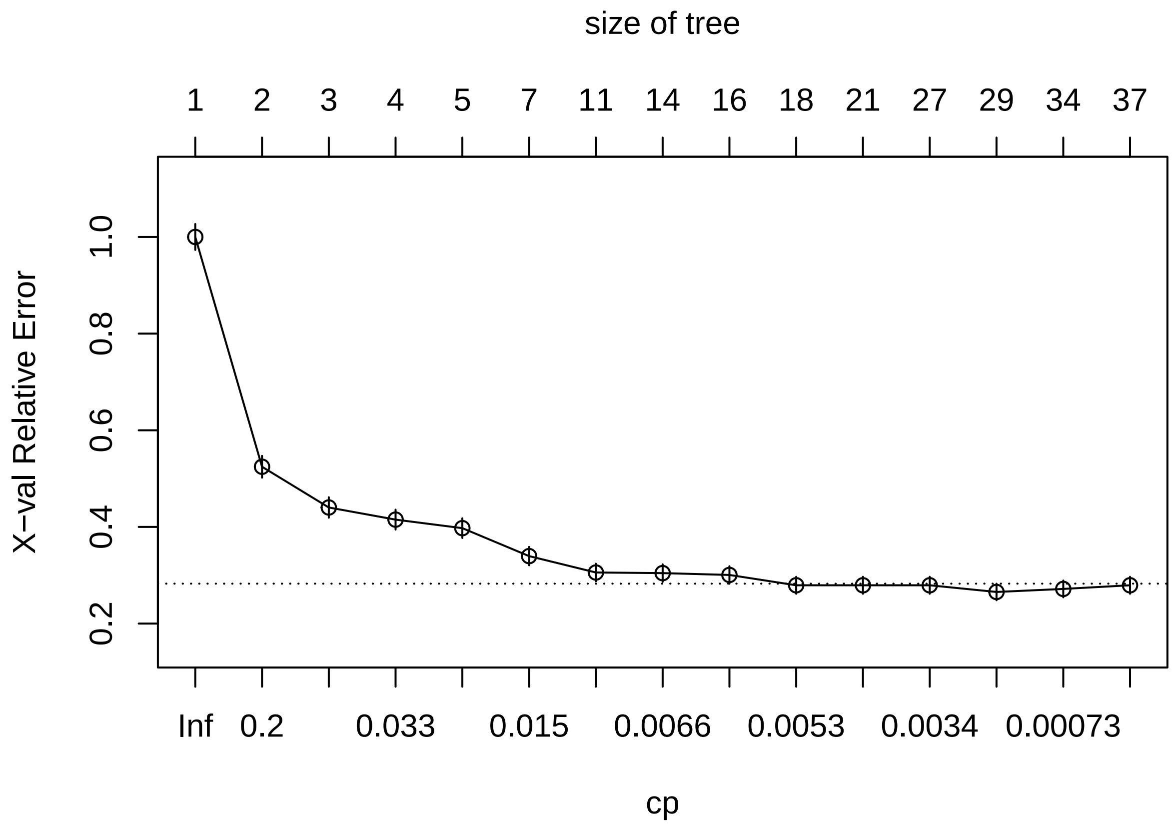
```
##        'Positive' Class : Yes
##
```

```r
# Best strategy for tree fitting, cp = 0
set.seed(1234) # for reproducibility of results
treeBalFit2 <- rpart(RainTomorrow ~., data = trainBal2, method = "class", cp = 0)
printcp(treeBalFit2)
```

