Spotify Data Clustering

Harshad Kumar Elangovan - 19200349

26/02/2020

Loading Data

The csv file contains data about audio features for a collection of songs extracted from the music streaming platform Spotify. For each song, measurements about audio features and popularity are recorded.

```
#Loading the data after setting up a working directory and placing the data file to
that directory.

musicdata<-read.csv("data_spotify_songs.csv",header = T)
#str(musicdata)
```

Modifying the dataset

This data has categorical data which cannot be used as key values in clustering. So, the irrelent columns are removed before working on clusters.

```
musicdata1<-musicdata[,-c(1,2,3)]
head(musicdata1)
```

```
##   song_popularity song_duration_ms acousticness danceability energy liveness
## 1              66           216933     0.010300        0.542  0.853    0.108
## 2              76           231733     0.008170        0.737  0.463    0.255
## 3              74           216933     0.026400        0.451  0.970    0.102
## 4              56           223826     0.000954        0.447  0.766    0.113
## 5              80           235893     0.008950        0.316  0.945    0.396
## 6              81           199893     0.000504        0.581  0.887    0.268
##   loudness speechiness   tempo audio_valence
## 1   -6.407      0.0498 105.256         0.370
## 2   -7.828      0.0792 123.881         0.324
## 3   -4.938      0.1070 122.444         0.198
## 4   -5.065      0.0313 172.011         0.574
## 5   -3.169      0.1240 189.931         0.320
## 6   -3.659      0.0624  90.578         0.724
```

Normalization of Dataset

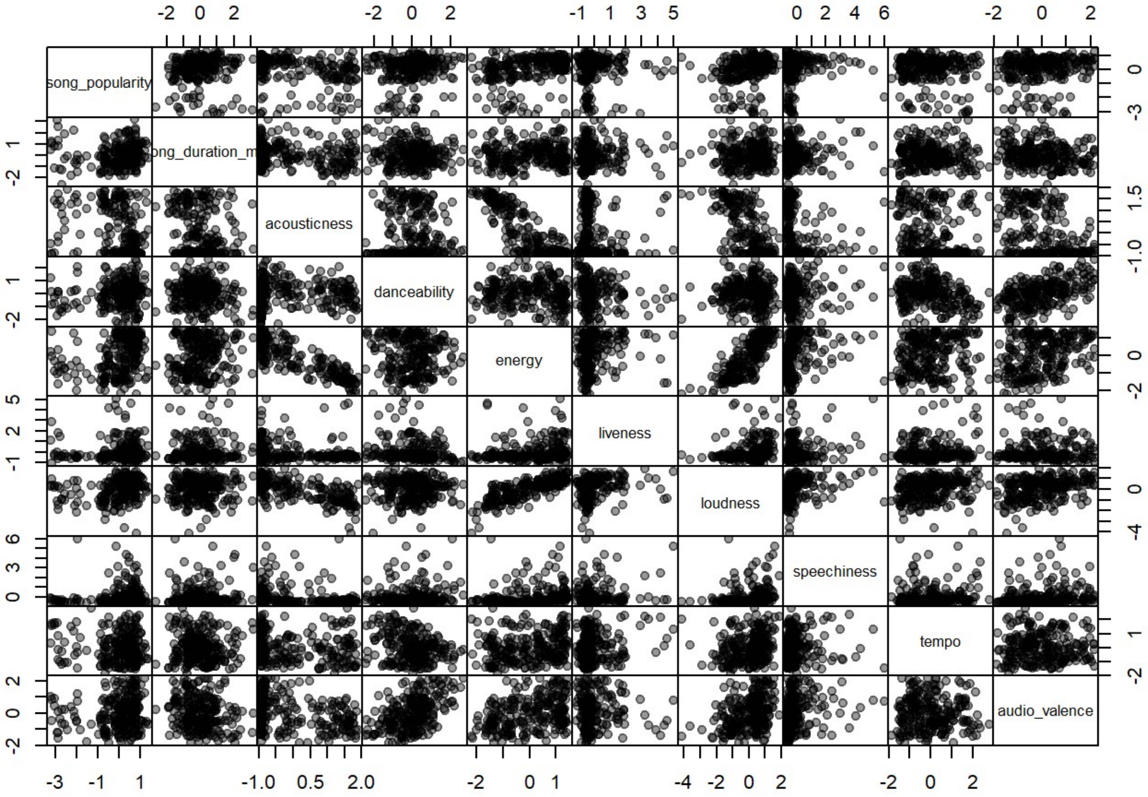
Since the values in few columns are extremely higher than other columns, we will have to scale the data for unbiased clustering of dataset.

```
meanval<-apply(musicdata1,2,mean)
sdval<-apply(musicdata1,2,sd)
musicdata1<-scale(musicdata1,meanval,sdval)
```

