Imperial College London

Dyson School of Design Engineering

DE4-SIOT | Sensing & IoT

Leah Pattison | 01052268
10th January 2019

Project Links

All files available at:
https://github.com/leahpattison/Sensing-IOT

Video available at:
https://youtu.be/r-I2Fc8hlMI
https://github.com/leahpattison/Sensing-IOT/tree/master/Video

Sensing API code:
https://github.com/leahpattison/Sensing-IOT/tree/master/Data_Collection

Application code:
https://github.com/leahpattison/Sensing-IOT/tree/master/Application

All data recorded available at:
https://docs.google.com/spreadsheets/d/1tdNvPJiN7f1mJFN0sjtZMy7s_MwcuzsVzBkgFAtgKRk/edit?usp=sharing
https://github.com/leahpattison/Sensing-IOT/tree/master/Time_Series/input

Time series analysis:
https://github.com/leahpattison/Sensing-IOT/tree/master/Time_Series

COURSEWORK 1
SENSING

INTRODUCTION & OBJECTIVES

Spotify is a music streaming platform with over 70 million subscribers worldwide, the second most popular service worldwide[1].

Spotify and AccuWeather collaborated to find that on sunnier day, higher-energy music was played and on rainy days lower energy music was played. In the UK in particular, they found that rainy days led to sadder music whilst in London, there was a large boost in dance music when the sun was out [2] .

Within this project, the weather will be tracked in London and the Spotify data from my personal account will be recorded to find if there is a correlation between the type of music listened to and the weather. The data will be stored on an online storage platform. This data will be used to create a webpage which automatically creates playlists for a user based on the weather in their location.

DATA SOURCES & SET UP

Two API's were used to collect data for this project; Spotify and Dark Sky weather. Both API's are based on Representational State Transfer (REST) principles. They use HTTP to GET, PUT, POST and DELETE data that is requested; GET was used within this project to retrieve the data. RESTful API's are more robust and use less bandwidth than its alternative SOAP[3].

SPOTIFY API

The Spotipy python wrapper for the Spotify API was used to request this data [4].

AUTHENTICATION

The Spotify API is based on REST principles which returns data as a JSON object. There is a rate limit per developer on the API but it is unstated. All requests to the Spotify API requires authentication, following the OAuth 2.0 authorization framework. The application must be registered with a Spotify developers account and by sending a valid OAuth access token with each request. This token is accessed through the user logging in when initially launching the application. All client ID information gained from creating a Spotify developers account is stored offline, protecting others' use of the key [5].

Within the first authorization request to the user of the app, a scope for authorization must be defined. The scopes that this application requests to access are:

- User-read-private
- User-read-currently-playing
- User-read-playback-sate
- User-read-recently-played

Any requests outside of this scope will be denied.

DATA COLLECTED

The data recorded for this API included:

- Timestamp
  UNIX start time of song
- Song details
  Artist, genre, song name, album name, album image uri, duration, track ID
- Track features
  Tempo, acousticness, liveness, danceability, speechiness, loudness, energy, instrumentalness

DATA FREQUENCY

This data was collected every three minutes. To meet the nyquist frequency the sampling rate should be 1.5 minutes. However when initialising the data, the spotify API blocked

WEATHER API

The Dark Sky weather API allows access to global past, current and predicted weather data [6].

AUTHENTICATION

The Requests python library was used to call the API which returns data as a JSON. The python JSON encoder and decoder library was used to decrypt the package. A Dark Sky secret key is required to access the API, this is applied to each account. The key is used to encode then later decode the data sent from the API. This must be kept a secret and was therefore uploaded locally onto the Rasberry Pi.

DATA COLLECTED

Sheffield Terrace in Notting Hill, London was used to track the weather with the latitude and longitude of (51.505, -0.196).

