Instituto Tecnológico de Autónomo de México

MSc Data Science

Analytical Methods Final Project

# Music Analysis – Spotify – Top 500 Rolling Stones

Work directed by

Dr. Luis Felipe González

Members :

Miguel Calvo Valente - 203129

Rodrigo I. Juárez J. - 145804

Alejandro A. Muñoz G. - 203021

Spring 2022

# Contents

**Contents** **1**

**1 Introduction** **2**
1.1 Objective . . . . . 2
1.2 Methodology . . . . . 2

**2 Data** **2**
2.1 EDA . . . . . 4

**3 Join by hashes** **6**
3.1 Second EDA after join . . . . . 12

**4 Principal Component Analysis (PCA)** **14**

**5 t-distributed Stochastic Neighbor Embedding** **22**

**6 Conclusions** **23**

**Bibliography** **24**

# 1 Introduction

Music is a universal pleasure through cultures and time. We were curious about what makes a song a "great song". To answer this question we worked with two datasets. The first one is a Spotify database that we found in Kaggle which contains more than 160,000 songs with all kinds of metrics about the song (name, author, bpm, acousticness, and more), the second one contains the 500 greatest songs of all time according to the Rolling Stones magazine. This second dataset just contains the name of the song, the artist and the year of release.

## 1.1 Objective

Our goal was to join these two datasets into one and to incorporate the information of the Spotify variables into the top 500 songs. With this new information, we made an analysis to compare if there were any differences between an average song in Spotify in comparison to the top 500.

## 1.2 Methodology

To make this analysis, we joined the tables by using hashes and similarity metrics. We did this because there was no trivial way of joining them, due to small differences between how the name of the songs' titles and artists were written in the datasets.

After joining the tables into one and creating a label to identify the song, we explored 2 dimensionality reduction techniques, PCA and TSNE, to try to find structure and differences between the two groups.

# 2 Data

Both datasets were found in Kaggle, you can find the dataset here for Spotify and 500 Greatest Songs of All Time Rolling Stones.

As mentioned before, the Spotify dataset contains more than 160.000 songs collected through the use of Spotify Web API. The dataset contains songs from 1921 to 2020. Each year has at most the top 2000 most popular songs at the date of retrieval.

- ID (Id of track generated by Spotify).
- Acousticness (Ranges from 0 to 1): the value that describes how acoustic a song is. Higher values mean that the song is most likely to be an acoustic one.
- Danceability (Ranges from 0 to 1): the relative measurement of the track being danceable.Higher values mean that the song is more danceable.

- Duration (Integer typically ranging from 200k to 300k): the length of the track. In milliseconds.
- Energy (Ranges from 0 to 1): the energy value of the track. Higher values mean that the song is more energetic.
- Instrumentalness (Ranges from 0 to 1): the relative ratio of the track being instrumental. Higher values mean that the song contains more instrumental sounds.
- Valence (Ranges from 0 to 1): the positiveness of the track. Higher values mean, the track evokes positive emotions (like joy) otherwise means, it evokes negative emotions (like anger, fear).
- Popularity (Ranges from 0 to 100): the popularity of the song.
- Tempo (Float typically ranging from 50 to 150): the tempo of the track in Beat Per Minute (BPM).
- Liveness (Ranges from 0 to 1): detects the presence of an audience in the recording. Higher values represent an increased probability that the track was performed live. A value above 0.8 provides a strong likelihood that the track is live.
- Loudness (Float typically ranging from -60 to 0): the overall loudness of a track in decibels (dB). Loudness values are averaged across the entire track and are useful for comparing the relative loudness of tracks. Loudness is the quality of a sound that is the primary psychological correlate of physical strength (amplitude). Values typical range between -60 and 0 dB.
- Speechiness (Ranges from 0 to 1): the relative length of the track containing any kind of human voice.
- Year (Ranges from 1921 to 2020): the release year of the track.
- Mode (0 = Minor, 1 = Major).
- Explicit (0 = No explicit content, 1 = Explicit content).
- key (All keys on octave encoded as values ranging from 0 to 11, starting on C as 0, C# as 1 and so on...).
- Artists (List of artists mentioned).
- Releasedate (Date of release mostly in yyyy-mm-dd format, however precision of date may vary).
- Name (Name of the song).

The data set of 500 Greatest Songs of All Time Rolling Stones just contains three variables:

- Song (Name of the song)
- Artist (Name of the artist)
- Year (Ranges from 1907 to 2004)

## 2.1 EDA

There were 2 Exploratory Data Analysis (EDA) performed: One to determine how to join both of the datasets and to explore the variables' distributions and correlations. The second one was performed to visualize if there was a direct way to characterize the top 500 Rolling Stones songs from the rest. This second EDA was done after joining the datasets, so we will cover it later on.

There is not much to explore in the data "500 Greatest Songs of All Time Rolling Stones". It is just a list of the songs and the authors. So, we will focus on the Spotify dataset.

First, we observed at the distributions of songs over the years 9. Here we noticed that, at most, each year contains 2000 songs, all of which represent the most popular songs for the given year at the date in which they were retrieved.

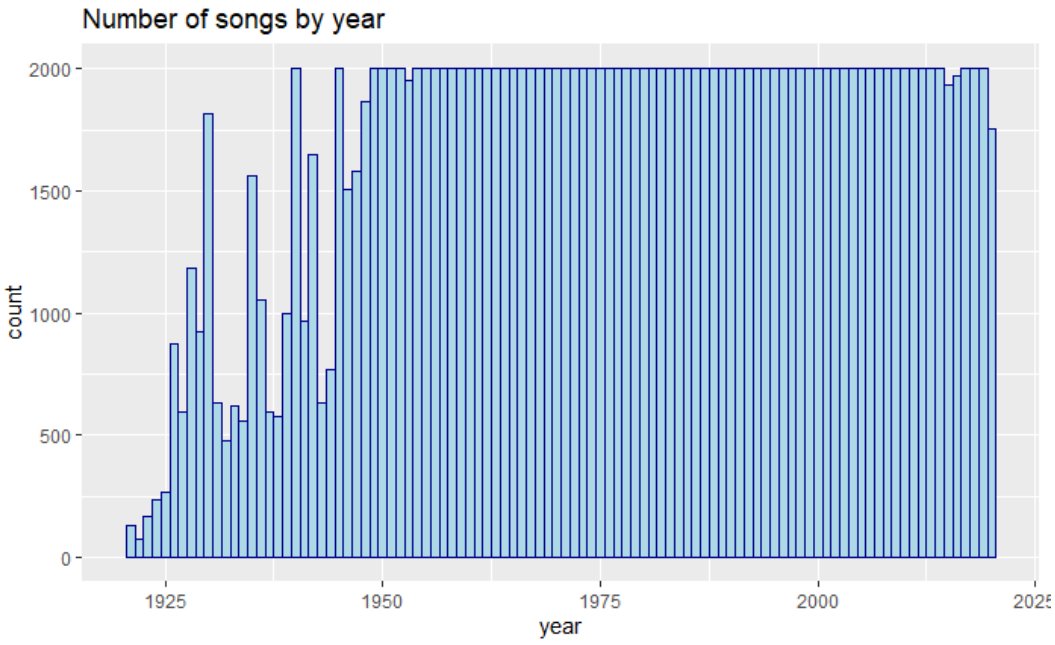


Figure 1: Histogram of songs by year

We take a quick look into the variables to try to find correlations between them. We can see some strong positive correlation between popularity and year, and loudness and energy. We can see that energy, loudness and popularity are negatively correlated with acousticness.

