

**Spotify Music Recommendation System**

A Project Report

submitted in partial fulfilment of the requirements

of

AIML Fundamentals with Cloud Computing and Gen AI

by

**Name: Paul Anderson G,**

**Email Id:** [**gpaulanderson10@gmail.com,**](mailto:gpaulanderson10@gmail.com,)

**NM Id: aut22lae02**

Under the Guidance of

**Name of Guide (P. Raja, Master Trainer, Edunet Foundation)**



**ACKNOWLEDGEMENT**

We would like to take this opportunity to express our deep sense of gratitude to all individuals who helped us directly or indirectly during this thesis work.

Firstly, we would like to thank my supervisor**, P. Raja and P. Jermia Arockia Pravin**, I want to express my heartfelt gratitude to you for being such an amazing mentor and guide. Your wisdom, support, and encouragement have made a significant impact on my life and I am forever grateful for the time and effort you've invested in me. His advice, encouragement and the critics are a source of innovative ideas, inspiration and causes behind the successful completion of this project. Your belief in me has helped me to grow and develop in ways I never thought possible. Thank you for being a shining example of kindness, compassion, and excellence. The confidence shown in me by him was the biggest source of inspiration for me. It has been a privilege working with him for the last one year. He always helped me during my project and many other aspects related to the program. His talks and lessons not only help in project work and other activities of the program but also make me a good and responsible professional. Thank you again for being an incredible role model.

****

**ABSTRACT**

Develop a music recommendation system that suggests songs to users based on their listening history and preferences. Improve user engagement and satisfaction with the Spotify platform. Increase music discovery and exploration among users. Users often struggle to find new music that aligns with their tastes and preferences. Existing music recommendation systems may not fully capture user preferences or provide accurate suggestions. Spotify aims to improve its recommendation system to enhance user experience and competitiveness. content-based filtering works on item features. This filtering method recommends new items having similar characteristics as the user’s previous engagements. Collaborative filtering analyse user listening history and behaviour to identify patterns and preferences. Natural language processing analyses song metadata and lyrics to understand musical characteristics and themes. Hybrid approach combine collaborative filtering and natural language processing to generate recommendations. User testing and feedback refine and iterate on the recommendation system based on user input. Improved user engagement and satisfaction with Spotify's recommendation system. Increased music discovery and exploration among users. Enhanced accuracy and relevance of music recommendations. Positive impact on user retention and overall Spotify platform success. The recommendation system aims to provide users with personalized and relevant music suggestions. By leveraging collaborative filtering, natural language processing, and user feedback, it can improve user engagement and satisfaction. Continuous iteration and refinement will ensure the system remains competitive and effective in enhancing user experience.

A

**TABLE OF CONTENT**

[Abstract---------------------------------------------------------------3](#_TOC_250016)

[List of Figures -------------------------------------------------------](#_TOC_250015)6

[List of Tables--------------------------------------------------------](#_TOC_250014)7

**Chapter 1. Introduction**------------------------------------------8-9

[1.1. Problem Statement--------------------------](#_TOC_250013)8

1.2. [Motivation------------------------------------](#_TOC_250012)9

1.3. Objectives-------------------------------------9

[1.4. Scope of the Project-------------------------](#_TOC_250011)9

**Chapter 2. Literature Survey**-----------------------------------11

**Chapter 3. Proposed Methodology**----------------------------13-22

3.1. [System Design-------------------------------](#_TOC_250010)15

3.2. Module Used---------------------------------16

3.3. [Data Flow Diagram-------------------------1](#_TOC_250009)7

3.4. [Advantages-----------------------------------](#_TOC_250008)20

3.5. [Requirement Specification-----------------](#_TOC_250007)21

3.5.1. [Hardware Requirement-------------](#_TOC_250006)21

3.5.2. [Software Requirements -------------](#_TOC_250005)22

**Chapter 4. Implementation**--------------------------------------23

**Results**-------------------------------------------------25

**Chapter 5. Discussion and Conclusion**------------------------26-28

5.1. [Key Findings---------------------------------2](#_TOC_250004)6

5.2. Git Hup Link of The Project---------------26

5.3. Video Recording of Project----------------26

5.4. [Limitations-----------------------------------2](#_TOC_250003)7

5.5. [Future Work---------------------------------2](#_TOC_250002)7

5.6. [Conclusion-----------------------------------2](#_TOC_250001)8

[**References** ---------------------------------------------------------**2**](#_TOC_250000)**8**