The JSON was processed to get the current weather data, with the following information:

- Time date
  Unix timestamp at time of call
- Cloud cover
  Percentage cloud cover, 0-1
- Wind speed
- Temperature
  Temperature in Fahrenheit
- Humidity
  Relative humidity between 0 and 1
- Icon
  Text summary of the data point
- Precipitation Intensity
  Inches of liquid water per hour

DATA FREQUENCY

The API has a limit of 1,000 free calls a day and therefore a call every 1.44 minutes is the maximum achievable frequency. A sample was collected once every 3 minutes in line with the Spotify collection.

DATA COLLECTION & STORAGE PROCESS

DATA COLLECTION

A Rasberry Pi 3 was used to run a script to request data from both API's and then store it. Try Except blocks were used to catch errors caused by the API's and return any issues that arose. The crontab scheduler was used on the Rasberry Pi to run the script at a set frequency of once every 3 minutes and when the Pi rebooted.

DATA STORAGE

All data collected by the API's was stored online on a google spreadsheet [7]. The Gpread API was used to append to an existing sheet.

AUTHENTICATION

To access this API, OAuth2.0 credentials had to be obtained from the Google Developers Console; these credentials were stored offline on the pi as a JSON. The oauth2client library was used to authorize all requests.

Data was backed up locally on the pi using a csv file.

BASIC CHARACTERISTICS

Generalised daily trends can be seen for the two week data collection period for the normalised temperature and humidity in *Figure 1*. The humidity can be seen to have an opposing relationship to temperature daily, but correlates over the whole data set.

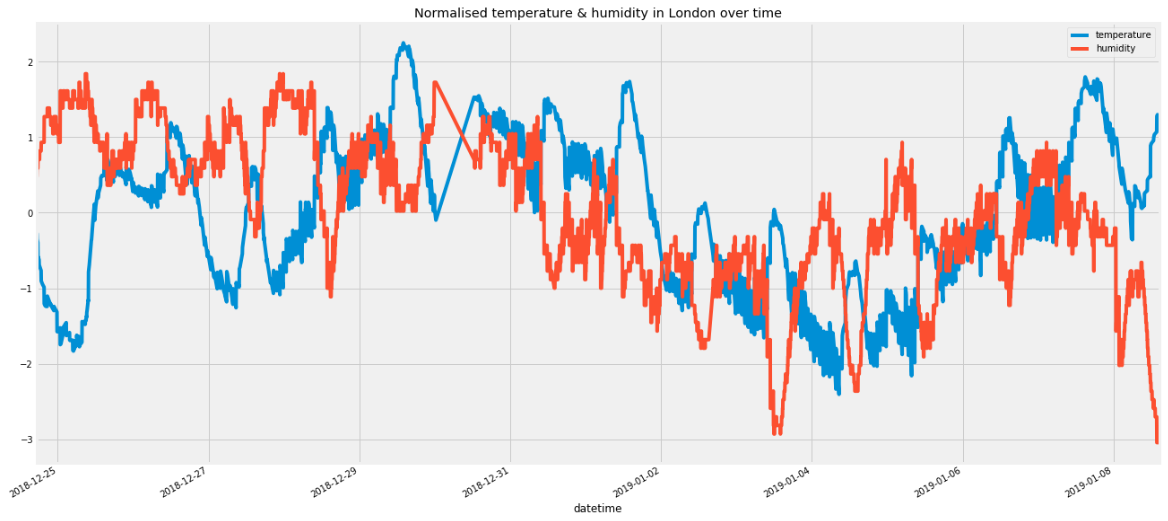


Figure 1 | The normalised temperature and humidity in London over the measured period

Figure 2 indicates the regularity of spotify usage. The audio features of the tracks played did not seem to correlate to time and can be found in the extended jupyter notebook file.

