

Project Report On

Music Recommendation System



Submitted in partial fulfillment for the award of

Post Graduate Diploma in Big Data Analytics (PGDBDA) From KnowIT(Pune)

Guided by:

Trupti Joshi Ma’am & Prasad Deshmukh Sir

Submitted By:

Shubham Bane (230943025011) Ritik Varma (230943025040) Mohit Bhagwat (230943025030)



## CERTIFICATE

TO WHOMSOEVER IT MAY CONCERN

This is to certify that

Shubham Bane (230943025011) Ritik Varma (230943025040) Mohit Bhagwat (230943025030)

Have successfully completed their project on

Music Recommendation System

Under the guidance of Trupti Joshi Ma’am and Prasad Deshmukh sir



## ACKNOWLEDGEMENT

This project Music Recommendation System was a great learning experience for us and we are submitting this work to CDAC KnowIT (Pune).

We all are very glad to mention the name Trupti Joshi Ma’am and Prasad Deshmukh Sir for his valuable guidance to work on this project. His guidance and support helped us to overcome various obstacles and intricacies during the course of project work.

We are highly grateful to Mr. Vaibhav Inamdar Manager (KnowIT), CDAC, for his guidance and support whenever necessary while doing this course Post Graduate Diploma in Big Data

Analytics (PGDBDA) through CDAC ACTS, Pune.



TABLE OF CONTENTS

ABSTRACT

1. INTRODUCTION
2. DATA COLLECTION AND FEATURES
3. SYSTEM REQUIREMENTS
   1. Software Requirements
   2. Hardware Requirements
4. FUNCTIONAL REQUIREMENTS
5. ARCHITECTURE
6. MACHINE LEARNING ALGORITHMS
7. DATA VISUALIZATION AND REPRESENTATION
8. CONCLUSION AND FUTURE SCOPE
9. REFERENCES



### Abstract

We have music applications like Spotify that uses content-based and collaborative filtering to recommend us songs similar to what we like. Through the integration of Apache Kafka, Apache Spark, MongoDB, and machine learning techniques. The primary goal of a music recommendation system is to enhance user experience by offering personalized and relevant song suggestions, ultimately leading to increased user engagement and satisfaction. This project explores data generation, storage, analysis, and visualization, providing valuable insights into user behavior.



### INTRODUCTION

Big data analytics become an important trend for organizations and enterprises that are interesting in providing innovative ideas for enhancing and increasing their business performance and decision-making. Recommender systems are a group of techniques that allow filtering through large samples and information space in order to give suggestion to users when needed. Currently, they are becoming highly popular and utilized in different areas such as musics, research articles, search queries, news, books, social tags, and music. Furthermore, there are other essential recommender systems basically applicable for specialist, collaborators, funny story, restaurant and hotels, dresses, monetary services, life insurance, passion associates which give online dating services and several other social media such as Twitter, LinkedIn, and Facebook.

The main focus of this work is collaborative filtering system. It is well known that collaborative filtering could be described as a procedure whereby automatic prediction (i.e., filtering) about the interests of a user is made by gathering taste or preferences information from many users. A music recommendation system based on collaborative filtering using apache spark. The performance analysis and evaluation of proposed approach are performed on a Million song dataset.



# Dataset Collection and Features

#### Data Sources

For our project, the dataset chosen is potify song dataset. The Spotify Song Dataset is a freely-available collection of audio features and metadata for a Spotify popular music tracks. The dataset does not include any audio, only the derived features and MongoDB for storage. The decision to use simulated data was made to ensure data privacy and to have full control over the dataset's structure and content.

#### Data Structure

The dataset comprises several collections, each representing different aspects of the recommender system such as Song Name, Artist Name ,Popularity and more. These collections are stored within a MongoDB database, providing a flexible and scalable storage solution.

