Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC Accredited with "A" Grade (CGPA : 3.18)
Department of Computer Science and Engineering (Data Science)

AY: 2023-24

Report on Mini Project

Big Data Engineering (DJ19DSL604)

AY: 2023-24

SONG RECOMMENDATION SYSTEM

Anuradha Yeole :60009210183

Foram Gandhi : 60009220174

Bhuvi Ghosh : 60009210191

Guided By

Adil Shaikh
(Assistant Professor)

CHAPTER 1: INTRODUCTION

In the flourishing field of big data analytics, the capacity to personalize content recommendations represents a significant advancement in enhancing user engagement and satisfaction. This report delves into a project aimed at developing a recommendation system for artists and songs tailored to individual user preferences. The project leverages a robust big data architecture to process and analyze extensive datasets, facilitating highly accurate and personalized music recommendations.

At the core of our solution is the integration of AWS Athena and Amazon S3. AWS Athena serves as an interactive query service that allows us to execute SQL queries directly against data stored in Amazon S3, providing fast, cost-effective, and scalable data analysis. The choice of Amazon S3 as our primary data storage solution offers durable, highly available, and secure storage, which is crucial for handling large volumes of data efficiently.

For data processing and analytics, PySpark, the Python API for Apache Spark, is utilized. PySpark offers extensive capabilities for big data processing, enabling us to handle complex data transformations and analysis with ease. Its ability to process data in-memory and support for multiple data sources and machine learning algorithms makes it an indispensable tool in our big data toolkit.

Finally, to visualize insights and results from our data, we incorporate Amazon QuickSight. QuickSight provides rich data visualization tools that allow us to create and publish interactive dashboards. These dashboards offer end-users and stakeholders intuitive and actionable insights into the performance of the recommendation system, enhancing decision-making processes.

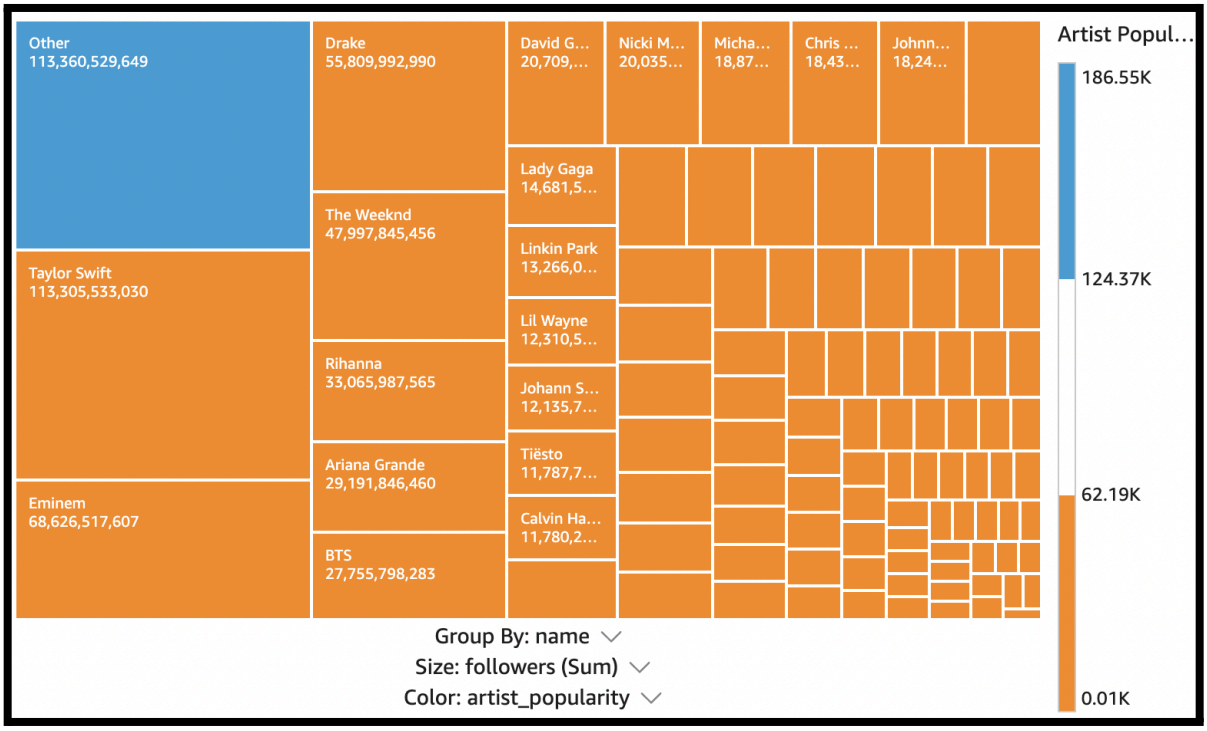
Together, these technologies form a comprehensive big data solution that not only meets the needs of our project but also sets a benchmark for efficiency, scalability, and customization in the realm of digital music services.

