# Data Mining Lab, Exercise 3

Team# 1: Poliakov Valerii, Holovnia Dmytro, Selvaraj Sinju

Datasets: f1.csv, zoo.csv

## Task# 1

*(3pt) Perform the calculations using the k-means algorithm for data from file “f#.csv”.*

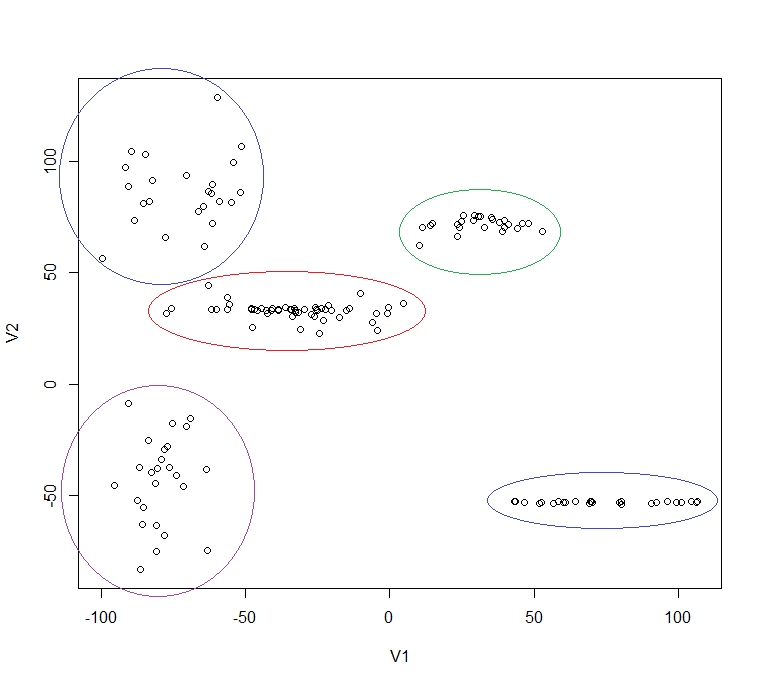
*a) Decide on the most likely number of clusters on the basis of the scatter plot of the data. For the chosen number of clusters, perform clustering using the k-means algorithm. Repeat the calculations 5 times. Write the values of SSB/TSS for each of the algorithm runs. Indicate the minimal and maximal value of SSB/TSS as well as its average value. Show the best solution (clusters) in the scatter plot. Indicate the centroids.*

Step 1. Read the data. Draw plot and guess number of clusters.

d<-read.csv(file="f1.csv", header = FALSE, sep=' ')

d

with(d, plot(V1, V2))



Dataset in the f1.csv file has no headers and whitespace used as a separator. Plot shows five distinct groups. Most likely there are 5 clusters.

Step 2. Repeat 5 times k-means clustering for five clusters. Note the SBB/TSS values for each iteration. SBB (Sum of squares between clusters) and TSS (Total Sum of Squares) are used as measures to evaluate the performance of the algorithm.

Script command:

km<-kmeans(d, centers=5)

Iteration 1:

Within cluster sum of squares by cluster:

[1] 10670.589 18327.261 10260.915 3436.462 10864.708

(between\_SS / total\_SS = 94.1 %)

Iteration 2:

