Classification Trees

Langtao Chen

Initial: March 1, 2019 Update: April 6, 2021

Contents

1. Data

As an example, let's use the customer churn dataset to demonstrate how to implement classification tree.

```
df <- read.csv('Telco-Customer-Churn.csv', stringsAsFactors = TRUE)
str(df)
```

```
## 'data.frame':    7043 obs. of  21 variables:
##  $ CustomerID      : Factor w/ 7043 levels "0002-ORFBO","0003-MKNFE",..: 5376 3963 2565 5536 6512 655
##  $ Gender          : Factor w/ 2 levels "Female","Male": 1 2 2 2 1 1 2 1 1 2 ...
##  $ SeniorCitizen   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Partner         : Factor w/ 2 levels "No","Yes": 2 1 1 1 1 1 1 1 2 1 ...
##  $ Dependents      : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 2 1 1 2 ...
##  $ Tenure          : int  1 34 2 45 2 8 22 10 28 62 ...
##  $ PhoneService    : Factor w/ 2 levels "No","Yes": 1 2 2 1 2 2 2 1 2 2 ...
##  $ MultipleLines   : Factor w/ 3 levels "No","No phone service",..: 2 1 1 2 1 3 3 2 3 1 ...
##  $ InternetService : Factor w/ 3 levels "DSL","Fiber optic",..: 1 1 1 1 2 2 2 1 2 1 ...
##  $ OnlineSecurity  : Factor w/ 3 levels "No","No internet service",..: 1 3 3 3 1 1 1 3 1 3 ...
##  $ OnlineBackup    : Factor w/ 3 levels "No","No internet service",..: 3 1 3 1 1 1 3 1 1 3 ...
##  $ DeviceProtection: Factor w/ 3 levels "No","No internet service",..: 1 3 1 3 1 3 1 1 3 1 ...
##  $ TechSupport     : Factor w/ 3 levels "No","No internet service",..: 1 1 1 3 1 1 1 1 3 1 ...
##  $ StreamingTV     : Factor w/ 3 levels "No","No internet service",..: 1 1 1 1 1 3 3 1 3 1 ...
##  $ StreamingMovies : Factor w/ 3 levels "No","No internet service",..: 1 1 1 1 1 3 1 1 3 1 ...
##  $ Contract        : Factor w/ 3 levels "Month-to-month",..: 1 2 1 2 1 1 1 1 1 2 ...
##  $ PaperlessBilling: Factor w/ 2 levels "No","Yes": 2 1 2 1 2 2 2 1 2 1 ...
##  $ PaymentMethod   : Factor w/ 4 levels "Bank transfer (automatic)",..: 3 4 4 1 3 3 2 4 3 1 ...
##  $ MonthlyCharges  : num  29.9 57 53.9 42.3 70.7 ...
##  $ TotalCharges    : num  29.9 1889.5 108.2 1840.8 151.7 ...
##  $ Churn           : Factor w/ 2 levels "No","Yes": 1 1 2 1 2 2 1 1 2 1 ...
```

The response if the Churn column, which is a binary variable.

```
summary(df)
```

```
##       CustomerID      Gender     SeniorCitizen    Partner    Dependents
##  0002-ORFBO:   1   Female:3488   Min.   :0.0000   No :3641   No :4933
##  0003-MKNFE:   1   Male  :3555   1st Qu.:0.0000   Yes:3402   Yes:2110
##  0004-TLHLJ:   1                 Median :0.0000
##  0011-IGKFF:   1                 Mean   :0.1621
##  0013-EXCHZ:   1                 3rd Qu.:0.0000
##  0013-MHZWF:   1                 Max.   :1.0000
##  (Other)   :7037
##      Tenure      PhoneService          MultipleLines     InternetService
##  Min.   : 0.00   No : 682     No              :3390   DSL        :2421
##  1st Qu.: 9.00   Yes:6361     No phone service: 682   Fiber optic:3096
##  Median :29.00                Yes             :2971   No         :1526
##  Mean   :32.37
##  3rd Qu.:55.00
##  Max.   :72.00
##
##              OnlineSecurity              OnlineBackup
##  No                 :3498   No                 :3088
```

```
##  No internet service:1526   No internet service:1526
##  Yes                :2019   Yes                :2429
##
##
##
##
##            DeviceProtection              TechSupport
##  No                 :3095    No                 :3473
##  No internet service:1526    No internet service:1526
##  Yes                :2422    Yes                :2044
##
##
##
##
##              StreamingTV              StreamingMovies           Contract
##  No                 :2810   No                 :2785   Month-to-month:3875
##  No internet service:1526   No internet service:1526   One year      :1473
##  Yes                :2707   Yes                :2732   Two year      :1695
##
##
##
##
##  PaperlessBilling                   PaymentMethod  MonthlyCharges
##  No :2872         Bank transfer (automatic):1544   Min.   : 18.25
##  Yes:4171         Credit card (automatic)  :1522   1st Qu.: 35.50
##                   Electronic check         :2365   Median : 70.35
##                   Mailed check             :1612   Mean   : 64.76
##                                                    3rd Qu.: 89.85
##                                                    Max.   :118.75
##
##   TotalCharges    Churn
##  Min.   :  18.8   No :5174
##  1st Qu.: 401.4   Yes:1869
##  Median :1397.5
##  Mean   :2283.3
##  3rd Qu.:3794.7
##  Max.   :8684.8
##  NA's   :11
```

