Report On

Movie Recommendation System

Submitted in partial fulfillment of the requirements of the Mini project in
Semester VIII of Final Year Computer Engineering

By
Prasanna Lad (Roll No. 22)
Devesh Panchal (Roll No. 31)
Rutuja Parab (Roll No. 32)

Supervisor
Prof. Sunil Katkar

University of Mumbai

Vidyavardhini's College of Engineering & Technology

Department of Computer Engineering

(2020-21)

Vidyavardhini's College of Engineering & Technology

Department of Computer Engineering

CERTIFICATE

This is to certify that the project entitled Movie Recommendation System is a bonafide work of Prasanna Lad (Roll No. 21), Devesh Panchal (Roll No. 31), Rutuja Parab (Roll No. 32) submitted to the University of Mumbai in partial fulfillment of the requirement for the Mini project in semester VIII of final Year Computer Engineering.

Supervisor

Prof. Sunil Katkar

Internal Examiner   External Examiner

Dr Megha Trivedi   Dr. H.V. Vankudre
Head of Department   Principal

ABSTRACT

Nowadays, the recommendation system has made finding the things easy that we need. Movie recommendation systems aim at helping movie enthusiasts by suggesting what movie to watch without having to go through the long process of choosing from a large set of movies which go up to thousands and millions that is time consuming and confusing. In this article, our aim is to reduce the human effort by suggesting movies based on the user's interests. To handle such problems, we introduced a model combining both content-based and collaborative approach. It will give progressively explicit outcomes compared to different systems that are based on content-based approach. Content-based recommendation systems are constrained to people, these systems don't prescribe things out of the box, thus limiting your choice to explore more. Hence, we have focused on a system that resolves these issues.

Recommender systems research has incorporated a wide variety of artificial intelligence techniques including machine learning, data mining, user modeling, case-based reasoning, and constraint satisfaction, among others. Recommender systems are tools for interacting with large and complex information spaces. They provide a personalized view of such spaces, prioritizing items likely to be of interest to the user. In most general terms, Recommendation systems are defined as the techniques used to predict the rating one individual will give to an item or social entity. These items can be books, movies, restaurants and things on which individuals have different preferences. These preferences are being predicted using two approaches first content-based approach which involves characteristics of an item and second collaborative filtering approaches which takes into account user's past behavior to make choices. In collaborative filtering, partners are chosen who will make recommendations because they share similar ratings history with the target user. One partner who have similar ratings to the target user may not be a reliable predictor for a particular item. So the past record of the partner of making a reliable recommendation also needs to be take into consideration which is dictated by trustworthiness of a partner. In order to keep track of past records of a recommender reputation systems comes into the picture those who actually assign reputation ratings to the partners.

INDEX

1. PROBLEM STATEMENT

This project is based on recommendation system that recommends different movies to the users. This project aims to calculate the similarities between different users and then recommend movie to them as per the ratings given by the different users of similar tastes. This will provide a precise recommendation to the user. This project is based on providing recommendations based on the similarity of the movies and the similarities in the user preferences.

The first thing we would need to collect is the preferences of the different people. These preferences will possibly be the ratings (as in case of a Movie Recommender System) of different users who have rated some movies in the past. The ratings will be on a scale of 1 to 5 where 1 depicts strong dislike and 5 suggest a strong liking for the movie. A rating of 3-4 would mean average opinion on that movie. After the data has been collected, a technique has to be devised to present similarities between different users and generate a better prediction for new users.

2. INTRODUCTION

In today's world, every customer is faced with multiple choices. For example, If I'm looking for a specific item without any specific idea of what I want, there's a wide range of possibilities. I might waste a lot of time browsing around on the internet and trawling through various sites hoping to strike gold. I might look for recommendations from other people. But if there was a site or app which could recommend me based on what my preferences are, that would be a massive help. Instead of wasting time on various sites, I could just get 10 recommendations tailored to my taste. This is what recommendation engines do and their power is being harnessed by most businesses these days. From Amazon to Netflix, Google to Goodreads, recommendation engines are one of the most widely used applications of Machine Learning techniques.

Recommendation System is an information tool which helps users to find out the items which they want from the large no of items available. Main goal of recommendation system is to forecast the rating which a specific user gives to an item. It helps the user to find the best solution from the available list of items. Many companies use recommendation system so that they can serve their user and raise their profit like Netflix, YouTube, Amazon and others Still now it is a good topic of research because to find what the user wants from available resource is a big challenge, as our choice keeps on changing with time. Nowadays what we purchase online is recommendation. For example, if we want to buy books, listen music, watch movies etc there is one recommendation system that is working in background which suggest the user based on his previous actions. Many platforms like Netflix which suggest movies, Amazon which suggest products, Spotify that suggest music, LinkedIn that is used for recommending jobs or any social networking sites which suggest users, all these work on recommendation system. By using these recommendation engine users can easily find out what he wants according to his/her choice. So to build an effective recommender system is also a challenge because user's preference keeps on changing with time.

