**IMDB SCORE PREDICTION PROJECT**

**INTRODUCTION**

Film industry has been expanded worldwide and now-a- days people, wherever on earth, have an opportunity to watch a movie on the very first day it is released. There is a huge sector behind the preparation phases of each film and lots of directors and movie stars have burst. Consistently every year hundreds of movies are being produced. These films have different or various types of genres, varying from comedy to romance or war to science fiction. In order to monitor and keep a track of every movie produced, an online platform was very much needed. Internet Movie Database (IMDB) is the most well-known and a popular platform to get information about a rich collection of movies. IMDB web site contains downloadable raw data about the movies, including data like cast, directors, genres, crew, scriptwriters, summaries, gross and even user ratings. This information is used for data mining on the films for making prediction on user ratings of the movies.

**BACKGROUND**

A commercial success film not only entertains its audience or viewers, but also facilitates movie companies to gain tremendous profit. A ton of factors for instance good directors, experienced actors are considerable for making good movies. However, famous directors and actors can always bring an expected box-office income but cannot ensure a highly rated imdb score. In this project, we attempt to use the IMDb dataset to predict what are the important factors that make a movie more successful than others using Data mining techniques such as Decision tree & K-NN. We take IMDB scores as response variable and focus on operating predictions by analyzing the other variables in the IMDB 5000 movie data. The results can help movie companies to understand the secret of generating a commercial success movie. So, we would like to analyze what kind of movies which are more successful. We would like to show the results of this analysis in an intuitive way by data visualization.

**Challenges experienced and how these were resolved**

Challenges which we faced are

* To identify the columns which don’t contribute to IMDB score. We resolved it by data visualization techniques like histograms, correlation plots, heat maps etc. to easily identify the patterns and to know to what extent the attributes have an influence on the IMDB score.
* Our goal is to build a model, which can help us predict if a movie is good or bad. Therefore, we don’t really want an exact score to be predicted. We resolved this challenge by categorizing the score into 4 different buckets like less than 4 represents bad movies,4~6 which represents average/ok movies, 6~8 corresponds to good movie, 8~10 corresponds to excellent movies respectively.
* We faced challenges in setting a seed value in algorithms (Classification Tree and K-N-N) as with different seed values the results are getting changed. We resolved this issue by trial and error methods (setting different seed values) to identify the rules and accuracy.
* Post cleaning the data we found the data disorganized and hard to understand. We tried to reorder the columns to make the dataset easier to be understood and also renamed the columns to make the names shorter.
* Faced challenges in knitting r-code. We resolved it by trying out different commands and changing global options, but finally could resolved it re-installing r-studio.
* Had to install multiple visualization packages. For the data exploration, we had installed different packages such as ggplot2, ggthemes, scales etc. for the data visualization so that pattern can be easily understood.

**Implementation**

There are different stages

* Data exploration
* Data cleaning
* Data visualization
* Data pre-processing
* Implementing algorithms
* Calculate & compare accuracy values

In data exploration, the data set is loaded into a variable using read.csv command. The data may consists of duplicate rows and unnecessary blank spaces or letters. The duplicated rows are removed and movie titles are cleaned. we want to know if genre is related to IMDB score. The string is divided into several substrings by the separator ‘|’, and save each substring along with its corresponding imdb score in the other data frame genres.df. Then a histogram is plotted for the score and genres to see if they are relative or not.

In data cleaning, the missing values are analysed by heatmap. Gross and budget has highest missing values so rows with missing values are deleted. Then the importance of the other variables with next highest missing values is checked by checking the means of the values present in that variable. The variable is deleted if the mean falls in similar range. There are some 0 values which should also be regarded as missing values. First NA is replaced with column average for facenumber\_in\_poster, then replace 0s in other predictors with NA, and lastly replace all NAs with their respective column mean. There are still some missing values which are marked as “ ”. The content ratings are replaced by 5 categories. for further analysis two colums are added : profit (gross – budget) and percentage return on investment (profit\*100/budget). In the dataset there are many columns that are irrelevent to the analysis. So columns like color and language are removed. In the variable country most of then are from usa and uk, so all others are grouped into a vaibale others.

In data visualization, histogram is generated based on year of the film and there aren’t many records of movies released before 1980. It’s better to remove those records because they might not be representative. Another thing that can be inferred from the top 20 movies based on the Profit earned plot, is that high budget movies tend to earn more profit. The trend is almost linear, with profit increasing with the increase in budget. Since profit earned by a movie does not give a clear picture about its monetary success over the years, the top 20 movies based on its Percentage Return on Investment analysis would provide better results. As hypothesized, the ROI is high for Low Budget Films and decreases as the budget of the movie increase. Analysis on the Commercial Success acclaimed by the movie (Gross earnings and profit earned) v.s. its IMDB Score says that there is not much correlation since most critically acclaimed movies do not do much well commercially. Movie with extremely high Facebook likes tend to have higher imdb score. But the score for movie with low Facebook likes vary in a very wide range can be analysed from a scatter plot.

