Unit3\_LaurenceBurden\_Code.R

larryburd

2023-09-11

##################################################################  
#  
# Author: Laurence Burden  
# IN400 - AI: Deep Learning and Machine Learning  
# Unit 3 Assignment / Modeule 2 Competency Assessment Part 2  
# Machine Learning Classification Using the Iris Dataset with R  
#  
##################################################################  
  
  
# Packages used  
packages <- c("tidyr", "ggplot2", "rpart", "rpart.plot", "glue")  
  
# Install needed packages  
installed\_packages <- packages %in% rownames(installed.packages())  
  
if (any(installed\_packages == FALSE)) {  
 install.packages(packages[!installed\_packages])  
}  
  
# Load packages  
invisible(lapply(packages, library, character.only = TRUE))  
  
# Import libraries  
library("tidyr")  
library("ggplot2")  
library("rpart")  
library("rpart.plot")  
  
# Load dataset  
data(iris)  
iris\_dataset <- iris  
  
# Explore data by peeking at it and its structure  
head(iris\_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  
## 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

dim(iris\_dataset)

## [1] 150 5

str(iris\_dataset)

## 'data.frame': 150 obs. of 5 variables:  
## $ Sepal.Length: num 5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...  
## $ Sepal.Width : num 3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...  
## $ Petal.Length: num 1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...  
## $ Petal.Width : num 0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...  
## $ Species : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1 ...

summary(iris\_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.:2.800 1st Qu.:1.600 1st Qu.:0.300   
## Median :5.800 Median :3.000 Median :4.350 Median :1.300   
## Mean :5.843 Mean :3.057 Mean :3.758 Mean :1.199   
## 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   
## Species   
## setosa :50   
## versicolor:50   
## virginica :50   
##   
##   
##

# Assign meaningful column names based on dataset repository  
colnames(iris\_dataset) <- c("Sepal.Length",  
 "Sepal.Width",  
 "Petal.Length",  
 "Petal.Width",  
 "Species")  
head(iris\_dataset)

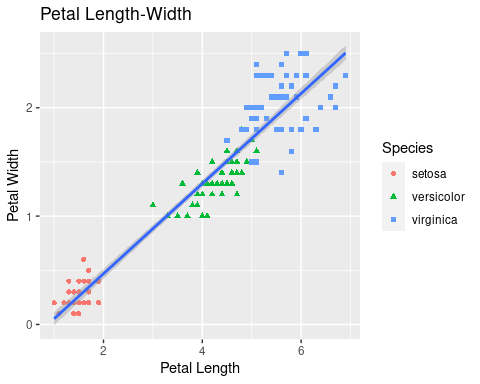
## Sepal.Length Sepal.Width Petal.Length Petal.Width Species  
## 1 5.1 3.5 1.4 0.2 setosa  
## 2 4.9 3.0 1.4 0.2 setosa  
## 3 4.7 3.2 1.3 0.2 setosa  
## 4 4.6 3.1 1.5 0.2 setosa  
## 5 5.0 3.6 1.4 0.2 setosa  
## 6 5.4 3.9 1.7 0.4 setosa

# Check classification number for flowers  
levels(iris\_dataset)

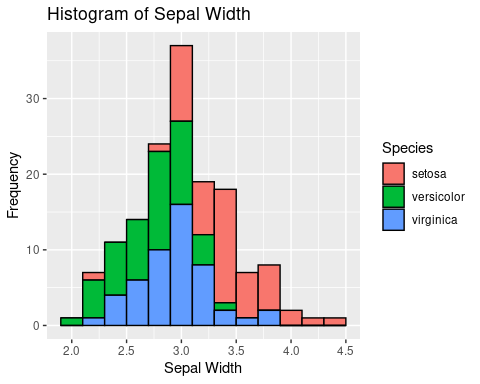
## NULL

# Visualizations  
# Scatter Plot  
sp <- ggplot(data = iris\_dataset,  
 aes(x = Petal.Length, y = Petal.Width)) +  
 geom\_point(aes(color = Species, shape = Species)) +  
 xlab("Petal Length") +  
 ylab("Petal Width") +  
 ggtitle("Petal Length-Width") +  
 geom\_smooth(method="lm")  
print(sp)

