Kiran Shahi

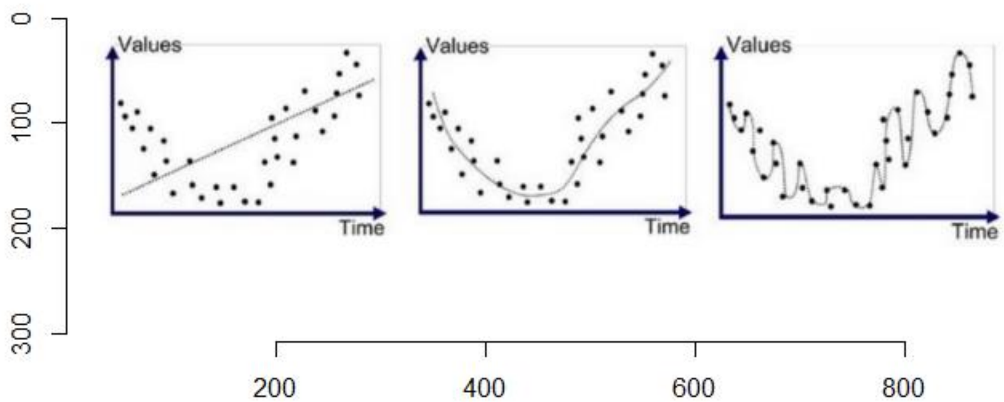
Lab Assignment # 7 - Nonlinear Regression and Decision Trees

Input

```
install.packages("caret")
install.packages("tidyr")
install.packages("ggplot2")
install.packages("lattice")
install.packages("rpart.plot")
install.packages("ggthemes")
install.packages("rattle")
```

\# 1. Entropy uses a log computation and Gini Impurity involves a square computation. The square computation is cheaper than log computation; therefore, we would consider using the Gini Impurity method instead of the Entropy method.

```
# 2. Load the library imager
library(imager)

im <-load.image("C:/Users/minti/Documents/McMaster/DataModelling/Threegraphs.PNG")
plot(im)
```



Graph 1- Underfitted: The model doesn't have sufficient degree of freedom to capture the underlying trend of the data.

Graph 2: Good Fit: The model hasn't captured any single noise of data and it fits too well. The model is unaffected by the noise in the data.

Graph 3: Overfitted: The model is better predicting from its training data but losing its ability to generalize to new data that it hasn't seen before.

The overfitting is an issue in ML because the model learns all the detail and noise from the training data which will deteriorate the prediction performance of the model.

3. # Read the CSV file

Iris<-read.csv("C:/Users/minti/Documents/McMaster/DataModelling/Lab Assignment/Week 7/Iris.csv")

print(Iris)

Load the caret package to partition the data

```
library(caret)
library(ggplot2)

# Split the data into training and testing data

# Use the dataset to create the partition (70% training, 30% testing)
index<-createDataPartition(Iris$Species, p=0.70, list=FALSE)
# Select 30% of the data for testing
testset <- Iris[-index,]

# Select 70% of the data to train the models
trainset <- Iris[index,]

# Summarize the data
summary(trainset)

# Use level function of the prediction column
levels(trainset$Species)

# Plot using ggplot2

# Use scatter plot
g <- ggplot(data=trainset, aes(x = SepalLengthCm, y = SepalWidthCm))+ geom_point()
print(g)

g <-g + geom_point(aes(color=Species, shape=Species)) + xlab("SepalLengthCm") +
ylab("SepalWidthCm") +ggtitle("SepalLengthCm-WidthCm")+geom_smooth(method="lm")
```

```
print(g)

## Box Plot
box <- ggplot(data=trainset, aes(x=Species, y=PetalLengthCm)) + geom_boxplot(aes(fill=Species)) + ylab("PetalLengthCm") + ggtitle("Iris Boxplot") +stat_summary(fun=mean, geom="point", shape=7, size=5)
print(box)

## Histogram
histogram <- ggplot(data=trainset, aes(x=PetalWidthCm)) + geom_histogram(binwidth=0.6, color="green", aes(fill=Species)) + xlab("PetalWidthCm") +  ylab("Frequency") + ggtitle("Histogram of Petal Width")
histogram

library(rpart)
library(rpart.plot)
set.seed(500)
?rpart

# Fit the model

fit <- rpart(Species ~SepalLengthCm + SepalWidthCm + PetalLengthCm + PetalWidthCm, method="class", data=trainset)
fit

# Displaying the results
printcp(fit)
```

```
# Visualizing cross-validation results
plotcp(fit)

# Summary of splits
summary(fit)



# plot tree


plot(fit, uniform=TRUE, main="Classification Tree for trainset")
text(fit, use.n=TRUE, all=TRUE, cex=0.8)



## Using rattle, tibble, and bitops package to produce some attractive tree plots
library(rattle)
library(tibble)
library(bitops)

fancyRpartPlot(fit)

# Use different colours
fancyRpartPlot(fit, palettes=c("Greens", "Reds"))

## Add a main title to the plot.

fancyRpartPlot(fit, main="Classification Tree for trainset", tweak=0.6)

# Checking how the tree performs on the training data
```

pred<-table(predict(fit,newdata = trainset,type="class"))

pred

4.

Check the accuracy on the testset data

pred_test<-predict(object = fit,newdata = testset,type="class")

table(testset$Species)

table(pred_test)

The training set accuracy is differ than test set accuracy by a lot due to overfitting. It means that the model is losing its ability to generalize to new data (testing data) that it hasn't seen before.

