Predicting and classifying Emotion in Spotify

Samuel Veliveli, Lucas Lisman, Lexi Van Metre

1

Introduction

Motivation/Background

- Companies and Organizations are very interested in seeing how their products and services are being perceived by their customers

- Emotions are inherently subjective and can be abstract. However, by quantifying a user's perceived emotions, greater analysis can be conducted to improve user experience, while also enhancing the service provided.

- We wanted to see if we could apply this to the field of music
    - Tailoring songs to moods can increase User Engagement
    - Increased User Engagement leads to more profit for the company
    - Positive feedback loop of better service and engagement, and profit.

Data Driven Question

“Can we predict a song's perceived emotion and use that predictive model to recommend a song based on a user’s mood with regards to their preferred genre on spotify?”

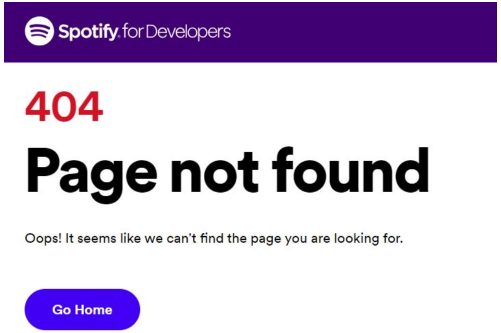
Dataset Background

- Dataset containing information on 114K songs found on Spotify
- Variables include the artist, album, and a variety of different measures (danceability, energy, popularity, speechiness etc)
- Updated 7 months ago

- We found it interesting how this dataset attempted to quantify things such as danceability, energy, and popularity. Things that seem categorical by nature.

Dataset Background cont.

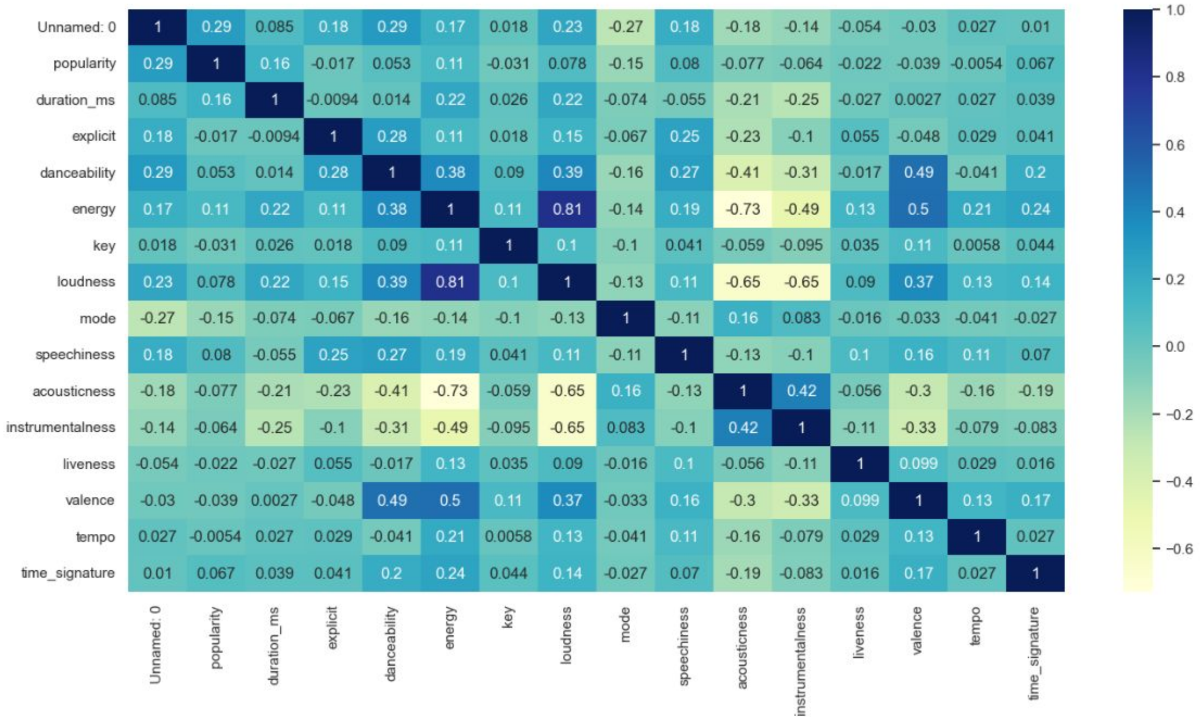
- Valence: A measure from 0.0 to 1.0 describing the musical positiveness conveyed by a track. Tracks with high valence sound more positive (e.g. happy, cheerful, euphoric), while tracks with low valence sound more negative (e.g. sad, depressed, angry)
- Developer's answer: This is an attribute that we compute based on a wide variety of inputs. Generally speaking, we use a set of agreed upon ideas of what happy or positive music sounds like.
  - This was originally developed at The Echo Nest.



