Artificial Intelligence and Machine Learning

Project Abstract Semester-IV

(Batch-2022)

Music Recommendation System

Supervised By:

Mrs. Shagun Sharma

Submitted By:

Name:

1.Kanwardeep Singh (2210990473)

2. Kanwarjeet Singh(2210990474)

3. Jaskirat Singh Maan (2210990445)

Group:G10

Department of Computer Science and Engineering
Chitkara University Institute of Engineering & Technology,
Chitkara University, Punjab

Title:- Music Recommendation System machine learning project This

project is based on Supervised learning.

Abstract:-

In today's digital age, the abundance of music content available online presents both opportunities and challenges for users seeking personalised recommendations. Music recommendation systems play a crucial role in assisting users in discovering new music that aligns with their tastes and preferences. However, existing systems often face limitations in accurately capturing the diverse and nuanced nature of individual music preferences.

This project aims to address these challenges by proposing a comprehensive framework for enhancing music recommendation systems. The framework encompasses various components, including user profiling, item modelling, and item-user profile matching, with a focus on leveraging state-of-the-art recommendation techniques.

Drawing from recent advancements in psychology, signal processing, machine learning, and musicology, our framework integrates both traditional collaborative filtering algorithms and content-based models. Additionally, we explore innovative approaches such as emotionbased and context-based recommendation models to better capture users' subjective music preferences.

Through a thorough survey of existing literature and empirical analysis, we evaluate the effectiveness of different recommendation techniques and identify areas for improvement. Our research culminates in the proposal of a novel recommendation model based on users' motivations, aiming to provide a more intuitive and personalised music recommendation experience.

By advancing the state of the art in music recommendation systems, this project contributes to the broader goal of enhancing user satisfaction and engagement in accessing and discovering music content online.

Supervised learning:-

Supervised learning is a type of machine learning algorithm that learns from labeled data. Labeled data is data that has been tagged with a correct answer or classification.

Unsupervised learning:-

Unsupervised learning, also known as unsupervised machine learning, uses machine learning (ML) algorithms to analyze and cluster unlabeled data sets. These algorithms discover hidden patterns or data groupings without the need for human intervention

Table 1 Difference between supervised and Unsupervised learning

Supervised Learning	Unsupervised Learning
Supervised learning algorithms are trained using labeled data.	Unsupervised learning algorithms are trained using unlabeled data.
Supervised learning model takes direct feedback to check if it is predicting correct output or not.	Unsupervised learning model does not take any feedback.
Supervised learning model predicts the output.	Unsupervised learning model finds the hidden patterns in data.
In supervised learning, input data is provided to the model along with the output.	In unsupervised learning, only input data is provided to the model.
The goal of supervised learning is to train the model so that it can predict the output when it is given new data.	The goal of unsupervised learning is to find the hidden patterns and useful insights from the unknown dataset.
Supervised learning needs supervision to train the model.	Unsupervised learning does not need any supervision to train the model.

Table of content

Introduction:-

In the ever-evolving landscape of digital content consumption, recommendation systems silently but significantly shape our daily interactions with online platforms. From suggesting videos to watch, music to stream, or products to purchase, these algorithms wield a profound influence, curating personalized experiences that align with our unique tastes and preferences. Despite their often understated role, recommendation systems are pivotal in enhancing user engagement and satisfaction, demanding continual innovation and thoughtful design to meet the evolving needs of modern consumers.

Historically, recommendation systems have relied on two primary approaches: content-based filtering and collaborative filtering. Content-based filtering analyzes item properties to make recommendations, while collaborative filtering identifies similarities between users to suggest relevant content. Furthermore, the real-time operation of these systems introduces additional complexities, requiring efficient data processing methods to handle dynamic user interactions seamlessly.

In response to these challenges, our pioneering algorithm transcends conventional boundaries by seamlessly blending content-based and collaborative filtering techniques. This novel hybrid approach capitalizes on the strengths of both methodologies, providing more accurate and personalized recommendations. At its core, our system evaluates each song in its database based on user preferences, leveraging insights gleaned from historical data and user interactions. By considering key metadata variables such as genre, mood, and tempo, our algorithm develops a nuanced understanding of user preferences, enabling swift and responsive recommendations.

