Movie Recommendation System

Nake Shrivatsa Aparanji[1], Bangarulakshmi Mahanthi[2]

[1] *Student, Department of Computer Science Engineering, Gitam University, Visakhapatnam.*
[2] *(Guide) Assistant Professor, Department of Computer Science Engineering, Gitam University, Visakhapatnam.*

Abstract

In the last few years, due to the exponential rise in the usage of applications like Amazon Prime, Netflix, YouTube etc., the need for recommendation system has hugely incremented. Nowadays, recommendation systems are frequently used in e-commerce, movie suggestion, online shopping etc.

Recommender systems are nothing but algorithms that are to suggest similar items to users, items could be movies, images, text, products etc. Movie recommendation is extremely crucial in the entertainment industry as it hugely impact the revenue of the industry. If a user is being suggested to watch movies of his/her liking, user tends to use a certain application for a longer time, which ultimately increases the company's revenue.

Sentiment analysis is a technical term for text mining that searches and gets important information from a text source, to help understand the social sentiment behind the text. It is an extremely common text classification tool that studies an input text and responds whether the text conveys a positive or negative meaning.

Introduction

At the backend, an ML model is created using content-based recommendation method. Python packages: scikit learn, pandas, NumPy and seaborn are used to build the model. The model is trained using a dataset from the Kaggle website (IMDB 5000 Movie Dataset), also extracting movies, that were released in 2017 or later, from Wikipedia. The metadata of these movies like cast, director, production company, genre, reviews etc. are extracted from The Move Data Base (TMDB) API. Next the pickling process will be done, where the python object model is converted to a class file. This class file can now be used to predict the incoming movies.

Flask framework will be utilized to make the connection between the HTML webpage and the python class file. Using flask, the input from the webpage is sent to the class file(model), which recommends movies and sends the result back to flask, which in turn returns the result to the webpage as output.

Content-based recommendation system uses metadata like genre, director name etc. to suggest similar movies to the user. For this project, I will be focusing on content-based recommendation system as I thing using metadata can provide a good deal of insight on understanding users preferences and help recommend movies. It helps to get a good gauge of users' likes and dislikes. Also, sentiment analysis is used to build a model to predict whether a review for a movie has a positive or a negative sentiment.

Methodology

Cosine similarity is a metric used to measure how similar the documents are irrespective of their size. Mathematically, it measures the cosine of the angle between two vectors projected in a multi-dimensional space.

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

Naive Bayes algorithm is a simple probabilistic classifier that calculates a set of probabilities by counting the frequency and combination of values in a given dataset. It is based on Bayes' theorem.

$$P(A|B)=\frac{P(B|A)P(A)}{P(B)}$$

Here,

P(A|B) is Posterior probability: Probability of hypothesis A on the observed event B.

P(B|A) is Likelihood probability: Probability of the evidence given that the probability of a hypothesis is true.

P(A) is the probability that A event occurs.

P(B) is the probability that B event occurs.

Term Frequency (TF) is defined as the number of times a word appears in a document divided by the total number of words in the document. Every document has its own frequency. Inverse Data Frequency (IDF) is log of the number of documents divided by the number of documents that contain a certain word 'w'. IDF determines the weight of rare words for all the documents in the corpus.

For example, say there are three movies from which the algorithm has to suggest similar movies – Avatar, Titanic, War. Now, the movie documents or the metadata are evaluated and a matrix is formed which gives us the similarity between movies.

i.e.,

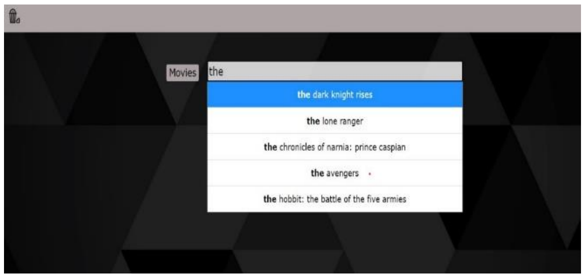
	Avatar	**Titanic**	**War**
Avatar	1	0.6	0.2
Titanic	0.6	1	0.4
War	0.2	0.4	1

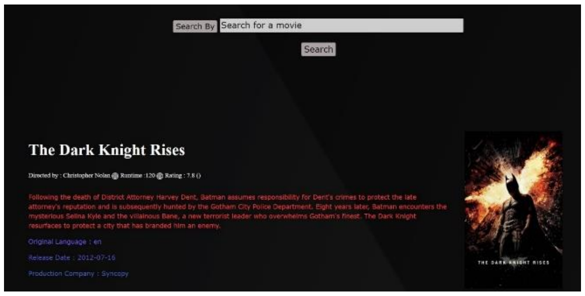
Now, to get movie similar to 'Avatar', the first is sliced and it is searched for second largest data value. Here, it is 0.6 i.e., 'Titanic', so we can conclude that 'Avatar' is similar to 'Titanic' than 'War'.