Plotting the data

The music data is then plotted to get an initial analysis of the data points. We will get a broader view on how the data points are scattered or overlapped among each other.

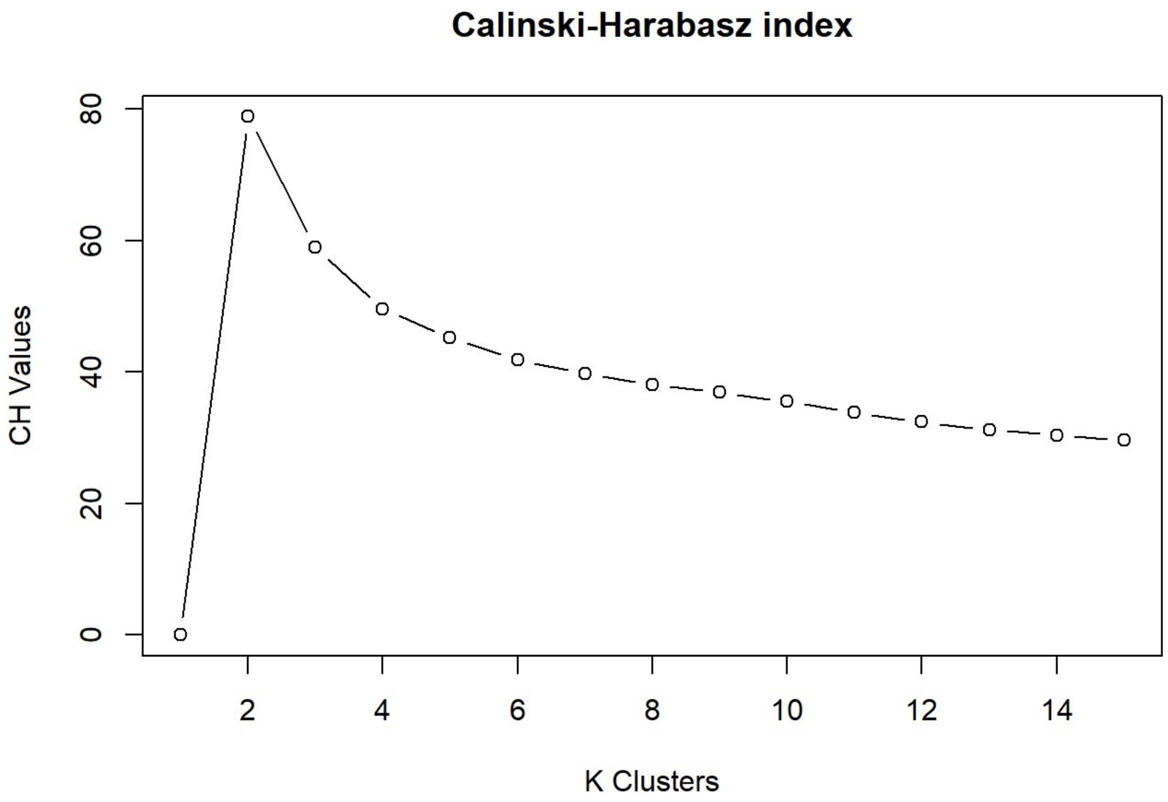
```
pairs(musicdata1,gap = 0,pch=19,col=adjustcolor(1,0.4))
```



From the plot, we can see that there is a possibility of having a minimum of two clusters. But we will have to check other possibilities as well. So, we will try to do an internal validation to verify our cluster’s count. For this, we will make use of CH Index with Kmeans.

