UCI MACHINE LEARNING REPOSITORY: IRIS DATASET

O Iris dataset é um conjunto de dados multivariado usado e tornado famoso pelo estatístico e biólogo britânico Ronald Fisher no seu artigo de 1936 "The use of multiple measurements in taxonomic problems" como um exemplo de análise discriminante linear. O conjunto de dados contém 3 classes de 50 instâncias cada, onde cada classe se refere a um tipo de planta de íris (Iris setosa, Iris virginica e Iris versicolor). Quatro características foram medidas de cada amostra: o comprimento e a largura das sépalas e das pétalas, em centímetros. *Com base na combinação dessas quatro características, Fisher desenvolveu um modelo discriminante linear para distinguir as espécies entre si.*

O que envolve definir os objetivos e os critérios de sucesso do projeto, compreender o contexto e o domínio do problema, identificar as fontes e a qualidade dos dados disponíveis, e determinar as técnicas e as ferramentas adequadas para a análise dos dados. ***O objetivo principal do projeto é classificar as plantas de íris em uma das três espécies com base nas suas características morfológicas, usando métodos de aprendizagem supervisionada. Isso é necessário porque as operações da empresa requerem a separação das mesmas classes, pois as classes setosa e virginica têm aplicações medicinais e são ingredientes em muitos medicamentos. Além disso, as folhas e raízes da classe versicolor são venenosas.*** *Os critérios de sucesso podem ser a precisão, a robustez e a interpretabilidade do modelo de classificação. O contexto e o domínio do problema podem ser explorados através de pesquisas bibliográficas sobre a biologia das plantas de íris, os métodos estatísticos usados por Fisher, e as aplicações práticas da classificação de plantas. As fontes e a qualidade dos dados podem ser avaliadas através da verificação da proveniência, da consistência, da completude e da relevância dos dados. As técnicas e as ferramentas adequadas para a análise dos dados podem ser selecionadas com base na natureza dos dados (numéricos, categóricos, multivariados), no tipo de problema (classificação), na complexidade do modelo (linear, não-linear) e na disponibilidade de recursos computacionais (memória, processamento, armazenamento).*

Uma compreensão dos dados referente ao CRISP-DM com a específica relação ao problema á frente pode ser obtida através da aplicação das seis fases do modelo de processo de mineração de dados: compreensão do negócio, compreensão dos dados, preparação dos dados, modelação, avaliação e implementação. Cada fase envolve uma série de tarefas que devem ser realizadas de forma iterativa e flexível, de acordo com os objetivos e requisitos do projeto.

1. *Na fase de compreensão do negócio, o objetivo é definir o problema a ser resolvido e os critérios de sucesso do projeto. Neste caso, o problema pode ser classificar as espécies de flores Iris com base nas medidas das suas características morfológicas, como o comprimento e a largura das sépalas e das pétalas. Os critérios de sucesso podem ser a accuracy, a precisão e a*

sensibilidade dos modelos de classificação. Sendo o principal a accuracy do modelo.

2. *Na fase de compreensão dos dados, o objetivo é coletar, descrever e explorar os dados disponíveis para o projeto.*

```
> head(dataset) # PRINT THE FIRST 10 LINES OF DATA (AS EXAMPLE)
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
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
```

Neste caso, os dados são provenientes do Iris dataset, um conjunto de dados clássico na área de aprendizado de máquina, que contém 150 instâncias de três espécies de flores Iris (setosa, versicolor e virginica), cada uma com quatro atributos numéricos (comprimento e largura das sépalas e das pétalas).

```
> dim(dataset) # GET DATASET DIMENSIONS (SPACE, ROWS, COLUMNS)
120   5 # O QUE SIGNIFICA QUE O DATASET É UMA TABELA COM 120
LINHAS E 5 COLUNAS.

> sapply(dataset, class) # SAPPLY() USED TO MAP FUNCTION TO
EACH ATTRIBUTE; CLASS FUNCTION RETURNS DATA CLASS TYPE FOR A
GIVEN ATTRIBUTE
Sepal.Length Sepal.Width Petal.Length  Petal.Width   Species
"numeric"    "numeric"    "numeric"    "numeric"     "factor"

> levels(dataset$Species) # LISTS ALL UNIQUE CLASS LEVELS
WITHIN THE SPECIES ATTRIBUTE OF DATASET
"setosa" "versicolor" "virginica"

> percentage <- prop.table(table(dataset$Species)) * 100 #
CREATE THE PERCENTAGE VARIABLE AND CALCULATE EACH INDIVIDUAL
LEVEL
> cbind(freq=table(dataset$Species), percentage=percentage) #
LISTS FREQUENCY AND PERCENTAGE OF EACH INDIVIDUAL CLASS LEVEL
WITHIN SPECIES ATTRIBUTE
           freq percentage
setosa       40   33.33333
versicolor   40   33.33333
virginica    40   33.33333
# Isto significa que cada classe compõem 33.(3)% do dataset.
```

Uma descrição estatística dos dados pode revelar informações como a média, o desvio padrão, o mínimo e o máximo de cada atributo, bem como a distribuição das classes. Uma exploração gráfica dos dados pode revelar informações como a correlação entre os atributos, a separabilidade das classes e a existência de outliers.

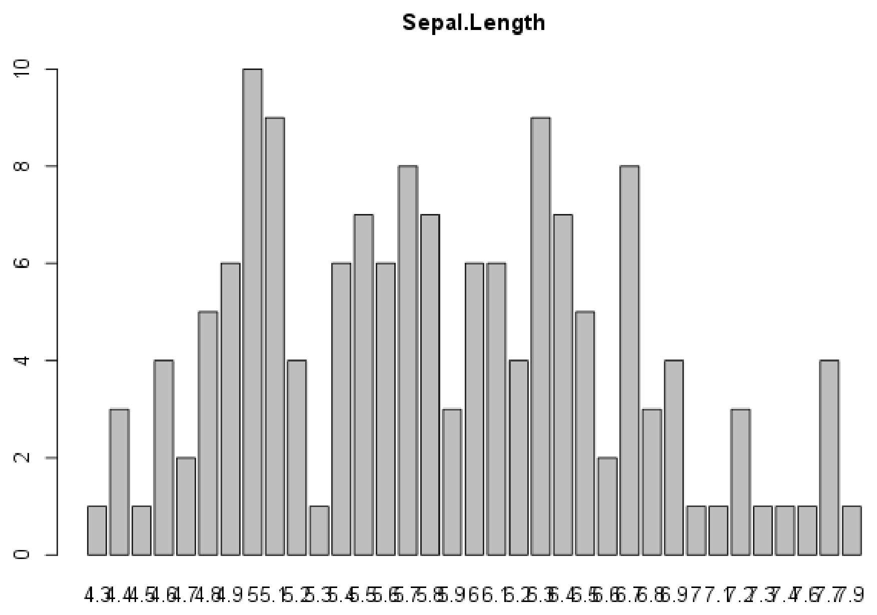
```
### MEAN: AVERAGE VALUE ###
> mean(Sepal.Length) # CALCULATE THE MEAN OF THE SEPAL LENGTH
ATTRIBUTE
5.843333
> mean(Sepal.Width) # CALCULATE THE MEAN OF THE SEPAL WIDTH
ATTRIBUTE
3.057333
> mean(Petal.Length) # CALCULATE THE MEAN OF THE PETAL LENGTH
ATTRIBUTE
3.758
> mean(Petal.Width) # CALCULATE THE MEAN OF THE PETAL WIDTH
ATTRIBUTE
1.199333

### MEDIAN: MIDDLE VALUE ###
> median(Sepal.Length) # CALCULATE THE MEDIAN OF THE SEPAL
LENGTH ATTRIBUTE
5.8
> median(Sepal.Width) # CALCULATE THE MEDIAN OF THE SEPAL WIDTH
ATTRIBUTE
3
> median(Petal.Length) # CALCULATE THE MEDIAN OF THE PETAL
LENGTH ATTRIBUTE
4.35
> median(Petal.Width) # CALCULATE THE MEDIAN OF THE PETAL WIDTH
ATTRIBUTE
1.3

### VARIANCE: SPREAD OF DATA ###
> var(Sepal.Length) # CALCULATE THE VARIANCE OF THE SEPAL
LENGTH ATTRIBUTE
0.6856935
> var(Sepal.Width) # CALCULATE THE VARIANCE OF THE SEPAL WIDTH
ATTRIBUTE
0.1899794
> var(Petal.Length) # CALCULATE THE VARIANCE OF THE PETAL
LENGTH ATTRIBUTE
3.116278
> var(Petal.Width) # CALCULATE THE VARIANCE OF THE PETAL WIDTH
ATTRIBUTE
0.5810063

### STANDARD DEVIATION ###
> sd(Sepal.Length) # CALCULATE THE STANDARD DEVIATION OF THE
SEPAL LENGTH ATTRIBUTE
0.8280661
> sd(Sepal.Width) # CALCULATE THE STANDARD DEVIATION OF THE
SEPAL WIDTH ATTRIBUTE
0.4358663
> sd(Petal.Length) # CALCULATE THE STANDARD DEVIATION OF THE
PETAL LENGTH ATTRIBUTE
1.765298
> sd(Petal.Width) # CALCULATE THE STANDARD DEVIATION OF THE
PETAL WIDTH ATTRIBUTE
0.7622377
```

```
### STATISTICAL SUMMARY ###
> summary(dataset) # CALCULATE THE STATISTICAL SUMMARY OF THE
DATASET
Sepal.Length    Sepal.Width     Petal.Length    Petal.Width
Species
Min.   :4.400   Min.   :2.000   Min.   :1.000   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 :4.300   Median :1.300
virginica :40
Mean   :5.853   Mean   :3.071   Mean   :3.778   Mean   :1.205
3rd Qu.:6.400   3rd Qu.:3.325   3rd Qu.:5.100   3rd Qu.:1.800
Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500

### ASYMMETRY ###
> skewness(Sepal.Length) # CALCULATE THE SKEWNESS OF THE SEPAL
LENGTH ATTRIBUTE (POSITIVE SKEWNESS INDICATES THAT THE MEAN OF
THE DATA VALUES IS LARGER THAN THE MEDIAN, AND THE DATA
DISTRIBUTION IS RIGHT-SKEWED; NEGATIVE SKEWNESS INDICATES THAT
THE MEAN OF THE DATA VALUES IS LESS THAN THE MEDIAN, AND THE
DATA DISTRIBUTION IS LEFT-SKEWED). SKEWNESS IS A MEASURE OF THE
ASYMMETRY OF THE PROBABILITY DISTRIBUTION OF A REAL-VALUED
RANDOM VARIABLE ABOUT ITS MEAN; MEANING THAT IT MEASURES THE
DEGREE OF DISTORTION FROM THE NORMAL DISTRIBUTION; IN OTHER
WORDS, SKEWNESS TELLS YOU THE AMOUNT AND DIRECTION OF SKEW
(LEFT OR RIGHT) IN A DATASET.
0.3086407
attr(,"method")
"moment"
> skewness(Sepal.Width) # CALCULATE THE SKEWNESS OF THE SEPAL
WIDTH ATTRIBUTE (POSITIVE SKEWNESS INDICATES THAT THE MEAN OF
THE DATA VALUES IS LARGER THAN THE MEDIAN, AND THE DATA
DISTRIBUTION IS RIGHT-SKEWED; NEGATIVE SKEWNESS INDICATES THAT
THE MEAN OF THE DATA VALUES IS LESS THAN THE MEDIAN, AND THE
DATA DISTRIBUTION IS LEFT-SKEWED). SKEWNESS IS A MEASURE OF THE
ASYMMETRY OF THE PROBABILITY DISTRIBUTION OF A REAL-VALUED
RANDOM VARIABLE ABOUT ITS MEAN; MEANING THAT IT MEASURES THE
DEGREE OF DISTORTION FROM THE NORMAL DISTRIBUTION; IN OTHER
WORDS, SKEWNESS TELLS YOU THE AMOUNT AND DIRECTION OF SKEW
(LEFT OR RIGHT) IN A DATASET.
0.3126147
attr(,"method")
"moment"
> skewness(Petal.Length) # CALCULATE THE SKEWNESS OF THE PETAL
LENGTH ATTRIBUTE (POSITIVE SKEWNESS INDICATES THAT THE MEAN OF
THE DATA VALUES IS LARGER THAN THE MEDIAN, AND THE DATA
DISTRIBUTION IS RIGHT-SKEWED; NEGATIVE SKEWNESS INDICATES THAT
THE MEAN OF THE DATA VALUES IS LESS THAN THE MEDIAN, AND THE
DATA DISTRIBUTION IS LEFT-SKEWED). SKEWNESS IS A MEASURE OF THE
ASYMMETRY OF THE PROBABILITY DISTRIBUTION OF A REAL-VALUED
RANDOM VARIABLE ABOUT ITS MEAN; MEANING THAT IT MEASURES THE
DEGREE OF DISTORTION FROM THE NORMAL DISTRIBUTION; IN OTHER
WORDS, SKEWNESS TELLS YOU THE AMOUNT AND DIRECTION OF SKEW
(LEFT OR RIGHT) IN A DATASET.
-0.2694109
attr(,"method")
"moment"
> skewness(Petal.Width) # CALCULATE THE SKEWNESS OF THE PETAL
WIDTH ATTRIBUTE (POSITIVE SKEWNESS INDICATES THAT THE MEAN OF
THE DATA VALUES IS LARGER THAN THE MEDIAN, AND THE DATA
```

