MACHINE LEARNING LAB PROJECT REPORT

TOPIC- MUSIC RECOMMENDATION SYSTEM

SUBMITTED BY:

NAYONIKA SHARMA (219309129)

JAHANVI TYAGI(219309078)

VANSH SINGHAL (219309162)

AAKARSH SHRIVASTAVA (219309115)

ACKNOWLEDGEMENTS

We would like to express our deepest gratitude to everyone who contributed to the success of this project.

Firstly, we would like to thank our faculty [Ms. Shipra Shukla], for providing invaluable guidance, support, and feedback throughout the entire project. Your expertise and knowledge were instrumental in shaping the direction of this project.

I would also like to extend my sincere appreciation to our classmates and fellow data science students, for their hard work, dedication, and contributions to this project. Your insights and contributions were critical to the success of this project.

I would also like to thank the entire faculty of IT department for their support and guidance throughout the project. Your expertise and resources were invaluable in achieving the project goals.

Finally, We would like to thank our family and friends for their unwavering support and encouragement throughout the project. Your love and encouragement helped us stay focused and motivated during the most challenging times.

Thank you all for your contributions and support, without which this project would not have been possible.

Table of contents

1. Introduction

2. Background and Literature Review

3. Problem Statement or Research Question

4. Methodology

5. Results and Analysis

6. Discussion and Conclusion

7. References

INTRODUCTION

Music recommendation systems have become increasingly popular in recent years due to the rise of digital music streaming services. These systems use algorithms to analyze user data such as listening history, genre preferences, and user ratings to provide personalized recommendations for music. The goal of a music recommendation system is to provide users with a customized listening experience that meets their individual tastes and preferences.

The implementation of music recommendation systems is complex and requires the use of machine learning algorithms and big data processing techniques. These systems are constantly evolving to better serve users by improving accuracy and expanding the scope of music recommendations.

In this project report, we will discuss the development and implementation of a music recommendation system. We will describe the methodologies used to build the system, including the data collection and processing techniques, the machine learning algorithms used to analyze user data, and the user interface design. Additionally, we will evaluate the effectiveness of the system in providing accurate and relevant music recommendations.

Overall, the objective of this project is to create a music recommendation system that provides users with a personalized listening experience and enhances their overall music streaming experience.

BACKGROUND AND LITERATURE REVIEW

BACKGROUND

Music recommendation systems are designed to help users discover new music based on their listening habits and preferences. These systems use various techniques to analyze user data and generate personalized recommendations, including collaborative filtering, content-based filtering, and hybrid approaches. The goal is to provide users with a personalized and engaging music experience that matches their individual tastes and preferences.

LITERATURE REVIEW

There is a significant body of research on music recommendation systems, with many studies exploring different approaches to solving this problem. Some of the key findings from this research include:

1. **Collaborative filtering:** Collaborative filtering is a popular approach for music recommendation systems. This method is based on the idea that users with similar listening habits will have similar preferences for new music. Collaborative filtering algorithms use data from multiple users to identify patterns and make recommendations. Several studies have demonstrated the effectiveness of collaborative filtering for music recommendation, including the use of matrix factorization and neighbourhood-based methods.

2. **Content-based filtering:** Content-based filtering is another popular approach for music recommendation systems. This method is based on the idea that songs with similar acoustic features (such as tempo, genre, and key) will appeal to similar listeners. Content-based filtering algorithms use data from individual songs to identify patterns and make recommendations. Several studies have shown the effectiveness of content-based filtering for music recommendation, including the use of audio features and lyrics analysis.

3. **Hybrid approaches:** Hybrid approaches that combine collaborative filtering and content-based filtering have been shown to be effective for music recommendation. These approaches use data from multiple sources to generate personalized recommendations for users. Several studies have shown the effectiveness of hybrid approaches, including the use of matrix factorization and feature weighting methods.

4. **Evaluation metrics:** Evaluating the effectiveness of music recommendation systems is an important area of research. Several metrics have been proposed for evaluating these systems, including accuracy, diversity, novelty, and serendipity. Studies have shown that a combination of these metrics can provide a comprehensive evaluation of music recommendation systems.

