Predicting The Next Song in a Playlist

Aidan Wall, Noah Masur, Ali Woodward

What we are trying to solve

Our group is aiming to create playlists with newly generated songs. We hope to do this by training a model to recognize patterns within playlists and as a byproduct be able to recommend a song based on previous tracks.

The current solution for this problem:

- The current state of the art solution is the built in recommended playlists that spotify has built in
- Others have tried to do playlist generation, but few have tried by use of deep learning techniques, or trying to analyze order in a playlist to predict the next song's attributes
- Spotify has a built in API function to get a track based on attributes, a function that we will be using in our project to generate songs

Why is it important?

Music is a universal way of bonding and entertainment. While dearly loved apps like Spotify and Apple Music have made discovering music a more pleasant experience than manually searching for songs, understanding the technicalities behind music recommendation and playlist curation is a fun way to better understand how exactly we are getting our music.

As a group, we wanted a better understanding of how we might use deep learning to generate our own music instead of relying on the apps we use daily.

Data

We downloaded 103 playlists from Spotify's API (ensuring that none of the curated playlists were based upon a singular artist). The dataset includes playlist IDs, track IDs, and the following attributes:

Mood: Danceability, Valence, Energy

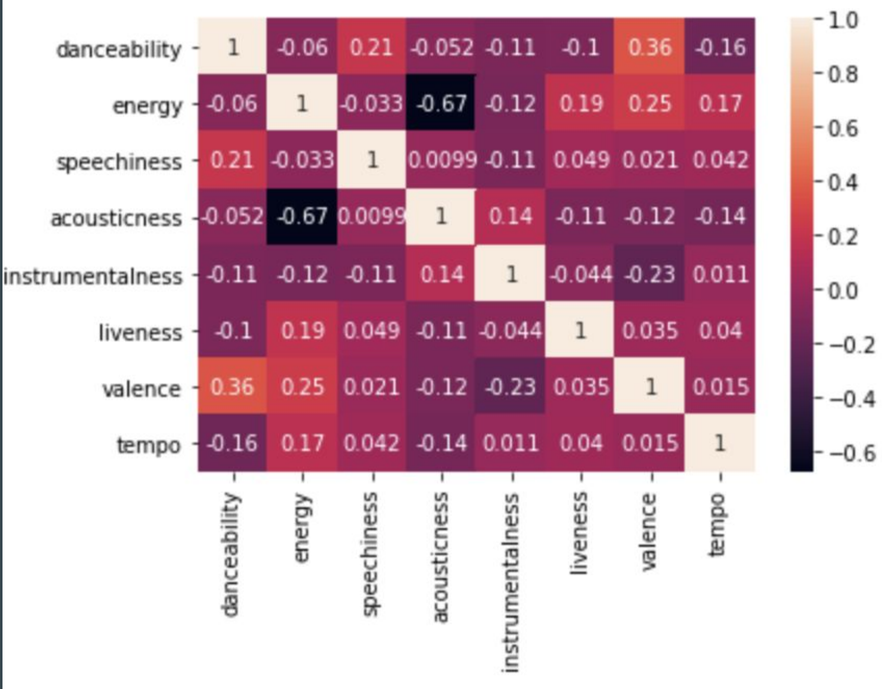
Properties: Loudness, Speechiness, Instrumentalness