Second Set of Variables on Balnced Training Data using SMOTE

```
##
## Classification tree:
## rpart(formula = RainTomorrow ~ ., data = trainBal2, method = "class",
##     cp = 0)
##
## Variables actually used in tree construction:
##  [1] Cloud3pm      Evaporation   Humidity3pm   Humidity9am   MaxTemp
##  [6] MinTemp       Pressure3pm   Pressure9am   Rainfall      Sunshine
## [11] Temp3pm       TempRange     WindGustSpeed WindSpeed3pm  WindSpeed9am
##
## Root node error: 795/1855 = 0.42857
##
## n= 1855
##
##            CP nsplit rel error  xerror     xstd
## 1  0.50566038      0   1.00000 1.00000 0.026810
## 2  0.07924528      1   0.49434 0.52453 0.022616
## 3  0.03396226      2   0.41509 0.44025 0.021196
## 4  0.03144654      3   0.38113 0.41509 0.020718
## 5  0.02075472      4   0.34969 0.39748 0.020367
## 6  0.01069182      6   0.30818 0.33962 0.019105
## 7  0.00691824     10   0.24780 0.30566 0.018279
## 8  0.00628931     13   0.22642 0.30440 0.018247
## 9  0.00566038     15   0.21384 0.30063 0.018150
## 10 0.00503145     17   0.20252 0.27925 0.017585
## 11 0.00377358     20   0.18742 0.27925 0.017585
## 12 0.00314465     26   0.16478 0.27925 0.017585
## 13 0.00125786     28   0.15849 0.26541 0.017201
## 14 0.00041929     33   0.15220 0.27170 0.017377
## 15 0.00000000     36   0.15094 0.27925 0.017585
```

```r
plotcp(treeBalFit2)
```



