u okay hun?

Powered by Spotify

01363020
Luke Hillery

Luke A Hillery 01363020
Dyson School of Design Engineering Imperial College London
DE4 SIoT

Project Links

Sensing & Data Collection code:

https://github.com/Kule123e45/u_okay_hun/tree/main/Data_Collection

Python Application code:

https://github.com/Kule123e45/u_okay_hun/tree/main/Web_Application/Python_Application

Web Application code:

https://github.com/Kule123e45/u_okay_hun/tree/main/Web_Application/Flask_Application

Collected data:

https://github.com/Kule123e45/u_okay_hun/blob/main/Time_Series_Analysis/u_okay_hun-results.csv

Time series analysis:

https://github.com/Kule123e45/u_okay_hun/tree/main/Time_Series_Analysis

Video:

https://vimeo.com/500234691

Live Data Dashboards:

https://www.lukehillery.com/u-okay-hun (password: iamnotokay)

https://thingspeak.com/channels/1277562

https://thingspeak.com/channels/1276455

INTRODUCTION AND OBJECTIVES

Project Overview

'U okay hun?' Is a web application designed to detect sudden changes in a Spotify user's listening patterns as a means of identifying changes in mood and behaviour associated with deteriorating mental health.

A model of a Spotify user's normal listening pattern is generated based on their music library or a favourite playlist they have selected. This is followed by continuous listening data collection. Features of each track played are recorded and analysed, both in real time and in hour-long listening 'sessions'. When consecutive listening sessions exhibit moods or characteristics that deviate significantly from the user's standard listening model, this triggers an automated text messaging system prompting the user, and aqn optional emergency contact, to change their current playback to a more positive selection. This intervention is designed to inform the user of how their music listening behaviour can have detrimental effects on their mental health and wellbeing.

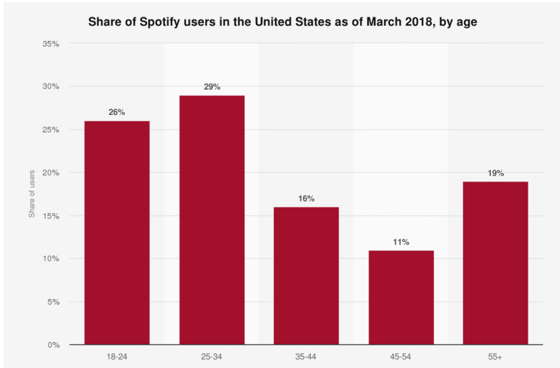


Fig 1. Spotify Users in the US by Age (Statista, 2018)

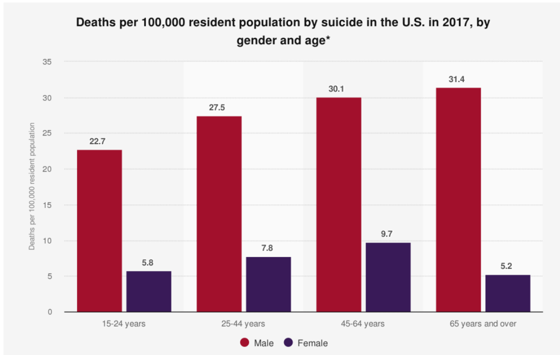


Fig 2. Deaths by suicide in the US by Age and Gender (Statista, 2017)

Suicide was the second leading cause of death among 15-29 year-olds globally in 2016 (1). This demographic has a large overlap with Spotify's user base, with 55% of active listeners aged 18-34 (based on available data from Statista, 2018). It is widely researched and proven that music is capable of altering moods and behaviours, yet little has been done to utilise the myriad of data available through the APIs provided by streaming platforms.

Having lost a close family friend to suicide late last year, I was left asking myself what more I could have done to save his life, what signs I might have missed and how I could have known how low he was feeling.

Research has shown that young people are especially likely to turn to music when in a negative mood (2), in fact withdrawal from socialisation and normal daily activity has been identified as a behaviour consistent with clinical depression and this often involves an increase in media consumption. However, they are often unaware how this consumption affects their mood. **'People with depression are most likely to use music to intensify a negative mood. They are also least aware of this tendency.'(3)**

Project Planning

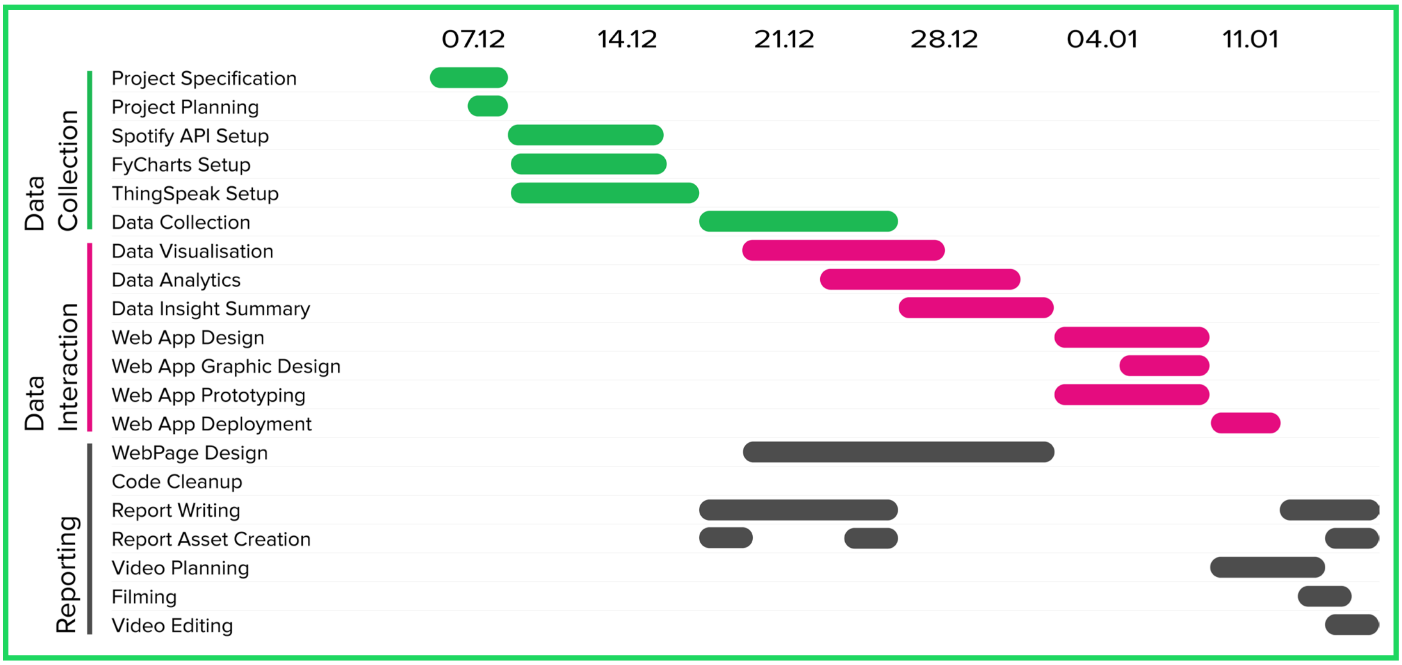


Fig 3. Project Planning Gantt Chart

Background

This report demonstrates an effective methodology for gathering data on the track features of songs played by a user on Spotify as a way to monitor changing moods. A python script accesses a Spotify user's playback data through the use of Spotify's API and stores the data in an online database for visualisation and analysis. This study looks at the correlation between track features supplied by the Spotify API 'track_feature' endpoint and compares the Spotify user's current behaviour to their normal listening patterns in terms of track valence, energy, danceability and tempo.

Spotify is a music streaming platform with over 70 million subscribers globally, the second most popular service in the world. Spotify have previously collaborated with weather data providers (such as AccuWeather) to find that weather has an effect on music preferences, however little has been done utilising music preferences as a means to estimate user moods.

