Classifier To Predict The Ratings Of Upcoming Movies

A. G. D. L. C. Priyanganie

2018

Classifier To Predict The Ratings Of Upcoming Movies

A dissertation submitted for the Degree of Master of Computer Science

A. G. D. L. C. Priyaganie
University of Colombo School of Computing
2018

Declaration

The thesis is my original work and has not been submitted previously for a degree at this or any other university/institute.

To the best of my knowledge it does not contain any material published or written by another person, except as acknowledged in the text.

Student Name: A. G. D. L. C. Priyanganie

Registration Number: 2014/MCS/062

Index Number: 14440621

_____

Signature: Date:

This is to certify that this thesis is based on the work of

Mr. A. G. D. L. C. Priyanganie

under my supervision. The thesis has been prepared according to the format stipulated and is of acceptable standard.

Certified by:

Supervisor Name: Dr. D. A. S. Atukorale

_____

Signature: Date:

Abstract

In today's world, movies are released rapidly in all around the world. Therefore, in a particular city, there can be more than one movie in theatres at a time. But people have a busy tight schedules in their life. So, they don't have enough time to watch each and every movie. Also, due to higher cost of living, people can't afford to watch all movies in theatres. So, people always try to find the better movies which are worth watching in a movie theatre. Therefore, they try to look for recommendations and non-spoiler reviews from other people.

As a solution for this problem, this research project proposes a model. This model was created by analyzing existing movie data, extracting features from it and identifying a relation between the features and the movie rating.

There are number of existing research work on movie rating prediction. But all of them have some kind of limitation such as less accuracy, inability to predict rating before movies are released etc. Motive of this research is to overcome most of those limitations.

The original dataset was taken from Kaggle and it was updated with some new and missing data retrieved from Facebook, Youtube and OMDb APIs. Most of the existing features were also modified so that they can contribute to the final model in better ways. In this process, the representation of some features were changed, some features were split into multiple features and some features were pruned to have only a selected values in them.

Multiple classifier algorithms were evaluated before developing the model and at the end, J48 decision tree algorithm with bagging was selected as it gave the best results. At this point, the output of the prediction model (i.e. the movie ranking) was given as an integer number between 1 to 10. However, in the real world, a person who's interested in knowing whether a movies is good or not, does not expect such an accuracy. A scale of "Great", "Ok" and "Poor" would be a good enough measure for this.

Taking this fact into consideration, the model was updated to output only 3 values for the rating, namely "High", "Medium" and "Low" which gave an higher accuracy than earlier.

Acknowledgements

I would like to express my gratitude to my supervisor, Dr. D.A.S. Atukorale, The Deputy Director, University of Colombo School of Computing, for the useful comments, remarks and engagement through the learning process of this master thesis.

Then, I wish to extend my gratitude to Prof K. P. Hewagamage, The Director, University of Colombo School of Computing, for giving me the opportunity to carry on with my research work.

Moreover, I wish to extend my deep sense of gratitude to the Non-Academic staff members of Department of Computer Science, Faculty of Science, University of Colombo School of Computing,

Finally, I would like to thank all who were silently involved in putting together this research project. I wish them nothing but the best.

Table of Content

List of Figures

List of Tables

List of Abbreviations

Abbreviation	***Explanation***
MSE	Mean Squared Error
SVM	Support Vector Machine Regression model
WEKA	Waikato Environment for Knowledge Analysis
KNN	K-Th Nearest Neighbor
IMDb	Internet Movie Database

Chapter 1: Introduction

1.1 The problem and motivation

In today's world, movies are released rapidly from all around the world. Therefore, in a particular city, there can be more than one movie in theatres at a time. But people have busy tight schedules in their life. So, they don't have enough time to watch each and every movie. Also, due to high cost of living, people can't afford to watch all movies in theatres. So, people always try to find better movies which are worth watching in a movie theatre. Therefore, they try to look for recommendations and non-spoiler reviews from other people.

To serve that purpose in a way, there are many popular websites which have movie ratings and movie information. Some of them are, IMDB (Internet Movie Database)[1], Rotten Tomatoes[2], Roger Ebert[3], Guardian[4] and Meta Critic[5] etc. But, even though there are lots of reviews and accurate ratings about old movies in those websites, the fresh movies either don't have a rating, or the ratings they have are not very accurate because the number of votes for a movie is low at the beginning. This error component only reduces when it receives more votes which takes time. Therefore, people who prefer watching movies just after they are released, do not have a way to find out how good a movie is, before they decide to watch them. So, if there is a model which can predict the rating of a movie before it is released, it will be important as it will help people a lot in aforementioned cases by saving both their time and money.

Not only those who watch movies but also the movie theater owners do not yet have a proper and promising way to predict movie ratings before they are released. If they could, they could get better and profitable movies to their theaters.

Even movie directors/producers could use such a model to improve their movies in the aspects such as which actors should be used, what's the best time to release a movie etc. On the other hand, if such a model is available for movie directors/producers, they could use that information to market their movies, to get more people to watch them. That would help them to reduce the cost of marketing stunts too, as the predictions of this suggested model will be trusted by everyone.

Those were the motivations for the idea of developing a model to predicate the rating of upcoming movies. However, for that we required a large amount of movie data with rating information. When selecting the dataset for this, it was more important to select a trusted dataset so that it gives more accurate results especially about the ratings. Obviously, if the number of voting is higher, rating accuracy is higher too. And for the number of voting to be the highest, there should be a higher web traffic to the relevant movie website. Therefore, a dataset should be taken from the movie rating website which has the highest web traffic.

Therefore, the movie websites with the highest traffic were selected according to the Alexa[6] index. Alexa is a traffic based index calculating system for websites. Below table contains the different movie websites with their Alexa ranking.

Movie website	Alexa Rating
Roger Ebert - http://www.rogerebert.com/	9215
Guardian - https://www.theguardian.com/film+tone/reviews	138
Rotten Tomatoes - https://www.rottentomatoes.com/	424
IMDB - http://www.imdb.com	57
Meta Critic - http://www.metacritic.com/	1315
MrQE - http://www.mrqe.com/	231187
Flixster - http://flixster.com	39180

Table 1.1: Movie websites with their Alexa ranking number as per July, 2017

As per Table 1.1, IMDB movie website has the highest Alexa ranking (i.e. 57). Therefore, a dataset will be selected from that website. IMDB is one of the main online databases which contains data related to movies, TV series etc. People rate movies and give reviews on IMDb. It helps other people to decide which movies they want to watch.

1.2 Objectives & Scope

The main objective of this research is to develop a model which can predict movie ratings before they are released.

In IMDB, there are lots of data related to movies such as director, writers, actors etc. Apart from those direct factors, relationships between different movies (eg. director's previous movies, actors'/actress' previous movies, previous parts of the same movie etc.) also can affect the rating of a movie indirectly. So the objective of this project is to use feature selection and feature extraction techniques to identify important features and develop a classifier which can classify new movies into rating categories such as 1 star, 2 stars, 3 stars etc. out of 10 stars.

To generate indirect properties of movies which can be important features, a new tool will be required to be implemented. Therefore that will be another objective of the project.

The scope of this project includes :

- Building a tool to clean and enhance the dataset
- Comparing classification algorithms
- Analyzing data to select and extract features, to identify important features
- Developing a model to classify new movies into rating categories

1.3 Deliverables

- Tool to generate indirect properties of movies and to enhance dataset
- Model to predict ratings of upcoming movies.

1.4 Organization of the Thesis

This dissertation presents a model which was developed to predict the IMDb rating of upcoming movies. The organization of this thesis is as follows. Chapter 1 gives an introduction to the domain and problem space. Chapter 2 discusses about related researches done so far. Chapter 3 presents the research methodology, details of analysis, information about dataset used and the solution. Chapter 4 presents about the proposed solution and

model. Chapter 5 presents the evaluation methods and results. Chapter 6 discusses about conclusion and future work.

Chapter 2: Literature Review

2.1 Predicting Movie Success Based on IMDB Data

In their research, Nithin V.R., Pranav M., Sarath Babu P.B. and Lijiya A. has developed a model to predict movie success and IMDB rating based on IMDB data[7]. They have done the prediction for movies which are released in the United States and in the English language from 2000 to 2012. Initially, they had taken dataset from IMDB. They have removed the movies which don't have any information about box office details. They have filled missing data fields of the dataset from Wikipedia and Rotten Tomatoes.

As the dataset was a collection of both nominal and numeric attributes, for the regression process they have had to convert corresponding nominal values to numeric values. They have taken correlation between all different features that were considered and the movie revenue. To avoid redundancy and irrelevant attributes, they have taken the correlation between the features themselves. When selecting the best feature subset, they have used the greedy backward procedure.

They have used Linear Regression model, Logistic Regression model, Support Vector Machine Regression model (SVM model) to predict the revenue and have compared the output from the each method. In the linear regression model, they have used standard least-squares linear regression[8]. For the logistic regression model, they have had to change the regression problem to a classification problem. So they have split revenue to buckets and generated a histogram by dropping movies to each bucket.

At the end of comparison, they have identified linear regression model was the most accurate model which was about 51% accurate, while logistic regression model and SVM model had 42.2% and 39% accuracies respectively. Even though they identified 20 features from the dataset at the beginning of the research, they have found only 7 features such as budget, director, writer etc. were the most significant features.

However, as per their conclusion of the research, the success percentage for all movies didn't look good for industrial use, and they believe that if their training set (i.e. 1050 movies)

were larger and if they considered additional features like social network data, News analysis, they could improve the performance of the model.

Another important fact they have ignored in their research is that correlation between movies themselves. For example, has the director of a particular movie directed any other popular movies? Etc. If they considered that they could have increased the accuracy furthermore.