#### DataSet Description :

#### It is a Spotify Songs dataset and contains 15 columns.

#### Attributes are:

#### SongName

#### ArtistName

#### Popularity

#### Danceability

#### Energy Key

#### Loudness

#### Mode

#### Speechiness

#### Acousticness

#### Instrumentalness

#### Liveness

#### Valence Tempo

#### Duration\_ms



### 2. SYSTEM REQUIREMENTS

Hardware Requirements

1. Computer: A computer with sufficient processing power and memory to run data processing and analysis tasks. A modern multicore processor and at least 8 GB of RAM are recommended.
2. Storage: Adequate storage space to store the generated dataset and any additional datasets if required. An SSD (Solid State Drive) is recommended for faster data access.
3. Internet Connection: A stable internet connection for downloading and installing software packages and libraries, as well as for any online resources needed during the project.

Software Requirements

1. Operating System: Windows 10 or higher
2. Python: The project heavily relies on Python for data generation, analysis, and machine learning. Ensure Python is installed on your system.
3. Python Libraries: Install the following Python libraries and dependencies using package managers like pip or conda:

NumPy: For numerical computing.

pandas: For data manipulation and analysis. scikitlearn: For machine learning tasks.

Matplotlib and Seaborn: For data visualization.

PyMongo: For interacting with MongoDB. Other libraries specific to your project's needs.

1. MongoDB: Install MongoDB to store and manage the synthetic dataset. Ensure the MongoDB server is running
2. Kafka: If your project uses Kafka for realtime data streaming, install and configure Kafka on your system.
3. Integrated Development Environment (IDE): Choose a Pythonfriendly IDE, such as PyCharm, Jupyter Notebook, Visual Studio Code, or your preferred text editor.

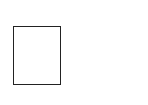


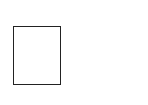
Visualization Software

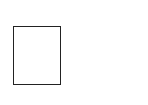
1. Tableau: If you plan to visualize and analyze data with Tableau, install Tableau Desktop.

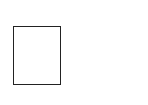
### 3. FUNCTIONAL REQUIREMENTS

1. Python 3:

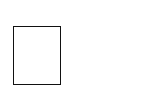
 Python is a general purpose and high level programming language.

 It is use for developing desktop GUI applications, websites and web applications.

 Python allows to focus on core functionality of the application by taking care of common programming tasks.

 Python is derived from many other languages, including ABC, Modula3, C, C++, Algol68, Small Talk, and Unix shell and other scripting languages.

(2) Apache Kafka:

What is Kafka: Apache Kafka is an opensource stream processing platform and distributed event streaming platform developed by the Apache Software Foundation.

Key Features: Kafka is designed to handle realtime data streams, making it a powerful tool for building and managing realtime data pipelines, eventdriven architectures, and applications that require highthroughput, fault tolerance, and scalability.

Overall, Apache Kafka serves as a backbone for realtime data processing, providing the infrastructure needed to handle large volumes of data, support eventdriven architectures, and enable applications with lowlatency requirements.

1. Tableau:

|  |  |
| --- | --- |
|  | * Data visualization is the graphical representation of information and data. |
|  | * It helps create interactive elements like charts, graphs, and maps, data visualization tools provide an accessible way to see and understand trends, outliers, and patterns in data. |
|  | * Tableau is widely used for Business Intelligence but is not limited to it. |
|  | * It helps create interactive graphs and charts in the form of dashboards and worksheets to gain business insights. |
|  | * All of this is made possible with gestures as simple as drag and drop. |

