Finding groups and patterns in a song dataset, Project.

Rafael Pérez Estrada
Escuela Nacional de Estudios Superiores, Universidad Nacional Autónoma de México
tinoco21.30@gmail.com

Fernando Nateras Bautista
Escuela Nacional de Estudios Superiores, Universidad Nacional Autónoma de México
fnaterasb1@gmail.com

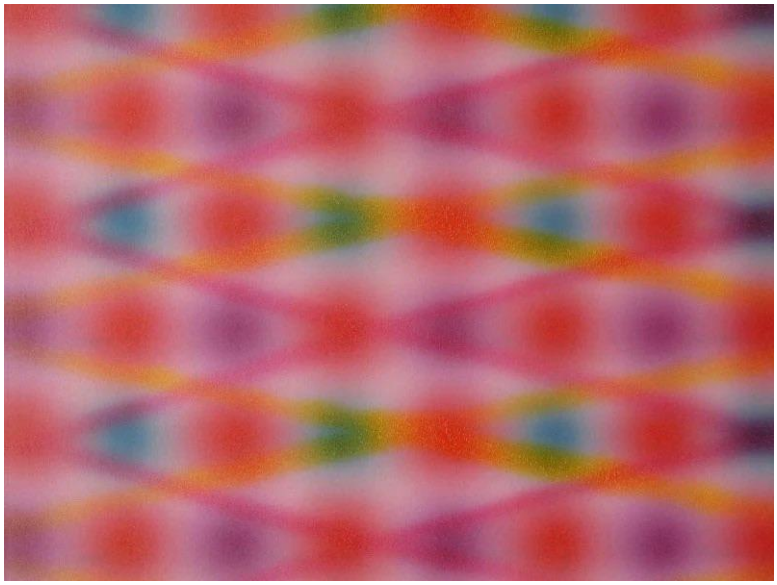

Figure 1: Illustration by Jozsef Bullas

ABSTRACT

In this work we apply different unsupervised learning techniques to find patterns and groups in our untagged data. Our work is based on a dataset of songs that we built with the Spotify API. The techniques we used are clustering and frequent patterns, additionally we used visualisation as a tool to have a better understanding of the data we have and its components. In this work we present our dataset, the methodology we used and the discussion of the results that we have obtained.

KEYWORDS

Unsupervised learning, patterns, visualisation, songs, features

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

ACM Reference Format:
Rafael Pérez Estrada and Fernando Nateras Bautista. 2021. Finding groups and patterns in a song dataset, Project.. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Unsupervised learning is a branch of machine learning where given data comes with no labels. Without the labels the algorithm is capable to find patterns or similar groups in the data. Unsupervised learning can also be a one step prior to a supervised learning technique, prepossessing.

Some of the most popular unsupervised learning techniques are, clustering, apriori algorithm, association rules, among others. In this project we use K-means clustering to find groups in our data and the apriori algorithm that help us to find frequent patterns in our data. Another tool we use is visualisation. Visualisation is not a technique of unsupervised learning but is a really helpful instrument to have an overview and a better understanding of the data we have. Visualisation is a wide branch of computer science, with its own rules and methods. Having a good graphic can be easy but also it can be really easy to make tremendous mistakes that lead to a bad data representation or a graphic where the people

cannot understand a thing. We widely recommend reading The Visual Display of Quantitative Information by Edward R. Tufte
Nowadays thanks to the streaming services we all have access to millions of songs within seconds, the songs we hear, the songs we like, the ones we add to our playlist, all of this is data that we can turn into information. With the help of unsupervised learning techniques we can turn this fuzzy data into something we can work with and obtain information of the songs, artists, users, etc. We used K-means clustering to discover groups in our dataset, so we can have a more general view of the songs we have, and identify their main features. Additionally we use the apriori algorithm to find the features that come together more often.

2 METHODOLOGY

We aim to evaluate our dataset with unsupervised learning techniques, so we can obtain interesting patterns or groups in our data as well we want to have an accurate description of our dataset after all the processing and give a good interpretation of our results. For this project we collected metadata of songs with the spotify api. In addition we manually added more attributes to each song. After we finished building our dataset we continue with the preprocessing.

