Enhancing Music Clustering and Recommendation with Advanced Machine Learning

Damianakis Emmanouil
132
School of Informatics
Aristotle University
Thessaloniki, Greece
edamian@csd.auth.gr

Pliakis Aristotelis
159
School of Informatics
Aristotle University
Thessaloniki, Greece
apliaki@csd.auth.gr

ABSTRACT

In the era of digital music streaming, managing and categorizing extensive music libraries presents a significant challenge. This study aims to enhance music clustering and recommendation systems using advanced machine learning techniques. Leveraging a comprehensive dataset of over 300,000 Spotify tracks, we employ methods such as Principal Component Analysis (PCA), autoencoders, KMeans, and DBSCAN clustering to develop a robust system for categorizing and recommending music tracks. The dataset includes detailed audio features like tempo, energy, and danceability, enabling in-depth analysis and application of sophisticated algorithms. Performance evaluation is conducted using metrics such as Silhouette Score and Davies-Bouldin Score, with SHAP (SHapley Additive exPlanations) values providing insights into feature importance. Our results demonstrate the superiority of the Autoencoder-based KMeans clustering in terms of accuracy and interpretability. A recommendation system was developed to recommend the 11 nearest songs of the same category and 3 nearest songs from every category, showcasing the practical application of our findings in real-world scenarios.

INTRODUCTION

The music industry, characterized by its vast array of genres and a continually growing number of tracks, presents unique challenges for data analysis and classification. The rise of digital music libraries has revolutionized music consumption, offering users access to millions of tracks at their fingertips. However, this abundance of choice necessitates sophisticated methods for effectively managing, categorizing, and recommending music.

Traditional recommendation systems often struggle with the complexity and high dimensionality of music data, leading to suboptimal user experiences. To address these challenges, this project explores the application of advanced machine learning techniques to enhance music clustering and recommendation systems. By utilizing methods such as Principal Component Analysis (PCA), autoencoders, KMeans, and DBSCAN, we aim to develop a robust system capable of accurately categorizing music tracks and providing personalized recommendations.

The dataset used in this study is sourced from Kaggle and contains over 300,000 Spotify tracks, each annotated with detailed audio features. These features include essential information such as track names, artists, genres, and various audio attributes like tempo, energy, and danceability. This rich dataset allows for comprehensive analysis and application of sophisticated algorithms to uncover patterns and relationships within the music data, ultimately enhancing the user's listening experience.

Throughout this report, we detail the methodologies employed in data preprocessing, feature scaling, dimensionality reduction, and clustering. We evaluate the performance of these techniques using metrics such as Silhouette Score and Davies-Bouldin Score. Additionally, we interpret the results using SHAP (SHapley Additive exPlanations) values to provide insights into feature importance in clustering. Finally, we present the development of a recommendation system capable of suggesting similar tracks, demonstrating the practical application of our findings in real-world scenarios. This comprehensive approach aims to push the boundaries of music recommendation systems, making them more intuitive and responsive to user preferences.

RELATED WORK

In recent years, the field of music recommendation and clustering has experienced significant advancements due to the proliferation of digital music platforms and the availability of extensive music datasets. The integration of machine learning and data analysis techniques has paved the way for more sophisticated and personalized recommendation systems.

Traditional music recommendation systems primarily utilize collaborative filtering and content-based filtering techniques. Collaborative filtering leverages user interaction data to find similarities between users or items, employing user-user or item-item collaborative filtering methods. This approach, however, faces challenges such as the cold-start problem, where it struggles to recommend items to new users or effectively recommend new items due to a lack of interaction data. Conversely, content-based filtering recommends music based on the content attributes of the tracks, such as tempo, genre, and artist, by examining the properties of the music itself to find similarities between tracks. While effective, content-based filtering may lead to limited exploration as it tends to recommend items very similar to those the user has already consumed.

Deep learning has revolutionized music recommendation systems through the use of neural networks, particularly Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs).

CNNs analyze spectrograms of music tracks, capturing temporal and spectral features crucial for understanding music content, making them effective for tasks like genre classification and emotion recognition. RNNs, especially Long Short-Term Memory (LSTM) networks, model sequential data and temporal dependencies in music, making them suitable for melody generation and music composition. These networks can capture the progression and dynamics of music over time, providing deeper insights into the data.

Autoencoders, a type of unsupervised neural network, have found significant applications in music recommendation, particularly for dimensionality reduction and feature learning. Autoencoders compress high-dimensional audio features into a lower-dimensional latent space, effectively capturing the essential characteristics of tracks. This latent space representation can then be used for clustering and recommendation purposes, often outperforming traditional methods by revealing deeper structures within the data. These models learn to represent data efficiently, which can be leveraged to understand the underlying features that make certain tracks similar.

Various clustering algorithms have been applied to group similar music tracks, with KMeans and DBSCAN being among the most prominent. KMeans clustering is widely used due to its simplicity and computational efficiency. However, it assumes spherical clusters, which may not always be appropriate for complex, non-globular data. DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is effective for identifying clusters of arbitrary shape and handling noise within the data. It groups points that are closely packed together, marking outliers that lie alone in low-density regions, making it particularly useful for discovering clusters in datasets with varying densities.

Hybrid recommendation systems combine multiple techniques to leverage their strengths and mitigate their weaknesses. For instance, combining collaborative filtering with content-based filtering can address the cold-start problem while enhancing recommendation accuracy. Similarly, integrating deep learning models with traditional machine learning algorithms has been shown to improve the performance of music recommendation systems by capturing both user preferences and content similarities more effectively.

