Final_Report_G27

December 6, 2023

#

Group 27: Movie Recommendation System

##

Author: Yichen Wang, Rui Zhao

0.0.1 1. Topic Summary

Our objective is to develop a movie recommendation system in order to enhance the user's experience by providing movie suggestions. The core functionally of this system is that when user provides us with one specific movie they watched before, we are able to process the movie metadata such as genre, topics, languages, etc. Then we output a result of a comprehensive recommendation list that offers the user a selection of films closely related with their input, such that giving the user a smoother and more tailored viewing experience.

0.0.2 2. Movie Dataset Files Overview

`movies_metadata.csv`

- **Description**: The main Movies Metadata file.
- **Content**: Contains information on 45,000 movies featured in the Full MovieLens dataset.
- **Features**:
  - Posters
  - Backdrops
  - Budget
  - Revenue
  - Release dates
  - Languages
  - Production countries
  - Production companies

`keywords.csv`

- **Description**: Contains the movie plot keywords for MovieLens movies.
- **Format**: Stringified JSON Object.

`credits.csv`

- **Description**: Consists of Cast and Crew Information for all movies.
- **Format**: Stringified JSON Object.

`links.csv`

- **Description**: Contains the TMDB and IMDB IDs of all the movies in the Full MovieLens dataset.

`links_small.csv`

- **Description**: Contains the TMDB and IMDB IDs of a small subset of 9,000 movies from the Full Dataset.

`ratings_small.csv`

- **Description**: A subset of 100,000 ratings from 700 users on 9,000 movies.

0.0.3 3. Goal & Background Info

Our goal is to build a recommendation system that recommends movies to the user based on their preference. Input of the system is a movie and the output is supposed to be a recommendation list consisting similar movies to the given one.

We plan to do this in **two** different approaches as suggested in our proposal.

1. Collaborative Filtering with Deep Learning

- **Techniques**:
  - **User-Based Collaborative Filtering**:
    - User-based collaborative filtering makes recommendations based on user-product interactions in the past. The assumption behind the algorithm is that similar users like similar products.
  - **Item-Based Collaborative Filtering**:
    - Item-based collaborative filtering makes recommendations also based on user-product interactions in the past. The assumption behind the algorithm is that users like similar products and dislike similar products, so they give similar ratings to similar products.

2. Content-Based Filtering with Machine Learning

- **Techniques**:
  - **Natural Language Processing (NLP)**:
    - Utilizes methods like TF-IDF or word embeddings for textual data analysis.
  - **Convolutional Neural Networks (CNNs)**:
    - Employed for analyzing visual content in movie posters.

0.0.4 4. Literature Review of Related Academic Paper

The paper "An Improved Approach for Movie Recommendation System" presents a method already done in our problem domain. It is used to improve the quality of movie recommendation systems. In the main content, it introduces a hybrid approach that combines Content-Based Filtering and Collaborative Filtering, using SVM as a classifier and a genetic algorithm, which are both used in our final project.

Content-Based Filtering based on users' preferences in this context involves analyzing movies and recommending similar ones. Collaborative filtering, on the other hand, relies on the preferences of other users, recommending movies that similar users have liked. And the hybrid approach in the paper aims to blend these two methods effectively.

The similarity of this hybrid approach to traditional collaborative and content-based filtering is its foundational elements. The content-based aspect of the hybrid approach still focuses on the properties of the movies themselves to make recommendations, such as genre, director or actors. Collaborative filtering is represented through the utilization of user ratings and preferences, similar to traditional systems that rely on user behavior and similarities among users. However, by combining these methods, the proposed approach aims to overcome the limitations of each individual method, such as the problem of overspecialization in content-based filtering and the cold start problem in collaborative filtering.

Overall, in the paper, the mentioned hybrid method integrates the personalized nature of content-based filtering with the broad user-based insights of collaborative filtering, potentially leading to more accurate, scalable, and high-quality movie recommendaions.