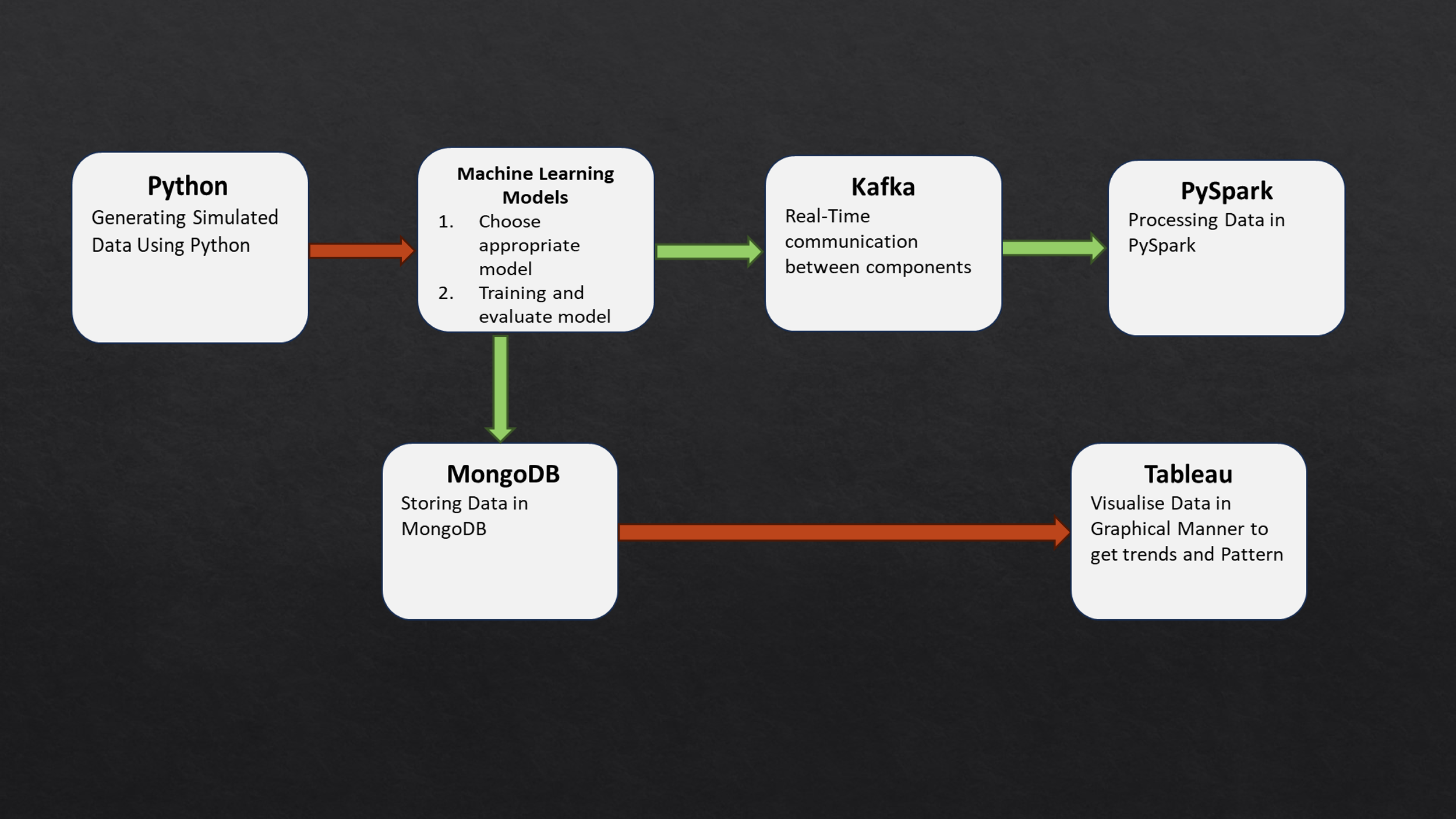


1. Streamlit UI Development:

* Streamlit is an open-source Python library for creating web applications and interactive data dashboards.
* Streamlit emphasizes simplicity, enabling developers to build applications with minimal code, making it ideal for rapid prototyping.
* It seamlessly integrates with popular data science libraries, allowing users to create interactive charts, graphs, and visualizations effortlessly.
* Streamlit provides easy-to-use widgets for user interaction, such as buttons, sliders, checkboxes, and more



# ARCHITECTURE





# MACHINE LEARNING ALGORITHMS

#### KMeans Clustering

Explanation:

KMeans Clustering is an unsupervised machine learning algorithm used for partitioning data into 'K' distinct, nonoverlapping clusters or groups. Each cluster represents a set of data points that are similar to each other based on selected features. Here's how KMeans Clustering works in our project:

#### Feature Selection:

We select relevant features from our song dataset, such as popularity, artist name, and song name. These features help define the similarity between songs.

