# Predictions using Decision Tree

R-Programming

Dr Akhter Raza

# Classification Trees C5.0

```
install.packages("C50")
require(C50)
data(iris)
```

```
head(iris)
```

```
  Species
1  setosa
2  setosa
3  setosa
4  setosa
5  setosa
6  setosa
```

```
str(iris)
```

# Classification Trees

Data is sorted we should reshuffle it

table(iris$Species)

```
> table(iris$Species)

    setosa versicolor  virginica
        50         50         50
```

# Preparing data for model

## Reshuffling commands

```
set.seed(9850)

g<- runif(nrow(iris))

irisr <- iris[order(g),]
```

# After shuffling

```
> set.seed(9850)
> g <- runif(nrow(iris))
> irisr <- iris[order(g),]
> str(irisr)
'data.frame':   150 obs. of  5 variables:
 $ Sepal.Length: num  7.1 5.1 6 5.4 5.8 6.9 7.7 5.5 5.7 4.4 ...
 $ Sepal.Width : num  3 3.8 2.2 3.9 2.7 3.1 3.8 2.6 2.6 3.2 ...
 $ Petal.Length: num  5.9 1.5 4 1.3 3.9 4.9 6.7 4.4 3.5 1.3 ...
 $ Petal.Width : num  2.1 0.3 1 0.4 1.2 1.5 2.2 1.2 1 0.2 ...
 $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 3$
```

# Classification Tree

#now check the order of species that is either random or not

```
str(irisr)
head(irisr)
```

# Classification Trees

first 100 rows for of shuffled data is used as training set

To eliminate target feature species from training data we use -5 which

# Format of C5.0

```
# C5.0(training_set,
target_column_training_data)
m1  <- C5.0(irisr[1:100,-5],irisr[1:100,5])
# to see the detailed output of model
m1 use summary command


summary(m1)
```

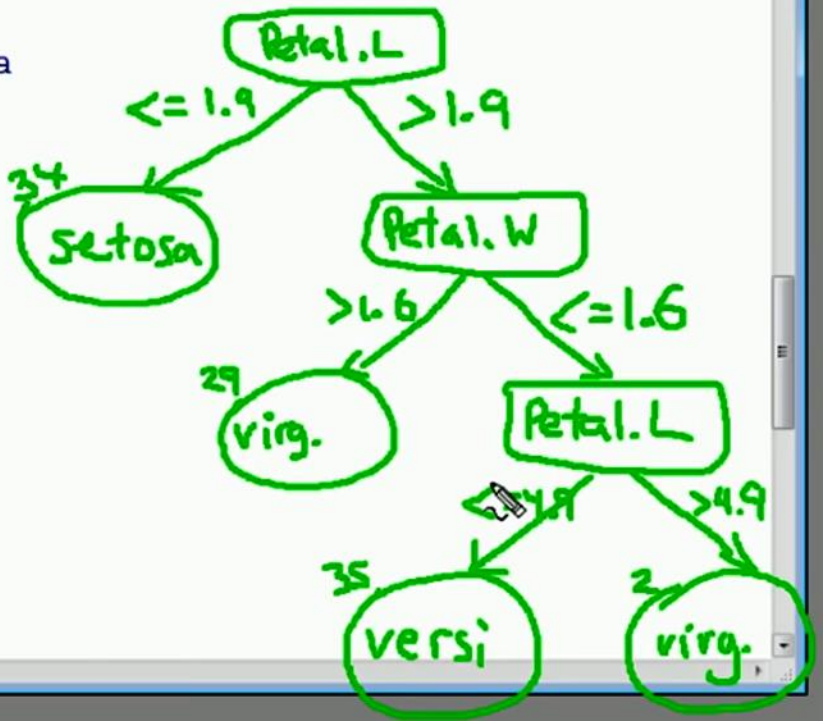
# Output of C5.0

```
> m1 <- C5.0(irisr[1:100,-5], irisr[1:100,5])
> m1

Call:
C5.0.default(x = irisr[1:100, -5], y = irisr[1:100, 5])

Classification Tree
Number of samples: 100
Number of predictors: 4

Tree size: 4
```

