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)
```

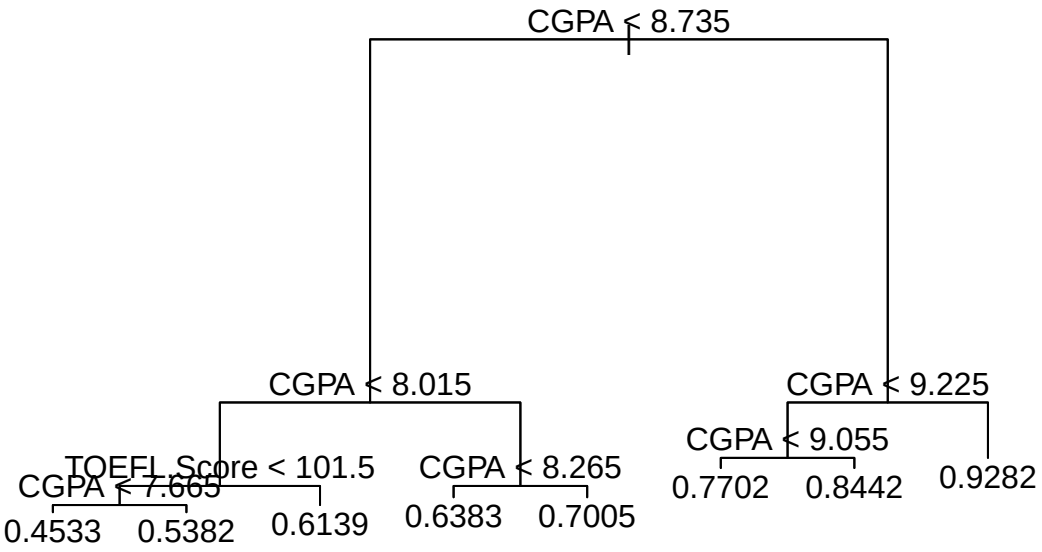
```
## [1] 500   8
```

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

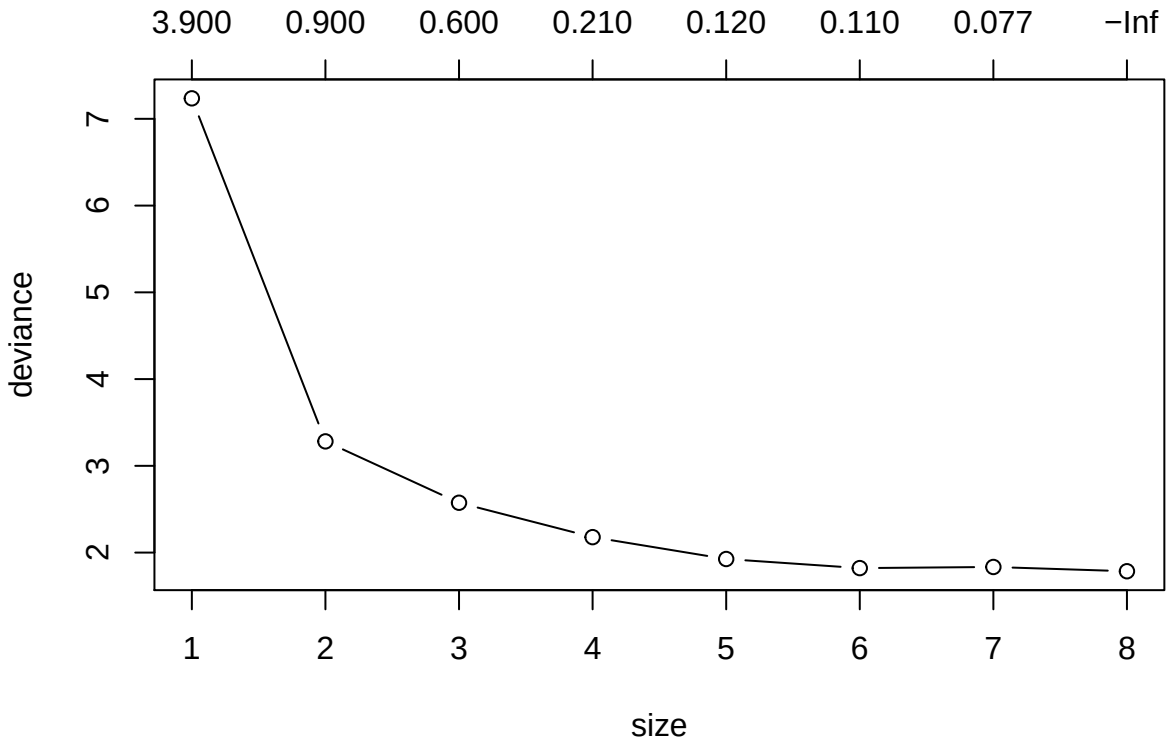
Chance of Admittance

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. Serial Number was removed as it is meta data