Output

 # 2. Load the library imager

> library(imager)

> im <- load.image("C:/Users/minti/Documents/McMaster/DataModelling/Threegraphs.PNG")

> plot(im)

> Iris<-read.csv("C:/Users/minti/Documents/McMaster/DataModelling/Lab Assignment/Week 7/Iris.csv")

> print(Iris)

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
1	1	5.1	3.5	1.4	0.2	Iris-setosa
2	2	4.9	3.0	1.4	0.2	Iris-setosa
3	3	4.7	3.2	1.3	0.2	Iris-setosa

4	4	4.6	3.1	1.5	0.2	Iris-setosa
5	5	5.0	3.6	1.4	0.2	Iris-setosa
6	6	5.4	3.9	1.7	0.4	Iris-setosa
7	7	4.6	3.4	1.4	0.3	Iris-setosa
8	8	5.0	3.4	1.5	0.2	Iris-setosa
9	9	4.4	2.9	1.4	0.2	Iris-setosa
10	10	4.9	3.1	1.5	0.1	Iris-setosa
11	11	5.4	3.7	1.5	0.2	Iris-setosa
12	12	4.8	3.4	1.6	0.2	Iris-setosa
13	13	4.8	3.0	1.4	0.1	Iris-setosa
14	14	4.3	3.0	1.1	0.1	Iris-setosa
15	15	5.8	4.0	1.2	0.2	Iris-setosa
16	16	5.7	4.4	1.5	0.4	Iris-setosa
17	17	5.4	3.9	1.3	0.4	Iris-setosa
18	18	5.1	3.5	1.4	0.3	Iris-setosa
19	19	5.7	3.8	1.7	0.3	Iris-setosa
20	20	5.1	3.8	1.5	0.3	Iris-setosa
21	21	5.4	3.4	1.7	0.2	Iris-setosa
22	22	5.1	3.7	1.5	0.4	Iris-setosa
23	23	4.6	3.6	1.0	0.2	Iris-setosa
24	24	5.1	3.3	1.7	0.5	Iris-setosa
25	25	4.8	3.4	1.9	0.2	Iris-setosa
26	26	5.0	3.0	1.6	0.2	Iris-setosa
27	27	5.0	3.4	1.6	0.4	Iris-setosa

28	28	5.2	3.5	1.5	0.2	Iris-setosa
29	29	5.2	3.4	1.4	0.2	Iris-setosa
30	30	4.7	3.2	1.6	0.2	Iris-setosa
31	31	4.8	3.1	1.6	0.2	Iris-setosa
32	32	5.4	3.4	1.5	0.4	Iris-setosa
33	33	5.2	4.1	1.5	0.1	Iris-setosa
34	34	5.5	4.2	1.4	0.2	Iris-setosa
35	35	4.9	3.1	1.5	0.1	Iris-setosa
36	36	5.0	3.2	1.2	0.2	Iris-setosa
37	37	5.5	3.5	1.3	0.2	Iris-setosa
38	38	4.9	3.1	1.5	0.1	Iris-setosa
39	39	4.4	3.0	1.3	0.2	Iris-setosa
40	40	5.1	3.4	1.5	0.2	Iris-setosa
41	41	5.0	3.5	1.3	0.3	Iris-setosa
42	42	4.5	2.3	1.3	0.3	Iris-setosa
43	43	4.4	3.2	1.3	0.2	Iris-setosa
44	44	5.0	3.5	1.6	0.6	Iris-setosa
45	45	5.1	3.8	1.9	0.4	Iris-setosa
46	46	4.8	3.0	1.4	0.3	Iris-setosa
47	47	5.1	3.8	1.6	0.2	Iris-setosa
48	48	4.6	3.2	1.4	0.2	Iris-setosa
49	49	5.3	3.7	1.5	0.2	Iris-setosa
50	50	5.0	3.3	1.4	0.2	Iris-setosa
51	51	7.0	3.2	4.7	1.4	Iris-versicolor

52	52	6.4	3.2	4.5	1.5 Iris-versicolor
53	53	6.9	3.1	4.9	1.5 Iris-versicolor
54	54	5.5	2.3	4.0	1.3 Iris-versicolor
55	55	6.5	2.8	4.6	1.5 Iris-versicolor
56	56	5.7	2.8	4.5	1.3 Iris-versicolor
57	57	6.3	3.3	4.7	1.6 Iris-versicolor
58	58	4.9	2.4	3.3	1.0 Iris-versicolor
59	59	6.6	2.9	4.6	1.3 Iris-versicolor
60	60	5.2	2.7	3.9	1.4 Iris-versicolor
61	61	5.0	2.0	3.5	1.0 Iris-versicolor
62	62	5.9	3.0	4.2	1.5 Iris-versicolor
63	63	6.0	2.2	4.0	1.0 Iris-versicolor
64	64	6.1	2.9	4.7	1.4 Iris-versicolor
65	65	5.6	2.9	3.6	1.3 Iris-versicolor
66	66	6.7	3.1	4.4	1.4 Iris-versicolor
67	67	5.6	3.0	4.5	1.5 Iris-versicolor
68	68	5.8	2.7	4.1	1.0 Iris-versicolor
69	69	6.2	2.2	4.5	1.5 Iris-versicolor
70	70	5.6	2.5	3.9	1.1 Iris-versicolor
71	71	5.9	3.2	4.8	1.8 Iris-versicolor
72	72	6.1	2.8	4.0	1.3 Iris-versicolor
73	73	6.3	2.5	4.9	1.5 Iris-versicolor
74	74	6.1	2.8	4.7	1.2 Iris-versicolor
75	75	6.4	2.9	4.3	1.3 Iris-versicolor

