1. Introduction
    a. Literature Review
        i. Traditionally, recommender systems have utilized collaborative filtering which was based on “interactions between users and items, e.g., clicks or ratings” but now recommender systems have now switched to content-based filtering which is based on recommending items similar to what the user likes. Markus Schedl researched methods that adopt deep learning techniques for content-based filtering such as deep neural networks to derive songs/artists.
        ii. One of the most popular streaming services, Spotify, uses an algorithmic recommendation system. An algorithmic recommendation system is “content suggestions generated by algorithms” produced by machine learning models that analyze Spotify’s users behaviors and actions. This source compares users' activity based on organic user-driven listening vs algorithmic programmed listening and compares the differences in what the users listen to when exposed to one method vs the other.
        iii. This source introduces 3 different recommendation systems hoping to “construct a complete music recommendation system, hoping to overcome the limitations” of content-based filtering and collaborative filtering. The 3 different recommendation systems they introduce are “The profile manager,” “The CB method,” and “The COL method.”
    b. Dataset Description: We are also changing our dataset to a cleaner dataset. When we tried to read our original dataset, it was very dirty and hard to read. Thus we found a cleaner dataset that is more neatly organized that utilizes data from Spotify. It shows clear tables and columns of a song's characteristics such as tempo, popularity, energy, etc.
    c. Dataset Link: https://www.kaggle.com/code/vatsalmavani/music-recommendation-system-using-spotify-dataset/input

2. Problem Definition and Motivation
    a. Problem:
        i. All music listeners, whether new or experienced, miss out on discovering new songs or artists due to the overwhelming volume of music available. The problem is that it’s nearly impossible for users to explore all potential songs of interest manually.
    b. Motivation:
        i. Our objective is to create a song recommendation system to introduce users to new songs and artists they haven't heard before that may be of interest. Before, our idea was to use data from what genres users personally explicitly said they like. (addition) However, we ran into a problem where our dataset doesn’t provide that specific information so it was hard to run with. Therefore we changed our approach for our music recommender system to recommend songs that are similar to a specific song. For example, users would be able to search up similar and related

songs to a specific song they may really like. In addition to that, we were going to surround our idea based on collaborative filtering, but again, since collaborative filtering is heavily reliant on user data that we could not access, we are changing our approach to content-based filtering.

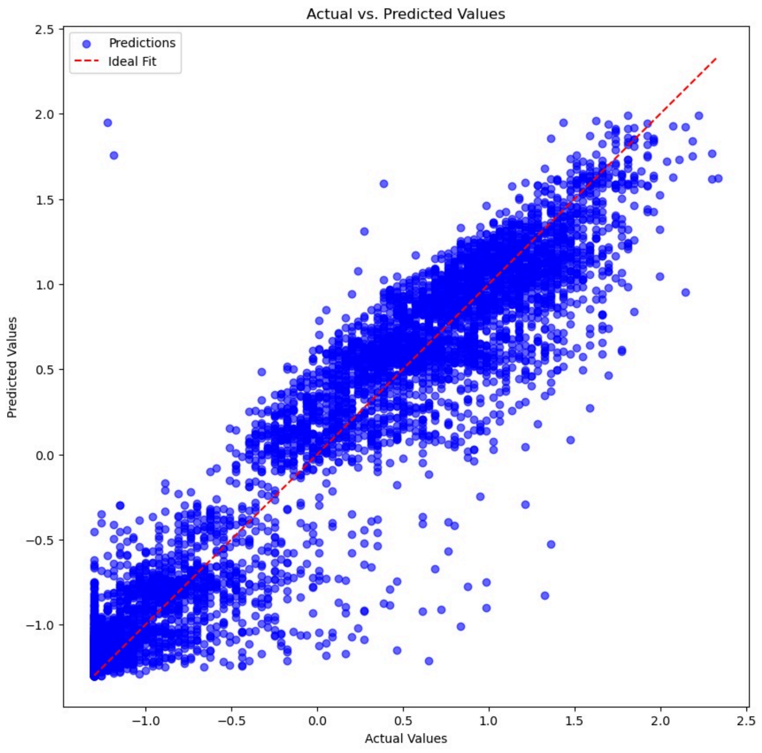
3. Methods
    a. Preprocessing Methods
        i. Data Cleaning: Handle null values, outliers, and unnecessary features
        ii. Feature Engineering: One-hot and ordinal encoding are two examples of the many we may need.
        iii. Dimensionality Reduction: Will need to reduce the dimensions while maintaining relationships and patterns in data.
        iv. Data Transformation: Normalize data to make data easier and more effective to train models with.
        v. Sampling Data: Since this is a very large subset, we will use a subset of data for training and testing.
    b. ML Models
        i. These are models we believe are relevant towards our project. Currently we will be using 1. K-means for clustering, 2. KNN for predicting similar data-points, and 3. Random Forest Regressor to predict how much a user might like the recommended song
            1. K-means
            2. K-nearest neighbors
            3. Random Forest Regressor