For the reviews part, the nave bayes algorithm is used to create a model that has the ability to predict whether a given sentence/paragraph tells positive or negative emotion. This model is then used to predict sentiment of the reviews that are extracted from the TMDB API for a particular movie.

Results

The website allows user to search for movies based on movie name, director, actor and production company. On clicking the 'search' button a gif is imbibed which shows that the movie is getting searched from the data set. On finding the movie, it is shown on screen as shown above. The details about the movie are extracted from the TMDB API and is shown on the web page. These details include a brief summary about the movie, director name, production company, a movie poster and the cast. The images of the cast is shown below, and they are in-built buttons. So, on clicking on these images, user gets information about the particular cast member.


Movies
the
the dark knight rises
the lone ranger
the chronicles of narnia: prince caspian
the avengers
the hobbit: the battle of the five armies



The Dark Knight Rises

Directed by : Christopher Nolan Runtime :120 Rating : 7.8 ()

Following the death of District Attorney Harvey Dent, Batman assumes responsibility for Dent's crimes to protect the late attorney's reputation and is subsequently hunted by the Gotham City Police Department. Eight years later, Batman encounters the mysterious Selina Kyle and the villainous Bane, a new terrorist leader who overwhelms Gotham's finest. The Dark Knight resurfaces to protect a city that has branded him an enemy.

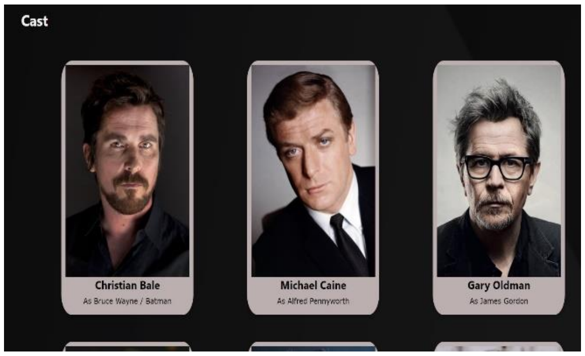
Original Language : en

Release Date : 2012-07-16

Production Company : Syncopy

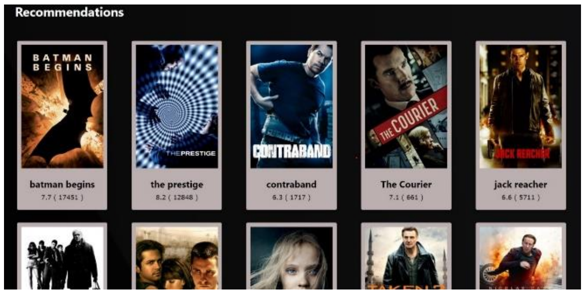
I have extensively used JavaScript while

creating the website. If the user types the key words of a movie, he/she will be given options to complete the movie name. Also, on hovering the mouse over the cast images, user can see that the color of the image changes, and he/she gets details regarding that cast.



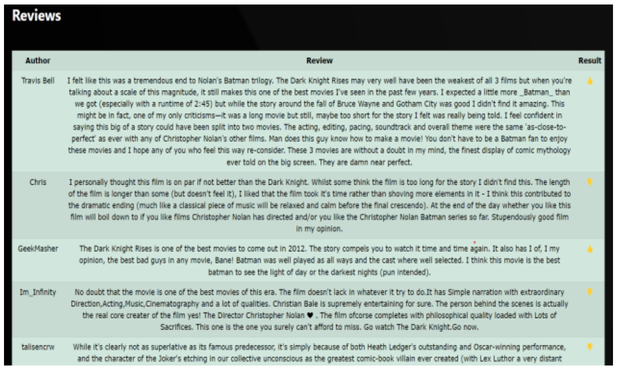
.

After getting a brief about the movie searched by the user, the user gets recommendation. The user will be able to see a number of similar movies to the movie he/she searches for. So, here we can see that when the user searches for 'the dark knight', user is recommended to watch 'batman begins', 'the prestige', 'contraband' etc. Also, these images are in-built buttons and therefore on clicking any of these images, the user is redirected to a page where he/she will get all details regarding that movie along with new recommendation.



