Caret-machinelearning

Han Wang

9/8/2017

1. Data structure and Basic Plotting

```
data("iris")
#structure
str(iris)
```

```
## 'data.frame':    150 obs. of  5 variables:
##  $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
##  $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
##  $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
##  $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
##  $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1 ...
```

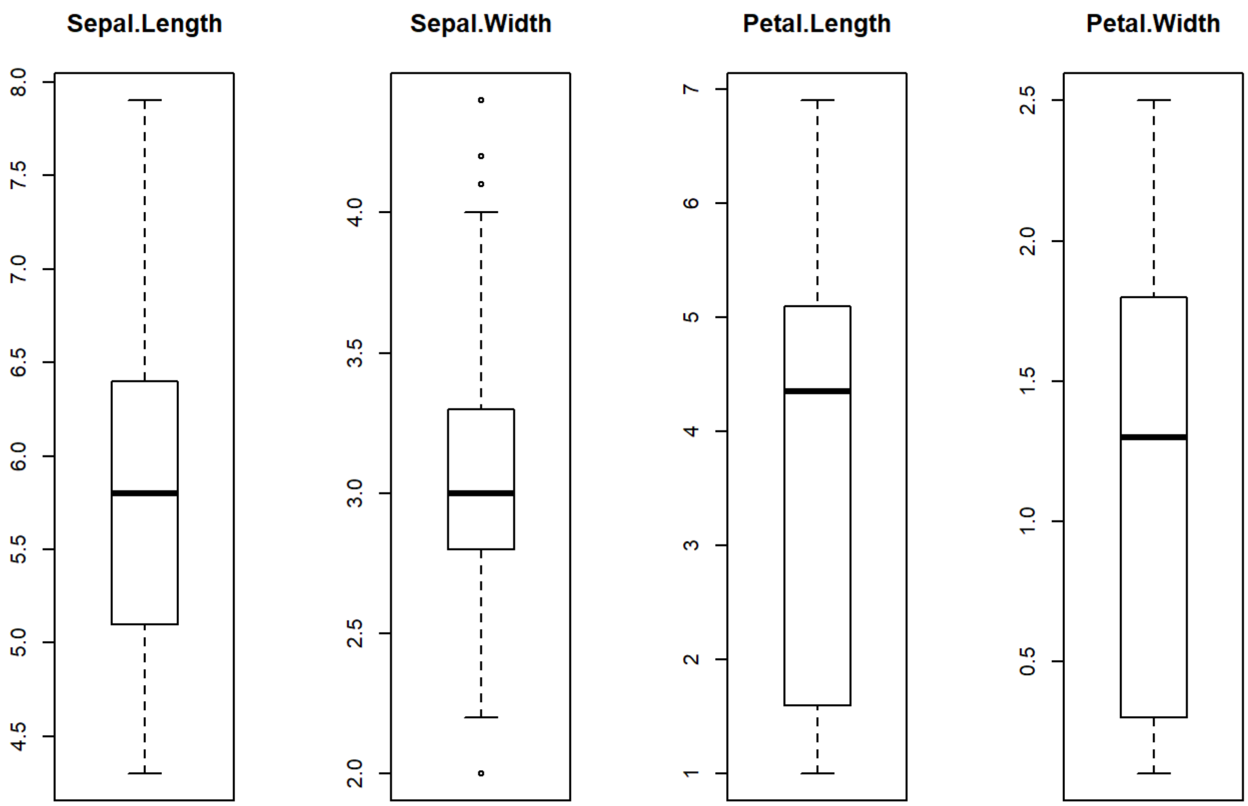
```
head(iris)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

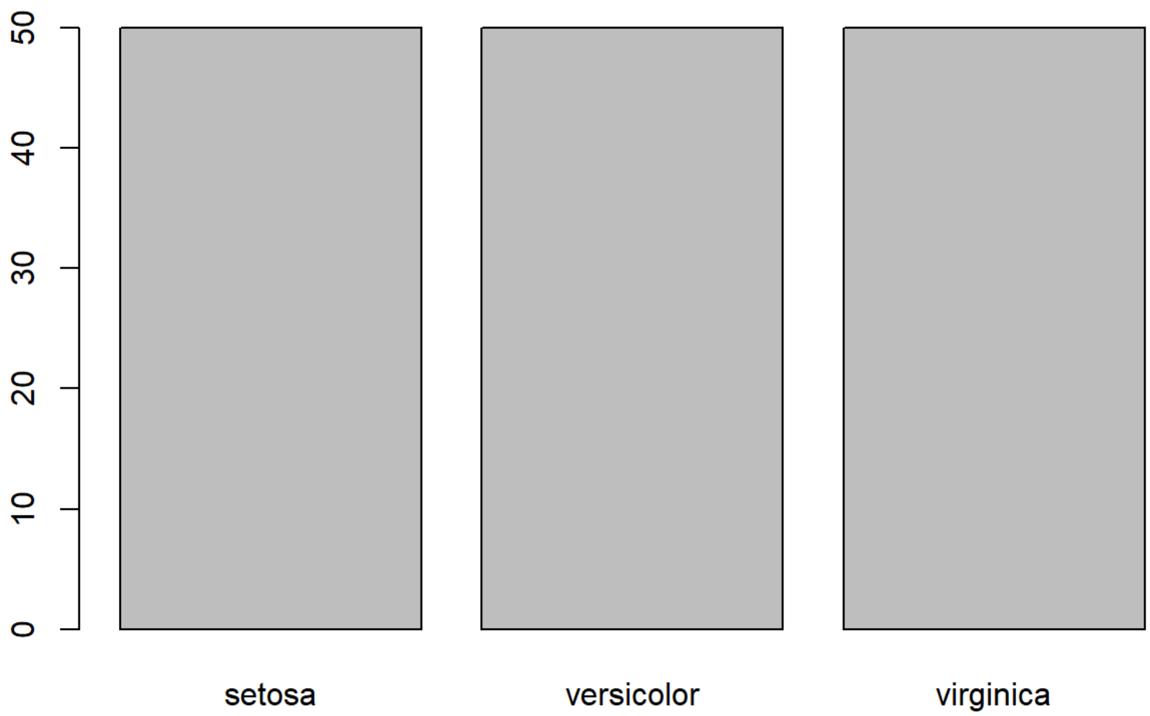
```
summary(iris)
```