Within this project, the local (GB) and global chart moods will be tracked and the Spotify data from my personal account will be recorded to find if there is a correlation between user, local and global moods. This data will be used to create a webpage which automatically alerts the user and an optional second contact when their listening behaviour deviates significantly towards a more negative sound than usual.

Objectives

The objective of this study is to quantify the mood of the user's music preference hourly and compare this to the daily local and global chart moods.

- Monitor playback state and record track features including genre, tempo, valence, energy and danceability

- Create 'sessions' based on songs listened to in a one-hour playback period

- Analyse the raw, session and chart data

- Analyse and infer possible meanings behind any trends discovered

- Present visualisation of the data collected

- Develop a web application implementing the data or conclusions drawn from the study

Considerations

PRIVACY & SECURITY

Careful consideration was given to privacy and data protection. In implementation, all sensitive variables were imported from a config file which was stored offline on the user's device. When the code for the project was committed to the GitHub repository, all client credentials, secrets and sensitive data were removed from the config file to protect the owner. Exported data was stored on Google Drive, which uses secure data centres.

A key aspect of the project was the amount of time that data could be gathered for. As the project continued, there was a threat that the data collection start date would have to be pushed back. For that reason, an underdeveloped version of the data collection script was implemented early in the project. Once the data was being collected, the code could not be changed without disrupting the periodicity of the data collected, invalidating previously collected data. This meant that code had to be worked around. Additional data collection scripts were added later.

FLASK/HTML

For the actuation part of coursework 2, Flask was utilised to develop a web application around the existing python functions. Flask is a python microframework for communicating with html. The backend of the application was python; which called the APIs and processed the reponses. Python functions are called when buttons are clicked on the web application, which interact with the Spotify API using Python's requests module. Custom HTML based on templates from w3schools was written to build the web application pages.

DATA SOURCES AND COLLECTION

Two APIs were used to collect data for this project; Spotify and Fycharts. Both APIs are based on Representational State Transfer (REST) principles. They utilise GET, PUT, POST and DELETE HTTP requests to access and manipulate requested data. RESTful APIs (those adhering to REST principles) are more robust and use less bandwidth than the alternative Simple Object Access Protocol (SOAP) APIs.

Spotify API with Spotipy Wrapper

AUTHENTICATION & AUTHORISATION

Spotify's API is a robust RESTful API with an unspecified request limit. All requests to the Spotify API require client (web application) authorization following the OAuth 2.0 authorization framework. This application must be registered to a Spotify Developer's account prior to sending any HTTP requests.

There are four optional flows to obtaining app authorisation;

1. Authorisation Code - suitable for long-running application in which the user grants permission only once.

2. Authorisation Code with Proof Key for Code Exchange (PKCE) - as above for mobile and desktop apps where it may be unsafe to store the client secret.

3. Implicit Grant - for client applications implemented entirely using JavaScript and running in the resource owner's own browser.

4. Client Credentials - for use in server-to-server authentication where endpoints do not access user information.

To access and manipulate user data, such as private playlists and playback data, user-level authentication with specified permissions is needed. User-level permissions can only be requested and approved through the authorisation code flows (1 and 2 above), This is achieved through a user log-in and authorisation prompt following the OAuth 2.0 framework, in which a 'bearer' and 'refresh' token are issued to the application. The bearer authentication token expires every hour and is refreshed upon expiration by sending a token refresh request carrying the cached refresh token. All client credentials, including client ID, client secret and redirect uri, are stored offline to protect the sensitive data. None of these details have been committed to the online repository.

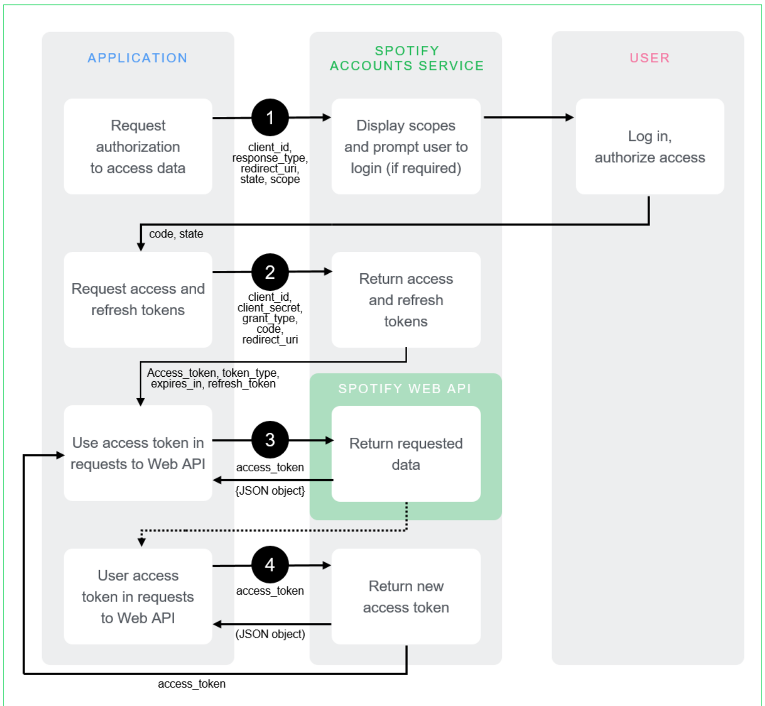


Fig 4. Spotify OAuth 2.0 Authorisation Code Flow (Spotify, 2021)

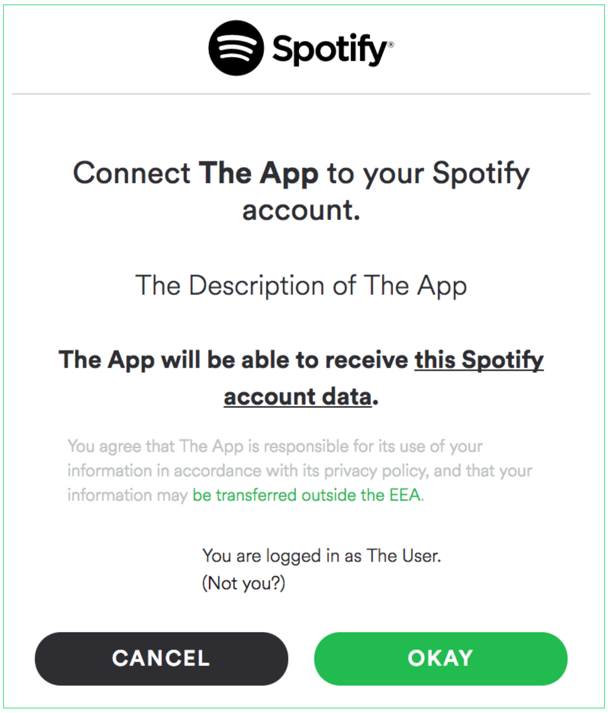


Fig 5. Spotify User Permission Prompt (Spotify, 2021)

Within the first authorization request to the Spotify user, a scope for app permissions must be defined. The scopes required for this application are:

- user-read-private
- user-read-currently-playing
- user-read-recently-played
- user-read-playback-state
- playlist-read-private
- playlist-read-collaborative
- user-library-read
- user-top-read

Requests requiring permissions out of the scope will be met with a HTTP 403 Forbidden error.

LIVE RAW DATA COLLECTION

The data collected using this API included:

- Currently Playing (Boolean T/F)

Returned in the .json file as 'True' or 'False', then converted into 1 or 0 respectively in the python script for data display.

- Tempo of Currently Playing Track

'The overall estimated tempo of a track in beats per minute (BPM). In musical terminology, tempo is the speed or pace of a given piece and derives directly from the average beat duration' (6). Returned as a floating point decimal (float).

