decision_tree.R

Magilan

Mon Oct 08 16:17:50 2018

```
library(tree)
library(rpart)
library(rpart.plot)
library(caret)
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```
library(bst)
```

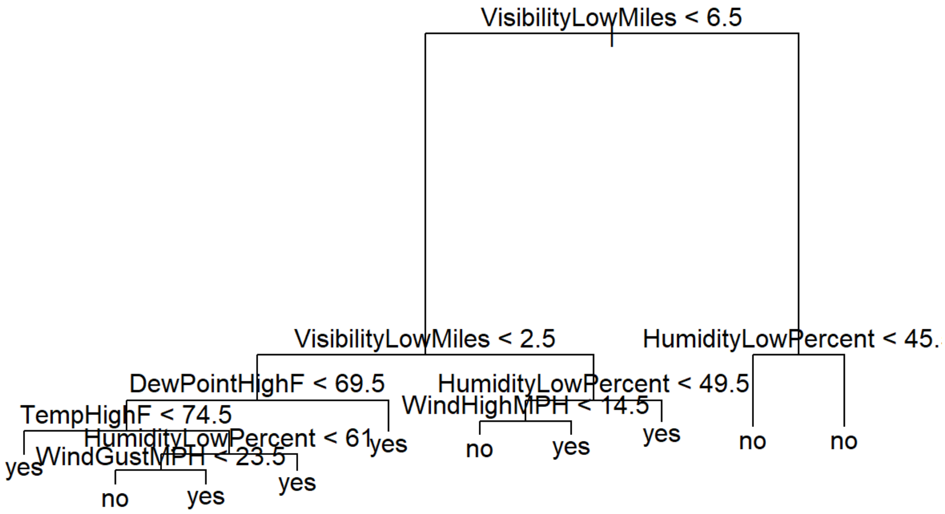
```
## Loading required package: gbm
```

```
## Loaded gbm 2.1.4
```

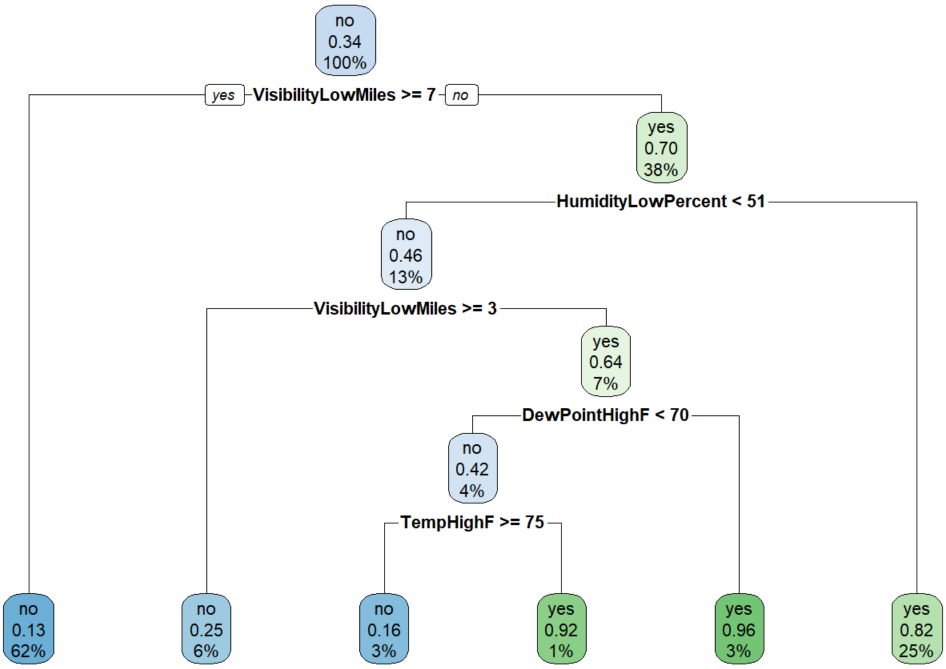
```
#Data Input

data <- read.csv("C:/Users/Magilan/Desktop/ML_project/austin_weather.csv",header = TRUE)
data1=na.omit(data,invert=FALSE)
attach(data1)

data2=data1[,-c(1,20,22)]
tree.model =tree(Rain ~. , data2,method = "class" )
summary(tree.model)
```

```
##
## Classification tree:
## tree(formula = Rain ~ ., data = data2, method = "class")
## Variables actually used in tree construction:
## [1] "VisibilityLowMiles" "DewPointHighF"      "TempHighF"
## [4] "HumidityLowPercent" "WindGustMPH"        "WindHighMPH"
## Number of terminal nodes:  10
## Residual mean deviance:  0.703 = 910.4 / 1295
## Misclassification error rate: 0.1479 = 193 / 1305
```