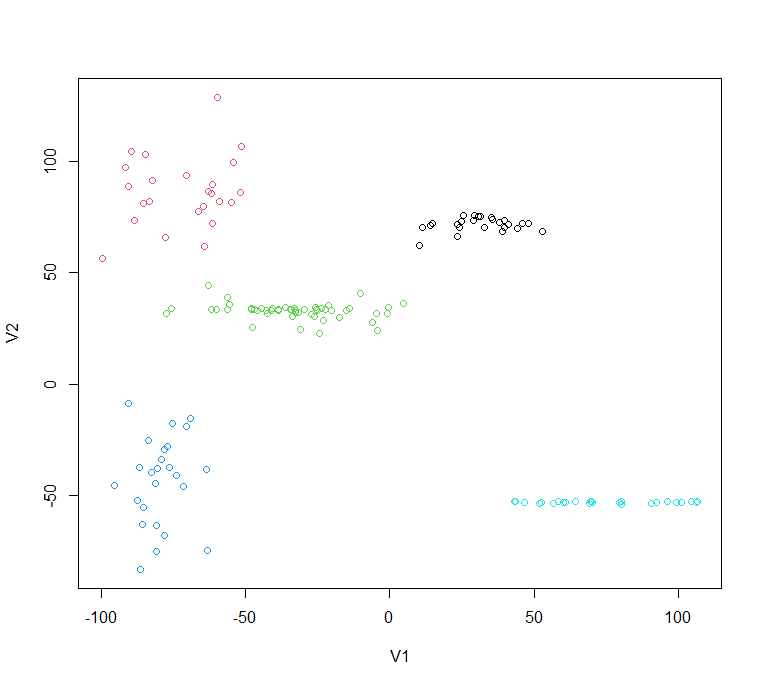
Within cluster sum of squares by cluster:

[1] 3436.462 10670.589 18327.261 10864.708 10260.915

(between\_SS / total\_SS = 94.1 %)

with(d, plot(V1, V2, col=km$cluster))

With the quality measure 94.1% diagram shows expected clusters



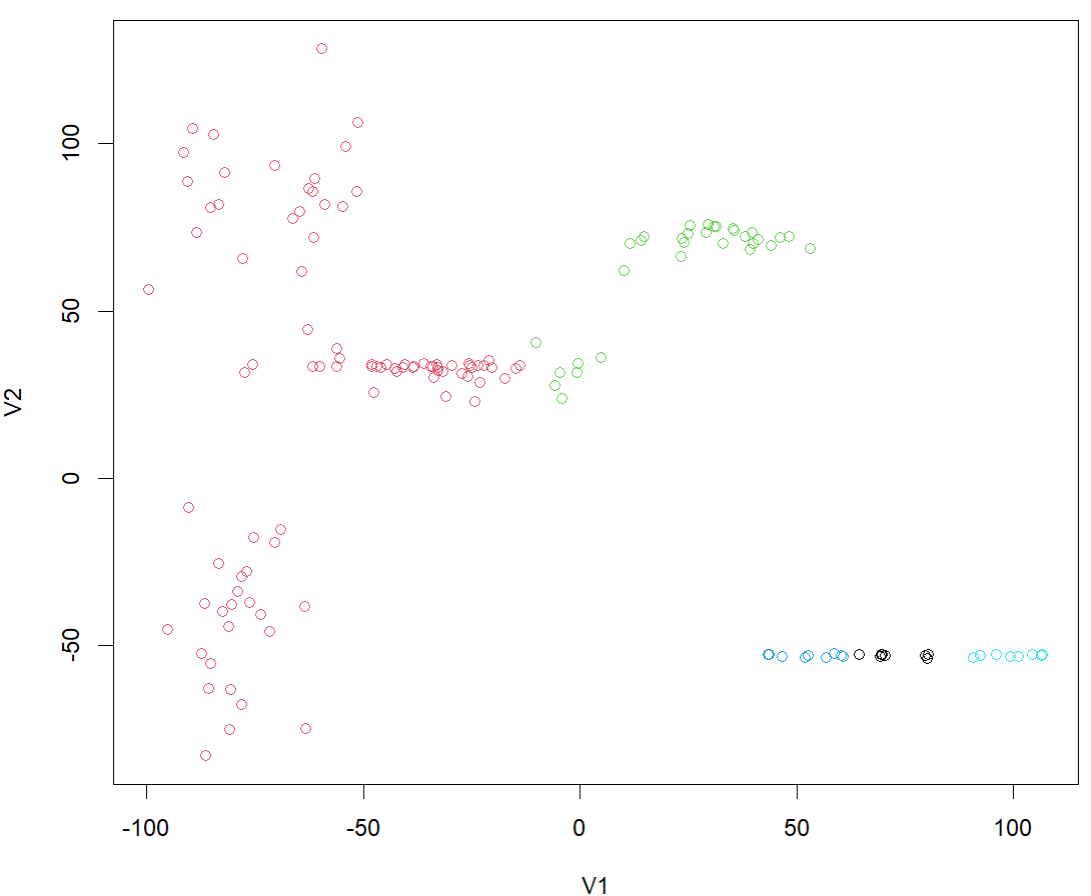
Iteration 3:

Within cluster sum of squares by cluster:

[1] 266.0465 274338.3389 18684.9885 385.3310 267.4781

(between\_SS / total\_SS = 67.9 %)

Quality measure is 67.9 which is worse than iteration 1 and 2. The diagram shows that algorithm identified 3 clusters in the right-bottom corner (where supposed to be only one) and remaining data split between remaining two clusters. For the greater number of clusters this might make sense, but we have only 5 so result is not very good.



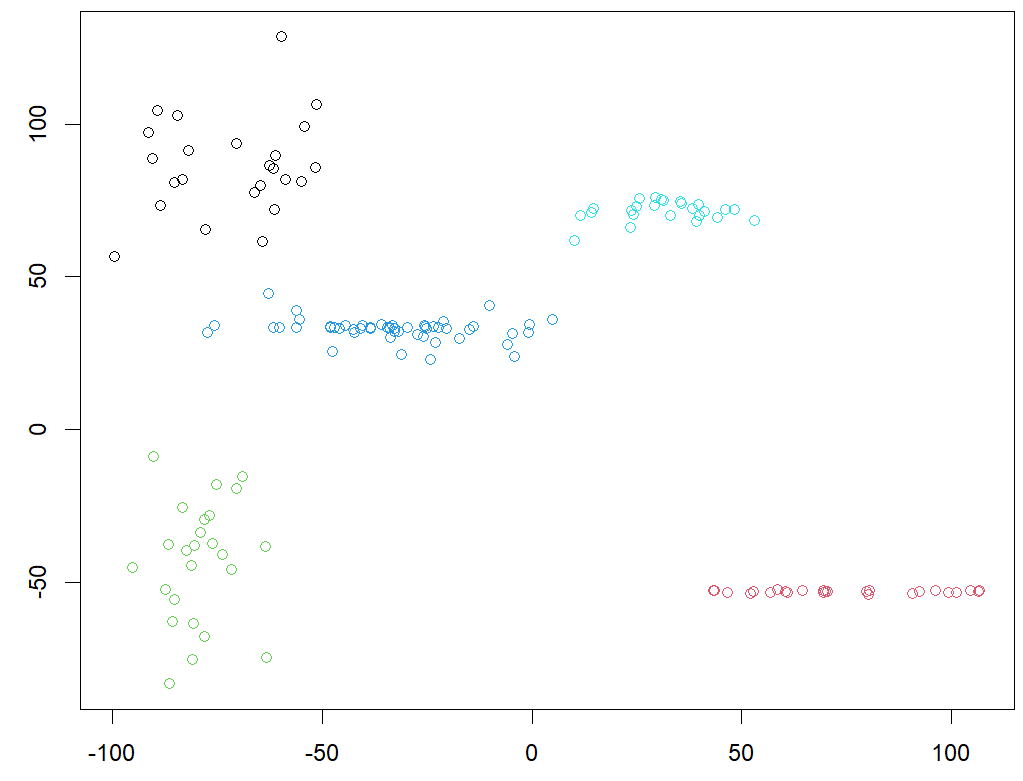
Iteration 4:

Within cluster sum of squares by cluster:

[1] 10670.589 10260.915 10864.708 18327.261 3436.462

(between\_SS / total\_SS = 94.1 %)

Again, very similar to Iteration 1 and 2



Iteration 5:

Within cluster sum of squares by cluster:

[1] 10670.589 10260.915 3436.462 10864.708 18327.261

(between\_SS / total\_SS = 94.1 %)

And last iteration also shows similar result.