- 'Valence' of Currently Playing Track

'A measure from 0.0 to 1.0 describing the musical positiveness conveyed by a track. Tracks with high valence sound more positive (e.g. happy, cheerful, euphoric), while tracks with low valence sound more negative (e.g. sad, depressed, angry)' (6). Returned as a float.

- Genre of Currently Playing Track

Returned as a text string (where available).

- 'Danceability' of Currently Playing Track

'Danceability describes how suitable a track is for dancing based on a combination of musical elements including tempo, rhythm stability, beat strength, and overall regularity. A value of 0.0 is least danceable and 1.0 is most danceable'(6). Returned as a float.

- 'Energy' of Currently Playing Track

'Energy is a measure from 0.0 to 1.0 and represents a perceptual measure of intensity and activity. Typically, energetic tracks feel fast, loud, and noisy. For example, death metal has high energy, while a Bach prelude scores low on the scale. Perceptual features contributing to this attribute include dynamic range, perceived loudness, timbre, onset rate, and general entropy' (6). Returned as a float.

Given the average length of a song is 3m30s, according to Nyquist's Theorem the sampling rate should be set to less that 1m45s. It was found experimentally that the rate limit on Spotify's API is around one request every 2 minutes, so it was decided that this would be the sampling frequency.

LIVE PROCESSED DATA COLLECTION

The python script responsible for collecting and posting the Spotify listening data also collated the raw data into 'sessions'. The data for each session was defined by the average feature value across each song played in that session's one-hour playback window. These averaged sessions provided more coherent data and removed much of the noise attributed by diverse listening patterns.

- Session Tempo
- Session Energy
- Session Danceability
- Session Valence

FyCharts

FyCharts is a 'fully-fledged installable python package for extracting top 200 and viral 50 charts from spotifycharts.com', created to fill the gap left when Spotify deprecated their own official Spotify charts API. It was utlised to collate lists of the top tracks trending both locally (in Great Britain) and globally.

AUTHENTICATION & AUTHORISATION

FyCharts does not require any authentication or authorisation in order to access the chart track lists. The Spotify API discussed previously was used to analyse the track lists to retrieve their audio features.

DATA COLLECTION

The chart data was collected using a python script scheduled using crontab scheduler. The unique Spotify id of each track was extracted in python, then the features of each track were extracted through the use of the Spotify API.
The mean and standard deviation of each feature value were calculated using python's statistics module, then these features were posted to a ThingSpeak database for visualisation and collation with the raw and session data.

Due to output limitation of the FyCharts API, it was later found that manually rnning the script each day was more reiable and accurate. TypeErrors frequently occurred when parsing the internaional chart data due to special characters and missing features.

Local (GB) Top 200 Songs

- Local Chart Tempo
- Local Chart Danceability
- Local Chart Valence
- Local Chart Energy

Global Top 200 Songs

- Global Chart Tempo
- Global Chart Danceability
- Global Chart Valence
- Global Chart Energy

DATA FREQUENCY

Since the data available to FyCharts is only updated daily, this data was sampled twice each day to ensure no aliasing occurred.

DATA STORAGE

ThingSpeak Data Collection

The data collected by the python scripts was posted to ThingSpeak, an IoT analytics platform service that allows live data stream aggregation, visualisation, and analysis in the cloud. It was selected due to it's built in functionality with MATLAB for complex analysis and visualisation tools. The data was posted using the platform's RESTful API, using GET requests that included the Channel's API Write Key and field IDs.

```
request = requests.get('https://api.thingspeak.com/update?api_key='+config.THINGSPEAK_CHANNEL_KEY+'&field1='+str(session_valence)...
```

The data was collected in 14 streams across two ThingSpeak channels as each channel is limited to 8 streams. The platform automatically timestamps and numbers each data entry for internal visualisation and dataset exporting.

ThingSpeak has a rate limit of one request every 15 seconds for student accounts however this was not an issue to the Spotify API being the limiting factor in data collection. The data entries could later be retrieved with another request.

```
prev_request = requests.get('https://api.thingspeak.com/channels/1277562/fields/1/last.json').json()
```

SYSTEM CHARACTERISTICS

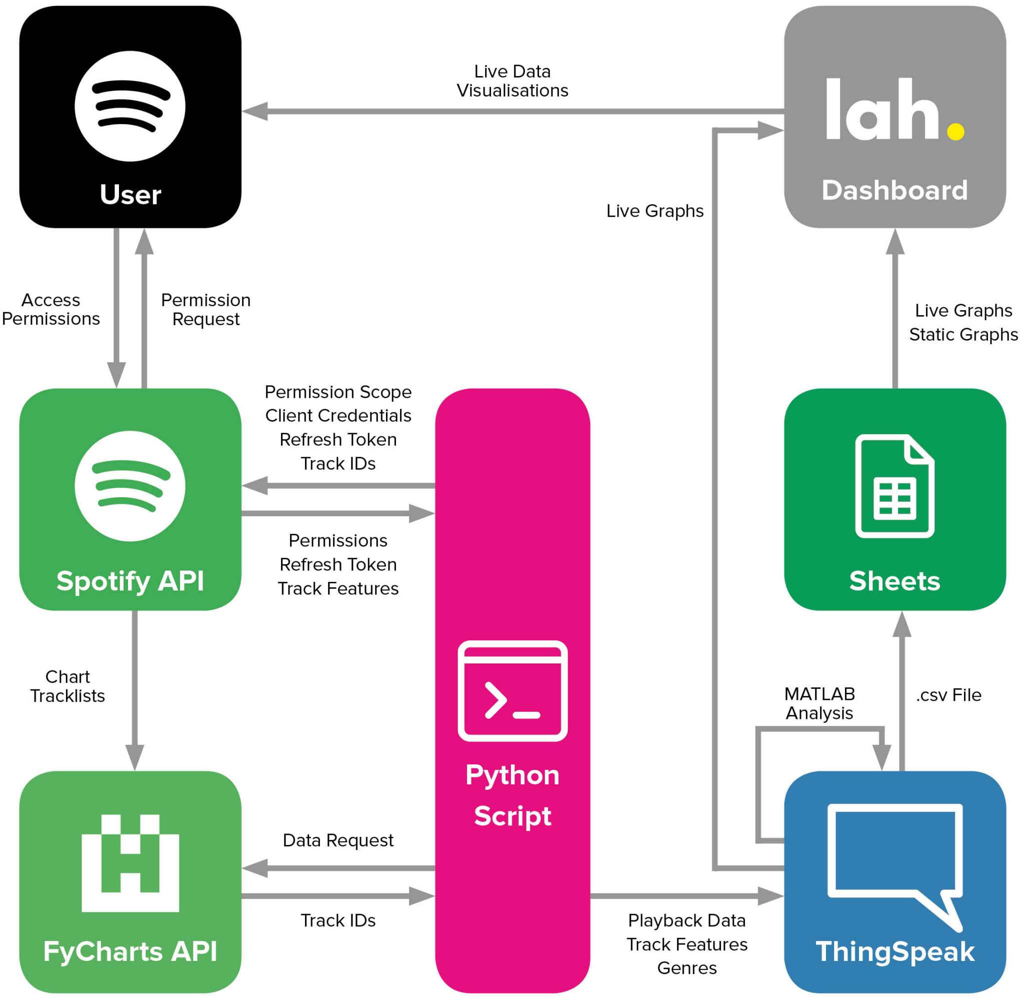


Fig 6. Data Collection System Diagram

Data Collection System

The python script collected authorised user data from the Spotify API, then collated the features of the currently playing track and posted these features to separate fields on a ThingSpeak channel along with playback as a binary value (0 = not playing, 1 = playing) and the track genre. The script also collated session data internally to publish hourly session summaries to separate fields on a further ThingSpeak channel.

DATA VISUALISATION

Basic Characteristics

VISUALISATION

The noisy data associated with diverse music preferences made raw data visualisation impractical and ineffectual. With a sampling rate of once every 2 minutes, over 5500 raw feature data points were collected and displayed over a 7-day listening period.

