Decision Trees and Random Forests

Cameron Chong

2019-03-27

```
library(tree)
library(MASS)
library(randomForest)
```

```
## randomForest 4.6-14
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
admissionsData <- read.csv("Admission_Predict_Ver1.1.csv", header = TRUE)
admissionsData <- admissionsData[,-1]
#head(admissionsData)
dim(admissionsData)
```

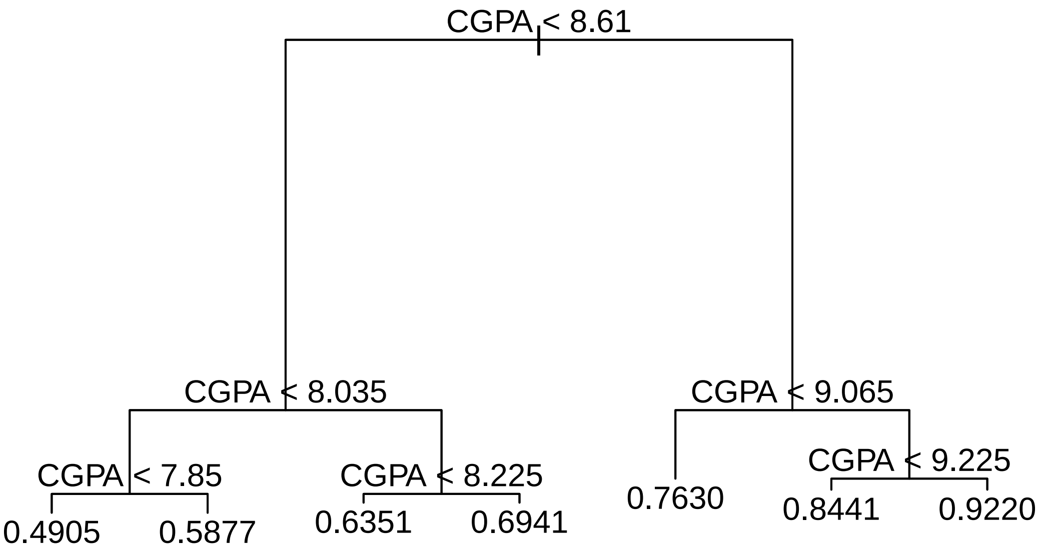
```
## [1] 500   8
```

```
trainindex <- sample(1:nrow(admissionsData), 350)
admissionsTrain <- admissionsData[trainindex, ]
admissionsTest <- admissionsData[-trainindex, ]
```