CHAPTER 2: DATA DESCRIPTION AND ANALYSIS

1. Data Description:

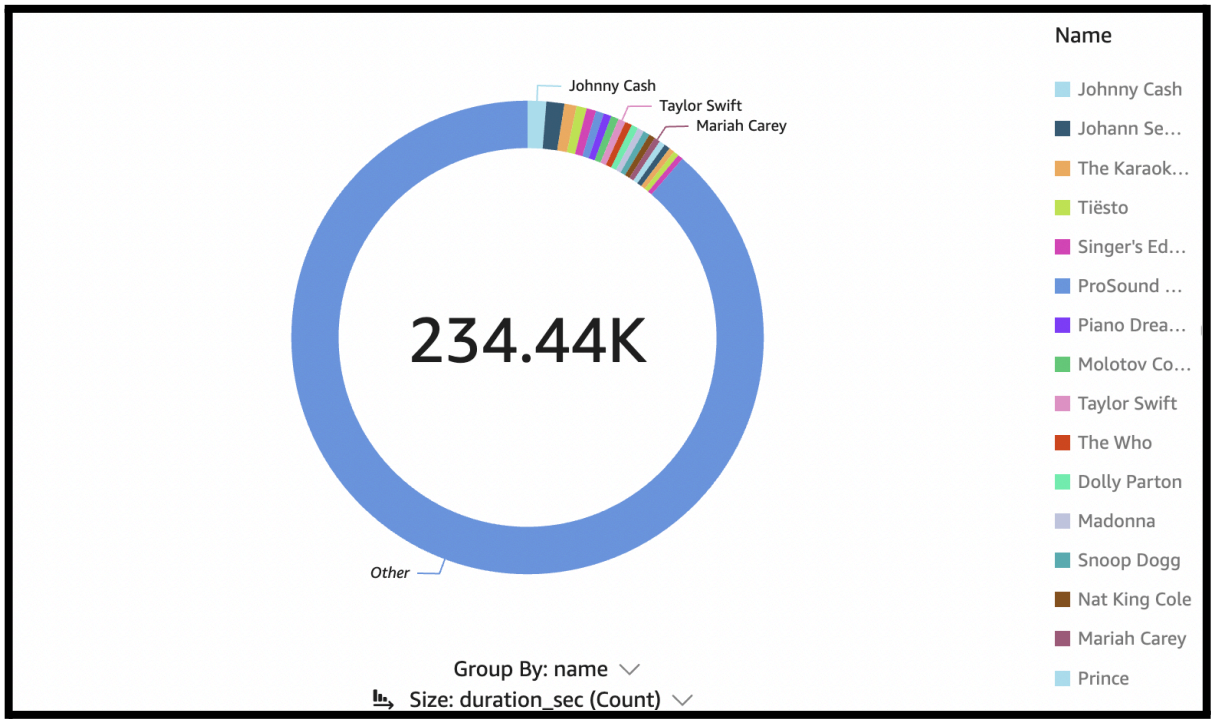
- followers: Number of followers for the track. (Integer)
- track_id: Unique identifier for the track. (String)
- artist_popularity: Popularity score for the artist. (Integer)
- artist_id: Unique identifier for the artist. (String)
- album_id: Unique identifier for the album. (String)
- duration_ms: Duration of the track in milliseconds. (Float)
- album_name: Name of the album. (String)
- name: Name of the artist. (String)
- duration_sec: Duration of the track in seconds. (Float)
- track_name: Name of the track. (String)
- track_popularity: Popularity score for the track. (Integer)
- label: Label of the track. (String)
- release_date: Release date of the track. (Date)
- album_popularity: Popularity score for the album. (Integer)
- genre: Genre of the track. (String)