Since customuer ID is a unique identifier for each customer, let's remove it. There are 11 missing values in TotalCharges column. We remove all missing values from the dataset.

```
df$CustomerID <- NULL
df <- na.omit(df)
```

2. Split Data into Training and Test Sets

Let's split the data into training set (50%) and test set (50%).

```
set.seed(123)
train <- sample(1:nrow(df), nrow(df)*0.5)
```

```
train_df <- df[train,]
test_df <- df[-train,]

# Num of observations in training set
nrow(train_df)
```

```
## [1] 3516
```

```
# Num of observations in test set
nrow(test_df)
```

```
## [1] 3516
```

3. Train A Classification Tree

We use the tree() method in the tree package to fit a classification tree to the training data. The syntax is very similar as the regression tree fit.

The tree() method will fit a regression tree if the response variable is continous, and a classification tree if the response variable is categorical.

```
library(tree)
```

```
## Warning: package 'tree' was built under R version 4.0.4
```

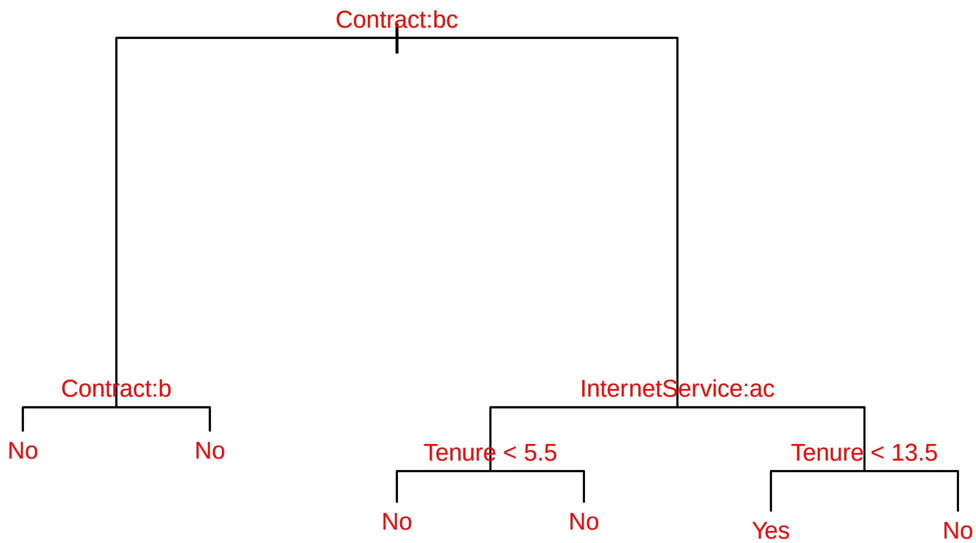
```
# Fit a decision tree
tree_churn <- tree(Churn ~., data = train_df)

# Summary of the decision tree
summary(tree_churn)
```

```
##
## Classification tree:
## tree(formula = Churn ~ ., data = train_df)
## Variables actually used in tree construction:
## [1] "Contract"        "InternetService" "Tenure"
## Number of terminal nodes:  6
## Residual mean deviance:  0.8835 = 3101 / 3510
## Misclassification error rate: 0.2162 = 760 / 3516
```

From the summary, we notice that only three variables (i.e., Contract, InternetService, and Tenure) are used to construct the tree.

