**Enterprise Cloud Computing and Big Data (BUDT737)**

**Project Title:**  Spotify Playlist Recommendation

**Brief Description:** Build a Spotify recommendation system to predict what a user wants to hear next. Offer analytics and insights on trends of what people listen to and it's developed over time.

**Team Members :**

1. Weslee Hwang
2. Navneeth Oruganti
3. Krithika Somasekhar

ORIGINAL WORK STATEMENT

|  | Name | Signature |
| --- | --- | --- |
| 1 | Weslee Hwang | Weslee |
| 2 | Navneeth Oruganti | Navneeth |
| 3 | Krithika Somasekhar | Krithika |

We the undersigned certify that the actual composition of this proposal was done by us

and is original work.

**Introduction**

We are a software firm “””””””””””””””””””””””””””””

In the ever-evolving digital landscape, music streaming platforms like Spotify have revolutionized the way we discover and consume music. At the forefront of this transformation, Spotify has embraced the power of personalized recommendations driven by cutting-edge algorithms to elevate the user experience. Recognizing the profound impact of playlists on modern musical exploration, our project aims to develop an advanced recommendation engine for Spotify's playlist curation system. By harnessing the capabilities of the Alternating Least Squares (ALS) algorithm and the scalability of PySpark, we seek to create playlists that not only resonate with individual users' preferences but also foster the discovery of new artists and reshape the dynamics of the music industry. This endeavor represents a fusion of data science, software engineering, and a deep understanding of the intricate relationship between listeners and their music, promising to revolutionize the way we experience and engage with music on streaming platforms.

[Slide Content]

• The Rise of Music Streaming

  - Music consumption transformed by platforms like Spotify

  - Personalized recommendations driving user engagement

• Project Overview

  - Develop advanced playlist recommendation engine for Spotify

  - Leverage Alternating Least Squares (ALS) algorithm

  - Harness the power of PySpark for scalable data processing

• Key Objectives

  - Elevate user experience with personalized playlists

  - Promote discovery of new artists and musical genres

  - Reshape the dynamics of the music industry

• Fusion of Data Science, Software Engineering, and Music

  - Deep understanding of listener preferences and patterns

  - Cutting-edge algorithms and technologies

  - Revolutionizing the way we explore and engage with music

**Data Description**

Data source**:** [**https://www.aicrowd.com/challenges/spotify-million-playlist-dataset-challenge#dataset**](https://www.aicrowd.com/challenges/spotify-million-playlist-dataset-challenge#dataset)

**What the data is -**

The Playlist Dataset consists of 1,000 slice files. These files have the naming convention of:

mpd.slice.\_STARTING\\_PLAYLIST\\_ID\\_-\\_ENDING\\_PLAYLIST\\_ID\_.json

Each slice file is a JSON dictionary with two fields:

1. Info
2. Playlist

→ The info field is a dictionary that contains general information about the particular slice:

  slice - the range of slices that in in this particular file - such as 0-999

  version - the current version of the MPD (which should be v1)

  description- a description of the MPD

  license - licensing info for the MPD

  generated\_on - a timestamp indicating when the slice was generated.

→ Each playlist is a dictionary that contains the following fields:

pid- integer - playlist id - the MPD ID of this playlist. This is an integer between 0 and 999,999.

name - string - the name of the playlist

description - optional string - if present, the description given to the playlist.  Note that user-provided playlist

descrptions are a relatively new feature of Spotify, so most playlists do not have descriptions.

modified\_at - seconds - timestamp (in seconds since the epoch) when this playlist was last updated.

Times are rounded to midnight GMT of the date when the playlist was last updated.

num\_artists - the total number of unique artists for the tracks in the playlist.

num\_albums- the number of unique albums for the tracks in the playlist

num\_tracks - the number of tracks in the playlist

num\_followers - the number of followers this playlist had at the time the MPD was created.

num\_edits - the number of separate editing sessions. Tracks added in a two hour window are considered

to be added in a single editing session.

duration\_ms - the total duration of all the tracks in the playlist (in milliseconds)

collaborative-  boolean - if true, the playlist is a collaborative playlist. Multiple users may

 contribute tracks to a collaborative playlist.

tracks - an array of information about each track in the playlist. Each element in the array is a dictionary

 with the following fields:

track\_name - the name of the track

track\_uri - the Spotify URI of the track

album\_name - the name of the track's album

album\_uri - the Spotify URI of the album

artist\_name- the name of the track's primary artist

artist\_uri - the Spotify URI of track's primary artist

duration\_ms - the duration of the track in milliseconds

 pos - the position of the track in the playlist (zero-based)

**Sample Observations from the Data -**

"playlists": [

        {

            "name": "instrumentals",

            "collaborative": "false",

            "pid": 150000,

            "modified\_at": 1483142400,

            "num\_tracks": 15,

            "num\_albums": 11,

            "num\_followers": 1,

            "tracks": [

                {

                    "pos": 0,

                    "artist\_name": "Sergei Rachmaninoff",

                    "track\_uri": "spotify:track:5CBzaWVsJgl4DD1oNInz9b",

                    "artist\_uri": "spotify:artist:0Kekt6CKSo0m5mivKcoH51",

                    "track\_name": "Vocalise, Op.34, No.14",

                    "album\_uri": "spotify:album:6NrlEDfBRZZU0nYiixWYec",

                    "duration\_ms": 327600,

                    "album\_name": "Rachmaninoff: Symphonic Dances, Op.45; Intermezzo \"Aleko\"; Vocalise, Op.34"

