MINI PROJECT

(2020 – 21)

Music Recommender System

MID – TERM REPORT

Institute of Engineering & Technology

Submitted By:

Ishika Singh (181500286)
Jaya Shukla (181500294)
Nishtha Kapoor (181500431)
Nitya Shukla (181500438)
Raman Shukla (181500554)

Supervised By:

Mr. Aamir Khan

(Assistant Professor)

Department of Computer Engineering & Applications

TABLE OF CONTENTS

ABSTRACT

On regular basis, individuals dependably require few advices when settling on the choices. Regardless of whether it is which movie to watch at Friday night or are there any fascinating new items accessible on the Amazon. Under this unique situation, we fabricate a framework that can consequently prescribe new melodies to customers dependent on their listening history before. Lots of businesses are nowadays using recommender systems for their benefit like Amazon and flip kart for selling products (e-commerce), WYNK music and GAANA.com for music streaming, for selling books, for movies, YouTube for videos recommendations. This helps both businesses and users as businesses are getting monetary benefit by attracting customers and users are getting benefitted by getting better services. Everybody is using recommender systems nowadays in various forms and day to day these are getting improved.

This project mainly focuses on only suggesting music for music lovers to help them listen to the songs those they may like.

This framework enables clients to find new collections or tunes making the melodic list accessible for tuning in. For music enthusiastic people music is life and also a larger part of everybody's lives because everything in this universe can be correlated to frequency and vibrations. Music helps us tuning ourselves with the universe and best thing about music is that nothing can relax you more than a pleasing melody. Due to all the good things about music and high demand of recommender systems in the market we have chosen to do this project.

INTRODUCTION

1.1 General Introduction to the topic

Rapid development of mobile devices and internet has made possible for us to access different music resources freely. The number of songs available exceeds the listening capacity of single individual. People sometimes feel difficult to choose from millions of songs. Moreover, music service providers need an efficient way to manage songs and help their costumers to discover music by giving quality recommendation. Thus, there is a strong need of a good recommendation system. Currently, there are many music streaming services, like Pandora, Spotify, etc. which are working on building high-precision commercial music recommendation systems. These companies generate revenue by helping their customers discover relevant music and charging them for the quality of their recommendation service. Thus, there is a strong thriving market for good music recommendation systems. Music recommender system is a system which learns from the users past listening history and recommends them songs which they would probably like to hear in future.

The Internet evolution continuously generates several changes in social habits related to communication and lifestyle. Nowadays lots of music industries like ***amazon music, spotify, wynk music, gaana.com*** are using recommender systems and the old - fashioned way of selling music has changed to a totally different cloud based.

The continue increasing in connection speed and trends in web development technologies, given rise to large web systems nowadays visited daily. Among these advanced systems, there are systems that allow users to listen music online without the need of downloading it to their personal computer.

With the promotion of the Internet and the advancement of E-trade, the Ecommerce destinations offer a great many items available for purchasing. Picking among such a large number of choices is challenging for buyers. So, clients typically lose all sense of direction in the huge space of ware data and can't discover the products they truly need.

Recommender frameworks have risen in light of this issue. A recommender framework for an E-trade site prescribes items that are probably going to meet client's requirements.
Suggestion frameworks have rapidly changed the way in which the life less sites can now interact and speak with their clients and users.

Recommender architecture is used by the E-business objective to propose and suggest items and services which are similar to their clients. There are many constraints and parameters on which an internet service provider recommends a user certain choices and options depending upon some restrained set of parameters. These parameters can include language, age, nationality, history, likes, ratings, purchase and many more.

Moreover, these strategies have been used extensively and are a piece of personalization on a site, since this enable the site to adjust to every client in accordance to the client or user.
There is a wide variety of these systems and new alternatives are constantly emerging increasingly improved. Some are simple players providing playlist functionality ***(prostopleer.com)***, others accompany the player with a recommendation system of similar artists ***(www.spotify.com)***, also there are complex collaborative systems in which hundreds of people leave comments on songs ***(www.pandora.com)*** and have the chance of interact with each other as in the newly emerging social networks.

