SIADS-696: Milestone II
Andre Onofre, Bowie Liu

Analyzing Musical Evolution and Predicting Song Virality in Pop Music

*** Project Report ***

1. Introduction

Pop music, characterized by its catchy melodies and widespread appeal, dominates current music charts, streaming platforms, and social media. Given its influence, understanding the elements that drive a pop song to virality and analyzing the evolutionary trends within the genre are crucial for artists looking to capitalize on these dynamics. This project aims to provide artists with actionable insights into their music's impact and potential trajectories for success, empowering them to adapt to the ever-evolving musical landscape.

Our research addresses two primary challenges: (**a**) identifying the changing patterns of audio features in pop music over time, and (**b**) predicting which songs will achieve high popularity or 'go viral.' By tackling these problems, we aim to offer tools that enhance artists' creative and strategic decisions, potentially transforming how music is produced and promoted.

Motivated by the transformative potential of data-driven insights in music production, our project implemented both supervised and unsupervised learning methods. We utilize unsupervised learning to cluster songs based on key audio features like danceability and valence as well as analyze the shifts across different decades to map the genre's evolution. For virality prediction, our supervised models assess the potential of a song becoming a hit based on its audio characteristics.

Compared to related projects, our approach uniquely combines historical analysis with future prediction using both unsupervised and supervised learning. This dual focus provides artists with a comprehensive toolkit, offering insights into both past trends and future virality. Our clustering reveals distinct shifts in pop music features over the decades, while our predictive models accurately forecast a song's potential popularity, giving artists a valuable resource for creating hit songs.

These findings not only highlight the dynamic nature of pop music but also provide foundational knowledge for predicting future trends, offering a significant advantage to artists and producers in a highly competitive industry.

2. Related Work

One similar application can be found here: Spotify for Artists. This website focuses on providing artists with detailed analytics in regards to their streams, listener demographics, and geographical distribution. However, it does not provide insights on specific features about the songs (i.e. danceability, energy, key, etc.) (aka musicality) and the genre evolution over time. Additionally, Spotify for Artists does not offer features for predicting song virality based on audio features. The current analytics are tailored to marketing and promotional strategies rather than delving deep into the music.

Another similar application can be found on this website: Solving Spotify Multiclass Genre Classification Problem. This project aims to develop a model that can accurately predict the genre of a Spotify track. While both our projects center on music genre analytics, our approach includes analyzing the musical evolution within a genre, rather than classifying tracks into predefined genres. Furthermore, while the example project classifies tracks across all possible genres, our focus is exclusively on pop music, which is currently the most dominant genre in the industry.

Last project that has similar application as our project is Predicting Song Popularity from Spotify Dataset, a Kaggle project that uses a regression model to predict the popularity score of Taylor Swift's songs. While our project also aims to predict popularity, we employed a classification model to determine whether a song qualifies as a 'hit.' Additionally, our model is designed to be applied to songs by any artist, not limited to just one.

3. Data Source

We used Spotify API to query audio features using the types of requests described on Table 1.

#	Description	API Call
1	Get Token to Access Content	**Request:** URL: https://accounts.spotify.com/api/token Headers: Authorization and Content Type Payload: Grant Type = Client Credentials **Response: Access Token** (Details: Client Credentials Flow)
2	Get Pop Track	**Request:** URL: https://api.spotify.com/v1/search Header: Authorization Token Parameters: q='genre:pop'; type='track' **Response: Track IDs** (Details: Search for Item)
3	Retrieve Audio Features for a list of Tracks	**Request:** URL: https://api.spotify.com/v1/artists/audio-features Header: Authorization Token Parameters: Track IDs **Response: Features for Tracks** (Details Get Several Tracks)
4	Get Top Artists by Genre	**Request:** URL: https://api.spotify.com/v1/search Headers: Authorization Token Parameters: q: "genre", type: "artist", limit **Response: List of Top Artists** (Details: Search for Item)
5	Fetch Top Tracks for Artist	**Request:** URL: https://api.spotify.com/v1/artists/{artist-id}/top-tracks Header: Authorization Token Parameters: Country **Response: List of Top Tracks for the Artists (Track IDs)** (Details: Get Artist's Top Tracks)
6	Retrieve Audio Features for a list of Tracks	**Request:** URL: https://api.spotify.com/v1/artists/audio-features Header: Authorization Token Parameters: Track IDs **Response: Features for Tracks** (Details: Get Several Tracks)

Table 1

In order to build the dataset, first the token was generated (# 1), then a list of pop tracks IDs was retrieved (# 2) and finally the features per track were gathered (# 3). This method provided 1300 records due to limits imposed by Spotify API. In order to have additional samples, we used a publicly available dataset (Kaggle Spotify Tracks Dataset) which provided an additional 1000 records. Merging the two datasets described above, the raw dataset was composed then, by the following columns and containing 2304 pop tracks information.

Even with 2304 records, we had only 5% of records which corresponded to viral tracks (popularity greater than 80), so other extract was done from Spotify. In order to fix the imbalance classes for our supervised learning model, a list of top artists for 'Pop' genre was created (# 3). With the list of top artists, the top tracks were fetched (# 4) and finally the audio features for each track were retrieved (# 5). This final extract (so called "Class

1 Enlargement Dataset") generated additional 190 'viral' tracks, making the final dataset with about 12% 'viral' tracks. The final dataset contains approximately 2,500 records spanning from 1942 to 2024, with over around 90% of the records dated after the 2000s. It's important to clarify that the additional ''viral' tracks were not included in our unsupervised method since the 'Popularity' feature, which is not an audio feature, was not utilized in our unsupervised learning models.

Finally, preprocessing was used for some of the variables:

- Release Date: convert from text to pd.datetime for easier date manipulation
- Popularity: to numeric value
- Explicit: from True and False to 1 and 0
- Duration_ms: from milliseconds to seconds

4. Feature Engineering

In order to finalize the features for supervised and unsupervised learning, it was created the **target variable** ("Virality") - tracks with popularity above 80 received "1" (viral), and 0 otherwise. The 'Release Year' was extracted from the release_date for easier date manipulation. Numerical variables were **normalized**. Missing values were filled in using the mean value for the corresponding numerical variable. Finally, any record still containing NAN was dropped. The final list of variables is displayed below in Table 2.

