DATA ENGINEERING PROJECT 2024

Stanislav Bulatskii, Tomàs Ockier Poblet, Hermes Barreiro Pena

Universitat Autònoma de Barcelona

ABSTRACT

In this project we needed to utilize the knowledge we have acquired in Data Engineering course with a dataset of our own choice. We chose 2 datasets with Spotify songs and with them made k-means and k-neighbours clustering models for countries and popular artists. In the process we discovered that the first dataset is not suitable for the task and the overall objectives of this project. With the second one with could accomplish good K-neighbours clustering model, which would give us a good accuracy score between 60 and 80 precents.

1. INTRODUCTION

The first dataset that we chose was one focused on the popularity of the songs, as the description of it explains:

> *This dataset presents the top songs currently trending for over 70 countries.*

And in theory it is updated daily. It consists of a total of 25 features with a datapoint count of 796253 at the time of writing. Some of this features are: spotify_id, name, artists, daily_rank, daily_movement, weekly_movement, country, snapshot_date,popularity, is_explicit, duration_ms, album_name, album_release_date, danceability, energy, key, loudness, mode, speechiness, acousticness, instrumentalness, liveness, valence, tempo, time_signature.

Note how there is a *snapshot_date* column, this is because the dataset is formed by many snapshot that are used to record the popularity of the songs in the data set. This results in far greater number of total datapoints in comparison to the unique songs represented in the dataset. We will develop more this fact further in the report, as it is part of the reasons that prompted us to change it.

The second dataset is much more well balanced overall, instead of focusing on the song popularity, it is just a compilation of approximately unique 30000 songs from Spotify.

It has all the features previously mentioned minus the snapshot date, it not a sequentially dataset, unlike the other one. On top of the features that correspond to characteristic of the songs themselves -e.g. tempo or loudness- it has genre classification and album and playlist metadata.

As we are going to explain, this dataset is far more suitable for this project.

On the next section of this document we are going to detail how the two datasets compare, explaining the reasoning behind our change. First we are going to dive into some basic statistical analysis of the first dataset, to then try to implement some of the models that we learned on class.

Following this section will be some implementation of models in the second dataset and their presentation of the their results.

Finally we will expose our conclusions to finish this report.

2. EXPLORING THE FIRST DATASET

The dataset is taken from Kaggle. Note it is updated daily, therefore the amount of data can change.

2.1. Basic Statistical Analysis

As explained previously, the dataset has nearly 800000 thousand different songs, with each one having 13 features that correspond to the characters of the songs itself -from now on, song data-, and other 8 that corresponds to song "metadata" like release data or popularity. Finally there are 3 identifiers, `spotify_id`, `name` and `artists`[1]. Will will be refereeing to each type of features by the names used here.

Each unique song is represented by their `spotify_id` not by their name. Because of the snapshots taken, a unique song can be represented multiple times in the dataset. Therefore, very popular artists are going to be overrepresented in the dataset. The most significant case is Taylor Swift, as it can be seen in the figure 2 and 1.

The idea from the beginning was to make clustering to determine the genre of the songs. Since the dataset did not include this information, it could be a good idea for us to create it.

When we started exploring the first thing we noticed is the correlations between the features, as seen on the heatmap in the figure 3. The correlation between most of the features is really small and we could not really use it for k-means.

The features that strongly correlate with each other are the ones related to the ranking of the songs and the some other features that are always going to have correlation, like loudness and energy.

[1] It can have multiple artists in case it is a collaboration.

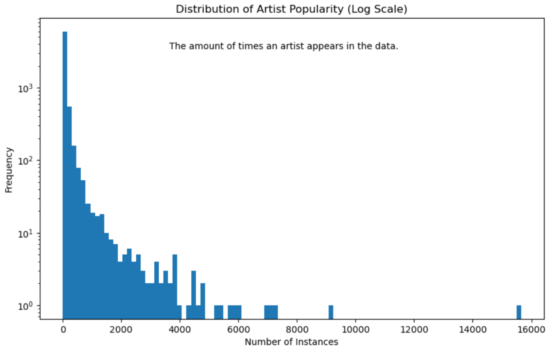


Fig. 1: Frequency in which the artists appear in the data. Note that the graph is in log scale. The data is clearly not well distributed. The right-most bin corresponds to Taylor Swift and the second right-most to Tate McRae. See figure 2 for more information.