Context: Liveness, Acousticness

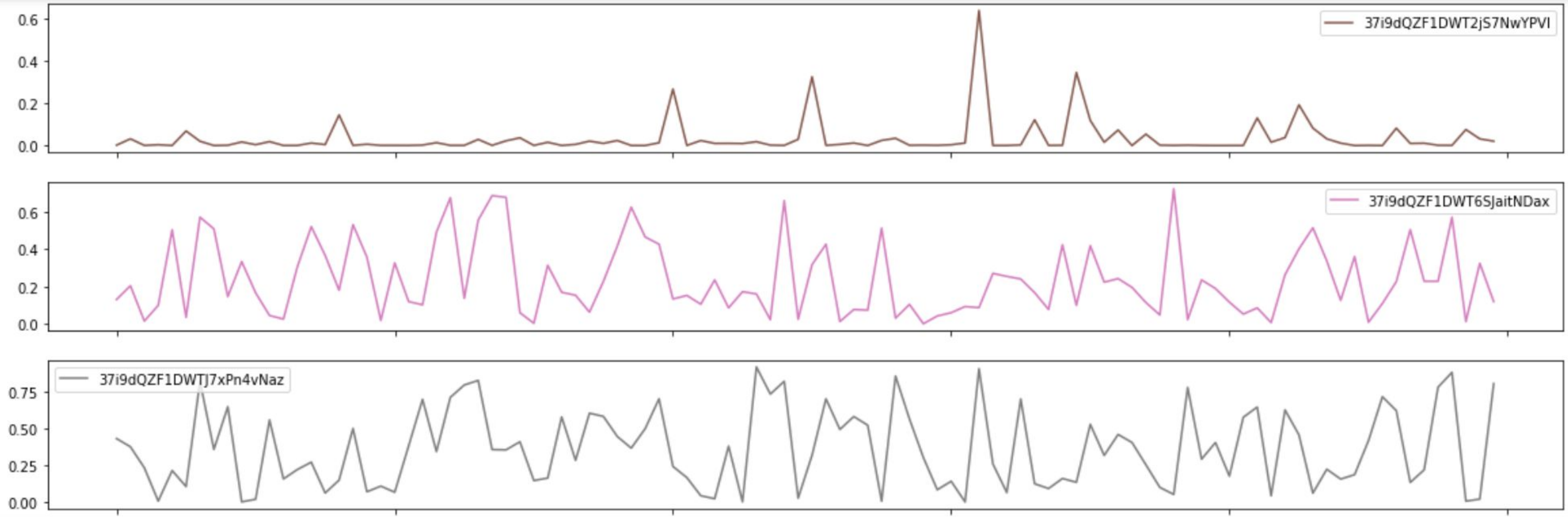
In total, the dataset included 8,025 songs with the length of each playlist ranging from 49 songs - 99 songs

What does an exploratory data analysis tell you about your data?

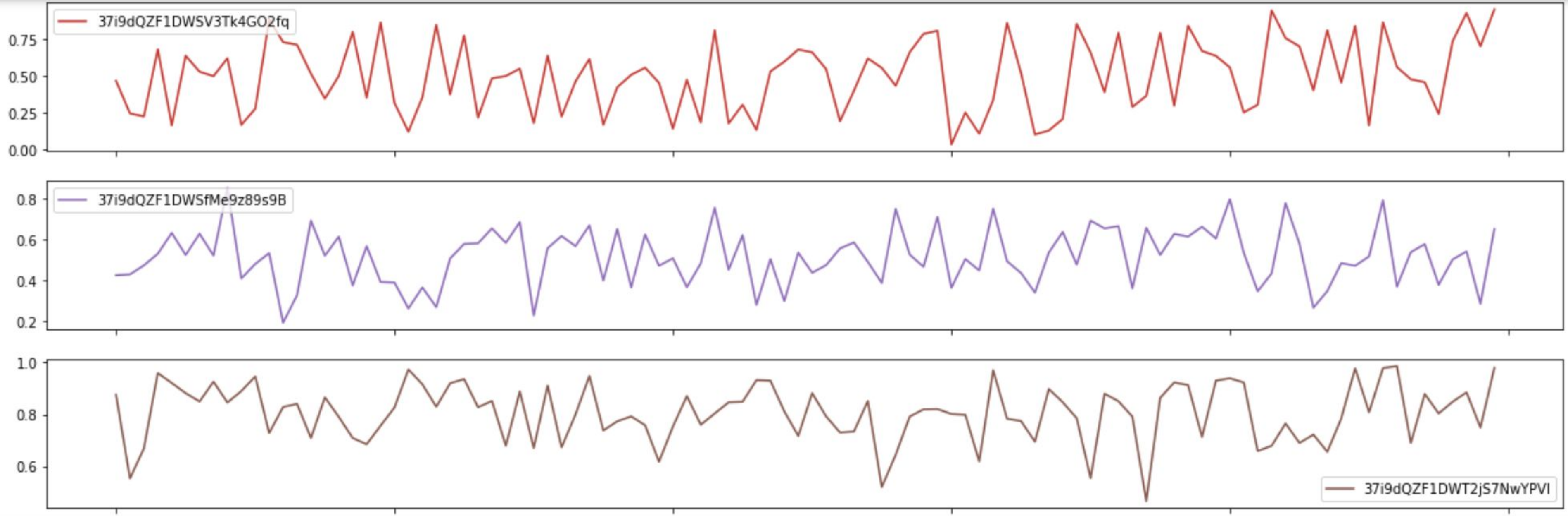
- Exploratory analysis shows us that each playlist across each genre or type of playlist is different.