Driving the mechanics of our system is a sophisticated deep learning classification model, meticulously trained to evaluate the likelihood of user enjoyment for each song based on input variables. This approach mirrors the success of industry-leading features such as Spotify's Discover Weekly Playlist, offering users curated selections tailored to their evolving tastes and listening habits. Moreover, our system extends its capabilities beyond individual song recommendations, seamlessly transitioning into the realm of playlist continuation—a task underscored by the ongoing demand for innovative solutions, as evidenced by Spotify's RecSys Challenge 2018.

As the digital landscape continues its rapid evolution, our system stands as a testament to innovation and progress in the realm of music recommendation. With an unwavering commitment to excellence, our algorithm heralds a new era of personalized music discovery, empowering users to embark on enriching musical journeys tailored precisely to their preferences. In doing so, we strive to enhance user experiences and foster deeper connections between individuals and the vast world of digital content.

Methodology:-

Our methodology for developing a music recommendation system encompasses several key steps aimed at creating a robust and personalized user experience.

Firstly, we begin with data collection and preprocessing. We gather music data from diverse sources such as online platforms, databases, and APIs. This raw data undergoes rigorous preprocessing to ensure consistency and reliability. We handle missing values, standardize data formats, and encode categorical variables. Additionally, we extract relevant features from the music data, including metadata such as genre, artist, album, mood, tempo, and user interactions such as play count and likes.

Next, we conduct exploratory data analysis (EDA) to gain insights into the characteristics and distributions of the music data. Through visualization and statistical analysis, we identify patterns, trends, and correlations within the dataset, informing subsequent modeling decisions.

For model development, we select appropriate machine learning algorithms tailored to recommendation tasks. This may include collaborative filtering, content-based filtering, or hybrid models that combine both approaches. Leveraging Python libraries such as scikit-learn, TensorFlow, or PyTorch, we train machine learning models on the preprocessed data. We perform feature engineering and hyperparameter tuning to optimize model performance, evaluating them using metrics such as precision, recall, and mean average precision (MAP).

A crucial aspect of our methodology involves the integration of content-based and collaborative filtering techniques to develop a hybrid recommendation system. By incorporating user preferences and historical interactions, we personalize recommendations, aiming to strike a balance between accuracy and diversity in suggestions.

Real-time processing and deployment are vital for ensuring timely and responsive recommendations. We implement efficient pre-processing methods to handle data in motion, utilizing Python frameworks like Flask or FastAPI to develop RESTful APIs for serving recommendation requests. Deployment on scalable cloud infrastructure such as AWS or Google Cloud Platform ensures reliability and performance, with monitoring and logging mechanisms in place to track system performance and user interactions in real-time.