```
##   Sepal.Length    Sepal.Width     Petal.Length    Petal.Width   
##  Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100  
##  1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300  
##  Median :5.800   Median :3.000   Median :4.350   Median :1.300  
##  Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199  
##  3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800  
##  Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500  
##        Species  
##  setosa    :50  
##  versicolor:50  
##  virginica :50  
##                 
##                 
##                 
```

```
input.val<-iris[,1:4]
output.val<-iris[,5]
par(mfrow=c(1,4))
for(i in 1:4) {
  boxplot(input.val[,i],main=names(iris)[i])    }
```

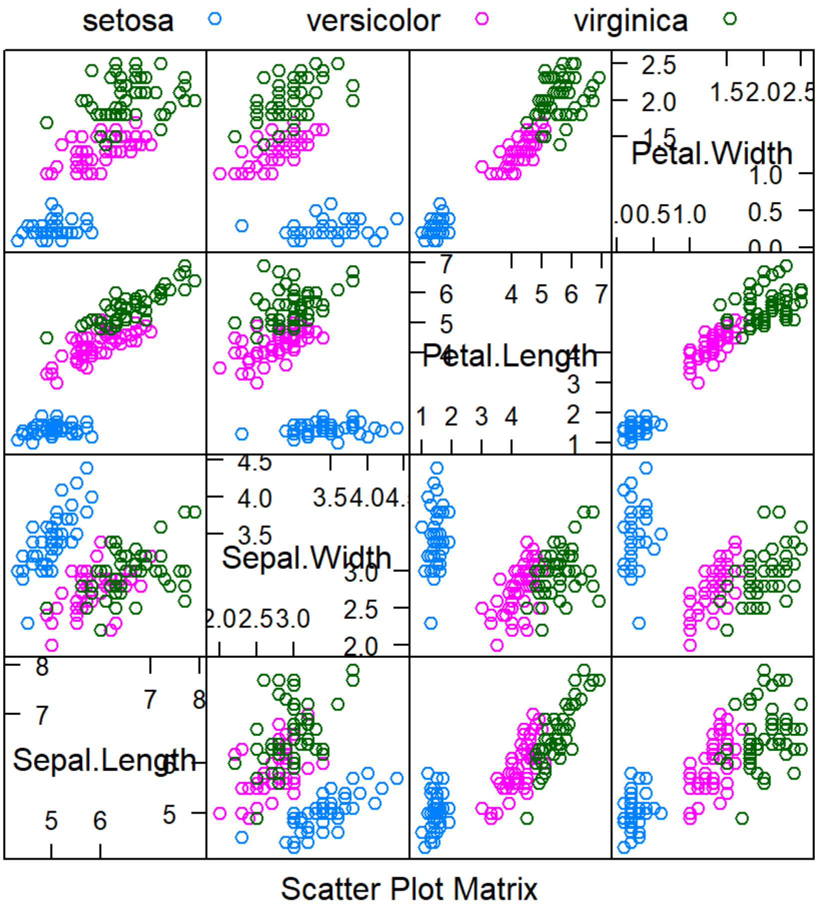



```
par(mfrow=c(1,1))
barplot(table(output.val))
```

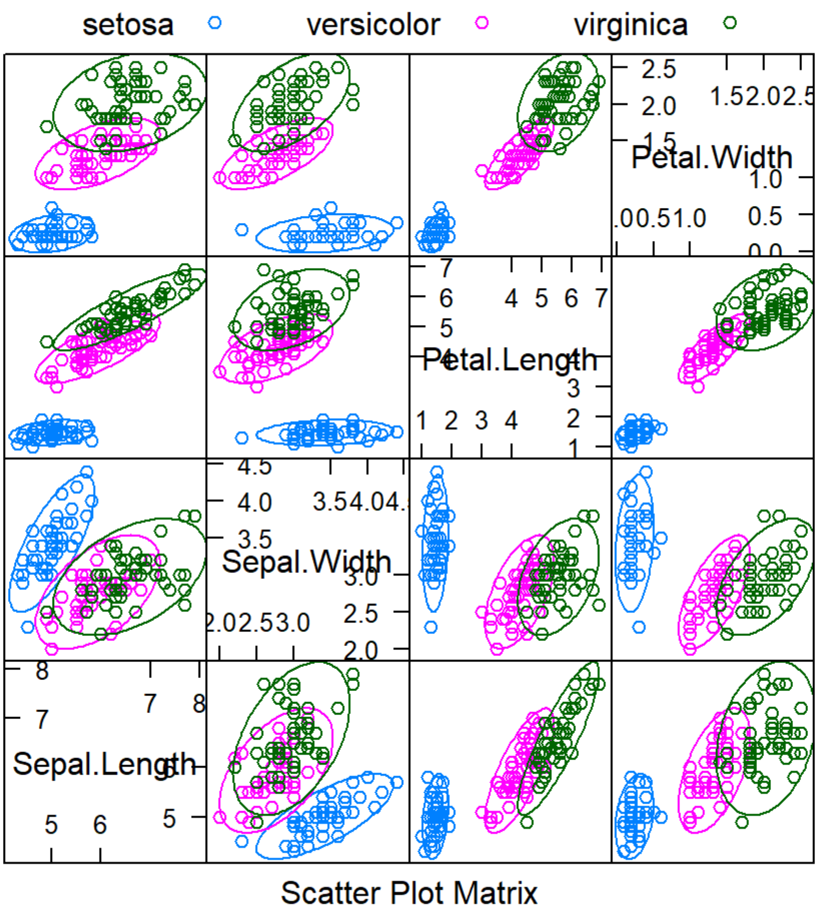


2. Multivarite plotting