```
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] 8 7 6 5 4 3 2 1
##
## $dev
## [1] 1.785065 1.834201 1.821147 1.926029 2.178254 2.574573 3.282046 7.236107
##
## $k
## [1]       -Inf 0.07735025 0.10931246 0.12207408 0.20572011 0.59725923
## [7] 0.89984470 3.91214202
##
## $method
## [1] "deviance"
##
## attr(,"class")
## [1] "prune"         "tree.sequence"
```

```
admissionTreeCV$dev
```

```
## [1] 1.785065 1.834201 1.821147 1.926029 2.178254 2.574573 3.282046 7.236107
```

```
admissionTreeCV$size
```

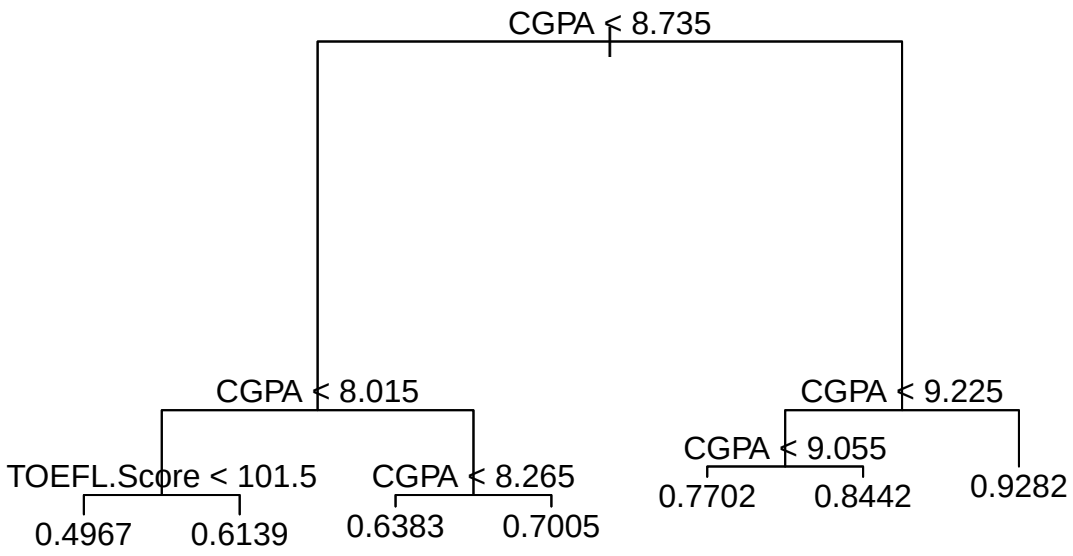
```
## [1] 8 7 6 5 4 3 2 1
```

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

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

Cross validation suggest 7 nodes would be best, so we will prune the tree using 7 terminal nodes.

