IMDB MOVIE RECOMMENDER SYSTEM

Akhil chaduvula

Computer science department
PES UNIVERSITY
Bengaluru, India
Email: akhilchaduvula@gmail.com

A . Nayan varma

Computer science department
PES UNIVERSITY
Bengaluru, India
Email: nayanvama3@gmail .com

Neeli Krishna Dheeraj

Computer science department
PES UNIVERSITY
Bengaluru
Email:neelidheeraj2001@gmail.com

***Abstract*—This paper focuses upon EDA on IMDB data and recommendation system using content based recommender system**

Keywords— IMDB , EDA , VISUALISATION , MOVIES, GENRE , STUDIO NAME , BOX PLOT , NLTK, WORDNETLEMMATIZER , PORTERSTEMMER , SKLEARN , TFIDFVECTORIZER.

I. INTRODUCTION

“ We see movies everyday, but how do we decide which movies to watch. It is famously known that a majority of the people majorly depend on the famous IMDb website. We take descriptions given by people for each movie and recommend others based on current watching movies using a content based recommender system. For this we take an existing dataset and train this model in order for our application to recommend similar movies accordingly. In this paper we start with EDA of data and end with a recommender system. “.

II. BACKGROUND

The field of entertainment is ever growing. Movies have become an inspiration for people to dream and imagine the possibilities of life.When they get bored and start watching movies they will soon run out of movies in their list and will have no idea on what to watch next . Our part comes from here. We take the movies they watched previously and recommend similar movies based on movie descriptions given by users at the start and after he got some profile we recommend based on collaborative filtering. For this we are using the IMDB movie data set .
Movies data set contain 85,855 movies , having 22 attributes about movies such as movie description, average rating, number of votes, genre, etc.

Ratings dataset contain 85,855 rating details from a wide range of demographics

Names dataset contains 297,705 cast members along personal attributes like birth details, death details, height, spouses, children, etc.

The title principals dataset contains 835,513 cast members roles in movies with attributes such as IMDb title id, IMDb name id, order of importance in the movie, role, and characters played.

III. LITERATURE REVIEW / PREVIOUS WORK

A. Paper 1:

The title of the paper is IMDb Explorer: Visual Exploration of a Movie Database.
This was published in August 2018.
This paper focuses on the Visualisations in an IMDb movie database. and the importance of the understandings that we get from that paper. Two very unique visualisations are taken and are combined to get some conclusions from the process. Also, a small experiment was conducted in the end to get out the pros and cons.
The assumptions made in this paper are that firstly we are only focussing on the data available with us and secondly only 28 variables(columns) are taken from the whole dataset.
The observations the found were that
a) Different visualization techniques additionally support the analysis on different levels while simple graphics show the data variables by visual variables like line thicknesses or colour coding, in particular the highlighting and linking between the views is effective.
This paper gave us two completely different ways of interpreting data namely, The movie Cosmos and the career lines. We also found out that conducting small experiments on the dataset gives out more limitations and scalability issues regarding the dataset.
The graphs / visualization part that is used in this paper are very hard to interpret because the methods which this paper used are not used very often so it would be great if this paper uses some popular techniques and get same insights

B. Paper 2:

The title is "Movie Success Prediction using Machine Learning Algorithms and their Comparison" and the year the paper was published was 15-17 December during the ICSCCC conference.
Since Our project is Predicting IMDb scores of different movies based upon different factors, we decided to use machine learning Algorithms to get hold of a good classification algorithm, we chose this paper to explore the

findings of the authors about different classification algorithms.
Because this paper is related to different Algorithms, it doesn't make any specific assumptions. The main findings from this paper are that through different algorithms such as SVM, Random Forest, Ada Boost, Gradient Boost, K-Nearest Neighbours were discussed, it was found out that Random Forest Gives out the best Accuracy(Which is not the only measure) followed by Gradient Boost, but these took up a lot of time to give out their scores. Whereas the others gave fewer accuracy scores but took very less time to show their results.
Our finding for this paper is very specific to our dataset. It is that. Movie success does not only depend on features related to movies. A number of the audience plays a very important role in a movie to become successful. Because the whole point is about audiences, the whole industry will make no sense if there is no audience to watch a movie. The number of tickets sold during a specific year can indicate the number of audiences of that year. This paper highlights the importance of the number of people rating and watching a particular movie. Though it is necessary to take the average of people's ratings, it is also important to maintain a threshold of the minimum number of people rating a movie.

