Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

SCHOOL OF COMPUTER SCIENCE ENGINEERING AND INFORMATION SYSTEMS

Course Code : CSC3005

Course Name : Fundamentals of Data Analytics

FINAL REVIEW

SLOT : D1+TD1

Music Recommendation System

TEAM MEMBERS:

S.Pradeep **(22BCS0075)**

S. Jeeva **(22BCS0142)**

P. Dhanush **(22BCS0167)**

FACULTY NAME:

Dr. P. Mohana Priya,

Assistant Professor Sr.Grade II

Contents

<u>Abstract :</u>

Music is an essential part of our daily lives. With the rise of streaming platforms, users have access to an overwhelming amount of music, making it challenging to discover new songs and artists that match their musical tastes. To solve this problem, music recommendation systems have become increasingly popular. These systems use machine learning algorithms to analyze user behavior and preferences, and provide personalized recommendations. In this project, we have developed a music recommendation system using Python. We have used a dataset consisting of user listening history, song information, and user preferences to train and test the recommendation model. The system is built using two popular recommendation techniques: collaborative filtering and content-based filtering. Collaborative filtering is a technique that recommends items based on the preferences of similar users. In our system, we have used user-item collaborative filtering, where we identify users who have similar listening histories and recommend songs that they have enjoyed but have not listened to yet. Content-based filtering, on the other hand, recommends items based on their characteristics. In our system, we have used contentbased filtering to recommend songs based on their 4 genre, mood, and tempo, which we have extracted using audio analysis libraries in Python. We have implemented our system using Python libraries such as Pandas, NumPy, and Scikit-learn. We have also used the Flask web framework to create a user interface for our system. The user can input their preferences, such as favorite artists or genres, and receive personalized recommendations based on the collaborative filtering and content-based filtering techniques. our music recommendation system using Python provides an efficient and effective way for users to discover new songs and artists that match their musical tastes. It demonstrates the potential of machine learning algorithms to improve the user experience inthe music industry.

Introduction:

In today's digital age, music is more accessible than ever before. With countless streaming platforms and services at our fingertips, discovering new music can be both exciting and overwhelming. To help navigate this vast musical landscape, recommendation systems have become invaluable tools. These systems leverage algorithms to analyze user preferences, historical listening data, and music attributes to suggest personalized recommendations. Whether you're searching for the next hit song or exploring niche genres, music recommendation systems offer tailored suggestions to enhance your listening experience. Let's delve deeper into how these systems work and how they've revolutionized the way we discover and enjoy music.

Objective

Personalization: Create a system that tailors recommendations based on the user's listening history, preferences, and behavior patterns.

Accuracy: Ensure that the recommended music aligns closely with the user's tastes and interests, minimizing irrelevant suggestions.

Diversity: Offer a diverse range of recommendations to expose users to new artists, genres, and styles they may enjoy but haven't explored yet.

Real-time Updates: Implement a system that continuously updates recommendations based on the user's evolving preferences and trends in the music industry.

User Feedback Integration: Allow users to provide feedback on recommended songs to improve the accuracy of future recommendations.

Seamless Integration: Integrate the recommendation system seamlessly into existing music streaming platforms or applications to enhance user experience.

Discovery: Facilitate music discovery by suggesting related artists, songs, or genres that users may find interesting based on their current preferences.

PROBLEM STATEMENT:

In today's digital music landscape, users are inundated with an overwhelming amount of music choices across various platforms and genres. However, the abundance of options often leads to decision paralysis and difficulty in discovering new music that aligns with individual tastes and preferences. Existing music recommendation systems, while providing some level of guidance, often fall short in delivering truly personalized and diverse recommendations. The primary challenge lies in accurately predicting user preferences based on limited data points such as listening history, likes, and dislikes. Traditional recommendation algorithms may struggle to capture the nuances of individual taste, resulting in recommendations that are either too generic or completely off the mark. This lack of accuracy can lead to user frustration and a diminished overall experience. Furthermore, existing recommendation systems often overlook the importance of diversity in music suggestions. While it's essential to cater to users' existing preferences, true music discovery requires exposure to a wide range of artists, genres, and styles. Without diverse recommendations, users may miss out on discovering hidden gems or exploring new musical territories.

