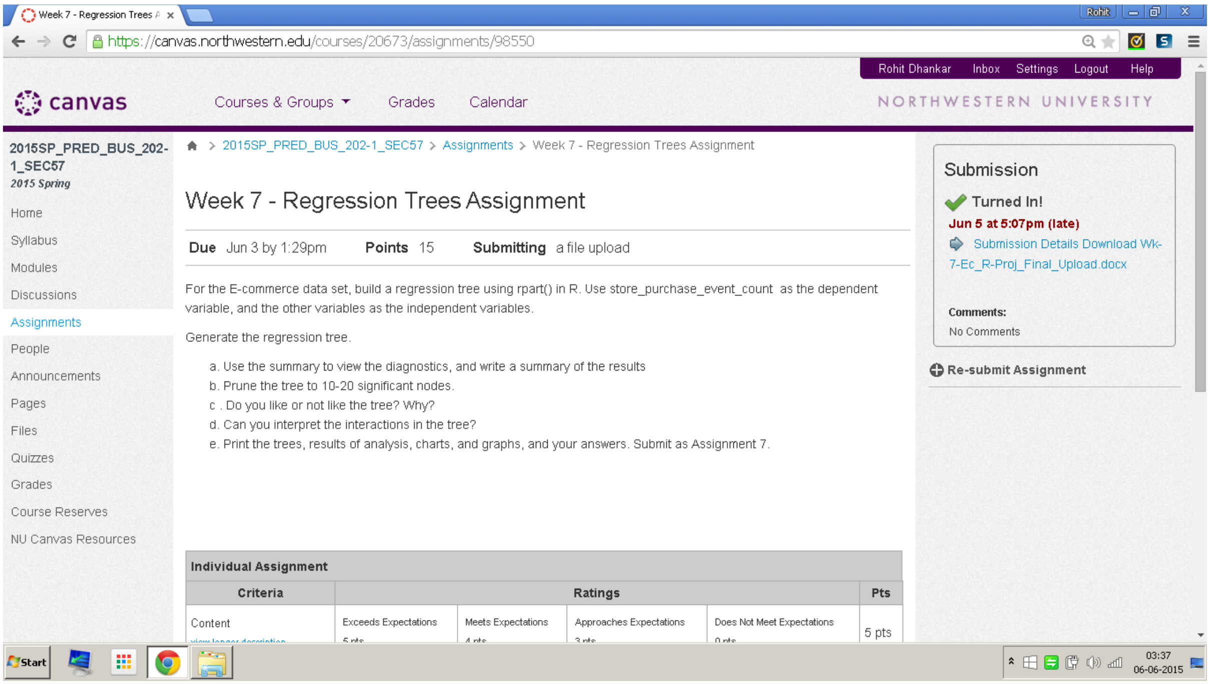
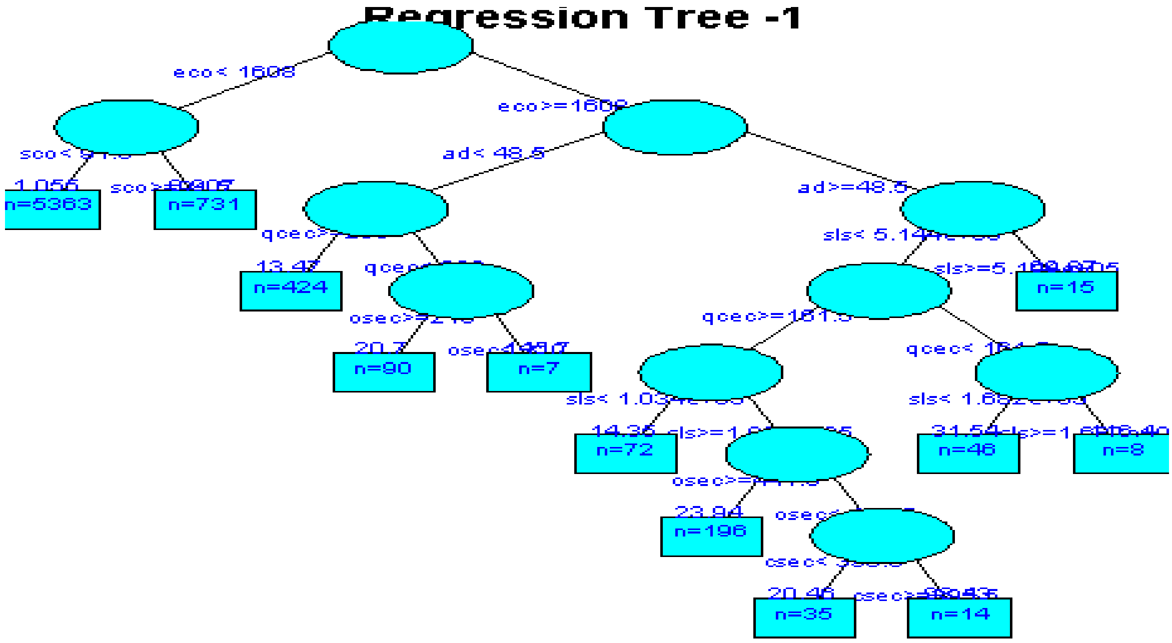


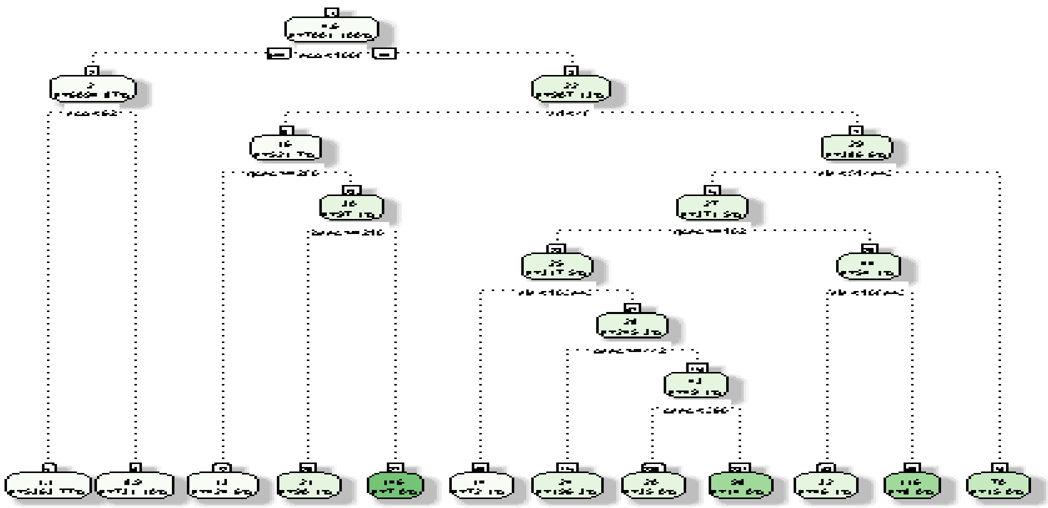
Wk-7-Ec_R-Proj_.R

Rohit
Sat Jun 06 01:46:51 2015

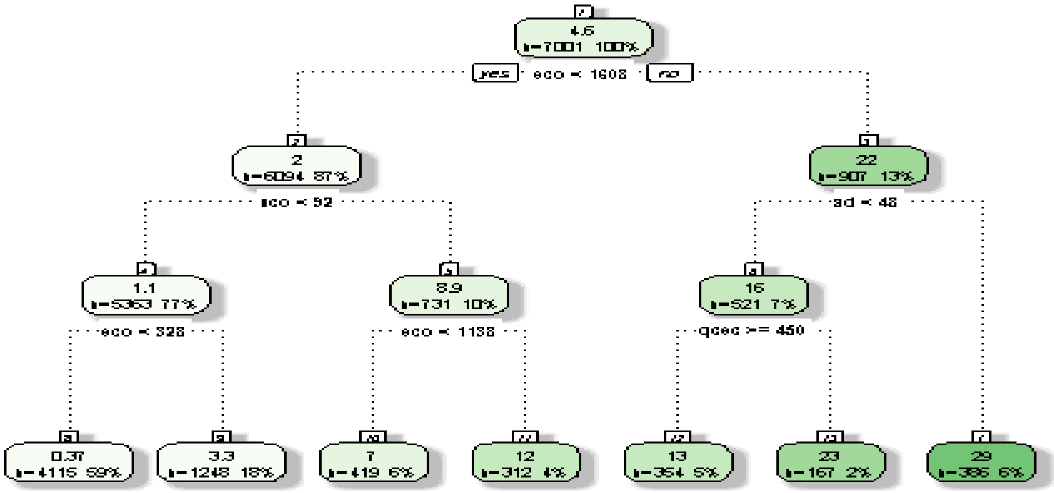
```
# Wk-7 EC - R- Project
ec <- read.csv("C:/STAT/Ec_R-Proj/ec.csv")
attach(ec)
library(caret)

library(rpart)
library(rattle)

inTrain <- createDataPartition(y=ec$spec,p=0.7, list=FALSE)
trn <- ec[inTrain,]
