STAT2004 Analytics for Observational Data

Semester 2, 2021

Final Project Report

PCA, Clustering, and Classification Based on Association Rules

Christopher Lyon (20160586)

Polawat Srichana (20696984)

Data Science, e.g., Bachelor of Science (Industrial and Applied Mathematics)

Declaration

The work presented in this report is my own work and all references are duly acknowledged.

This work has not been submitted, in whole or in part, in respect of any academic award at Curtin University or elsewhere.

cllyon
_____

Christopher Lyon (20160586)

(31/10/22)

_____Pan_____

(Polawat Srichana 20696984)

(Date)

1. Introduction

This project aims to analyze a multivariate dataset with a selected R package. The selected dataset is Agaricus, which describes the physical characteristics of the Agaricus and Lepiota mushrooms in detail. Secondly, the selected R package is arulesCBA which is a package that serves classification by association for data processing.

2. Methodology

- **Principle Component Analysis (PCA)**

It is a technique used to examine multivariate data to determine the link between the variables, producing a more understandable, smaller, more complex matrix. PCA was applied to make the features (Features) lower in size. As a result, modeling will take less time (Maćkiewicz, A., & Ratajczak, W, 1993).

1. Imagine there are 50 variables or traits, but only 50 of them are used. We must order the factors. Any variable that impacts it, for instance, is utilized. It is unnecessary to utilize any variable because they are all essentially irrelevant. Principal Component Analysis (PCA) is the name for this straightforward idea.
2. Our primary data are unaffected by this tenet. The new data's display gives smaller and simpler data to use

Each component produced by PCA is an entirely uncorrelated variable. The first component, which describes the most data, has the biggest variance, which gets lower as the components follow. Optimal choices account for 80–90% of the variance.

- **Clustering**

Clustering is an unsupervised learning approach in machine learning. We attempt to group things with comparable characteristics and traits by splitting a big set of objects into smaller groups using clustering or cluster analysis in R. Compared to the objects of other sets, the objects of a subset are more like one another. Instead of being a fix for classification problems, clustering is a method for navigating classification challenges. Many algorithms use clustering techniques to address classification problems. The efficiency of these algorithms, techniques used to group things into clusters, and even the definition of a cluster differ.

Imagine that you have to categorize the items in a dataset with n rows and m columns. The information in this dataset can be shown as n points in m-dimensional space. We employ a clustering algorithm to establish the distance between these objects and divide them into groups. (Madhulatha, T. S. ,2012).

- **K-means Clustering**

K-means is one of the data mining methods in the Unsupervised Learning category: learning without teaching. The main function of K-means is clustering to group similar data into the same group (Likas, A., Vlassis, N., & Verbeek, J. J.,2003).

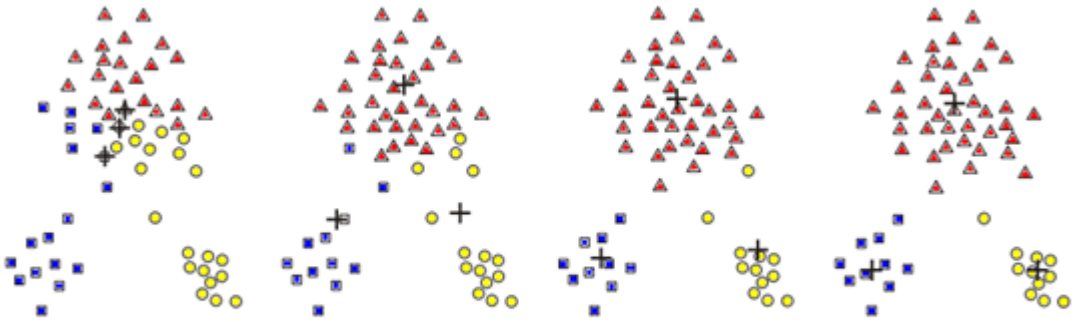
An example of clustering is market segmentation. Group our customers into segments (or clusters). We do not know if the group will be 2, 3, 4, 5, or maybe 10; it is up to us to decide what clustering results make the most sense.

k is an integer that specifies the number of segments required from the data set. Mean tells us that the data used with this algorithm must be a number that can only be "averaged."