LITERATURE REVIEW

s.no	Paper title	Author & Year of Publication	pros	cons
1.	Contentbased Recommender Systems	Pasquale Lops, Marco de Gemmis and Giovanni Semeraro. 2010	Learning of profile is made easy. Quality improves over time. Considers implicit feedback	Does not completely Overcome the problem of overspecialization and serendipity.
2.	Hybrid Recommender Systems	Robin Burke 2010	The survey shows combine techniques for improved performance. It improves the user preferences for suggesting items to users.	
3.	Association rule Mining for recommendation system on the book sale	Luo Zhenghua. 2012	The website based on this has shown great performance.	It does not recommend quality content to the users. Does not consider new user cold start problem Not very efficient in terms of performance

4.	Collaborative filtering for recommender systems: Userbased and Itembased CF	Gilbert Badaro, Hazem Hajj, Wassim El-Hajj and Lama Nachman. 2013	solves the problem of finding the ratings of unrated items in a user-item ranking matrix. It improves the data sparsity problem.	It does not consider the demographic features which would give better results and solve the user coldstart problem.
5.	ContentBased Filtering, Collaborative Filtering, and Association Rule Mining	Anand Shanker Tewari, Abhay Kumar, and Asim Gopal Barman. 2014	It considers various parameters like content & quality of the book by doing collaborative filtering of rating of other buyers. It does not have performance problems. It builds the recommendation offline.	It still lacks the new user cold-start problem.

<u>DATA EXPLOARATION:</u>

Music Catalog Data:

Explore metadata about songs, albums, and artists.

Investigate attributes such as genre, release date, popularity, and acoustic features (e.g., tempo, danceability, energy).

Identify any inconsistencies or missing values in the data.

Visualize distributions of genres, popularity scores, and other relevant attributes

User Interaction Data:

Analyze user listening history, including played tracks, skipped tracks, likes, dislikes, and ratings.

Examine patterns in user behavior, such as favorite genres, artists, or time of day for listening sessions.

Calculate user engagement metrics, such as session duration, frequency of interactions, and diversity of listened genres.

Identify any outliers or anomalies in user behavior.

Contextual Data:

Incorporate contextual factors such as time, location, weather, and user mood.

Analyze how these factors influence music preferences and listening patterns.

Visualize trends in music consumption based on contextual variables.

Identify correlations between contextual factors and user interactions with music.

Collaborative Filtering Data:

Explore user-item interaction matrices, representing user preferences for different songs or artists.

Analyze patterns of user similarity or dissimilarity based on their listening behavior.

Investigate the sparsity of the user-item matrix and its impact on recommendation quality.

Visualize user-item matrices and similarity matrices to gain insights into user preferences and item relationships.

Content-Based Filtering Data:

Examine features extracted from music audio files, such as spectrograms, MFCCs (Mel-frequency cepstral coefficients), and other acoustic features.

Analyze correlations between acoustic features and user preferences.

Visualize distributions of acoustic features for different genres or artists.

Identify patterns in feature space that distinguish between different music styles or moods.

Data Preprocessing:

Clean and preprocess data to handle missing values, outliers, and inconsistencies.

Normalize or scale features as needed for modeling.

Encode categorical variables and timestamps into numerical representations.

Split data into training, validation, and test sets for model development and evaluation.

METHODS

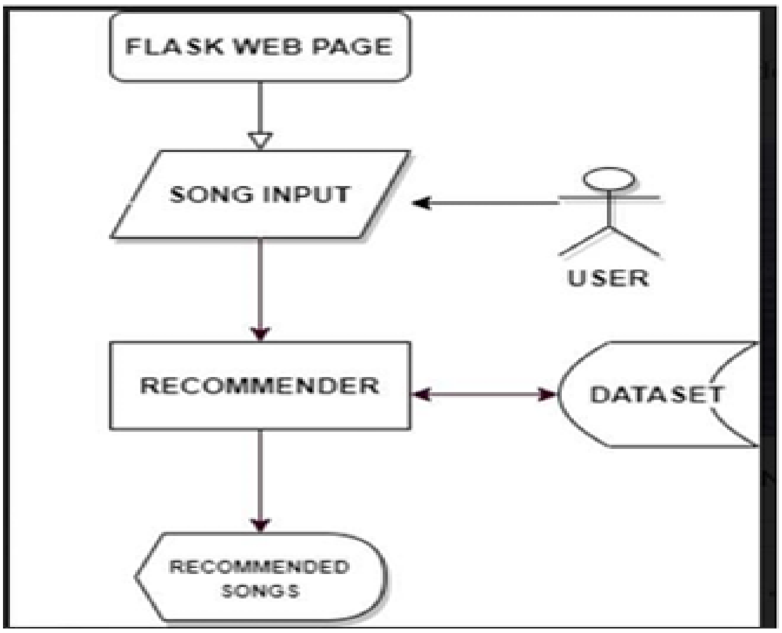
• A music recommendation system is a type of recommender system that suggests songs or artists to users based on their preferences, behavior, and contextual information. The goal is to provide personalized recommendations that match users' tastes, interests, and current mood, thereby enhancing their music listening experience. These systems typically employ various techniques, algorithms, and data sources to analyze user data and music metadata to generate relevant suggestions.