2.2 Predicting IMDB movie ratings using Google Trends

Deniz Demir, Olga Kapralova, and Hongze Lai, in their research, have developed a model to predict IMDB movie ratings using Google search frequencies for movie related information[9]. They have used IMDB dataset and Google search frequencies as the input to train the model. For that, for they have used 400 movies in America between a period of 3 years. 50% movies of them are good in IMDB rating (rating is greater than 6) and others are bad in IMDB rating (rating is less than or equal to 6). For each movie, they have collected movie title, movie director, movie actors and the release date.

They have used two different approaches to predict movie popularity. One approach was combining Google Trends and Google AdWords statistics, and the other one was using Google Trends statistics only.

In the first approach, they have compared the performance of the logistic classifier, SVM model, and multilayer perceptron. However, none of those methods gave more than 55% accuracy in the output. So the conclusion was all of those methods performed similarly to a fair coin toss, which did not give any important output.

In the second approach, they have used only Google Trends data. Then they have tested the same 3 models used in the first approach. Interestingly, the accuracies have become a little bit higher in the second approach where SVM model has got 72% accuracy while other two have got around 60% accuracy[9].

As per their findings, in general, the number of Google search queries for a particular movie starts going up one week before the movie is released and it reaches the highest around the release date. Then, after about 4 months the trend of search goes away. They also have observed that long term post-release period's Google search activity has higher prediction

ability than pre-release and short term post-release search activity. That means, for a better prediction they need to wait till at least 3-4 months after the movie is released. But at that time, the value of a prediction may not be much of worth.

2.3 Predicting movie rating using the MovieLens dataset

If we consider a particular movie, different viewers can have different opinions about it, and the way they enjoy it is different. So eventually, the rating they give is obviously different. Based on this fact, Yashodhan Karandikar has conducted a research on predicting the rating of a movie from a particular user's viewpoint[10].

In his dataset, he has had 1 million ratings from 6040 users on 3900 movies. For each user, he had age, gender, occupation and zip code. For each move, he had title and genre. In his research, he has used 3 different methods develop a model; Linear Regression, Collaborative Filtering, and Latent Factor model. He has first divided the data set into two using 80%-20% split and used the latter as the test data set. Then he has further divided the former into two again using 80%-20% split, and used the new 20% as the validation set and rest as the training set.

After comparing the results of 3 methods, Yashodhan has come to a conclusion that Latent Factor model tends to perform better than the other two methods.

However, the dataset of his research has quite a limitation, which is that he has only taken movie genre as the features of movies. But obviously, that's not quite enough. For example, if someone likes action movies, we can't expect they will like every action movie. There can be many reasons not to. Therefore, when selecting features of movies, at least the basic features like director and actors etc, should be used.

2.4 Predicting rating for Amazon movies

When a movie is first released, there are a few number of ratings and reviews. That makes the accuracy of overall rating lower. This is called cold-start problem. Rajiv Pasricha has tried to address this problem for Amazon Movies, in his research[12]. In Amazon Movies, users can both rate movies and write reviews for them. Rajiv has claimed that cold-start

problem can be mitigated to some extent if you consider both rating and reviews instead of just the rating.

He has used text extraction and text analysis tools to start analyzing reviews. He has trained two supervised learning algorithms on the data, linear regression, and logistic regression. To evaluate these two, he has used Mean Squared Error (MSE) method, mainly because of its easiness to use.

He has taken 50000 reviews from Amazon and split them into two as 90%-10%. Then he has selected 10% as the test dataset. He then has split the other part again into two as 90%-10% and taken as training dataset and validation dataset respectively.

After analyzing the results he has found that simple regression model is more accurate for his dataset, and hence concluded that simple regression model is the most effective at handling cold-start users and items.

2.5 Identified Research Gap

Among all above movie rating prediction researches, most of them had used features which are available only after the release of movies. Therefore those models are unable to predict the movie rating before the movies are released. The only a few researches, which had not used features available only after the release, had a less accuracy compared to the others. So the motive of this research was to develop a high accuracy model to predict movie ratings before they are released.

Chapter 3: Analysis and Design

3.1 Introduction to dataset

The Initial dataset was taken from Kaggle[13]. There were data about 5043 movies which were extracted from IMDb website and there were 28 features in the initial dataset. They are,

1. Movie title
2. Movie IMDb link
3. Color (i.e whether it is a color, or black and white movie)
4. Genres
5. Budget
6. Duration
7. Gross
8. Number of voted users
9. Cast total facebook likes
10. Face number in poster
11. Plot keywords
12. Number of users for reviews
13. Language
14. Country
15. Content rating
16. Title year
17. Imdb score
18. Aspect ratio
19. Number critic for reviews
20. Director name
21. Director facebook likes
22. 1st actor name
23. 1st actor facebook likes
24. 2nd actor name
25. 2nd actor facebook likes

26. 3rd actor name
27. 3rd actor facebook likes
28. Movie facebook likes

3.2 Data Preprocessing

Kaggle dataset had both movies and tv series data. Tianxin Yu[14] has used the same dataset for his research and he had used all the data to predict rate. In the dataset, TV Series were rated based on both series and episode wise.

Ex : Prison Break TV series' IMDb rating was 8.4[19]. And there were IMDb ratings for its individual episodes too. S1.E21's rating was 9.5 and S4.E10's rating was 8.3. This rating format different between movies and TV series can make confusions for the model as those 2 are completely different in nature and in a way it can be wrong to predict one's rating based on other's information.

Therefore, it was decided to remove TV series from the dataset. But the problem was the dataset didn't have any attributes to differentiate a movie from a TV series. To achieve that, an external API called OMDb[15] was used to filter movie data. The general format of an OMDb API call is like this.

http://www.omdbapi.com/?i={imdb-id}

Eg.:- Sample OMDb API call to get information about "Logan" movie and response is below.

http://www.omdbapi.com/?i=tt3315342

OMDb API response


```
{
  "Title":"Logan",
  "Year":"2017",
  "Rated":"R",
  "Released":"03 Mar 2017",
  "Runtime":"137 min",
  "Genre":"Action, Drama, Sci-Fi",
  "Director":"James Mangold",
  "Writer":"James Mangold (story by), Scott Frank (screenplay by), James Mangold (screenplay by), Michael Green (screenplay by)",
  "Actors":"Hugh Jackman, Patrick Stewart, Dafne Keen, Boyd Holbrook",
   "Plot":"In the near future, a weary Logan cares for an ailing Professor X, somewhere on the Mexican border. However, Logan's attempts to hide from the world, and his legacy, are upended when a young mutant arrives, pursued by dark forces.",
  "Language":"English, Spanish",
  "Country":"Canada, Australia, USA",
  "Awards":"Nominated for 1 Oscar. Another 11 wins & 46 nominations.",
  "Poster":"https://images-na.ssl-images-amazon.com/images/M/MV5BMjQwODQwNTg4OV5BMl5BanBnXkFtZTgwMTk4MTAzMjI@._V1_SX300.jpg",
  "Ratings":[
    {
      "Source":"Internet Movie Database",
      "Value":"8.1/10"
    },
    {
```

```
      "Source":"Rotten Tomatoes",
      "Value":"93%"
    },
    {
      "Source":"Metacritic",
      "Value":"77/100"
    }
  ],
  "Metascore":"77",
  "imdbRating":"8.1",
  "imdbVotes":"442,722",
  "imdbID":"tt3315342",
  "Type":"movie",
  "DVD":"23 May 2017",
  "BoxOffice":"$226,276,809",
  "Production":"20th Century Fox",
  "Website":"http://www.foxmovies.com/movies/logan",
  "Response":"True"
}
```


This response had “Type” attribute, of which the possible values could be “movie”, “series” or “episode”. In this case only the movies with type “movie” were selected. Since it was needed to call this API for every entry in the dataset to find out whether it’s a movie or a TV series, some other opportunities were also met in the process.

1. In the initial dataset, there were a lot of missing values/empty values. Most of them could be filled with the values came from OMDb API responses.
2. Three addition features were taken from OMDb API response. They are,
    i. Writer name
    ii. 4th actor name
    iii. Production Company

A movie rating highly depends on its story and the content. But there were no direct way to get a measure of its quality of the story. But, using the “writer”, movie content could be measured to some extent in an indirect way. Normally, most of the movies have a main actor and actress. Apart from that, there can be a supportive main actor and supportive main actress. So, that way, A movie can have four major characters. But in the initial dataset, there were only 3 actors. Using OMDb API, another actor detail could be extracted.

Most of time, movie quality depends on its production company. Even when the movie story is good, actors are very famous, but if the production company has lack of technology capabilities, final movie definitely will be a low quality output. Therefore, movie production company also indirectly affects to the movie rating.

3.3 Data Exploration

3.3.1 Feature Selection

In the initial dataset, all the fields could be divided into two types.

1. Data which can be received even before a movie is released such as budget, actors/actresses names, director name, language etc.
2. Data which can be received only after the release of the movie such as movie facebook likes, number of users for reviews etc.

Therefore, the features which can be found before the release were chosen as the feature set. So, features such as "gross income, number of users who reviewed, Number of critics who reviewed, movie facebook likes" were removed from the dataset.

One important thing to note here is that even though the "number of voted users" feature also can be found only after the release of a movie, that feature wasn't removed from the dataset. The reason will be explained later.

IMDb score distribution of the initial dataset was like in Figure 3.1. According to that, IMDb score was skewed to the higher numbers. 80% values of IMDb score was lying between 5 and 8. Without a normally distributed IMDb score, accuracy of the prediction model could be less accurate. Therefore, additional IMDb movie data had to be added to initial dataset such that IMDb score gives a normalized distribution.