Due to the lack of information in the collection of this features we are not capable of properly interpreted this results. Since we do not have an understanding of what ”energy” or ”loudness” are for example, we cannot properly treat them. However, we can probably join the features, as done in a Principal Component Analysis (PCA).

2.2. PCA and Selection of Features

Since we have two clear groups of numerical features, song popularity and song data we need to decide which one we keep to perform predictions or clustering. This of course depends on the which features or aspect of the song we want to cluster or classify.

First we want to show an interesting aspect of a dataset, that tells us how to perform a PCA on it.

As it can be seen in figure 5 the PCA cannot be performed with the repeated songs taken into account, as this aspect is the main culprit of this behaviour.

If we take one unique song that is present on two different datapoints, we can see how its identifier and song data are exactly the same.

We saw this fact early on in the project, and to both reduce the number of datapoints and get rid of this behaviour we dropped the duplicated on over model that we tried to create.

2.3. First dataset K-neighbours Clustering

After which we created a k-means clustering model to see the contents of the clusters.

At first we made a clusters based on 7 artist and tried to see the amount of said artists in each cluster. The fist thing we noticed is how only Taylor Swift is dominant in any clusters, other artist did not have such amount of songs in different clusters, so after that we checked the correlation heat map in the 3rd cluster. The results did not change from the general correlation heat map.

We decided to try and make a k-neighbours classification based on artists and see the accuracy of it for k-neighbours from 1 to 50. The results were very bad and the accuracy, and we will soon know why it is that what (describe here please the process of what we do in code)

2.4. First dataset K-means Clustering

Seeing as we got a bad representation of the artists in the data set -see figure 2, we tried to make the k-means clustering using 6 top artists with at least 50 unique songs.

This was in bane as, the accuracy once again was poor. This is to be expected, since the main features we used do not correlate that well and have not a lot of value to define a song.

In the dataset we can see that there is only one artist with at least 50 unique songs and it is Taylor Swift2b, since we had to drop the duplicated songs, this is a very important fact. Now could understand why the K-clustering model is bad, we do not have so many unique values from different artists to work with.

After narrowing down the amount of unique songs to 5 per artist we now have a big selection of artists to try and make a k-means model. But the accuracy is very depended on the combination of the artists, so we made code to search for the first combination which is at least 70 percent. And after some time the code gives us combination of five artists *Taylor Swift*, *PRO8L3M*, *Dětský sbor Camerata*, *Shallipopi* and *Free Finga*.