3. BLOCK DIAGRAM
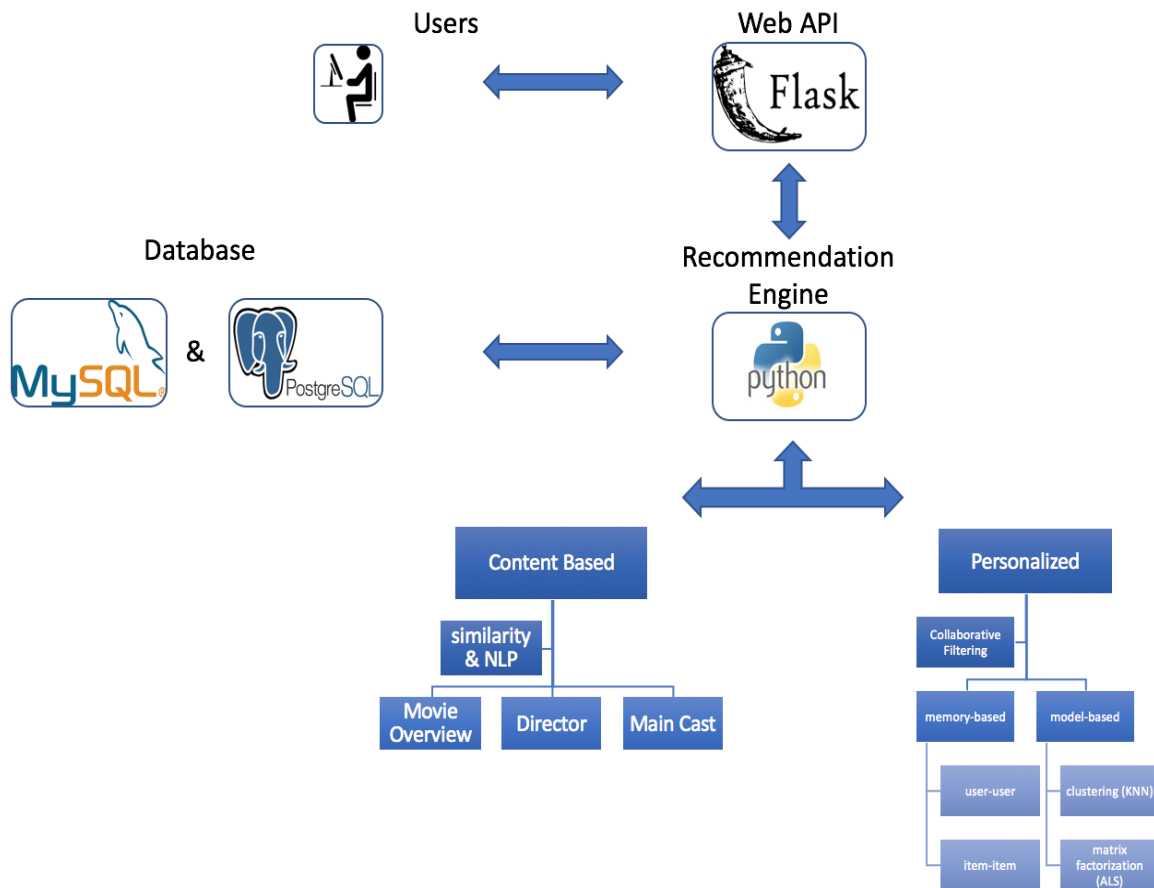
Collaborative filtering based systems (CF based systems):

Collaborative filtering system recommends items based on similarity measures between users and/or items. The system recommends items preferred by similar users. This is based on the scenario where a person asks his friends, who have similar tastes, to recommend him some movies.

Advantages of collaborative filtering based systems:

- It is dependent on the relation between users which implies that it is content-independent.
- CF recommender systems can suggest serendipitous items by observing similar-minded people's behavior.
- They can make real quality assessment of items by considering other peoples experience.

