Lecture

IAQS | INSTITUTE OF ACTUARIAL & QUANTITATIVE STUDIES

Mrs. Indrani Sen

Class: SY BSc

Subject : STATISTICAL MODELLING IN R

Subject Code: SMR

Chapter: Unit 4 Chp 1

Chapter Name: Decision tree

Classification

Classification maps data into predefined groups or classes.

It is referred to as supervised learning.

E.g Good or Bad customer for loan approval

Pattern recognition can be used e.g face recognition

Identifying a pattern on the face which recognizes a criminal

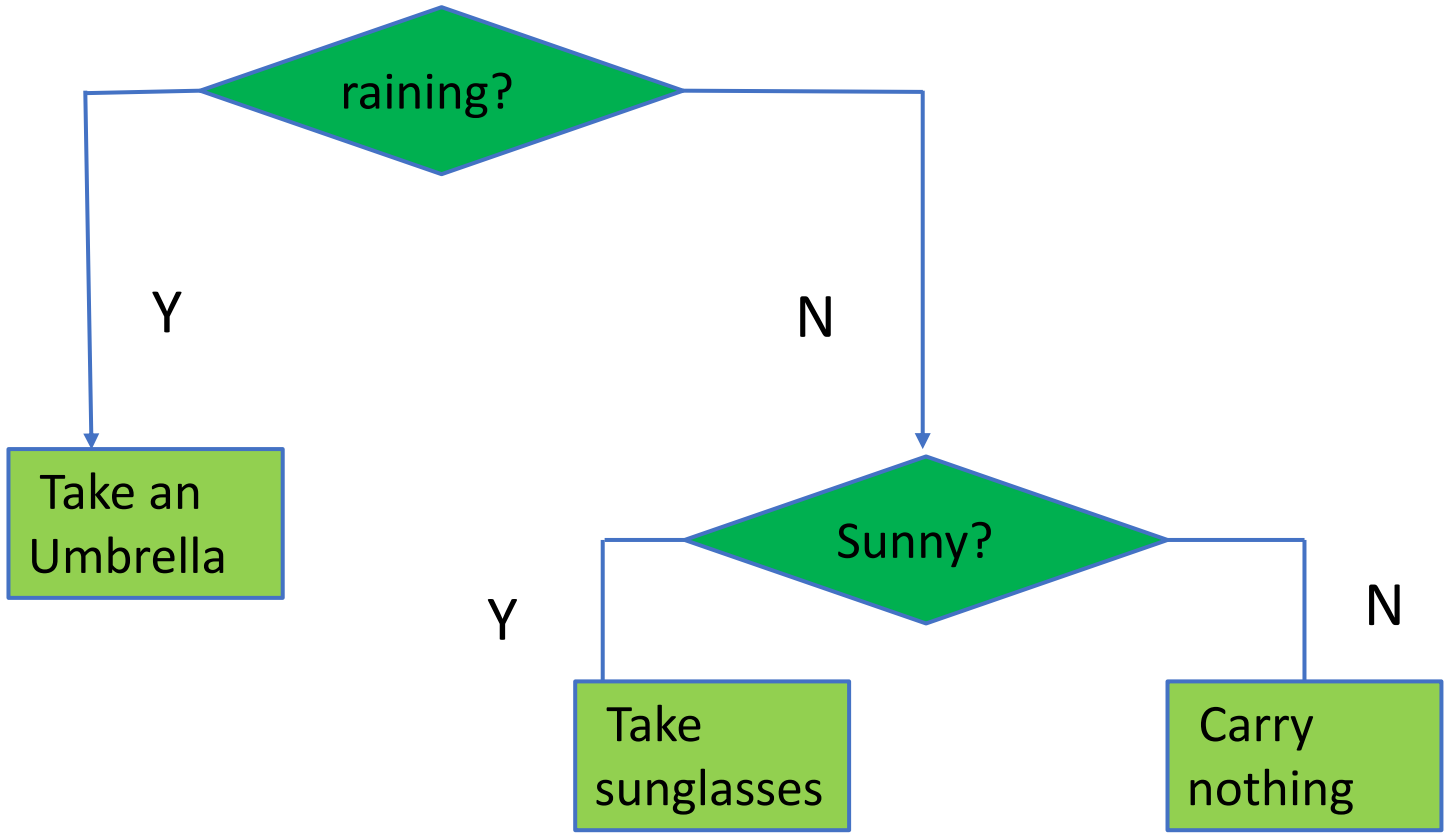
Classification and regression trees

The classification tree splits the response variable into mainly two classes *Yes* or *No*

