Machine Learning_Iris_caret~Package

Mohamed Nachid

boussiala.nachid@univ-alger3.dz

```
#install.packages("caret")
#install.packages("kernlab")
#install.packages("randomForest")
#install.packages("ellipse")

library(caret)
library(kernlab)
library(randomForest)
library(ellipse)
```

Load The Data

attach the iris dataset to the environment

```
data(iris)
colnames(iris)
```

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

```
colnames(iris) <- c("Sepal_Length","Sepal_Width","Petal_Length","Petal_Width","Species")
colnames(iris)
```

```
## [1] "Sepal_Length" "Sepal_Width"  "Petal_Length" "Petal_Width"  "Species"
```

create a list of 80% of the rows in the original dataset we can use for training

```
index <- createDataPartition(iris$Species, p=0.80, list=FALSE)
```

select 20% of the data for Test

```
test <- iris[-index,]
dim(test)
```

```
## [1] 30  5
```

use the remaining 80% of data to training

```
train <- iris[index,]
dim(train)
```

```
## [1] 120   5
```

list types for each attribute

```
sapply(train, class)
```

```
## Sepal_Length  Sepal_Width Petal_Length  Petal_Width      Species
##    "numeric"    "numeric"    "numeric"    "numeric"     "factor"
```

take a peek at the first 6 rows of the data

```
head(train)
```

```
##   Sepal_Length Sepal_Width Petal_Length Petal_Width Species
## 1          5.1         3.5          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
## 7          4.6         3.4          1.4         0.3  setosa
```

list the levels for the class

```
levels(train$Species)
```

```
## [1] "setosa"     "versicolor" "virginica"
```

summarize the class distribution

```r
percentage <- prop.table(table(train$Species)) * 100
cbind(freq=table(train$Species), percentage=percentage)
```

```
##            freq percentage
## setosa       40   33.33333
## versicolor   40   33.33333
## virginica    40   33.33333
```

summarize attribute distributions

```r
summary(train)
```

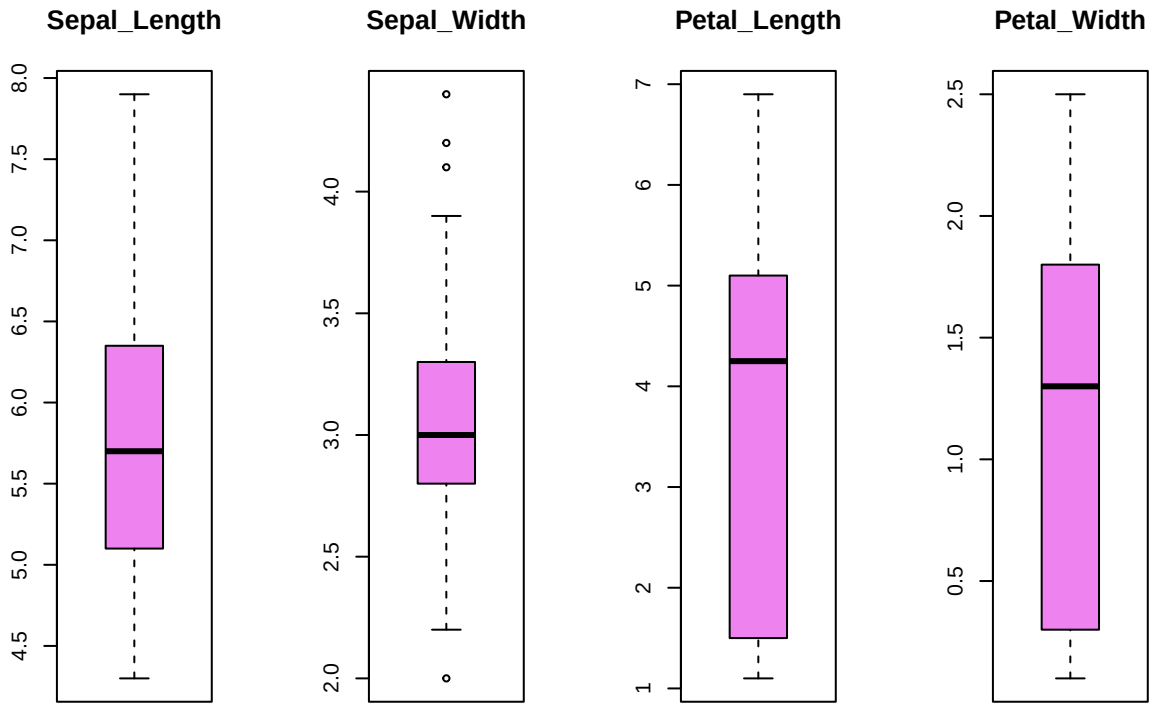
```
##   Sepal_Length    Sepal_Width     Petal_Length    Petal_Width
##  Min.   :4.300   Min.   :2.000   Min.   :1.100   Min.   :0.100
##  1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.500   1st Qu.:0.300
##  Median :5.700   Median :3.000   Median :4.250   Median :1.300
##  Mean   :5.805   Mean   :3.055   Mean   :3.731   Mean   :1.192
##  3rd Qu.:6.325   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    :40
##  versicolor:40
##  virginica :40
##
##
##
```