Figure 1.2(Recommendor Architecture)

1.2 Why Recommender System?

As the web moved from a proprietor model to an open publicly supporting model and enabled individuals to contribute unreservedly, it saw an exponential ascent in the measure of substance accessible, which was something to be thankful for.

More or less, a recommender framework is a framework that predicts client reaction to an assortment of choices. Anticipating what the client may get next is the basic point of a recommender framework.

There is a broad class of web applications that include foreseeing the client's reaction to choices. Such an office is known as a recommender framework.

In today's time services like Netflix, Spotify, YouTube offers their customers with plenty of choices. A person using such a service is known as client or user and to assist user, services can use information filtering to recommend items to users. Items can be things such as books, music, movies, news etc.

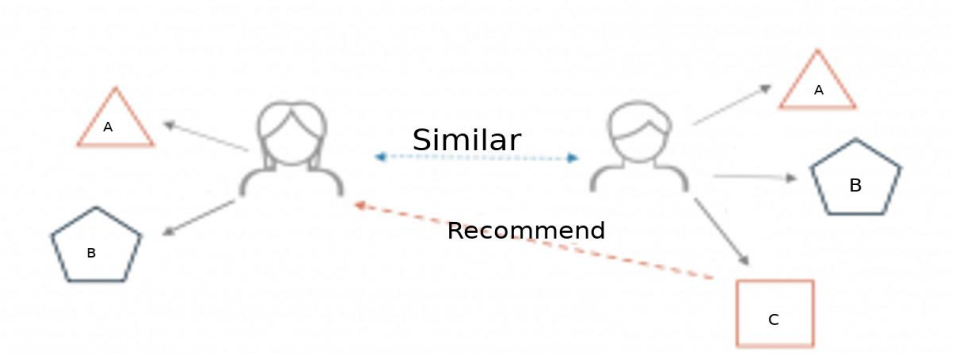
Recommender systems need information for functioning, data about a particular user. This particular data can be fetched directly or indirectly. Directly collecting data means that user of a particular service gives feedback and review of the item. Indirectly means that system will analyse the user's interaction

with the particular service consisting of history and present services.

The most common approaches people have used implementing various recommender systems are collaborative filtering and content - based models. These algorithms aim to find similarity between various user's various songs and artists.

In other words, a recommender system is a simple algorithm which aims at providing the most relevant information to a user by discovering patterns in a dataset. The algorithm rates the items and shows the user the items that they would rate highly. An example of recommendation in action is when you visit Amazon (Prime Video) and you notice that some items are being recommended to you or when Netflix recommends certain movies to you. They are also used by Music streaming applications such as Spotify to recommend music that you might like.

A simple illustration of how recommender system works in context of an e-commerce website:



In the above illustration, two users buy the same items A and B from an e-commerce store. When this happens the similarity index of these two users is computed. Depending on the score the system can recommend item C to the other user because it detects that those two users are similar in terms of the items they purchase.

Every user has different preferences and likes. In addition, even the taste of a single user can vary depending on a large number of factors, such as mood or season or type of activity the user is

doing. For example, the type of music one would like to hear while exercising differs greatly from the type of music he'd listen to while cooking the dinner. They must explore new domains to discover more about the user, while still making the most of what is already known about of the user.