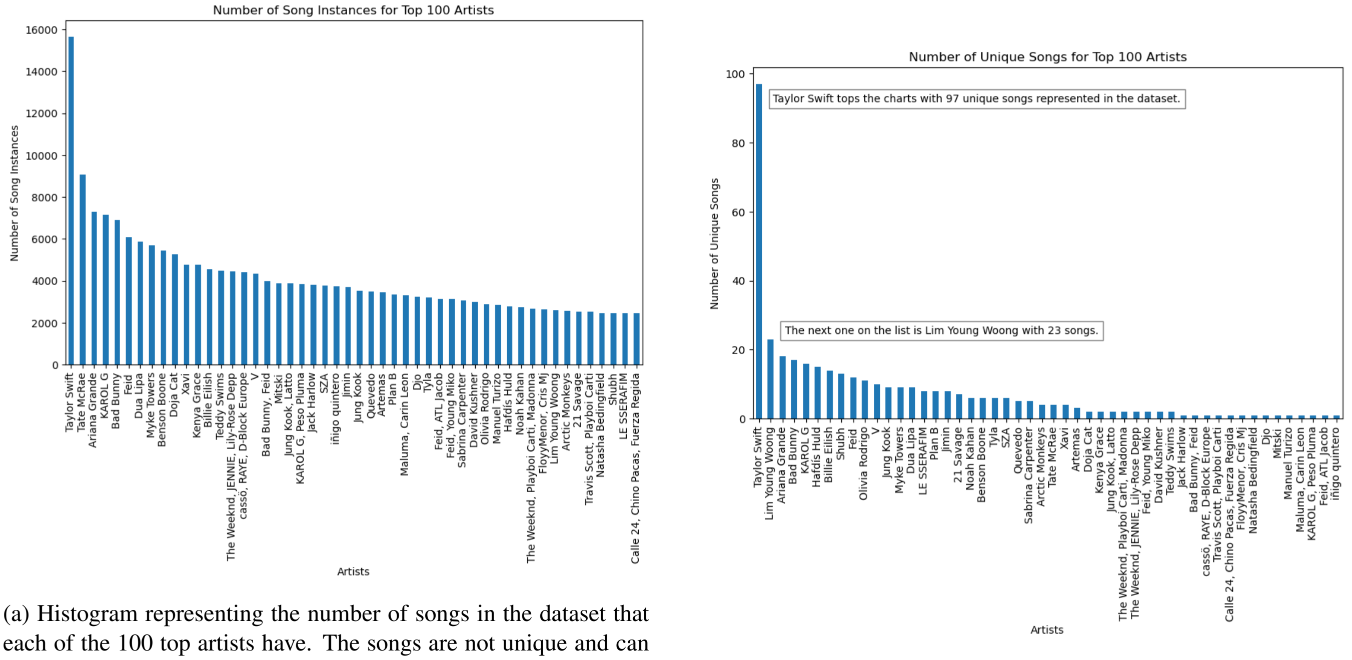
2.5. Changing the Dataset

At the time we came to the conclusion that this was not enough data make either artist prediction or genre classification.

From what we can see, it is clear that this dataset is not the bet for our current objectives, since there are not enough unique values for K-neighbours, and the features are not correlated enough in order to perform a K-means.

The value of the dataset in theory relies in the popularity. It seems that it can be ideal to make a prediction model by country of the popularity. Because the popularity is presented as a sequential value that changes day by day. Since we do not have the tools to analyze this kind of data, we believe that this was not the right dataset for this project.

Other methods out of the scope of this course seem to work in doing. A random forest classifier seems to work in some capacity on a small subset of the dataset, in trying to predict the artist for each song. Since it is not an algorithm given in class we decided to not included in this report.



(a) Histogram representing the number of songs in the dataset that each of the 100 top artists have. The songs are not unique and can be repeated.

(b) The number of unique song that each of the top 100 artists have.

Fig. 2: From this two histograms it is clear the there is a problem with first dataset. The most popular artists are overrepresented. From the total of 6957 artist here are represented the top 100. From graph 2b, we can see that only the top 50 most represented artist have more than 5 songs.