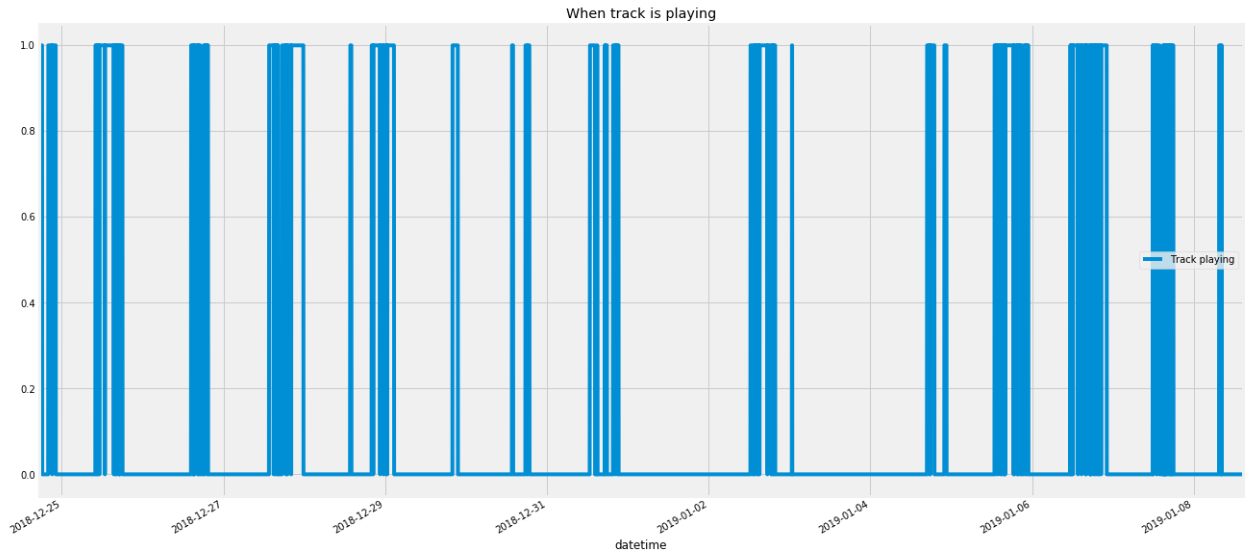


Figure 2 | Whether a spotify track is playing or not

DATA ACTUATION

A web application was created to create playlists for the user based on the weather in their current location. A regression model was created using the data collected in *Coursework 1 : Sensing* to predict the audio features of a song being played based on weather features. The flow of data of the web app can be seen in *Figure 3*.

The application was created using flask; a python microframe work for communicating with html. The backend of the application was python; this dealt with all machine learning algorithms and called the APIs. Python functions are called when an html button is clicked and posts information. A spotify bootstrap was installed to format the html and inline css was added to further add to it.