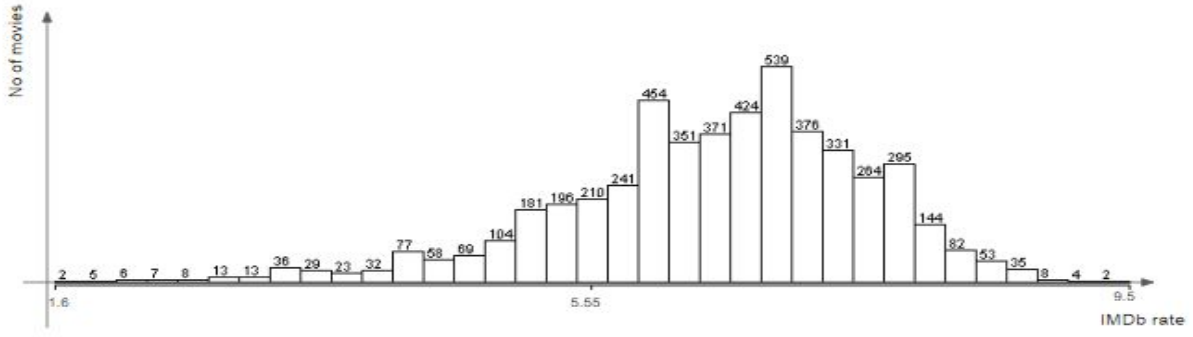

Figure 3.1: IMDb score distribution of initial dataset

In the initial dataset, there were some movies having IMDb score greater than 7 with a very little vote counts (less than 100). But the movie ratings are not accurate until they get a considerable amount of votes. This is highly applicable to movies having a high ratings. But for the movies with low ratings, this may not be relevant. Based on that assumption, movies with IMDb rating greater than 6 were ignored if number of votes were less than 10000.

Here, prediction is going to be the IMDb rating. So, when the output model is being created, IMDb rate is the class variable. If there are too much distinct values for the class variable, the output model will be overfitting. In the dataset, there were 80+ different values for IMDb score. To avoid that, 10 values for the IMDb score were defined. Initial IMDb ratings were grouped into 10 groups and they were labeled as shown in Table 3.1.

Updated Class variable value	**IMDb rating group**
1	0 - 1
2	1.1 - 2
3	2.1 - 3
4	3.1 - 4
5	4.1 - 5
6	5.1 -6
7	6.1 - 7
8	7.1 - 8
9	8.1 - 9
10	9.1 - 10

Table 3.1: IMDb rating groups and updated label names

In the initial dataset, there were movies from year 1936 to 2017. Old movies and latest movies had different technologies, and other different aspects. Some old movies were “black and white” movies. But there were no black and movies in last 2 decades. The “Color” field of Kaggle dataset contained the value of either “Color” or “Black and White”. If the model was created with old movies and latest movies together, the movie prediction model can have

unnecessary complications. Therefore, before creating the model, films were filtered based on its year. If the movie was created on or after 1980, it has latest technologies. Based on that assumption, if the movie title year is 1980 or after that only, movie was taken to the dataset. Aspect ratio feature was also removed from the dataset, as it's relevant only for old movies.

In the initial dataset there were 47 languages. 90% of movies of the initial dataset were English movies. Figure 2 shows the distribution of languages in the dataset. As most of the movies were English movies, only they were considered to build the movie rating prediction model. All around the year, USA and UK were the majority of English movie producing/releasing countries. Based on that assumption, country feature was recategorized like this.

- USA
- UK
- USA and UK
- Other Countries

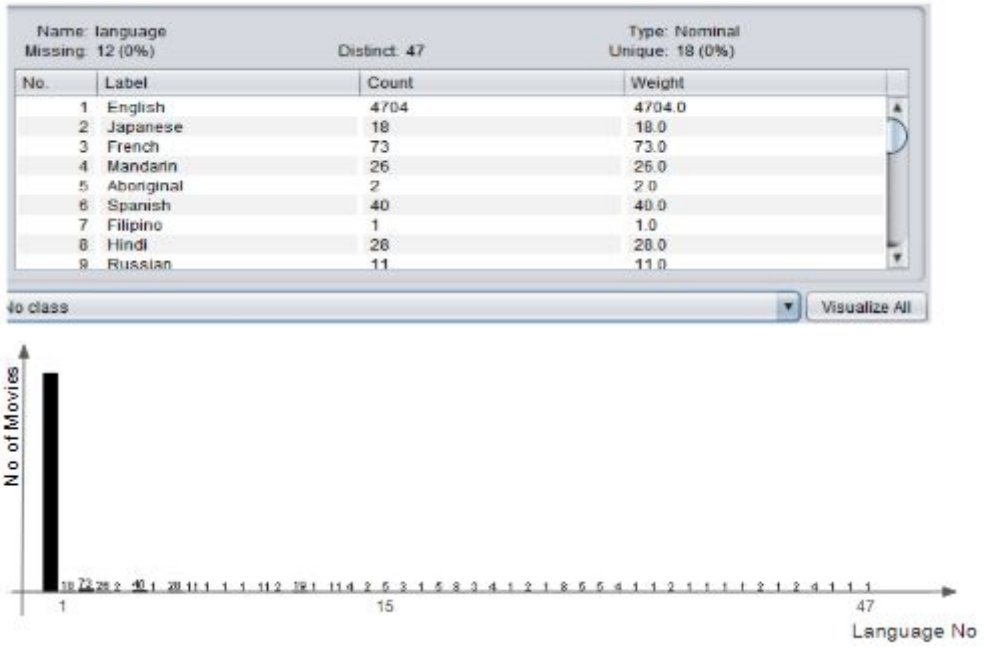

Figure 3.2: Language distribution of Initial dataset

3.3.2 Facebook User likes

There were 3000+ distinct values for each actor, writer, and director fields. More than 2000+ values had only one occurrence in the dataset. If the output model is created with such kind of data, output model could overfit for the dataset. If some numeric value can replace actors', writers' and directors' names, data problem of those fields can be solved. One of the solutions for this is using facebook likes of pages of those persons, if there are any. Nowadays, Facebook is the most popular social media network. If a person is getting popular, they usually create a Facebook page so that their fan base can follow them up. Then their fans have to subscribe to those pages. In the initial dataset, there already were facebook likes features for the 1st actor, 2nd actor, 3rd actor, and director. So, Facebook API was used to get facebook likes of 4th actor and the writer.

General format of Facebook API call to get user account id from the user name is shown below.

https://graph.facebook.com/v2.10/search?q={actor/director/writer name}&type=page&fields=fan_count,name,is_verified&access_token={access token}

Ex: Sample Facebook API call for an actor called "Prabas" and output API response with account ids is shown below.

https://graph.facebook.com/v2.10/search?q=Prabhas&type=page&fields=fan_count,name,is_verified&access_token=226478960815387|RoqpAAxoK_US5X0ZxnxtCMis0Ac

Facebook API response


```
{
  "data": [
    {
      "fan_count": 10241604,
      "name": "Prabhas",
      "is_verified": true,
      "id": "378233035640910"
    },
    {
      "fan_count": 3147,
      "name": "Actor Prabhas",
      "is_verified": false,
      "id": "702438293284308"
    },
    {
      "fan_count": 3027371,
      "name": "Prabhas  FC",
      "is_verified": false,
      "id": "258564894300101"
    },
```

```
    {
      "fan_count": 600113,
      "name": "Prabhas  Actor",
      "is_verified": false,
      "id": "1657001787885440"
    }
  ],
  "paging": {
    "cursors": {
      "before": "MAZDZD",
      "after": "MjQZD"
    },
                                                                                                    "next":
"https://graph.facebook.com/v2.10/search?access_token=226478960815387\u00257CRoqpAAxoK_US5X0ZxnxtCMis0Ac&pretty=1&fields=fan_count\u00
252Cname\u00252Cis_verified&q=Prabhas&type=page&limit=25&after=MjQZD"
  }
}
```


In above API call there is a field called “fan_count” and it is the number of subscribers of the the page of the actor. And there are multiple user ids for the searched user name. So, most matching user id can be taken as the correct user account details. So “fan_count” of the most matching response was taken as the facebook like count. Algorithm shown in Figure 3 was used to process the whole data set with Facebook API.

```
private static int getMostRelaventUserFBLikes(JSONArray jsonArray, String userName) throws IOException {
    int resultsInputSize = jsonArray.length();
    String[] nameValues = userName.split(" ");
    System.out.println(nameValues.length);
    int totalLikes = 0;
    for (int i = 0; i < resultsInputSize; i++) {
        JSONObject userJsonObject = jsonArray.getJSONObject(i);
        String userId = userJsonObject.get("id").toString();
        String resultedUserName = userJsonObject.get("name").toString();
        boolean isVerifiedUser = userJsonObject.getBoolean("is_verified");
        int fanLikes = userJsonObject.getInt("fan_count");
        for (int j = 0; j < nameValues.length; j++) {
            String namePart = nameValues[j];
            if (resultedUserName.contains(namePart)) {
                if (totalLikes < fanLikes) {
                    totalLikes = fanLikes;
                }
                break;
            }
        }
    }
    return totalLikes;
}
```

Figure 3.3: Algorithm used to process Facebook likes for a given user name

The algorithm in Figure 3.3 can be explained in simple words like this. First Facebook API is executed with Actor/director/writer name. There are more than one matching user ids in the response JSON body for a given user name. So, there are multiple JSON objects with different user ids. Looping through each JSON object, there is a check to verify the username value which is coming from JSON object against the searched username. If contains, “fan_count” value of the relevant object is taken as the total Likes. Looping all the objects in result JSON array, the maximum like count is taken as the expected facebook likes for the given user.