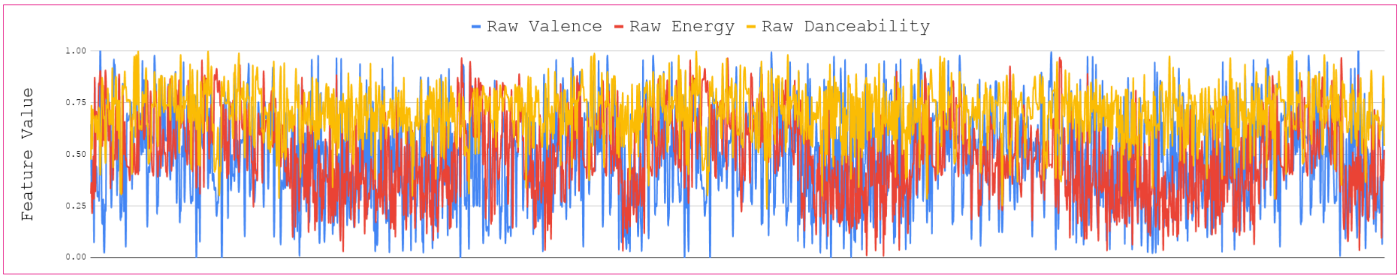


Fig 7. 7-Day Raw Track Features

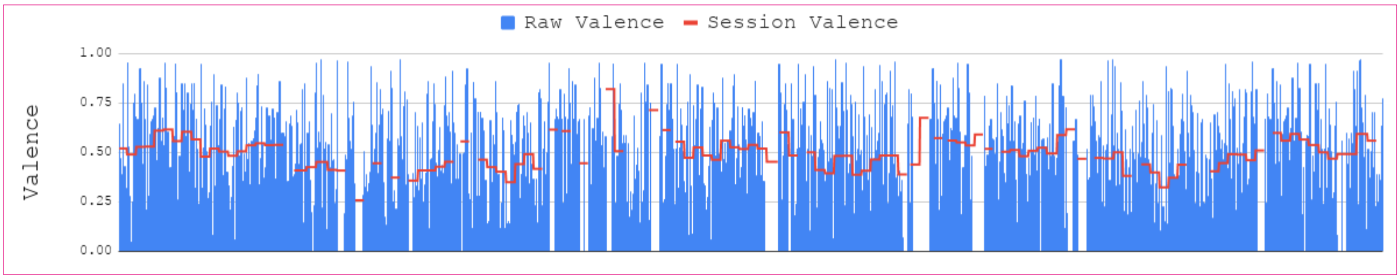


Fig 8. 7-Day Raw Valence Reading with Session Data

Listening Sessions

VISUALISATION

Visualising the listening behaviour in ‘sessions’ provided a much clearer view of the features. They were displayed with the global and local chart average features to display any obvious correlation between the behaviour of the user and that of wider populations. Despite removing much of the noise, there were still no obviously visible correlations between the user’s track playback characteristics and those deduced from local and global listening behaviours. Additional analysis was needed to identify less obvious correlations between datapoints. Figures 9 and 10 show the average valence and tempo values respectively for all listening sessions across a 5-day period. The mean valence and tempo of the local and global charts are also indicated on the respective graph in red (local) and blue (global). It can be inferred from the graphs that the local and global charts deviate very little from each other in terms of average values for track features. Additional visualisations of all track features can be found in the Appendix.

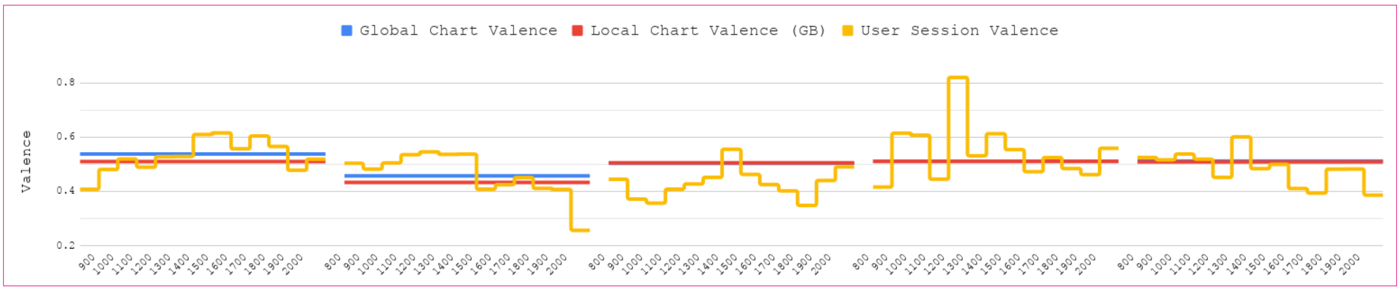


Fig 9. 5-Day Session Valence vs Local and Global Pattern

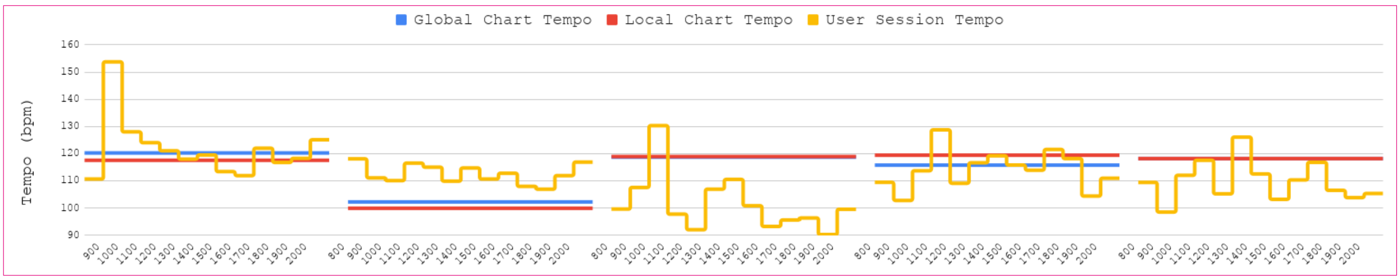


Fig 10. 5-Day Session Tempo vs Local and Global Pattern

ACTUATION

The web application ‘u okay hun’ is designed to identify sudden changes in a Spotify user’s listening behaviour that deviate towards a more negative sound. The user can provide their own phone number and an optional emergency contact to be notified by SMS when this occurs, in an effort to identify and intervene in situations where the listener’s mental health may be deteriorating.

Twilio API

Twilio is a service that can deliver SMS alerts, notifications, and reminders with the Programmable Messaging API. It can also provide customer interaction through two-way SMS messaging with the Conversations API. Once registered, Twilio provides a client id and authentication token/client secret, which can then be used to send HTTP POST requests to the Messaging API. With a trial account, SMS messages may only be sent to numbers registered to a Twilio account.

```
message = 'Hello from Twilio API'
to = '+447952165272'
def TextAlert(message, to):
    message = client.messages \
                .create(
                    body=message,
                    from_='+14343255105',
                    to=to
                )
```

Fig 11. Sample Twilio SMS

The Intervention

THE THEORY

‘Research has demonstrated that young people are even more likely to turn to media when they are in a negative mood (3). In fact, withdrawal from socialization and normal daily activity has been identified as a behaviour consistent with clinical depression and this often involves an increase in general media use (4). This increased engagement with media includes music listening, with emotional dependecy on music also tending to increase during periods of depression (5).’(2)

In a study examining the factors that influenced the outcome of music listening for people with symptoms of depression, it was found that ‘a key factor was the level of awareness and consciousness with which individuals selected music’ (2). Some participants were able to list a genre or artist which caused their mood to deteriorate, but only upon direct request. Other particpants were made aware of the negative affects of the music on their mood only by comments from their friends and family.

On the whole It was found that ‘intervening conditions including insights gained from friends, family, a therapist or through self-reflection resulted in increased awareness’ (2).

