Machine Learning for Drug Adverse Event Discovery

r-bloggers.com/machine-learning-for-drug-adverse-event-discovery

Data

Let’s create fake drug adverse event data where we can visually identify the clusters and see if our machine learning algorithm can identify the clusters. If we have millions of rows of adverse event data, clustering can help us to summarize the data and get insights quickly.

Let’s assume a drug AAA results in adverse events shown below. We will see in which group (cluster) the drug results in what kind of reactions (adverse events).
In the table shown below, I have created four clusters:

> - Route=ORAL, Age=60s, Sex=M, Outcome code=OT, Indication=RHEUMATOID ARTHRITIS and Reaction=VASCULITIC RASH + some noise
> - Route=TOPICAL, Age=early 20s, Sex=F, Outcome code=HO, Indication=URINARY TRACT INFECTION and Reaction=VOMITING + some noise
> - Route=INTRAVENOUS, Age=about 5, Sex=F, Outcome code=LT, Indication=TONSILLITIS and Reaction=VOMITING + some noise
> - Route=OPHTHALMIC, Age=early 50s, Sex=F, Outcome code=DE, Indication=Senile osteoporosis and Reaction=Sepsis + some noise

Below is a preview of my data. You can download the data here

```
head(my_data)
  route age sex outc_cod              indi_pt              pt
1  ORAL  63   M       OT RHEUMATOID ARTHRITIS VASCULITIC RASH
2  ORAL  66   F       OT RHEUMATOID ARTHRITIS VASCULITIC RASH
3  ORAL  66   M       OT RHEUMATOID ARTHRITIS VASCULITIC RASH
4  ORAL  57   M       OT RHEUMATOID ARTHRITIS VASCULITIC RASH
5  ORAL  66   M       OT RHEUMATOID ARTHRITIS VASCULITIC RASH
6  ORAL  66   M       OT RHEUMATOID ARTHRITIS VASCULITIC RASH
```

Hierarchical Clustering

To perform hierarchical clustering, we need to change the text to numeric values so that we can calculate distances. Since age is numeric, we will remove it from the rest of the variables and change the character variables to multidimensional numeric space.

```
age = my_data$age
my_data = select(my_data,-age)
```

Create a Matrix

```
my_matrix = as.data.frame(do.call(cbind, lapply(mydata, function(x)
table(1:nrow(mydata), x))))
```

Now, we can add the age column:

```
my_matrix$Age=age
head(my_matrix)
INTRAVENOUS OPHTHALMIC ORAL TOPICAL F M DE HO LT OT RHEUMATOID ARTHRITIS Senile
osteoporosis
1           0          0    1       0 0 1  0  0  0  1                    1
0
2           0          0    1       0 1 0  0  0  0  1                    1
0
3           0          0    1       0 0 1  0  0  0  1                    1
0
4           0          0    1       0 0 1  0  0  0  1                    1
0
5           0          0    1       0 0 1  0  0  0  1                    1
0
6           0          0    1       0 0 1  0  0  0  1                    1
0
  TONSILLITIS URINARY TRACT INFECTION Sepsis VASCULITIC RASH VOMITING Age
1           0                       0      0               1        0  63
2           0                       0      0               1        0  66
3           0                       0      0               1        0  66
4           0                       0      0               1        0  57
5           0                       0      0               1        0  66
6           0                       0      0               1        0  66
```