4. WORKING
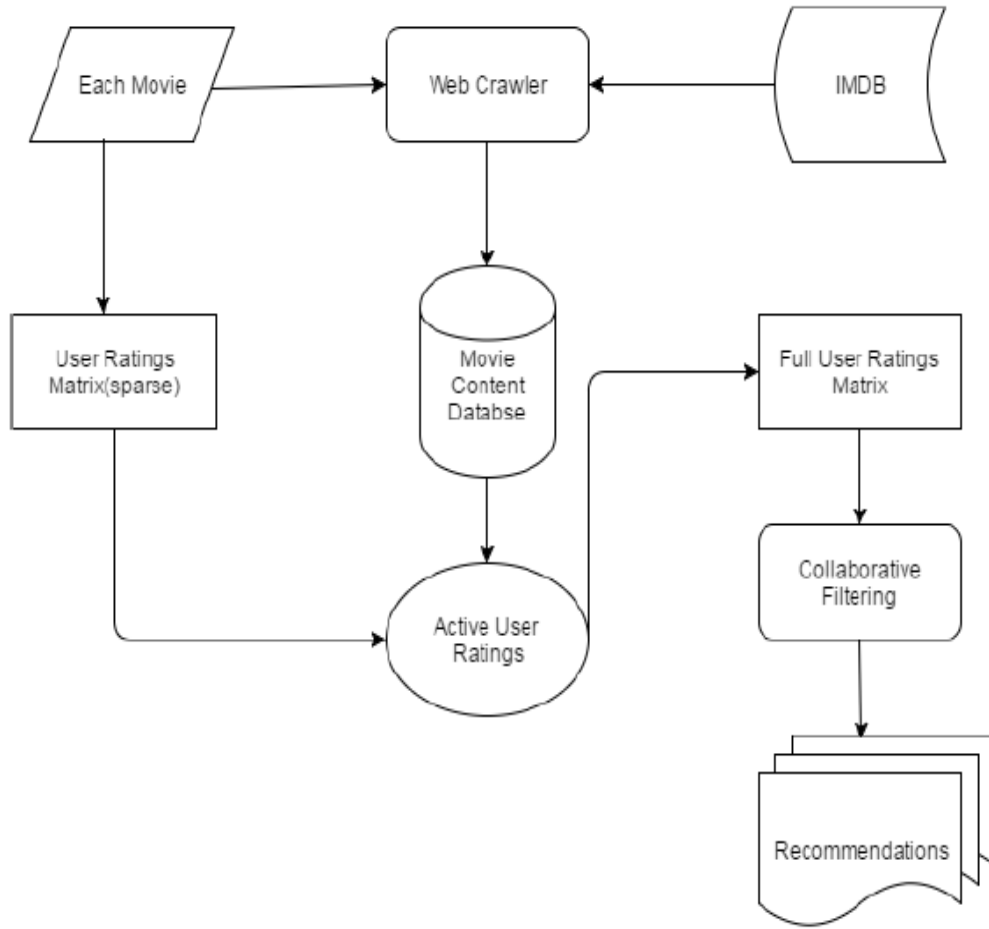
Collaborative filtering memory based:

The memory-based approach uses user rating data to compute the similarity between users or items. Typical examples of this approach are neighbourhood-based CF and item-based/user-based top-N recommendations. For example, in user based approaches, the value of ratings user u gives to item i is calculated as an aggregation of some similar users' rating of the item:

$$r_{u,i} = \text{aggr}_{u' \in U}\ r_{u',i}$$

where U denotes the set of top N users that are most similar to user u who rated item i. Some examples of the aggregation function include:

$$r_{u,i} = \frac{1}{N} \sum_{u' \in U} r_{u',i}$$

$$r_{u,i} = k \sum_{u' \in U} \text{simil}(u, u') r_{u',i}$$

where k is a normalizing factor defined as , and

$$k = 1 / \sum_{u' \in U} |\text{simil}(u, u')|$$

where ru is the average rating of user u for all the items rated by u.

The neighborhood-based algorithm calculates the similarity between two users or items, and produces a prediction for the user by taking the weighted average of all the ratings. Similarity computation between items or users is an important part of this approach. Multiple measures, such as Pearson correlation and vector cosine based similarity are used for this.

The Pearson correlation similarity of two users x, y is defined as

$$\text{simil}(x, y) = \frac{\sum_{i \in I_{xy}} (r_{x,i} - \bar{r_x})(r_{y,i} - \bar{r_y})}{\sqrt{\sum_{i \in I_{xy}} (r_{x,i} - \bar{r_x})^2} \sqrt{\sum_{i \in I_{xy}} (r_{y,i} - \bar{r_y})^2}}$$

where I_{xy} is the set of items rated by both user x and user y.

The cosine-based approach defines the cosine-similarity between two users x and y as:

$$\text{simil}(x, y) = \cos(\vec{x}, \vec{y}) = \frac{\vec{x} \cdot \vec{y}}{||\vec{x}|| \times ||\vec{y}||} = \frac{\sum_{i \in I_{xy}} r_{x,i} r_{y,i}}{\sqrt{\sum_{i \in I_x} r_{x,i}^2} \sqrt{\sum_{i \in I_y} r_{y,i}^2}}$$

The user based top-N recommendation algorithm uses a similarity-based vector model to identify the k most similar users to an active user. After the k most similar users are found, their corresponding user-item matrices are aggregated to identify the set of items to be recommended. A popular method to find the similar users is the Locality-sensitive hashing, which implements the nearest neighbor mechanism in linear time.

The advantages with this approach include: the explainability of the results, which is an important aspect of recommendation systems; easy creation and use; easy facilitation of new data; content-independence of the items being recommended; good scaling with co-rated items.

There are also several disadvantages with this approach. Its performance decreases when data gets sparse, which occurs frequently with web-related items. This hinders the scalability of this approach and creates problems with large datasets. Although it can efficiently handle new users because it relies on a data structure, adding new items becomes more complicated since that representation usually relies on a specific vector space. Adding new items requires inclusion of the new item and the re-insertion of all the elements in the structure.

5. SOFTWARE AND HARDWARE REQUIREMENTS.

5.1 Software requirements:

- Language: Python and HTML.

  - ➢ **Python** is a general-purpose coding language—which means that, unlike HTML, CSS, and JavaScript, it can be used for other types of programming and software development besides web development. That includes back end development, software development, data science and writing system scripts among other things.

  - ➢ **HTML** stands for Hyper Text Markup Language. HTML is the standard markup language for creating Web pages. HTML describes the structure of a Web page. HTML consists of a series of elements. HTML elements tell the browser how to display the content.

- Libraries: Pandas and Pickle.

  - ➢ **Pandas** is an open source Python package that is most widely used for data science/data analysis and machine learning tasks. It is built on top of another package named Numpy, which provides support for multi-dimensional arrays.

  - ➢ The Python pickle module is another way to serialize and deserialize objects in Python. It differs from the json module in that it serializes objects in a binary format, which means the result is not human readable. However, it's also faster and it works with many more Python types right out of the box, including your custom-defined objects.

- Python Framework: Flask.

  - ➢ **Flask** is a micro web framework written in Python. It is classified as a micro framework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions.