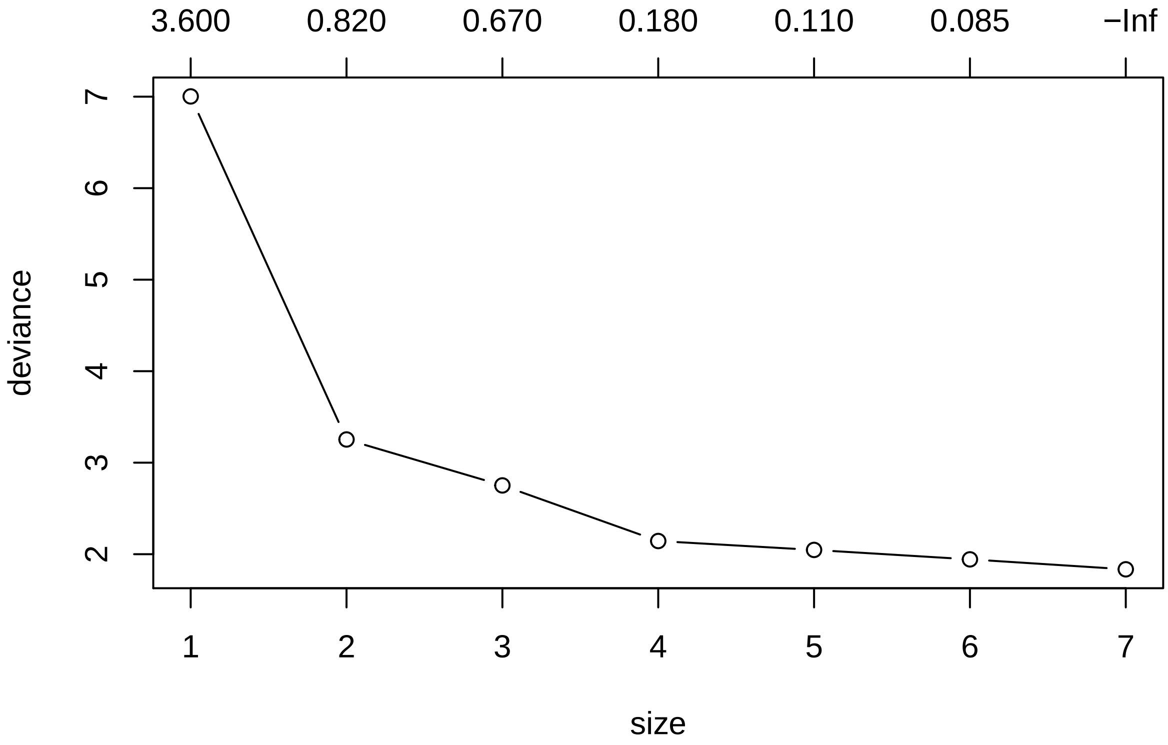
Chance of Admittance

I am going to do a 70/30 split of training and testing data. There are 500 observations, so we will have 350 training observations and 150 testing points.

```
set.seed(110101010)
admissionTree <- tree(Chance.of.Admit~., data = admissionsTrain)
plot(admissionTree)
text(admissionTree, pretty=0)
```



```
admissionTreeCV <- cv.tree(admissionTree, FUN = prune.tree, K = 10)
plot(admissionTreeCV, type = "b")
```



```
admissionTreeCV
```

```
## $size
## [1] 7 6 5 4 3 2 1
##
## $dev
## [1] 1.834740 1.943470 2.046524 2.143788 2.751490 3.253592 7.002862
##
## $k
## [1]       -Inf 0.08540037 0.10771131 0.18420149 0.66899699 0.81811334
## [7] 3.61885317
##
## $method
## [1] "deviance"
##
## attr(,"class")
## [1] "prune"         "tree.sequence"
```

```
admissionTreeCV$dev
```

```
## [1] 1.834740 1.943470 2.046524 2.143788 2.751490 3.253592 7.002862
```

```
admissionTreeCV$size
```

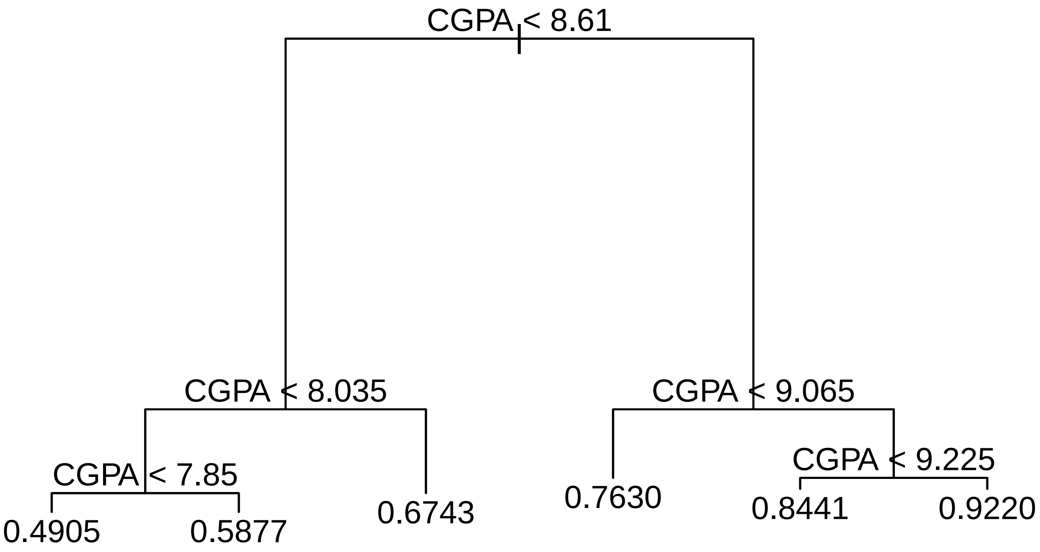
```
## [1] 7 6 5 4 3 2 1
```

```
which.min(admissionTreeCV$dev)
```