# Summary(m1)

R Console

```
Class specified by attribute `outcome'

Read 100 cases (5 attributes) from undefined.data

Decision tree:

Petal.Length <= 1.9: setosa (34)
Petal.Length > 1.9:
:...Petal.Width > 1.6: virginica (29)
    Petal.Width <= 1.6:
    :...Petal.Length <= 4.9: versicolor (35)
        Petal.Length > 4.9: virginica (2)

Evaluation on training data (100 cases):

             Decision Tree
```



# Confusion Matrix

```
Evaluation on training data (100 cases):

           Decision Tree
         ----------------
         Size      Errors

            4    0( 0.0%)   <<


          (a)   (b)   (c)    <-classified as
         ----  ----  ----
           34                 (a): class setosa
                 35           (b): class versicolor
                       31     (c): class virginica
```

misclassification rate is 0 is over fitting

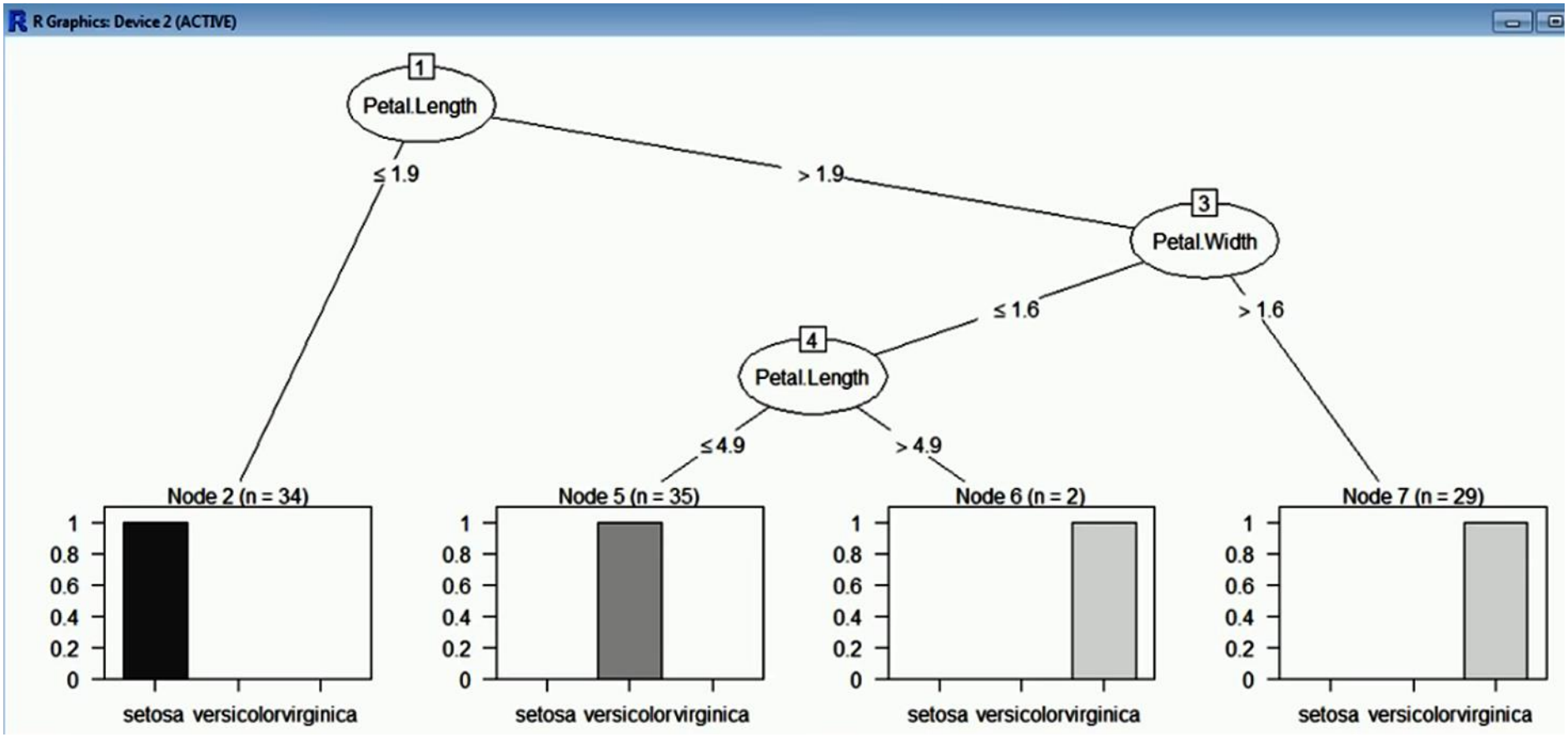
# Predictions from C5.0

```
p1 <- predict(m1, df)

P1

table(irisr[101:150,5],p1)
```