- Method of K-means

K-means has four steps as follows:

1. Set the number of groups first, e.g., two groups, meaning K=2 (designated as C1 and C2), and randomly randomize the x and y-axis to C1 and C2 to get C1(x1, y1) and C2(x2, y2). Each point C is called a centroid.
2. To see the position of each member, who is closer to each other, let that person be a member of that C. From here, we can know which group member is between C1 and C2.
3. Re-adjust the x and y of C1 and C2 to be in the middle of the group.
4. Repeat steps 2 and 3 until C1 and C2 do not change.



5.

Figure 1: K-mean Clustering

Source: https://datamining-techniques.blogspot.com/2012/09/k-means-k-means-clustering.html

Figure (a) is the grouping in the first step, where the number of 3 groups is determined, and the starting center is set. In which the "+" symbol represents the center of the three groups, the object is assigned to a group with the center closest to the object, as shown in Figure (b). The center has changed, and a new relationship is formed between the object and the center. Moreover, group objects into groups with the center closest to the object, as shown in Figure (c). Repeat this process until the center does not change to obtain the result, as shown in Figure (d) (Hartigan, J. A., & Wong, M. A.,1979).

- **Association Rules (arules)**

  One of the more used data mining methods is association rules. Several datasets within a large data group are correlated using association rules. Various approaches can be used to find the correlation rule. Nevertheless, the Apriori algorithm is the most well-known and frequently employed (Hahsler, M., Johnson, I., Kliegr, T., & Kucha, J. ,2019).

  Market basket analysis, which establishes the relationship between the goods buyers typically purchase concurrently, is one application of Association Rules. Used to coordinate Marketing Campaigns Support

  An item with a cause to set the resulting item can be used to represent the Association Rule. Implied $I = \{ i_1, i_2, \dots, i_m \}$ is a collection of items. The set of transactions $D = \{ t_1, t_2, \dots, t_m \}$ is the set of transactions, and t is a subset of I. The transactions in D are where the transaction ID number is brought in. For instance, Organizations listing five transactions as following Table 1 & 2

ID	Jam	Oil	Spice	Beer
1	1	1	0	0
2	0	0	0	1
3	0	1	1	0
4	1	1	1	0

Table 1: market basket table

Create a co-occurrence table or a table that counts the frequency of events using the transaction data. To determine how the following events are related to one another:

ID	Jam	Oil	Spice	Beer
Jam	2*	2	1	0
Oil	2	4*	2	0
Spice	1	1	2*	0
Beer	0	0	0	1

Table 2: market basket co-occurrence table

"*" tells how many times that item has been purchased.

Then create a rule from possible relationships using the "IF condition Then result," e.g.,

1. If Jam, Then Oil

2. If Jam, Then Spice,

The total number of possible rules is calculated from the equation $2^n - 1$, where n is the total number of Item types. For example, the table has a total of 4 types of items, so the total number of possible rules is$2^4 - 1 = 15$ rules. It will not say which events happened first. Just say that those events only happened together. Indicators are used to find rules that are of interest and correct.

1. Support Factor is a value that indicates how often events A and B occur.

$$A \rightarrow B: Support\ Factor = (A\ U\ B)$$

2. Confident Factor is a value that says When event B occurs, what is the probability that event A will occur?

$$A \rightarrow B : Confident\ Factor = P(A|B)$$

3. Results

This data set includes descriptions of hypothetical samples corresponding to 23 species of gilled mushrooms in the Agaricus and Lepiota Families. Each species is identified as definitely edible, poisonous, or of unknown edibility and not recommended. This latter class was combined with the poisonous one. The Guide clearly states that there is no simple rule for determining the edibility of a mushroom; no rule like “leaflets three, let it be” for Poisonous Oak and Ivy.

The information on each attribute of the Mushroom dataset is shown in Appendices.

Before performing the k-means clustering, an elbow plot was created to determine the optimal number of clusters. Unfortunately, no single number of clusters has a noticeably higher increase in the variance explained. Multiple k-means clustering models will be created with varying numbers of clusters to determine which has the best performance.

