**Week 3 Writeup - EDA Insights**

**Ryan O’Hara, Jack Metzger**

**Dataset Partition Strategy**

Our main priority will be constructing a latent space where similar songs can be grouped for recommendation purposes. Since we don’t have direct feedback data indicating whether a user would like or dislike a given song, we're exploring alternative ways to evaluate the quality of our recommendations. Right now, we’re considering two classification-based strategies to assess how well our latent space reflects meaningful similarities - this is where the data splitting is important.

The first approach would be a 70-15-15 split of the data (train-validation-test) and uses genre tags as labels. One idea is to train a classifier to predict genre, then evaluate whether the recommended songs share genre tags with those in the same cluster. This would tell us whether our system is "genre aware" - whether it groups songs that share characteristics.

The second option is similar but focuses on using the latent coordinates along with the genre tags to train the classifier directly. Either way, the 70-15-15 split gives us enough training data to build a model while still holding out enough for validation and final testing. This strategy helps us measure how well semantic similarity in the latent space aligns with genre-based similarity, without the user preference data.

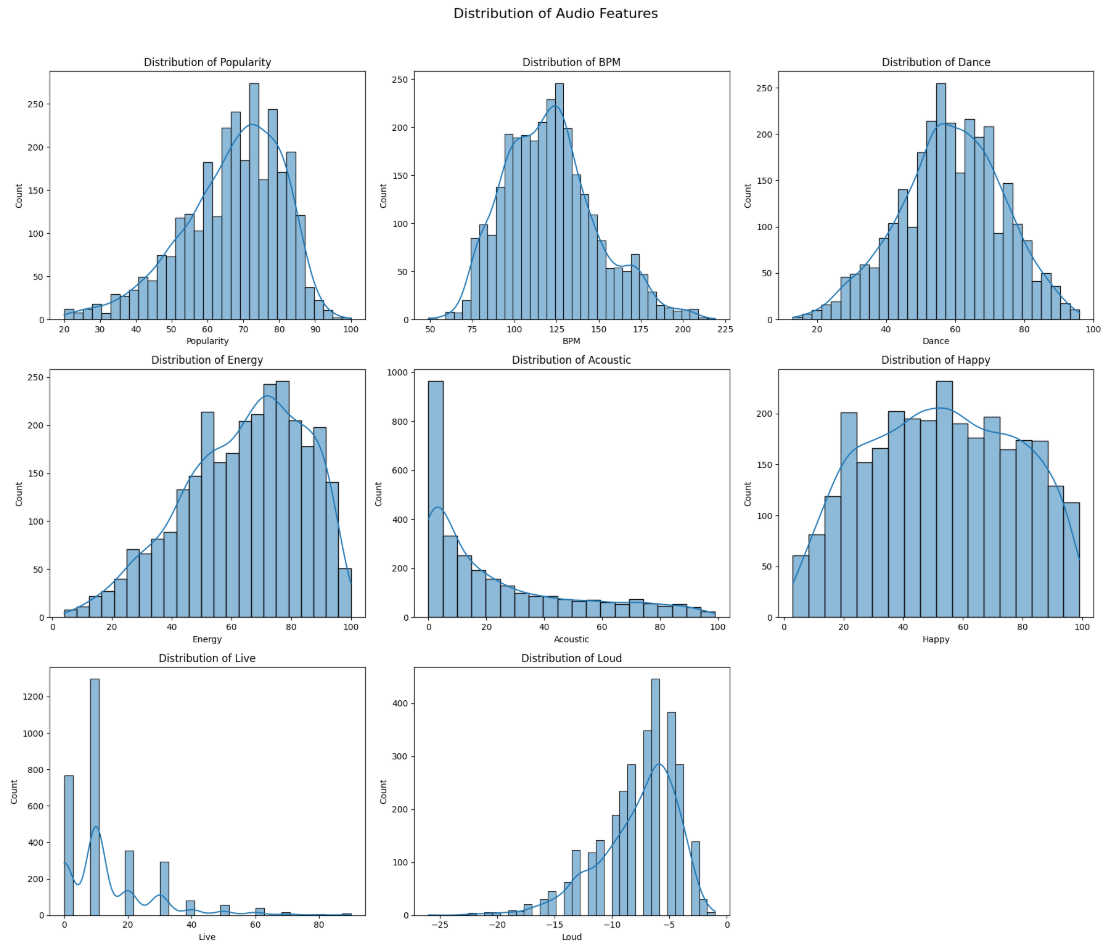
**Data Cleaning**

In addition to the week 2 data cleaning steps, additional steps were taken this week to refine the dataset. First, redundant columns (‘time’, ‘time\_bucket’, ‘Artist\_Clean’, and ‘Song\_Clean’) were dropped from the dataset. The dataset was then filtered to only keep rows with a popularity score greater or equal to 20 (popularity is a metric rated on a scale from 0 -100 and considers the number of plays a track has and how recent those plays were). This was done to keep recommendations in the realm of relevant music. We believe there should be a baseline level of popularity as this will increase the chances a user listens and enjoys the recommendation.

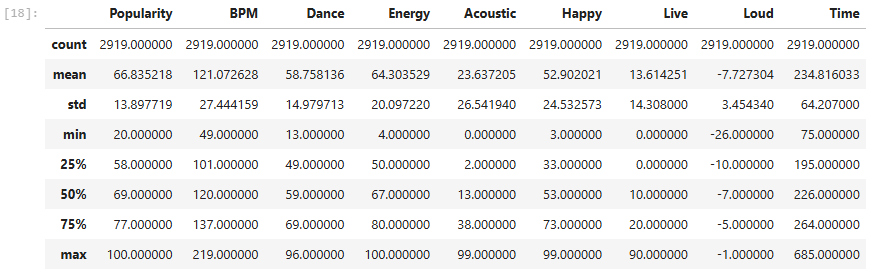