```
#Relation of length and width of sepal and petal for three species
featurePlot(x=input.val, y=output.val, plot="pairs", auto.key=list(columns=3))
```

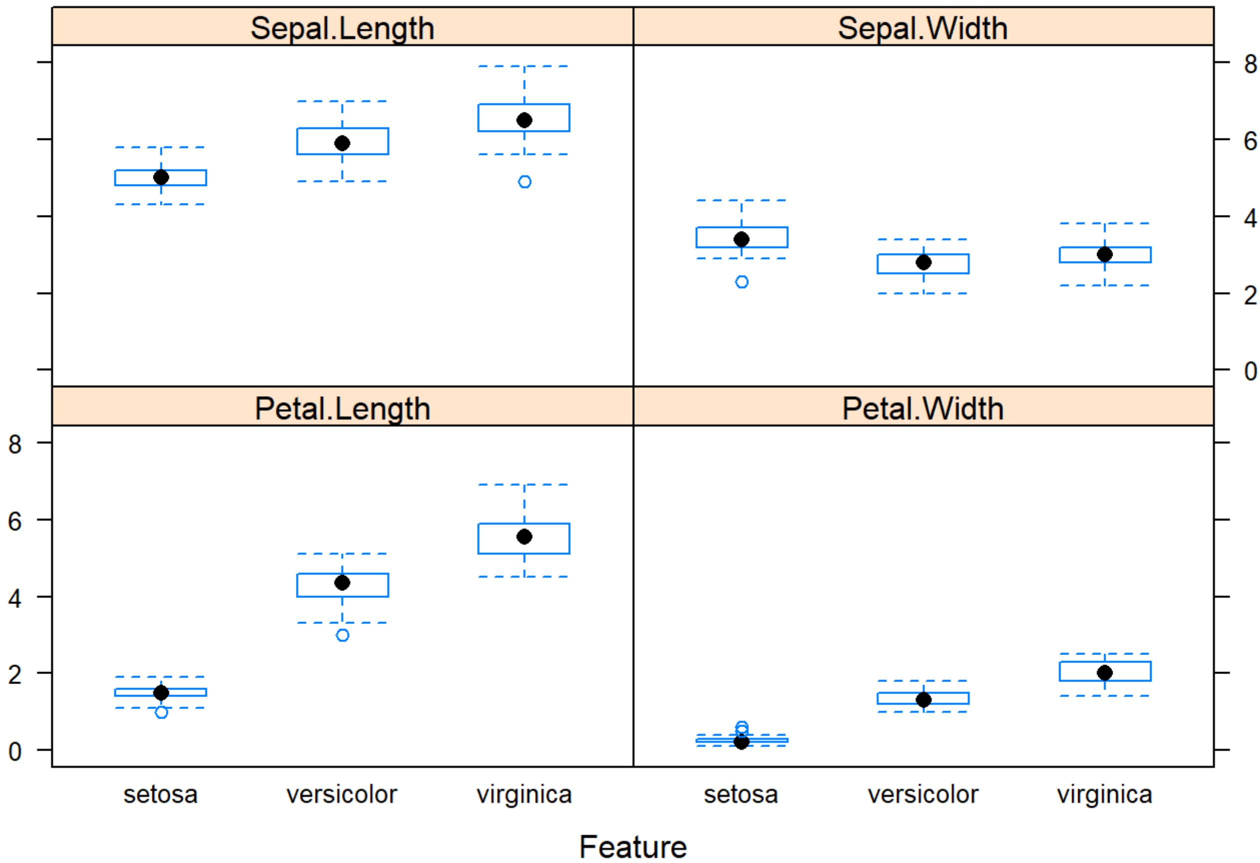