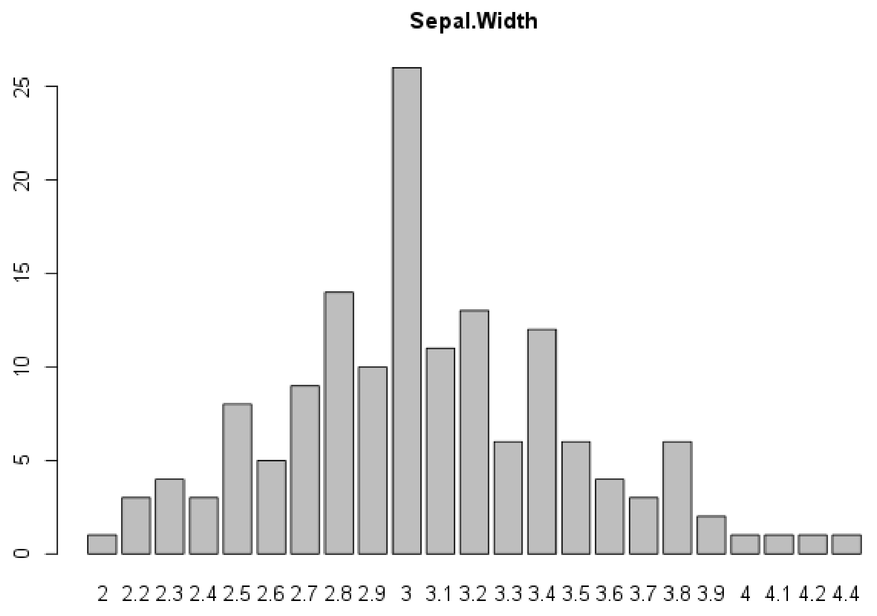
```
DISTRIBUTION IS RIGHT-SKEWED; NEGATIVE SKEWNESS INDICATES THAT
THE MEAN OF THE DATA VALUES IS LESS THAN THE MEDIAN, AND THE
DATA DISTRIBUTION IS LEFT-SKEWED). SKEWNESS IS A MEASURE OF THE
ASYMMETRY OF THE PROBABILITY DISTRIBUTION OF A REAL-VALUED
RANDOM VARIABLE ABOUT ITS MEAN; MEANING THAT IT MEASURES THE
DEGREE OF DISTORTION FROM THE NORMAL DISTRIBUTION; IN OTHER
WORDS, SKEWNESS TELLS YOU THE AMOUNT AND DIRECTION OF SKEW
(LEFT OR RIGHT) IN A DATASET.
-0.1009166
attr(,"method")
"moment"

### KURTOSIS ###
> kurtosis(Sepal.Length) # CALCULATE THE KURTOSIS OF THE SEPAL
LENGTH ATTRIBUTE (IF THE KURTOSIS IS GREATER THAN 3, THEN THE
DATASET HAS HEAVIER TAILS THAN A NORMAL DISTRIBUTION (MORE IN
THE TAILS) (LEPTOKURTIC DISTRIBUTION); IF THE KURTOSIS IS EQUAL
TO 3, THEN THE DATASET HAS HA NORMAL DISTRIBUTION; IF THE
KURTOSIS IS LESS THAN 3, THEN THE DATASET HAS LIGHTER TAILS
THAN A NORMAL DISTRIBUTION (LESS IN THE TAILS) (PLATYKURTIC
DISTRIBUTION)); KURTOSIS IS A MEASURE OF THE "TAILEDNESS" OF
THE PROBABILITY DISTRIBUTION OF A REAL-VALUED RANDOM VARIABLE;
IN OTHER WORDS, KURTOSIS TELLS YOU THE HEIGHT AND SHARPNESS OF
THE CENTRAL PEAK, RELATIVELY TO THAT OF A STANDARD BELL CURVE.
-0.6058125
attr(,"method")
"excess"
> kurtosis(Sepal.Width) # CALCULATE THE KURTOSIS OF THE SEPAL
WIDTH ATTRIBUTE (IF THE KURTOSIS IS GREATER THAN 3, THEN THE
DATASET HAS HEAVIER TAILS THAN A NORMAL DISTRIBUTION (MORE IN
THE TAILS) (LEPTOKURTIC DISTRIBUTION); IF THE KURTOSIS IS EQUAL
TO 3, THEN THE DATASET HAS HA NORMAL DISTRIBUTION; IF THE
KURTOSIS IS LESS THAN 3, THEN THE DATASET HAS LIGHTER TAILS
THAN A NORMAL DISTRIBUTION (LESS IN THE TAILS) (PLATYKURTIC
DISTRIBUTION)); KURTOSIS IS A MEASURE OF THE "TAILEDNESS" OF
THE PROBABILITY DISTRIBUTION OF A REAL-VALUED RANDOM VARIABLE;
IN OTHER WORDS, KURTOSIS TELLS YOU THE HEIGHT AND SHARPNESS OF
THE CENTRAL PEAK, RELATIVELY TO THAT OF A STANDARD BELL CURVE.
0.1387047
attr(,"method")
"excess"
> kurtosis(Petal.Length) # CALCULATE THE KURTOSIS OF THE PETAL
LENGTH ATTRIBUTE (IF THE KURTOSIS IS GREATER THAN 3, THEN THE
DATASET HAS HEAVIER TAILS THAN A NORMAL DISTRIBUTION (MORE IN
THE TAILS) (LEPTOKURTIC DISTRIBUTION); IF THE KURTOSIS IS EQUAL
TO 3, THEN THE DATASET HAS HA NORMAL DISTRIBUTION; IF THE
KURTOSIS IS LESS THAN 3, THEN THE DATASET HAS LIGHTER TAILS
THAN A NORMAL DISTRIBUTION (LESS IN THE TAILS) (PLATYKURTIC
DISTRIBUTION)); KURTOSIS IS A MEASURE OF THE "TAILEDNESS" OF
THE PROBABILITY DISTRIBUTION OF A REAL-VALUED RANDOM VARIABLE;
IN OTHER WORDS, KURTOSIS TELLS YOU THE HEIGHT AND SHARPNESS OF
THE CENTRAL PEAK, RELATIVELY TO THAT OF A STANDARD BELL CURVE.
-1.416857
attr(,"method")
"excess"
> kurtosis(Petal.Width) # CALCULATE THE KURTOSIS OF THE PETAL
WIDTH ATTRIBUTE (IF THE KURTOSIS IS GREATER THAN 3, THEN THE
DATASET HAS HEAVIER TAILS THAN A NORMAL DISTRIBUTION (MORE IN
THE TAILS) (LEPTOKURTIC DISTRIBUTION); IF THE KURTOSIS IS EQUAL
TO 3, THEN THE DATASET HAS HA NORMAL DISTRIBUTION; IF THE
KURTOSIS IS LESS THAN 3, THEN THE DATASET HAS LIGHTER TAILS
```

```
THAN A NORMAL DISTRIBUTION (LESS IN THE TAILS) (PLATYKURTIC
DISTRIBUTION)); KURTOSIS IS A MEASURE OF THE "TAILEDNESS" OF
THE PROBABILITY DISTRIBUTION OF A REAL-VALUED RANDOM VARIABLE;
IN OTHER WORDS, KURTOSIS TELLS YOU THE HEIGHT AND SHARPNESS OF
THE CENTRAL PEAK, RELATIVELY TO THAT OF A STANDARD BELL CURVE.
-1.358179
attr(,"method")
"excess"

### FREQUENT SUB-INTERVAL TABLE ###
> table(cut(Sepal.Length,3)) # FREQUENT SUB-INTERVAL TABLE OF
THE SEPAL LENGTH VARIABLE: THE FREQUENT SUB-INTERVAL TABLE IS A
TABLE THAT SHOWS THE NUMBER OF OBSERVATIONS THAT FALL INTO EACH
OF SEVERAL NON-OVERLAPPING CATEGORIES; IT IS A SPECIAL CASE OF
A HISTOGRAM DIVIDED INTO NON-OVERLAPPING INTERVALS; FREQUENT
SUB-INTERVAL TABLES ARE USED TO SUMMARIZE CATEGORICAL, COUNT,
OR CONTINUOUS DATA.
(4.3,5.5] (5.5,6.7] (6.7,7.9]
59        71        20
> table(cut(Sepal.Width,3)) # FREQUENT SUB-INTERVAL TABLE OF
THE SEPAL WIDTH VARIABLE: THE FREQUENT SUB-INTERVAL TABLE IS A
TABLE THAT SHOWS THE NUMBER OF OBSERVATIONS THAT FALL INTO EACH
OF SEVERAL NON-OVERLAPPING CATEGORIES; IT IS A SPECIAL CASE OF
A HISTOGRAM DIVIDED INTO NON-OVERLAPPING INTERVALS; FREQUENT
SUB-INTERVAL TABLES ARE USED TO SUMMARIZE CATEGORICAL, COUNT,
OR CONTINUOUS DATA.
(2,2.8] (2.8,3.6] (3.6,4.4]
47        88        15
> table(cut(Petal.Length,3)) # FREQUENT SUB-INTERVAL TABLE OF
THE PETAL LENGTH VARIABLE: THE FREQUENT SUB-INTERVAL TABLE IS A
TABLE THAT SHOWS THE NUMBER OF OBSERVATIONS THAT FALL INTO EACH
OF SEVERAL NON-OVERLAPPING CATEGORIES; IT IS A SPECIAL CASE OF
A HISTOGRAM DIVIDED INTO NON-OVERLAPPING INTERVALS; FREQUENT
SUB-INTERVAL TABLES ARE USED TO SUMMARIZE CATEGORICAL, COUNT,
OR CONTINUOUS DATA.
(0.994,2.97]  (2.97,4.93]  (4.93,6.91]
50            54            46
> table(cut(Petal.Width,3)) # FREQUENT SUB-INTERVAL TABLE OF
THE PETAL WIDTH VARIABLE: THE FREQUENT SUB-INTERVAL TABLE IS A
TABLE THAT SHOWS THE NUMBER OF OBSERVATIONS THAT FALL INTO EACH
OF SEVERAL NON-OVERLAPPING CATEGORIES; IT IS A SPECIAL CASE OF
A HISTOGRAM DIVIDED INTO NON-OVERLAPPING INTERVALS; FREQUENT
SUB-INTERVAL TABLES ARE USED TO SUMMARIZE CATEGORICAL, COUNT,
OR CONTINUOUS DATA.
(0.0976,0.9]    (0.9,1.7]    (1.7,2.5]
50            54            46
```