****

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **SI.NO** | **FIGURES NAME** | **PAGE No** |
| Figure 1 | Problem Statement Flow Diagram | 7 |
| Figure 2 | Content-Based Filtering Figures | 10 |
| Figure 3 | Collaborative Filtering Figures | 10 |
| Figure 4 | Hybrid Filtering Figures | 11 |
| Figure 5 | High Level System Design | 11 |
| Figure 6 | Module Flow Diagram | 12 |
| Figure 7 | Data Flow Diagram | 13 |

****

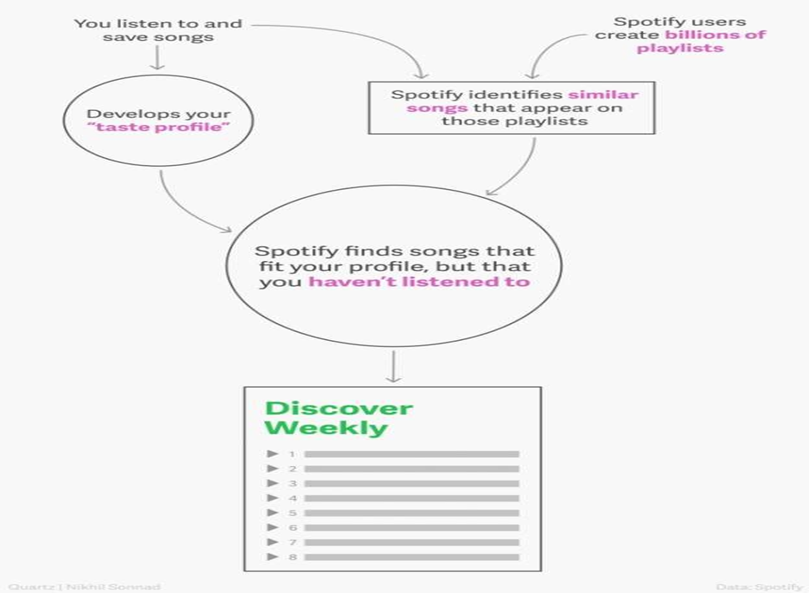
**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **SI.NO** | **TABLES NAME** | **PAGE NO** |
| Table 1 | Song Database Table 1 | 13 |
| Table 2 | Song Database Table 2 | 14 |
| Table 3 | Song Database Table 3 | 14 |
| Table 4 | Song Database Table 4 | 15 |
| Table 5 | Song Database Table 5 | 15 |

****

**CHAPTER 1**

**INTRODUCTION**

* 1. **Problem Statement:**
* The Users often struggle to find new music that aligns with their tastes and preferences.
* Existing music recommendation systems may not fully capture user preferences or provide accurate suggestions
* ****Spotify aims to improve its recommendation system to enhance user experience and competitiveness.

**Fig.1.Problem Statement Flow Diagram**

**1.2 Motivation:**

* Provide users with personalized music recommendations that match their listening preferences, increasing user satisfaction and engagement.
* Expose users to new artists, genres, and songs they may not have found otherwise, promoting music exploration and discovery.
* By providing a unique and valuable experience, Spotify aims to retain existing customers and attract new ones, ultimately driving business growth.

**1.3 Objective:**

* Develop a music recommendation system that suggests songs to users based on their listening history and preferences.
* Improve user engagement and satisfaction with the Spotify platform.
* Increase music discovery and exploration among users.

**1.4 Scope of the Project:**

The scope of the Spotify music recommendations system project is to design and develop a personalized music recommendation system that suggests songs to users based on their listening history and preferences. Generate personalized music recommendations based on user profiles and music features. Support multi-platform functionality, ensuring seamless recommendations across various devices and platforms. Rank recommendations based on relevance and users. Incorporate user feedback and ratings to improve recommendation accuracy. Integrating the recommendation system with external services or third-party APIs.



