PROJECT 3:
RECOMMENDER SYSTEM FOR IMDB 5000+ MOVIES

GROUP 17:

ABISHEK GANESH

MAHMOOD M NAUMANI

SRI KAVYA RAVELLA

Table of Contents

Problem Statement

Recommender systems are used by all the e-commerce sites to recommend items to users, Recommendation systems can be broadly classified into two types (1) content based recommendation systems and (2) collaborative filtering techniques.

In this project, we build a content based recommendation system for IMDB Movies, which recommends similar movies to a user based on the current movie, Content based recommender systems don't need user data to provide recommendations but are based on the similarity of the content between the movies.

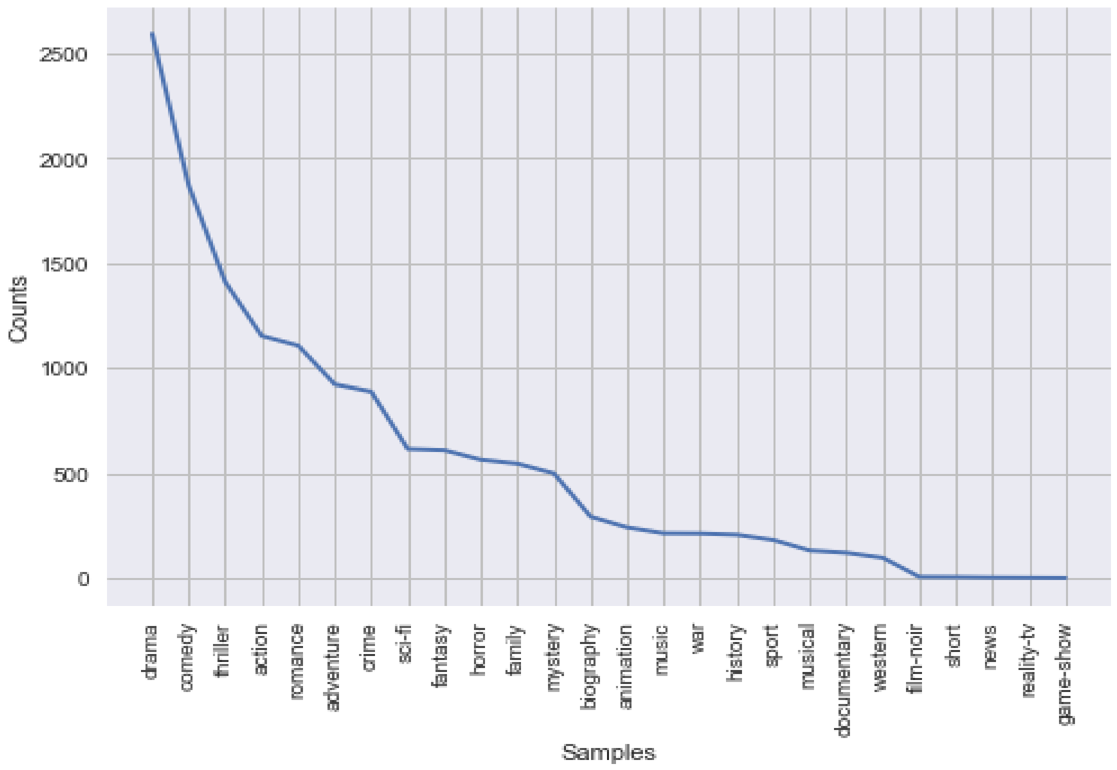
Data Collection and Pre-Processing

We have collected our data from Kaggle, Our Data consists of 5040 records and 28 features. While cleaning of our data we have removed special characters separators and Null values from our data set. We also removed multilingual characters from the data, and standardized the plot_keywords field. After cleaning and preprocessing we have around 4900 records in our dataset.

Data Visualization

We plotted frequency distribution for genres to see which genre has the highest occurrence.