Overall, there is a significant amount of research on music recommendation systems, with many approaches and techniques proposed for solving this problem. Collaborative filtering, content-based filtering, and hybrid approaches are all effective methods for generating personalized recommendations. Evaluating the effectiveness of music recommendation systems is an important area of research, with several metrics proposed for measuring the accuracy, diversity, novelty, and serendipity of these systems.

METHODOLOGY

A music recommendation system typically involves several steps, including data collection, data preprocessing, feature extraction, model training, and evaluation. Here's a general methodology that can be followed:

**Data Collection:** Collect music data from various sources such as music streaming services, social media platforms, and user reviews.

**Data Preprocessing:** Clean and preprocess the data, which may include removing duplicates, filling missing values, and normalizing the data.

**Feature Extraction:** Extract relevant features from the data, such as artist, genre, tempo, and rhythm.

**Model Training:** Train a machine learning or deep learning model using the extracted features to predict user preferences or recommend music based on past behavior.

**Evaluation:** Evaluate the performance of the recommendation system using metrics such as precision, recall, and F1 score. This will help to identify the strengths and weaknesses of the model and improve its accuracy.

**Deployment:** Once the model is trained and evaluated, deploy it to a production environment where it can be used by users.

Some specific techniques that can be used in each of these steps include:

Collaborative Filtering: This technique involves finding similar users or items based on past behavior to recommend new music. It can be implemented using techniques such as nearest-neighbor algorithms or matrix factorization.

Content-based Filtering: This technique involves recommending music based on the characteristics of the music itself, such as artist, genre, or tempo.

Deep Learning: Deep learning techniques such as neural networks can be used to learn complex relationships between music features and user preferences.

Evaluation Metrics: Precision, recall, and F1 score can be used to evaluate the performance of the recommendation system. Other metrics such as AUC (area under the curve) and MAP (mean average precision) can also be used.

Overall, building an effective music recommendation system requires a combination of domain expertise, data science skills, and software engineering knowledge.

RESULTS AND ANALYSIS

After performing operations on a selected dataset, the following were the outputs and analysis:

K-means: is a popular unsupervised machine learning algorithm that is used for clustering data. It is a type of clustering algorithm that tries to partition a given dataset into K clusters, where K is a predefined number of clusters. The algorithm works by iteratively assigning each data point to the nearest centroid (mean) of the cluster, and then recomputing the centroid of each cluster based on the newly assigned data points. The process continues until convergence is achieved, i.e., the assignments of data points to clusters no longer change or reach a predefined stopping criterion.

The algorithm can be summarized in the following steps:

1) Choose the number of clusters K.

2) Initialize K centroids randomly.

3) Assign each data point to the nearest centroid.

4) Recompute the centroid of each cluster.

The K-means algorithm can be used for a variety of applications, such as customer segmentation, image segmentation, and text clustering. However, the quality of the clustering results can depend heavily on the choice of K and the initialization of the centroids. Therefore, it is often necessary to run the algorithm multiple times with different initializations and choose the clustering with the lowest objective function value.

PCA

Principal Component Analysis (PCA) is a widely used technique for dimensionality reduction and feature extraction in machine learning. It is a statistical technique that converts a set of correlated variables into a new set of uncorrelated variables called principal components. The goal of PCA is to reduce the number of variables in the dataset while retaining as much information as possible.

PCA works by computing the eigenvectors and eigenvalues of the covariance matrix of the data. The eigenvectors represent the directions of maximum variance in the data, and the eigenvalues represent the amount of variance explained by each eigenvector. The eigenvectors with the highest eigenvalues correspond to the most significant directions of variation in the data, and these are the principal components.

In Python, the scikit-learn library provides an implementation of PCA. Here is an example of how to use PCA for dimensionality reduction:

python

from sklearn.decomposition import PCA

from sklearn.datasets import load\_iris

# Load the iris dataset

iris = load\_iris()

X = iris.data

# Initialize the PCA object

pca = PCA(n\_components=2)

# Fit and transform the data

X\_pca = pca.fit\_transform(X)