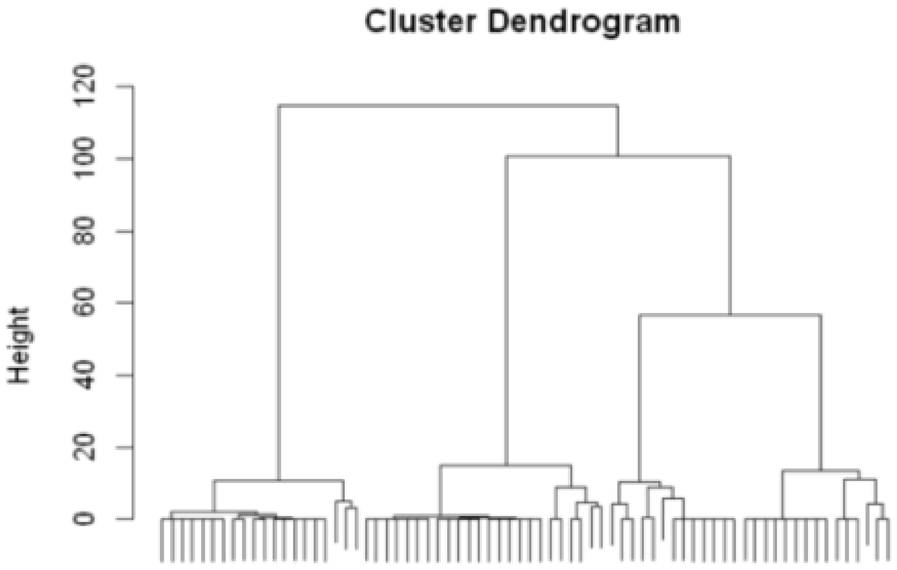
Let’s normalize our variables using *caret package*.

```
library(caret)
preproc = preProcess(my_matrix)
my_matrixNorm = as.matrix(predict(preproc, my_matrix))
```

Next, let’s calculate distance and apply hierarchical clustering and plot the dendrogram.

```
distances = dist(my_matrixNorm, method = "euclidean")
clusterdrug = hclust(distances, method = "ward.D")
plot(clusterdrug, cex=0.5, labels = FALSE,cex=0.5,xlab = "", sub = "",cex=1.2)
```

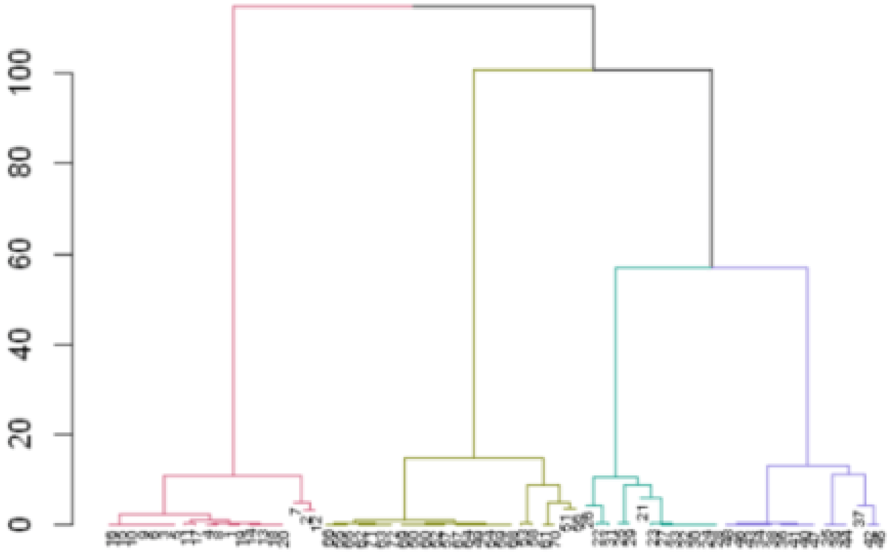
You will get this plot:



From the dendrogram shown above, we see that four distinct clusters can be created from the fake data we created. Let’s use different colors to identify the four clusters.

```
library(dendextend)
dend <- as.dendrogram(clusterdrug)
install.packages("dendextend")
library(dendextend)
# Color the branches based on the clusters:
dend <- color_branches(dend, k=4) #, groupLabels=iris_species)
# We hang the dendrogram a bit:
dend <- hang.dendrogram(dend,hang_height=0.1)
# reduce the size of the labels:
# dend <- assign_values_to_leaves_nodePar(dend, 0.5, "lab.cex")
dend <- set(dend, "labels_cex", 0.5)
plot(dend)
```

Here is the plot:



