



iJRASET

International Journal For Research in
Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 9 Issue: I Month of publication: January 2021

DOI: <https://doi.org/10.22214/ijraset.2021.32936>

www.ijraset.com

Call:  08813907089

E-mail ID: ijraset@gmail.com

Movie Recommender System

A Nayan Varma¹, Akhil Chaduvula², Neeli Dheeraj Krishna³

^{1, 2, 3}Students, Department of Computer Science and Engineering, PES University, Bangalore-560085, India

Abstract: This paper focuses upon EDA on IMDB data and building of a recommendation system using Content and Collaborative based recommender system based on K nearest neighbours and NLP techniques.

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 a fact 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. LITERATURE SURVEY

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.

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

- 1) Different visualisation 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 / visualisation 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

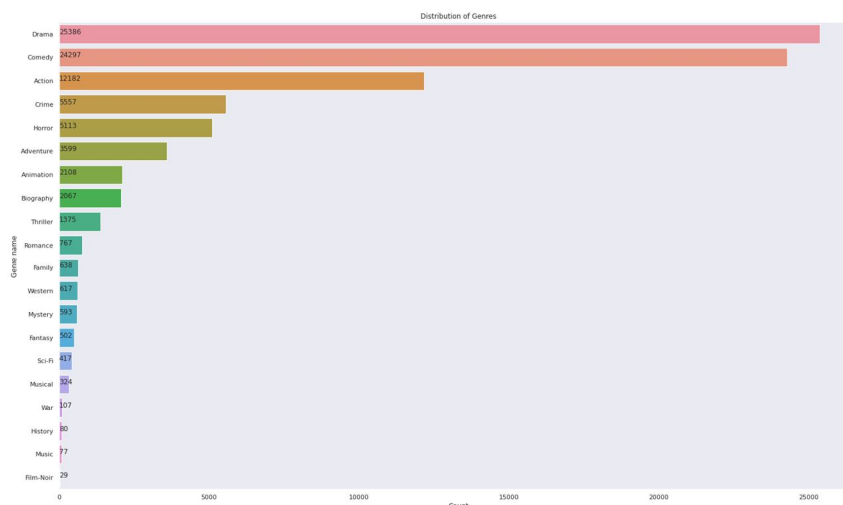
The title is "Movie Success Prediction using Machine Learning Algorithms and their Comparison" and the year the paper was published was 15-17 December during the ICSCC conference.

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.

834

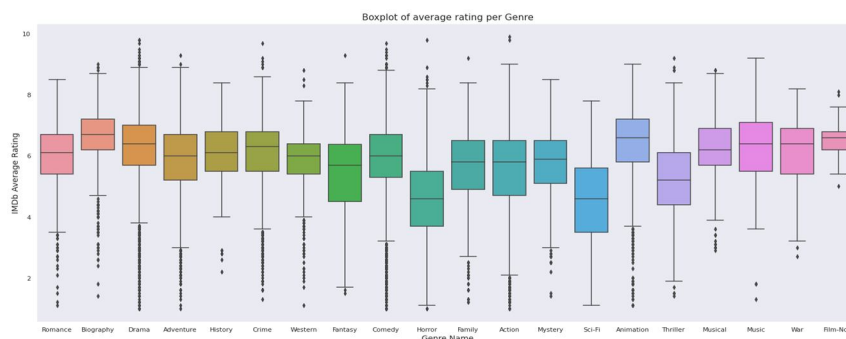
Most of the best performing production companies are Hollywood based with some Chinese companies in between which comes from the fact that most of the movies produced are from Hollywood . But one intuition we can gain is though India , France and some other countries perform better in terms of no of 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 its 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.

D. Implementation in Python

- 1) *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.
- 2) *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 recommended movies.

E. 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

F. Content Based Filtering using Descriptions

- 1) *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 stop-words from the list (Stop words are the words which occur very often and doesn't have much importance)

IV. EXPERIMENTAL RESULTS

- 1) *Stemming*: We stemmed each word which we got after removing stop words so that parts of speech does not differentiate words
- 2) *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
- 3) *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
- 4) *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)', 3.9655172413793185),
('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),
```

- 1) *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

V. 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.

We are providing additional links containing the more detailed EDA and the complete practical project.

EDA: https://colab.research.google.com/drive/1bWA-fCrDplPmu94I9oX-YBt_eEwMAsfY?usp=sharing

Project: https://colab.research.google.com/drive/18iVSq_THpRegOmBHsowu-PtqPFx27jvw?usp=sharing

REFERENCES

- [1] Visual Exploration of a Movie Database. https://www.researchgate.net/publication/325596292_IMDb_Explorer_Visual_Exploration_of_a_Movie_Database
- [2] Movie Success Prediction using Machine Learning Algorithms and their Comparison "<https://ieeexplore.ieee.org/document/8703320>"
- [3] Literature Summary of Prescriptive Analytics "<https://www.sciencedirect.com/science/article/pii/S0268401218309873>"



10.22214/IJRASET



45.98



IMPACT FACTOR:
7.129



IMPACT FACTOR:
7.429



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089  (24*7 Support on Whatsapp)