Accousticness



Energy



37i9dQZF1DWSV3Tk4GO2fq
0.75
0.50
0.25
0.00
37i9dQZF1DWSfMe9z89s9B
0.8
0.6
0.4
0.2
1.0
0.8
0.6
37i9dQZF1DWT2jS7NwYPVI

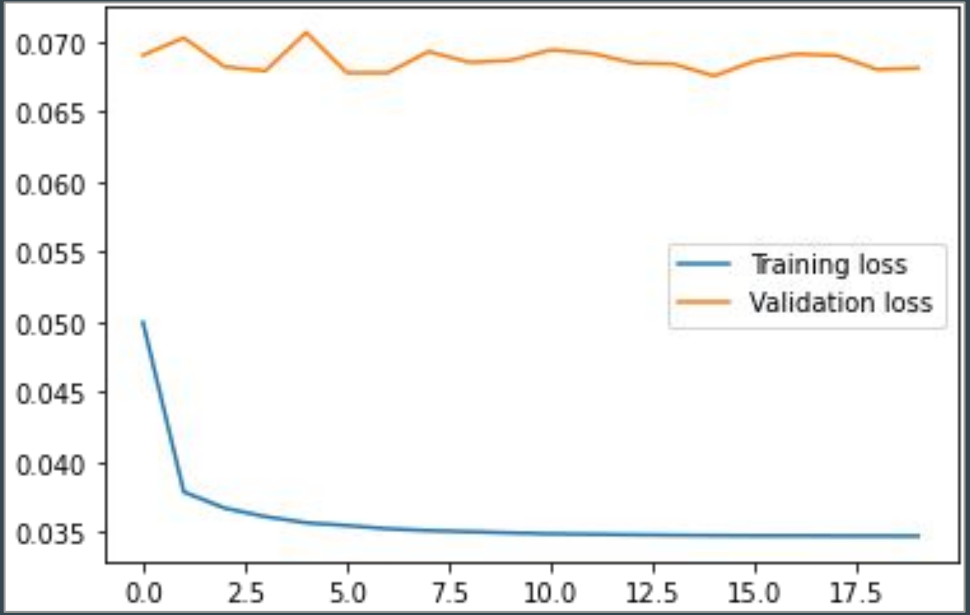
Model

- Data
    - 5 steps back
- Model
    - Layers
        - LSTM layer: 64 nodes
        - Dropout layer: 0.2
        - LSTM layer: 32 nodes
        - Dropout layers: 0.2
    - Parameters
        - Optimizer: Adam
        - Loss: mean squared error
        - Epochs: 20
        - Batch size: 16

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm_12 (LSTM)              (None, 5, 64)             18432

 dropout_10 (Dropout)        (None, 5, 64)             0

 lstm_13 (LSTM)              (None, 32)                12416

 dropout_11 (Dropout)        (None, 32)                0

 dense_2 (Dense)             (None, 7)                 231

=================================================================
Total params: 31,079
Trainable params: 31,079
Non-trainable params: 0
```

Results

- Decrease training loss but not validation loss
- Good at training on the data we have but not on unseen data
    - Overfit
- Prediction from
    - Losing My religion - REM
    - Bigmouth Strikes Again - The Smiths
    - Love Will Tear Us Apart - Joy Divisions
    - Age of consent - New Order
    - This Charming Man - The Smiths
- Results in
    - Where is my Mind - The Pixies



Future steps for improvement

- More powerful computer could train on million playlist dataset
- More hypertuning of parameters
- Actually generate full playlists instead of a couple songs.
- More playlists to train on need to update api commands to refresh token (when downloading auth token would time out after an hour there are more playlists we can grab)

A slide with links to any references you used

https://machinelearningmastery.com/how-to-develop-lstm-models-for-time-series-forecasting/

https://machinelearningmastery.com/multivariate-time-series-forecasting-lstms-keras/

https://www.youtube.com/watch?v=tepxdcepTbY

https://github.com/bnsreenu/python_for_microscopists/blob/master/181_multivariate_timeseries_LSTM_GE.py