# Machine learning

2024-03-15

**Building a Decision Tree Classifier for Iris Dataset in R**

```
# Load required libraries
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.3.3
```

```
## Loading required package: ggplot2
```

```
## Loading required package: lattice
```

```
library(rpart)
library(rattle)
```

```
## Warning: package 'rattle' was built under R version 4.3.3
```

```
## Loading required package: tibble
```

```
## Loading required package: bitops
```

```
## Rattle: A free graphical interface for data science with R.
## Version 5.5.1 Copyright (c) 2006-2021 Togaware Pty Ltd.
## Type 'rattle()' to shake, rattle, and roll your data.
```

```
# Load the iris dataset
data(iris)

# Print unique species values
print(unique(iris$Species))
```

```
## [1] setosa     versicolor virginica
## Levels: setosa versicolor virginica
```

```
# Print column names
print(names(iris[,1:4]))
```

```
## [1] "Sepal.Length" "Sepal.Width"  "Petal.Length" "Petal.Width"
```

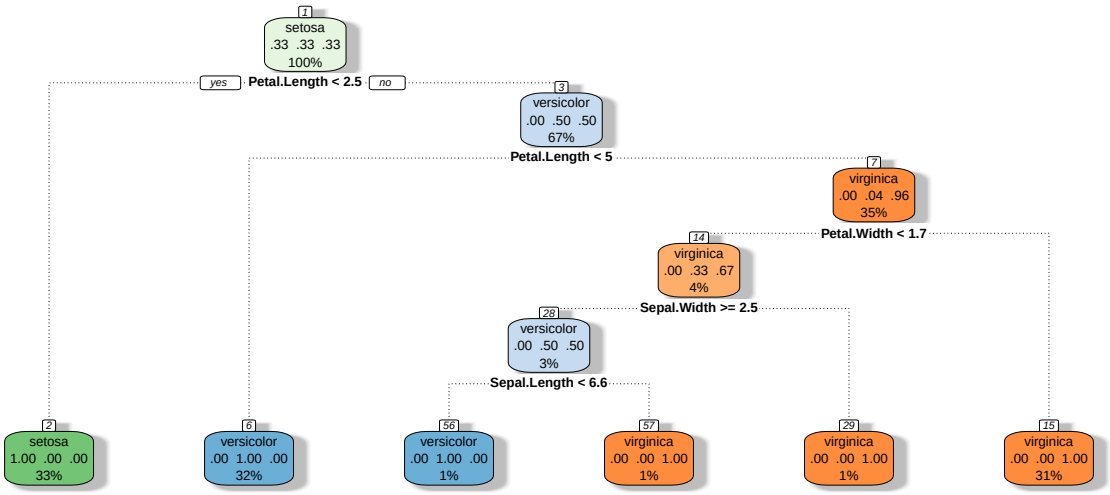
```
# Print number of items in the dataset
print(nrow(iris))
```

```
## [1] 150
```

```
# Split the dataset into training and test sets
set.seed(121)
index = createDataPartition(iris$Species, p = 0.5, list = FALSE)
train = iris[index,]
test = iris[-index,]

# Fit a decision tree classifier
clf_full = rpart(Species ~ ., data = train,
                 control = rpart.control(minsplit = 2, minbucket = 1, cp = 0))

# Plot the decision tree
fancyRpartPlot(clf_full)
```




Rattle 2024−Mar−15 00:48:13 Admin

### Predicted Class for First Test Sample

```
print(test[1,])
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 3          4.7         3.2          1.3         0.2  setosa
```

```
predict(clf_full, newdata = test[1,])
```

```
##   setosa versicolor virginica
## 3      1          0         0
```

### Calculating Training Set Classification Accuracy

```
total = nrow(train)
correct = 0
for(i in 1:total){
  if(train$Species[i] == predict(clf_full, newdata = train[i,], type
                                 = "class")){
    correct = correct + 1
  }
}
accuracy = correct / total
accuracy
```

```
## [1] 1
```

### Calculating Test Set Classification Accuracy

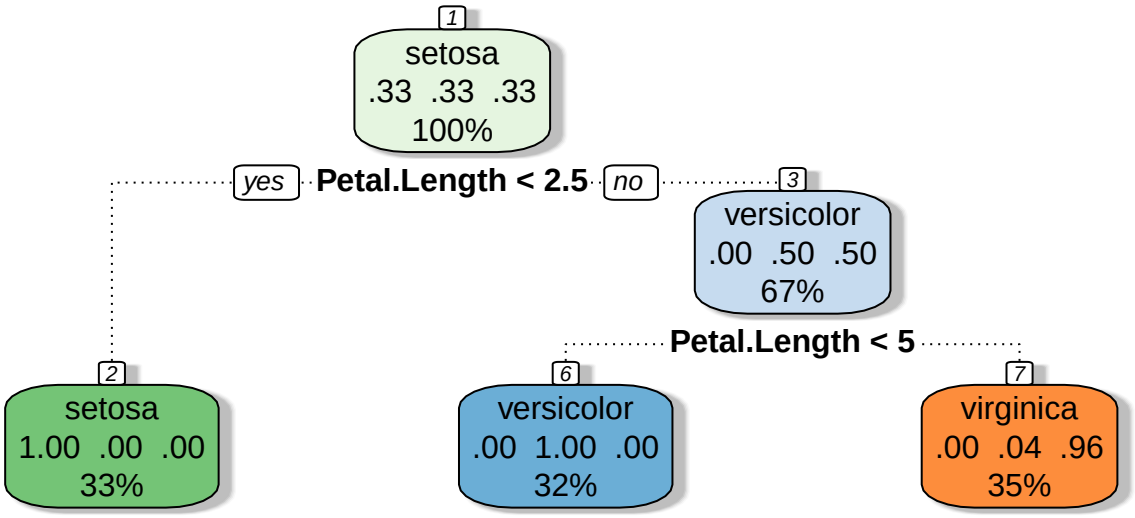
```
sum(predict(clf_full, newdata = test, type = "class") ==
      test$Species)/nrow(test)
```

```
## [1] 0.88
```

### Regularization

```
clf_small = rpart(Species~., data =train, control =
                    rpart.control(maxdepth = 2))

fancyRpartPlot(clf_small)
```




Rattle 2024−Mar−15 00:48:14 Admin

**Calculating Training and Test Set Classification Accuracy for Small Decision Tree**

```
train_acc = sum(predict(clf_small, newdata = train, type = "class") ==
                  test$Species)/nrow(train)
test_acc = sum(predict(clf_small, newdata = test, type = "class") ==
                 test$Species)/nrow(test)
print(cat("Training accuracy: ", train_acc, "Test accuracy: ",
          test_acc))
```

```
## Training accuracy:  0.9866667 Test accuracy:  0.9066667NULL
```