Feature Name	Description
Artist	Artist Name
Name	Name of the song (Track Name)
Album	Album containing the track
Release Date	Date track was released
Popularity	Numbers between 80 and 100 indicates very high popularity
Duration_s	The duration of the track in milliseconds
Explicit	Whether or not the track has explicit lyrics
Danceability	A value of 0.0 to 1.0 that indicates how suitable the track is for dancing
Energy	Measure the intensity based on dynamic range, perceived loudness, etc.
Key	The key the track is in, which can convey mood
Mode	The modality of the track where Major is represented by 1 and Minor is 0
Valence	Describes the musical positiveness conveyed by a track
Tempo	The estimated beats per minute (BPM) of a track
Loudness	The overall loudness of each track measured in dB
Liveness	Detects the presence of an audience in recording on a scale of 0.0 to 1.0
Instrumentalness	Predicts whether a track contains no vocals on a scale of 0.0 to 1.0
Acousticness	A measure from 0.0 to 1.0 on whether the track is acoustic
Speechness	Detects presence of spoken words in a track on a scale of 0.0 to 1.0

Time Signature	The time signature (meter) is a notational convention to specify how many beats are in each bar (or measure). The time signature ranges from 3 to 7.
Virality	1 (= Viral Track) or 0 (= Non Viral Track)
Release Year	Extract 'Year' from release_date for easier date manipulation

Table 2

5. Supervised Learning

5.1 Methods Description

Step #	Name	Description
1	Crete X and y	Divide the dataset in independent variables (X) **(*)** and dependent variable (y) ("Virality")
2	Create Training and Test sets	Divide the dataset in Training Set (80% of the samples) and Testing Set (20% of the samples)
3	SMOTE	As only 12% of the dataset samples are for class 1 ('viral'), SMOTE was used to balance the dataset and get better results during training and testing
4	Define Classifiers	4 types of classifiers were used. Please, see details on section 5.1.1 below
5	Define Grid Search	For each of the supervised learning models, it was defined a range of hyper-parameters to be tested ("grid search") to find the best classifier and best set of hyper-parameters for our use case
6	Fit Classifiers	Using the hyper-parameters grid and the training set, all models were fit
7	Evaluation	The models evaluation was done considering their accuracy, F1 and ROC-AUC scores (see details on Chapter 5.2)
8	Elect best classifier	Based on the scores results from the evaluation, the best classifier and its best set of hyper-parameters were found. Such combination of best classifier with its best set of hyper-parameters became the "Best Model" which was used for the subsequent analysis (see Chapters 5.2 and 5.3)

Table 3

(*) List of Independent Variables: danceability, energy, key, loudness, mode, acousticness, instrumentalness, liveness, valence, tempo, duration and explicit. For detailed descriptions about each variable, please refer to Chapter 4.

5.1.1 Classifiers

Model	Description
Logistic Regression	A statistical method used to model the probability of a binary outcome based on one or more predictor variables, employing a logistic function to map input variables to the probability of the output belonging to a particular class [1]. It fits well to our case as we are working to predict a binary outcome ("virality") based on inputs variables (described above)
Random Forest	A machine learning algorithm that constructs a multitude of decision trees during training and outputs the mode of the classes (classification) of the individual trees [2]. It is a very popular classifier as it avoids overfitting due to its ensemble mechanisms, so it helps to get a good classifier without overfitting.

K-Neighbors	Simple and effective when dealing with small to medium datasets (which is our case) and where decision boundaries are non-linear.
Support Vector Machine	Versatile in handling both linear and non-linear classification tasks through the use of kernel functions and robustness against overfitting, especially in cases of small to medium-sized datasets (which is our case)

Table 4

5.2 Supervised Evaluation

5.2.1 Overall Results

We compared the models based on their F1, Accuracy and ROC-AUC in order to cover the model performance in multiple aspects. F1 score combines precision and recall into a single metric, providing a balanced evaluation by considering both false positives and false negatives, which is crucial in scenarios with imbalanced class distributions. Accuracy calculates the arithmetic mean of sensitivity and specificity, offering a balanced assessment of the classifier's performance across different classes. ROC-AUC provides a comprehensive measure of classifier performance by considering the area under the Receiver Operating Characteristic curve, weighted by class distribution, making it robust to imbalanced datasets.

The Table 5 below summarizes the hyper-parameters grid searches with the proper metrics. 5-fold cross validation was used and the scores below are shown with its mean metric across the multiple cross-validation folds, along with the standard deviation of the metric (in parentheses).

Logistic Regression Parameters	**Tested Range / Options**	**Accuracy**	**F1**	**ROC-AUC**	**Best Grid (Highest Score)**
Inverse of regularization strength ('C')	0.001, 0.01, 0.1, 1.0, 10	0.733 (0.068)	0.716 (0.092)	**0.830 (0.085)**	C: 10 Penalty: L2
Penalty ('penalty')	None: no penalty used L1: Lasso regularization L2: Ridge regularization Elasticnet: L1 and L2 used				

K-Nearest Neighbors Parameter	**Tested Range / Options**	**Accuracy**	**F1**	**ROC-AUC**	**Best Grid (Highest Score)**
Number of Neighbors ('n_neighbors')	1-20	0.928 (0.022)	0.929 (0.023)	**0.972 (0.019)**	n_neighbors: 15 weights: 'distance' leaf_size: 10 p: 1
Weight function used in prediction ('weights')	Uniform Distance				
Leaf Size ('leaf_size')	1, 5, 10				
Power parameter ('p')	'l1': manhattan_distance 'l2': euclidean_distance				

SVC Parameter	**Tested Range / Options**	**Accuracy**	**F1**	**ROC-AUC**	**Best Grid (Highest Score)**

Regularization parameter ('C')	0.1, 1, 10	0.921 (0.019)	0.923 (0.022)	**0.973 (0.017)**	C: 10 gamma: auto Kernel: 'rbf'
Kernel coefficient ('gamma')	'scale": 1 / (n_features * X.var()) 'auto': 1 / n_features				

Random Forest Parameters	**Tested Range / Options**	**Accuracy**	**F1**	**ROC-AUC**	**Best Grid (Highest Score)**
Number of trees in the forest ('n_estimators')	10-50	0.940 (0.007)	0.939 (0.007)	**0.986 (0.005)**	n_estimators: 40 max_depth: None min_samples_split: 3 min_samples_leaf: 2 max_features: 'sqrt'
Max depth of the tree ('max_depth')	3, 5, None (expanded until all leaves are pure)				
Min number of samples to split a node ('min_samples_split')	1, 2, 3				
Min number of samples to be a leaf ('min_samples_leaf')	1, 2, 3				
Number of features to consider when looking for the best split ('max_features')	- sqrt(Number of Features) - log2(Number of Features) - Total Number Features				

Table 5

As it can be noticed on the tables above, the best classifier in all scores is the 'Random Forest' with the hyper-parameters grid indicated on the table above. From now on, such a classifier with its best grid will be called "Best Classifier" and it will be used for the subsequent analysis.