1.3 Hardware and Software Requirements

The particular software and hardware requirements are:

1. Software Requirements:

- Python
- Anaconda Navigator
- Jupyter Notebook
- Jupyter Lab
- Google Colab
- VS Code

} anyone from these

2. Hardware Requirements:

- Windows10
- Ubuntu
- Mac

} anyone from these

- i3 or higher
- 4GB RAM or higher

TECHNOLOGY USED IN THE PROJECT

- ## Python:

  Python is an interpreted, high-level, general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant whitespace.
  I'll be using python instead of R as python is a general-purpose programming language whereas on the other hand R is suitable for statistics, not for software development.
  Python as an interpreted scripting language has lot of disadvantages as well. Basically, in terms of speed of execution and memory management. Scientific computing libraries have been developed to overcome basic disadvantages of Python.
  Some of the commonly used libraries of Python are **NUMPY, PANDAS, SEABORN, KERAS, MATPLOTLIB, SKLEARN.**

- ## Machine Learning:

  Machine Learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. The process of learning begins with observations or data, such as examples, direct experience, or instruction, in order to look for patterns in data and make better decisions in the future based on the examples that we provide. The primary aim is to allow the computers learn automatically without human intervention or assistance and adjust actions accordingly.

- ## Django:

  Django is a high – level Python Web Framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel. It's free and open – source.

APPROACHES USED IN THE PROJECT

There are various types of recommender systems with different approaches and some of them are discussed as below:

- **Content-based Filtering System:**

  In content-based filtering, items are recommended based on comparisons between item profile and user profile.
  A user profile is content that is found to be relevant to the user in form of keywords. A user profile might be seen as a set of assigned keywords (terms, features) collected by algorithm from items found relevant (or interesting) by the user.
  A set of keywords of an item is the Item profile.
  Consider a scenario where a person goes to buy his favourite cake 'X' to a pastry shop. Unfortunately, cake 'X' has been sold out and as a result of this the shopkeeper recommends the person to buy cake 'Y' which is made up of ingredients similar to cake 'X'. This is an instance of content-based filtering.

  In other words, content-based systems use metadata such as genre, producer, actor, musician to recommend items say music or movies.

  You can get music recommendations from certain artists because you liked their music. Content-based systems are based on the idea that if you liked a certain item you are most likely to like something that is similar to it.

  ## Advantages of content-based filtering are:

  - They are capable of recommending unrated items.
  - Content-based recommender systems need only the rating of the concerned user, not any other user of the system.

  ## Disadvantages of content-based filtering are:

  - It does not work for a new user who has not rated any item yet.
  - Limited Content Analysis - The recommendation does not work if the system fails to distinguish the items that a user likes from the items that he does not like.

- ## Collaborative Filtering-based System:

  In collaborative filtering, the behaviour of a group of users is used to make recommendations to other users. The recommendation is based on the preference of other users. A simple example would be recommending a song to a user based on the fact that their friend liked the song.
  There are two types of collaborative models ***Memory-based*** methods and ***Model-based*** methods.
  The advantage of memory-based techniques is that they are simple to implement and the resulting recommendations are often easy to explain. They are further divided into two:

  - ## User-based collaborative filtering:

    These systems recommend products to a user that similar users have liked. For measuring the similarity between two users we can either use Pearson Correlation or Cosine Similarity.
    In other words, in this model, products are recommended to a user based on the fact that the products have been liked by users similar to the user.
    For example, if A and B like the same song and a new song come out that A like, then we can recommend that song to B because A and B seem to like the same song.
    Although computing user-based CF is very simple, but it suffers from several problems. One main issue is that user's preference can change over time. It indicates that precomputing the matrix based on their neighbouring users may lead to bad performance. To tackle this problem, we can apply item-based CF.