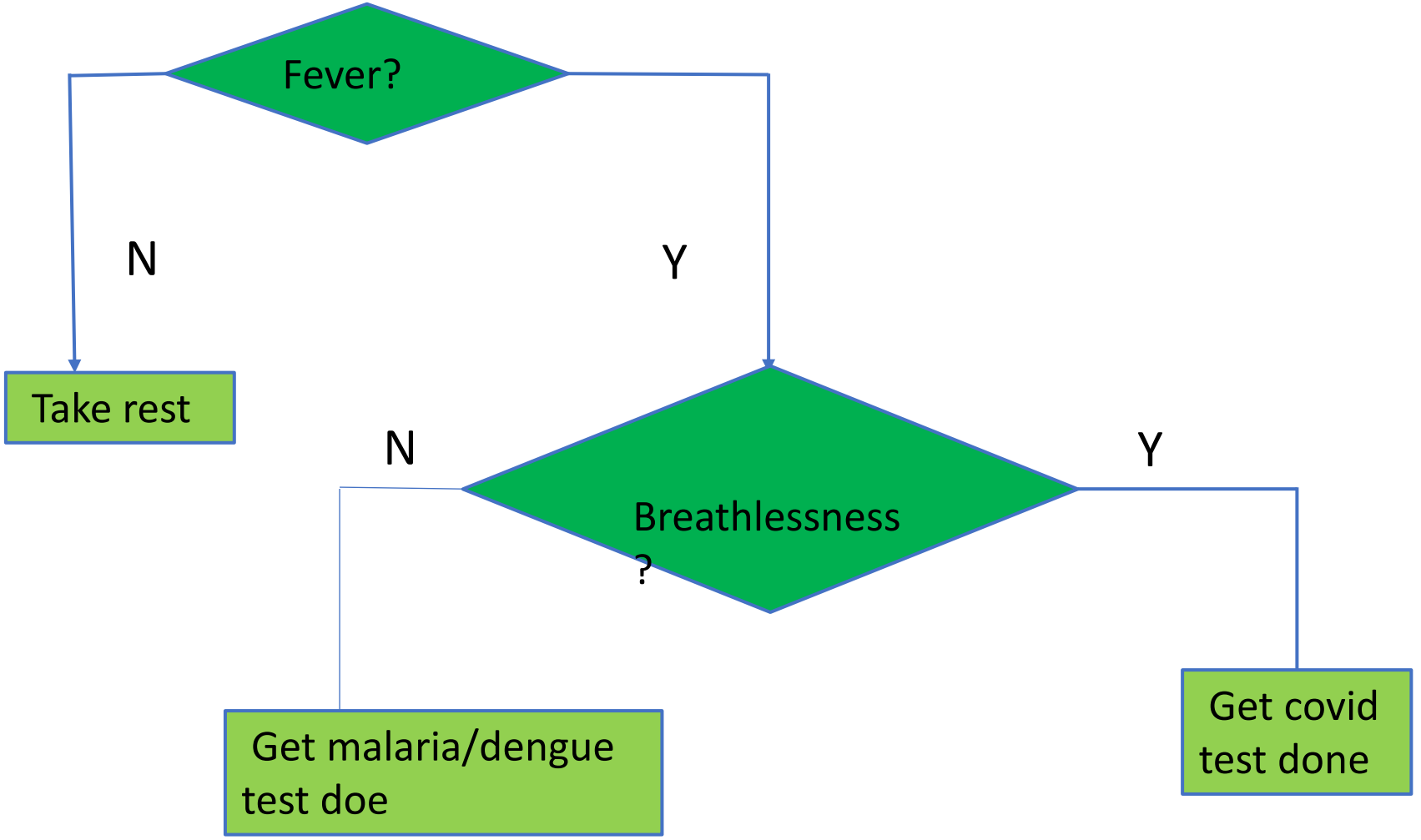
also can be numerically categorized as 1 or 0.

This is the reason why classification tree is applied when there is a need for categorical variable for categorical outcome.

Decision tree to carry specific things while going out



MEDICAL DIAGNOSIS



Decision tree algorithm

The core algorithm for building decision trees called ID3 by J. R. Quinlan which employs a topdown, greedy search through the space of possible branches with no backtracking.

ID3 uses Entropy and Information Gain to construct a decision tree.

