Clustering Iris dataset without using plots

January 28, 2018

0.0.1 FInding the best model without using plots to custer the iris dataset

Initialize and load data in kernel

```
In [1]: rm(list=ls())
        result<-NULL
        rawdata<-read.table("iris.txt", header=TRUE)
```

Scaling attributes in iris

```
In [2]: data<-rawdata

        for (i in 1:4){
          data[,i]<-(rawdata[,i]-min(rawdata[,i]))/(max(rawdata[,i])-min(rawdata[,i]))
        }
        head(data)
```

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
0.22222222	0.6250000	0.06779661	0.04166667	setosa
0.16666667	0.4166667	0.06779661	0.04166667	setosa
0.11111111	0.5000000	0.05084746	0.04166667	setosa
0.08333333	0.4583333	0.08474576	0.04166667	setosa
0.19444444	0.6666667	0.06779661	0.04166667	setosa
0.30555556	0.7916667	0.11864407	0.12500000	setosa

Parsing through number of clusters from 3 to 10 while choosing all combinations of attributes and storing the cluster vs actual classification in table result

```
In [3]: for(k in 1:4){
            #k dictatates the number of attributes to choose
            #going from 1 to 4 and t has the combinations of
            #k attributes choosen
            t<-combn(4,k, FUN = NULL, simplify = TRUE)
            #i has the number of clusters, going from 3 to 10
            for(i in 3:10){
                #j-iterates through all combinations of attributes
                #and t[,j] has the column number of attributes
                for(j in 1:(length(t)/k)){
                    set.seed(1)
```

```
                model<-kmeans(data[,t[,j]],i)
                result<-rbind(result
                              ,cbind(i,paste( unlist(t[,j]), collapse=','),k
                                          ,(table(model$cluster, data[,5]))))
        }}}
```

```
In [4]: head(result)
```

	i		k	setosa	versicolor	virginica
1	3	1	1	40	5	1
2	3	1	1	0	14	37
3	3	1	1	10	31	12
1	3	2	1	31	1	5
2	3	2	1	1	27	19
3	3	2	1	18	22	26

In above sample, first column gives the value of k(number of clusters); 2nd column has the attribute chosen, where 1-Sepal.Length, 2-Sepal.Width, 3-Petal.Length, 4-Petal.Width; third column is the number of attributes to be chosen.
Each row depicts a cluster. and the values in columns-4 to 6, show the number of elements of each species within the cluster

Converting the result matrix to a dataframe with numeric values in columns 4,5,6

```
In [5]: result1<- as.data.frame.array(result)
```

```
        result1[,4]<-as.numeric(as.character(result1[,4]))
        result1[,5]<-as.numeric(as.character(result1[,5]))
        result1[,6]<-as.numeric(as.character(result1[,6]))
```

Since each row is a cluster, I am calculating the accuracy of each cluster as largest classification of points as a single species within cluster divided by total number of points in the cluster.
I am storing the result in 7th column of result1 data frame

```
In [6]: for (i in 1:nrow(result1)){
          result1[i,7]<-max(result1[i,4:6])/sum(result1[i,4:6])
        }
```

```
In [7]: head(result1)
```

	i	V2	k	setosa	versicolor	virginica	V7
1	3	1	1	40	5	1	0.8695652
2	3	1	1	0	14	37	0.7254902
3	3	1	1	10	31	12	0.5849057
1.1	3	2	1	31	1	5	0.8378378
2.1	3	2	1	1	27	19	0.5744681
3.1	3	2	1	18	22	26	0.3939394

Now, to compute accuracy of the model, I had to choose between mean and minimum accuracy of all clusters within the model. I chose minimum as I wanted to choose a model which has all clusters clearly classified