  - **Item-based collaborative filtering:**

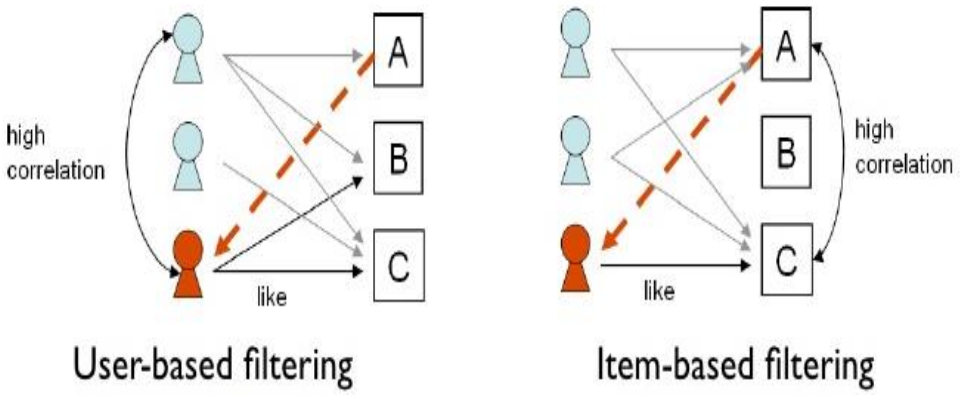
    Instead of measuring the similarity between users, the item-based CF recommends items based on their similarity with the items that the target user rated.
    These systems identify similar items based on user's previous ratings.
    For example - if users A, B, and C gave a 5-star rating to songs X and Y then when a user D listens song Y, they also get a recommendation to listen X because the system identifies song X and Y as similar based on the ratings of users A, B, and C.

Advantages of collaborative filtering-based systems:

- It is dependent on the relation between users which implies that it is content-independent.
- They can make real quality assessment of items by considering other people's experience.

Disadvantages of collaborative filtering-based systems:

- These systems cannot provide recommendations for new items since there are no user ratings on which to base a prediction.
- In order for CF based system to work, group with similar characteristics are needed.



PROBLEM DEFINITION / PROBLEM STATEMENT

Recommendation system help users find and select items like music, movies, or books from the huge number available on the web or in the other electronic information sources.

The main motive behind this project is to simply design and implement a music recommendation system by using the item – based collaborative filtering system and understand the working or designing behind the system.

OBJECTIVE

Learning Objective:

Learning objective of doing this project was to first to learn about machine learning and its key concepts and various data mining techniques and algorithms. Other goal was to learn different types of machine learning algorithm and how to use them.

Just learning algorithms doesn't makes you an engineer the real task is to learn which is the right algorithm to apply for a specific project.

Outcome Objective:

The main objective in terms of outcome was to create a framework for users which can help them by suggesting the right songs for them.

This project aims to find the correlation and similarity between different music lovers, their tastes and various songs so that if a user's taste is similar to that of the other user, we can recommend the songs of one to another on the basis of similar taste.

Or if a song is similar to that of the other song, we can suggest that song to the user that listens the first one.

The other objective of this project is to reduce the time that user generally wastes on looking for the right song.

DATASET USED FOR THE PROJECT

We used data provided by ***Million Song Data Challenge*** hosted by Kaggle. It was released by ***Columbia University Laboratory*** for the Recognition and Organization of Speech and Audio.

The data is open, meta-data, audio - content analysis, etc. are available for all the songs. It is also very large and contains around 48 million (userid, songid, play count) triplets collected from histories of over one million users and metadata (280 GB) of millions of songs. But the users are anonymous here and thus information about their demography and timestamps of listening events is not available. The feedback is implicit as play-count is given instead of explicit ratings.

This dataset included the song listening history of around million people but because the dataset was too big and also the processing of such a large dataset is highly memory and CPU-intensive, so we had to take the subset of the dataset for our project.

The dataset consists of various attributes like user id, song counts, language, artist, genre, and year.

IMPLEMENTATION

Using Django Built-in Login System We have Implemented registration, login, logout, password reset, and password change.

On the log in page, we need a form with **username** and **password**, a button with the main action (log in) and two alternative paths: sign up page and password reset page.

On the sign up page, we should have a form with four fields: **username**, **email address**, **password**, and **password confirmation**. The user should also be able to reach the log in page.