1. Collaborative Filtering: This method analyzes user behavior and preferences to recommend music items that similar users have enjoyed. It identifies patterns in user interactions, such as listening history, ratings, and social connections, to suggest songs or artists that align with the user's taste.
2. Content-Based Filtering: Content-based filtering recommends music items based on their attributes and characteristics, such as genre, tempo, mood, and instrumentation. It analyzes features extracted from songs, such as audio signals, lyrics, and metadata, to

match them with users' preferences and provide personalized recommendations.

3. Hybrid Recommendation Systems: Hybrid systems combine collaborative filtering and content-based filtering approaches to leverage the strengths of both methods. By integrating user behavior analysis with song feature analysis, these systems offer more accurate and diverse recommendations, catering to a wider range of user preferences.
4. Deep Learning Models: Deep learning techniques, such as neural networks, can be employed to build recommendation systems that can handle large and complex datasets. These models can learn intricate patterns from user behavior and music features, allowing them to generate highly personalized recommendations with improved accuracy.

MODELING:



RESULTS:

CORELATION MATRIX OUTPUT:

```
[1] import os
    import numpy as np
    import pandas as pd

    import seaborn as sns
    import plotly.express as px
    import matplotlib.pyplot as plt
    %matplotlib inline
    import tensorflow as tf

    from sklearn.cluster import KMeans
    from sklearn.preprocessing import StandardScaler
    from sklearn.pipeline import Pipeline
    from sklearn.manifold import TSNE
    from sklearn.decomposition import PCA
    from sklearn.metrics import euclidean_distances
    from scipy.spatial.distance import cdist
    from matplotlib import pyplot as plt
    from matplotlib.pyplot import plot
    from sklearn.model_selection import train_test_split

    import warnings
    warnings.filterwarnings("ignore")
```

Read data

```
[2] data = pd.read_csv("/content/data.csv")
    genre_data = pd.read_csv('/content/data_by_genres.csv')
    year_data = pd.read_csv('/content/data_by_year.csv')
    artist_data = pd.read_csv('/content/data_by_artist.csv')
```

print(data.info())

```
print(data.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 54582 entries, 0 to 54581
Data columns (total 19 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   valence           54582 non-null  float64
 1   year              54582 non-null  int64
 2   acousticness      54582 non-null  float64
 3   artists           54581 non-null  object
 4   danceability      54581 non-null  float64
 5   duration_ms       54581 non-null  float64
 6   energy            54581 non-null  float64
 7   explicit          54581 non-null  float64
 8   id                54581 non-null  object
 9   instrumentalness  54581 non-null  float64
 10  key               54581 non-null  float64
 11  liveness          54581 non-null  float64
 12  loudness          54581 non-null  float64
 13  mode              54581 non-null  float64
 14  name              54581 non-null  object
 15  popularity        54581 non-null  float64
 16  release_date      54581 non-null  object
 17  speechiness       54581 non-null  float64
 18  tempo             54581 non-null  float64
dtypes: float64(14), int64(1), object(4)
memory usage: 7.9+ MB
None
```

print(genre_data.info())

```
[4] print(genre_data.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2973 entries, 0 to 2972
Data columns (total 14 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   mode              2973 non-null   int64  
 1   genres            2973 non-null   object 
 2   acousticness      2973 non-null   float64
 3   danceability      2973 non-null   float64
 4   duration_ms       2973 non-null   float64
 5   energy            2973 non-null   float64
 6   instrumentalness  2973 non-null   float64
 7   liveness          2973 non-null   float64
 8   loudness          2973 non-null   float64
 9   speechiness       2973 non-null   float64
 10  tempo             2973 non-null   float64
 11  valence           2973 non-null   float64
 12  popularity        2973 non-null   float64
 13  key               2973 non-null   int64  
dtypes: float64(11), int64(2), object(1)
memory usage: 325.3+ KB
None
```