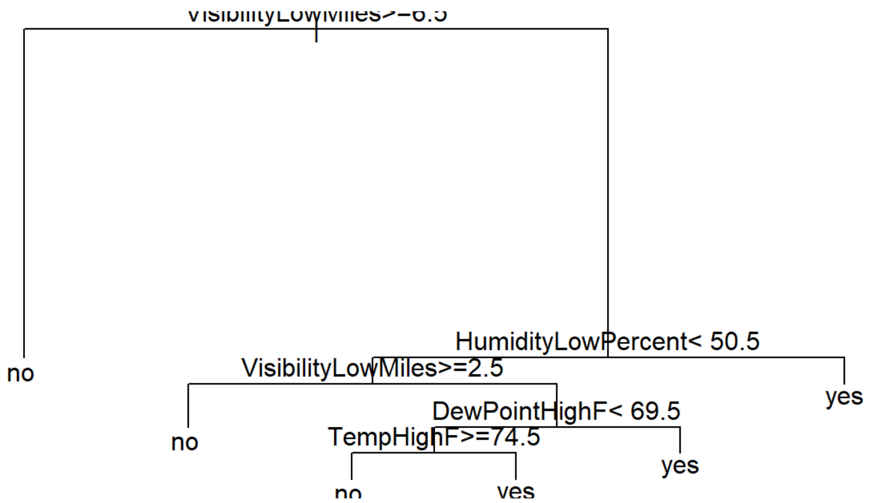
```
plot(tree.model )
text(tree.model ,pretty =0)
```



```
# Train And Test Data

index <- createDataPartition(Rain, p = 0.7, list = FALSE)
train = data1[index,-c(1,20,22)]
test = data1[-index,-c(1,20,22)]
test.Y = Rain[-index]

# Tree Model

tree.model1 = rpart(Rain ~ . ,data = train, method = "class")
rpart.plot(tree.model1)
```



```
plot(tree.model1)
text(tree.model1,pretty = 0)
```



```
tree.pred = predict(tree.model1 ,test, type = "class")
table(tree.pred,test.Y)
```

```
##          test.Y
## tree.pred  no yes
##       no  232  44
##       yes  25  89
```