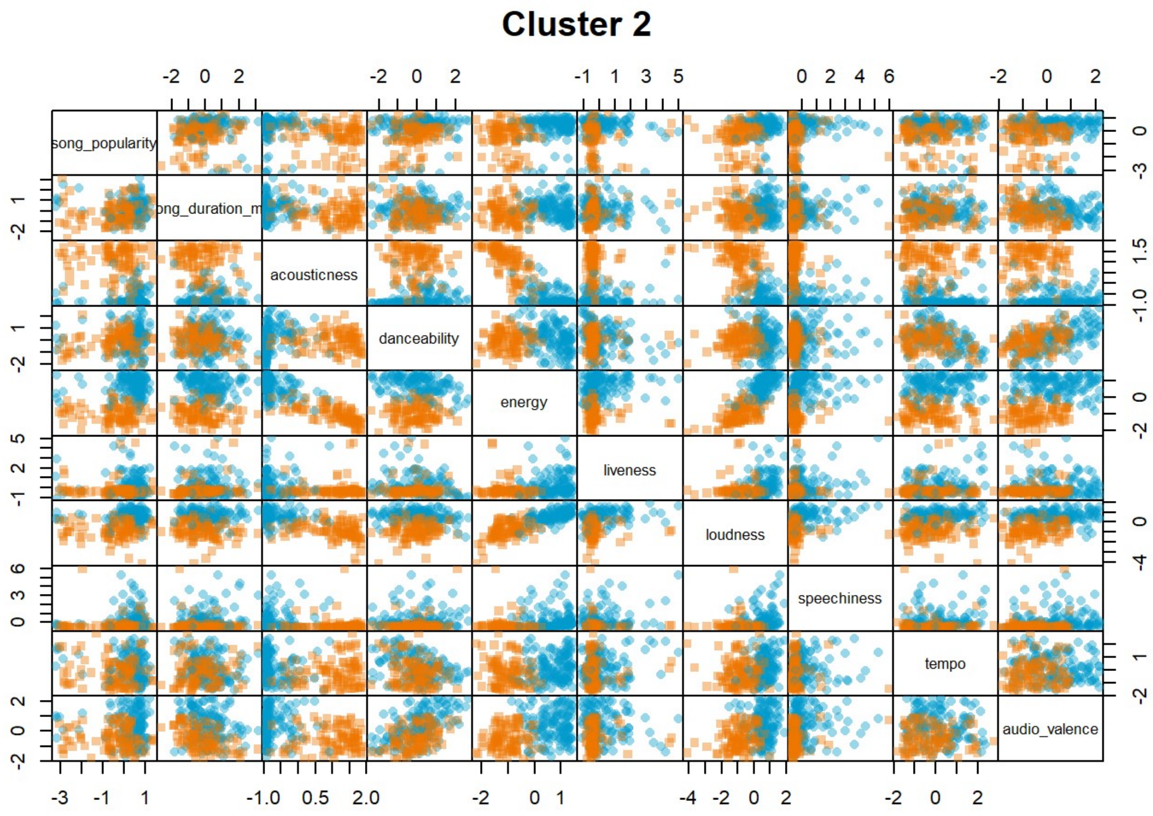
Calinski-Harabasz Index

```
k<-15
wss<-bss<-rep(NA,k)
for (i in 1:k) {
  fitmodel<- kmeans(musicdata1,centers = i,nstart = 50)
  wss[i]<-fitmodel$tot.withinss
  bss[i]<-fitmodel$betweenss
}

#computing CH Index
N<-nrow(musicdata1)
ch<-(bss/(1:k-1))/(wss/(N-1:k))
ch[1]<-0

plot(1:k,ch,type = "b",ylab = "CH Values",xlab = "K Clusters",main="Calinski-Harabasz index")
```