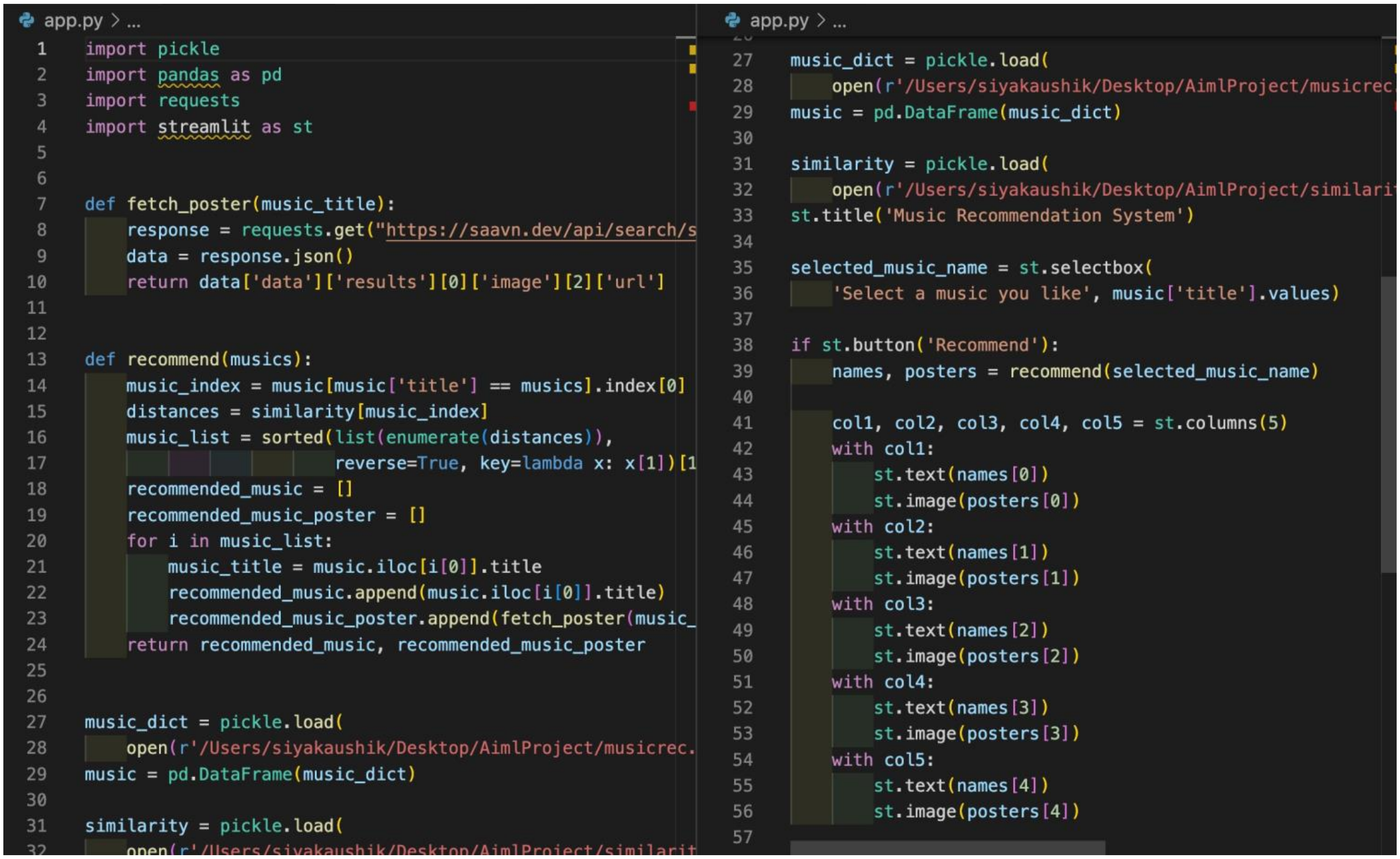
Continuous evaluation and optimization are essential for refining the recommendation system over time. We conduct extensive evaluation through A/B testing or user studies to measure user satisfaction and engagement. By monitoring system performance and user feedback, we identify areas for improvement, iterating on the system design and algorithms to optimize performance and enhance user experience iteratively.

By following this methodology and leveraging Python libraries for machine learning, we aim to develop a music recommendation system that delivers personalized and engaging recommendations, enriching the music listening experience for users.

Source Code:-

```
import numpy as np
import pandas as pd
df=pd.read_csv("ex.csv")
df
df.isnull().sum()
df.dropna(inplace=True)
df.isnull().sum()
df.duplicated().sum()
df=df.drop_duplicates()
df.duplicated().sum()
df.shape
df.head()
df['User-Rating']
l=[]
for i in df['User-Rating']:
    l.append(i[1:3])
l
df['Album/Movie']=df['Album/Movie'].str.replace(' ','')
df['Singer/Artists']=df['Singer/Artists'].str.replace(' ','')
df
df['Singer/Artists']=df['Singer/Artists'].str.replace(',',' ')
df
df['tags']=df['Singer/Artists']+' '+df['Genre']+' '+df['Album/Movie']+' '+df['User-Rating']
df['tags'][0]
new_df=df[['Song-Name','tags']]
new_df
new_df['tags']=new_df['tags'].apply(lambda x:x.lower())
new_df
from sklearn.feature_extraction.text import CountVectorizer
cv=CountVectorizer(max_features=2000)
vectors=cv.fit_transform(new_df['tags']).toarray()
vectors.shape
cv.get_feature_names_out()
from sklearn.metrics.pairwise import cosine_similarity
similarity=cosine_similarity(vectors)
sorted(list(enumerate(similarity[0])),reverse=True,key=lambda x:x[1])
new_df.rename(columns ={'Song-Name':'title'},inplace=True)
def recommend(music):
    music_index=new_df[new_df['title']==music].index[0]
    distances=similarity[music_index]
    music_list=sorted(list(enumerate(distances)),reverse=True,key=lambda x:x[1])[1:6]
    for i in music_list:
        print(new_df.iloc[i[0]].title)
recommend('Coca Cola')
df.head(50)
import pickle
pickle.dump(new_df.open('musicrec.pkl','wb'))
pickle.dump(similarity.open('similarities.pkl','wb'))
```