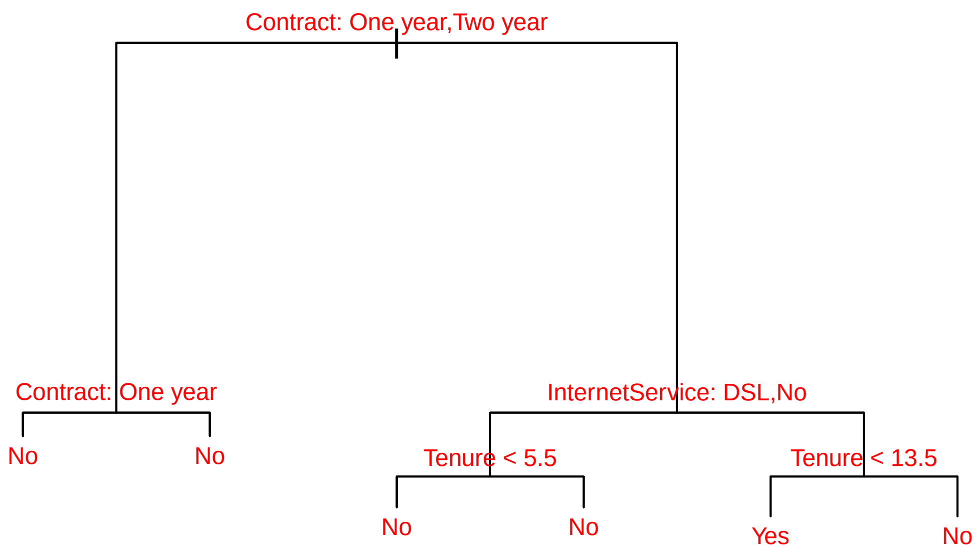
```
# Plot the decison tree
plot(tree_churn)
text(tree_churn, cex = 0.75, col = 'red')
```



From the above, we can see that factor labels have been shortened. To show the full names of factor levels, set the pretty parameter as FALSE.

```
# Plot the decison tree
plot(tree_churn)

text(tree_churn, cex = 0.75,
     col = 'red', pretty = FALSE)
```



The classification tree predicts a customer will churn her service if she has a month-to-month contract, fiber-optic internet service, and tenure < 13.5.

4. Test Performance of the Classification Tree

As we log tranform the response variable, the predicted value of the response needs to be transformed back to the original scale.

```
yhat <- predict(tree_churn, newdata = test_df, type = 'class')
yobs <- test_df$Churn

library(caret)
```

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

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

```
confusionMatrix(yhat, yobs, positive = 'Yes')
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   No  Yes
```

```
##        No  2461  594
##        Yes  128  333
##
##                Accuracy : 0.7947
##                  95% CI : (0.7809, 0.8079)
##     No Information Rate : 0.7363
##     P-Value [Acc > NIR] : 4.719e-16
##
##                   Kappa : 0.3694
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.35922
##             Specificity : 0.95056
##          Pos Pred Value : 0.72234
##          Neg Pred Value : 0.80556
##              Prevalence : 0.26365
##          Detection Rate : 0.09471
##    Detection Prevalence : 0.13111
##       Balanced Accuracy : 0.65489
##
##        'Positive' Class : Yes
##
```

5. Prune the Tree

From the plot of the classification tree shown in section 4, we notice that the tree has unnecessarily complicated structure:

- If the customer has a one year or two year contract, we can simply predict that the customer will not churn her service. So there is no need to further evaluate if the contract is one year or not. That is to say, the 1st and 2nd terminal notes can be combined into a single one.
- If the customer has a month-to-month contract and a DSL or no internet service, we don't need to further evaluate tenure. The customer churn can be classified as No. That is to say, the 3rd and 4th terminal notes in the tree plot in section 4 can be combined into a single one.

Thus, the above decision tree can be pruned.

Generally speaking, an unpruned tree may overfit the data (low bias, high variance). By pruning the tree, we may improve its performance.

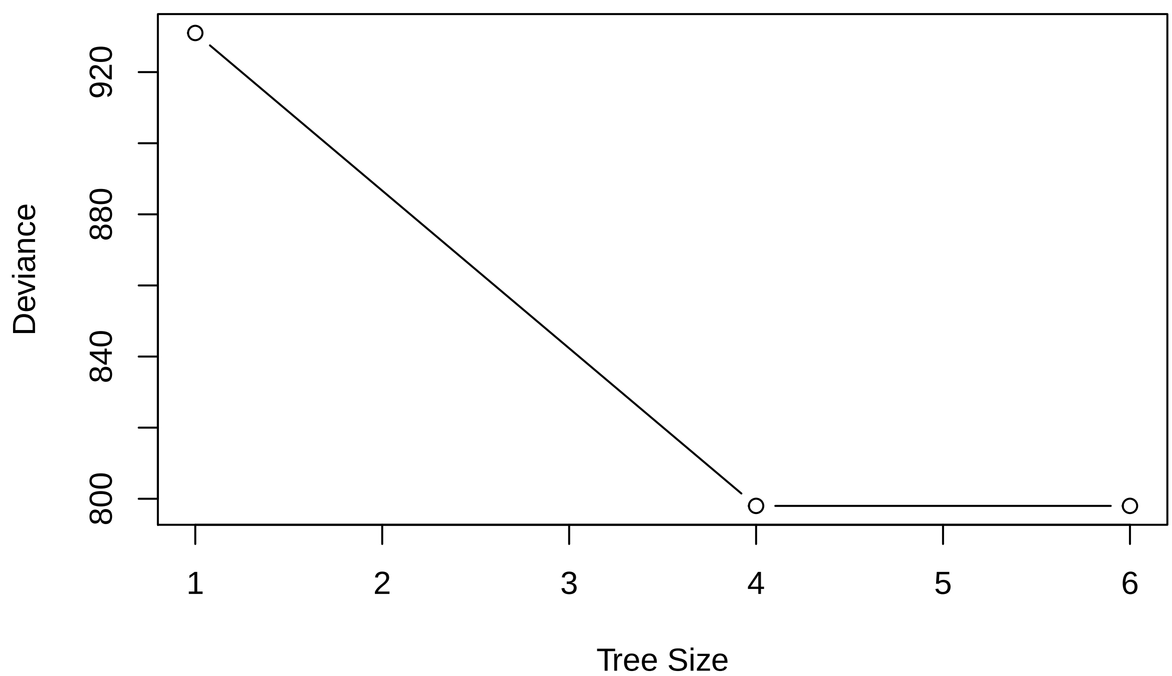
Now, we check whether pruning the tree might lead to improved interpretation and some times better performance.

```r
cv_churn <- cv.tree(tree_churn, FUN=prune.misclass)
cv_churn
```

```
## $size
## [1] 6 4 1
##
## $dev
## [1] 798 798 931
```

```
##
## $k
## [1]      -Inf  0.00000 60.66667
##
## $method
## [1] "misclass"
##
## attr(,"class")
## [1] "prune"         "tree.sequence"
```

```
plot(cv_churn$size, cv_churn$dev, type='b',
     xlab = 'Tree Size', ylab = 'Deviance')
```