0.0.5 5. Timeline and Contribution

We first started with a research on how to do the collaborative filtering from both the user-based side and the item-based side. During the research, we learned several ways to calculate the similarities from the user's side and the item's side. However, we noticed that using a scikit library `scikit-surprise` can help us build a recommender system in a more efficient way.

Basically, we separated the works between two different models. Yichen Wang is in charge of the Collaborative filtering algorithm and Rui Zhao is in charge of the Content-based filtering. We both started with our code in the data preprocessing and cleaning stage and then proceed to the actual similarity evaluation process. Finally, we exchanged ideas about how we should display the performance of the model and decided to use different methods.

0.0.6 6. Model Evaluation

[7]:
```
from IPython.display import Image
x1 = Image(filename="result_2000.png")
x2 = Image(filename="result_5000.png")
x3 = Image(filename="result_10000.png")
x4 = Image(filename="user-based.png")
x5 = Image(filename="item-based.png")
```

[8]:
```
x4
```

[8]:
```
predictions_user_based = sim_user.test(testset)
rmse_user_based = accuracy.rmse(predictions_user_based)
```

```
RMSE: 0.9673
```

[9]: x5

[9]:

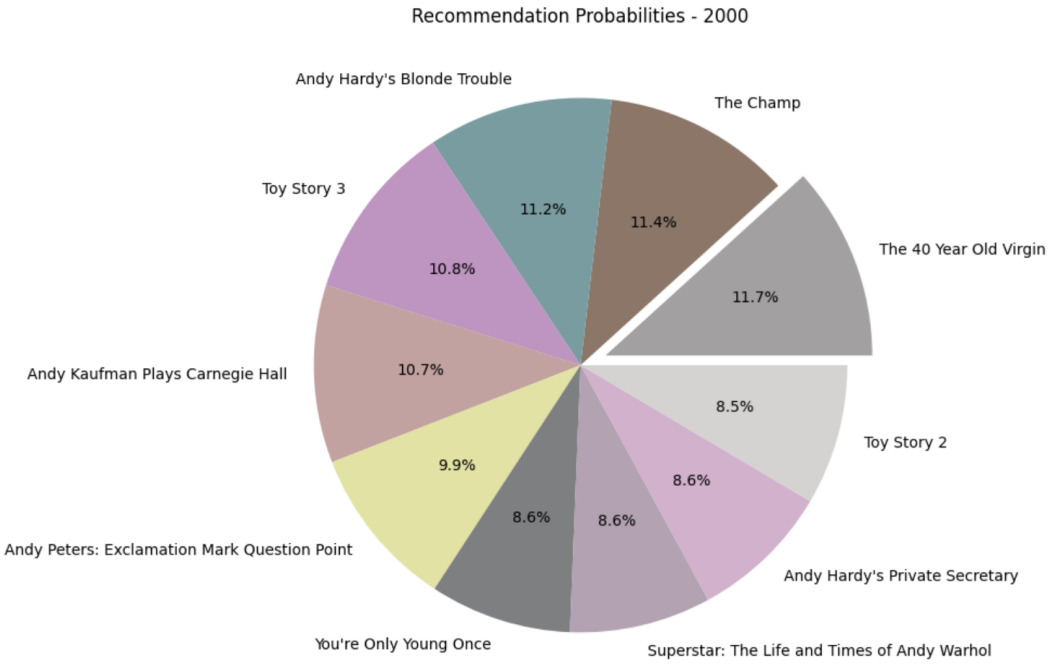
```
predictions_item_based = sim_item.test(testset)
rmse_item_based = accuracy.rmse(predictions_item_based)
```

```
RMSE: 0.9906
```

As we can see from the performance of our collaborative filtering models from the user-based and item-based, the accuracy score of the item-based method is higher than the user-based method. This is predictable since our model is highly dependent on the movie dataset and rating is only a part of the evaluation process to find similarities.