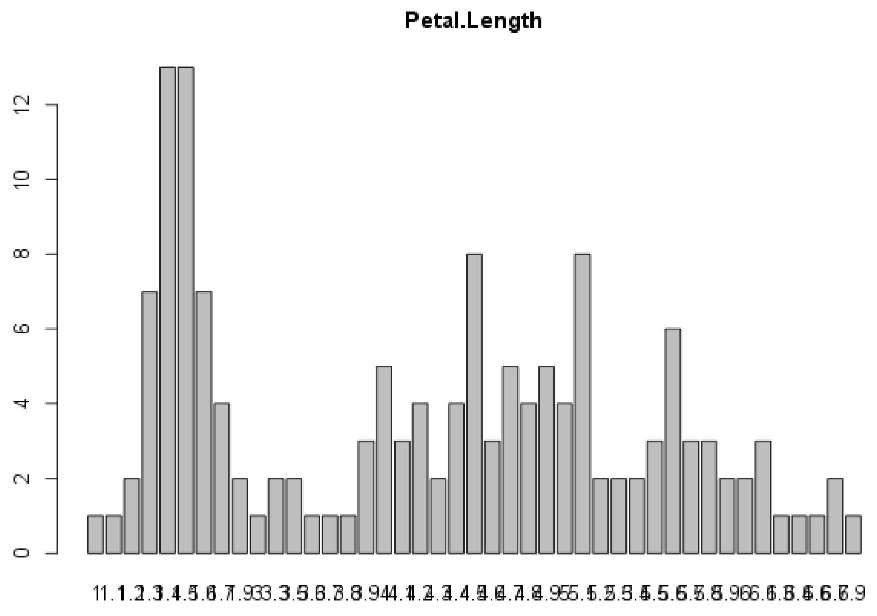
```
### THE PROPORTION OF THE VARIABLES BY SPECIES ###
> prop.table(xtabs(Sepal.Length~Species)) # THE PROPORTION OF
THE SEPAL LENGTH VARIABLE BY SPECIES
Species
setosa versicolor  virginica
0.2855676  0.3386195  0.3758129
> prop.table(xtabs(Sepal.Width~Species)) # THE PROPORTION OF
THE SEPAL WIDTH VARIABLE BY SPECIES
Species
setosa versicolor  virginica
0.3737462  0.3020061  0.3242477
> prop.table(xtabs(Petal.Length~Species)) # THE PROPORTION OF
THE PETAL LENGTH VARIABLE BY SPECIES
Species
setosa versicolor  virginica
0.1296789  0.3778606  0.4924605
> prop.table(xtabs(Petal.Width~Species)) # THE PROPORTION OF
THE PETAL WIDTH VARIABLE BY SPECIES
Species
setosa versicolor  virginica
0.06837132 0.36853808 0.56309061

################## DATA SUMMARY: GRAPHS ##################
### BARPLOTS ###
> barplot(table(Sepal.Length),main="Sepal.Length") # BARPLOT OF
THE SEPAL LENGTH VARIABLE
```



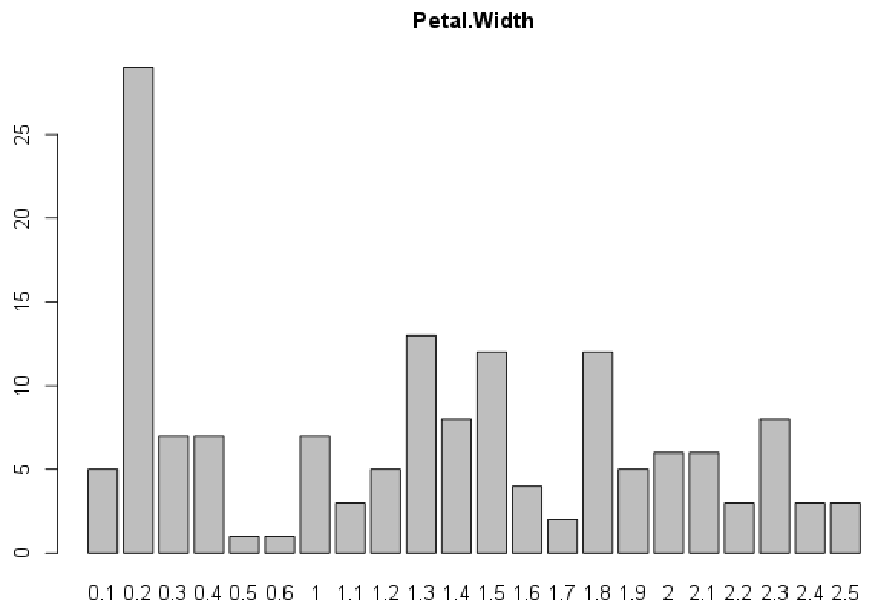
```
> barplot(table(Sepal.Width),main="Sepal.Width") # BARPLOT OF
THE SEPAL WIDTH VARIABLE
```