‘It appears that people with depression are most likely to use music to intensify a negative mood. They are also least aware of this tendency.’(2)

THE INTERVENTION

‘U okay hun?’ aims to increase awareness of music’s influence on mood in Spotify users. It provides the prompt necessary for self-reflection on music choice without the need for a human continuously observing and analysing the changing listening behaviour. The myriad of data available from streaming platforms such as Spotify is an untapped resource in positive social interventions such as this. Although this is not a faultless solution, it could be an invaluable tool in helping to detect deteriorating mental health in spotify users with minimal investment.

TEST CASE

Although I did not have access to his full music library, I tested the application using a list of tracks discovered on my family friend’s laptop by his parents after he took his own life. The mood of these tracks fell signifcantly below the average valence of tracks on Spotify and of the the user model generated from my own music library.

```
==============================================
**SESSION SUMMARY**
Session mean valence: 0.387, deviation = 0.250
Session mean energy: 0.351, deviation = 0.152
Session mean danceability: 0.673, deviation = 0.148
Session mean tempo: 105.356, deviation = 33.770
==============================================
```

Fig 12. Session Summary Example

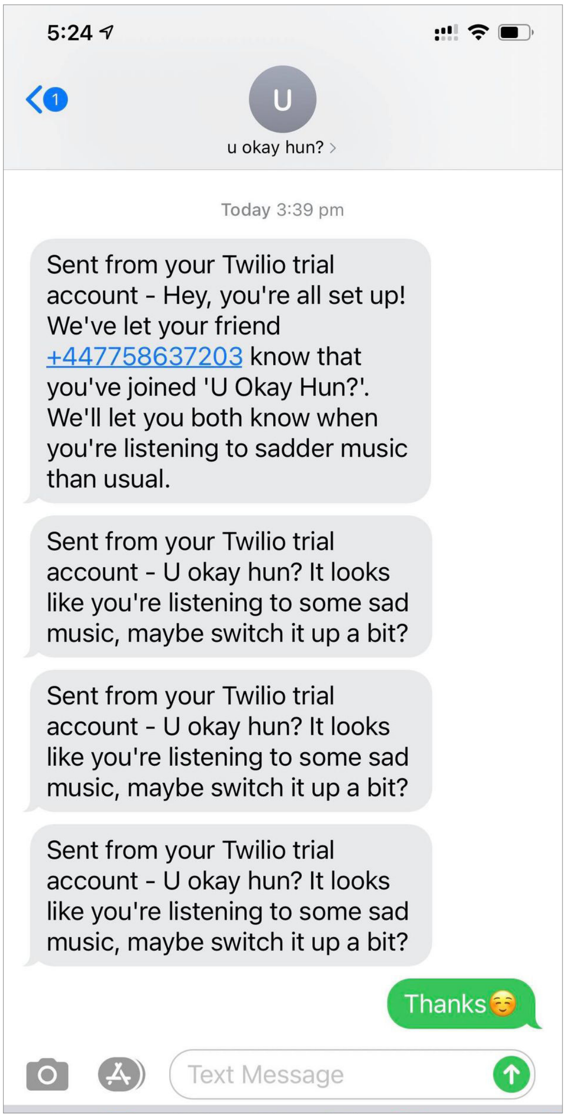


Fig 13. Twilio-powered SMS Intervention

Web Application

Please see the video linked on the ‘Project Links’ page of this report for a live demonstration of the application.

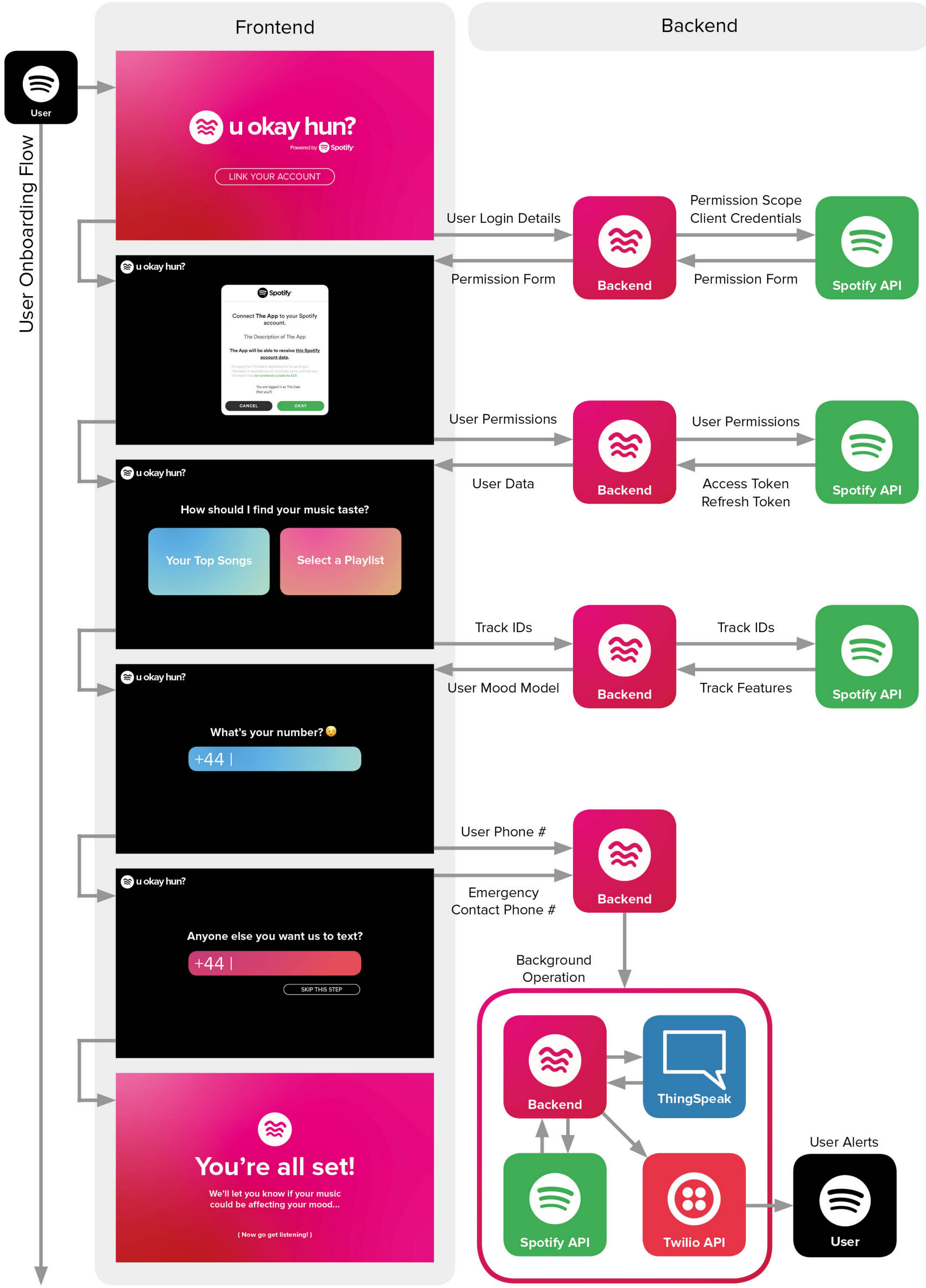


Fig 14. u okay hun? System Diagram

DATA ANALYTICS

Correlation

A strong correlation was found between the track features in the local and global charts. No correlation was found between the user's mood and the mood of these charts. More data from a greater number or users is required to confirm whether a user's mood is affected by the most popualr tracks in that region. The features of each track were also analysed for correlations and no obvious correlations were found. Additional data is necessary to infer any conclusions.

