IDS-705: Model Based Approach to Music Genre Assignment

Clarissa Ache, Preet Khowaja, John Owusu Duah, & Cameron Ratliff

April 7, 2022

Abstract

The SlackBot platform was used to create the chatbot, "slash" commands were used to create interactivity, and AWS AppRunner to deploy the application. The elective data is hosted on DynamoDB, and AppRunner can read and write from the database.

1. Introduction

As a top-level descriptor, musical genres represent a simplification of a song's intricate details and are of great interest as summaries of shared characteristics in music pieces. With the emergence of digital music platforms like Spotify and Apple Music, the importance of genres to help listeners to query huge catalogs of readily available music cannot be exaggerated. However, despite their importance, music genres remain a poorly defined concept, which has been at the center of controversy and dispute. Another challenge presented by the scale that digital music distribution platforms present is the need to find efficient ways to annotate old and new catalogs of music. Dannenberg [2] reports that manually labeling of 100,000 songs for Microsoft's MSN music search engine required about 30 musicologists for a year. Also, what was once a trade secret of elite music record labels, leveraging the technical attributes of popular music to inform music projects that receive financial backing has now become trite knowledge and common practice. Musicians, through their own volition or under the influence of record labels have often adapted the sound of their music to mimic prevailing popular music. The Recording Academy's response to this adaptation has been to create new genres and sub-genres to classify music that sounds like a fusion of music belonging to separate genres. Their genre and sub-genre assignment approach is subjective, and while it offers musicians novelty and opportunities to gain recognition in award categories that did not exist before, it comes at a cost of obfuscating the technical attributes of popular music. There is evidence that audio signals carries information about music genre [16]. This makes the study of a model-based approach for music genre assignment using audio signals a nontrivial task. This study seeks to use a model-based approach to eliminate subjectivity in music genre assignment and its related problems. We aim to give music record labels and independent musicians a scientific method of mapping popular music to genres so they leverage this knowledge to improve their bottom line.

This study seeks to bring clarity to the task of music genre assignment by assigning genres to music using technical attributes as features with a supervised machine learning approach in lieu of the subjective status quo. Before the supervised machine learning could do this, it had to be trained on a data set of songs with genres assigned by an unsupervised machine learning model. The existing genres have been discarded due to reasons discussed in the section 2, and new genres or cluster labels have been assigned to each song using this unsupervised machine learning model based on the structure of the technical attributes of the songs. Ultimately, the value of this study lies in how well music pieces can be grouped into separate clusters. Several experiments using different techniques have been carried out to this end, and each has been evaluated and discussed.

The paper is organized as follows: In the next section we situate our study in the context of past work in the field. Section 3 - Data, outlines a thorough exploration of the data set and a discussion of the challenges and drawbacks of the data and its implication on the study. Each experiment and its corresponding justification is discussed in section 4. Data preprocessing, feature extraction, machine learning models have also been outlined in this section. Section 4, Methods, also outlines how training and validation sets are composed for both supervised and unsupervised machine learning models and a discussion on the choice of performance evaluation metrics used and why they are appropriate. Section 5 - Results, shows complete performance assessment for each experiment/analysis with the appropriate performance metrics. A comparison of the performance of baseline model to other models has been presented here. Finally, a discussion of how the results of the experiments answer the problem we seek to solve is outlined in section 6 - Conclusion.

2. Background

The music genre classification problem requires an analysis of the existing taxonomy of genres. Pachet [3] proved, after studying the field music genre taxonomies used in the music industry, that it is not straightforward to build up such a hierarchy of genres. Since a robust classification relies on careful thought of taxonomy, we throw light on some critical issues with the prevailing. Pachet [3] showed that a general agreement on genre taxonomies does not exist. They found out that different domains and regions of the world assigned different genres to the same songs. They noticed that some widely used terms like rock or pop denote different sets of songs and those hierarchies of genres are differently structured from one taxonomy to the other. Our exploration of widely used music genres the world over reveals inconsistent variations in the manner genres are defined. Some instances are:

- Latin music is geographically defined
- Pop (an abbreviation of popular) music is assigned to any music piece that appeals to a very wide audience.
- Post-rock is characterized by a focus on exploring textures and timbre over traditional rock song structures, chords, or riffs.

This semantic confusion leads to redundancies and a fuzzy boundary between music genres and alienate objective technical attributes of music from their classification. The decision to discard the existing genre assignments of music in our study was informed by the above inadequacies of the status quo. As a principle, it is important for music genres to account for the possibility of adding new genres as the taste of listeners evolve. In the past, new genres have emerged typically as the result of some merging different genres or the discovery of new mellifluous sound patterns. Any model-based approach to classify music genres should account for this and we propose that as shifts in the taste of audiences occur, the models and approaches proposed be recomputed and reevaluated to capture new genres.

There have been a number of studies on clustering and classifying the genre of music using features from a myriad of sources. What most of these attempts have in common is that they rely on the temporal structure of a music piece as an important predictor of the genre. Most of the music genre classification algorithms resort to the so-called bag-of-features approach, [15] which models audio signals by their long-term statistical distribution of short time features. Features commonly exploited for music genre classification can be roughly classified into timbral texture, rhythmic, pitch content ones, or their combinations [16].In one of the earliest research studies on the subject matter, Soltau [1], in building a music type recognition system that can be used to index and query multimedia databases into rock, pop, techno and classic genres, transformed the acoustic signal of music into a series of acoustic events. Frequencies of these events and transitions in the sequence were calculated and combined into one vector which contains the temporal structure information of the sequence. These feature vectors were used to train a neural network to recognize the music type. The recognition rate of their final model was 86.1%. It is important to note that early research in this domain, which occurred in latter part of the 1990's, does not address issues related to the taxonomy of genres. Instead, a lot of effort was spent developing models to classify music according to existing genre labels. Some approaches focused on the direct similarity between the feature sets rather than the temporal structures. Cilibrasi [18] used a distance function between feature vectors and generated trees by the distances to visualize the similarity between samples of classical, rock and jazz[3]. Peng [19] used features from the signals to perform a k-means clustering, and used some labels as constraints to perform a constraint-based clustering to group a set of songs by their artists.