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

Cross validation suggest 6 nodes would be best, so we will prune the tree using 6 terminal nodes.

```r
pruneAdmissionTreeCV <- prune.tree(admissionTree, best=6)
plot(pruneAdmissionTreeCV)
text(pruneAdmissionTreeCV, pretty = 0)
```



```r
summary(pruneAdmissionTreeCV)
```

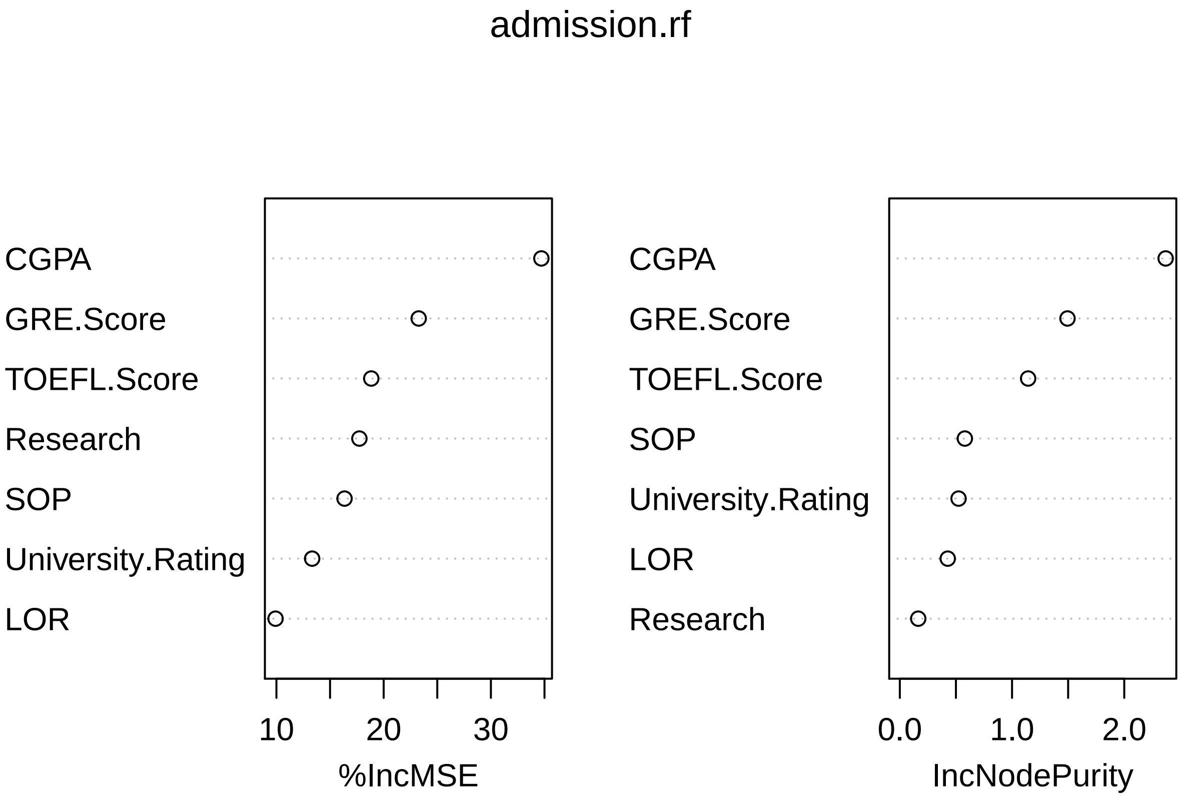
```
##
## Regression tree:
## snip.tree(tree = admissionTree, nodes = 5L)
## Variables actually used in tree construction:
## [1] "CGPA"
## Number of terminal nodes:  6
## Residual mean deviance:  0.004598 = 1.582 / 344
## Distribution of residuals:
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -0.247700 -0.032990  0.006437  0.000000  0.045730  0.209500
```

```r
set.seed(1000101010)
admission.rf <- randomForest(Chance.of.Admit~., data = admissionsTrain, importance = TRUE)
admission.rf
```

```
##
## Call:
##  randomForest(formula = Chance.of.Admit ~ ., data = admissionsTrain,      importance = TRUE)
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 2
##
##           Mean of squared residuals: 0.00415925
##                     % Var explained: 79.14
```

Since Random Forest uses out-of-bag which is similar to cross validation so no cross validation was performed. We can look at the importance of the variables.