Both Actor/director/writer name feature and his facebook like feature are used to show actor/director and writer. And facebook like attribute is used to represent the actor/director and writer information in numeric way. So, Actor/director/writer name feature can be removed from the dataset.

3.3.3 Movie Trailer

Before a movie is released, movie fans watch movie trailers. Assume someone is waiting a movie trailer. If he/she really likes the trailer, that person hits the like button, otherwise hits unlike button of the video. Also, that person might watch that trailer multiple times and share with others if the movie looks good. That shows movie trailer views, trailer likes and dislikes can indirectly affect the movie rating. Therefore, movie dataset should be enriched with movie trailer details as well. For that, Youtube API was used to retrieve movie trailer views, trailer likes and dislikes counts.

General format of Youtube API call to search movie for given search phrase is shown below.

https://www.googleapis.com/youtube/v3/search?part=id%2Csnippet&maxResults=25&order=relevance&q={URL encoded movie name + trailer}&type=Video&fields=items(id%2Fkind%2Cid%2FvideoId%2Csnippet%2Ftitle%2Csnippet%2Fthumbnails%2Fdefault%2Furl)&key={accesstoken}

Ex: Sample Youtube API call for a movie called “Logan” and output API response with video ids is shown below

Sample Youtube API call

https://www.googleapis.com/youtube/v3/search?part=id%2Csnippet&maxResults=25&order=relevance&q=logan%20trailer&type=Video&fields=items(id%2Fkind%2Cid%2FvideoId%2Csnippet%2Ftitle%2Csnippet%2Fthumbnails%2Fdefault%2Furl)&key=AIzaSyA61PT6tBhdVyhJmifm9TIFmU7kYwJsQ4k

Youtube API response


```
{
 "items": [
  {
   "id": {
    "kind": "youtube#video",
    "videoId": "DekuSxJgpbY"
   },
   "snippet": {
    "title": "Logan Trailer #2 (2017) | Movieclips Trailers",
    "thumbnails": {
     "default": {
      "url": "https://i.ytimg.com/vi/DekuSxJgpbY/default.jpg"
     }
```

```
   }
   }
  },
  {
   "id": {
    "kind": "youtube#video",
    "videoId": "Div0iP65aZo"
   },
   "snippet": {
    "title": "Logan | Official Trailer [HD] | 20th Century FOX",
    "thumbnails": {
     "default": {
      "url": "https://i.ytimg.com/vi/Div0iP65aZo/default.jpg"
     }
    }
   }
  },
  {
   "id": {
    "kind": "youtube#video",
    "videoId": "wZXWqzoMViQ"
   },
   "snippet": {
    "title": "Logan - Trailer Review",
    "thumbnails": {
     "default": {
      "url": "https://i.ytimg.com/vi/wZXWqzoMViQ/default.jpg"
     }
    }
   }
  }
 ]
}
```


In the Youtube API response there were multiple JSON objects with movies details came to search query. In response, there were some results which were made by fans and some results were full movie video details. Therefore, when each JSON objects was evaluated, following conditions were used to filter out the most suitable movie ids.

1. Video name of search result item is contained “Trailer” keyword.
2. Video name of search result item should not contained “FanMade” keyword.
3. Video name of search result item should not contained “Honest Trailers” keyword.

If a video is followed all above 3 condition, then another API was executed with the video Id to get the trailer information. Trailer with maximum like count is taken as the movie trailer.

General format of Youtube API call to get trailer information for a given movie id is shown below.

https://www.googleapis.com/youtube/v3/videos?id={videoid}&key={authorization token}&fields=items(id,snippet(channelId,title,categoryId),statistics)&part=snippet,statistics

Ex: Sample Youtube API call to get trailer information for a videoId called “Div0iP65aZo” and output API response with trailer information is shown below

https://www.googleapis.com/youtube/v3/videos?id=Div0iP65aZo&key=AIzaSyA61PT6tBhdVyhJmifm9TIFmU7kYwJsQ4k&fields=items(id,snippet(channelId,title,categoryId),statistics)&part=snippet,statistics.

Youtube API response


```
{
 "items": [
  {
   "id": "Div0iP65aZo",
   "snippet": {
    "channelId": "UC2-BeLxzUBSs0uSrmzWhJuQ",
    "title": "Logan | Official Trailer [HD] | 20th Century FOX",
    "categoryId": "1"
   },
   "statistics": {
    "viewCount": "27393616",
    "likeCount": "234506",
    "dislikeCount": "4080",
    "favoriteCount": "0",
    "commentCount": "32351"
   }
  }
 ]
}
```


3.3.4 Adjusting Movie budget

If a movie has so many technologies, most famous actors, best directors, then movie budget may be too high. Therefore, budget can affect the movie rating both direct and indirect ways. But, if movie is taken a cost of nine millions dollars in 1980, same movie will be taken more higher cost in 2017 due to cumulative inflation.

Cumulative inflation of dollar for last 100 years as shown in Figure 3.4 So, when comparing movies in different title years, if there is no adjustment to budget, the final model can be inaccurate. So, all the budgets in movie list were calculated compared to the one year(2017).

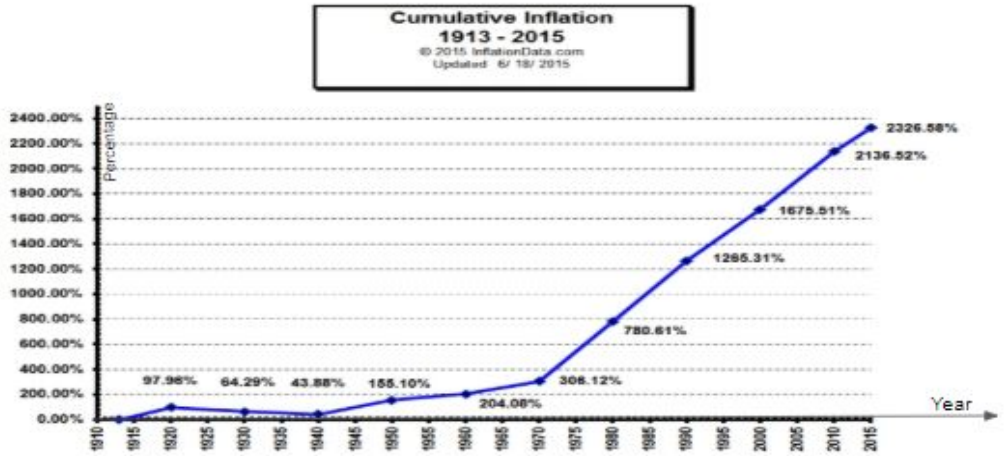

Figure 3.4: Cumulative Inflation for past 100 years

3.3.5 Movie Production Categorization

There were 200+ distinct values for the production company in the dataset. In there, same production company was in different names. And also, there are several children production companies under one parent production company[15]. So, production companies were grouped based on its parent company. All the production value were rearranged according to Table 3.2.

Updated Production	**Production name in dataset**
20th Century Fox	20th century fox twentieth century fox 21st century films fox 2000 pictures fox searchlight fox
Walt Disney Pictures	walt disney pictures walt disney productions walt disney studios walt disney home entertainment walt disney feature animation walt disney films touchstone pictures Pixar Miramax Disney buena vista dimension films
Warner Bros. Pictures	warner bros. Pictures warner bros warner brothers pictures warners bros. Pictures warner home video warner independent pictures warner brothers warner independent new line cinema hbo video hbo films new line home
Universal Pictures	universal pictures dreamworks animation universal studios Dreamworks

	focus world focus features universal film Universal mca universal
Columbia Pictures	columbia pictures sony pictures Tristar columbia trista sony
Paramount Pictures	paramount pictures mtv films paramount
MGM Holdings	mgm home entertainment united artists orion pictures Mirror mgm animation Mgm samuel goldwyn
Lions Gate Motion Picture Group	lions gate films pantelion films codeBlack films codeBlack entertainment lionsgate entertainment lionsgate films lionsgate pictures Lionsgate liongate films lions gate anchor bay films anchor bay entertainment summit entertainment artisan entertainment roadside attractions starz
The Weinstein Company	weinstein
AMC Networks	ifc films amc pictures sundance
Magnolia Pictures	magnolia pictures
Other	All other names

Table 3.2: Production companies and its child companies

After rearranging the production feature, there were 12 distinct productions as shown in Figure 3.5.