However in 2004, Shao [5], in their paper titled Unsupervised Classification of Music Genres Using Hidden Markov Models, identified and sought to solve the problem of existing human genre assignment by clustering songs using intrinsic rhythmic structural properties of music such as beat, tempo, and time signature. These periodicities contain obvious time-sequential information which were used to train a Hidden Markov Model. In a second step, they embedded the distance between every pair of music pieces

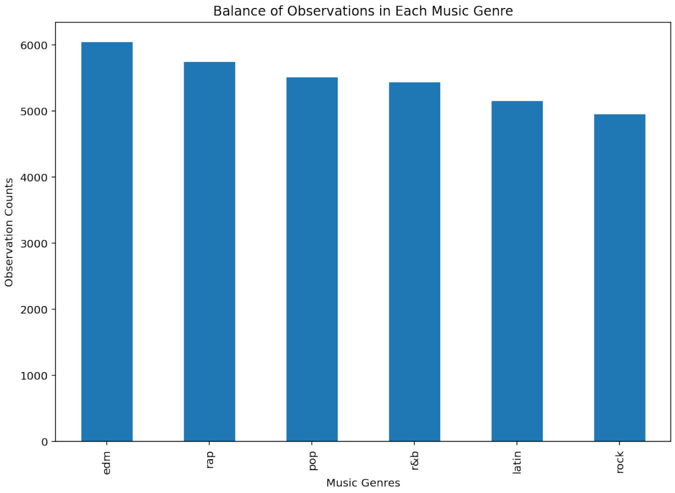


Figure 1: Distribution of Number of Songs in Each Genre

(HMMs) into a distance matrix and performed clustering to generate desired clusters. The results of this approach was mixed. Some genres of music, like jazz proved to be more difficult to classify than other genres like classical.

Tsai [6]. studied an unsupervised paradigm for music genre classification by partitioning a collection of unknown music recordings into several clusters such that each cluster contains recordings in only one genre and different clusters represent different genres. The features they used were Timbre-based features. Timbre-based feature vectors were extracted from music recording is then represented by a Gaussian Mixture Model (GMM). They assumed that each piece of music has its own genre pattern that reflects in the distribution of Timbre-based features over a span of time. An agglomerative clustering algorithm was used to group together the recordings deemed similar to one another. The Rand index and purity was used to determine the optimal number of clusters and evaluate the approach. They conclude by stating that there is the need to explore other audio features that relate more closely to music genre. This study uses attributes of songs provided by Spotify as features. While some attributes like loudness and tempo - speed or pace of a given piece derived directly from the average beat duration, are measured directly from the music pieces, other elements like danceability are an amalgamation of other elements like tempo, rhythm stability, beat strength, and overall regularity. These features are readily available for every music piece on Spotify's massive database which is representative of the greater population of all published music. This makes it an ideal set of features to use for music genre reassignment. An in-depth discussion of all the features has been outlined in section 3.

There are limitations the previous work studied. First, their data sets have less than 100 samples for each genre. This can be attributed to their choice of features and lack of access to curated digital music at the time of their publication. Also, the methodology of feature extraction and transformation does not scale for large volumes of music available on digital music platforms. Spotify's Application Programming Interface (API), one of the largest digital music service repositories, gives us access to a disproportionately wider range of audio features and higher number of samples per genre. Although, we restricted our data set to approximately 5,500 samples per genre, with greater computing power, more samples can be extracted and analysed, which will yield results with greater statistical power.

3. Data

The success of a model-based approach is largely dependent on the data. Any data sourced to solve a problem should be representative of the population of the domain of the problem. The data set for this study has been sourced from Spotify's API and it is a collection of 32,833 music pieces of approximately 5,500 songs randomly sampled from existing music genres of pop, rap, electronic dance music (EDM), rock, latin, and r&b.

The unit of observation is a music piece belonging to an existing genre. We note that more music genres exist, however, by selecting music pieces belonging to a sub-sample of existing music genres, we hope to draw conclusions that will be generalizable to the greater population of music pieces. The features

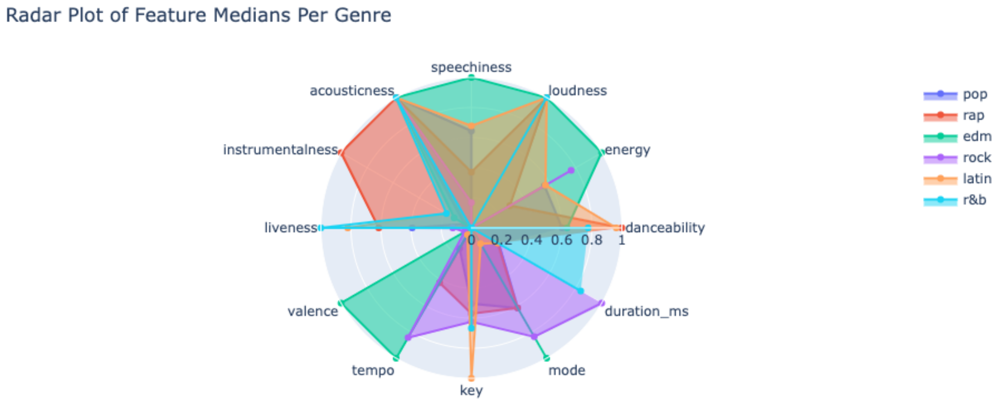


Figure 2: A radar plot of the composition of the original genres in terms of the 12 characteristics of their songs helps visualize the composition of the preexisting groups.

used for the study are the technical attributes of music pieces measured by Spotify. A description of each feature has been outlined in the Appendix 0.1.

Although this study is limited in that it does not verify if Spotify's measurements and estimation of these technical attributes are accurate, no other research on this subject matter has had access to a richer, scalable feature space. For every single music piece in Spotify's massive database, these features are readily variable, thus this study is underpinned by the assumption that there is no incentive for Spotify to measure and estimate these attributes incorrectly for any sub-group of music pieces.

Exploratory Data Analysis (EDA) commenced with an exploration of the data set with existing music genres intact. A check for balance across all six (6) genres revealed an approximately equal distribution as shown in Figure 1.

Radar plot in Figure 2, shows the relationship between each genre and the feature space. We can see that rap music has higher values for speechiness compared to other genres. We can also see that EDM music pieces are louder than other genres. The radar plot indicates that there are some overlaps across most genres and in terms of features selected for this study, and this is expected since most music pieces have a common baseline structure and instruments.

The correlation between individual features was explored and a correlation matrix of the features used in the study is shown in figure 3 in appendix.

We see that the highest correlation value of 0.68 is between loudness and energy. The next highest correlation value of 0.33 is shared by valence and danceability. Overall, there is little correlation between the individual features and this implies that each feature offers unique information to the task at hand.

We reduced the feature space from 12 dimensions to 2 dimensions, while preserving the variance within within the data set so as to visualize and gain a cursory understanding of the underlying structure of the samples. Van-der-Maaten and Hinton [10] presented a visualization technique which is a variation of Stochastic Neighbor Embedding (Hinton and Rosweis [9]) called t-distributed stochastic neighbourhood embedding (t-SNE). They showed that t-SNE visualizes high-dimensional data by giving each data point a location in a two or three-dimensional map. They also showed that t-SNE is easier to optimize and produces significantly better visualizations by reducing the tendency to crowd points together at the center of the map or plot. This proved true when we tried to visualize data set using Principal Component Analysis (PCA). Figure 4 shows a 2-dimensional representation of the features using the t-SNE technique.

