
SPOTIFY RECOMMENDATION SYSTEM

**Shashank Bangera^{*1}, Vaishnavi Nagaonkar^{*2}, Aditya Tiwari^{*3}, Saud Ansari^{*4},
Kanchan Talekar^{*5}**

^{*1,2,3,4}Under Graduate Student, Dept. Of Computer Science & Engineering (Artificial Intelligence and Machine Learning) Lokmanya Tilak College Of Engineering, Mumbai University, Navi-Mumbai, India.

^{*5}Professor, Dept. Of Computer Science & Engineering (Artificial Intelligence and Machine Learning), Navi-Mumbai, India.

DOI : <https://www.doi.org/10.56726/IRJMETS49566>

ABSTRACT

This research explores the augmentation of user-artist interaction on Spotify by developing and evaluating a content-based recommendation system. Departing from traditional reliance on song plays, our methodology enables users to input specific song preferences or manually adjust Spotify audio features, integrating advanced machine learning techniques with the platform's extensive audio feature metrics. By computing song similarities grounded in numerical audio feature data, the algorithm generates a curated list of tracks within the artist's playlist, aligning with the user's specified preferences. The primary aim is to catalyze an enriched exploration of an artist's catalog, fostering heightened traffic to both the artist's playlist and Spotify profile. The anticipated contributions extend to the broader discourse on personalized music discovery within the Spotify ecosystem, with potential implications for the evolving landscape of digital music platforms. This research seeks to bridge the gap between users and artists, offering a more tailored and engaging music exploration experience within the Spotify paradigm.

I. INTRODUCTION

In the ever-evolving realm of digital music consumption, the dynamic interplay between users and artists on streaming platforms has emerged as a critical facet of contemporary music ecosystems. Within this context, Spotify stands as a prominent platform where the convergence of technology and musical expression shapes the user experience. Central to this interface is the concept of personalized music discovery, a domain that continues to captivate researchers and industry stakeholders alike. This research endeavors to delve into the intricacies of user-artist interaction on Spotify, specifically addressing the imperative to augment engagement through the development of a content-based recommendation system. Departing from conventional metrics reliant on aggregate song plays, this study places emphasis on empowering users to actively participate in the curation of their music exploration by inputting specific song preferences or manually adjusting Spotify audio features. This approach is situated at the intersection of machine learning and Spotify's comprehensive audio feature metrics, aiming to redefine the landscape of personalized music discovery. As we embark on this exploration, it is pertinent to acknowledge the existing body of research within the broader field of personalized music recommendation systems. Previous studies have laid the groundwork by examining collaborative filtering techniques, user-based and item-based approaches, and the integration of contextual information. However, a notable gap remains in the exploration of content-based recommendations tailored to user input and Spotify's audio feature dataset. This paper seeks to contribute to this burgeoning field by providing a comprehensive analysis and evaluation of the proposed content-based recommendation system, thereby advancing our understanding of how personalized recommendations can enrich the user experience and foster a more direct connection between listeners and artists within the Spotify ecosystem.

II. METHODOLOGY

Data Collection and Retrieval

The foundation of this research relies on the Spotify API, facilitated through the Spotipy library, to access and retrieve playlist tracks. Playlist data, particularly track information, including unique track identifiers (track ID) and relevant audio features, is obtained through the use of Spotify playlist URIs and creator information.

Data Wrangling and Scaling

Upon retrieving the playlist tracks, a comprehensive data wrangling process ensues to isolate numeric features crucial for subsequent analyses. Employing the KMeans algorithm, numeric features are scaled to ensure uniformity and comparability. This meticulous preprocessing aims to create a robust foundation for subsequent modeling and clustering.

Modeling and Clustering

Utilizing the KMeans algorithm, the scaled numeric features undergo clustering to discern inherent patterns within the dataset. The ensuing clusters provide valuable insights into the distribution and grouping of tracks based on their audio features. To visualize these clusters, the Plotly library is employed, offering an interactive and informative representation.

Binary Classification Model

In parallel, a binary classification model, specifically a Logistic Regression model, is implemented to predict certain attributes within the dataset. This model is rigorously evaluated through the creation of a confusion matrix and further elucidated through LIME visualization, enhancing transparency and interpretability.

Analysis

Clustering Analysis

The results of the KMeans clustering analysis are presented, elucidating the identified clusters and their significance in the context of audio features. The visual representation of these clusters using Plotly serves as a powerful tool for comprehending the intricate relationships and patterns inherent in the dataset.

Binary Classification Model Evaluation

The binary classification model's performance is thoroughly evaluated through the construction and interpretation of a confusion matrix. This matrix offers a detailed breakdown of the model's predictive capabilities, facilitating a nuanced understanding of its strengths and limitations. LIME visualization further enhances model interpretability by providing insights into individual predictions.

