**Week 13 Writeup - Final**

**Ryan O’Hara, Jack Metzger**

**Project Overview**

With the amount of music being released each week, people tend to get stuck listening to the same songs or artists over and over due to the amount of time it takes to explore and find similar music that matches what you like. Different music platforms like Apple and Spotify do a good job with recommendations, but they often prioritize matching genres or suggest what other users are listening to. This typically leads to missing out on music that matches the style or meaning behind a song that you love.

For this project, we wanted to take a more creative and personalized approach - the goal is to develop a recommendation system that suggests songs based on their semantic similarity to a given track. The idea is to capture how songs feel similar - not just because they’re both “pop” or “indie”, but because they share similar lyrics, tempo, emotion, or energy. For example, if someone’s listening to a slow breakup song, we can recommend others with that same lyrical theme and musical type - even if the artist or genre is different.

This kind of model could make music discovery more engaging, relevant, and personalized. Instead of relying on popularity or genre, listeners get recommendations that truly match the feel of what they like at the time. It can also help discovery of less popular songs and artists that would otherwise be overlooked, which gives platforms a way to boost discovery and user satisfaction at the same time!

With an estimated yearly impact of $325 million, the system's emphasis on lyrics and metadata-driven similarity improves user pleasure and encourages the discovery of lesser-known artists and generates economic value for streaming platforms through greater engagement and retention.

**Data Exploration**

We selected a high-quality dataset of 4,095 English-language songs from 17 public Spotify playlists, enhanced with audio characteristics, metadata, and complete lyrics, to assist our semantic similarity-based recommendation engine. We then cleaned and refined it to ensure quality and consistency. After removing duplicates, remixes, and non-English tracks, as well as scraping accurate track-level genre tags from MusicBrainz and full lyrics from Genius, the dataset was nearly halved in size - leaving us with the 4K songs. Additional deep cleaning removed remastered versions, near-duplicates, and tracks missing key features.

The lyrics are essential to embedding and clustering algorithms since they offer significant variance in sentiment and narrative tone, as sourced from [Genius.com](http://genius.com). Advanced feature engineering and unsupervised modeling are made possible by this dataset, which provides a strong basis for building a latent space that captures both musical and thematic similarities.

**EDA**

Every track has comprehensive lyrical and auditory elements, guaranteeing analysis that is both thorough and accurate. Recommendation models that accommodate casual listeners, genre aficionados, and those looking for more experimental or ambient sounds are made possible by the selection's harmony between lively, high-energy music and slower, moodier pieces.

The collection reflects modern production standards, with most tracks professionally recorded and mastered, leaning slightly toward digitally produced content in line with streaming trends. Emotional tones range from joyful to introspective, supported by lyric data that adds thematic depth for mood or story driven recommendations. Consistent structural features like track length and volume help maintain modeling stability, while the diversity in rhythm, texture, and energy makes the dataset adaptable for personalized recommendations across varied listening contexts.

**A group of blue and white graphs

AI-generated content may be incorrect.**

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

A screenshot of a graph

AI-generated content may be incorrect.

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

A screenshot of a computer screen

AI-generated content may be incorrect.

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

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.

We did our exploratory data analysis, concentrating on audio feature distributions, genre representation, and inter-feature correlations pertinent to semantic similarity on 2,919 songs.

We split the data into 70-15-15 for training, validation, and testing. In order to evaluate musical coherence through supervised genre categorization, we employ genre labels as a stand-in for assessing latent space quality.

The variables like loudness, liveness, and acousticness are important factors that can cause normalization, which have strong correlations, such as energy, loudness, and acousticness, which provide feature engineering with information. In line with actual consumption patterns, genre analysis showed a natural imbalance favoring pop, rock, and indie.

In addition to giving influence on data preprocessing such as feature selection and categorical normalization, these results confirm the dataset's diversity and structural integrity. They also offer the framework for embedding and clustering models that capture nuanced emotional and musical similarity.

**Data Preprocessing**

We put in place a thorough preparation pipeline that addressed redundancy, scale variance, feature relevance, and data leakage to prepare the dataset for semantic similarity modeling. To designate primary and subgenres and eliminate uncommon subgenres to guarantee modeling validity, genre tags were made simpler. To counteract scale imbalance and stop any feature from unduly affecting similarity calculations, numerical features, including danceability, loudness, and BPM, were min-max normalized. Before scaling, the train-validation-test split was carried out to maintain model integrity, and genre information was improved to prevent semantic leaking. To guarantee that all features represent inherent musical and poetic qualities, popularity was carefully included, and no user-derived data was used.

