CFGDegree Project: ANALYSIS OF SONG POPULARITY FROM SPOTIFY

by Alisa, Amily, Blessing, Hassanat and Madina

INTRODUCTION

● Aims and objectives of the project

The aim of our research is to investigate potential factors that might affect song popularity. By understanding what makes a song popular, artists and songwriters may better accommodate listener preferences in their production. In addition, listeners who have no idea what songs to listen to may find it easier to make choices when creating their playlists.

● Roadmap of the report

In this report, we will walk through the steps we took to gather our data, how the analysis was carried out and the final conclusions. We will also talk about the limitations and the challenges we encountered during the analysis.

BACKGROUND

The catalogue of the songs to be analysed and the popularity scores of tracks were obtained from Spotify, one of the largest music streaming service providers, with over 433 million monthly active users. According to Spotify, song popularity is calculated by algorithm and is based, in the most part, on the total number of plays the track has had and how recent those plays are. Generally speaking, songs that are being played a lot now will have a higher popularity than songs that were played a lot in the past.

Our analysis was mostly exploratory. We checked the correlation among a variety of songs' attributes to answer two questions:

- What factors affect a song's popularity?
- How did these factors affect the song's popularity?

STEPS SPECIFICATIONS

Step1: Data gathering

To answer our questions,we collected and developed data with the aid of Spotify API (track, artist, audio information), Kaggle dataset & Wikipedia (artist details), azlyrics (lyrics information). We came up with potential factors for different song popularity and categorised them into four groups:

1. Track-related: release date, duration
2. Artist-related: follower number, artist popularity, genres, age, nationality, gender, if it's a band
3. Audio-related: acousticness, danceability, energy (measure of intensity and activity), instrumentalness (measure of vocal involvement), key, liveness, loudness, mode (major or minor), speechiness (measure of presence of spoken words), tempo, time signature (number of beats in each bar), valence (sentiment)

4. Lyrics-related: explicit content or not, total number of words, average word length, average sentence length, repetition rate of words, proportion of unique words, common words in a song, length of the lines of lyrics, count of certain common words (you, yeah, love, go, etc.)

We split ourselves into three groups to work on the data sources:

Spotify API (track, artist, audio information)

We used sp.playlist() and sp.playlist_items() to fetch information about the tracks. To ensure we had the version with highest popularity (spotify doesn't have a function to rank tracks by popularity, and it's not done in default), we used sp.search() to search for the tracks with the same track name and artist name, and then picked the most popular one by applying a for loop. This gave us basic track information.

To get the artist information and audio features, we applied sp.artist() and sp.audio_features() respectively. After the gathering process, we combined these two files with the track information file by mapping dictionaries.

Kaggle dataset & Wikipedia (artist details)

Wikipedia scraping: We scraped the infobox sections for each artist to get their personal details and appended this to a dataframe that already contained the artist id and name. These details include birth name, birth date, place of birth, and whether or not the artist name was a band.

Gender dataset: This dataset was downloaded from Kaggle. It contained the artist name, gender and race of the dataset. The gender and wikipedia datasets were joined together using a left join to produce the artist-details dataframe.

Azlyrics (lyrics information)

For lyrics scraping, we used azlyrics.com websites. Using regex and built-in string methods, we formated the song/artist name, so that it could be used to generate the url for each song. By generating this url, we were able to scrape the text using the BeautifulSoup library. Then, lyrics metrics such as average word length, most frequent words etc were calculated and appended to a dataframe that already contained the artist id and track id.

Step2: Data Pre-processing

We did a number of cleaning after gathering the data:

Spotify API (track, artist, audio information)

We first removed duplicated songs, then we removed songs without a name and/or collaborated songs (more than one name in the field). Some of the artists had the same name but they were different people with different artist_id's. We removed these songs as well to prevent confusion.

Kaggle dataset & Wikipedia (artist details)

Using regex, we extracted the year from the birth year column which included date and month. We also extracted the country from the birth_place column which included state and country with the help of

country_converter (a python package). The gender and category columns had a number of missing rows due to lack of information.

Azlyrics (lyrics information)

After getting the lyrics for each song, we removed all the punctuation and case folded words (into lowercase) prior to the analysis. The lyrics were preprocessed in order to extract metrics, such as average word length. In some cases, stop words including articles, pronouns, prepositions etc. were removed to analyse most repeated words and complexity of the lyrics.

Some of the songs could not be found on the websites we used, so we removed these songs from the analysis.

Step3: In-depth Analysis

We combined all the datasets from ***Step2*** and dropped columns that will be redundant for our analysis.Then we had a closer look at our data and decided to eliminate some of the details in our columns (e.g. change release date accuracy from date to decade) and combine information into bigger groups.

To make it easier for graphing, we grouped our factors into three categories according to the data types.

(NB: column names are explained in Step1)

Unique data: artist_name

Numerical data:

duration(ms), artist_follower_no, artist_popularity, acousticness, danceability, energy, instrumentalness, key, liveness, loudness, mode, speechiness, tempo, time_signature, valence, repetitions_rate, mean_sentence_len, mean_word_len, unique_words_rate, total_words, total_char, count_word_i, count_word_you, count_word_oh, count_word_know, count_word_it, count_word_like, count_word_got, count_word_and, count_word_yeah, count_word_we, count_word_one, count_word_love, count_word_the, count_word_go, count_word_time, count_word_get, count_word_me, count_word_when, count_word_come, count_word_all

Categorical data:

release_date, release_year, explicit, artist_genres, birth_year, birth_decade, artist_country, gender, band

Firstly, we applied a heatmap to show the correlation between song popularity and all the numerical data in the dataset. We then checked the relationship between artists follower number, artist popularity and song popularity using line regression. This is because they showed some sort of positive correlation with song popularity.