5.2.2 Feature Importance

Figure 1 shows the importance of each feature with regards to its contributing to prediction success and failure. The top 4 features are valence, explicit, acousticness, and energy.

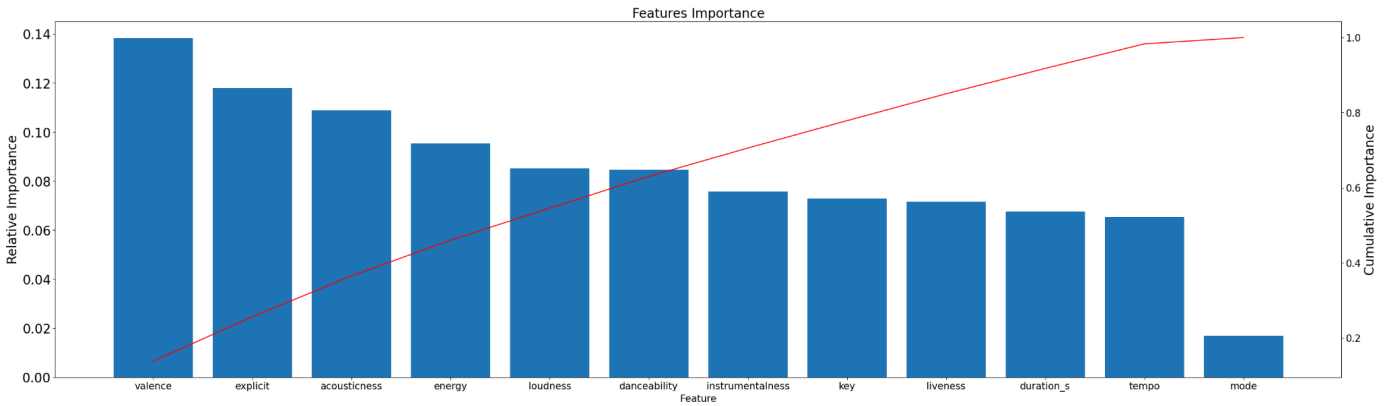

Figure 1

5.2.3 Sensitivity Analysis

Sensitivity analysis was done using the hyper-parameter Number of trees in the forest ('n_estimators') and the result is shown on Figure 2.

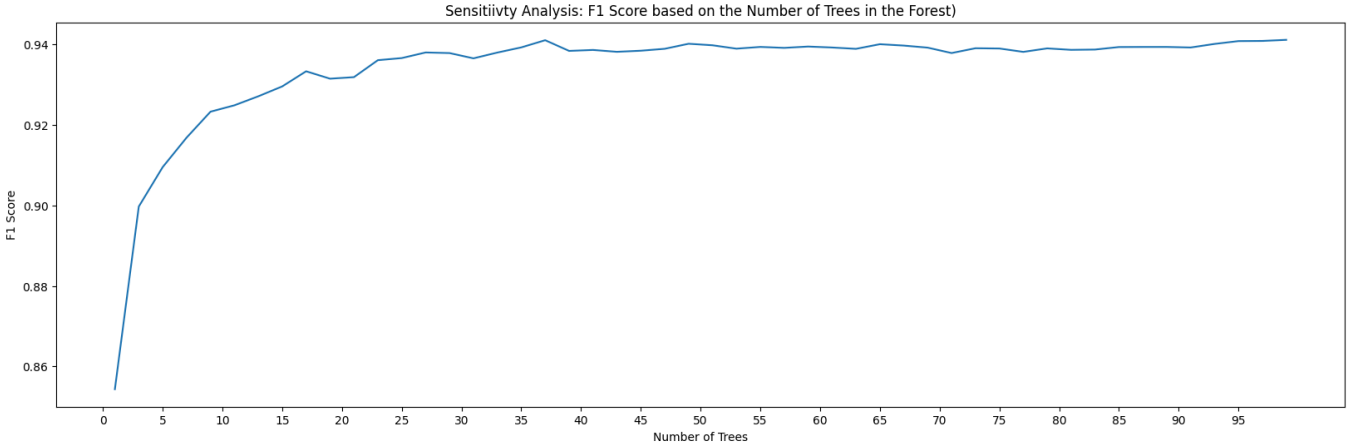

Figure 2

As per the chart above, the model performance is reasonably sensitive to the number of trees until around 30-40 trees (F1 score goes from 0.85 to 0.94). After that point, even adding more trees, the performance almost doesn't change.

5.2.4 Important tradeoffs

As it can be seen on Figure 3, the maximum F1 (0.6) is around the threshold 0.5. Smaller threshold leads to higher recall and smaller precision. Greater threshold leads to higher precision and small recall.

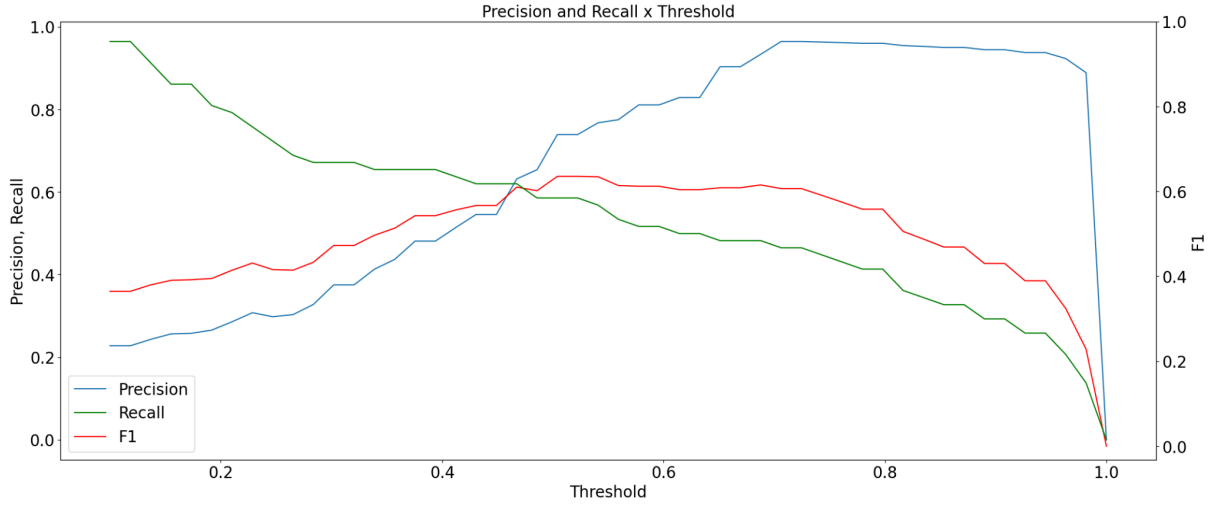

Figure 3

5.3 Failure Analysis

On the test set, overall, there were 426 true negatives, 34 true positives, 12 false positives and 27 false negatives. For False Positive failures (12), it is possible to notice that for half of the failures, the probability calculation for class 0 (not viral) and class 1 (viral) are around 0.45 and 0.55, respectively. This leads to the suspicion that these items are borderline samples. Adjusting the cutover to around 0.45 should reduce the number of false positives. For False Negative failures (27), we checked if the records were outliers or if the predictions were close to the border (e.g. 0.45), but it was not the case. So, our hypothesis is that, because of the dataset imbalance (many more samples for class 0), the classifier is biased towards the majority class.