Summary table for all 5 iterations

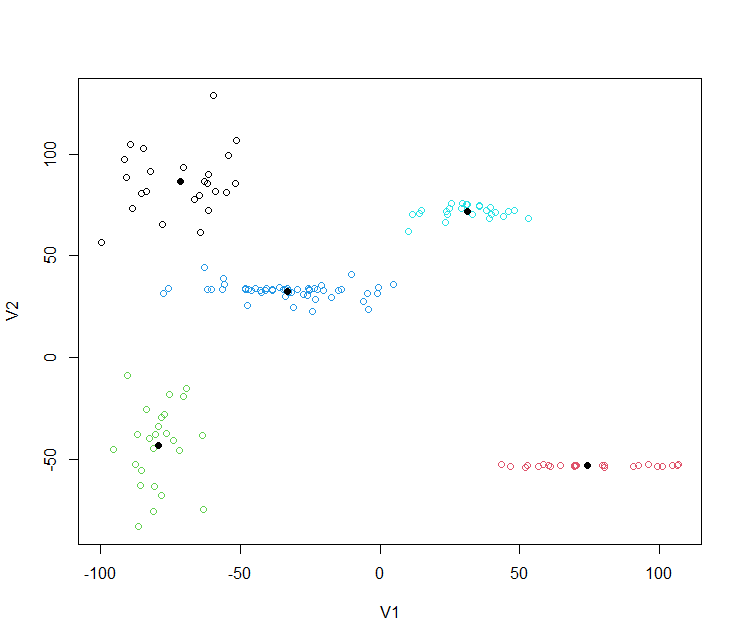
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Iteration 1 | Iteration 2 | Iteration 3 | Iteration 4 | Iteration 5 |
| SSB/TSS | 94.1% | 94.1% | 67.9% | 94.1% | 94.1% |

Minimal value: 67.9%

Maximal value: 94.1%

Average value: 88.86%

The solution with centroids shown on the diagram below.



*b) Identify the number of clusters using the Elbow Method. Draw the chart that shows how WSS depends on k. Compare the results with the results of Task 1a.*

Step 3. Use Elbow Method to identify number of clusters.

Script performs N iterations of k-means algorithm with number of clusters equal current iteration number from 1 to N. Algorithm runs K times in each iteration.

N <- 15 # Number of iterations

K <- 5 # Number of algorithm runs in each iteration

results <- list()

for (i in 1:N) {

km <- kmeans(d, centers = i, nstart = K)

results[[i]] <- km

}

wss <- sapply(results, function(x) x$tot.withinss)

wss

plot(wss, type = "b", xlab = "Iteration", ylab = "WSS")

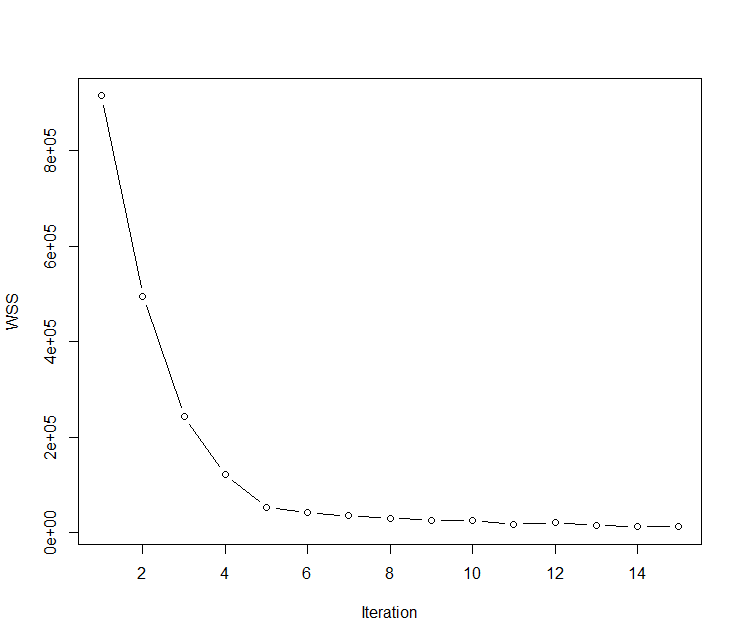
WSS in each iteration are:

[1] 914637.42 [2] 494055.40 [3] 244007.56 [4] 123037.13 [5] 53559.94

[6] 42632.39 [7] 35810.32 [8] 31048.06 [9] 26675.12 [10] 26372.47

[11] 18161.88 [12] 21246.19 [13] 15605.51 [14] 12704.59 [15] 13443.24

And the chart is on the diagram below. On the diagram, we see that after 5 clusters, the WSS improves very slowly, so our initial assumption about 5 clusters seems correct.