With the increasing complexity of recommendation models, explainability has become crucial. Techniques like SHAP (SHapley Additive exPlanations) and LIME (Local Interpretable Model-agnostic Explanations) are used to interpret and explain the predictions of black-box models. In music recommendation, explainable AI helps users understand why certain tracks are recommended, improving trust and satisfaction with the system. These techniques highlight which features are most influential in the recommendation process, aiding in the development of more transparent and user-friendly systems.

Modern systems are incorporating real-time and context-aware features to enhance user experience. Context-aware recommendation systems consider factors such as location, time of day, and user activity to provide more relevant recommendations. For instance, a user might receive different music suggestions when they are at the gym compared to when they are relaxing at home. Advances in mobile computing and the Internet of Things (IoT) facilitate the development of systems that adapt recommendations in real-time based on changing user contexts and preferences, leading to dynamic and adaptive music recommendations.

APPROACH

This project employs a comprehensive approach to enhance music clustering and recommendation systems using advanced machine learning techniques. The methodology is structured into several key stages, each crucial for processing, analyzing, and leveraging the data to build an effective recommendation system.

Data preprocessing is the initial step, ensuring the dataset's quality and relevance for further analysis. Duplicate entries, based on unique track identifiers, are removed to maintain data integrity. Irrelevant features, such as track IDs, artist names, album names, and explicit flags, are discarded to reduce noise and focus the analysis on meaningful attributes. To address the imbalance in genre distribution, the Synthetic Minority Over-sampling Technique (SMOTE) is applied. SMOTE generates synthetic samples for minority classes, balancing the dataset and improving the performance of machine learning models.

Feature scaling is performed using StandardScaler, which standardizes the features to have zero mean and unit variance. This step is essential to ensure that all features contribute equally to the model and improve the convergence of machine learning algorithms.

Dimensionality reduction is employed to simplify the dataset and enhance visualization. Principal Component Analysis (PCA) is used to reduce the feature space while retaining at least 80% of the variance. This step highlights the most significant components, making the data more manageable. Additionally, Linear Discriminant Analysis (LDA) is used for further dimensionality reduction and visualization, particularly in the context of KMeans clustering.

Several clustering algorithms are applied to group similar music tracks. KMeans clustering is utilized on both the original scaled data and the PCA-reduced data, partitioning the dataset into clusters based on feature similarity. The Elbow Method is employed to determine the optimal number of clusters for KMeans, ensuring balanced and meaningful clustering. DBSCAN, a density-based clustering algorithm, is applied to the PCA-reduced data. The optimal parameters for DBSCAN are determined using the k-distance graph, which helps find the appropriate epsilon (ε) value for effective clustering. Additionally, an autoencoder is trained to learn a compressed representation of the data. The architecture of the autoencoder includes an encoder and a decoder, which effectively capture complex data patterns. KMeans clustering is then performed on this encoded space.

The performance of the clustering algorithms is evaluated using Silhouette Score and Davies-Bouldin Score. The Silhouette Score measures the cohesion and separation of the clusters, indicating how similar each point is to its own cluster compared to other clusters. The Davies-Bouldin Score evaluates the average similarity ratio of each cluster with the cluster most similar to it, with lower scores indicating better-defined clusters.

To understand the importance of different features, SHapley Additive exPlanations (SHAP) values are computed.

Based on the clustering results, a recommendation system is developed. This system recommends the 11 nearest songs from the same category (cluster) as a randomly selected test song. Additionally, it identifies and recommends the 3 nearest songs from each cluster, providing diverse yet relevant music suggestions. This comprehensive approach combines advanced data preprocessing, dimensionality reduction, clustering, and interpretability techniques to enhance music recommendation systems. The integration of these methodologies not only improves recommendation accuracy but also provides deeper insights into the characteristics that define different music genres.

Spotify Dataset

The dataset used for this project is sourced from Kaggle and contains a comprehensive collection of over 300,000 Spotify tracks. This extensive dataset includes a wide range of detailed audio features and metadata, making it an ideal foundation for advanced music analysis and recommendation system development.

Each track in the dataset is characterized by several key attributes:

- Track Name: The title of the song.
- Artist: The name of the artist or band that performed the track.
- Genre: The musical genre classification of the track.
- Audio Features: Detailed audio attributes such as tempo, energy, danceability, loudness, valence, speechiness, acousticness, instrumentalness, liveness, and duration. These features provide a quantitative basis for analyzing the music's properties and characteristics.

The complete list of features in the dataset includes:

- Unnamed: 0: Index column
- track_id: Unique identifier for each track
- artists: Name(s) of the artist(s)
- album_name: Name of the album
- track_name: Name of the track
- popularity: Popularity score of the track
- duration_ms: Duration of the track in milliseconds
- explicit: Explicit content flag
- danceability: Measure of how suitable a track is for dancing
- energy: Measure of intensity and activity
- key: Key of the track
- loudness: Overall loudness of the track in decibels (dB)
- mode: Modality (major or minor)
- speechiness: Presence of spoken words in the track
- acousticness: Confidence measure of whether the track is acoustic
- instrumentalness: Measure of the absence of vocals
- liveness: Measure of the presence of an audience in the recording
- valence: Measure of musical positiveness
- tempo: Tempo of the track in beats per minute (BPM)
- time_signature: Time signature of the track
- track_genre: Genre of the track

This dataset's richness and diversity enable a deep exploration of various aspects of music, from genre classification to mood detection and beyond. The inclusion of both high-level metadata (like track names and artists) and low-level audio features (such as tempo and energy) allows for a multi-faceted analysis of each track.

