***Machine Learning Project in R***

Install the packages we are going to use today. Packages are third party add-ons or libraries that we can use in R.

install.packages("caret")

You can install the caret packages and all packages that you might need by typing:

install.packages("caret", dependencies=c("Depends", "Suggests"))

Load the package:

library(caret)

The caret package provides a consistent interface into hundreds of machine learning algorithms.

***Iris flowers dataset***

The dataset contains 150 observations of iris flowers. There are four columns of measurements of the flowers in centimeters. The fifth column is the species of the flower observed. All observed flowers belong to one of three species.

Load the iris data the easy way:

# attach the iris dataset to the environment

data(iris)

# rename the dataset

dataset <- iris

***Characteristics of the Dataset***

# dimensions of dataset

dim(dataset)

# list types for each attribute

sapply(dataset, class)

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

head(dataset)

# list the levels for the class

levels(dataset$Species)

# summarize the class distribution

percentage <- prop.table(table(dataset$Species)) \* 100

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

# summarize attribute distributions

summary(dataset)

***Visualize***

Two types of plots:

1. Univariate plots to better understand each attribute:

# boxplot for each attribute on one image

par(mfrow=c(1,4))

for(i in 1:4) {

boxplot(x[,i], main=names(iris)[i])

}  
  
  
# barplot for class breakdown

plot(y)

1. Multivariate plots to better understand the relationships between attributes:

# scatterplot matrix

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

# 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)

***Algorithms***

5 different algorithms:

* Linear Discriminant Analysis (LDA)
* Classification and Regression Trees (CART).
* k-Nearest Neighbors (kNN).
* Support Vector Machines (SVM) with a linear kernel.
* Random Forest (RF)

# a) linear algorithms

set.seed(7)

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

# b) nonlinear algorithms

# CART

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)

# c) advanced algorithms

# 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)

***Select the 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)

Create a plot of the model evaluation results and compare the spread and the mean accuracy of each model. There is a population of accuracy measures for each algorithm because each algorithm was evaluated 10 times.

# compare accuracy of models

dotplot(results)