```
pruneAdmissionTreeCV <- prune.tree(admissionTree, best=7)
plot(pruneAdmissionTreeCV)
text(pruneAdmissionTreeCV, pretty = 0)
```




```
summary(pruneAdmissionTreeCV)
```

```
##
## Regression tree:
## snip.tree(tree = admissionTree, nodes = 8L)
## Variables actually used in tree construction:
## [1] "CGPA"        "TOEFL.Score"
## Number of terminal nodes:  7
## Residual mean deviance:  0.003917 = 1.344 / 343
## Distribution of residuals:
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -0.223900 -0.028310  0.006087  0.000000  0.031750  0.183300
```

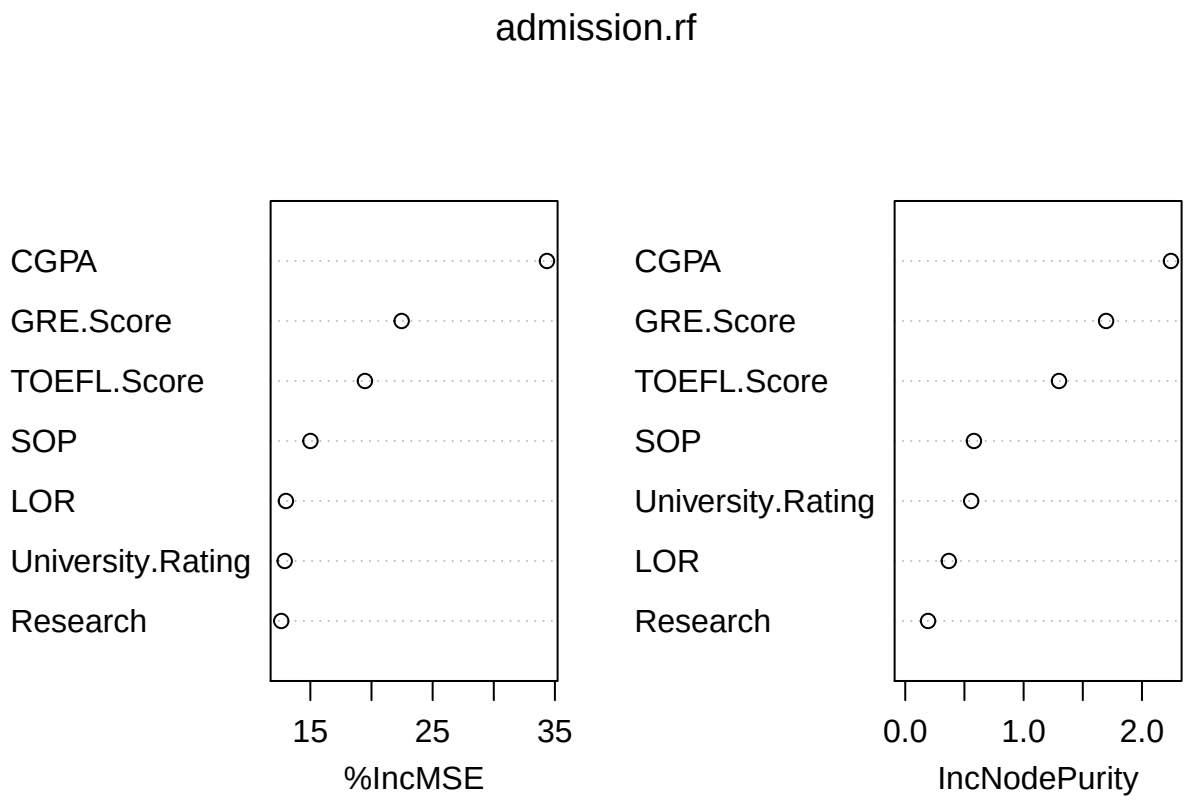
```
set.seed(1000101010)
admission.rf <- randomForest(Chance.of.Admit~., data = admissionsTrain, importance = TRUE)
admission.rf
```