tst <- ec[-inTrain,]
dim(trn); dim(tst)

## [1] 7001    9

## [1] 2999    9

m<-rpart(spec~.,data=trn,method="anova")
par(mai=c(0.1,0.1,0.1,0.1))
plot(m,main="Regression Tree
-1",col=3,compress=TRUE,branch=0.2,uniform="TRUE")
text(m,cex=0.6,col=4,use.n=TRUE,fancy=TRUE,fwidth=0.4,fheight=0.4,bg=c(5)
)
```



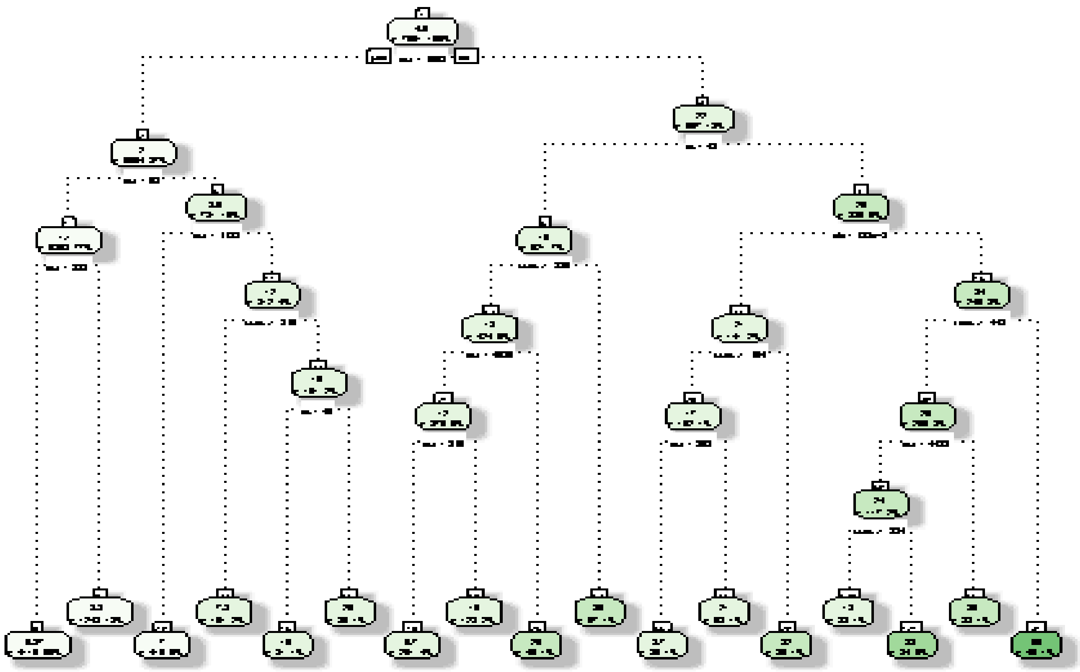

```r
fancyRpartPlot(m)
```



```r
m_minsplit500cp0.001<-rpart(spec~.,data=trn,