Important Terminology related to Decision Trees

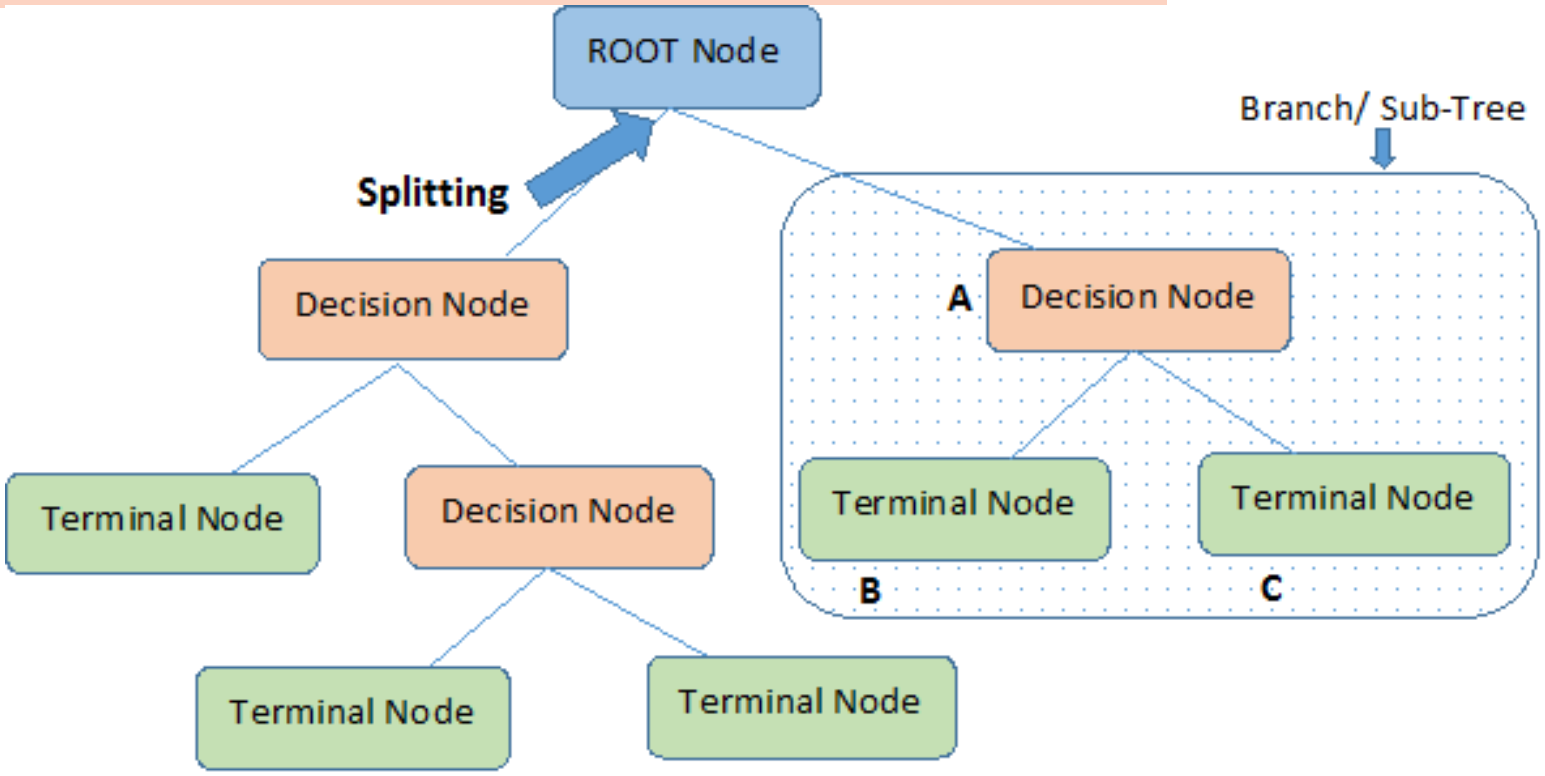
Let’s look at the basic terminology used with Decision trees:

Root Node: It represents entire population or sample and this further gets divided into two or more homogeneous sets.

Splitting: It is a process of dividing a node into two or more sub-nodes.

Decision Node: When a sub-node splits into further sub-nodes, then it is called decision node.

Leaf/ Terminal Node: Nodes do not split is called Leaf or Terminal node.



Note:- A is parent node of B and C.

