MOVIE RECOMMENDATION SYSTEM

BY VRISHANK WARRIER & VEDIKA SURVE

roll no. 16014022096 and 16014022084

Table of Contents :

Introduction :

Users frequently struggle to find movies that match their preferences among a vast array of options. To address this challenge, we've developed an expert system for personalized movie recommendations. Our system saves time and minimizes frustration by offering accurate suggestions tailored to individual tastes.

Introduction :

PROBLEM STATEMENT:

Develop an intelligent movie recommendation system that offers personalized movie suggestions based on user preferences such as genre and favorite cast members. The system should provide accurate and enjoyable recommendations, enhancing the overall movie-watching experience for users.

Introduction :

OBJECTIVE:

The objective of this project is to design and implement an intelligent movie recommendation system that provides personalized movie suggestions to users based on their genre preferences, favorite actors, and similar movies they enjoy.

PEAS :

PERFORMANCE

- generate personalized movie recommendations based on user input
- execution time
- accuracy of recommendations
- responsiveness

ENVIRONMENT

- VSCode and Python,
- laptop with adequate processing power and memory.
- fuzzywuzzy module.

ACTUATORS

- monitor
- console

SENSORS

- keyboard
- mouse

Expert Systems :

REPRESENTATION :

1. Rule-based representation
2. Fuzzy logic

Expert Systems :

FEATURES :

1. Knowledge Base
2. Inference Engine
3. User Interface
4. Explanation Facility
5. Learning and Update

Features :

KNOWLEDGE BASE :

- Data Structures Used:
  - List
  - Dictionary

Features :

KNOWLEDGE BASE :

- Data Structures Used:
  - List
  - Dictionary
- Categories :
  - Hollywood Movies

code snippet for Hollywood:

```
hollywood = [
    {
        "title": "the shawshank redemption",
        "genres": ["drama"],
        "cast": ["tim robbins", "morgan freeman"]
    },
    {
        "title": "the godfather",
        "genres": ["drama", "crime"],
        "cast": ["marlon brando", "al pacino"]
    },
```

Features :

KNOWLEDGE BASE :

- Data Structures Used:
  - List
  - Dictionary
- Categories :
  - Hollywood Movies
  - Bollywood Movies

code snippet for Bollywood:

```
bollywood = [
    {
        "title": "dilwale dulhania le jayenge",
        "genres": ["romance", "drama"],
        "cast": ["shah rukh khan", "kajol"],
    },
    {
        "title": "3 idiots",
        "genres": ["comedy", "drama"],
        "cast": ["aamir khan", "kareena kapoor", "r. madhavan"],
    },
```

Features :

KNOWLEDGE BASE :

- Data Structures Used:
  - List
  - Dictionary
- Categories :
  - Hollywood Movies
  - Bollywood Movies
  - Anime

code snippet for Anime:

```
animes = [
    {
        "title": "your name",
        "genres": ["romance", "drama", "fantasy"],
        "cast": ["ryunosuke kamiki", "mone kamishiraishi"]
    },
    {
        "title": "attack on titan",
        "genres": ["action", "drama", "fantasy"],
        "cast": ["yuki kaji", "yui ishikawa"]
    },
```

Features :

INFERENCE ENGINE :

- Functions of the Inference Engine
- Utility Functions
- Integration with User Interface

Features :

INFERENCE ENGINE :

- Functions of the Inference Engine
  - Recommend by Genre

```
if sm == "any":
    recommendations = [movie["title"] for movie in knowledegebase if any(genre in preferred_genres for genre in movie["genres"])]
elif sm == "all" :
    recommendations = [movie["title"] for movie in knowledegebase if all(genre in preferred_genres for genre in movie["genres"])]
```

Features :

INFERENCE ENGINE :

- Functions of the Inference Engine
  - Recommend by Cast

```
recommendations = [movie["title"] for movie in knowledegebase if any(actor in preferred_cast for actor in movie["cast"])]
```

Features :

INFERENCE ENGINE :

- Functions of the Inference Engine
  - Recommend by Genre and Cast

```
recommendations = [movie["title"] for movie in knowledegebase if
                   all(genre in preferred_genres for genre in movie["genres"]) and any(actor in preferred_cast for actor in movie["cast"])]
```

Features :

INFERENCE ENGINE :