5.4 Discussion

5.4.1 What did you learn?

We learned that when we have class imbalance, a tool called “Synthetic Minority Oversampling Technique” (SMOTE) can be used to create synthetic samples for the minority class and “balance” the dataset classes. Further, I learned that there are variants for the algorithm as follows:

- Borderline SMOTE: Borderline samples will be detected and used to generate new synthetic samples [6]
- SVM SMOTE: use Support Vector Machine algorithm to detect sample to use for generating new synthetic samples [7]
- ADASYN: Adaptive Synthetic algorithm. It generates a different number of samples depending on an estimate of the local distribution of the class to be oversampled [8]
- KMeans SMOTE: Apply a KMeans clustering before to over-sample using SMOTE [9]

5.4.2 Surprise

We were surprised with the number of trees necessary to have the best results with the Random Forest Classifier. Although the standard configuration in Sciki-Learn is 100 trees, it was necessary much less trees (40) to have good results in our model.

5.4.3 Challenges

The most challenging aspect of this supervised learning project was to find ways to improve F1 score. At first, there were only 5% of records for class 1 (viral track) which led to very low F1 (i.e. 0.2). Even with intense work to calibrate the classifier hyper-parameters, the results continued poor. After extracting additional samples from Spotify for viral tracks, and achieving around 12% of class 1 samples and using SMOTE, F1 improved to 0.64.

5.4.4 How could you extend your solution with more time/resources?

If there was more time and resources, we would collect more samples of class 1 tracks in order to improve F1 score even more. In order to illustrate the possibility to improve F1 score with additional class 1 samples, we modified the threshold to consider a track viral from 80 to small values so that we had additional viral tracks on the dataset. With such an approach, the best model was retrained and F1 scores were captured on Table 6.

Threshold (Popularity Greater than…)	% Class 1 Samples	F1 (Test Set)
80	12%	0.64
75	21%	0.72
70	32%	0.75
65	47%	0.81

Table 6

It can be noticed that with a more balanced dataset, F1 score improves from 0.64 to 0.81, confirming our hypothesis that low F1 scores were due to class imbalance.

5.5 Ethical Considerations

One ethical consideration is about voluntary participation. Although this is an academic project, we don’t have explicit written approval from the artists to use information about their songs. It is expected that this scenario is legally covered by Spotify as it exposes such information using Spotify APIs.

6. Unsupervised Learning

6.1 Methods Description

In our project, we decided to cluster songs based on their audio features and observe how these clusters evolve over time, capturing shifts in pop music evolution. To select the best unsupervised learning method, we first optimized each model's parameters to capture their highest evaluation scores. Based on these scores, we chose the best model to perform clustering on pop music tracks over the decades. Due to limited data from

earlier decades on Spotify, we combined records from the 2000s and 2010s and compared their combined clusters to pop music in the 2020s. After establishing our best-performing model, we calculated the shift in centroid movement between the clusters of different decades. This approach provided insights into how pop music has evolved in recent years compared to earlier periods. We explored various clustering models and their evaluation metrics before concluding that using t-distributed Stochastic Neighbor Embedding (t-SNE) with K-means clustering provided the most meaningful insights for our data.

6.1.1 Model Exploration

We employed several dimensionality reduction and clustering methods to analyze pop music evolution. t-SNE was selected for its ability to handle complex, non-linear relationships, making it ideal for visualizing high-dimensional audio data and identifying subtle shifts and trends over time. Principal Component Analysis (PCA) was used for its efficiency in reducing dimensionality while preserving maximum variance, helping us understand the primary factors driving musical evolution. Hierarchical Clustering was utilized to reveal intrinsic hierarchical structures within the data, providing a comprehensive view of cluster relationships and uncovering nuanced patterns. Despite its computational intensity, it was beneficial for understanding the granularity of musical evolution. K-means Clustering was chosen for its simplicity and effectiveness in creating well-defined clusters, particularly suitable for large datasets. Its iterative adjustment of cluster centroids allowed us to identify optimal groupings of songs based on audio features, facilitating a straightforward analysis of musical styles and their evolution. To pick the best-performing model, we will combine dimensionality reduction methods and clustering techniques. However, we must first discuss the feature representation and hyperparameter tuning for the models to ensure optimal performance.

6.1.2 Feature Representations

We selected audio features such as danceability, duration, energy, tempo, loudness, liveness, instrumentalness, and acousticness to characterize the musicality of songs. Features like popularity, virality, and release date were excluded from our unsupervised learning to focus on clustering based on musical attributes rather than external factors. We also omitted the key feature, as the mode feature already provides insights into the song's mood. Speechiness was excluded because it focuses on spoken words, which are less relevant to our analysis of musical elements. However, we included the explicit feature as it helps to understand the emotional tone of the song, adding depth to our analysis of its characteristics.

6.1.3 Hyperparameter Tuning

We employed several techniques to determine the optimal parameters for our clustering and dimensionality reduction methods:

- **PCA**: Cumulative explained variance to retain the most informative features while reducing dimensionality.

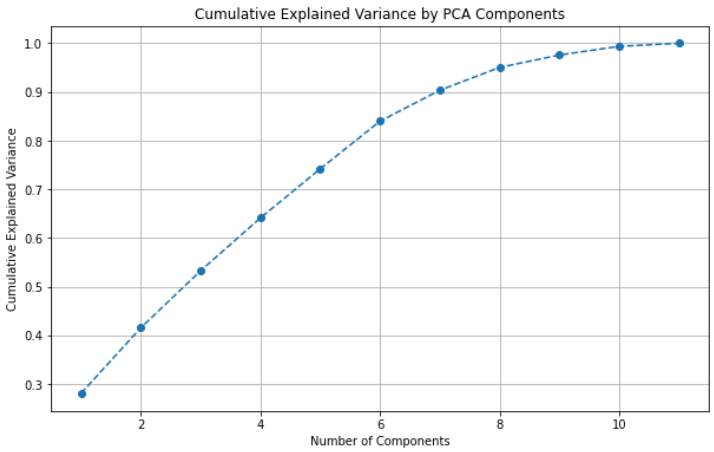

Figure 4

- **K-Means and Hierarchical Clustering**: The Elbow Method and the Silhouette Score to identify the ideal cluster count.