- Dataset: Movielens.
  - ➢ **MovieLens** is a web-based recommender system and virtual community that recommends movies for its users to watch, based on their film preferences using collaborative filtering of members' movie ratings and movie reviews. It contains about 11 million ratings for about 8500 movies.
- Cloud Host: Heroku.
  - ➢ **Heroku** is a cloud platform as a service supporting several programming languages. One of the first cloud platforms, Heroku has been in development since June 2007, when it supported only the Ruby programming language, but now supports Java, Node.js, Scala, Clojure, Python, PHP, and Go.

5.2 Hardware Requirements.

- Processor: Intel(R) Core(TM) i3-4030U CPU @ 1.90GHz   1.90 GHz.
- RAM: 4.00 GB.
- System type: 64-bit operating system, x64-based processor.

6. CODE

-------------------->>>>>>>>>>>>>>>>>>>>app.py

```python
from flask import Flask, render_template, request
import pandas as pd
import numpy as np
from flask_table import Table, Col

#building flask table for showing recommendation results
class Results(Table):
    id = Col('Id', show=False)
    title = Col('Recommendation List')


app = Flask(__name__)

#Welcome Page
@app.route("/")
def welcome():
    return render_template('welcome.html')

#Rating Page
@app.route("/rating", methods=["GET", "POST"])
def rating():
    if request.method=="POST":
        return render_template('recommendation.html')
    return render_template('rating.html')

#Results Page
@app.route("/recommendation", methods=["GET", "POST"])
def recommendation():
    if request.method == 'POST':
        #reading the original dataset
        movies = pd.read_csv('movies.csv')

        #separating genres for each movie
        movies = pd.concat([movies, movies.genres.str.get_dummies(sep='|')], axis=1)

        #dropping variables to have a dummy 1-0 matrix of movies and their genres
        ## IMAX is not a genre, it is a specific method of filming a movie, thus removed
```

```
###we do not need movieId for this project
categories = movies.drop(['title', 'genres', 'IMAX', 'movieId'], axis=1)

#initializing user preference list which will contain user ratings
preferences = []

#reading rating values given by user in the front-end
Action = request.form.get('Action')
Adventure = request.form.get('Adventure')
Animation = request.form.get('Animation')
Children = request.form.get('Children')
Comedy = request.form.get('Comedy')
Crime = request.form.get('Crime')
Documentary = request.form.get('Documentary')
Drama = request.form.get('Drama')
Fantasy = request.form.get('Fantasy')
FilmNoir = request.form.get('FilmNoir')
Horror = request.form.get('Horror')
Musical = request.form.get('Musical')
Mystery = request.form.get('Mystery')
Romance = request.form.get('Romance')
SciFi = request.form.get('SciFi')
Thriller = request.form.get('Thriller')
War = request.form.get('War')
Western = request.form.get('Western')

#inserting each rating in a specific position based on the movie-genre matrix
preferences.insert(0, int(Action))
preferences.insert(1,int(Adventure))
preferences.insert(2,int(Animation))
preferences.insert(3,int(Children))
preferences.insert(4,int(Comedy))
preferences.insert(5,int(Crime))
preferences.insert(6,int(Documentary))
preferences.insert(7,int(Drama))
preferences.insert(8,int(Fantasy))
preferences.insert(9,int(FilmNoir))
preferences.insert(10,int(Horror))
preferences.insert(11,int(Musical))
preferences.insert(12,int(Mystery))
```

```
        preferences.insert(13,int(Romance))
        preferences.insert(14,int(SciFi))
        preferences.insert(15,int(War))
        preferences.insert(16,int(Thriller))
        preferences.insert(17,int(Western))

        #This funtion will get each movie score based on user's ratings through dot product
        def get_score(a, b):
            return np.dot(a, b)

        #Generating recommendations based on top score movies
        def recommendations(X, n_recommendations):
            movies['score'] = get_score(categories, preferences)
            return movies.sort_values(by=['score'], ascending=False)['title'][:n_recommendations]

        #printing top-20 recommendations
        output= recommendations(preferences, 20)
        table = Results(output)
        table.border = True
        return render_template('recommendation.html', table=table)

if __name__ == '__main__':
    app.run(debug = True)
```

