Decision Trees

Decision Trees

For this example, I will use the Kaggle Titantic Test and Train datasets. Let's read these in and clean them first.

Data Cleaning

Recall that Decision Trees can only split nominally. This means that any quantitative (numerical) variables will have to be discretized/binned or removed; numerical ordinal data or data with too many categories should also be consolidated.

Let's consider each attribute. Passenger ID - not useful - so we can remove it. If we want to look up a passenger ID later, we can remove the column after we clean and prep everything, so that it stays aligned. For this case, I do not need it. So I will remove it now. I am also going to remove the Cabin, Name, and the Ticket. I like to use temp data frames so as to keep the originals.

```
## Clean and prepare the data

## Look at the structure
(str(TitanicTestData))
```

```
## 'data.frame':    418 obs. of  11 variables:
##  $ PassengerId: int  892 893 894 895 896 897 898 899 900 901 ...
##  $ Pclass     : int  3 3 2 3 3 3 3 2 3 3 ...
##  $ Name       : Factor w/ 418 levels "Abbott, Master. Eugene Joseph",..: 210 409 273 414 182 370 85 5
##  $ Sex        : Factor w/ 2 levels "female","male": 2 1 2 2 1 2 1 2 1 2 ...
##  $ Age        : num  34.5 47 62 27 22 14 30 26 18 21 ...
##  $ SibSp      : int  0 1 0 0 1 0 0 1 0 2 ...
##  $ Parch      : int  0 0 0 0 1 0 0 1 0 0 ...
##  $ Ticket     : Factor w/ 363 levels "110469","110489",..: 153 222 74 148 139 262 159 85 101 270 ...
##  $ Fare       : num  7.83 7 9.69 8.66 12.29 ...
##  $ Cabin      : Factor w/ 76 levels "A11","A18","A21",..: NA NA NA NA NA NA NA NA NA NA ...
##  $ Embarked   : Factor w/ 3 levels "C","Q","S": 2 3 2 3 3 3 2 3 1 3 ...
```

```
## NULL
```

```
(str(TitanicTrainData))
```

```
## 'data.frame':    891 obs. of  12 variables:
##  $ PassengerId: int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Survived   : int  0 1 1 1 0 0 0 0 1 1 ...
##  $ Pclass     : int  3 1 3 1 3 3 1 3 3 2 ...
##  $ Name       : Factor w/ 891 levels "Abbing, Mr. Anthony",..: 109 191 358 277 16 559 520 629 417 58
##  $ Sex        : Factor w/ 2 levels "female","male": 2 1 1 1 2 2 2 2 1 1 ...
##  $ Age        : num  22 38 26 35 35 NA 54 2 27 14 ...
##  $ SibSp      : int  1 1 0 1 0 0 0 3 0 1 ...
##  $ Parch      : int  0 0 0 0 0 0 0 1 2 0 ...
##  $ Ticket     : Factor w/ 681 levels "110152","110413",..: 524 597 670 50 473 276 86 396 345 133 ...
##  $ Fare       : num  7.25 71.28 7.92 53.1 8.05 ...
##  $ Cabin      : Factor w/ 147 levels "A10","A14","A16",..: NA 82 NA 56 NA NA 130 NA NA NA ...
##  $ Embarked   : Factor w/ 3 levels "C","Q","S": 3 1 3 3 3 2 3 3 3 1 ...
```

```
## NULL
```

```
TempTitanicTrain <- TitanicTrainData
TempTitanicTest <- TitanicTestData

# Remove PassengerID, Name, Ticket, and Cabin
TempTitanicTrain <-TempTitanicTrain[ , -which(names(TempTitanicTrain) %in%
                            c("PassengerId","Name","Ticket","Cabin"))]
(head(TempTitanicTrain, n=10))
```

```
##    Survived Pclass    Sex Age SibSp Parch    Fare Embarked
## 1         0      3   male  22     1     0  7.2500        S
## 2         1      1 female  38     1     0 71.2833        C
## 3         1      3 female  26     0     0  7.9250        S
## 4         1      1 female  35     1     0 53.1000        S
## 5         0      3   male  35     0     0  8.0500        S
## 6         0      3   male  NA     0     0  8.4583        Q
## 7         0      1   male  54     0     0 51.8625        S
## 8         0      3   male   2     3     1 21.0750        S
## 9         1      3 female  27     0     2 11.1333        S
## 10        1      2 female  14     1     0 30.0708        C
```

```
TempTitanicTest <- TempTitanicTest[ , -which(names(TempTitanicTest) %in%
                            c("PassengerId","Name","Ticket","Cabin"))]
(head(TempTitanicTest, n=10))
```

```
##    Pclass    Sex  Age SibSp Parch    Fare Embarked
## 1       3   male 34.5     0     0  7.8292        Q
## 2       3 female 47.0     1     0  7.0000        S
## 3       2   male 62.0     0     0  9.6875        Q
## 4       3   male 27.0     0     0  8.6625        S
## 5       3 female 22.0     1     1 12.2875        S
## 6       3   male 14.0     0     0  9.2250        S
## 7       3 female 30.0     0     0  7.6292        Q
## 8       2   male 26.0     1     1 29.0000        S
## 9       3 female 18.0     0     0  7.2292        C
## 10      3   male 21.0     2     0 24.1500        S
```

Next - check how many NAs or missing values.

```
(head((is.na(TempTitanicTrain))))
```

```
##      Survived Pclass   Sex   Age SibSp Parch  Fare Embarked
## [1,]    FALSE  FALSE FALSE FALSE FALSE FALSE FALSE    FALSE
## [2,]    FALSE  FALSE FALSE FALSE FALSE FALSE FALSE    FALSE
## [3,]    FALSE  FALSE FALSE FALSE FALSE FALSE FALSE    FALSE
## [4,]    FALSE  FALSE FALSE FALSE FALSE FALSE FALSE    FALSE
## [5,]    FALSE  FALSE FALSE FALSE FALSE FALSE FALSE    FALSE
## [6,]    FALSE  FALSE FALSE  TRUE FALSE FALSE FALSE    FALSE
```

```
(sum(is.na(TempTitanicTrain)))
```

```
## [1] 179
```

```
(head((is.na(TempTitanicTest))))
```

```
##      Pclass   Sex   Age SibSp Parch  Fare Embarked
## [1,]  FALSE FALSE FALSE FALSE FALSE FALSE    FALSE
```

```
## [2,]  FALSE FALSE FALSE FALSE FALSE FALSE    FALSE
## [3,]  FALSE FALSE FALSE FALSE FALSE FALSE    FALSE
## [4,]  FALSE FALSE FALSE FALSE FALSE FALSE    FALSE
## [5,]  FALSE FALSE FALSE FALSE FALSE FALSE    FALSE
## [6,]  FALSE FALSE FALSE FALSE FALSE FALSE    FALSE
```

```
(sum(is.na(TempTitanicTest)))
```

```
## [1] 87
```

The AGE variable has a lot of missing items - let's think about what do do about this. We can remove the age variable, we can remove the rows with NA, or we can try to fill in the missing ages with the age mean or median. Filling in the values is not always a good idea because we are trying to build a predictor (a decision-maker). Using false ages may generate potentially incorrect results. So - we can remove the column or remove the rows with NA. There is no perfect choice. So let us see how many rows we have left after removing the rows with NA.

```
## How many rows are complete?
cat("The Titanic Test data has a total of ", nrow(TempTitanicTest), "rows.")
```

```
## The Titanic Test data has a total of  418 rows.
```

```
cat("The Titanic Train data has a total of ", nrow(TempTitanicTrain), "rows.")