method="anova",control=rpart.control(minsplit=500, cp=0.001))
library(rattle)
fancyRpartPlot(m_minsplit500cp0.001)
```




```
m_minsplit100cp0.001<-rpart(spec~.,data=trn,
method="anova",control=rpart.control(minsplit=100, cp=0.001))
fancyRpartPlot(m_minsplit100cp0.001)
```

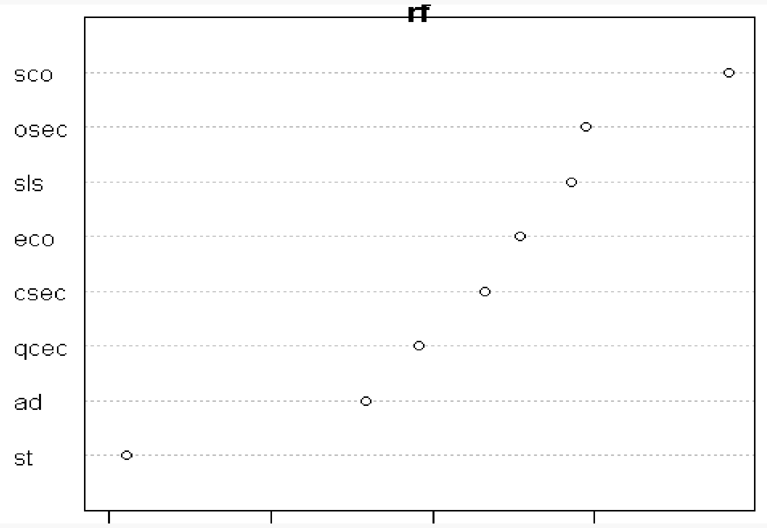


```
## INTERACTIONS ##
## As seen from plot Interaction is as defined below --
# Root Node - Split of variable eco < 1610 here n=7001.
