

Initial Plan: using sentiment analysis to determine moods based on Spotify playlists

Author: Aaron Cheung

Supervisor: Dr Daniela Tsaneva

Project Description

Spotify is the world's largest music streaming service, with over 574 million users, including 226 million paying subscribers in more than 180 markets (**Spotify, n.d**). With these music streaming services becoming more popular and integrated within our everyday lives, it is interesting to analyse the type of music that we listen to, and the moods and emotions that are related to them. When we listen to music, it can affect how we feel – and conversely, we can choose what music we listen to depending on our current mood since it can “strengthen and change the emotions of the listener” as we enjoy listening to music that fits our situation (**Helmholz, Meyer, Robra-Bissantz, 2019**).

For my project, I want to analyse Spotify playlists to look for trends or patterns, by classifying/predicting the moods or emotions determined within the songs in these playlists. With the COVID-19 pandemic resulting in widespread lockdowns in 2020 and 2021, I want to see if our music listening habits changed as a consequence of this. For example, were the biggest hits prior to 2020 quite energetic and happy? And if so, was there a noticeable difference in emotion in the biggest hits of 2020 onwards? By using sentiment analysis, I intend to derive a predicted mood from a Spotify playlist (by analysing the lyrics in these playlists' songs) and use this data to see if and how the world changed the way they listened to music.

Although there are other streaming platforms, such as Apple Music and YouTube Music, I have chosen to focus on Spotify due to it being the significantly largest music platform. Additionally, with Spotify having social media-style features (public playlists, sharing, collaboration) as well as many pre-made playlists which can be accessed through their API, I felt that Spotify was the most appropriate platform to focus on.

Aims and Objectives

To approach this project, I intend to gather the Spotify playlists through the use of the Spotify Web API (**Spotify for Developers, n.d**). With their API, I can retrieve playlists and should be able to get the data/audio features of the songs in the playlist. To get the words from each song, I will use the Genius API to extract this data (**Genius, n.d**). Genius is a song lyric website which also has an API that I can use to access their database of song lyrics. Additionally, to make use of Spotify's Web API, I will use the Python Spotipy library, which is a lightweight Python library that gives me access to the data provided by the API. As a result, Python is the coding language that I will be using for this project.

With Spotify's API, the data that can be retrieved from each song can be used for further analysis of energy and mood – for example, some of the track features are: danceability, energy, tempo, and valence (a measure of 0.0 to 1.0 which describes the “positiveness” in a track, where a higher value represents a more positive, cheerful sound). In my implementation, the statistics that will likely be the most useful would be the valence, energy, and possibly danceability and tempo, though valence will be the most relevant.

For the sentiment analysis of the song lyrics, I will need to do some research and reading into the different approaches of using Natural Language Processing (NLP) methods. One possible way I could go is with the use of the Natural Language Toolkit (NLTK) Python library, which can be used to analyse textual data. This library also has a tool called VADER (Valence Aware Dictionary and Sentiment Reasoner) which I could use. There is also another option of using the scikit-learn to create a classifier - this research is what will be done and planned within the weekly plans. To add on to this, as there are different models of

sentiment analysis (e.g., Naïve Bayes Classifier, Support Vector Machine), part of my report may include a comparison of the accuracy of each one with regard to the playlists being analysed.

To display my findings and results of the models and the predicted moods of playlists, there are libraries and tools, such as pandas, matplotlib, and scikit-learn, of which I will read into which would be the most appropriate for me to utilise. For a user to see my findings, I will likely create a basic website (possibly with Flask, as I have past experience with it on a previous module) which will have my analysis and graphs of the playlist/music data. One feature that is not crucial to my project, but something I might look into as I develop my project, is the ability to allow users to log in their Spotify account and get information about their own playlists. This is something that *should* be somewhat simple to implement as the API supports this. However, as this is not a critical component of the project, it will only be attempted if I have enough time.

Aims

My overall aim of the project is to create a sentiment analysis tool that outputs a predicted mood from a Spotify playlist by analysing the lyrics in each of the playlist's songs.