Since each song carried multiple genres, a method was needed to clean up/simplify the genre parameters to better classify each song as well as further differentiate the songs. Each song was narrowed down to 2 genres; a main genre and subgenre. This will allow for better classification and will result in better recommendations. The ‘Genre Tags’ column was split to create a new column for each genre (up to 5 per song) while normalizing text’s symbols, spaces, slashes, etc. Similar/synonyms genre names were standardized to reduce noise and overlapping categories. These included changing all genre “rap” labels to “hip hop”, “new wave” to “80s”, and “classic rock” to “rock”. Next, two genre sets were defined; major genres and secondary genres. Major genres comprises the traditional music genres (rock, pop, hip hop, etc) where secondary contains all the major genres as well as eras and substyles. A function was designed to first look for compound genres in the first/primary genre column (i.e. ‘Indie Rock’). If a compound genre was found it was split and used as the primary and secondary genre (i.e. Primary: Indie, Secondary: Rock). If a compound genre is not found, the genre one becomes the primary and genre two becomes the secondary.

**EDA Types + Visuals**

*Distributions and Summary Statistics*

****

**Figure 1:** Distribution histograms for each of the 8 numeric audio features (popularity, BPM, dance, energy, acoustic, happy, live, loud).



**Table 1**: Descriptive summary statistics for the Spotify song recommendation dataset.

After eliminating primary and secondary genre combinations with less than 5 occurrences, the final dataset contained 2919 songs. Figure 1 shows the distribution for each audio feature in the dataset with the descriptive statistics for each listed in table 1.

Popularity: Songs in the dataset follow a left skewed unimodal distribution in relation to popularity with a mean score of 66.8/100 and a median score of 69.0. The middle 50% of the data (IQR) falls between 58 - 77. Songs with a popularity score of less than 20 were removed from the dataset, reducing the intensity of the negative skew in the data.

BPM: BPM follows a unimodal right skewed distribution with a mean of 121.07 and a median of 121. The minimum is 49 and maximum is 219, giving a wide range of 170. The IQR is 101 to 137.

Dance: Dance is measured on a scale from 0-100. It follows a relatively normal distribution and is by far the most normal of all the audio features. The mean and median dance scores are 58.76 and 59 respectively. The minimum is 13 and the maximum is 96, giving a range of 83. The IQR is 49 to 69.

Energy: Energy is measured on a scale from 0-100. It follows a left skewed unimodal distribution with a mean of 64.30 and a median of 67. The minimum is 4 and the maximum is 100, giving a wide range of 96. The IQR is 50 to 80.