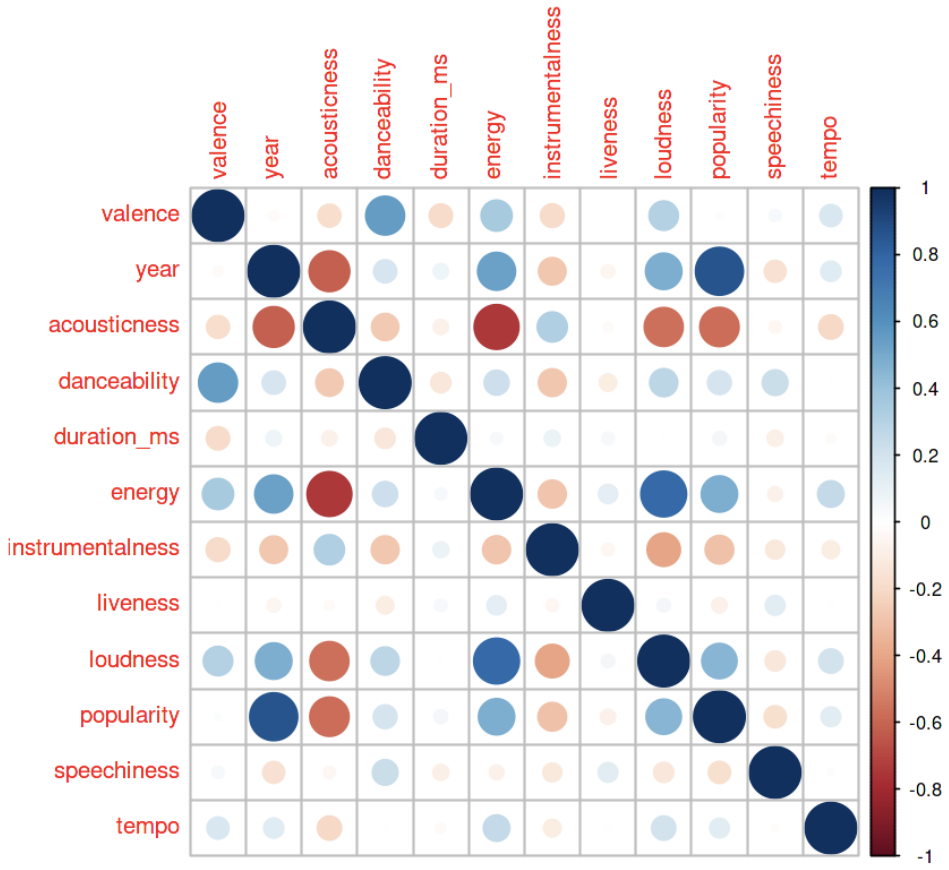


Figure 2: Correlation plot Spotify data

Both datasets contain a column for the artist(s) and the name of the song. It would seem to be fairly easy to simply create a joint column of artist-name and join both the datasets. We initially tried this approach but quickly observed that there were many wrong matches. For this reason, we decided to use an approach with hashes.

# 3 Join by hashes

MinHash is a widley spread used method to quickly find similar texts. It can reduce the overall amount of comparisons done between a collection of texts by only comparing the "significant" associations.

A brief summary on the MinHash procedure is as follows:

1. For the given set of texts, we retrieve their k-shingles. These are k-long subsequences of characters. For example, for k = 4, the k-shingles of the text "This is an example" would be: [This,his ,is i,s is, is ,is a,s an, an ,an e,n ex, exa,exam,xamp,ampl,mple]

2. We associate a numeric hash function to each of these substrings. The mapping of this hash function must be as random as possible, so that even similar strings like "Maria" and "Marie" are mapped to non correlated values in the hash function image.

3. We define the minhash of each text as the minimum value that each transformed k-shingle takes after being transformed by the hashing function.

4. By definition and construction, the minhash of similar sentences must have a high probability of being the same. For example, for the texts "My dog has been eating this whole time" and "My dog has been sleeping this whole time", there are several shared k-shingles, all of which have a uniform probability of being the minimum hash, thus, the likelihood of them sharing a minhash is quite large.

5. Having retrieved the minhashes, we can now only make the comparisons between the texts with shared minhashes instead of comparing each text with every other one, reducing considerably the number of comparisons.

We decided to use a MinHash approach for two reasons: 1. Comparing the Jaccard similarity between the k-shingles of every of song with the other ones would be too computationally costly, unviable even. 2. There are titles or artists that do not necessarily match between both datasets. For example, in the Spotify dataset we could have "The Jimi Hendrix Experience", but in the Rolling Stones one we could have "Jimi Hendrix". Thus, using minhashes both reduces the amount of operations and time required to match songs, and also increases the probability of finding not identically matching strings.

The value of k and the amount of hash functions are in some sort a hyperparameter; they depend on the problem at hand and have to be tuned in order to provide the best solutions. If k is too small, we run into too many false positives, that is, too many minhases that matched by pure luck. If we increase it too much, we risk losing certain matches (f.e. for k=10, then "Yesterday" and "Yesterday - Original Version" would share no matches even though they refer to the same song).

Also, if we use too few hashing functions, then we risk losing a possible comparison because, by sheer luck, two very similar songs could have a different minhash. And if we increase it too much, we end up doing too many computations.

By trial and error, we tried different values for k, in the range of [2, 6], and tried a number of hashing functions from [2, 4]. With k=4 and 4 different hashing functions, we managed to retrieve 440 of the 500 Rolling Stones songs. The remaining 60 could either not be detected by bad luck in all of the hashing functions, which is relatively improbable, or could simply not be included in the Spotify dataset. We consider 440 songs is a good amount to perform our analysis on given our purposes.

Regarding the details on data preprocessing:

- The Rolling Stones dataset has 500 songs. The Spotify one has 169,909. That means that the maximum amount of comparison we could have performed was $\frac{(500)(501)}{2} \times 169,909 = 21,281,102,250$. Through the MinHash procedure, we ended up with only 1,243,402 comparisons.
- There were 5 songs in the Rolling Stones dataset dated before 1948, through the years 1906 to 1909. Also, the last song from Rolling Stones is from 2008, whilst the Spotify dataset goes all the way to 2020. We decided to remove all of the songs before 1948 and after 2008 from both datasets for the following reasons:
  - We can reduce the number of overall computations and comparisons, which can help us test more values for k in less time.
  - We consider that the analysis we will perform later makes more sense for songs in the same range of time, since songs that are too old or too recent can differ considerably from the selected range (which on its own is quite large, but we will later on make an analysis based on ranges of years).
- There were several songs in the Spotify dataset written in languages like Japanese, Arabic, etc. and thus contained non-usual characters and letters, whilst the Rolling Stones set contained only letters from the English alphabet. Thus, any comparison between songs with non-usual and the Rolling Stone ones would be unnecessary, so we removed them.
- All of the songs in the Rolling Stones set have fairly regular names. By that, we mean that they do not contain word such as 'Sonata', 'Danse', 'Chapter', 'Nocturne', 'Prelude', etc. We made a list of words that we consider would not appear in the titles of song in that set. We validated that indeed there were no instances of them, and later proceeded to remove all songs in the Spotify dasatet which contained instances of these words.

- All 3 previous steps eventually reduced the Spotify dataset from 169,909 songs to 115,726.
- To compare songs, we decided to make a variable which combines the name and artist, taking apart any non-alphabet character, introducing some spaces between them to avoid overlapping k-shingles, and lower-casing everything to avoid losing potential matches. For example, the register for "What's Going On", "Marvin Gaye" would yield the following value: "whats going on marvin gaye".
- Having done this preprocessing, we were now able to calculate sensible k-shingles, convert them through the hash functions, retrieve the minhashes for every song and for every hashing function, and join the matching minhashes from both datasets:

| id_rolling_stones <int> | cubeta <chr> | id_spotify <int> |
|---|---|---|
| 1 | 1--2102236732 | 27 |
| 1 | 1--2102236732 | 38 |
| 1 | 1--2102236732 | 62 |
| 1 | 1--2102236732 | 154 |
| 1 | 1--2102236732 | 166 |
| 1 | 1--2102236732 | 396 |
| 1 | 1--2102236732 | 488 |
| 1 | 1--2102236732 | 520 |
| 1 | 1--2102236732 | 521 |
| 1 | 1--2102236732 | 590 |

Figure 3: Matching minhashes between datasets. The column **cubeta** is the number of the hashing function, followed by –, an finally the minhash calculated by that hasing function for that song in both datasets. So, the Rolling Stones song with id 1 matched through the minhash 2102236732 with the songs in Spotify with ids 27, 38, 62, ...

