Rensselaer

why not change the world?®

Lab exercises: Naïve Bayes, kNN & k-Means, Model Training, Prediction and Evaluation

Ahmed Eleish

ITWS-4600/ITWS-6600/MATP-4450/CSCI-4960 Group 1, Lab 1, September 27th, 2024

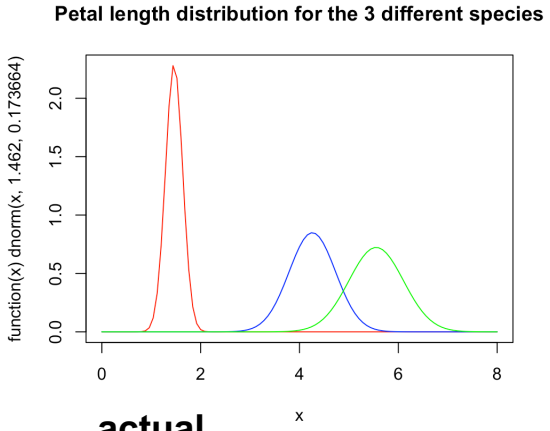
Tetherless World Constellation
Rensselaer Polytechnic Institute

Lab 02 – part 1 review

Lab 02 - part 2

Files:
https://rpi.box.com/s/vg2rrl65qp7xruegyq3c06lzsvmg8j6v

Naïve Bayes




```
## Call the NaiveBayes Classifier Package e1071, which auto calls the Class package ##
library("e1071")

#Train classifier
classifier<-naiveBayes(iris[,1:4], iris[,5])

# evaluate classification
table(predict(classifier, iris[,-5]), iris[,5], dnn=list('predicted','actual'))

# examine class means and standard deviations for petal length
classifier$tables$Petal.Length

# plot normal distributions at the means of the classes

# one class
plot(function(x) dnorm(x, 1.462, 0.1736640), 0, 8, col="red", main="Petal length distribution for the 3 different species")

# another class
curve(dnorm(x, 4.260, 0.4699110), add=TRUE, col="blue")

# the final class
curve(dnorm(x, 5.552, 0.5518947 ), add=TRUE, col = "green")
```

	actual		
predicted	setosa	versicolor	virginica
setosa	50	0	0
versicolor	0	47	3
virginica	0	3	47

Exercise 1:

- Repeat the naïve bayes analysis using the abalone dataset.
- Try 3 different subsets of features not just all features at once.
- Compare models using contingency tables.
- Plot the distribution of classes along 3 different features.

K-Nearest Neighbors

```
# read dataset
abalone <- read.csv(url("https://archive.ics.uci.edu/ml/machine-learning-databases/abalone/abalone.data"), header = FALSE, sep = ",")

# rename columns
colnames(abalone) <- c("sex", "length", 'diameter', 'height', 'whole_weight', 'shucked_wieght', 'viscera_wieght', 'shell_weight', 'rings' )

# add new column abalone$age.group with 3 values based on the number of rings
abalone$age.group <- cut(abalone$rings, br=c(0,8,11,35), labels = c("young", 'adult', 'old'))

# drop the sex column (categorical variable)
abalone.norm <- abalone[,-1]

# optionally normalize
#normalize <- function(x) {return ((x - min(x)) / (max(x) - min(x))) }
#abalone.norm[1:7] <- as.data.frame(lapply(abalone.norm[1:7], normalize))
```