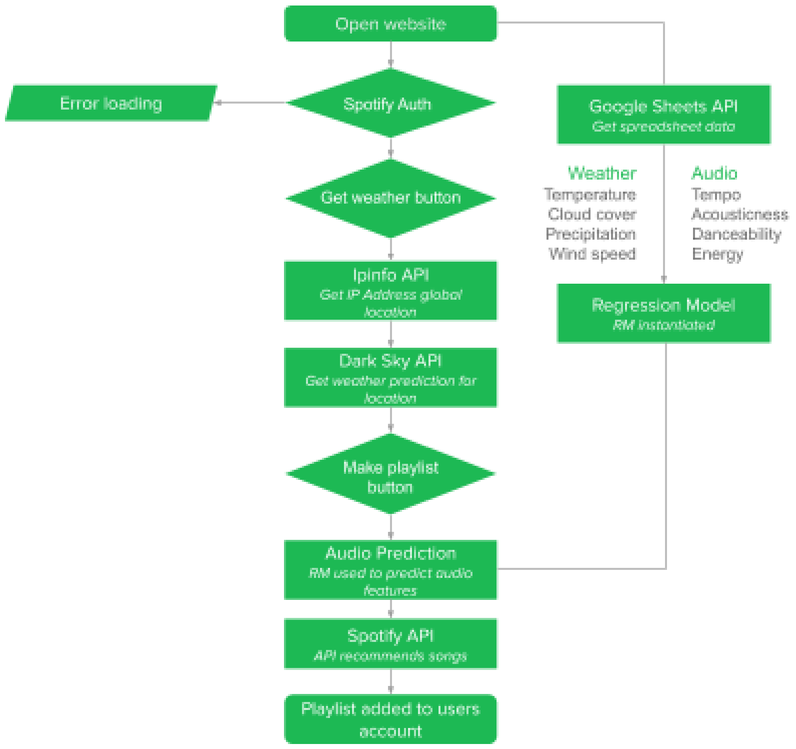
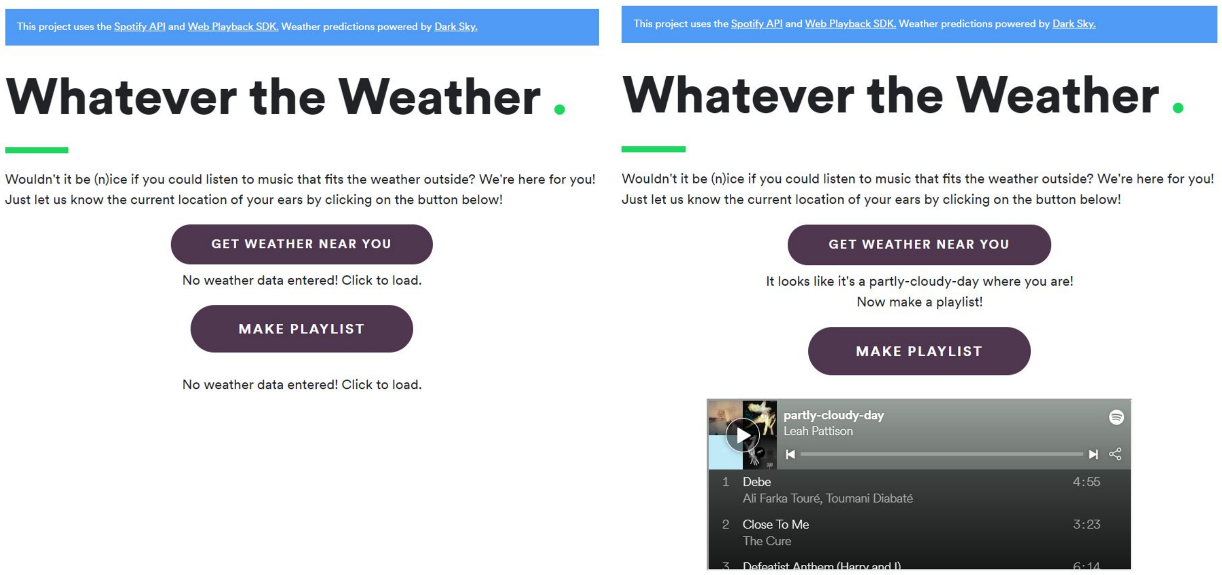


FIGURE 3 | Flow chart for whatever the weather web application

To use the application, the user has to have a Spotify premium account. On entering, the user is prompted to log in which sends an Auth2.0 request to the Spotify API. On loading of the website, the google sheets API is used to load the data collected in *Coursework 1.* A multi-output random forest regressor is then used to fit the data using a regression model.



This project uses the Spotify API and Web Playback SDK. Weather predictions powered by Dark Sky.

Whatever the Weather .

Wouldn't it be (n)ice if you could listen to music that fits the weather outside? We're here for you! Just let us know the current location of your ears by clicking on the button below!

GET WEATHER NEAR YOU

No weather data entered! Click to load.

MAKE PLAYLIST

No weather data entered! Click to load.

This project uses the Spotify API and Web Playback SDK. Weather predictions powered by Dark Sky.

Whatever the Weather .

Wouldn't it be (n)ice if you could listen to music that fits the weather outside? We're here for you! Just let us know the current location of your ears by clicking on the button below!

GET WEATHER NEAR YOU

It looks like it's a partly-cloudy-day where you are!
Now make a playlist!