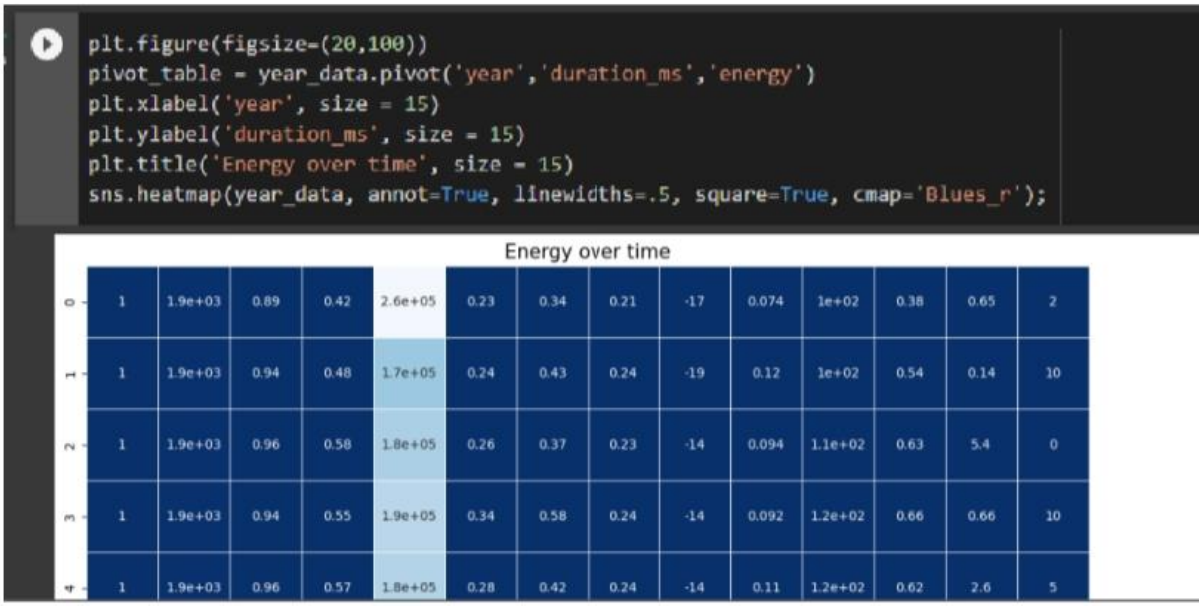
HEAT MAPS year_data.pivot('year','duration_ms','energy').head()

```
[7] year_data.pivot('year','duration_ms','energy').head()
```

duration_ms	156881.657475	165469.746479	168999.412815	171553.425466	177942.362162	182227.944500	184986.924460	184993.598374	189356.126298	191046.707627
year										
1921	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1922	NaN	0.237815	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1923	NaN	NaN	NaN	NaN	0.262406	NaN	NaN	NaN	NaN	NaN
1924	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0.344347
1925	NaN	NaN	NaN	NaN	NaN	NaN	0.278594	NaN	NaN	NaN

5 rows × 100 columns

```
plt.figure(figsize=(20,100))
pivot_table = year_data.pivot('year','duration_ms','energy')
plt.xlabel('year', size = 15)
plt.ylabel('duration_ms', size = 15)
plt.title('Energy over time', size = 15)
sns.heatmap(year_data, annot=True, linewidths=.5, square=True, cmap='Blues_r');
```



Correlation Matrix:

```
df = pd.read_csv('/kaggle/input/-spotify-tracks-dataset/dataset.csv')
df.head()
```

```
plot_df = pd.read_csv('/content/dataset.csv')
plot_df.head()
```

	Unnamed: 0	track_id	artists	album_name	track_name	popularity	duration_ms	explicit	danceability	energy	...	loudness	mode	speech
0	0	5SuOikwiRyPMVoIQDJUgSV	Gen Hoshino	Comedy	Comedy	73	230666	False	0.676	0.4610	...	-6.746	0	
1	1	4qPNDBW1i3p13qLCt0Ki3A	Ben Woodward	Ghost (Acoustic)	Ghost - Acoustic	55	149610	False	0.420	0.1660	...	-17.235	1	
2	2	1iJBSr7s7jYXzM8EGcbK5b	Ingrid Michaelson;ZAYN	To Begin Again	To Begin Again	57	210826	False	0.438	0.3590	...	-9.734	1	
3	3	6lfxq3CG4xtTiEg7opyCyx	Kina Grannis	Crazy Rich Asians (Original Motion Picture Sou...	Can't Help Falling In Love	71	201933	False	0.266	0.0596	...	-18.515	1	
4	4	5vjLSffimiIP26QG5WcN2K	Chord Overstreet	Hold On	Hold On	82	198853	False	0.618	0.4430	...	-9.681	1	