# Child Node -9 , where n=1075 -- sco < 88,eco < 368 and qcec >= 22 is the
result of Interaction between - sco , eco and qcec.

# Predicting from data Trainig ..
```

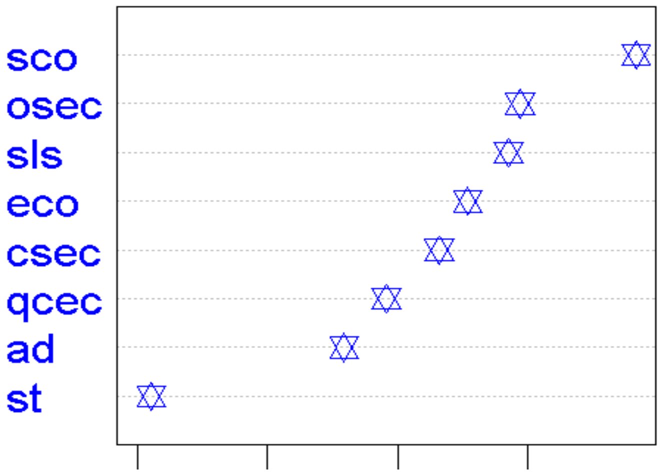
```r
TrnPred_m<-predict(m, newdata=trn)
head(TrnPred_m)

##         1         2         3         4         9        10
##  1.055379  1.055379  8.906977 20.457143 13.466981  1.055379

x<-as.matrix(TrnPred_m)