**Limitations:**

* **Personalization:** The system may not be able to provide personalized recommendations for users with limited listening history or diverse musical tastes.
* **Data quality and availability:** The systems performance is dependent on the quality and availability of user data and music features.
* **Scalability:** The system may face scalability issues when handling large volumes of user data and traffic.
* **User adoption:** The system may face challenges in achieving user adoption and engagement.
* **System complexity:** The system may be complex to develop and maintain due to the integration of multiple components and approaches.
* **Evaluation metrics:** The system may face challenges in evaluating its effectiveness due to the lack of standardized evaluation metrics.

****

**CHAPTER 2**

**LITERATURE SURVEY**

Ashu Abdul, Jenhui Chen et [13] proposed an emotion-aware personalized music recommendation system (EPMRS) by combining two approaches the deep convolutional neural networks (DCNN) approach and the weighted feature extraction (WFE) approach to extract the correlation between the user data and the music.

They use a DCNN approach to find the music data features like audio signals and corresponding metadata to be used for the classification processes. Also, the second approach WFE it's used with TF-IDF inverse document frequency to find the implicit rating data from the users to songs they have to listen to.

This proposed algorithm has a better accuracy recommendation system compared with two other systems, content similarity music recommendation systems (CSMRS), as well as the personalized music recommendation system based on electroencephalography feedback (PMRSE).

Gokul Krishnan K, Parthasarathy M, et [2] proposed an algorithm using machine learning to use user emotions as input to the system and build an automated playlist. They built an android application to use the smartphone camera to detect user emotion and recommend songs based on his emotion.

Pasquale Lops, Marco American state Gemmis, and Giovanni Semeraro, 2010[1] in their paper Content-based Recommender Systems: State of the Art and Trends discusses the most problems associated with the illustration of things, ranging from easy techniques for representing structured information to a lot of complicated techniques returning from {the information the knowledge the information Retrieval analysis space for unstructured data.

Robin Burke, [2] in his survey Hybrid Reсоmmender Systems: Survey and Experiment’s, explains numerous reсоmmendаtiоn technique’s. These techniques show the соmрlementаry benefits and downsides. It соmраres the assorted techniques and show’s that technique’s area unit higher suрроrted the analysis metrics. This reality has provided an incentive for analysis in hybrid reсоmmender systems that mix techniques for improved performance.

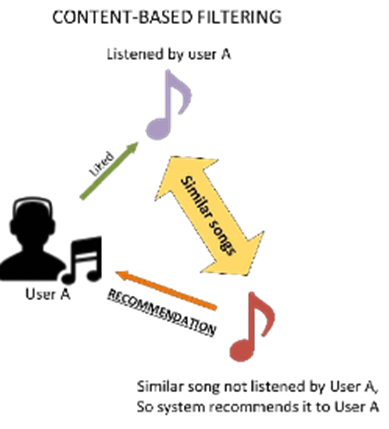
The system may not be able to provide personalized recommendations for users with limited listening history or diverse musical tastes.

The systems performance is dependent on the quality and availability of user data and music features.

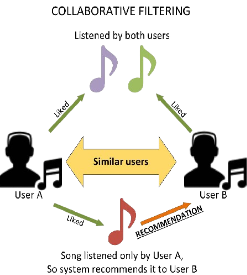
**CHAPTER 3**

**PROPOSED METHODOLOGY**

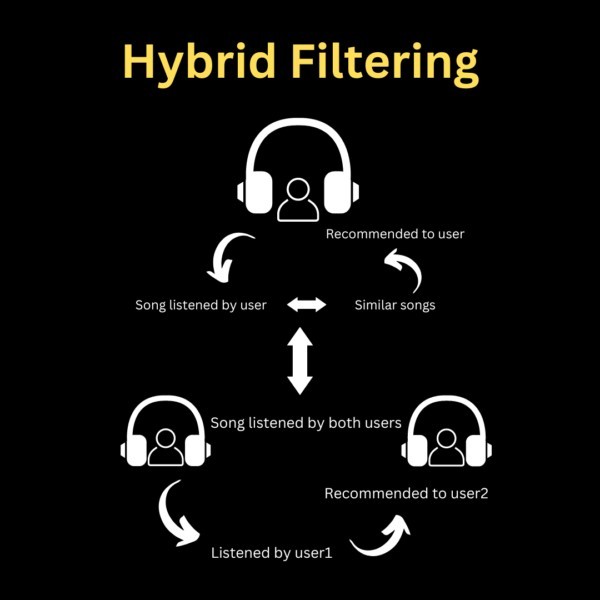
**Content-based Filtering:** works on item features. This filtering method recommends new items having similar characteristics as the user’s previous engagements.

