Model for Titanic Data

install.packages('rattle') install.packages('rpart.plot') install.packages('RColorBrewer') install.packages('ggplot2') install.packages('rpart') install.packages('randomForest') install.packages('caret') install.packages('e1071')

library(ggplot2)

## Warning: package 'ggplot2' was built under R version 3.1.3

library(randomForest)

## Warning: package 'randomForest' was built under R version 3.1.2

## randomForest 4.6-10  
## Type rfNews() to see new features/changes/bug fixes.

library(rpart)

## Warning: package 'rpart' was built under R version 3.1.2

library(rattle)

## Warning: package 'rattle' was built under R version 3.1.2

## Rattle: A free graphical interface for data mining with R.  
## Version 3.4.1 Copyright (c) 2006-2014 Togaware Pty Ltd.  
## Type 'rattle()' to shake, rattle, and roll your data.

library(rpart.plot)

## Warning: package 'rpart.plot' was built under R version 3.1.2

library(RColorBrewer)

## Warning: package 'RColorBrewer' was built under R version 3.1.2

library(caret)

## Warning: package 'caret' was built under R version 3.1.2

## Loading required package: lattice

library(e1071)

## Warning: package 'e1071' was built under R version 3.1.2

#Read the train data.  
trn <- read.csv("ttn\_train.csv",header=TRUE,stringsAsFactors=FALSE)  
#Check how many rows and columns in the data  
dim(trn)

## [1] 891 12

# Check the data types and what sort of values are there in the data set  
str(trn)

## '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 : chr "Braund, Mr. Owen Harris" "Cumings, Mrs. John Bradley (Florence Briggs Thayer)" "Heikkinen, Miss. Laina" "Futrelle, Mrs. Jacques Heath (Lily May Peel)" ...  
## $ Sex : chr "male" "female" "female" "female" ...  
## $ 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 : chr "A/5 21171" "PC 17599" "STON/O2. 3101282" "113803" ...  
## $ Fare : num 7.25 71.28 7.92 53.1 8.05 ...  
## $ Cabin : chr "" "C85" "" "C123" ...  
## $ Embarked : chr "S" "C" "S" "S" ...

#Convert the columns such as Pclass,Sex,Sibsp etc to Factor columns as these have very few unique values.  
  
trn$Sex <- as.factor(as.character(trn$Sex))  
trn$Survived <- as.factor(as.character(trn$Survived))  
trn$Pclass <- as.factor(as.character(trn$Pclass))  
trn$SibSp <- as.factor(as.character(trn$SibSp))  
trn$Parch <- as.factor(as.character(trn$Parch))  
trn$Embarked <- ifelse(trn$Embarked =="","S",trn$Embarked)  
trn$Embarked <- as.factor(as.character(trn$Embarked))  
summary(trn$Embarked)

## C Q S   
## 168 77 646

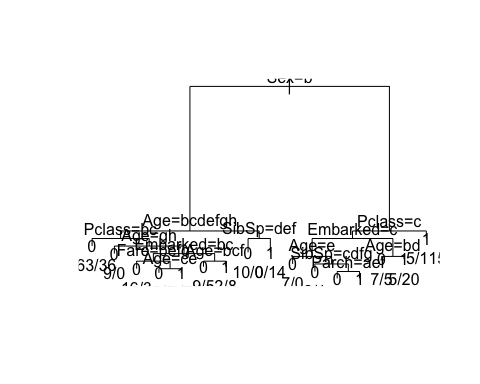
#Only Age has 177 misisng records. I will impute these with median values of age. There are other ways to do as well but I will keep it simple and use median age  
  
med\_age <- median(trn$Age,na.rm=TRUE)  
#impute the data  
trn$Age <- ifelse(is.na(trn$Age),med\_age,trn$Age)  
  
  
summary(trn)

