Prooject #1 (Iris dataset)

Ismael Isak

03/15/2022

Introduction

This first project is going to be going through the basics of loading a dataset, validation, summarizing it, visualizing it, and finally building predictive models.

I will be using plain language and explaining each step as I go along. I won't be including how to install R or R Studio but this project should be enough for the basics of machine learning.

Loading the data

First I will load the data from the dataset package in R. I will be using the Iris dataset, which is a well known dataset created by Edgar Anderson (1897-1969). He was an American botanist who revolutionized is field by introducing botanical genetics in hihs 1941 book Introgressive Hybridization. His work on the Iris species along with statistician R.A. Fisher helped develop examples of statistical classification which is an important part of machine learning.

```
data("iris")
dataset <- iris
```

Validation and training

```
# 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,]

dim(dataset)
```

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

Exploring and summarizing the data

```r
#list the classes in dataset
sapply(dataset, class)
```

```
## Sepal.Length  Sepal.Width Petal.Length  Petal.Width      Species
##    "numeric"    "numeric"    "numeric"    "numeric"     "factor"
```

```r
#looking through the dataset
head(dataset)
```

```
##    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
## 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
## 10          4.9         3.1          1.5         0.1  setosa
```

```r
#Going through the levels of dataset classes
levels(dataset$Species)
```

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

```r
#Summarize the class distribution
percentage <- prop.table(table(dataset$Species))*100
cbind(freq=table(dataset$Species), percentage=percentage)
```

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

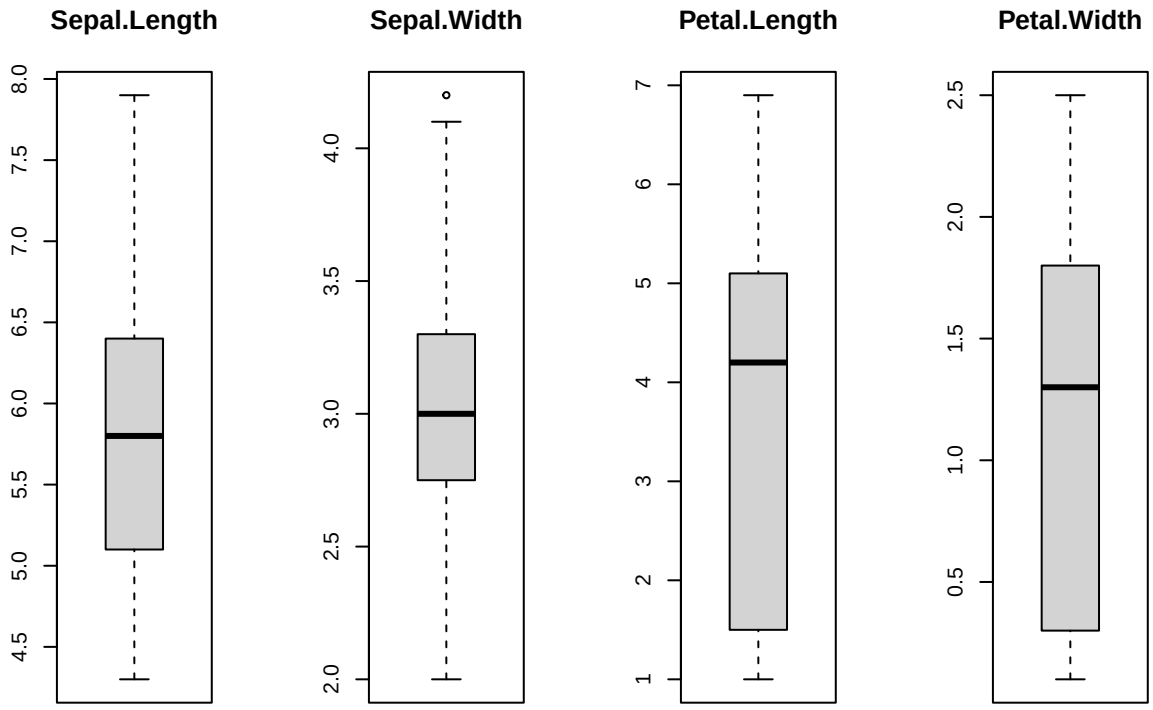
```r
#Summarize the attributes of the distributions
summary(dataset)
```

```
##   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.:2.775   1st Qu.:1.500   1st Qu.:0.300
##  Median :5.800   Median :3.000   Median :4.200   Median :1.300
##  Mean   :5.846   Mean   :3.049   Mean   :3.765   Mean   :1.192
##  3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
##  Max.   :7.900   Max.   :4.200   Max.   :6.900   Max.   :2.500
##        Species
##  setosa    :40
##  versicolor:40
##  virginica :40
##
##
##
```