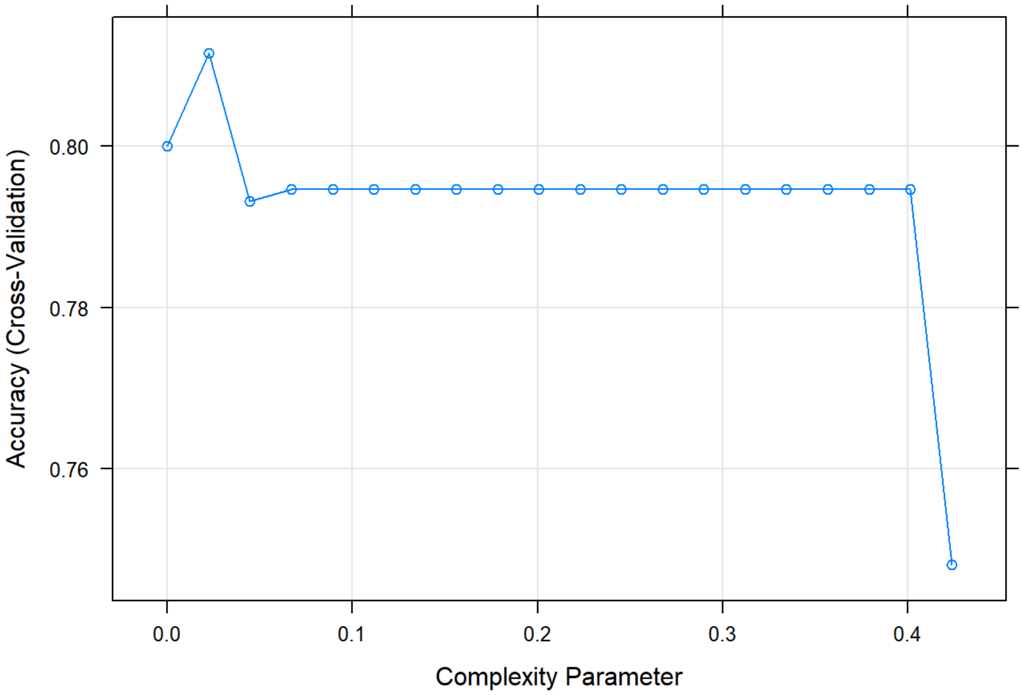
```
confusionMatrix(tree.pred,test.Y)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  no yes
##        no  232  44
##        yes  25  89
##
##                Accuracy : 0.8231
##                  95% CI : (0.7815, 0.8597)
##     No Information Rate : 0.659
##     P-Value [Acc > NIR] : 4.143e-13
##
##                   Kappa : 0.5923
##  Mcnemar's Test P-Value : 0.03024
##
##             Sensitivity : 0.9027
##             Specificity : 0.6692
##          Pos Pred Value : 0.8406
##          Neg Pred Value : 0.7807
##              Prevalence : 0.6590
##          Detection Rate : 0.5949
##    Detection Prevalence : 0.7077
##       Balanced Accuracy : 0.7859
##
##        'Positive' Class : no
##
```

```
# Cross Validation

model <- train(
  Rain ~., data = data1[,-c(1,20,22)], method = "rpart",
  trControl = trainControl("cv", number = 10),
  preProcess = c("center","scale"),
  tuneLength = 20
  )
model
```

```
## CART
##
## 1305 samples
##   18 predictor
##    2 classes: 'no', 'yes'
##
## Pre-processing: centered (18), scaled (18)
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 1176, 1175, 1174, 1174, 1174, 1174, ...
## Resampling results across tuning parameters:
##
##   cp          Accuracy   Kappa
##   0.00000000  0.8000565  0.5500857
##   0.02230352  0.8115894  0.5621310
##   0.04460703  0.7931803  0.5388793
##   0.06691055  0.7947070  0.5452138
##   0.08921407  0.7947070  0.5452138
##   0.11151758  0.7947070  0.5452138
##   0.13382110  0.7947070  0.5452138
##   0.15612462  0.7947070  0.5452138
##   0.17842813  0.7947070  0.5452138
##   0.20073165  0.7947070  0.5452138
##   0.22303517  0.7947070  0.5452138
##   0.24533868  0.7947070  0.5452138
##   0.26764220  0.7947070  0.5452138
##   0.28994572  0.7947070  0.5452138
##   0.31224923  0.7947070  0.5452138
##   0.33455275  0.7947070  0.5452138
##   0.35685627  0.7947070  0.5452138
##   0.37915978  0.7947070  0.5452138
##   0.40146330  0.7947070  0.5452138
##   0.42376682  0.7480188  0.3657507
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was cp = 0.02230352.
```