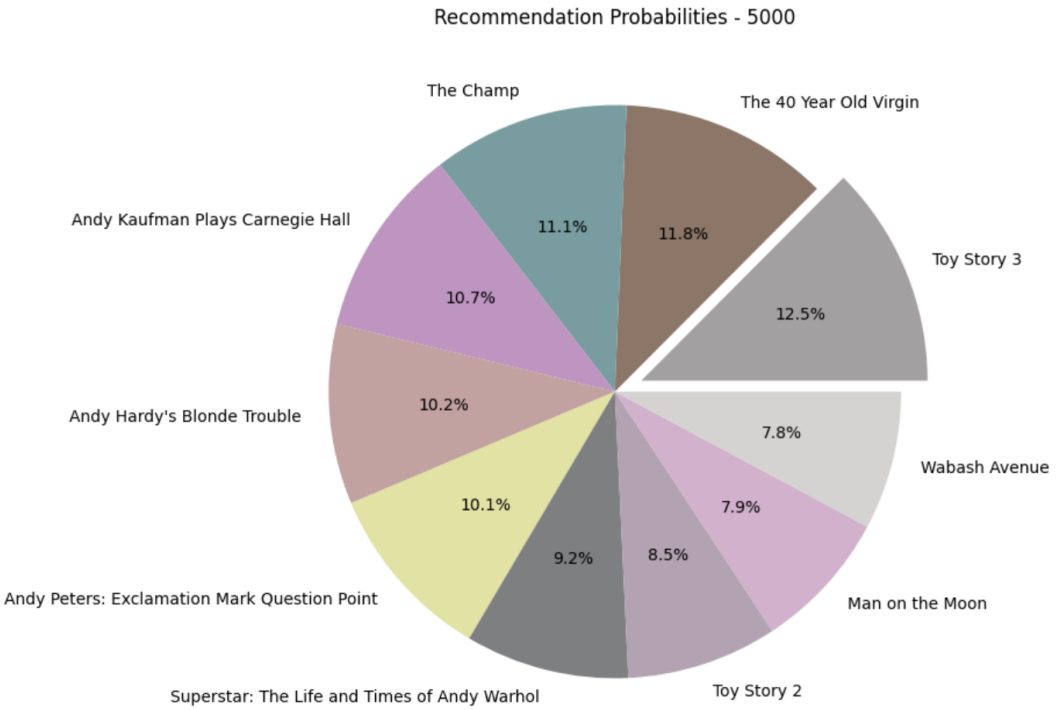
[10]: x1

[10]:



