**1) Iris Flowers Classification ML Project :**

**INTRODUCTION**

This is my first project using R por machine learning. The project makes use of the Iris database provided with R package, and it consistsof a multivariate classification project. Let's get to it!

**LOAD AND SPLIT DATA**

**code**

> library(caret)

> library(ggplot2)

> data(iris)

> dataset <- iris

> function(createDataPartition)

+ createDataPartition(dataset$Species, p=0.80, list = FALSE)

function(createDataPartition)

createDataPartition(dataset$Species, p=0.80, list = FALSE)

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

> validation\_index <- createDataPartition(dataset$Species, p=0.80, list = FALSE)

> # select 20% of the data for validation

> validation <- dataset[-validation\_index, ]

> # use the remaining 80% of data to training and testing the models

> dataset <- dataset[validation\_index, ]

**SUMMARIZE DATASET**

> dim(dataset)

**[1] 120 5**

> #list type of each attribute

> sapply(dataset, class)

**Sepal.Length Sepal.Width Petal.Length Petal.Width Species**

**"numeric" "numeric" "numeric" "numeric" "factor"**

> # take a peek at the first 5 rows of the data

> head(dataset)

**Sepal.Length Sepal.Width Petal.Length Petal.Width Species**

**2 4.9 3.0 1.4 0.2 setosa**

**3 4.7 3.2 1.3 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**

**8 5.0 3.4 1.5 0.2 setosa**

> #list the level for the class

> levels(dataset$Species)

**[1] "setosa" "versicolor" "virginica"**

> #summarize the class distribution

> percentage <- prop.table(table(dataset$Species))\*10

> cbind(freq = table(dataset$Species), percentage=percentage)

**freq percentage**

**setosa 40 3.333333**

**versicolor 40 3.333333**

**virginica 40 3.333333**

> summary(dataset)

**Sepal.Length Sepal.Width Petal.Length Petal.Width Species**

**Min. :4.300 Min. :2.000 Min. :1.100 Min. :0.100 setosa :40**

**1st Qu.:5.100 1st Qu.:2.800 1st Qu.:1.600 1st Qu.:0.300 versicolor:40**

**Median :5.800 Median :3.000 Median :4.350 Median :1.350 virginica :40**

**Mean :5.854 Mean :3.056 Mean :3.764 Mean :1.201**

**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**

**VISUALIZE DATASET**

> #split input and output

> x <- dataset[ , 1:4]

> y <- dataset[ , 5]

**UNIVARIATE PLOTS**

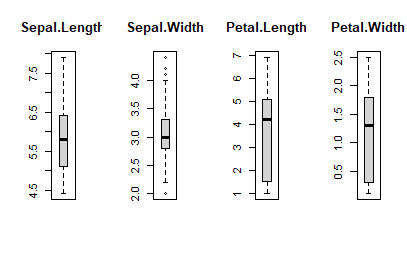
> #boxplot for each attribute on new image

> par(mfrow=c(1,4)) # show plots in 1 row and 4 columns

> for (i in 1:4) {

+ boxplot(x[, i], main=names(dataset)[i])

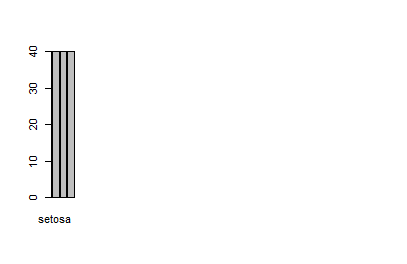
+ }



> # barplot for class breakdown

> # par(mfrow=c(1,1)) # show plots in 1 row and 4 columns

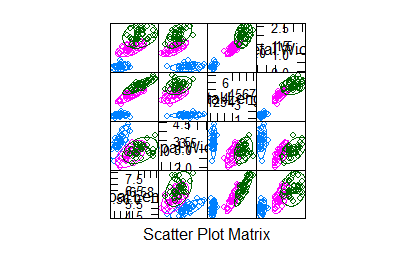
> plot(y)



**MULTIVARIATE PLOTS**

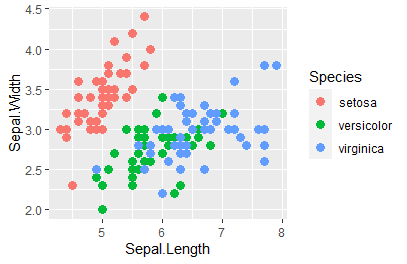
> # scatterplot matrix

> featurePlot(x=x, y=y, plot="ellipse")



