Homework #5

CS5402 — Intro To Data Mining

Illya Starikov

Due Date: July 24th, 2018

1 DBSCAN

a) The table would look as follows:

Point	x	y	Density	Designation
A1	1	4	3	Core Point
A2	5	2	3	Core Point
A3	4	3	3	Core Point
A4	5	6	2	Border Point
A5	2	5	2	Border Point
A6	5	4	4	Core Point
A7	1	2	2	Border Point
A8	3	1	1	Noise

b) The points in the two clusters would be:

- $\{A1, A5, A7\}$
- $\{A2, A3, A4, A6\}$

The clusters are formed by first identifying the main points; this is done via calculating the ϵ, or the density of the points (i.e., how many points are around those points). After the core points have been identified, the border points are found via observing which points are close (within an ϵ value) to the core points. Finally, the remaining points are simply noise.

2 Bagging Vs Boosting

2.1 Iris

Looking at the confusion matrix of using bagging:

```
  a  b  c   <-- classified as
 50  0  0 |  a = Iris-setosa
  0  0 50 |  b = Iris-versicolor
  0  0 50 |  c = Iris-virginica
```

And comparing to the confusion matrix of boosting:

```
  a  b  c   <-- classified as
 50  0  0 |  a = Iris-setosa
  0 45  5 |  b = Iris-versicolor
  0  1 49 |  c = Iris-virginica
```

We see that boosting is far more accurate; the kappa statistic agrees, where bagging had a kappa statistic of 0.5 versus the boosting kappa statistic of 0.94.

2.2 Iris

Looking at the confusion matrix of bagging:

```
 445  13 |    a = benign
   7 234 |    b = malignant
```

And comparing to the confusion matrix of the boosting:

```
   a   b   <-- classified as
 445  13 |    a = benign
  17 224 |    b = malignant
```

We see there is very little difference in accuracy. The kappa statistic agrees; where bagging had a kappa statistic of 0.937 versus the boosting kappa statistic of 0.9046.

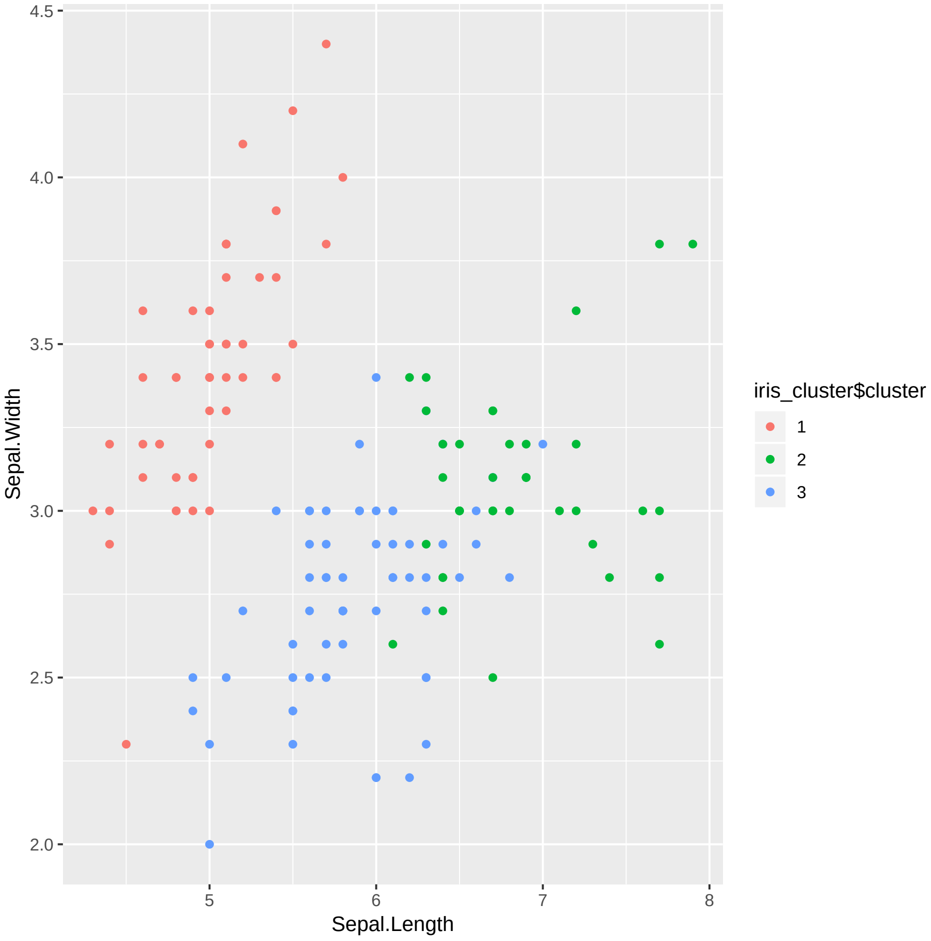
3 KMeans In R

```r
library(ggplot2)

iris_copy <- data.frame(iris[1:4])
iris_cluster = kmeans(iris_copy, 3)

table(iris_cluster$cluster, iris$Species)

# could not get regular plot to work so this is what i got to work
# requires packages ggplot2 + labeling (and dependencies)
iris_cluster$cluster <- as.factor(iris_cluster$cluster)
ggplot(iris, aes(Sepal.Length, Sepal.Width, color = iris_cluster$cluster))
     + geom_point()
```