- Lastly, we calculate the Jaccard similarity between the k-shingles of the resulting matches. These gives us a score between 0 and 1 to measure the similarity between any two given songs. For every score above 0.3, we kept the maximum scoring match for every song in the Rolling Stones dataset.

We managed to retrieve 440 songs from the Rolling Stones set contained in the Spotify one. Some results that summarise this are as follows:

- Highly scoring matches, above 0.7, do refer to the songs with the same name and artist, as expected.

| Score | Rolling Stones - Title | Spotify - Title | Rolling Stones - Artist | Spotify - Artist |
|---|---|---|---|---|
| 0.937 | I Never Loved a Man (the Way I Love You) | I Never Loved a Man (The Way I Love You) | Aretha Franklin | ['Aretha Franklin'] |
| 0.930 | Wholl Stop the Rain | Who'll Stop The Rain | Creedence Clearwater Revival | ['Creedence Clearwater Revival'] |
| 0.927 | With a Little Help From My Friends | With A Little Help From My Friends | The Beatles | ['The Beatles'] |
| 0.926 | You Are the Sunshine of My Life | You Are The Sunshine Of My Life | Stevie Wonder | ['Stevie Wonder'] |
| 0.926 | Killing Me Softly With His Song | Killing Me Softly with His Song | Roberta Flack | ['Roberta Flack'] |
| 0.925 | The Night They Drove Old Dixie Down | The Night They Drove Old Dixie Down | The Band | ['The Band'] |
| 0.925 | Bad Moon Rising | Bad Moon Rising | Creedence Clearwater Revival | ['Creedence Clearwater Revival'] |
| 0.925 | I Heard It Through the Grapevine | I Heard It Through The Grapevine | Marvin Gaye | ['Marvin Gaye'] |
| 0.923 | Whole Lotta Shakin Going On | Whole Lotta Shakin' Going On | Jerry Lee Lewis | ['Jerry Lee Lewis'] |
| 0.922 | Do You Believe in Magic | Do You Believe in Magic? | The Lovin' Spoonful | ["The Lovin' Spoonful"] |

Figure 4: Matches with the highest score

| Score | Rolling Stones - Title | Spotify - Title | Rolling Stones - Artist | Spotify - Artist |
|---|---|---|---|---|
| 0.833 | Papas Got A Brand New Bag | Papa's Got A Brand New Bag | James Brown | ['James Brown'] |
| 0.833 | Shes Not There | She's Not There | The Zombies | ['The Zombies'] |
| 0.833 | Beautiful Day | Beautiful Day | U2 | ['U2'] |
| 0.830 | Goodbye Yellow Brick Road | Goodbye Yellow Brick Road | Elton John | ['Elton John'] |
| 0.829 | Spanish Harlem | Spanish Harlem | Ben E. King | ['Ben E. King'] |
| 0.826 | Hey Ya! | Hey Ya! | OutKast | ['OutKast'] |
| 0.824 | The Twist | The Twist | Chubby Checker | ['Chubby Checker'] |
| 0.822 | Leader of the Pack | Leader Of The Pack | The Shangri-Las | ['The Shangri-Las'] |
| 0.818 | Hound Dog | Hound Dog | Elvis Presley | ['Elvis Presley'] |
| 0.818 | Walk This Way | Walk This Way | Aerosmith | ['Aerosmith'] |
| 0.818 | One More Time | One More Time | Daft Punk | ['Daft Punk'] |

Figure 5: Matches with a score around 0.8

| Score | Rolling Stones - Title | Spotify - Title | Rolling Stones - Artist | Spotify - Artist |
|---|---|---|---|---|
| 0.758 | Paranoid | Paranoid | Black Sabbath | ['Black Sabbath'] |
| 0.755 | Do Right Woman â€" Do Right Man | Do Right Woman, Do Right Man | Aretha Franklin | ['Aretha Franklin'] |
| 0.754 | The Tracks of My Tears | The Tracks Of My Tears | Smokey Robinson and the Miracles | ['Smokey Robinson & The Miracles'] |
| 0.750 | Love Me Tender | Love Me Tender | Elvis Presley | ['Elvis Presley'] |
| 0.744 | Sunshine of Your Love | Sunshine Of Your Love | Cream | ['Cream'] |
| 0.740 | (Were Gonna) Rock Around the Clock | (We're Gonna) Rock Around The Clock | Bill Haley and His Comets | ['Bill Haley & His Comets'] |
| 0.738 | Midnight Train to Georgia | Midnight Train to Georgia | Gladys Knight and the Pips | ['Gladys Knight & The Pips'] |
| 0.738 | Candle in the Wind | Candle In The Wind 1997 | Elton John | ['Elton John'] |
| 0.737 | I Got a Woman | I've Got a Woman | Ray Charles | ['Ray Charles'] |

Figure 6: Matches with a score around 0.7

- Medium scoring matches show some interesting results. On this range, songs do not match perfectly anymore. Rather, we find several instances of "Remastered" songs, "XXXX Version", "Part 1", etc. which mostly do refer to the same song, or at least a version of the song, that we can use for the purpose of our analysis. It's also interesting to observe how the MinHash method catches non-exact artists, such as "Bob Marley" and "Bob Marley & The Wailers". Nevertheless, low scores also bring miss-matches either in the title or the artist. We treated this false positives by hand, searching each of them individually in the Spotify dataset. There were only 30 of these, which took some time to retrieve but at least was far less time consuming than searching the 500 songs from the beginning.

| Score | Rolling Stones - Title | Spotify - Title | Rolling Stones - Artist | Spotify - Artist |
|---|---|---|---|---|
| 0.636 | Gloria | Gloria | Them | ['Them'] |
| 0.634 | Good Lovin | Groovin' | The Young Rascals | ['The Young Rascals'] |
| 0.633 | My Sweet Lord | My Sweet Lord (2000) - 2014 Mix | George Harrison | ['George Harrison'] |
| 0.632 | 96 Tears | 96 Tears | ? and the Mysterians | ['? & The Mysterians'] |
| 0.632 | The Sounds of Silence | The Sounds of Silence | Simon and Garfunkel | ['Simon & Garfunkel'] |
| 0.632 | Why Do Fools Fall In Love | Why Do Fools Fall in Love | Frankie Lymon and The Teenagers | ['Frankie Lymon', 'die Teenagers'] |
| 0.630 | Oh | Leah | Roy Orbison | ['Roy Orbison'] |
| 0.627 | We Gotta Get Out of This Place | We Gotta Get Out Of This Place (US Versior | The Animals | ['The Animals'] |
| 0.625 | What'd I Say | What'd I Say, Pt. 1 & 2 | Ray Charles | ['Ray Charles'] |
| 0.625 | The Loco-Motion | The Locomotion | Little Eva | ['Little Eva'] |
| 0.623 | I Want to Know What Love Is | I Want to Know What Love Is - 1999 Remas | Foreigner | ['Foreigner'] |

Figure 7: Matches with a score around 0.6. Non highlighted cells indicate matches that are not very interesting to analyze. Yellow highlighted cells are interesting matches. For example, "My Sweet Lord" does match with "My Sweet Lord (2000) - 2014 Mix", since they are basically the same song. Or "We Gotta Get Out of This Place" matching with "We Gotta Get Out of This Place [US Version]" is also something we were expecting for the method to catch. Orange highlighted cells are false positives, which we later addressed manually.

