2. In the last class, I covered C4.5 decision tree classification technique where the dependent variable is categorical and splitting is based on information gain. Please read about decision tree regression where the dependent variable is continuous and splitting is based on standard deviation reduction: http://www.saedsayad.com/decision\_tree\_reg.htm. You will now have assignments on both. Implement decision tree classification with Titanic dataset (predict Survived) and decision tree regression Energy efficiency Dataset (outcome y1 or y2) in R. You can use rpart library but there are other options.

**decision tree classification**

library(rpart)

titanic\_data <- read.csv("A:\\sem-2 - spring-2017\\Advance-Data-Science\\Assignments\\Assignment-3\\Titanic\_train.csv", header = TRUE, sep = ",")

head(titanic\_data)

table(titanic\_data$Survived)

titanic\_data\_tree <- rpart(Survived ~ Pclass + Sex + Age + SibSp + Parch + Fare + Embarked, data = titanic\_data, method = "class")

library(rattle)

library(rpart.plot)

library(RColorBrewer)

fancyRpartPlot(titanic\_data\_tree)

prp(titanic\_data\_tree)

prediction <- predict(titanic\_data\_tree, titanic\_data, type = "class")

solution <- data.frame(Pclass = titanic\_data$Pclass, Age = titanic\_data$Age, Sex= titanic\_data$Sex,

SibSp = titanic\_data$SibSp ,Parch = titanic\_data$Parch, Fare = titanic\_data$Fare,

Embarked= titanic\_data$Embarked, Survived = prediction)

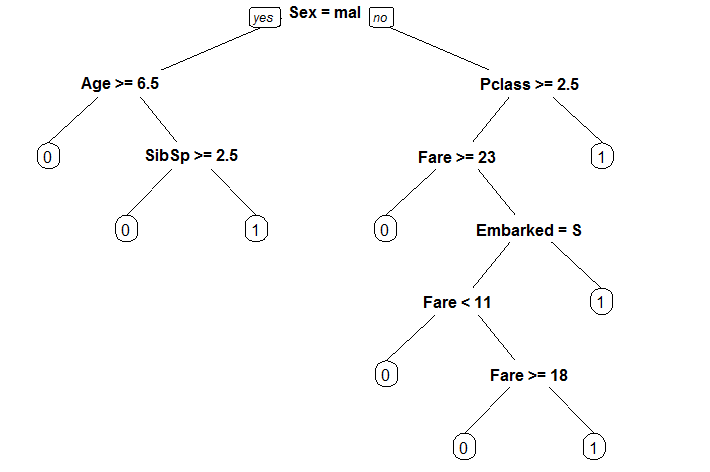
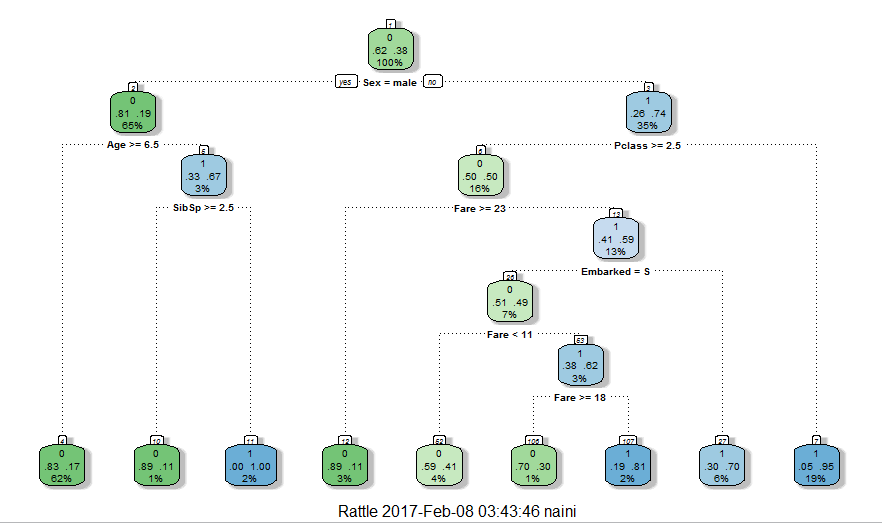
my\_tree\_prediction <- rpart(Survived ~ Pclass + Sex + Age + SibSp + Parch + Fare + Embarked,

data = solution, method = "class", control = rpart.control(minsplit = 50, cp = 0))