5 rows × 21 columns

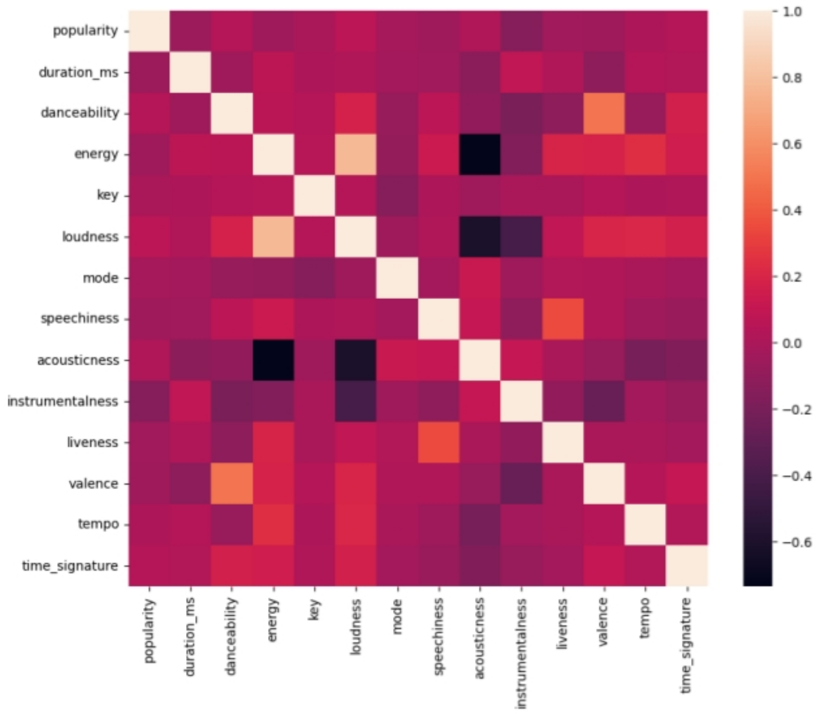
```
df = df[['popularity','duration_ms','danceability', 'energy',
'key', 'loudness', 'mode', 'speechiness', 'acousticness',
'instrumentalness', 'liveness', 'valence', 'tempo', 'time_signature']]
df.head()
```

```
[11] df = df[['popularity','duration_ms','danceability', 'energy',
        'key', 'loudness', 'mode', 'speechiness', 'acousticness',
        'instrumentalness', 'liveness', 'valence', 'tempo', 'time_signature']]
     df.head()
```

	popularity	duration_ms	danceability	energy	key	loudness	mode	speechiness	acousticness	instrumentalness	liveness	valence	tempo	time_signature
0	73.0	230666.0	0.676	0.4610	1.0	-6.746	0.0	0.1430	0.0322	0.000001	0.3580	0.715	87.917	4.0
1	55.0	149610.0	0.420	0.1660	1.0	-17.235	1.0	0.0763	0.9240	0.000006	0.1010	0.267	77.489	4.0
2	57.0	210826.0	0.438	0.3590	0.0	-9.734	1.0	0.0557	0.2100	0.000000	0.1170	0.120	76.332	4.0
3	71.0	201933.0	0.266	0.0596	0.0	-18.515	1.0	0.0363	0.9050	0.000071	0.1320	0.143	181.740	3.0
4	82.0	198853.0	0.618	0.4430	2.0	-9.681	1.0	0.0526	0.4690	0.000000	0.0829	0.167	119.949	4.0

```
plt.figure(figsize=(10,8))
sns.heatmap(data=cor)
```

```
[13] plt.figure(figsize=(10,8))
     sns.heatmap(data=cor)
```