2

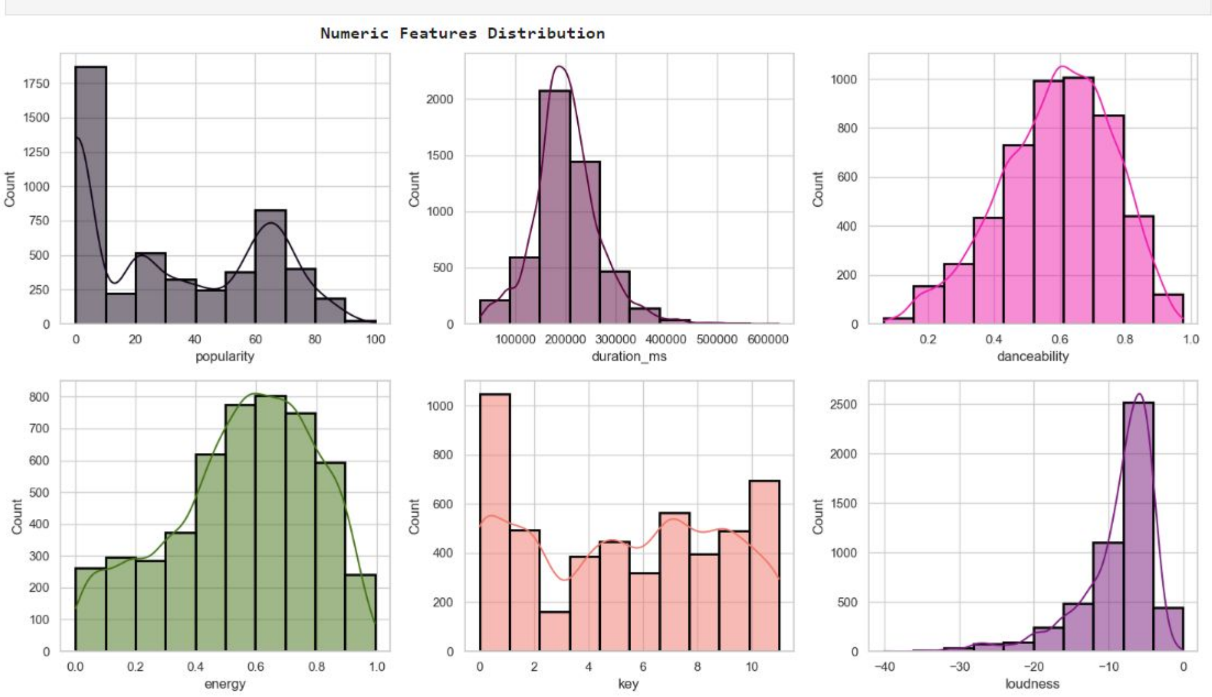
EDA

EDA



Unnamed: 0 1 0.29 0.085 0.18 0.29 0.17 0.018 0.23 -0.27 0.18 -0.18 -0.14 -0.054 -0.03 0.027 0.01
popularity 0.29 1 0.16 -0.017 0.053 0.11 -0.031 0.078 -0.15 0.08 -0.077 -0.064 -0.022 -0.039 -0.0054 0.067
duration_ms 0.085 0.16 1 -0.0094 0.014 0.22 0.026 0.22 -0.074 -0.055 -0.21 -0.25 -0.027 0.0027 0.027 0.039
explicit 0.18 -0.017 -0.0094 1 0.28 0.11 0.018 0.15 -0.067 0.25 -0.23 -0.1 0.055 -0.048 0.029 0.041
danceability 0.29 0.053 0.014 0.28 1 0.38 0.09 0.39 -0.16 0.27 -0.41 -0.31 -0.017 0.49 -0.041 0.2
energy 0.17 0.11 0.22 0.11 0.38 1 0.11 0.81 -0.14 0.19 -0.73 -0.49 0.13 0.5 0.21 0.24
key 0.018 -0.031 0.026 0.018 0.09 0.11 1 0.1 -0.1 0.041 -0.059 -0.095 0.035 0.11 0.0058 0.044
loudness 0.23 0.078 0.22 0.15 0.39 0.81 0.1 1 -0.13 0.11 -0.65 -0.65 0.09 0.37 0.13 0.14
mode -0.27 -0.15 -0.074 -0.067 -0.16 -0.14 -0.1 -0.13 1 -0.11 0.16 0.083 -0.016 -0.033 -0.041 -0.027