As a result, the dataset was clear, comprehensible, and objective, making it ideal for content-based recommendation.

**Feature Engineering**

Categorical variables are defined as “data [that] represents discrete values or labels that fall into distinct categories or groups” ([medium.com](http://medium.com), Chandima Jayamina). Categorical variables can be nominal, ordinal, or binary. Nominal variables have no rank or order (genre/subgenres). Ordinal variables have an ordered structure (camelot). Binary variables contain two states (yes or no/ present or not present). Traditional machine learning methods require numerical data, making categorical variables in their raw form unsuitable for ML models. The categorical variables in our dataset, ‘camelot’, ‘genre’, and ‘subgenre’, needed to be encoded in order to be used in a machine learning model. The concept of one-hot encoding works to transform categorical data into multidimensional binary vectors, enabling the features to be used in mathematical machine learning models.

*Genre and Subgenre (Nominal)* - Genre and subgenre were one-hot encoded using standard one-hot encoding. The goal was to transform each unique genre or subgenre into a separate binary column. Sklearn’s OneHotEncoder package was used to achieve this.

After applying one-hot encoding, each of the 11 primary genres (such as rock, pop, hip-hop, etc.) and their 42 associated subgenres are represented as separate binary columns in the dataset. To ensure statistical reliability and balanced representation across categories, each genre and subgenre included has a minimum of 15 songs.

*Camelot* - Camelot is defined as “a term that refers to the concept of key signatures, chord progressions, and harmonic relationships between notes and chords… In essence, Camelot is a mathematical concept that helps musicians understand the relationships between different keys, chords, and scales. It’s based on the idea that certain notes and chords have a special affinity with each other” ([clrn.org](http://clrn.org)).

The Camelot Wheel is a tool for DJs to understand and apply harmonic mixing (dj.studio). Our data labels camelot on a key scale of 1-12 paired with a mode (A or B), giving 24 total combinations. Although ordinal, camelot follows a circular scale, not a linear one. This means key 12B is adjacent to 1A. Using a linear ordinal system, the distance between 12B and 1A would be large where in reality they are just as close as 1A and 2A.

For lyrical analysis, we tokenized cleaned song lyrics to prepare them for embedding. Using a straightforward simple\_tokenize function, each lyric string was split into individual words, balancing efficiency and accuracy for our dataset size. These tokens were then embedded using pre-trained FastText English word vectors (crawl-300d-2M-subword), with each song represented by the mean of its token embeddings. Songs without valid tokens were assigned a zero vector. This process transformed lyrical content into a numerical form that captures semantic relationships, enabling use in downstream models such as clustering or recommendation engines.

We also engineered an interaction term called Power, combining the normalized energy and loud features. With a correlation of 0.73, this term captures the relationship between dynamism and volume, especially relevant for genres like heavy rock or dance. Additional preprocessing included distribution normalization—BPM values were square root transformed to reduce right skew, and Acoustic values were also square root transformed to better spread their distribution. While we avoided dimensionality reduction at this stage to retain maximum information, these preprocessing steps ensured the dataset remained both structured and rich in musical detail for modeling.

**Model Training**

To provide a starting point for our semantic song recommendation system, we implemented the KNN modeling by using 300-dimensional FastText lyric embeddings that evaluate textual closeness in high-dimensional space by using cosine similarity. We evaluated recommendations based on neighborhood cohesiveness and compactness across a range of K values using this unsupervised method, which is independent of label dependence. We found that while a larger K decreased cohesion, it had no discernible effect on the top-ranked results.

Data was split for test, training, and validation, and embedding lists were transformed into a uniform NumPy array as part of the preprocessing step. Rap and pop/indie songs with similar styles demonstrated substantial grouping, according to qualitative analysis; however, country and EDM tracks had poorer semantic coherence because of their scant or deceptive lyric content.

Recommendations often featured popular tunes, indicating both centrality in the embedding space and possible popularity bias. The lack of audio features and user feedback is one of the limitations, which encourages the use of multimodal inputs, clustering techniques, and semi-supervised learning in the future to improve suggestions.

**Advanced Modeling**

By improving lyric embeddings and including audio data, we investigated two improved methods to increase music suggestions, building on our basic KNN model.