## `geom\_smooth()` using formula = 'y ~ x'



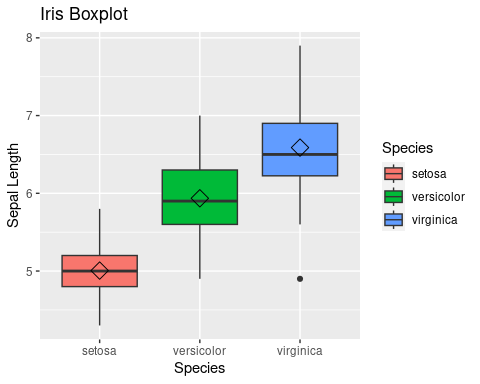
# Histogram  
histogram <- ggplot(iris\_dataset,  
 aes(x = Sepal.Width)) +  
 geom\_histogram(binwidth = 0.2, color = "black",  
 aes(fill = Species)) +  
 xlab("Sepal Width") +  
 ylab("Frequency") +  
 ggtitle("Histogram of Sepal Width")  
  
print(histogram)



# Box plot  
box <- ggplot(data = iris\_dataset,  
 aes(x = Species, y = Sepal.Length)) +  
 geom\_boxplot(aes(fill = Species)) +  
 ylab("Sepal Length") +  
 ggtitle("Iris Boxplot") +  
 stat\_summary(fun.y = mean, geom = "point", shape = 5, size = 4)

## Warning: The `fun.y` argument of `stat\_summary()` is deprecated as of ggplot2 3.3.0.  
## ℹ Please use the `fun` argument instead.

print(box)



# Classification  
# First split data in 80-20 train/test data  
n <- seq\_len(nrow(iris\_dataset))  
index <- sample(n, length(n) \* 0.8)  
trainset <- iris\_dataset[index,]  
testset <- iris\_dataset[-index,]  
  
# Peek at the training data  
head(trainset)

## Sepal.Length Sepal.Width Petal.Length Petal.Width Species  
## 88 6.3 2.3 4.4 1.3 versicolor  
## 99 5.1 2.5 3.0 1.1 versicolor  
## 87 6.7 3.1 4.7 1.5 versicolor  
## 92 6.1 3.0 4.6 1.4 versicolor  
## 64 6.1 2.9 4.7 1.4 versicolor  
## 9 4.4 2.9 1.4 0.2 setosa

dim(trainset)

## [1] 120 5

str(trainset)

## 'data.frame': 120 obs. of 5 variables:  
## $ Sepal.Length: num 6.3 5.1 6.7 6.1 6.1 4.4 5.2 6.1 7.4 7.9 ...  
## $ Sepal.Width : num 2.3 2.5 3.1 3 2.9 2.9 2.7 3 2.8 3.8 ...  
## $ Petal.Length: num 4.4 3 4.7 4.6 4.7 1.4 3.9 4.9 6.1 6.4 ...  
## $ Petal.Width : num 1.3 1.1 1.5 1.4 1.4 0.2 1.4 1.8 1.9 2 ...  
## $ Species : Factor w/ 3 levels "setosa","versicolor",..: 2 2 2 2 2 1 2 3 3 3 ...

# Peek at test data  
head(testset)

## Sepal.Length Sepal.Width Petal.Length Petal.Width Species  
## 2 4.9 3.0 1.4 0.2 setosa  
## 11 5.4 3.7 1.5 0.2 setosa  
## 16 5.7 4.4 1.5 0.4 setosa  
## 25 4.8 3.4 1.9 0.2 setosa  
## 31 4.8 3.1 1.6 0.2 setosa  
## 34 5.5 4.2 1.4 0.2 setosa

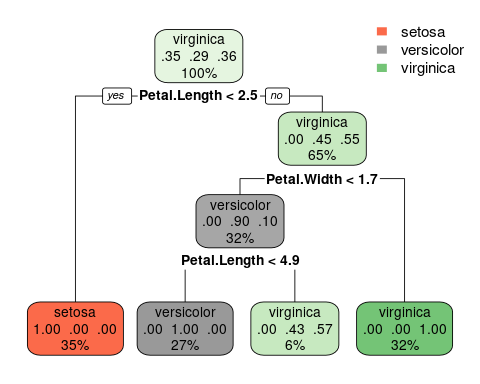
dim(testset)