WEATHER DATASET

outlook	temperature	humidity	windy	play
overcast	hot	high	FALSE	yes
overcast	cool	normal	TRUE	yes
overcast	mild	high	TRUE	yes
overcast	hot	normal	FALSE	yes
rainy	mild	high	FALSE	yes
rainy	cool	normal	FALSE	yes
rainy	cool	normal	TRUE	no
rainy	mild	normal	FALSE	yes
rainy	mild	high	TRUE	no
sunny	hot	high	FALSE	no
sunny	hot	high	TRUE	no
sunny	mild	high	FALSE	no
sunny	cool	normal	FALSE	yes
sunny	mild	normal	TRUE	yes

Outlook

Sunny

hot	high	FALSE	no
hot	high	TRUE	no
mild	high	FALSE	no
cool	normal	FALSE	yes
mild	normal	TRUE	yes

Overcast

hot	high	FALSE	yes
cool	normal	TRUE	yes
mild	high	TRUE	yes
hot	normal	FALSE	yes

Rainy

temperature	humidity	windy	play
mild	high	FALSE	yes
cool	normal	FALSE	yes
cool	normal	TRUE	no
mild	normal	FALSE	yes
mild	high	TRUE	no

Windy

TRUE

hot	high	no
mild	normal	yes

FALSE

hot	high	no
mild	high	no
cool	normal	yes

Humidity

High

sunny hot high TRUE no

Normal

sunny mild normal TRUE yes

Humidity

High

hot	high	FALSE	no
mild	high	FALSE	no

Normal

sunny cool normal FALSE yes

Windy

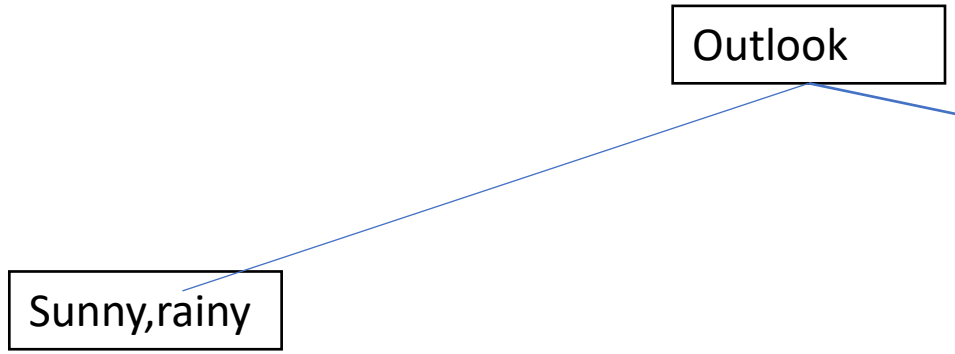
?