First of all, the model achieved a slight 2% accuracy boost over FastText by using Word2Vec embeddings based on Twitter, which better captured colloquial language and idiomatic idioms.

Secondly, a hybrid KNN\_Full model achieved our most significant result to date with an accuracy of 49% by combining these embeddings with normalized musical parameters like danceability, valence, and BPM.

When 100 inputs were manually evaluated using binary relevance scoring, structured characteristics were shown to increase consistency and reduce genre incompatibilities, particularly for rap tracks. But genres with scant lyrics, like EDM, continue to be difficult. To further improve suggestion quality, future research will investigate adaptive feature tuning and genre weighting.

**Model Development and Model Evaluation**

During this stage, we created KNN\_Weighted, a hybrid recommendation model that assigns equal weight to normalized audio characteristics and 200-dimensional GloVe lyric embeddings to musical and poetic elements.

 By improving semantic representation and balancing acoustic metadata, this method improves on previous models and makes recommendations that are more logical and genre-sensitive.

 KNN\_Weighted improved electronic music suggestions and lessened popularity bias by achieving a 55% accuracy rate, our highest to date, using cosine similarity and manual binary relevance rating over 100 test queries. Future goals include genre weighting, dataset enlargement, and top-N stochastic selection to improve diversity and resilience, even though country music is still difficult.

**Model Selection**

We thoroughly assessed nine KNN-based models with different embedding types, feature compositions, and weighting strategies to finalize our recommendation system. In the end, we chose a weighted full-feature model that struck the optimum balance between interpretability, generalization, and accuracy.

This successful setup makes use of normalized information and 200-dimensional GloVe-Twitter lyric embeddings, such as tempo, valence, and energy, as well as one-hot encoded genre vectors, using a weighting system of 10% genre, 45% lyrics, and 45% metadata.

It outperformed all previous models and showed a substantial bias-variance tradeoff, achieving 60% accuracy on both the validation and test sets with K=6 and cosine similarity. While maintaining transparency and tunability in feature contributions, the model greatly improved recommendations for previously underperforming genres such as country and electronic, making it a dependable and intelligible option for cold-start music selection.

**Data-Centric AI**

Our investigation concentrated on noise reduction, feature enhancement, and and dataset balancing, prioritizing data refinement over algorithmic complexity. Important tactics included removing underrepresented genres and popularity outliers, converting characteristics like acousticness and BPM for improved distribution, and adding artist and subgenre details to metadata.

 A cosine-distance KNN model with a consistent 45/45/10 weighting for lyrics, metadata, and genre, respectively, was built using these improvements. A richer background was included in Week 10; however, accuracy was somewhat lower than with the more broadly applicable Week 9 model, which was eventually used.

The need to maintain consistent weighting, the dangers of overcomplicating feature space, and the delicate balance between pruning and diversity were all highlighted in the lessons learned.

To further increase suggestion quality, future developments will concentrate on metadata standards, improved lyric preparation, and balanced genre sampling.

**Ethical Considerations**

There are ethical concerns with representation and confidence. Our music recommendation algorithm limits the visibility and income of underrepresented genres and independent musicians.

Sometimes, the way we use language unintentionally favors certain styles, artist names, or genres connected to cultural identity. For example, these patterns can act as a stand-in for broader cultural associations. Traits that might reveal bias. To mitigate these risks, we employed data balancing, post-recommendation auditing, and controlled genre weighting, acknowledging that some bias preserves cultural authenticity.

 There are trade-offs, though, as overcorrecting could erode genre identification or skew listening patterns in the actual world. To guarantee that fairness and relevance stay in line with changing musical landscapes, ethical management calls for openness in feature weighting, ongoing observation of suggestion diversity, and active interaction with artists and users.

**Final Model Deployment and Monitoring**

We put in place a strong deployment and monitoring framework that included model serialization, environment replication, and adaptive deployment techniques to guarantee the scalability and dependability of our music recommendation system.

Python's pickle module was used to serialize the finished model for portability and reuse for the data called “requirement.txt” to ensure reproducibility. For efficiency, deployment starts in batch mode and moves to real-time inference as the dataset expands.

User feedback scores, latency, and engagement indicators are used to track performance, and a three-tier alert system directs assistance. User ratings are incorporated to preserve relevance during retraining, which starts off monthly and progresses to weekly as the data amount increases.

Frequent data updates and dynamic feedback weighting reduce drift and guarantee that the model adjusts to changing musical tastes while preserving a constant level of recommendation quality.