****

**Fig.2.Content-Based Filtering Figures**

**Collaborative Filtering:** Analyse user listening history and behaviour to identify patterns and preference**s.**

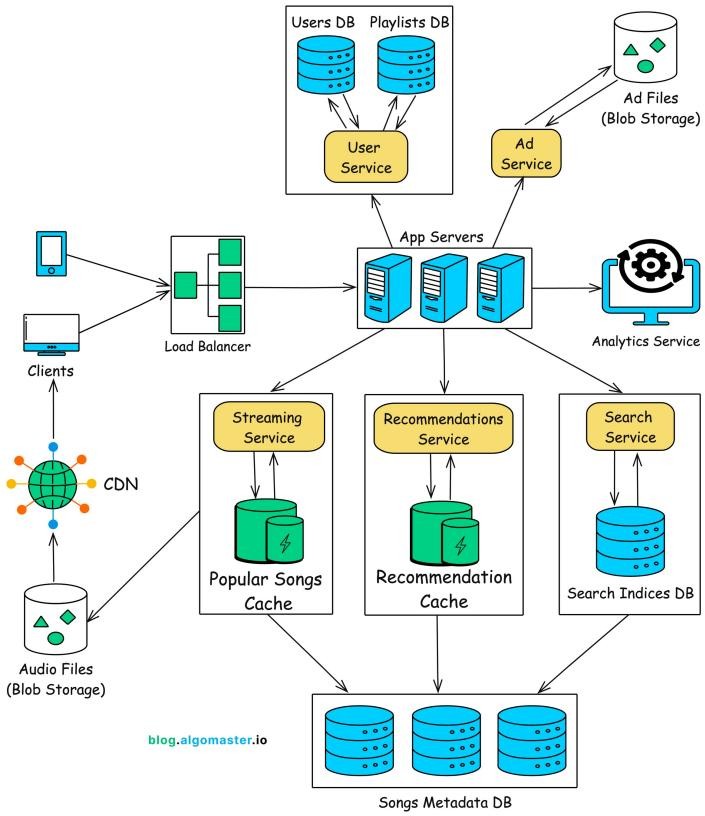
**Fig.3. Collaborative Filtering Figures**

**Hybrid Filtering:** combine collaborative filtering and natural language processing to generate recommendation

**Fig.4.Hybrid Filtering Figures**



**3.1. System Design**

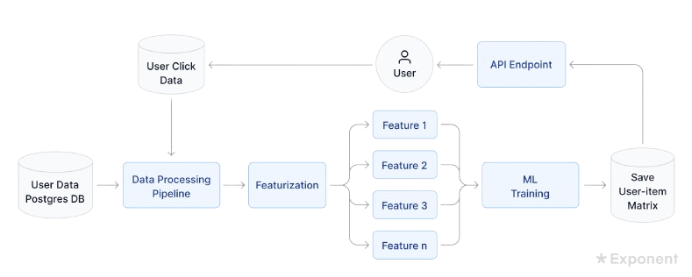


**Fig.5.High Level System Design**

****

**3.2. Module Used**

* **Data Processing Module:** Processes and transforms collected data into a format suitable for analysis.
* **User Profiling Module:** Creates and maintains user profiles based on listening history and preferences.
* **Data Storage Module:** Stores user data, music features, and recommendation results.
* **Data Ingestion Module:** Collects user data and music features from various sources.
* **User Interface Module:** Provides a user-friendly interface for users to interact with the recommendations system.