It is worth noting that the recommendation is achieved through content-based filtering and hence the user's priorities are not considered. The features of a movie are taken into consideration and any movie that matches in terms of features, will be displayed as a recommended movie.

Well, the second part is to filter the reviews and predict whether they show positive or negative emotions. For that, I have built a Naïve Bayes' model that has been trained to filter the reviews. So, when a movie is searched by the user, using the TMDB API, reviews corresponding to that movie is extracted. These reviews are then fed to the Python model which predicts the polarity. If the emotion is positive, I display an 'OK' emoji and if negative emotion is predicted then I display 'NOT OK' emoji.



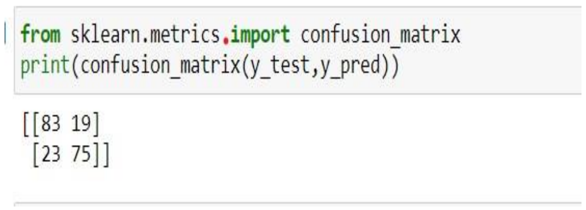
Reviews

Author	Review	Result
Travis Bell	I felt like this was a tremendous end to Nolan's Batman trilogy. The Dark Knight Rises may very well have been the weakest of all 3 films but when you're talking about a scale of this magnitude, it still makes this one of the best movies I've seen in the past few years. I expected a little more _Batman_ than we got (especially with a runtime of 2:45) but while the story around the fall of Bruce Wayne and Gotham City was good I didn't find it amazing. This might be in fact, one of my only criticisms—it was a long movie but still, maybe too short for the story I felt was really being told. I feel confident in saying this big of a story could have been split into two movies. The acting, editing, pacing, soundtrack and overall theme were the same 'as-close-to-perfect' as ever with any of Christopher Nolan's other films. Man does this guy know how to make a movie! You don't have to be a Batman fan to enjoy these movies and I hope any of you who feel this way re-consider. These 3 movies are without a doubt in my mind, the finest display of comic mythology ever told on the big screen. They are damn near perfect.	👍
Chris	I personally thought this film is on par if not better than the Dark Knight. Whilst some think the film is too long for the story I didn't find this. The length of the film is longer than some (but doesn't feel it), I liked that the film took it's time rather than shoving more elements in it - I think this contributed to the dramatic ending (much like a classical piece of music will be relaxed and calm before the final crescendo). At the end of the day whether you like this film will boil down to if you like films Christopher Nolan has directed and/or you like the Christopher Nolan Batman series so far. Stupendously good film in my opinion.	👎
GeekMasher	The Dark Knight Rises is one of the best movies to come out in 2012. The story compels you to watch it time and time again. It also has I of, I my opinion, the best bad guys in any movie, Bane! Batman was well played as all ways and the cast where well selected. I think this movie is the best batman to see the light of day or the darkest nights (pun intended).	👍
Im_Infinity	No doubt that the movie is one of the best movies of this era. The film doesn't lack in whatever it try to do.It has Simple narration with extraordinary Direction,Acting,Music,Cinematography and a lot of qualities. Christian Bale is supremely entertaining for sure. The person behind the scenes is actually the real core creater of the film yes! The Director Christopher Nolan ♥ . The film ofcorse completes with philosophical quality loaded with Lots of Sacrifices. This one is the one you surely can't afford to miss. Go watch The Dark Knight.Go now.	👎
talisencrw	While it's clearly not as superlative as its famous predecessor, it's simply because of both Heath Ledger's outstanding and Oscar-winning performance, and the character of the Joker's etching in our collective unconscious as the greatest comic-book villain ever created (with Lex Luthor a very distant	👎

Conclusion

The recommendation of a movie using cosine similarity was a success, achieving and efficiency close to 83%. This efficiency can be easily improved if we add extra movie features

Naïve Bayes' algorithm was implemented to perform sentiment analysis, and the results are truly appreciable.



```
from sklearn.metrics import confusion_matrix
print(confusion_matrix(y_test,y_pred))
```

```
[[83 19]
 [23 75]]
```

This is the result I got after building a Naïve Bayes' model to filter the reviews.

REFERENCES

[1] https://towardsdatascience.com/natural-language-processing-feature-engineering-using-tf-idf-e8b9d00e7e76

[2] http://docs.python.org/library/pickle.htm

[3] Machine Learning, Tom Mitchell, McGraw Hill

[4] https://flask.palletsprojects.com/en/2.0.x/

[5]https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/

[6] https://www.w3schools.com/js/

[7] https://towardsdatascience.com/how-to-build-a-movie-recommendation-system-67e321339109