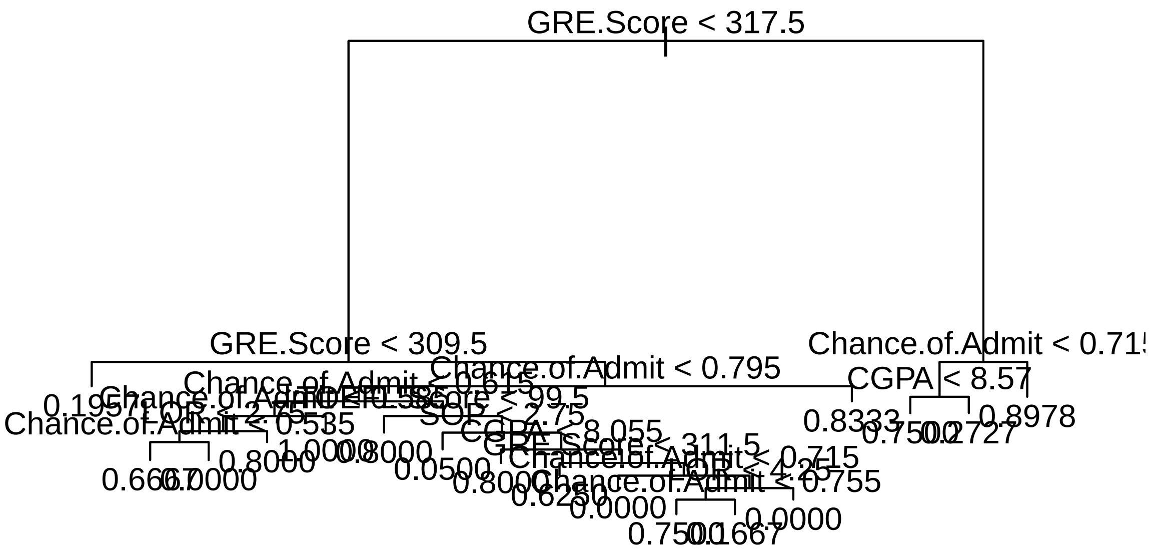
```
varImpPlot(admission.rf)
```



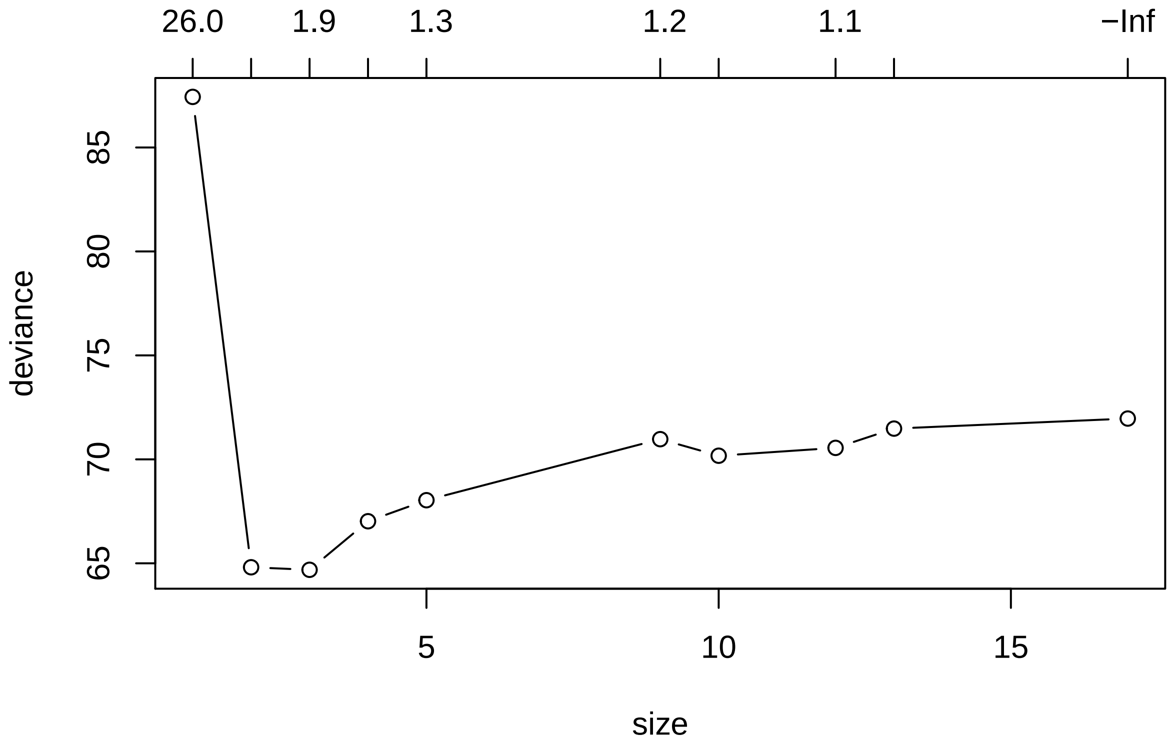
As seen from the Importance Plot the most important variables are CGPA, GRE Score and TOEFL scores when using chance of admission as a response variable.