```

```
## The Titanic Train data has a total of  891 rows.
```

```
TotalCompleteRowsTrain <- (nrow(TempTitanicTrain[complete.cases(TempTitanicTrain),]))
TotalCompleteRowsTest <- (nrow(TempTitanicTest[complete.cases(TempTitanicTest),]))
cat("The Titanic Train data has a total of ", TotalCompleteRowsTrain, "complete rows.")
```

```
## The Titanic Train data has a total of  712 complete rows.
```

```
cat("The Titanic Test data has a total of ", TotalCompleteRowsTest, "complete rows.")
```

```
## The Titanic Test data has a total of  331 complete rows.
```

```
## The above tells us that we will still have a large testing and training
## set - even if we remove all rows with NA

TempTitanicTrain <- TempTitanicTrain[complete.cases(TempTitanicTrain),]
TempTitanicTest <- TempTitanicTest[complete.cases(TempTitanicTest),]
## double check - both of these should be 0 - which they are
(nrow(TempTitanicTrain[!complete.cases(TempTitanicTrain),]))
```

```
## [1] 0
```

```
(nrow(TempTitanicTest[!complete.cases(TempTitanicTest),]))
```

```
## [1] 0
```

Now its time to discretize/ bin some of the variables where it may be appropriate.

```
## Let's look at the str and tables
(str(TempTitanicTest))
```

```
## 'data.frame':    331 obs. of  7 variables:
##  $ Pclass  : int  3 3 2 3 3 3 3 2 3 3 ...
##  $ Sex     : Factor w/ 2 levels "female","male": 2 1 2 2 1 2 1 2 1 2 ...
##  $ Age     : num  34.5 47 62 27 22 14 30 26 18 21 ...
```

```
##  $ SibSp   : int  0 1 0 0 1 0 0 1 0 2 ...
##  $ Parch   : int  0 0 0 0 1 0 0 1 0 0 ...
##  $ Fare    : num  7.83 7 9.69 8.66 12.29 ...
##  $ Embarked: Factor w/ 3 levels "C","Q","S": 2 3 2 3 3 3 2 3 1 3 ...
```

```
## NULL
```

```
(str(TempTitanicTrain))
```

```
## 'data.frame':    712 obs. of  8 variables:
##  $ Survived: int  0 1 1 1 0 0 0 1 1 1 ...
##  $ Pclass  : int  3 1 3 1 3 1 3 3 2 3 ...
##  $ Sex     : Factor w/ 2 levels "female","male": 2 1 1 1 2 2 2 1 1 1 ...
##  $ Age     : num  22 38 26 35 35 54 2 27 14 4 ...
##  $ SibSp   : int  1 1 0 1 0 0 3 0 1 1 ...
##  $ Parch   : int  0 0 0 0 0 0 1 2 0 1 ...
##  $ Fare    : num  7.25 71.28 7.92 53.1 8.05 ...
##  $ Embarked: Factor w/ 3 levels "C","Q","S": 3 1 3 3 3 3 3 3 1 3 ...
```

```
## NULL
```

```
## change "survived" to a factor
TempTitanicTrain$Survived=factor(TempTitanicTrain$Survived)
##Pclass is the classification of the ticket - 1st, 2nd, 3rd.
## Make this into a factor
TempTitanicTrain$Pclass=factor(TempTitanicTrain$Pclass)
TempTitanicTest$Pclass=factor(TempTitanicTest$Pclass)
## Age is quantitative and must be binned and discretized
## Check Age for errors and look at values
(freq=table(TempTitanicTrain$Age))
```

```
##
## 0.42 0.67 0.75 0.83 0.92    1    2    3    4    5    6    7    8    9   10
##    1    1    2    2    1    7   10    6   10    4    3    3    4    8    2
##   11   12   13   14 14.5   15   16   17   18   19   20 20.5   21   22   23
##    4    1    2    6    1    5   17   13   26   25   15    1   24   27   15
## 23.5   24 24.5   25   26   27   28 28.5   29   30 30.5   31   32 32.5   33
##    1   30    1   23   18   18   25    2   20   25    2   17   18    2   15
##   34 34.5   35   36 36.5   37   38   39   40 40.5   41   42   43   44   45
##   15    1   18   22    1    6   10   14   13    2    6   13    5    9   12
## 45.5   46   47   48   49   50   51   52   53   54   55 55.5   56   57   58
##    2    3    9    9    6   10    7    6    1    8    2    1    4    2    5
##   59   60   61   62   63   64   65   66   70 70.5   71   74   80
##    2    4    3    3    2    2    3    1    2    1    2    1    1
```

```
## We see that there are some incorrect ages: .42, .67, .75, .83, and .92 are not correct.
## Remove those rows first
(freq=table(TempTitanicTest$Age))  ## Same errors in Test data
```

```
##
## 0.17 0.33 0.75 0.83 0.92    1    2    3    5    6    7    8    9   10 11.5
##    1    1    1    1    1    3    2    1    1    3    1    2    2    2    1
##   12   13   14 14.5   15   16   17   18 18.5   19   20   21   22 22.5   23
##    2    3    2    1    1    2    7   13    3    4    8   17   16    1   11
##   24   25   26 26.5   27   28 28.5   29   30   31   32 32.5   33   34 34.5
##   17   11   12    1   12    7    1   10   15    6    6    2    6    1    1
##   35   36 36.5   37   38 38.5   39   40 40.5   41   42   43   44   45   46
```

```
##    5    9    1    3    3    1    6    5    1    5    5    4    1    9    3
##   47   48   49   50   51   53   54   55   57   58   59   60   61   62   63
##    5    5    3    5    1    3    2    6    3    1    1    3    2    1    2
##   64   67   76
##    3    1    1
```

```
## Place NA for any ages that are < 1
TempTitanicTrain$Age <- ifelse(TempTitanicTrain$Age < 1, "NA", TempTitanicTrain$Age)
(freq=table(TempTitanicTrain$Age))
```

```
