CS6120 Project - Movie Recommendation

TITLE.

GROUP 24: Xuan Zhang, Yi Chen, Meishan Li, Jia Xu, Qia Lin

DATE: 08-10-2022

Introduction

In recent years more and more people are choosing to watch movies on the Internet. The variety and number of movies are also increasing. People often don't know which one to watch instead of facing so many movies. If there is a good recommendation application for movies, these recommendations will help people to choose a movie of the type they want to watch. We will implement a movie recommendation tool by using unsupervised learning models to group movies, and using cosine similarity to select movies the users want to watch more effectively.

Introduction

1. TMDB 5000 Movies dataset

- Kaggle dataset
- Tmdb 5000 movies
- Tmdb 5000 credits

2. BoW

- Bag of Words model
- Converts a sentence into a vector representation.

3. K-Means

- K-means clustering algorithm
- An iterative clustering analysis algorithm
- Data is divided into k groups by computing
- Cluster centroids & objects assigned to them represent a cluster

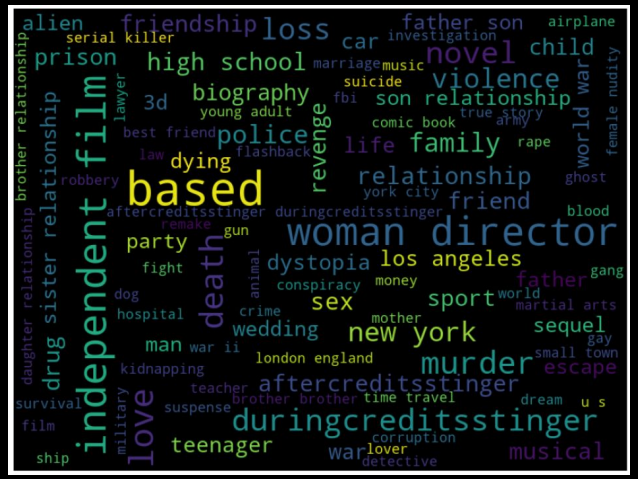
Method

- ➔ Data Pre-processing and Analysis
- ➔ Data Clean and Vectorize the features
- ➔ Cluster Movies
- ➔ Generate Recommendations
- ➔ Evaluation

Method

Data Pre-processing and Analysis

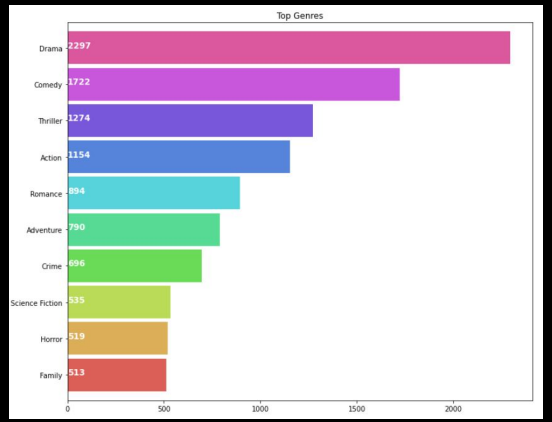
Heat Map



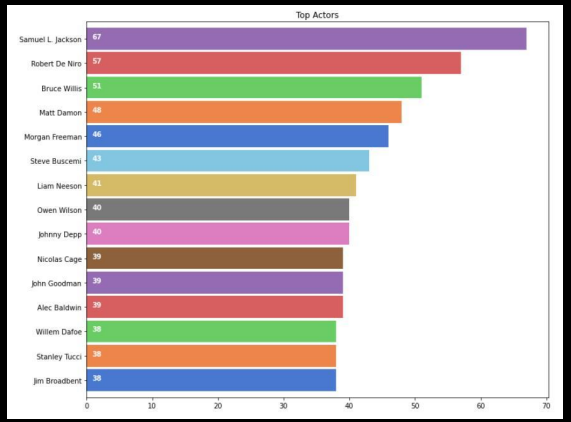
WordCloud

Method

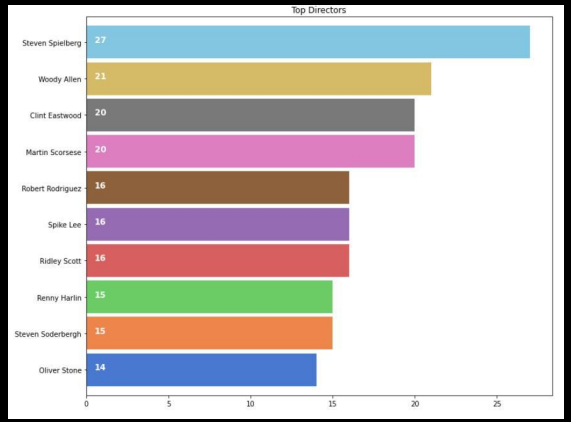
Data Pre-processing and Analysis



Top Genres



Top Actors



Top Directors

Method

Data Clean and Vectorize the features

BoW Vector for Genres

```
0    [1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
1    [1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
2    [1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
3    [1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, ...
4    [1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
Name: genres_bin, dtype: object
```