Frequency distribution for Genre



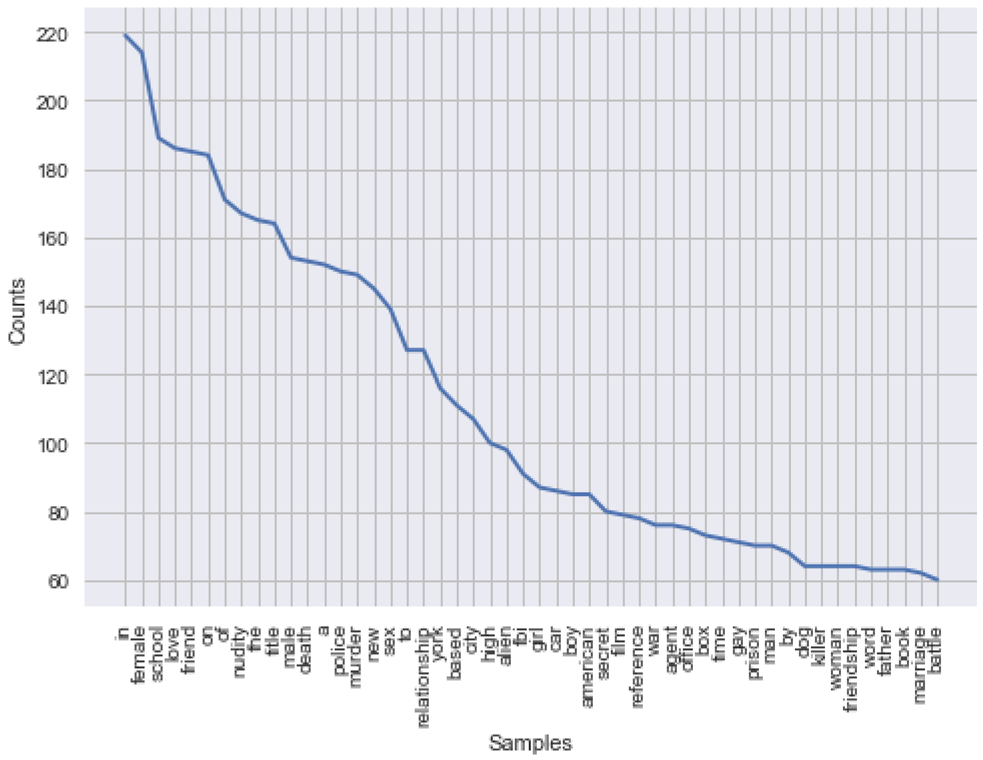
We could see that Drama was the highest occurring genre followed by Comedy, Thriller and Action which were somewhat similar in numbers. The least count were for Documentary, Western, News, Reality-tv and Game-show.

Word Cloud for Genre

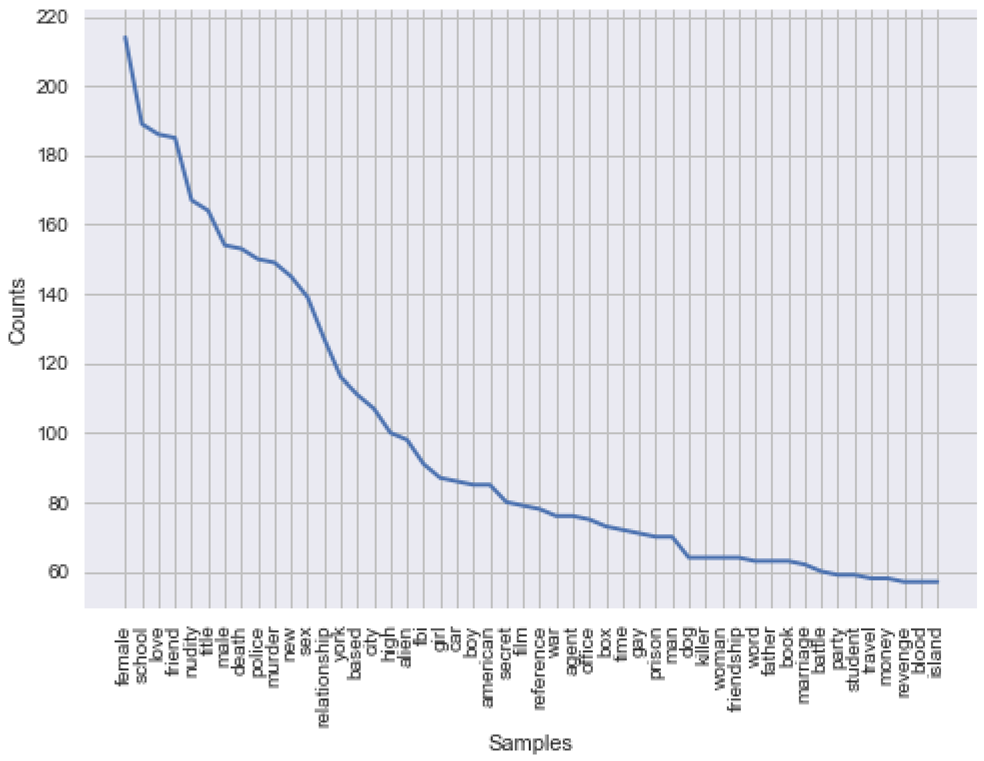
music
sci
noir
horror
show
western
fi
sport
action
film
short
animation
mystery
musical
tv
adventure
news
reality
comedy
crime
thriller
family
drama
fantasy
biography
war
game
romance
documentary
history