Objectives

To achieve my aim, these are my objectives:

- Register and get access to Spotify API developer tools, Genius API for lyrics
- Get song features and data from a specific playlist (using Spotipy)
- Compare NLP models and create classifier
- Conduct sentiment analysis on song lyrics by using different NLP models
- Present playlist mood and data back to the user
 - Create simple website interface to show data visualisation and findings
- Research information regarding the effect of music on our mood, and any potential correlation with the impact of COVID

Feasibility and Risks

- There should be no risks with getting access to API tools as both Spotify and Genius offer free access as far as I know. I believe they only require an account to be made so that I can then create an API client for my application and set up a Spotify Developer Dashboard.
- With regards to retrieving data from Spotify playlists, there should be no risk involved because the playlists used in my project will be publicly available. As the API allows me to access public playlists, this should be fine.
- For the creation of a web interface to display my analysis, there may be a risk involved in terms of the time that I have to make it. Although I only plan on making a basic website, which shouldn't be too complex, it has been a while since I have created a website (and it also is not an area that I am particularly interested in). So, it is important that I do some background research on the tools that I need to deploy the website and to implement my Spotify data on it. I do have past experience with using Flask, but that was in a first-year coursework, so whilst it was a while ago, I have some experience.
- As I have a lack of experience with machine learning and NLP, there is a risk involved since time will need to be spent researching, experimenting, and getting

used to using these tools. As a result, I have included this consideration in my weekly work plan.

Ethics Approval

In my project, I intend to use Spotify playlists which are publicly available. For example, Spotify has their own public playlists of the top 100 songs streamed each year (e.g., Top 100 Songs of 2020 playlist). This is the data I will use for the sentiment analysis, as these songs will be indicative of the biggest global hits across the years.

If I do end up adding the feature of allowing users to login their Spotify account to look at the moods of their own playlist music, I will have to ensure that no data is stored, and that any analysis is anonymous and not saved.

After emailing Dr Katarzyna Stawarz of the ethics committee, I understand that I will not require ethics approval for my project, provided that I do not ask anyone to test out my work, and that I test my project only with own data or Spotify's public playlists.

Supervisor Meetings

For my project, I will plan on having meetings at least once a week with my supervisor Dr Daniela Tsaneva. In these meetings, I will discuss what work I have completed, as well as what work needs to be done for the following week. Also, I can show how much progress I have made through the use of a Gantt chart – a horizontal diagram which is used to monitor progress across tasks that need to be completed. A Gantt chart will be shown with this report which is what I will use during my project.