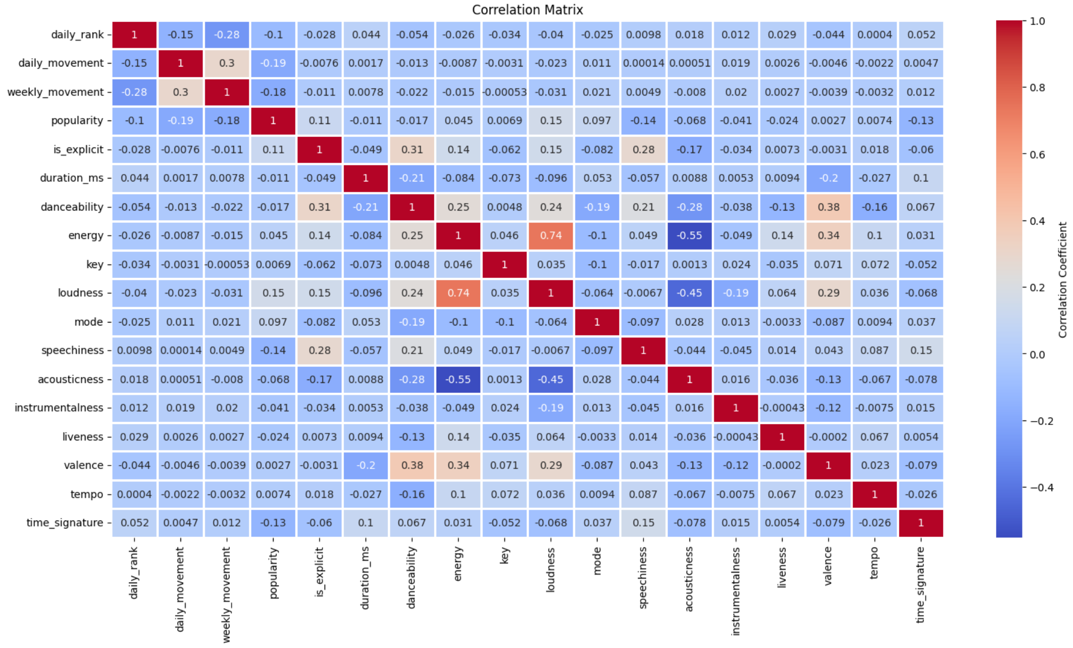
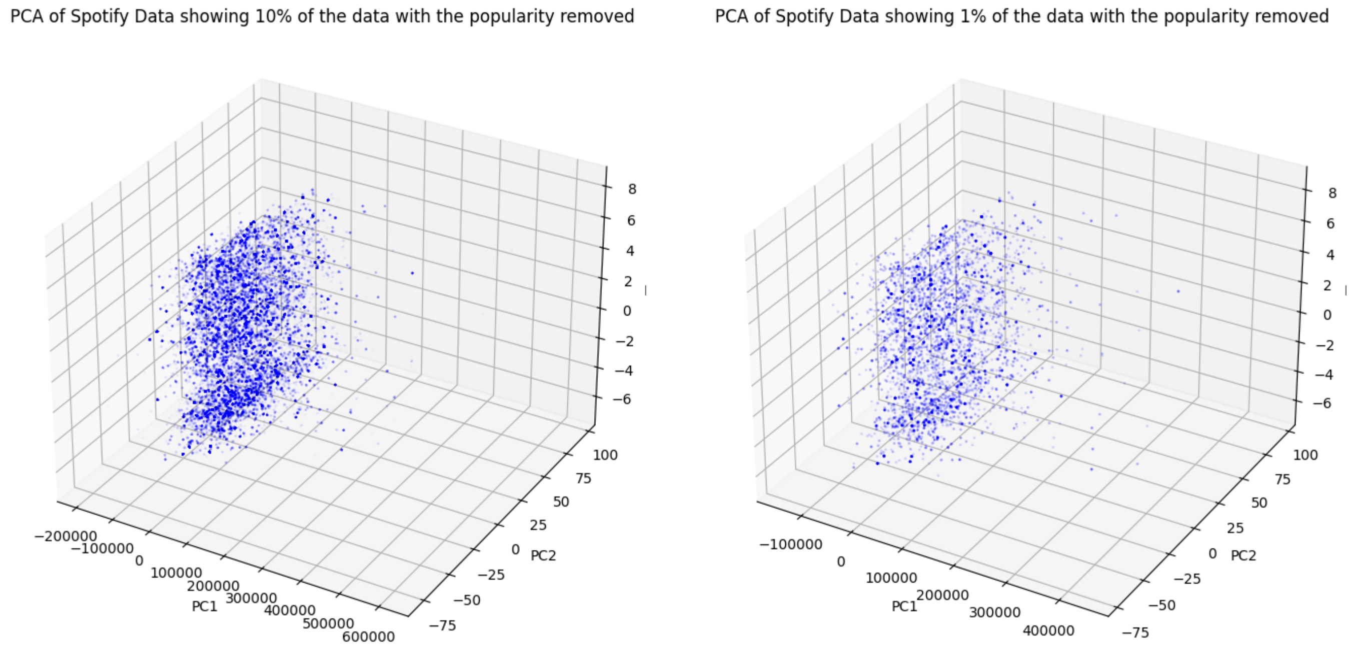


Fig. 3: Correlation matrix showing the correlation values between the numerical features of the first dataset. As seen, there is a positive correlation between some of the song data features like between energy and loudness specially. Moreover, the features related to the popularity are related to each other inside their own group. However, when comparing them to the song characteristics, we can clearly see that there is barely correlation. We believe that is falls under the category of statistical noise.