```
featurePlot(x=input.val,y=output.val,plot = "ellipse", auto.key=list(columns=3))
```



Scatter Plot Matrix

```
#Range of length and width of sepal and petal for three species
featurePlot(x=input.val,y=output.val,plot = "box")
```



3. Data partition

```
set.seed(123)
# training-80%
validation.index<-createDataPartition(iris$Species,p=0.8,list=FALSE)
training.data<-iris[validation.index,]
# testing- 20%
testing.data<-iris[-validation.index,]
```

4.1 Cross-validation

```
control<-trainControl(method="cv", number=10)
metric<-"Accuracy"
```

4.2 Model formulation

```
#linear model
set.seed(123)
lda.model<-train(Species~.,data=training.data,method="lda",metric=metric, trControl=control)
```

```
## Loading required package: MASS
```

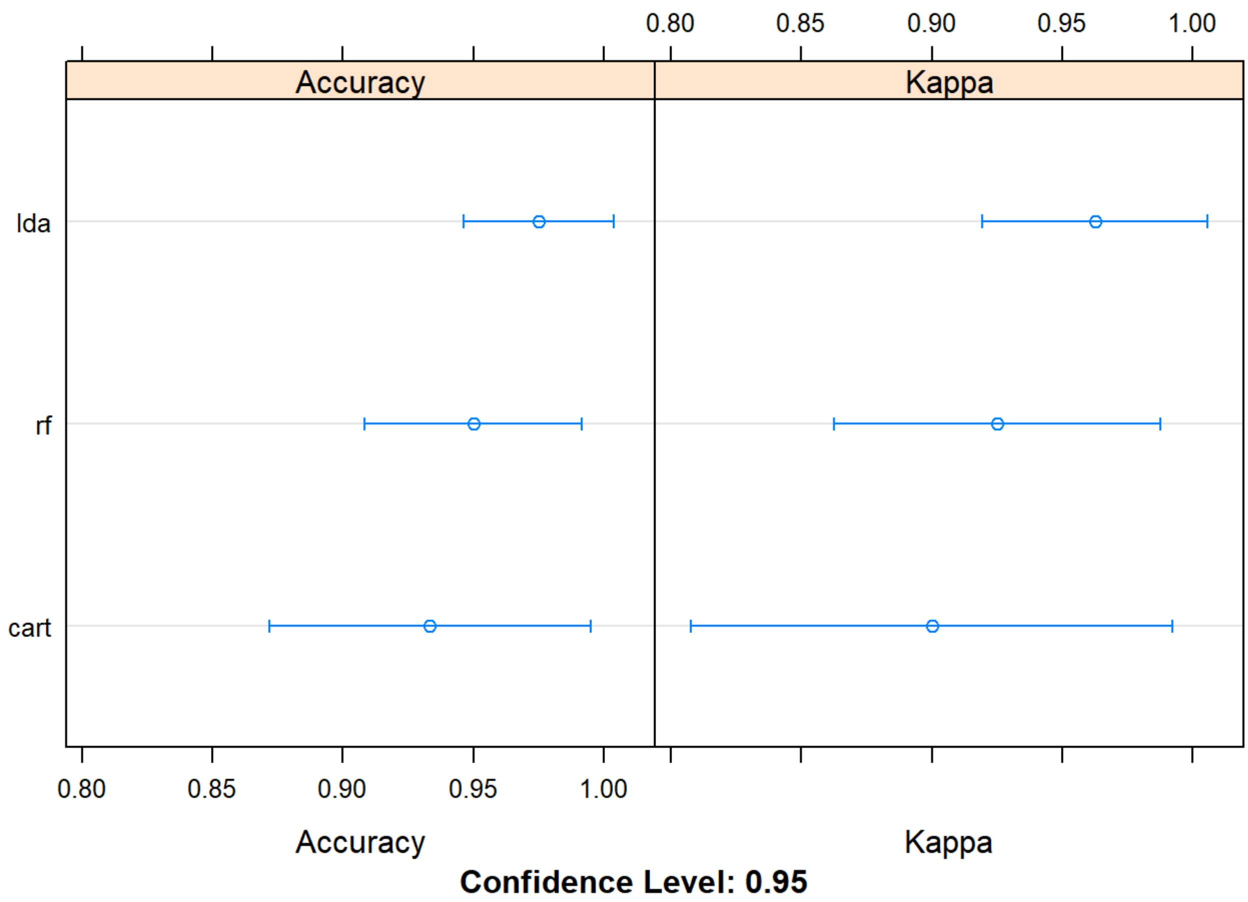
```
#decision tree
set.seed(123)
cart.model<-train(Species~.,data=training.data,method="rpart",metric=metric, trControl=control)