| Score | Rolling Stones - Title | Spotify - Title | Rolling Stones - Artist | Spotify - Artist |
|---|---|---|---|---|
| 0.556 | Whole Lotta Love | Whole Lotta Love - 2012 Remaster | Led Zeppelin | ['Led Zeppelin'] |
| 0.556 | Aint It a Shame | Ain't That A Shame | Fats Domino | ['Fats Domino'] |
| 0.549 | No Woman, No Cry | No Woman No Cry | Bob Marley | ['Bob Marley & The Wailers'] |
| 0.549 | Personal Jesus | Personal Jesus - Acoustic | Depeche Mode | ['Depeche Mode'] |
| 0.549 | Nuthin But a G Thang | Nuthin' But A "G" Thang | Dr. Dre | ['Dr. Dre', 'Snoop Dogg'] |
| 0.538 | Young Americans | Young Americans - 2016 Remaster | David Bowie | ['David Bowie'] |
| 0.538 | I Want to Hold Your Hand | I Want To Hold Your Hand - Remastered 20( | The Beatles | ['The Beatles'] |
| 0.538 | Sweet Dreams (Are Made of This) | Sweet Dreams (Are Made of This) | Eurythmics | ['Eurythmics', 'Annie Lennox', 'Dave Stewart |
| 0.537 | Something | Something - 2019 Mix | The Beatles | ['The Beatles'] |

Figure 8: Matches with a score around 0.5. Refer to the previous figure for an explanation on the color coding. At this range we can find some more interesting matches, such as "Personal Jesus" with "Personal Jesus - Acoustic", which might not be the exact same song, but we consider it to be a good match for our purposes. Also, "Eurythmics" matched with "['Eurythmics', 'Annie Lennox', 'Dave Stewart']", the latter being the correct complete set of artists.

- Even low scores proved to find several correct matches, but there were also

various false positives in this range as expected.

| Score | Rolling Stones - Title | Spotify - Title | Rolling Stones - Artist | Spotify - Artist |
|---|---|---|---|---|
| 0.424 | Cortez the Killer | Cortez the Killer - 2016 Remaster | Neil Young | ['Neil Young', 'Crazy Horse'] |
| 0.422 | Hit the Road Jack | Blackjack | Ray Charles | ['Ray Charles'] |
| 0.420 | Everyday | Everyday | Buddy Holly and the Crickets | ['Buddy Holly'] |
| 0.414 | I Cant Stop Loving You | I Can't Stop Loving You | Ray Charles | ['Roy Orbison'] |
| 0.411 | Earth Angel | Earth Angel (Will You Be Mine) | The Penguins | ['The Penguins'] |
| 0.405 | Push It | Push It | Salt 'n Pepa | ['Salt-N-Pepa'] |
| 0.404 | Walking in the Rain | Just Walking In the Rain | The Ronettes | ['The Champs'] |
| 0.400 | Purple Haze | Purple Haze | The Jimi Hendrix Experience | ['Jimi Hendrix'] |
| 0.400 | Good Times | Good Times - 2018 Remaster | Chic | ['CHIC'] |
| 0.395 | Crazy | Faded Love | Patsy Cline | ['Patsy Cline'] |

Figure 9: Matches with a score around 0.4. Refer to the previous figure for an explanation on the color coding. This range still shows various correct matches, such as "Buddy Holly and the Crickets" and "Buddy Holly", or "Earth Angel" and "Earth Angel (Will You Be Mine)", but there are also some false positives like "Hit the Road Jack" and "Blackjack", or "Crazy" and "Faded Love".

After correcting for the false positives, we proceeded to get the variables from the Spotify dataset for each of the 440 songs. Also, we sampled 2500 different songs from the Spotify set (the one we reduced up to 115,726 registers). We added a column to indicate whether the song was from the top 500 Rolling Stones set (TRUE) or not (FALSE). With this new filtered set, we can proceed to perform a second EDA to find if there are clear differences between the top 500 songs and the rest, as well as the proposed dimensionality reduction techniques previously mentioned.

## 3.1 Second EDA after join

After joining the two tables with hashes, we added all the information from the variables of the Spotify set into the top 500. We did an analysis of the two groups' distribution (belonging to top 500 and not top 500) to see if we could find any differences. 10 and 11.