##
##    1   10   11   12   13   14 14.5   15   16   17   18   19    2   20 20.5
##    7    2    4    1    2    6    1    5   17   13   26   25   10   15    1
##   21   22   23 23.5   24 24.5   25   26   27   28 28.5   29    3   30 30.5
##   24   27   15    1   30    1   23   18   18   25    2   20    6   25    2
##   31   32 32.5   33   34 34.5   35   36 36.5   37   38   39    4   40 40.5
##   17   18    2   15   15    1   18   22    1    6   10   14   10   13    2
##   41   42   43   44   45 45.5   46   47   48   49    5   50   51   52   53
##    6   13    5    9   12    2    3    9    9    6    4   10    7    6    1
##   54   55 55.5   56   57   58   59    6   60   61   62   63   64   65   66
##    8    2    1    4    2    5    2    3    4    3    3    2    2    3    1
##    7   70 70.5   71   74    8   80    9   NA
##    3    2    1    2    1    4    1    8    7
```

```
TempTitanicTest$Age <- ifelse(TempTitanicTest$Age < 1, "NA", TempTitanicTest$Age)
(freq=table(TempTitanicTest$Age))
```

```
##
##    1   10 11.5   12   13   14 14.5   15   16   17   18 18.5   19    2   20
##    3    2    1    2    3    2    1    1    2    7   13    3    4    2    8
##   21   22 22.5   23   24   25   26 26.5   27   28 28.5   29    3   30   31
##   17   16    1   11   17   11   12    1   12    7    1   10    1   15    6
##   32 32.5   33   34 34.5   35   36 36.5   37   38 38.5   39   40 40.5   41
##    6    2    6    1    1    5    9    1    3    3    1    6    5    1    5
##   42   43   44   45   46   47   48   49    5   50   51   53   54   55   57
##    5    4    1    9    3    5    5    3    1    5    1    3    2    6    3
##   58   59    6   60   61   62   63   64   67    7   76    8    9   NA
##    1    1    3    3    2    1    2    3    1    1    1    2    2    5
```

```
## Remove NAs

TempTitanicTrain <- TempTitanicTrain[complete.cases(TempTitanicTrain),]
TempTitanicTest <- TempTitanicTest[complete.cases(TempTitanicTest),]

## Now we can discretize the Age
TempTitanicTrain$Age[TempTitanicTrain$Age <= 22] <- 1
TempTitanicTrain$Age[TempTitanicTrain$Age > 22 & TempTitanicTrain$Age <=38] <- 2
TempTitanicTrain$Age[TempTitanicTrain$Age > 38] <- 3
TempTitanicTrain$Age=factor(TempTitanicTrain$Age)
(TempTitanicTrain$Age)
```

```
##   [1] 1 2 2 2 2 3 1 2 1 3 3 1 3 1 3 1 2 2 2 1 2 3 2 1 3 3 2 3 1 1 1 3 2 2 1
##  [36] 1 3 1 3 2 3 1 2 3 1 1 3 3 2 1 1 2 2 1 1 2 2 2 3 2 1 2 2 1 2 1 2 2 2 1
##  [71] 3 2 3 3 2 2 2 2 1 2 2 2 1 2 3 1 1 1 1 1 3 2 2 1 1 2 2 3 1 2 3 2 1 3 2
## [106] 2 2 1 2 1 2 1 2 1 1 1 2 3 2 3 3 1 3 3 3 1 2 3 3 2 1 1 3 3 2 3 3 1 1 3
## [141] 1 3 2 2 3 1 3 3 3 2 2 1 1 2 3 3 3 2 2 2 3 1 1 2 2 1 3 2 2 1 2 2 2 3 2
```

```
## [176] 2 1 2 3 2 1 1 1 1 2 2 3 3 2 3 3 1 2 2 1 2 3 2 2 2 3 2 3 2 3 2 2 2 3 2
## [211] 3 3 2 1 2 3 2 2 3 2 3 3 3 2 3 2 1 1 2 2 1 3 1 2 1 2 2 2 2 1 3 1 3 1 2
## [246] 2 2 1 2 2 3 2 2 3 2 3 1 2 2 1 2 3 2 2 1 3 2 1 2 3 3 3 1 2 2 2 2 2 3 2
## [281] 3 2 1 2 2 1 2 3 2 3 2 2 3 2 2 1 1 1 2 1 2 1 1 3 1 2 2 1 1 2 1 2 1 2 2
## [316] 2 1 2 3 2 2 3 2 1 2 1 2 3 2 1 2 3 2 1 2 1 1 2 2 1 2 1 2 2 3 1 3 1 1 2
## [351] 3 2 3 1 2 2 3 1 2 3 3 2 2 3 2 3 3 3 2 3 3 2 3 3 2 2 2 1 2 2 1 1 3 3 3
## [386] 2 2 3 2 3 1 3 3 1 3 2 2 1 1 2 1 1 2 2 2 2 2 3 2 3 2 2 2 2 1 3 3 3 3 2
## [421] 1 1 2 3 3 2 1 2 3 1 2 3 3 1 2 3 1 2 1 1 3 3 3 2 3 2 2 1 2 2 3 3 2 1 1
## [456] 2 3 2 2 3 3 2 1 3 3 1 2 3 3 2 2 3 3 2 3 2 2 2 2 1 3 3 2 2 2 2 3 2 2 3
## [491] 1 1 1 3 3 1 2 3 3 2 3 2 2 2 3 1 2 1 3 3 1 3 2 1 1 1 2 2 2 3 3 3 3 2 1
## [526] 2 2 3 3 2 3 2 1 2 1 3 2 2 1 1 3 2 1 1 1 1 2 3 2 3 3 3 3 3 1 2 1 2 2 3
## [561] 3 3 1 2 3 2 3 1 2 2 2 3 1 2 3 2 1 2 2 2 2 1 2 2 2 3 2 2 1 2 2 3 1 2 1
## [596] 2 3 3 2 2 3 3 2 1 2 2 3 1 2 1 3 2 2 2 3 3 3 1 3 3 1 1 2 2 2 1 3 1 3 1
## [631] 2 3 3 2 2 2 2 2 1 3 2 2 3 1 3 2 2 3 2 3 2 2 2 3 1 3 2 2 2 1 1 1 3 2 1
## [666] 3 1 2 1 1 2 2 1 3 2 2 3 3 3 1 3 1 3 3 2 3 1 3 2 3 2 2 3 2 3 2 3 2 1 1
## [701] 1 3 2 2 1 2 2 3 2 1 2 2
## Levels: 1 2 3
```

```
TempTitanicTest$Age[TempTitanicTest$Age <= 22] <- 1
TempTitanicTest$Age[TempTitanicTest$Age > 22 & TempTitanicTest$Age <=38] <- 2
TempTitanicTest$Age[TempTitanicTest$Age > 38] <- 3
TempTitanicTest$Age=factor(TempTitanicTest$Age)
(TempTitanicTest$Age)
```

```
##   [1] 2 3 3 2 1 1 2 2 1 1 3 2 3 3 2 2 1 2 3 3 3 1 3 3 1 2 3 3 2 2 2 1 1 2 3
##  [36] 3 2 3 2 3 3 2 2 2 1 2 1 2 2 2 1 2 1 1 1 1 3 2 3 2 1 2 2 2 2 3 2 2 3 3
##  [71] 3 2 1 1 1 2 2 2 3 2 1 2 3 2 2 1 2 1 1 3 2 1 3 1 1 2 2 1 2 2 1 1 3 2 2
## [106] 3 3 2 2 2 2 3 1 2 3 2 3 2 1 2 2 2 1 2 2 2 3 2 3 2 3 2 3 1 1 2 2 3 1 1
## [141] 3 2 3 2 2 1 2 3 1 1 3 2 1 3 3 2 3 1 2 3 3 3 2 2 2 2 2 2 1 3 2 3 3 3 2
## [176] 1 1 1 3 2 3 2 1 1 1 3 1 3 1 1 3 3 3 3 3 1 3 2 3 1 2 2 2 2 1 1 2 1 2 1
## [211] 2 1 3 2 1 2 3 2 1 2 3 3 1 2 2 2 3 2 2 1 2 2 2 3 2 3 2 3 3 3 1 1 2 3 1
## [246] 3 1 2 1 2 2 2 2 3 2 1 3 2 1 3 3 1 2 2 2 3 2 1 2 3 1 2 2 2 2 3 2 1 3 3
## [281] 3 3 2 1 2 2 2 2 2 1 3 2 1 2 3 3 3 3 1 1 3 3 2 1 2 2 3 1 3 2 3 1 3 2 1 2
## [316] 3 1 2 2 2 1 1 3 1 2 3 2 2 2 3 3
## Levels: 1 2 3
```

```
## check it
(freq=table(TempTitanicTrain$Age))
```

```
##
##   1   2   3
## 186 311 215
```

```
(freq=table(TempTitanicTest$Age))
```

```
##
##   1   2   3
##  87 142 102
```

```
##Look at the str again
(str(TempTitanicTest))
```