- **K-Mean Clustering**

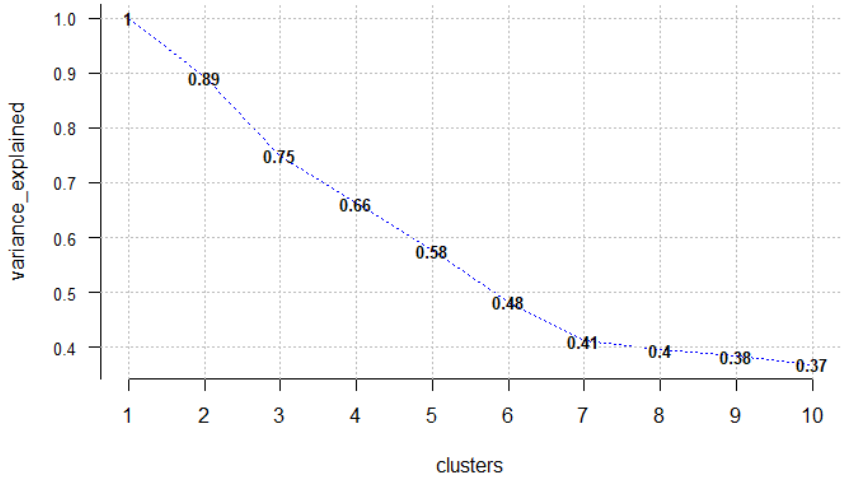


Figure 2: Variance

For modeling the k-means clustering, 2, 3, 6, and 9 clusters were used. Although there are only two classes (edible and poisonous), the increased number of clusters could better describe subsets of these classes rather than attempting to cluster the classes. For the 2-cluster model, the apparent error rate is 0.27, meaning that 27% of the mushrooms were misclassified. Figure 3 shows the overlap in the clusters in (k = 2). The 3-cluster model had a lower apparent error rate at 0.238. The 6-cluster model also had a lower apparent error rate at 0.107, and the 9-cluster model at 0.075.

As the number of clusters increased, the overlap between clusters increased, but the apparent error rate was lowered. The result means that although more clusters were used than classes exist, the clusters created could better describe the edible and poisonous mushroom subsets.