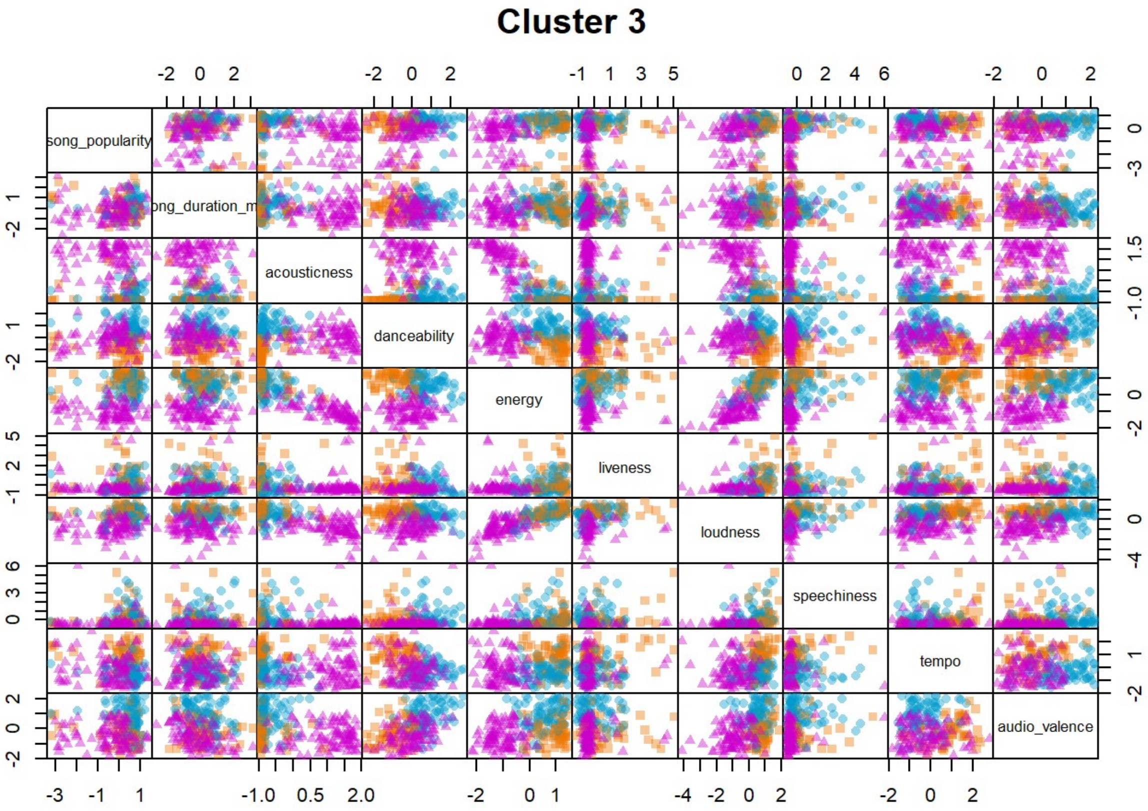


From the plot, we see that there is a sharp upward line for K=2 and then the CH Index value reduces. So, there is a high possibility that there are only two clusters. But we can compare it with three clusters and come to a conclusion.

```
fitmodel2<-kmeans(musicdata1,centers = 2,nstart = 50)
fitmodel3<-kmeans(musicdata1,centers = 3,nstart = 50)

symb<-c(15,16,17)
col<-c("darkorange2","deepskyblue3","magenta3")

pairs(musicdata1,gap=0,pch=symb[fitmodel2$cluster],col=adjustcolor(col[fitmodel2$cl
uster],0.4),main="Cluster 2")
```