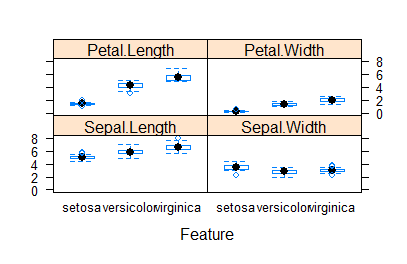
> # Zoom on one of the graphs

> ggplot(data=iris, aes(x=Sepal.Length, y=Sepal.Width, color=Species))+geom\_point(size=3)



> # box and whisker plots for each attribute

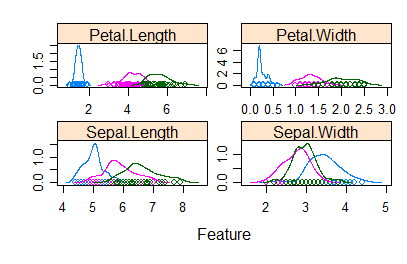
> featurePlot(x=x, y=y, plot="box")



> # density plots for each attribute by class value

> scales <- list(x=list(relation="free"), y=list(relation="free"))

> featurePlot(x=x, y=y, plot="density", scales=scales)



**TEST HARESS-10-FOLD CROSS VALIDATION**

> # Run algorithms using 10-fold cross validation

> control <- trainControl(method="cv", number=18)

> metric <- "Accuracy" #correctly predicted/total number of instances

**BUILD MODELS**

**# linear algorithms**

set.seed(7)

fit.lda <- train(Species~., data = dataset, method = "lda", metric = metric, trControl = control)

**nonlinear algorithms**

> set.seed(7)

> fit.cart <- train(Species~., data=dataset, method="rpart", metric=metric, trControl=control)

**#KNN**

> set.seed(7)

> fit.knn <- train(Species~., data=dataset, method="knn", metric=metric, trControl=control)

**ADVANCED ALGORITHM**

#SVM

> set.seed(7)

> fit.svm <- train(Species~., data=dataset, method="svmRadial", metric=metric, trControl=control)

**#RANDOM FOREST**

> set.seed(7)

> fit.rf <- train(Species~., data = dataset, method="rf", metric = metric, trControl = control)

**# 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: 18

Accuracy

Min. 1st Qu. Median Mean 3rd Qu. Max. NA's

lda 0.8333333 1 1 0.9828042 1 1 0

cart 0.8333333 1 1 0.9735450 1 1 0

knn 0.6666667 1 1 0.9735450 1 1 0

svm 0.8333333 1 1 0.9656085 1 1 0b

rf 0.8333333 1 1 0.9656085 1 1 0

Kappa

Min. 1st Qu. Median Mean 3rd Qu. Max. NA's

lda 0.75 1 1 0.9739583 1 1 0

cart 0.75 1 1 0.9604377 1 1 0

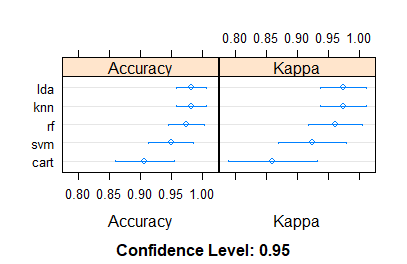
knn 0.50 1 1 0.9600694 1 1 0

svm 0.75 1 1 0.9482849 1 1 0

rf 0.75 1 1 0.9482849 1 1 0

> #comapre Accuracy of Models

> dotplot(results)



> #summarize Best model

> print(fit.lda)

Linear Discriminant Analysis

120 samples

4 predictor

3 classes: 'setosa', 'versicolor', 'virginica'

No pre-processing

Resampling: Cross-Validated (18 fold)

Summary of sample sizes: 114, 113, 113, 112, 113, 114, ...

Resampling results:

Accuracy Kappa

0.9828042 0.9739583

**MAKE PREDICTION**

> # estimate skill of LDA on the validation dataset

> predictions <- predict(fit.lda, validation)

> confusionMatrix(predictions, validation$Species)

Confusion Matrix and Statistics

Reference

Prediction setosa versicolor virginica

setosa 10 0 0

versicolor 0 9 0

virginica 0 1 10

Overall Statistics

Accuracy : 0.9667

95% CI : (0.8278, 0.9992)

No Information Rate : 0.3333

P-Value [Acc > NIR] : 2.963e-13

Kappa : 0.95

Mcnemar's Test P-Value : NA

Statistics by Class:

Class: setosa Class: versicolor Class: virginica

Sensitivity 1.0000 0.9000 1.0000

Specificity 1.0000 1.0000 0.9500

Pos Pred Value 1.0000 1.0000 0.9091

Neg Pred Value 1.0000 0.9524 1.0000

Prevalence 0.3333 0.3333 0.3333

Detection Rate 0.3333 0.3000 0.3333

Detection Prevalence 0.3333 0.3000 0.3667

Balanced Accuracy 1.0000 0.9500 0.9750