4. Methods

Our methodology to address the problem of subjective music genre assignment involves a clustering algorithms to assign music pieces to new genres and a supervised machine learning model to reclassify existing songs and classify songs to new genres or clusters. Figure 5 below is a flowchart of our methodology.

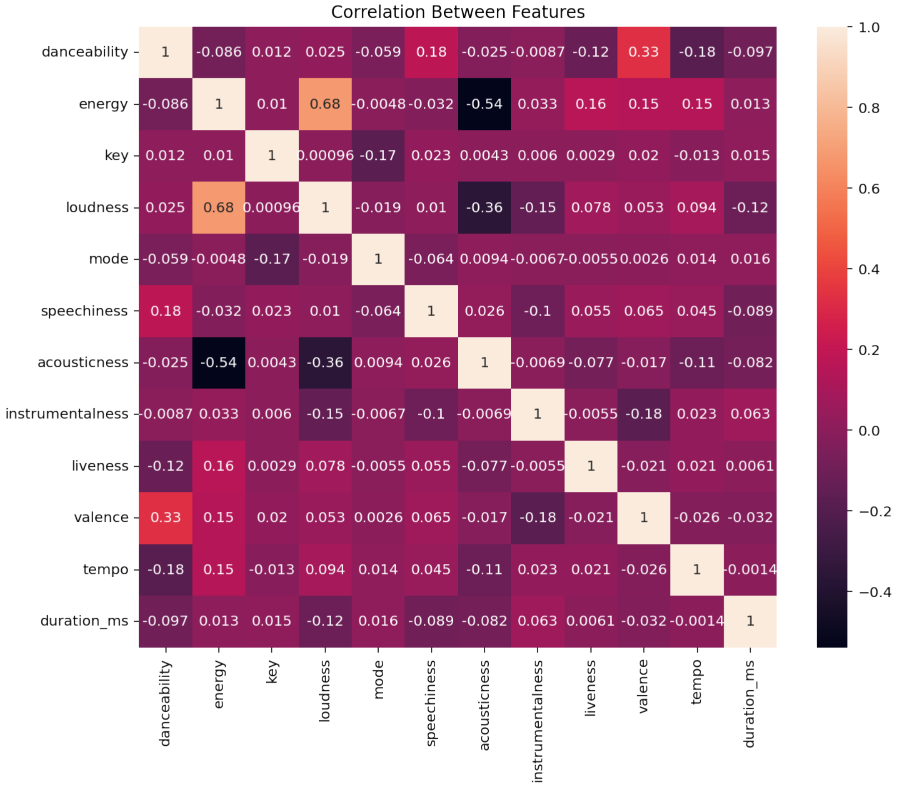


Figure 3: Data features are not highly correlated, which means Spotify provides a good set of measurements on different aspects of a song.

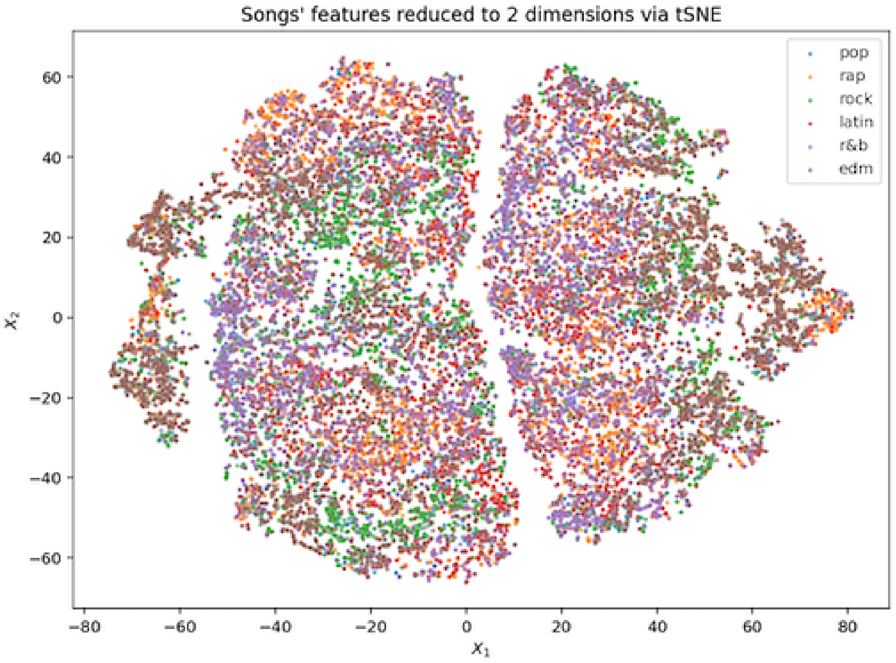


Figure 4: Plot of Song Data colored by genre via t-SNE demonstrates how current genres assigned by Spotify are not visibly separable in 2 dimensions.