```
## 'data.frame':    331 obs. of  7 variables:
##  $ Pclass  : Factor w/ 3 levels "1","2","3": 3 3 2 3 3 3 3 2 3 3 ...
##  $ Sex     : Factor w/ 2 levels "female","male": 2 1 2 2 1 2 1 2 1 2 ...
##  $ Age     : Factor w/ 3 levels "1","2","3": 2 3 3 2 1 1 2 2 1 1 ...
##  $ SibSp   : int  0 1 0 0 1 0 0 1 0 2 ...
```

```
##  $ Parch   : int  0 0 0 0 1 0 0 1 0 0 ...
##  $ Fare    : num  7.83 7 9.69 8.66 12.29 ...
##  $ Embarked: Factor w/ 3 levels "C","Q","S": 2 3 2 3 3 3 2 3 1 3 ...
```

```
## NULL
```

```
(str(TempTitanicTrain))
```

```
## 'data.frame':    712 obs. of  8 variables:
##  $ Survived: Factor w/ 2 levels "0","1": 1 2 2 2 1 1 1 2 2 2 ...
##  $ Pclass  : Factor w/ 3 levels "1","2","3": 3 1 3 1 3 1 3 3 2 3 ...
##  $ Sex     : Factor w/ 2 levels "female","male": 2 1 1 1 2 2 2 1 1 1 ...
##  $ Age     : Factor w/ 3 levels "1","2","3": 1 2 2 2 2 3 1 2 1 3 ...
##  $ SibSp   : int  1 1 0 1 0 0 3 0 1 1 ...
##  $ Parch   : int  0 0 0 0 0 0 1 2 0 1 ...
##  $ Fare    : num  7.25 71.28 7.92 53.1 8.05 ...
##  $ Embarked: Factor w/ 3 levels "C","Q","S": 3 1 3 3 3 3 3 3 1 3 ...
```

```
## NULL
```

```
## Now we will discretize Sibsp (number of siblings on board)
## and Parch (number of parents or children)
## WHile I will not do this here - it might also be interesting to
## add these to create a vew attribute called Family

(freq=table(TempTitanicTrain$SibSp))
```

```
##
##   0   1   2   3   4   5
## 469 183  25  12  18   5
```

```
(freq=table(TempTitanicTest$SibSp))
```

```
##
##   0   1   2   3   4   5   8
## 213  97  11   4   4   1   1
```

```
(freq=table(TempTitanicTrain$Parch))
```

```
##
##   0   1   2   3   4   5   6
## 519 110  68   5   4   5   1
```

```
(freq=table(TempTitanicTest$Parch))
```

```
##
##   0   1   2   3   4   5   6
## 246  50  29   3   1   1   1
```

```
## Given that so much of the data is at 0 or 1, I will group
## SibSp and Parch so that they are 0 (none) or 1 (one or more)

TempTitanicTrain$SibSp[TempTitanicTrain$SibSp == 0] <- 0
TempTitanicTrain$SibSp[TempTitanicTrain$SibSp > 0] <- 1
TempTitanicTrain$SibSp=factor(TempTitanicTrain$SibSp)
(TempTitanicTrain$SibSp)
```

```
##   [1] 1 1 0 1 0 0 1 0 1 1 0 0 1 0 0 1 1 0 0 0 0 1 1 1 0 0 1 1 0 1 1 1 1 1 0
##  [36] 1 1 0 1 1 0 0 0 1 1 0 1 1 0 0 1 1 0 1 0 1 0 0 0 0 0 0 0 0 1 1 1 0 0 0
```

```
##  [71] 1 1 0 0 0 0 1 0 0 0 1 0 0 0 0 1 0 1 0 0 0 1 0 1 1 1 0 0 1 0 0 0 0 1 1
## [106] 0 0 0 1 0 0 0 1 0 0 1 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 1 0 0 1 1 0 0
## [141] 1 0 0 0 1 1 0 0 1 0 0 0 1 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 1 0
## [176] 0 0 0 0 1 0 0 0 0 1 0 0 1 0 1 0 0 0 0 0 0 1 0 0 1 1 1 0 1 0 0 0 0 0 1
## [211] 1 0 0 1 1 0 0 0 0 0 1 0 1 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 1 1 1
## [246] 0 0 1 1 0 1 0 1 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 1 0 0 0 1 1 0 0 0 0 0 1
## [281] 0 0 1 1 0 0 0 1 1 0 0 0 1 0 1 1 0 0 1 0 0 0 0 0 0 0 1 0 1 0 0 1 0 1 1
## [316] 0 0 0 0 0 0 0 0 1 1 0 1 0 1 0 1 0 1 0 0 0 0 0 1 1 1 0 0 0 1 0 1 1 1 1
## [351] 1 0 1 0 1 0 0 0 0 1 0 1 0 1 0 0 0 0 0 0 0 0 0 1 0 1 0 0 1 1 0 0 1 0 0
## [386] 1 1 0 0 1 0 0 0 0 1 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 1
## [421] 1 1 0 0 0 0 0 0 1 1 1 1 0 1 1 1 0 0 0 0 0 1 1 1 0 0 1 0 0 0 0 1 0 0 0
## [456] 0 1 0 1 1 0 0 0 0 1 0 0 1 0 1 1 0 1 1 0 0 1 0 0 1 0 1 0 1 1 1 1 0 1 1
## [491] 1 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 1 1 1 0 0 0 0 0 0 1 1 0 0 1 0 0 0 1
## [526] 1 0 0 1 1 0 0 0 0 0 1 0 0 0 1 1 1 1 0 0 0 1 0 0 0 0 0 1 0 1 0 0 0 1 0
## [561] 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 1 0 1 1 1 0 0 0 0 0 1 0 1 1 1 0 1 1 0 1
## [596] 0 1 0 0 0 1 1 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 1 1 0 0
## [631] 0 0 0 0 0 0 1 0 1 1 0 0 0 0 0 0 1 0 0 0 1 0 0 1 0 1 1 0 0 0 1 0 1 1 0 0
## [666] 1 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 1 0 1 0 0 1 1 0 0 0 1 0 1 0 1 0 0 1 0 0
## [701] 0 0 0 0 0 0 0 0 0 0 0 0 0
## Levels: 0 1
```

```
TempTitanicTrain$Parch[TempTitanicTrain$Parch == 0] <- 0
TempTitanicTrain$Parch[TempTitanicTrain$Parch > 0] <- 1
TempTitanicTrain$Parch=factor(TempTitanicTrain$Parch)
(TempTitanicTrain$Parch)
```

```