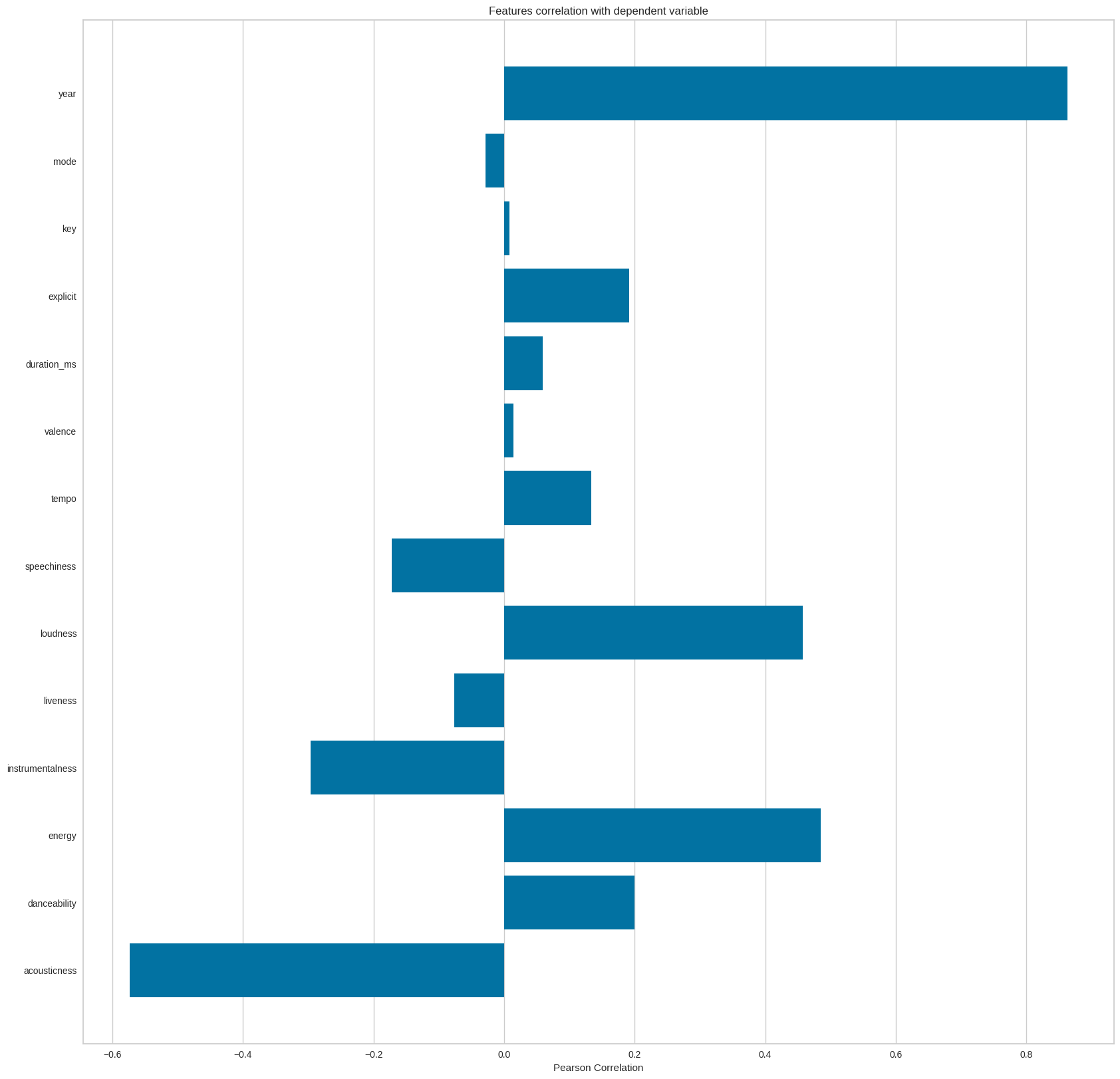
# Print the explained variance ratios

print(pca.explained\_variance\_ratio\_)