split input and output

```r
x <- train[,1:4]
y <- train[,5]
```

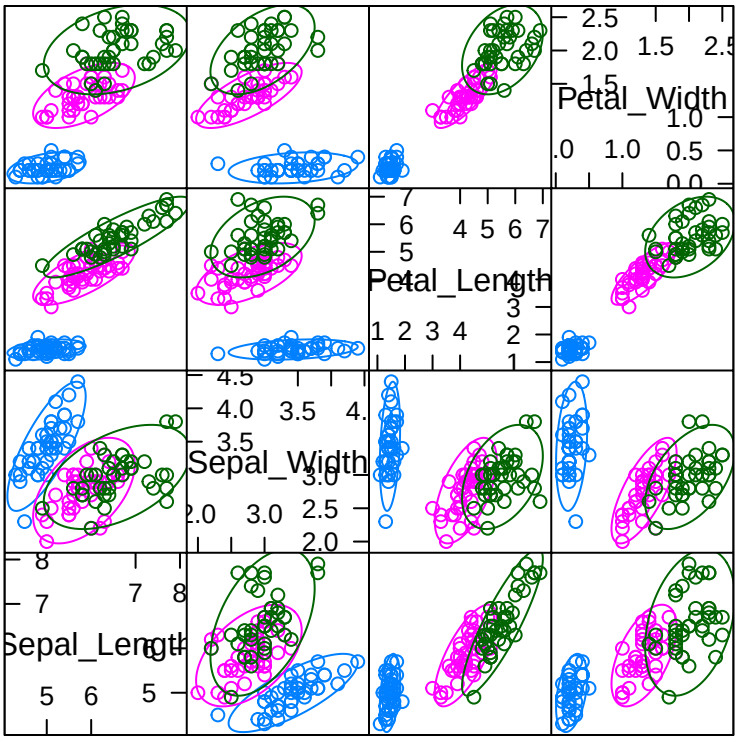
boxplot for each attribute on one image

```r
par(mfrow=c(1,4))
for(i in 1:4) {
  boxplot(x[,i], main=names(iris)[i], col='violet')}
```