C. Paper 3

The title of this paper is "Literature Summary of Prescriptive Analytics" and was published in April 2019.
This paper basically focuses on Prescriptive Analytics.
Business Analytics is that branch of Analytics that enables organizations to make better, faster and quicker Decisions. There are three branches of business Analytics namely:-
Descriptive Analytics:- This branch mainly answers questions like "What Happened?", "When it happened?" and "What is happening now?".
Predictive Analytics:- This branch answers questions like "What will happen?" and "Why will it happen?".
Prescriptive Analytics:- This branch answers questions like "What can be done?" and "How it can be done?"
Usually in Descriptive Analytics and Predictive analytics are given more attention as compared to that of Prescriptive Analytics. As a step to work towards Prescriptive analytics, this article has been written. Prescriptive Analytics utilizes artificial intelligence, optimization algorithms and expert systems in a probabilistic context to provide adaptive, automated, constrained, time-dependent and optimal decisions. Below is a diagram to represent the roles of all kinds of business analytics.
This paper is a generic paper which can be the next step in the future of Analytics since this aspect is not used much in analytics.
This paper further goes on discussing the different categories of classification of Predictive and prescriptive analytics like Probabilistic Models, data mining models, Statistical Analysis, Mathematical Models, Simulation, Logic-based Models. The main takeaway from this paper is that normally everyone does only Descriptive and Predictive Analysis for Analysis of data. It is also important to dive deep into Prescriptive Analysis so that further descriptive analysis can be done.
The main issues with this paper was that it did not focus on the errors that could occur in prescriptive Analytics. An occurrence of error in the Prescriptive Analytics could lead to drastic changes in further Descriptive Analytics. This could also lead to making wrong estimates in the final result.

ASSUMPTIONS MADE ABOUT THE DATA & AREA OF PROBLEM THAT WE SEEK TO SOLVE

The First assumption we made about the data is that the data is a representative sample of all kinds of movies like it contains max. possible combinations of samples and it is not biased in any way or biased to any genre.e user's description about movies will give clear idea ab

The second assumption about the data is that thout movie and cover main features of other attributes

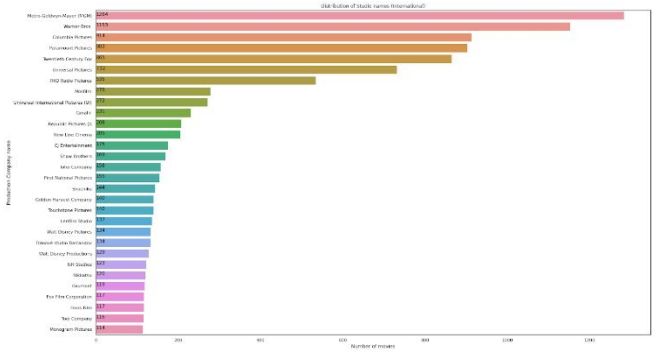
As we have seen in previous models which have used attributes like ratings, genre , country to recommend movies they may not be as much representative of user descriptions as this attribute contains detailed description of movie recommending movies based on this would be better.

But our recommender system doesn't perform well if users are biased to like movies of a particular hero or genre or any other dominating attributes

IV .PROPOSED SOLUTION

The Solution we have used to counter the problem mentioned above is by using Content Based Filtering. Content based filtering is a mechanism in which the model recommends different items based on the similarity between features of different inputs. We introduce our solution with various EDA graphs and proceed further with Content Based Filtering

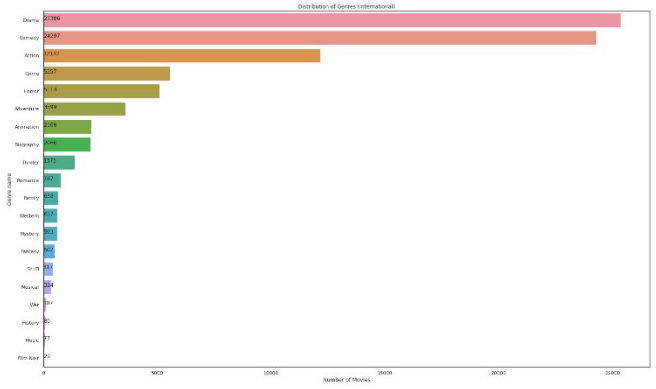
A. Distribution of studio names



The above analysis is about distribution of studio names with respect to each production
company . we can clearly see the domination of mgm, warner bros, columbia pictures in the number of movies produced . Most of the best performing production companies are Hollywood based with some chinese companies in between which comes from the fact that most of the movies produced are from hollywood . But one intuition we can gain is though india , france and some other