Visualizing the data

```
#Univariate plot analysis using boxplots
x <- dataset[,1:4]
y <- dataset[,5]

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

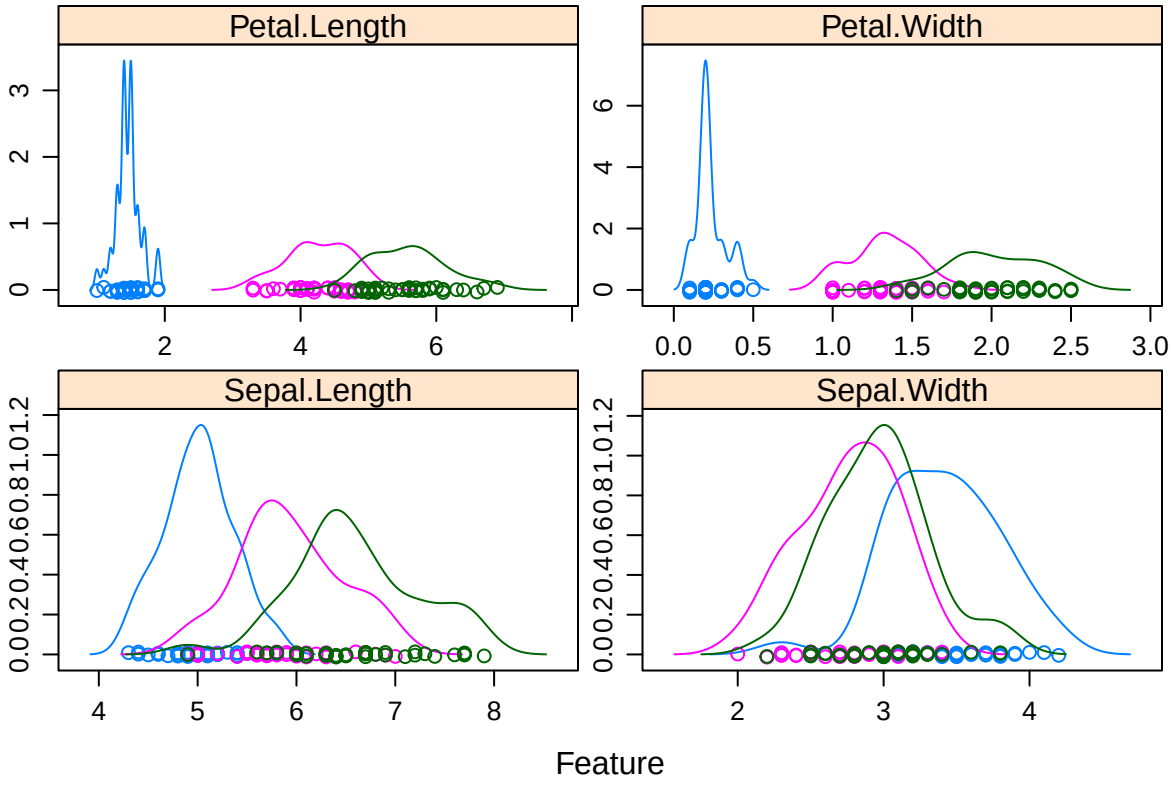



```
par(mfrow = c(1, 2))

#Analyzing the class distribution of the data
plot(y)

#Multivariate plot analysis using ellipse and boxplots
featurePlot(x,y,plot = "ellipse")
featurePlot(x,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)
```



Creating the predictive models

```
#Run algorithms using 10-fold cross validation
control <- trainControl(method="cv", number=10)

# 1)Linear algorithms
fit.lda <- train(Species~., data=dataset, method="lda", metric="Accuracy"
                 , trControl=control)

# 2)Nonlinear algorithms
# CART
fit.cart <- train(Species~., data=dataset, method="rpart", metric="Accuracy",
                  trControl=control)