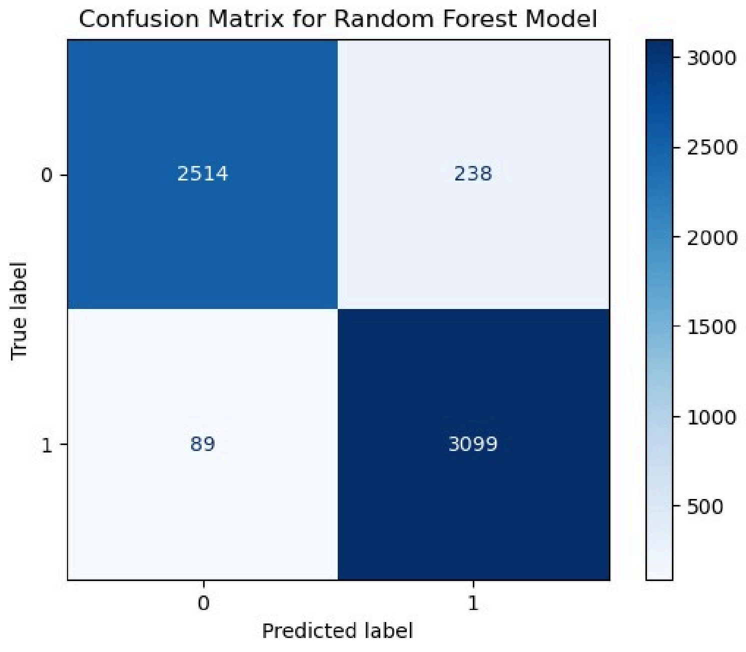
        Out of these relevant models for our project, we implemented the Random Forest Regressor (RFR) model in relevance to content based filtering. We were able to use RFR to predict preferences on attributes such as energy, popularity, etc. by training the model to predict a user's preferability toward a song based on its popularity. The model is trained from previous song instances with a determined popularity level. The arbitrary features of the song, like acoustics and genre, are used as the relational identifiers to find similarity with instantiated songs and the new song. If there is a similar match based on the metrics, then it will predict a similar popularity. For example, if two songs share the exact same metrics, it will be incredibly likely for them to share the same predicted popularity according to the model. There are intuitive benefits for us utilizing the RFR model. The RFR model will allow us to train off of a song's measured popularity, seeing their patterns, and determining what leads to a higher or lower popularity based on the song's features. Furthermore, a majority of people tend to follow the trending songs for their song preferences, as it tends to be indicative of a song being "good."

        The purpose of us choosing K Means is predicated on the case of the RFR model not being able to understand what songs are of similar styles/interests. K Means will group songs of similar values together, however due to the unsupervised nature of the model the feature labels will be lost when clustering. This will still yield a better outcome in comparison to the RFR model due to the clustering being geared toward a song's characteristics. Subsequently allowing for predicted user interest to not be strictly determined off a similar song's arbitrary popularity, rather catered to the unique music profile of the hypothetically entered song instance.

Lastly, we chose K-nearest neighbors (KNN) since it is a supervised learning model that can detect similar songs by similar neighbors (Euclidean distance) . We can generate songs more similar to themselves rather than a generalized group. It will show the K most similar songs to the one we input, yielding an objectively higher similarity between the songs suggested.

4. Results and Discussion
   a. Random Forest RegressorVisualization(s)





      i. Quantitative metrics
         1. Cross-validation scores (neg MSE): [-0.08660587 -0.09356522 -0.09719014 -0.09392039 -0.09292569]