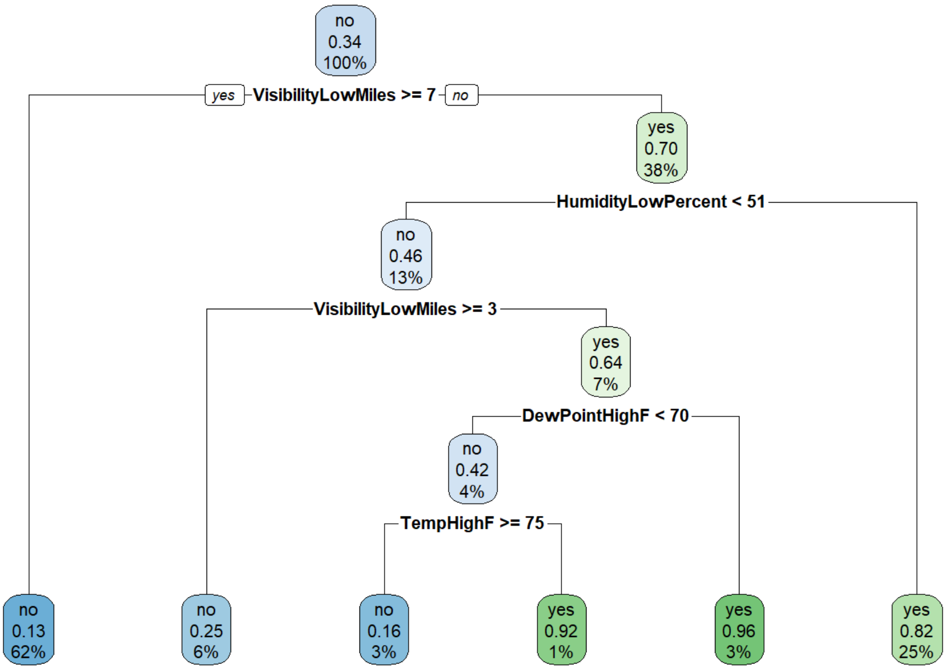
```
plot(model)
```



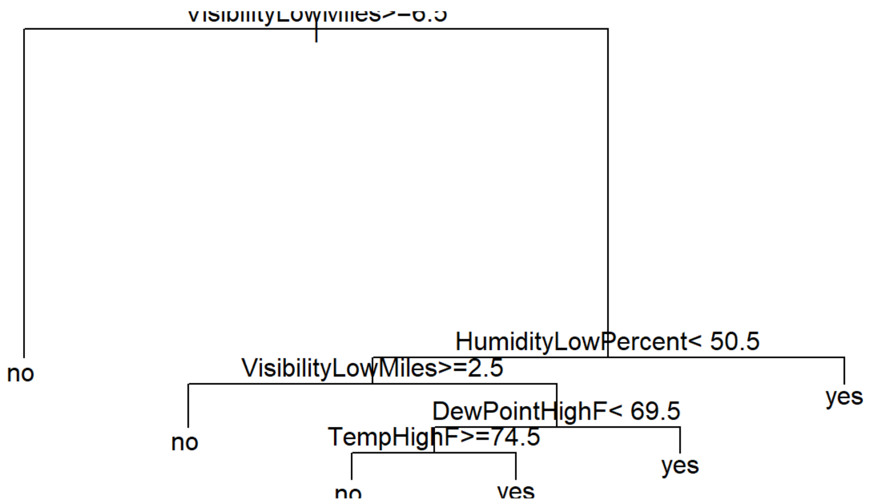
```
k=model$bestTune
k
```

```
##           cp
## 2 0.02230352
```

```
# Prunning

ptree<-  prune(tree.model1,cp=0.022303)
rpart.plot(ptree)
```



```
plot(ptree)
text(ptree,pretty = 0)
```



