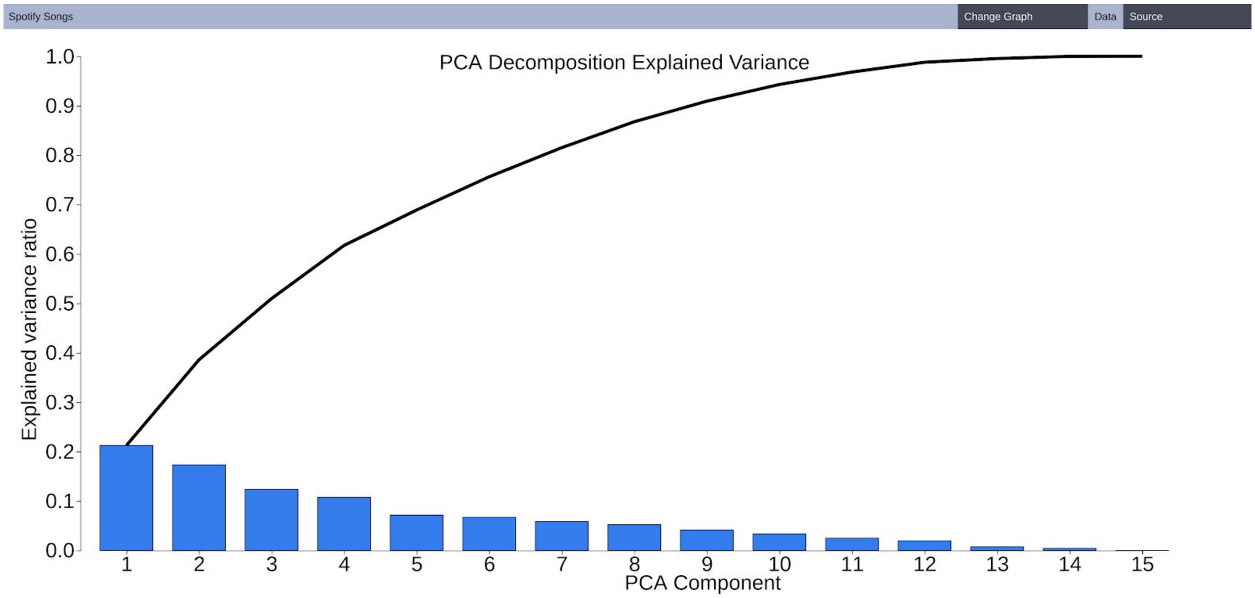
Ivan Lin
CSE564 - Spring 2021
Klaus Mueller

The back end and data processing was done with python 3.8.5. It uses the Flask, scikit-learn, and numpy. The preprocessing script also includes matplotlib. In order to obtain all requirements, a requirements.txt file is included.

In order to run the code, refer to the README for explicit instructions. The file contains a flask application located in app.py that when run, serves the application on localhost:8080. Much of the information is preprocessed for the sake of speed, with the information stored in csv files served statically. “Pca.py” contains the script for preprocessing the data. “Pca_reduction.csv” contains the information of the components generated by PCA and “spotify_numeric_norm.csv” stores the processed numeric attributes.

I used the Spotify Songs dataset I formed from the first lab. I made no changes to the data, except for rescaling the “Loudness” attribute, which previously was all negative decibel values. Since decibels measure loudness on a log scale, I noticed that results of the clustering algorithm were noticeably better and clearer to visualize when I changed it back to a positive scale. I also scaled the data from 0 to 1 for similar reasons - without the scaling, PCA decomposition was dominated by two dimensions and the explained variance of the other dimensions were negligible.

