Project Proposal: Evaluating Trends through Clustering

Khawar Murad Ahmed
Shreeman Gautam

ACM Reference Format:
Khawar Murad Ahmed and Shreeman Gautam. 2022. Project Proposal: Evaluating Trends through Clustering. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

The question that we would like to study has slightly changed. The main questions we'd like to study are the following: How does music differ as time passes? What were the different musical trends across generations? For that, we are using a dataset that we found on kaggle.

The dataset that we plan on using is a list of around 160,000 songs from Spotify that was found in Kaggle. It includes columns such as the name of the song, artist, the date in which it was released, danceability, acousticness, energy, tempo, liveness, loudness, popularity, and valence among others. During pre-processing, while the dataset was from 1920-2020, we decided that our dataset would only include songs from 1960-2020 just to make it more convenient to work with (roughly speaking, while we are more familiar with music from the last 20 years, 1960s was manageable while the 1920s was not). Currently, we are running clustering to learn more about the dataset we have. For now, we ran K-means, but we intend on running agglomerative clustering as well. The reason we did clustering (which is talked about later as well) is to see what kind of separation takes place through the years and which also has the potential to identify trends. We chose to remove popularity and year from our data when we ran the clustering experiment to see how the clustering algorithm would cluster it blindly (i.e.: Without seeing the time it was released). For that reason, we are also eager to see whether agglomerative clustering gives us a different result (such datasets can be interpreted in many different ways). We also intend on using regression to predict popularity, and identifying different clusters also gives us a means of understanding musical trends to see what factor is influential.

Ultimately, we have decided that the success of our project would depend on how good of a job our regression model does with predicting popularity. The reason we think that this evaluation method is pertinent to our project is that if we have produced a regression model that can predict popularity with relatively high degree of accuracy, that means that we have done an excellent job of understanding current musical trends.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

2 TECHNIQUES AND RESULTS

As part of K-means clustering, we ran experiments to see how data would cluster for different number of clusters. We have varied the number of clusters from n = 3 to n = 6. K-means clustering was done using scikit-learn's K-means module. EVen thouhg, K-means is random, we have observed that from n = 3 to n = 6, approximately the same clusters are produced albeit in a different order with every run of the code. So for instance, for n = 6, we get 6 clusters where:

- In run = 1, we see that cluster-0 has 53459 songs, cluster-1 has 4505 songs, cluster-2 has 365 songs, cluster-3 has 40565 songs, cluster-4 has 22743 songs and cluster-5 has 19 songs.
- In run = 2, we see that cluster-0 has 40395 songs, cluster-1 has 22806 songs, cluster-2 has 4503 songs, cluster-3 has 362 songs, cluster-4 has 53572 songs and cluster-5 has 18 songs.

A closer look at the 2 runs tells us that:

- cluster-0 of run 1 is approximately the same as cluster-4 of run 2.
- cluster-1 of run 1 is approximately the same as cluster-2 of run 2.
- cluster-2 of run 1 is approximately the same as cluster-3 of run 2.
- cluster-3 of run 1 is approximately the same as cluster-0 of run 2.
- cluster-4 of run 1 is approximately the same as cluster-1 of run 2.
- cluster-5 of run 1 is approximately the same as cluster-5 of run 2.

We did 10 runs each of the code and each run produced 6 clusters(in different orders) with approximately the same 6 values(same thing holds for n = 3 to n = 5 clusters). This proves to us that despite, the randomness of K-means, approximately the same 6 clusters are produced albeit in different orders.

We then put our clusters into histograms. As an example, for n = 3, we get 3 clusters. For each cluster, we made a histogram. On the x-axis of the histogram, we specify 6 intervals where each interval represents roughly a decade[1960-1969, 1970-1979, 1980-1989, 1990-1999, 2000-2009, 2010-2020]. On, the y-axis, we count the number of songs for that decade. We did the same thing for the other n values as well.