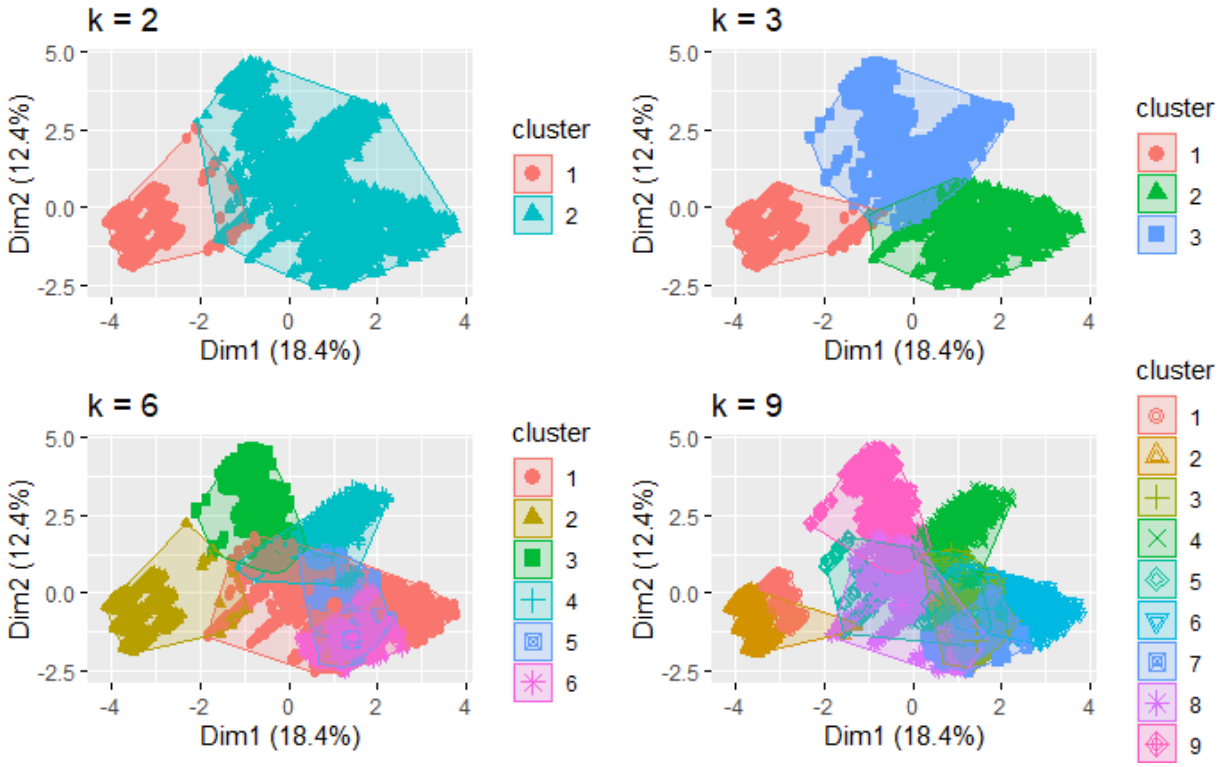


Figure 3: K-mean Clustering

```
> table(k$cluster, mushroom.class)
   mushroom.class
      e    p
1   192 1744
2  4016 2172
> table(k2$cluster, mushroom.class)
   mushroom.class
      e    p
1    36 1752
2  4172 2164
> table(k3$cluster, mushroom.class)
   mushroom.class
      e    p
1    98 1758
2  2963  694
3  1147 1464
> table(k4$cluster, mushroom.class)
   mushroom.class
      e    p
1   228 1336
2  2768  768
3   114 1760
4  1098   52
```

```
> table(k6$cluster, mushroom.class)
   mushroom.class
      e    p
1 1433  592
2   25 1749
3  789    0
4   41 1358
5 1728  199
6  192   18
> table(k9$cluster, mushroom.class)
   mushroom.class
      e    p
1    0  576
2    0 1156
3 1728  179
4    0 1358
5  276  492
6  704   65
7  192   18
8  528   72
9  780    0
```