Actual attribute selection

Binary subsets

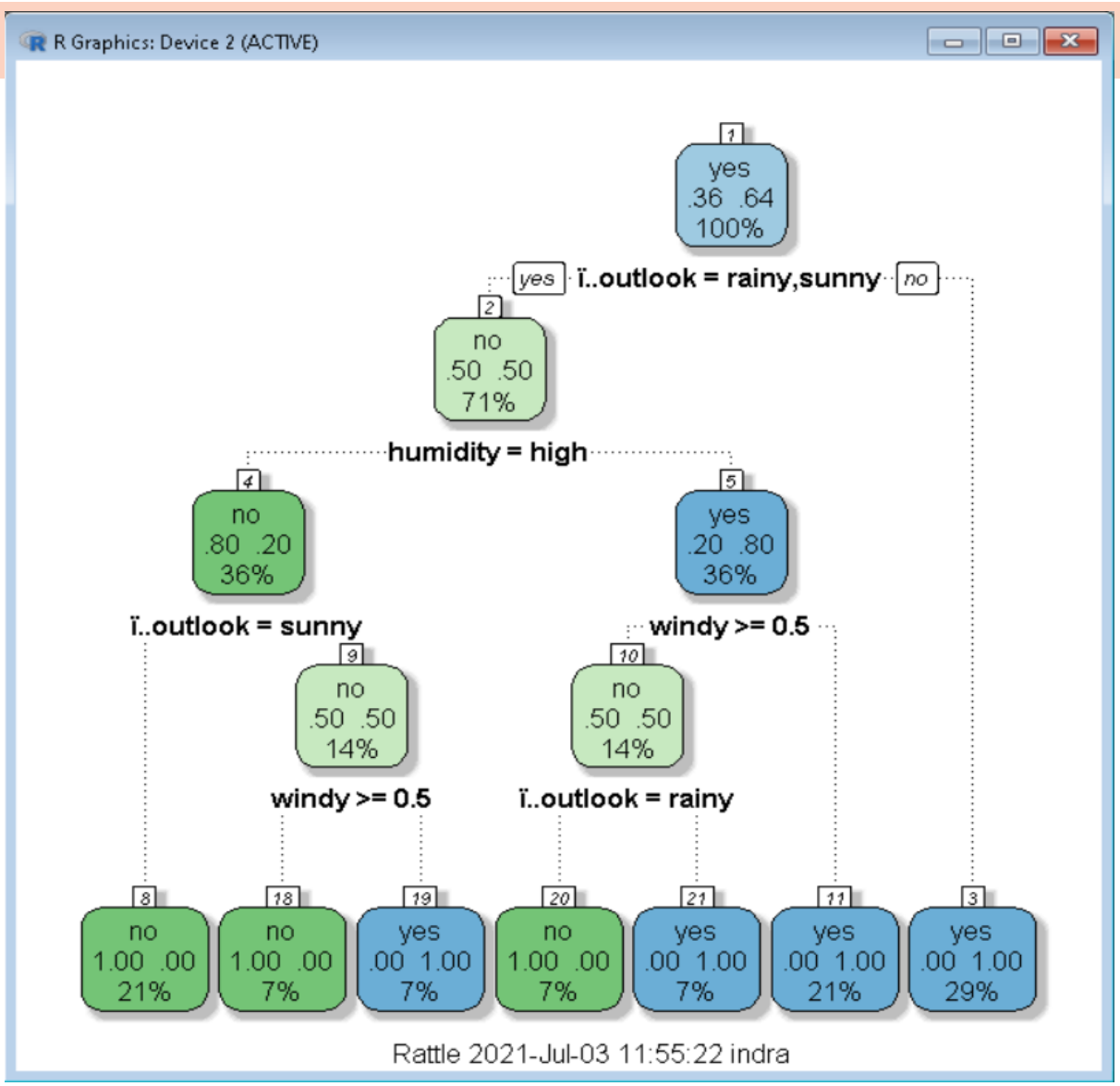
Outlook ([sunny,overcast],rainy)(sunny,[overcast,rainy])([sunny,rainy],overcast)

Temperature([cool,hot],mild)([cool,mild],hot)([mild,hot],cool)



outlook	temperature	humidity	windy	play
rainy	mild	high	FALSE	yes
rainy	cool	normal	FALSE	yes
rainy	cool	normal	TRUE	no
rainy	mild	normal	FALSE	yes
rainy	mild	high	TRUE	no
sunny	hot	high	FALSE	no
sunny	hot	high	TRUE	no
sunny	mild	high	FALSE	no
sunny	cool	normal	FALSE	yes
sunny	mild	normal	TRUE	yes

hot	high	FALSE	yes
cool	normal	TRUE	yes
mild	high	TRUE	yes
hot	normal	FALSE	yes


R Graphics: Device 2 (ACTIVE)
1
yes
.36 .64
100%
yes
ï..outlook = rainy,sunny
no
2
no
.50 .50
71%
humidity = high
4
no
.80 .20
36%
5
yes
.20 .80
36%
ï..outlook = sunny
windy >= 0.5
9
no
.50 .50
14%
10
no
.50 .50
14%
windy >= 0.5
ï..outlook = rainy
8
no
1.00 .00
21%
18
no
1.00 .00
7%
19
yes
.00 1.00
7%
20
no
1.00 .00
7%
21
yes
.00 1.00
7%
11
yes
.00 1.00
21%
3
yes
.00 1.00
29%
Rattle 2021-Jul-03 11:55:22 indra

Entropy

A decision tree is built top-down from a root node and involves partitioning the data into subsets that contain instances with similar values (homogenous).

ID3 algorithm uses entropy to calculate the homogeneity of a sample.

If the sample is completely homogeneous the entropy is zero and if the sample is an equally divided it has entropy of one.

Entropy=$-p \log_2 p - q \log_2 q$

Information gain=Entropy(parent)-weighted sum of entropy(children)

Attribute selection using entropy

Attribute	Entropy	Information gain
Outlook	0.693536139	0.2467
Temperature	0.911063393	0.0292
Wind	0.892158928	0.0481

Gini Gain

Similar to entropy, which had the concept of information gain,

gini gain is calculated when building a decision tree to help determine which attribute gives us the most information about which class a new data point belongs to.