76	76	6.6	3.0	4.4	1.4 Iris-versicolor
77	77	6.8	2.8	4.8	1.4 Iris-versicolor
78	78	6.7	3.0	5.0	1.7 Iris-versicolor
79	79	6.0	2.9	4.5	1.5 Iris-versicolor
80	80	5.7	2.6	3.5	1.0 Iris-versicolor
81	81	5.5	2.4	3.8	1.1 Iris-versicolor
82	82	5.5	2.4	3.7	1.0 Iris-versicolor
83	83	5.8	2.7	3.9	1.2 Iris-versicolor
84	84	6.0	2.7	5.1	1.6 Iris-versicolor
85	85	5.4	3.0	4.5	1.5 Iris-versicolor
86	86	6.0	3.4	4.5	1.6 Iris-versicolor
87	87	6.7	3.1	4.7	1.5 Iris-versicolor
88	88	6.3	2.3	4.4	1.3 Iris-versicolor
89	89	5.6	3.0	4.1	1.3 Iris-versicolor
90	90	5.5	2.5	4.0	1.3 Iris-versicolor
91	91	5.5	2.6	4.4	1.2 Iris-versicolor
92	92	6.1	3.0	4.6	1.4 Iris-versicolor
93	93	5.8	2.6	4.0	1.2 Iris-versicolor
94	94	5.0	2.3	3.3	1.0 Iris-versicolor
95	95	5.6	2.7	4.2	1.3 Iris-versicolor
96	96	5.7	3.0	4.2	1.2 Iris-versicolor
97	97	5.7	2.9	4.2	1.3 Iris-versicolor
98	98	6.2	2.9	4.3	1.3 Iris-versicolor
99	99	5.1	2.5	3.0	1.1 Iris-versicolor

100	100	5.7	2.8	4.1	1.3	Iris-versicolor
101	101	6.3	3.3	6.0	2.5	Iris-virginica
102	102	5.8	2.7	5.1	1.9	Iris-virginica
103	103	7.1	3.0	5.9	2.1	Iris-virginica
104	104	6.3	2.9	5.6	1.8	Iris-virginica
105	105	6.5	3.0	5.8	2.2	Iris-virginica
106	106	7.6	3.0	6.6	2.1	Iris-virginica
107	107	4.9	2.5	4.5	1.7	Iris-virginica
108	108	7.3	2.9	6.3	1.8	Iris-virginica
109	109	6.7	2.5	5.8	1.8	Iris-virginica
110	110	7.2	3.6	6.1	2.5	Iris-virginica
111	111	6.5	3.2	5.1	2.0	Iris-virginica
112	112	6.4	2.7	5.3	1.9	Iris-virginica
113	113	6.8	3.0	5.5	2.1	Iris-virginica
114	114	5.7	2.5	5.0	2.0	Iris-virginica
115	115	5.8	2.8	5.1	2.4	Iris-virginica
116	116	6.4	3.2	5.3	2.3	Iris-virginica
117	117	6.5	3.0	5.5	1.8	Iris-virginica
118	118	7.7	3.8	6.7	2.2	Iris-virginica
119	119	7.7	2.6	6.9	2.3	Iris-virginica
120	120	6.0	2.2	5.0	1.5	Iris-virginica
121	121	6.9	3.2	5.7	2.3	Iris-virginica
122	122	5.6	2.8	4.9	2.0	Iris-virginica
123	123	7.7	2.8	6.7	2.0	Iris-virginica

124	124	6.3	2.7	4.9	1.8	Iris-virginica
125	125	6.7	3.3	5.7	2.1	Iris-virginica
126	126	7.2	3.2	6.0	1.8	Iris-virginica
127	127	6.2	2.8	4.8	1.8	Iris-virginica
128	128	6.1	3.0	4.9	1.8	Iris-virginica
129	129	6.4	2.8	5.6	2.1	Iris-virginica
130	130	7.2	3.0	5.8	1.6	Iris-virginica
131	131	7.4	2.8	6.1	1.9	Iris-virginica
132	132	7.9	3.8	6.4	2.0	Iris-virginica
133	133	6.4	2.8	5.6	2.2	Iris-virginica
134	134	6.3	2.8	5.1	1.5	Iris-virginica
135	135	6.1	2.6	5.6	1.4	Iris-virginica
136	136	7.7	3.0	6.1	2.3	Iris-virginica
137	137	6.3	3.4	5.6	2.4	Iris-virginica
138	138	6.4	3.1	5.5	1.8	Iris-virginica
139	139	6.0	3.0	4.8	1.8	Iris-virginica
140	140	6.9	3.1	5.4	2.1	Iris-virginica
141	141	6.7	3.1	5.6	2.4	Iris-virginica
142	142	6.9	3.1	5.1	2.3	Iris-virginica
143	143	5.8	2.7	5.1	1.9	Iris-virginica
144	144	6.8	3.2	5.9	2.3	Iris-virginica
145	145	6.7	3.3	5.7	2.5	Iris-virginica
146	146	6.7	3.0	5.2	2.3	Iris-virginica
147	147	6.3	2.5	5.0	1.9	Iris-virginica

148 148 6.5 3.0 5.2 2.0 Iris-virginica

149 149 6.2 3.4 5.4 2.3 Iris-virginica

150 150 5.9 3.0 5.1 1.8 Iris-virginica

> # Load the caret package to partition the data

> library(caret)

> library(ggplot2)

> # Use the dataset to create the partition (70% training, 30% testing)

> index<-createDataPartition(Iris$Species, p=0.70, list=FALSE)

> # Select 30% of the data for testing

> testset <- Iris[-index,]

> # Select 70% of the data to train the models

> trainset <- Iris[index,]

> # Summarize the data

> summary(trainset)

Id SepalLengthCm SepalWidthCm PetalLengthCm PetalWidthCm Species

Min. : 1.00 Min. :4.400 Min. :2.00 Min. :1.200 Min. :0.100 Length:105

1st Qu.: 36.00 1st Qu.:5.100 1st Qu.:2.80 1st Qu.:1.600 1st Qu.:0.300 Class :character

Median : 75.00 Median :5.800 Median :3.00 Median :4.300 Median :1.300 Mode :character

