**Week 8 Writeup - Model development and evaluation**

**Ryan O’Hara, Jack Metzger**

**Model Development**

In Week 8, we reframed the playlist recommendation task as a similarity-search problem, using the KNN model which was our ideal choice This was the best option we could select among the other models which is identifying songs most similar to a given query track without matching on the target variable because it was non-parametric, it required little training time. It enabled complex feature engineering, which kept the system both practical and understandable for all parties involved. Therefore, the KNN models allow the improvement of suggestion quality by revealing subtle, hidden musical linkages through high-dimensional embeddings, such as those found in lyrics.

The procedure demonstrates how to integrate textual and numerical data into a reliable pipeline to generate feature vectors from Spotify track data. Tokenization, lyrics embedding, and the extraction of numerical metadata, such as danceability or pace, occur after data intake. To ensure that metadata and embeddings had an equal influence, we scaled and normalized the feature blocks using a square-root transformation prior to concatenation. This architecture produces a well-structured matrix appropriate for similarity-based recommendation algorithms and allows for flexible changes between metadata and lyrical semantics without changing their original scales.

We used three KNN Parameters, which are 5, 10, and 15, to investigate the effects of various feature representations on the quality of music recommendations. V1, which only deals with metadata, provides an interpretable baseline by using only numeric audio metadata with cosine distance. To capture more profound textual similarities, V2, which focuses on embeddings only, concentrated on semantic content using 200-dimensional GloVe lyric embeddings, which also employed cosine distance. To manage their influence, V3, which is a hybrid of embeddings and metadata, merges both by concatenating and block-weighting the metadata and embedding features. The metadata weight ratio (α) was chosen to be .50 for the V3 model, meaning the embeddings were weighted as 50% of the calculated cosine distance and the song metadata (popularity, dance, sqrt\_BPM, sqrt\_acoustic, happy, and power) contributed the other 50%. Our group felt that a 50/50 contribution split made sense as lyrics and melody are often viewed as half the song. Although there was no formal survey, from our own personal experiences and asking a number of people outside the project if they enjoyed music for more the lyrics, the melody, or both; responses were generally split evenly between the three categories. For this specific iteration of the model, k was only set to k=1. This is because we are currently generating the top recommendation for our testing set, so running a model with a different k parameter would produce the same results for what we are currently evaluating.

**Model Complexity, Hyperparameters, and Reproducibility**

The KNN recommender models require a training cost of O(1), which involves storing the features. Querying cost is O(k · d), which is manageable for datasets, allowing us to use a neural network, such as an ANN. Memory is closely related to the feature matrix. Nonetheless, a compact vector with expressive power is guaranteed by the unique weighting technique.

The key parameter k plays a distinct role in recommendation, and making it smaller and narrower results in brittle suggestions. The k value is 1 for this current version of the model. The weight parameter controlled the tradeoff between audio features and semantic lyrics and was roughly adjusted to 0.5 to preserve a balanced influence. In contrast to Euclidean distance, which was quickly abandoned because of its poorer qualitative performance, cosine distance was chosen for its scale invariance and high alignment with text-based embedding geometries.

To ensure reproducibility, all sources of randomness in the pipeline which is combining the embeddings and neighbor queries, were consistently producing good recommendations. We are retraining the recommendation systems to enhance both replication and transparency.

**Model Evaluation**

The model showed significant gains in accuracy after this iteration. The new KNN\_weighted model weighted the lyric embeddings to contribute 50% to the distance metric and the other song/melody metadata was weighted to contribute the other 50%, achieving a score of 55/100 on the testing set. While the accuracy score didn’t necessarily reflect how much better the recommendations were, our group is just a few small adjustments away from having a very high quality recommendation model.

The recommendations that were graded as good were significantly closer to the input song than they have been in previous iterations, so while the score assigned to each is either a 1 or 0, the 1’s were much better than they have been in the past.

The model showed significant improvement in certain genres, specifically in the electronic genre. This genre has given us trouble in the past as there are fewer lyrics to base the recommendations off of and it relies heavily on the actual beats. This iteration handled these songs better than any other model.