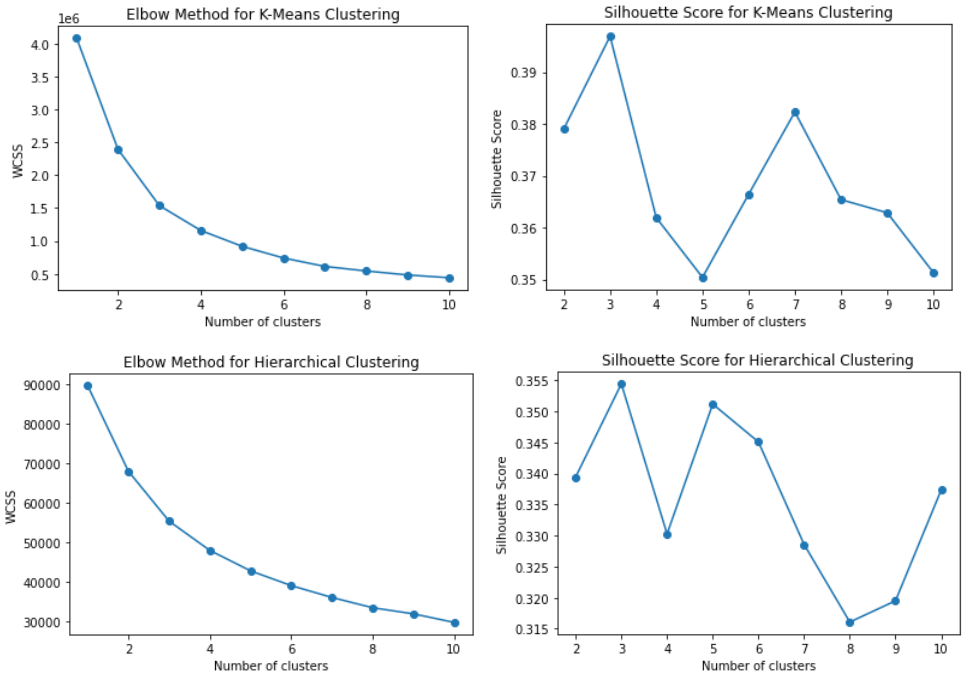

Figure 5

- **t-SNE**: Experimentation with various perplexity values to fine-tune the balance between local and global data aspects.

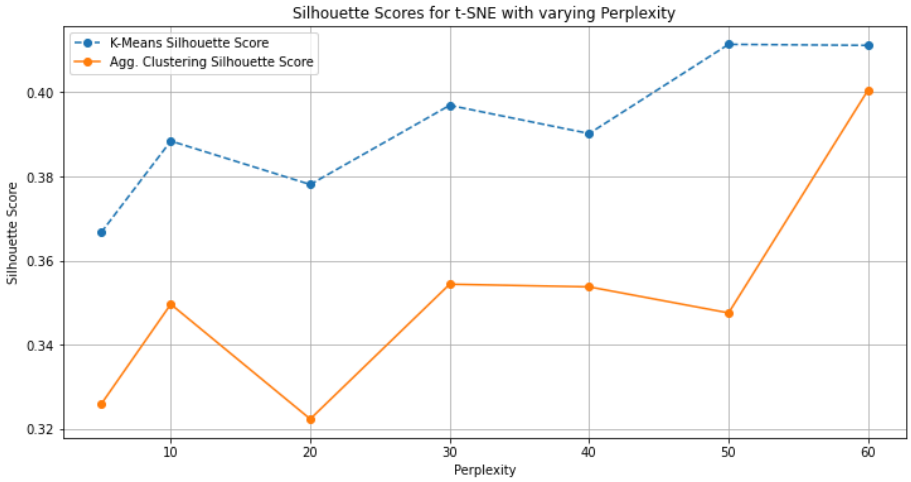

Figure 6

Optimal parameters:

- **PCA**: 7 components (Number of components at the 90% threshold of CEV)
- **K-Means and Hierarchical Clustering**: 3 clusters (Number of clusters with highest Silhouette Score)
- **t-SNE**: Perplexity of 50 for K-Means and 60 for Hierarchical Clustering (Perplexity at the highest Silhouette Score)

6.1.4 Optimal Model Selection

To achieve optimal performance and clustering analysis, we will combine dimensionality reduction techniques with clustering algorithms. By integrating t-SNE and PCA for dimensionality reduction, we can effectively reduce the complexity of our high-dimensional audio features while preserving essential information. This preprocessing

step ensures that the subsequent clustering methods, such as Hierarchical Clustering and K-means, can operate more efficiently and produce clearer and more distinct clusters.

Table 7 depicts the evaluation metrics for different combinations of dimensionality reduction and clustering methods:

Method	Silhouette Score	Davies-Bouldin Index
PCA with K-Means	0.163599	1.938777
t-SNE with K-Means	0.385450	0.820739
PCA + t-SNE with K-Means	0.385450	0.820739
PCA with Hierarchical Clustering	0.226291	1.545270
t-SNE with Hierarchical Clustering	0.400531	0.888063
PCA + t-SNE with Hierarchical Clustering	0.400531	0.888063

Table 7

Despite the identical scores of PCA + t-SNE with K-Means and t-SNE with K-Means, we chose to use t-SNE with K-Means exclusively for its simplicity and efficiency.

In addition, we chose t-SNE with K-Means over t-SNE with Hierarchical Clustering because it is better suited for our task of comparing cluster movements between different time periods. The significantly lower Davies-Bouldin Index (0.820739 vs. 0.888063) indicates superior cluster separation, which is essential for accurately tracking changes in musical styles over time. While Hierarchical Clustering has a slightly higher Silhouette Score (0.400531 vs. 0.385450), the computational efficiency and scalability of K-Means make it more practical for handling our high-dimensional audio data. These advantages ensure that we can efficiently and effectively compare the evolution of music clusters across different decades.

Figure 7 below shows the t-SNE with K-Means clusters of the subsetted data between 2000 and 2024.