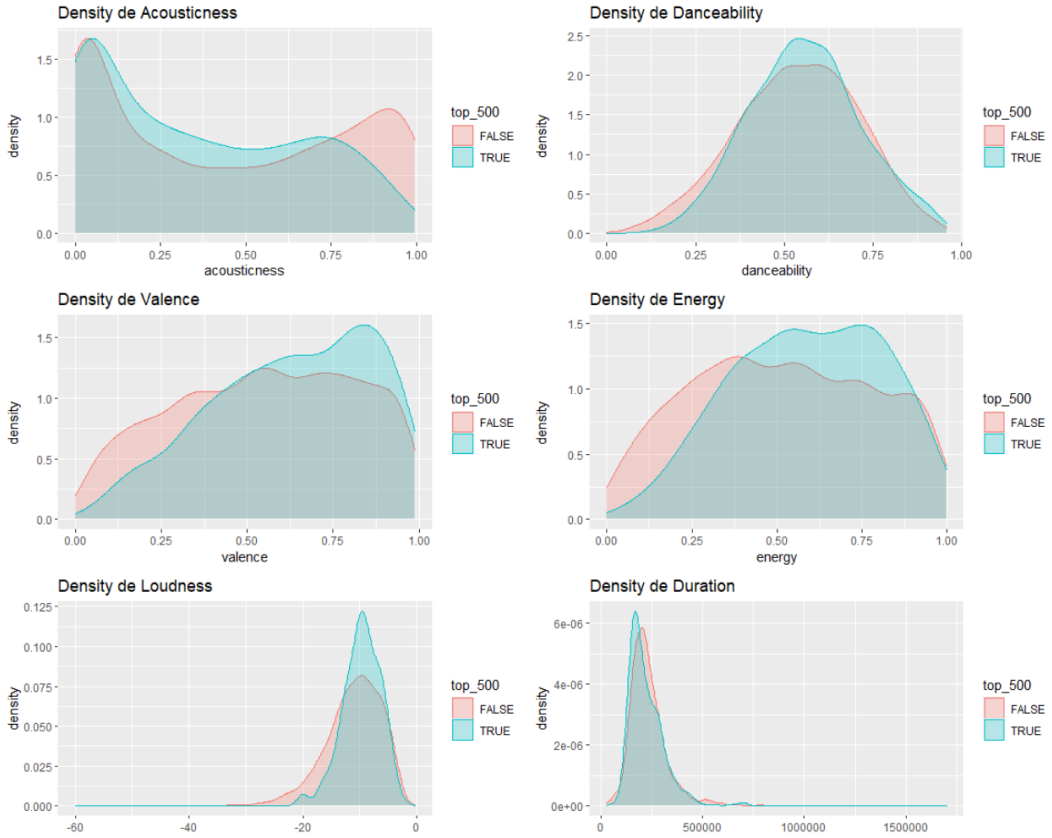


Figure 10: Density plots of the joint datasets

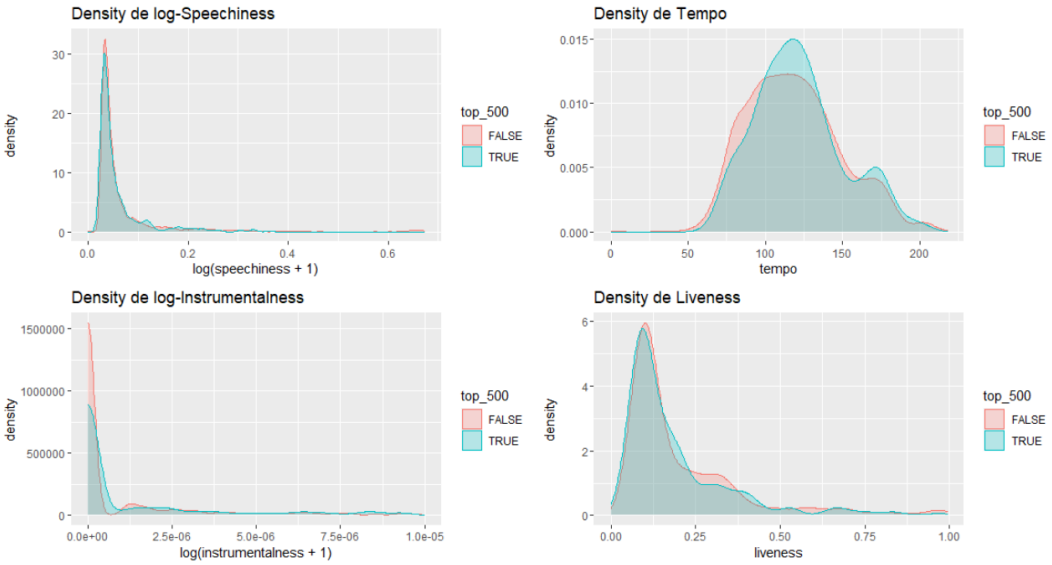


Figure 11: Density plots of the joint datasets

From this density plots we cannot see any differences in the distributions on all

the variables. This means that we cannot differentiate a song belonging to the top 500 to other songs by just looking at these variables independently.

Based on this initial analysis, we decided to apply techniques such as PCA and TSNE in order to find differences or segmentations of the groups. These techniques consider the interaction of variables so, we hope to see different results.

# 4 Principal Component Analysis (PCA)

Principal components is the singular value decomposition applied to a column-centered data matrix. This operation converts the low rank matrix approximation problem into one of approximations that seek to explain most of the variance (including covariance) of the variables in the data matrix.

We centered the variables since the column means do not have important or interesting information for our purposes and because we are more interested in having an interpretation in terms of variances and covariances than making an approximation of the original data.

Initially, to make our data sets comparable with respect to time, we only took the songs that intersected with respect to the span of years of the Rolling Stones top 500 and the Spotify database, keeping songs from 1948 to 2008.

After doing the principal components analysis of our numerical variables (acousticness, danceability, speechiness, tempo, valence, energy, instrumentalness, liveness, loudness and duration), we observe that the first dimension explains 28.6% the variance, the second the 14.8% and the third 11.9%. In total, the first three dimensions explain 55.3% of the variance. We can see graphically the percentage of explained variance. In addition, we can graphically observe how the variables are ordered by the amount of original variance they describe in the two first dimensions.

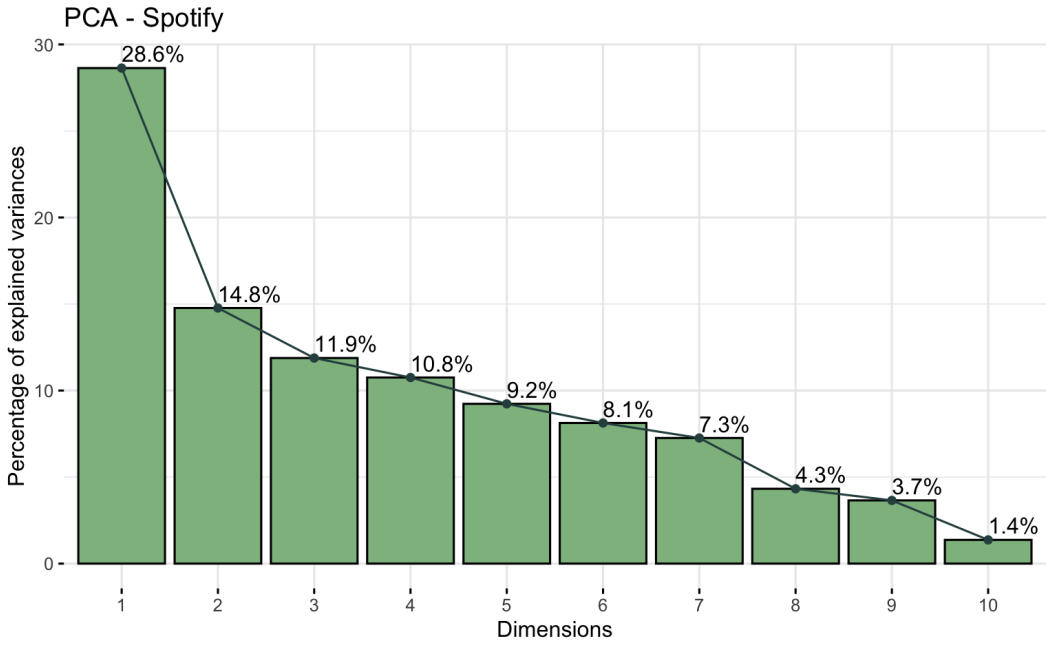


Figure 12: Variance explained by dimension

Graphically, we can observe the contribution of the variables for the two main dimensions. This is a very interesting analysis since we can see the direction and magnitude of these variables represented in vectors.

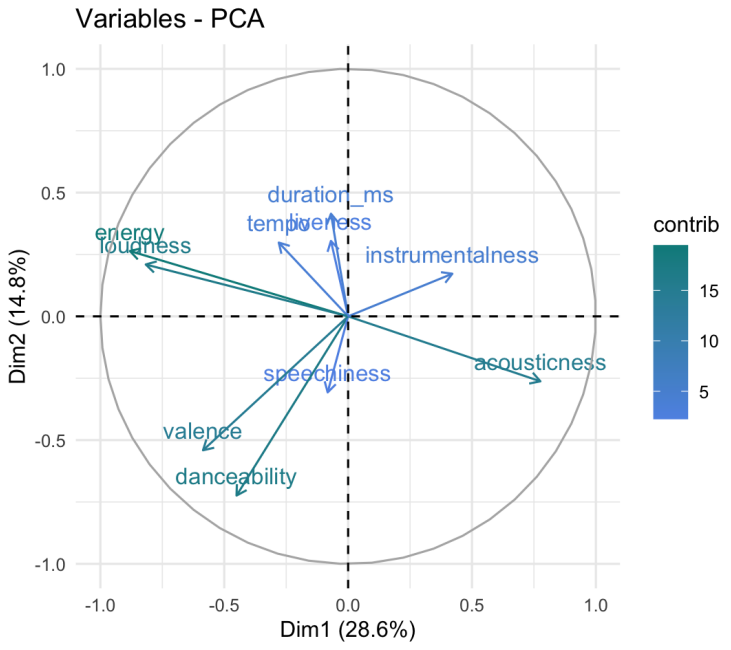


Figure 13: Contribution of variables to the main dimensions

We realized that the first component separates songs with acousticness and instrumentalness from those with energy and loudness (acousticness & instrumentalness vs. energy & loudness). We also note that the second principal component separates songs that have a certain duration and liveness from those that have danceability and valence (duration_ms & liveness vs. danceability & valence). We looked at our graph of the two main components of all our songs and realized that there is a certain pattern by years, so we filtered by decades (from 1948 to 2008). In total we worked with six decades and we did the analysis of principal components for each decade.


Figure 14: Principal components by year

We observed that the percentage of variance explained in the two principal components is very similar to the same analysis done with the complete database for all the years. However, the PCA is found in what proportion of variance is explained by the different variables, since it does change depending on the time in which this analysis is carried out.