(a) Showing a random sampling of 10% of the songs.

(b) Showing a random sampling of the 1% of the songs.

Fig. 4: Showing the first three PCA components and reducing the size of the scatter dots, we can clearly see how the data it organized in columns that conform the 3rd PCA. As we understand this is because in the 3rd component the popularity features are clustered together -Not the snapshot data, just `popularity`, `dairly_rank`, `weekly_movement` and `daily_movement`-. When taking out the popularity related variables its when this behaviour disappears.

3. SECOND DATASET

As explained previously we decided to change the data set. To not change the type of data what were are dealing with, another Spotify song dataset was chosen. It is also from Kaggle.

It more data points -i.e. unique songs-, and fewer features. Due to this fact, we are able to for example select 50 artists with more than 20 unique songs to execute some models.

However, before showing the model implementation we are going to compare both datasets to show how the second is a better choice than for the objectives that we set up originally.

The same type of histograms that we used in the first dataset can be found on figure applied to the second one. We can observe both a greater variety of genres in the most popular artists as a more historical variety. As an example, we can see that the most popular artist is Queen a band that has not been as popular as popular as Taylor Swift in recent years. Since the dataset is just a compilation of songs, it does not exhibit the bias found in the first one, were it was mostly composed of the most popular songs of the last few years.

Moreover, it also has new features, apart for the previously covered ”song data” features; for each song we can find the genre of the playlist to which it belongs, as well as the sub genre. Therefore, we are able to use both this features to make predictions about them, and see if there are enough features and correlations between to identify the genres or sub genre.

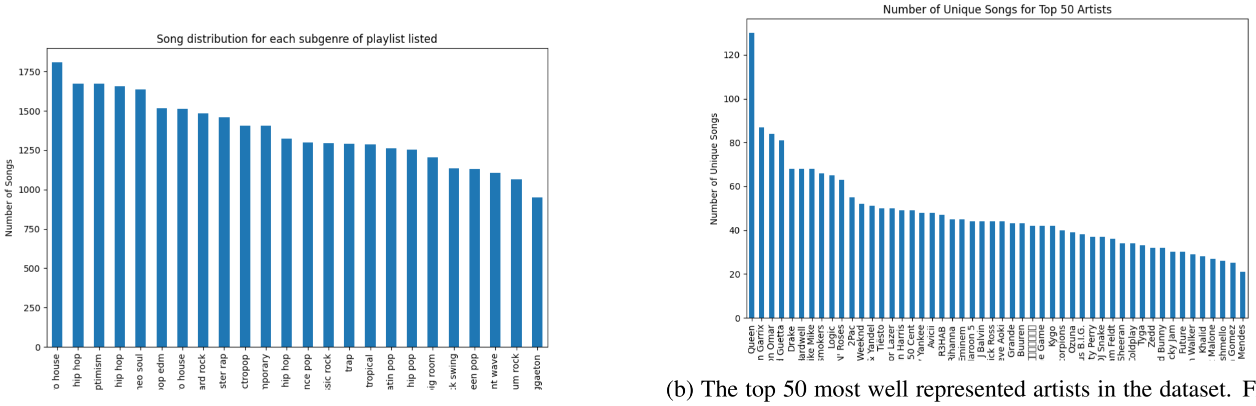
3.1. Second Dataset K-Means and K-neighbours

After beginning to make the classification and clustering, first we did a test to find a close to optimal value for the number of neighbors hyperparameter. To accomplish this we simply did a for loop over a range of value, and using the cross validation technique taught to us in class, we found the number that best improve. We had to reduce the dimensional of the data using a PCA pretty aggressively as with a lot of features it would take several minutes. A smaller subset of the data was also selected.

We saw that the accuracy keep improving very slightly as the number of neighbors increased. At more than 40 it seem to reach an equilibrium, so we keep 40 as our optimal value. We will use this KNN hyperparameter from now on.

However in doing some changes, we saw that the accuracy in some occasions was poorer. Upon doing so research we understood that the number of neighbors depended mainly on the amount of data, as it exists a rule that says that this hyperparameter should be the square root of the amount of data points used on the model.