Fig.1. Code of Music Recommendation System



```
import pickle
import pandas as pd
import requests
import streamlit as st


def fetch_poster(music_title):
    response = requests.get("https://saavn.dev/api/search/s
    data = response.json()
    return data['data']['results'][0]['image'][2]['url']


def recommend(musics):
    music_index = music[music['title'] == musics].index[0]
    distances = similarity[music_index]
    music_list = sorted(list(enumerate(distances)),
                        reverse=True, key=lambda x: x[1])[1
    recommended_music = []
    recommended_music_poster = []
    for i in music_list:
        music_title = music.iloc[i[0]].title
        recommended_music.append(music.iloc[i[0]].title)
        recommended_music_poster.append(fetch_poster(music_
    return recommended_music, recommended_music_poster


music_dict = pickle.load(
    open(r'/Users/siyakaushik/Desktop/AimlProject/musicrec.
music = pd.DataFrame(music_dict)

similarity = pickle.load(
    open(r'/Users/siyakaushik/Desktop/AimlProject/similari
st.title('Music Recommendation System')

selected_music_name = st.selectbox(
    'Select a music you like', music['title'].values)

if st.button('Recommend'):
    names, posters = recommend(selected_music_name)

    col1, col2, col3, col4, col5 = st.columns(5)
    with col1:
        st.text(names[0])
        st.image(posters[0])
    with col2:
        st.text(names[1])
        st.image(posters[1])
    with col3:
        st.text(names[2])
        st.image(posters[2])
    with col4:
        st.text(names[3])
        st.image(posters[3])
    with col5:
        st.text(names[4])
        st.image(posters[4])
```