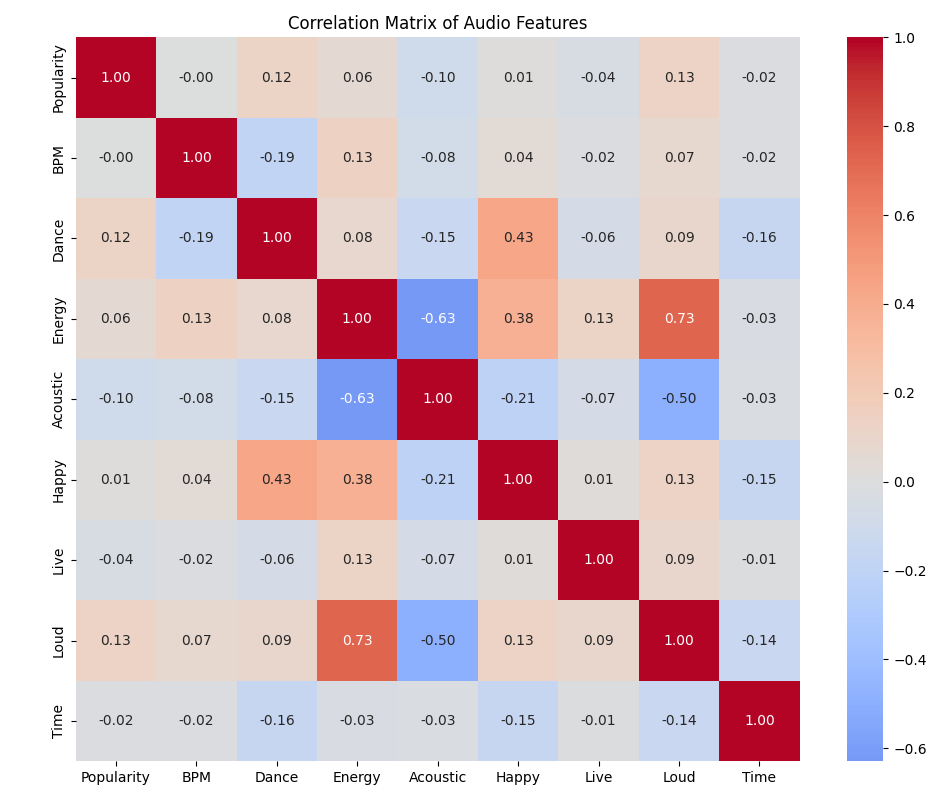
Acoustic: Acoustic is measured as confidence on a scale from 0-100, with 100 being very high confidence the track is acoustic. It follows a very strong right skew. The mean is 23.6 with a median of 13. There is a range of 99 (0 min and 99 max). The IQR is 2 to 38. The distribution peak is heavily centered between 0 and 5, meaning many of the songs have a very low acoustic confidence level. This variable should be considered for transformation into a binary predictor (acoustic or not acoustic) based on a calculated decision threshold.

Happy: Happy is measured on a scale from 0 to 100. It follows a normal distribution showing characteristics of uniform. The middle of the data follows a relatively uniform distribution with tails on either end. This distribution has a mean of 52.9 and median of 53. The range is 96 with a minimum of 3 and maximum of 99. The IQR is 33 to 73.

Live: Live is measured on a confidence scale of 0 to 100, with 100 being very high confidence. A score above 80 strongly suggests the performance was live. The distribution of live is unimodal (at 10) and heavily skewed right. This makes sense as most music available on streaming sites are recorded in the studio. The distribution has a mean of 13.6 and a median of 10. The range is from 0 to 90 with an IQR of 0 to 20. This feature is a good candidate for transformation into a binary feature (is live or is not live) based on the threshold score of 80.

Loud: Loud is measured as the average decibels of a track across its whole duration. It follows a left skewed unimodal distribution. The distribution has a mean of -7.72 and a median of -7. The range is -26 to -1 with an IQR of -10 to -5. This feature is a good candidate to normalize from 0 to 1 in order to eliminate the negative values from the dataset.

*Feature Correlation Analysis*



**Figure 2:** Correlation matrix of all audio features. Red signals a positive correlation, blue signals a negative correlation, and gray signals no correlation.

The following popular correlation coefficient thresholds are used to evaluate relationships in absolute value terms ([geekforgeeks.com](http://geekforgeeks.com)).

No relationship: 0 to +- .25

Weak relationship: +- 0.25 to +- 0.50

Moderate relationship: +- .50 to +- .75

Strong relationship: > +- .75

As seen in figure 2, most audio features show no or weak correlation with a few showing stronger relationships.