Now, let’s create cluster groups with four clusters.

```
clusterGroups = cutree(clusterdrug, k = 4)
```

Now, let’s add the clusterGroups column to the original data.

```
my_data= cbind(data.frame(Cluster=clusterGroups), my_data, age)
head(my_data)
Cluster route sex outc_cod              indi_pt             pt age
1        1   ORAL    M       OT RHEUMATOID ARTHRITIS VASCULITIC RASH   63
2        1   ORAL    F       OT RHEUMATOID ARTHRITIS VASCULITIC RASH   66
3        1   ORAL    M       OT RHEUMATOID ARTHRITIS VASCULITIC RASH   66
4        1   ORAL    M       OT RHEUMATOID ARTHRITIS VASCULITIC RASH   57
5        1   ORAL    M       OT RHEUMATOID ARTHRITIS VASCULITIC RASH   66
6        1   ORAL    M       OT RHEUMATOID ARTHRITIS VASCULITIC RASH   66
```

Number of Observations in Each Cluster

```
observationsH=c()
for (i in seq(1,4)){
  observationsH=c(observationsH,length(subset(clusterdrug, clusterGroups==i)))
}
observationsH
=as.data.frame(list(cluster=c(1:4),Number_of_observations=observationsH))
observationsH
cluster Number_of_observations
1       1                      20
2       2                      13
3       3                      15
4       4                      24
```

What is the most common observation in each cluster?

Let’s calculate column average for each cluster.

```
z=do.call(cbind,lapply(1:4, function(i)
round(colMeans(subset(my_matrix,clusterGroups==i)),2)))
colnames(z)=paste0('cluster',seq(1,4))
z
cluster1 cluster2 cluster3 cluster4
INTRAVENOUS                 0.00     0.00     1.00     0.00
OPHTHALMIC                  0.00     0.00     0.00     0.92
ORAL                        1.00     0.08     0.00     0.08
TOPICAL                     0.00     0.92     0.00     0.00
F                           0.10     0.85     0.80     1.00
M                           0.90     0.15     0.20     0.00
DE                          0.00     0.00     0.00     0.83
.....
```

Next, most common observation in each cluster:

```
Age=z[nrow(z),]
z=z[1:(nrow(z)-1),]
my_result=matrix(0,ncol=4,nrow=ncol(mydata))
for(i in seq(1,4)){
    for(j in seq(1,ncol(mydata))){
q = names(mydata)[j]
q = as.vector(as.matrix(unique(mydata[q])))
my_result[j,i]=names(sort(z[q,i],decreasing = TRUE)[1])
    }}

colnames(my_result)=paste0('Cluster',seq(1,4))
rownames(my_result)=names(mydata)
my_result=rbind(Age,my_result)
my_result <- cbind(Attribute =c("Age","Route","Sex","Outcome Code","Indication
preferred term","Adverse event"), my_result)
rownames(my_result) <- NULL
my_result
```

```
Attribute cluster1 cluster2 cluster3 cluster4
Age         61.8           17.54           5.8           44.62
Route         ORAL           TOPICAL   INTRAVENOUS    OPHTHALMIC
Sex          M           F           F            F
Outcome Code  OT           HO           LT            DE
Indication
preferred term RHEUMATOID ARTHRITIS   URINARY TRACT INFECTION TONSILLITIS Senile
osteoporosis
Adverse event VASCULITIC RASH  VOMITING  VOMITING    Sepsis
```

Summary

We see that we have created the clusters using hierarchical clustering. From cluster 1, for male in the 60s, the drug results in vasculitic rash when taken for rheumatoid arthritis. We can interpret the other clusters similarly. Remember, this data is not real data. It is fake data made to show the application of clustering for drug adverse event study. From, this short post, we see that clustering can be used for knowledge discovery in drug adverse event reactions. Specially in cases where the data has millions of observations, where we cannot get any insight visually, clustering becomes handy for summarizing our data, for getting statistical insights and for discovering new knowledge.

, , , ,

, ,
in