Mean : 75.21 Mean :5.809 Mean :3.05 Mean :3.728 Mean :1.191

3rd Qu.:113.00 3rd Qu.:6.300 3rd Qu.:3.30 3rd Qu.:5.100 3rd Qu.:1.800

Max. :150.00 Max. :7.700 Max. :4.40 Max. :6.700 Max. :2.500

> # Use level function of the prediction column

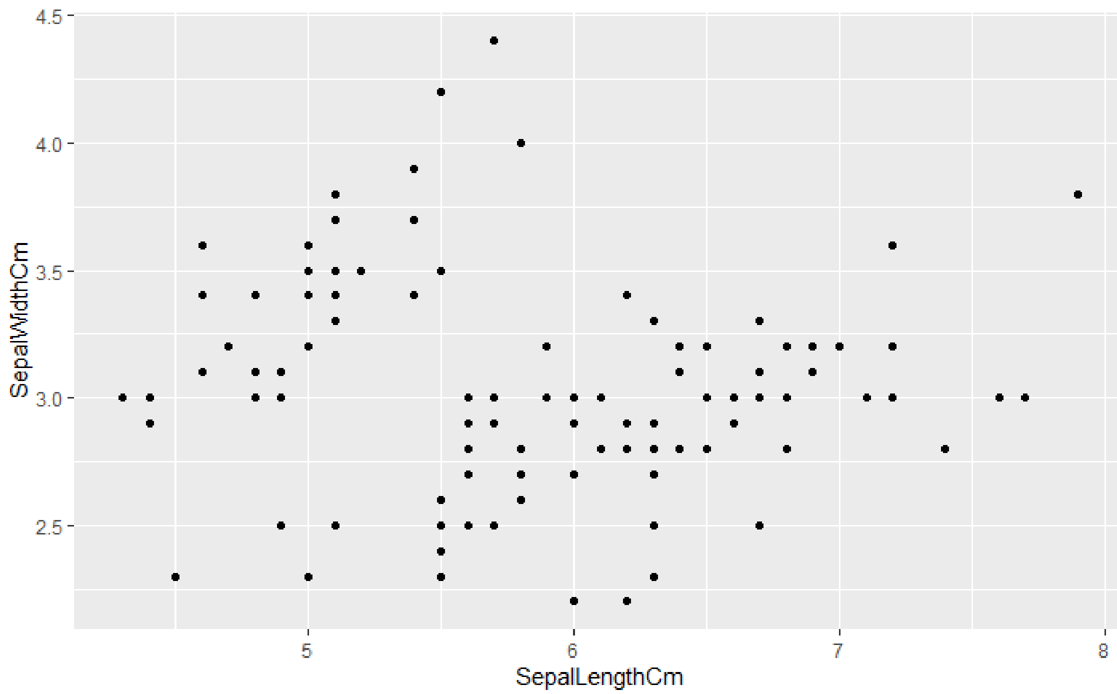
> levels(trainset$Species)

NULL

> # Use scatter plot

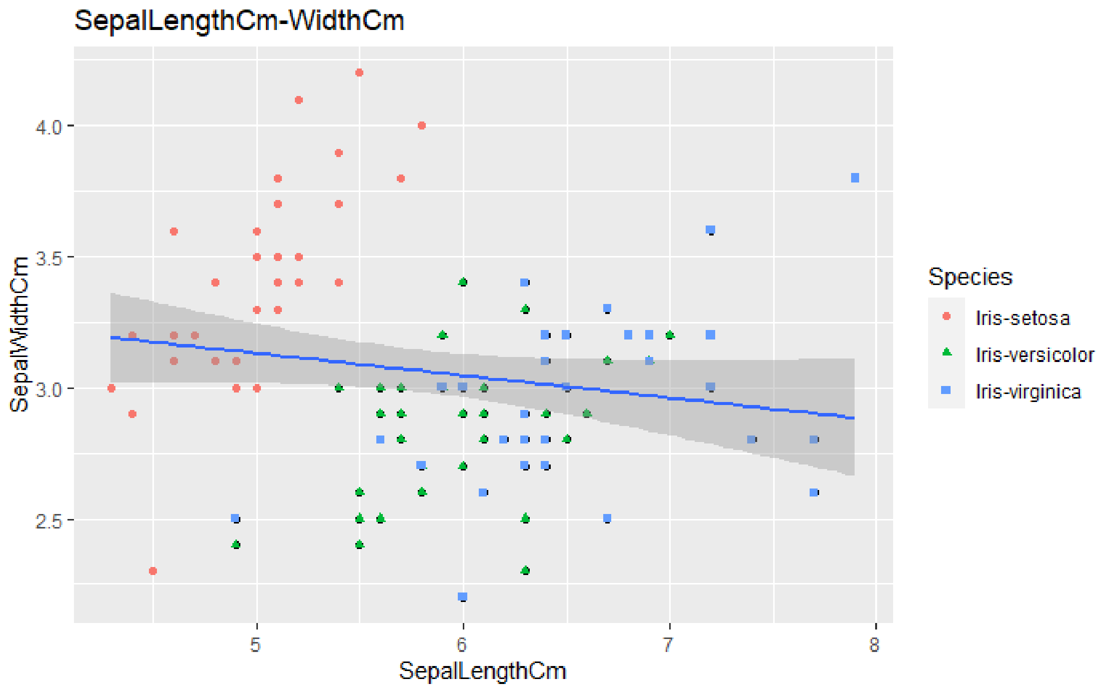
> g <- ggplot(data=trainset, aes(x = SepalLengthCm, y = SepalWidthCm))+ geom_point()

> print(g)



> g <-g + geom_point(aes(color=Species, shape=Species)) + xlab("SepalLengthCm") + ylab("SepalWidthCm") +ggtitle("SepalLengthCm-WidthCm")+geom_smooth(method="lm")

> print(g)