2.1 Preprocessing

The preprocessing consists in dropping all the columns we did not need, so we do not have interference. After this we transformed our qualitative data into quantitative data. To make this we assigned a number to all the unique values in a column. We did this to all the columns that were not in a numerical format.

Something really important that we became aware of is the importance of data normalization. For this project it was really important to normalize our data because we have some attributes that were in the range of 0 to 1 and on the other hand, we have attributes that go all the way up to 100000. So this bigger attributes were the ones that really define the cluster groups. This was not what we wanted, we wanted that all the attributes to have the same weight at the time of making the clustering. To make the normalization of our data we used StandardScaler form the scikit-learn library.

2.2 Methods

For our analysis we used two unsupervised learning techniques and data visualisation.

2.2.1 K-Means.

K-Means is a form of clustering that with a given k it finds the k number of groups in the data. It chooses k centroids and with distance measures it assigns the data to its closest centroid then it computes the mean of all the data points and it chooses new centroid. This process repeats until the centroids do not change position.K-Means helped us to make groups in our data, it made different clusters with similar songs in each of it.
For this project we used the K-Means implementation from the scikit-learn library.

2.2.2 Frequent patterns.

To get to know the frequent patterns in our data we use the Apriori algorithm. Some of the attributes we had were continuous data, so before performing this task we discretize these attributes.
We used this technique to discover which features come together more frequently and find patterns that satisfies a minimum threshold value for support and confidence. From the we obtain information that can complement the groups information we have and we can have a wider overview of the data.
For this project we used the frequent patterns implementation from the mlxtend library.

2.2.3 Visualisation.

As we know, visualisation is a powerful tool when it comes to communicate and represent our data. With good data representations is more easy for us and for the other people to understand the data and results without knowing a lot of the processes we have made. We made a streamlit dashboard with different graphics and visualisations.

3 DATASET

The experiments were made in a dataset we built, with data of 1001 songs. The songs were gathered by us. With the help of the Spotify API, we got the technical features of each song, such as, danceability, key, instrumentalness, etc. later on we will describe each of these attributes. In addition of these technical features, the spotify api also provides the name of the song, and the name of the artist(s). To have more accurate information we added columns that we filled manually, these columns are gender, type, country, number of integrants, song language, do we like?, latitude and longitude of the country. Dataset distributios is shown in Table 1.

3.1 Attributes description.

- **Danceability:** describes how suitable a track is for dancing based on a combination of musical elements. A value of 0.0 is least danceable and 1.0 is most danceable.
- **Energy:** is a measure from 0.0 to 1.0 and represents a perceptual measure of intensity and activity.
- **Key:** The key the track is in.
- **Loudness:** the overall loudness of a track in decibels (dB).
- **Mode:** indicates the modality (major or minor) of a track. Major is represented by 1 and minor is 0.
- **Speechiness:** detects the presence of spoken words in a track. The more exclusively speech-like the recording (e.g. talk show, audio book, poetry), the closer to 1.0 the attribute value.
- **Acousticness:** a confidence measure from 0.0 to 1.0 of whether the track is acoustic. 1.0 represents high confidence that the track is acoustic.
- **Instrumentalness:** predicts whether a track contains no vocals. The closer the instrumentalness value is to 1.0, the greater likelihood the track contains no vocal content.
- **Liveness:** detects the presence of an audience in the recording.
- **Valence:** a measure from 0.0 to 1.0 describing the musical positiveness conveyed by a track.
- **Tempo:** the overall estimated tempo of a track in beats per minute (BPM).
- **duration_ms:** the duration of the track in milliseconds.
- **time_signature:**the time signature (meter) is a notational convention to specify how many beats are in each bar (or measure).