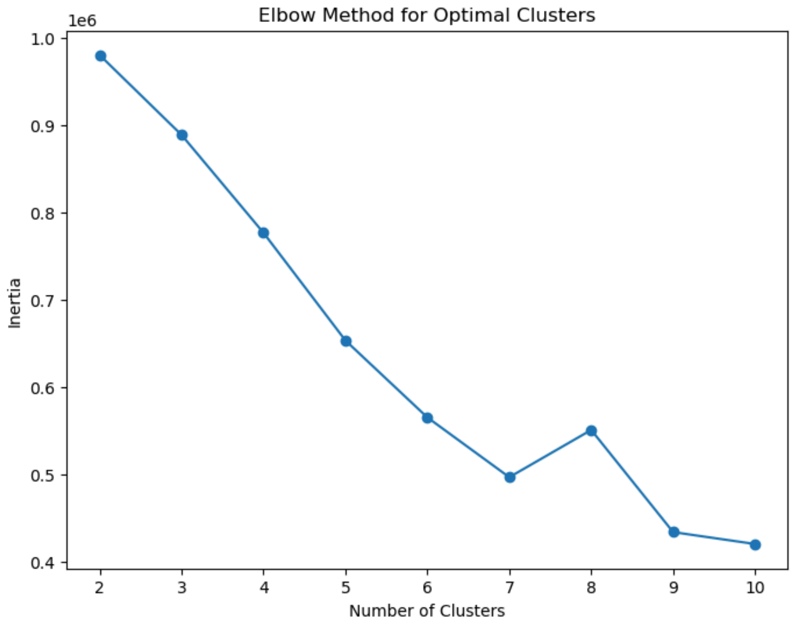
2. Root Mean Squared Error for each fold: [0.29428875 0.30588433 0.31175333 0.30646433 0.30483716]
3. Mean Root Mean Squared Error: 0.30464557787548807

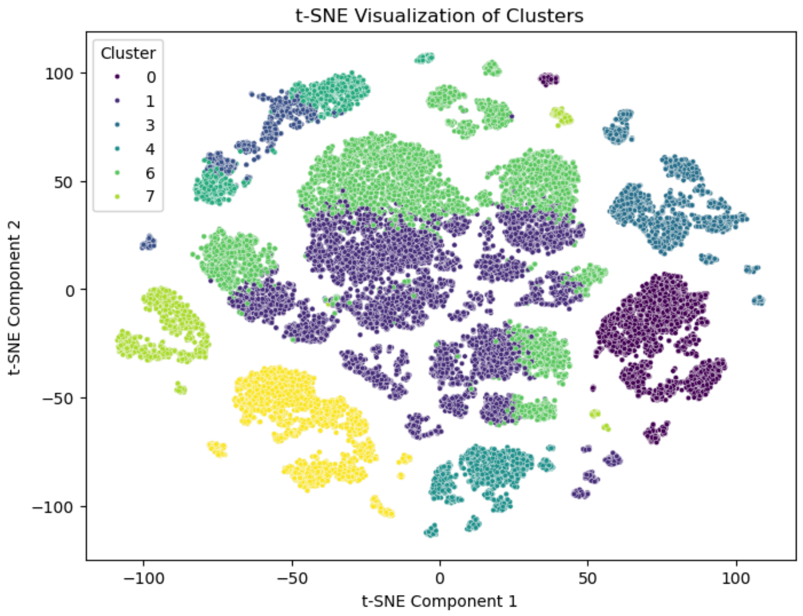
ii. Analysis

The Random Forest Regressor (RFR) demonstrated robust and consistent performance in predicting the popularity metric, as evidenced by our cross-validated Root Mean Squared Error (RMSE) scores. The low variance among the RMSE values across the five folds indicates that the model maintains stable predictive accuracy regardless of the specific data split, highlighting its reliability and generalization capabilities. This consistent performance can be attributed to the RFR's inherent ability to capture complex feature interactions and mitigate overfitting through ensemble averaging of multiple decision trees. Additionally, the effective feature selection and relevance likely played a crucial role, ensuring that the most predictive variables were leveraged efficiently. However, the RMSE value's interpretation is contingent on the scale of the popularity variable; since ours is scaled between 0 and 100, it indicates a relatively low error. Overall, the RFR model's strong performance suggests that it effectively models the underlying patterns in the data.

b. K Means

i. Visualization(s)