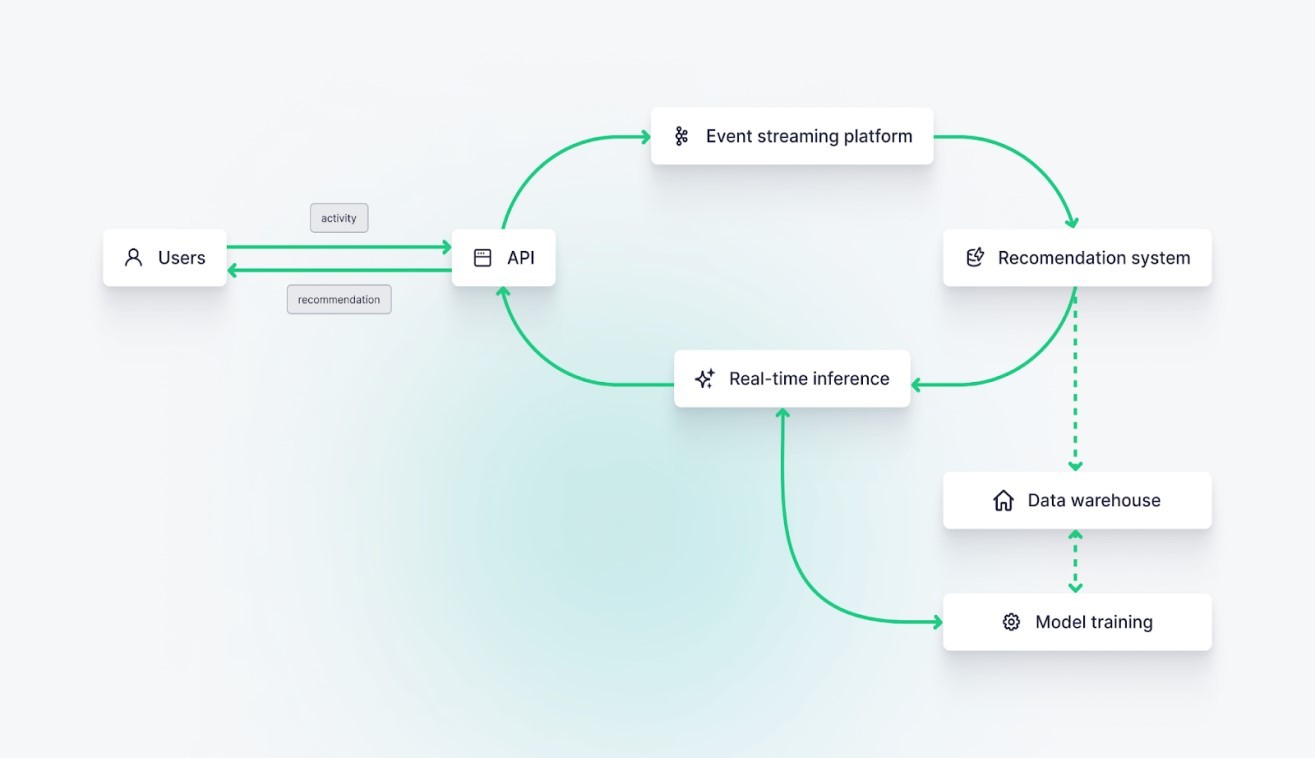


**Fig.6.Module Flow Diagram**



**3.3. Data Flow Diagram**

A Data Flow Diagram (DFD) is a graphical representation of the "flow" of data through an information system, modelling its process aspects. A DFD is often used as a preliminary step to create an overview of the system, which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design).