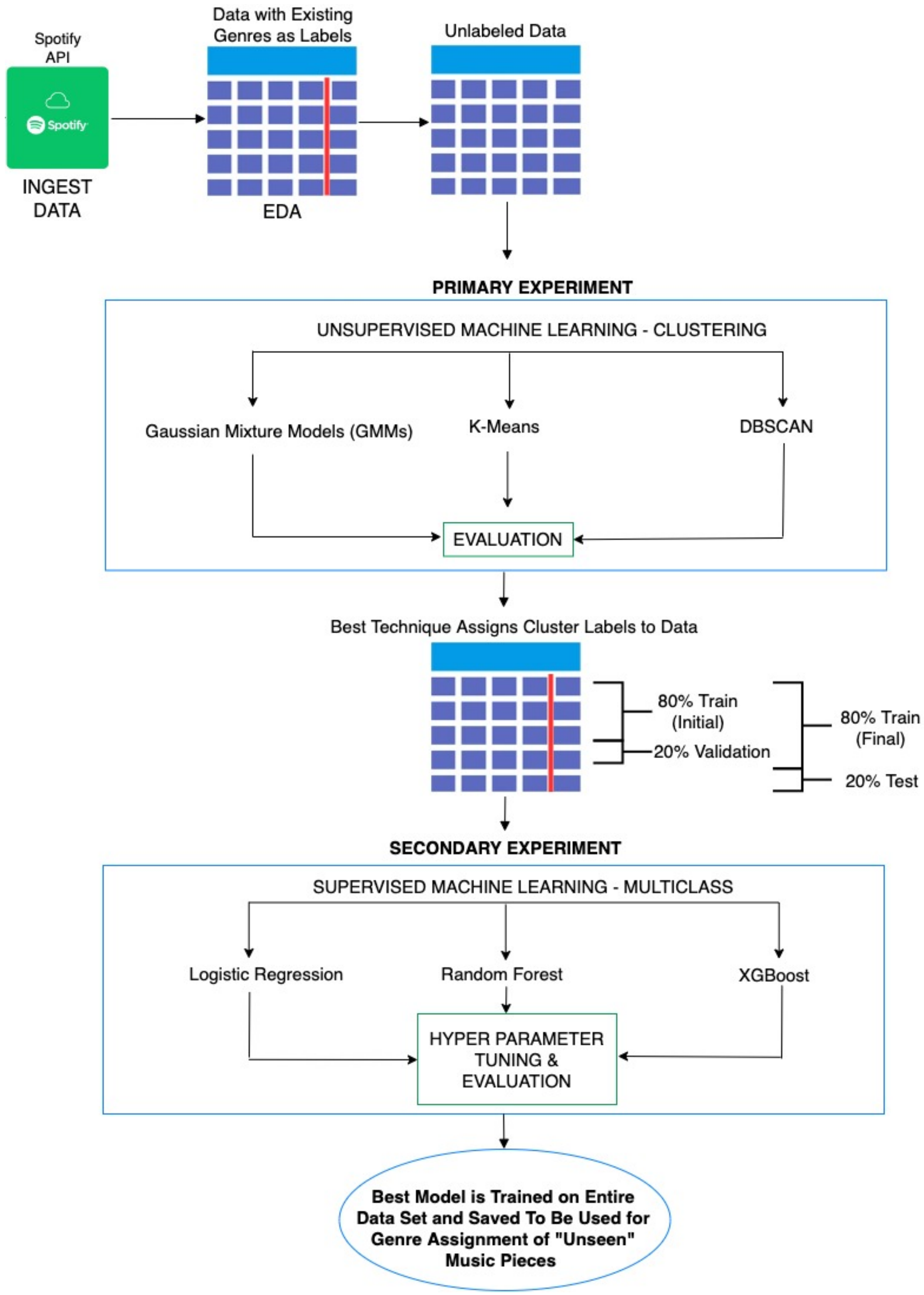

Spotify API
Spotify
INGEST DATA
Data with Existing Genres as Labels
EDA
Unlabeled Data
PRIMARY EXPERIMENT
UNSUPERVISED MACHINE LEARNING - CLUSTERING
Gaussian Mixture Models (GMMs)
K-Means
DBSCAN
EVALUATION
Best Technique Assigns Cluster Labels to Data
80% Train (Initial)
20% Validation
80% Train (Final)
20% Test
SECONDARY EXPERIMENT
SUPERVISED MACHINE LEARNING - MULTICLASS
Logistic Regression
Random Forest
XGBoost
HYPER PARAMETER TUNING & EVALUATION
Best Model is Trained on Entire Data Set and Saved To Be Used for Genre Assignment of "Unseen" Music Pieces

Figure 5: Flowchart of methods

Unsupervised Machine Learning - Clustering

The primary experiment as shown in Figure 5 involves running clustering algorithms to assign individual music pieces into well-defined genres. In order to implement clustering, we first normalize our features. Since we work with the motivation that existing Spotify labels of genres in our data are not informative of technical attributes, we must re-classify without the assumption that the ideal number of genres is six. DBSCAN and OPTICS clustering methods do not require inputs of the number of clusters and so we initially investigate the utility of both these methods. However, these methods are extremely resistant to outliers and do not predict well on clusters with high variance in densities. When used on this dataset, they leave up to 30,000 songs unclassified when they produce less than 10 clusters. Alternatively, the number of clusters goes up to 300 when ensuring most points are assigned to a cluster. The many failings of these methods make it an unsuitable pursuit for the purpose of our problem.

We then switch to K-means clustering and Gaussian Mixture Models to cluster our data and help explain it's distribution in the high dimensional space we have. Both approaches take a different perspective to identifying patterns in the data, and the motivations for selecting these methods is discussed in the next paragraph. We tune both types of clustering and evaluate their performance based on the Davies-Bouldin score and Silhouette score, finally picking the K-means clustering algorithm with 5 clusters as a final classification process.

The primary advantage of moving to K-means [20] and Gaussian Mixture Models is to avoid the large number of clusters and unclassified points. Gaussian Mixture Models [13] have the added advantage of assigning predicted probabilities to each observation's likelihood of falling into a particular cluster, unlike K-means. The reason this could be useful is that genres can overlap the same way real music genres sometimes do. Eliminating the uncertainty that comes from confidence scores deprives our model of this nuanced representation of music genres. On the other hand, K-means produces distinct clusters when the patterns that exist are globular or spherical. K-means can scale-up for more genres, songs and numbers of clusters with less computational power required, in comparison to Gaussian Mixture Models. Gaussian Mixture models are probability distributions and not algorithmic assignments. Hence, they come with the assumption that each cluster has a Gaussian distribution, which we verify. For these reasons, we examine both K-means and Gaussian Mixture Models.

Before we delve into using these models, we identify the metrics we will use to evaluate them. In order to gain a good clustering, we hope to increase the distance between clusters and decrease the distance between points within a cluster. The first metric that is useful to do this is the Davies-Bouldin Index (DBI) [11], as mentioned above. This index measures the average similarity of each cluster with its most similar cluster. For our purposes, a DBI close to 0 means the clusters are distinct from each other and less dispersed within themselves. This is what we are interested in looking at. However, relying on a single metric can be naive, and so we also take into account the Silhouette score [12] of each model. This score also measures cluster similarity, but scores each data point from -1 to 1, where 1 represents a point being assigned to the right cluster and -1 indicates an incorrect clustering. Clustering that has an average score of 0 indicates largely overlapping clustering for most data points.

With these popular and well-defined metrics set for our evaluation, we first cluster our data using k-means. In order to determine the optimal number of clusters we use the elbow method, which plots the distortion (average of the squared distances from the cluster centers of the respective clusters), against k = $[2, 3, 4, 5, 6, 7, 8, 9]$. The optimal value found is k = 5, where the distortion is least (see figure 6). Once the data is clustered, each song is assigned new labels which correspond to these clusters i.e. new genres.

In order to explore a method with fuzzy clustering, we use Gaussian Mixture models next. Scikit-learn [14] defined this model as generalizing k-means clustering "to incorporate information about the covariance structure of the data as well as the centers of the latent Gaussians", which means, there is a probabilistic or "weighted" assignment. We tune this model to obtain the optimal number of clusters as 8, plotting the BIC (Bayesian Information Criterion) against the range of k values 2-9, as for k-means above. The BIC is lowest at k = 8 as shown in figure 7. We also tune the Gaussian model for the covariance type by evaluating the DBI and Silhouette score for each type of covariance matrices in the scikit-learn functionality [14]. Spherical was the best fit for our data. Our final Gaussian Mixture model had 8 clusters and these were assigned as labels to our data.

In order to evaluate both types of clustering we measure their DBI and Silhouette scores and compare them to each other. These scores are included in the results section below.