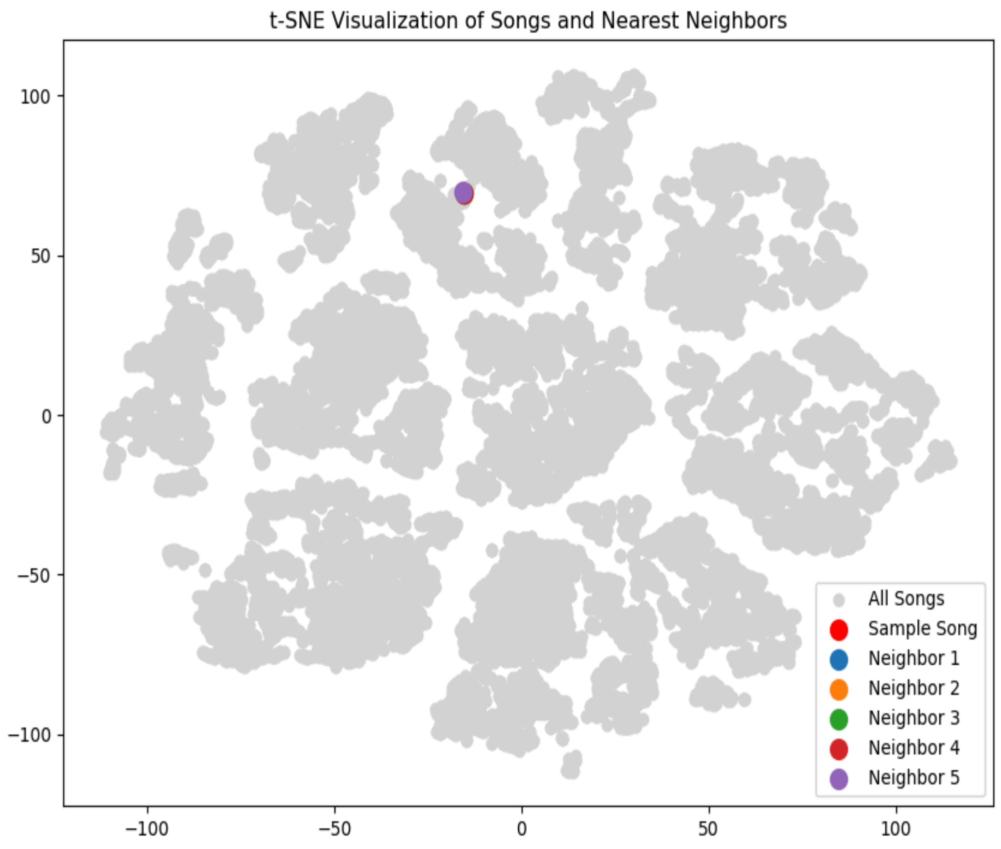


ii. Quantitative metrics
   1. Silhouette Score for 2 clusters: 0.32401411537665503
      Silhouette Score for 3 clusters: 0.13915033981407682
      Silhouette Score for 4 clusters: 0.15425173009870227
      Silhouette Score for 5 clusters: 0.20019064873310627
      Silhouette Score for 6 clusters: 0.23306320448929127
      Silhouette Score for 7 clusters: 0.26006253197698326
      Silhouette Score for 8 clusters: 0.21845200543074625
      Silhouette Score for 9 clusters: 0.25134226470191506
   2. Average Silhouette Score for cluster range 2 to 9: 0.22256585507768453

iii. Analysis

The K-Means clustering analysis yielded relatively modest silhouette scores, with the highest score of approximately 0.324 achieved when using two clusters. This indicates a fair level of separation between the two largest clusters, suggesting that the dataset may naturally divide into two broad categories. However, as the number of clusters increases beyond two, the silhouette scores generally decline or show minimal improvement, averaging around 0.222 across clusters ranging from two to nine. These lower scores imply that additional clusters do not significantly enhance the distinctiveness of the groupings and may even lead to overlapping or less well-defined clusters. The performance could be influenced by several factors, such as the high dimensionality introduced by one-hot encoding the 'genres' feature, which can dilute the cluster distinctions, or the inherent variability and overlap in the data attributes like artist frequencies and genre distributions. Additionally, K-Means assumes spherical cluster shapes and equal variance, which might not align well with the actual data structure. The t-SNE visualization further supports this by potentially showing overlapping clusters in the reduced two-dimensional space. Overall, the silhouette scores suggest that while a basic clustering structure exists, our current feature engineering and clustering approach may need refinement, such as exploring alternative encoding methods, dimensionality reduction techniques, or different clustering algorithms better suited to the data's intrinsic patterns.

c. K-nearest Neighbors
   i. Visualization(s)



   ii. Quantitative metrics
      1. Diversity Score: 0.57
   iii. Analysis

The K-Nearest Neighbors (KNN) model effectively identified songs similar to the sample track based on a comprehensive set of audio and metadata features, including valence, energy, acousticness, and instrumentalness. The recommended songs from artists exhibit comparable high energy levels and varying degrees of acoustic and instrumental qualities, aligning well with the characteristics of the sample song. The Euclidean distance metric ensures that the recommendations are closely matched in the multidimensional feature space of the music. The diversity score of 0.57 indicates a moderate level of variety among the recommended songs, suggesting that while the selections share core similarities with the sample, there is enough diversity to cater to different listener preferences within the same general style. This balance enhances user experience by providing familiar yet distinct options, preventing the recommendations from being overly homogeneous. Overall, the model demonstrates a solid performance in delivering relevant and varied song suggestions.