# Confusion matrix of predicted

```
> p1
 [1] virginica  setosa     versicolor virginica  versicolor setosa     setosa
 [8] versicolor versicolor versicolor versicolor virginica  virginica  setosa
[15] versicolor virginica  virginica  virginica  versicolor virginica  setosa
[22] virginica  virginica  setosa     virginica  setosa     setosa     versicolor
[29] setosa     versicolor setosa     virginica  virginica  virginica  setosa
[36] virginica  versicolor virginica  setosa     setosa     virginica  setosa
[43] virginica  virginica  virginica  setosa     virginica  virginica  versicolor
[50] setosa
Levels: setosa versicolor virginica
> table(irisr[101:150,5], Predicted= p1)
            Predicted
             setosa versicolor virginica
  setosa         16          0         0
  versicolor      0         12         3
  virginica       0          0        19
```

# Output of plot(m1)



# Function rpart()

Syntax of rpart(DV~ IV)

. dot indicate all predictors

```
rpart (Species ~ Sepal.Length +
Sepal.Width + Petal.Length +
Petal.Width, data = data_name ,
method="class")
```

# rpart command

use method = class for classification

```
m3 <- rpart (Species ~ ., data =
irisr[1:100,] , method="class")
```

# Output of rpart

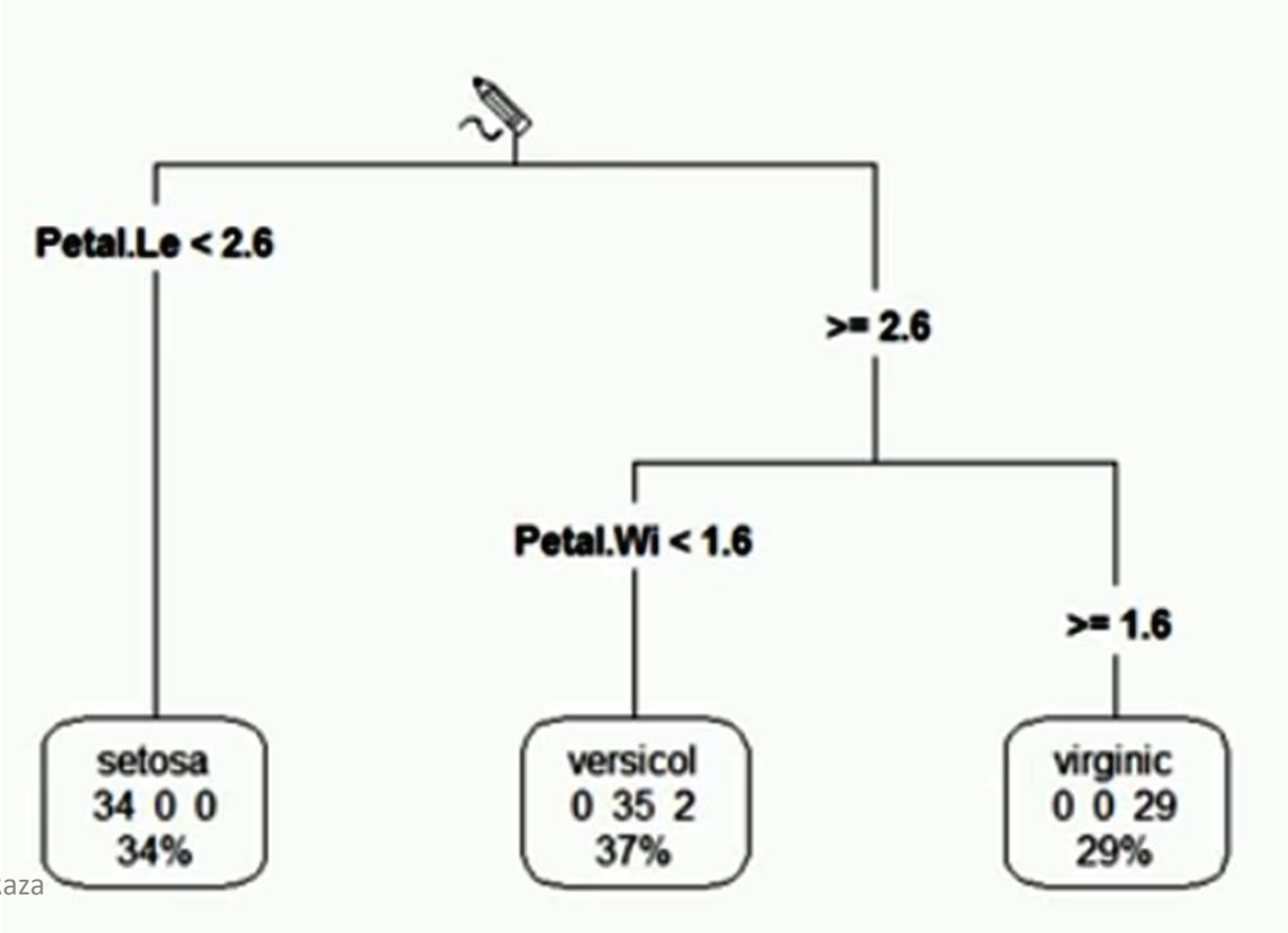
```
> m3 <- rpart(Species ~ ., data=irisr[1:100,], method="class")
> m3
n= 100

node), split, n, loss, yval, (yprob)
      * denotes terminal node

1) root 100 65 versicolor (0.34000000 0.35000000 0.31000000)
  2) Petal.Length< 2.6 34  0 setosa (1.00000000 0.00000000 0.00000000) *
  3) Petal.Length>=2.6 66 31 versicolor (0.00000000 0.53030303 0.46969697)
    6) Petal.Width< 1.65 37  2 versicolor (0.00000000 0.94594595 0.05405405)