Secondly, we proceeded to analyse the categorical factors. Here, we created density plots that showed the distribution of song popularity for each of the variables.

Thirdly, we subsetted the dataset and examined the artists with a popularity score greater than 86. There were 20 of them and we analysed their genres and countries using barplots.

Finally, we did an analysis of the artist details where we examined their gender, age and origin using scatterplots and Facetgrids.

IMPLEMENTATION AND EXECUTION

● Development approach and team member roles

Our development approach included data gathering, data cleaning, data analysis and data visualisations which happened sequentially then report writing and slides creation. Our responsibilities were as follows:

Group member	Tasks
Alisa	● Lyrics scraping ● Lyrics data processing ● Data analysis ● Data visualisation
Amily	● Spotify API ● Wikipedia data processing ● Data cleaning ● Data analysis ● Data visualisation ● Report writing ● Notebook creation & organisation ● Github creation & organisation
Blessing	● Wikipedia scraping ● Wikipedia data processing ● Data visualisation ● Slides creation
Hassanat	● Wikipedia scraping ● Wikipedia data processing ● Report writing ● Project management ● Github organisation
Madina	● Lyrics scraping ● Lyrics data processing ● Lyrics data analysis ● Notebook organisation ● Keeping meeting minutes

● Tools and libraries

Some of the python packages we made use of were:

1. csv: used to read and write data in csv format
2. wptools: python library used to get data from Wikipedia
3. country_converter: used to extract country information in our data
4. spotipy: python package used to get data from Spotify API
5. pprint: used for ‘pretty print’, makes life easier in dealing with API
6. selenium: used it to imitate the real behaviour user, to prevent the azlyrics website from blocking our requests for scraping.
7. bs4: makes is easier to scrape from web pages
8. nltk: natural language processing tool kit, used to detect common words
9. random: generate random numbers
10. pandas: used for analysis of dataframe

11. regex: used to match expressions in data, it was mostly used during cleaning
12. numpy: python library used for array manipulation
13. matplotlib & seaborn: python library used for visualisations
14. time: python package that converts time into consistent format
15. statistics: calculates statistics of numerical data

● Implementation process

During the project development we faced several challenges and gained a lot of the knowledge on data cleaning, and data analysis, as well as project management during the work on this project. One of the most challenging stages of the project was result analysis, when we found out that our initial theses were not proved, and we found almost no correlation between lyrics and song popularity. But then we understood that no correlation is also an insight, and it also made us look at data from a new perspective and rechallenge initial theses.

One of the decisions to change did happen, when we discovered rate limiters on music lyrics websites. In particular, when we were scraping websites using built-in python libraries, our requests got banned, which changed our whole scraping pipeline.

More implementation challenges are going to be discussed below.

● Implementation challenges

Some other challenges we encountered were:

1. Artist names on Spotify and Wikipedia were not written in the same way so we had to make some manual changes before scraping.
2. Wikipedia had a time limit for scraping so we had to scrape four times with each scraping being about 15mins long.
3. The birth place column from Wikipedia required a lot of cleaning. Asides from missing data, some entries did not include countries, just states, some contained multiple countries and were formatted in an inconsistent way.We used a country_converter package and some code to solve this problem.
4. To diversify our data sources, we decided to pick a dataset from Kaggle. This resulted in us having a lot of missing rows for the gender column. It was not easy to find already prepared data for artists' gender.
5. Getting the lyrics was the most challenging part of the data gathering process. Initially, we tried to get an API token from lyrics.com and request the data we needed but the website did not send the token to us. We then switched to scraping musixmatch.com but we got banned after multiple requests even while using multiple proxies and user agents; we had to purchase multiple proxies from proxy.webshare.io. Finally, we were able to scrape azlyrics.com by using the selenium python package to imitate real user behaviour.

● Agile development

We used an agile approach for the duration of the project. Although we created fixed tasks and schedules in JIRA, we moved back and forth between these processes. We conducted SCRUM meetings every Monday-Thursday after class and had a sprint review every Sunday via Google Meets. Our daily communication happened through Slack.

Code review was done via GitHub, we each created a branch and pushed our work there. To collectively participate in the project notebook, report and slides; we used Google slides, Google docs and Canva. This enabled us to make our changes and suggestions in real time. We also practiced paircoding when working on some parts of code in subgroups.

RESULT REPORTING

Earlier we grouped our factors into four groups: track-related, artist related, audio related and lyrics related:

For track-related factors, we found that duration has no correlation with song popularity, indicating that length of the song doesn't affect song popularity. According to the diagram, the majority of older songs have a higher score. However, song popularity of older songs is more condensed than current songs. This indicates, good songs stand up by time.

For artist-related factors, artist popularity and follower number had the strongest positive correlation with song popularity.

1) Artists follower number and artist popularity are positively correlated (relatively weak correlation)
2) Artist popularity and song popularity are positively correlated (relatively weak correlation)
3) There is a weak correlation between artist gender and his/her age at release (female artists become less popular earlier and quit singing earlier)
4) There are less female artists in the dataset than male, but the popularity is similar for both genders

For audio-related factors, there was no interesting correlation between any factors and song popularity

For lyrics-related factors, the majority of explicit songs have a higher song popularity than non-explicit songs. There was no other significant relation between lyrics and popularity.

CONCLUSION

Other than release year and whether the song is explicit, song popularity is generally affected by artists' popularity, follower number, genres, country of origin of the artist, and whether or not the artist is a band. This indicates that a song's popularity is highly dependent on the artist related factors and least dependent on the audio related factors.