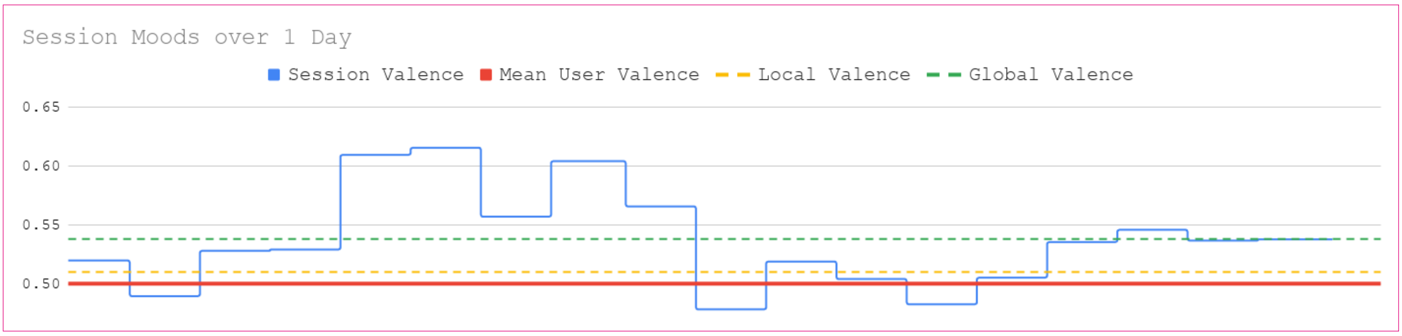


Fig 15. Session, Local Daily and Global Daily Valence

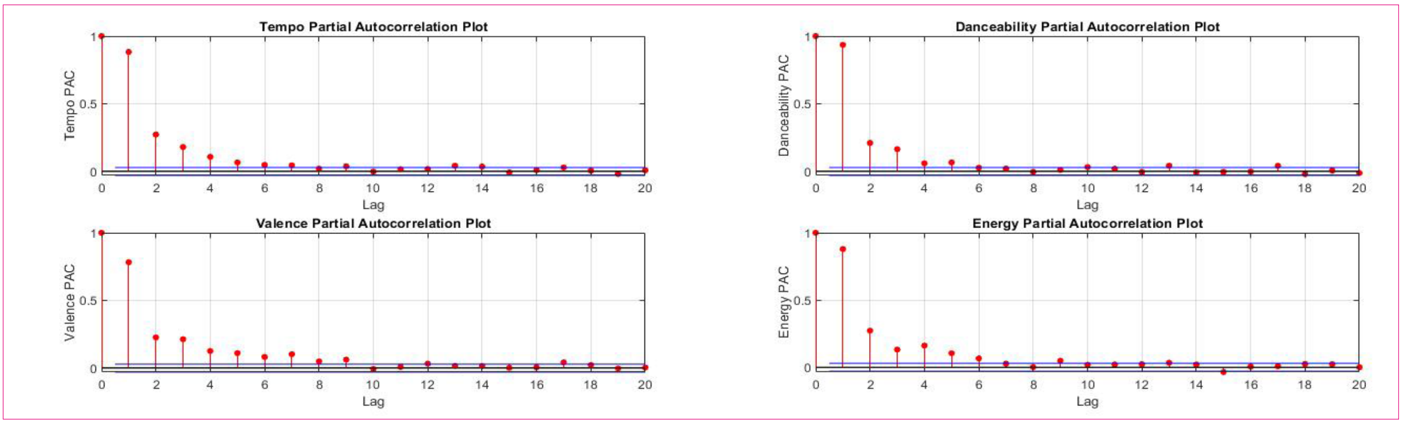


Fig 16. Raw Track Feature Partial Autocorrelation

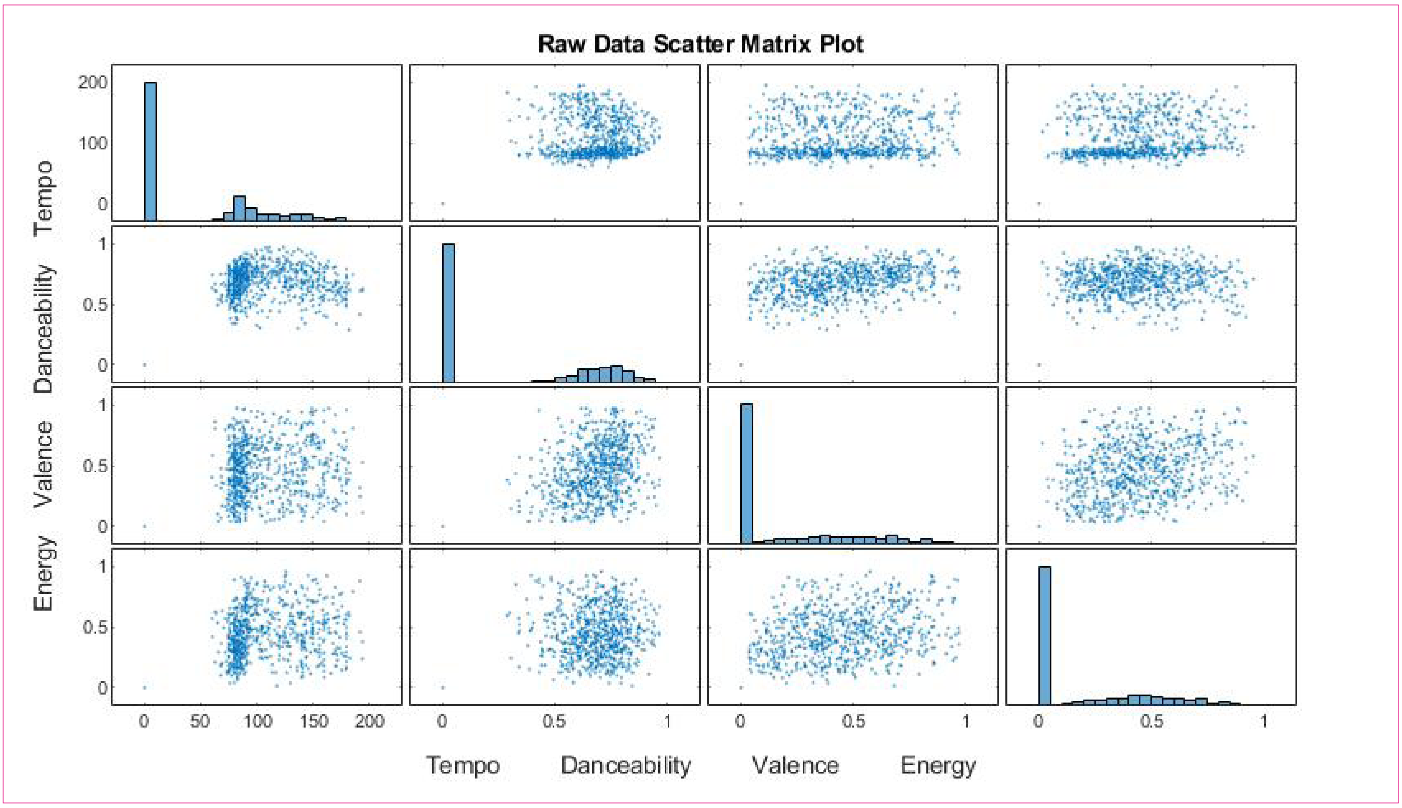


Fig 17. Raw Track Feature Scatter Matrix

DISCUSSION

Improvements

IMPROVING THE USER MODEL

The app currently uses a very basic model for user behaviour based on the user's most frequently played songs on Spotify. A more complex model should be developed to take into consideraton seasonal and global shift in music preferences, as well as noticing a gradual change in musical preferences in individual users over time. To implement these features, significantly more data is needed spanning many more users.

HYPERLOCALISED MOODS

In its current implementation, 'local' mood refers to the mood of the top 200 songs in Britain. Application such as 'Spotify's Musical Map of the World' by Eliot van Buskirk (https://spotifymaps.github.io/musicalcities/) show that it is possible to build hyperlocal models of listening behaviour. Utilising the highly localised playlists generated by this app would allow for a much more accurate 'local mood' to be measured and analysed for correlation to the user.