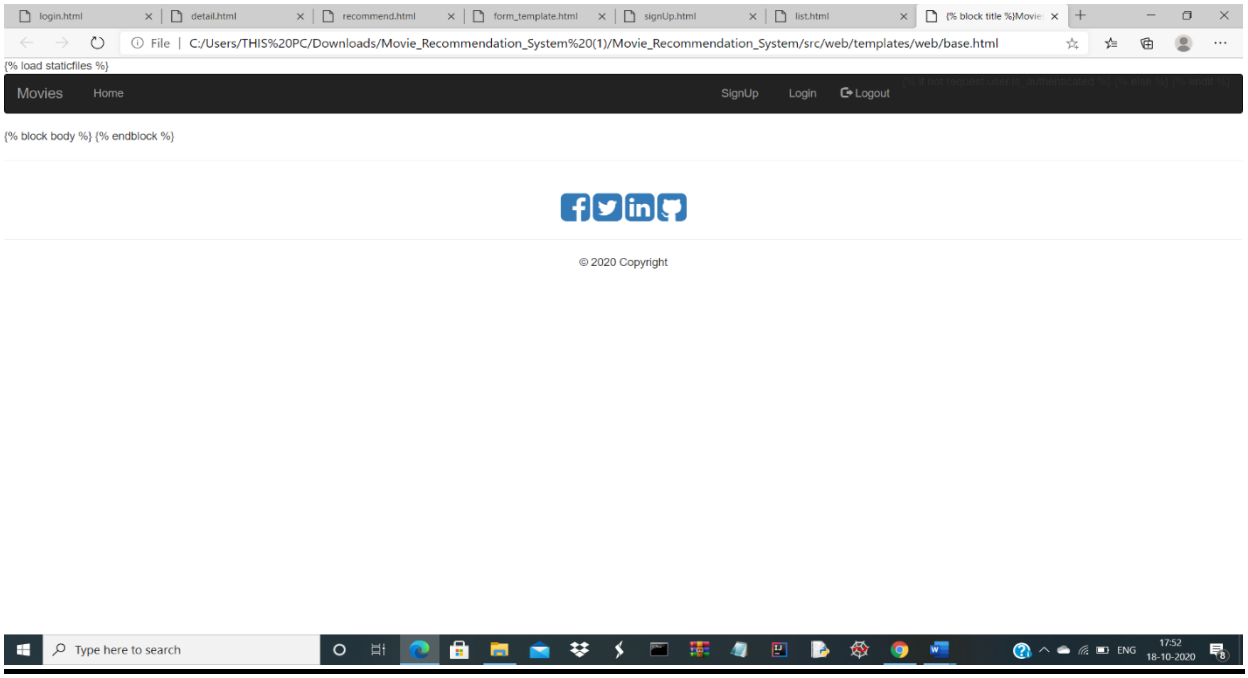
On the password reset page, we will have a form with just the **email address**.

Then, after clicking on a special token link, the user will be redirected to a page where they can set a new password:

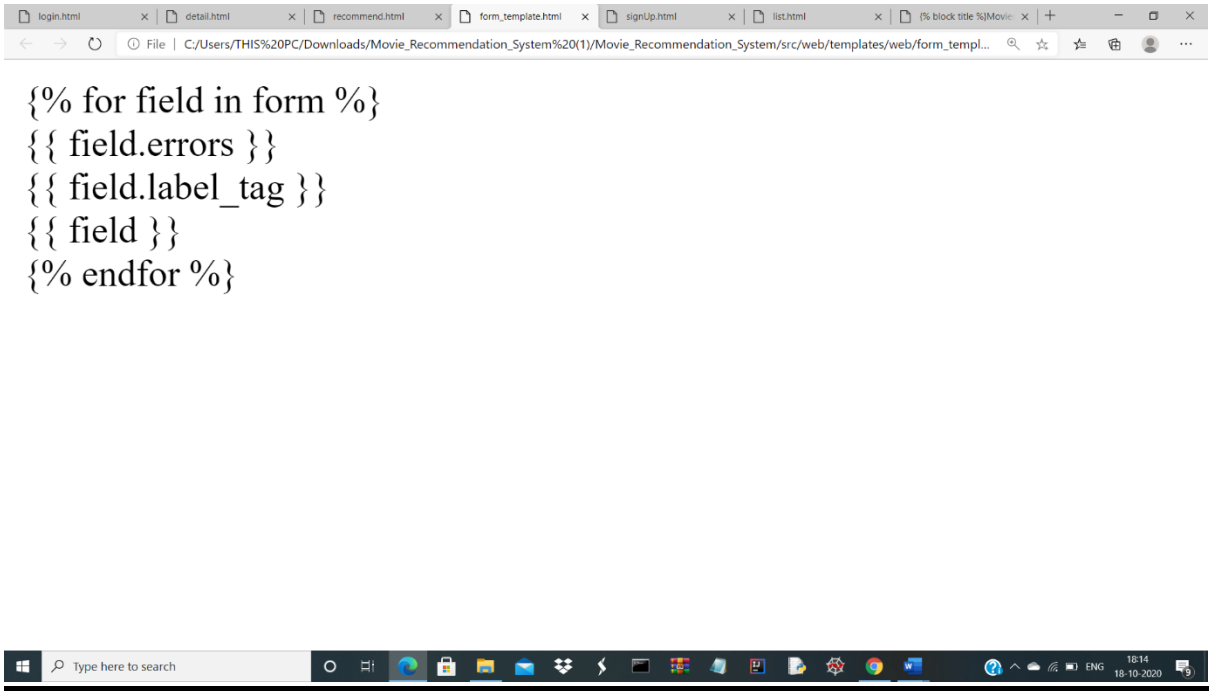
To manage all this information, we can break it down in a different app. In the project root, in the same page where the **manage.py** script is.

Below are some Screen shots of basic structure of these pages.