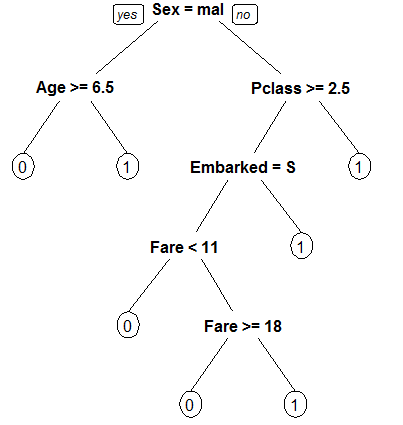
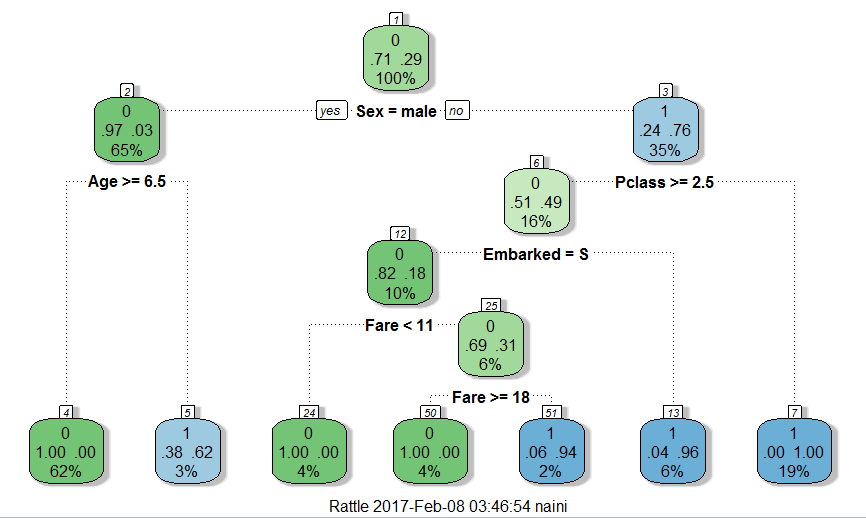
prp(my\_tree\_prediction)

fancyRpartPlot(my\_tree\_prediction)

Train Data:

Prediction:

**decision tree regression**

library(rpart)

Energy\_efficiency <- read.csv("A:\\sem-2 - spring-2017\\Advance-Data-Science\\Assignments\\Assignment-3\\Energy\_Efficiency\_data.csv", header = TRUE, sep = ",")

head(Energy\_efficiency)

data\_y1 <- rpart(Y1~X1 + X2 + X3 + X4 +X5 + X6 +X7 +X8, method="anova", data=Energy\_efficiency)

data\_y2 <- rpart(Y2~X1 + X2 + X3 + X4 +X5 + X6 +X7 +X8, method="anova", data=Energy\_efficiency)

printcp(data\_y1)

plotcp(data\_y1)

printcp(data\_y2)

