Lab 2 Association Analysis -1

The dataset was discretizised as described in the instructions and as part of my my ”extra task” for the submission will be using different a different cluster technique and a different number of clusters.

# Clustering

## Clustering with SimpleKmeans, k = 3

The result of the K-means clustering with k = 3 resulted in the following result.

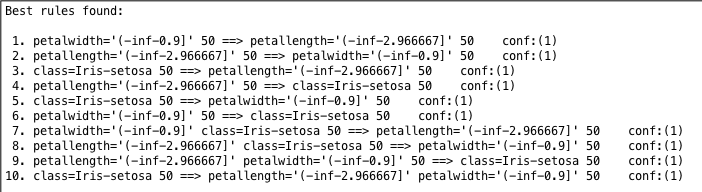
|  |  |  |  |
| --- | --- | --- | --- |
| Iris-setosa (2) | Iris-versicolor (0) | Iris-virginica (1) |  |
| 50 | 0 | 0 | Iris-setosa (2) |
| 0 | 48 | 2 | Iris-versicolor (0) |
| 0 | 7 | 43 | Iris-virginica (1) |

The klustering seems to return stable results with only 9 missclasifications of 150 observations which should be deemd like a good result with no prior knowleged to the data.

# Association

Since no real goals (except running the algorithm) with this part of the task was described I ran them algorithm with default settings delta = 0.05, a lowerboundminsuport = 0.1 and sorted by highest confidence selecting the top 10 rules.

The first part of the print-out describes how many cycles of generating large item-sets were preformed and their respective sizes.



The top 10 rules are rules with 100 % percent confidence since both rule and consequence were 100 support. In other words in the 50 observations where petalwidth was observed with (-inf – 0.9) the petallength was (–inf-2.966667) 50 times so the confidence is calculated as follows (50/150)/(50/150) and resulting in a 100 confidence.

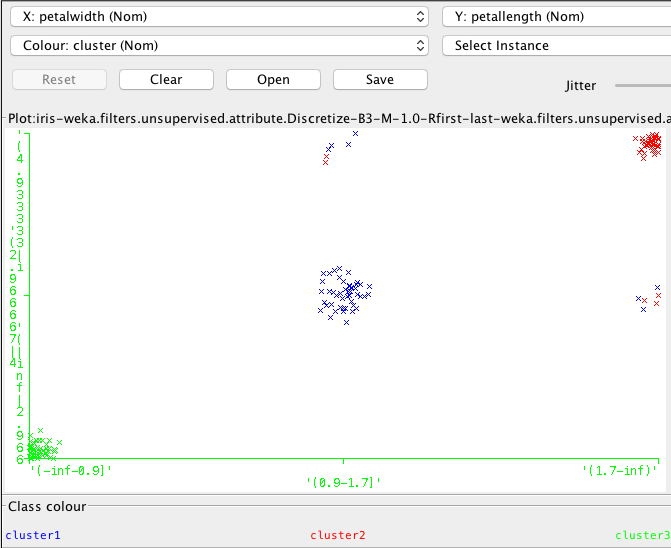
# K –means

Since we are interested in a little bit of more interesting rules I increased the number of rules to 100 and kept the support level after seeing how many high-support rules were generated. As instructed the variable “class” was removed.

When looking for rules I decided on searching for ones with high confidence since the way we are applying it to the data we are in a way using unsupervised machine learning techniques to do supervised learning since we are supposed to strive for as good rules in a sense that we want the correct classification.

9. petallength='(-inf-2.966667]' petalwidth='(-inf-0.9]' 50 ==> cluster=cluster3 50 conf:(1)

This rule is good in a sense it is describing the third cluster ( the green one) and how it is occuring 100 % of the times these two variables are in their respective bins (intervalls)



Another good rule is

26. petallength='(4.933333-inf)' petalwidth='(1.7-inf)' 40 ==> cluster=cluster2 40 conf:(1)

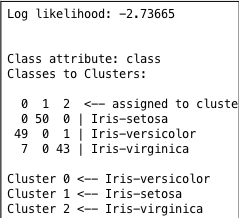
Since it is describing the blue cluster that is in the middle of the graph above.

This rule is describing and successfully identifying the the red cluster in the upper right corner.