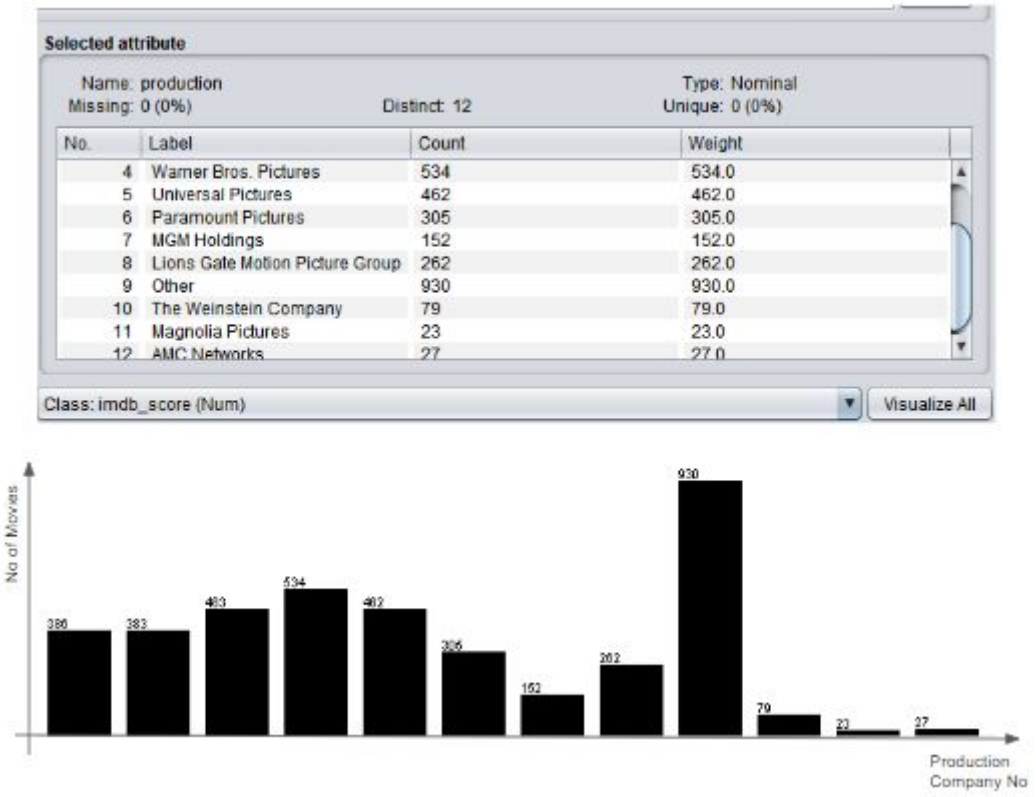

Figure 3.5: Production company distribution after rearranging production values

3.3.6 Movie Genres Categorization

In the initial dataset, there were 15+ distinct genres with low frequencies. And each movie had 1 to 4 genres in "|" separated format. For example, "Action| Crime| Drama". Even though this means the movie has multiple genres, for the classifier it is a single string. Therefore, due to the higher number of genre combinations, this feature had 900+ distinct values. But the classifier wasn't able to capture the real meaning.

So, A priority was assigned a priority to each genre and was picked only one for each movie, so that it makes sense for the classifier. The priority order was like this.

1. Animation
2. Comedy
3. Horror
4. Romance

5. Sci-Fi
6. Biography
7. Adventure
8. Action
9. Thriller
10. Documentary
11. Other

3.4 Algorithm Selection

From the initial dataset, 66% of data was taken as the training data and rest of data was used as testing data. As a basic performance comparison, following 6 classification was applied to training and tested data.

- **Nearest neighbours** was chosen since it is flexible to features and robust to noisy training data.
- **Decision tree J48** was chosen since it handles non-linear effects.
- **Random forest** was chosen due to reduction in overfitting.
- **Naive Bayes** was chosen as it is very simple and fast.
- **Linear regression** was chosen due to reduction in overfitting.
- **Logistic regression** was chosen due to more robust and handling non-linear effects .

Performance of above classification against the movie dataset are shown in Table 3.3.

Algorithm	**Accuracy**
Nearest Neighbours(IBk)	36.6373 %
Decision tree J48	42.4376 %
Random Forest	47.21 %
Naive Bayes	9.6916 %
Linear regression	18.12%
Logistic regression	41.63%

Table 3.3: Basic classification accuracy against the movie dataset

According to Table 3.3, Naive Bayes and Linear regression classifiers have the lower accuracy percentage. The accuracy of other four classifiers are nearly similar to 40%.

3.4.1 Post Optimization - Changing Genre format

In the analyzed dataset, only one genre was selected for each movie. Due to that, some important information regarding secondary genres were lost. That might be one of reason for the reduced accuracy of the classifications output. Therefore, it was required an update the dataset such that it represents all genres in each movies. So, original value of genre was transformed to multiple set of binary features.

Introduced features were :

- isAction
- isAnimation
- isAdventure
- isComedy
- isHorror
- isRomance
- isBiography
- isThriller
- isSci-Fi
- isDocumentary
- isDrama
- isMusic

Movie dataset with the multiple genre format is evaluated with different classifiers.

Algorithm	**Accuracy**
Nearest Neighbours	36.417 %
Decision tree J48	40.2349 %
Random Forest	47.5037 %
Naive Bayes	12.9956 %
Linear regression	19.54%

Logistic regression	43.3921 %

Table 3.4: Accuracy of classification with multiple genres types

Due to genre update, accuracy of 3 classifications were slightly increased. They were Naive Bayes, Linear regression and Logistic regression. Accuracy of Nearest neighbour and Random forest were not changed.

3.5 Normalization and Standardization

For rescaling data, two methods can be used[18]. They are:

- Normalization
- Standardization

3.5.1 Normalization

The mathematical function for the normalization is shown in Eq. (3.1).

$$x_{new} = \frac{x - x_{min}}{x_{max} - x_{min}} \qquad (3.1)$$

In simple words, we can use normalization to scale all numeric variables in the range of [0,1].

Normalization is useful when the data has varying scales, because that can lead to biased classifications.

Each classifiers were evaluated with normalized attributes. Output accuracy is shown in Table 3.5.

Algorithm	**Accuracy**
Nearest Neighbours	36.417 %
Decision tree J48	38.7665 %
Random Forest	47.21 %
Naive Bayes	12.9222 %

Table 3.5: Accuracy of classification with normalized data

There was no significant accuracy increment of classifiers after movie data were nominalized.

3.5.2 Standardization

The mathematical function for standardization is shown in Eq. (3.2).

$$x_{new} = \frac{x - \mu}{\sigma} \qquad (3.2)$$

In simple words, after applying standardization on the dataset, it will transform the mean to zero and variance to one.

Each classifiers are evaluated with standardized attributes. Output accuracy is shown in Table 3.6.

Algorithm	**Accuracy**
Nearest Neighbours	36.417 %
Decision tree J48	40.6755 %
Random Forest	46.4023 %
Naive Bayes	12.9956 %
Logistic regression	43.5389 %

Table 3.6: Accuracy of classification with standardized data

There was no significant accuracy increment of classifiers after movie data were standardized.

3.6 Feature Selection

There are 3 types of feature selection algorithms.[17]
They are :

- Filter methods
- Wrapper methods
- Embedded methods

3.6.1 Filter methods

Filter methods have several types of for selection methods. And also this is known as **single factor analysis**. The predictive power of each individual variable is evaluated. An

attribute evaluator and a ranker is used to rank all the features in the dataset. The number of features wanted to select from the feature vector can always be defined. Omit the features one at a time that have lower ranks and see the predictive accuracy of the classification algorithm.Weights put by the ranker algorithms are different than those by the classification algorithm.

Information gain attribute evaluator, Chi squared test are some examples to the filter methods.

Filter method is useful for feature selections of data mining tasks. In data mining, there can be thousands or millions of features and it's need to reduce the number of features to fifty. So in order to fo that filter method can be used.

InfoGainAttributeEval attribute evaluator with ranker search method was used to rank the features of the movie dataset. Ranking of the features in the dataset is shown in Table 3.7.

Feature name	**ranking**
actor_1_facebook_likes	0.27103
actor_3_facebook_likes	0.26763
actor_2_facebook_likes	0.26674
actor_4_facebook_likes	0.2662
duration	0.21506
budget	0.20047
director_facebook_likes	0.14068
production	0.12357
movie_trailer_likes	0.07541
isDrama	0.06645
movie_trailer_views	0.06154
isHorror	0.037

isBiography	0.03623
country	0.03526
movie_trailer_unlikes	0.02735
isComedy	0.02207
isAction	0.01125
isRomance	0.00697
isThriller	0.00619
isAdventure	0.00561
isDocumentary	0.0053
isAnimation	0.00293
isMusic	0.00265
isSci-Fi	0.0019

Table 3.7: Ranking of the features with InfoGainAttributeEval attribute evaluator

Each classifies are evaluated by removing one by one features from the lowest value.

- **Nearest Neighbours**

Nearest neighbour classifiers is evaluated with removing one by one features with lowest rank. Accuracy comparison is shown in Table 3.8.

Removed Feature	**Accuracy percentage**
With all features	36.417 %
isSci-Fi	36.931 %
isMusic	37.0044 %
isAnimation	36.931 %
isDocumentary	37.001%

isAdventure	36.931 %
isThriller	36.931 %
isRomance	36.1233 %
isAction	37.812 %
isComedy	36.4905 %
movie_trailer_unlikes	36.4905 %
country	35.8297 %
isBiography	35.022 %
isHorror	35.7562 %
movie_trailer_views	35.1689 %
isDrama	33.627 %
movie_trailer_likes	33.9207 %

Table 3.8: Nearest neighbour classifier accuracy changing with the features reduction

For, nearest neighbour classifier, when "**isAction"** feature is removed, its highest accuracy is received. The highest accuracy is 37.812%. So, for this classifier selected features are country, duration, budget, isDrama, isComedy, isBiography, isHorror, isDirector_facebook_likes, Actor_1_facebook_likes, Actor_2_facebook_likes, Actor_3_facebook_likes, Actor_4_facebook_likes, Movie_trailer_views, Movie_trailer_likes, Movie_trailer_unlikes, Production and ImdbScore.

- **Decision tree - J48**

J48 classifiers is evaluated with removing one by one features with lowest rank. Accuracy comparison is shown in Table 3.9.

Removed Feature	Accuracy percentage
With all the features	40.2349 %

isSci-Fi	40.5286 %
isMusic	41.4831 %
isAnimation	39.3539 %
isDocumentary	37.0044 %
isAdventure	36.931 %
isThriller	40.3818 %
isRomance	39.9413 %
isAction	39.6476 %
isComedy	39.6476 %
movie_trailer_unlikes	39.8678 %
country	38.9134 %
isBiography	40.6021 %
isHorror	39.4273 %
movie_trailer_views	39.5007 %
isDrama	39.721 %
movie_trailer_likes	38.9134 %

Table 3.9: J48 classifier accuracy changing with the features reduction

For, J48 classifier, when "**isMusic"** feature is removed, highest accuracy of the classifier is received. The highest accuracy percentage is 41.4831%. So, selected features are duration, budget, isDrama, isAnimation, isAction, isAdventure, isComedy, isBiography, isDocumentary, isHorror, isThriller, isRomance, isDirector_facebook_likes, Actor_1_facebook_likes, Actor_2_facebook_likes, Actor_3_facebook_likes, Actor_4_facebook_likes, Movie_trailer_views, Movie_trailer_likes, Movie_trailer_unlikes, Production and ImdbScore.

