Regression Trees

Dr. Saad

Introduction: Tree Based Algorithms or **CARTs**

- **Tree-Based models or algorithms** can be applied to both **regression** and **classification** projects.

- **Tree-Based models** refer to many algorithms such, **Regression Trees**, **Decision Trees**, and **ensemble models** such as **Random Forest**, **Bagging**, and **Gradient Boosting Machines (or GBMs)**.

- These algorithms involve **stratifying or segmenting** the predictor space into a number of regions.

- Tree-Based models: are simple to understand, easy-to-use, easy to interpret, and, when used in ensembles, the have excellent accuracy.

- They are used to make decisions, explore the data and make predictions. (Even by non-data scientists like managers)

- They do not need pre-processing, which make them a good start for beginners. And they are **Flexible** or **non-linear** models.

Terminology

- A Decision tree is a **hierarchical structure** with nodes and directed edges.
- **Node**: A question or prediction
- **Root node**: The top node at the top with no parent, it involves a question that gives two answers (**children**).
- **leaf nodes or terminal nodes** The nodes at the bottom. The have one **parent** and no **children**. (**The leaves are used for prediction**)
- **Internal nodes** the nodes that are neither the root node or the leaf nodes. The have one **parent** and give two **children**.
- **Branch**: The segments connecting the parent with the children nodes.
- **Maximum Depth**: the distance between the root node (depth $= 0$) and the final leaf.

Diagram Showing the Terminology 01

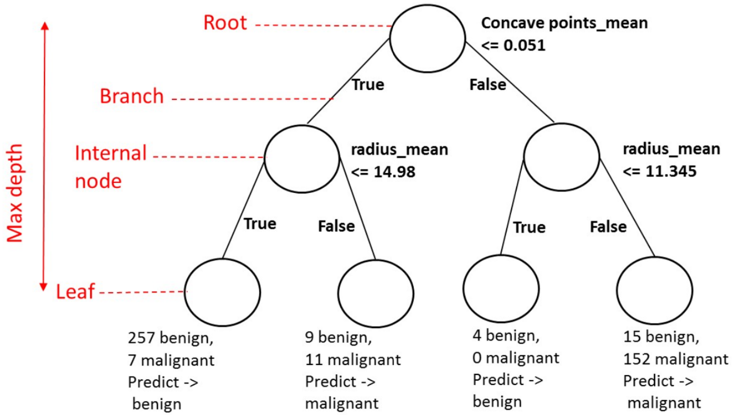
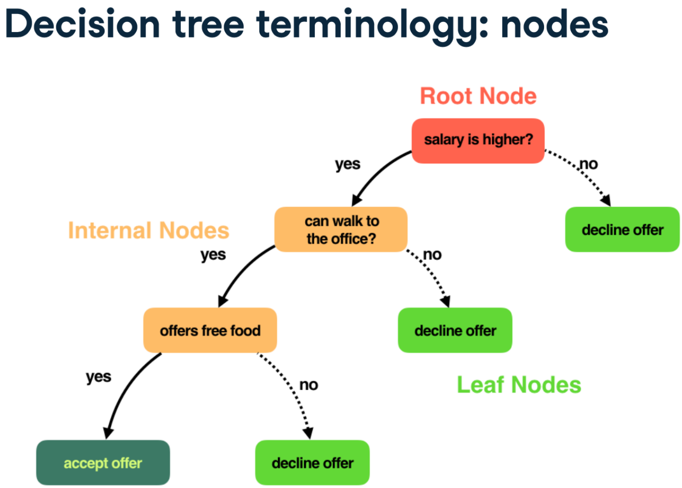


Diagram Showing the Terminology 02



How Regression Tree Works

Baseball data, with three variables, **years**, **Hits**, to predict the **Salary**.

Fitting Regression Tree (Motivation Example)

```r
library(ISLR)
library(tree)
data("Hitters")
str(Hitters)
```