Summary

After completing a prototype of the application, its concept and execution was reviewed by a practicing counsellor:

"Suicidal thoughts can often be temporary, so simple primary interventions like asking someone if they are ok, and/or signposting them to where they can get help immediately, such as Samaritans or SHOUT, could just save a life. In fact Small Talk Saves Lives is a campaign run by National Rail, in partnership with Samaritans and staff across the wider rail industry, based on the very idea that a little small talk, and just asking someone if they need any help, can be all it takes to interrupt someone's suicidal thoughts."

- Linda, Practicing Counsellor and Samaritans Listening Volunteer

NEXT STEPS

Future Work

APP DEPLOYMENT

The next step would be deploy the web application to a web server to allow other Spotify users to sign up and analyse their own listening pattern.

SONG LYRIC SENTIMENT ANALYSIS

To further improve the mood analysis of both the user library and the live plaback data, sentiment analysis could be implemented to determine positivity and negativity values based on the lyrics of the songs. Many of the songs categorised as 'positive' by their valence values have lyrics that contradict this, and vice versa. Adding sentiment analysis powered by a neural network would greatly improve the mood analysis of each track.

CORRELATION BETWEEN SUICIDE RATES AND LISTENING PATTERNS

Data available on deaths by suicide in geographical areas could be analysed for correlations to Spotify listening patterns in those areas in a given timeframe to determine if there is a significant correlation between listening behaviour and seriously deteriorating mental health geographically.

Potential Impact

MENTAL HEALTH CHARITIES

With many charities such as Mind and Samaritans struggling to cope, particularly in the socially isolated 'new normal' caused by the Covid-19 pandemic, the need for automated solutions in mental health interventions is growing rapidly. Although interventions like the one discussed in this report could never replace the invaluable support offered by the volunteers staffing phone lines and email accounts at these charities, there is a space that this category of minor intevention can occupy. Further development could include signposting to other online wellbeing apps such as Calm or Headspace, or providing crisis support via direct links to charities offering 24/7 support via phone, email ot text.

'U okay hun?' offers a way of creating mass awareness of the way music affects the mood of a generation. It takes less than a minute to register for the application (in its current state) and has the potential to save countless lives. Failing that, it can at least pull someone out of a downward spiral of negative music and mood that they would otherwise remain unaware of.

7. References

(1) https://www.who.int/news-room/fact-sheets/detail/suicide.

(2) Dillman Carpentier FR, Brown JD, Bertocci M, Silk JS, Forbes EE, Dahl RE. Sad Kids, Sad Media? Applying Mood Management Theory to Depressed Adolescents' Use of Media. Media Psychology [Internet]. Media Psychology; 2008;11(1):143–66. Available from: https://dx.doi.org/10.1080/15213260701834484.

(3) Stewart J, Garrido S, Hense C, Mcferran K. Music Use for Mood Regulation: Self-Awareness and Conscious Listening Choices in Young People With Tendencies to Depression. Frontiers in Psychology [Internet]. Frontiers in Psychology; 2019;10. Available from: https://dx.doi.org/10.3389/fpsyg.2019.01199.

(4) O'Keeffe GS, Clarke-Pearson K. The impact of social media on children, adolescents and families. Pediatrics [Internet]. Pediatrics; 2011;127. Available from: https://pediatrics.aappublications.org/content/127/4/800.

(5) Mcferran KS. Contextualising the relationship between music, emotions and the well-being of young people: A critical interpretive synthesis. Musicae Scientiae. Musicae Scientiae; 2016;20(1):103–21. Available from: https://doi.org/10.1177%2F1029864915626968.

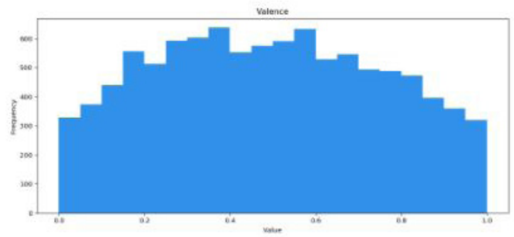
(6) https://developer.spotify.com/documentation/web-api/reference/tracks/get-audio-features/

APPENDIX

Spotify Track Features

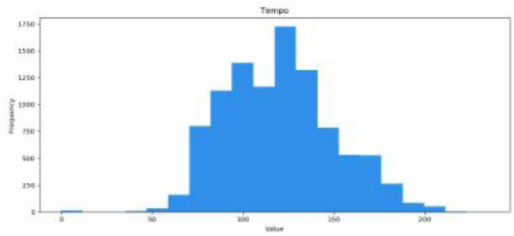
Valence

A measure from 0.0 to 1.0 describing the musical positiveness conveyed by a track. Tracks with high valence sound more positive (e.g. happy, cheerful, euphoric), while tracks with low valence sound more negative (e.g. sad, depressed, angry). The distribution of values for this feature look like this:



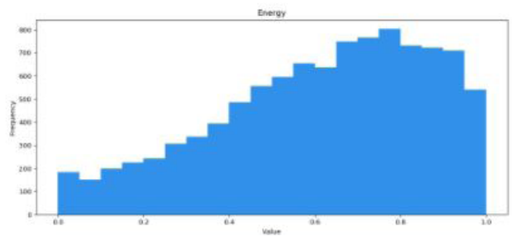
Tempo

The overall estimated tempo of a track in beats per minute (BPM). In musical terminology, tempo is the speed or pace of a given piece and derives directly from the average beat duration. The distribution of values for this feature look like this:



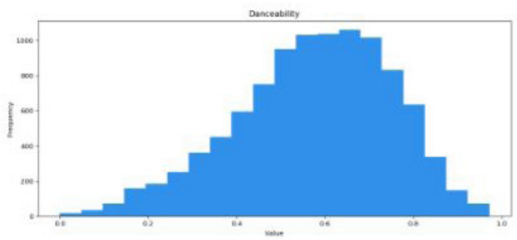
Energy

Energy is a measure from 0.0 to 1.0 and represents a perceptual measure of intensity and activity. Typically, energetic tracks feel fast, loud, and noisy. For example, death metal has high energy, while a Bach prelude scores low on the scale. Perceptual features contributing to this attribute include dynamic range, perceived loudness, timbre, onset rate, and general entropy. The distribution of values for this feature look like this:



Danceability

Danceability describes how suitable a track is for dancing based on a combination of musical elements including tempo, rhythm stability, beat strength, and overall regularity. A value of 0.0 is least danceable and 1.0 is most danceable. The distribution of values for this feature look like this:



Spotify API Permissions

`playlist-read-collaborative`

Description	Include collaborative playlists when requesting a user's playlists.
Visible to users	Access your collaborative playlists.

Endpoints that require the `playlist-read-collaborative` scope

- Get a List of Current User's Playlists
- Get a List of a User's Playlists

`user-read-playback-state`

Description	Read access to a user's player state.
Visible to users	Read your currently playing content and Spotify Connect devices information.

Endpoints that require the `user-read-playback-state` scope

- Get a User's Available Devices
- Get Information About The User's Current Playback
- Get the User's Currently Playing Track

`user-read-recently-played`

Description	Read access to a user's recently played tracks.
Visible to users	Access your recently played items.

Endpoints that require the `user-read-recently-played` scope

- Get Current User's Recently Played Tracks

`playlist-read-private`

Description	Read access to user's private playlists.
Visible to users	Access your private playlists.

Endpoints that require the `playlist-read-private` scope

- Check if Users Follow a Playlist
- Get a List of Current User's Playlists
- Get a List of a User's Playlists

`user-read-private`

Description	Read access to user's subscription details (type of user account).
Visible to users	Access your subscription details.