We also plotted frequency distribution for plot keywords with stop words and then without the stop words to better understand which key words are occurring highly.

Frequency Distribution for Plot_keywords

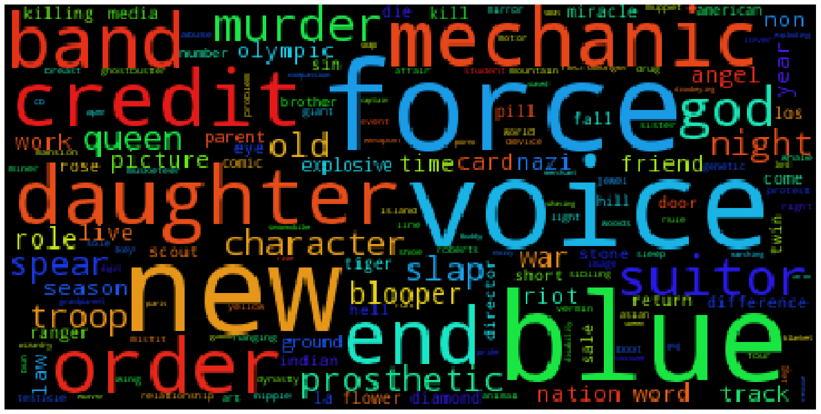


Frequency Distribution for Plot_keywords with stop words removed



The word cloud for Plot_keywords helped us to visualize the most common words used to describe the movie plot , we can see that the words such as force, voice and new were the most common ones followed by mechanic, credit, band and daughter.

Word Cloud for Plot_keywords



Content Based Recommendation

Recommender systems typically produce a list of recommendations in one of two ways – through collaborative and content-based filtering or the personality-based approach.

Collaborative filtering approaches building a model from a user's past behavior (items previously purchased or selected and/or numerical ratings given to those items) as well as similar decisions made by other users. This model is then used to predict items (or ratings for items) that the user may have an interest in. Content-based filtering approaches utilize a series of discrete characteristics of an item in order to recommend additional items with similar properties.

Similarity Based Recommendation system

We created a panda's data frame for Movie_title, Plot_keywords and Genres fields in our data set and created a Matrix based on CountVectorizer for both Genres and Plot_keywords. We have then calculated the similarity scores using Cosine , Jaccard, Euclidean and Manhattan similarities.

Cosine Similarity

Cosine similarity is a measure of similarity between two non-zero vectors of an inner product space that measures the cosine of the angle between them. The cosine of 0° is 1, and it is less than 1 for any other angle. It is thus a judgment of orientation and not magnitude: two vectors with the same orientation have a cosine similarity of 1, two vectors at 90° have a similarity of 0, and two vectors diametrically opposed have a similarity of -1, independent of their magnitude.

Jaccard Similarity

The Jaccard Similarity or coefficient measures similarity between finite sample sets, and is defined as the size of the intersection divided by the size of the union of the sample sets

Euclidean Similarity

Euclidean metric is the "ordinary" straight-line distance between two points in Euclidean space. The Euclidean similarity gives the smallest distance as the closest or the most similar point in Euclidean space.

Manhattan Similarity

The distance between two points in a grid based on a strictly horizontal and/or vertical path (that is, along the grid lines), as opposed to the diagonal or "as the crow flies" distance. The Manhattan distance is the simple sum of the horizontal and vertical components We have used the similarity score to sort our data and recommend the movies with highest similarity scores between them.