We have consistently had trouble with country songs - both with this genre as the input genre and as the recommended song. We believe this is a tricky genre for us because musically it can be very similar to many other genres, including rock, pop, alternative, and indie. These recommendations are consistently very far off, and had we adjusted for this genre, our accuracy would likely improve by at least 10-15%.

We plan to introduce some light genre labeling next to account for some of this country genre bias in our model. While the goal of this project was to create a recommendation model that transcends the traditional “genre” labeling, we believe that if we lightly weight genre labels for certain genres like country, we can nudge songs closer together than clearly need a little adjustment. We can either weigh other genres at a lower rate or not at all so we remain consistent with our original mission.

Other than that, we believe our model is extremely close to being a ~70% accurate model, to which there are many other iterations we can tweak to make slight improvements in accuracy. While we don’t anticipate doing this for the scope of this class, in the grand scheme of all songs created, our sample size of 3,000 songs is relatively small. We believe that in the future, given more time, resources, and compute power, we would achieve substantial gains simply by getting our sample to 20,000 songs.

As seen in the table below, we made another substantial jump in accuracy, and while it appears to be only 6%, we are confident that the jump made by this model was much more significant than that.

|  |  |  |  |
| --- | --- | --- | --- |
| Model | Accuracy | Confidence Interval | K Parameter |
| KNN | 41/100 | 36 to 46 | 1 |
| KNN\_Twitter | 43/100 | 38 to 48 | 1 |
| KNN\_Full | 49/100 | 44 to 54 | 1 |
| KNN\_Weighted | 55/100 | 50-60 | 1 |

**Table 1:** Comparison of the three model performances with +- 5 confidence intervals. KNN\_Full showed the best performance. KNN\_Weighted provided the best results.

**Future Work:**

The KNN\_Weighted model outperformed the other two versions, as predicted. Still, there’s room for improvement as we aim to push the model’s accuracy closer to our group’s target of 70%. One persistent challenge has been the country genre—both as an input and a recommendation. (Note: Due to the subjective nature of recommendations and the vast options of songs to recommend; 55% is a solid accuracy metric for this model where it may be viewed as low for other ML models/problems. Random song recommendations would have a hit rate far below 50%).

Our dataset includes a range of one-hot encoded genres and subgenres pulled from the metadata in weeks 1 and 2. We plan to integrate these genre features into our model. We hope that doing so might resolve some of the genre-specific shortcomings, such as those seen with country music. However, we will be cautious about overemphasizing genre labels in the model. We believe excessive reliance on genre labels risks overfitting and may hinder broader music discovery. One of our model’s core objectives is to move beyond rigid genre classifications.

As mentioned above, our recommendations are improving with each successive model. It is important to recognize the subjective nature of grading a recommended song. In addition, our recommendations are being pulled from a pool containing slightly under 2,000 songs. That is a solid sample size considering our current resources, but still limited when compared to the vast catalog available on platforms like Spotify. With this in mind, it's possible that some recommendations may be optimal within the model's constraints, even if they appear inaccurate. To address this, one idea is to integrate more songs into our model training set during week 13 for a total of 3,000 - 4,000 songs. With our current defined pipeline, it would be possible to attempt this scale up in an efficient way.

Lastly, if we want to incorporate randomness in the model, we can set the k parameter to a higher number (i.e. 5, 10, 15, etc.) and code the model to randomly (full random or weighted random) select a recommendation from the top 10 recommendations for that song. This could help integrate some diversity and randomness into the model, and will enable the model to provide a different recommendation for the same song, adding a layer of complexity.

By integrating some or all of the suggestions, we are confident in continuing to increase our accuracy metric while creating a more robust and user-friendly model.

**Reference**

**1.**Sangeetha, A., Tejashree, M., Roshini, B. (2025). Spotify Dataset: Recommendation of Popularity Songs (Genre) Using Machine Learning Classification Techniques. In: Madureira, A.M., Abraham, A., Bajaj, A., Kahraman, C. (eds) Hybrid Intelligent Systems. HIS 2023. Lecture Notes in Networks and Systems, vol 1227. Springer, Cham.