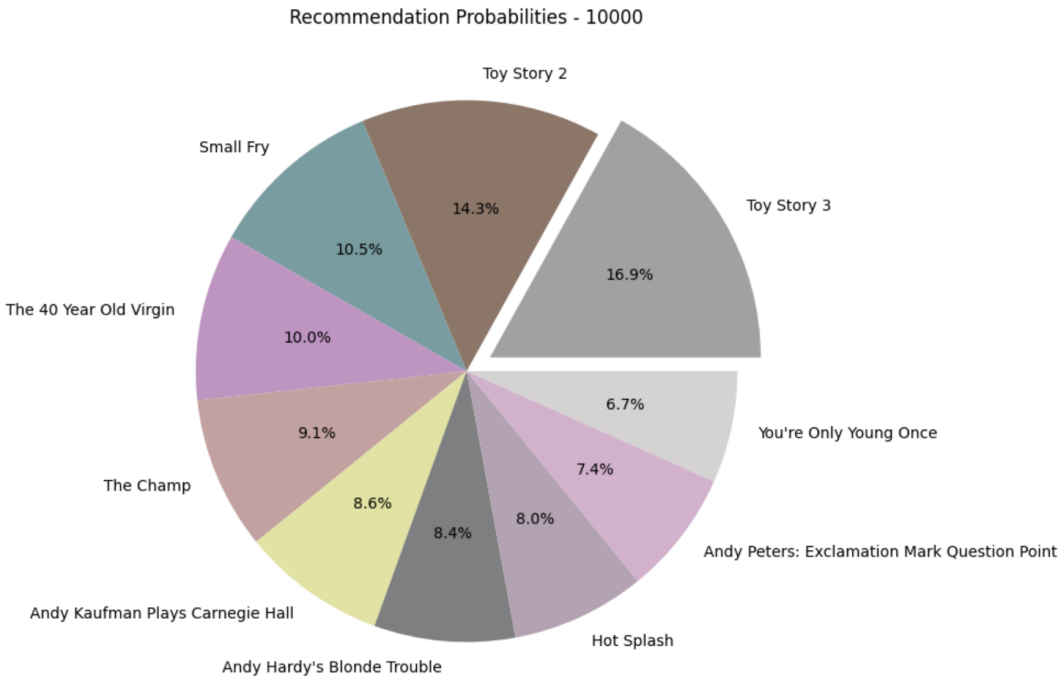
[5]: x2

[5]:


Recommendation Probabilities - 5000
The Champ
The 40 Year Old Virgin
Andy Kaufman Plays Carnegie Hall
Toy Story 3
Andy Hardy's Blonde Trouble
Wabash Avenue
Andy Peters: Exclamation Mark Question Point
Man on the Moon
Superstar: The Life and Times of Andy Warhol
Toy Story 2
11.1%
11.8%
12.5%
10.7%
10.2%
7.8%
10.1%
7.9%
9.2%
8.5%

[6]: x3

[6]:


Recommendation Probabilities - 10000
Toy Story 2
Small Fry
Toy Story 3
The 40 Year Old Virgin
You're Only Young Once
The Champ
Andy Peters: Exclamation Mark Question Point
Andy Kaufman Plays Carnegie Hall
Hot Splash
Andy Hardy's Blonde Trouble
14.3%
10.5%
16.9%
10.0%
6.7%
9.1%
7.4%
8.6%
8.0%
8.4%

In our content-based filtering model, we inputed the movie 'Toy Story' and tried for 3 different parameter tunings, 2000/5000/10000, for the tfidf vectorizer. As we can see from the pie chart which displayed the probabilities of recommending certain movies to the given input, we find that with higher number of features, comes with closer recommendations. In the model when we chosed 2000 features, 'Toy Story 3' comes at fourth and 'Toy Story 2' comes at last. When we chosed 5000, 'Toy Story 3' comes at first and 'Toy Story 2' comes at eighth. When we used the max_features = 10000, we see that the first two recommended movies are the 'Toy Story 3' and 'Toy Story 2'. This makes a lot more sense because when you watched 'Toy Story', it is of high posibility that you may want to watch the continuing series of it, which are 'Toy Story 3' and 'Toy Story 2'. This proved that our model of the content based filtering is a success.

0.0.7 7. Challenges and Limitations

- **Collaborative Filtering:**
  - **Challenges:** The first challenge in every data science/machine learning project is always the data itself. We need to explore the dataset for those features that we can apply and also remove the trivial data points so that useless data won't get in the way of the training process. That's when we did a large amount of data cleaning attempts and finally came with a standard process of how to construct a great and clean dataset ready for training and testing. The second challenge comes when we need to build a reliable model for us to fit the data. We decided to go for svd and knn for their outstanding performance in such tasks of simplify the data and find similar items in a quick manner.
  - **Limitations:** The limitation of this model is that we are more focused on the user's experience on the movies so that we used the userid to predict and find similar movies for our recommendation list. It's not ideal for us to use the model given inputs of movieids.
- **Content-Based Filtering:**
  - **Challenges:** The challenge we faced when dealing with how to build a content-based model was a lot harder than the previous one. We had to do a bit more research than the collaborative filtering and also learn the concepts like TF-IDF and word embedding to decide which one fits better. The parsing of reviews and features also was an important part. We had to do more data cleaning and processing because we were trying to derive 5 features out of the original 24 features.
  - **Limitations:** The limitation of this model that we discovered is that, it's not straigh-forward about how we should proceed to the evaluation metrics since we are doing an unsupervised learning process. However, we were able to deal with this using the max_features parameter inside the tf-idf algorithm.