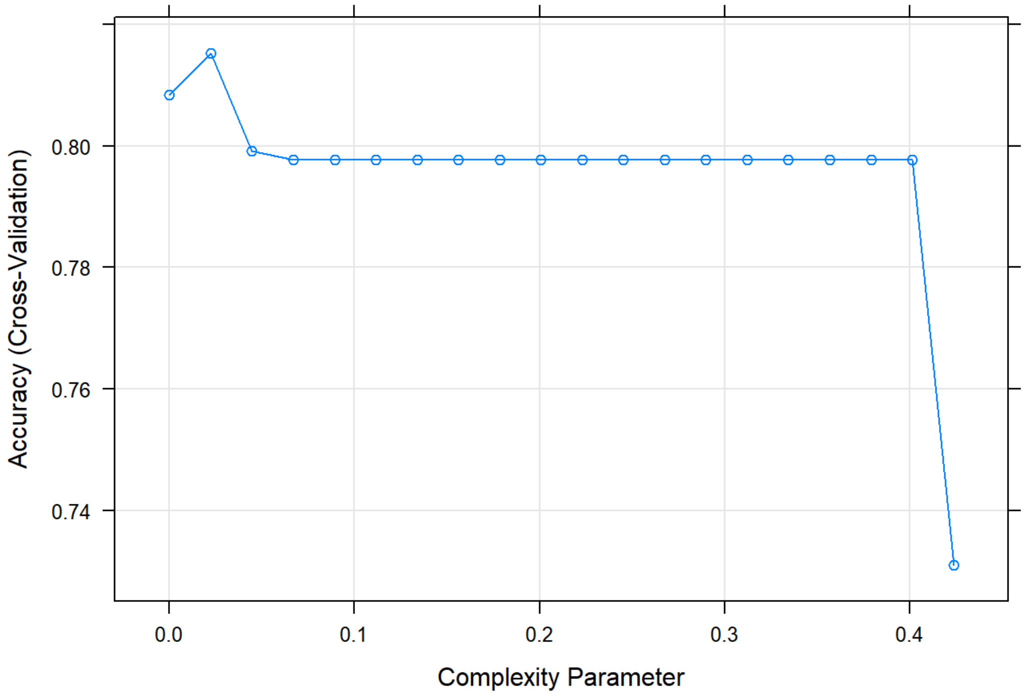
```
ptree.pred = predict(ptree ,test, type = "class")
table(ptree.pred,test.Y)
```

```
##            test.Y
## ptree.pred  no yes
##        no  232  44
##        yes  25  89
```

```
confusionMatrix(ptree.pred,test.Y)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  no yes
##        no  232  44
##        yes  25  89
##
##                Accuracy : 0.8231
##                  95% CI : (0.7815, 0.8597)
##     No Information Rate : 0.659
##     P-Value [Acc > NIR] : 4.143e-13
##
##                   Kappa : 0.5923
##  Mcnemar's Test P-Value : 0.03024
##
##             Sensitivity : 0.9027
##             Specificity : 0.6692
##          Pos Pred Value : 0.8406
##          Neg Pred Value : 0.7807
##              Prevalence : 0.6590
##          Detection Rate : 0.5949
##    Detection Prevalence : 0.7077
##       Balanced Accuracy : 0.7859
##
##        'Positive' Class : no
##
```

```
# Using Gini Indexing

model1 <- train(
  Rain ~., data = data1[,-c(1,20,22)],parms = list(split = "gini"),
  method = "rpart",
  trControl = trainControl("cv", number = 10),
  preProcess = c("center","scale"),
  tuneLength = 20
)
plot(model1)
```



```
model1$bestTune
```

```
##           cp
## 2 0.02230352
```

```
tree.pred.gini = predict(model1 ,test)
table(tree.pred.gini,test.Y)
```

```
##                test.Y
## tree.pred.gini  no yes
##             no  228  39
##             yes  29  94
```

```
confusionMatrix(tree.pred.gini,test.Y)
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction  no yes
##        no  228  39
##        yes  29  94
##                                           
##                Accuracy : 0.8256          
##                  95% CI : (0.7843, 0.862) 
##     No Information Rate : 0.659           
##     P-Value [Acc > NIR] : 1.695e-13       
##                                           
##                   Kappa : 0.6049          
##  Mcnemar's Test P-Value : 0.2751          
##                                           
##             Sensitivity : 0.8872          
##             Specificity : 0.7068          
##          Pos Pred Value : 0.8539          
##          Neg Pred Value : 0.7642          
##              Prevalence : 0.6590          
##          Detection Rate : 0.5846          
##    Detection Prevalence : 0.6846          
##       Balanced Accuracy : 0.7970          
##                                           
##        'Positive' Class : no              
## 
```