# As its a Regression problem - we need to Minimize Sum of Squared Errors
#
# Bagging - Bagging will have bias similar to the individual models
# but a reduced variance as we are averaging over individual models.

library(caret)
mBAG<-train(spec~.,method="treebag",data =trn)

print(mBAG$finalModel)

##
## Bagging regression trees with 25 bootstrap replications

## Random Forests

library(randomForest)

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

rf <- randomForest(spec~., data=trn, mtry=6, importance = TRUE,ntree=10)
yhat_tst <- predict(rf, tst)
yhat_trn <- predict(rf, trn)
mean((yhat_tst - tst$spec)^2)

## [1] 308.2885

mean((yhat_trn - trn$spec)^2)

## [1] 58.06644

rf

##
## Call:
##  randomForest(formula = spec ~ ., data = trn, mtry = 6, importance = TRUE,
ntree = 10)
##                Type of random forest: regression
##                      Number of trees: 10
## No. of variables tried at each split: 6
##
##           Mean of squared residuals: 290.6614
##                     % Var explained: -3.2

#Relative Importance
importance(rf)

##      IncNodePurity
## st       10660.97
## sls     285456.29
## sco      383617.35
```

```
## eco      253963.51
## csec     232026.07
## osec     294298.86
## qcec     191254.35
## ad       158977.83
```

```
varImpPlot(rf)
```



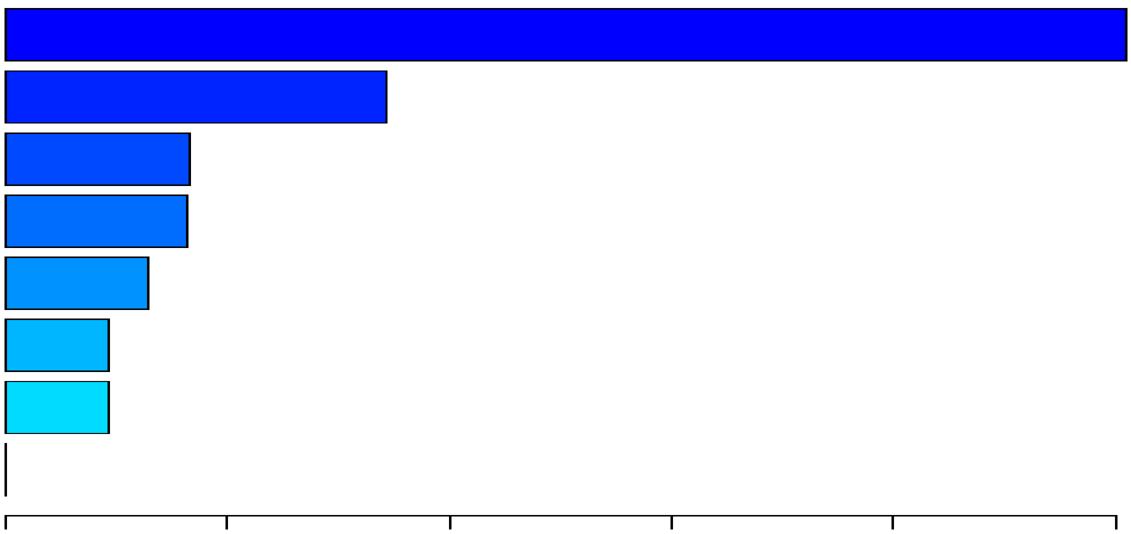

```
varImpPlot(rf, type=2, pch=11, col=4, cex=2, main="")
```




```
## Boosting ----
library(gbm)
```

```
## Loading required package: survival
##
## Attaching package: 'survival'
##
## The following object is masked from 'package:caret':
##
##     cluster
##
## Loading required package: splines
## Loading required package: parallel
## Loaded gbm 2.1.1
```

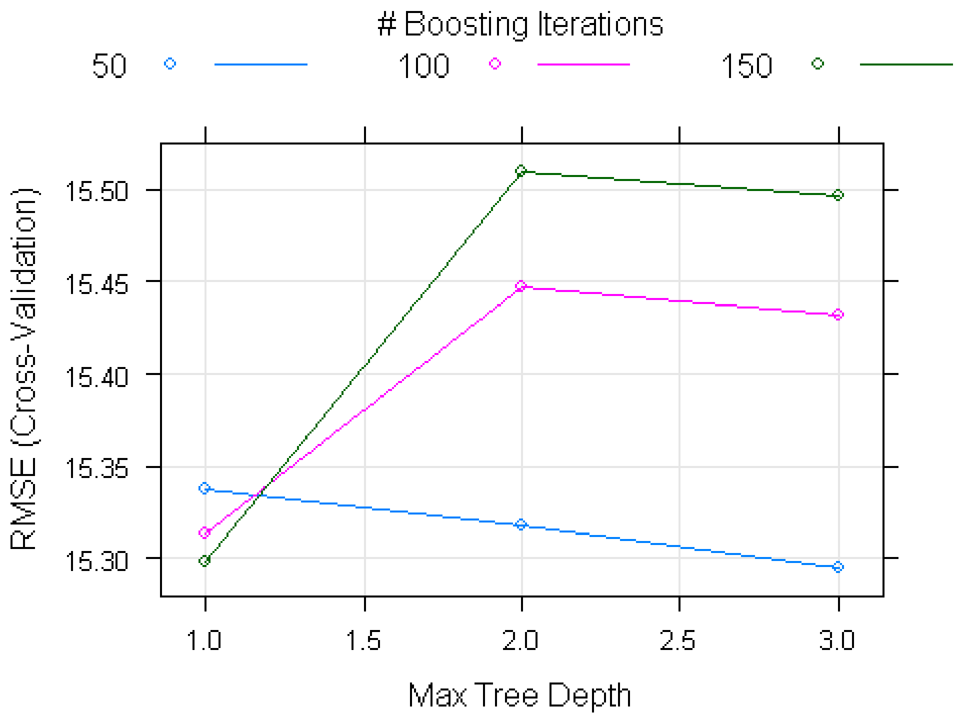
```
boost <- gbm(spec~. , data=trn,distribution = 'gaussian', n.trees = 10,
interaction.depth = 4)
summary(boost)
```



```
##       var   rel.inf
## eco   eco 50.436622
## sco   sco 17.220223
## csec csec  8.287838
## qcec qcec  8.249882
## osec osec  6.445820
## ad     ad  4.699561
## sls   sls  4.660054
## st     st  0.000000
```