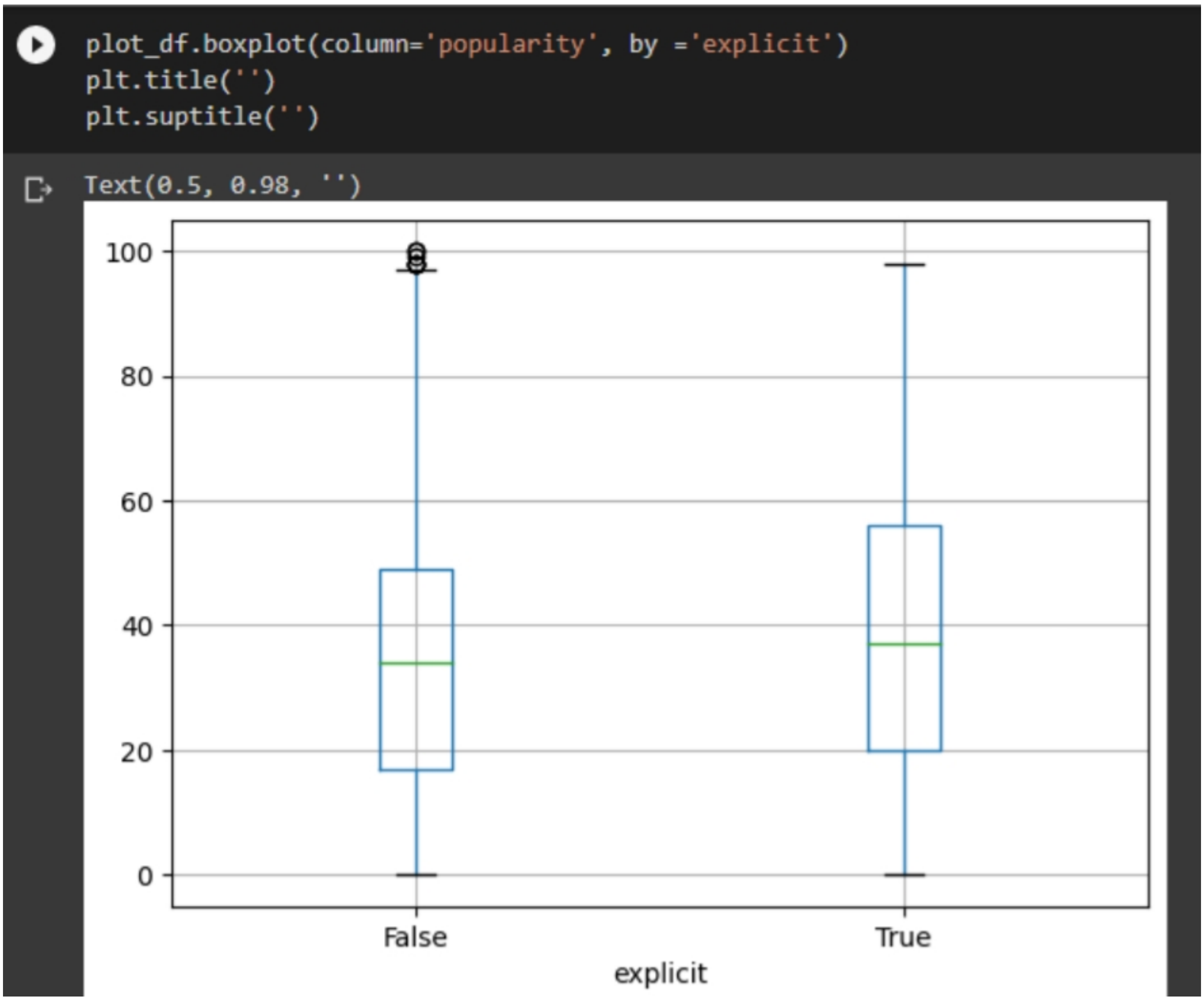


Box Plot

plot_df.boxplot(column='popularity', by ='explicit')

plt.title('')