- **song_name:** name of the song.
- **artist(s):** name of the artist(s) that appear in the song.
- **gender(of the atrist(s)):** gender of the artist(s), F for female, M for Male, V for songs where there are more than just women or men, it also applies to non binary artist(s).
- **type(solo, colab, grpup/band):** type of agrupation, S for solo, C for colab and B for band.
- **country:** the country where the main artist is from.
- **number_integrants:** number of participants in the song.
- **song_language:** language of the song or No/Lang if it does not have vocals.
- **label:** this marks if we liked or not the song, F for Fernando liked or R for Rafa liked it and RF or FR for both of us liked.
- **lat:** latitude of the country.
- **lon:** longitude of the country.

Table 1: Dataset stats

Songs Dataset	
Set	Songs
Rafa's	501
Fer's	500
Total	1001

4 EXPERIMENTS AND RESULTS

4.1 K-Means

For clusterin we based oir experimentes in three subsets, one is conformed of the songs only Fer liked, the other is conformed with the songs only Rafa liked and the third one is made of the songs both of us liked.

Our firs step in the clustering task with K-Means was to set a value for the k in each subset, for this we used the kelbow visualizer and the Silhouette visualiser.

For the Fer's and Rafa's set we used a k value of 8, we based this decision on the graphic that the kelbow visualiser provides, in the graphic it told us to use a value 6 for the number of clusters but we decided to use a k of 8 to have more groups so we can have more detailed groups.

This value gave us an output of 8 different groups of our data. The groups we obtained are explicitly on the jupyter notebook at our github. But mainly we can observe the groups in our streamlit app, because in the notebook we obtained the groups normalized so its kind of hard to interpret the results. In the streamlit app we got the labels of each song and when we put the labels in the dataset for the attributes we obtained the mean, standard deviation and the mode, so we can have a better view of the groups we have and a better interpretation of the songs in each cluster. Firstly can observe that the main features it took to make the groups are the location, energy, loudness and danceability. Because of the nature of the values of country it has a little more weight in making the groups.

4.2 Apriori

In this work we wanted to analyze frequent patterns in the music we both liked. We used the apriori algorithm so we needed to choose a threshold. First, we decided to choose a very high threshold because there were many frequent patterns due to the number of attributes. When we chose such high threshold, we were concerned about the patterns that we found, this was because the patterns were not very interesting or important at all. Many of these patterns were even obvious, like the English language comes together with united states or England, also that with a number grater than 1 in the number of integrants it comes together with colab or bands, etc. That is why we decided to choose a lower threshold and we certainly found some more interesting patterns. We think this is because the different music taste we have. As we just took a random sample of our music taste and we like very different type of music, we did not have enough data so that interesting patterns could really stand out, so in this case the only way to do it was to lower the threshold. After we lowered the threshold, we fund some very interesting patterns in our music taste. First, we found out that the song we both most listen to are in English. Also, we found out that we like energetic music but with low valence, that means that we prefer energetic and loud music, but also sad or angry (negative music) such as rap or hip hop. Furthermore, another very interesting pattern was that we like music in Korean but most of the time when the singer is female. We found a lot of patterns like this and we probably should keep analyzing them but for the purpose of this project we mentioned just some of them.

5 CONCLUSIONS

In conclusion we have found which kind of music we listen to, not only by knowing the tracks but by analyzing the groups we obtained and its patterns. Another thing is that we not only van make groups of songs we like we can also make the groups by the artist and with enough information of the artists and the songs of them we could get to know the features of the artist and also discover similar artists.

We have come to the conclusion that this work sets a base for developing something bigger, with a bigger and richest dataset. It could grow to be a recommendation system for new songs you might like or to recommend new artists. In fact we thought about how selecting different attributes can affect the clustering of the songs, that is why we think this project could also lead to a dynamical recommendation system that could generate playlist according to certain factors such as loudness, valence, energy, etc. So you could cluster your own music and get new music suggestions. In a next part of this project we would try to use a much bigger dataset and we would try to choose the attributes more carefully so that we could generate this cluster-frequent patterns based playlists.

At the end many questions were left unanswered, but we found out that this method could really work well on generating music clusters with the music features that the spotify API can provide us.