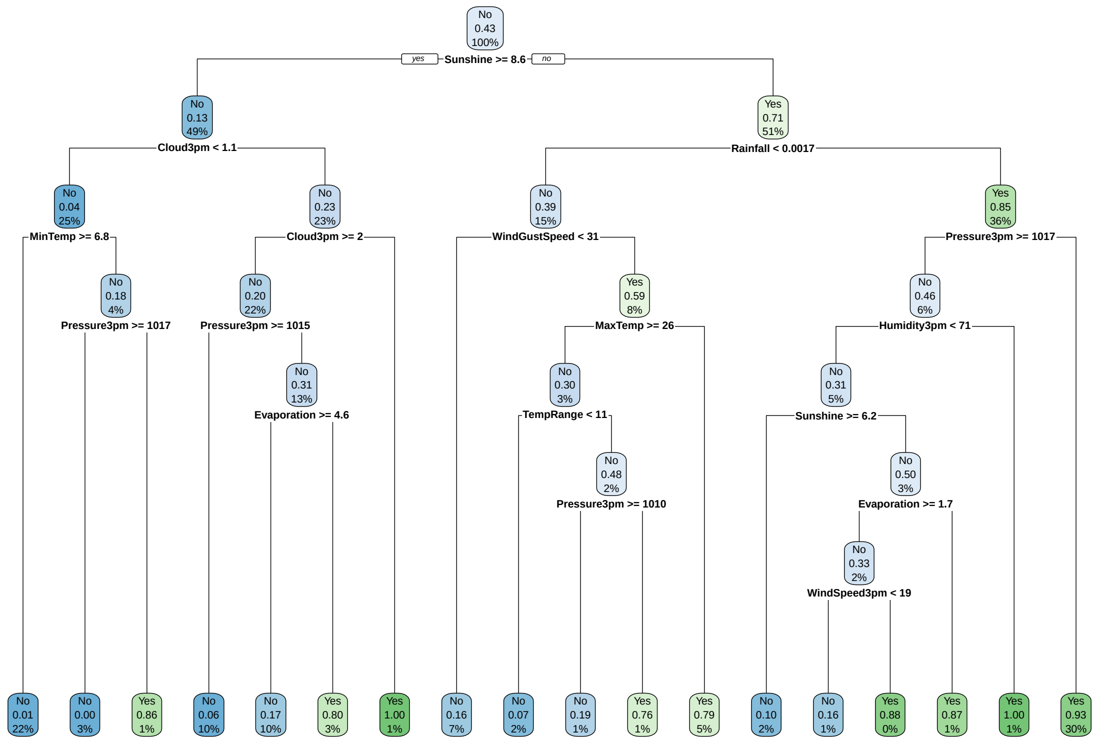

```r
#rpart.plot(treeBalFit2)
```

```r
xerror <- treeBalFit2$cptable[,"xerror"]
imin.xerror <- which.min(xerror)
treeBalFit2$cptable[imin.xerror, ]
```

```
##          CP      nsplit   rel error      xerror        xstd
## 0.001257862 28.000000000 0.158490566 0.265408805 0.017200974
```

```r
upper.xerror <- xerror[imin.xerror] + treeBalFit2$cptable[imin.xerror, "xstd"]
icp <- min(which(xerror <= upper.xerror))
cp <- treeBalFit2$cptable[icp, "CP"]
```

```r
treeBal2 <- prune(treeBalFit2, cp = cp)
rpart.plot(treeBal2)
```



```r
#Classification Rules
rpart.rules(treeBal2, style = "tall")
```

```
## RainTomorrow is 0.00 when
##     Sunshine >= 8.6
##     Pressure3pm >= 1017
##     Cloud3pm < 1.1
##     MinTemp < 6.8
##
## RainTomorrow is 0.01 when
##     Sunshine >= 8.6
##     Cloud3pm < 1.1
##     MinTemp >= 6.8
##
## RainTomorrow is 0.06 when
##     Sunshine >= 8.6
##     Pressure3pm >= 1015
##     Cloud3pm >= 2.0
##
## RainTomorrow is 0.07 when
##     Sunshine < 8.6
##     Rainfall < 0.0017
##     WindGustSpeed >= 31
##     MaxTemp >= 26
##     TempRange < 11
##
## RainTomorrow is 0.10 when
##     Sunshine is 6.2 to 8.6
##     Pressure3pm >= 1017
##     Rainfall >= 0.0017
##     Humidity3pm < 71
```

```
##
## RainTomorrow is 0.16 when
##     Sunshine < 6.2
##     Pressure3pm >= 1017
##     Rainfall >= 0.0017
##     Humidity3pm < 71
##     Evaporation >= 1.7
##     WindSpeed3pm < 19
##
## RainTomorrow is 0.16 when
##     Sunshine < 8.6
##     Rainfall < 0.0017
##     WindGustSpeed < 31
##
## RainTomorrow is 0.17 when
##     Sunshine >= 8.6
##     Pressure3pm < 1015
##     Cloud3pm >= 2.0
##     Evaporation >= 4.6
##
## RainTomorrow is 0.19 when
##     Sunshine < 8.6
##     Pressure3pm >= 1010
##     Rainfall < 0.0017
##     WindGustSpeed >= 31
##     MaxTemp >= 26
##     TempRange >= 11
##
## RainTomorrow is 0.76 when
##     Sunshine < 8.6
##     Pressure3pm < 1010
##     Rainfall < 0.0017
##     WindGustSpeed >= 31
##     MaxTemp >= 26
##     TempRange >= 11
##
## RainTomorrow is 0.79 when
##     Sunshine < 8.6
##     Rainfall < 0.0017
##     WindGustSpeed >= 31
##     MaxTemp < 26
##
## RainTomorrow is 0.80 when
##     Sunshine >= 8.6
##     Pressure3pm < 1015
##     Cloud3pm >= 2.0
##     Evaporation < 4.6
##
## RainTomorrow is 0.86 when
##     Sunshine >= 8.6
##     Pressure3pm < 1017
##     Cloud3pm < 1.1
##     MinTemp < 6.8
##
```

```
## RainTomorrow is 0.87 when
##     Sunshine < 6.2
##     Pressure3pm >= 1017
##     Rainfall >= 0.0017
##     Humidity3pm < 71
##     Evaporation < 1.7
##
## RainTomorrow is 0.88 when
##     Sunshine < 6.2
##     Pressure3pm >= 1017
##     Rainfall >= 0.0017
##     Humidity3pm < 71
##     Evaporation >= 1.7
##     WindSpeed3pm >= 19
##
## RainTomorrow is 0.93 when
##     Sunshine < 8.6
##     Pressure3pm < 1017
##     Rainfall >= 0.0017
##
## RainTomorrow is 1.00 when
##     Sunshine >= 8.6
##     Cloud3pm is 1.1 to 2.0
##
## RainTomorrow is 1.00 when
##     Sunshine < 8.6
##     Pressure3pm >= 1017
##     Rainfall >= 0.0017
##     Humidity3pm >= 71
```