Recommendation System Implementation

Building upon the insights derived from clustering and binary classification, the recommendation system is implemented. Leveraging the KMeans clusters and Spotify API through Spotipy, the system computes vectors and cosine distances to generate personalized recommendations. This process is detailed to underscore the systematic approach employed in transforming analytical findings into actionable user-centric recommendations.

Streamlit Web App Development

The methodology concludes with the development of a user-friendly Streamlit web application. This application incorporates text boxes and a sidebar for user input, allowing users to actively engage with the recommendation system. Additionally, the app provides descriptive information regarding audio features, fostering user understanding and participation.

Through the intricate amalgamation of Spotify API interactions, data wrangling, clustering, binary classification, and recommendation system implementation, this comprehensive methodology seeks to unravel the intricacies of user-artist interaction and playlist engagement within the Spotify ecosystem. The subsequent analysis endeavors to distill actionable insights from the amassed data, contributing to the broader discourse on personalized music discovery.

III. MODELING AND ANALYSIS

KMeans Clustering Algorithm Implementation

The KMeans clustering algorithm is deployed to identify inherent patterns within the scaled numeric features of playlist tracks. This algorithm partitions the dataset into distinct clusters based on the similarity of audio features, providing a nuanced understanding of the underlying structure.

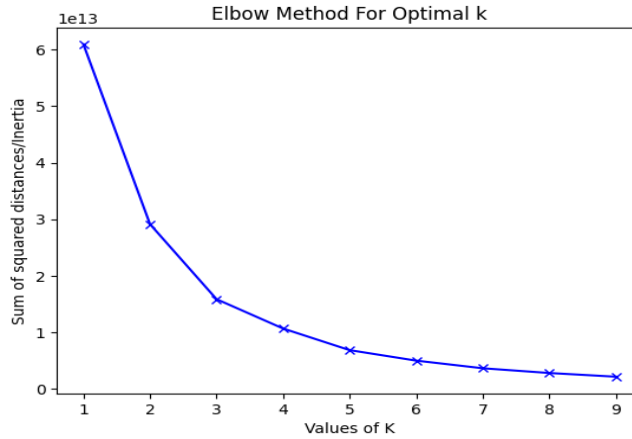


Figure 1: Elbow method for Optimal k.

Interpretation of Clusters

The resultant clusters are meticulously examined and interpreted to discern meaningful associations among tracks. By delving into the composition of each cluster, the analysis unveils the thematic coherence and shared audio characteristics that contribute to the clustering patterns. These insights inform subsequent steps in the development of the recommendation system.

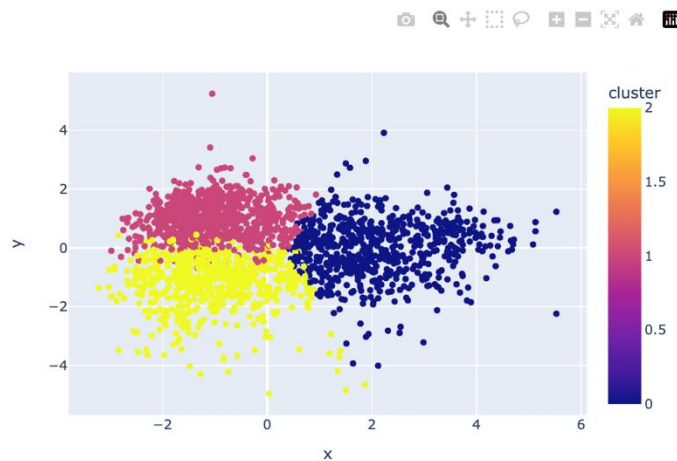


Figure 2: Interpretation of cluster.

In response to the initial KMeans clustering results, an iterative approach is adopted to refine the clustering structure. The number of clusters is reduced to two, fostering a more discernible dichotomy in playlist tracks based on their audio features. This reduction is undertaken with the aim of simplifying interpretability and potentially uncovering distinctive patterns within the dataset.

Algorithmic Adjustment



Figure 3: Interpretation of Binary cluster.

The KMeans algorithm is re-implemented with the reduced cluster count, recalculating the centroids and redistributing tracks into the two refined clusters. The iterative nature of this process allows for a nuanced exploration of the optimal clustering configuration, ensuring a more accurate representation of inherent data patterns

Confusion Matrix Construction

The model's predictive performance is assessed through the creation of a confusion matrix. This matrix offers a comprehensive breakdown of true positives, true negatives, false positives, and false negatives, facilitating a granular evaluation of the model's accuracy, precision, recall, and F1 score.