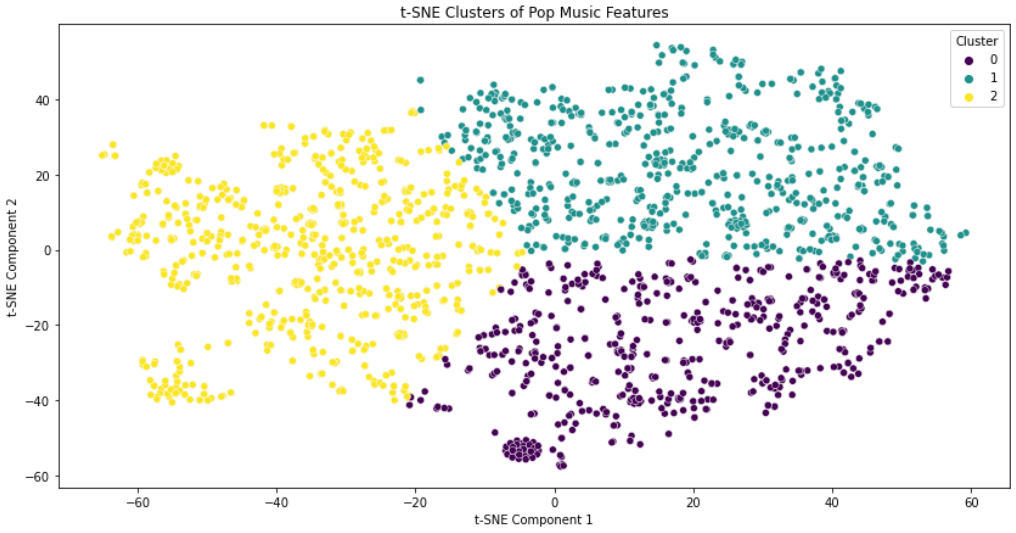

Figure 7

6.2 Unsupervised Evaluation

6.2.1 Choice of Evaluation Metrics

To evaluate the effectiveness of our clustering methods and the choice of feature representation, we utilized the following metrics:

- **Silhouette Score:** This metric assesses the cohesion within clusters and the separation between clusters. A higher Silhouette Score indicates better-defined and more distinct clusters. This is crucial for understanding the clear boundaries between different sub-genres of pop music and determining how well each song fits within its assigned cluster compared to other clusters, thereby providing insight into the quality and reliability of our clustering results.
- **Davies-Bouldin Index:** This index measures the average similarity ratio of each cluster with its most similar cluster. A lower Davies-Bouldin Index indicates better clustering as clusters are more distinct from each other. This ensures that each identified sub-genre of pop music has unique characteristics and minimal overlap between clusters, verifying that our clustering algorithm effectively differentiates between different musical styles.

By employing both the Silhouette Score and the Davies-Bouldin Index, we achieve a comprehensive evaluation of clustering quality, balancing cluster cohesion and distinctness to select the most effective model for categorizing pop music into meaningful sub-genres.

6.2.2 Results and Justification

After performing t-SNE with K-means clustering to categorize a diverse collection of pop music into distinct sub-genres, we were able to identify three primary clusters: Electropop, Dance Pop, and Acoustic Pop. Each cluster exhibits unique characteristics in terms of musical attributes, such as danceability, energy, and acousticness. The cluster assignments are determined based on Daniel Silver's research paper on Genre Complexes in Popular Music (See Reference 11). These clusters provide a detailed understanding of the varied landscape of pop music, highlighting the distinct elements that define each sub-genre. In terms of model evaluation, the t-SNE with K-means clustering model achieved a Silhouette Score of 0.40 and Davies-Bouldin Index at 0.87 on the subsetted data from 2000 to 2024. See Table 8 and Figure 8 below for details.

Evaluation Metrics	Score
Silhouette Score	0.40
Davies-Bouldin Index	0.87

Table 8

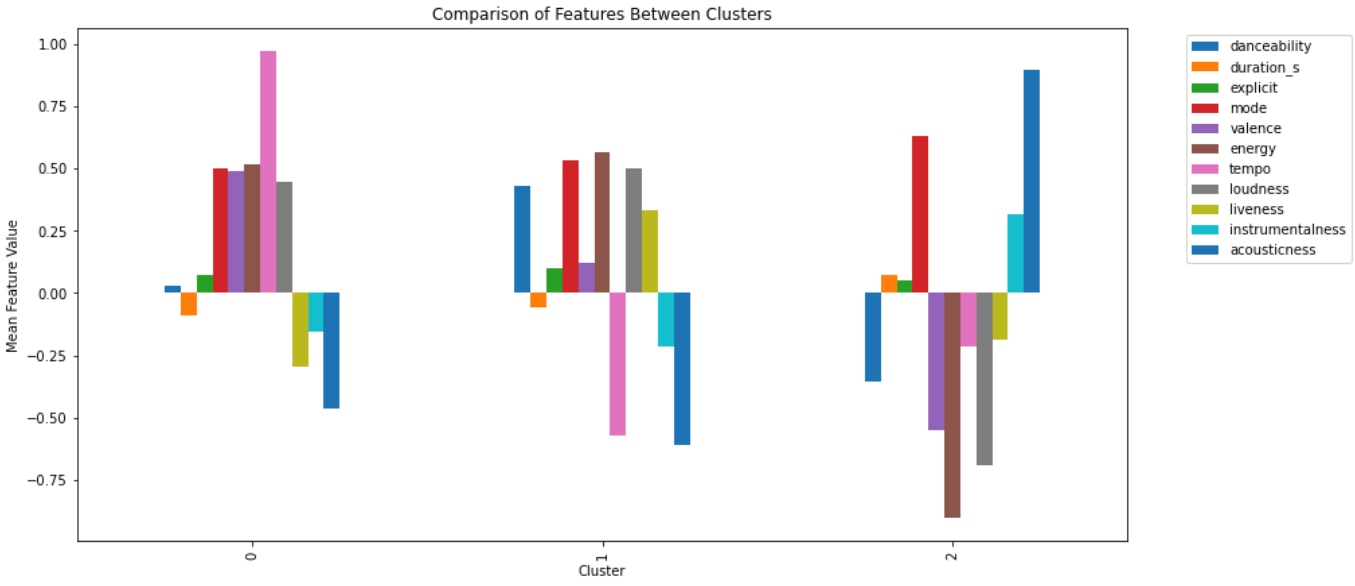

Figure 8

6.2.3 Cluster Interpretations

- **Electropop (Cluster 0)**: Slightly above-average danceability, faster tempos, moderately explicit, positive valence, moderately loud and energetic, lower liveness and acousticness.