- Functions of the Inference Engine
  - Recommend by Similar Movie Title

```
choice = input ("Choose recommend by\n1. By genre.\n2. By cast.\n3. By genre and cast.\nEnter 1, 2 or 3: ")

if choice == '1':
    preferred_genres = [genre for movie in knowledgebase if movie["title"] == movie_title for genre in movie["genres"]]
    recommend_by_genre(knowledgebase, preferred_genres, "all")

elif choice == '2':
    preferred_cast = [cast for movie in knowledgebase if movie["title"] == movie_title for cast in movie["cast"]]
    recommend_by_cast(knowledgebase, preferred_cast)

elif choice == '3':
    preferred_genres = [genre for movie in knowledgebase if movie["title"] == movie_title for genre in movie["genres"]]
    preferred_cast = [cast for movie in knowledgebase if movie["title"] == movie_title for cast in movie["cast"]]
    recommend_by_genre_and_cast(knowledgebase, preferred_genres, preferred_cast)
```

Features :

INFERENCE ENGINE :

- UTILITY FUNCTIONS
  - Spelling Checker for Genre and Cast

```
for word in words_to_check:
    if word in main_list:
        correct_list.append(word)
    else:
        correct_word, similarity_ratio = process.extractOne(word, main_list)
        if similarity_ratio >= 70:
            correct_list.append(correct_word)
        else:
            incorrect_list.append(correct_word)
```

Features :

INFERENCE ENGINE :

- UTILITY FUNCTIONS
  - Spelling Checker for Movie Title

```
if word_to_check in main_list:
    return word_to_check

else:
    correct_word, similarity_ratio = process.extractOne(word_to_check, main_list)
    if similarity_ratio >= 70 :
        return correct_word
    else:
        print(f"This {word_to_check} does not exist in our knowledge base. Please try again with correct spelling.")
        update_knowledgebase(knowledgebase, word_to_check.lower())
        return []
```

Features :

INFERENCE ENGINE :

- UTILITY FUNCTIONS
  - Update knowledgebase

```
choice = input(f"Would you like to add details for {movie_title}? (yes or no) ")
if choice.lower() == "yes":
    new_movie_genres = input(f"Enter the genres for the new movie '{movie_title}': ").split(",")
    new_movie_cast = input(f"Enter the cast for the new movie '{movie_title}': ").split(",")

    new_movie_genres = convert_into_lower(new_movie_genres)
    new_movie_cast = convert_into_lower(new_movie_cast)

    new_movie = {
        "title": movie_title,
        "genres": new_movie_genres,
        "cast": new_movie_cast
    }
    knowledgebase.append(new_movie)
```

Features :

INFERENCE ENGINE :

- UTILITY FUNCTIONS
  - list of genre, cast and movie title

```
for movie in knowledgebase:
    for item in movie["genres"]:
        if item not in main_list:
            main_list.append(item)
```

```
for movie in knowledgebase:
    for item in movie["cast"]:
        if item not in main_list:
            main_list.append(item)
```

```
main_list = [movie["title"] for movie in knowledgebase]
```

Features :

USER INTERFACE :

- User Interaction Flow
  - Knowledge Base selection
  - Input preference
  - Recommendation type selection
  - View Recommendation
  - Update Preferences
  - Error Handling

Code:

- Knowledge base :

knowledgebase.py

- Inference engine :

inference_engine.py

- Utility function :

utility.py

- User Interface :

user_interface.py

- Google collab link :

https://colab.research.google.com/drive/1lgQoiI6cFf2wFYVZoZ2Gzo1wkaxM5ORV?usp=sharing

Usage:

```
Choose your Film Industry type:
1. Hollywood
2. Bollywood.
3. Anime.
4. Exit.
Enter 1, 2, 3 or 4: 1
Choose your recommendation type:
1. By genre
2. By cast
3. Similar genre and cast
4. Similar movies/animes.
Enter 1, 2,3 or 4: 1
LIST OF GENRES : ['drama', 'crime', 'action', 'adventure', 'sci-fi', 'thriller', 'comedy', 'romance', 'fantasy', 'music']
Enter your preferred genre , separated by commas: drma, crme

Recommended movies based on your preferred genres ['drama', 'crime']:
the shawshank redemption
the godfather
the dark knight
interstellar
joker
love and other drugs
crazy stupid love
about time
passengers
the vow
forever my girl
```