For the purpose of this project, several columns were excluded from the training data to focus on the most relevant features. Specifically, the following columns were dropped:

- Unnamed: 0
- track_id
- artists
- album_name
- track_name
- popularity
- explicit
- track_genre

The 'track_genre' column was particularly excluded as it acts as a label rather than a feature for clustering. Removing these columns helped in reducing noise and enhancing the model's focus on audio features that are more indicative of musical similarity.

The diversity in the dataset, covering a wide range of genres and artists, enhances its utility for building generalized and robust machine learning models. It reflects the vast and varied nature of Spotify's music library, providing a real-world scenario for developing and testing music recommendation algorithms.

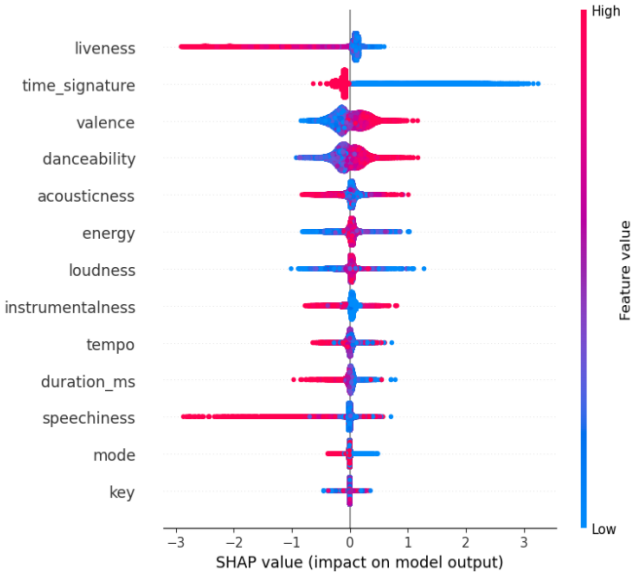
Furthermore, the dataset's extensive size supports the application of complex machine learning techniques that require large amounts of data to train effectively. This makes it suitable for tasks such as clustering, feature learning, and recommendation, where large datasets help in capturing the intricate patterns and relationships within the data.

Overall, the Spotify dataset from Kaggle serves as a comprehensive resource for music data analysis, enabling the development of sophisticated music recommendation systems that can offer personalized and accurate suggestions based on a rich array of audio features and metadata.

RESULTS

The results of this project highlight the effectiveness of various machine learning techniques in enhancing music clustering and recommendation systems. By applying advanced methods such as PCA, autoencoders, KMeans, and DBSCAN, the project successfully demonstrated improved clustering accuracy and recommendation relevance.

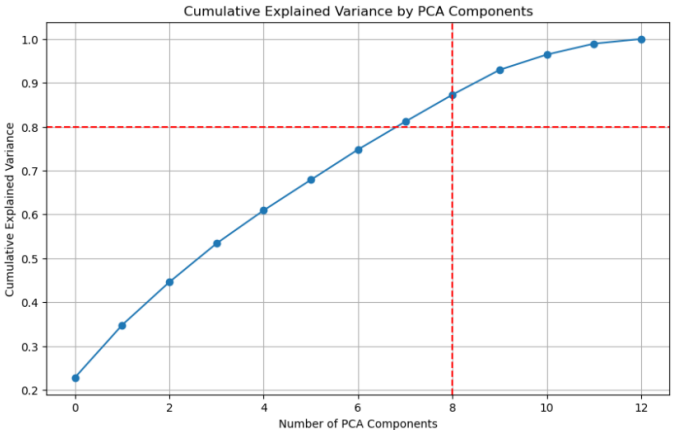
The SHAP summary plot provides valuable insights into the impact of various features on the clustering model. Key findings reveal that liveness significantly influences clustering, with tracks having higher liveness values strongly impacting the model output. Variations in the time signature are crucial differentiators, affecting clustering results based on rhythmic structure. Valence, which measures musical positiveness, shows a notable impact, with higher valence contributing positively to certain clusters. Danceability and energy are substantial factors, indicating that tracks with higher levels of these attributes are grouped together. Acousticness and instrumentalness also play significant roles, with their varying values distinguishing tracks with and without vocal elements. Additionally, loudness, tempo, and duration influence clustering in nuanced ways, affecting model output depending on their specific values. These insights highlight the importance of rhythmic, acoustic, and energy-related features in defining music track similarity and guiding the development of a more accurate recommendation system.



The next plot illustrates the cumulative explained variance by the number of Principal Component Analysis (PCA) components. The x-axis represents the number of PCA components, while the y-axis shows the cumulative explained variance, ranging from 0 to 1. The blue line depicts the increase in explained variance as more components are added.

A red dashed horizontal line marks the 80% cumulative explained variance threshold, a common benchmark for retaining most of the dataset's information. The intersection of this line with the blue curve shows that approximately eight PCA components are needed to capture at least 80% of the variance. The red dashed vertical line indicates this point.