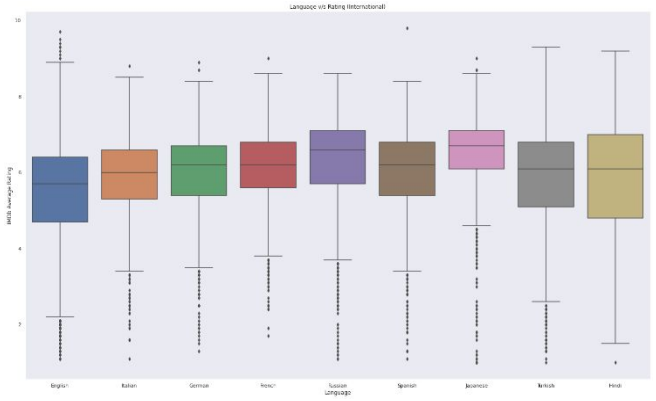
countries perform better in terms of no of movies produced than hong kong , companies of that country origin are not much seen in the list which says there is little monopoly in those countries above

B. No of movies v/s genre



As we can observe from the data the most films produced are in the genre of Drama(25386) and comedy(24297). This is due to the liking of people in the early and late 20th Century. Due to the many wars and spy thrillers Action(12182) and crime(5557) have also gained populace recently. Horror(5113) are less which can be attributed to the increase in technology in recent times.The Animation(2108) is also less in number which is due to recent advancement in technology.

C. language v/s rating (International)



The above is a box plot of the language in which the movie is made and imdb average it got. In the above model it's shown that english movies have relatively low imdb average with some outliers . This shows that
Much of english movies produced are not performing that well at the box office. Japanese movies have a high average and relatively low inter quartile range . Though they have many outliers which may be due to some other factors, we can intuitively say that japanese movies are guaranteed to get a better imdb rating . We can also see that movies in Hindi have a high inter quartile range and average nearly in the middle. We can say language is not a factor of success in India .

From A and C we can see that studio names , language don't have much significant impact on movie ratings or recommendations. The same implies for many of the variables so it would be better if we recommend movies based on descriptions given by users .
To do content based recommendation based on descriptions given by the user we first need to convert it into some numerical machine interpretable form . The above will take care of the cold start problem . After the user got some profile and started watching movies we use collaborative filtering having user -user similarity to predict movies what similar users have used

IMPLEMENTATION IN PYTHON

Introduction:

Content based filtering: Initially we have used Content based filtering. This is basically used to give a cold start. Basically Content based filtering is a type of filtering that gives out recommendations based on content that is similar to what the majority users watch . This basically gives out a Cold Start. Cold Start case is the case in which there is no previous viewing data about the user. This happens usually when the user is new to the recommendation system.

When a Cold start occurs, usually the system opts for using a content based recommendation system. Once the system has used a content based recommendation system, the system is now ready to use collaborative filtering if the user wants any more recommendations.

Collaborative filtering:

It is used to basically recommend users who have already used the recommender system before and have given their feedback for Movies that they have already seen. What this does is that it checks for the feedback given by other users for the same movie/ Similar movies and gives out the

Libraries used:

1) Pandas
2) nltk.corpus.stopwords
3) nltk.tokenize.word_tokenizer
4) nltk.stem.wordNetLemmatizer
5) nltk.stem.PorterStemmer
6) sklearn.feature_extraction.text.TfIdfVectorizer
7) sklearn.neighbours.nearest neighbours
8) sklearn.feature_extraction.text.CountVectorizer
9) sklearn.feature_extraction.text.TfIdfVectorizer

Content based filtering using descriptions :
Preprocessing :

As most of the variables are of string type , the size of data is high and the percentage of NAN's is low we dropped rows having NAN's.
We then made description attribute from string type to list where each entry is a word from description and we removed stopwords from the list (Stop words are the words which occur very often and doesn't have much importance)

V . Experimental Results

Stemming : We stemmed each word which we got after removing stop words so that parts of speech does not differentiate words

Lemmatization : After stemming to overcome the problem of different words having the same meaning we use Lemmatization in which words of similar meanings are mapped to a single word.

Feature_Extraction: we have used TfIdf vectorizer to find real importance of each word in given data and use this information

Cosine Similarity: After feature extraction and finding which words are important according to tf_idf we will find similarity between descriptions given by user using cosine similarity and store them in a matrix which will be used further to check given a movie we find whose descriptions are much similar and give those movies as suggestions

Nearest Neighbours: After obtaining the matrix, we then cluster all the similar elements in a particular matrix. and then we fit our data using the nearest neighbour classification. It takes n nearest neighbours and classifies according to a majority vote.