                },

                {

                    "pos": 1,

                    "artist\_name": "Sergei Rachmaninoff",

                    "track\_uri": "spotify:track:4SnVAVSTV4iZxCPhVovfNb",

                    "artist\_uri": "spotify:artist:0Kekt6CKSo0m5mivKcoH51",

                    "track\_name": "Symphony No.1 in D minor, Op.13: 3. Larghetto",

                    "album\_uri": "spotify:album:2piZ6h6BZRAIAGp25nPJAS",

                    "duration\_ms": 541000,

                    "album\_name": "Rachmaninov: Symphony No.1; The Isle of Dead"

                },

                {

                    "pos": 2,

                    "artist\_name": "Sergei Rachmaninoff",

                    "track\_uri": "spotify:track:50YW5s4ZsaZrzN12E6CUAj",

                    "artist\_uri": "spotify:artist:0Kekt6CKSo0m5mivKcoH51",

                    "track\_name": "Piano Concerto No. 2 in C Minor, Op. 18: I. Moderato",

                    "album\_uri": "spotify:album:41eImZRW2jKn5tr0hkBFxj",

                    "duration\_ms": 561960,

                    "album\_name": "Rachmaninov: Piano Concertos Nos. 1-4 / Rhapsody On A Theme of Paganini"

                }

]

**Why is the data of Interest?**

The Business Case

Spotify, the giant in the realm of music streaming, recognized the enormous potential of personalized music recommendations. Not only do such systems deepen user engagement and retention, but they also act as a powerful tool for artist discovery, potentially shaping the very dynamics of the music industry. Seeking to refine their algorithms and push the boundaries of their user experience, Spotify reached out to our software firm. Our recognized expertise in big data analytics and machine learning positioned us perfectly to tackle this challenge.

The Technical Deep Dive

We understood that traditional collaborative filtering methods, while insightful, wouldn't fully capture the nuanced interplay of users and tracks in Spotify's vast ecosystem. Therefore, we proposed a solution centered around the Alternating Least Squares (ALS) algorithm. ALS offered several key advantages:

Scalability: Efficiently handling Spotify's massive user and song datasets was paramount. ALS's iterative matrix factorization approach gracefully scales to accommodate high-dimensional data.

Implicit Feedback: Unlike explicit ratings, ALS elegantly models implicit user feedback such as plays, skips, and playlist additions, providing a more holistic view of user preferences.

Flexibility: The algorithm could seamlessly integrate both user-song interaction data and metadata about the tracks themselves (e.g., genre, tempo, release year), promising even richer recommendations.

**The Development Blueprint**

**Data Engineering: Spotify's rich data required meticulous preprocessing. We established pipelines to transform raw logs, handle missing values, and carefully partition data into training and validation sets.**

**Model Building & Refinement: Hyperparameter tuning was critical. We experimented with parameters like the number of latent factors, regularization terms, and iterations to optimize the performance of our ALS model.**

**Evaluation: Beyond standard accuracy metrics, we designed custom evaluation frameworks. These would mirror Spotify's business objectives: user retention, new song adoption, and promoting long-tail artists.**

Delivering Value

Through tireless iterations and a deep understanding of Spotify's goals, our ALS-driven recommendation engine achieved results that surpassed expectations. Our playlists resonated with users, driving increased listening times and demonstrating the system's ability to adapt to ever-evolving preferences. Additionally, our models began to highlight lesser-known artists aligned with user tastes, opening the door to new musical journeys and potential shifts in artist popularity.

We delivered to Spotify not just a technical solution, but a powerful tool with the potential to transform the music experience for both listeners and creators.

**Research Questions**

1. Playlist Diversity Impact: How does the diversity of artists and genres within user-created playlists affect the predictive accuracy of our music recommendation algorithm? This question seeks to understand the complexity of recommendation systems in handling varied musical tastes.

2. Contextual Curation: Can incorporating contextual information from user-generated playlist titles enhance the personalization and relevance of our recommendations? This examines the potential of playlist titles to provide additional insights for more targeted song suggestions.

3. Temporal Dynamics: Are there discernible patterns in user playlist creation and music listening behavior that correlate with specific times of the day, days of the week, or seasons? This question aims to explore the influence of temporal factors on musical preferences and playlist curation.

4. Influence of Playlist Attributes: What role do factors such as a playlist's thematic cohesion, artist popularity, and a user's historical interactions play in the likelihood of a song being added to a particular playlist? This investigates the multifaceted criteria users might unconsciously apply when expanding their playlists.