Positive Correlation:

1. **Loud and Energy (+0.73)** : This relationship is intuitive/expected. Generally the louder a song is, the more energy it is said to have.
2. **Happy and Dance (+0.43)** : This relationship is not a surprise as it is expected that happier songs will make the listener more likely to dance. Sad songs are not always the most danceable, as dancing can be seen as an expression of happiness and energy.
3. **Happy and Energy (+0.38)**: Similar to the happy/dance relationship, the more energy a song has the more likely it is to make the listener feel happy.

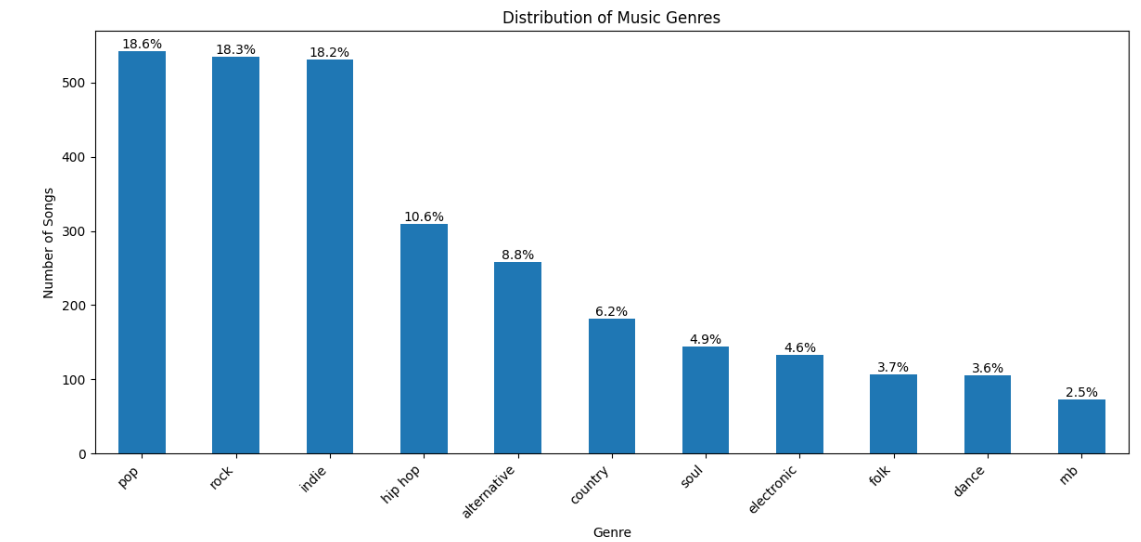
Negative Correlation:

1. **Acoustic and Energy (- 0.63)**: This was not a surprise as acoustic songs are generally softer and do not have added electrical amplification (Merriam-Webster).
2. **Acoustic and Loud (- 0.50)**: This was not a surprise for the same reasons as the acoustic/energy relationship.

Surprising No Correlation:

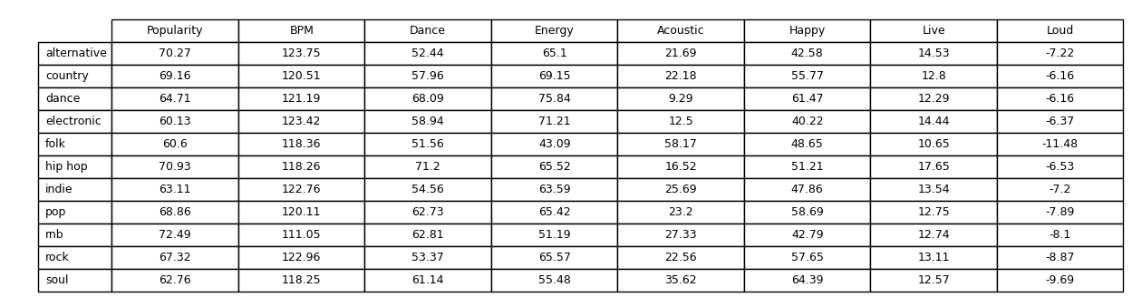
1. **Energy and Dance (0.08)**: Interestingly, energy and dance showed no relationship. This is surprising as happy showed a positive correlation with both energy and dance. The assumption was energy and dance to therefore show a positive relationship as more energy could cause someone to be more likely to dance (thinking about club music). However, this assumption does not hold true in the data and there is no correlation between energy and dance.