- **Dance Pop (Cluster 1)**: High danceability, slightly shorter durations, often in a major key with positive valence, high energy, moderately loud, higher liveness, low instrumentalness and acousticness.
- **Acoustic Pop (Cluster 2)**: Low danceability, longer durations, fewer explicit lyrics, often in a major key but convey more somber emotions, very low energy, slower tempos, quieter, higher instrumentalness and acousticness.

6.2.4 Cluster Evolution

Figure 9 illustrates the cluster visualizations for the respective time periods, enabling us to visually track the changes within each cluster over recent years. There are 985 data points in the 2000s-2010s period and 946 data points in the 2020s period.

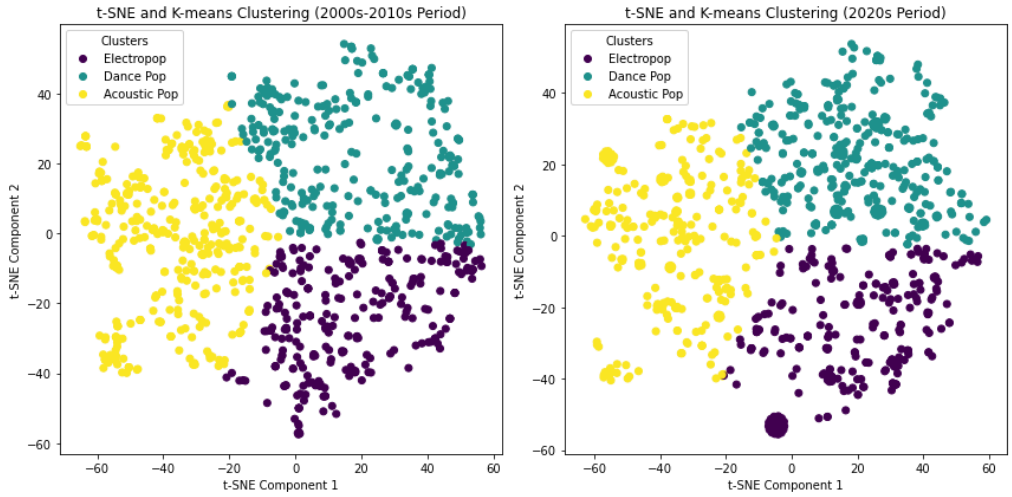

Figure 9

By analyzing the centroid movement of clusters between the 2000s-2010s and 2020s, we identified notable trends in our sub-genres. Electropop has become more danceable, shorter, with increased explicit content, reduced energy, and fewer live elements, resulting in a polished electronic sound. Dance Pop also evolved to be more danceable and shorter, with moderately increased explicit lyrics and decreased energy, giving it a studio-produced feel. Acoustic Pop saw a subtle rise in danceability and energy, shorter durations, and cleaner lyrics, leading to a more neutral, acoustic sound. These shifts reflect how pop music has adapted to changing listener preferences and production techniques. Table 9 shows the differences in audio feature shifts between the two periods.

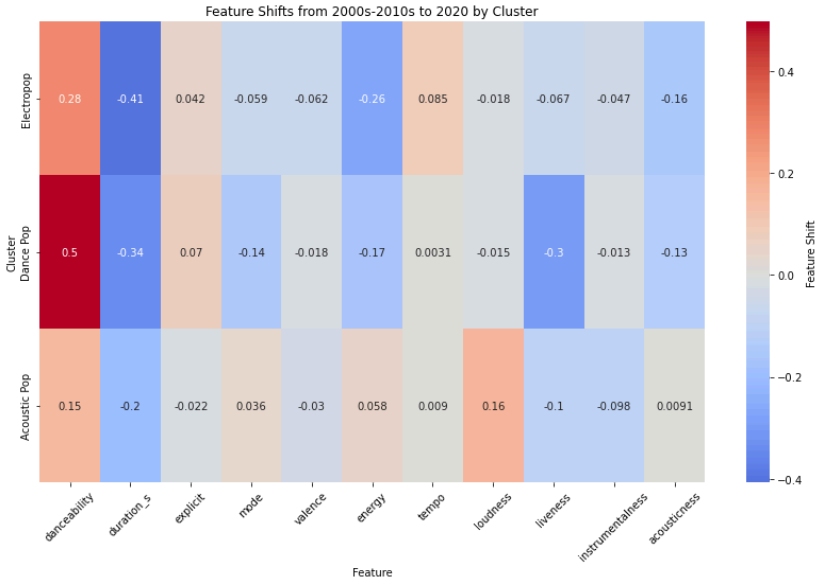

Table 9

6.2.5 Sensitivity Analysis

Figure 10 below shows the sensitivity analysis of our unsupervised model. We aimed to select the optimal number of clusters using the Silhouette score as the evaluation metric, based on varying perplexity values.

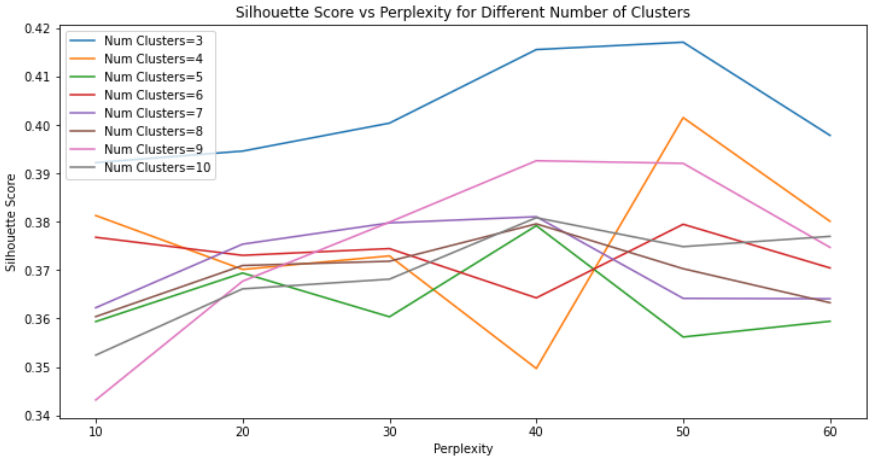

Figure 10

The sensitivity analysis of the clustering model reveals that the performance is moderately sensitive to the choice of hyperparameters, particularly the perplexity and the number of clusters. Higher perplexity values, such as 40 and 50, generally yielded better silhouette scores, indicating more well-defined clusters. Specifically, a perplexity of 50 with 3 clusters achieved the highest Silhouette Score of 0.417072. Conversely, the lowest Davies-Bouldin Index, which indicates better cluster separation, was observed at a perplexity of 30 with 9 clusters. Given our emphasis on the Silhouette Score, we decided to go with a perplexity of 50 and 3 clusters to optimize clustering performance. These results aligned with our earlier model parameter optimizations and suggest that the model's effectiveness in capturing the data structure varies with different hyperparameter settings, emphasizing the importance of careful selection of perplexity and the number of clusters.