We compared the similarity score and recommendation between the list generated from plot_keywords and Genres and could see that the list recommended by plot_keywords is more accurate as there are more words to describe the movie and the plot related to the movie.

Cosine, Euclidean and Manhattan Distances based on Plot

	Name	Plot	cos	euc	man
3	The Dark Knight Rises	deception imprisonment lawlessness police offi...	1	0	0
4794	Funny Ha Ha	Mumblecore	0	8	2.828427

3448	Snow Queen	3d	0	8	2.828427
579	Die Hard 2	airport fuel plane police terrorist	0.338062	8	2.828427
2305	Dwegons and Leprechauns	Dwegons	0	8	2.828427
2304	Ramanujan	Ramanujan	0	8	2.828427
380	The Devil's Own	friendship murder northern ireland police offi...	0.428571	8	2.828427
4476	The Horror Network Vol. 1	Anthology	0	8	2.828427
3072	The Second Best Exotic Marigold Hotel	Sequel	0	8	2.828427
2215	Fired Up	Sitcom	0	8	2.828427

Cosine, Euclidean and Manhattan Distances based on Genres

	Name	genres	cos	jac	euc	man
915	From Paris with Love	action thriller	1	1	0	0
343	A Good Day to Die Hard	action thriller	1	1	0	0
2462	The Prince	action thriller	1	1	0	0
350	Unstoppable	action thriller	1	1	0	0
2002	Red Sky	action thriller	1	1	0	0
2052	Haywire	action thriller	1	1	0	0
4721	Royal Kill	action thriller	1	1	0	0
973	Into the Storm	action thriller	1	1	0	0

Recommendation System based on K Nearest Neighbor

The k-nearest neighbors algorithm (k-NN) is a non-parametric method used for classification and regression. The input consists of the k closest training examples in the feature space. the output is a class membership. An object is classified by a majority vote of its neighbors, with the object being assigned to the class most common among its *k* nearest neighbors

The neighbors are taken from a set of objects for which the class (for *k*-NN classification) or the object property value (for *k*-NN regression) is known. This can be thought of as the training set for the algorithm, though no explicit training step is required.

In this project, we have used KNN- based recommendation with features such as Genres, Movie_plot, gross, num_voted_users, budget and imdb_score together to see how the movies are recommended. The recommendations are based on the following criteria.

Genres, movie_title, plot_keywords, language, budget, title_year, imdb_score, num_user_for_reviews, num_critic_for_reviews, gross

Below are the recommendations for the Movie “Ocean s Thirteen” using KNN- algorithm.

KNN - Recommendations Output

Recommendations
The Italian Job
Ocean s Twelve
Ocean s Eleven
3000 Miles to Graceland
The Ladykillers
Killer Elite
Hannibal Rising
Out of the Furnace
Death Sentence
Taken 2
Road to Perdition
The Killer Inside Me
Fast Five
Takers

Challenges

- The quality of the data from IMDB movies could be better in a sense that the plot_keywords are not very accurate or standardized hence by improving the description of the movie plot we could improve the recommendations. One way to achieve this is to manually scrape data from sites like IMDB.
- This is an Item-Item content based recommendation system which doesn’t make use of any user data, with user data we can recommend different movies based on user preferences.
- The data size affects the performance of the model so ideally this model should be moved to a Big Data environment

Conclusion

Using textual similarity between movies to recommend has a lot of potential as we are looking at the item-item similarity to create this model. But in out attempt to create a Recommender system, we were not able to test the validity of the recommendation generated for the model for this we hoping to do tests on users and see what their opinion is on the recommendations given by the model.

References

- http://scikit-learn.org/stable/modules/generated/sklearn.metrics.jaccard_similarity_score.html
- http://scikit-learn.org/stable/modules/generated/sklearn.metrics.pairwise.euclidean_distances.html
- http://scikit-learn.org/stable/modules/generated/sklearn.metrics.pairwise.cosine_similarity.html
- http://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html
- https://www.youtube.com/watch?v=gCaOa3W9kM0 - 8 Recommender Systems
- https://www.youtube.com/watch?v=TSv6eLAOt78 - Netflix recommendation system
- Dataset: https://www.kaggle.com/deepmatrix/imdb-5000-movie-dataset