MAKE PLAYLIST

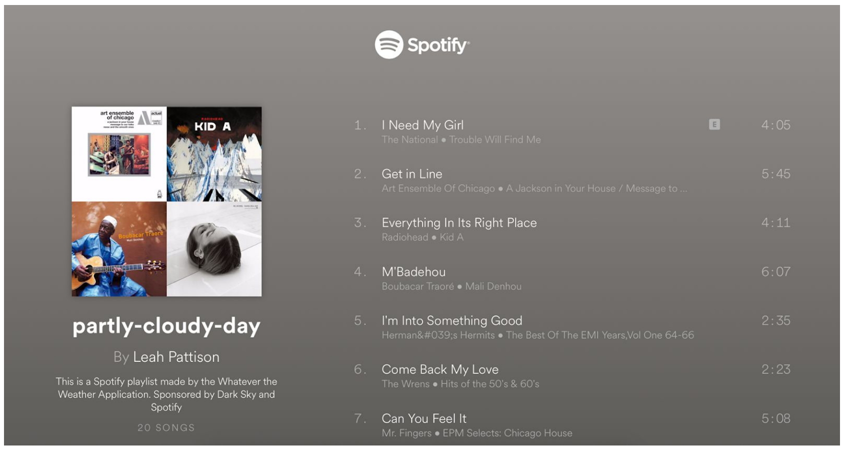
partly-cloudy-day
Leah Pattison

```
1  Debe                                      4:55
   Ali Farka Touré, Toumani Diabaté
2  Close To Me                               3:23
   The Cure
3  Defeatist Anthem (Harry and I)            6:14
```

a. b.

Figure 4 | The interactive application; Whatever the Weather

As shown in *Figure 4* there are two buttons on the interface. The first button uses the IPinfo API [8] ; this finds the location of the IP address of the computer requesting the website. The Dark Sky API is used to get the weather in this location. When the 'Make Playlist' button is clicked, the regression model is used to predict the audio features. These include; tempo, acousticness, liveness, danceability, speechiness, loudness, energy, instrumentalness. The Spotify Recommendations API is then used to get song recommendations based on these values. This is then published to the users spotify account. It is named with respect to the weather summary given by Dark Sky API, as shown in *Figure 5,* and given a description. The spotify API is then called again in an iframe to create an embedded link to the newly created playlist. This playlist can be opened in spotify or played in browser.



partly-cloudy-day

By Leah Pattison

This is a Spotify playlist made by the Whatever the Weather Application. Sponsored by Dark Sky and Spotify

20 SONGS

```
1.  I Need My Girl                                                  E   4:05
    The National • Trouble Will Find Me
2.  Get in Line                                                         5:45
    Art Ensemble Of Chicago • A Jackson in Your House / Message to ...
3.  Everything In Its Right Place                                       4:11
    Radiohead • Kid A
4.  M'Badehou                                                           6:07
    Boubacar Traoré • Mali Denhou
5.  I'm Into Something Good                                             2:35
    Herman&#039;s Hermits • The Best Of The EMI Years,Vol One 64-66
6.  Come Back My Love                                                   2:23
    The Wrens • Hits of the 50's & 60's
7.  Can You Feel It                                                     5:08
    Mr. Fingers • EPM Selects: Chicago House
```

Figure 5 | One of the created playlists' details

DATA ANALYTICS, INFERENCES & INSIGHTS

PROCESSING

Within this, temperature, humidity and tempo are the key features being discussed however the full data analytics can be found within the submitted jupyter notebook file. The Spotify API did not log data when a song was not playing and the Rasberry Pi crashed a few times, thus there were gaps in this data set. The two data sets were combined and given the same time stamp. Data for temperature was linearly interpolated whilst missing data for Spotify was recorded as a none-type. The temperature was resampled to 30 minutes between data points to give a smoother curve.