-------------------->>>>>>>>>>>>>>>>>>>>rating.html

```
<!DOCTYPE html>
    <head>
        <html lang="en" dir="ltr">
        <meta charset="utf-8">
        <title>Rating</title>
        <style type="text/css">
         body
         {
            border: 2px solid black;
            padding: 25px;
            background: url("http://www.baltana.com/files/wallpapers-5/Light-Background-
Wallpaper-16379.jpeg");
            background-repeat: repeat;
```

```html
        background-size: 2000px 2000px;        }
    </style>
  </head>


  <body>
    {% extends "template.html" %}
    {% block content %}
    <h1>Please rate each genre based on your preference </h1>
    <form action="{{ url_for('recommendation') }}" method="POST">
      <div>
      <p> <strong><font size="4" color="black">What is your rating to <font
color="Maroon">Action </font>genre?</strong><br/>Example: Die Hard</p>

        <input type="radio" name="Action" value="0" checked>I hate this genre!<br>
        <input type="radio" name="Action" value="1" > 1<br>
        <input type="radio" name="Action" value="2"> 2<br>
        <input type="radio" name="Action" value="3"> 3<br>
        <input type="radio" name="Action" value="4"> 4<br>
        <input type="radio" name="Action" value="5"> 5
      </div>


      <div>
      <p> <strong><font size="4" color="black">What is your rating to <font
color="Maroon">Adventure </font>genre?</strong><br/>Example: Mission: Impossible</p>

        <input type="radio" name="Adventure" value="0" checked>I hate this genre!<br>
        <input type="radio" name="Adventure" value="1" > 1<br>
        <input type="radio" name="Adventure" value="2"> 2<br>
        <input type="radio" name="Adventure" value="3"> 3<br>
        <input type="radio" name="Adventure" value="4"> 4<br>
        <input type="radio" name="Adventure" value="5"> 5
      </div>


      <div>
      <p> <strong><font size="4" color="black">What is your rating to <font
color="Maroon">Animation </font>genre?</strong><br/>Example: Toy Story </p>

        <input type="radio" name="Animation" value="0" checked>I hate this genre!<br>
        <input type="radio" name="Animation" value="1" > 1<br>
        <input type="radio" name="Animation" value="2"> 2<br>
```

```
      <input type="radio" name="Animation" value="3"> 3<br>
      <input type="radio" name="Animation" value="4"> 4<br>
      <input type="radio" name="Animation" value="5"> 5
   </div>


   <div>
   <p> <strong><font size="4" color="black">What is your rating to <font
color="Maroon">Children's </font>genre?</strong><br/>Example: Alice in Wonderland </p>


      <input type="radio" name="Children" value="0" checked>I hate this genre!<br>
      <input type="radio" name="Children" value="1" > 1<br>
      <input type="radio" name="Children" value="2"> 2<br>
      <input type="radio" name="Children" value="3"> 3<br>
      <input type="radio" name="Children" value="4"> 4<br>
      <input type="radio" name="Children" value="5"> 5
   </div>


   <div>
    <p> <strong><font size="4" color="black">What is your rating to <font
color="Maroon">Comedy </font>genre?</strong><br/>Example: Dumb and Dumber </p>


      <input type="radio" name="Comedy" value="0" checked>I hate this genre!<br>
      <input type="radio" name="Comedy" value="1" > 1<br>
      <input type="radio" name="Comedy" value="2"> 2<br>
      <input type="radio" name="Comedy" value="3"> 3<br>
      <input type="radio" name="Comedy" value="4"> 4<br>
      <input type="radio" name="Comedy" value="5"> 5
   </div>


   <div>
   <p> <strong><font size="4" color="black">What is your rating to <font
color="Maroon">Crime </font>genre?</strong><br/>Example: Goodfellas </p>

      <input type="radio" name="Crime" value="0" checked>I hate this genre!<br>
      <input type="radio" name="Crime" value="1" > 1<br>
      <input type="radio" name="Crime" value="2"> 2<br>
      <input type="radio" name="Crime" value="3"> 3<br>
      <input type="radio" name="Crime" value="4"> 4<br>
      <input type="radio" name="Crime" value="5"> 5
```

```
</div>

<div>
<p> <strong><font size="4" color="black">What is your rating to <font
color="Maroon">Documentary </font>genre?</strong><br/>Example: The Decline of Western
Civilizations </p>

    <input type="radio" name="Documentary" value="0" checked>I hate this genre!<br>
    <input type="radio" name="Documentary" value="1" > 1<br>
    <input type="radio" name="Documentary" value="2"> 2<br>
    <input type="radio" name="Documentary" value="3"> 3<br>
    <input type="radio" name="Documentary" value="4"> 4<br>
    <input type="radio" name="Documentary" value="5"> 5
</div>

<div>
<p> <strong><font size="4" color="black">What is your rating to <font
color="Maroon">Drama </font>genre?</strong><br/>Example: The Godfather </p>

    <input type="radio" name="Drama" value="0" checked>I hate this genre!<br>
    <input type="radio" name="Drama" value="1" > 1<br>
    <input type="radio" name="Drama" value="2"> 2<br>
    <input type="radio" name="Drama" value="3"> 3<br>
    <input type="radio" name="Drama" value="4"> 4<br>
    <input type="radio" name="Drama" value="5"> 5
</div>

<div>
<p> <strong><font size="4" color="black">What is your rating to <font
color="Maroon">Fantasy </font>genre?</strong><br/>Example: Harry Potter </p>

    <input type="radio" name="Fantasy" value="0" checked>I hate this genre!<br>
    <input type="radio" name="Fantasy" value="1" > 1<br>