- **Random forest**

Random forest classifiers is evaluated with removing one by one features with lowest rank. Accuracy comparison is shown in Table 3.10.

Removed Feature	**Accuracy percentage**
With all the features	47.5037 %
isSci-Fi	45.815 %
isMusic	45.1542 %
isAnimation	45.5213 %
isDocumentary	46.5492 %
isAdventure	45.0073 %
isThriller	45.815 %
isRomance	43.9794 %
isAction	45.4479 %
isComedy	45.815 %
movie_trailer_unlikes	44.42 %
country	45.0073 %
isBiography	44.5668 %
isHorror	44.3465 %
movie_trailer_views	45.301 %
isDrama	44.2731 %
movie_trailer_likes	42.8781 %

Table 3.10: Random forest classifier accuracy changing with the features reduction

For, random forest classifier, when the classifier is evaluated with all the features its highest accuracy is received. The highest accuracy is 47.5037%.

- **Naive Bayes**

Naive bayes classifiers is evaluated with removing one by one features with lowest rank. Accuracy comparison is shown in Table 3.11.

Removed Feature	**Accuracy percentage**
With all the features	12.9956 %
isSci-Fi	12.5551 %
isMusic	12.4816 %
isAnimation	12.4082 %
isDocumentary	12.2614 %
isAdventure	12.1145 %
isThriller	12.188 %
isRomance	11.9677 %
isAction	11.9677 %
isComedy	11.7474 %
movie_trailer_unlikes	13.1424 %
country	12.9956 %
isBiography	13.069 %
isHorror	13.583 %
movie_trailer_views	16.9604 %
isDrama	16.5932 %

movie_trailer_likes	15.5653 %

Table 3.11: Naive Bayes classifier accuracy changing with the features reduction

For, Naive bayes classifier, when "**movie_trailer_views**" feature is removed, the classifiers' highest accuracy is received. The accuracy percentage is 16.9604%. So, selected features are duration, budget, isDrama, isDirector_facebook_likes, Actor_1_facebook_likes, Actor_2_facebook_likes, Actor_3_facebook_likes, Actor_4_facebook_likes, Movie_trailer_views, Movie_trailer_likes, Production and ImdbScore.

Even better accuracy can be get after removing features using the ranking algorithm, but it can lead to overfit the model.

3.6.2 Wrapper methods

Wrapper methods consider the selection of a set of features as a search problem, where different combinations are prepared, evaluated and compared to other combinations. Wrapper methods are useful for feature selection for machine learning tasks. In machine learning test, we are aware of the features that are really prominent. In that case all the wrapper method can be used to find out the best subset of features that performs the best with a particular classification algorithm. Selected search techniques are defend on time and memory can used.

ClassifierSubsetEval with BestFirst search technique is one example for wrapper methods.

After evaluating the dataset with ClassifierSubsetEval evaluator with BestFirst search technique, following features are selected.

- Budget
- isDrama
- Director_facebook_likes
- Actor_1_facebook_likes
- Actor_2_facebook_likes
- Actor_3_facebook_likes
- Actor_4_facebook_likes
- Movie_trailer_likes

- Imdb rate

After that, classifiers were evaluated with above 8 features.

Naive bayes classifiers is evaluated with removing one by one features with lowest rank. Accuracy comparison is shown in Table 3.12.

Algorithm	**Accuracy**
Nearest Neighbours	35.1689 %
Decision tree J48	38.3994 %
Random Forest	43.9794 %
Naive Bayes	6.7548 %
Logistic regression	35.0954 %

Table 3.12: Comparison between classifier accuracy after feature reduction from ClassifierSubsetEval

- **Embedded methods**

Embedded methods are also called as selling case methods. Also call this a regularization regression model. This is a technique that regularize the estimates or shrink the coefficient towards zero.

3.7 Improving Accuracy of classification

3.7.1 Improving accuracy of J48

Three methods were used to check the possibility of improving the accuracy of J48.

1. Find confidenceFactor and minNumObj so that, algorithm gives a better performance.
2. Apply Bagging with J48 classifier.
3. Apply AdaBoost with J48 classifier.

Each of methods has described below.

- **Find better values for confidenceFactor and minNumObj**

Cross-validated Parameter selection(CVParameterSelection)[20] is used to get the most accurate confidenceFactor(C) and minNumObj(M) values.

First, C and M were chose with the following conditions.

- M - 10 values from 1.0 to 10.0 (1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0)
- C - 9 values from 0.1 to 0.9 (0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9)

Then the dataset was evaluated with J48 classifier with above values. When C equals to 0.1 and M equals to 10, the best accuracy percentage was received. Accuracy percentage was 45.2072%.

Both C and M were selected from the CVParameterSelection, were edges of selected range. If the value range is changed, output C and M values might be changed. So, the range of C and M was changed and CVParameterSelection was executed again.

At this time C and M were chosen like below.

- M - 20 values from 0.0 to 20.0
- C - 19 values from 0.05 to 0.95

After that, the dataset was evaluated with J48 with above values. When C equals 0.1 and M equals to 14, the best accuracy percentage was received. Accuracy percentage was 45.8063 %.

- **Apply Bagging with J48**

Bagging is using with decision trees and it helps to reduce the variance and avoid the overfitting. Bagging stands for Bootstrap aggregation. When there is a sample and sub samples are taken repeatedly in place called as bootstrapping. A model is trained on each of the bootstrap samples and then aggregated models of the all sample models are return as the final model is called as Bagging.

When Bagging with J48 classifier evaluated with the dataset, the accuracy percentage was increased. Accuracy percentage is 48.4581 %.

- **Apply Boosting with J48**

Boosting uses ensemble techniques to create a sequence of increasingly complex predictors out of building block make out of very simple predictors. It helps to reduce bias and variance. In weka, there is AdaBoost and it stands for adaptive boosting.

In AdaBoost, initially a simple classifier(ex:- J48) has been fitted on the data which splits the data into just two regions like shown in Figure 3.6. That is iteration one. Whatever the class is correctly classified in iteration 01 will be given less weight edge in the in iteration two. In that iteration, higher edge from misclassified class and again another weak classifier will be fitted on the data. It will change the weight again for the iteration three.

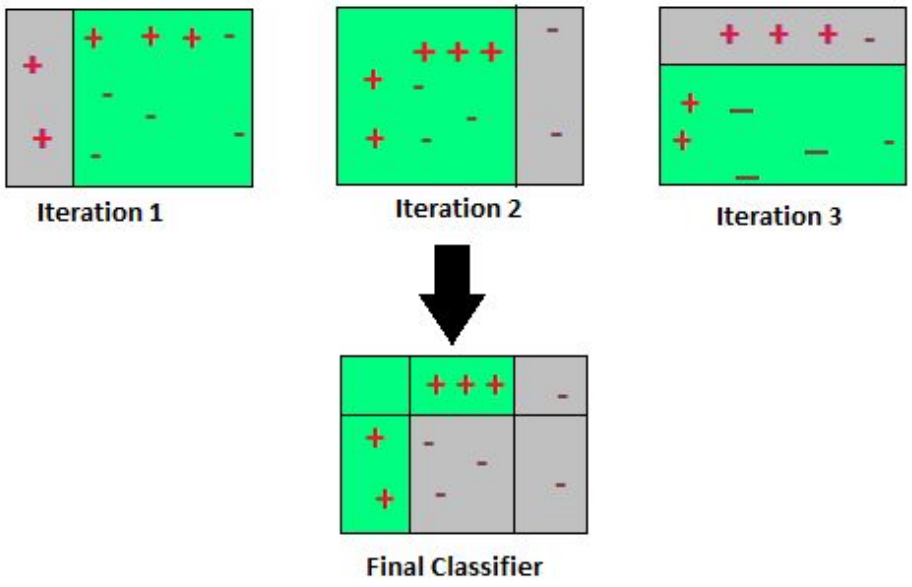

Figure 3.6: How works AdaBoostM1

In above diagrams, some minus symbols' weight has been increased once it finishes the iterations. Weights are automatically calculated for each classifier at each classifier at each iteration base on the error rate. After all the iterations, final classifier is a strong classifier which predicts the class with better accuracy.

The movie dataset was evaluated with AdaBoost algorithm with J48 classifier and accuracy was decreased slightly. Accuracy percentage is 45.5067%.

Table 3.13 contains the accuracy comparison for the above 3 methods.

	J48(C=0.1 and M = 14)	J48 with Bagging	J48 with Boosting
Accuracy percentage	45.8063 %	48.4581 %	45.5067%

Table 3.13: Accuracy Comparison of J48 classifier

3.7.2 Improving accuracy of K-Nearest neighbour classifier(IBk)

KNN algorithm accuracy is depend on the distance between nodes. Therefore, calculating the distance between nodes might be one reason to the accuracy of KNN classifier. There are several methods to get the distance between nodes in KNN. Two of them are Euclidean distance and Manhattan distance

- **Euclidean distance**

Euclidean distance means the straight line distance between two points which are lies in euclidean space. The Euclidean distance between p and q points is the length of the line segment connecting them (Figure 3.7).