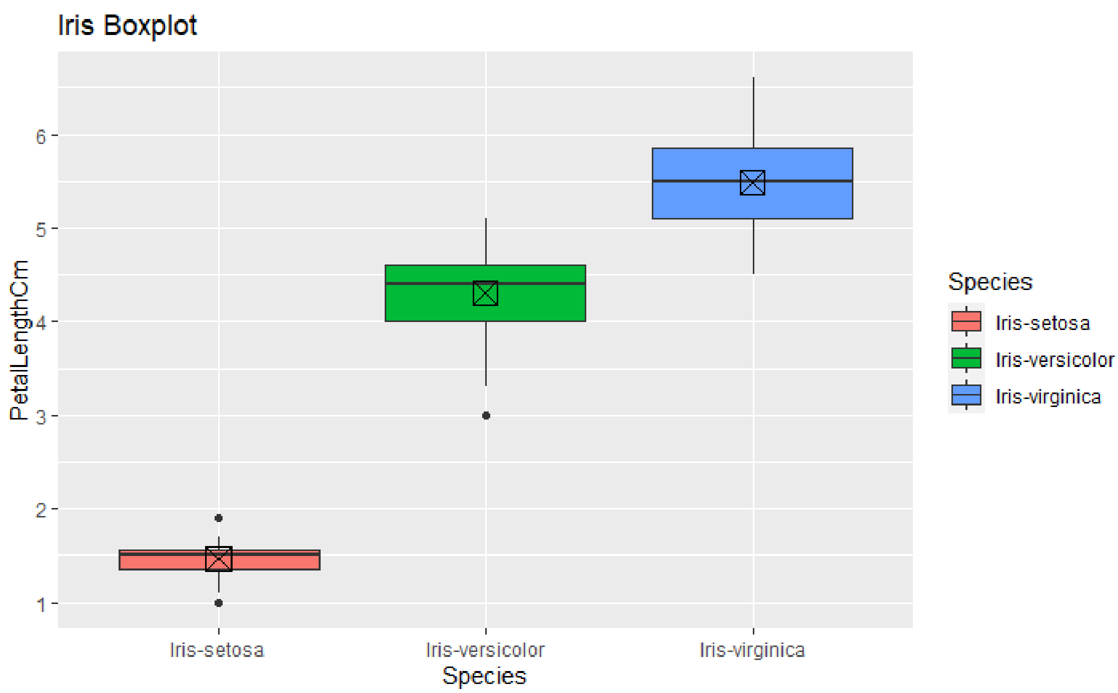


`geom_smooth()` using formula 'y ~ x'

> ## Box Plot

> box <- ggplot(data=trainset, aes(x=Species, y=PetalLengthCm)) + geom_boxplot(aes(fill=Species)) + ylab("PetalLengthCm") + ggtitle("Iris Boxplot") +stat_summary(fun=mean, geom="point", shape=7, size=5)

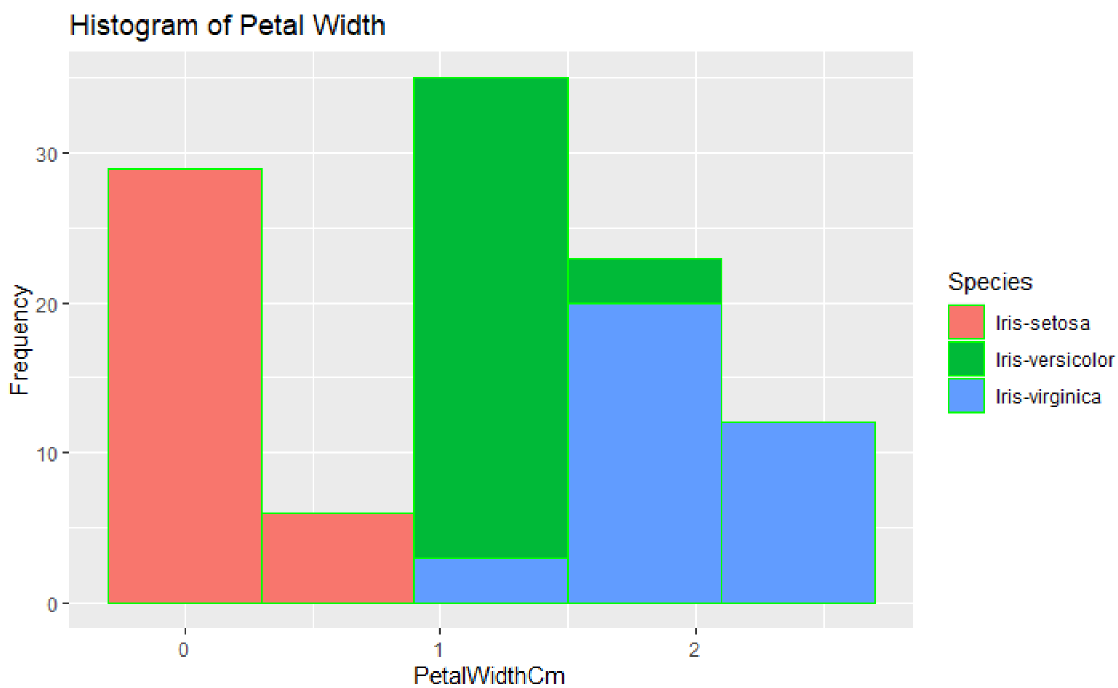
> print(box)



> ## Histogram

> histogram <- ggplot(data=trainset, aes(x=PetalWidthCm)) + geom_histogram(binwidth=0.6, color="green", aes(fill=Species)) + xlab("PetalWidthCm") + ylab("Frequency") + ggtitle("Histogram of Petal Width")

> histogram



> library(rpart)

> library(rpart.plot)

> set.seed(500)

> ?rpart

> fit <- rpart(Species ~SepalLengthCm + SepalWidthCm + PetalLengthCm + PetalWidthCm, method="class", data=trainset)

> fit

n= 105

node), split, n, loss, yval, (yprob)