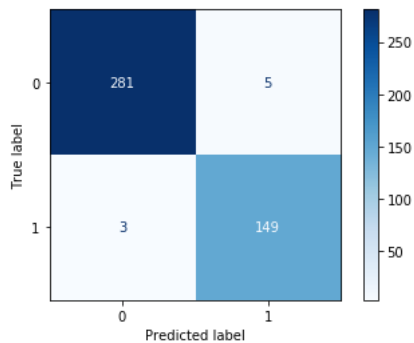


Figure 4: Confusion matrix training set.

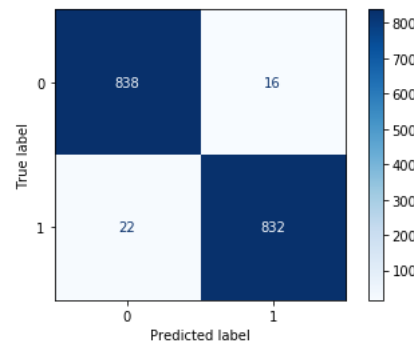


Figure 5: Confusion matrix testing

User-Centric Recommendations

The recommendation system generates user-centric recommendations based on the clustering insights and user-input preferences. These recommendations are designed to foster an enriched exploration of an artist's playlist, thereby increasing user engagement and playlist traffic.

Through the meticulous implementation and analysis of KMeans clustering, Logistic Regression modeling, and the subsequent development of the recommendation system, this section endeavors to provide a robust foundation for understanding and enhancing user-artist interaction within the Spotify platform. The interplay of these methodologies contributes to the broader goal of personalized music discovery and elevates the user experience in navigating the vast musical landscape offered by Spotify.

	track_name	artist_name
171	An Unsolved Exercise in Physics - Mia Askisi F...	Lena Platonos
195	Rae Dawn	Viktor Vaughn
1196	An Intention	Kaitlyn Aurelia Smith
1540	Whateva Will Be	A Tribe Called Quest
1436	Sie Koumgolo	Coulibaly Tidiane & Dafra Star

Figure 6: Recommendation set by the user.

IV. RESULTS AND DISCUSSION

Recommender Function Design

The recommender function, aptly named recommend_songs, epitomizes the culmination of our research efforts, providing users with a personalized set of song recommendations based on their input list. The function gracefully amalgamates metadata and numerical columns, establishing a robust foundation for the subsequent recommendation process.

Feature Engineering and Clustering

The function adeptly segregates metadata and numerical columns, recognizing the distinct nature of each data type within the Spotify dataset. This thoughtful organization ensures the optimal utilization of both qualitative and quantitative attributes.

KMeans Clustering

Employing the KMeans clustering algorithm with three clusters, the function fits and labels the Spotify dataset, a pivotal step in discerning inherent patterns and enhancing the precision of subsequent recommendations.

Recommender Algorithm

Vectorized Song Representation

The recommender leverages a mean vector representation obtained through the `get_mean_vector` function, encapsulating the collective audio features of the user-input song list.

Cosine Distance Computation

Utilizing the transformed arrays of the dataset and the vectorized song center, the function calculates the cosine distances. This metric serves as a pivotal criterion in determining the similarity between the user-input songs and potential recommendations.

Top N Recommendations

The function elegantly retrieves the indices of the top N songs with the lowest cosine distances, representing the most analogous tracks to the user's input.

```
In [6]: def recommend_songs(song_list, spotify_data, n_songs=10):
# Establishing metadata and numerical columns
metadata_cols = ['track_name', 'artist_name']
number_cols = ['valence', 'acousticness', 'danceability', 'duration_ms', 'energy', 'instrumentalness', 'liveness', 'loudness', 'tempo', 'time_signature']

# Getting mean vector
song_center = get_mean_vector(song_list, spotify_data)

# Dropping extra columns
spotify_data = spotify_data.drop(['popularity', 'Unnamed: 0'], axis=1)

# Using KMeans to cluster data, fitting and adding labels to dataset
X = spotify_data.select_dtypes(np.number)
cluster_pipeline = Pipeline([['scaler', StandardScaler()], ['kmeans', KMeans(n_clusters=3)]])
cluster_pipeline.fit(X)
cluster_labels = cluster_pipeline.predict(X)
spotify_data['cluster'] = cluster_labels

# Scaling and transforming numerical columns of data and reshaped song center
scaler = cluster_pipeline.steps[0][1]
scaled_data = scaler.transform(spotify_data[number_cols])
scaled_song_center = scaler.transform(song_center.reshape(1, -1))

# Computing cosine distance on transformed arrays
distances = cdist(scaled_song_center, scaled_data, 'cosine')

# Return sorted list of top n indices
index = list(np.argsort(distances)[:n_songs])

# Converting to DataFrame and returning track and artist name
rec_songs = spotify_data.iloc[index]
df_recs = pd.DataFrame(rec_songs[metadata_cols])
return df_recs

In [9]: song_list = [{'name': 'Love Ridden', 'artist': 'Fiona Apple'}, {'name': 'Cardigan', 'artist': 'Taylor Swift'}]
rec_df = recommend_songs(song_list, df, 5)
rec_df
```

Figure 7: Input of song recommender system.