speechiness 0.18 0.08 -0.055 0.25 0.27 0.19 0.041 0.11 -0.11 1 -0.13 -0.1 0.1 0.16 0.11 0.07
acousticness -0.18 -0.077 -0.21 -0.23 -0.41 -0.73 -0.059 -0.65 0.16 -0.13 1 0.42 -0.056 -0.3 -0.16 -0.19
instrumentalness -0.14 -0.064 -0.25 -0.1 -0.31 -0.49 -0.095 -0.65 0.083 -0.1 0.42 1 -0.11 -0.33 -0.079 -0.083
liveness -0.054 -0.022 -0.027 0.055 -0.017 0.13 0.035 0.09 -0.016 0.1 -0.056 -0.11 1 0.099 0.029 0.016
valence -0.03 -0.039 0.0027 -0.048 0.49 0.5 0.11 0.37 -0.033 0.16 -0.3 -0.33 0.099 1 0.13 0.17
tempo 0.027 -0.0054 0.027 0.029 -0.041 0.21 0.0058 0.13 -0.041 0.11 -0.16 -0.079 0.029 0.13 1 0.027
time_signature 0.01 0.067 0.039 0.041 0.2 0.24 0.044 0.14 -0.027 0.07 -0.19 -0.083 0.016 0.17 0.027 1
Unnamed: 0
popularity
duration_ms
explicit
danceability
energy
key
loudness
mode
speechiness
acousticness
instrumentalness
liveness
valence
tempo
time_signature
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6

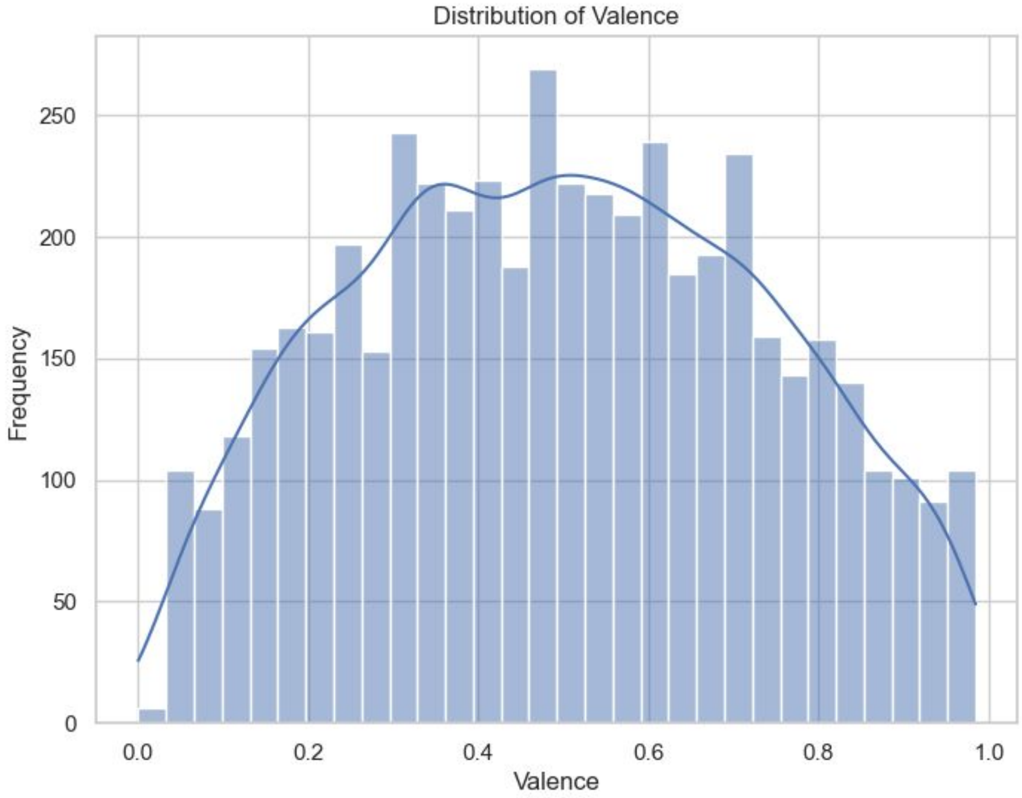
EDA Cont.