13. petallength='(2.966667-4.933333]' petalwidth='(0.9-1.7]' 48 ==> cluster=cluster1 48 conf:(1)

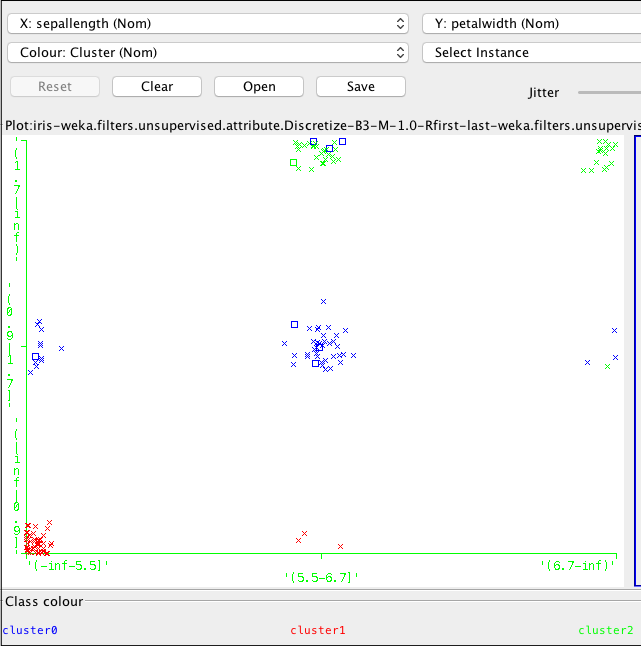
A general observation is that petallength and petalwidth are better at identifying the clusters than other combinations of the variables since the combination of petallength/width creates the best separation between the clusters.

# EM with three clusters



The clustering for the EM algorithm improves a bit, but not that much. So I expect similar results as the K-means with k = 3

I found the same rules as in K-means k =3 but instead of presenting them again I found some other good rules to showcase. These are not as good but offers some variation to the interpretation.



27. sepallength='(5.5-6.7]' petalwidth='(0.9-1.7]' 38 ==> cluster=cluster1 38 conf:(1)

44. sepallength='(-inf-5.5]' sepalwidth='(2.8-3.6]' petallength='(-inf-2.966667]' 36 ==> cluster=cluster2 36 conf:(1)

88. sepallength='(5.5-6.7]' sepalwidth='(2.8-3.6]' petalwidth='(1.7-inf)' 18 ==> cluster=cluster3 18 conf:(1)

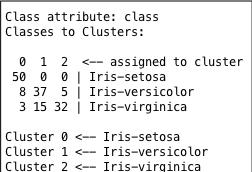
151. sepallength='(5.5-6.7]' sepalwidth='(2.8-3.6]' petallength='(4.933333-inf)' 16 ==> cluster=cluster3 15 conf:(0.94)

Rule 27 identifies the blue cluster in the middle. 44 finds the red cluster in the part left corner and 88 finds the top right green cluster. These three have all 100 % confidence since the observations in each cluster belongs to the same clusterlable. Worth noting is that although they have the same confidence as the rules presented in the K-means section they are not as good since the number of observations identify fewer observations. To identify more observations with these two variables we need to accept worse rules.

Rule 151 also identifies the green cluster 3 but only has 94 % percent confidence since there are observations belonging to a different clusterlabel among the observations.

## Bins = 6

When discretizising the variables in 6 bins i hoped it would maybe separate the classes a bit better to get a better result with K-means with K = 3.



Apperantly this was not such a good idea since it gave me a missclasification rate at around 20 % which is worse than the two previous experiments.

6. petalwidth='(-inf-0.5]' 49 ==> petallength='(-inf-1.983333]' cluster=cluster1 49 conf:(1)

13. sepallength='(5.5-6.1]' petallength='(3.95-4.933333]' 21 ==> cluster=cluster2 21 conf:(1)

20. sepallength='(6.1-6.7]' petallength='(4.933333-5.916667]' 20 ==> cluster=cluster3 20 conf:(1)

With these three rules we can capture 90 osbervations of the 150 which is decent, but since the clustering it self is not that good not even if we could find rules that identifies all clusters the result is still worse than the previous two experiments since here we have a higher missclassification rate.

But if we didn’t have the classlabels we wouldn’t be able to tell and might have been more content with this result. This is the difference between supervised and unsupervised learning, in this case we perform unsupervised learning with the answer avaliable which is kind of cheating since in real life we would probably have used some sort och supervised learning to find the correct classlabels.