Table 3: K-mean Clustering

- **PCA**

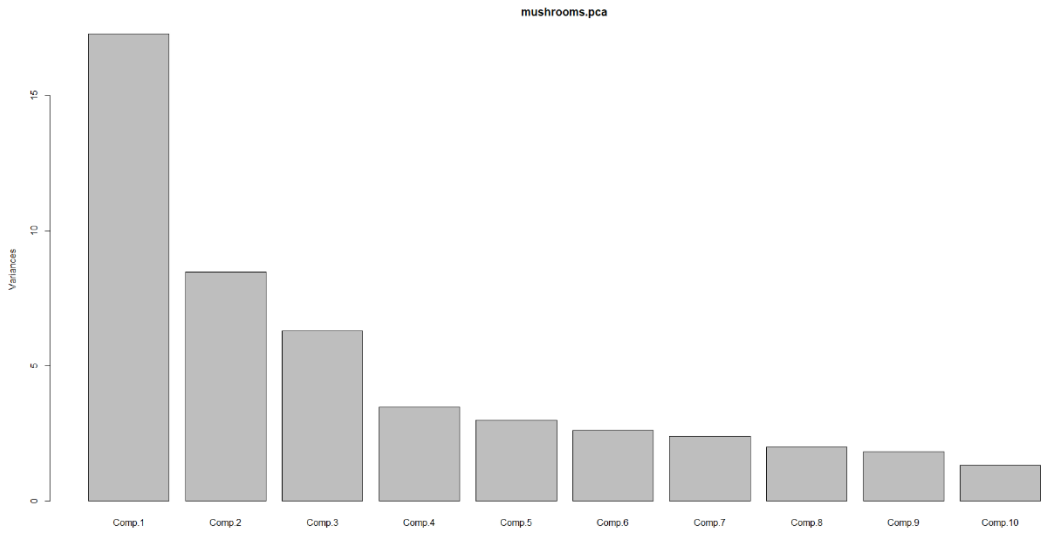


Figure 4: Importance of Components

From Table3 and Figure 4, 50% of the total variance is component 1&2.

```
Importance of components:
                          Comp.1    Comp.2    Comp.3     Comp.4     Comp.5     Comp.6     Comp.7     Comp.8     Comp.9
Standard deviation     4.1570062 2.9133606 2.5077815 1.86523603 1.72768905 1.61456214 1.54626926 1.41445120 1.35331936
Proportion of Variance 0.3375873 0.1658110 0.1228582 0.06796611 0.05831173 0.05092539 0.04670841 0.03908416 0.03577877
Cumulative Proportion  0.3375873 0.5033983 0.6262565 0.69422264 0.75253437 0.80345976 0.85016816 0.88925232 0.92503109
                          Comp.10    Comp.11    Comp.12    Comp.13     Comp.14     Comp.15     Comp.16     Comp.17     Comp.18
Standard deviation     1.15178833 1.01066269 0.72595206 0.53276447 0.465970650 0.420258973 0.323180255 0.278235058 0.252425360
Proportion of Variance 0.02591613 0.01995434 0.01029534 0.00554492 0.004241719 0.003450315 0.002040396 0.001512337 0.001244775
Cumulative Proportion  0.95094722 0.97090155 0.98119689 0.98674181 0.990983533 0.994433848 0.996474244 0.997986581 0.999231356
                          Comp.19      Comp.20      Comp.21
Standard deviation     0.1583429890 0.1071970730 5.274733e-02
Proportion of Variance 0.0004898042 0.0002244867 5.435327e-05
Cumulative Proportion  0.9997211600 0.9999456467 1.000000e+00
```