*Genre*

****

**Figure 3:** Bar chart showing the distribution of the different primary genres in the dataset.

The songs in the dataset are generally dominated by 3 genres (pop, rock, and indie) accounting for 55.1% of the data. The genre with the lowest allocation of songs is mb with 2.5%. There is enough representation in each group to make a model. There are about 73 songs in the mb category in all the data, sufficient to make diverse recommendations for that genre. It should be noted that the left skew in genre distribution may create a general bias towards recommending songs in the more popular genres (pop, rock, indie, hip hop). However, this bias may not be a large concern, as the distribution above is representative of a random sampling of songs in pop culture. This weighted representation of genres may improve our recommendation system's accuracy, as it better reflects how genres are actually distributed in pop culture and the music industry, rather than treating all genres as equally popular. In other words; by maintaining the natural weight of genres in our dataset, our model can make more relevant recommendations that reflect real-world music popularity patterns, rather than assuming all genres have equal representation.



**Table 2:** Mean audio features by genre.

**EDA Insights**

The EDA provided insights into both the structure of the dataset and the challenges and opportunities for building a recommendation system. The analysis revealed a mix of skewed and normal distributions across audio features, which gave insight into some preprocessing steps we will have to conduct such as normalization or transformation of skewed variables like acoustic, live, and loud. The weak to moderate correlations between features like happy with dance and energy point to potential clusters of emotional or mood-based song attributes that could improve recommendations. Surprisingly, some expected relationships like energy and dance showed no meaningful correlation. The genre distribution, while imbalanced, actually reflects real world music consumption trends, which may help the model deliver more practical recommendations aligned with user expectations. One challenge is identifying if any sub genres are acting as proxies for primary genres (and how this might affect the model). Overall, the EDA highlighted key preprocessing steps and modeling considerations that will be important in solving the problem of generating accurate and personalized music recommendations.

**Data Problems**

1. Variety of genre tags

When songs receive multiple genre tags, they present issues with ambiguity, complexity, and scalability... Overlapping genre borders are intricate for traditional databases to handle, leading to uneven classification. The accuracy of genre tagging can be enhanced by incorporating user input and AI-powered models.

2. Unbalanced popularity

Music popularity data is frequently skewed due to algorithmic biases and mainstream impact and engagement measurements that favor well-known artistsThis distortion impacts fairness in rankings, recommendations, and discovery.n. Achieving representational balance can be facilitated by improved weighing models and diversified exposure.

3. Genre Labeling Inconsistency

Subjective interpretations, changing musical styles, and platform-specific classification techniques are the causes of inconsistent genre designation. Mismatches in searchability and recommendations result from this heterogeneity. Consistency can be increased by implementing adaptive tagging models and uniform criteria.

4. Creating Binary Variables for ‘Acoustic’ and ‘Live’

Certain audio features like ‘acoustic’ and ‘live’ could be more effective if transformed into binary features to better reflect their real world interpretation. In the real world a song is either acoustic or not acoustic and recorded live or recorded in the studio. Transforming these features into binary would help with interpretation, give clearer decision boundaries within the model in relation to these features, reduce noise, reduce dimensionality, and save computational power.

For ‘live’ a threshold of > 80 is previously mentioned as being the cut off for an assumption the song is a live performance.

For ‘acoustic’ a similar threshold would make sense as the distribution follows a similar heavy right skew pattern as ‘live’. Other methods can be explored to calculate a threshold based on the proportion of live songs in the Spotify library.

5. Normalizing Variables

Normalizing all of the predictor features to a scale of 0 to 1 will be a benefit to the model. Not all the features follow the same scale. Omitting the two potential binary variables (acoustic and live), 4 of the remaining 6 features (popularity, dance, energy, happy) lie on a scale of 0 to 100.

BMP falls on a scale of 49 to 219 in the dataset. This is a very arbitrary scale with many observations having the potential to be considerably larger in magnitude than features on the 0 to 100 scale. This will introduce unintended weight bias in the model. This holds especially true in deep learning models, where standardizing variables is a standard practice and an essential part of data processing.

Loud falls on a scale of -26 to -1 (measuring in decibels). This scale is also arbitrary and will introduce weight bias for the same reasons as BMP. In addition, the negative values will negatively affect the model.

Overall the benefits of standardizing features in the dataset before model training will help model performance. It will help prevent unintended weight bias, improve convergence speed, save computation power, and will be essential in developing a deep learning model.