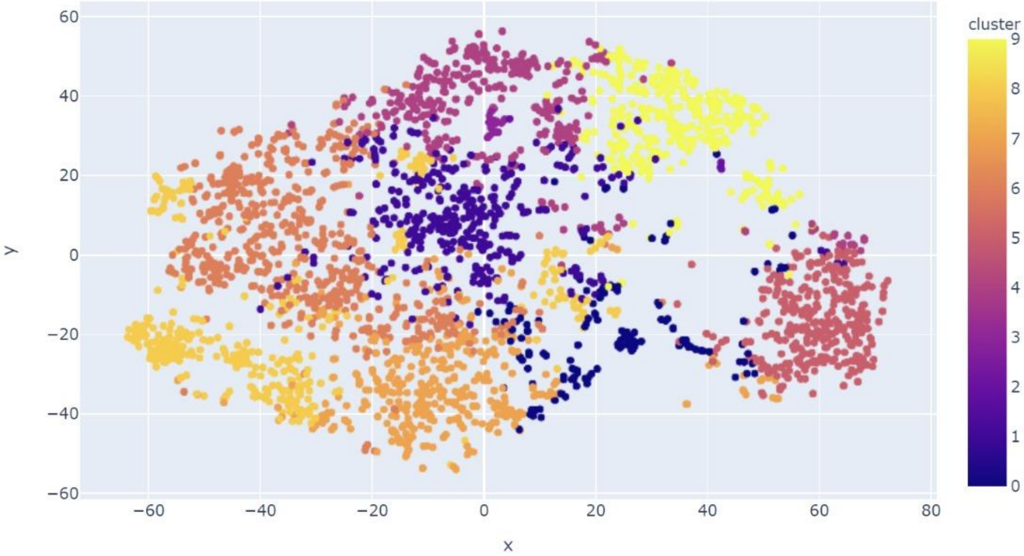
plt.suptitle('')



K-MEANS CLUSTERING:

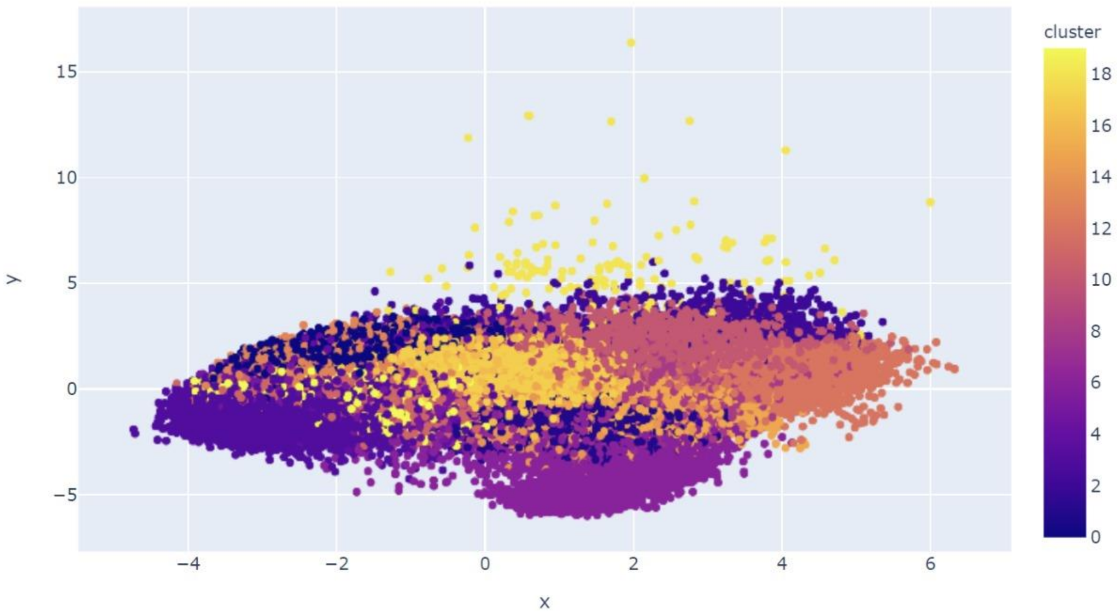
```
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import Pipeline
cluster_pipeline = Pipeline([('scaler', StandardScaler()), ('kmeans', KMeans(n_clusters=10, n_jobs=-1))])
X = genre_data.select_dtypes(np.number)
cluster_pipeline.fit(X)
```

```
genre_data['cluster'] = cluster_pipeline.predict(X)

# Visualizing the Clusters with t-SNE

from sklearn.manifold import TSNE

tsne_pipeline = Pipeline([('scaler', StandardScaler()), ('tsne',
TSNE(n_components=2, verbose=1))])

genre_embedding = tsne_pipeline.fit_transform(X)

projection = pd.DataFrame(columns=['x', 'y'], data=genre_embedding)

projection['genres'] = genre_data['genres']

projection['cluster'] = genre_data['cluster']

fig = px.scatter(
    projection, x='x', y='y', color='cluster', hover_data=['x', 'y', 'genres'])

fig.show()

[t-SNE] Computing 91 nearest neighbors...

[t-SNE] Indexed 2973 samples in 0.005s...

[t-SNE] Computed neighbors for 2973 samples in 0.322s...

[t-SNE] Computed conditional probabilities for sample 1000 / 2973

[t-SNE] Computed conditional probabilities for sample 2000 / 2973

[t-SNE] Computed conditional probabilities for sample 2973 / 2973

[t-SNE] Mean sigma: 0.777516
```




```
song_cluster_pipeline = Pipeline([('scaler', StandardScaler()),
('kmeans', KMeans(n_clusters=20,
verbose=False, n_jobs=4))
], verbose=False)


X = data.select_dtypes(np.number)
number_cols = list(X.columns)
song_cluster_pipeline.fit(X)
song_cluster_labels = song_cluster_pipeline.predict(X)
data['cluster_label'] = song_cluster_labels
# Visualizing the Clusters with PCA
from sklearn.decomposition import PCA
pca_pipeline = Pipeline([('scaler', StandardScaler()), ('PCA',
PCA(n_components=2))])
song_embedding = pca_pipeline.fit_transform(X)
projection = pd.DataFrame(columns=['x', 'y'], data=song_embedding)
```

```
projection['title'] = data['name']
projection['cluster'] = data['cluster_label']

fig = px.scatter(
    projection, x='x', y='y', color='cluster', hover_data=['x', 'y', 'title'])
fig.show()
```