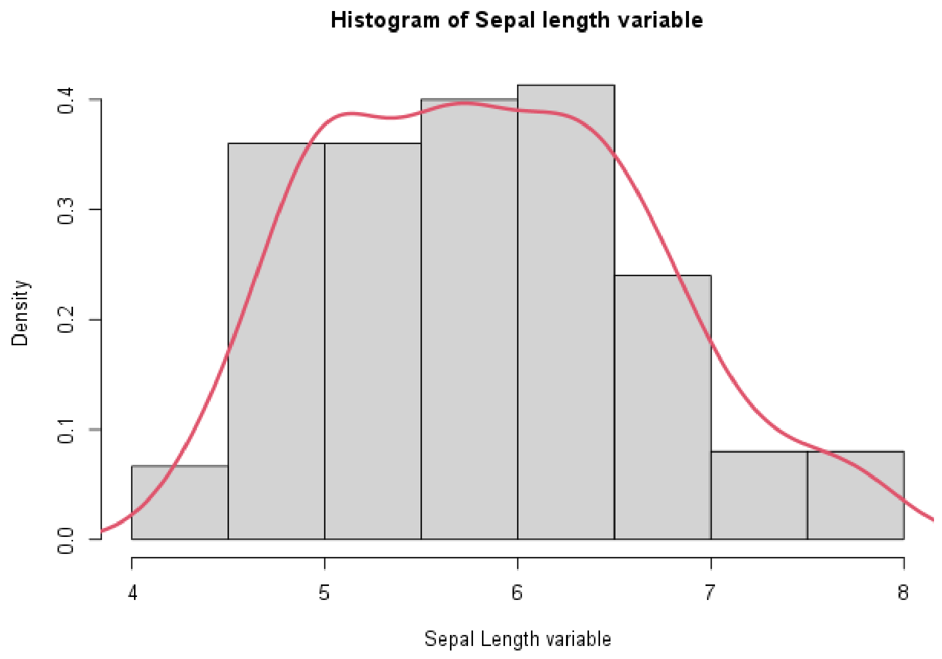
```
> barplot(table(Petal.Length),main="Petal.Length") # BARPLOT OF
THE PETAL LENGTH VARIABLE
```



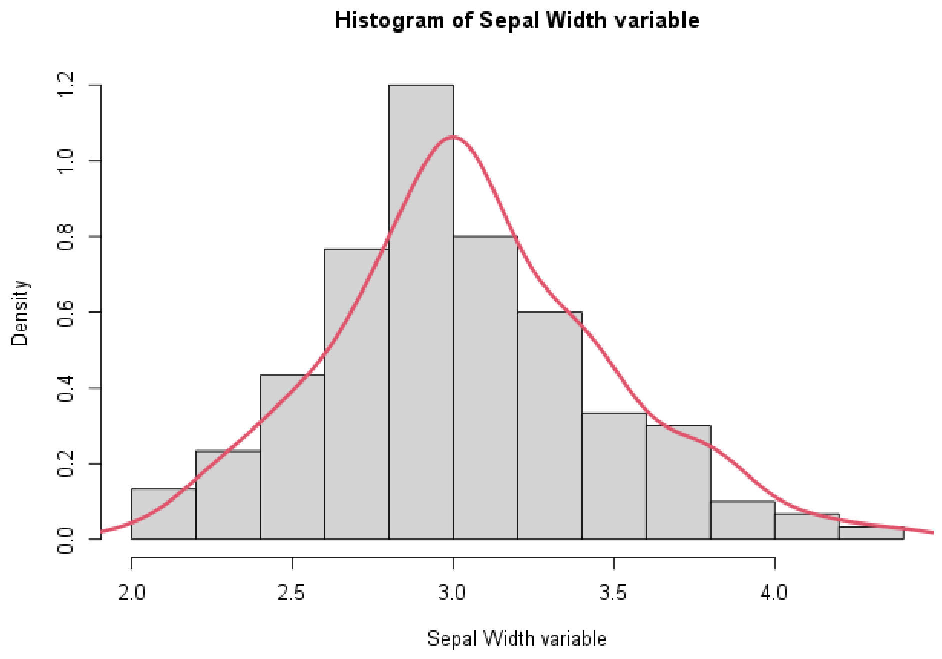
```
> barplot(table(Petal.Width),main="Petal.Width") # BARPLOT OF
THE PETAL WIDTH VARIABLE
```



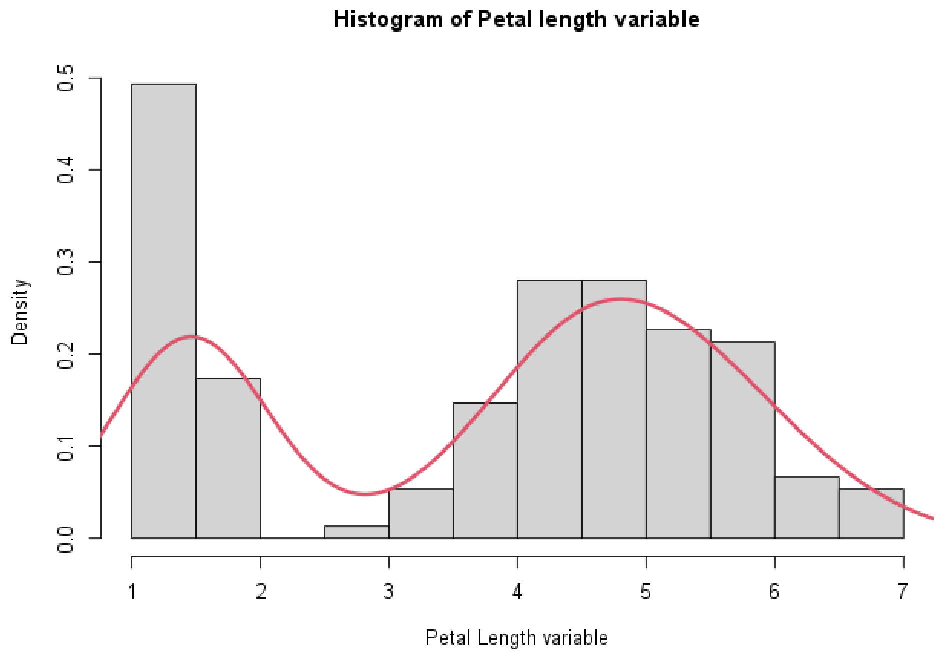
```
### HISTOGRAMS ###
# CREATE HISTOGRAM OF SEPAL LENGTH VARIABLE
> hist(Sepal.Length,freq = F,xlab = "Sepal Length variable",
main = "Histogram of Sepal length variable")
> lines(density(Sepal.Length),col=2,lwd=3) # ADDING THE DENSITY
CURVE OVER THE POLT
```



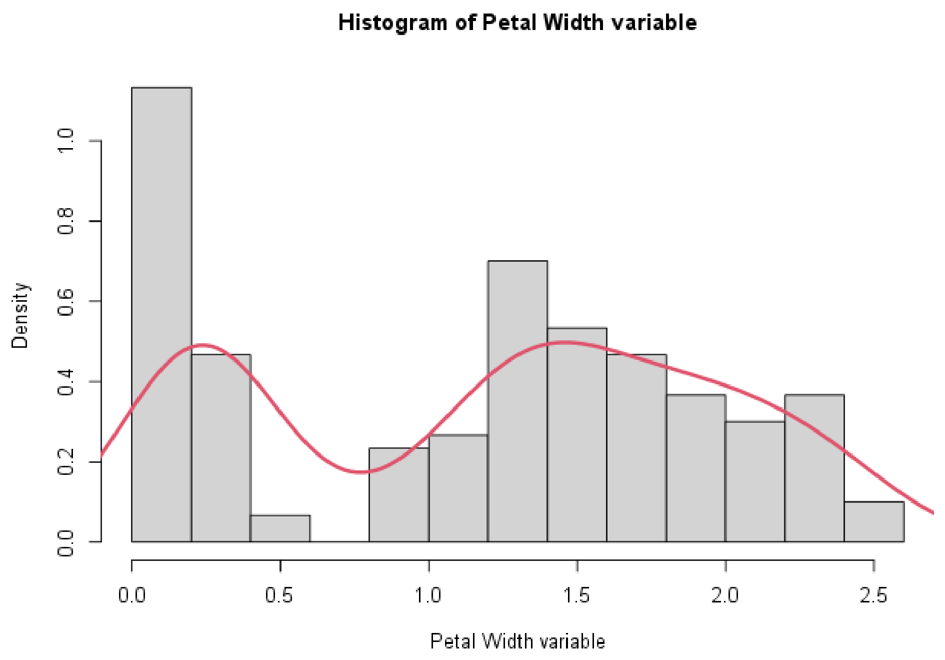
```
# CREATE HISTOGRAM OF SEPAL WIDTH VARIABLE
> hist(Sepal.Width,freq = F,xlab = "Sepal Width variable", main
= "Histogram of Sepal Width variable")
> lines(density(Sepal.Width),col=2,lwd=3) # ADDING THE DENSITY
CURVE OVER THE POLT
```