##   [1] 0 0 0 0 0 0 1 1 0 1 0 0 1 0 0 1 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 1 0
##  [36] 0 1 0 0 0 1 0 0 1 1 0 0 1 0 0 1 0 0 1 0 0 0 0 1 0 0 0 0 0 0 1 1 0 0 0
##  [71] 0 1 0 0 1 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0 0 0 0 0 0
## [106] 0 0 1 0 0 0 0 0 0 0 1 0 1 1 0 0 0 0 1 1 0 0 1 0 0 0 1 1 1 0 0 1 1 0 0
## [141] 1 0 0 0 1 1 1 0 1 0 0 0 0 1 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
## [176] 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 1 1 0 1 0 0 1 1 0 0 1 1
## [211] 1 0 0 1 0 1 0 0 1 1 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1 0 0
## [246] 0 0 1 1 0 1 0 0 0 1 1 0 0 0 1 0 0 0 1 1 0 1 0 0 0 0 0 1 1 0 0 0 0 0 1
## [281] 0 0 1 0 0 1 0 1 0 1 0 0 0 0 0 0 0 0 1 0 1 0 0 0 1 0 0 0 1 0 0 1 0 0 0
## [316] 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 1 0 1 0 0 1 1 0 0 0 0 0 0 0 1 1 1
## [351] 1 0 1 0 0 0 1 1 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 1 1 0 0
## [386] 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1
## [421] 1 1 0 1 0 0 1 1 1 1 0 0 0 0 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0
## [456] 0 0 0 1 1 0 0 1 0 1 0 0 0 0 0 1 0 0 1 0 0 0 0 0 1 0 1 0 1 1 0 1 0 0 0
## [491] 1 0 0 0 0 0 0 0 0 0 0 1 0 0 1 1 0 0 1 1 0 0 0 0 1 0 0 0 1 0 1 0 0 0 0 0
## [526] 0 0 0 1 0 0 0 0 0 0 0 1 1 0 0 1 1 1 1 1 0 0 1 0 1 0 0 0 0 1 0 0 0 1 0 0 0
## [561] 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 1 1 0 0
## [596] 0 1 1 0 0 1 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 1 1 0 0
## [631] 0 0 0 0 0 1 0 1 1 1 0 0 0 0 0 0 0 0 1 0 0 1 0 1 1 0 0 1 1 1 0 1 0 0
## [666] 1 0 0 0 0 0 0 0 0 0 0 1 1 0 1 1 0 1 1 0 1 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0
## [701] 0 1 1 0 0 0 0 1 0 0 0 0
## Levels: 0 1
```

```
TempTitanicTest$SibSp[TempTitanicTest$SibSp == 0] <- 0
TempTitanicTest$SibSp[TempTitanicTest$SibSp > 0] <- 1
TempTitanicTest$SibSp=factor(TempTitanicTest$SibSp)
(TempTitanicTest$SibSp)
```

```
##   [1] 0 1 0 0 1 0 0 1 0 1 0 1 1 1 1 0 0 1 0 1 0 0 1 1 0 0 0 1 1 1 1 0 0 0 0
```

```
##  [36] 0 0 1 0 0 0 0 1 0 1 1 1 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 1 0 0 1 1
##  [71] 0 0 0 1 1 1 0 0 1 0 0 0 1 1 0 1 0 0 0 0 0 0 1 1 1 0 1 0 1 0 0 0 0 0 0
## [106] 0 0 0 0 0 0 1 1 0 1 0 0 1 0 1 0 0 1 0 0 0 0 0 1 0 0 1 0 1 0 0 1 1 0 0
## [141] 1 0 0 0 1 1 1 0 0 1 0 1 1 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 1 1 0 1 1 1
## [176] 0 0 0 0 0 0 0 0 0 1 1 0 1 0 1 1 0 0 1 1 0 1 1 1 0 1 0 0 0 0 0 1 0 0 1
## [211] 0 0 0 1 1 0 1 1 0 0 1 1 1 0 1 0 1 0 0 1 0 0 0 0 0 1 1 0 1 1 0 0 0 0 0
## [246] 1 0 0 1 0 0 0 0 0 0 1 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1
## [281] 0 1 1 1 1 0 0 1 0 0 0 1 0 0 0 1 0 1 0 0 1 0 1 1 0 0 0 1 0 0 0 0 1 1 0
## [316] 1 0 0 0 1 0 0 1 0 1 1 1 1 0 0 0
## Levels: 0 1
```

```
TempTitanicTest$Parch[TempTitanicTest$Parch == 0] <- 0
TempTitanicTest$Parch[TempTitanicTest$Parch > 0] <- 1
TempTitanicTest$Parch=factor(TempTitanicTest$Parch)
(TempTitanicTest$Parch)
```

```
##   [1] 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 1 0 0 0 0 1 0 0 0 0 1
##  [36] 0 0 0 0 0 0 1 0 0 1 1 1 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0
##  [71] 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0
## [106] 0 0 0 0 0 0 1 1 0 1 0 0 0 0 1 1 1 1 0 0 0 0 1 1 0 0 1 0 1 0 0 0 1 1 0
## [141] 0 1 1 0 1 0 1 0 1 0 0 0 1 0 1 0 1 0 0 1 0 1 0 0 0 0 0 0 0 0 1 0 1 1 0
## [176] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 1 0 0 1 0 1 0 0 0 0 0 0 0 0 0 1 1
## [211] 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 1 0 0 1 0 0 0 1 0 1 1 1 1 0 0 0 0 0 0
## [246] 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 0 0 0 1 0 0 0 1 0 0 0 0 0 0 1 0 0 1 1
## [281] 0 0 0 1 1 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 1 0 0 1 0 0 0 1 0 1 1 0 1 0 0
## [316] 1 0 0 0 0 1 0 0 0 0 1 1 0 0 0 0
## Levels: 0 1
```

```
(freq=table(TempTitanicTrain$SibSp))
```

```
##
##   0   1
## 469 243
```

```
(freq=table(TempTitanicTest$SibSp))
```

```
##
##   0   1
## 213 118
```

```
(freq=table(TempTitanicTrain$Parch))
```

```
##
##   0   1
## 519 193
```

```
(freq=table(TempTitanicTest$Parch))
```

```
##
##   0   1
## 246  85
```

```
## Continue the process...look at the currect str
(str(TempTitanicTest))
```

```
## 'data.frame':    331 obs. of  7 variables:
##  $ Pclass  : Factor w/ 3 levels "1","2","3": 3 3 2 3 3 3 3 2 3 3 ...
##  $ Sex     : Factor w/ 2 levels "female","male": 2 1 2 2 1 2 1 2 1 2 ...
```

```
##  $ Age     : Factor w/ 3 levels "1","2","3": 2 3 3 2 1 1 2 2 1 1 ...
##  $ SibSp   : Factor w/ 2 levels "0","1": 1 2 1 1 2 1 1 2 1 2 ...
##  $ Parch   : Factor w/ 2 levels "0","1": 1 1 1 1 2 1 1 2 1 1 ...
##  $ Fare    : num  7.83 7 9.69 8.66 12.29 ...
##  $ Embarked: Factor w/ 3 levels "C","Q","S": 2 3 2 3 3 3 2 3 1 3 ...
```

```
## NULL
```

```
(str(TempTitanicTrain))
```