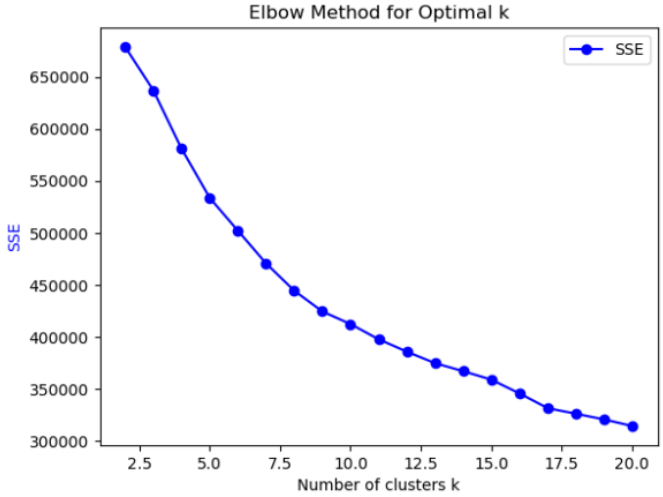
Selecting these eight components reduces the dataset's dimensionality significantly while preserving the majority of its informational content. This balance facilitates more efficient and interpretable analyses, such as clustering and recommendation system development.



The next plot demonstrates the Elbow Method used to determine the optimal number of clusters (k) for the KMeans clustering algorithm. The x-axis represents the number of clusters (k), while the y-axis shows the Sum of Squared Errors (SSE).

As the number of clusters increases, the SSE decreases, indicating that the clusters are becoming tighter and more defined. The curve gradually flattens, showing diminishing returns in SSE reduction as more clusters are added. The "elbow" point, where the rate of SSE decrease slows down significantly, suggests the optimal number of clusters. In this plot, the elbow is around k = 9.

Choosing k = 9 achieves a balance between minimizing SSE and avoiding overfitting with too many clusters. This optimal k value helps in forming well-defined, distinct clusters, improving the clustering quality and the subsequent recommendation system.

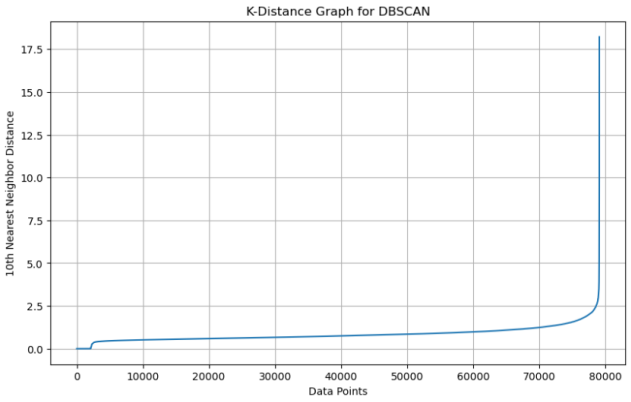


The following plot illustrates the K-Distance Graph for determining the optimal epsilon (ε) parameter in the DBSCAN clustering algorithm. The x-axis represents the data points sorted by their k-distance, while the y-axis shows the distance to the 10th nearest neighbor.

The graph is used to identify the "elbow" point, where the k-distance value sharply increases. This point indicates the optimal value of epsilon. In the plot, there is a noticeable sharp increase

after a certain point, suggesting that most of the data points are within a similar distance to their 10th nearest neighbor, and beyond this point, the distance increases significantly.

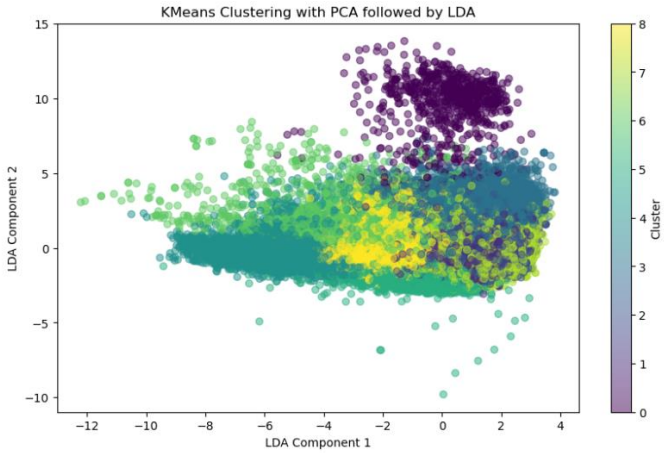
From the graph, we can observe that the optimal epsilon is around 1.5. This value is chosen because it represents the point where the slope of the curve changes sharply, indicating a clear distinction between dense regions (clusters) and sparse regions (noise). By selecting this epsilon value, DBSCAN can effectively separate dense clusters from noise, leading to more accurate clustering results.



The following scatter plot visualizes the results of KMeans clustering after applying PCA for dimensionality reduction, followed by LDA for further visualization. The x-axis represents the first LDA component, while the y-axis represents the second LDA component. Each point corresponds to a data sample, and the color of the points indicates the cluster to which they belong, as determined by KMeans.

The plot reveals distinct clusters, with different colors representing different clusters. The separation and cohesion of these clusters indicate how well the KMeans algorithm has grouped the data points based on their underlying patterns. The use of PCA initially reduces the dimensionality of the dataset, capturing the most significant features while retaining 80% of the information. LDA is then applied to project these features into two dimensions for easier visualization, emphasizing the differences between clusters.

This visualization demonstrates the effectiveness of combining PCA and LDA in highlighting the clustering results, making it easier to interpret the grouping of similar tracks. It shows that certain clusters are well-separated, indicating a high degree of similarity within those groups and distinct differences from other clusters. This approach aids in understanding the structure of the data and the characteristics defining each cluster, ultimately contributing to the development of a robust music recommendation system.