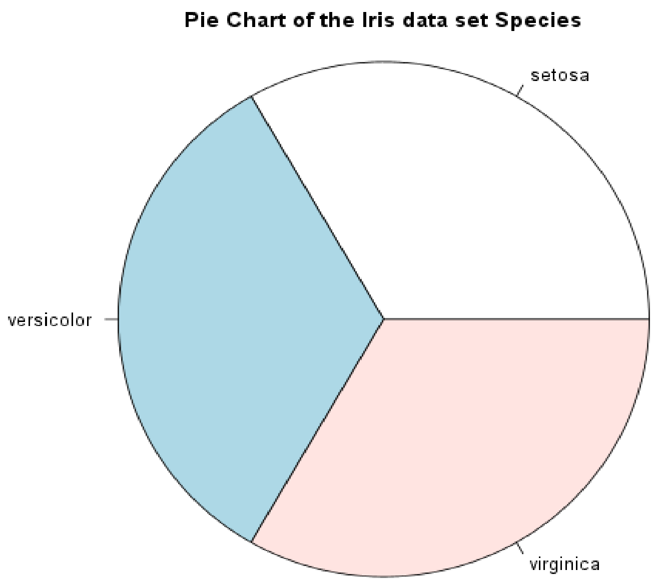
```
# CREATE HISTOGRAM OF PETAL LENGTH VARIABLE
> hist(Petal.Length,freq = F,xlab = "Petal Length variable",
main = "Histogram of Petal length variable")
> lines(density(Petal.Length),col=2,lwd=3) # ADDING THE DENSITY
CURVE OVER THE POLT
```



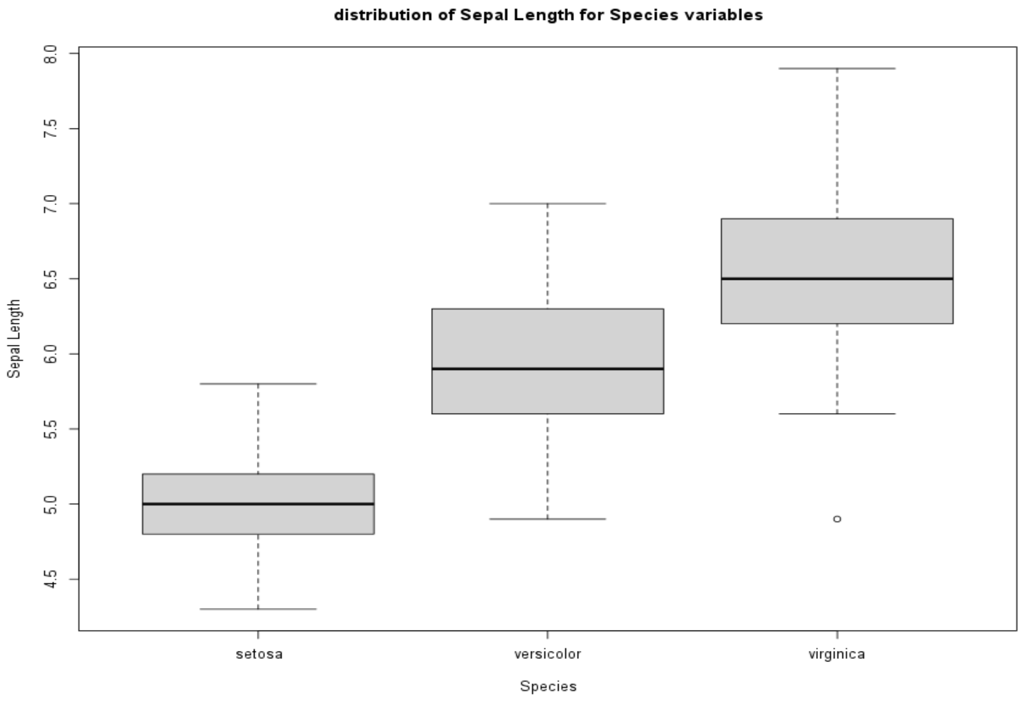
```
# CREATE HISTOGRAM OF PETAL WIDTH VARIABLE
> hist(Petal.Width,freq = F,xlab = "Petal Width variable", main
= "Histogram of Petal Width variable")
> lines(density(Petal.Width),col=2,lwd=3) # ADDING THE DENSITY
CURVE OVER THE POLT
```



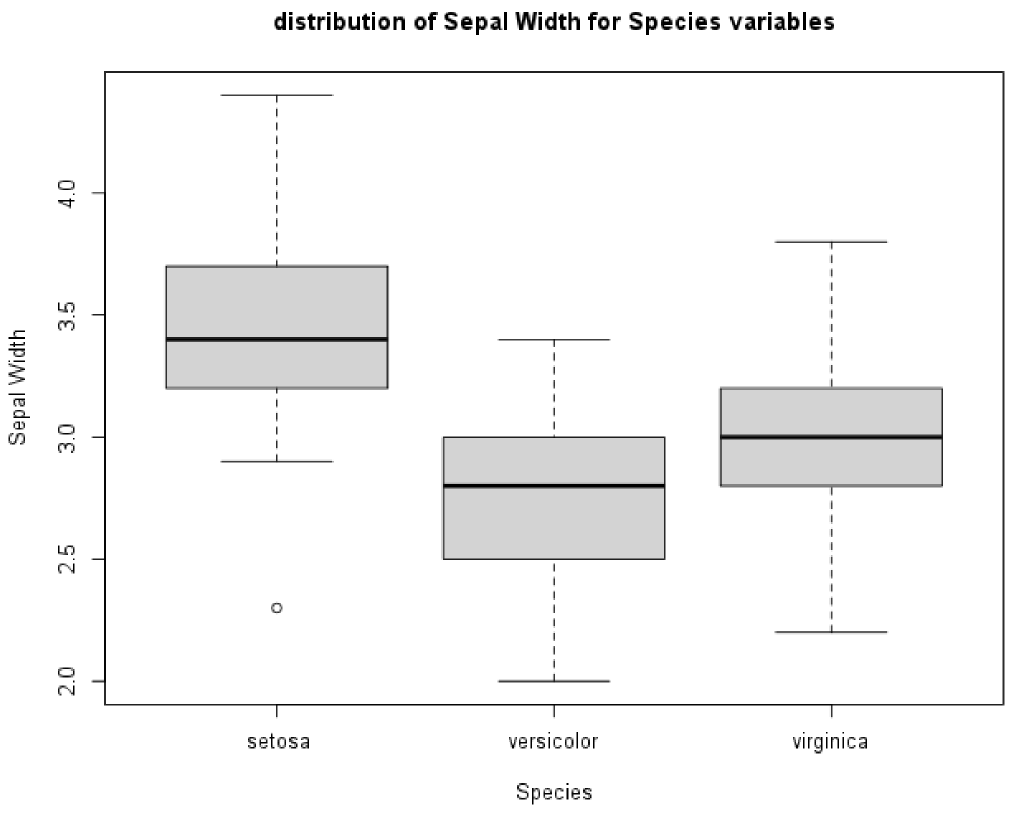
```
# NOTICE THE SHAPE OF THE DATA, MOST ATTRIBUTES EXHIBIT A
NORMAL DISTRIBUTION.
# PIE TO SHOW THE SPECIES
> pie(table(Species), main = "Pie Chart of the Iris data set
Species", radius = 1)
```



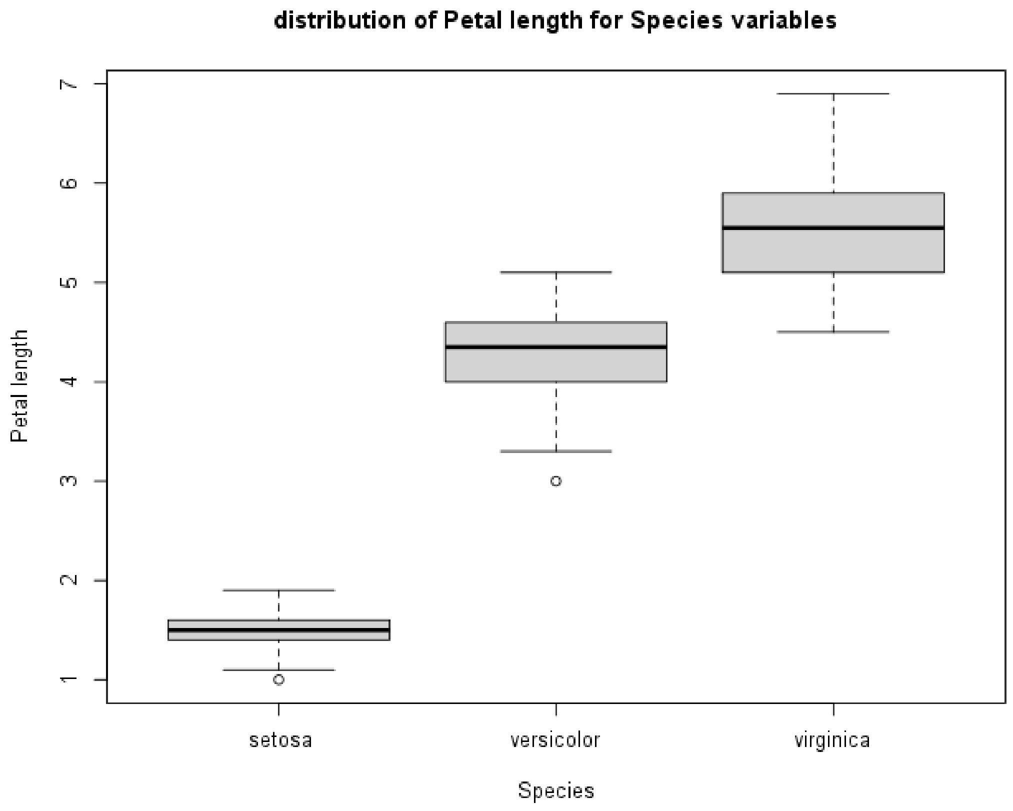
```
### BOXPLOTS ###
# COMPARE EACH VARIABLE WITH THE SPECIES VARIABLE
> boxplot(Sepal.Length~Species, ylab = "Sepal Length", main =
"distribution of Sepal Length for Species variables") # BOXPLOT
OF THE SEPAL LENGTH VARIABLE BY SPECIES
```