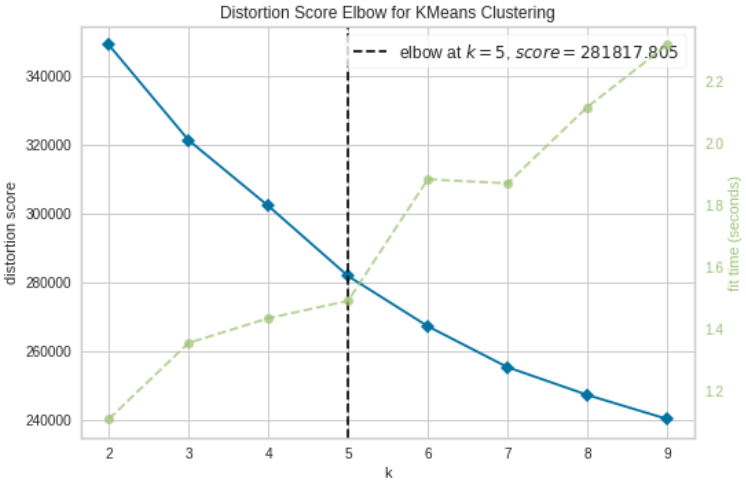


Figure 6: The elbow graph for finding the optimal number of clusters, k, for k-means clustering

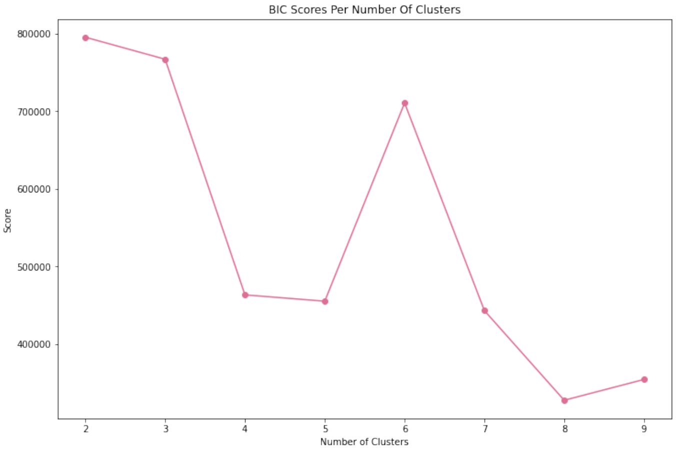


Figure 7: The BIC graph for finding the optimal number of clusters, k, for Gaussian Mixture clustering

	K-means Model	Gaussian Mixture Model
Davies-Bouldin Index (DBI)	2.166	1.999
Silhouette Score	0.105	0.09

Table 1: Evaluation Scores for K-means and Gaussian Mixture Model

Supervised Machine Learning - Multiclass Classification

To achieve our objective of developing a model-based approach of music genre assignment, it is necessary to build a model that learns how genres are assigned by the best clustering technique and can assign any music piece to genres based its technical attributes. Following best practice, the first iteration of this experiment was to assess the performance of a simple model. A multinomial logistic regression model was first implemented, after which a random forest model was implemented. Since we anticipate that this method can be replicated on the entire corpus of music pieces in Spotify's library, a random forest model was selected because of its strength in working with large data sets and interpretability [17]. Finally, XGBoost, a boosting algorithm, was selected because of its ability to reduce variance and improve the prediction power by training a sequence of weak models. To evaluate the performance of these models, it was important to select a metric that measures how a model correctly classifies music genres out of all predictions made for each genre, and measures what proportion of positive predictions of music genres were identified correctly for each genre. This metric is a harmonic mean of the precision and recall is defined as the F1 score or the Sorensen-Dice coefficient. The macro average F1 score is computed by taking the arithmetic mean of all per-genre F1 scores. This method treats all classes and since the number of genres or classes change over time and no genre is given a higher weighting or importance than the other, we used macro average F1 score to evaluate the above-mentioned supervised machine learning model. The model which scored the highest macro average F1 score on the validation data set was selected as the best model. This model (saved as a pickle file) is available to be used for genre assignment in lieu of the existing flawed approach.

5. Results

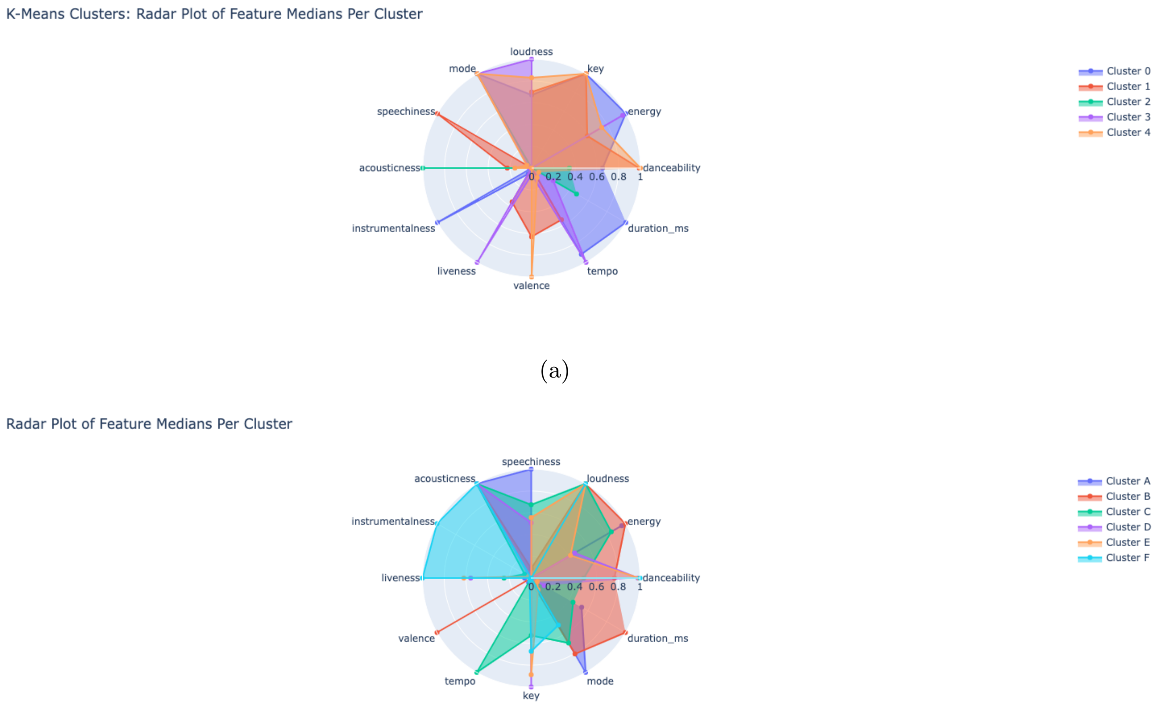
Unsupervised Machine Learning - Clustering

The results of our clustering methods warranted a discussion on which model is better in the context of our problem. As observable in Table 1, our k-means clustering has a slightly higher DBI than our Gaussian mixture model but the difference is quite negligible. On the other hand, the Silhouette score is very informative. The k-means model has a Silhouette score close to 1 as compared to the Gaussian Mixture model's score which is close to 0. This means that the Gaussian clusters are not very distinct in their assignment of songs to genres and create room for a significant lack in clarity on what the clusters mean. This fuzziness could have been viewed positively if the goal of our genre creation been different. Musical genres tend to overlap largely in musical theory and Gaussian Models account for that overlap and contain confidence scores for observations belonging in a particular genre. However, in order to be able to explain what each genre's technical attributes are, we prefer a harder clustering as is obtained from k-means. The scatter plots in Figures 9 show how both k-means and Gaussian clusters are easier to define than previous genres, and so, future songs belonging to these genres must be easier to create for practical purposes.