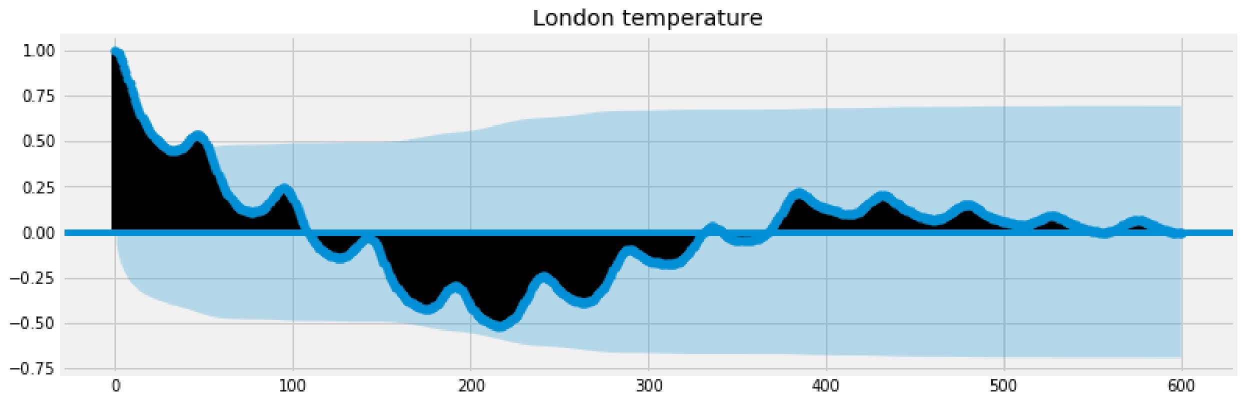


Figure 6 | Seasonal decomposition plots for temperature and tempo

Seasonality trends were observed. As can be seen in *Figure 7*, temperature exhibits a strong seasonality which is representative of the daily temperature cycle. This was also shown through weakly correlated autocorrelation plots. Comparably, the audio features have next to no seasonal correlation.

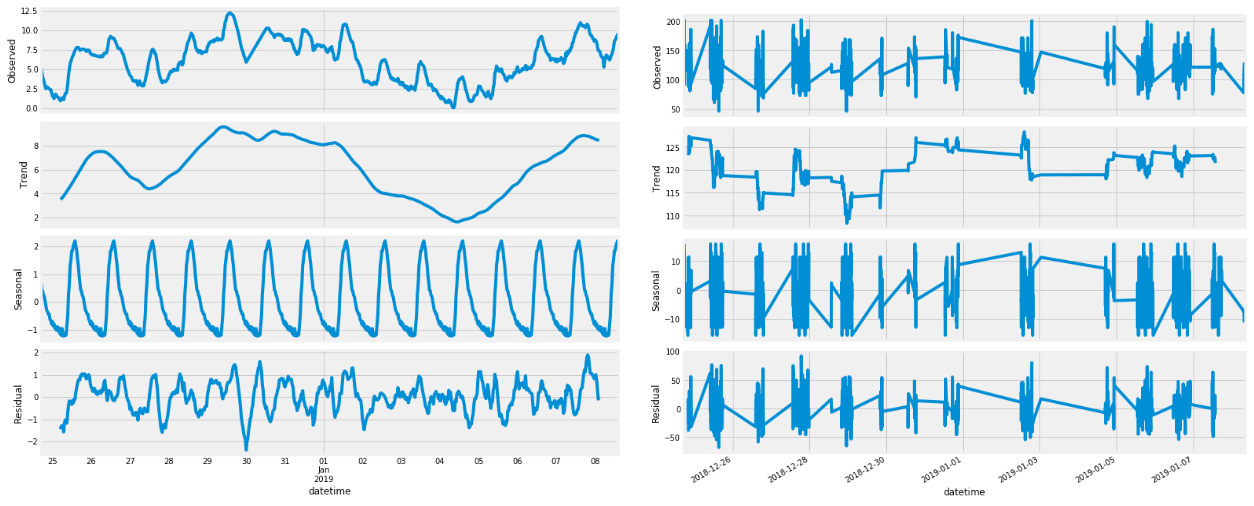


Figure 7 | Seasonal decomposition plots for temperature and tempo

DISCUSSION

It was found that there was no correlation between the audio features of the songs listened to and the weather in London. This can be seen in the cross correlation plots in *Figure 6*. A correlation can be seen between loudness and energy but correlation between some audio features is to be expected. A correlation was also found between humidity and temperature. Looking at the autocorrelation plots, there is a statistically significant correlation for temperature offset to itself by a day.