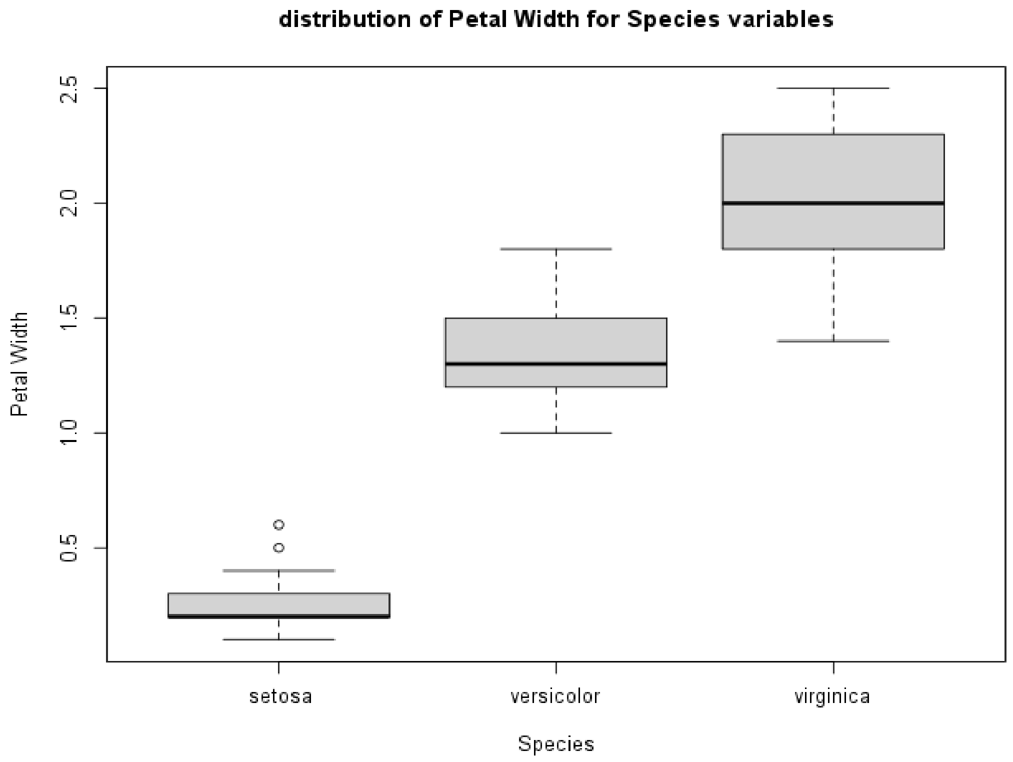
```
> boxplot(Sepal.Width~Species, ylab = "Sepal Width", main =
"distribution of Sepal Width for Species variables") # BOXPLOT
OF THE SEPAL WIDTH VARIABLE BY SPECIES
```



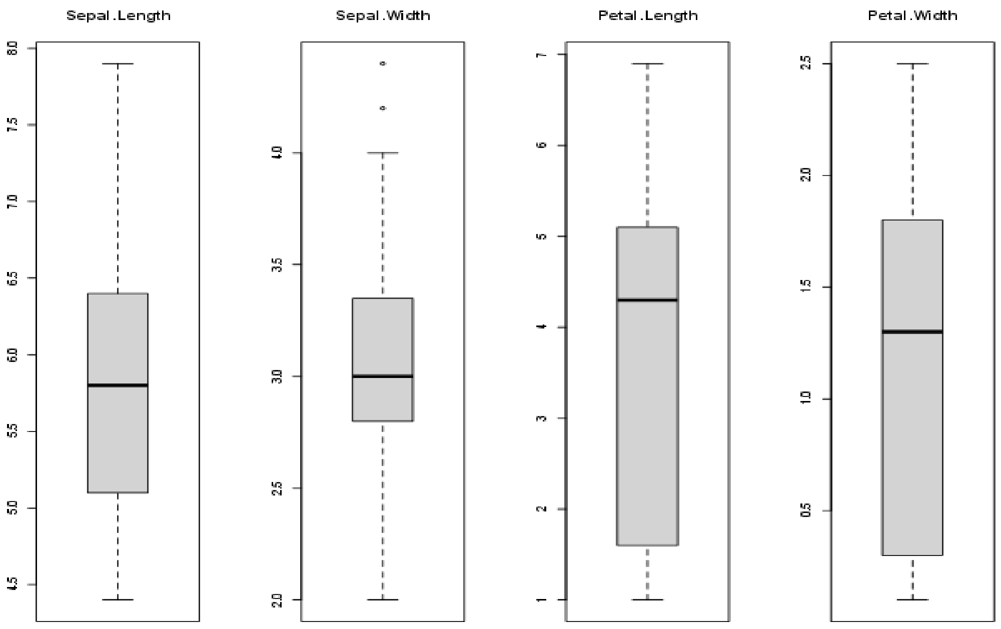
```
> boxplot(Petal.Length~Species, ylab = "Petal length", main =
"distribution of Petal length for Species variables") # BOXPLOT
OF THE PETAL LENGTH VARIABLE BY SPECIES
```



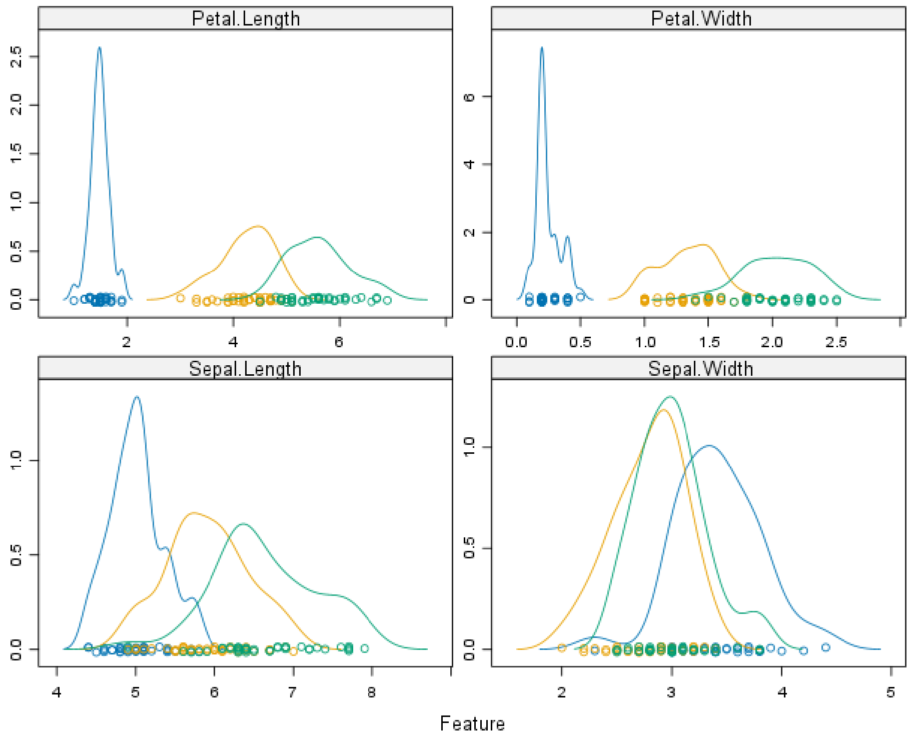
```
> boxplot(Petal.Width~Species, ylab = "Petal Width", main =
"distribution of Petal Width for Species variables") # BOXPLOT
OF THE PETAL WIDTH VARIABLE BY SPECIES
```