    <input type="radio" name="Fantasy" value="2"> 2<br>
    <input type="radio" name="Fantasy" value="3"> 3<br>
    <input type="radio" name="Fantasy" value="4"> 4<br>
    <input type="radio" name="Fantasy" value="5"> 5
</div>

<div>
```

```
<p><strong><font size="4" color="black"> What is your rating to <font
color="Maroon">Film-Noir </font>genre?</strong><br/>Example: The Big Sleep </p>


      <input type="radio" name="FilmNoir" value="0" checked>I hate this genre!<br>
      <input type="radio" name="FilmNoir" value="1" > 1<br>
      <input type="radio" name="FilmNoir" value="2"> 2<br>
      <input type="radio" name="FilmNoir" value="3"> 3<br>
      <input type="radio" name="FilmNoir" value="4"> 4<br>
      <input type="radio" name="FilmNoir" value="5"> 5
   </div>


   <div>
    <p> <strong><font size="4" color="black">What is your rating to <font
color="Maroon">Horror </font>genre?</strong><br/>Example: The Shining </p>


      <input type="radio" name="Horror" value="0" checked>I hate this genre!<br>
      <input type="radio" name="Horror" value="1" > 1<br>
      <input type="radio" name="Horror" value="2"> 2<br>
      <input type="radio" name="Horror" value="3"> 3<br>
      <input type="radio" name="Horror" value="4"> 4<br>
      <input type="radio" name="Horror" value="5"> 5
   </div>


   <div>
   <p><strong><font size="4" color="black"> What is your rating to <font
color="Maroon">Musical </font>genre?</strong><br/>Example: La La Land </p>


      <input type="radio" name="Musical" value="0" checked>I hate this genre!<br>
      <input type="radio" name="Musical" value="1" > 1<br>
      <input type="radio" name="Musical" value="2"> 2<br>
      <input type="radio" name="Musical" value="3"> 3<br>
      <input type="radio" name="Musical" value="4"> 4<br>
      <input type="radio" name="Musical" value="5"> 5
   </div>


   <div>
    <p><strong><font size="4" color="black"> What is your rating to <font
color="Maroon">Mystery </font>genre?</strong><br/>Example: Memento </p>


      <input type="radio" name="Mystery" value="0" checked>I hate this genre!<br>
```

```
        <input type="radio" name="Mystery" value="1" > 1<br>
        <input type="radio" name="Mystery" value="2"> 2<br>
        <input type="radio" name="Mystery" value="3"> 3<br>
        <input type="radio" name="Mystery" value="4"> 4<br>
        <input type="radio" name="Mystery" value="5"> 5
    </div>


    <div>
    <p> <strong><font size="4" color="black">What is your rating to <font
color="Maroon">Romance </font>genre?</strong><br/>Example: The Notebook </p>


        <input type="radio" name="Romance" value="0" checked>I hate this genre!<br>
        <input type="radio" name="Romance" value="1" > 1<br>
        <input type="radio" name="Romance" value="2"> 2<br>
        <input type="radio" name="Romance" value="3"> 3<br>
        <input type="radio" name="Romance" value="4"> 4<br>
        <input type="radio" name="Romance" value="5"> 5
    </div>
    </div>


    <div>
    <p> <strong><font size="4" color="black">What is your rating to <font
color="Maroon">Sci-Fi </font>genre?</strong><br/>Example: Star Wars </p>


        <input type="radio" name="SciFi" value="0" checked>I hate this genre!<br>
        <input type="radio" name="SciFi" value="1" > 1<br>
        <input type="radio" name="SciFi" value="2"> 2<br>
        <input type="radio" name="SciFi" value="3"> 3<br>
        <input type="radio" name="SciFi" value="4"> 4<br>
        <input type="radio" name="SciFi" value="5"> 5
    </div>


    <div>
    <p> <strong><font size="4" color="black">What is your rating to <font
color="Maroon">War </font>genre?</strong><br/>Example: Braveheart </p>


        <input type="radio" name="War" value="0" checked>I hate this genre!<br>
        <input type="radio" name="War" value="1" > 1<br>
        <input type="radio" name="War" value="2"> 2<br>
        <input type="radio" name="War" value="3"> 3<br>
```

```
        <input type="radio" name="War" value="4"> 4<br>
        <input type="radio" name="War" value="5"> 5
    </div>


    <div>
    <p> <strong><font size="4" color="black">What is your rating to <font
color="Maroon">Thriller </font>genre?</strong><br/>Example: Seven </p>

        <input type="radio" name="Thriller" value="0" checked>I hate this genre!<br>
        <input type="radio" name="Thriller" value="1" > 1<br>
        <input type="radio" name="Thriller" value="2"> 2<br>
        <input type="radio" name="Thriller" value="3"> 3<br>
        <input type="radio" name="Thriller" value="4"> 4<br>
        <input type="radio" name="Thriller" value="5"> 5
    </div>


    <div>
    <p> <strong><font size="4" color="black">What is your rating to <font
color="Maroon">Western </font>genre?</strong><br/>Example: The Good, The Bad and The
Ugly </p>

        <input type="radio" name="Western" value="0" checked>I hate this genre!<br>
        <input type="radio" name="Western" value="1" > 1<br>
        <input type="radio" name="Western" value="2"> 2<br>
        <input type="radio" name="Western" value="3"> 3<br>
        <input type="radio" name="Western" value="4"> 4<br>
        <input type="radio" name="Western" value="5"> 5
    </div>


    <div>
    <h3><font color="midnightblue">If you're sure about these ratings, go ahead and click
this button!</h3>
    </div>
    <button type="submit" class="btn btn-success"><font color="navy"><strong>Get
Recommendations!</strong></button>

    </form>
    {% endblock %}
  </body>
</html>
```