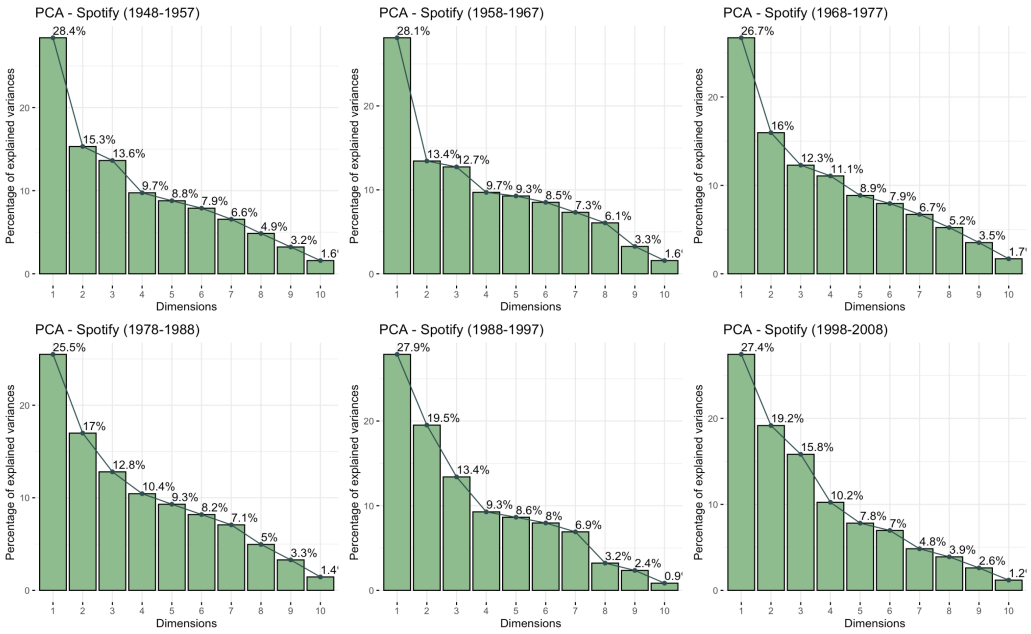


Figure 15: Variance explained by dimension for each decade

By doing the analysis by decades, we realized how the variables that explained the variance within each set changed. For example, for the first decade (1948-1957), for the first dimension the variables acousticness & instrumentalness vs. energy & loudness are the ones that explain the first dimension, however, for the decade from 78 to 87, energy & loudness vs. acousticness are the variables of the first dimension. Graphically we can see how the correlations of these variables evolved and their importance from decade to decade. We can see that the variable accousticness is the most important to make a segregation of the dataset for all the decades.

We're not observing a clear segregation of the top 500 list and the non-top 500 ones. Only for the first two decades, we can see that the top 500 songs are biased towards the left. For the first decade, they were more characterized by their energy loudness and less acousticness instrumentalness in comparison to the rest. In the second one, they were more characterized by their energy valence and less acousticness in comparison to the non-top 500. However, the songs of the top 500 list are closer to the axes, while those that do not belong to the top-500 are more dispersed in the following representations of the two main dimensions (specially in the last four decades).

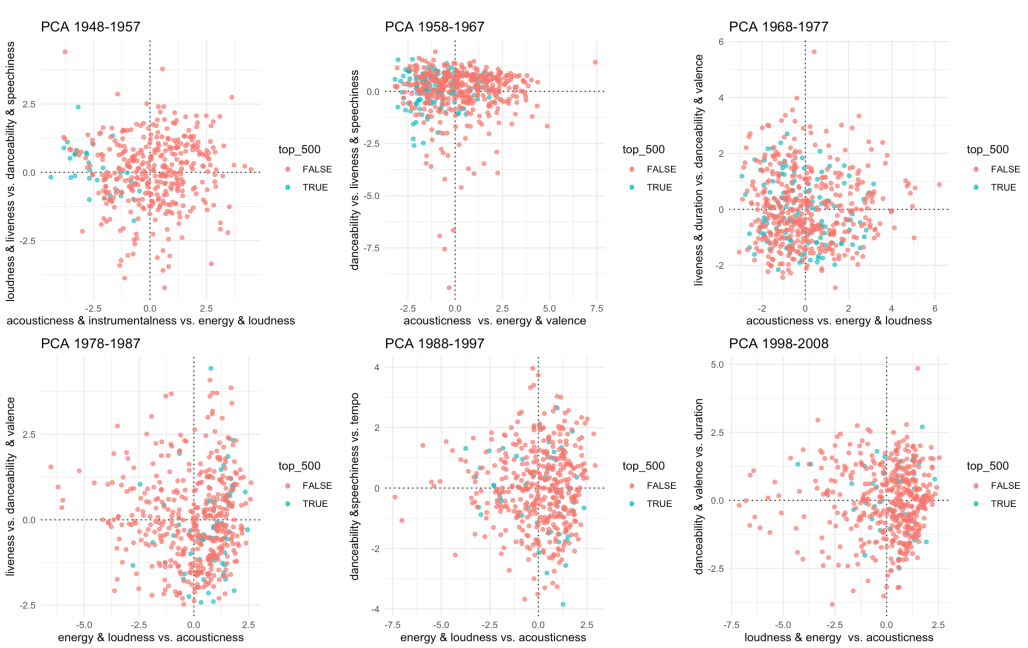


Figure 16: Two principal dimensions for each decade

Finally, we decided to do an analysis of the most recent songs vs the top 500. This decision was made in order to be able to do a complete analysis of the dataset.

We did the principal components analysis. We found that together, the two main components explained approximately 34.7% of our variance, which is really poor. We don't have variables that help us segregate our data set very well. However, we can infer which are the variables that would help us to do so. The results are very similar to our previous analysis of the dataset that had songs up to 2008: acousticness vs. energy & loudness in the first dimension and durationy vs. danceability & valence for the second dimension.