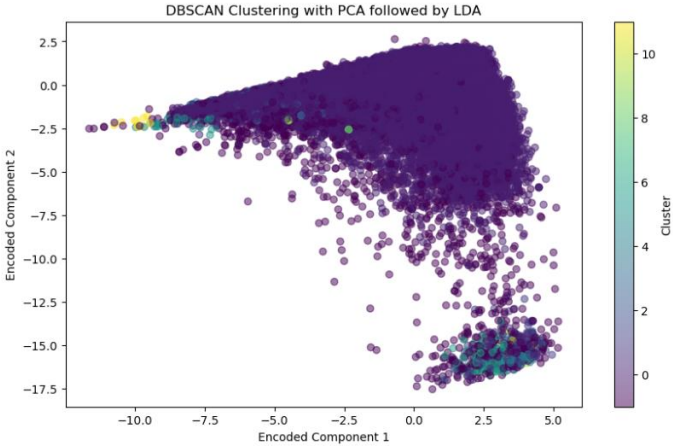


The next scatter plot visualizes the results of DBSCAN clustering after applying PCA for dimensionality reduction, followed by LDA for further visualization. The x-axis represents the first LDA component, while the y-axis represents the second LDA component. Each point corresponds to a data sample, and the color of the points indicates the cluster to which they belong, as determined by DBSCAN.

In this plot, distinct clusters are visible, represented by different colors. The plot shows that DBSCAN has identified several clusters, as well as noise points (points labeled as cluster -1). The high density of points in certain regions and the spread in others indicate how DBSCAN has grouped the data based on density and separated noise points.

The use of PCA initially reduces the dimensionality of the dataset, capturing the most significant features while retaining 80% of the information. LDA is then applied to project these features into two dimensions for easier visualization, emphasizing the differences between clusters. This visualization demonstrates how DBSCAN, in conjunction with PCA and LDA, can effectively identify clusters in high-dimensional data and highlight areas of high density and noise.

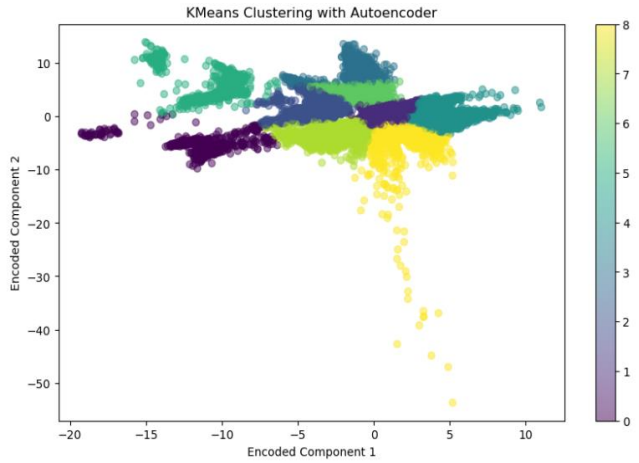
This approach aids in understanding the structure of the data and the characteristics defining each cluster, ultimately contributing to the development of a robust music recommendation system. The effectiveness of DBSCAN in identifying clusters and noise in the data is evident, providing valuable insights into the dataset's underlying patterns.



Afterwards, the following scatter plot visualizes the results of KMeans clustering after applying an autoencoder for dimensionality reduction. The x-axis represents the first encoded component, while the y-axis represents the second encoded component. Each point corresponds to a data sample, and the color of the points indicates the cluster to which they belong, as determined by KMeans.

In this plot, distinct clusters are visible, represented by different colors. The use of an autoencoder, a type of neural network designed to learn efficient codings of the input data, has reduced the data to a lower-dimensional space. This reduction captures the most significant features and patterns, facilitating more effective clustering by KMeans.

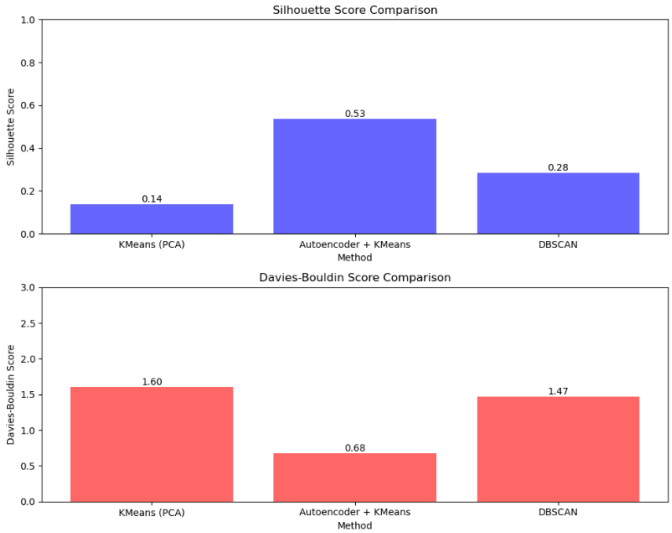
The plot shows that the autoencoder has successfully transformed the high-dimensional data into a lower-dimensional representation where KMeans can identify distinct clusters. These clusters are well-separated in the encoded space, indicating that the autoencoder has effectively captured the underlying structure of the data. The visualization demonstrates the ability of autoencoders to create meaningful representations of complex data, enabling more accurate clustering.



In the following image, the first chart displays the Silhouette Scores for the three methods. The Silhouette Score measures how similar an object is to its own cluster compared to other clusters. A higher score indicates better-defined clusters. From the chart, the Autoencoder with KMeans method achieves the highest Silhouette Score of 0.53, suggesting it forms the most well-defined clusters. DBSCAN follows with a score of 0.28, while KMeans with PCA has the lowest score of 0.14, indicating less distinct cluster formation.