Since we have used two methods , one is the content based filtering and the other is the collaborative filtering.
The image shown below is the result for Content based filtering.

```
a = recommend_system("The Samaritan",sigmoid_sim)
```

```
['Beyond Re-Animator', "What's Your Number?", 'The Box',
'Inconceivable', 'Bridget', 'Saving Christmas', 'Chasing Bullitt',
'Dom Hemingway', "Nobody's Fool", 'Rainbow Time']
```

The below image is the result for collaborative filtering:

```
Enter User Id 108
Recommended Movies are:
[('Harry Potter and the Deathly Hallows: Part 1 (2010)',
 ('Black Swan (2010)', 3.9655172413793105),
 ('Devil (2010)', 3.0714285714285716),
 ('Easy A (2010)', 3.3461538461538463),
 ('Social Network, The (2010)', 3.875),
 ('Machete (2010)', 3.3636363636363638),
 ('Scott Pilgrim vs. the World (2010)', 4.0),
 ('Expendables, The (2010)', 3.125),
 ('Salt (2010)', 3.3125),
 ('Karate Kid, The (2010)', 3.3333333333333335),
 ('Inception (2010)', 4.189320388349515),
 ('Toy Story 3 (2010)', 4.142857142857143),
 ('Get Him to the Greek (2010)', 3.15),
 ('Iron Man 2 (2010)', 3.6451612903225805),
 ('Death at a Funeral (2010)', 2.75),
 ('Kick-Ass (2010)', 3.857142857142857),
 ('How to Train Your Dragon (2010)', 3.757142857142857),
 ("She's Out of My League (2010)", 3.0),
 ("Valentine's Day (2010)", 3.0),
 ('Avatar (2009)', 3.856060606060606),
 ('Invictus (2009)', 3.5833333333333335),
 ('Blind Side, The  (2009)', 3.9347826086956523),
 ('Ninja Assassin (2009)', 3.2857142857142856),
 ('Fantastic Mr. Fox (2009)', 4.0),
 ('Up in the Air (2009)', 3.92),
 ('Zombieland (2009)', 3.765957446808511),
 ('Men Who Stare at Goats, The (2009)', 3.1785714285714284),
 ('Julie & Julia (2009)', 3.7222222222222223),
 ('(500) Days of Summer (2009)', 3.9324324324324325),
 ('Hangover, The (2009)', 3.705357142857143),
 ('Up (2009)', 4.153225806451613),
 ('X-Men Origins: Wolverine (2009)', 3.3214285714285716),
 ('Moon (2009)', 4.0625),
 ('Crank: High Voltage (2009)', 2.0),
 ('17 Again (2009)', 2.875),
 ('Observe and Report (2009)', 2.5714285714285716),
 ('I Love You, Man (2009)', 3.5357142857142856),
 ('Funny People (2009)', 3.5),
 ("He's Just Not That Into You (2009)", 3.1666666666666665),
 ('Watchmen (2009)', 3.8020833333333335),
 ("Dr. Horrible's Sing-Along Blog (2008)", 4.434782608695652),
```

Checking:
Since this is a Recommender System, we have manually checked ourselves, with data(Movies that we have known ourselves). and cross evaluated with similarity scores as well. We found that initially movies that were recommended are static and are based on one or two movies which the user was initially interested in .

But after the user get some profile and start watching much more movies he moves between various clusters and gets the movies recommended based on his interest and movies which are recently watched gets high priority

Conclusion:

The Cinematic Industry is one of those few industries that can influence people and become part of everyone's lives. It is ever growing and has a huge scope of growing even more. There are a variety of movies out in the market, which may lead to a huge confusion as to which movie to watch. People watch movies mostly to relax and stay satisfied. Hence it is necessary that they select the right movie in as less time as possible as many users waste most of their time trying to find the right movie and finally end up watching nothing .

This is where our recommendation system plays a huge role. Our recommendation system is capable of recommending users who are new to the system(Content Based filtering) , it is also capable of supporting users who have already used our system,(collaborative based filtering). This kind of system is very effective as it is capable of recommending users based on their taste.

We feel that this system can also be improved if we are capable of grabbing the users' watch history apart from their recommending history. This can greatly improve the performance of our system.

VI. REFERENCES

Paper1: Visual Exploration of a Movie Database. https://www.researchgate.net/publication/325596292_IMDb_Explorer_Visual_Exploration_of_a_Movie_Database.

Paper2: Movie Success Prediction using Machine Learning Algorithms and their Comparison

"https://ieeexplore.ieee.org/document/8703320"

Paper 3: Literature Summary of Prescriptive Analytics

"https://www.sciencedirect.com/science/article/pii/S0268401218309873"