6.3 Discussion

6.3.1 What did you learn?

From the results, we learn that pop music in recent years has undergone notable transformations across different sub-genres. In Electropop, there is a clear trend towards more danceable and shorter songs with a polished, electronic sound, characterized by less acousticness and energy. Dance Pop follows a similar trend, becoming more danceable, shorter, and studio-produced, with a moderate increase in explicit content and a decrease in live elements and energy. On the other hand, Acoustic Pop shows a nuanced evolution with slight increases in danceability, energy, and loudness, while maintaining a cleaner lyrical content and a more acoustic nature. Overall, the shift in pop music reflects a preference for more refined and produced sounds, shorter durations, and varying degrees of danceability and acoustic elements depending on the sub-genre. This evolution indicates a broader trend in pop music towards high-energy, danceable tracks with a polished production quality.

6.3.2 Surprises

What surprised me about the results is the noticeable decrease in energy levels across both Electropop and Dance Pop clusters, despite an increase in danceability. This suggests a shift in production styles where songs are crafted to be more rhythmically engaging and dance-friendly, yet they do not necessarily convey the same intensity or loudness as before. Additionally, the consistent shortening of song duration across all clusters is surprising, indicating a trend towards more concise musical formats. Another unexpected finding is the slight increase in acoustic elements and major keys in Acoustic Pop, which contrasts with the general trend towards electronic and studio-produced sounds in other sub-genres. This highlights a diverse evolution within pop music, where different sub-genres are adapting unique characteristics in response to changing listener preferences.

6.3.3 Challenges

Managing data imbalance across different periods might have been challenging, particularly given the greater availability of data in more recent decades. This challenge was addressed by focusing on these recent decades to ensure a richer dataset for analysis. Additionally, balancing the trade-off between retaining significant

information and reducing dimensions with PCA or t-SNE was difficult. This was managed by experimenting with different dimensionality reduction techniques and evaluating their impact on clustering results. Ensuring the clusters were meaningful and distinct presented its own challenges. To validate the robustness of the clusters, metrics like the silhouette score and Davies-Bouldin index were used, which provided a rigorous framework for cluster validation.

6.3.4 How could you extend your solution with more time/resources?

With more time and resources, the analysis could be enhanced by incorporating additional features such as lyrical content using NLP techniques or more granular audio features with tools like Librosa. Experimenting with more sophisticated clustering algorithms, such as DBSCAN and Gaussian Mixture Models, and comparing their performance with K-means and hierarchical clustering could provide deeper insights. Conducting a more detailed temporal analysis, potentially examining year-by-year changes instead of decade-by-decade, would help capture more granular trends. Additionally, developing interactive visualizations to explore clusters dynamically would allow users to better understand the evolution of music features over time.

6.4 Ethical Considerations

One of the primary ethical issues in unsupervised learning is the potential for bias. Since these models identify patterns without predefined labels, they are susceptible to biases present in the data. If the dataset is not representative, the clusters may perpetuate these biases. In our project, aimed at discovering the evolution of pop music, we sampled exclusively from the pop genre on Spotify. To mitigate bias, we randomly sampled from various decades to ensure a comprehensive representation of pop music over time.

Another ethical concern is the misuse of unsupervised learning results. Clusters can be misinterpreted or misapplied, leading to incorrect conclusions or harmful actions, such as stereotyping or unfair targeting. To prevent this, we educated stakeholders on the appropriate use and limitations of clustering results. We provided clear context and charts for interpreting the clusters, along with transparent communication about the methods used. By addressing these ethical considerations, we promote responsible and fair use of unsupervised learning technology.

7. Statement of Work

Andre	Bowie
Document Proposal Report Chapters 1 to 4 Supervised Learning (Chapter 5) Stand-up 1 Stand-up 2	Idea and Concept Document Proposal Records retrieval from Spotify (using API) Dataset construction Unsupervised Learning Analysis (Chapter 6) Stand-up 2

Table 10

8. References

[1] An Introduction to Statistical Learning with Applications in R" by Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani (Springer, 2013)

[2] "Random Forests" by Leo Breiman (Machine Learning, 2001)

[3] Scikit-Learn Documentation
https://imbalanced-learn.org/stable/references/generated/imblearn.ensemble.BalancedRandomForestClassifier.html

[4] “How to tune hyperparameters of tSNE” by Nikolay Oskolkov
https://towardsdatascience.com/how-to-tune-hyperparameters-of-tsne-7c0596a18868

[5] “Mastering t-SNE” (t-distributed stochastic neighbor embedding) by Sachinsoni
https://medium.com/@sachinsoni600517/mastering-t-sne-t-distributed-stochastic-neighbor-embedding-0e365ee898ea

[6] Borderline SMOTE Documentation
https://imbalanced-learn.org/stable/references/generated/imblearn.over_sampling.BorderlineSMOTE.html

[7] SVM SMOTE Documentation
https://imbalanced-learn.org/stable/references/generated/imblearn.over_sampling.SVMSMOTE.html

[8] ADASYN Documentation
https://imbalanced-learn.org/stable/references/generated/imblearn.over_sampling.ADASYN.html

[9] K Means SMOTE Documentation
https://imbalanced-learn.org/stable/references/generated/imblearn.over_sampling.KMeansSMOTE.html

[10] “Clustering and Dimensionality Reduction” by Guy Shtar and Shiri Margel
https://www.imperva.com/blog/clustering-and-dimensionality-reduction-understanding-the-magic-behind-machine-learning/

[11] “Genre Complexes in Popular Music” by Daniel Silver
https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4874668/

Appendix

Spotify API URL

- https://developer.spotify.com/documentation/web-api/concepts/api-calls

Kaggle Data

- Provided the top 100 records of Spotify track dataset found on Kaggle in the zip file
- Original Kaggle dataset extracted is more than 20MB

Code and Data Use

- **api_df**, **clean_artists_df, Dataset_Concat_Normalized,** and **Dataset_Concat_not_Normalized** are imported to run MS2_SupervisedML_Virality, which is the supervised learning step
- **preprocessed_df** is used to run both MS2_UnsupervisedML_Optimization and MS2_UnsupervisedML_Results, which are notebooks for model optimizations and unsupervised learning results
- The steps to run the notebooks are indicated in the respective notebook file names