The tree trained with the balanced second set of variables gave Rainfall the second most importance of all the variables, which is a big difference because RainToday was not an important variable in the first set of variables. Sunshine, Cloud3pm, Humidity3pm, and TempRange are common important variables between the two sets.

```
#Checking important variables
importanceBal2 <- treeBal2$variable.importance
importanceBal2 <- round(100*importanceBal2/sum(importanceBal2), 1)
importanceBal2[importanceBal2 >= 1]
```

```
##      Sunshine      Rainfall      Cloud3pm   Humidity3pm     TempRange
##          15.7          12.3          11.9          11.3          10.3
##      Cloud9am   Pressure3pm   Pressure9am       Temp9am       MaxTemp
##          10.2           4.3           4.0           3.8           3.0
##       MinTemp       Temp3pm   Evaporation WindGustSpeed        Season
##           2.8           2.7           2.5           2.0           1.4
```

The model using the balanced second set of variables performs similar to the model created with the first set when evaluating the predictions on the same set of training data.

Accuracy improves from 88% to 88.7%. Sensitivity decreases from 87.2% to 84%. Specificity improves from 88.3% to 89.7%. Balanced accuracy decreased from 87.7% to 86.9%

```
#Evaluation of second model using Training Set
#Confusion matrix-train
pred_trainBal2 <- predict(treeBal2, train2, type = 'class') # using unbalanced train data
#Make sure to state positive class in the confusion matrix.
```

```r
confusionMatrix(pred_trainBal2, train2$RainTomorrow, positive="Yes")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   No  Yes
##        No  1046   42
##        Yes  120  223
##
##                Accuracy : 0.8868
##                  95% CI : (0.8692, 0.9028)
##     No Information Rate : 0.8148
##     P-Value [Acc > NIR] : 7.033e-14
##
##                   Kappa : 0.6632
##
##  Mcnemar's Test P-Value : 1.451e-09
##
##             Sensitivity : 0.8415
##             Specificity : 0.8971
##          Pos Pred Value : 0.6501
##          Neg Pred Value : 0.9614
##              Prevalence : 0.1852
##          Detection Rate : 0.1558
##    Detection Prevalence : 0.2397
##       Balanced Accuracy : 0.8693
##
##        'Positive' Class : Yes
##
```

The second set of variables appears to be overfitting the model, because our metrics are worse using the second set. Accuracy decreases from 83.6% to 80.3%, Sensitivity decreases from 75% to 59.7%, Specificity remained the same at 85.8%, and Balanced Accuracy decreased from 80.6% to 72.8%.

```r
#Test Set Evaluation of Balanced Model 2
#Confusion matrix-test
pred_testBal2 <- predict(treeBal2, test2, type = 'class') # using test data
confusionMatrix(pred_testBal2, test2$RainTomorrow, positive="Yes")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  No Yes
##        No  248  31
##        Yes  41  46
##
##                Accuracy : 0.8033
##                  95% CI : (0.7588, 0.8428)
##     No Information Rate : 0.7896
##     P-Value [Acc > NIR] : 0.2848
##
##                   Kappa : 0.4348
##
##  Mcnemar's Test P-Value : 0.2888
##
```

```
##            Sensitivity : 0.5974
##            Specificity : 0.8581
##         Pos Pred Value : 0.5287
##         Neg Pred Value : 0.8889
##             Prevalence : 0.2104
##         Detection Rate : 0.1257
##   Detection Prevalence : 0.2377
##      Balanced Accuracy : 0.7278
##
##       'Positive' Class : Yes
##
```

Balanced Model 1 performs better with the test data and should be the model that is implemented.