This is done in a similar way to how information gain was calculated for entropy, except instead of taking a weighted sum of the entropies of each branch of a decision, we take a weighted sum of the gini impurity.

$Gini=1-p^2-q^2$

Gini gain=Gini(parent)-weighted sum of gini(children)

Gini index

Attribute	Gini	Gini gain
Outlook	0.342857143	0.1163
Temperature	0.44047619	0.0187
Wind	0.428571429	0.0306
Humidity	0.367346939	0.0918

Reading the csv file

```
> x=read.csv("d:/weather3.csv")

> x

    outlook temperature humidity windy play
1  overcast         hot     high FALSE  yes
2  overcast        cool   normal  TRUE  yes
3  overcast        mild     high  TRUE  yes
4  overcast         hot   normal FALSE  yes
```

Partitioning the sample and training the model

```
> s=sample(14,10)

> weather_tr=x[s,]

> weather_test=x[-s,]
```

```
> weather_tr
    outlook temperature humidity windy play
8     rainy        mild   normal FALSE  yes
7     rainy        cool   normal  TRUE   no
1  overcast         hot     high FALSE  yes
3  overcast        mild     high  TRUE  yes
5     rainy        mild     high FALSE  yes
11    sunny         hot     high  TRUE   no
14    sunny        mild   normal  TRUE  yes
9     rainy        mild     high  TRUE   no
13    sunny        cool   normal FALSE  yes
6     rainy        cool   normal FALSE  yes
```

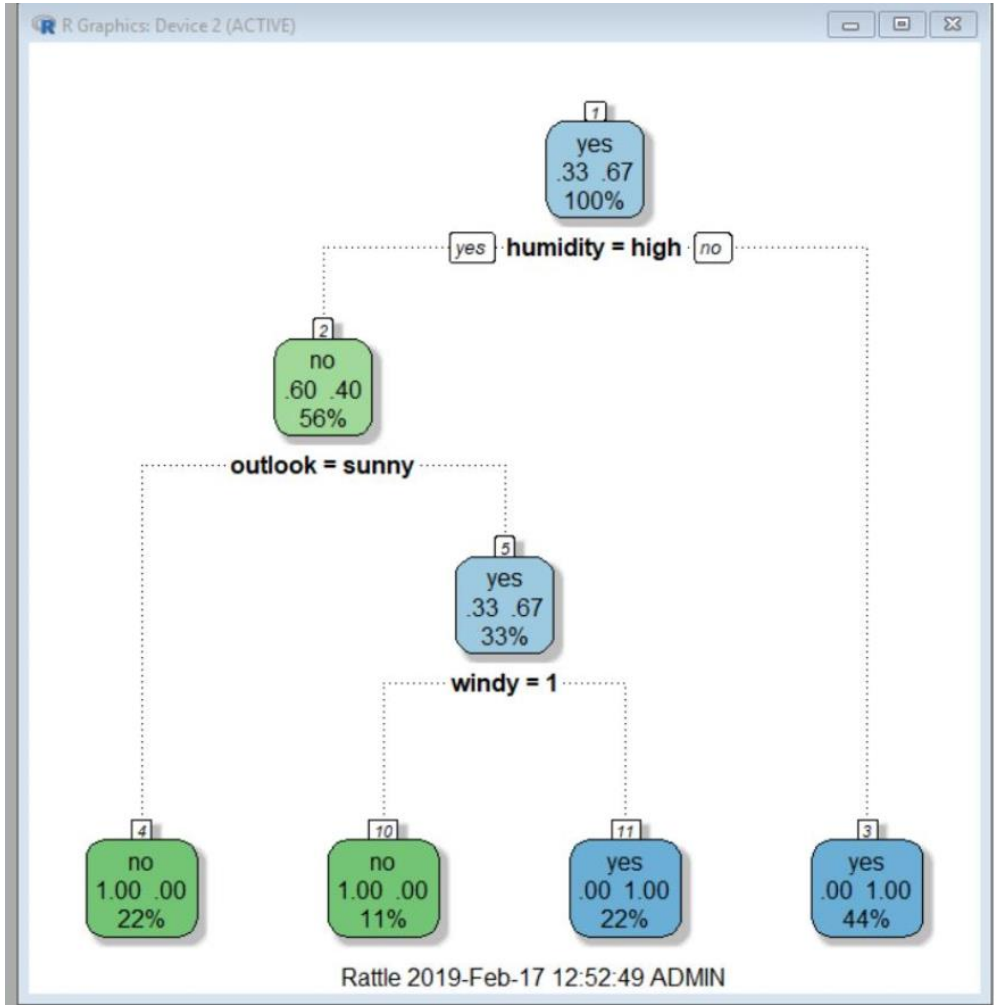
Partitioning the tree has error

```
> dtree=rpart(play~.,weather_tr,method="class",)

> fancyRpartPlot(dtree)

Error in apply(model$frame$yval2[, yval2per], 1, function(x) x[1 + x[1]]) :

  dim(X) must have a positive length
```