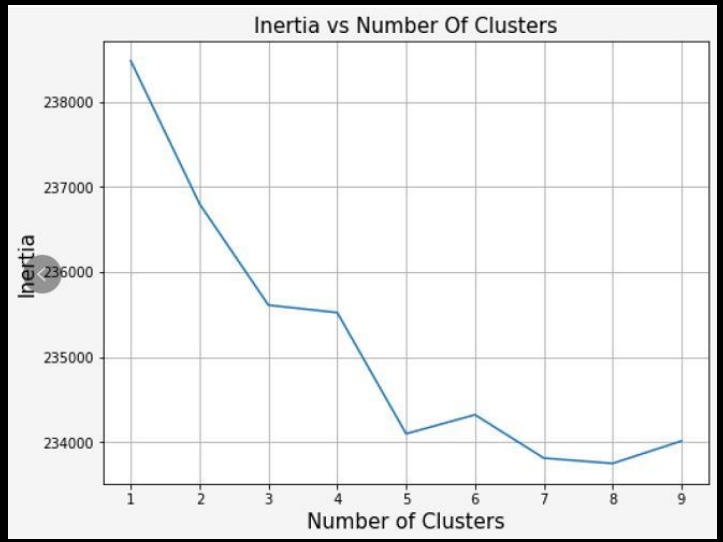
BoW Vector for Cast

```
0    [1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
1    [0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, ...
2    [0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, ...
3    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, ...
4    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
Name: cast_bin, dtype: object
```

Method
Cluster Movies

Our Elbow = 5

$$J(c,\mu) = \sum_{i=1}^{m} \left\| x^{(i)} - \mu_{c^{(i)}} \right\|^2$$



Method

Cluster Movies

CLUSTER 1: Popular Movies: ['Whiplash', 'Fight Club', 'Fury', "One Flew Over the Cuckoo's Nest", 'The Godfather: Part II', 'The Green Mile', 'Cinderella', "We're the Millers", 'The Twilight Saga: Breaking Dawn - Part 2', 'The Wolf of Wall Street']

CLUSTER 2: Popular Movies: ['Mad Max: Fury Road', 'Dawn of the Planet of the Apes', 'The Hunger Games: Mockingjay - Part 1', 'Terminator Genisys', 'The Dark Knight', 'Inception', 'Gone Girl', 'Rise of the Planet of the Apes', 'The Maze Runner', 'Pulp Fiction']

CLUSTER 3: Popular Movies: ['Minions', 'Interstellar', 'Deadpool', 'Guardians of the Galaxy', 'Jurassic World', 'Pirates of the Caribbean: The Curse of the Black Pearl', 'Big Hero 6', 'Captain America: Civil War', 'The Martian', 'Batman v Superman: Dawn of Justice']

CLUSTER 4: Popular Movies: ['Frozen', 'Forrest Gump', 'Twilight', 'Bruce Almighty', 'The Twilight Saga: Eclipse', 'The Twilight Saga: New Moon', 'The Age of Adaline', 'The Fault in Our Stars', 'Amélie', 'Sex Tape']

CLUSTER 5: Popular Movies: ['The Imitation Game', 'The Godfather', 'The Shawshank Redemption', 'Inside Out', "Schindler's List", 'Titanic', 'Fifty Shades of Grey', '12 Years a Slave', 'Blade Runner', 'Psycho']

Method

Generate Recommendations

1. Computing the cosine distance

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}},$$

2. Generate recommendations and Predict score

Method

Generate Recommendations

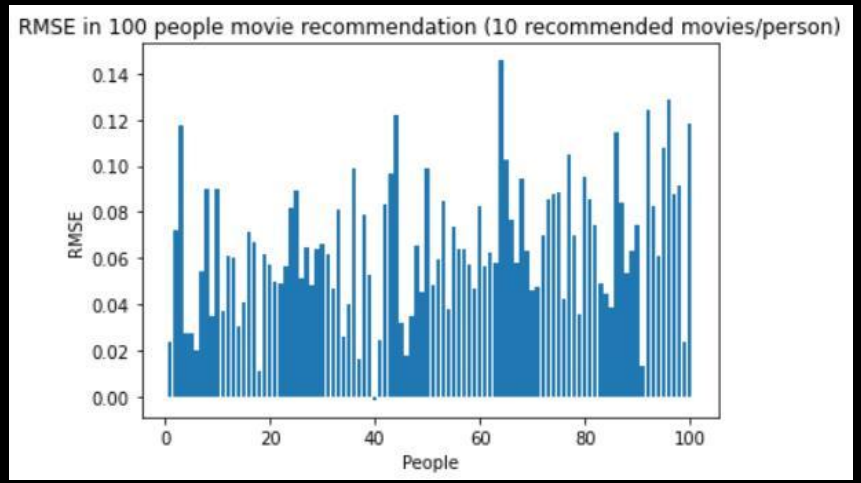
```
Selected Movie:  Catch Me If You Can

Recommended Movies:

Saving Private Ryan | Genres: Drama, History, War | Rating: 7.9
War Horse | Genres: Drama, War | Rating: 7.0
Lincoln | Genres: History, Drama | Rating: 6.7
Close Encounters of the Third Kind | Genres: Science Fiction, Drama | Rating: 7.2
Amistad | Genres: Drama, History, Mystery | Rating: 6.8
Wall Street | Genres: Crime, Drama | Rating: 7.0
The Funeral | Genres: Crime, Drama | Rating: 7.3
American Hustle | Genres: Drama, Crime | Rating: 6.8
The Wolf of Wall Street | Genres: Crime, Drama, Comedy | Rating: 7.9
Capote | Genres: Crime, Drama | Rating: 6.8
```