#### Normalization:

Before applying KMeans, we normalize the data to ensure that features with different scales do not dominate the clustering process.

#### Benefits:

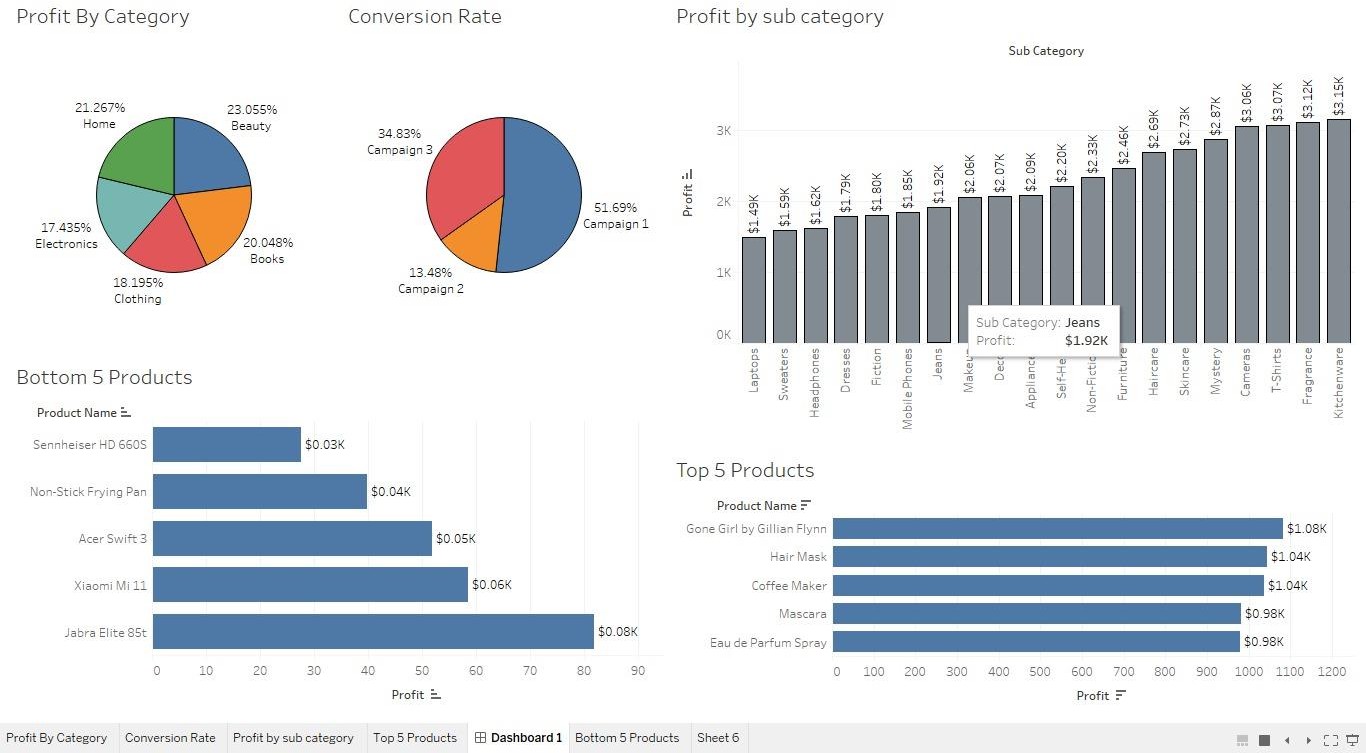
* Personalized Experience
* Improved Song Recommendations
* Enhanced User Experience