Table 4: Importance of Components

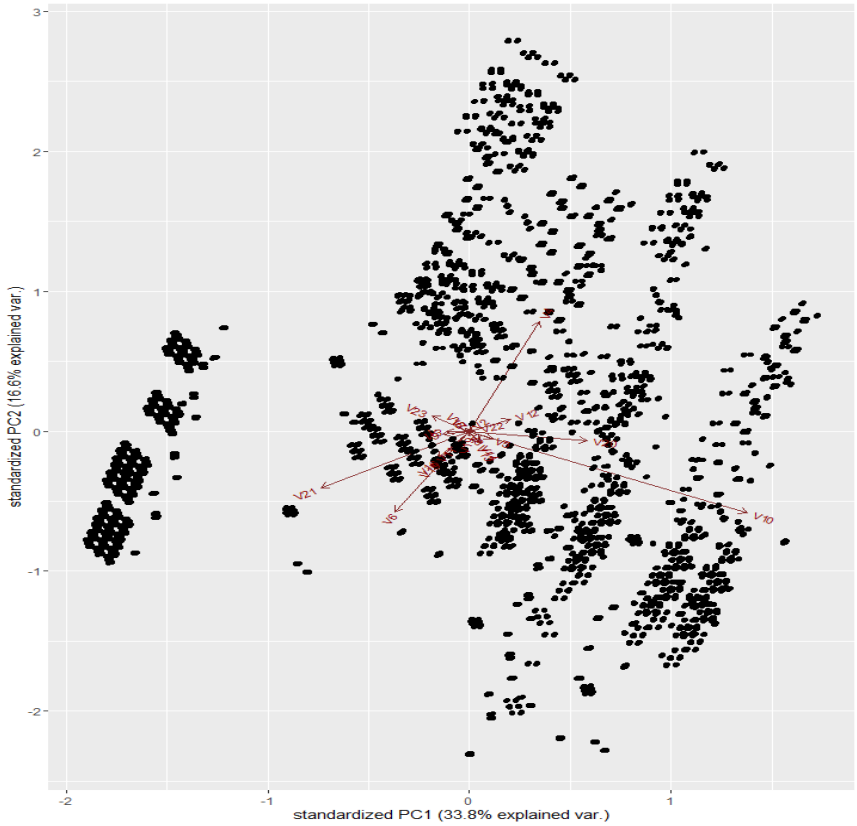


Figure 5: Biplot PCA

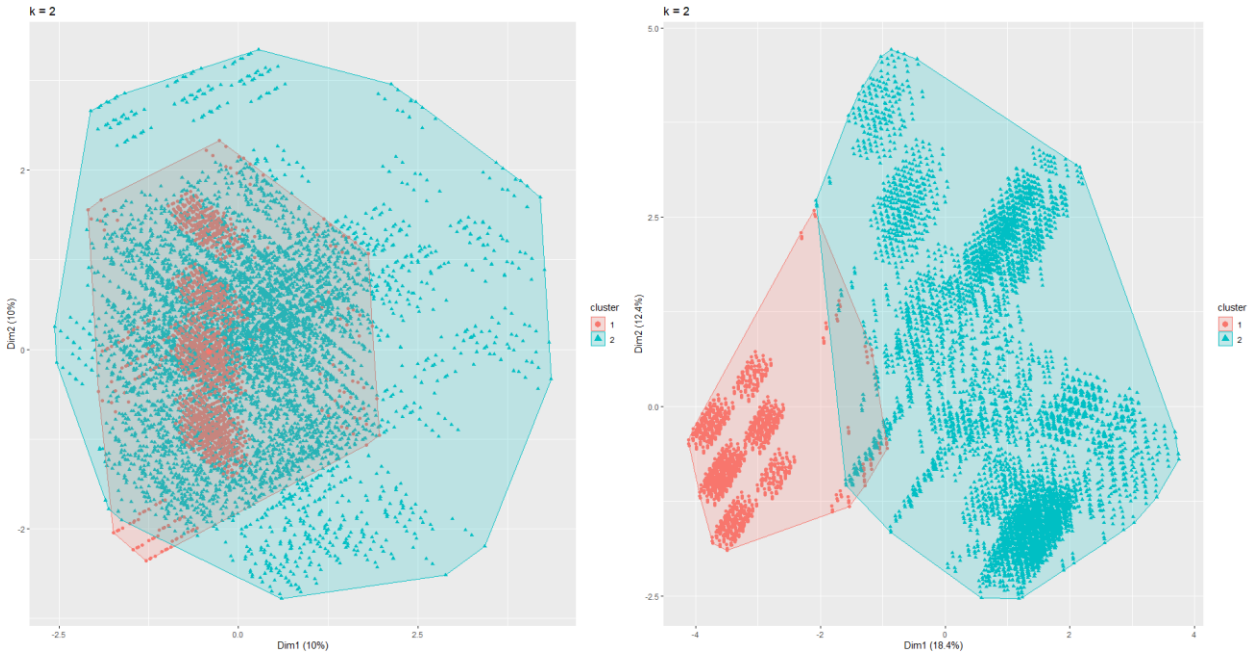


Figure 6: PCA with K-mean Clustering

- From Figure 4, biplot PCA has not had much information can be seen due to the simplicity of the plot. However, this plot tells us that the most important property is V10, as seen from the length of the arrow. Furthermore, from the graph, there is a cluster of data on the left that may be identified as another group.
- As a result of Figure 5, the results were not very satisfactory due to the huge overlap between the 2 clusters in RHS. On the other hand, after applying PCA, it decreases overlapping and increases density, but it still has overlapping in the central. Using more clusters may deal with an overlapping problem.

- **arulesCBA**

The arulesCBA package provides many functions, but the two used here are CBA () and inspect (). The former function mines for association rules in the dataset given within the scope of the input formula. After determining the association rules, they are then pruned only to include those leading to the best-performing model using the confidence and support statistics.

A 4-fold cross-validation process was used to determine the effectiveness of the association rules-based classification model. Over the cross-validation stages, the first few association patterns mined from the dataset for use in the classification are the same $(V6 = n, V9 = b, V19 = o) => (V1 = e)$. This pattern has a support of 0.33, meaning that it occurs in ~1/3 of all observations, and a confidence of 1, meaning that anywhere that $(V6 = n, V9 = b, V19 = o)$ occurs, the mushroom is edible. Most rules found and used in the classification model have a confidence of 1, except for the base case where no other rules apply, which classifies as the majority class.

Method 1	e	p	Method 2	e	p
e	4204	0	e	4206	44
p	4	3916	p	2	3872

Table 2: ArulesCBA

After running through the cross-validation, the final apparent error rate is 0.0005 for the model using the first pruning method and 0.0057 for the model using the second pruning method. This performed much better than the previous k-means clustering models.