Numeric Features Distribution
Count
popularity
duration_ms
danceability
energy
key
loudness

EDA Targeted

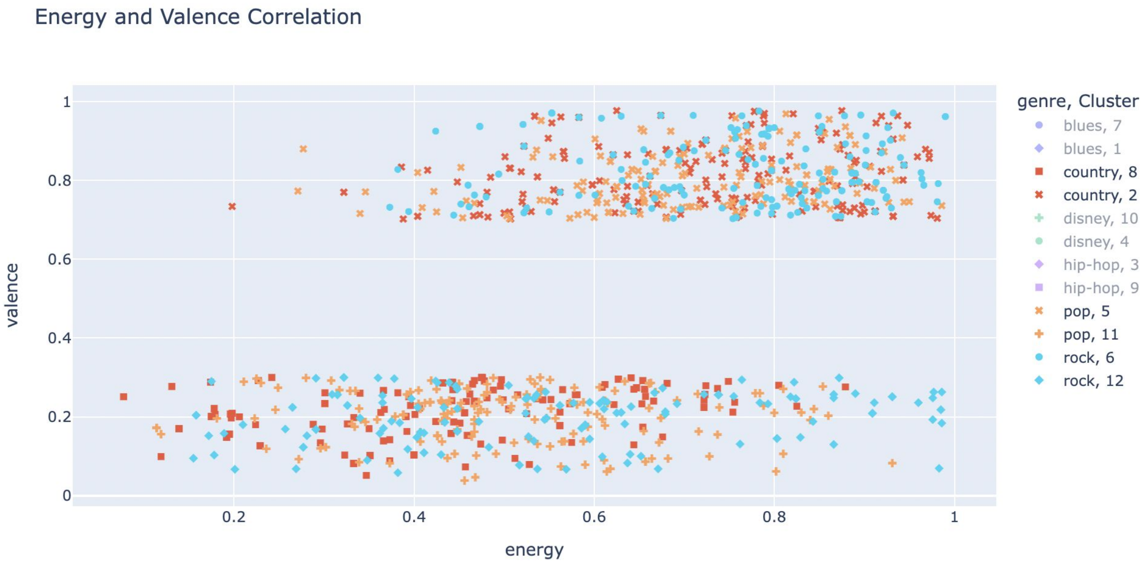
Key Features:

- Relative normal distribution
- Able to clearly find ranges for emotions
- 0.0 - 0.3: negative
- 0.7-1.0: positive



Interactive Plot

Switch over to Jupyter Notebook



3

Methods

Song Recommendation

```python
# TODO: Experiment with inputting different song names!
find_song_recommendation('Die A Happy Man', df)
```

```
Present Without A Bow by Kacey Musgraves;Leon Bridges
Dear Mama by Sidhu Moose Wala
Si Te La Encuentras Por Ahí by Feid
La La La (Reprise) by Arjun Kanungo
Fortunate Son by Creedence Clearwater Revival
```

```python
song_rec_by_mood_genre('Happy', 'pop', df)
```

```
Woman by Doja Cat
All Around The World by Justin Bieber;Ludacris
Munbe Vaa by Naresh Iyer;Shreya Ghoshal
Kannukkulle (From "Sita Ramam (Tamil)") by Vishal Chandrashekhar;Haricharan;Sinduri Vishal
Sol by A.R. Rahman;Rakshita Suresh
```

Demonstration on Jupyter Notebook

Model

Model: Predictive + Classifying algorithm within each genre on whether the song is positive or Negative and output recommendation

- **Turn valence into categorical**
- **Genres Analyzed - Pop, Country, Hip-Hop, Disney, Blues**
- **Have 5 models, one for each genre, and compare accuracy. Have separate models for happy and sad.**
- **Connect predictive algorithm to song recommendation function**
  - **RandomForest for predictive**
  - **Clustering for song recommendation**