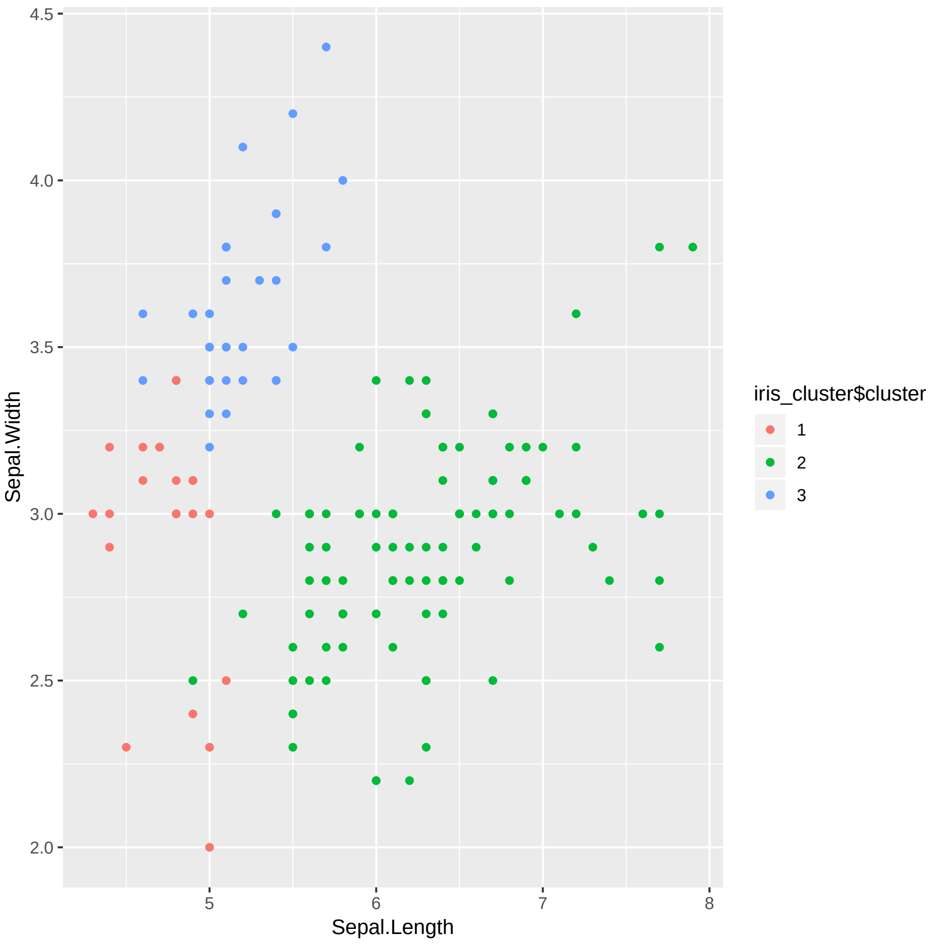


4 DBSCAN In R

```
library(fpc)
iris_cluster = dbscan(iris_cluster, eps=0.42, MinPts = 5)

table(iris_cluster$cluster, iris$Species)

# could not get regular plot to work so this is what i got to work
# requires packages ggplot2 + labeling (and dependencies)
iris_cluster$cluster <- as.factor(iris_cluster$cluster)
ggplot(iris, aes(Sepal.Length, Sepal.Width, color = iris_cluster$cluster))
     + geom_point()
```



5 Frames For Days

```
refund = c(
     "y",
     "n",
     "n",
     "y",
     "n",
     "n",
     "y",
     "n",
     "n",
     "n"
)
status = c(
     "single",
     "married",
     "single",
     "married",
     "divorced",
     "married",
     "divorced",
     "single",
     "married",
     "single"
)

income_k = c(
     125,
     100,
     70,
     120,
     95,
     60,
     220,
     85,
     75,
     90
)
class = c(
     FALSE,
     FALSE,
     FALSE,
     FALSE,
     TRUE,
     FALSE,
     FALSE,
     TRUE,
     FALSE,
     TRUE
)

tax_info = data.frame(refund, status, income_k, class)
```

6 She’s So Mean

```
tax_info <- data.frame(refund, status, income_k, class)
total <- 0

for (i in 1:nrow(tax_info)) {
    total <- total + tax_info$income_k[i]
}

average <- total / nrow(tax_info)
print(average)
```

7 Animals

```r
library(party)

data <- read.csv("C:\\temp\\animal data.csv")
data_frame = data.frame(data)

tree = ctree(Name ~ BloodType + GivesBirth + CanFly + LivesInWater, data=data_frame)
plot(tree, type="simple")
```