## [1] 30 5

str(testset)

## 'data.frame': 30 obs. of 5 variables:  
## $ Sepal.Length: num 4.9 5.4 5.7 4.8 4.8 5.5 4.5 4.8 7 5 ...  
## $ Sepal.Width : num 3 3.7 4.4 3.4 3.1 4.2 2.3 3 3.2 2 ...  
## $ Petal.Length: num 1.4 1.5 1.5 1.9 1.6 1.4 1.3 1.4 4.7 3.5 ...  
## $ Petal.Width : num 0.2 0.2 0.4 0.2 0.2 0.2 0.3 0.3 1.4 1 ...  
## $ Species : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 2 2 ...

# Build classification model using a decision tree classifier  
# First, fit the data  
fit <- rpart(Species~., data = trainset, method = 'class')  
print(rpart.plot(fit))



## $obj  
## n= 120   
##   
## node), split, n, loss, yval, (yprob)  
## \* denotes terminal node  
##   
## 1) root 120 77 virginica (0.3500000 0.2916667 0.3583333)   
## 2) Petal.Length< 2.45 42 0 setosa (1.0000000 0.0000000 0.0000000) \*  
## 3) Petal.Length>=2.45 78 35 virginica (0.0000000 0.4487179 0.5512821)   
## 6) Petal.Width< 1.65 39 4 versicolor (0.0000000 0.8974359 0.1025641)   
## 12) Petal.Length< 4.85 32 0 versicolor (0.0000000 1.0000000 0.0000000) \*  
## 13) Petal.Length>=4.85 7 3 virginica (0.0000000 0.4285714 0.5714286) \*  
## 7) Petal.Width>=1.65 39 0 virginica (0.0000000 0.0000000 1.0000000) \*  
##   
## $snipped.nodes  
## NULL  
##   
## $xlim  
## [1] 0 1  
##   
## $ylim  
## [1] 0 1  
##   
## $x  
## [1] 0.39936557 0.09746226 0.70126887 0.50000000 0.36582075 0.63417925 0.90253774  
##   
## $y  
## [1] 0.90919253 0.05753979 0.65111594 0.39303935 0.05753979 0.05753979 0.05753979  
##   
## $branch.x  
## [,1] [,2] [,3] [,4] [,5] [,6] [,7]  
## x 0.3993656 0.09746226 0.7012689 0.5000000 0.3658208 0.6341792 0.9025377  
## NA 0.09746226 0.7012689 0.5000000 0.3658208 0.6341792 0.9025377  
## NA 0.39936557 0.3993656 0.7012689 0.5000000 0.5000000 0.7012689  
##   
## $branch.y  
## [,1] [,2] [,3] [,4] [,5] [,6] [,7]  
## y 1.008005 0.1563523 0.7499284 0.4918519 0.1563523 0.1563523 0.1563523  
## NA 0.7998175 0.7998175 0.5417409 0.2836644 0.2836644 0.5417409  
## NA 0.7998175 0.7998175 0.5417409 0.2836644 0.2836644 0.5417409  
##   
## $labs  
## [1] "virginica\n.35 .29 .36\n100%" "setosa\n1.00 .00 .00\n35%"   
## [3] "virginica\n.00 .45 .55\n65%" "versicolor\n.00 .90 .10\n32%"   
## [5] "versicolor\n.00 1.00 .00\n27%" "virginica\n.00 .43 .57\n6%"   
## [7] "virginica\n.00 .00 1.00\n32%"   
##   
## $cex  
## [1] 0.865625  
##   
## $boxes  
## $boxes$x1  
## [1] 0.29175687 -0.01992904 0.59366017 0.39239130 0.24842945 0.52657055  
## [7] 0.78514643  
##   
## $boxes$y1  
## [1] 0.84163003 -0.01002271 0.58355344 0.32547685 -0.01002271 -0.01002271  
## [7] -0.01002271  
##   
## $boxes$x2  
## [1] 0.5069743 0.2148536 0.8088776 0.6076087 0.4832121 0.7417879 1.0199290  
##   
## $boxes$y2  
## [1] 1.0080050 0.1563523 0.7499284 0.4918519 0.1563523 0.1563523 0.1563523  
##   
##   
## $split.labs  
## [1] ""  
##   
## $split.cex  
## [1] 1 1 1 1 1 1 1  
##   
## $split.box  
## $split.box$x1  
## [1] 0.2391753 NA 0.5496384 0.3398098 NA NA NA  
##   
## $split.box$y1  
## [1] 0.7685675 NA 0.5104909 0.2524144 NA NA NA  
##   
## $split.box$x2  
## [1] 0.5595558 NA 0.8528993 0.6601902 NA NA NA  
##   
## $split.box$y2  
## [1] 0.8310675 NA 0.5729909 0.3149144 NA NA NA