4

Evaluation/Results

Model Evaluation

	Genre & Mood	Accuracy	OOB Error Rate
0	Blues (+)	0.873706	0.207039
1	Blues (-)	0.892340	0.109731
2	Country (+)	0.837573	0.201566
3	Country (-)	0.884540	0.142857
4	Disney (+)	0.882143	0.121429
5	Disney (-)	0.860714	0.217857
6	Hip-Hop (+)	0.832347	0.213018
7	Hip-Hop (-)	0.856016	0.143984
8	Pop (+)	0.858316	0.145791
9	Pop (-)	0.876797	0.158111

- All models have satisfactory accuracy (> .80)
  - Blues (-) has the highest (0.89)
- All models have a low out of bag error rate (< .22)
  - Disney (-) has the highest

Model Evaluation

Danceability and **Energy** are two of the (3) most important features for prediction in all of the models

Variability when get past top three (however importance also varies)

Importance

	Genre & Mood	1	2	3
0	Blues (+)	Danceability	Duration	Energy
1	Blues (-)	Danceability	Energy	Speechiness
2	Country (+)	Danceability	Energy	Liveness
3	Country (-)	Energy	Danceability	Loudness
4	Disney (+)	Danceability	Energy	Speechiness
5	Disney (-)	Danceability	Energy	Instrumentalness
6	Hip-Hop (+)	Energy	Danceability	Liveness
7	Hip-Hop (-)	Acousticness	Energy	Danceability
8	Pop (+)	Liveness	Energy	Tempo
9	Pop (-)	Energy	Danceability	Acousticness

Evaluation of Models (Precision)

	Genre & Mood	Precision
0	Blues (+)	0.866667
1	Blues (-)	1.000000
2	Country (+)	0.892857
3	Country (-)	0.945455
4	Disney (+)	1.000000
5	Disney (-)	0.833333
6	Hip-Hop (+)	0.971014
7	Hip-Hop (-)	1.000000
8	Pop (+)	1.000000
9	Pop (-)	0.909091

Precision is the key metric for our model

- All models have high precision (> 0.80)
- Models with a precision of 1.0 did not have any false positive predictions

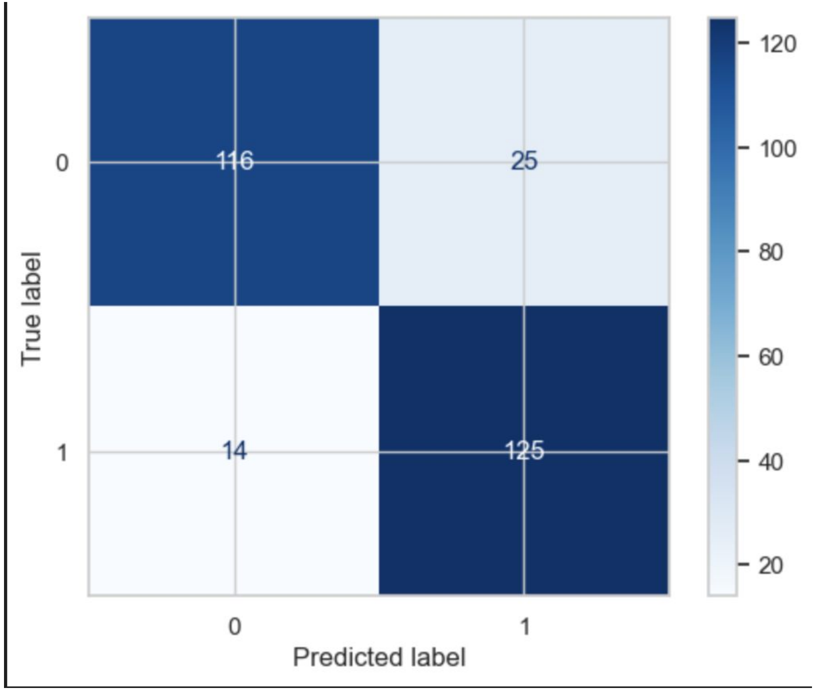
Results of Model (Disney (-))