    7) Petal.Width>=1.65 29  0 virginica (0.00000000 0.00000000 1.00000000) *
```

# Output of rpart

```
1) root 100 65 versicolor (0.34000000 0.35000000 0.31000000)
  2) Petal.Length< 2.6 34  0 setosa (1.00000000 0.00000000 0.00000000) *
  3) Petal.Length>=2.6 66 31 versicolor (0.00000000 0.53030303 0.46969697)
    6) Petal.Width< 1.65 37  2 versicolor (0.00000000 0.94594595 0.05405405) *
    7) Petal.Width>=1.65 29  0 virginica (0.00000000 0.00000000 1.00000000) *
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
```

# Output of rpart.plot

```
rpart.plot(m3)
rpart.plot(m3, type=3, extra=101, fallen.leaves=T)
```



# Confusion matrix of p1 and p3

```
p3 <- predict(m3, irisr[101:150,], type="class" )

table(irisr[101:150,5],predicted=p3)

# we compare the predicted results of two algorithms
c5.0 and  classifiction tree

table(irisr[101:150,5],p1)
```

# Output of rpart.plot

```
> p3 <- predict(m3, irisr[101:150,], type="class")
> table(irisr[101:150,5], predicted= p3)
            predicted
             setosa versicolor virginica
  setosa         16          0         0
  versicolor      0         13         2
  virginica       0          2        17
```

# Comparison of p1 and p3

```
> table(irisr[101:150,5], predicted= p3)
            predicted
             setosa versicolor virginica
  setosa         16          0         0
  versicolor      0         13         2
  virginica       0          2        17
> table(irisr[101:150,5], predicted= p1)
            predicted
             setosa versicolor virginica
  setosa         16          0         0
  versicolor      0         12         3
  virginica       0          0        19
```

# Questions?