5. Collaborative Filtering Efficacy: How does a collaborative filtering approach, leveraging both user-to-playlist and playlist-to-track interactions, compare to traditional user-item collaborative methods in terms of enhancing music recommendation quality? This question delves into the effectiveness of different collaborative filtering techniques in the context of music recommendations.

**Business Problems -**

Implementing a recommendation system, especially in the context of a music streaming service like Spotify, can address several business-oriented problems.

1.Improved User Engagement: By offering tailored playlist suggestions, users are more likely to stay on the platform and discover new music from other artists. This leads to an increase in user satisfaction and engagement.

2. Increased Content Discovery: By introducing users to songs and artists they might not have otherwise come across , recommendation engines can expand listeners' musical horizons. This improves the user experience and encourages a wider variety of artists to be featured on the platform.

3. Retention and Loyalty: Personalized interactions help users feel appreciated and understood, which has a big impact on loyalty and retention rates. A user is more likely to remain loyal to the platform if they regularly discover new and entertaining music through suggestions.

4. Ad Revenue Optimization: For platforms that rely on ad revenue, increased engagement means more opportunities to serve ads. Moreover, understanding user preferences can lead to more targeted and effective advertising, increasing ad value and revenue.

5. Market Competitiveness: Offering a superior music discovery and recommendation experience can set a platform apart from its competitors. In a crowded market, the ability to provide highly personalized and satisfying user experiences can be a significant differentiator.

6. Reducing Skip Rates: By accurately predicting and recommending songs that align with a user's taste, the system can reduce the frequency of song skips, leading to a smoother and more enjoyable listening experience.

7. Optimizing Playlist Curation Costs: Automated, data-driven recommendations can reduce the need for manual playlist curation, lowering operational costs associated with content management and curation.

8. User Feedback Loop: Recommendations can serve as a mechanism for collecting implicit user feedback (e.g., skips, saves, plays) to continuously refine and improve the recommendation algorithms, ensuring they remain effective over time.

By addressing these business challenges, a recommendation system not only improves the user experience but also drives key business metrics such as engagement, retention, and revenue.

**Methodology**

In our project we have used the following techniques to successfully complete the project-

We employed a hybrid recommendation system that combines collaborative filtering and content-based methods. The Apache Spark framework facilitated the processing of large-scale data, and the ALS (Alternating Least Squares) algorithm was pivotal in identifying latent factors and user-item interactions. Additionally, feature engineering and similarity measures were utilized to enhance the recommendation relevance and diversity.

**Exploratory Data Analysis (EDA):**

* Analyzing the distribution of playlist lengths, number of followers, and track popularity.
* Visualizing patterns in genre representation and co-occurrence, revealing potential biases or trends within the data.
* Investigating correlations between metadata (such as tempo, key, energy) and user preferences, offering insights for content-based recommendations.
* Data Preprocessing:

**Normalization and Scaling:**

* We normalized numerical features to ensure they were on a similar scale, preventing any one feature from dominating the models.
* Missing values in the dataset were addressed using techniques like imputation (e.g., mean or median replacement) or deletion when appropriate. This maintained the integrity of the data.

**Scalable Collaborative Filtering:**

* We employed the Alternating Least Squares (ALS) algorithm, implemented within the Apache Spark framework, to model complex user-item interactions.
* ALS effectively handles the large-scale data characteristic of Spotify, while its ability to model implicit feedback (plays, skips, playlist additions) provides a comprehensive understanding of user preferences.

**Content-Based Filtering:**

* To complement ALS, we utilized metadata-driven similarity measures (e.g., cosine similarity) to identify relationships between tracks based on their inherent characteristics (genre, tempo, release year).
* This content-based approach enhances recommendations, especially in situations of limited collaborative data.

**Feature Engineering:**

* We leveraged Apache Spark's capabilities to derive meaningful features from both user-interaction data and track metadata.
* This includes aggregate statistics (artist popularity, genre affinity, time-based preferences) to enrich our models.

**Hybrid Recommendation Strategy:**

* The insights from the ALS-based collaborative filtering model and content-based filtering are integrated to produce refined recommendations.
* We experimented with techniques such as weighted ensembling and cascading approaches to achieve an optimal balance.

Despite their vast collection of songs and artists, Spotify noticed a growing trend of listeners being stuck in their musical bubbles, rarely venturing beyond their usual playlists. This not only limited the listener's experience but also restricted the exposure of numerous talented artists.

The executives at Spotify laid out their vision in front of the Harmonica team: to create a dynamic and intuitive playlist recommendation system that could introduce listeners to new songs, artists, and genres, thereby enriching their musical journey and creating a more vibrant and interconnected music ecosystem.

They began by sampling playlists to understand patterns and preferences, meticulously analyzing the relationships between different songs, artists, and genres. Using advanced algorithms and machine learning models, they started crafting a system that could not only analyze a user's current music taste but also predict and recommend songs that the listener is likely to enjoy, even if they're from genres or artists they haven't explored before.

People were discovering songs they never knew they'd love, artists from corners of the world they hadn't heard of, and genres they'd never thought to explore. Spotify's library of music was no longer a vast, intimidating ocean, but a world of possibilities, accessible and personalized for every listener.