Additionally, we can compare scatter plot for different number of clusters.

# plot 5th iteration result

with(d, plot(V1, V2, col=results[[5]]$cluster))

points(results[[5]]$centers, col=1,pch=16,cex=1)

|  |  |
| --- | --- |
| 5 clusters | 6 clusters |
| 7 clusters | 14 clusters |

On the scatter plot representation, we see that 7 clusters might make sense but not very different from 5 clusters. At the same time, 14 cluster definitely looks very redundant.

## Task 2:

*File “zoo.csv” contains variables describing 7 types of animals: mammal, fish, bird,*

*invertebrate, insect, amphibian and reptile. Determine clusters in data from “zoo.csv” using the*

*k-means algorithm. Compare your results with the known classification presented in file*

*“zoo\_full.xlsx” (see the last column “type”). Which of the animals were often misclassified?*

Solution:

R script finds 7 clusters in zoo data, binds this information to the dataset and writes new file zoo\_clustered.csv for further analysis.

d <- read.csv("zoo.csv", header = TRUE, sep = ",")

d

km <- kmeans(d, centers=7, nstart=10)

km

Clustering vector:

[1] 4 4 5 4 4 4 4 5 5 4 4 1 5 5 7 2 1 4 5 5 1 1 4 1 2 7 7 3 4 3 2 4 3 1 5 4 4 1 5 2 2 1 2 1 4 4 2 4 4 4 4 2 7 6 4 4 1 1 1 1

[61] 5 5 5 4 4 4 5 4 4 4 4 1 6 5 5 3 5 5 1 1 5 5 5 1 3 7 5 1 2 7 7 7 5 3 4 1 3 2 4 5 1

Within cluster sum of squares by cluster:

[1] 19.100000 12.600000 8.571429 24.451613 44.608696 3.000000 12.375000

(between\_SS / total\_SS = 82.2 %)

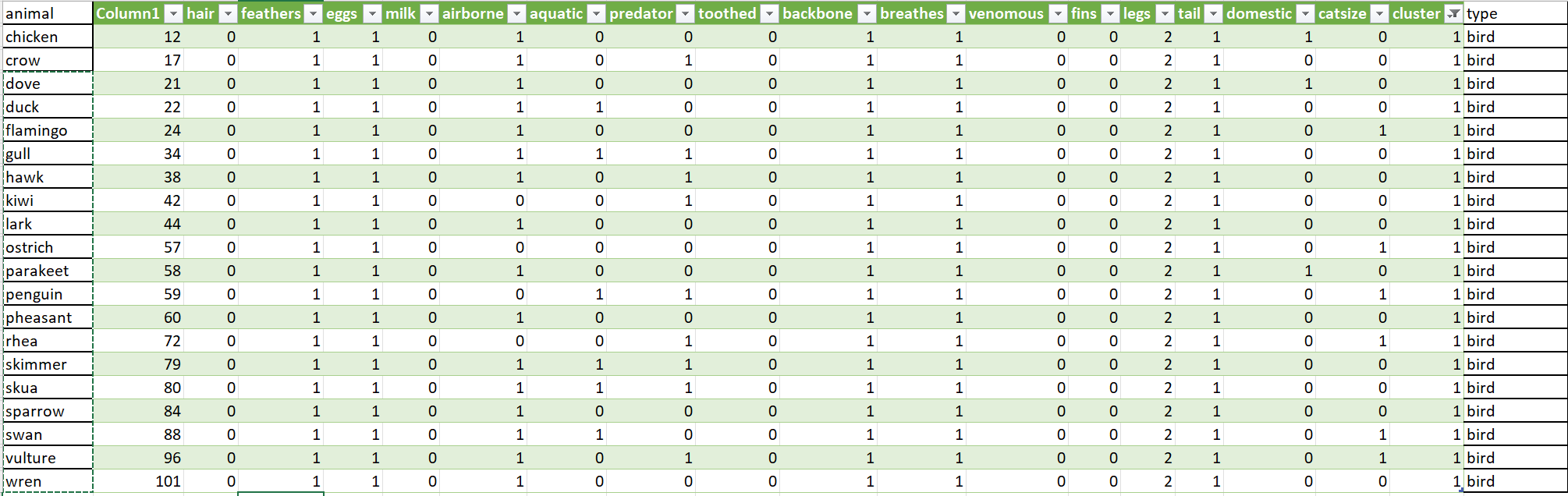
d\_clustered <- cbind(d, cluster=km$cluster)

write.csv(d\_clustered, "zoo\_clustered.csv")