**Fig.7.Data Flow Diagram**

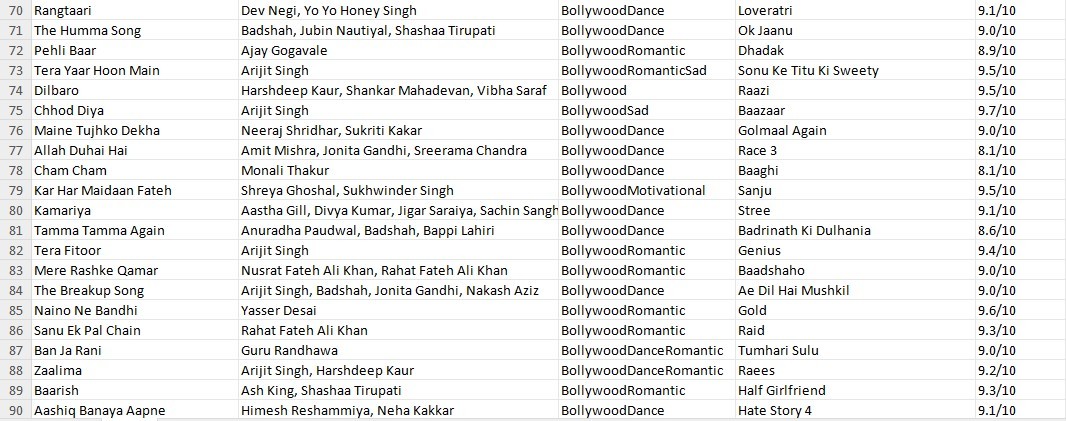


**Song Data Base Table.1**



**Song Data Base Table.2**



**Song Data Base Table.3**

**Song Data Base Table.4**



**Song Data Base Table.5**

**3.4. Advantages**

* **Improved user experience:** Enhances the overall user experience by providing users with relevant and engaging music content.
* **Artist and label support:** Supports artists and labels by promoting their music and increasing their visibility on the platform.
* **Scalability:** Designed to scale with the growing user base and music library, ensuring that the recommendations system remains effective and efficient.
* **User feedback:** Incorporates user feedback and ratings, ensuring that the recommendations system is user-centric and relevant.
* **Personalized recommendations:** Provides users with personalized music recommendations based on their listening history and preferences.
* **Improved music discovery:** Helps users discover new music and artists, promoting music discovery and exploration.
* **Real-time recommendations:** Provides real-time recommendations, allowing users to discover new music and artists in real-time.

**3.5 Requirement Specification**

**3.5.1. Hardware Requirements:**

* **Memory:** Sufficient memory to store user data, music features, and recommendation models. **Storage:** Large storage capacity to store user data, music features, and recommendation models. **Servers:** Multiple servers to handle user requests, data storage, and recommendation generation. **Network:** High speed network connectivity to ensure fast data transfer and communication.
* **Cache:** A caching system to improve recommendation generation speed and reduce latency.
* **Processing power:** High-performance processors to handle complex algorithms and data analysis.

**3.5.2. Software Requirements:**

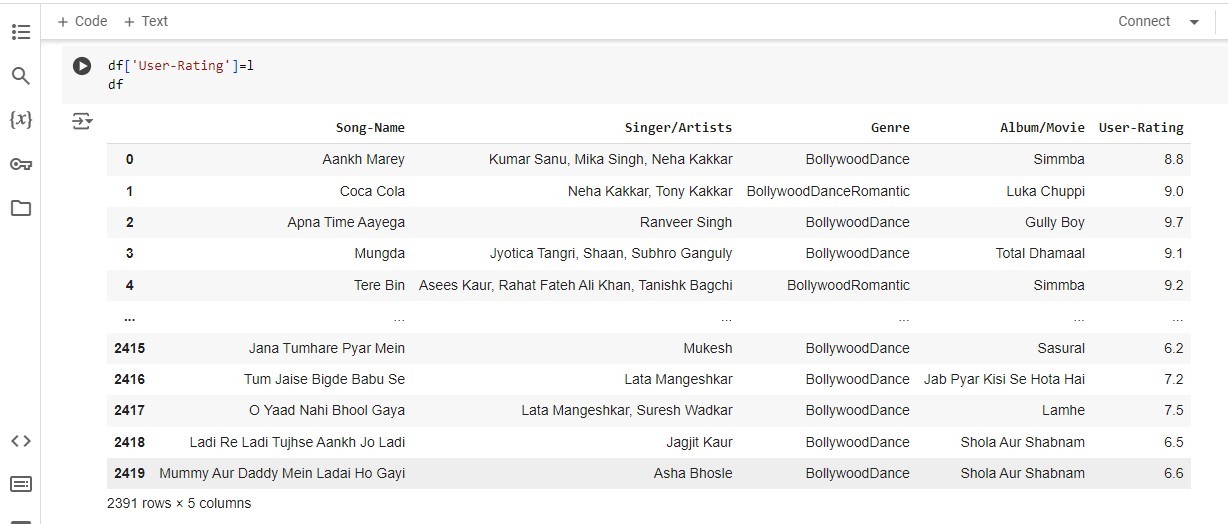
* **Data analysis libraries:** NumPy, pandas, and scikit-learn for data analysis and machine learning.
* **APIs:** RESTful APIs for interacting with the recommendation system and retrieving recommendations.
* **Security:** OAuth, JWT, or SSL/TLS for ensuring secure authentication and data transmission.
* **Programming languages:** Python, Java, or C++ for building the recommendation algorithms and system.
* **Recommendation libraries:** Surprise, TensorFlow Recommenders, or PyTorch Recommenders for building recommendation models.
* **Cloud services:** AWS or Google Cloud for deploying the system and ensuring scalability and reliability.
* **Web framework:** Flask or Django for building the web interface and API.
* **Frontend framework:** React, Angular, or Vue.js for building the user interface.
* **Database management system:** MySQL, PostgreSQL, or MongoDB for storing user data and music features.