Research

```
set.seed(1388582293)
researchTree <- tree(Research~., data = admissionsTrain)
plot(researchTree)
text(researchTree, pretty=0)
```



```r
researchTreeCV <- cv.tree(researchTree, FUN = prune.tree, K = 10)
plot(researchTreeCV, type = "b")
```



```r
which.min(researchTreeCV$dev)
```

```
## [1] 8
```

```r
researchTreeCV$dev
```

```
##  [1] 71.96252 71.47878 70.55073 70.17400 70.97002 68.03508 67.02237
##  [8] 64.68752 64.80862 87.43248
```

```r
researchTreeCV$dev
```

```
##  [1] 71.96252 71.47878 70.55073 70.17400 70.97002 68.03508 67.02237
##  [8] 64.68752 64.80862 87.43248
```

```
researchTreeCV$size
```

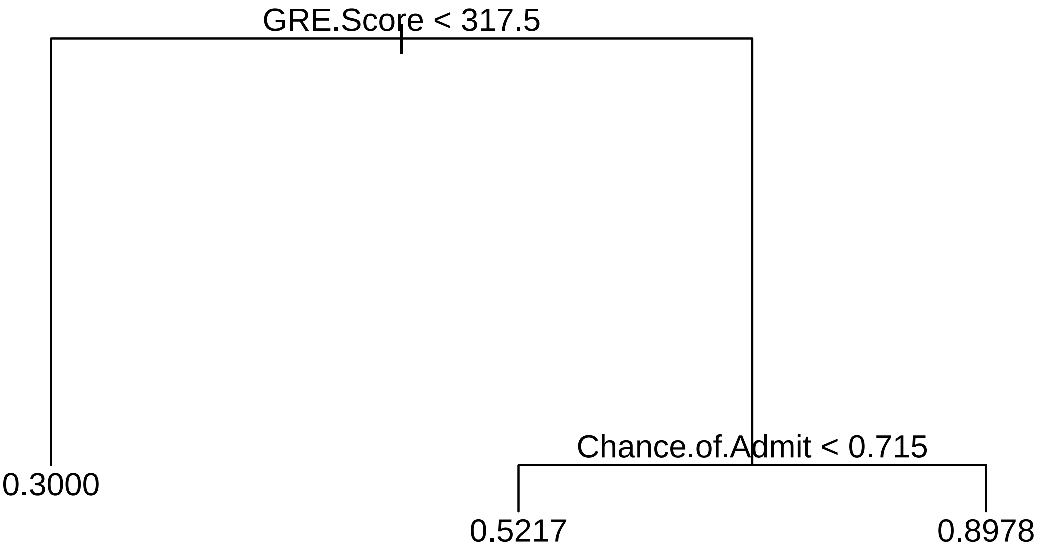
```
##  [1] 17 13 12 10  9  5  4  3  2  1
```

```
which.min(researchTreeCV$dev)
```

```
## [1] 8
```

Cross Validation Suggests 3 terminal nodes would be best.

```
pruneResearchTreeCV <- prune.tree(researchTree, best=3)
plot(pruneResearchTreeCV)
text(pruneResearchTreeCV, pretty = 0)
```



```
summary(pruneResearchTreeCV)
```

```
##
## Regression tree:
## snip.tree(tree = researchTree, nodes = c(6L, 2L))
## Variables actually used in tree construction:
## [1] "GRE.Score"       "Chance.of.Admit"
## Number of terminal nodes:  3
## Residual mean deviance:  0.1677 = 58.21 / 347
## Distribution of residuals:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -0.8978 -0.3000  0.1022  0.0000  0.1022  0.7000
```