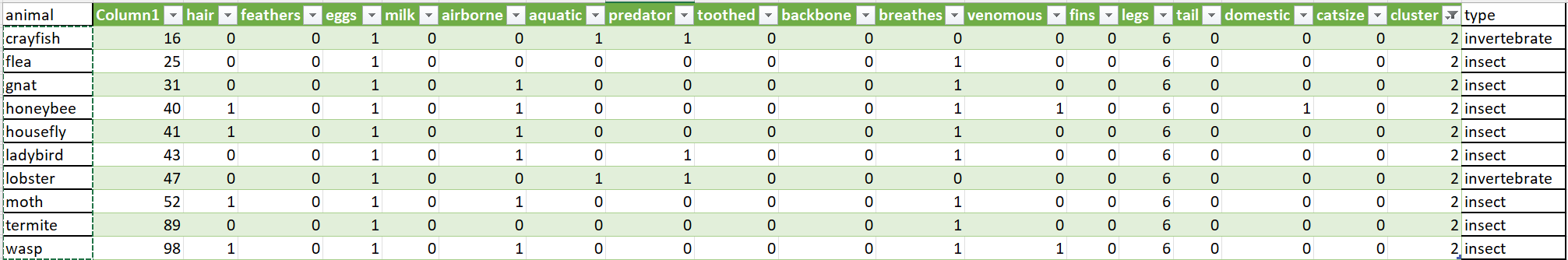
File zoo\_clustered.csv imported to Excel and two columns (animal and type) from zoo\_full.xlsx added to zoo\_clustered data to compare results.

Clusters are:

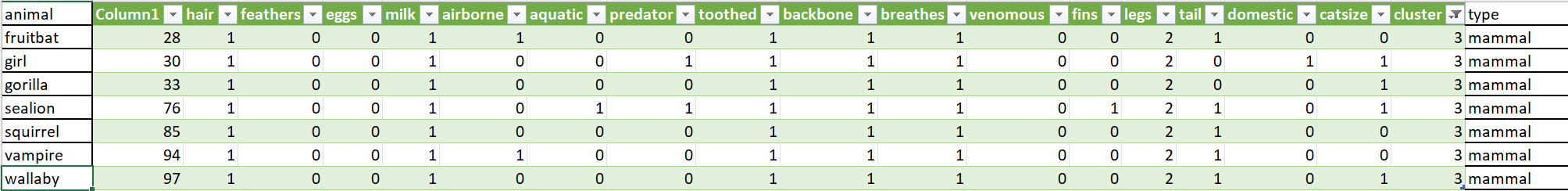
Cluster 1 – birds, seems correct.