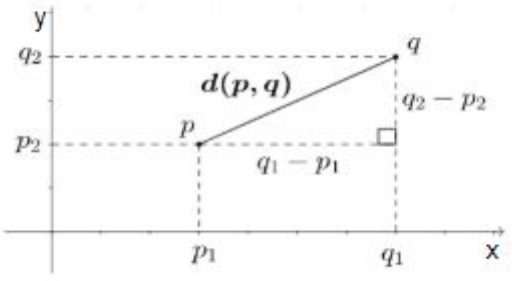

Figure 3.7: Euclidean distance

The mathematical formula to get the Euclidean distance between p and q which lies in a N dimension space is in Eq (3.3)

$$d(\mathbf{p}, \mathbf{q}) = d(\mathbf{q}, \mathbf{p}) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2}$$

$$= \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}.$$

(3.3)

When KNN classifier was evaluated with Euclidean distance function, the output accuracy was 36.417%.

- **Manhattan distance**

Manhattan distance is the distance between the projection of the points on the axis.
If (x1,x2) and (y1,y2) are points, then Manhattan distance is |x1-x2| + |y1-y2|
When KNN classifier was evaluated with Manhattan distance function, the output accuracy is 36.8576 %.

When KNN classifier is evaluated with Manhattan distance function, accuracy is better comparing to the classifier with Euclidean distance.

3.8 Attribute values Reduction

3.8.1 Reduce no of different class attributes.

In previous analyses, there were 10 class attributes. They were taken by rounding to the most closest upper integer value(As shown in the Table 3.1). Therefore, if a movies' actual IMDb rating is 6.4, it's class variable should be 7. If that movie rate was tested from the movie rating predicting model and output rating was received as 6, then that value was taken as an incorrect value. But the predicted rating is roughly correct. Therefore, if movie rating can be categorized into a few types instead of 10 attributes, the accuracy of output model might be increased. If no of class attributes can be reduced from a meaningful way, the accuracy of the evaluating algorithms might be increased.

Movie rating categories are reduced to 3 different types.

They are,

- If the movie Imdb rating is higher than 7.1, movie rating is changed as "HIGH".
- If the movie Imdb rating is between 5.1 and 7, movie rating is changed is "MEDIUM"

- If the movie Imdb rating is lower than 5, movie rating is changed is “LOW”

- **Accuracy Comparison for the dataset with 3 class attributes and single genre for a given movie**.

Classifier	**Accuracy percentage**
Nearest Neighbours - IBk	49.2658 %
Decision tree J48	58.0029 %
Random Forest	61.2335 %
Naive Bayes	27.9001 %
Logistic regression	60.793 %

Table 3.14: Accuracy Comparison for the classifier with 3 class attributes and single genre

According to the Table 3.14, the imdb rate prediction accuracy of each selected algorithm was increased from a significant value.

- **Accuracy Comparison for the dataset with 3 class attributes and multiple genres for a given movie.**

Classifier	**Accuracy percentage**
Nearest Neighbours - IBk	55.9471 %
Decision tree J48	62.9956 %
Random Forest	67.6946 %
Naive Bayes	36.3436 %
Logistic regression	63.2438 %

Table 3.15: Accuracy Comparison for the classifier with 3 class attributes and multiple genre

:

According to the Table 3.15, the imdb rate prediction accuracy of each selected algorithms were increased. And also, accuracy percentages of every algorithms were higher than when the same movies has one selected genre only.

3.6.2 Reduce no of different Instance attributes with equal frequency.

- **Accuracy Comparison for the dataset with single genres for a given movie.**

Some instance attribute (Budget, director facebook likes, actors facebook likes) ranges were too big. Minimum value of budget for the selected dataset was 200 and maximum value of budget for the same dataset was 14536445000. And the number of distinct values were also too high, which can make the output model overfit to the given dataset. Therefore, Budget, director facebook likes and actors facebook likes features were categorized into 3 categories with equal frequencies. And, there were only one value to the genre attribute for a selected movie. Accuracy comparison for the movie dataset with categorized instance attributes is shown in Table 3.16

Classifier	**Accuracy percentage**
Nearest Neighbours - IBk	49.1189 %
Decision tree J48	64.978 %
Random Forest	61.674 %
Naive Bayes	41.7034 %
Logistic regression	64.4361 %

Table 3.16: Accuracy Comparison for the classifier with reduced instance attribute values and single genre

After reducing the no of different class attributes into three and instance attributes are categorized, the IMDb rate prediction accuracy of each selected algorithm was increased from a significant value than the dataset set with uncategorized instance attributes.

- **Accuracy Comparison for the dataset with multiple genres for a given movie.**

Classifier	Accuracy percentage
Nearest Neighbours - IBk	57.1953 %
Decision tree J48	65.2893 %
Random Forest	58.0764 %
Naive Bayes	62.4816 %
Logistic regression	64.8135 %

Table 3.17: Accuracy Comparison for the classifier with reduced instance attribute values and multiple genres

According to Table 3.17, the IMDb rate prediction accuracy of each selected algorithms were increased. If the dataset has 3 different values for the class attribute and some instance attributes are categorized, the accuracy percentage of the dataset from Naive bayes algorithm is increased significantly. It is nearly 2 times big accuracy than the accuracy of the dataset with uncategorized instance attributes.

3.6.3 Reduce no of different Instance attributes with defined ranges.

Algorithm	Accuracy percentage
Nearest Neighbours - IBk	57.4156 %
Decision tree J48	64.5374 %
Random Forest	66.3869 %
Naive Bayes	63.8032 %
Logistic regression	65.1351 %

Table 3.18: Accuracy Comparison for the classifier with reduced predefined instance attribute values

According to Table 3.18, when instance attributes are categorized with defined ranges, all classifiers accuracy percentage is increased. Random forest, Naive Bayes and Logistic regression classifiers accuracy is increased from a significant value.

When applying Bagging classifier with J48 to the same dataset, the accuracy percentage increased to 66.8135 %.

Accuracy comparison of each classification for the same dataset with standardization has shown in Table 3.19.

Algorithm	**Accuracy percentage**
Nearest Neighbours - IBk	57.4156 %
Decision tree J48	65.0514 %
Random Forest	66.7401 %
Naive Bayes	63.7298 %
Logistic regression	66.8135 %

Table 3.19: Accuracy Comparison for the classifier with reduced class variables and standardization

Features are selected from the CfsSubsetEval evaluator and then accuracy was evaluated of each classifiers. But accuracy of each classifier is lower than above values. After applying bagging with J48, 67.6946 % of accuracy percentage was received. With AdaBoostM1 classifier, 67.0152 % of accuracy percentage received. After analysing the dataset, the more accurate percentage is received to Bagging with J48 classifier and its value is 67.6946 %. C and M values for this occasion is 0.01 and 2.

Chapter 4: Proposed Solution

4.1 Selection of a better classifier

Based on the analysis done in Chapter 03, the best accuracy percentage was 67.6946% and it was given by J48 classifier with bagging.

4.2 Introduction to the solution

Before applying the J48 classifier with bagging, the dataset and features were updated to get the maximum accuracy without overfitting. First, the class variable which was the Movie ranking was split into 3 bins, namely HIGH, MEDIUM, and LOW. Actors' facebook likes and Movie budget were also split into 5 categories, VERY-HIGH, HIGH, MEDIUM, LOW, and VERY-LOW.

Genre combinations which were in the original dataset were converted into separate binary features. Duration, Trailer views, Trailer likes and dislikes were taken as integers and they were standardized, while Production company was taken as a nominal value. When standardizing, the standard deviation and mean of the dataset is recorded.

Then the J48 classifier with bagging was applied to the dataset with ConfidenceFactor=0.01 and MinNumObj=2, which gave the highest accuracy. Then the model was saved to predict the rating of the test data.

When applying test data to the model, values such as Duration, Trailer views, Trailer likes and dislikes etc. were updated as per the recorded standard deviation and mean of the original dataset.

Chapter 5: Evaluation and Results

Around 50 movies from IMDb was taken randomly to evaluate the prediction model. When getting the movie data, they was taken such that their ratings covered most of the movie ratings. When getting movies, those movies was chosen if and only if they have more than 25000 number of votes. This was done to ensure the credibility of the rating. In the very initial stage of a movie, the movie rating may be very high or very low. But those may not be accurate due to the low number of votes. With a large numbers of votes, IMDb rate of movies will be more accurate. Because of those reasons, when picking movies to evaluate the model, movies with more than 25000 of votes was picked only. Those is the ground truth data for the evaluation.

In addition to the IMDb data, some data was taken from other sources as well. For example, number of facebook likes was taken from the facebook pages, and movie trailer likes was taken from Youtube. To make sure the data taken from those other sources are valid and credible, data was taken from verified facebook pages and official movie trailers in Youtube only. There were cases that I couldn't find verified pages. In such cases, I tried to find the most matching pages with higher likes.

When evaluating the model, for every selected movie, there was a predicted movie rating and an actual rating which comes from IMDb data. So, by comparing those two values, the prediction model could be evaluated.

When evaluating the model with sample 50 movie, 54.6102% accuracy percentage was received.

Chapter 6: Conclusion and Future Works

6.1 Conclusion

In today's world, movies are released rapidly in all around the world. Therefore, in a particular city, there is more than one movie in theatres at a time. But people have a busy tight schedules in their life. So, they don't have enough time to watch each and every movie. Also, due to higher cost of living, people can't afford to watch all movies in theatres. So, people always try to find the better movies which are worth watching in a movie theatre. Therefore, they try to look for recommendations and non-spoiler reviews from other people.

As a solution for this problem, this research project proposes a model. This model was created by analyzing existing movie data, extracting features from it and identifying a relation between the features and the movie rating.