In the next section we talk about the results of the supervised learning model we trained on our data to predict the labels of our k-means clustering, which are representative of our new genres.

Supervised Machine Learning - Multiclass Classification

Cluster labels of each song generated by the k-means algorithm are appended to the data. To gain a cursory understanding of the performance of a classifier on the entire data set and to determine if hyperparameter tuning was necessary, we first split our data into training (80%) and test(20%) data sets. Logistic Regression algorithm (with initial hyperparameters of saga solver), which was the baseline model, is fit on standardized training data and the model was evaluated on how it generalizes to unseen standardized test data. We chose a saga solver because it supports L1 regularization and works well for multinomial regression of large data sets. The process was iterated for a Random Forest and an



(a)

(b)

Figure 8: Clusters produced by K-means (a) and Gaussian Mixture Models (b) are represented in terms of the mean values of each of the music features in these radar plots. Without seeing the music that composes these ”new genres”, we can appreciate some characteristics of the songs that belong to them in a single visual.

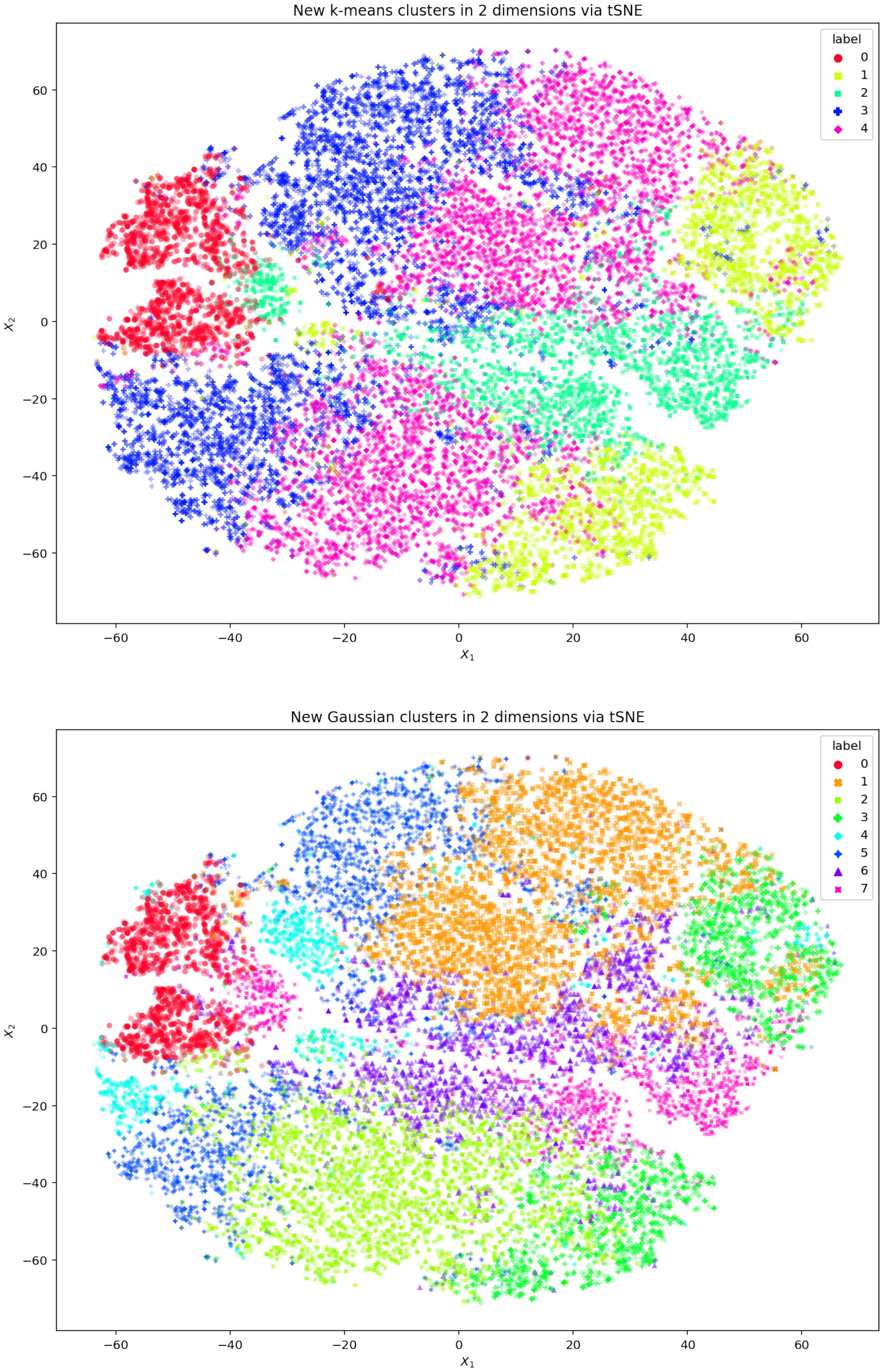


Figure 9: Clusters produced by K-means (a) and Gaussian Mixture Models (b) are represented in two dimensions using the t-SNE technique. Compared to original genres, both options of classifications look more distinguishable, even with features reduced to 2 dimensions.

XGBoost algorithms and their summaries can be found in the appendix. We found out from our cursory investigation that the logistic regression classifier achieved near perfect scores on all relevant evaluation metrics. Classification reports on the test data have been outline in table 2 below.

	Precision	Recall	F1-score	Support
cluster 0	0.970297	0.985915	0.978044	497.00000
cluster 1	0.997775	0.993355	0.995560	903.00000
cluster 2	0.987915	0.985930	0.986922	995.00000
cluster 3	0.994005	0.990766	0.992383	1841.00000
cluster 4	0.992719	0.994423	0.993571	2331.00000
accuracy	0.991320	0.991320	0.991320	0.99132
macro avg	0.988543	0.990078	0.989296	6567.00000
weighted avg	0.991350	0.991320	0.991329	6567.00000

Table 2: Classification Report of Logistic Regression Classifier

We visualized how well the logistic regression model classified the validation data set into clusters with a confusion matrix shown in figure 10 below. Confusion matrix of Random Forest XGBoost Classifiers have been outlined in the Appendix.