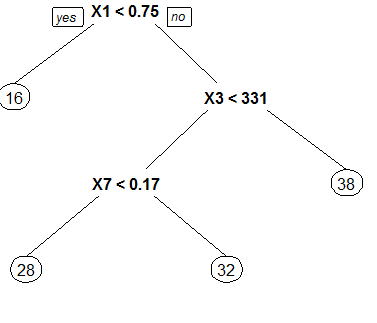
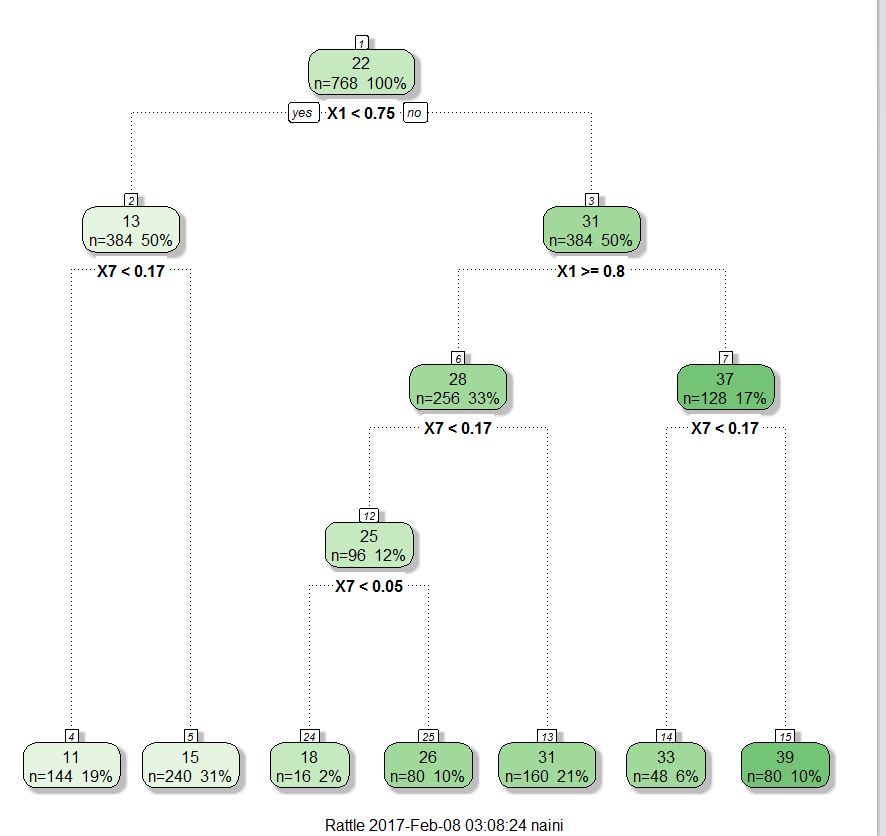
plotcp(data\_y2)

library(rattle)

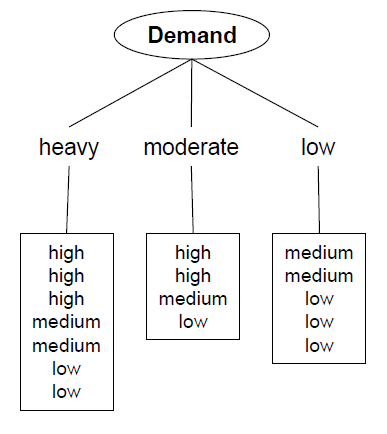
library(rpart.plot)

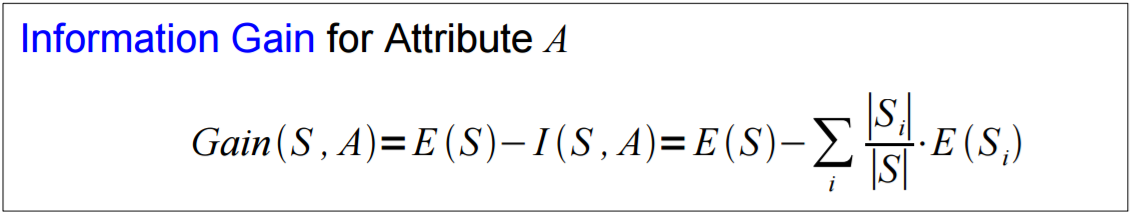
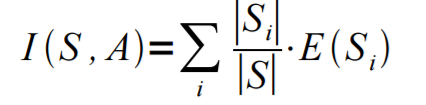
library(RColorBrewer)

prp(data\_y2) fancyRpartPlot(data\_y1)

1. Read pruning, entropy, and information gain in decision tree. Compute entropy manually from slide, put your steps and result in word file.