If some optimization needed to be made it should have



(a) Subgenres present in the data set that the amount of times present. As we can see it a pretty even distribution. The same happens with the 4 types of genre present. This lead us to believe that the dataset was purposely build to arrive at this distributions.

(b) The top 50 most well represented artists in the dataset. From a glance we can see how this distribution of artist is much more well balanced. As well as having greater genre and historical variety, with very different types of artists from different years being represented.

Fig. 5: These two graphics indicate that this dataset is of greater quality for the task that we set up to do. Moreover, looking at graphs for the distribution of artists in the same way as in figure 1, and the equal distribution of the 4 types of genres, we can conclude that this dataset is of more quality that the first one for our objectives.

been in a case per case basis. Looking at this values in the code, it can be seen how it was handpicked and changes from model to model.

The second dataset worked better in regards of K-neighbours classification for countries and top artists. We choose random 6 artists with at least 50 unique songs and all countries (five of them). After all the computation is done the average accuracy for K-neighbours clustering is around 60-80%, as well as presenting good recall and precision values. However this metric for the K-Means is still bad, sometimes even worse, because the features are the same and do not help us that much.

For the roughly the same amount of artists to predict as genres, the results seemed to be better for the artists, with an accuracy oscillating around 80%. This is a result that surprised us, as the artists intuitively seem the more complicated value to predict. This can cause of the more variety in the artist, if you randomly pick artists form a well balanced dataset, perhaps they are likely to be more different in regard to the differences between the overall genres of the songs in the dataset. This fact would need to be verified with a more careful analysis.

This leads to believe that the K-Means is overall not suited from this data. We think that the correlation between the song features are not strong enough for this type of algorithm to be accurate. To the best of our knowledge, we think that unsupervised clustering does not work with this type of data, as we could get good results with KNN and Random Forests.

For the popularity classification, because it is a continuous variables, we defined bins in order to get discrete values. Five different clusters were made (0-20, 20-40, 40-60, 60-80, 80-100). The overall accuracy of K-neighbours models was once again great, and just as expected the K-means accuracy was just as bad as all the previous examples. In the end all K-neighbours classification models for the second dataset had good accuracy and can be depicted as success in the task we gave ourselves in the start of the project.

In terms of a comparison between subgenre and genre classification, it seems that in the case of the genre it model perform better. Since the amount of genre is much smaller than the subgenres seem to be an intuitive results. When using the same parameters and the optimal way of sampling the data, the results of the genre classification were always better by roughly 20%-30%.

4. FINAL REMARKS

Before the conclusion we want to make final remarks on some aspects that were not covered properly in the report:

- Both datasets did not have missing data.
- Do to not understanding the song's features and the natures of the datasets. We did not search for outliers. In the first dataset Taylor Swift could have been considered one, however, since she represented a large amount of our data we opted to not considered as such.
- When needed we always performed a PCA, due to performance reasons and visualization of the clusters or predictions, despite them not being really necessary for trustful to evaluate the performance of a model.

5. CONCLUSIONS

From this project we understood that some data validation and checking when starting to work in a dataset can save time and resources. We did not do simple plots or statistical analysis at the beginning with the first dataset, those mistakes resulted in time wasted. Since we could not make the models worked, we did researching on alternatives, when in reality we should have searched for another dataset or adjusted our expectations.

As thing to improve, we should have organized better the code. Because we did not found a reliable and easy way to share a Jupyter Notebook -Git, Colab, VSCode...- we opted to share the snipped of code -the cells- directly. This resulted in the code being disorganized as the project developed. Moreover, because we shared individual cells, the code often had the same variables. Thus we ended with repeated variables and we often just had the cells to be self contained. Because of the redefine of variables we often reloaded the values from zero, for example, loading the dataset from the downloaded CSV.

Overall it was a project that we enjoyed, specially in the end when we had our ideas clear and we did some research in order to use the models and algorithms properly. At the end we had some models that could serve the task of genre, subgenre and artist predictions, despite this types of models not being the ideal ones for this particular set of data, as they do not work as better as regression or machine learning models for large quantities of data. Also we feel that a better job could have been done with more time or better time management, as there are still some hypothesis to be verified and a more careful analysis of our conclusions could had been made.