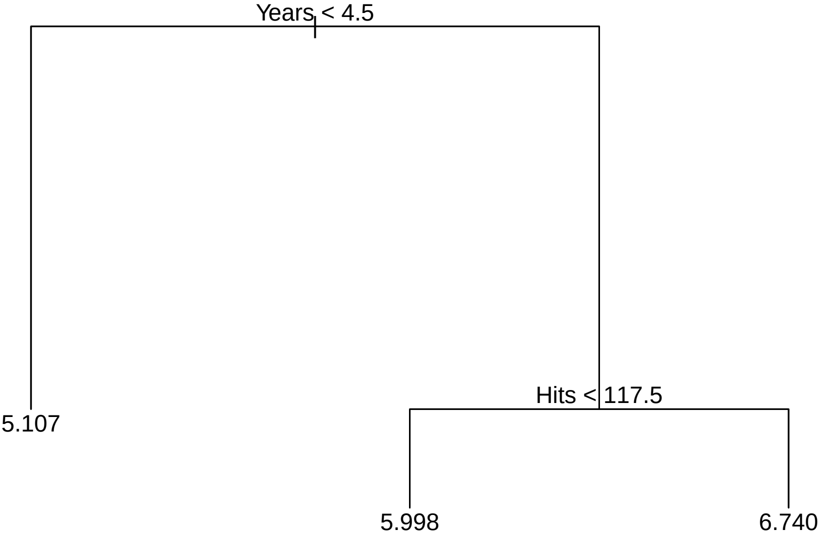
```
## 'data.frame':    322 obs. of  20 variables:
##  $ AtBat    : int  293 315 479 496 321 594 185 298 323 401 ...
##  $ Hits     : int  66 81 130 141 87 169 37 73 81 92 ...
##  $ HmRun    : int  1 7 18 20 10 4 1 0 6 17 ...
##  $ Runs     : int  30 24 66 65 39 74 23 24 26 49 ...
##  $ RBI      : int  29 38 72 78 42 51 8 24 32 66 ...
##  $ Walks    : int  14 39 76 37 30 35 21 7 8 65 ...
##  $ Years    : int  1 14 3 11 2 11 2 3 2 13 ...
##  $ CAtBat   : int  293 3449 1624 5628 396 4408 214 509 341 5206 ...
##  $ CHits    : int  66 835 457 1575 101 1133 42 108 86 1332 ...
##  $ CHmRun   : int  1 69 63 225 12 19 1 0 6 253 ...
##  $ CRuns    : int  30 321 224 828 48 501 30 41 32 784 ...
##  $ CRBI     : int  29 414 266 838 46 336 9 37 34 890 ...
##  $ CWalks   : int  14 375 263 354 33 194 24 12 8 866 ...
##  $ League   : Factor w/ 2 levels "A","N": 1 2 1 2 2 1 2 1 2 1 ...
##  $ Division : Factor w/ 2 levels "E","W": 1 2 2 1 1 2 1 2 2 1 ...
##  $ PutOuts  : int  446 632 880 200 805 282 76 121 143 0 ...
##  $ Assists  : int  33 43 82 11 40 421 127 283 290 0 ...
##  $ Errors   : int  20 10 14 3 4 25 7 9 19 0 ...
##  $ Salary   : num  NA 475 480 500 91.5 750 70 100 75 1100 ...
##  $ NewLeague: Factor w/ 2 levels "A","N": 1 2 1 2 2 1 1 1 2 1 ...
```

```r
my_data <- dplyr::select(Hitters, Salary, Years, Hits)
my_data <- na.omit(my_data)
tree_model <- tree(log(Salary) ~ Years + Hits,
                   data = my_data,
                   control = tree.control(
                     nobs = nrow(my_data),
                     mincut = 50
                   ))
print(tree_model)
```

```
## node), split, n, deviance, yval
##       * denotes terminal node
##
## 1) root 263 207.20 5.927
##   2) Years < 4.5 90  42.35 5.107 *
##   3) Years > 4.5 173  72.71 6.354
##     6) Hits < 117.5 90  28.09 5.998 *
##     7) Hits > 117.5 83  20.88 6.740 *
```

```r
summary(tree_model)
```

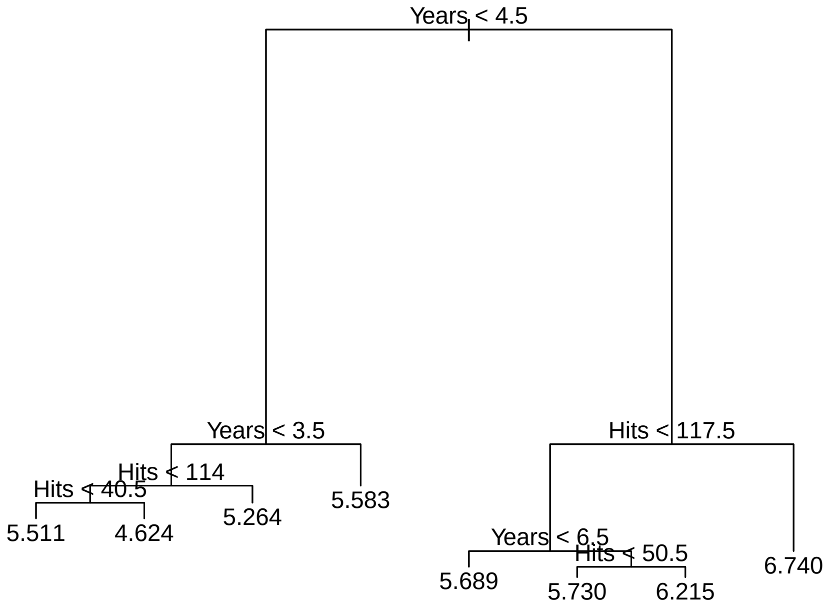
```
##
## Regression tree:
## tree(formula = log(Salary) ~ Years + Hits, data = my_data, control = tree.control(nobs = nrow(my_data),
##     mincut = 50))
## Number of terminal nodes:  3
## Residual mean deviance:  0.3513 = 91.33 / 260
## Distribution of residuals:
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -2.24000 -0.39580 -0.03162  0.00000  0.33380  2.55600
```


Years < 4.5
5.107
Hits < 117.5
5.998
6.740

Fitting Regression Tree (No control)

```r
tree_model2 <- tree(log(Salary) ~ Years + Hits,
                    data = my_data)
print(tree_model2)
```

```
## node), split, n, deviance, yval
##       * denotes terminal node
##
##  1) root 263 207.200 5.927
##    2) Years < 4.5 90  42.350 5.107
##      4) Years < 3.5 62  23.010 4.892
##        8) Hits < 114 43  17.150 4.727
##         16) Hits < 40.5 5  10.400 5.511 *
##         17) Hits > 40.5 38   3.280 4.624 *
##        9) Hits > 114 19   2.069 5.264 *
##      5) Years > 3.5 28  10.130 5.583 *
##    3) Years > 4.5 173  72.710 6.354
##      6) Hits < 117.5 90  28.090 5.998
##       12) Years < 6.5 26   7.238 5.689 *
##       13) Years > 6.5 64  17.350 6.124
##         26) Hits < 50.5 12   2.689 5.730 *
##         27) Hits > 50.5 52  12.370 6.215 *
##      7) Hits > 117.5 83  20.880 6.740 *
```


Years < 4.5
Years < 3.5
Hits < 117.5
Hits < 114
Hits < 40.5
5.511
4.624
5.264
5.583
Years < 6.5
Hits < 50.5
5.689
5.730
6.215
6.740