Evaluation

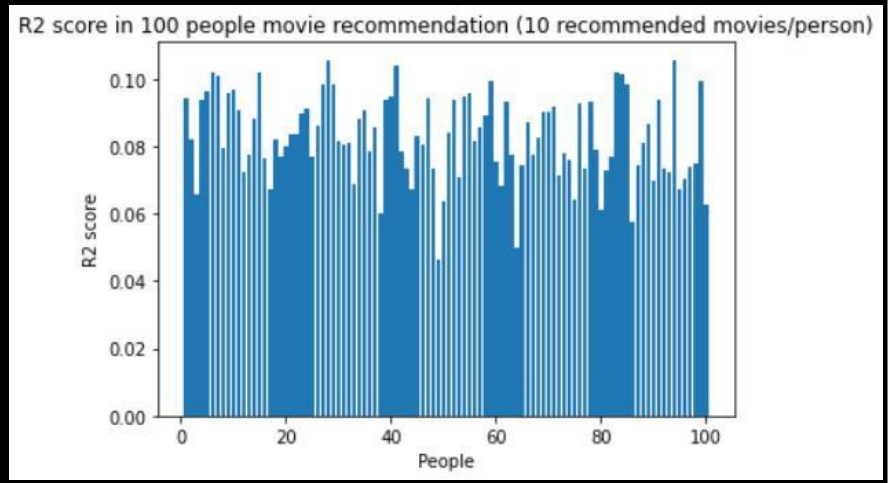
We will calculate the average error between the actual 10 movies the person who watched before and the result of 10 recommended movies predicted in our model.

$$\mathrm{RMSD} = \sqrt{\frac{\sum_{i=1}^{N} (x_i - \hat{x}_i)^2}{N}}$$

Evaluation



RMSE



R2 score

Results

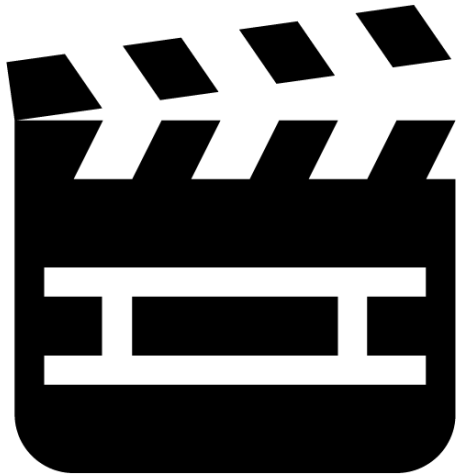
In order to test the accuracy of the system, we randomly selected 100 users to predict 10 films they might like, using the 10 films they had seen as the basis for their recommendations. Then compare the feature vector of the recommendation to the base by RMSE method. The figures from RMSE section easily show that RMSE is below 0.15, and the R2 Score is below 0.1. It indicates that our system provides decent recommendations for users, which are similar to their original preference, in an efficient way.

```
Selected Movie:  Catch Me If You Can

Recommended Movies:

Saving Private Ryan | Genres: Drama, History, War | Rating: 7.9
War Horse | Genres: Drama, War | Rating: 7.0
Lincoln | Genres: History, Drama | Rating: 6.7
Close Encounters of the Third Kind | Genres: Science Fiction, Drama | Rating: 7.2
Amistad | Genres: Drama, History, Mystery | Rating: 6.8
Wall Street | Genres: Crime, Drama | Rating: 7.0
The Funeral | Genres: Crime, Drama | Rating: 7.3
American Hustle | Genres: Drama, Crime | Rating: 6.8
The Wolf of Wall Street | Genres: Crime, Drama, Comedy | Rating: 7.9
Capote | Genres: Crime, Drama | Rating: 6.8
```

Conclusion

We did a good analysis of this dataset to select the important features and vectorize them by using the bag of words model. We made a nice clustering with K-means for large data to reduce the computational complexity and computational time. When making recommendations, we first predict which cluster the referee belongs to, and use cosine as an indicator to recommend movies. Finally, our model achieves good results on RMSE and R-square.

Future Works



We can try to use other ways (like TF-IDF, PMI, and Neural word embedding) to do vectorization and collaborative filtering in this project. The collaborative filtering will be based on the User file and Item file to fill out empty of these two files. We think it will make this model of this project better. In addition, some parts of the movie data we did not use in this project. It is possible to extract useful features from these movie data. And we can use forward feature selection to select good features. It will help this model of this project to do a better recommendation.