scatterplot matrix

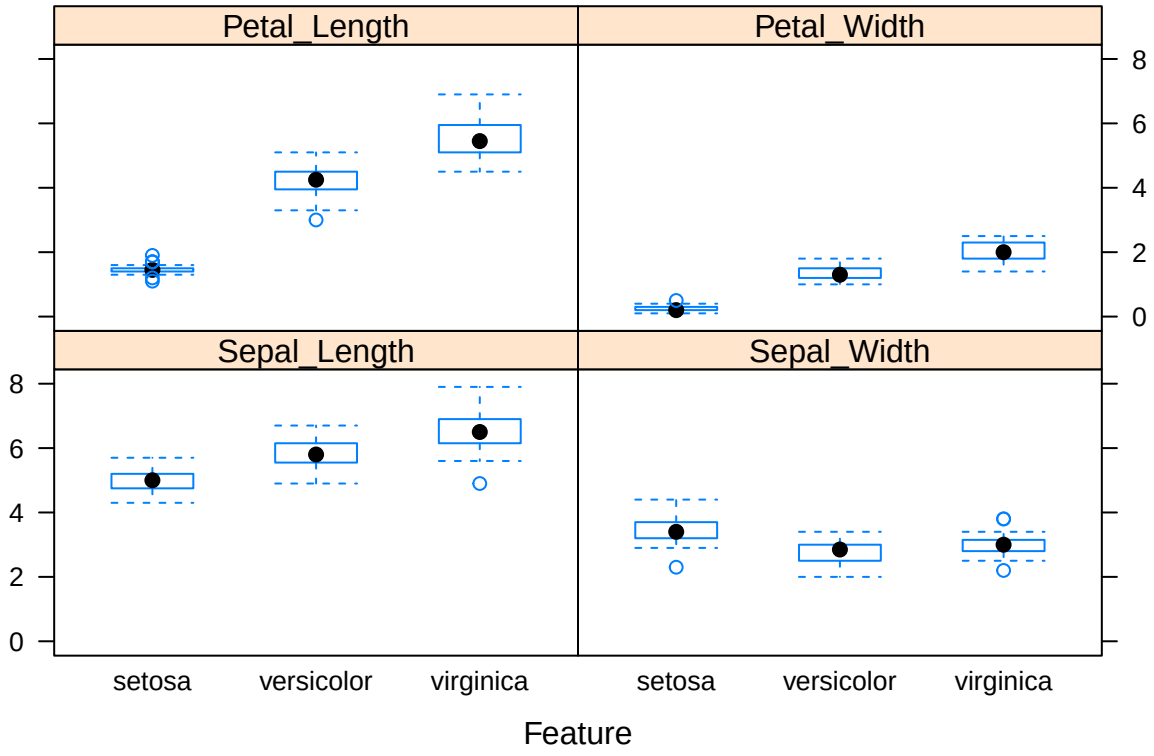
```
featurePlot(x, y, 'ellipse')
```



Matrice de nuages de points

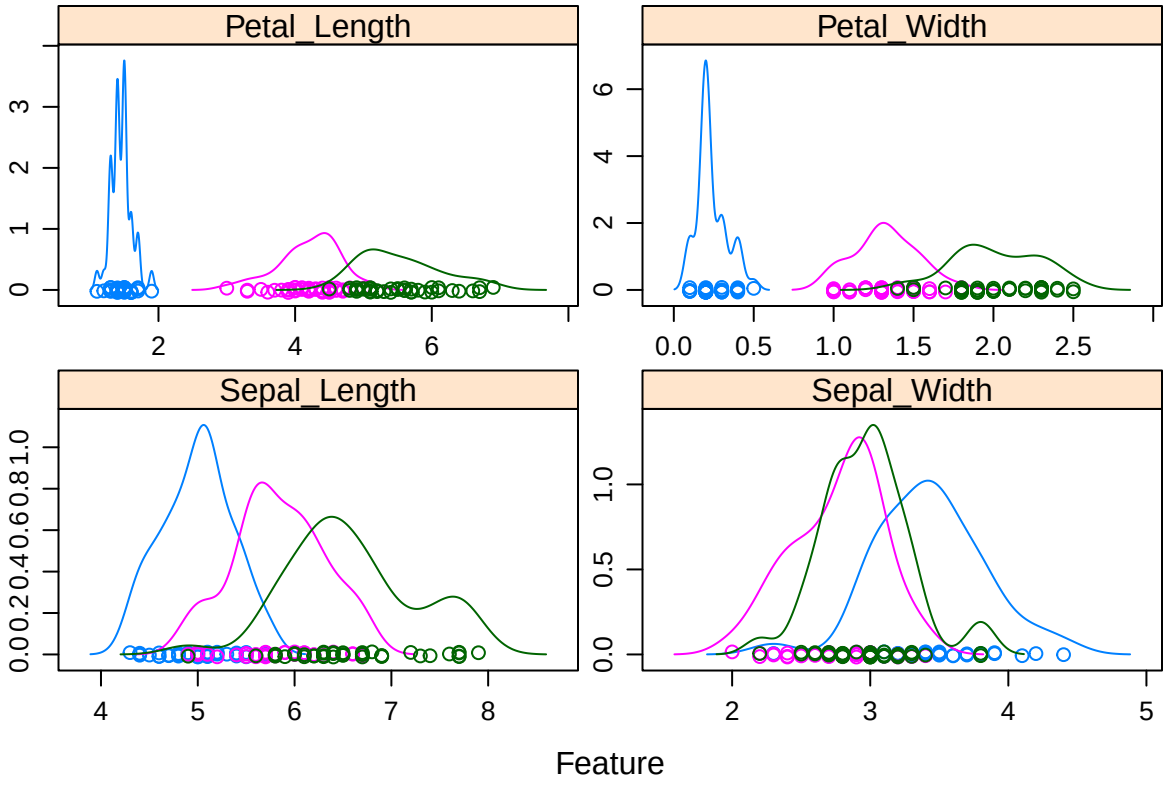
box and whisker plots for each attribute

```
featurePlot(x, y, "box")
```



density plots for each attribute by class value

```
scales <- list(x=list(relation="free"), y=list(relation="free"))
featurePlot(x, y, "density", scales=scales)
```



Run algorithms using 10-fold cross validation

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

Build Models

a) linear algorithms

```
set.seed(7)
fit_lda <- train(Species~., data=train, method="lda", metric=metric, trControl=control)
```

b) nonlinear algorithms

RPART

```
set.seed(7)
fit_cart <- train(Species~., data=train, method="rpart", metric=metric, trControl=control)
```

kNN

```
set.seed(7)
fit_knn <- train(Species~., data=train, method="knn", metric=metric, trControl=control)
```

c) advanced algorithms

SVM

```
set.seed(7)
fit_svm <- train(Species~., data=train, method="svmRadial", metric=metric, trControl=control)
```

Random Forest

```
set.seed(7)
fit_rf <- train(Species~., data=train, method="rf", metric=metric, trControl=control)
```