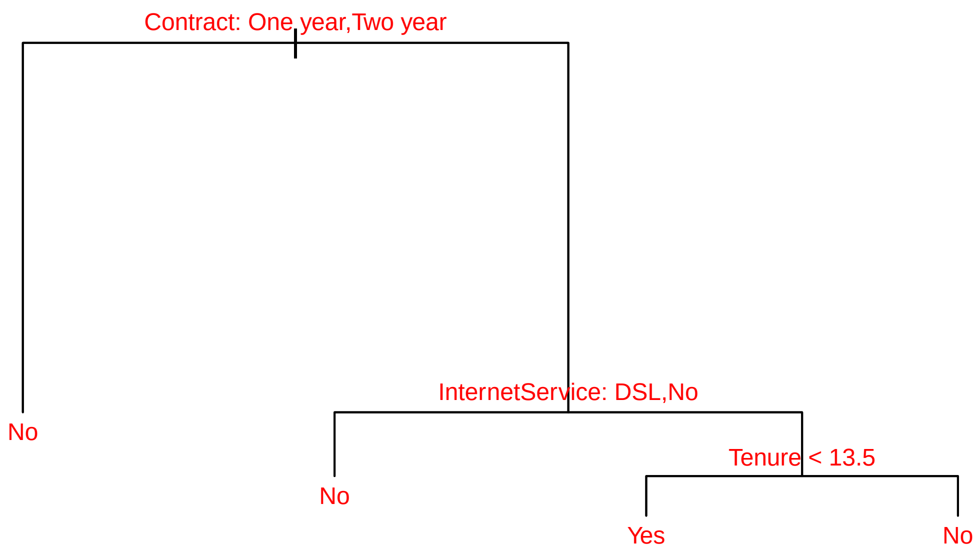


We find the misclassification would be the same if we prune the tree to keep 4 terminal notes. We can apply the prune.misclass() function in to prune the tree.

```
tree_churn_pruned <- prune.misclass(tree_churn, best = 4)

# Plot the pruned tree
plot(tree_churn_pruned)

text(tree_churn_pruned, cex = 0.75,
     col = 'red', pretty = FALSE)
```



We can test the performance of the pruned tree. As shown below, the pruned tree has the same performance as the unpruned tree. However, the pruned tree has a better interpretation as a smaller set of decisions rules are generated from the pruned tree. In addition, the pruned tree tends to have lower variance.

```
yhat2 <- predict(tree_churn_pruned, newdata = test_df, type = 'class')

library(caret)
confusionMatrix(yhat2, yobs, positive = 'Yes')
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   No  Yes
##        No  2461  594
##        Yes  128  333
##
##                Accuracy : 0.7947
##                  95% CI : (0.7809, 0.8079)
##     No Information Rate : 0.7363
##     P-Value [Acc > NIR] : 4.719e-16
##
##                   Kappa : 0.3694
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.35922
```

```
##            Specificity : 0.95056
##         Pos Pred Value : 0.72234
##         Neg Pred Value : 0.80556
##             Prevalence : 0.26365
##         Detection Rate : 0.09471
##   Detection Prevalence : 0.13111
##      Balanced Accuracy : 0.65489
##
##       'Positive' Class : Yes
##
```