4. Discussion

From this project, the PCA has a few problems. It may be due to the use of a small number of clusters; therefore, it is not possible to separate the clusters. In addition, Componet1&2 are still 50% of the total variability. Whereas the cluster division for K-means yielded satisfactory results at K=3 and very satisfactory results for ArulesCBA with error rate 0.0005. Even though we must use “Catencoder” to convert any value to any number, the result is still satisfied for Arules and K-mean

5. Conclusion

This project summary offers information and methods that will be helpful for upcoming initiatives. Even with poor outcomes, PCA offers methods for reducing dimensions and locating variable boundaries. The cluster offers information about the data clusters that should be excluded. Mining

association rules have a collection of variables to produce the greatest lift under predetermined circumstances.

6. References

Maćkiewicz, A., & Ratajczak, W. (1993). Principal components analysis (PCA). *Computers & Geosciences*, *19*(3), 303-342.

Abdi, H., & Williams, L. J. (2010). Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, *2*(4), 433-459.

Madhulatha, T. S. (2012). An overview on clustering methods. *arXiv preprint arXiv:1205.1117*.

Likas, A., Vlassis, N., & Verbeek, J. J. (2003). The global k-means clustering algorithm. *Pattern recognition*, *36*(2), 451-461.

Hartigan, J. A., & Wong, M. A. (1979). Algorithm AS 136: A k-means clustering algorithm. *Journal of the royal statistical society. series c (applied statistics)*, *28*(1), 100-108.

Hahsler, M., Johnson, I., Kliegr, T., & Kucha, J. (2019). Associative Classification in R: arc, arulesCBA, and rCBA. *R Journal*, *9*(2).

Laurent, A., & Hauschild, M. Z. (2015). Normalisation. In *Life cycle impact assessment* (pp. 271-300). Springer, Dordrecht.

Hegland, M. (2007). The apriori algorithm–a tutorial. *Mathematics and computation in imaging science and information processing*, 209-262.)

7. Appendices

```
library(factoextra)
library(flexclust)
library(fpc)
library(clustertend)
library(cluster)
library(ClusterR)
library(CatEncoders)
library(devtools)
install_github("vqv/ggbiplot")

mushrooms = read.csv('C:/Users/Pan/Downloads/agaricus-lepiota.data', header = FALSE, col.names = )
head(mushrooms)


nrow(mushrooms)
ncol(mushrooms)
ls(mushrooms)
any(is.na(mushrooms))
summary(mushrooms)
mushrooms$V17 <- NULL
mushroom.class = as.factor(mushrooms$V1)
mushrooms$V1 <- NULL
ncol(mushrooms)

head(mushroom.class)


for(i in 1:ncol(mushrooms)){

  labs = LabelEncoder.fit(as.factor(mushrooms[,i]))
  mushrooms[,i]=transform(labs,as.factor(mushrooms[,i]))
}

mushrooms
mushrooms_s <- as.data.frame(lapply(mushrooms, scale))

mushrooms.pca = princomp(mushrooms)
screeplot(mushrooms.pca)

summary(mushrooms.pca)
library(ggbiplot)
ggbiplot(mushrooms.pca)

opt<-Optimal_Clusters_KMeans(mushrooms_s, max_clusters=10, plot_clusters=TRUE, criterion="silhouette")
opt1<-Optimal_Clusters_KMeans(mushrooms_s, max_clusters=10, plot_clusters = TRUE)

opt2<-Optimal_Clusters_KMeans(mushrooms.pca$scores, max_clusters=10, plot_clusters = TRUE)
```

```
k = kmeans(mushrooms.pca$scores[,1:10], centers=2, nstart=25)
k2 = kmeans(mushrooms_s, centers=2, nstart=25)
k3 = kmeans(mushrooms_s, centers=3, nstart=25)

k4 = kmeans(mushrooms_s, centers=4, nstart=25)
k6 = kmeans(mushrooms_s, centers=6, nstart=25)
k9 = kmeans(mushrooms_s, centers=9, nstart=25)


p <- fviz_cluster(k, geom = "point", mushrooms.pca$scores[,1:10]) + ggtitle("k = 2")
p2 <- fviz_cluster(k2, geom = "point", mushrooms_s) + ggtitle("k = 2")
p3 <- fviz_cluster(k3, geom = "point", mushrooms_s) + ggtitle("k = 3")
p4 <- fviz_cluster(k4, geom = "point", mushrooms_s) + ggtitle("k = 4")
p6 <- fviz_cluster(k6, geom = "point", mushrooms_s) + ggtitle("k = 6")
p9 <- fviz_cluster(k9, geom = "point", mushrooms_s) + ggtitle("k = 9")

library(gridExtra)
grid.arrange(p2,p3, p6, p9, nrow=2)


grid.arrange(p, p2, nrow=1)

table(k$cluster, mushroom.class)
table(k2$cluster, mushroom.class)
table(k3$cluster, mushroom.class)
table(k4$cluster, mushroom.class)
table(k6$cluster, mushroom.class)
table(k9$cluster, mushroom.class)
```

```r
library(arulesCBA)



data = read.csv('C:/Users/Chris/Downloads/agaricus-lepiota.data', header=FALSE)

#Cleaning
for(i in 1:23){
  data[,i] = as.factor(data[,i])
}

t.m1=matrix(c(0,0,0,0), nrow=2)
t.m2=matrix(c(0,0,0,0), nrow=2)



classify = function(method, data, dt){
  data.test = data[dt,]
  data.train = data[-dt,]


  cl = CBA(V1~., data=data, pruning=method)
  inspect(cl$rules, linebreak = TRUE)
  pred = predict(cl, data.test)

  t = table(pred, data.test$V1)

  print(t)


  return(t)
}


error = function(t){
  return(round((t[1,2]+t[2,1])/sum(t)*100, 2))
}

summ = function(t){
  print(t)
  print(error(t))
}


#4-fold cross validation (8124/4=2031)
for(i in 1:4){
  dt = seq(1+floor(nrow(data)*(i-1)*1/4), floor(nrow(data)*i*1/4))

  t.m1 = t.m1 + classify('M1', data, dt)
  t.m2 = t.m2 + classify('M2', data, dt)
}
```