In data preprocessing, some variables are removed. We have 1660 directors, and 3621 actors in this data. Since all the names are so different for the whole dataset, there is no point to use names to predict score. Same with plot keywords, they are too diverse to be used in the prediction and movie link is also a redundant variable. For the purpose of data exploration, two variables are added based on existing variables: profit and return\_on\_investment\_perc. In order to avoid multicollinearity, these two added variables are removed. Based on the heatmap, there are some high correlations (greater than 0.7) between predictors. According to the highest correlation value 0.95, actor\_1\_facebook\_likes is highly correlated with the cast\_total\_facebook\_likes, and both actor2 and actor3 are also somehow correlated to the total. So they are modifed into two variables: actor\_1\_facebook\_likes and other\_actors\_facebook\_likes. There are high correlations among num\_voted\_users, num\_user\_for\_reviews and num\_critic\_for\_reviews. Our goal is to build a model, which can help us predict if a movie is good or bad. So we don’t really want an exact score to be predicted, we only want to know how good or how bad is the movie. Therefore, the score is binned into 4 buckets: less than 4, 4~6, 6~8 and 8~10, which represents bad, OK, good and excellent respectively. Columns are reordered to make the dataset easier to be understood. And also the columns are renamed to make the names shorter. The data is split into training, validation and test sets with the ratio of 6:2:2.

In the next step, algorithms are applied. For a full-grown tree, a plot is generated along with classification rules. From these rules, we can conclude that movies with a lot of votes in imdb website tend to have a higher score, which really makes sense because popular movies will have a lot of fans to vote high scores for them. On the contrary, if a movie has fewer votes, it can still be a good movie if its duration is longer. It is surprise to see that movies make less profit are good, but ok if they make more profit. Best pruned tree is a cross-validation procedure. We set a smallest value as a complexity parameter. Using the variables, we construct a tree and prune it by the lowest complexity parameter. The model is applied on training set and validation set, then a confusion matrix is generated. Similarly, the model is applied to test data and a confusion matrix is generated to determine the accuracy.

In K-NN classification, the data is prepared. Dummy variables are required for categorical variables. So, a copy of data is used so that original data can be used in future. Useful variables are selected and data is partitioned into training and validation sets. Then, the data is normalized. We will set k as 1 to 20, and build 20 different models. We calculate each model’s classification accuracy, and find the best k according to the highest accuracy. The model is applied on test set and a confusion matrix is generated to determine accuracy.

**Results**

From our analysis, below are the results

* We found no much difference in the averages of imdb score related to different genres, almost all the averages are in the same range of 6~8. Therefore, we thought that the predictor “genres” could be removed because it’s not really related to the score. (Obtained from 4 from implementation part).
* Removed Aspect ratio attribute as it does not have much influence on IMDB Score.
* Color of the movie,language,names of the directors,actors,plot keywords, movie link does not have much influence.
* Movie with extremely high Facebook likes tend to have higher imdb score.
* Classification rules:

1. If (user\_vote >= 541000) then class = (8,10].
2. If (84000 <= user\_vote < 541000) then class = (6,8].
3. If (user\_vote < 84000) and (duration > 103) then class = (6,8].
4. If (user\_vote < 84000) and (duration < 103) and (budget <=8300000) then class = (6,8].
5. If (user\_vote < 84000) and (duration < 103) and (budget >=8300000) and (user\_vote < 34000) then class = (4,6].
6. If (user\_vote < 84000) and (duration < 103) and (budget >=8300000) and (user\_vote > 34000) and (gross <= 17000000) then class = (6,8].
7. If (user\_vote < 84000) and (duration < 103) and (budget >=8300000) and (user\_vote > 34000) and (gross >= 17000000) and (critic\_review\_ratio < 1.5) then class = (4,6].
8. If (user\_vote < 41000) and (duration < 103) and (budget >=8300000) and (user\_vote > 34000) and (gross >= 17000000) and (critic\_review\_ratio > 1.5) then class = (6,8].

* From these rules, we can conclude that movies with a lot of votes on imdb website tend to have a higher score, which really makes sense because popular movies will have a lot of fans to vote high scores for them. On the contrary, if a movie has fewer votes, it can still be a good movie if its duration is shorter (rule #3). It is surprising to see that movies make less profit are good, but ok if they make more profit (rule #4).
* Accuracy table for different models:

|  |  |  |
| --- | --- | --- |
| Dataset | Decision Tree | K-NN |
| Training | 0.753 |  |
| Validation | 0.724 | 0.720 |
| Test | 0.708 | 0.686 |

* For Decision tree model, we have a higher accuracy for training data because the tree was built based on the training data.
* Based on the overall performance, we find the best model is decision tree, which gives a high accuracy around 0.753.