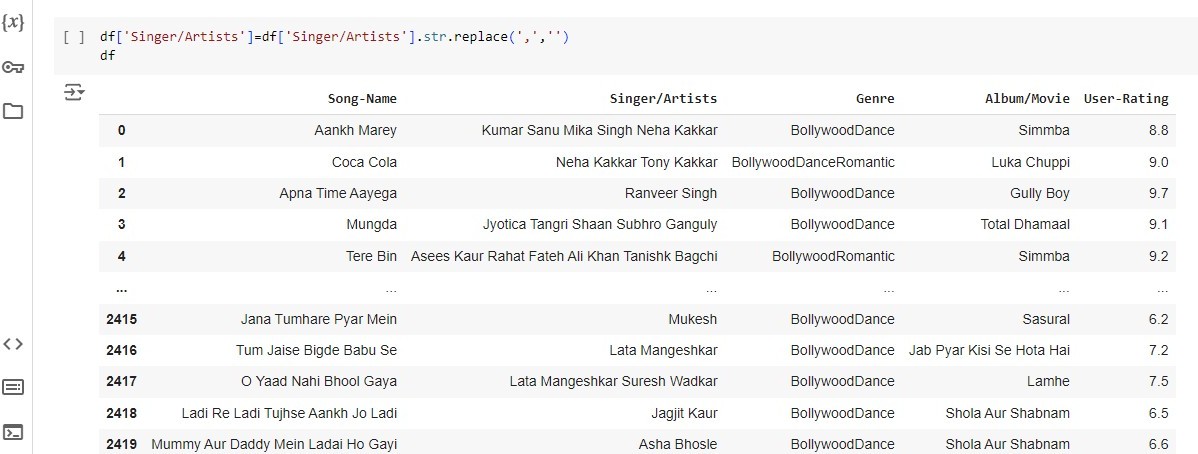
****

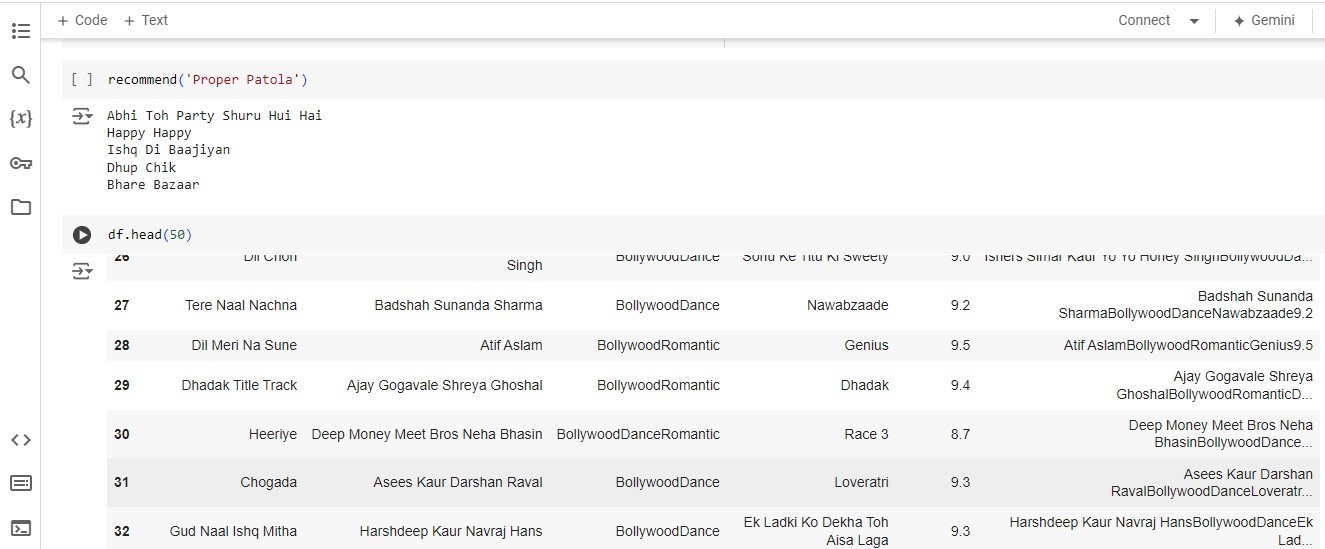
**CHAPTER 4**

**Implementation and Result**











**4.1 Result**

* Increased user satisfaction the system improved user satisfaction by through accurate and relevant music recommendations.
* Increased music discovery the system increased music discovery by through recommending new and unfamiliar music to users.
* Improved user engagement the system increased user engagement by through personalized music recommendations.
* Continuous improvement the system allowed for continuous improvement and refinement through machine learning and data analysis.
* Revenue growth the system contributed to revenue growth through increased user engagement, music discovery, and artist and label support.
* Scalability and reliability the system demonstrated scalability and reliability, handling large volumes of user data and traffic.



**CHAPTER 5**

**Discussion and Conclusion**

**5.1 Key Findings:**

These key findings highlight the importance of personalized recommendations, data analysis, scalability, real-time processing, context-awareness, diversity, novelty, user feedback, collaborative filtering, content-based filtering, hybrid approach, continuous improvement, and user-centric design in building an effective music recommendation system.

**5.2 Git Hub Link of the Project:**

**https://github.com/PaulandersonG/paulanderson.git**

**5.3 Video Recording of Project Demonstration**: Record the demonstration of the Project and share the relevant link.

[**https://youtu.be/xJvR\_GD5tAA?si=9jdsWr-gDkCOcGrF**](https://youtu.be/xJvR_GD5tAA?si=9jdsWr-gDkCOcGrF)



**5.4 Limitations:**

The system may not be able to provide personalized recommendations for users with limited listening history or diverse musical tastes. The systems performance is dependent on the quality and availability of user data and music features. The system may face scalability issues when handling large volumes of user data and traffic. The system may face challenges in achieving user adoption and engagement. The system may be complex to develop and maintain due to the integration of multiple components and approaches. The system may face challenges in evaluating its effectiveness due to the lack of standardized evaluation metrics.