# Use the fit to perform predictions on the training set  
pred\_train <- predict(object = fit, newdata = trainset[,1:4], type = "class")  
print(pred\_train)

## 88 99 87 92 64 9 60   
## versicolor versicolor versicolor versicolor versicolor setosa versicolor   
## 128 131 132 33 112 43 76   
## virginica virginica virginica setosa virginica setosa versicolor   
## 27 104 97 127 22 41 75   
## setosa virginica versicolor virginica setosa setosa versicolor   
## 125 118 82 73 80 103 3   
## virginica virginica versicolor virginica versicolor virginica setosa   
## 148 74 65 68 18 57 111   
## virginica versicolor versicolor versicolor setosa versicolor virginica   
## 137 39 48 32 141 102 45   
## virginica setosa setosa setosa virginica virginica setosa   
## 150 12 59 84 40 140 55   
## virginica setosa versicolor virginica setosa virginica versicolor   
## 107 58 98 122 50 106 110   
## virginica versicolor versicolor virginica setosa virginica virginica   
## 121 30 130 96 81 116 115   
## virginica setosa virginica versicolor versicolor virginica virginica   
## 44 6 35 67 144 119 15   
## setosa setosa setosa versicolor virginica virginica setosa   
## 139 91 7 69 126 94 24   
## virginica versicolor setosa versicolor virginica versicolor setosa   
## 37 19 5 4 114 62 149   
## setosa setosa setosa setosa virginica versicolor virginica   
## 20 10 29 21 123 38 17   
## setosa setosa setosa setosa virginica setosa setosa   
## 23 109 47 142 14 26 52   
## setosa virginica setosa virginica setosa setosa versicolor   
## 113 120 49 36 54 83 13   
## virginica virginica setosa setosa versicolor versicolor setosa   
## 77 129 133 124 53 135 28   
## versicolor virginica virginica virginica virginica virginica setosa   
## 108 56 136 134 1 8 138   
## virginica versicolor virginica virginica setosa setosa virginica   
## 85   
## versicolor   
## Levels: setosa versicolor virginica

# Metrics  
# Determine accuracy metrics with a confusion matrix  
print(table(pred\_train, trainset$Species))

##   
## pred\_train setosa versicolor virginica  
## setosa 42 0 0  
## versicolor 0 32 0  
## virginica 0 3 43

# Use the fit to perform predictions on the test data  
pred\_test <- predict(object = fit, newdata = testset[,1:4], type = "class")  
print(pred\_test)

## 2 11 16 25 31 34 42   
## setosa setosa setosa setosa setosa setosa setosa   
## 46 51 61 63 66 70 71   
## setosa versicolor versicolor versicolor versicolor versicolor virginica   
## 72 78 79 86 89 90 93   
## versicolor virginica versicolor versicolor versicolor versicolor versicolor   
## 95 100 101 105 117 143 145   
## versicolor versicolor virginica virginica virginica virginica virginica   
## 146 147   
## virginica virginica   
## Levels: setosa versicolor virginica

# Confusion matrix on test data  
print(table(pred\_test, testset$Species))

##   
## pred\_test setosa versicolor virginica  
## setosa 8 0 0  
## versicolor 0 13 0  
## virginica 0 2 7