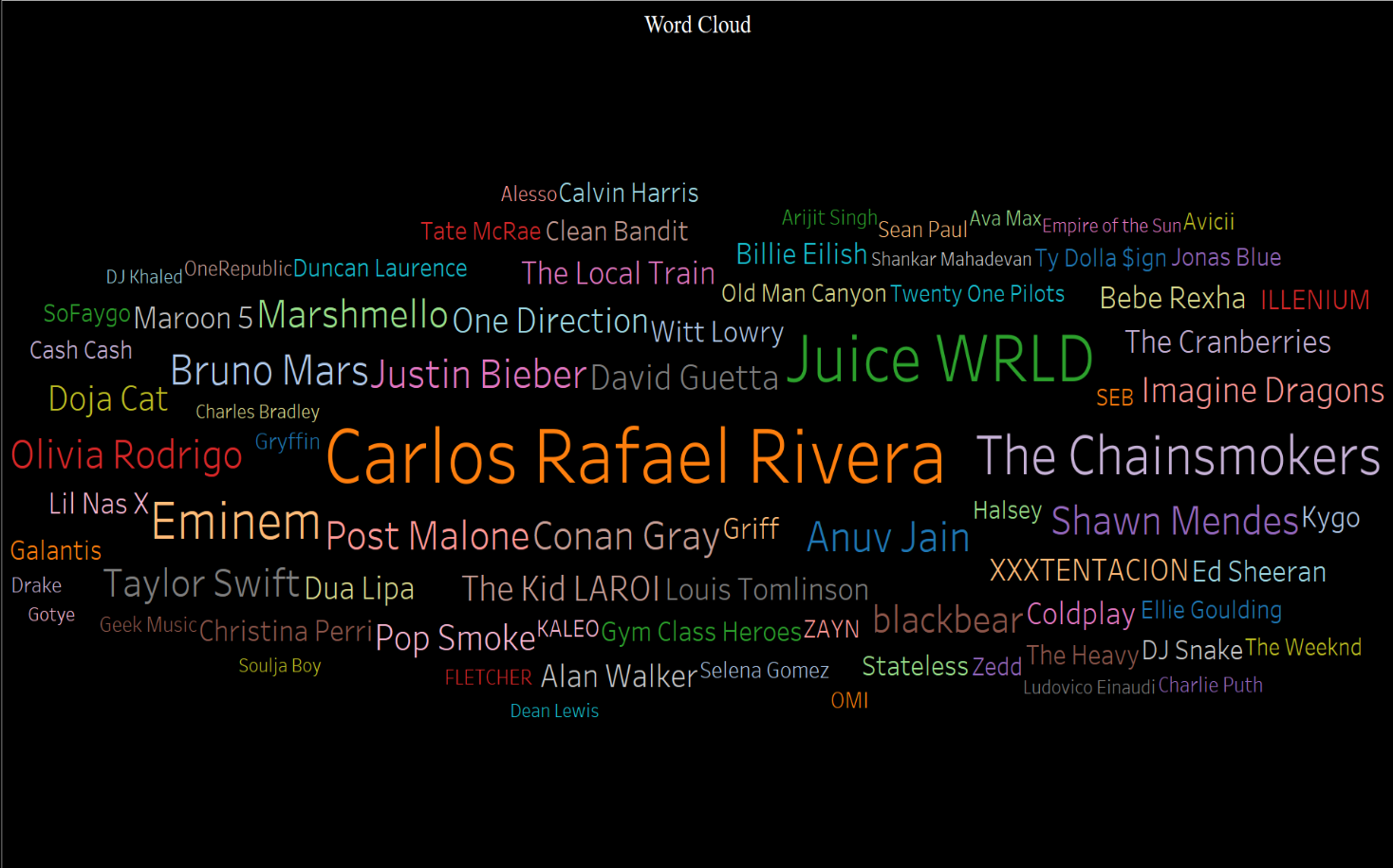
#### Conclusion:

K-Means Clustering is a powerful tool for song recommendation in our project. Apply K-means clustering algorithm using PySpark to group users or items based on similar features, enhancing recommendation accuracy.