SPLITTING THE TREE

```
> dtree=rpart(play ~.,weather_tr, method="class",
+       control = rpart.control(minsplit = 1, minbucket = 1, maxdepth=20 ))
> fancyRpartPlot(dtree)
p=predict(dtree,weather_test)
Table(weather_test[,5],dtree)
```

DECISION TREE



Decision Tree - Regression

Decision tree builds regression or classification models in the form of a tree structure.

It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed.

The final result is a tree with **decision nodes** and **leaf nodes**. A decision node (e.g., Outlook) has two or more branches (e.g., Sunny, Overcast and Rainy), each representing values for the attribute tested.

Leaf node (e.g., Hours Played) represents a decision on the numerical target. The topmost decision node in a tree which corresponds to the best predictor called **root node**.

Decision trees can handle both categorical and numerical data.

Weather dataset

Outlook	temp	Humidity	Windy	Hours_played
Rainy	Hot	High	FALSE	26
Rainy	Hot	High	TRUE	30
Overcast	Hot	High	FALSE	48
Sunny	Mild	High	FALSE	46
Sunny	Cool	Normal	FALSE	62
Overcast	Cool	Normal	TRUE	43
Rainy	Mild	High	FALSE	36
Rainy	Cool	Normal	FALSE	38
Sunny	Mild	Normal	FALSE	48
Rainy	Mild	Normal	TRUE	48
Overcast	Mild	High	TRUE	62
Overcast	Hot	Normal	FALSE	44
Sunny	Mild	High	TRUE	30

Standard Deviation Reduction

A decision tree is built top-down from a root node and involves partitioning the data into subsets that contain instances with similar values (homogenous).

We use standard deviation to calculate the homogeneity of a numerical sample. If the numerical sample is completely homogeneous its standard deviation is zero.

The attribute with the highest standard deviation reduction is chosen for partitioning

Attribute:play						
play	stdev					
14	9.63					
					Attribute SD	SDR
Attribute:outlook	count	stdev	P		8.60	1.03
overcast	4	4.031128874	0.29			
rainy	5	8.700574694	0.36			
sunny	5	12.15	0.36			
					Attribute SD	SDR
Attribute:temperature	count	stdev	P		10.01	-0.38
cool	4	12.13	0.29			
hot	4	10.34005158	0.29			
mild	6	8.38	0.43			
					Attribute SD	SDR
Attribute:wind	count	stdev	P		9.78	-0.15
FALSE	8	8.416607732	0.57			
TRUE	6	11.60459679	0.43			
					Attribute SD	
Attribute: humidity	count	stdev	P		9.77	-0.14
High	7	10.11364001	0.5			
Normal	7	9.433981132	0.5			

TRAINING AND TESTING DATASET

> weather_tr

	Outlook	temp	Humidity	Windy	Hours_played
5	Sunny	Cool	Normal	FALSE	52
9	Rainy	Cool	Normal	FALSE	38
3	Overcast	Hot	High	FALSE	46
13	Overcast	Hot	Normal	FALSE	44
14	Sunny	Mild	High	TRUE	30
12	Overcast	Mild	High	TRUE	52
1	Rainy	Hot	High	FALSE	25
8	Rainy	Mild	High	FALSE	35
10	Sunny	Mild	Normal	FALSE	46