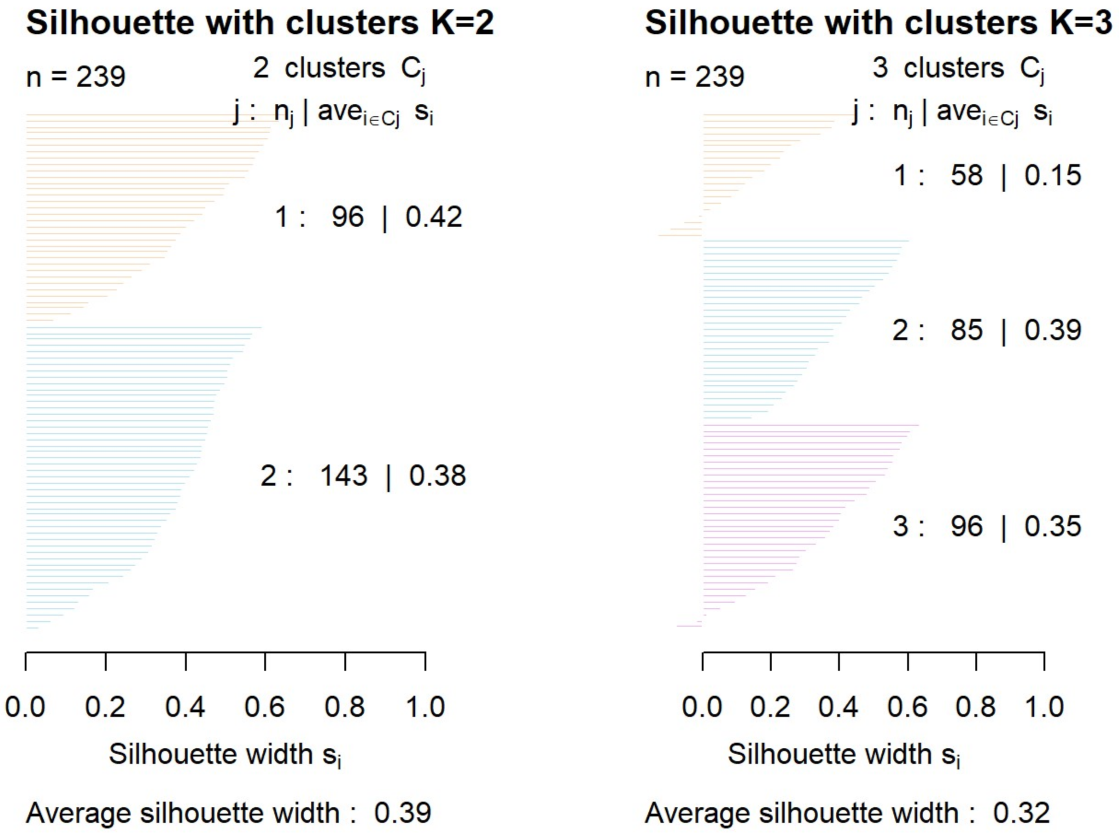
```
pairs(musicdata1,gap=0,pch=symb[fitmodel3$cluster],col=adjustcolor(col[fitmodel3$cl
uster],0.4),main="Cluster 3")
```



Silhouette

When we visualize both the plots, we can see that there is minimal overlap between the data points in the cluster 2 compared with cluster 3. So, we can come to an assumption that this dataset can be categorized using two clusters. Lets try to check the optimal cluster again using Silhouette.

```
library("cluster")

#Calculating Euclidean distance

distance<-dist(musicdata1,method = "euclidean")^2

#This distance shows the distance between the data points. This can be used to anal
yze if the data points are close or very far from the neighbouring one and can acco
unt for dissimilarity points.

sil2<-silhouette(fitmodel2$cluster,distance)
sil3<-silhouette(fitmodel3$cluster,distance)

par(mfrow=c(1,2))
plot(sil2,col=adjustcolor(col[1:2],0.3),main="Silhouette with clusters K=2")

plot(sil3,col=adjustcolor(col[1:3],0.3),main="Silhouette with clusters K=3")
```



This validation confirms that the model with two cluster is better than the model with three cluster as the model with two cluster has a higher Silhouette value than the model with three clusters.

External validation

We can verify the partition between the clusters by comparing them using External Validation. We can make use of the Rand Index and Adjusted Rand Index for this validation.

```
library(e1071)
```

```
## Warning: package 'e1071' was built under R version 3.6.2
```

```
#k=2
tab<-table(fitmodel2$cluster,musicdata$genre)
tab
```

```
##
##      acoustic pop rock
##   1        90   5    1
##   2        10  75   58
```

```
classAgreement(tab)
```

```
## $diag
## [1] 0.6903766
##
## $kappa
## [1] 0.5098251
##
## $rand
## [1] 0.7342569
##
## $crand
## [1] 0.4741481
```

The result shows that the Rand Index for clusters with k =2 is 77% which is close to 100% with adjusted Rand Index approximately 50% and proves that there is a good agreement among the clusters.

Conclusion

The cluster analysis on the Spotify audio file dataset shows us that there are two types of Clusters which segregated the whole data points. To confirm this, we have made use of Internal and external validations on data.This validation has confirmed that the dataset is split to two clusters and it gives a fit model. This cluster analysis can be used further in analyzing any new data that might come up in future, by plotting it in graph and calculating its dependency among these clusters.