The random forest regressor used yielded an R2 of 0.02 which is not statistically significant. For the future, the MLP classifier regression could be explored to find less obvious correlations between the data.

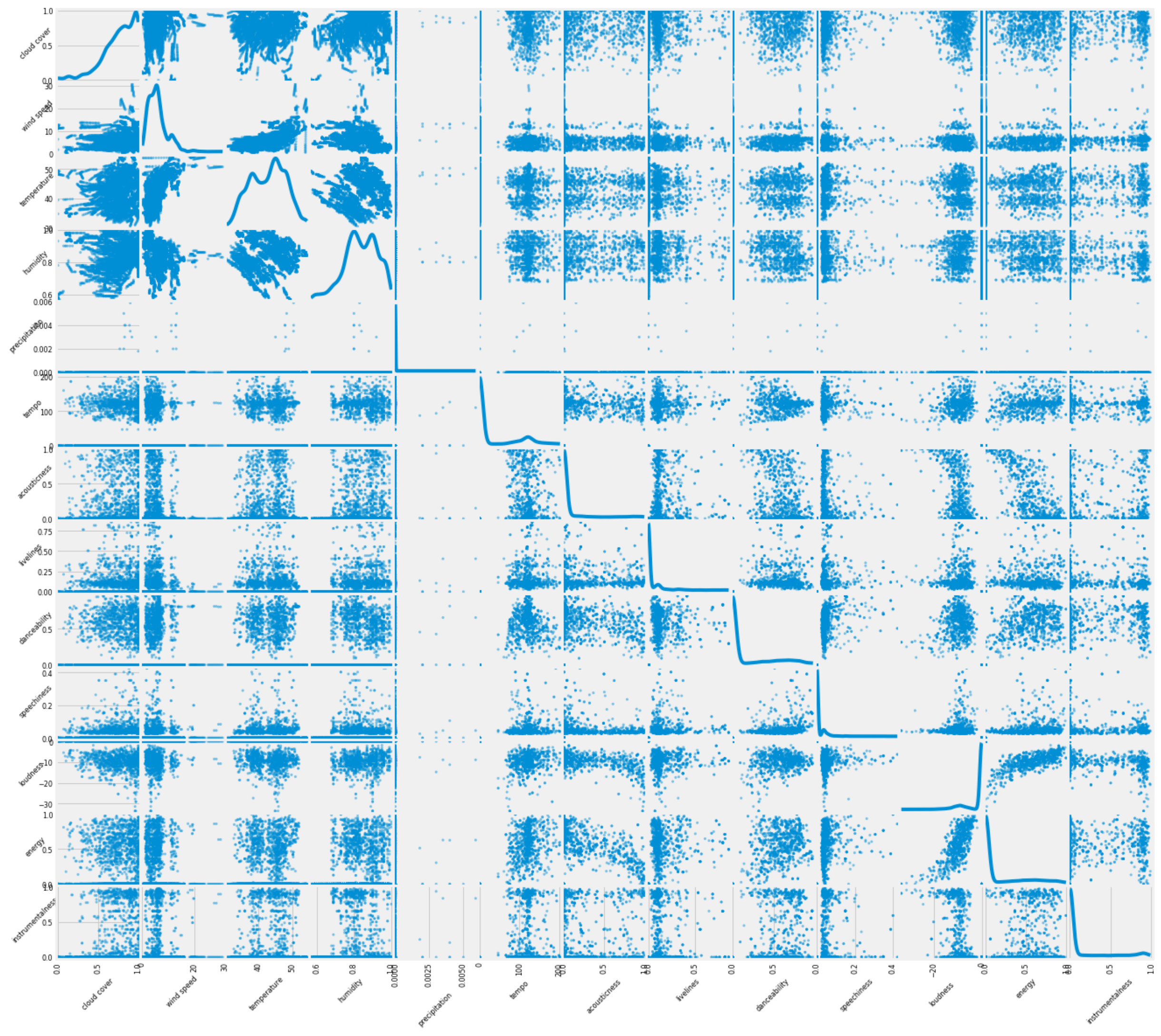


Figure 8 | Scatter plot comparison of all variables from both data sets

FUTURE WORK & IMPACT

It is likely in the coming months that this application will be developed to include some of the below features.