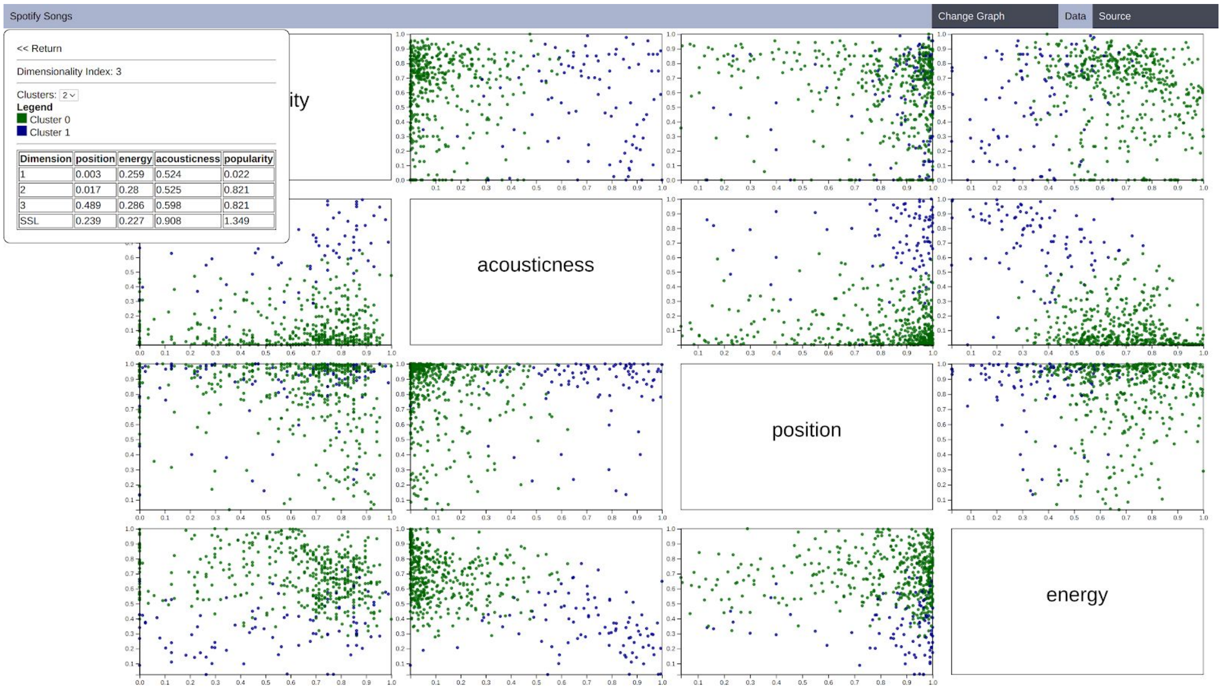
The first page is the scree plot, matching the components generated by the scikit-learn PCA decomposition to the value for their explained variance ratio. The line is the cumulative value of the explained variance.



Scree Plot

By clicking on a bar, the user chooses a dimensionality index. If the user were to select bar 3, the program would use a dimensionality index of 3. This generates a scatter plot matrix measuring the four most influential attributes based on their sum square loadings on the three most influential matrices. The y-axis for each scatter plot is the attribute in the same row while the x-axis for each scatter plot is the attribute in the same column. So the second scatter plot

The legend is available in a condensed form. By hovering over the legend, the user is able to expand it to see a table. On the table, each row represents the *n* PCA components determined by the user-selected dimensionality index, *n*. The columns with attribute names are measures of the loading of each attribute on a PCA dimension, determined by the attribute's dot product with each PCA component. The last row is the sum-square of the loadings (SSL).



Dimensionality Index: 3

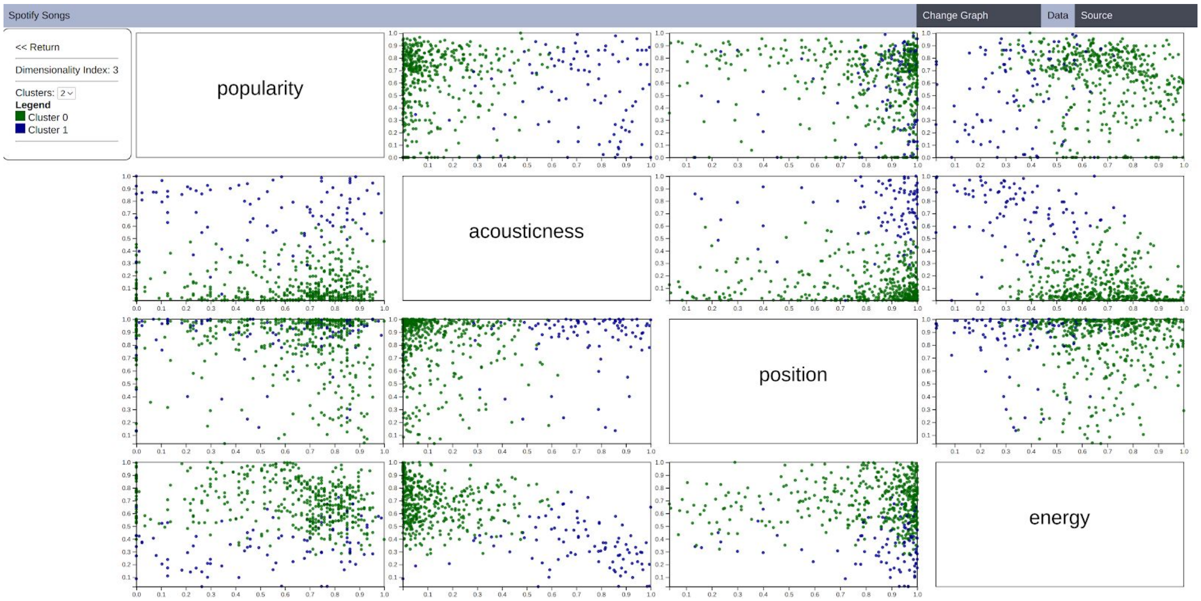
Clusters: 2

Legend

🟩 Cluster 0

🟦 Cluster 1

Dimension	position	energy	acousticness	popularity
1	0.003	0.259	0.524	0.022
2	0.017	0.28	0.525	0.821
3	0.489	0.286	0.598	0.821
SSL	0.239	0.227	0.908	1.349