d. Model Comparisons (Strengths and trade-offs)

The K-Nearest Neighbors (KNN), K-Means clustering, and Random Forest Regressor (RFR) models each offer unique approaches to analyzing the dataset, with distinct strengths and tradeoffs. KNN excels in providing personalized song recommendations by identifying similar items based on Euclidean distance, ensuring high relevance through closely matched features. However, its performance is moderately limited by a diversity score of 0.57, indicating that while recommendations are relevant, they may lack sufficient variety. Additionally, KNN can struggle with high-dimensional data, potentially impacting scalability and efficiency. In contrast, K-Means clustering aims to uncover inherent groupings within the data, achieving fair separation with a silhouette score of 0.324 for two clusters. This approach is effective for identifying broad categories but faces challenges as the number of clusters increases, with diminishing silhouette scores suggesting overlapping or less distinct groupings. The high dimensionality from one-hot encoding genres and K-Means' assumption of spherical clusters further limit its effectiveness in capturing nuanced relationships. On the other hand, the RFR model demonstrates robust and consistent performance in predicting the popularity metric, evidenced by low and stable cross-validated RMSE scores. Its ability to capture complex feature interactions and mitigate overfitting through ensemble averaging makes it highly reliable for regression tasks. However, RFR operates as a black-box model, offering less interpretability compared to clustering methods and being specialized for prediction rather than exploratory data analysis.

Overall, KNN is ideal for recommendation tasks with a focus on similarity, K-Means is suited for identifying general groupings within the data, and RFR excels in accurate and reliable prediction of numerical outcomes. Each model complements different analytical objectives, with KNN and K-Means being more exploratory and RFR providing strong predictive capabilities, while also presenting limitations related to feature dependence, interpretability, and scalability.

5. Next Steps

Based on the analyses of the K-Nearest Neighbors (KNN), K-Means clustering, and Random Forest Regressor (RFR) models, the next steps should focus on enhancing model performance, addressing identified limitations, and exploring complementary techniques to maximize the insights and utility derived from the data.

For the KNN model, consider implementing dimensionality reduction methods like Principal Component Analysis (PCA) to mitigate the challenges posed by high-dimensional data, and experiment with different distance metrics or hyperparameters to improve recommendation diversity beyond the current score of 0.57. Exploring hybrid recommendation systems that incorporate additional criteria such as user preferences or temporal dynamics could also enhance relevance and variety.

For the K-Means clustering analysis, experimenting with alternative clustering algorithms such as DBSCAN or hierarchical clustering may yield better-defined clusters, especially if the data does not naturally conform to spherical shapes. Additionally, refining feature engineering by reducing the impact of one-hot encoded genres through techniques like embedding or feature selection could improve silhouette scores and cluster distinctiveness.

Lastly, in the case of the RFR model, further hyperparameter tuning and exploring ensemble methods like Gradient Boosting or XGBoost could potentially enhance predictive accuracy. Conducting a thorough feature importance analysis might also reveal new insights or suggest additional features to incorporate. Additionally, implementing cross-validation strategies with more folds or different sampling techniques can provide a more robust evaluation of model performance.

6. References

a. M. Schedl, “Deep Learning in Music Recommendation Systems,” *Frontiers*, Aug. 29, 2019. https://www.frontiersin.org/journals/applied-mathematics-and-statistics/articles/10.3389/fams.2019.00044/full?ref=https%3A%2F%2Fgithubhelp.com
b. A. Anderson, L. Maystre, I. Anderson, R. Mehrotra, and M. Lalmas, “Algorithmic Effects on the Diversity of Consumption on Spotify,” *Proceedings of The Web Conference 2020*, pp. 2155–2165, Apr. 2020, doi: https://doi.org/10.1145/3366423.3380281.
c. H.-C. Chen and A. L. P. Chen, “A Music Recommendation System Based on Music and User Grouping,” *Journal of Intelligent Information Systems*, vol. 24, no. 2–3, pp. 113–132, Mar. 2005, doi: https://doi.org/10.1007/s10844-005-0319-3.

Name	Proposal Contributions
Benjamin Jin	Implemented KNN model + Presentation
Jason Lee	Implemented K-Means model
Jiyoon Lee	Implemented K-Means model
Julien Sanchez	Final Report Write-Up + Model Selections + Recording
Mirbaan Balagam	Implemented KNN model