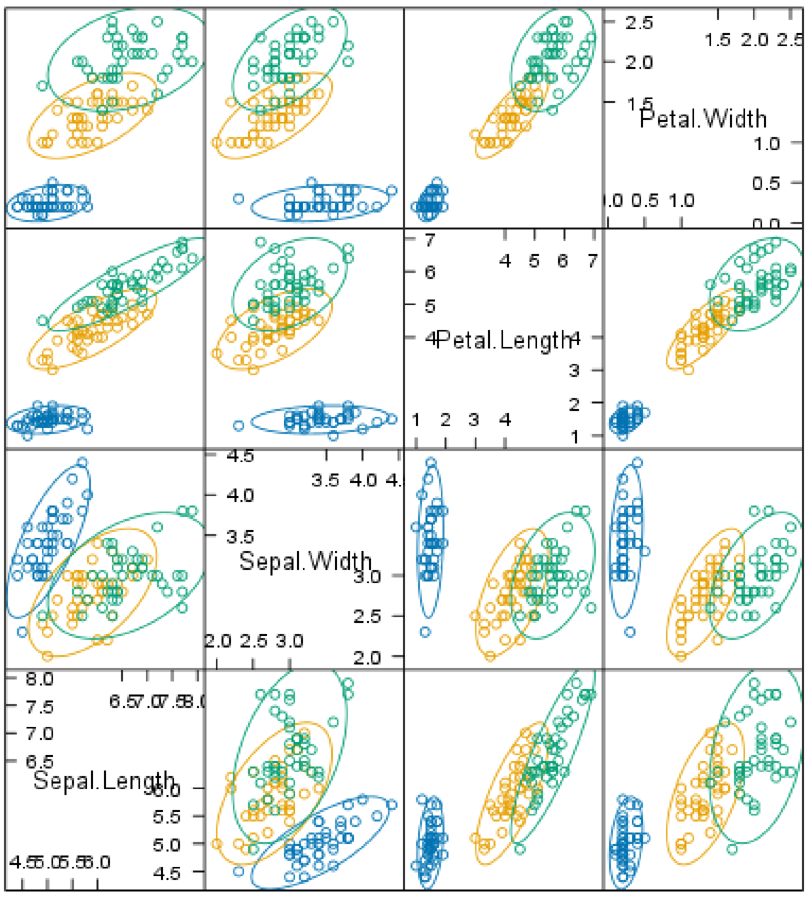
```
# FEATURE BOXPLOTS:
par(mfrow=c(1,4)) # SET THE PLOT TO 1 ROW AND 4 COLUMNS
for(i in 1:4) { # LOOP THROUGH THE FIRST 4 COLUMNS
  boxplot(x[,i], main=names(iris)[i]) # CREATE BOXPLOT FOR EACH
COLUMN
} # END OF LOOP
```



```
### DENSITY PLOTS ###
> scales <- list(x=list(relation="free"),
y=list(relation="free")) # SET THE SCALES FOR THE PLOT: FREE
SCALES
> featurePlot(x=x, y=y, plot="density", scales=scales) # CREATE
THE PLOT FOR THE DENSITY OF THE VARIABLES
```



```
# GET THE PLOT FOR NUMERIC COLUMNS:
> pairs(iris[,1:4]) # CREATE THE PLOT FOR THE NUMERIC COLUMNS
OF THE DATASET
> featurePlot(x=x, y=y, plot="ellipse") # CREATE THE PLOT FOR
THE ELLIPSE OF THE VARIABLES
```



Scatter Plot Matrix

```
# CHECK THE COVARIANCE AND CORRELATION BETWEEN VARIABLES
> cov(dataset[ ,1:4]) # COVARIANCE MATRIX OF THE FIRST 4
COLUMNS; THE COVARIANCE IS A MEASURE OF THE JOINT VARIABILITY
OF TWO RANDOM VARIABLES.
Sepal.Length Sepal.Width Petal.Length Petal.Width
Sepal.Length   0.70458754 -0.03152941    1.3079706  0.52155742
Sepal.Width   -0.03152941  0.18578151   -0.2832269 -0.09905882
Petal.Length   1.30797059 -0.28322689    3.1030000  1.26860504
Petal.Width    0.52155742 -0.09905882    1.2686050  0.55928291
> cor(dataset[ ,1:4]) # CORRELATION MATRIX OF THE FIRST 4
COLUMNS; THE CORRELATION IS A MEASURE OF THE LINEAR ASSOCIATION
BETWEEN TWO RANDOM VARIABLES. (CORRELATION IS POSITIVE WHEN THE
VALUES INCREASE TOGETHER (POSITIVE LINEAR CORRELATION);
CORRELATION IS ~ ZERO WHEN THERE IS NO LINEAR ASSOCIATION
BETWEEN VARIABLES; CORRELATION IN NEGATIVE WHEN THE VALUES
DECREASES TOGETHER (NEGATIVE LINEAR CORRELATION).)
Sepal.Length Sepal.Width Petal.Length Petal.Width
Sepal.Length   1.00000000 -0.08714594    0.8845851   0.8308428
Sepal.Width   -0.08714594  1.00000000   -0.3730288  -0.3073096
Petal.Length   0.88458509 -0.37302879    1.0000000   0.9629855
Petal.Width    0.83084283 -0.30730961    0.9629855   1.0000000
```

3. *Na fase de preparação dos dados, o objetivo é selecionar, limpar, transformar e integrar os dados que serão usados na modelação. Neste caso, os dados já estão em um formato adequado para a modelação, pois não apresentam valores ausentes, inconsistências ou ruídos. No entanto, pode-se realizar algumas operações como normalizar ou padronizar os atributos para evitar que alguns tenham mais influência que outros na modelação, ou reduzir a dimensionalidade dos dados para eliminar atributos irrelevantes ou redundantes. Visto que os dados já estão bastante padronizados, normalizados, e ajustados, falta-nos só preparar as variáveis para a análise e modelação dos mesmos.*

```
### SPLIT DATA INTO TRAIN/TEST SETS ###
> data_split <- createDataPartition(dataset$Species, p = 0.8,
list = FALSE) # CREATE A 80/20 SPLIT OF THE DATA FOR
TRAINING/TESTING PURPOSES
> test <- dataset[-data_split,] # SET 20% OF DATA AS TEST SET
> dataset <- dataset[data_split,] # SET THE REMAINING 80% OF
DATA FOR TRAINING

### SEPARATE THE TRAINING DATA FROM THE LABELS ###
> x <- dataset[,1:4] # SET THE FIRST 4 COLUMNS AS THE TRAINING
DATA
> y <- dataset[,5] # SET THE 5TH COLUMN AS THE LABELS
```

4. *Na fase de modelação, o objetivo é aplicar técnicas de mineração de dados para construir um ou mais modelos que atendam aos objetivos do projeto. Neste caso, como se trata de um problema de classificação supervisionada, pode-se aplicar técnicas como árvore de decisão, k-nearest-neighbours, regressão logística, rede neural artificial ou máquina de vetores de suporte. Cada técnica possui seus próprios parâmetros que devem ser ajustados para obter o melhor desempenho possível. Além disso, deve-se dividir os dados em conjuntos de treinamento e teste para evitar o sobreajuste dos modelos.*

```
# ALGORITHMS ASSESSED USING 10-FOLD CROSS VALIDATION. METRIC
FOR ASSESSMENT IN THIS EXAMPLE IS ACCURACY.
> control <- trainControl(method="cv", number=10) # SET THE
CONTROL FOR THE TRAINING (10-FOLD CROSS VALIDATION)
> metric <- "Accuracy" # SET THE METRIC FOR THE TRAINING
(ACCURACY)

### LINEAR DISCRIMINANT ANALYSIS ###
> set.seed(7) # SET THE SEED FOR REPRODUCIBILITY OF THE RESULTS
> fit.lda <- train(Species~., data=dataset, method="lda",
metric=metric, trControl=control) # TRAIN THE MODEL USING THE
LDA METHOD (LINEAR DISCRIMINANT ANALYSIS)

### CLASSIFICATION AND REGRESSION TREES ###
> set.seed(7) # SET THE SEED FOR REPRODUCIBILITY OF THE RESULTS
> fit.cart <- train(Species~., data=dataset, method="rpart",
metric=metric, trControl=control) # TRAIN THE MODEL USING THE
CART METHOD (CLASSIFICATION AND REGRESSION TREES)
```

```
### K-NEAREST NEIGHBORS ###
> set.seed(7) # SET THE SEED FOR REPRODUCIBILITY OF THE RESULTS
> fit.knn <- train(Species~., data=dataset, method="knn",
metric=metric, trControl=control) # TRAIN THE MODEL USING THE
KNN METHOD (K-NEAREST NEIGHBORS)

### SUPPORT VECTOR MACHINES ###
> set.seed(7) # SET THE SEED FOR REPRODUCIBILITY OF THE RESULTS
> fit.svm <- train(Species~., data=dataset, method="svmRadial",
metric=metric, trControl=control) # TRAIN THE MODEL USING THE
SVM METHOD (SUPPORT VECTOR MACHINES)

### RANDOM FOREST ###
> set.seed(7) # SET THE SEED FOR REPRODUCIBILITY OF THE RESULTS
> fit.rf <- train(Species~., data=dataset, method="rf",
metric=metric, trControl=control) # TRAIN THE MODEL USING THE
RF METHOD (RANDOM FOREST)
```