So, I aggrregate result1 by columns i (k in clustering) and V2(attributes chosen) and find the minimum accuracy of cluster within each model.

```
In [8]: aggresult<-aggregate(result1[,c(1,2,3,7)]
                             , by=list(result1$i,result1$V2,result1$k)
                             , function(x) min(as.character(x)))

In [9]: aggresult[order(aggresult$V7, decreasing=TRUE),c(4,5,7)]
```

	i	V2	V7
26	3	4	0.923076923076923
27	4	4	0.923076923076923
74	3	3,4	0.923076923076923
106	3	2,3,4	0.92
119	8	1,2,3,4	0.904761904761905
28	5	4	0.891891891891892
53	6	1,4	0.888888888888889
69	6	2,4	0.888888888888889
101	6	1,3,4	0.888888888888889
20	5	3	0.885714285714286
97	10	1,3,4	0.875
102	7	1,3,4	0.875
104	9	1,3,4	0.875
66	3	2,4	0.867924528301887
110	7	2,3,4	0.857142857142857
77	6	3,4	0.851851851851852
78	7	3,4	0.851851851851852
109	6	2,3,4	0.851851851851852
23	8	3	0.846153846153846
117	6	1,2,3,4	0.84
30	7	4	0.833333333333333
54	7	1,4	0.833333333333333
49	10	1,4	0.823529411764706
55	8	1,4	0.823529411764706
56	9	1,4	0.823529411764706
25	10	4	0.8125
31	8	4	0.8125
79	8	3,4	0.8
98	3	1,3,4	0.786885245901639
42	3	1,3	0.78
88	9	1,2,3	0.545454545454545
99	4	1,3,4	0.545454545454545
100	5	1,3,4	0.545454545454545
3	4	1	0.538461538461538
4	5	1	0.538461538461538
68	5	2,4	0.536585365853659
111	8	2,3,4	0.533333333333333
8	9	1	0.53125
1	10	1	0.526315789473684
39	8	1,2	0.526315789473684
41	10	1,3	0.526315789473684
47	8	1,3	0.526315789473684
48	9	1,3	0.526315789473684
107	4	2,3,4	0.525
108	5	2,3,4	0.525
67	4	2,4	0.523809523809524
29	6	4	0.5
33	10	1,2	0.5
38	7	1,2	0.5
63	8	2,3	0.5
113	10	1,2,3,4	0.5

In the data above, i gives the number of clusters to create and V2 gives the attributes to be chosen where
1-Sepal.Length,
2-Sepal.Width
3-Petal.Length
4-Petal.Width

** The following values of number of clusters(i) and attributes(V2) give the highest accuracy of clustering **

```
In [10]: head(aggresult[order(aggresult$V7, decreasing=TRUE),c(4,5,7)])
```

	i	V2	V7
26	3	4	0.923076923076923
27	4	4	0.923076923076923
74	3	3,4	0.923076923076923
106	3	2,3,4	0.92
119	8	1,2,3,4	0.904761904761905
28	5	4	0.891891891891892

To check some good and bad models based on clustering

0.0.2 Examples of Good models

number of clusters(i)=3, attributes(V2)=4(Petal.Width)

```
In [11]: set.seed(1)
         model<-kmeans(data[,c(4)],3)

         table(model$cluster, data[,5])
```

```
   setosa versicolor virginica
1      50          0         0
2       0         48         4
3       0          2        46
```

Number of clusters(i)=8, Attributes(V2)=1,2,3,4(Sepal.Length,Sepal.Width,Petal.Length,Petal.Width)

```
In [12]: set.seed(1)
         model<-kmeans(data[,c(1,2,3,4)],8)

         table(model$cluster, data[,5])
```

```
   setosa versicolor virginica
1      28          0         0
2       0         22         2
3       0          2        19
4       0         14         1
```

```
  5      22           0          0
  6       0           0         11
  7       0           0         16
  8       0          12          1
```

In the above model,
Cluster 1 and 5 would be Setosa
Cluster 2,4 and 8 would be versicolor
Cluster 3,6 and 7 would be virginica

0.0.3 examples of bad models

```
In [13]: tail(aggresult[order(aggresult$V7, decreasing=TRUE),c(4,5,7)],n=10)
```

	i	V2	V7
113	10	1,2,3,4	0.5
120	9	1,2,3,4	0.5
11	4	2	0.440677966101695
9	10	2	0.432432432432432
14	7	2	0.421052631578947
16	9	2	0.4
10	3	2	0.393939393939394
12	5	2	0.382978723404255
13	6	2	0.382978723404255
15	8	2	0.375

Number of clusters(i)=3, Attribute(V2)=2(Sepal.Width)

```
In [14]: set.seed(1)
         model<-kmeans(data[,c(2)],3)

         table(model$cluster, data[,5])
```

```
  setosa versicolor virginica
1     31          1         5
2      1         27        19
3     18         22        26
```