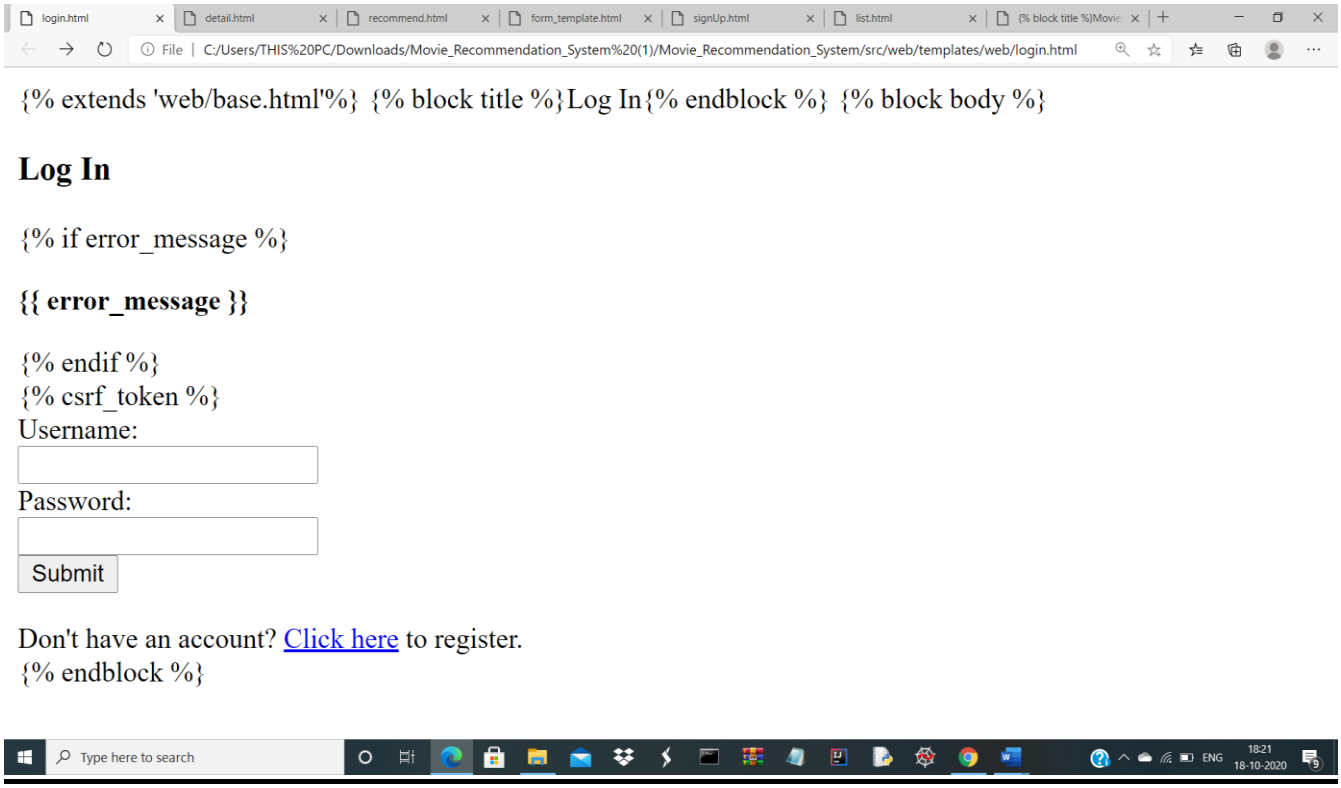
- BASE PAGE


login.html
detail.html
recommend.html
form_template.html
signUp.html
list.html
{% block title %}Movie
File | C:/Users/THIS%20PC/Downloads/Movie_Recommendation_System%20(1)/Movie_Recommendation_System/src/web/templates/web/base.html
{% load staticfiles %}
Movies
Home
SignUp
Login
Logout
{% block body %} {% endblock %}
© 2020 Copyright
Type here to search
ENG
17:52
18-10-2020