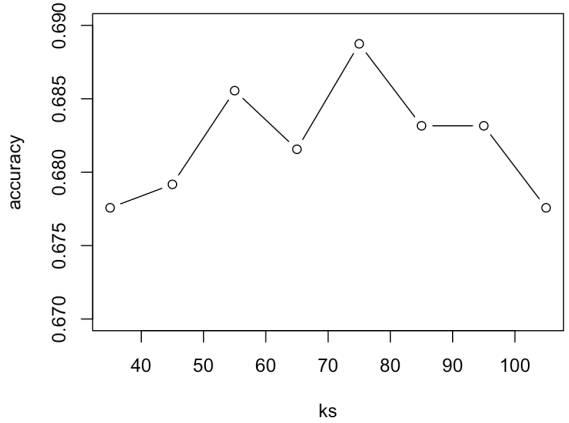
K-Nearest Neighbors

```
# sample 2924 from 4177 (~70%)
s_abalone <- sample(4177,2924)

#Abalone.norm.train <-abalone.norm[s_abalone,]
#abalone.norm.test <-abalone.norm[-s_abalone,]

## create train & test sets based on sampled indexes
abalone.train <-abalone[s_abalone,]
abalone.test <-abalone[-s_abalone,]

sqrt(2924)
k = 55
# k = 80

# train model & predict
KNNpred <- knn(train = abalone.train[1:7], test = abalone.test[1:7], cl = abalone.train$age.group, k = k)

# create contingency table/ confusion matrix
contingency.table  <- table(KNNpred,abalone.test$age.group)
```

	actual		
predicted	**young**	**adult**	**old**
young	**338**	**94**	**18**
adult	**97**	**417**	**154**
old	**2**	**28**	**105**

K-Nearest Neighbors

```
contingency.matrix = as.matrix(contingency.table)

sum(diag(contingency.matrix))/length(abalone.test$age.group)

accuracy <- c()
ks <- c(35,45,55,65,75,85,95,105)

for (k in ks) {

  KNNpred <- knn(train = abalone.train[1:7], test = abalone.test[1:7], cl = abalone.train$age.group, k = k)
  cm = as.matrix(table(Actual=KNNpred, Predicted = abalone.test$age.group, dnn=list('predicted','actual')))
  accuracy <- c(accuracy,sum(diag(cm))/length(abalone.test$age.group))
}

plot(ks,accuracy,type = "b", ylim = c(0.67,0.69))
```

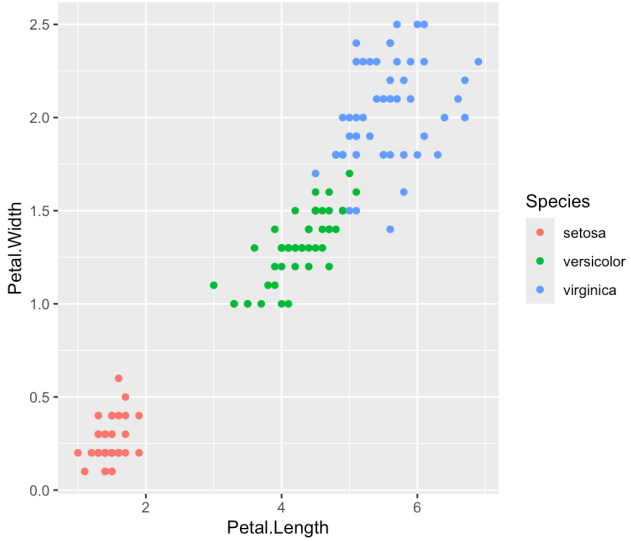


Exercise 2:

- Repeat the kNN analysis using the iris dataset.
- Try 2 different subsets of features.
- Compare models using contingency tables and accuracy plots.