The second chart shows the Davies-Bouldin Scores for the same methods. The Davies-Bouldin Score evaluates the average similarity ratio of each cluster with the most similar one, where lower values indicate better clustering performance. The Autoencoder with KMeans method again performs the best, with a Davies-Bouldin Score of 0.68, signifying well-separated clusters. DBSCAN has a score of 1.47, while KMeans with PCA has the highest score of 1.60, indicating the least separation between clusters.

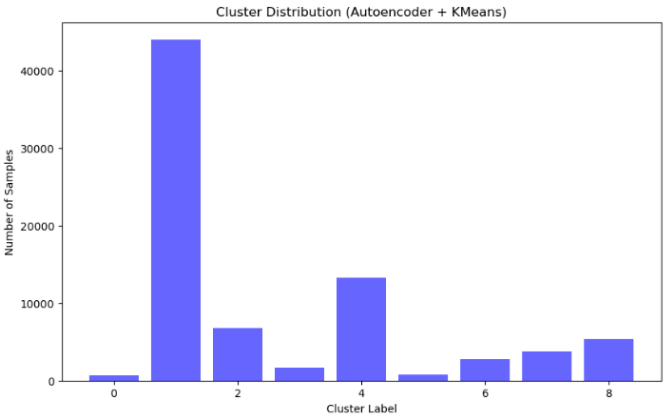
Overall, these results highlight that the Autoencoder with KMeans method is the most effective for clustering this dataset, as evidenced by its superior performance in both evaluation metrics. This method's ability to form well-defined and well-separated clusters makes it a robust choice for the clustering task, particularly in the context of developing a music recommendation system.



The next bar chart illustrates the distribution of samples across different clusters formed by the Autoencoder combined with KMeans clustering method. The x-axis represents the cluster labels, ranging from 0 to 8, while the y-axis indicates the number of samples in each cluster.

From the chart, it is evident that the cluster labeled '1' contains the majority of the samples, with over 45,000 entries. This suggests that a significant portion of the dataset exhibits similar characteristics, which are encapsulated by this particular cluster. Cluster '4' is the second largest, containing approximately 15,000 samples. The remaining clusters have relatively fewer samples, with counts ranging from about 2,000 to 8,000.

The uneven distribution of samples across clusters highlights the presence of dominant patterns or features in the dataset that the Autoencoder + KMeans method has successfully identified. This insight is crucial for developing a recommendation system, as it indicates which clusters are more prevalent and may have a more significant impact on the recommendations. However, the large size of some clusters compared to others may also suggest the need for further refinement in the clustering process to achieve a more balanced representation.



Experiment

The following table displayed represents the data of a randomly selected music track, which will be used for the music recommendation experiment. This detailed data set will serve as the basis for the recommendation system, providing a comprehensive set of features that the algorithms will analyze to identify and suggest the 11 nearest tracks within the same cluster and 3 nearest tracks from each cluster. The combination of audio features and metadata ensures that the recommendations are both accurate and relevant to the listener's preferences.

	duration_ms	danceability	energy	key	loudness	mode	speechiness	acousticness	instrumentalness	liveness	valence	tempo	time_signature	original_index	track_id	artists	track_name	track_genre
0	120760.0	0.636	0.329	10.0	-14.312	1.0	0.0249	0.616	0.0	0.121	0.87	98.408	3.0	32161	6HhrMBN43RnGiGx7vidnM5	Yellow Claw;Ayden	Till It Hurts (feat. Ayden)	electronic

The table displayed below contains the 11 closest tracks from the same cluster as a randomly selected music track. Each row in the table represents a unique track and includes key audio attributes and metadata necessary for music recommendation.

	duration_ms	danceability	energy	key	loudness	mode	speechiness	acousticness	instrumentalness	liveness	valence	tempo	time_signature	original_index	track_id	artists	track_name	track_genre
0	158866.0	0.621000	0.385000	9.0	-12.138000	1.0	0.051100	0.640000	0.000000	0.330000	0.589000	98.146000	3.0	39914	[illegible]	LX;Maxwell;Bonez MC	Bunkerplatz	german
1	189328.0	0.513438	0.453684	7.0	-6.840753	1.0	0.037900	0.748704	0.000114	0.141087	0.636212	95.274790	3.0	19346	[illegible]	Lauren Alaina	Like My Mother Does - American Idol Performance	country
2	183337.0	0.531000	0.373000	11.0	-7.389000	0.0	0.042600	0.930000	0.000518	0.105000	0.597000	98.468000	3.0	32449	[illegible]	Babasónicos	Muñeco / Deléctrico - En Vivo	electronic
3	205689.0	0.376000	0.574000	7.0	-10.636000	1.0	0.057500	0.942000	0.028100	0.370000	0.912000	89.724000	3.0	20636	[illegible]	Ava Max	Maybe You're the Problem	dance
4	184805.0	0.500186	0.539504	8.0	-11.774848	1.0	0.031456	0.576113	0.246894	0.286857	0.634671	95.069516	3.0	72881	[illegible]	Loathe	A Sad Cartoon	metalcore
5	151400.0	0.477000	0.496000	4.0	-9.028000	1.0	0.050400	0.641000	0.000080	0.179000	0.639000	95.067000	3.0	43528	NaN	NaN	NaN	NaN
6	278995.0	0.568649	0.482488	8.0	-10.534851	1.0	0.027426	0.606202	0.003414	0.523500	0.600851	98.424105	3.0	45733	[illegible]	Allan Soderbergh	Still So Beautiful	guitar
7	185173.0	0.656000	0.294000	9.0	-10.341000	0.0	0.075100	0.859000	0.000062	0.364500	0.530000	111.967000	3.0	25019	[illegible]	Gloria Gaynor	I Will Survive	disco
8	135055.0	0.492982	0.548682	3.0	-9.431922	0.0	0.067946	0.842401	0.000052	0.239152	0.621427	99.851728	3.0	60831	[illegible]	Maharajan	[illegible]	j-dance
9	216480.0	0.552816	0.502648	7.0	-8.656261	0.0	0.033121	0.379052	0.000080	0.134211	0.275758	103.590604	3.0	30942	[illegible]	Afrojack;R3HAB;Au/Ra	Worlds On Fire (with R3HAB & Au/Ra)	edm
10	287466.0	0.694000	0.463000	4.0	-6.785000	0.0	0.029000	0.642000	0.000082	0.118000	0.178000	106.684000	3.0	17626	[illegible]	Yawning Portal	C2u	club