CONCLUSION AND FUTURE DEVELOPEMENT:

• A Robust Music Recommendation System Can Greatly Enhance User Experience By Offering Personalized Suggestions Tailored To Individual Tastes And Preferences. By Leveraging Advanced Algorithms And User Data Analysis, These Systems Can Help Users Discover New Music, Rediscover Old Favorites, And Continuously Engage With Their Musical Interests. As Technology Continues To Evolve, The Potential For Music Recommendation Systems To Refine Their Accuracy And Effectiveness Remains Promising, Making Them Indispensable Tools For Music Enthusiasts Worldwide.

REFERENCES:

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

2. Hu, L., Li, X., & Cui, L. (2018). Context-aware music recommendation based on deep learning. IEEE Access, 6, 13240-13248.

3.Oord, A. V. D., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., ... & Kavukcuoglu, K. (2016). Wavenet: A generative model for raw audio. arXiv preprint arXiv:1609.03499.

4. Ricci, F., Rokach, L., & Shapira, B. (2011). Introduction to recommender systems handbook. Springer Science & Business Media.

5. Celma, O ., & Herrera, P. (2008). A new approach to evaluating novel recommendations. In Proceedings of the 2nd ACM workshop on Music information retrieval with user-centered and multimodal strategies (pp. 1-8).

6.Zhang, J., & Luo, J. (2019). Scalable music recommendation with poisson factorization. IEEE Transactions on Multimedia, 21(3), 557 567.

7. Luo Zhenghua, "Realization of Individualized Recommendation System on Books Sale," IEEE 2012 International Conference on Management of e-Commerce and e-Government. pp.10-13.

8. Tewari, A.S. Kumar, and Barman, A.G, "Book recommendation system based on combining features of content-based filtering, collaborative filtering and association rule mining," International Advance Computing Conference (IACC), IEEE, pp 500 – 503, April 2014.

9. Robin Burke, "Hybrid Recommender Systems: Survey and Experiments", California State University, Department of Information Systems and Decision Sciences, Vol. 12, No. 4, pp. 331-370,March 2012

10. Anil Poriya, Neev Patel, Tanvi Bhagat, and RekhaSharma, “Non Personalized Recommender SystemsandUser-basedCollaborative Recommender Systems”, International Journal of Applied Information Systems (IJAIS), FCS, Vol. 6, No. 9, March 2014.

DECLARATION

We hereby declare that the report entitled **"Music Recommendations system"** submitted by us, for the **CSC3005-Fundamentals of Data Analytics** to Vellore Institute of Technology is a record of bonafide work carried out by me under the supervision of **Dr.P.Mohana Priya.**

We further declare that the work reported in this report has not been submitted and will not be submitted, either in part or in full, for any other courses in this institute or any other institute or university.

Place:Vellore

Date:3.5.2024

S. Pradeep

Signature of the Candidate 1

S. Teena

Signature of the Candidate 2

P. Dost.

Signature of the Candidate 3