From n = 3 to n = 6, we get 3+4+5+6=18 histograms but we will show 3 histograms from n = 6 clusters because:

- We observe that these 3 histograms contain the majority of songs(from a total of 121656 songs in the pre-processed data, these 3 histograms contain 22741+53474+40550 = 116765 songs while the other 3 histograms contains 121656-116765 = 4891 songs)
- n = 3 to n = 5 also contain 3 histograms each, where the 3 histograms end up containing the majority of the songs

- Each of the 3 histograms indicate a clustering based on a generation. Generally, a generation is defined as 25 years. Generally, we see that songs from 1960-1979(20 years) ended up being a majority in the 1st cluster, songs from 1980-1999(20 years) ended up being a majority in the 2nd cluster and songs from 2000-2020(21 years) ended up being a majority in the 3rd cluster.

The last bullet point goes against our hypothesis because we thought that one decade would dominate one cluster(what we are saying is that we expected cluster-0 to have a majority of songs from the 1960-1969, cluster-1 to have a majority of songs from 1970-1979, ... cluster-5 to have a majority of songs from 2010-2020). Even though our former hypothesis was proven false, we can now demonstrably say that generally, songs from a generation end up being a majority in one cluster.

Here are the histograms[Figures 1-3]:

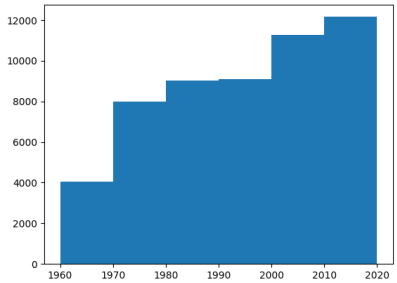

Figure 1: Cluster-0

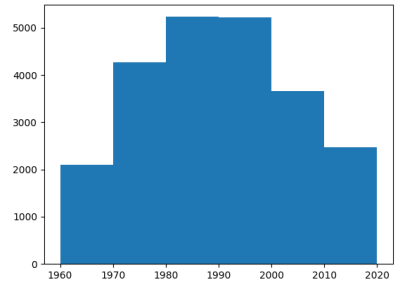

Figure 2: Cluster-1

Now, we have to emphasize that these are general trends. During our experiments from n = 3 to n = 6, we noticed that songs from 1960-1969 and 2010-2020 would end up in a cluster together. You can see that in figure 3 where the number of songs from 2010-2020 is the second highest population in the histogram. Here is another histogram from n = 4 which shows the same thing**[Figure 4]**, but the populations are bigger for 1960-1969 and 2010-2020 compared to figure 3:

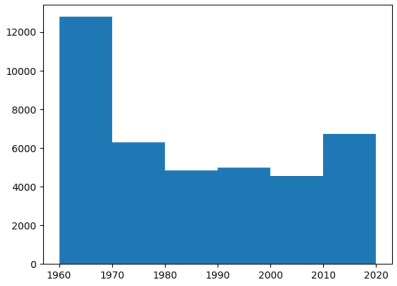

Figure 3: Cluster-2

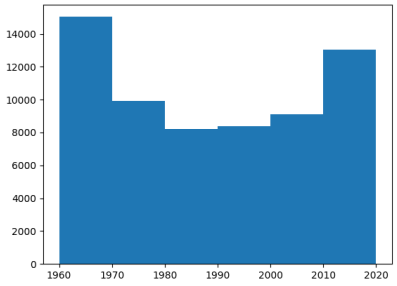

Figure 4: 1960-69and2010-2020

3 NEXT STEPS

As previously mentioned, our main motivation for using clustering was to find out more about the dataset we have and look for similarities. For example, one of the more surprising findings was that there was a cluster that would group music from 1960s and 2010s in the same cluster, consistently. Because of that, we are going to look more into the properties of each cluster and see what feature(s) dominates each cluster. How this might inform our regression model is, if for example we take the example we just gave with 1960s and 2010s music being clustered together, then we might include some of the 1960s music in our model to predict the 2010s trend. That is why we are keen to use hierarchical clustering to see if that algorithm picks up a different trend altogether, or whether it gives us similar results.

For regression, our general plan is to train a model with music from some number of years, and use it to predict the popularity of the following years. For example, we use music from 2010-2015s to train, and use that model to predict the popularity of music from 2016-2018. How successful we are would depend on how accurately we would have predicted the popularity.