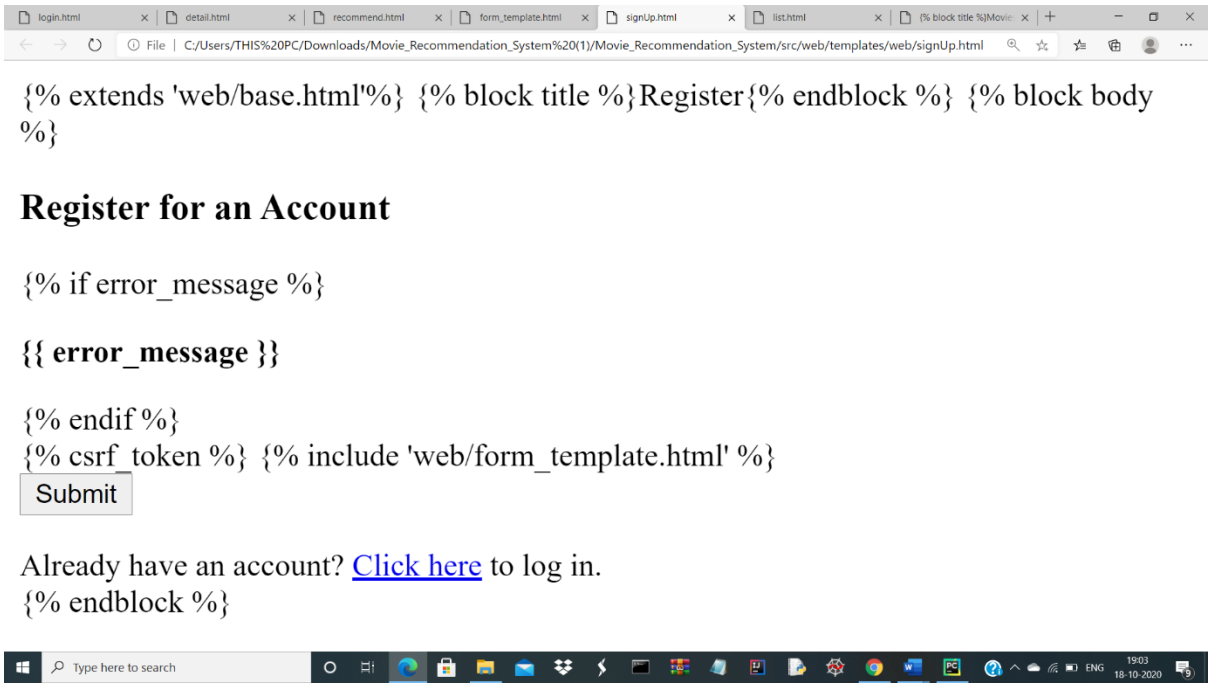
- FORM TEMPLATE



- LOGIN PAGE


login.html
detail.html
recommend.html
form_template.html
signUp.html
list.html
{% block title %}Movie
File | C:/Users/THIS%20PC/Downloads/Movie_Recommendation_System%20(1)/Movie_Recommendation_System/src/web/templates/web/login.html
{% extends 'web/base.html'%} {% block title %}Log In{% endblock %} {% block body %}
Log In
{% if error_message %}
{{ error_message }}
{% endif %}
{% csrf_token %}
Username:
Password:
Submit
Don't have an account? Click here to register.
{% endblock %}
Type here to search
ENG
18:21
18-10-2020

- SIGN UP PAGE


login.html
detail.html
recommend.html
form_template.html
signUp.html
list.html
File | C:/Users/THIS%20PC/Downloads/Movie_Recommendation_System%20(1)/Movie_Recommendation_System/src/web/templates/web/signUp.html
{% extends 'web/base.html'%} {% block title %}Register{% endblock %} {% block body %}
Register for an Account
{% if error_message %}
{{ error_message }}
{% endif %}
{% csrf_token %} {% include 'web/form_template.html' %}
Submit
Already have an account? Click here to log in.
{% endblock %}
Type here to search
ENG
19:03
18-10-2020

PROGRESS

PART 1 -Completed

Our first job is to integrate our dataset,To integrate both triplet_file and metadata_file, we are going to use a popular Python library called pandas.

We first define the two files we are going to work with.

We then read the table of triplet_file using pandas and define the 3 columns as user_id, song_id and listen_count (`df` here means dataframe).

We also read the metadat_file and going to combine the metadata_file with triplets_file. Whenever we combine 2 or more datasets, there will be duplicate columns. Here we drop the duplicates between 2 datasets using song_id.

Using command `song_df.head()` allows us to visualize the combined data.

PART 2- Pending

Our second step is data transformation, where we're going to select a subset of this data (the first 10,000 songs). We then merge the song and artist_name into one column, aggregated by number of time a particular song is listened too in general by all users.

PART 3- Pending

Our next step is to create a ML personalized song recommender system by leveraging the item similarity based collaborative filtering model. Recall that recommender system is divided into 2 types: content based and collaborative based. Content based system predicts what a user like based on what that user like in the past. Collaborative based system predict what a particular user like based on what other similar users like.

PART 4- Partially Completed

Building a barebones Django backend as a REST AP

FUTURE SCOPE

Designing a personalised music recommender is complicated, and it is challenging to thoroughly understand the users' needs and meet their requirements. As the future research direction will be mainly focused on usercentric music recommender systems. A survey among athletes showed practitioners in sport and exercise environments tend to select music in a rather arbitrary manner without full consideration of its motivational characteristics. Therefore, future music recommender should be able to lead the users reasonably choose music. To the end, we are hoping that through this study we can build the bridge among isolated research in all the other disciplines.