```
## 'data.frame':    712 obs. of  8 variables:
##  $ Survived: Factor w/ 2 levels "0","1": 1 2 2 2 1 1 1 2 2 2 ...
##  $ Pclass  : Factor w/ 3 levels "1","2","3": 3 1 3 1 3 1 3 3 2 3 ...
##  $ Sex     : Factor w/ 2 levels "female","male": 2 1 1 1 2 2 2 1 1 1 ...
##  $ Age     : Factor w/ 3 levels "1","2","3": 1 2 2 2 2 3 1 2 1 3 ...
##  $ SibSp   : Factor w/ 2 levels "0","1": 2 2 1 2 1 1 2 1 2 2 ...
##  $ Parch   : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 2 2 1 2 ...
##  $ Fare    : num  7.25 71.28 7.92 53.1 8.05 ...
##  $ Embarked: Factor w/ 3 levels "C","Q","S": 3 1 3 3 3 3 3 3 1 3 ...
```

```
## NULL
```

```
## The last step is to discretize the Fare
(freq=table(TempTitanicTrain$Fare))
```

```
##
##        0   4.0125        5   6.2375   6.4375     6.45   6.4958     6.75
##        7        1        1        1        1        1        2        2
##    6.975   7.0458     7.05   7.0542    7.125   7.1417    7.225   7.2292
##        2        1        6        2        4        1        6        8
##     7.25   7.4958   7.5208     7.55     7.65   7.7333   7.7417     7.75
##       10        3        1        2        4        2        1       14
##    7.775   7.7958      7.8   7.8542    7.875   7.8792   7.8875   7.8958
##       14        6        1       13        1        1        1       23
##    7.925   8.0292     8.05   8.1583      8.3   8.3625   8.4042   8.4333
##       18        1       29        1        1        1        1        1
##   8.5167   8.6542   8.6625   8.6833     8.85        9   9.2167    9.225
##        1        1       12        1        1        2        1        2
##     9.35    9.475   9.4833      9.5   9.5875    9.825   9.8375   9.8417
##        2        1        1        8        2        2        1        1
##   9.8458  10.1708  10.4625     10.5  10.5167  11.1333  11.2417     11.5
##        1        1        2       24        1        3        2        4
##       12   12.275  12.2875    12.35   12.475   12.525    12.65   12.875
##        1        1        1        2        4        1        1        1
##       13  13.4167     13.5  13.7917  13.8583       14  14.1083     14.4
##       41        1        4        1        1        1        1        2
##  14.4542  14.4583     14.5       15  15.0458  15.2458     15.5    15.55
##        6        1        5        1        1        2        2        1
##  15.7417    15.75    15.85     15.9       16     16.1     16.7     17.4
##        2        1        4        2        1        6        2        1
##     17.8       18    18.75  18.7875  19.2583     19.5  20.2125    20.25
##        2        3        3        2        4        2        2        2
##   20.525   20.575       21   21.075   22.025   22.525       23       24
##        3        2        6        4        1        1        4        2
##    24.15  25.5875  25.9292       26    26.25  26.2833  26.2875  26.3875
##        5        1        2       30        6        1        3        1
```

```
##    26.55       27  27.7208    27.75     27.9     28.5  28.7125       29
##       13        2        3        4        6        1        1        2
##   29.125     29.7       30  30.0708     30.5  30.6958       31   31.275
##        5        2        5        2        4        1        2        7
##  31.3875  32.3208     32.5       33     33.5  34.0208   34.375  34.6542
##        4        1        1        2        1        1        4        1
##     35.5    36.75  37.0042     38.5       39     39.4     39.6  39.6875
##        3        2        2        1        4        1        1        6
##   40.125  41.5792     46.9     47.1     49.5  49.5042  50.4958  51.4792
##        1        3        6        1        1        2        1        1
##  51.8625       52  52.5542     53.1       55  55.4417     55.9  56.4958
##        1        5        3        5        1        1        2        4
##  56.9292       57  57.9792     59.4   61.175  61.3792  61.9792  63.3583
##        2        2        2        1        1        1        1        1
##       65     66.6     69.3       71  71.2833     73.5    75.25  76.2917
##        2        2        2        2        1        5        1        1
##  76.7292  77.2875  77.9583  78.2667    78.85     79.2    79.65  81.8583
##        3        2        3        2        2        3        3        1
##  82.1708  83.1583   83.475     86.5  89.1042       90  91.0792     93.5
##        1        3        2        3        1        4        2        2
##  106.425    108.9 110.8833  113.275      120   133.65    134.5 135.6333
##        2        2        3        3        4        1        2        3
## 146.5208   151.55 153.4625 164.8667 211.3375    211.5  227.525 247.5208
##        1        4        3        2        3        1        3        2
##  262.375      263 512.3292
##        2        4        3
```

```
(freq=table(TempTitanicTest$Fare))
```

```
##
##        0   3.1708   6.4958     6.95        7     7.05    7.225   7.2292
##        1        1        1        1        1        1        7        5
##     7.25   7.2833     7.55   7.5792   7.6292     7.65    7.725   7.7333
##        4        1        1        1        1        2        1        2
##     7.75    7.775   7.7958   7.8208   7.8292     7.85   7.8542   7.8792
##        6        9        4        1        1        1        8        3
##   7.8958    7.925     8.05   8.5167   8.6625   8.9625    9.225    9.325
##        7        5        9        1        8        1        1        1
##     9.35      9.5   9.6875     10.5     11.5  12.1833  12.2875    12.35
##        1        3        1       11        2        2        1        2
##  12.7375       13  13.4167     13.5   13.775  13.8583  13.8625     13.9
##        1       17        1        3        3        2        1        2
##  14.1083     14.4  14.4542     14.5  15.0333  15.0458     15.1  15.2458
##        1        1        1        2        1        1        1        3
##    15.55  15.7417    15.75     15.9       16     16.1     16.7     17.4
##        1        1        1        1        1        2        1        1
##       18  20.2125    20.25   20.575       21   21.075   22.025   22.525
##        1        1        1        2        7        1        2        2
##       23    24.15     25.7       26    26.55  27.4458  27.7208    27.75
##        3        1        1       18        5        1        5        1
##     28.5  28.5375       29   29.125     29.7       30     30.5  31.3875
##        1        1        1        1        2        1        1        3
##     31.5  31.6792     32.5   34.375    36.75  37.0042       39     39.4
##        3        1        2        1        2        1        3        1
##  39.6875  41.5792     42.4     42.5     45.5     46.9     47.1       50
```

```
##        1        1        1        1        1        2        1        1
##  50.4958  51.4792  51.8625       52  52.5542     53.1  55.4417    57.75
##        1        1        1        1        1        1        3        2
##     59.4       60   61.175  61.3792  61.9792  63.3583       65    69.55
##        3        2        1        1        1        1        3        1
##  71.2833     73.5  75.2417    75.25  76.2917    78.85     79.2  81.8583
##        1        2        2        1        1        1        2        2
##  82.2667  83.1583       90     93.5  106.425    108.9    134.5 135.6333
##        2        3        1        2        1        1        3        1
## 136.7792 146.5208   151.55 164.8667 211.3375    211.5 221.7792  227.525
##        2        1        2        2        1        4        3        1
## 247.5208  262.375      263 512.3292
##        1        5        2        1
```

```
## We can see that the range is within 0 - 550
## We could also have gotten the min and max

## Place into three groups: 1: Low, 2: Mediun, 3: High  Fare rate
TempTitanicTrain$Fare[TempTitanicTrain$Fare <= 10] <- 1
TempTitanicTrain$Fare[TempTitanicTrain$Fare > 10 & TempTitanicTrain$Fare <=30] <- 2
TempTitanicTrain$Fare[TempTitanicTrain$Fare > 30] <- 3
TempTitanicTrain$Fare=factor(TempTitanicTrain$Fare)
#(TempTitanicTrain$Fare)

TempTitanicTest$Fare[TempTitanicTest$Fare <= 10] <- 1
TempTitanicTest$Fare[TempTitanicTest$Fare > 10 & TempTitanicTest$Fare <=30] <- 2
TempTitanicTest$Fare[TempTitanicTest$Fare > 30] <- 3
TempTitanicTest$Fare=factor(TempTitanicTest$Fare)
#(TempTitanicTest$Fare)

(freq=table(TempTitanicTrain$Fare))
```

```
##
##   1   2   3
## 236 275 201
```

```
(freq=table(TempTitanicTest$Fare))
```

```
##
##   1   2   3
## 102 127 102
```

```
(str(TempTitanicTest))
```