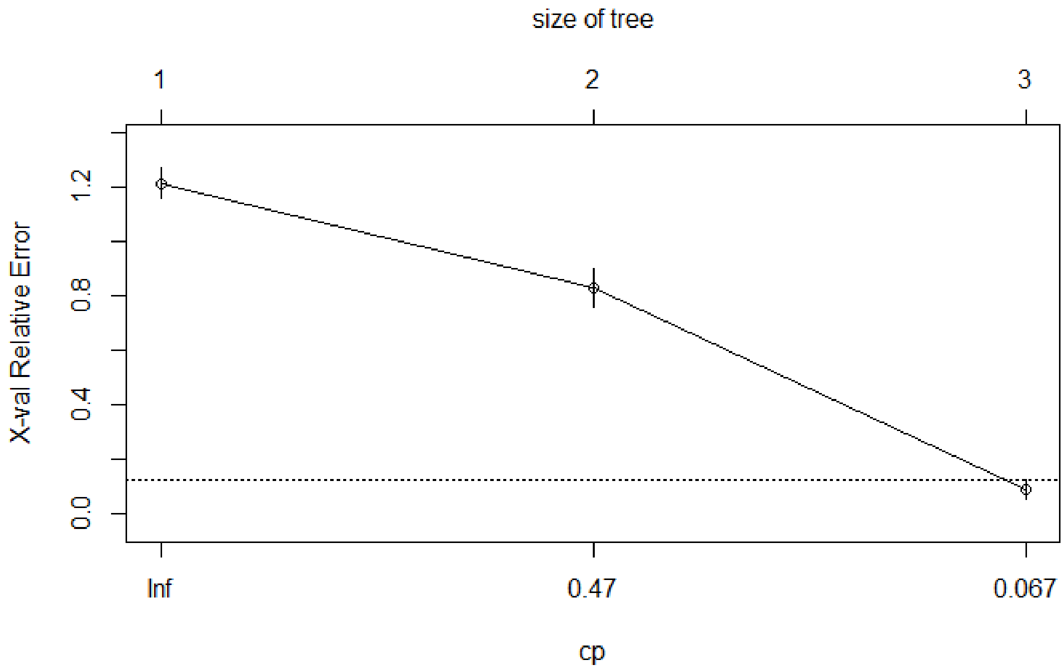
*** denotes terminal node**

1) root 105 70 Iris-setosa (0.33333333 0.33333333 0.33333333)

2) PetalLengthCm< 2.6 35 0 Iris-setosa (1.00000000 0.00000000 0.00000000) *

```
  3) PetalLengthCm>=2.6 70 35 Iris-versicolor (0.00000000 0.50000000
0.50000000)

    6) PetalLengthCm< 4.75 33  1 Iris-versicolor (0.00000000 0.96969697
0.03030303) *

    7) PetalLengthCm>=4.75 37  3 Iris-virginica (0.00000000 0.08108108
0.91891892) *

> # Displaying the results

> printcp(fit)


Classification tree:

rpart(formula = Species ~ SepalLengthCm + SepalWidthCm + PetalLengthCm +

    PetalWidthCm, data = trainset, method = "class")


Variables actually used in tree construction:

[1] PetalLengthCm


Root node error: 70/105 = 0.66667


n= 105


       CP nsplit rel error   xerror     xstd

1 0.50000      0  1.000000 1.214286 0.057482

2 0.44286      1  0.500000 0.828571 0.072790

3 0.01000      2  0.057143 0.085714 0.033978

> # Visualizing cross-validation results
```

> plotcp(fit)



> # Summary of splits

> summary(fit)

Call:

rpart(formula = Species ~ SepalLengthCm + SepalWidthCm + PetalLengthCm + PetalWidthCm, data = trainset, method = "class")

n= 105

	CP	nsplit	rel error	xerror	xstd
1	0.5000000	0	1.00000000	1.21428571	0.05748199
2	0.4428571	1	0.50000000	0.82857143	0.07278975
3	0.0100000	2	0.05714286	0.08571429	0.03397821

Variable importance

```
PetalLengthCm  PetalWidthCm SepalLengthCm  SepalWidthCm
        34           31           22           13

Node number 1: 105 observations,    complexity param=0.5
  predicted class=Iris-setosa      expected loss=0.6666667  P(node) =1
    class counts:    35    35    35
   probabilities: 0.333 0.333 0.333
  left son=2 (35 obs) right son=3 (70 obs)
  Primary splits:
      PetalLengthCm < 2.6  to the left,  improve=35.00000, (0 missing)
      PetalWidthCm  < 0.8  to the left,  improve=35.00000, (0 missing)
      SepalLengthCm < 5.45 to the left,  improve=25.25684, (0 missing)
      SepalWidthCm  < 3.25 to the right, improve=11.46552, (0 missing)
  Surrogate splits:
      PetalWidthCm  < 0.8  to the left,  agree=1.000, adj=1.000, (0 split)
      SepalLengthCm < 5.45 to the left,  agree=0.933, adj=0.800, (0 split)
      SepalWidthCm  < 3.25 to the right, agree=0.810, adj=0.429, (0 split)

Node number 2: 35 observations
  predicted class=Iris-setosa      expected loss=0  P(node) =0.3333333
    class counts:    35     0     0
   probabilities: 1.000 0.000 0.000

Node number 3: 70 observations,    complexity param=0.4428571
```

predicted class=Iris-versicolor expected loss=0.5 P(node) =0.6666667

class counts: 0   35   35

probabilities: 0.000 0.500 0.500

left son=6 (33 obs) right son=7 (37 obs)

Primary splits:

PetalLengthCm < 4.75 to the left, improve=27.547090, (0 missing)

PetalWidthCm < 1.75 to the left, improve=24.346570, (0 missing)

SepalLengthCm < 5.75 to the left, improve= 7.329060, (0 missing)