REGRESSION MODEL

The datasets had no correlation between them which meant the R was not significant. This could be due to the small dataset that was collected over only two weeks. Within this time, there was little variation in the weather. The time period over which this was collected could have also brought in some inconsistencies into the data. For example, the festive christmas music contributed to a few days of the Spotify data which is an abnormal listening habit compared to the rest of the year. The rest of the Spotify data set was also specific to a singular Spotify user with their own musical habits which would make the data biased.

Therefore to improve the regression model data should be collected over a longer time period and across multiple users, perhaps in different locations. This could perhaps then yield a correlation between the datasets.

WHATEVER THE WEATHER APP

The application could include more interaction for the user. For instance, choosing the genres that the Spotify Recommendations API chooses from would increase the likelihood that the user liked the playlists created for them. Another option for users could be ‘weather basher’ playlists in which the regression model was trained to output the opposite acoustic song properties correlated to the weather, reflecting the opposite mood.

With the Spotify API this project is easily scalable. Spotify provides authentication methods which allow multiple users to log into the application. The handling of the API calls to spotify would have to be modified to handle more users as Spotify has an API call limit per application per day. If many users were active on the application these could all be sent in one request to the API.

SPOTIFY

There is a large scope for spotify to expand its user specific ‘suggested playlists’ to include weather or location based suggestions relative to the users listening habits. Spotify already has already created a Musical Map which is an interactive map displaying the most popular songs for areas with a high spotify user rate [10]. An extension of this could be a region wide ‘feels like’ playlist to represent the musical mood in each city based on the weather. This could call on data from their previous research with AccuWeather.

REFERENCES

[1] Sanchez, D. (2018). *Apple Music, Not Spotify, Ranks as the Most Popular Music Streaming Service*. [online] Digital Music News. Available at: https://www.digitalmusicnews.com/2018/03/29/verto-analytics-study-apple-music-spotify/ [Accessed 5 Jan. 2019].

[2] Insights. (2017). *Spotify, Accuweather Reveal How Weather Affects Music Listening*. [online] Available at: https://insights.spotify.com/us/2017/02/07/spotify-accuweather-music-and-weather/ [Accessed 5 Jan. 2019].

[3] Rouse, M. (2019). *What is RESTful API? - Definition from WhatIs.com*. [online] SearchMicroservices. Available at: https://searchmicroservices.techtarget.com/definition/RESTful-API [Accessed 8 Jan. 2019].

[4] Spotipy.readthedocs.io. (2017). *Welcome to Spotipy! — spotipy 2.0 documentation*. [online] Available at: https://spotipy.readthedocs.io/en/latest/ [Accessed 1 Dec. 2018].

[5] Developer.spotify.com. (2016). *Web API | Spotify for Developers*. [online] Available at: https://developer.spotify.com/documentation/web-api/ [Accessed 1 Dec. 2018].

[6] Darksky.net. (2019). *Dark Sky*. [online] Available at: https://darksky.net/dev [Accessed 1 Jan. 2019].

[7] Gspread.readthedocs.io. (2019). *API Reference — gspread 3.1.0 documentation*. [online] Available at: https://gspread.readthedocs.io/en/latest/api.html [Accessed 1 Jan. 2019].

[8] Ipinfo.io. (2018). *IP Address API and Data Solutions - geolocation, company, carrier info, type and more - IPinfo IP Address Geolocation API*. [online] Available at: https://ipinfo.io/ [Accessed 10 Jan. 2019].

[9] Sp-bootstrap.global.ssl.fastly.net. (2019). [online] Available at: https://sp-bootstrap.global.ssl.fastly.net/8.0.0/sp-bootstrap.css [Accessed 10 Jan. 2019].

[10] MUSICAL CITIES:Insights. (2018). *Musical Map: Cities of the World*. [online] Available at: https://insights.spotify.com/uk/2015/07/13/musical-map-of-the-world/ [Accessed 7 Jan. 2019].