#Random Forest
set.seed(123)
rf.model<-train(Species~.,data=training.data,method="rf", metric=metric, trControl=control)
```

5. Performance assessing

```
result.model<-resamples(list(lda=lda.model,cart=cart.model,rf=rf.model))
summary(result.model)
```

```
##
## Call:
## summary.resamples(object = result.model)
##
## Models: lda, cart, rf
## Number of resamples: 10
##
## Accuracy
##        Min. 1st Qu. Median   Mean 3rd Qu. Max. NA's
## lda  0.9167  0.9375 1.0000 0.9750       1    1    0
## cart 0.7500  0.9167 0.9583 0.9333       1    1    0
## rf   0.8333  0.9167 0.9583 0.9500       1    1    0
##
## Kappa
##       Min. 1st Qu. Median   Mean 3rd Qu. Max. NA's
## lda  0.875  0.9062 1.0000 0.9625       1    1    0
## cart 0.625  0.8750 0.9375 0.9000       1    1    0
## rf   0.750  0.8750 0.9375 0.9250       1    1    0
```

```
#Visualize
dotplot(result.model)
```



6. Prediction

```
pred.result<-predict(lda.model,testing.data)
confusionMatrix(pred.result,testing.data$Species)
```

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction   setosa versicolor virginica
##   setosa         10          0         0
##   versicolor      0         10         0
##   virginica       0          0        10
##
## Overall Statistics
##
##                Accuracy : 1
##                  95% CI : (0.8843, 1)
##     No Information Rate : 0.3333
##     P-Value [Acc > NIR] : 4.857e-15
##
##                   Kappa : 1
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: setosa Class: versicolor Class: virginica
## Sensitivity                 1.0000            1.0000           1.0000
## Specificity                 1.0000            1.0000           1.0000
## Pos Pred Value              1.0000            1.0000           1.0000
## Neg Pred Value              1.0000            1.0000           1.0000
## Prevalence                  0.3333            0.3333           0.3333
## Detection Rate              0.3333            0.3333           0.3333
## Detection Prevalence        0.3333            0.3333           0.3333
## Balanced Accuracy           1.0000            1.0000           1.0000
```