Select Best Model

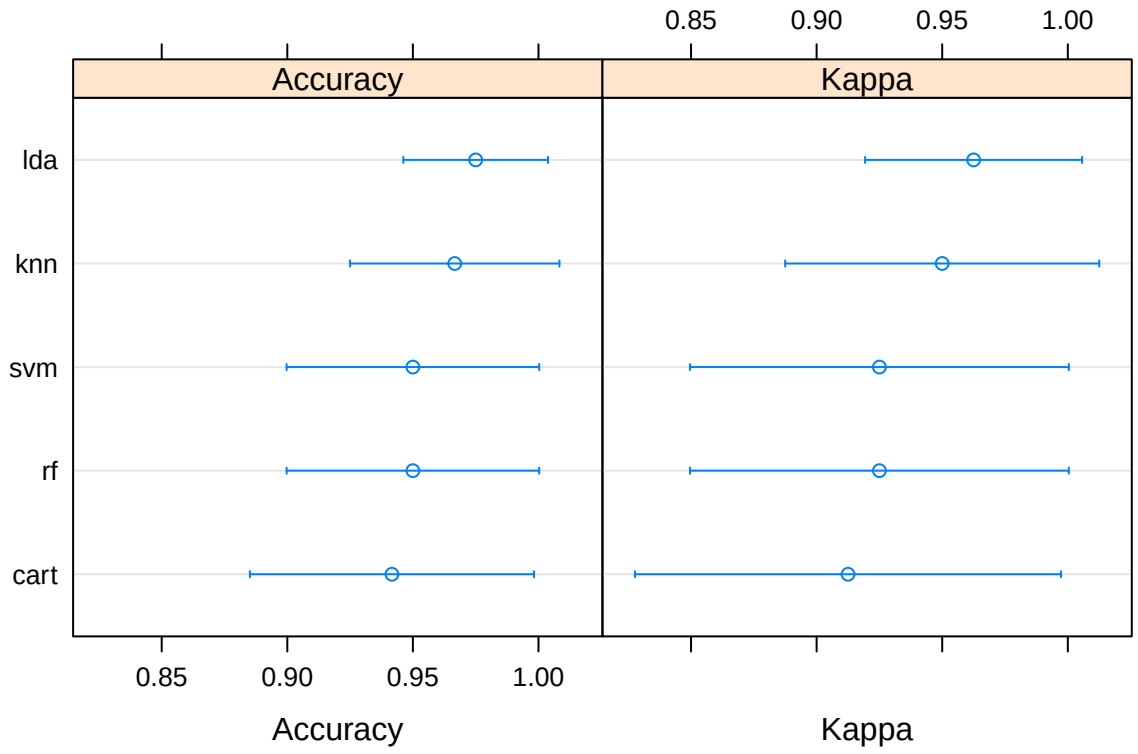
summarize accuracy of models

```
results <- resamples(list(lda=fit_lda, cart=fit_cart, knn=fit_knn, svm=fit_svm, rf=fit_rf))
summary(results)
```

```
##
## Call:
## summary.resamples(object = results)
##
## Models: lda, cart, knn, svm, rf
## Number of resamples: 10
##
## Accuracy
##           Min.   1st Qu. Median      Mean 3rd Qu. Max. NA's
## lda  0.9166667 0.9375000      1 0.9750000       1    1    0
## cart 0.8333333 0.8541667      1 0.9416667       1    1    0
## knn  0.8333333 0.9375000      1 0.9666667       1    1    0
## svm  0.8333333 0.9166667      1 0.9500000       1    1    0
## rf   0.8333333 0.9166667      1 0.9500000       1    1    0
##
## Kappa
##       Min. 1st Qu. Median   Mean 3rd Qu. Max. NA's
## lda  0.875 0.90625      1 0.9625       1    1    0
## cart 0.750 0.78125      1 0.9125       1    1    0
## knn  0.750 0.90625      1 0.9500       1    1    0
## svm  0.750 0.87500      1 0.9250       1    1    0
## rf   0.750 0.87500      1 0.9250       1    1    0
```

compare accuracy of models

```
dotplot(results)
```




```
lda_accuracy <- fit_lda$results$Accuracy
cart_accuracy <- fit_cart$results$Accuracy
knn_accuracy <- fit_lda$results$Accuracy
svm_accuracy <- fit_lda$results$Accuracy
rf_accuracy <- fit_lda$results$Accuracy

knitr::kable(cbind(lda_accuracy, cart_accuracy, knn_accuracy,
                   svm_accuracy, rf_accuracy))
```

lda_accuracy	cart_accuracy	knn_accuracy	svm_accuracy	rf_accuracy
0.975	0.9416667	0.975	0.975	0.975
0.975	0.7416667	0.975	0.975	0.975
0.975	0.3333333	0.975	0.975	0.975

summarize Best Model

```r
print(fit_lda)
```

```
## Linear Discriminant Analysis
##
## 120 samples
##   4 predictor
##   3 classes: 'setosa', 'versicolor', 'virginica'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 108, 108, 108, 108, 108, 108, ...
## Resampling results:
##
##   Accuracy  Kappa
##   0.975     0.9625
```

estimate skill of LDA on the validation dataset

```r
predictions <- predict(fit_lda, test)
confusionMatrix(predictions, test$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
```