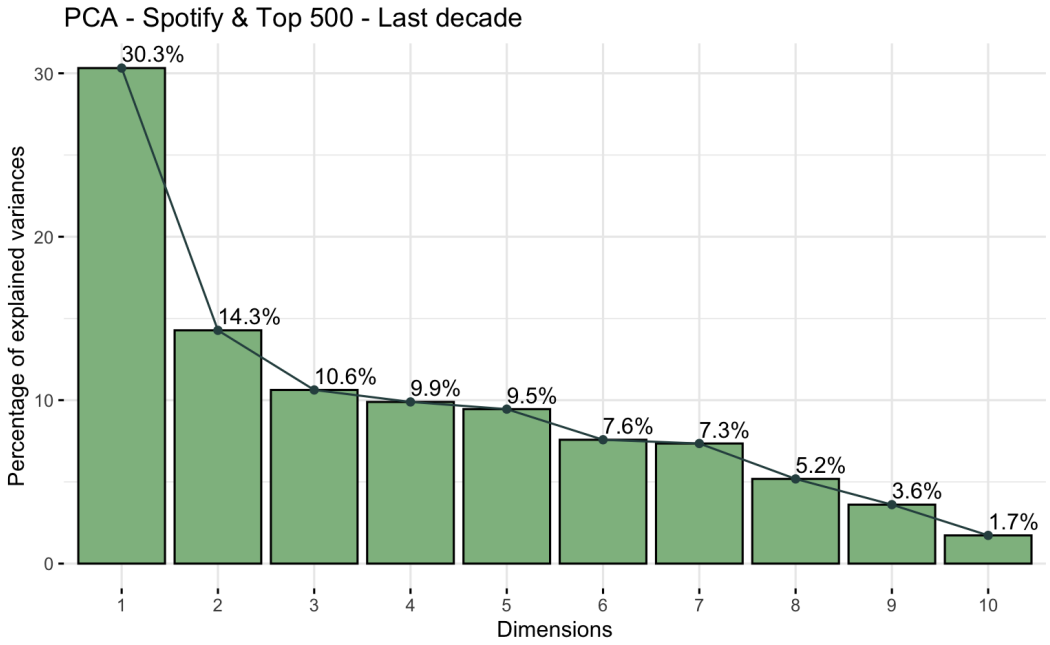


Figure 17: Variance explained by dimension for the entire dataset

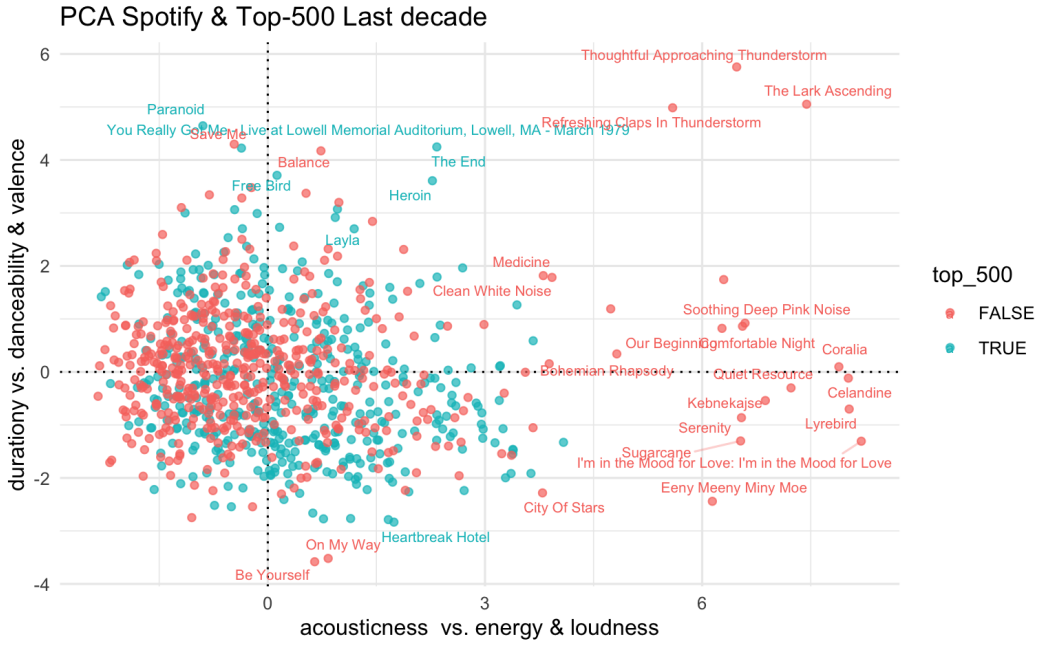


Figure 18: Two principal dimensions of the full dataset

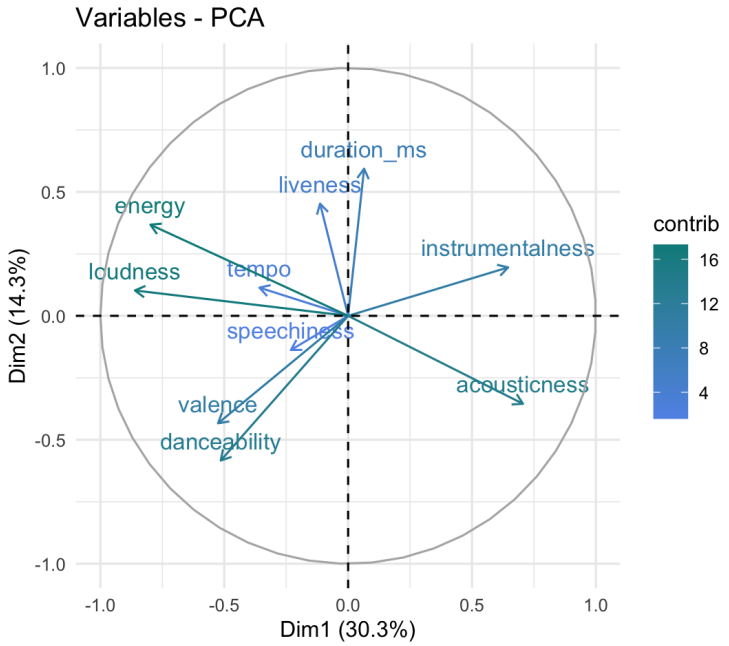


Figure 19: Variables that contribute to the two main dimensions

When making the comparison of the top-500 songs of spotify vs. the compilation of songs from spotify, regarding the analysis of main components, the top 500 songs explain 40% of their variance in their first two dimensions, while 46% for songs from Spotify. That is, for the Spotify data, the variance is better explained with the variables of acousticness vs. energy & loudness and duration vs. danceability & valence.

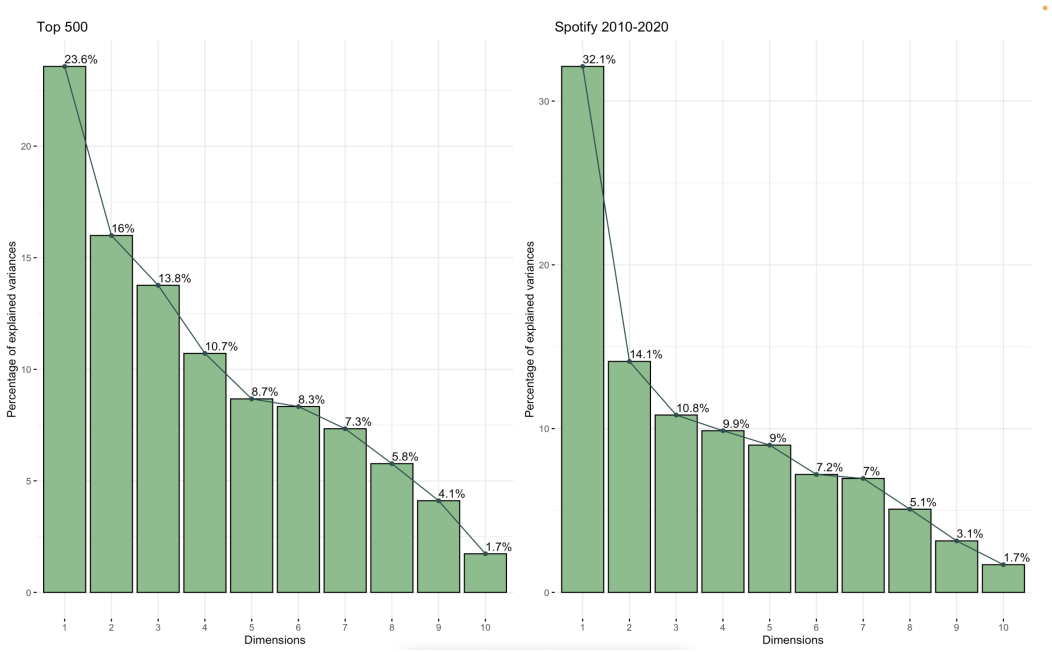


Figure 20: Variance explained for both datasets

In addition, we can see that the songs are grouped better in the top 500 dataset than in the Spotify one.