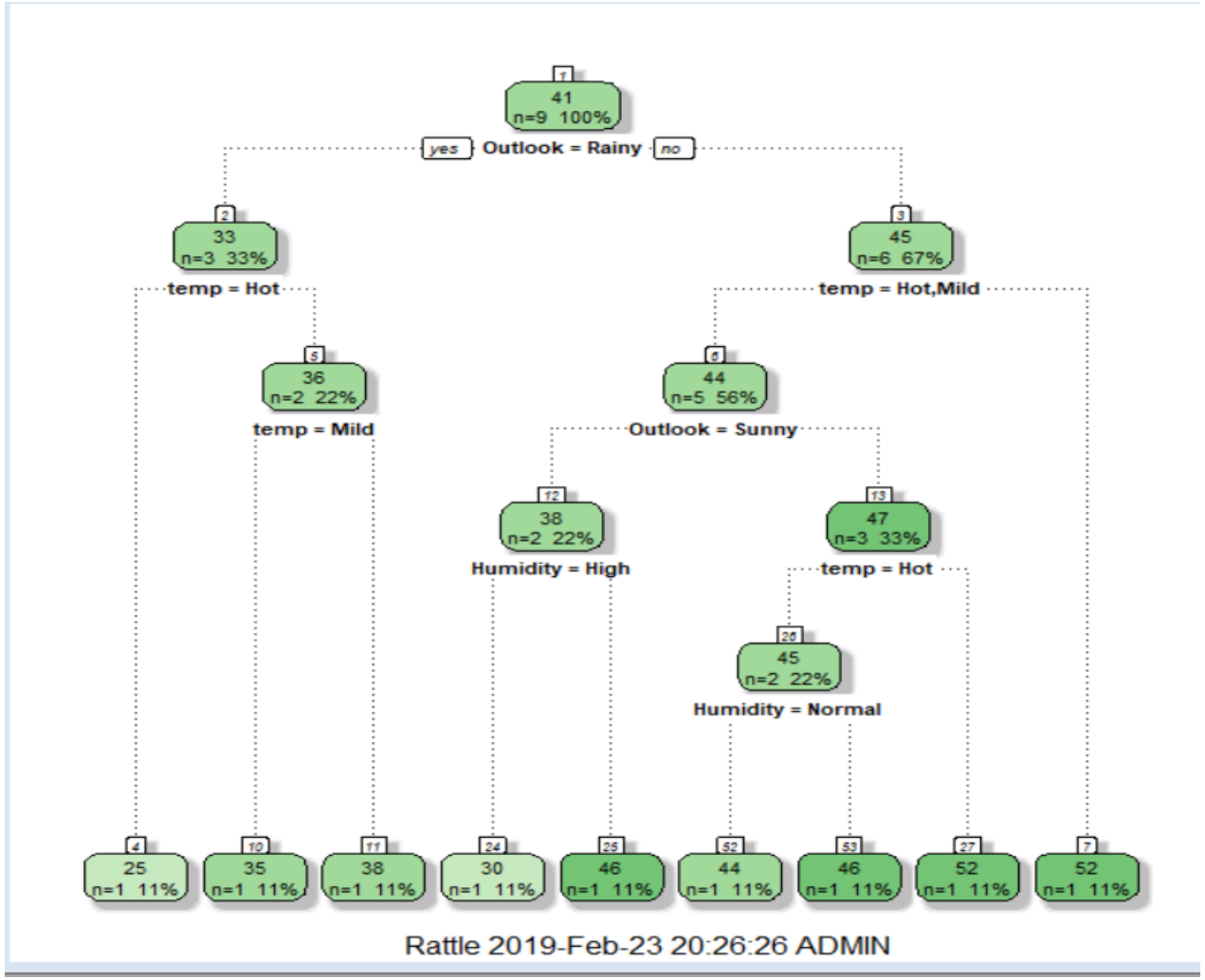
> weather_test

	Outlook	temp	Humidity	Windy	Hours_played
2	Rainy	Hot	High	TRUE	30
4	Sunny	Mild	High	FALSE	45
6	Sunny	Cool	Normal	TRUE	23
7	Overcast	Cool	Normal	TRUE	43
11	Rainy	Mild	Normal	TRUE	48

SPLITTING THE TREE

```
> dtree=rpart(Hours_played ~.,weather_tr, method="anova",
+ control = rpart.control(minsplit = 1, minbucket = 1, maxdepth=20 ))
> fancyRpartPlot(dtree)
```

REGRESSION TREE



PREDICTION

```
> pred=predict(dtree,weather_test)

> pred

 2  4  6  7 11

25 30 52 52 35
```

```
> actual_predict=data.frame(cbind(actuals=weather_test$Hours_played, predicteds=pred))

> actual_predict

   actuals predicteds

2      30         25

4      45         30

6      23         52

7      43         52

11     48         35
```