Spotify Research

- **Size/features in the dataset**
  - ~50 million songs on spotify
  - 
```
{
  "duration_ms" : 255349,
  "key" : 5,
  "mode" : 0,
  "time_signature" : 4,
  "acousticness" : 0.514,
  "danceability" : 0.735,
  "energy" : 0.578,
  "instrumentalness" : 0.0902,
  "liveness" : 0.159,
  "loudness" : -11.840,
  "speechiness" : 0.0461,
  "valence" : 0.624,
  "tempo" : 98.002,
  "id" : "06AKEBrKUckW0KREUWRnvT",
  "uri" : "spotify:track:06AKEBrKUckW0KREUWRnvT",
  "track_href" : "https://api.spotify.com/v1/tracks/06AKEBrKUckW0KREUWRnvT",
  "analysis_url" : "https://api.spotify.com/v1/audio-
analysis/06AKEBrKUckW0KREUWRnvT",
  "type" : "audio_features"
}
```

  - 

| | id | title | first_artist | all_artists | danceability | energy | key | loudness | mode | acousticness | instrumentalness | liveness | valence | tempo | duration_ms | time_signature |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 94 | 5IF6IBPqbclVR7SQKmCIyA | Just Like You | Louis Tomlinson | [Louis Tomlinson] | 0.703 | 0.628 | 0 | -5.914 | 1 | 0.36400 | 0.000000 | 0.384 | 0.471 | 138.032 | 205217 | 4 |
| 95 | 2UCI5rt3PkM9pXtARihaaQ | Bedroom Floor | Liam Payne | [Liam Payne] | 0.625 | 0.684 | 1 | -7.147 | 1 | 0.34200 | 0.000064 | 0.107 | 0.217 | 119.932 | 188234 | 4 |
| 96 | 2kqAtjOtQPAR0OiYUJR43k | Can We Dance | The Vamps | [The Vamps] | 0.640 | 0.820 | 1 | -4.729 | 0 | 0.00312 | 0.000000 | 0.189 | 0.583 | 130.108 | 192711 | 4 |
| 97 | 3wGCtxNjZe3GrZfkojZ1FB | I'm a Mess | Jasmine | [Jasmine] | 0.636 | 0.487 | 11 | -7.123 | 0 | 0.09460 | 0.000000 | 0.111 | 0.647 | 97.022 | 190500 | 4 |
| 98 | 3jnQF0OxLiAEFFcMdBgJ9s | Oh Cecilia (Breaking My Heart) | The Vamps | [The Vamps] | 0.746 | 0.844 | 11 | -5.506 | 1 | 0.03150 | 0.000000 | 0.318 | 0.662 | 100.027 | 196684 | 4 |

  - Spotify api audio features documentation: https://developer.spotify.com/documentation/web-api/reference/tracks/get-audio-features/
  - 18 features listed above for audio features
    - Only 17 usable?
  - Can scrape up to 99 songs at once
    - Create something to scrape 99 songs every x amount of time?
  - In existing blogs, they had “liked” playlist and “disliked” playlist

- **How to scrape/add more data and labels**
  - https://machinelearningknowledge.ai/tutorial-how-to-use-spotipy-api-to-scrape-spotify-data/
  - We will use spotify api to get data, tutorial above ^
  - Also have other features like popularity, explicit, audio analysis about beats, bars, pitch, etc.

    - Spotify api documentation: https://developer.spotify.com/documentation/web-api/reference/tracks/
    - https://towardsdatascience.com/how-to-create-large-music-datasets-using-spotipy-40e7242cc6a6
    -

- **How to train model/target classification**
    - https://medium.com/deep-learning-turkey/build-your-own-spotify-playlist-of-best-playlist-recommendations-fc9ebe92826a
    - https://techxplore.com/news/2019-09-spotify-songs.html
        - Found that random forest created the highest accuracy
    - https://towardsdatascience.com/breaking-spotifys-algorithm-of-music-genre-classification-31ecf8453af1
        - Create own algorithm to classify songs by genre
    - Probably use random forest and augmented data sets to create best model?
    - More complex - try and get waveform info of song and train based on that to see if songs are similar or not?
        - https://benanne.github.io/2014/08/05/spotify-cnns.html
    - *Idea for overall recommendations concept (similar to https://towardsdatascience.com/making-your-own-discover-weekly-f1ac7546fedb )*
        - *Train the model on a personal Spotify user's library with a score of some kind for each song (subjective rating, or frequency in playlists, or # of plays)*
        - *Then have the model predict that score for each song in a randomly selected large test set of Spotify songs (or a large test set of Spotify recommendations)*
        - *Then recommend songs from the test set with optimal score*
    - More accurate: genre classifier
    -
- **Further applications/usefulness of model that we can expand on in blog**
    - Predict next "top hits" based on current popular songs
    - Create custom playlists for a group gathering based on everyone's playlists
    - Create playlist by "mood" - study beats, party beats, customized etc
- **Existing solutions/blogs**
    - https://machinelearningknowledge.ai/mini-ml-project-predicting-song-likeness-from-spotify-playlist/
    - https://medium.com/mlreview/spotify-analyzing-and-predicting-songs-58827a0fa42b
    - https://cs230.stanford.edu/files_winter_2018/projects/6970963.pdf