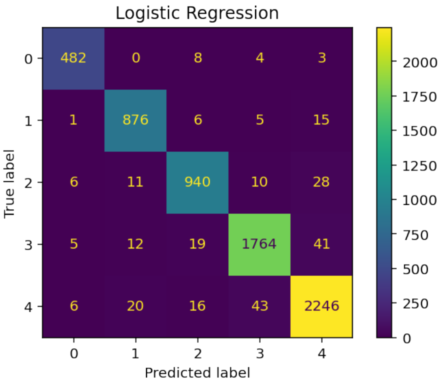


Figure 10: Confusion Matrix of Multinomial Classifier

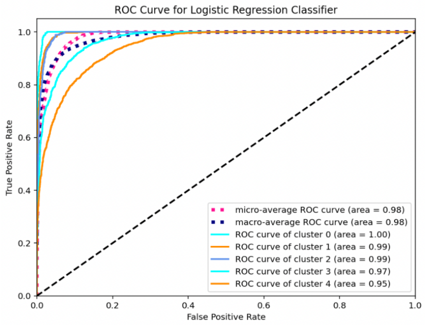


Figure 11: ROC Curves of Multinomial Classifier

We also computed the Area Under Curve (AUC) of the logistic regression classifier for each cluster. Receiver Operator Characteristics (ROC) curves of the logistic regression classifier has been outlined in figure 15. We noticed that evaluating the decision function of values in the feature space of the test data produces more accurate results than evaluating the predicted probability of observations belonging to clusters. Since Random Forest and XGBoost Classifiers are tree based classifiers, we were limited in using the predicted probabilities to plot ROC curves, and their results mirrored each other. Their ROC curves have been included in the appendix.

6. Conclusions

At the conclusion of this report, we have a working model for classifying new songs into technically-oriented genres in a way that informs the creation of future music. We add value to existing clustering of music into genres by using a variety of machine learning methods and exploring their outputs with a practical perspective. This is different from classifying songs into existing genres. Since our objective was to assign music pieces to well-defined genres, the k-means clustering algorithm was the best performing model. Even though the probabilistic genre assignment of the Gaussian-Mixture Model does not address our problem, it shows much promise in generating confidence scores for music pieces at the intersection of one or multiple genres. We recommend the scope of future work on this subject matter include building models and functions that map these confidence scores to new genres.

A limitation of this study is that some features like danceability and speechiness are confidence scores generated by Spotify's machine learning models. The uncertainty of these confidence scores and machine learning models are an integral part of the assumption we have made in our analysis. The trade-off with this limitation is that we now have access to a rich feature space with which to solve the problem of subjective music genre assignment. We recommend that future work should be carried to shed light on machine learning models used by digital streaming platforms, although we do not have reason to suspicious of such models, there is the need to assess whether their confidence scores are consistent across all genres of music. Once music pieces were categorized into well-defined clusters, the goal of building a supervised machine learning model to learn how the data was clustered and classify existing and new music pieces into these new clusters or genres is a straightforward task. To conclude, we recommend that this approach be replicated with a larger corpus of music from all genres so that the contentious issue of subjective genre assignment can be put to bed.

Roles

- Clarissa: The presenter is responsible for collecting the pieces of information to build out a story that communicates the project findings in the correct level of detail and tailored to the target audience. The presenter must highlight the relevance of the project findings in the broader landscape of research in the industry of choice and produce the presentation slides and the github summary.

- Preet: The writer's main responsibility is to construct the project report that communicates in detail the modeling process, including decisions made and the reasoning behind, project outcomes, motivations and conclusions. The writer is also responsible for note taking during meetings, tracking decisions, and mediating discussions on any modeling choices that are decided by the team.

- John: The primary coder will ensure the full cycle of ML is carried out correctly in clean python scripts / jupyter notebooks with auxiliary notes and descriptions. The primary coder will also generate plots that describe outcomes of the model and their performance.

- Rashaad: The checker is responsible for making sure not only the coding of elements are carried out correctly but also the report and presentation reflect consistently the project findings.

References

[1] H. Soltau, T. Schultz, M. Westphal and A. Waibel, “Recognition of Music Types”, In Proc. a/ IEEE ICASSPYB, pp.1137-1140, 1998.

[2] R. Dannenberg, J. Foote, G. Tzanetakis, and C. Weare, “Panel: new directions in music information retrieval,” in Proc. Int. Computer Music Conf., Habana, Cuba, Sept. 2001.

[3] F. Pachet and D. Cazaly, “A taxonomy of musical genres,” in Proc. Content Based Multimedia Information Access (RIAO), Paris, France, 2000.

[4] F. Pachet, J.J. Aucouturier, A. La Burthe, A. Zils, and A. Beurive, “The cuidado music browser: an end-to-end electronic music distribution system,” Multimedia Tools Applicat., 2004, Special Issue on the CBMI03 Conference, Rennes, France, 2003.

[5] X. Shao, C. Xu, and M. Kankanhalli, “Unsupervised classification of musical genre using hidden Markov model,” in Proc. IEEE Int. Conf. Multimedia Explore (ICME), Taibei, Taiwan, 2004, pp. 2023–2026.

[6] W. Tsai, D. Bao, “Clustering Music Recordings Based on Genres” in 2010 International Conference on Informational Science and Applications, Taiwan, 2010

[7] D. Temperley and E.W Marven, “Pitch-Class Distribution and the Identification of Key” in Music Perception by University of California Press, 2008

[8] G. James, D. Witten, T. Hastie, R. Tibshirani, “An Introduction to Statistical Learning with Applications in R” , Springer, Second Edition, 2014, page 499.

[9] G. Hinton, S. Roweis “Stochastic Neighbor Embedding” in Advances in Neural Processing Systems 15, 2003

[10] L. van der Maaten, G. Hinton “Visualizing data using t-SNE” in Journal of Machine Learning Research 9(2605):2579-2605, 2008

[11] A. K. Singh, S. Mittal, P. Malhotra and Y. V. Srivastava, ”Clustering Evaluation by Davies-Bouldin Index(DBI) in Cereal data using K-Means,” 2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC), 2020, pp. 306-310, doi: 10.1109/ICCMC48092.2020.ICCMC-00057.

[12] K. R. Shahapure and C. Nicholas, ”Cluster Quality Analysis Using Silhouette Score,” in 2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA), Sydney, Australia, 2020.

[13] McLachlan, G.J. and Rathnayake, S. (2014), On the number of components in a Gaussian mixture model. WIREs Data Mining Knowl Discov, 4: 341-355. https://doi.org/10.1002/widm.1135

[14] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, 2011, pp. 2825-2830.

[15] N. Scaringella, G. Zoia, and D. Mlynek, “Automatic genre classification of music content: A survey,” IEEE Signal Processing Magazine, 2006 vol. 23, no.2, pp. 133–141.

[16] G. Tzanetakis and P. Cook, “Musical genre classification of audio signals,” IEEE Trans. Speech and Audio Processing, vol. 10, no. 5, pp 293–302, Jul. 2002.