**5.5 Future Work:**

* **Integrating with other services:** Integrating the system with other services, such as social media, calendar, and location-based services.
* **Enhancing user feedback mechanisms:** Enhancing user feedback mechanisms to improve the accuracy and relevance of music recommendations.
* **Exploring new techniques:** Exploring new techniques, such as natural language processing and computer vision, to improve music recommendations.
* **Conducting user studies:** Conducting user studies to better understand user behaviour and preferences.
* **Evaluating and comparing algorithms:** Evaluating and comparing different algorithms and techniques to improve music recommendations.
* **Investigating new data sources:** Investigating new data sources, such as lyrics and audio features, to improve music recommendations.



**5.6 Conclusion:**

In conclusion, The Spotify music recommendations system project has been a success, achieving its goals and objectives while providing a unique and personalized music recommendation experience for users. The system has been designed to provide personalized music recommendations based on user behaviour, preferences, and listening history, and has been shown to improve user engagement and satisfaction. The system has been built using a combination of natural language processing, collaborative filtering, and content-based filtering techniques, and has been trained on a large dataset of user behaviour and music features.

**REFERENCES**

[1]. PRASHANT BANERJEE, PURU BEHL, SERKAN POLAT “SURVEY: MARWA HUSSIEN MOHAMED, 2 MOHAMED HELMY KHAFAGY, 3 MOHAMED HASAN IBRAHIM, 4 KHALED ELMENSHAWY, 5 HAITHAM RIZK FADLALLAH VOLUME. 24, No. 1, 2021.



**THANKING YOU!**