-------------------->>>>>>>>>>>>>>>>>>>>>recommendation.html

```html
<!DOCTYPE html>
  <head>
    <html lang="en" dir="ltr">
    <meta charset="utf-8">
    <title>Recommendations</title>
    <style type="text/css">
     body
      {
       border: 2px solid black;
       padding: 0px;
       background: url("https://toscon14.files.wordpress.com/2014/05/seminar-presenter-pic.jpg");
       background-repeat: no-repeat;
       background-size: 450px 450px;
       background-position: center center;
       }
    </style>
  </head>

  <body bgcolor="Lavender">
    {% extends "template.html" %}
    {% block content %}
      <h1><font color="midnightblue"><strong>Here are the top-20 movies that you'll most probably like:</strong></font></h1>
      <p>{{ table }}</p>
    {% endblock %}
  </body>
</html>
```
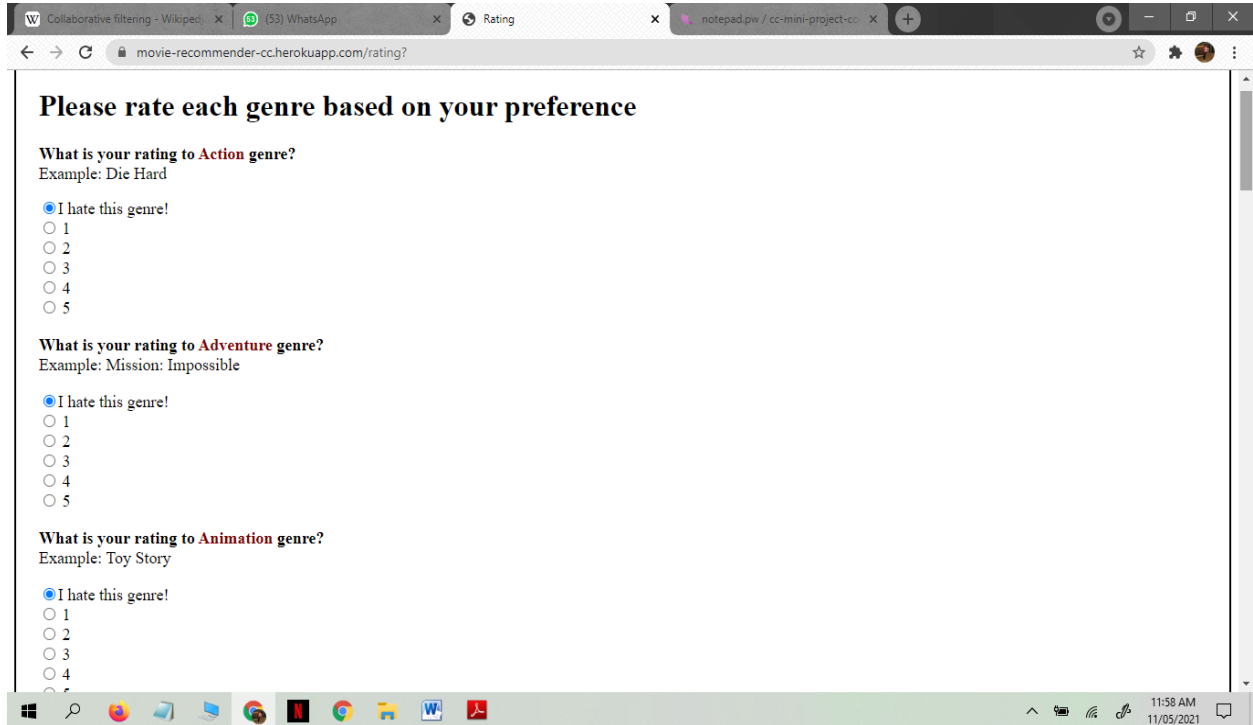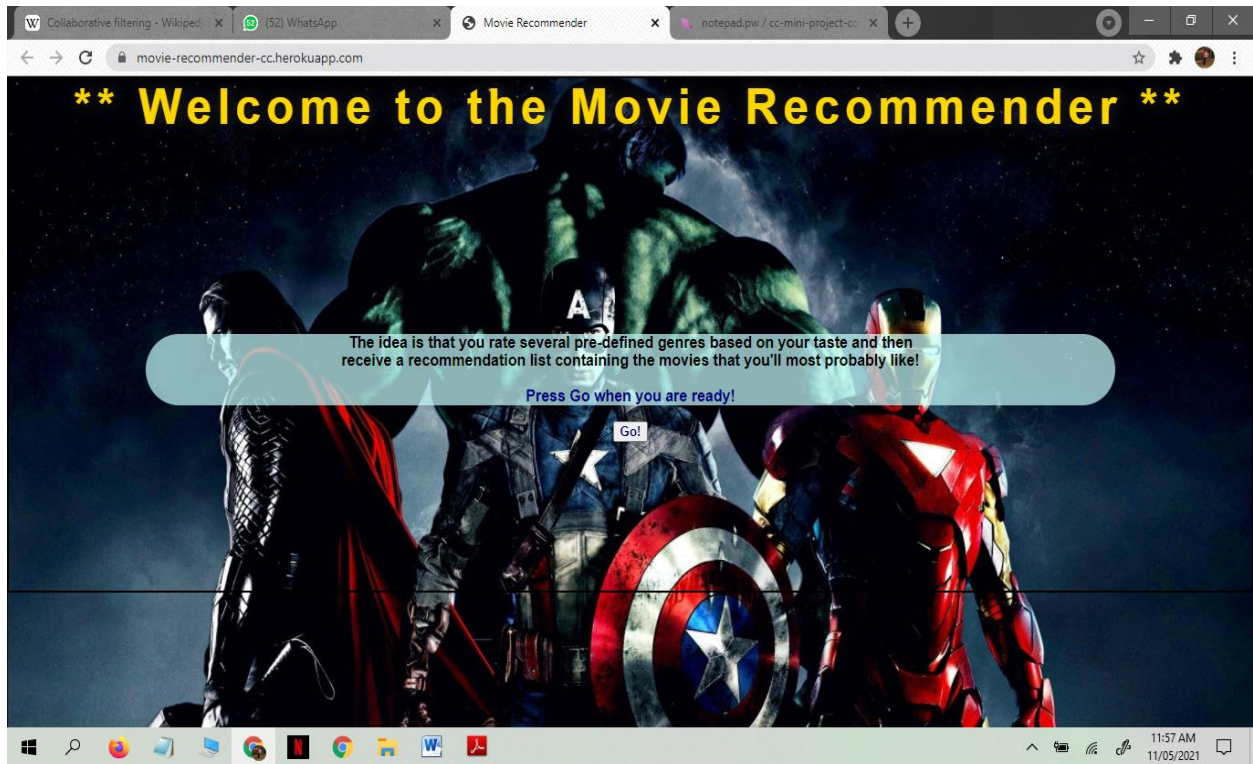
7. RESULT