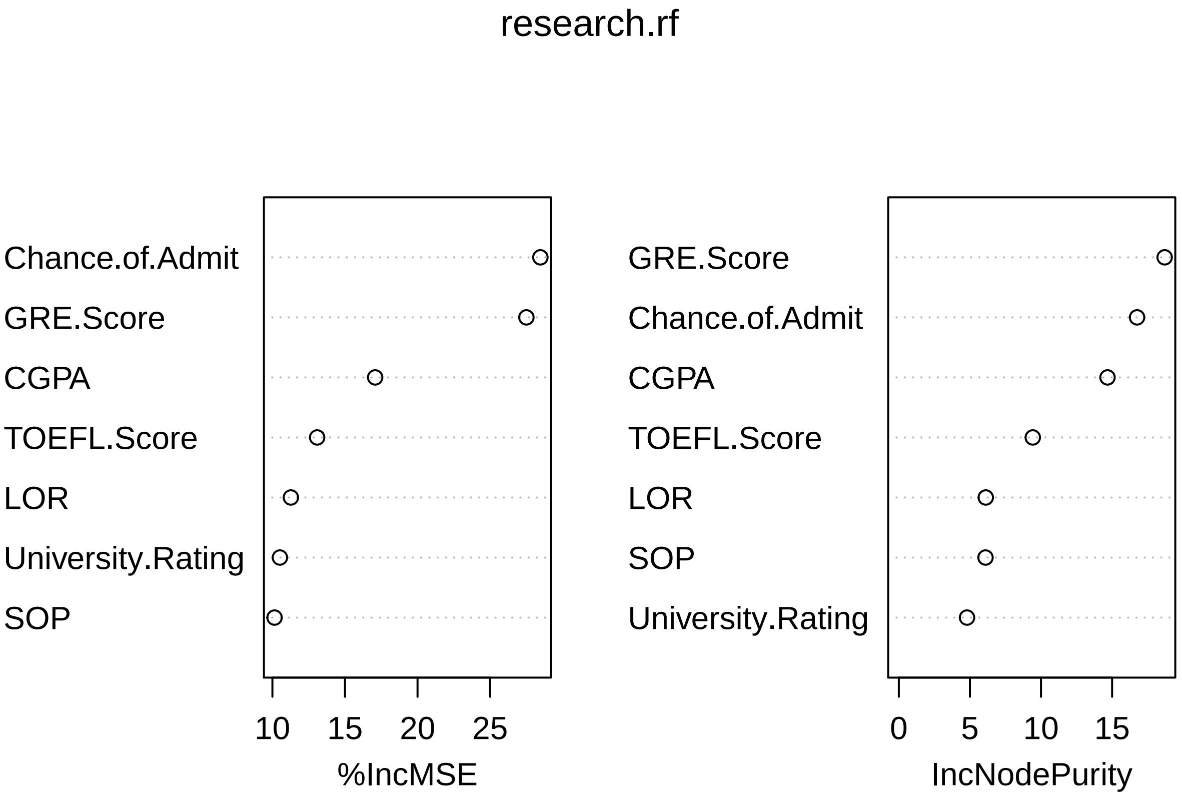
```
set.seed(1413755523)
research.rf <- randomForest(Research~., data = admissionsTrain, importance = TRUE)
```

```
## Warning in randomForest.default(m, y, ...): The response has five or fewer
## unique values. Are you sure you want to do regression?
```

```
research.rf
```

```
##
## Call:
##  randomForest(formula = Research ~ ., data = admissionsTrain,      importance = TRUE)
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 2
##
##           Mean of squared residuals: 0.168409
##                     % Var explained: 31.99
```

```
varImpPlot(research.rf)
```


research.rf
Chance.of.Admit
GRE.Score
CGPA
TOEFL.Score
LOR
University.Rating
SOP
%IncMSE
GRE.Score
Chance.of.Admit
CGPA
TOEFL.Score
LOR
SOP
University.Rating
IncNodePurity