5. *Na fase de avaliação, o objetivo é avaliar os modelos construídos em termos da sua qualidade e relevância para o problema do negócio. Neste caso, pode-se usar métricas como accuracy, precisão e sensibilidade para medir o quão bem os modelos classificam as espécies de flores Iris. Também pode-se usar técnicas como matriz de confusão ou curva ROC para visualizar o desempenho dos modelos. Além disso, deve-se verificar se os modelos atendem aos critérios de sucesso definidos na fase de compreensão do negócio.*

```
### SUMMARIZE THE RESULTS ###
> results <- resamples(list(lda=fit.lda, cart=fit.cart,
knn=fit.knn, svm=fit.svm, rf=fit.rf)) # CREATE A LIST OF THE
RESULTS OF THE MODELS
> summary(results) # SUMMARIZE THE 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  0.833333 0.9375000 1.0000000 0.9666667 1.0000000    1    0
cart 0.833333 0.9166667 0.9166667 0.9333333 0.9791667    1    0
knn  0.916667 1.0000000 1.0000000 0.9833333 1.0000000    1    0
svm  0.916667 0.9375000 1.0000000 0.9750000 1.0000000    1    0
rf   0.833333 0.9375000 1.0000000 0.9666667 1.0000000    1    0

Kappa
      Min. 1st Qu. Median   Mean 3rd Qu. Max. NA's
lda  0.750 0.90625  1.000 0.9500 1.00000    1    0
cart 0.750 0.87500  0.875 0.9000 0.96875    1    0
knn  0.875 1.00000  1.000 0.9750 1.00000    1    0
svm  0.875 0.90625  1.000 0.9625 1.00000    1    0
rf   0.750 0.90625  1.000 0.9500 1.00000    1    0
```

Como se pode observar o melhor modelo é quase em todas as 'runs' o k-nearest-neighbours, ás vezes é equiparado pelo modelo support vector machines. Tendo estes uma média (em accuracy) de 91.6%.

```
### MODEL SUMMARIES ###
> print(fit.lda) # PRINT THE SUMMARY OF THE LDA MODEL
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
0.9666667  0.95

> print(fit.cart) # PRINT THE SUMMARY OF THE CART MODEL
CART

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 across tuning parameters:

cp    Accuracy   Kappa
0.00  0.9333333  0.9000
0.45  0.7583333  0.6375
0.50  0.3333333  0.0000

Accuracy was used to select the optimal model using the largest
value.
The final value used for the model was cp = 0.

> print(fit.knn) # PRINT THE SUMMARY OF THE KNN MODEL
k-Nearest Neighbors

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 across tuning parameters:

k  Accuracy   Kappa
5  0.9750000  0.9625
7  0.9666667  0.9500
9  0.9833333  0.9750

Accuracy was used to select the optimal model using the largest
```

```
value.
The final value used for the model was k = 9.

> print(fit.svm) # PRINT THE SUMMARY OF THE SVM MODEL
Support Vector Machines with Radial Basis Function Kernel

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 across tuning parameters:

C     Accuracy  Kappa
0.25  0.950     0.9250
0.50  0.975     0.9625
1.00  0.975     0.9625

Tuning parameter 'sigma' was held constant at a value of
0.7667022
Accuracy was used to select the optimal model using the largest
value.
The final values used for the model were sigma = 0.7667022 and
C = 0.5.

> print(fit.rf) # PRINT THE SUMMARY OF THE RF MODEL
Random Forest

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 across tuning parameters:

mtry  Accuracy   Kappa
2     0.9583333  0.9375
3     0.9666667  0.9500
4     0.9666667  0.9500

Accuracy was used to select the optimal model using the largest
value.
The final value used for the model was mtry = 3.

### EVALUATE EVERY MODEL ON TEST DATA ###
### LINEAR DISCRIMINANT ANALYSIS ###
> predictions <- predict(fit.lda, test) # PREDICT THE TEST DATA
USING THE LDA MODEL
> confusionMatrix(predictions, test$Species) # CREATE A
CONFUSION MATRIX OF THE PREDICTIONS
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

### CLASSIFICATION AND REGRESSION TREES ###
> predictions <- predict(fit.cart, test) # PREDICT THE TEST
DATA USING THE CART MODEL
> confusionMatrix(predictions, test$Species) # CREATE A
CONFUSION MATRIX OF THE PREDICTIONS
Confusion Matrix and Statistics

            Reference
Prediction   setosa versicolor virginica
  setosa         10          0         0
  versicolor      0         10         2
  virginica       0          0         8

Overall Statistics

               Accuracy : 0.9333
                 95% CI : (0.7793, 0.9918)
No Information Rate : 0.3333
P-Value [Acc > NIR] : 8.747e-12

Kappa : 0.9

Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: setosa Class: versicolor Class:
virginica
Sensitivity                 1.0000            1.0000
```

```
0.8000
Specificity                 1.0000            0.9000
1.0000
Pos Pred Value              1.0000            0.8333
1.0000
Neg Pred Value              1.0000            1.0000
0.9091
Prevalence                  0.3333            0.3333
0.3333
Detection Rate              0.3333            0.3333
0.2667
Detection Prevalence        0.3333            0.4000
0.2667
Balanced Accuracy           1.0000            0.9500
0.9000

### K-NEAREST NEIGHBORS ###
> predictions <- predict(fit.knn, test) # PREDICT THE TEST DATA
USING THE KNN MODEL
> confusionMatrix(predictions, test$Species) # CREATE A
CONFUSION MATRIX OF THE PREDICTIONS
Confusion Matrix and Statistics

            Reference
Prediction   setosa versicolor virginica
  setosa         10          0         0
  versicolor      0         10         1
  virginica       0          0         9

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            1.0000
0.9000
Specificity                 1.0000            0.9500
1.0000
Pos Pred Value              1.0000            0.9091
1.0000
Neg Pred Value              1.0000            1.0000
0.9524
Prevalence                  0.3333            0.3333
0.3333
Detection Rate              0.3333            0.3333
0.3000
Detection Prevalence        0.3333            0.3667
0.3000
Balanced Accuracy           1.0000            0.9750
0.9500
```

Também se concluí que o modelo mais sensível e com menos falsos positivos e negativos é o k-nearest-neighbours. Ou seja, tal modelo será utilizado pela empresa, para o negócio previsto.

```
### SUPPORT VECTOR MACHINES ###
> predictions <- predict(fit.svm, test) # PREDICT THE TEST DATA
USING THE SVM MODEL
> confusionMatrix(predictions, test$Species) # CREATE A
CONFUSION MATRIX OF THE PREDICTIONS
Confusion Matrix and Statistics

Reference
Prediction   setosa versicolor virginica
setosa         10          0         0
versicolor      0          9         2
virginica       0          1         8

Overall Statistics

Accuracy : 0.9
95% CI : (0.7347, 0.9789)
    No Information Rate : 0.3333
    P-Value [Acc > NIR] : 1.665e-10

                  Kappa : 0.85

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: setosa Class: versicolor Class:
virginica
Sensitivity                 1.0000            0.9000
0.8000
Specificity                 1.0000            0.9000
0.9500
Pos Pred Value              1.0000            0.8182
0.8889
Neg Pred Value              1.0000            0.9474
0.9048
Prevalence                  0.3333            0.3333
0.3333
Detection Rate              0.3333            0.3000
0.2667
Detection Prevalence        0.3333            0.3667
0.3000
Balanced Accuracy           1.0000            0.9000
0.8750

### RANDOM FOREST ###
> predictions <- predict(fit.rf, test) # PREDICT THE TEST DATA
USING THE RF MODEL
> confusionMatrix(predictions, test$Species) # CREATE A
CONFUSION MATRIX OF THE PREDICTIONS
Confusion Matrix and Statistics

            Reference
Prediction   setosa versicolor virginica
  setosa         10          0         0
  versicolor      0         10         2
  virginica       0          0         8

Overall Statistics

               Accuracy : 0.9333
                 95% CI : (0.7793, 0.9918)
No Information Rate : 0.3333
P-Value [Acc > NIR] : 8.747e-12

Kappa : 0.9
Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: setosa Class: versicolor Class:
virginica
Sensitivity                 1.0000            1.0000
0.8000
Specificity                 1.0000            0.9000
1.0000
Pos Pred Value              1.0000            0.8333
1.0000
Neg Pred Value              1.0000            1.0000
0.9091
Prevalence                  0.3333            0.3333
0.3333
Detection Rate              0.3333            0.3333
0.2667
Detection Prevalence        0.3333            0.4000
0.2667
Balanced Accuracy           1.0000            0.9500
0.9000
```

6. *Na fase de implementação, o objetivo é colocar os modelos em uso prático no ambiente do negócio. Neste caso, os modelos podem ser usados para classificar novas instâncias de flores Iris que não fazem parte do conjunto de dados original. Os modelos devem ser monitorados e atualizados periodicamente para garantir que continuem produzindo resultados confiáveis e relevantes.* ***Logo, uma possível proposta de implementação de negócio em relação as aplicações médicas das classes setosa e virginica, das flores Iris, e a precaução contra a classe venenosa, versicolor, da flor Iris são a:***
   — ***...exploração empresarial das aplicações médicas das classes setosa e virginica, e a precaução contra a classe versicolor, por parte de um laboratório de análise e produção de produtos naturais derivados das flores Iris. Este laboratório poderia usar o modelo K-Nearest-Neighbours para identificar as espécies das flores coletadas, e extrair os princípios ativos de cada uma delas.*** *Por exemplo, a classe setosa poderia fornecer corantes naturais para cosméticos ou alimentos, a classe virginica poderia fornecer substâncias anti-inflamatórias ou antioxidantes para medicamentos ou suplementos, e a classe versicolor poderia fornecer toxinas ou alérgenos para testes ou vacinas.* ***Tal laboratório poderia vender esses produtos para indústrias***

farmacêuticas, alimentícias ou cosméticas, ou para instituições de pesquisa ou saúde.