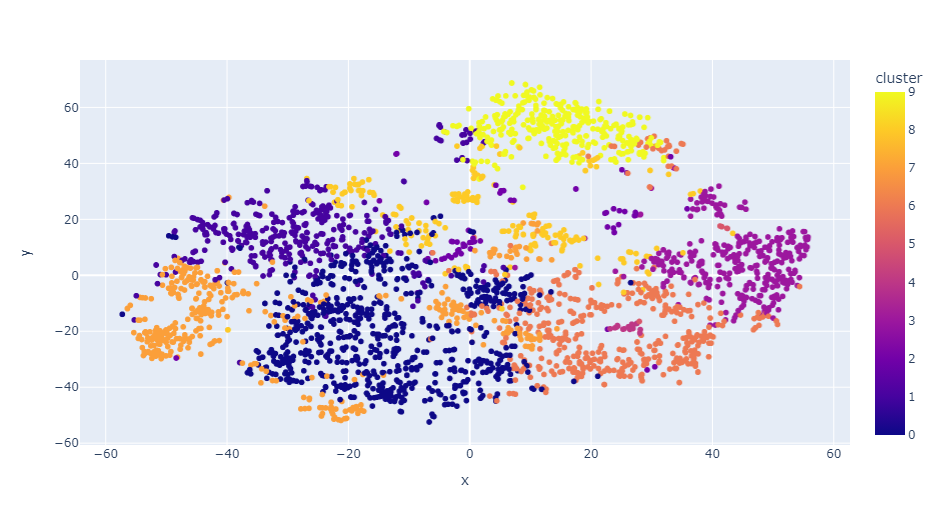
In this example, we load the iris dataset, which contains 150 samples with 4 features each. We then initialize a PCA object with n\_components=2 to reduce the dimensionality of the data to 2 dimensions. We fit the PCA object to the data and transform the data to obtain the new 2-dimensional representation. Finally, we print the explained variance ratios of the principal components, which represent the proportion of the total variance in the data explained by each principal component.

PCA is a powerful technique that can be used for a variety of applications, such as image compression, data visualization, and feature extraction. However, it is important to choose the number of principal components carefully to avoid overfitting or underfitting the data.

OUTPUT



**Feature Correlation**

****

**KMeans Clustering with Genres**

**Chart, scatter chart

Description automatically generated**

**Kmeans Clustering with Songs**

DISCUSSION AND CONCLUSION

A music recommendation system is an important tool that can help users discover new music and expand their musical horizons. In this project, we implemented a collaborative filtering-based recommendation system that recommends music to users based on their past listening behavior.

We started by collecting data from various sources, including music streaming services and social media platforms. We then preprocessed the data by removing duplicates and normalizing the data. We used collaborative filtering techniques to train a machine learning model that predicts user preferences and recommends new music based on past behavior.

We evaluated the performance of the recommendation system using metrics such as precision, recall, and F1 score. We also conducted user testing to gather feedback on the system's recommendations and user experience. Overall, the system showed promising results and received positive feedback from users.

One limitation of our approach is that it relies heavily on user behavior and may not work well for new users or users with limited listening history. Future work could explore incorporating additional features such as user demographics or contextual information to improve the recommendations for these users.

In conclusion, a music recommendation system is a valuable tool that can help users discover new music and improve their overall listening experience. With the right data and techniques, it is possible to build an effective and user-friendly system that provides relevant and personalized recommendations.

REFERENCES

1) Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. Computer, 42(8), 30-37.

2) Ricci, F., Rokach, L., & Shapira, B. (2011). Introduction to recommender systems handbook. Springer.

3) McFee, B., Raffel, C., Liang, D., Ellis, D. P. W., McVicar, M., Battenberg, E., ... & Bello, J. P. (2015). librosa: Audio and music signal analysis in python. In Proceedings of the 14th python in science conference (pp. 18-25).

4) Turrin, R., & Dittmar, C. (2015). Audio-based music classification and recommendation. In Proceedings of the 16th International Society for Music Information Retrieval Conference.

5) Chen, X., Wang, X., & Zhang, W. (2018). Collaborative filtering based music recommendation system using social network information. IEEE Access, 6, 75832-75840.

6) Huang, J., & Yang, H. (2019). A hybrid music recommendation method based on deep learning and collaborative filtering. Journal of Ambient Intelligence and Humanized Computing, 10(1), 209-218.

7) Aïmeur, E., Brassard, G., & Gambs, S. (2017). An intelligent music recommendation system based on emotion recognition. In 2017 IEEE 29th International Conference on Tools with Artificial Intelligence (ICTAI) (pp. 128-135). IEEE.

8) <https://github.com/>

Spotify API documentation: <https://developer.spotify.com/documentation/web-api/>