SepalWidthCm < 2.45 to the left, improve= 3.123862, (0 missing)

Surrogate splits:

PetalWidthCm < 1.55 to the left, agree=0.914, adj=0.818, (0 split)

SepalLengthCm < 5.75 to the left, agree=0.757, adj=0.485, (0 split)

SepalWidthCm < 2.65 to the left, agree=0.671, adj=0.303, (0 split)

Node number 6: 33 observations

predicted class=Iris-versicolor expected loss=0.03030303 P(node) =0.3142857

class counts: 0   32   1

probabilities: 0.000 0.970 0.030

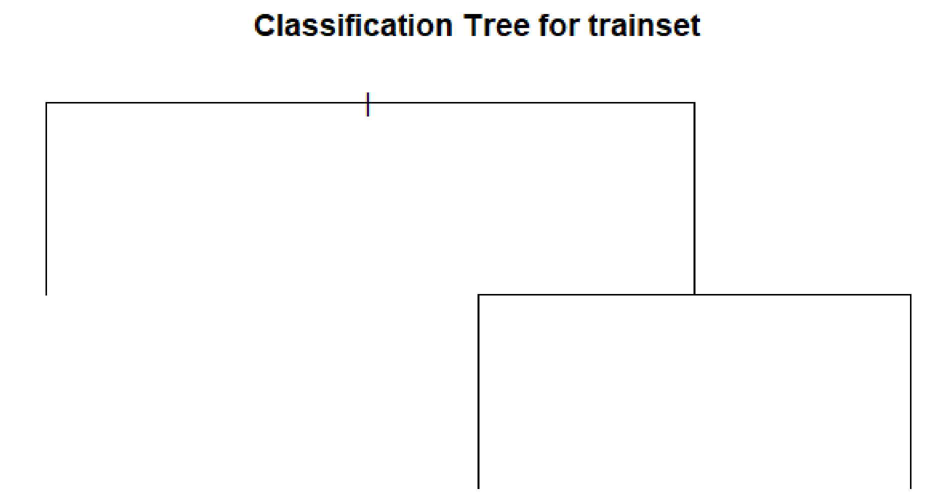
Node number 7: 37 observations

predicted class=Iris-virginica   expected loss=0.08108108 P(node) =0.352381

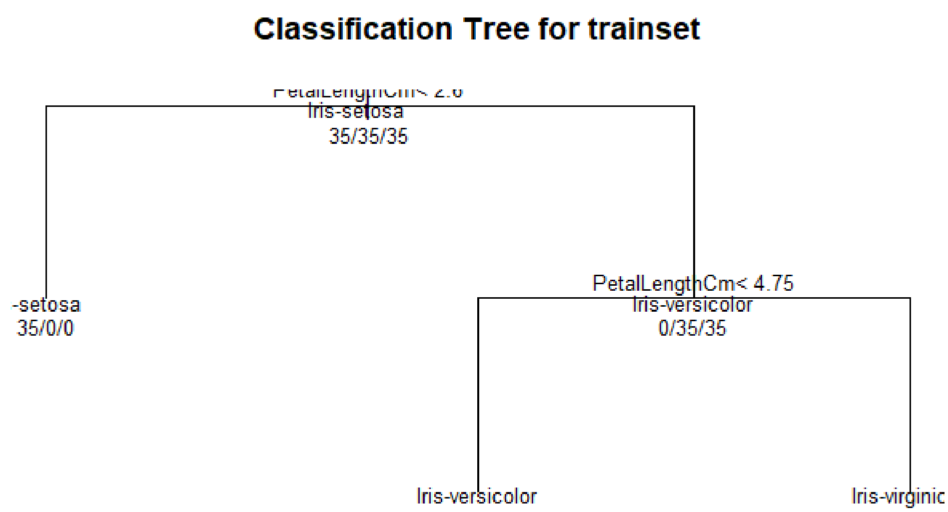
class counts: 0   3   34

probabilities: 0.000 0.081 0.919

> plot(fit, uniform=TRUE, main="Classification Tree for trainset")



> text(fit, use.n=TRUE, all=TRUE, cex=0.8)



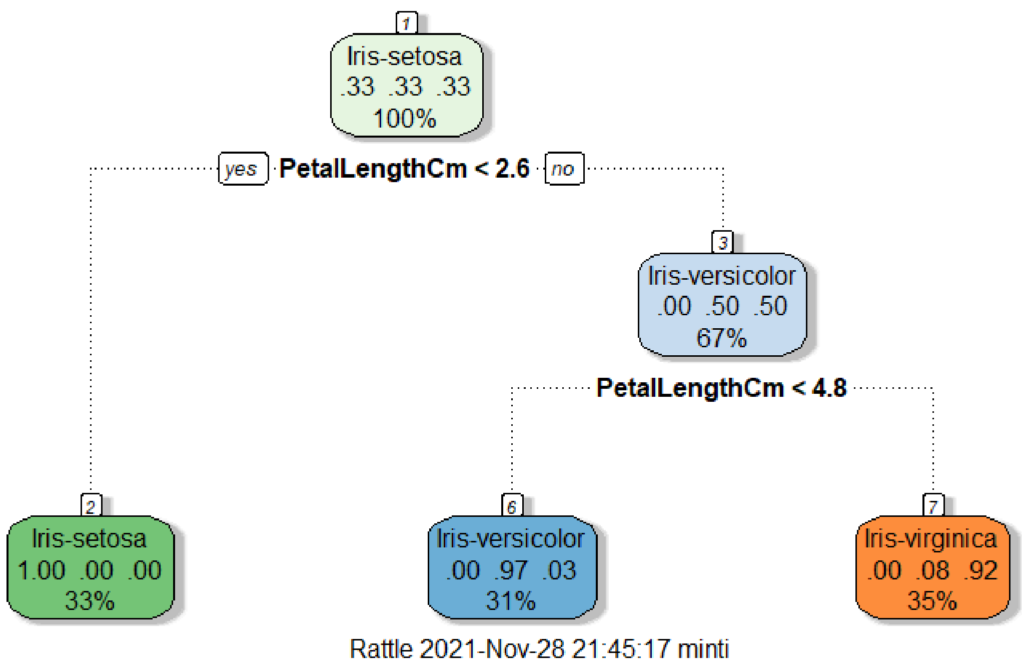
> ## Using rattle, tibble, and bitops package to produce some attractive tree plots