- TPR: 0.899
- FPR: 0.177

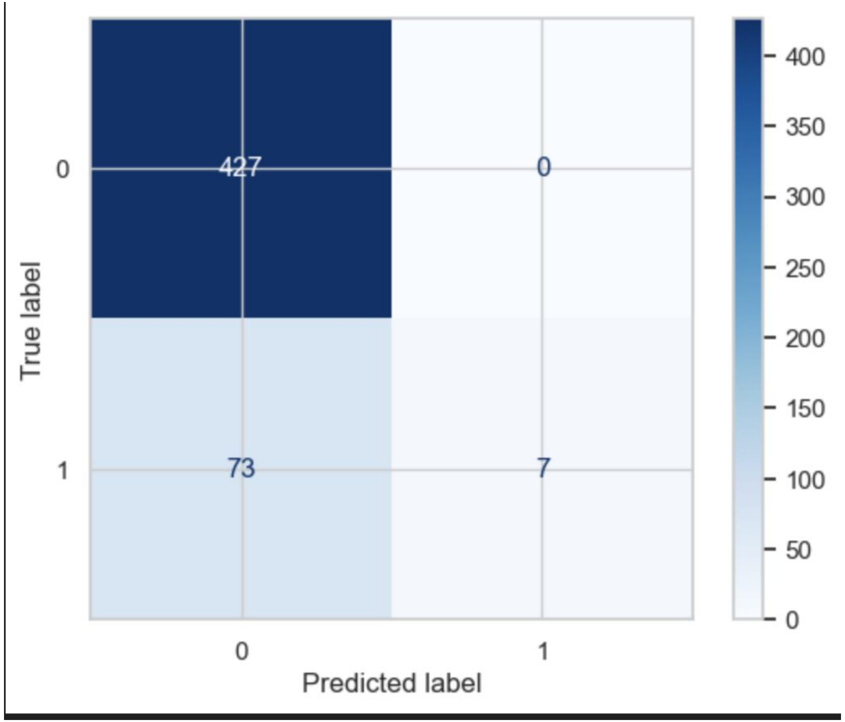
Most points are predicted correctly

- (> 80.0) for both parts of binary

There are however songs that are not associated with negative emotions that are predicted to be negative



Results of Model (Hip-Hop (-))



- TPR: 0.088
- FPR: 0.00

There are 0 false positives

However...

A low true positive rate is not great

5

Conclusion/Future Work

Conclusion

- This project showed a brief method on how we can attempt to quantity categorical things like emotions and conduct analysis on them
- Limitations to the model: we didn’t do the entire emotional scale and we only used five genres and within those genres there were only around a thousand songs.
- A limitation to the song recommendation model is that it only can pull and use songs that were in the dataset. However, the Random Forest model can take a song and its factors and compute its valence in order to determine if the song is positive or negative.
- The model would not perform that well on songs that evoke multiple emotions or on songs that seem to have a very neutral emotional pull.
- High precision shows that you can be confident that a song is associated with positive (or negative) emotions when it is predicted that way

Future work

- **Integrating the Random Forest model with the Song Recommendation Function.** This would allow any music platform to use the machine learning algorithm as it is not limited to one dataset with specific songs. The random forest model would cover all songs.
- **Incorporate temporal dynamics** by considering how the mood of songs or user preferences may change over time. This could involve analyzing trends, seasonal variations, or even user-specific patterns throughout the day, week, or year.
- **Integrate user engagement metrics** into your recommendation system. Track metrics such as how often users skip songs, like songs, the duration of song listens, and user feedback on recommendations. Use this information to continuously refine the recommendation algorithm.
- **Enhance playlist diversity** by incorporating songs from different eras, artists, or sub-genres within the preferred genre. This can prevent playlist fatigue and introduce users to a wider range of music within their preferred mood and genre. It also could incorporate multiple genres within the same mood category to curate a playlist.
- **Allow playlists to evolve over time** based on user interactions and feedback. The system could learn from user preferences, and as they discover and like new songs, the playlist could adapt to incorporate those preferences.
- **Focus on making your machine learning models more interpretable**. Users may appreciate explanations for why a particular song was recommended. Techniques like LIME (Local Interpretable Model-agnostic Explanations) can help provide insights into model predictions.
- **Access information on location of listeners**. This could help curate playlists for different speaking languages or if certain living populations like a certain music genre. Location could also allow for the music curated to be more in touch with the season or particular holiday.