Attribute Information:

1. cap-shape: bell=b,conical=c,convex=x,flat=f, knobbed=k,sunken=s
2. cap-surface: fibrous=f,grooves=g,scaly=y,smooth=s
3. cap-color: brown=n,buff=b,cinnamon=c,gray=g,green=r, pink=p,purple=u,red=e,white=w,yellow=y
4. bruises?: bruises=t,no=f
5. odor: almond=a,anise=l,creosote=c,fishy=y,foul=f, musty=m,none=n,pungent=p,spicy=s
6. gill-attachment: attached=a,descending=d,free=f,notched=n
7. gill-spacing: close=c,crowded=w,distant=d
8. gill-size: broad=b,narrow=n
9. gill-color: black=k,brown=n,buff=b,chocolate=h,gray=g, green=r,orange=o,pink=p,purple=u,red=e, white=w,yellow=y
10. stalk-shape: enlarging=e,tapering=t
11. stalk-root: bulbous=b,club=c,cup=u,equal=e, rhizomorphs=z,rooted=r,missing=?
12. stalk-surface-above-ring: fibrous=f,scaly=y,silky=k,smooth=s
13. stalk-surface-below-ring: fibrous=f,scaly=y,silky=k,smooth=s
14. stalk-color-above-ring: brown=n,buff=b,cinnamon=c,gray=g,orange=o, pink=p,red=e,white=w,yellow=y
15. stalk-color-below-ring: brown=n,buff=b,cinnamon=c,gray=g,orange=o, pink=p,red=e,white=w,yellow=y
16. veil-type: partial=p,universal=u
17. veil-color: brown=n,orange=o,white=w,yellow=y
18. ring-number: none=n,one=o,two=t
19. ring-type: cobwebby=c,evanescent=e,flaring=f,large=l, none=n,pendant=p,sheathing=s,zone=z
20. spore-print-color: black=k,brown=n,buff=b,chocolate=h,green=r, orange=o,purple=u,white=w,yellow=y
21. population: abundant=a,clustered=c,numerous=n, scattered=s,several=v,solitary=y
22. habitat: grasses=g,leaves=l,meadows=m,paths=p, urban=u,waste=w,woods=d