> library(rattle)

> library(tibble)

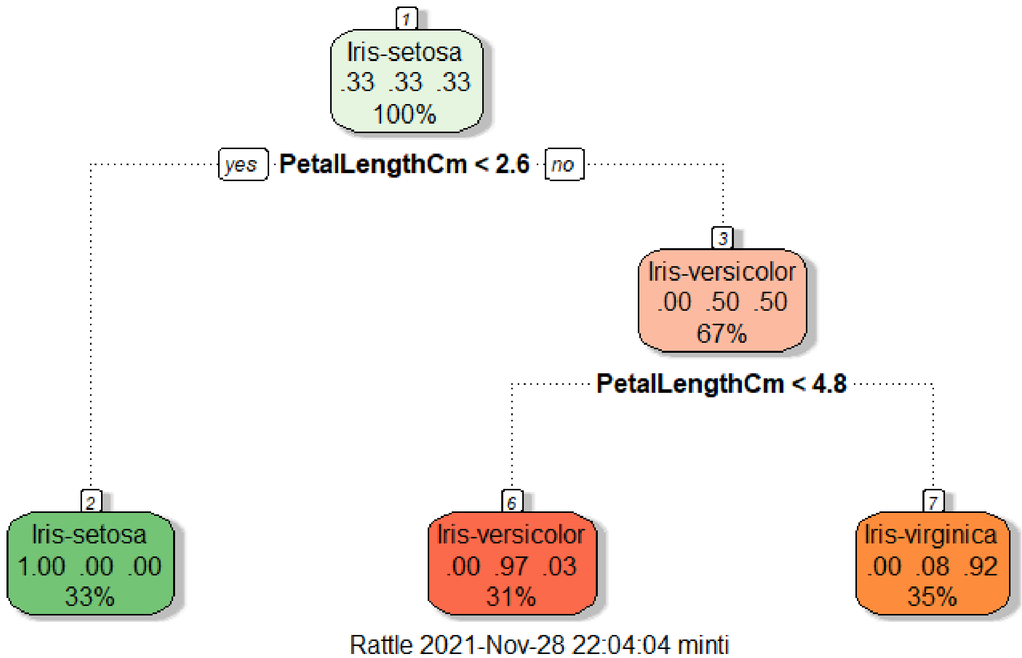
> library(bitops)

> fancyRpartPlot(fit)

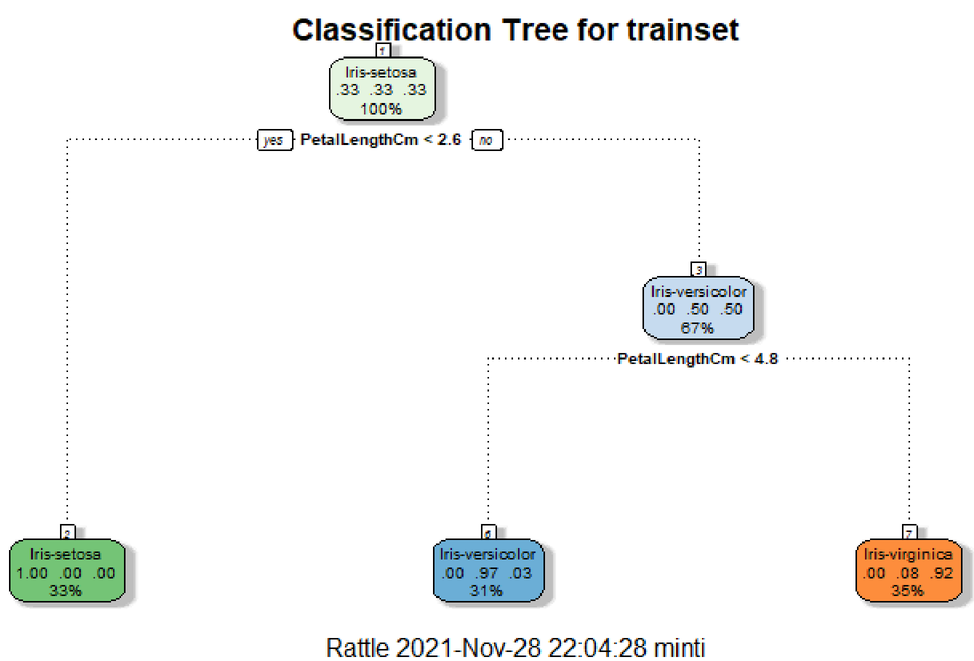


> # Use different colours

> fancyRpartPlot(fit, palettes=c("Greens", "Reds"))



> fancyRpartPlot(fit, main="Classification Tree for trainset", tweak=0.6)



> # Checking how the tree performs on the training data

> pred<-table(predict(fit,newdata = trainset,type="class"))

> pred

```
   Iris-setosa Iris-versicolor  Iris-virginica
            35              33              37
> # 4.
> # Check the accuracy on the testset data
> pred_test<-predict(object = fit,newdata = testset,type="class")
> table(testset$Species)

   Iris-setosa Iris-versicolor  Iris-virginica
            15              15              15
> table(pred_test)
pred_test
   Iris-setosa Iris-versicolor  Iris-virginica
            15              12              18
>
> # The training set accuracy is differ than test set accuracy by a lot due to overfitting. It means that the model is losing its ability to  generalize to new data (testing data) that it hasn't seen before.
```