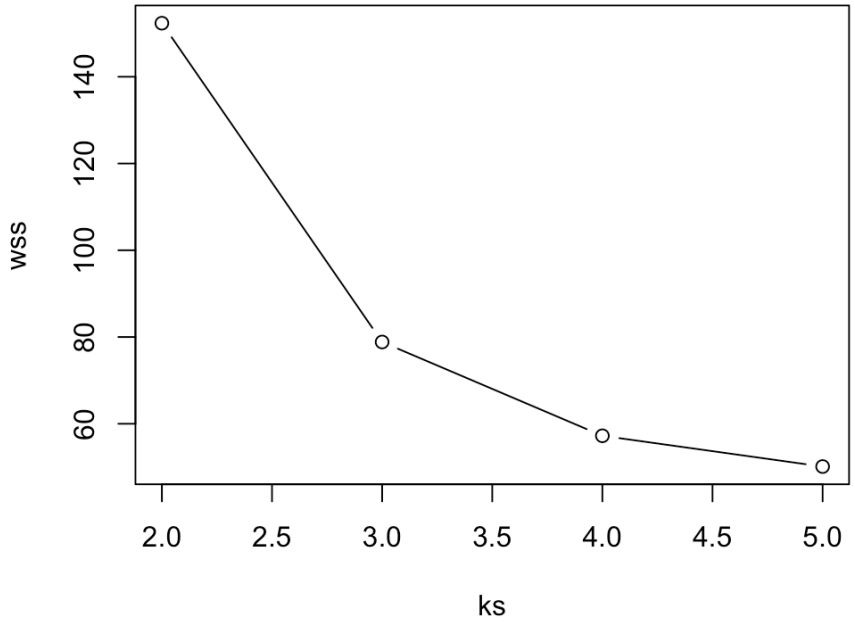
K-Means

```
# Plot iris petal length vs. petal width, color by species
ggplot(iris, aes(x = Petal.Length, y = Petal.Width, colour = Species)) +
  geom_point()

# set seed for random number generator
set.seed(123)

# run k-means
iris.km <- kmeans(iris[,-5], centers = 3)

assigned.clusters <- as.factor(iris.km$cluster)

ggplot(iris, aes(x = Petal.Length, y = Petal.Width, colour = assigned.clusters)) +
  geom_point()
```



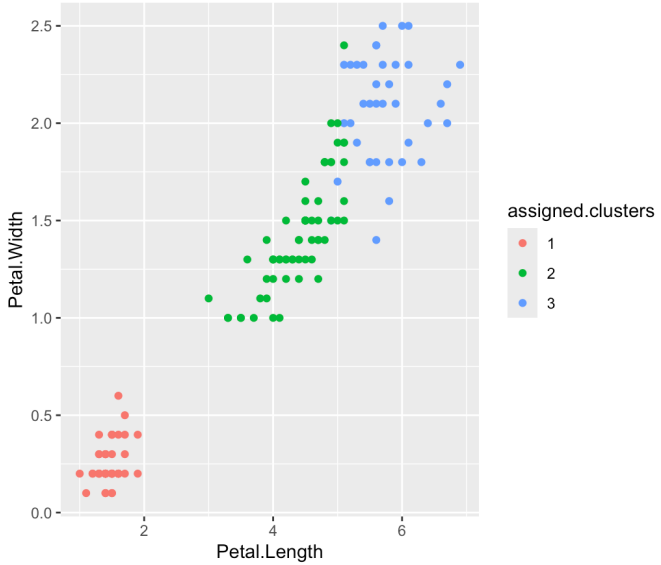
K-Means

```
wss <- c()
ks <- c(2,3,4,5)

for (k in ks) {

  iris.km <- kmeans(iris[,-5], centers = k)

  wss <- c(wss,iris.km$tot.withinss)

}

plot(ks,wss,type = "b")
```



K-Means

```
labeled.clusters <- as.character(assigned.clusters)

labeled.clusters[labeled.clusters==1] <- "setosa"
labeled.clusters[labeled.clusters==2] <- "versivolor"
labeled.clusters[labeled.clusters==3] <- "virginica"


table(labeled.clusters, iris[,5])
```



Exercise 3:

- Run k-means analysis using the abalone & iris datasets.
- Try different values of *k* for both.
- Evaluate clustering using Plot the best clustering output for both.

Please push to your github repository:

1. All your code in a *.R or *.MD file
2. All text outputs (contingency tables)
3. All plots (group colored scatter plots, kNN accuracy plots, k-Means “elbow” plots)

Thanks!
Have a great weekend!*

* Good luck with the job search!!!