[17] J. Mourtada, S. Gaiffas and E. Scornet, “AMF: Aggregated Mondrian Forests for Online Learning” IEEE Trans. Speech and Audio Processing, Big Data Research, 2015 vol. 3, pp 1-2.

[18] Algorithmic clustering of music based on string compression. In Computer Music Journal on (Vol.28, no. 4, pp. 49-67). IEEE1-2.

[19] Peng, W., Li, T., Ogihara, M. (2007). Music Clustering with Constraints. In ISMIR (pp. 27-32)

[20] Jin X., Han J. (2011) K-Means Clustering. In: Sammut C., Webb G.I. (eds) Encyclopedia of Machine Learning. Springer, Boston, MA. https://doi.org/10.1007/978-0-387-30164-8_425

[21] Sklearn.metrics: 3.3.2.9. Precision, recall and F-measures
https://scikit-learn.org/stable/modules/model_evaluation

Appendix

0.1 Data Dictionary

- Tempo: The overall estimated tempo of a music piece in beats per minute (BPM). In musical terminology, tempo is the speed or pace of a given piece and derives directly from the average beat duration.

- Loudness: The overall loudness of a track in decibels (dB). Loudness is the quality of a sound that has a primary correlation with amplitude. Loudness values are averaged across the entire track and are useful in comparing the relative loudness of tracks. Values typically range between -60 and 0 db.

- Key: Integers map to pitches using standard Pitch Class notation. Although a technical attribute of music, keys are often detectable by the human ear. Temperley et al [7] show that listeners identify 3 key by monitoring the distribution of pitch classes in a piece and compare it to an ideal distribution for each key. For example 0 = C, 1 = C, 2 = D, and so on. If no key was detected the value is -1.

- Mode: Integers map to the type of scale from which its melodic content is derived. Major is represented by 1 and minor is 0.

- Speechiness: Detects the presence of spoken words in a track. The more exclusively speech-like the recording, the closer to 1.0 the attribute value. Values above 0.66 describe the tracks that are probably made of entire spoken words.

- Liveness: Detects the presence of an audience in the recording. Higher liveness values represent an increased probability that the track was performed live. A value above 0.8 provides strong likelihood that the track is live.

- Instrumentalness: Measures whether a music piece contains no vocals. "Ooh" and "aah" sounds are treated as instrumental in this context. The closer the instrumentalness value is 1.0, the greater likelihood the music piece contains no vocal content and vice versa.

- Danceability: Describes how suitable a music piece is for dancing based on a combination of music elements including tempo, rhythm stability, beat strength, and overall regularity. A value of 0.0 is least danceable and 1.0 is most danceable.

- Energy: Represents the perceptual measure of intensity and activity. Perceptual features contributing to this attribute include dynamic range, perceived loudness, timbre, onset rate, and general entropy. Typically, energetic tracks feel fast, loud, and noisy. For example, death metal has high energy, while Bach scores low on the scale.

- Acousticness: A confidence measure from 0.0 to 1.0 of whether the track is acoustic. 1.0 represents high confidence the track is acoustic and vice versa.

- Valence: A measure from 0.0 to 1.0 describing the musical positiveness conveyed by a music piece. 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)

- Duration: The duration of a music piece in milliseconds.

Table 3: Classification Report of Random Forest Classifier

	Precision	Recall	F1-score	Support
cluster 0	0.964000	0.969819	0.966901	497.00000
cluster 1	0.953210	0.970100	0.961581	903.00000
cluster 2	0.950455	0.944724	0.947581	995.00000
cluster 3	0.966046	0.958175	0.962094	1841.00000
cluster 4	0.962709	0.963535	0.963122	2331.00000
accuracy	0.960560	0.960560	0.960560	0.96056
macro avg	0.959284	0.961270	0.960256	6567.00000
weighted avg	0.960579	0.960560	0.960553	6567.00000

Table 4: Classification Report of XGBoost Classifier

	Precision	Recall	F1-score	Support
cluster 0	0.964000	0.969819	0.966901	497.00000
cluster 1	0.953210	0.970100	0.961581	903.00000
cluster 2	0.950455	0.944724	0.947581	995.00000
cluster 3	0.966046	0.958175	0.962094	1841.00000
cluster 4	0.962709	0.963535	0.963122	2331.00000
accuracy	0.960560	0.960560	0.960560	0.96056
macro avg	0.959284	0.961270	0.960256	6567.00000
weighted avg	0.960579	0.960560	0.960553	6567.00000

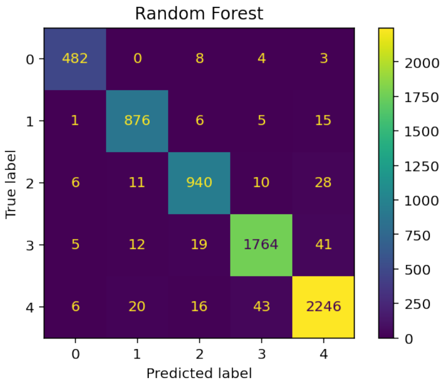


Figure 12: Confusion Matrix of Random Forest Classifier

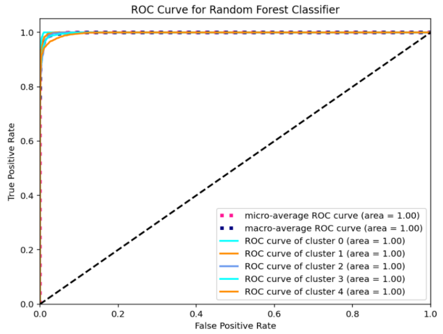


Figure 13: ROC Curves of Random Forest Classifier

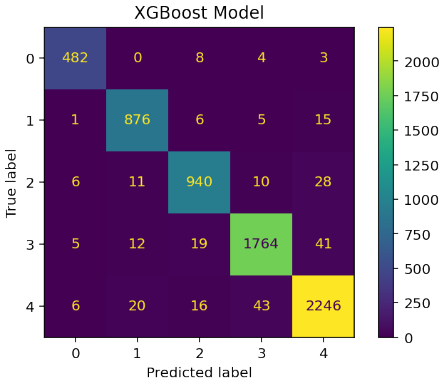


Figure 14: Confusion Matrix of XGBoost Classifier

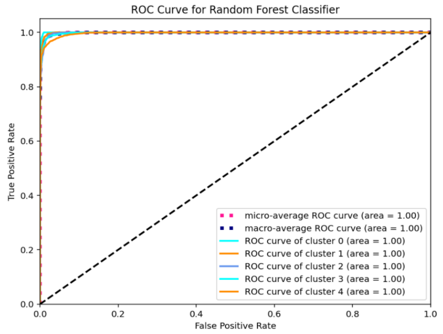


Figure 15: ROC Curves of XGBoost Classifier