## PassengerId Survived Pclass Name Sex   
## Min. : 1.0 0:549 1:216 Length:891 female:314   
## 1st Qu.:223.5 1:342 2:184 Class :character male :577   
## Median :446.0 3:491 Mode :character   
## Mean :446.0   
## 3rd Qu.:668.5   
## Max. :891.0   
##   
## Age SibSp Parch Ticket Fare   
## Min. : 0.42 0:608 0:678 Length:891 Min. : 0.00   
## 1st Qu.:22.00 1:209 1:118 Class :character 1st Qu.: 7.91   
## Median :28.00 2: 28 2: 80 Mode :character Median : 14.45   
## Mean :29.36 3: 16 3: 5 Mean : 32.20   
## 3rd Qu.:35.00 4: 18 4: 4 3rd Qu.: 31.00   
## Max. :80.00 5: 5 5: 5 Max. :512.33   
## 8: 7 6: 1   
## Cabin Embarked  
## Length:891 C:168   
## Class :character Q: 77   
## Mode :character S:646   
##   
##   
##   
##

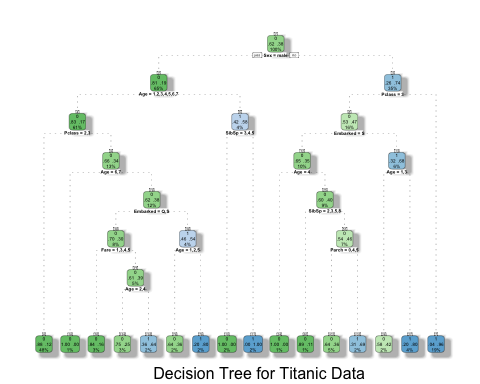
#make 10 buckets of the age.  
trn$Age <- as.factor(as.character(floor(trn$Age/10)))  
  
#make buckets of the fare of size 50.  
trn$Fare <- as.factor(as.character(floor(trn$Fare/50)))  
  
#It provides meadin,mean and other summary stats. I could see that 687 values for cabin are blank andfor Embarked there are 2 such values.  
# I will remove cabin column as well as Ticket columns form the data set  
  
trn$Cabin <- NULL  
trn$Ticket <- NULL  
trn$PassengerId <- NULL  
trn$Name <- NULL  
  
summary(trn)

## Survived Pclass Sex Age SibSp Parch Fare   
## 0:549 1:216 female:314 2 :397 0:608 0:678 0 :730   
## 1:342 2:184 male :577 3 :167 1:209 1:118 1 :108   
## 3:491 1 :102 2: 28 2: 80 10: 3   
## 4 : 89 3: 16 3: 5 2 : 24   
## 0 : 62 4: 18 4: 4 3 : 9   
## 5 : 48 5: 5 5: 5 4 : 11   
## (Other): 26 8: 7 6: 1 5 : 6   
## Embarked  
## C:168   
## Q: 77   
## S:646   
##   
##   
##   
##

# Now let us look at some of the graphs. In Data analysis we found that age Pclass and sex were quite good columns for model and let us visually see them.  
  
trn\_idx<- createDataPartition(trn$Survived, times=1,p=0.7,list=FALSE)  
  
trn\_data <- trn[trn\_idx,]  
tst\_data <- trn[-trn\_idx,]  
  
frml <- as.formula('Survived ~ .')  
  
#create a decision tree and then plot it and see how it looks like  
rt <- rpart(frml, trn\_data, control = rpart.control(cp = 0.005))  
plot(rt)  
text(rt, use.n = TRUE)



#We can see the tree and it is as what we expected sex being most important then p class and age etc. But tree doesnt look good.  
#let us use other methods to show the trees.  
  
fancyRpartPlot(rt,sub="Decision Tree for Titanic Data")



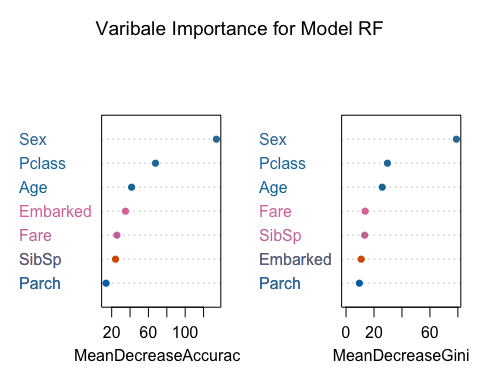
#Now this decision tree looks really nice.  
  