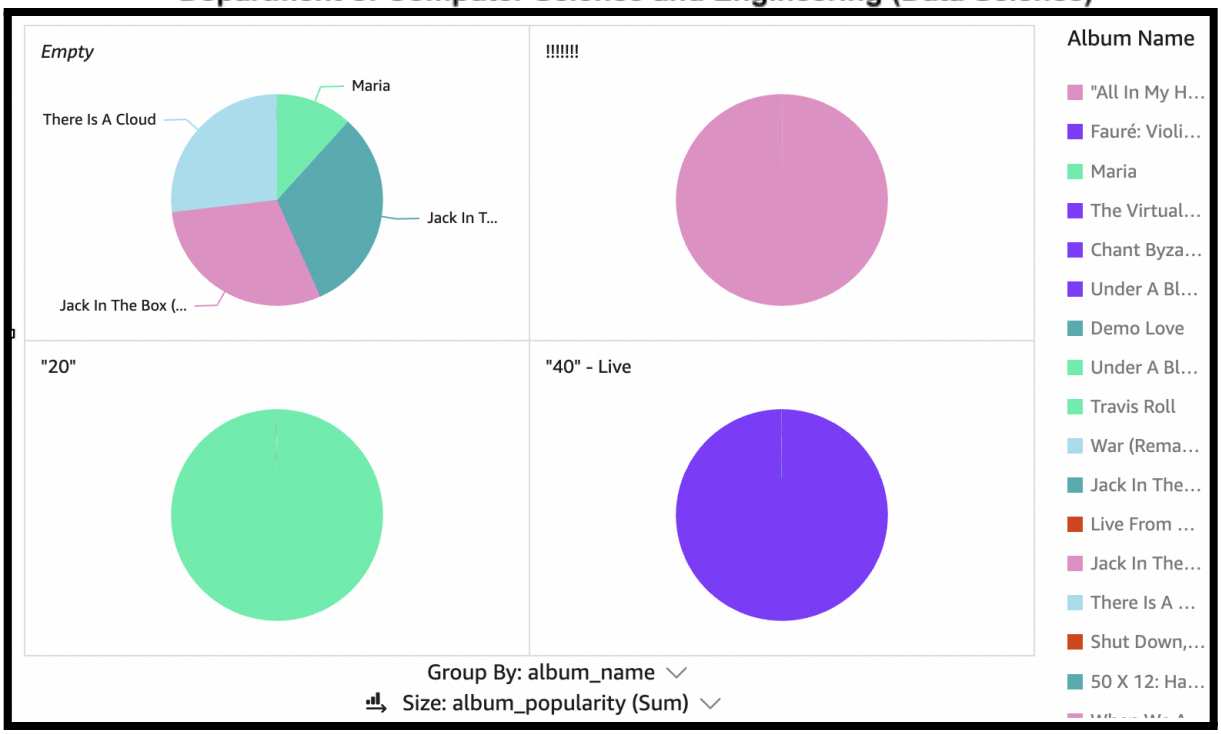
2. DATA ANALYSIS



The visualization provided is a treemap that represents the distribution of artist popularity and their follower counts within a given dataset. Each block in the treemap corresponds to an artist, with the size of the block indicating the sum of followers and the color shade representing the artist's popularity level. For instance, artists like Eminem and Taylor Swift, who occupy large blocks, have a significant number of followers, illustrating their high popularity. Eminem has approximately 68.6 billion followers, while Taylor Swift is close with about 113.3 billion. The color gradient, ranging from lighter to darker shades, helps in visually distinguishing between artists with varying levels of popularity, with darker shades