```
## 'data.frame':    331 obs. of  7 variables:
##  $ Pclass  : Factor w/ 3 levels "1","2","3": 3 3 2 3 3 3 3 2 3 3 ...
##  $ Sex     : Factor w/ 2 levels "female","male": 2 1 2 2 1 2 1 2 1 2 ...
##  $ Age     : Factor w/ 3 levels "1","2","3": 2 3 3 2 1 1 2 2 1 1 ...
##  $ SibSp   : Factor w/ 2 levels "0","1": 1 2 1 1 2 1 1 2 1 2 ...
##  $ Parch   : Factor w/ 2 levels "0","1": 1 1 1 1 2 1 1 2 1 1 ...
##  $ Fare    : Factor w/ 3 levels "1","2","3": 1 1 1 1 2 1 1 2 1 2 ...
##  $ Embarked: Factor w/ 3 levels "C","Q","S": 2 3 2 3 3 3 2 3 1 3 ...
```

```
## NULL
```

```
(str(TempTitanicTrain))
```

```
## 'data.frame':    712 obs. of  8 variables:
##  $ Survived: Factor w/ 2 levels "0","1": 1 2 2 2 1 1 1 2 2 2 ...
##  $ Pclass  : Factor w/ 3 levels "1","2","3": 3 1 3 1 3 1 3 3 2 3 ...
##  $ Sex     : Factor w/ 2 levels "female","male": 2 1 1 1 2 2 2 1 1 1 ...
##  $ Age     : Factor w/ 3 levels "1","2","3": 1 2 2 2 2 3 1 2 1 3 ...
##  $ SibSp   : Factor w/ 2 levels "0","1": 2 2 1 2 1 1 2 1 2 2 ...
##  $ Parch   : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 2 2 1 2 ...
##  $ Fare    : Factor w/ 3 levels "1","2","3": 1 3 1 3 1 3 2 2 3 2 ...
##  $ Embarked: Factor w/ 3 levels "C","Q","S": 3 1 3 3 3 3 3 3 1 3 ...
```

```
## NULL
```

```
## Our datasets are finally clean and ready to use.
## Let's move the temp to new dataframe names
CleanTest <- TempTitanicTest
CleanTrain <- TempTitanicTrain
```

Decision Trees

The goal is to create decision tree from the training data that can classify a row as survive or not survive.Then, we will use the testing data to see how well the Decision Tree works.

```
######################################## BUILD Decision Trees ----------------------------
#install.packages("rpart")
#install.packages('rattle')
#install.packages('rpart.plot')
#install.packages('RColorBrewer')
#install.packages("Cairo")
#install.packages("COREIearn")
library(rpart)
library(rattle)
```

```
## Rattle: A free graphical interface for data science with R.
## Version 5.2.0 Copyright (c) 2006-2018 Togaware Pty Ltd.
## Type 'rattle()' to shake, rattle, and roll your data.
```

```
library(rpart.plot)
library(RColorBrewer)
library(Cairo)


# rpart will train the decision tree based on the train data.
fit <- rpart(CleanTrain$Survived ~ ., data = CleanTrain, method="class")
summary(fit)
```

```
## Call:
## rpart(formula = CleanTrain$Survived ~ ., data = CleanTrain, method = "class")
##   n= 712
##
##           CP nsplit rel error    xerror       xstd
## 1 0.45486111      0 1.0000000 1.0000000 0.04547228
## 2 0.02777778      1 0.5451389 0.5451389 0.03841173
## 3 0.01041667      2 0.5173611 0.5555556 0.03867204
## 4 0.01000000      5 0.4826389 0.5277778 0.03796499
##
```

```
## Variable importance
##      Sex   Pclass     Fare    Parch Embarked      Age    SibSp
##       62       18        9        6        2        2        1
##
## Node number 1: 712 observations,    complexity param=0.4548611
##   predicted class=0  expected loss=0.4044944  P(node) =1
##     class counts:   424   288
##    probabilities: 0.596 0.404
##   left son=2 (453 obs) right son=3 (259 obs)
##   Primary splits:
##       Sex      splits as  RL,  improve=98.826010, (0 missing)
##       Pclass   splits as  RRL, improve=38.578210, (0 missing)
##       Fare     splits as  LRR, improve=29.769390, (0 missing)
##       Embarked splits as  RLL, improve=13.133150, (0 missing)
##       Parch    splits as  LR,  improve= 9.560419, (0 missing)
##   Surrogate splits:
##       Parch splits as  LR, agree=0.669, adj=0.089, (0 split)
##
## Node number 2: 453 observations
##   predicted class=0  expected loss=0.205298  P(node) =0.636236
##     class counts:   360    93
##    probabilities: 0.795 0.205
##
## Node number 3: 259 observations,    complexity param=0.02777778
##   predicted class=1  expected loss=0.2471042  P(node) =0.363764
##     class counts:    64   195
##    probabilities: 0.247 0.753
##   left son=6 (102 obs) right son=7 (157 obs)
##   Primary splits:
##       Pclass   splits as  RRL, improve=28.7162300, (0 missing)
##       Fare     splits as  LRR, improve= 4.5583940, (0 missing)
##       Embarked splits as  RLL, improve= 3.5307820, (0 missing)
##       SibSp    splits as  RL,  improve= 1.1558980, (0 missing)
##       Parch    splits as  RL,  improve= 0.5907937, (0 missing)
##   Surrogate splits:
##       Fare     splits as  LRR, agree=0.764, adj=0.402, (0 split)
##       Age      splits as  LRR, agree=0.645, adj=0.098, (0 split)
##       Embarked splits as  RLR, agree=0.637, adj=0.078, (0 split)
##
## Node number 6: 102 observations,    complexity param=0.01041667
##   predicted class=0  expected loss=0.4607843  P(node) =0.1432584
##     class counts:    55    47
##    probabilities: 0.539 0.461
##   left son=12 (13 obs) right son=13 (89 obs)
##   Primary splits:
##       Fare     splits as  RRL, improve=2.8072770, (0 missing)
##       SibSp    splits as  RL,  improve=2.1381380, (0 missing)
##       Embarked splits as  RLL, improve=1.9508090, (0 missing)
##       Age      splits as  RRL, improve=0.9170437, (0 missing)
##       Parch    splits as  RL,  improve=0.3529412, (0 missing)
##
## Node number 7: 157 observations
##   predicted class=1  expected loss=0.05732484  P(node) =0.2205056
##     class counts:     9   148
```

```
##    probabilities: 0.057 0.943
##
## Node number 12: 13 observations
##   predicted class=0  expected loss=0.1538462  P(node) =0.01825843
##     class counts:    11     2
##    probabilities: 0.846 0.154
##
## Node number 13: 89 observations,    complexity param=0.01041667
##   predicted class=1  expected loss=0.494382  P(node) =0.125
##     class counts:    44    45
##    probabilities: 0.494 0.506
##   left son=26 (73 obs) right son=27 (16 obs)
##   Primary splits:
##       Embarked splits as  RLL, improve=1.290615000, (0 missing)
##       SibSp    splits as  RL,  improve=0.969248300, (0 missing)
##       Age      splits as  RLL, improve=0.293143600, (0 missing)
##       Fare     splits as  RL-, improve=0.145804800, (0 missing)
##       Parch    splits as  LR,  improve=0.008667737, (0 missing)
##
## Node number 26: 73 observations,    complexity param=0.01041667
##   predicted class=0  expected loss=0.4657534  P(node) =0.1025281
##     class counts:    39    34
##    probabilities: 0.534 0.466
##   left son=52 (27 obs) right son=53 (46 obs)
##   Primary splits:
##       SibSp    splits as  RL,  improve=1.50268000, (0 missing)
##       Age      splits as  RRL, improve=0.50363050, (0 missing)
##       Fare     splits as  RL-, improve=0.37099940, (0 missing)
##       Parch    splits as  RL,  improve=0.05043379, (0 missing)
##       Embarked splits as  -RL, improve=0.02717982, (0 missing)
##   Surrogate splits:
##       Fare  splits as  RL-, agree=0.740, adj=0.296, (0 split)
##       Parch splits as  RL,  agree=0.671, adj=0.111, (0 split)
##
## Node number 27: 16 observations
##   predicted class=1  expected loss=0.3125  P(node) =0.02247191
##     class counts:     5    11
##    probabilities: 0.312 0.688
##
## Node number 52: 27 observations
##   predicted class=0  expected loss=0.3333333  P(node) =0.03792135
##     class counts:    18     9
##    probabilities: 0.667 0.333
##
## Node number 53: 46 observations
##   predicted class=1  expected loss=0.4565217  P(node) =0.06460674
##     class counts:    21    25
##    probabilities: 0.457 0.543
```