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

# 3)Advanced algorithms
# SVM
fit.svm <- train(Species~., data=dataset, method="svmRadial", metric="Accuracy",
                 trControl=control)

# Random Forest
```

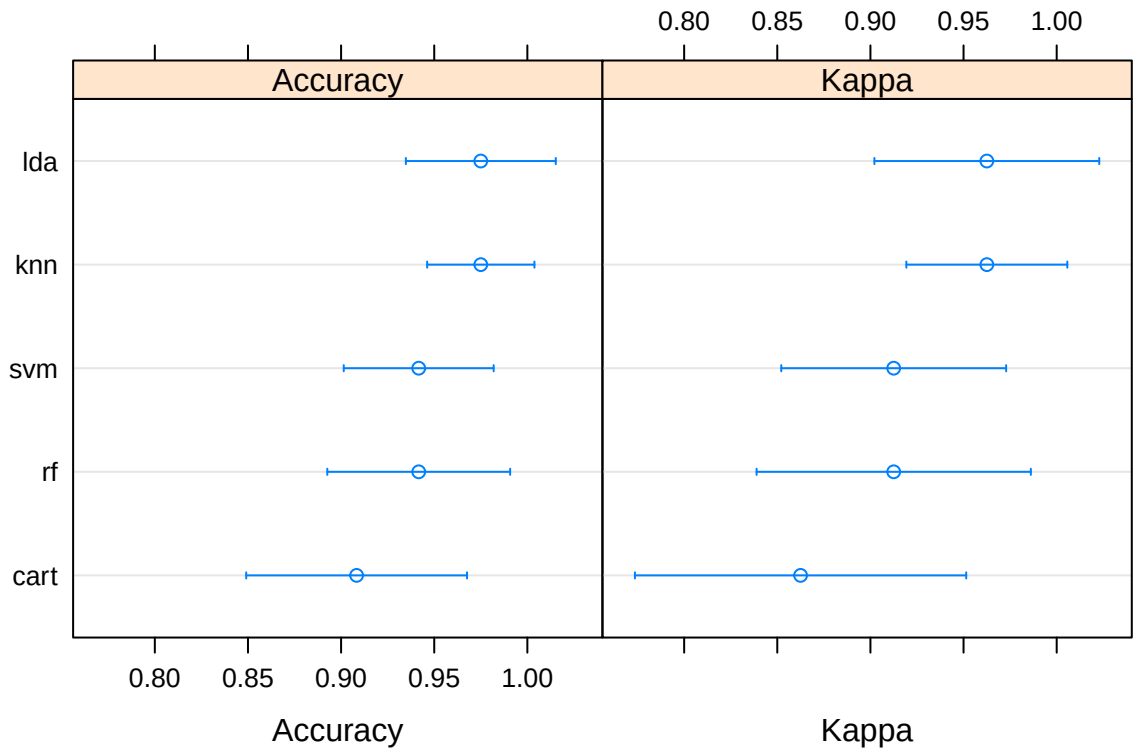
```
fit.rf <- train(Species~., data=dataset, method="rf", metric="Accuracy",
                trControl=control)
```

Selecting the best model and summarizing the results

```
#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  0.8333333 1.0000000 1.0000000 0.9750000 1.0000000    1    0
## cart 0.7500000 0.8541667 0.9166667 0.9083333 0.9791667    1    0
## knn  0.9166667 0.9375000 1.0000000 0.9750000 1.0000000    1    0
## svm  0.8333333 0.9166667 0.9166667 0.9416667 1.0000000    1    0
## rf   0.8333333 0.9166667 0.9583333 0.9416667 1.0000000    1    0
##
## Kappa
##       Min. 1st Qu. Median   Mean 3rd Qu. Max. NA's
## lda  0.750 1.00000 1.0000 0.9625 1.00000    1    0
## cart 0.625 0.78125 0.8750 0.8625 0.96875    1    0
## knn  0.875 0.90625 1.0000 0.9625 1.00000    1    0
## svm  0.750 0.87500 0.8750 0.9125 1.00000    1    0
## rf   0.750 0.87500 0.9375 0.9125 1.00000    1    0
```

```
#Compare accuracy of models
dotplot(results)
```



Confidence Level: 0.95

```
#Summarizing the best model
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
```

Making predictions based on the best model

```
#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         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
```