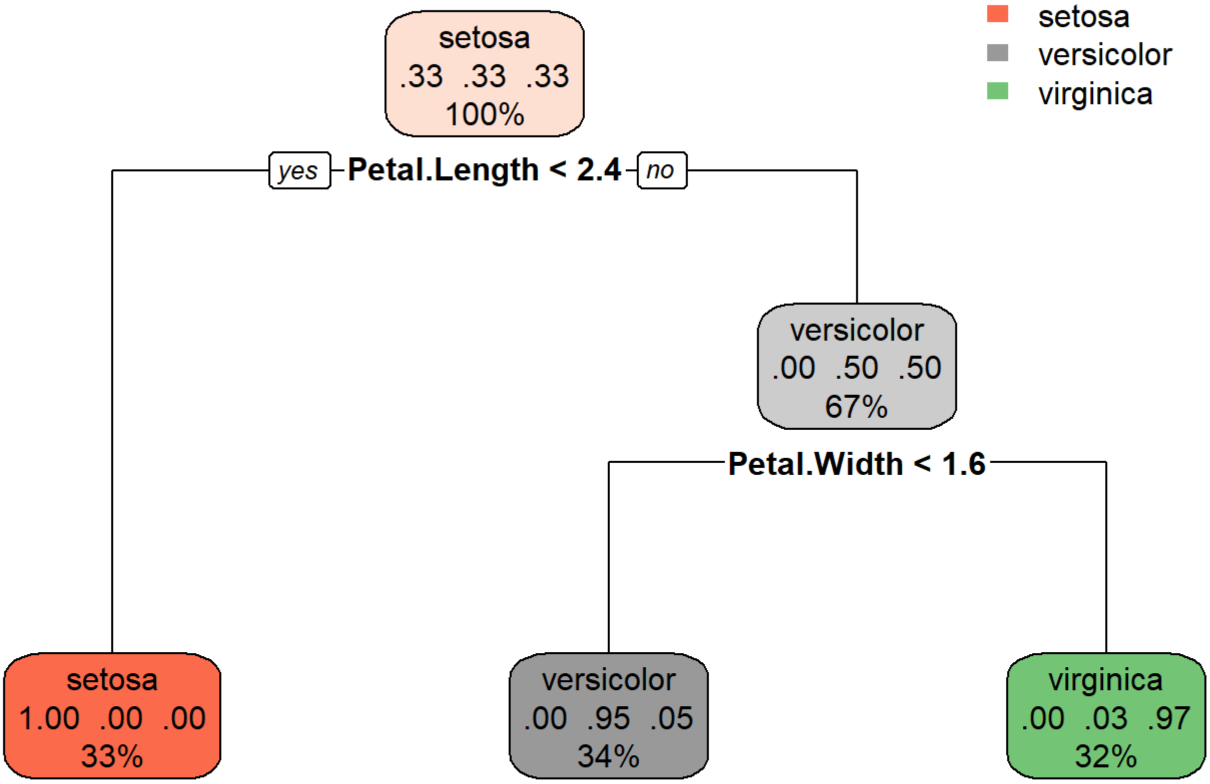
Analysis: Iris dataset contains the variables of width and length of petal and sepal for three species (50 samples for each). As is showed in the boxplots, the petal length varies significantly in three species, while the sepal width of them is relatively similar. Virginica is the biggest in size generally, and setosa is particularly small in width. For Virginica and versicolor, a long sepal(petal) usually comes with a wide petal(sepal). This positive correlation is not obvious for setosa. Also, setosa can be more easily distinguished from the other two apecies, for its unique petal and sepal size.

Comparing the three models from 10-fold validation, linear model gives the most accurate classification (0.97 on average), while decision tree has the poorest performace(0.93 on average). In terms of kappa statistic, which is a more robust metric that takes random chance in to consideration, linear model still gives best performance. Therefore linear model is selected for prediction on testing data.

From the results of confusion matrx, it seems that linear model would give perfect prediction with 100% accuracy, which might be too optimistic in reality. Sample collection and sample size could be a reason, which requires further sampling and analysis.

Comparison of three algorithm: Linear: The most common and simple regression model. Linear relationship assumption. Easy to interpret.

Decision tree: Classification tree model is usually obtained by recursive binary partition, with node impurity as the split criterion. Although it may not give accurate predicition compared with advanced machine learning algorithm, decision tree can help visualize the process and easy to interpret. Using iris as an example:



We can see the split criteria clearly from the chart. However, overfitting is a fundamental problem so appropriate pruning methods should be used, which often requires domain expertise and can be really challenging.

Random forest: More advanced as it includes a large number of classification trees and selects the mode as most trees give (mean? for regression tree), and therefore the prediction is normally better than a single tree and can avoid the overfitting problem. Less interpretable than decision tree (ensemble method), and generally less accurate than other algorithms(as trees).