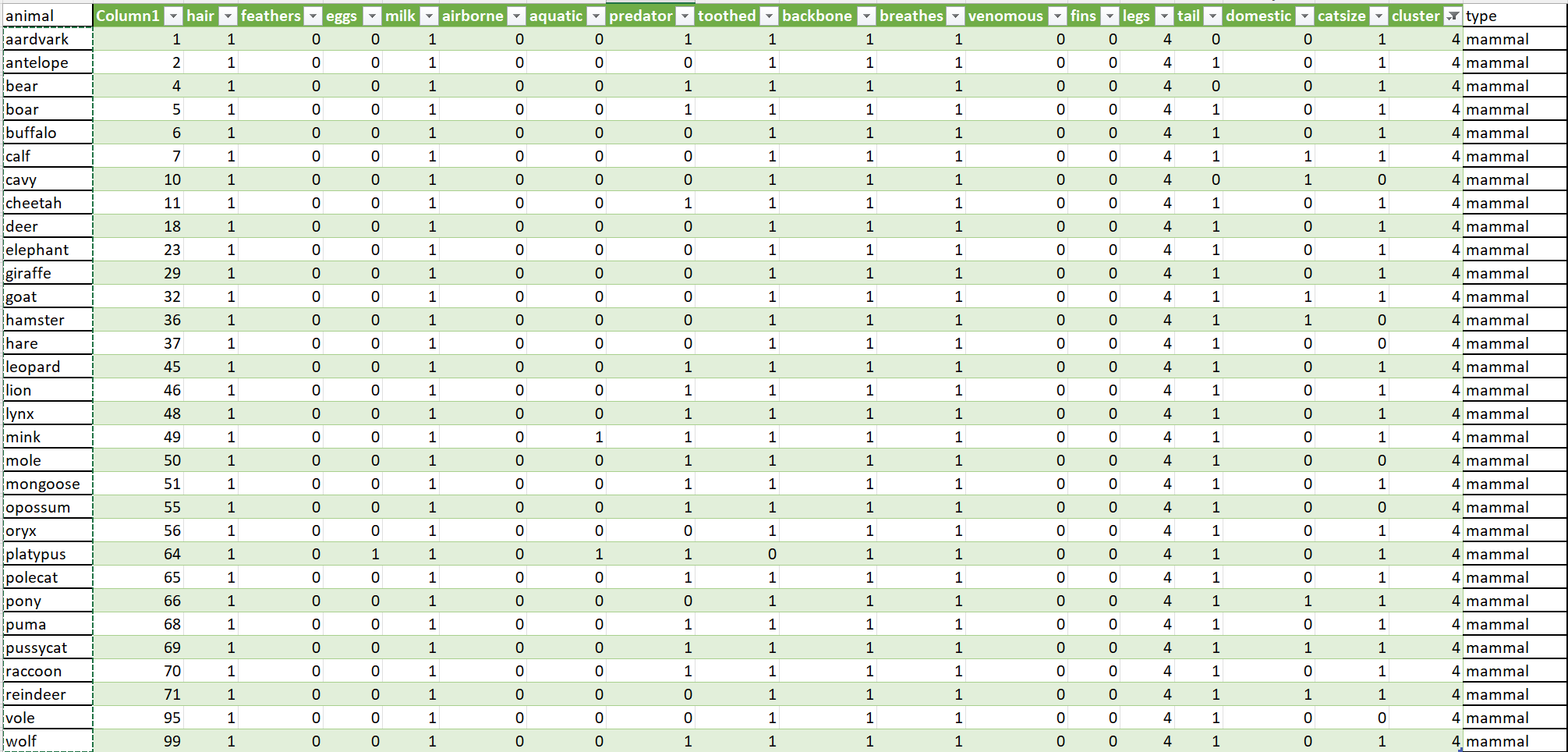
Cluster 2 – mostly insects, but also invertebrates: “crayfish” and “lobster”