○ I hate this genre!
○ 1
○ 2
○ 3
○ 4
○ 5

What is your rating to Thriller genre?
Example: Seven

○ I hate this genre!
○ 1
○ 2
● 3
○ 4
○ 5

What is your rating to Western genre?
Example: The Good, The Bad and The Ugly

○ I hate this genre!
○ 1
○ 2
○ 3
○ 4
● 5

If you're sure about these ratings, go ahead and click this button!

[Get Recommendations!]

Here are the top-20 movies that you'll most probably like:

Recommendation List
Shenandoah (1965)
Guns Of Fort Petticoat, The (1957)
Last Of The Mohicans, The (1992)
Australia (2008)
Duck, You Sucker (A.K.A. Fistful Of Dynamite, A) (Giù La Testa) (1971)
I Will Fight No More Forever (1975)
Two Mules For Sister Sara (1970)
100 Rifles (1969)
Alamo, The (2004)
Seraphim Falls (2006)
Legends Of The Fall (1994)
Horse Soldiers, The (1959)
Rio Lobo (1970)
Alamo, The (1960)
They Died With Their Boots On (1941)
Django (1966)
High Noon (1952)
Oklahoma! (1955)
Joe Kidd (1972)
Mclintock! (1963)

8. CONCLUSION

Recommender systems are a powerful new technology for extracting additional value for a business from its user databases. These systems help users find items they want to buy from a business. Recommender systems benefit users by enabling them to find items they like. Conversely, they help the business by generating more sales. Recommender systems are rapidly becoming a crucial tool in E-commerce on the Web. Recommender systems are being stressed by the huge volume of user data in existing corporate databases, and will be stressed even more by the increasing volume of user data available on the Web. New technologies are needed that can dramatically improve the scalability of recommender systems.

Recommender systems open new opportunities of retrieving personalized information on the Internet. It also helps to alleviate the problem of information overload which is a very common phenomenon with information retrieval systems and enables users to have access to products and services which are not readily available to users on the system. We come up with a strategy that focuses on dealing with user's personal interests and based on his previous reviews, movies are recommended to users. This strategy helps in improving accuracy of the recommendations. A personal profile is created for each user, where each user has access to his own history, his likes, ratings, comments, password modification processes. It also helps in collecting authentic data with improved accuracy and makes the system more responsive.

9. REFERENCES

1. Peng, Xiao, Shao Liangshan, and Li Xiuran. "Improved Collaborative Filtering Algorithm in the Research and Application of Personalized Movie Recommendations", 2013 Fourth International Conference on Intelligent Systems Design and Engineering Applications, 2013.

2. Munoz-Organero, Mario, Gustavo A. Ramíez-González, Pedro J. Munoz-Merino, and Carlos Delgado Kloos. "A Collaborative Recommender System Based on Space-Time Similarities", IEEE Pervasive Computing, 2010.

3. Al-Shamri, M.Y.H.. "Fuzzy-genetic approach to recommender systems based on a novel hybrid user model", Expert Systems With Applications, 200810.

4. Hu Jinming. "Application and research of collaborative filtering in e-commerce recommendation system", 2010 3rd International Conference on Computer Science and Information Technology, 07/2010.

5. Xu, Qingzhen Wu, Jiayong Chen, Qiang. "A novel mobile personalized recommended method based on money flow model for stock exchange.(Researc", Mathematical Problems in Engineering, Annual 2014 Issue.

6. Yan, Bo, and Guanling Chen. "AppJoy : personalized mobile application discovery", Proceedings of the 9th international conference on Mobile systems applications and service MobiSys 11 MobiSys 11, 2011.

7. Davidsson C, Moritz S. Utilizing implicit feedback and context to recommend mobile applications from first use.In: Proc. of the Ca RR 2011. New York: ACM Press, 2011. 19-22.http://dl.acm.org/citation.cfm?id=1961639[doi:10.1145/1961634.1961639].

8. Bilge, A., Kaleli, C., Yakut, I., Gunes, I., Polat, H.: A survey of privacy-preserving collaborative filtering schemes. Int. J. Softw. Eng. Knowl. Eng. 23(08), 1085–1108 (2013)CrossRefGoogle Scholar.

9. Calandrino, J.A., Kilzer, A., Narayanan, A., Felten, E.W., Shmatikov, V.: You might also like: privacy risks of collaborative filtering. In: Proceedings of the IEEE Symposium on Security and Privacy, pp. 231–246, Oakland, CA, USA (2011).

10. Research.ijcaonline.org.