Fig.2. Code Of Visual Studio Code

Output:-

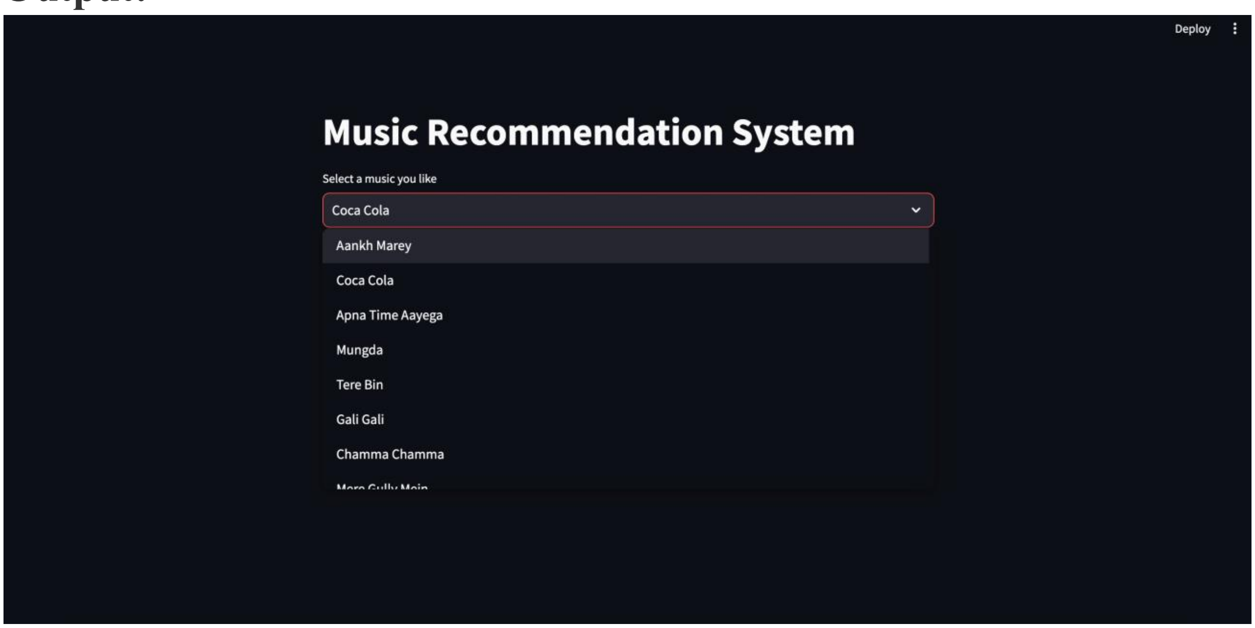


Fig.3. Output of Music Recommendation System

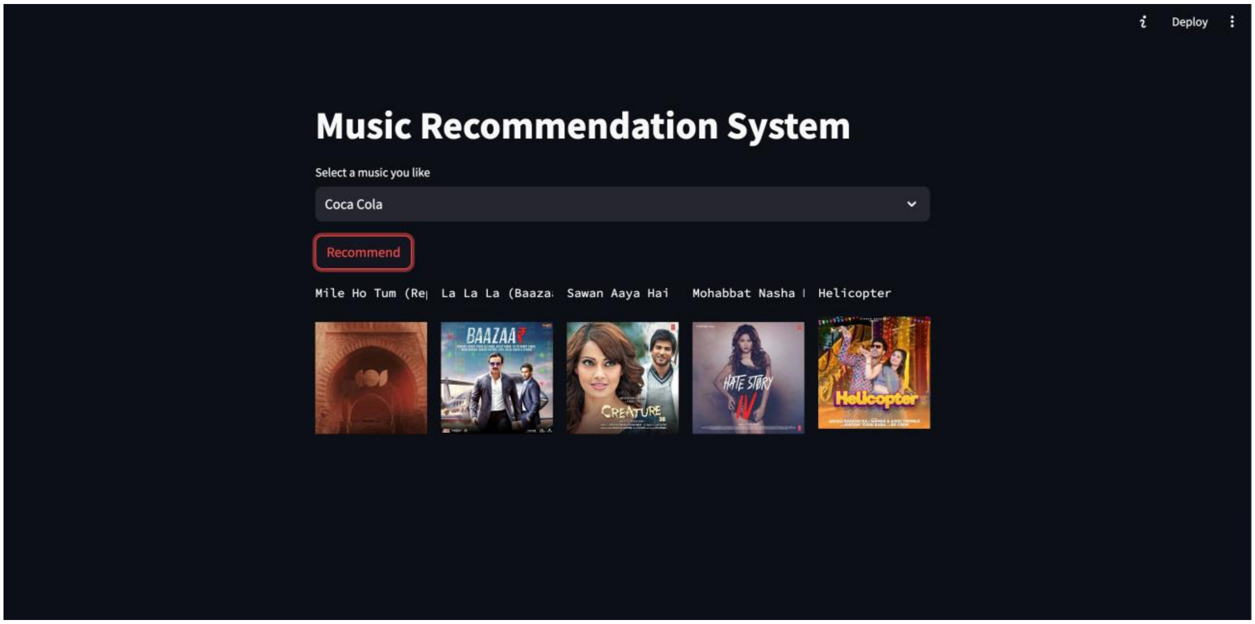


Fig.4. Output of Music Recommendation System

References :-

[1]Schedl, M. (2019). Deep learning in music recommendation systems. Frontiers in Applied Mathematics and Statistics, 5, 457883.

[2]Wen, X. (2021). Using deep learning approach and IoT architecture to build the intelligent music recommendation system. Soft Computing, 25(4), 3087-3096. [3] Dataset