Endpoints that require the `user-read-private` scope

- Search for an Item
- Get Current User's Profile

`user-top-read`

Description	Read access to a user's top artists and tracks.
Visible to users	Read your top artists and content.

Endpoints that require the `user-top-read` scope

- Get a User's Top Artists and Tracks

`user-read-currently-playing`

Description	Read access to a user's currently playing content.
Visible to users	Read your currently playing content.

Endpoints that require the `user-read-currently-playing` scope

- Get the User's Currently Playing Track

`user-library-read`

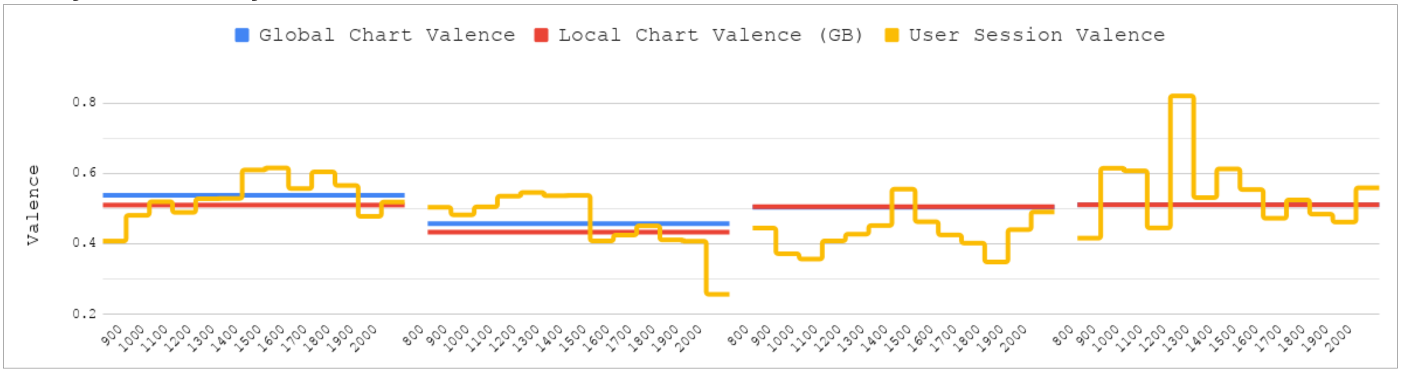
Description	Read access to a user's "Your Music" library.
Visible to users	Access your saved content.

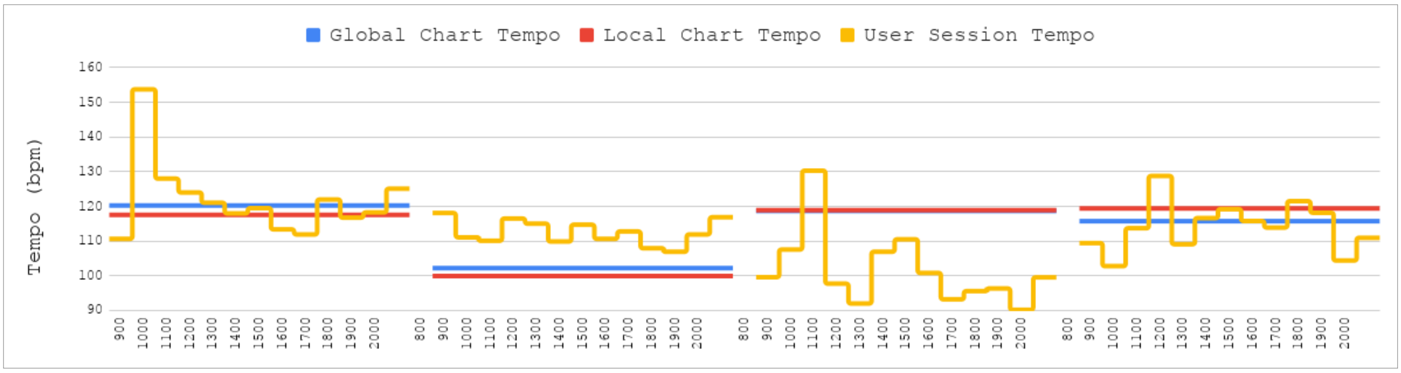
Endpoints that require the `user-library-read` scope

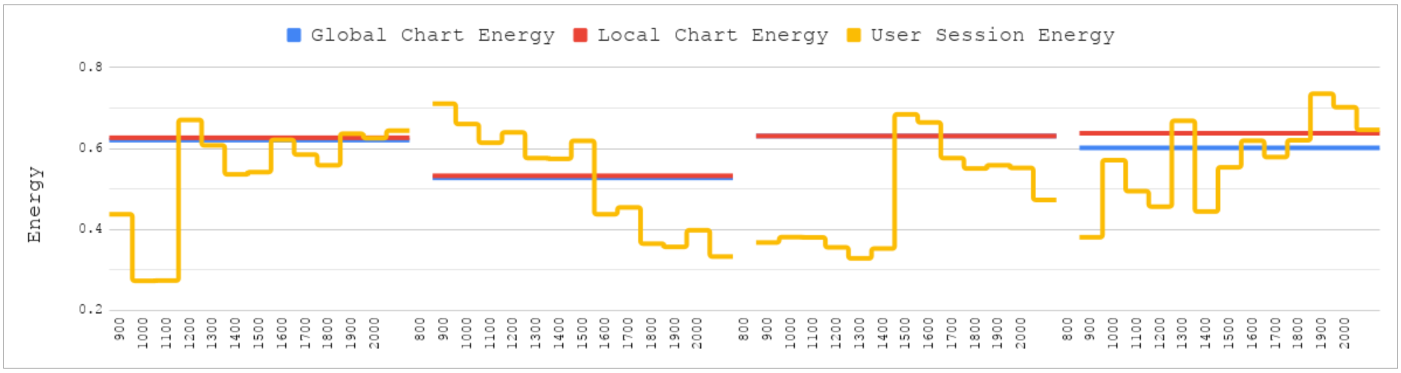
- Check User's Saved Albums
- Check User's Saved Tracks
- Get Current User's Saved Albums
- Get a User's Saved Tracks

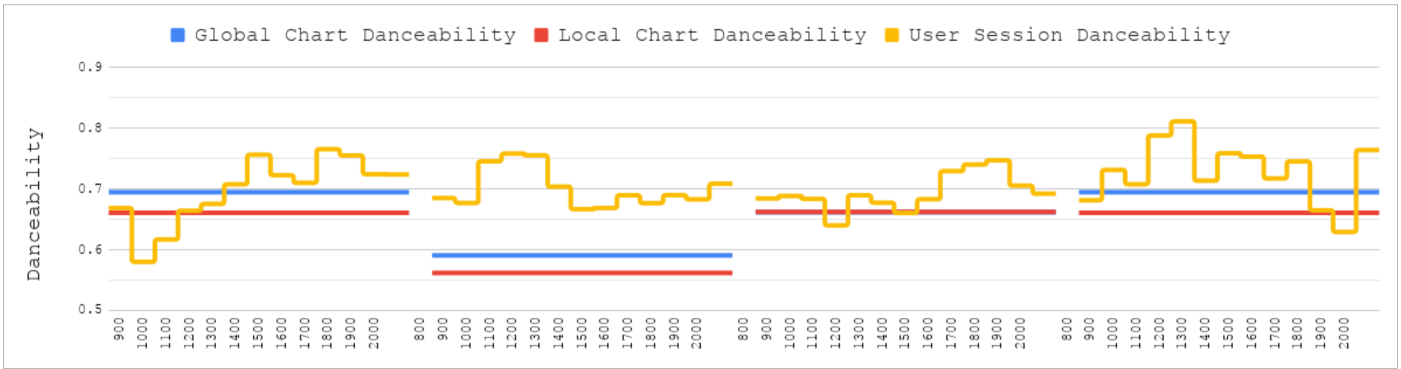
Data Visualisations

4-Day Features by Session









Powered by Spotify