Usage:

```
Choose your Film Industry type:
1. Hollywood
2. Bollywood.
3. Anime.
4. Exit.
Enter 1, 2, 3 or 4: 2
Choose your recommendation type:
1. By genre
2. By cast
3. Similar genre and cast
4. Similar movies/animes.
Enter 1, 2,3 or 4: 2
LIST OF CAST : ['shah rukh khan', 'kajol', 'aamir khan', 'kareena kapoor', 'r. madhavan', 'gracy singh', 'ayushmann khurrana', 'tabu', 'radhika apte', 'sakshi tanwar', 'sagarika
 ghatge', 'amitabh bachchan', 'kangana ranaut', 'ranveer singh', 'alia bhatt', 'deepika padukone', 'shahid kapoor', 'salman khan', 'sonakshi sinha']
Enter your preferred actor , separated by commas: amir khaan

Recommended movies based on your preferred cast ['aamir khan']:
3 idiots
lagaan
dangal
```

Usage:

```
Choose your Film Industry type:
1. Hollywood
2. Bollywood.
3. Anime.
4. Exit.
Enter 1, 2, 3 or 4: 2
Choose your recommendation type:
1. By genre
2. By cast
3. Similar genre and cast
4. Similar movies/animes.
Enter 1, 2,3 or 4: 4
LIST OF MOVIES : ['dilwale dulhania le jayenge', '3 idiots', 'lagaan', 'andhadhun', 'dangal', 'chak de! india', 'kabhi khushi kabhie gham', 'queen', 'gully boy', 'padmaavat',
ajrangi bhaijaan', 'dabangg']
Enter the title of a movie: lagaaan
Choose recommend by
1. By genre.
2. By cast.
3. By genre and cast.
Enter 1, 2 or 3: 1

Recommended movies based on your preferred genres ['drama', 'sport']:
lagaan
chak de! india
```

Usage:

```
Choose your Film Industry type:
1. Hollywood
2. Bollywood.
3. Anime.
4. Exit.
Enter 1, 2, 3 or 4: 1
Choose your recommendation type:
1. By genre
2. By cast
3. Similar genre and cast
4. Similar movies/animes.
Enter 1, 2,3 or 4: 4
LIST OF MOVIES : ['the shawshank redemption', 'the godfather', 'the dark knight', 'interstellar', 'joker', 'the ugly truth', 'love and other drugs', 'crazy stupid love', 'about
time', 'tenet', 'passengers', 'the vow', 'forever my girl', 'no strings attached', 'the hating game', 'a nice girl like you']
Enter the title of a movie: one piece
This one piece does not exist in our knowledge base. Please try again with correct spelling.
Would you like to add details for one piece? (yes or no) yes
Enter the genres for the new movie 'one piece': action
Enter the cast for the new movie 'one piece': emily rudd, inaki godoy

'one piece' added to the knowledge base.
```

Usage:

```
Choose your Film Industry type:
1. Hollywood
2. Bollywood.
3. Anime.
4. Exit.
Enter 1, 2, 3 or 4: 1
Choose your recommendation type:
1. By genre
2. By cast
3. Similar genre and cast
4. Similar movies/animes.
Enter 1, 2,3 or 4: 2
LIST OF CAST : ['tim robbins', 'morgan freeman', 'marlon brando', 'al pacino', 'christian bale', 'heath ledger', 'matthew mcconaughey', 'anne hathaway', 'joaquin phoenix', 'robe
rt de niro', 'gerard butler', 'katherine heigl', 'jake gyllenhaal', 'steve carell', 'ryan gosling', 'domhnall gleeson', 'rachel mcadams', 'john david washington', 'robert pattin
son', 'jennifer lawrence', 'chris pratt', 'channing tatum', 'alex roe', 'jessica rothe', 'natalie portman', 'ashton kutcher', 'lucy hale', 'austin stowell', 'leonardo nam', 'emi
ly rudd', ' inaki godoy']
Enter your preferred actor , separated by commas: emily rudd

Recommended movies based on your preferred cast ['emily rudd']:
one piece
```

Usage:

```
Choose your Film Industry type:
1. Hollywood
2. Bollywood.
3. Anime.
4. Exit.
Enter 1, 2, 3 or 4: 1
Choose your recommendation type:
1. By genre
2. By cast
3. Similar genre and cast
4. Similar movies/animes.
Enter 1, 2,3 or 4: 1
LIST OF GENRES : ['drama', 'crime', 'action', 'adventure', 'sci-fi', 'thriller', 'comedy', 'romance', 'fantasy', 'music']
Enter your preferred genre , separated by commas: action

Recommended movies based on your preferred genres ['action']:
the dark knight
tenet
one piece
```

Conclusion :

- Our movie recommendation system delivers personalized suggestions based on user preferences, enhancing the movie-watching experience.
- By allowing users to input their preferences for genres and favorite cast members, the system generates tailored recommendations that match their interests.
- With intuitive interfaces and interactive features, it ensures accurate and enjoyable recommendations, saving time and boosting user satisfaction.