# DATA VISUALIZATION AND REPRESENTATION



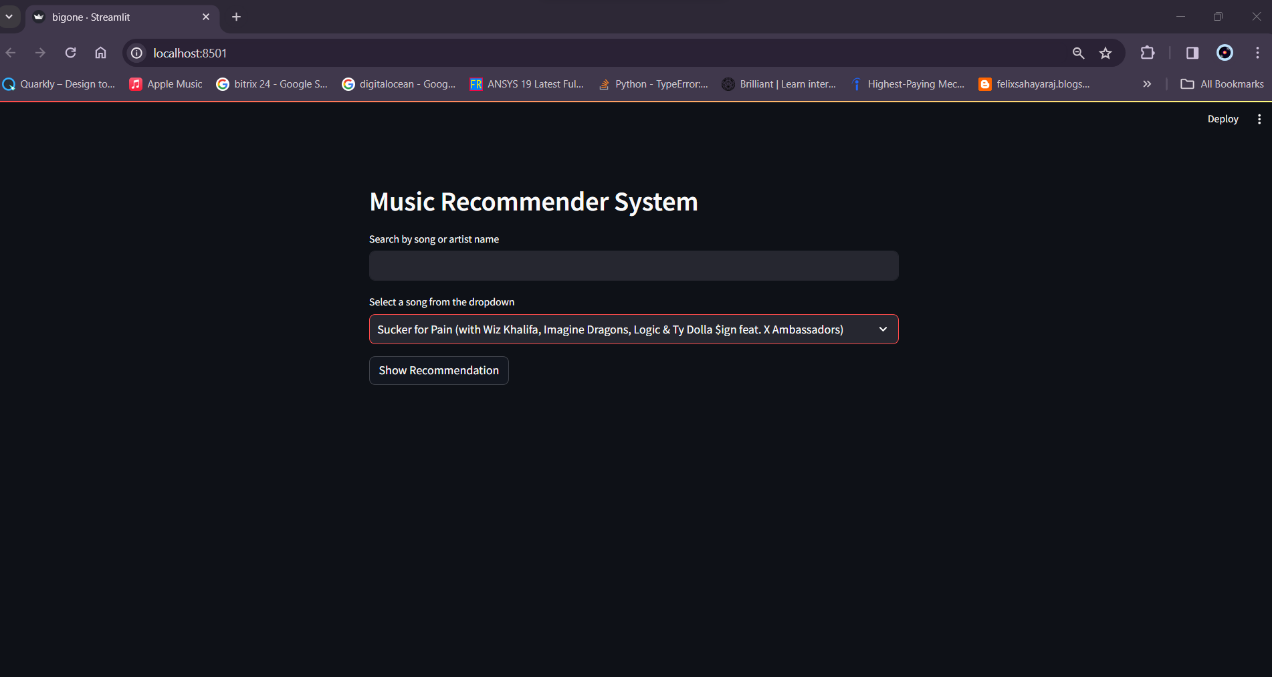


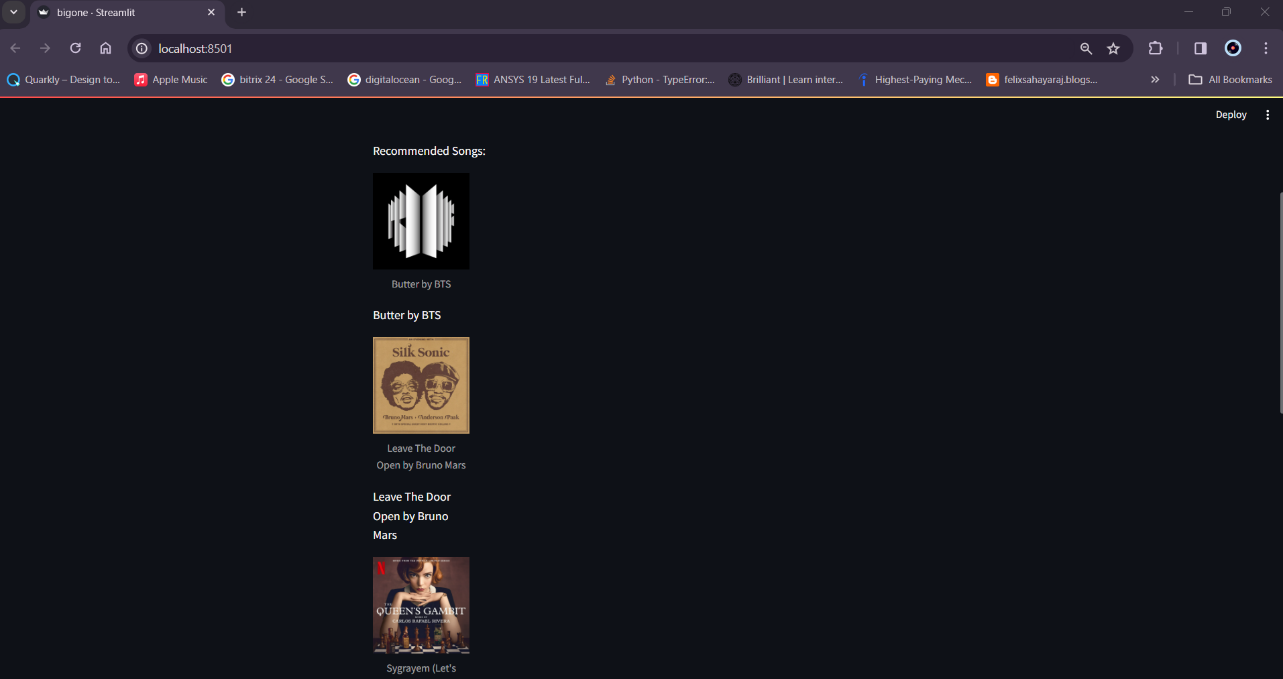


**CONCLUSION AND FUTURE SCOPE**

**Conclusion:**

In conclusion, music recommender system should consider the music genre information to increase the quality of music recommendations. By doing the analysis and recommendation using pyspark we conclude that each user is recommended with a unique playlist based on their interest. By focusing on key components such as k-means clustering as K-means clustering algorithm clusters the pieces in the music list dynamically adapting the number of clusters. We recommend pieces of music based on the clusters







### Future Scope:

1. Real-Time Dataset Usage:

Utilizing real-time datasets involves continuously updating and analyzing data as it becomes available. This approach allows recommendation systems to adapt quickly to changing user preferences and item availability. Real-time datasets enable the system to provide up-to-the-minute personalized recommendations, enhancing user experience and satisfaction.

1. Highly Personalized Recommendations at Scale:

Scalability is crucial for recommendation systems dealing with a large number of users and items. Advanced machine learning algorithms, such as collaborative filtering, content-based filtering, or hybrid models, are employed to efficiently process vast datasets and generate personalized recommendations for individual users, even in scenarios with millions of users and items.

1. Integration of Multi-Modal Data:

Multi-modal data involves incorporating diverse types of information, such as text, images, and audio, to enhance the recommendation system.Audio features, for example, can be analyzed to understand user preferences based on their listening habits.Analyzing album cover images can provide visual cues for item recommendations, considering aesthetic preferences or genre-specific visual patterns.

1. Incorporating Audio Features:

Extracting features from audio data can involve using techniques like audio fingerprinting, sentiment analysis of spoken content, or genre classification.These audio features can then be integrated into the recommendation algorithms, allowing the system to recommend music, podcasts, or other audio content that aligns with a user's preferences based on both historical interactions and audio feature analysis.

1. Analyzing Album Cover Images:

Image analysis of album covers can be employed to understand visual preferences of users.Computer vision techniques, including image recognition and deep learning, can be applied to extract relevant information from album cover images. Integrating this visual data into the recommendation system allows for a more comprehensive understanding of user tastes, especially in scenarios where visual aesthetics play a significant role in user choices.

In summary, the combination of real-time dataset usage, scalability, and the integration of multi-modal data, including audio features and album cover image analysis, enhances the capability of recommendation systems to provide highly personalized suggestions at scale, even in scenarios with a large and diverse user-item space



**References**

1. MongoDB. [https://[www.mongodb.com/]](http://www.mongodb.com/)
2. Kafka. [https://kafka.apache.org/]
3. Python. [[https://ww](http://www.python.org/)w.py[thon.](http://www.python.org/)org/]
4. scikit-learn. [https://scikit-learn.org/]