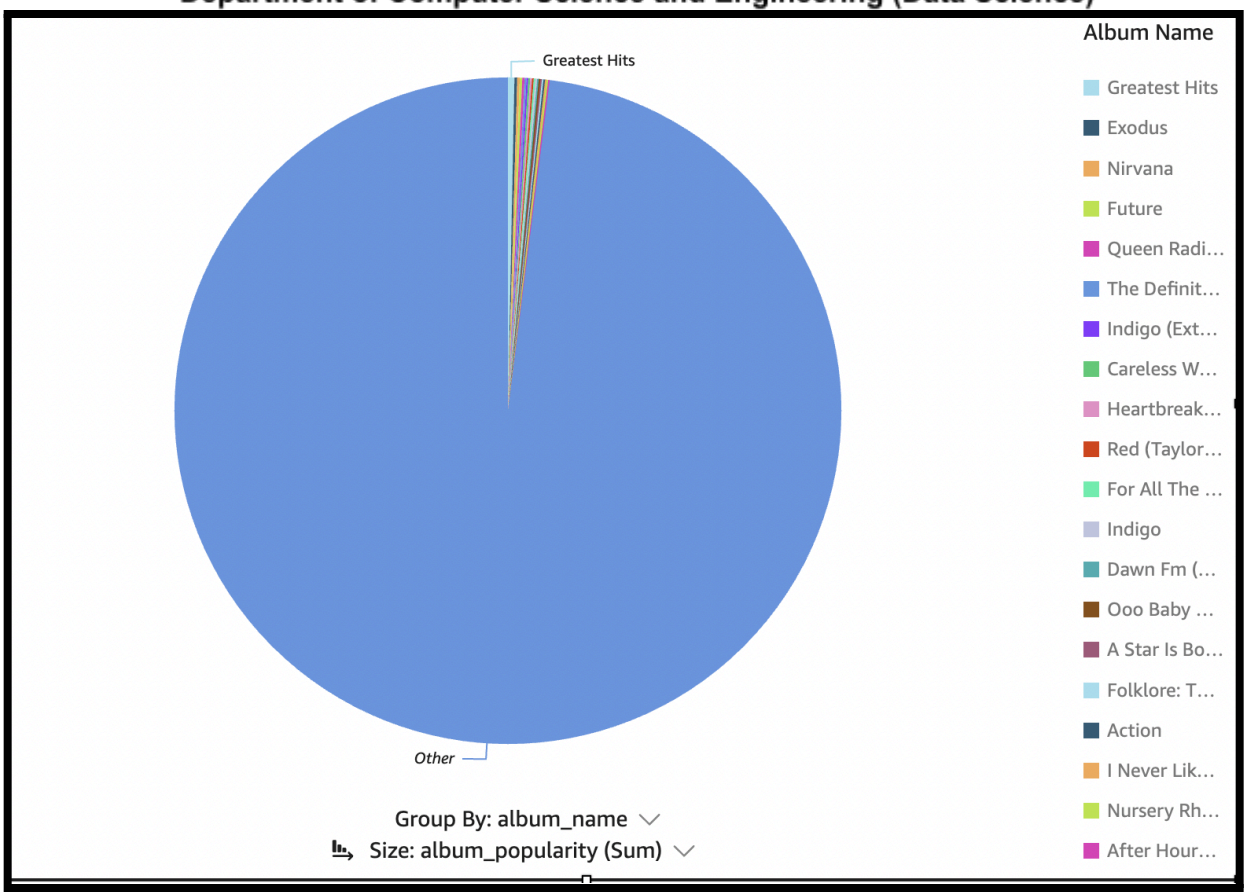
representing higher popularity. This treemap effectively utilizes space and color to convey complex data in a highly accessible format, making it easier to identify which artists dominate in terms of both followers and popularity.





The provided image showcases a series of pie charts representing the popularity distribution of different music albums across various categories, each labeled distinctly at the top.