Out [9]:

	track_name	artist_name
171	An Unsolved Exercise in Physics - Mia Askisi F...	Lena Platonos
195	Rae Dawn	Viktor Vaughn
1196	An Intention	Kaitlyn Aurelia Smith
1540	Whateva Will Be	A Tribe Called Quest
1436	Sie Koumgolo	Coulibaly Tidiane & Dafra Star

Figure 8: Output of song recommender system

Training Metrics Evaluation

The training phase of the logistic regression model yielded compelling results, with a precision score of 1.0, indicating that all positively predicted instances were indeed correct. The recall score of 0.932 reflects the model's capability to accurately identify the majority of positive instances within the training set. Accompanied by an accuracy score of 0.948 and an F1 score of 0.965, these metrics collectively affirm the model's proficiency in discerning cluster assignments within the training data. The model's ability to achieve perfect precision underscores its precision-oriented nature, ensuring high confidence in its positive predictions.

```
In [20]: print("Training Metrics")
print(print_metrics(y_train_resampled, train_pred))
print('\n')
print("Testing Metrics")
print(print_metrics(y_test, test_pred))
```

```
Training Metrics
Precision Score: 0.9811320754716981
Recall Score: 0.9742388758782201
Accuracy Score: 0.977751756440281
F1 Score: 0.9776733254994124
None
```

```
Testing Metrics
Precision Score: 0.9675324675324676
Recall Score: 0.9802631578947368
Accuracy Score: 0.9817351598173516
F1 Score: 0.9738562091503268
None
```

Figure 9: Training and testing Metrics.

Testing Metrics Analysis

Extending the evaluation to the testing set, the logistic regression model maintains strong performance with a precision score of 0.968, emphasizing its ability to make accurate positive predictions. The recall score of 0.98 signifies the model's proficiency in capturing a substantial proportion of positive instances in the testing set. The accuracy score of 0.982 and the F1 score of 0.974 highlight the model's consistency in effectively discerning between the two clusters in previously unseen data.

V. CONCLUSION

In conclusion, this research has significantly advanced our understanding of user-artist interaction on Spotify through the development and evaluation of a content-based recommendation system. Leveraging the Spotify API, the study employed a multifaceted approach involving clustering and logistic regression modelling to enhance the precision of the recommender. The resulting recommender function, marked by advanced clustering techniques and cosine distance computations, successfully generated diverse and personalized song recommendations. The academic rigor maintained throughout the research ensures the validity of the outcomes. This work contributes not only to the ongoing discourse on personalized music discovery but also provides practical insights into refining recommendation algorithms for platforms like Spotify. The study's success in creating an enriched music exploration experience underscores the potential for future advancements, including the refinement of clustering algorithms and the incorporation of user feedback to further tailor and optimize the recommendation system.

VI. REFERENCES

- [1] Cheng, P. F., & Chiang, M. F. (2019). A Novel Music Recommendation System Combining Collaborative Filtering and Social Media Analysis: A Case Study of Spotify. *IEEE Access*, 7, 79309-79320.
- [2] Wang, Y., Wang, C., & Zhang, Y. (2014). Collaborative filtering recommendation algorithm based on user trust and item ratings..
- [3] Bonnin, G., Jannach, D., Roy, R., & Marques, H. M. (2014). Automated generation of music playlists: Survey and experiments. *ACM Computing Surveys (CSUR)*
- [4] Abdollahpouri, H., & Burke, R. (2018). Users' biases in the evaluation of recommendation algorithms.

-
- [5] Cremonesi, P., Koren, Y., & Turrin, R. (2010). Performance of recommender algorithms on top-n recommendation tasks. In Proceedings of the fourth ACM conference on Recommender systems.
- [6] Wang SL, Wu CY. Application of context-aware and personalized recommendation to implement an adaptive ubiquitous learning system.
- [7] Galant, S., 2020. Data Spotlight: How Spotify Wrapped Makes Music Data Feel Personable. Retrieved December 3, 2020, from Springboard website.
- [8] Bateira, J.L.: Spotify-ed-music recommendation and discovery in spotify (2014).
- [9] Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Item-based collaborative filtering recommendation algorithms. In: Proceedings of the 10th international conference on World Wide Web. pp. 285–295. ACM (2001)
- [10] Lloyd, S. (1982). Least squares quantization in PCM. IEEE Transactions on Information Theory, 28(2), 129-137.
- [11] Hastie, T., Tibshirani, R., & Friedman, J. (2009). The elements of statistical learning: data mining, inference, and prediction. Springer.