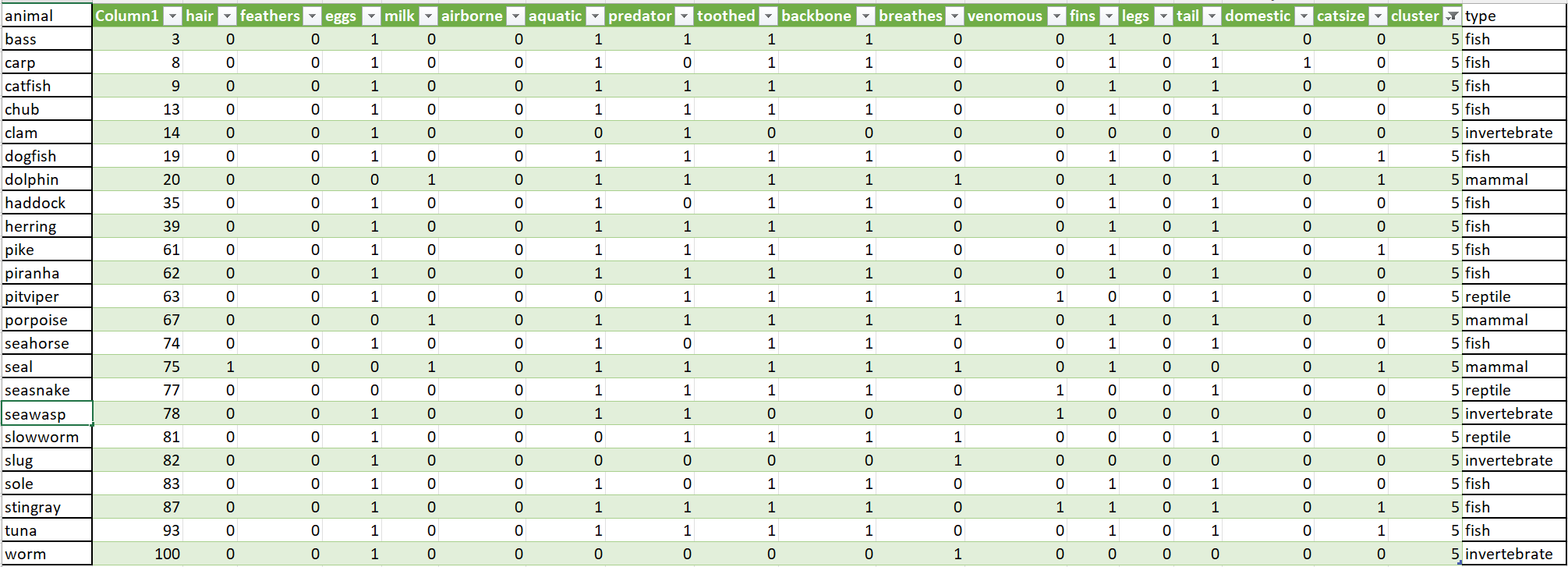
Cluster 3 – mammals. Type seems correct but this cluster inhabitants are very unlikely have many in common.



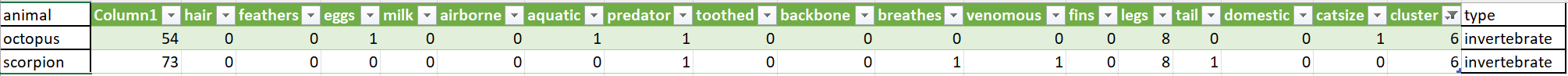
Cluster 4 – all mammals. Nothing strange here.



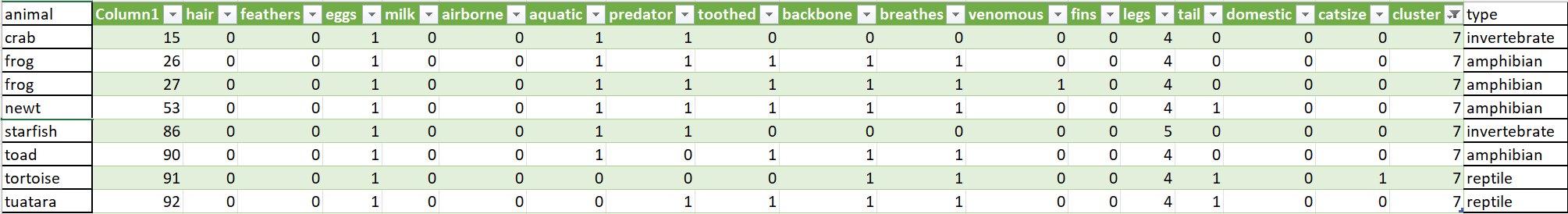
Cluster 5 – Mix of fish, invertebrate, mammal and one reptile - pitviper. Mammals are dolphin, porpoise and seal. Invertebrates are: clam, seawasp, slug, worm.



Cluster 6 – invertebrates. There must be more, but algorithm selected only two.



Cluster 7 – also mix of amphibian, invertebrate and reptile.



Conclusions.

1. Sea mammals like dolphin, porpoise and seal mistakenly considered as fish and such an error even humans do.
2. Amphibian, invertebrate and reptiles are difficult to distinguish by the attributes in this dataset.