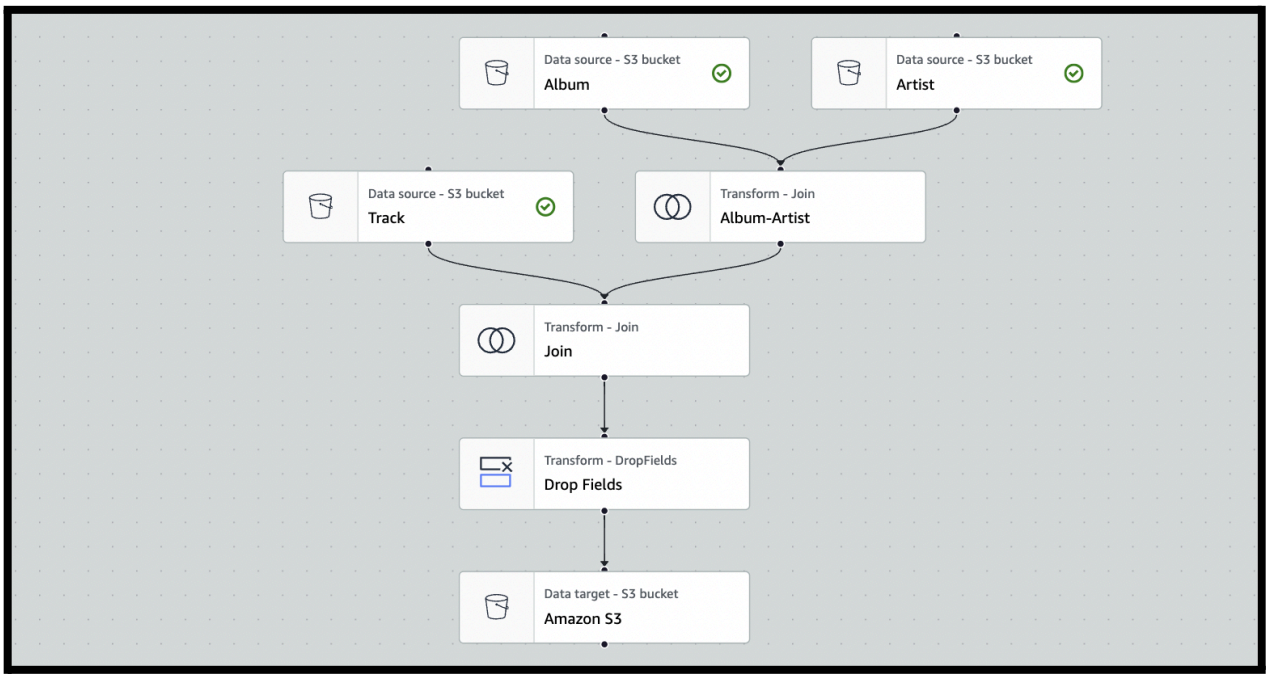
- The first chart, labeled "Empty," displays a diverse distribution with multiple albums like "There Is A Cloud," "Maria," and "Jack In The Box" among others, each having a share of popularity as indicated by the variously sized and colored segments.
- The second chart, labeled "!!!!!," features a single album represented as a fully pink pie chart, indicating that this album solely occupies the popularity within its category.
- In the third chart, labeled "20," a solid green pie chart indicates that a single album dominates its category entirely, similar to the second chart.
- The fourth chart, labeled "40" - Live," follows the pattern of the second and third, showing a purple pie chart representing the complete dominance of one album in its respective category.



The pie chart displayed visualizes the distribution of album popularity within music dataset. The vast majority of the chart is dominated by a single category labeled "Other," indicating that most albums in the dataset do not have a significant individual popularity. However, a sliver at the top represents the "Greatest Hits" album, suggesting that this particular album stands out in terms of popularity among all albums considered. This visualization effectively highlights the disparity in popularity, with "Greatest Hits" markedly more popular than the rest, which are grouped into a single, large segment.

CHAPTER 3: DESIGN OF DATA PIPELINE

1. ETL Pipeline:



Here are the steps involved in this ETL process:

1. Read Data from S3
    - The script reads three CSV files from an S3 bucket (s3://projectbhuvi/staging/):
        - albums.csv
        - artists.csv
        - track.csv
    - These files are read into Spark DataFrames using the glueContext.create_dynamic_frame.from_options method.