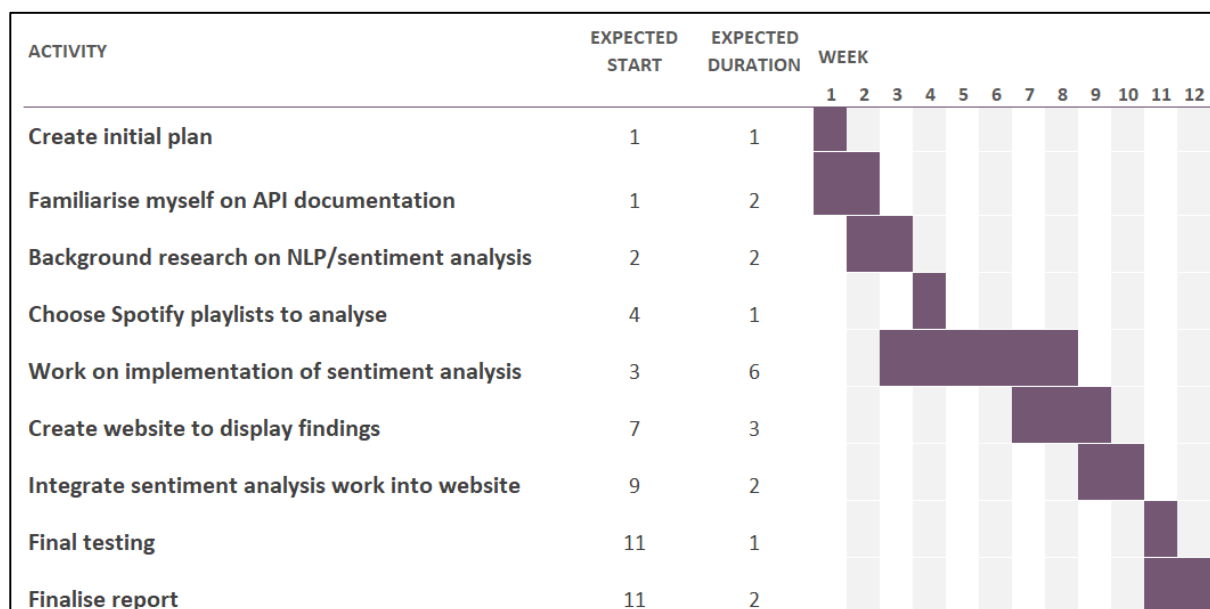


Figure 1 - Gantt chart progress tracker

Work Plan and Progress Tracking

Alongside my Gantt chart, I will have a weekly schedule/list of objectives that I will try to complete to ensure that I am on track and do not fall behind on work.

Week 1 (29/1/2024)

- Read up on previous archived projects to see how final year projects should be structured and developed
- Meet with supervisor to discuss initial ideas and the feasibility of the project – what features are crucial and what features I can leave as extras if I have time
- Read up on API documentation for Spotify and Genius – there are some tutorials I can look through to get an idea of how to utilise some of the API tools
- Complete initial plan, submit on PATS and send copy to supervisor

Week 2 (5/2/2024)

- Read up on sentiment analysis - research natural language processing models for this
- Look up and read through papers regarding music listening habits, relations to mood and emotions, effect of COVID, etc.
- Get access to Spotify and Genius API tools, and experiment on Python with them
- Briefly look into how I will deploy a web interface at the end – could look back at old Flask coursework

Week 3 – 4 (12/2/2024 & 19/2/2024)

- Continue using the API tools and make a start on retrieving playlist data
- Decide on the NLP models I will compare in my project. Start looking into implementation of these in my code

Week 5 (26/2/2024)

- Implement at least one NLP model with sentiment analysis on song lyrics
- Test sentiment analysis on a whole playlist
- Begin comparing NLP models to see which is most suitable for my sentiment analysis use

Week 6 - 7 (4/3/2024 & 11/3/2024)

- Use chosen NLP model to perform sentiment analysis on a range of playlists (e.g., each of the previous 10 years of top hits on Spotify) to look for changes in moods
- Start setting up basics of a website

Week 8 (18/3/2024)

- Finish developing back-end of website
- Start final report and write about my sentiment analysis implementation – will need to be concurrently reading related research papers regarding the music habit changes

Easter Break (25/3/2024 – 14/4/2024)

- Use spring break time only for catching up of any work that has taken longer than expected

Week 9 – 10 (15/4/2024 & 22/4/2024)

- Integrate sentiment analysis work into a website interface
- Work on front-end of website
- Allow user to login and use own playlist data (**not critical – only do as extra feature**)

Week 11 – 12 (29/4/2024 & 6/5/2024)

- Final testing of project
- Complete draft of final report
- Final checks of report and submission on **10/5/2024**

Reference list:

Genius. (n.d.). *Getting Started*. [online] Available at: <https://docs.genius.com/> [Accessed: 4/2/2024].

Helmholz, P., Meyer, M., and Robra-Bissantz, S. (2019). *Feel the Moosic: Emotion-based Music Selection and Recommendation*. BLED 2019 Proceedings, p. 3. Available at: <https://aisel.aisnet.org/cgi/viewcontent.cgi?article=1010&context=bled2019> [Accessed: 31/1/2024].

Spotify. (n.d.). *About Spotify*. [online] Available at: <https://newsroom.spotify.com/company-info/> [Accessed: 31/1/2024].

Spotify for Developers. (n.d.). *Web API*. [online] Available at: <https://developer.spotify.com/documentation/web-api> [Accessed: 4/2/2024].