```
# Once trained we can test our tree on the test data.
predicted= predict(fit,CleanTest, type="class")
(head(predicted,n=10))
```

```
##  1  2  3  4  5  6  7  8  9 10
##  0  0  0  0  0  0  1  0  1  0
```

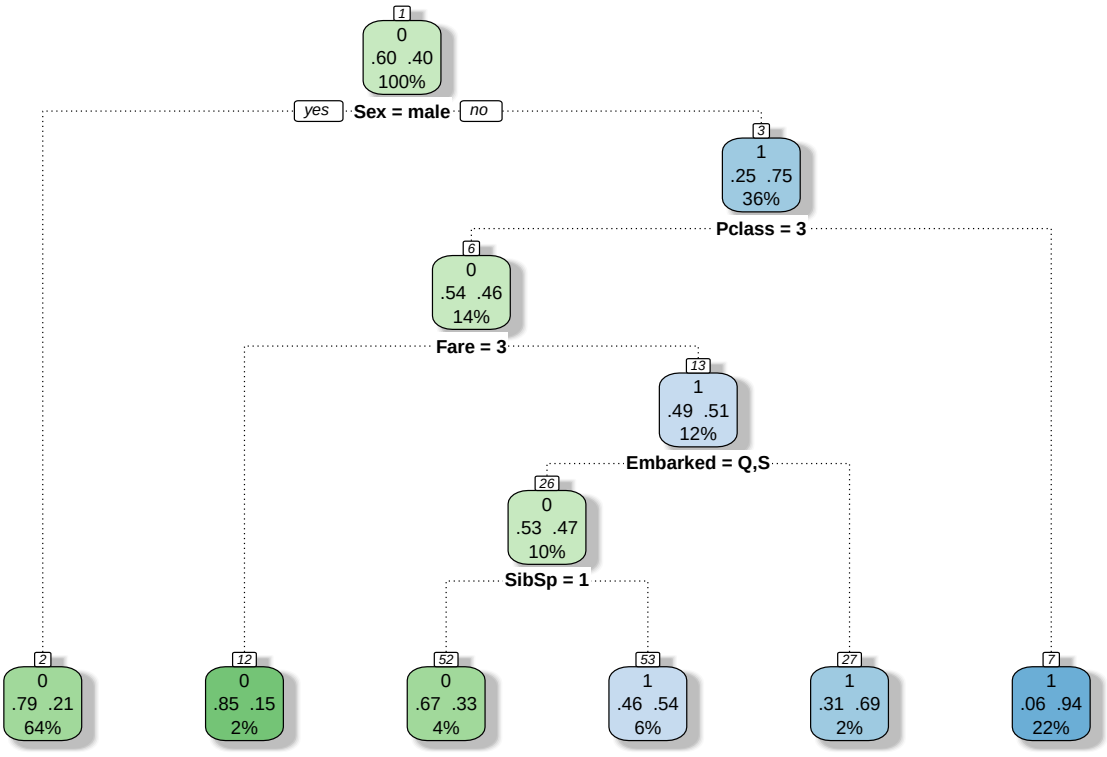
```
## Levels: 0 1
```

```
(head(CleanTest, n=10))
```

```
##    Pclass    Sex Age SibSp Parch Fare Embarked
## 1       3   male   2     0     0    1        Q
## 2       3 female   3     1     0    1        S
## 3       2   male   3     0     0    1        Q
## 4       3   male   2     0     0    1        S
## 5       3 female   1     1     1    2        S
## 6       3   male   1     0     0    1        S
## 7       3 female   2     0     0    1        Q
## 8       2   male   2     1     1    2        S
## 9       3 female   1     0     0    1        C
## 10      3   male   1     1     0    2        S
```

In the next segment we will illustrate the decision tree.

```
fancyRpartPlot(fit)
```




```
submit <- data.frame(PassengerGender = CleanTest$Sex, Survived = predicted)
(head(submit, n=10))
```

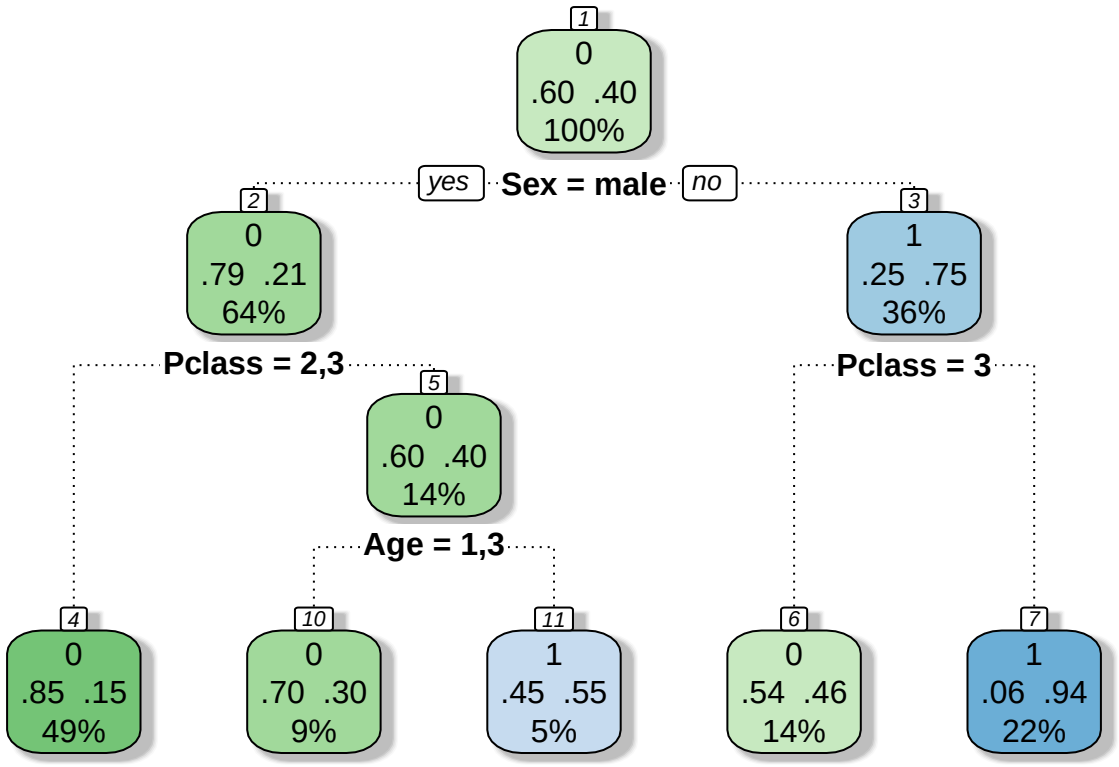
```
##    PassengerGender Survived
## 1             male        0
## 2           female        0
## 3             male        0
## 4             male        0
## 5           female        0
```

```
## 6              male        0
## 7            female        1
## 8              male        0
## 9            female        1
## 10             male        0
```

```
write.csv(submit, file = "TitanicPrediction.csv", row.names = FALSE)

## Let's reduce the tree size
fit2 <- rpart(Survived ~ Pclass + Sex + Age,
              data=CleanTrain,
              method="class",
              control=rpart.control(minsplit=2, cp=0))

fancyRpartPlot(fit2)
```




```
## Save the Decision Tree as a jpg image
jpeg("DecisionTree_Titanic.jpg")
fancyRpartPlot(fit2)
dev.off()
```

```
## pdf
##   2
```