Reference :

RESEARCH

The movie recommendation system was inspired by the article "How to Build a Movie Recommendation System" from Towards Data Science.

LINK

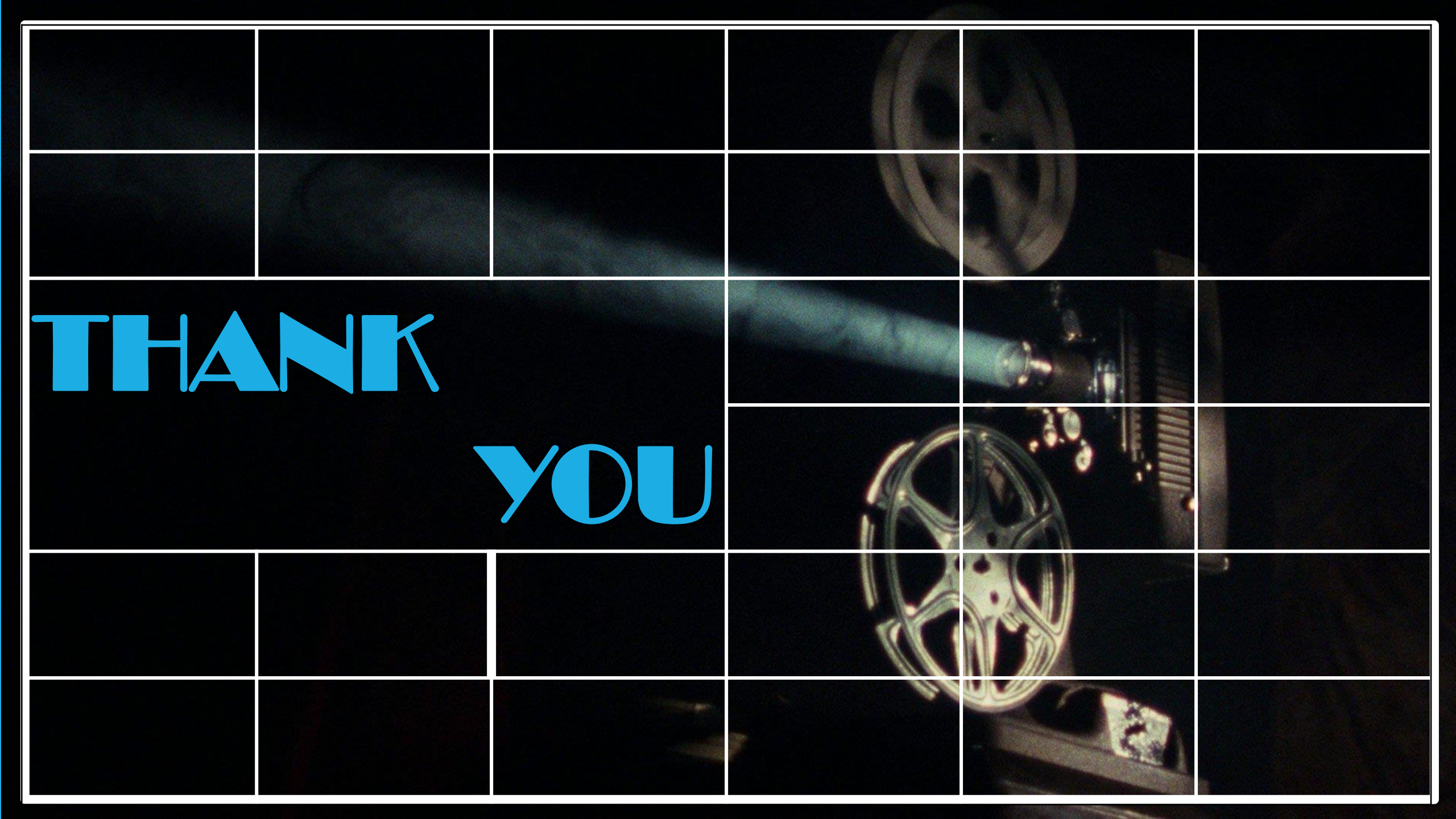
The reference article can be found at the following link: https://towardsdatascience.com/how-to-build-a-movie-recommendation-system-67e321339109

KNOWLEDGE BASE

The knowledge base was taken from imdb website and google search engines

LINK

https://www.imdb.com/


THANK
YOU