**Demand = Heavy: 3-high, 2-medium, 2-low**

E (Demand = Heavy) = -(3/7)(log2(3/7) - (2/7)(log2(2/7) - (2/7)(log2(2/7)

= 1.552

**Demand = Moderate: 2-high, 1-medium, 1-low**

E (Demand = Moderate) = -(2/4)(log2(2/4) - (1/4)(log2(1/4) - (1/4)(log2(2/4)

= 1.5

**Demand = Low: 2-medium, 3-low**

E (Demand = Low) = -(2/5)(log2(2/5) - (3/5)(log2(3/5)

= 0.968

**I(S, Demand) = (7/16)( 1.552) + (4/16)( 1.5) +(5/16)( 0.968)**

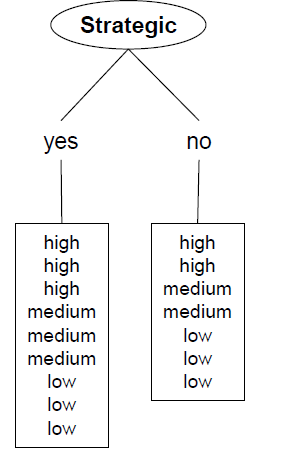
**= 1.3565**

**E (S) = -(5/16)(log2(5/16) - (5/16)(log2(5/16) - (6/16)(log2(6/16)**

**= 1.578**

**Gain(S, A) = E (S) - I(S, A)**

**= 0.22**



**Strategic = yes: 3-high, 3-medium, 3-low**

E (Strategic = yes) = -(3/9) (log2(3/9) -(3/9) (log2(3/9) -(3/9) (log2(3/9)

= 1.584

**Strategic = no: 2-high, 2-medium, 3-low**

E (Strategic = no) = -(2/7) (log2(2/7) - (2/7) (log2(2/7) - (3/7) (log2(3/7)

= 1.556

**I(S, Strategic) = (9/16)(** 1.584**) + (7/16)(** 1.556**)**

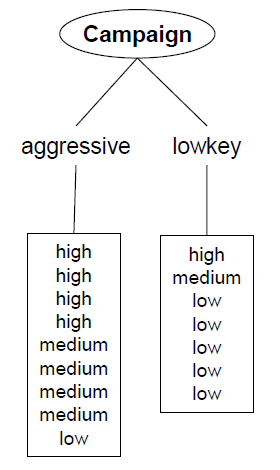
**= 1.572**

**E (S) = -(5/16)(log2(5/16) - (5/16)(log2(5/16) - (6/16)(log2(6/16)**

**= 1.578**

**Gain(S, A) = E (S) - I(S, A)**

**= 0.006**



**Campaign = aggressive: 4-high, 4-medium, 1-low**

E (Campaign = aggressive) = -(4/9) (log2(4/9) - (4/9) (log2(4/9) - (1/9) (log2(1/9)

= 1.391

**Campaign = lowkey: 1-high, 1-medium, 5-low**

E (Campaign = aggressive) = -(1/7) (log2(1/7) - (1/7) (log2(1/7) - (5/7) (log2(5/7)

= 1.148

**I(S, Campaign) = (9/16)(** 1.391**) + (7/16)(** 1.148**)**

**= 1.284**

**E (S) = -(5/16)(log2(5/16) - (5/16)(log2(5/16) - (6/16)(log2(6/16)**

**= 1.578**

**Gain(S, A) = E (S) - I(S, A)**

**= 0.294**

**Conclusion:**

For Decision tree classification, the implemented model predicts the survival of the passenger in a titanic data set. Data set is processed using Rpart library and the prediction is derived using the same from the processed data

For Decision tree regression, we have processed energy efficiency data set to create decision nodes and leaf nodes where dependent variable is continuous and splitting is based on standard deviation. Rpart library is used to depict the same

For pruning, entropy, and information gain in decision tree, I Computed entropy manually and came to conclusion that Campaign has the highest information gain and it should be considered as root node.