```
# as seen from Summary - eco   eco 60.626854 and sco   sco 10.632023 have
high Relative Influence .
# We plot them separately ...
par(mfrow=c(1,2))
plot(boost, i='eco')
plot(boost, i='sco')
```

```
boost.pred <- predict (boost, tst, n.trees=10)
mean((boost.pred - tst$spec)^2)

## [1] 276.3682

ctr <- trainControl(method = "cv", number = 10)
boost.caret <- train(spec~.,
trn,method='bstTree',preProc=c('center','scale'),trControl=ctr)

## Loading required package: bst

# yes
boost.caret

## Boosted Tree
##
## 7001 samples
##    8 predictor
##
## Pre-processing: centered, scaled
## Resampling: Cross-Validated (10 fold)
##
## Summary of sample sizes: 6300, 6301, 6302, 6302, 6301, 6300, ...
##
## Resampling results across tuning parameters:
##
##   maxdepth  mstop  RMSE      Rsquared   RMSE SD   Rsquared SD
##   1          50    15.33740  0.2352452  4.248241  0.06221489
##   1         100    15.31355  0.2390788  4.226653  0.06484887
##   1         150    15.29802  0.2407234  4.229935  0.06566608
##   2          50    15.31806  0.2394373  4.169020  0.06417995
##   2         100    15.44771  0.2310972  4.093270  0.05860468
##   2         150    15.50997  0.2288551  4.050762  0.05757364
##   3          50    15.29485  0.2397998  4.179463  0.04864371
##   3         100    15.43241  0.2332206  4.045777  0.05172717
##   3         150    15.49668  0.2321351  4.017721  0.06180625
##
## Tuning parameter 'nu' was held constant at a value of 0.1
## RMSE was used to select the optimal model using  the smallest value.
## The final values used for the model were mstop = 50, maxdepth = 3 and nu
##  = 0.1.

plot(boost.caret)
```




```
## Here, with some better tuned parameters, we compare prediction - accuracy with random
forests.??
boost.caret.pred <- predict(boost.caret, tst)
mean((boost.caret.pred - tst$spec)^2)

## [1] 214.432

## RF - > mean((boost.pred - tst$spec)^2) == [1] 373.8108
## boost.caret.pred - mean((boost.caret.pred - tst$spec)^2) == [1] 311.0449

## boost.caret.pred  is better as - mean == 311.0449 is lower .
```