```
##
## 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: 0.00368334
##                     % Var explained: 82.07
```

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.

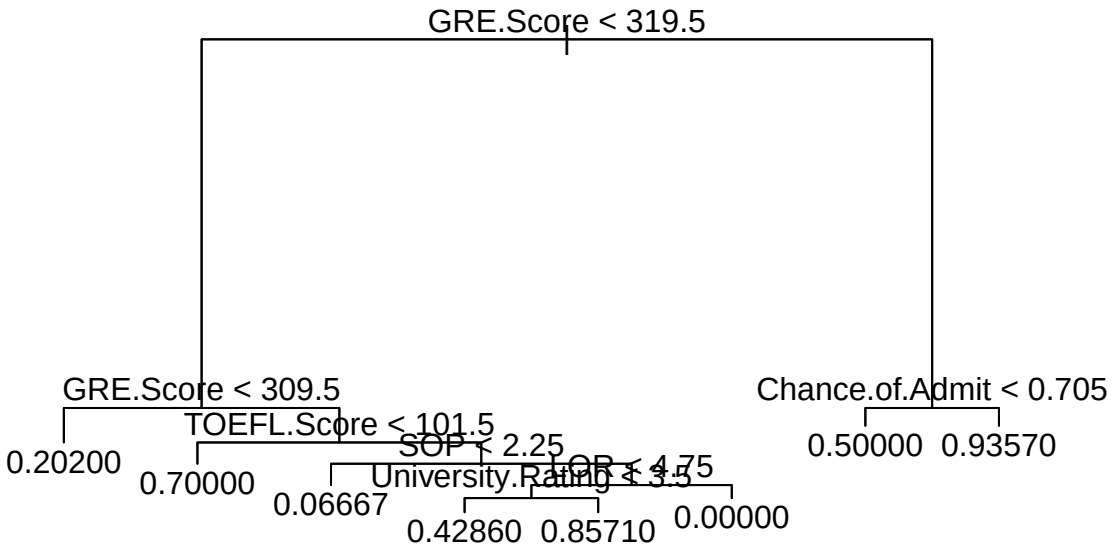
```
varImpPlot(admission.rf)
```



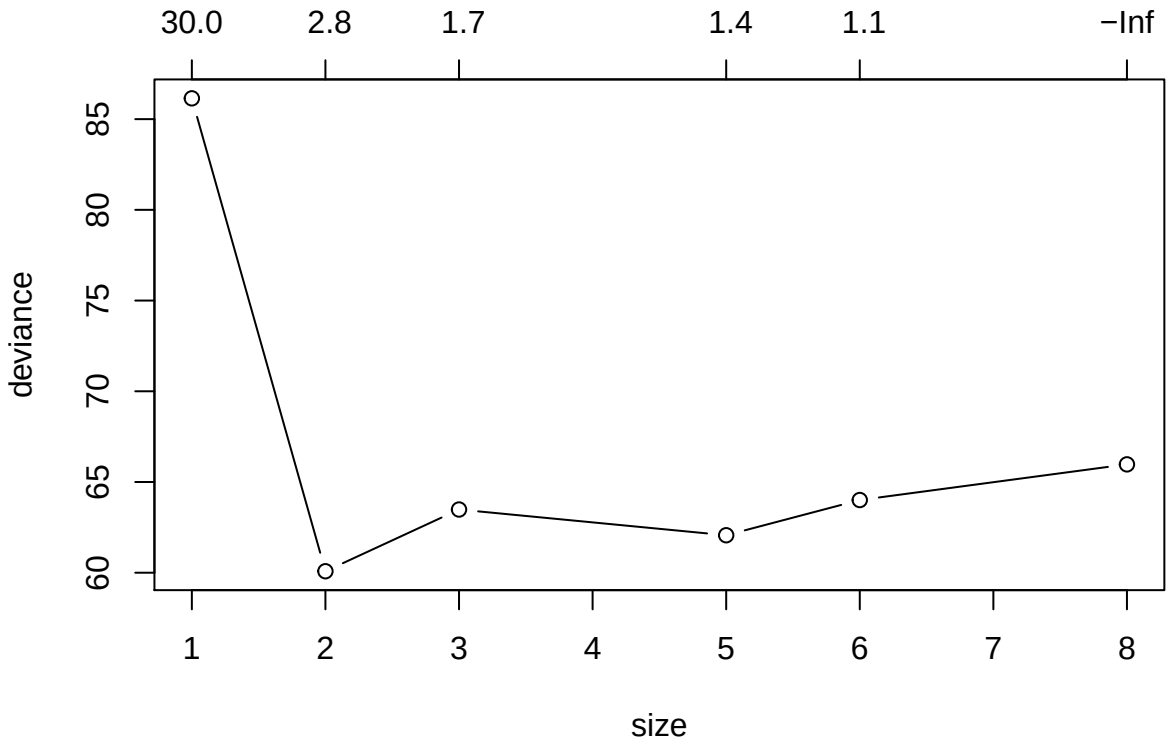
As seen from the Importance Plot the most important variables are CGPA, GRE Score and TOEFL scores when using chance of admission as a response variable.

Research

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




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




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

