**Random Forest**

**Example- Company Data**

'data.frame': 400 obs. of 11 variables:

$ Sales : num 9.5 11.22 10.06 7.4 4.15 ...

$ CompPrice : int 138 111 113 117 141 124 115 136 132 132 ...

$ Income : int 73 48 35 100 64 113 105 81 110 113 ...

$ Advertising: int 11 16 10 4 3 13 0 15 0 0 ...

$ Population : int 276 260 269 466 340 501 45 425 108 131 ...

$ Price : int 120 83 80 97 128 72 108 120 124 124 ...

$ ShelveLoc : Factor w/ 3 levels "Bad","Good","Medium": 1 2 3 3 1 1 3 2 3 3 ...

$ Age : int 42 65 59 55 38 78 71 67 76 76 ...

$ Education : int 17 10 12 14 13 16 15 10 10 17 ...

$ Urban : Factor w/ 2 levels "No","Yes": 2 2 2 2 2 1 2 2 1 1 ...

$ US : Factor w/ 2 levels "No","Yes": 2 2 2 2 1 2 1 2 1 2 ...

**From above data frame, 3 variables are factor and rest all are numeric.**

**Out of this we need to find which records can be considered for high sales.**

**To classify, we consider Sales into 3 categories as Low, Medium and High and will give equal weightage to all categories based on cutoff values.**

**Data classification using variable Sales 🡺**

**So, we have 33.3 % records each with low, medium and high, so we consider top 33.3% of sales value as high and rest 66.6% as low.**

**Based on this sorting we are getting cutoff value as 8.67, so sales above than 8.5 will be considered as high sales value.**

**Now, based on cutoff value we will create another variable as “SalesC” with levels High and Low.**

**Now sales ratio of High:Low is 1:3 which is kind of imbalanced, so we required to balance it if we find any difficulty in classification.**

**Model-1 🡺**

**Confusion Matrix**

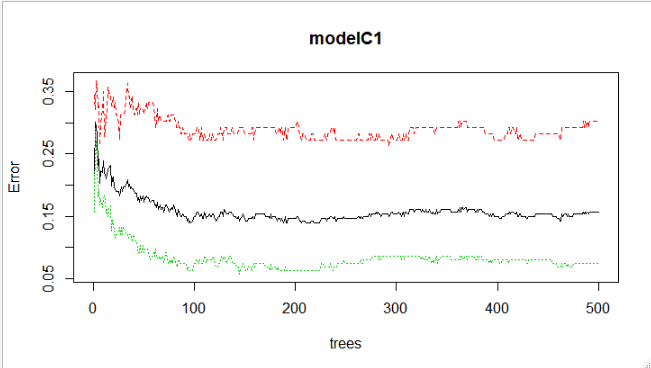
Predicted

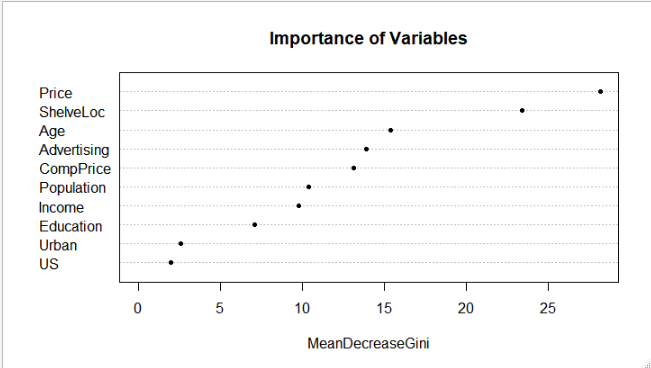
Actual High Low

High 24 12

Low 8 73

**Accuracy 🡺0.8292**





**Model-2 with Normalized Value 🡺**

**Confusion Matrix**

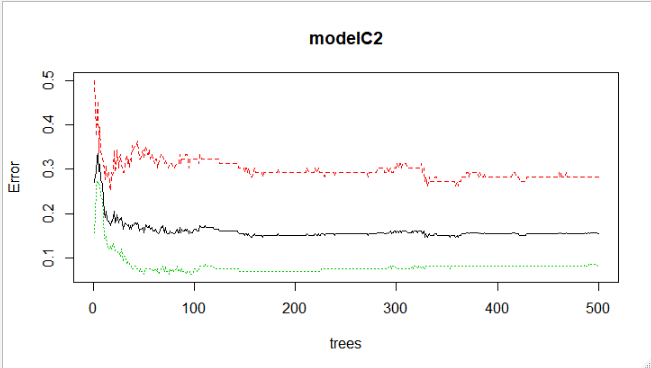
Predicted

Actual High Low

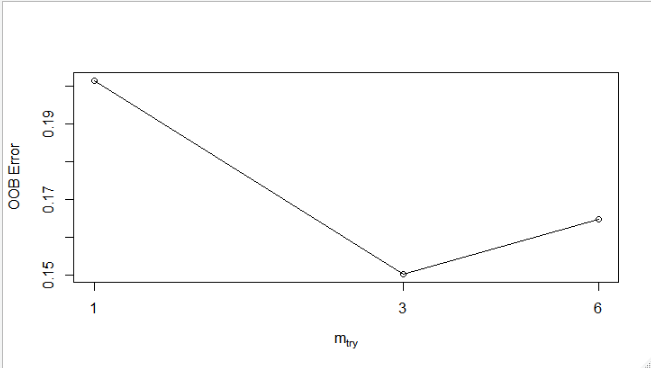
High 21 15

Low 9 72

**Accuracy 🡺0.7948**



**Turning the Random Forest**



**Model-3 🡺**

**Confusion Matrix**

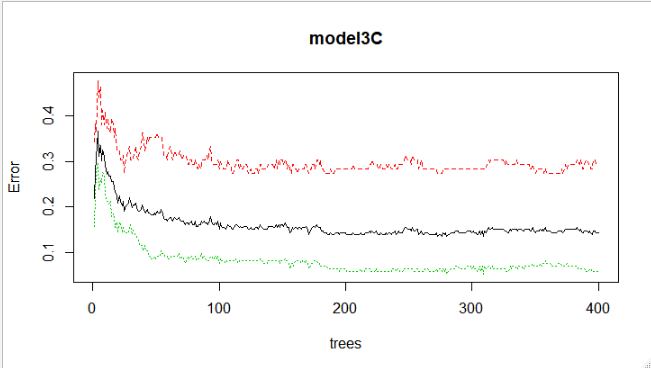
Predicted

Actual High Low

High 25 11

Low 7 74

**Accuracy 🡺0.8461**



**Based on above information we can infer that Model -3 is good model with accuracy 84.61%**