Project - Recommender System

Objective

Develop a content-based movie recommender system.

Examples

1. get_movie_recomendation (‘The Dark Knight’)
    The Dark Knight Rises
    Batman Begins
    Batman Returns
    Batman
    Batman Forever

2. get_movie_recomendation (‘The Shawshank Redemption’)
    Prison
    Penitentiary
    1982
    Flying By
    Buffalo '66

3. get_movie_recomendation (‘Frozen’)
    Aladdin
    Spirit: Stallion of the Cimarron
    Pocahontas
    The Legend of Hercules
    The Book of Life

Datasets

1. tmdb_5000_credits.csv
    - movie_id - A unique identifier for each movie
    - title - Title of the movie
    - cast - The name of lead and supporting actors
    - crew - The name of Director, Editor, Composer, Writer etc.

2. tmdb_5000_movies.csv
    - budget - The budget in which the movie was made
    - genres - The genres of the movie, Action, Comedy ,Thriller etc.
    - homepage - A link to the homepage of the movie
    - id - This is infact the movie_id as in the first dataset
    - keywords - The keywords or tags related to the movie
    - original_language - The language in which the movie was made
    - original_title - The title of the movie before translation or adaptation
    - overview - A brief description of the movie
    - popularity - A numeric quantity specifying the movie popularity
    - production_companies - The production house of the movie
    - production_countries - The country in which it was produced
    - release_date - The date on which it was released
    - revenue - The worldwide revenue generated by the movie
    - runtime - The running time of the movie in minutes
    - spoken_languages - The spoken languges in the movie
    - status - "Released" or "Rumored"
    - tagline - Movie's tagline
    - title - Title of the movie
    - vote_average - average ratings the movie recieved
    - vote_count - the count of votes recieved

Requirements

Implement a content-based movie recommender.

1. Given a movie title as input, return the 5 most “relevant” movie titles.
2. From the **movies** dataset:
    a. must use **overview**
    b. must use **genres** - since this is a “packed” field, you will need to create at least 1 new variable from this field
    c. must use **keywords**
3. From the **credits** dataset:
    a. must use **cast** or **crew** - since these are “packed” fields, you will need to create at least 1 new variable from one of these fields
4. Can use any other field(s) from either dataset.
5. Must use at least 4 text mining preprocessing techniques (e.g. tokenization, case, punctuation, stop words, stemming, lemmatization, etc.).
6. Must use at least 1 vectorization technique (e.g. TfidfVectorizer, CountVectorizer, HashingVectorizer, etc.).
7. Must use at least 1 similarity measuring technique (e.g. cosine, euclidean, etc.).
8. The submitted movie recommender must display the output for the 3 movies listed in the examples on page 1.
9. The submitted movie recommender must be implemented with all 8 requirements listed above.

Scoring Rubric

Task	**Points**	**Due Date**
Form a team (4 people)	2	3/3/2023
Submit project proposal (1 paragraph)	2	3/24/2023
Meet with professor to show progress (Jupyter notebook) - All team members must be present to earn points	2	4/21/2023
Project (well-documented Jupyter notebook containing all the project code including all the output cells)	20	5/5/2023
Oral presentation (in class) - All team members must be present to earn points	4	5/5/2023
Total	**30**	

Project Submission Notes

1. Submit a well-documented Jupyter notebook containing all the project code including all the output cells.
    - Each project team must submit 1 Jupyter notebook by email prior to the project due date.
    - Prior to submission, do the following in the Jupyter notebook:
        - *Kernel → Restart and run all*
    - Use *Markdown* formatting to clearly explain code sections. In addition, add comments in each cell for further clarity.
    - After each cell that modifies data, display the first few rows of the resulting dataframe (i.e. *df.head()*).
2. Submitting a basic content-based movie recommendation system that works AND meets the minimum requirements listed above will earn an average grade of 80%.
3. The remaining 20% can be earned through a combination of the following steps:
    - Thorough data handling (i.e. additional preprocessing techniques, handling missing data, etc.)

- Create and use additional innovative features
- Use an additional vectorization technique
  - Show difference in results with the additional vectorization technique used
- Use an additional similarity measuring technique
  - Show difference in results with the additional similarity measuring technique used
- Creativity
  - One possible idea is listed below:
    - Create a weighted rating for each movie and eliminate movies below a certain threshold. Students decide on a weighted rating formula and on the threshold.
  - It is up to the students to think about other creative solutions. Please feel free to discuss ideas with the professor.

References

1. The Netflix Prize (ft. Anne-Marie Kermarrec) - The Netflix prize is the most mythical data science competition in History. It lasted over 2 years, gathered over 20,000 teams and led to major progress in machine learning. This video features Anne-Marie Kermarrec, Director of Research at INRIA, Rennes, scientist of the IC School at EPFL, and CEO of Mediego.
2. From the Labs: Winning the Netflix Prize - AT&T Labs researchers and million dollar Netflix Prize co-winners, Chris Volinsky and Robert Bell, describe their three-year quest to improve the collaborative filtering algorithm Netflix relies on to make millions of movie recommendations.