```
## [1] 5
```

```
researchTreeCV$dev
```

```
## [1] 65.97168 64.00638 62.06575 63.48388 60.08173 86.14524
```

```
researchTreeCV$dev
```

```
## [1] 65.97168 64.00638 62.06575 63.48388 60.08173 86.14524
```

```
researchTreeCV$size
```

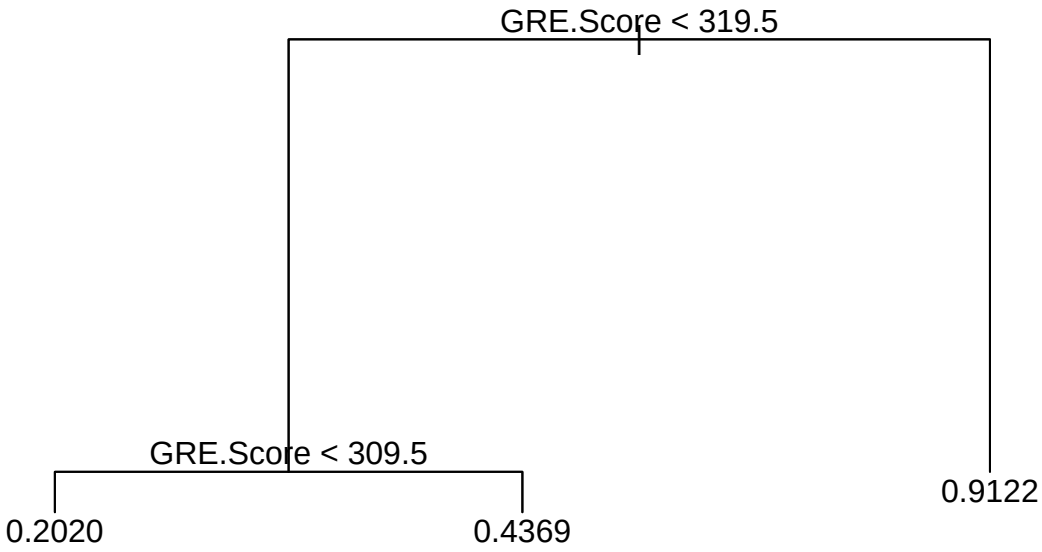
```
## [1] 8 6 5 3 2 1
```

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

```
## [1] 5
```

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(3L, 5L))
## Variables actually used in tree construction:
## [1] "GRE.Score"
## Number of terminal nodes:  3
## Residual mean deviance:  0.1532 = 53.16 / 347
## Distribution of residuals:
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -0.91220 -0.20200  0.08784  0.00000  0.08784  0.79800
```

```
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: 0.1607849
##                     % Var explained: 34.35
```

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