#Now let us use random forest  
  
rf <- randomForest(frml,data =trn\_data,importance=TRUE,ntree=1000,mtry=4,nodesize=10)  
rf

##   
## Call:  
## randomForest(formula = frml, data = trn\_data, importance = TRUE, ntree = 1000, mtry = 4, nodesize = 10)   
## Type of random forest: classification  
## Number of trees: 1000  
## No. of variables tried at each split: 4  
##   
## OOB estimate of error rate: 16%  
## Confusion matrix:  
## 0 1 class.error  
## 0 360 25 0.06493506  
## 1 75 165 0.31250000

importance(rf)

## 0 1 MeanDecreaseAccuracy MeanDecreaseGini  
## Pclass 47.56215 49.964339 67.68920 29.605150  
## Sex 94.20706 120.098199 133.94593 78.931143  
## Age 30.91374 29.959217 41.66896 25.875964  
## SibSp 28.57512 -1.092670 24.15399 13.303568  
## Parch 13.11412 3.538524 13.86651 9.536308  
## Fare 23.16542 6.511525 25.72154 13.668319  
## Embarked 12.64010 37.051130 35.02233 10.841372

#The Sex is most important followed by Paclass and Age and Embarked.Now let us plot these.  
  
colset <- c("#0072B2", "#D55E00" ,"#CC79A7","#DD79A7","#1179A7","#2279A7","#3379A7","#4479A7","#5579A7")  
varImpPlot(rf,color=colset,gcolor=colset,pch=16,main="Varibale Importance for Model RF")



#Now let us predict and see hows it doing  
pred <- predict(rf,tst\_data)  
tb <- table(pred,tst\_data$Survived)  
#check the results  
  
confusionMatrix(tb,positive='1')

## Confusion Matrix and Statistics  
##   
##   
## pred 0 1  
## 0 150 35  
## 1 14 67  
##   
## Accuracy : 0.8158   
## 95% CI : (0.7639, 0.8605)  
## No Information Rate : 0.6165   
## P-Value [Acc > NIR] : 1.592e-12   
##   
## Kappa : 0.5946   
## Mcnemar's Test P-Value : 0.004275   
##   
## Sensitivity : 0.6569   
## Specificity : 0.9146   
## Pos Pred Value : 0.8272   
## Neg Pred Value : 0.8108   
## Prevalence : 0.3835   
## Detection Rate : 0.2519   
## Detection Prevalence : 0.3045   
## Balanced Accuracy : 0.7857   
##   
## 'Positive' Class : 1   
##

# Now let us try on actual test data

tst\_dt <- read.csv("ttn\_test.csv",header=TRUE,stringsAsFactors=FALSE) tst\_dtSex)) tst\_dtSurvived)) tst\_dtPclass)) tst\_dtSibSp)) tst\_dtParch)) tst\_dtEmbarked =="","S",tst\_dtEmbarked <- as.factor(as.character(tst\_dtFare <- ifelse(is.na(tst\_dtFare)

# parch has extra level so fix that

tst\_dtParch==9,6,tst\_dtParch <- ifelse(tst\_dtParch) tst\_dtParch))

# replcae the missing age values with med\_age

tst\_dtAge),med\_age,tst\_dt$Age)

# make 10 buckets of the age.

tst\_dtAge/10)))

# make buckets of the fare of size 50.

tst\_dtFare/50)))

# remove unwanted columns

tst\_dtTicket <- NULL tst\_dt$Name <- NULL

levels(tst\_dtParch) levels(tst\_dtAge) levels(tst\_dtFare) levels(tst\_dtEmbarked)

frml <- as.formula('Survived ~ .') #use full training data now for final training rf <- randomForest(frml,data =trn,importance=TRUE,ntree=1000,mtry=4,nodesize=5) rf importance(rf) varImpPlot(rf,color=colset,gcolor=colset,pch=16,main="Varibale Importance for Model RF") pred <- predict(rf,tst\_dt) tst\_dt$Survived <- pred td <- tst\_dt[,c('PassengerId','Survived')]

head(td)

write.csv(td,file='td.csv',row.names=FALSE)