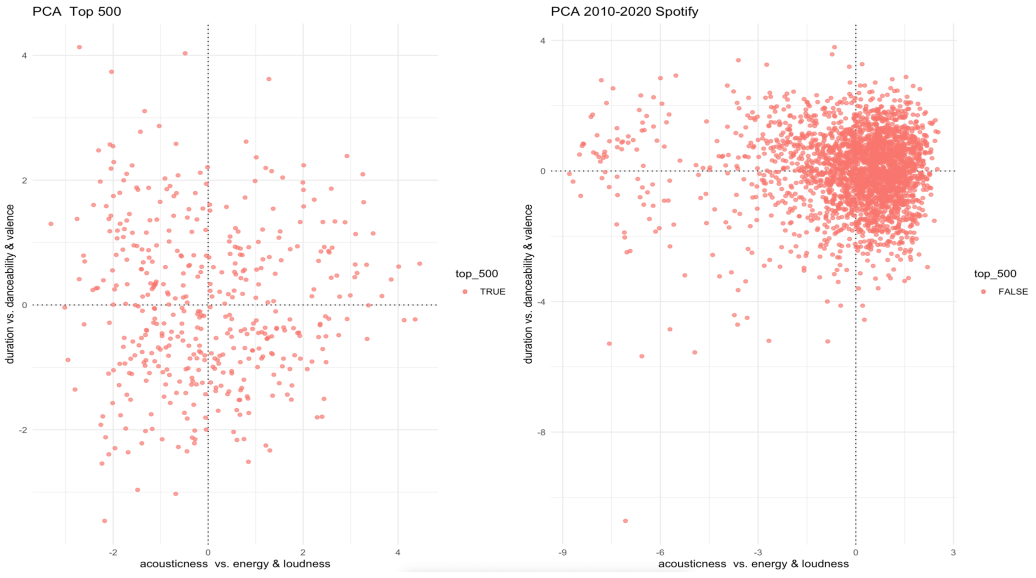


Figure 21: Principal two dimensions for both datasets

We can see that the explanatory variables of the variance in the two main dimensions are very similar, however, they change with respect to the size of the vector in the comparative graph. The only 2 variables that change quadrant are durationms

instrumentalness, meaning that songs from the top 500 last longer and had more relative instrumental presence with respect to the whole song.

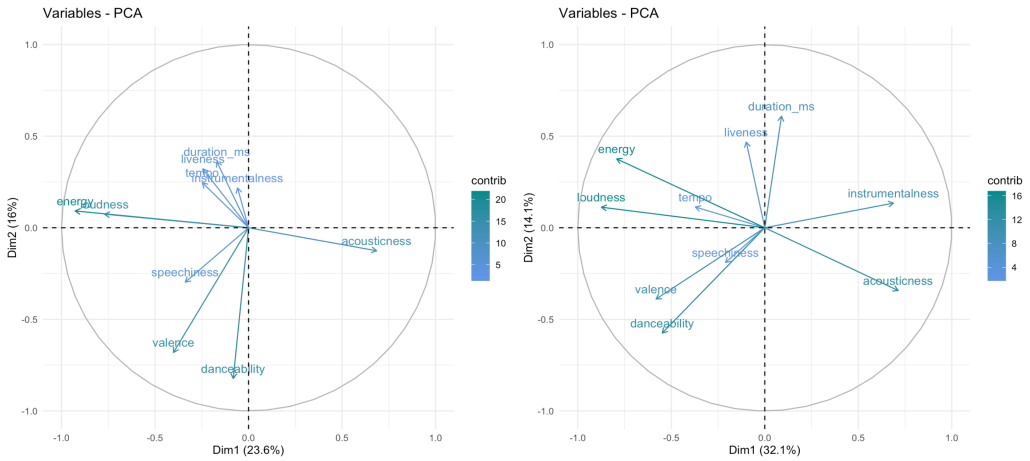


Figure 22: Variables that contribute to the two principal dimensions for both datasets

# 5 t-distributed Stochastic Neighbor Embedding

t-SNE is a method that reduces the dimensionality of the data, but in comparison to PCA, it focuses on keeping close and similar data together in the representation, at the cost of losing information about distances of very different cases.

We applied the t-SNE algorithm to our data and could not find any consistent aggrupation through different iterations and random seeds, as shown in figure 24. To verify if t-SNE effectively converged on well behaved data regardless of the seeds, we applied this same algorithm to the MNIST dataset. We did find consistency in the latter, showing that our data is simply not well behaved enough to find consistent clusters 23.

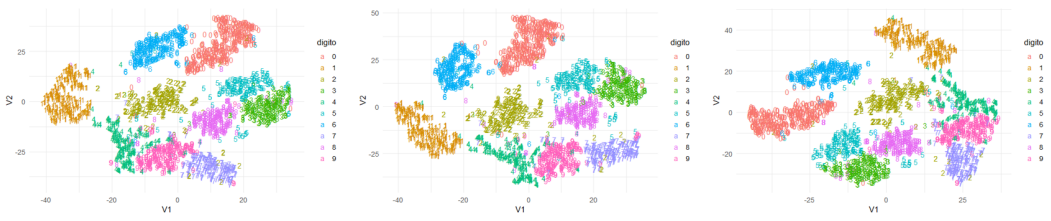


Figure 23: Projection of MNIST data ith t-SNE with three random seeds

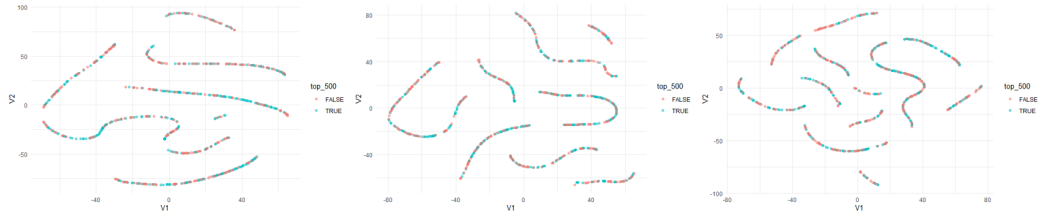


Figure 24: Projection of our data with three random seeds

# 6 Conclusions

Using the MinHash technique, we were able to successfully capture several coincidences, whether they matched perfectly or that they varied in a small manner, such as containing the word "Remastered" or having an artist being spelled differently. The procedure was not perfect, since we could not find 55 songs, and 30 of the 440 found ones were false positives. Regarding the not found, it is highly likely that they were simply not contained in the Spotify dataset. The false positives could perhaps be reduced if we used more hashing functions. We were quite satisfied with the results of the MinHash method, which enabled us to move on to the more analysis-oriented techniques such as PCA and T-SNE.

The main objective for using PCA was to do an exploratory analysis of the data and to later be able to do some predictive modeling to create a song that will be considered in the top 500 in the future, based on the characteristics that were found. PCA was used to be able to reduce the dimensionality of our data to project each data point only on the first principal components, i.e. to get lower dimensional data and preserve as much variation of the data as possible. However, our results lead us to conclude that either this method is not very good in two dimensions to capture separations, or that the variables at hand do not allow for such expected segregation.

We performed t-SNE in order to find a low-dimensionality clustered representation of the data, and to verify if the top 500 songs would be in some degree grouped together and apart from the rest of the songs. Nevertheless, this method showed that there is no clear segregation between the top 500 songs and other songs. We conclude the same as in PCA, that either the method is not good enough, or that these variables do not provide enough power to segregate the top 500 songs from the rest.

# Bibliography

[1] Good, Richard P, Daniel Kost, and Gregory A Cherry: *Introducing a unified pca algorithm for model size reduction.* IEEE Transactions on Semiconductor Manufacturing, 23(2):201–209, 2010.

[2] Felipe, González: *Métodos analíticos, itam-2022.* `https://felipegonzalez.github.io/metodos-analiticos-mcd-2022/`.

[3] Wikipedia: *Principal component analysis.* `https://en.wikipedia.org/wiki/Principal_component_analysis`.

[4] Andrii, Samoshyn: *Dataset of songs in spotify.* `https://www.kaggle.com/datasets/mrmorj/dataset-of-songs-in-spotify`.

[5] Magazine, Rolling Stones: *The 500 greatest songs of all time.* `https://www.rollingstone.com/music/music-lists/best-songs-of-all-time-1224767/`.

[6] StatQuest: *Statquest: t-sne, clearly explained.* `https://www.youtube.com/watch?v=NEaUSP4YerM`.

[7] Polcari, Fabio: *Spotify songs popularity analysis.* `https://www.kaggle.com/code/fpolcari/spotify-songs-popularity-analysis/notebook`.