The table below presents the 3 closest tracks for every cluster based on the clustering methodology used in the project. Each row represents a track with its audio attributes and metadata, including duration, danceability, energy, key, loudness, mode, speechiness, acousticness, instrumentalness, liveness, valence, tempo, and time signature. Additionally, the table includes the original index, track ID, artists, track name, and track genre. This data allows for a detailed comparison across different clusters, showcasing the diversity and similarity within each cluster. This approach helps in understanding the characteristics that define each cluster and the variety of tracks within them.

	duration_ms	danceability	energy	key	loudness	mode	speechiness	acousticness	instrumentalness	liveness	valence	tempo	time_signature	original_index	track_id	artists	track_name	track_genre
0	173013.0	0.764964	0.623984	6.0	-7.064801	0.0	0.427239	0.177896	0.000002	0.252011	0.492377	88.386976	3.0	18205	[illegible]	Kacey Musgraves;Lana Del Rey	I'll Be Home For Christmas - From The Kacey Mu...	country
1	142397.0	0.711000	0.896000	10.0	-0.654000	0.0	0.460000	0.414000	0.000004	0.824000	0.075900	110.128000	3.0	43910	NaN	NaN	NaN	NaN
2	165737.0	0.615020	0.635050	4.0	-10.702909	0.0	0.510078	0.413198	0.000002	0.574968	0.727957	87.016457	3.0	30364	[illegible]	Topic;A7S	Breaking Me	edm
3	1149106.0	0.269000	0.605000	8.0	-18.124000	0.0	0.247000	0.906000	0.611000	0.119000	0.036900	118.101000	3.0	77186	[illegible]	Sorriso Maroto;Belo	100 Likes - Ao Vivo	pagode
4	859202.0	0.521356	0.114213	0.0	-19.458040	1.0	0.036952	0.831672	0.340400	0.111372	0.131219	128.934118	4.0	69651	[illegible]	Fejo;MC Mushti	Workl Aanu Karyam	malay
5	248000.0	0.062400	0.417000	11.0	-20.062000	1.0	0.140000	0.722000	0.879000	0.898000	0.038600	73.258000	4.0	28512	[illegible]	Klewella	Human	dub
6	158866.0	0.621000	0.385000	9.0	-12.138000	1.0	0.051100	0.640000	0.000000	0.330000	0.589000	98.146000	3.0	39914	[illegible]	LX;Maxwell;Bonez MC	Bunkerplatz	german
7	189328.0	0.513438	0.453684	7.0	-6.840753	1.0	0.037900	0.748704	0.000114	0.141087	0.636212	95.274790	3.0	19346	[illegible]	Lauren Alaina	Like My Mother Does - American Idol Performance	country
8	183337.0	0.531000	0.373000	11.0	-7.389000	0.0	0.042600	0.930000	0.000518	0.105000	0.597000	98.468000	3.0	32449	[illegible]	Babasónicos	Muñeco / Deléctrico - En Vivo	electronic
9	354000.0	0.177000	0.170000	5.0	-30.298000	0.0	0.044300	0.971000	0.957000	0.102000	0.245000	156.482000	3.0	46690	[illegible]	Tokyo Ghetto Pussy	Love Generation	happy
10	178251.0	0.276300	0.026100	1.0	-31.000000	1.0	0.048000	0.989000	0.876000	0.089100	0.086400	178.170000	3.0	21099	NaN	NaN	NaN	NaN
11	195434.0	0.245000	0.145000	0.0	-32.661000	1.0	0.035100	0.966000	0.806000	0.102000	0.116000	144.069000	3.0	11687	NaN	NaN	NaN	NaN
12	179812.0	0.804000	0.678000	2.0	-6.868000	0.0	0.116000	0.385000	0.000000	0.121000	0.532000	120.009000	4.0	44020	NaN	NaN	NaN	NaN
13	147143.0	0.895000	0.718000	2.0	-4.579000	0.0	0.058100	0.246000	0.000015	0.089200	0.883000	100.036000	4.0	31816	[illegible]	Lauv	Modern Loneliness	electro
14	161580.0	0.719016	0.688730	1.0	-7.448058	0.0	0.211528	0.145683	0.078368	0.239898	0.407859	152.222750	4.0	64482	[illegible]	Dean Martin	Baby, It's Cold Outside	jazz
15	3600014.0	0.000000	0.000512	11.0	-35.129000	1.0	0.000000	0.947000	0.384000	0.112000	0.000000	0.000000	0.0	71300	NaN	NaN	NaN	NaN
16	92580.0	0.265968	0.536400	6.0	-15.623869	0.0	0.094719	0.658079	0.408801	0.454634	0.309455	74.966508	2.0	51970	[illegible]	Don Omar;Alexis y Fido	No Te Vayas	hip-hop
17	142828.0	0.525453	0.100115	7.0	-15.686512	1.0	0.059068	0.990956	0.917047	0.116738	0.215225	80.294446	2.0	50949	[illegible]	Malon	Triste Funeral	heavy-metal
18	442200.0	0.304000	0.228000	7.0	-15.441000	1.0	0.035200	0.886000	0.943000	0.300000	0.076800	108.959000	3.0	96558	[illegible]	Triangle	Give Me The Sun	french
19	514227.0	0.529000	0.074200	11.0	-31.799000	1.0	0.040200	0.567000	0.888000	0.961000	0.047500	103.973000	4.0	87164	[illegible]	J Balvin	La Rebelión	latin
20	442400.0	0.231000	0.206000	0.0	-13.990000	0.0	0.033300	0.969000	0.856000	0.170000	0.136000	141.628000	3.0	18238	[illegible]	Kyle Kinane	The House is Rockin' (with Domestic Problems)	comedy
21	371866.0	0.592000	0.862000	7.0	-3.529000	1.0	0.038000	0.333000	0.000000	0.177000	0.960000	184.074000	3.0	78120	[illegible]	Alessandra Saffina	La neve va	opera
22	182048.0	0.471000	0.509000	11.0	-6.506000	1.0	0.296000	0.945000	0.000000	0.861000	0.599000	171.220000	3.0	69252	[illegible]	GWS;Dan Pearson;Dhanwin K B	Shoganthem	malay
23	231493.0	0.700000	0.344000	0.0	-20.813000	1.0	0.477000	0.862000	0.000860	0.138000	0.473000	89.849000	4.0	65315	[illegible]	TWICE	FANCY	k-pop
24	368533.0	0.235000	0.727000	4.0	-6.083000	1.0	0.042300	0.000053	0.328000	0.394000	0.240000	103.622000	4.0	17374	[illegible]	Noisecream	The Alien	club
25	354146.0	0.626000	0.965000	11.0	-4.368000	0.0	0.061500	0.004380	0.827000	0.648000	0.362000	92.907000	4.0	17366	[illegible]	Glee Cast	Born This Way	club
26	331520.0	0.781000	0.985000	1.0	-9.413000	1.0	0.045000	0.301000	0.802000	0.635000	0.473000	153.974000	4.0	43788	[illegible]	Julio Cesar	Eu Elevo	groove