Dimensionality Index: 3

Clusters: 2

Legend

🟩 Cluster 0

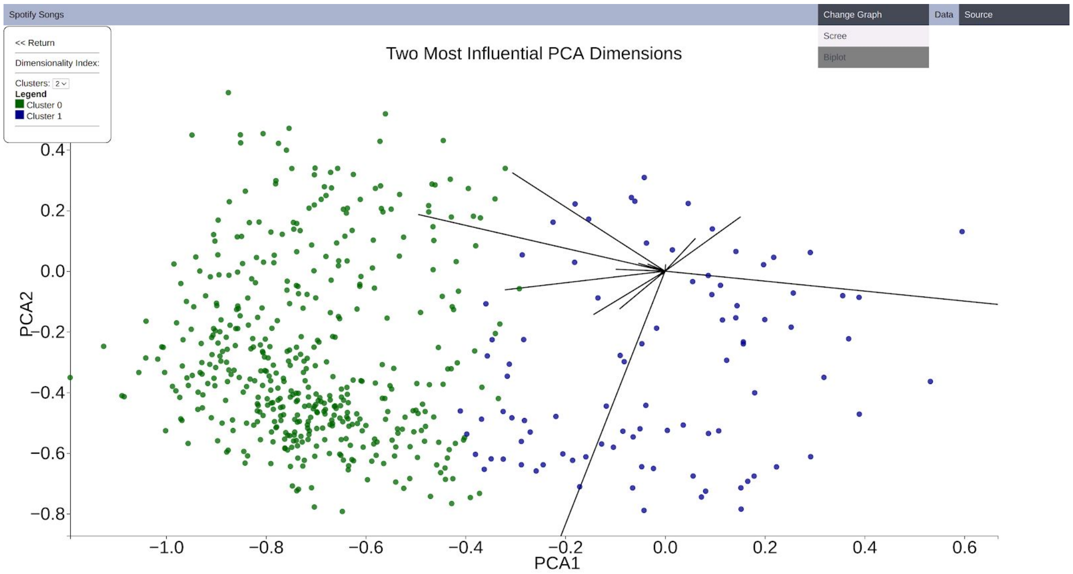
🟦 Cluster 1

Scatter matrix

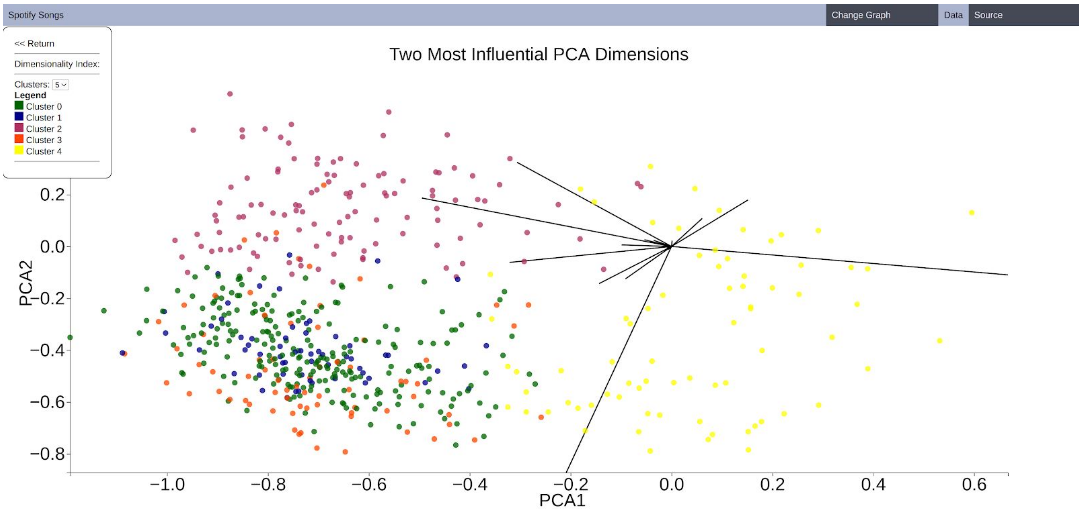
To return the scree plot and change the dimensionality index, there is a “Return” link in the legend. To view the biplot of all attributes based on the coordinate graph defined by the two most influential PCA components, there is a button to change between the scree and the PCA biplot in the top right corner labelled “Change Graph.”

The scree plot graphs all the points on the coordinate plane defined on the x-axis by the most influential PCA components and on the y-axis by the second most influential component. The vectors representing each attribute is also graphed as a vector from the origin.

Using the legend, there is also a dropdown to change the number of clusters that are displayed. This option is present on both the PCA biplot and the scatter matrix.



Clusters: 2



Clusters: 5

PCA Biplot