The original dataset has some unwanted data, as well as missing data. The former was fixed by cleaning up the dataset and the latter was fixed by calling external APIs such as OMDb API, Facebook API and Youtube API etc. In addition to that, most of available features were modified as well. For example, some feature had huge amount of unique values with a few occurrences. Since it can lead to overfitting, values of these features were split into a few bins. That increased the accuracy of the model.

Multiple classifier algorithms were evaluated before developing the model and at the end, J48 decision tree algorithm with bagging was used as it gave the best results (48.4581%). At this point, the output of the prediction model (i.e. the movie ranking) was given as an integer number between 1 to 10. However, in the real world, a person who's interested in knowing whether a movies is good or not, does not expect such an accuracy. A scale of "Great", "Ok" and "Poor" would be a good enough measure for this.

Taking this fact into consideration, the model was updated to output only 3 values for the rating, namely "High", "Medium" and "Low" which gave an accuracy of "67.6946%". This is a little good accuracy, but it can be improved as well. Some methods which can be used to improve the model are discussed in the next sub chapter.

6.2 Future Works

Since some years now, most of movies come with both 2D and 3D format. Nowadays, if a movie fan is going to watch a movie to the cinema and, if that movie has both 2D and 3D format, selecting 3D format is the trend. Therefore, a movie has 3D format might be highly depend to the movie rating. So, in future, when creating a data model to predict the IMDb rating of upcoming movie's rating, "Has3DFormat"feature can be considered a new feature.

The output model can be implemented as a automated tool such that it is more user friendly. The tool which implement from this project need to add a lot of user inputs to predict a movie rating. But in future, movie rating can be predicted entering the movie name only. Then there can be a google search inside the tool and there should be a way to load the relevant IMDb Page. In that page, there is all the information like, movie title with year, genres, run time etc related to movie like in Figure 6.1. For every html field of that page there is a key. Therefore, the wanted data when predicting a movie can be read by using the HTML keys in IMDb page. So, all the wanted data to predict the rate can be read by the implementation itself. Then searched movie data should be the input to the movie prediction model from the implementation. IMDb rate should be final output from the implementation. If there is this kind of application in future, it will be a big trend which is using to search the movie rating before watching it.

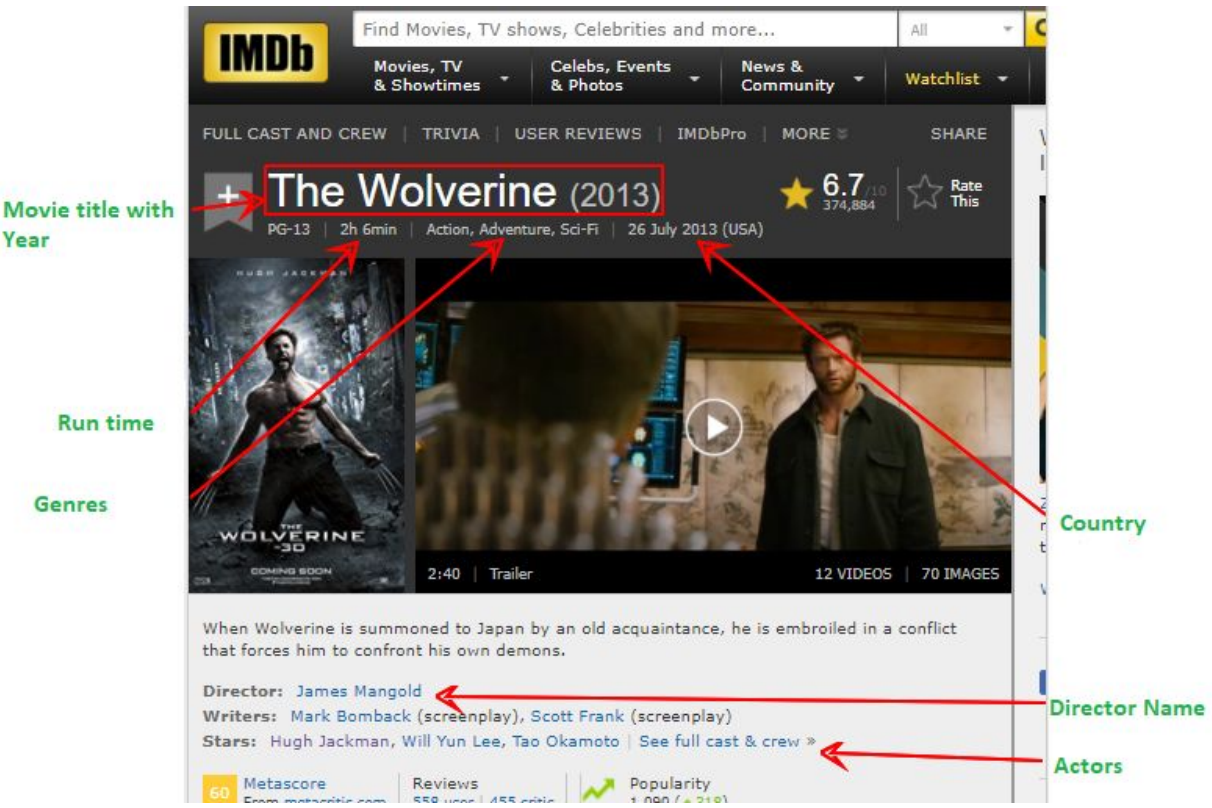

Figure 6.1: IMDb page with movie information

References

[1] IMDb. (2017). IMDb - Movies, TV and Celebrities. [online] Available at: http://www.imdb.com [Accessed 19 Jun. 2017].

[2] Rottentomatoes.com. (2017). Rotten Tomatoes: Movies | TV Shows | Movie Trailers | Reviews. [online] Available at: https://www.rottentomatoes.com/ [Accessed 1 Jul. 2017].

[3] Ebert, R. (2017). Movie Reviews and Ratings by Film Critic Roger Ebert | Roger Ebert. [online] Rogerebert.com. Available at: http://www.rogerebert.com/ [Accessed 2 Jul. 2017].

[4] The Guardian. (2017). Film + Reviews | Film | The Guardian. [online] Available at: https://www.theguardian.com/film+tone/reviews [Accessed 2 Jul. 2017].

[5] Metacritic.com. (2017). Metacritic - Movie Reviews, TV Reviews, Game Reviews, and Music Reviews. [online] Available at: http://www.metacritic.com/ [Accessed 7 Jul. 2017].

[6] Alexa Internet. (2017). Keyword Research, Competitor Analysis, & Website Ranking | Alexa. [online] Available at: http://www.alexa.com/ [Accessed 20 Apr. 2017].

[7] Nithin, V., Pranav, M., Sarath Babu, P. and Lijiya, A. (2014). Predicting Movie Success Based on IMDB Data. International Journal of Data Mining Techniques and Applications, [online] 03(2278-2419), pp.365-368. Available at: https://www.researchgate.net/publication/282133920_Predicting_Movie_Success_Based_on_IMDB_Data [Accessed 7 May 2017].

[8] Cohen, J., Cohen, P., West, S. and Aiken, L. (2003). Applied multiple regression/correlation analysis for the behavioral sciences. 2nd ed. Hillsdale, NJ: Lawrence Erlbaum Associates.

[9] Deniz Demir, Olga Kapralova, Hongze Lai, "Predicting IMDB Movie Ratings Using Google Trends," Dept.Elect.Eng,Stanford Univ., California, December, 2012

[10] Yashodhan, K. (2015). Movie Rating Prediction using the MovieLens dataset. [online] Available at: https://cseweb.ucsd.edu/~jmcauley/cse255/reports/wi15/Yashodhan_Karandikar.pdf [Accessed 15 Jun. 2017].

[11] Cohen, J., Cohen, P., West, S. and Aiken, L. (2003). Applied multiple regression/correlation analysis for the behavioral sciences. 2nd ed. Hillsdale, NJ: Lawrence Erlbaum Associates.

[12] Pasricha, R. (2015). Rating Prediction for Amazon Movies. [online] Available at: https://cseweb.ucsd.edu/~jmcauley/cse255/reports/fa15/025.pdf.

[13] Kaggle.com. (2018). Kaggle: Your Home for Data Science. [online] Available at: https://www.kaggle.com [Accessed 1 Feb. 2018].

[14] Yutianxin.com. (2018). On predicting the movie ratings. [online] Available at: http://yutianxin.com/static/portfolio/files/movie_box_office_machine_learning.pdf [Accessed 1 Feb. 2018].

[15]. Omdbapi.com. (2018). OMDb API - The Open Movie Database. [online] Available at: http://www.omdbapi.com/ [Accessed 6 Feb. 2018].

[16] Omdbapi.com. (2018). OMDb API - The Open Movie Database. [online] Available at: http://www.omdbapi.com/ [Accessed 6 Feb. 2018].

[17] J. Brownlee, "An Introduction to Feature Selection - Machine Learning Mastery", Machine Learning Mastery, 2018. [Online]. Available: https://machinelearningmastery.com/an-introduction-to-feature-selection/. [Accessed: 17- Mar- 2018].

[18] "Standardization vs. normalization | Data Mining Blog - www.dataminingblog.com", Dataminingblog.com, 2018. [Online]. Available: http://www.dataminingblog.com/standardization-vs-normalization/. [Accessed: 18- Mar- 2018].

[19] P. Scheuring, D. Purcell, W. Miller, A. Nolasco and D. Riots, "Prison Break (TV Series 2005–2017)", IMDb, 2018. [Online]. Available: http://www.imdb.com/title/tt0455275/. [Accessed: 25- Mar- 2018].

[20] "weka - Optimizing parameters", Weka.wikispaces.com, 2018. [Online]. Available: https://weka.wikispaces.com/Optimizing+parameters. [Accessed: 13- Mar- 2018].