CONCLUSIONS

The project aimed to develop a music recommendation system using a variety of machine learning techniques and data analysis methods. The dataset used was the Spotify Tracks Dataset, which includes a wide range of audio features and metadata for numerous tracks. Through this dataset, several preprocessing steps, clustering algorithms, and evaluation metrics were employed to achieve the final recommendation system.

The first step involved using SMOTE to handle class imbalance, ensuring that the model training was not biased toward the more frequent classes. Standard scaling was applied to standardize the features, followed by PCA for dimensionality reduction. The explained variance plot indicated that using eight principal components would retain 80% of the information, which balanced dimensionality reduction with information preservation.

Clustering was performed using KMeans, DBSCAN, and an autoencoder-based approach. The elbow method was used to determine the optimal number of clusters for KMeans, identifying that nine clusters provided a good balance between the complexity and the explained variance. For DBSCAN, the k-distance graph helped in determining the optimal epsilon value, ensuring meaningful clusters based on density. The clustering results were visualized using PCA and LDA, providing a clear representation of the data distribution and the effectiveness of each clustering method.

SHAP analysis was used to identify the most important features contributing to the clustering results. Features such as liveness, time_signature, and valence had significant impacts on the model output, highlighting the factors that distinguish different clusters of tracks.

Comparative analysis of the clustering methods using silhouette scores and Davies-Bouldin scores revealed that the autoencoder combined with KMeans performed the best. The silhouette score for the autoencoder-based clustering was significantly higher, indicating better-defined clusters. The Davies-Bouldin score was lower for the autoencoder-based method, suggesting more compact and well-separated clusters.

The music recommendation system was designed to provide recommendations based on the closest tracks within the same cluster and across different clusters. The results included a recommendation of the 11 nearest tracks from the same cluster and three nearest tracks from each cluster. This approach ensures that the recommendations are not only similar but also diverse, enhancing the user experience by exposing them to a variety of tracks that they might enjoy.

In conclusion, this project demonstrated the effective use of machine learning techniques for developing a music recommendation system. The combination of data preprocessing, feature selection, clustering, and evaluation metrics provided a robust framework for analyzing and recommending music tracks. The findings highlight the importance of using advanced techniques like autoencoders for capturing complex patterns in the data, leading to more accurate and meaningful recommendations. This project lays the groundwork for further enhancements and refinements in music recommendation systems.

ACKNOWLEDGMENTS

This paper was built for educational purposes for the lesson “Advanced Machine Learning” of the MSc program “Artificial Intelligence” of Department of Informatics at Aristotle University of Thessaloniki.

REFERENCES

[1] Pandya, Maharshi. "Spotify Tracks Dataset." Kaggle, 2020.

[2] Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). "SMOTE: Synthetic Minority Over-sampling Technique." Journal of Artificial Intelligence Research, 16, 321-357

[3] Jolliffe, I. T. (2002). "Principal Component Analysis."

Springer.

[4] Davies, D. L., & Bouldin, D. W. (1979). "A Cluster Separation Measure." IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-1(2), 224-227.

[5] Rousseeuw, P. J. (1987). "Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis." Journal of Computational and Applied Mathematics, 20, 53-65.

[6] Hinton, G. E., & Salakhutdinov, R. R. (2006). "Reducing the Dimensionality of Data with Neural Networks."