2. Join Data Frames
    - The Album and Artist DataFrames are joined using the Join.apply method, with artist_id being the join key.
    - The resulting joined DataFrame is called AlbumArtist_node1714672184066.
3. Join with Track Data
    - The Track DataFrame is joined with the AlbumArtist_node1714672184066
    - DataFrame using the Join.apply method, with track_id being the join key.
    - The resulting joined DataFrame is called Join_node1714672623389.
4. Drop Fields
    - The DropFields.apply method is used to drop the id and ``.track_id```` fields from the Join_node1714672623389 DataFrame.
    - The resulting DataFrame after dropping the fields is called DropFields_node1714673786195.
5. Write to S3
    - The final DropFields_node1714673786195 DataFrame is written to an S3 bucket (s3://projectbhuvi/datawarehouse1/) in Parquet format using the glueContext.write_dynamic_frame.from_options method.
    - The compression used for the Parquet files is Snappy.

CHAPTER 4: RESULT ANALYSIS

1.Data Processing and Model Building:

- Feature Engineering: The system utilizes features like artist popularity, followers, and genres from the dataset to form a comprehensive understanding of each artist. These features are critical in identifying patterns and preferences in user behavior.
- Normalization and Scaling: Attributes such as 'track_popularity' are scaled using MinMaxScaler to ensure that the model treats all features equally, improving the recommendation accuracy.

2. Recommendation Engine Performance:

- K-Nearest Neighbors Algorithm: The recommendation engine is powered by the K-Nearest Neighbors (KNN) algorithm, which is used to find similar songs based on the feature matrix that combines artist information and song popularity.
- Interactive Song Recommendations: The system allows users to input a song name, returning the top 5 recommended songs that are closest in feature space to the input song. This functionality demonstrates the system's responsiveness and user-friendly interface.

3. **Evaluation and User Interaction**:

- Sample Output: A test case with the song "Amazing Grace" produced recommendations like "Whispering Hope" and "Just As I Am", which suggests that the system effectively recognizes thematic and auditory similarities between tracks.

- Accuracy and Relevance: The recommendations provided are based on a blend of popularity and artist-related features, ensuring that the suggestions are not only popular but also relevant to the user's specific tastes.

4. Scalability and Robustness:

- Handling Large Datasets: The use of AWS services like Athena for querying and S3 for storage ensures that the system can handle large datasets efficiently, making it scalable and robust for commercial applications.
- Automation with AWS Glue: The automated data cataloging and ETL processes handled by AWS Glue contribute to a streamlined workflow, enabling quick updates and maintenance of the recommendation system.

CHAPTER 5: CONCLUSION AND FUTURE SCOPE

Conclusion : In leveraging AWS services such as S3, Glue, Athena, and PySpark, we have successfully engineered and prepared the music dataset for the development of a robust recommendation system. Through this process, several key achievements and insights have been obtained:

1. Data Integration and Transformation: AWS Glue provided a powerful platform for seamlessly integrating and transforming the raw music data stored in S3. By defining ETL jobs and leveraging Glue's serverless architecture, we efficiently processed and cleaned the data to ensure its readiness for analysis.
2. Automated Metadata Discovery: The AWS Glue Crawler enabled automated discovery and cataloging of metadata within the dataset. By intelligently scanning the S3 bucket and inferring schema definitions, the crawler facilitated the creation of a centralized metadata repository, enhancing data governance and accessibility.
3. Querying and Analysis: AWS Athena emerged as a valuable tool for querying and analyzing the prepared dataset using standard SQL queries. With its serverless query service, Athena provided on-demand access to data stored in S3, enabling ad-hoc exploration and insights generation without the need for complex infrastructure provisioning.

2. Future Scope

1. Personalization: Implement advanced algorithms, such as collaborative filtering and content-based filtering, to personalize recommendations based on user preferences, listening history, and behavior patterns.
2. Integration of External Data: Explore integrating external data sources such as social media activity, user demographics, and music reviews to enrich user profiles and improve recommendation accuracy.
3. Dynamic Updates: Implement mechanisms for real-time updates and feedback loops to continuously improve the recommendation system based on user interactions and feedback.