Analyze IMDB score using Data Mining Algorithms

Amrit Mohan Kaushik
amrit18022@iiitd.ac.in
M.Tech. CSE - 2018
Indraprastha Institute of Information & Technology
Course: Data Minning

***Abstract*— The success of a movie entertains the audience and also helps the movie makers to gain high profits. The success can be measured by the ratings given to that movie by the critics. IMDB is the most famous critic portal which rates the movie on the basis of different features like Facebook likes, actors, directors, etc. This IMDB score is useful for others to decide how the movie is. Here, this report is aimed to analyze that IMDB scores only.**

I. Introduction

This project is aimed to analyze the IMDB score using data mining algorithms. The success of movies not only entertains the audience but also makes film companies to gain a huge amount of profit. A lot of factors such as good directors, experienced actors are considered for creating good movies. However, famous directors and actors can make good earning on box office but they do not help to get a high IMDB score. IMDB score is a trustworthy score to judge a movie cause it considers a lot of important features and not only focus on how many people are liking it or not. Now let's discuss some overview of the dataset.

A. Overview of The Dataset

The dataset contains some information related to 5000 movies. This data spans over 100 years and 66 countries. It has 28 different features for each movie. IMDB score is the target or response variable and 27 attributes are used as features to predict the IMDB scores. The data have features related to the cast of the movie like actors names, director names, likes the cast got on Facebook. Also, this dataset contains information about reviews and comments given by the customers like count of critic reviews, number of likes, etc. The main information about movies is also given in the dataset like the title of the movie, year, color, duration, etc. Each feature has it's own impact on the target variable.

II. Data Exploration, Pre-processing and Feature Selection

A. Handling Null values

There are many attributes that contain null value in it. The highest number of null values are in the attribute gross and budget and so on. For handling these null values we replace those null values with the mean value for some attributes and for some, we replace with the maximum occurring value of that attribute.

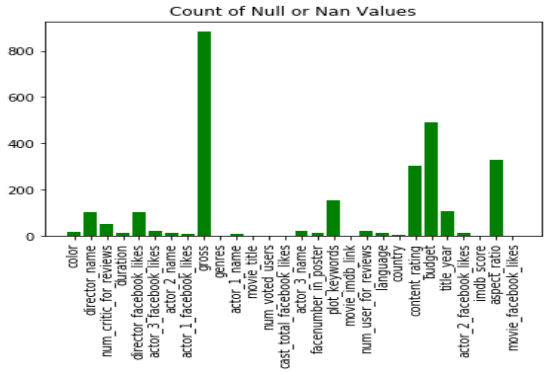

Fig. 1. "Null value count"

B. Add New Feature

The gross value and budget of the movie are given in the dataset. We can calculate the profit earned by the movie using these attributes by subtracting the budget from gross value.

C. Finding Correlation between Attributes

Correlation is a good measure to find that how important or impacting a feature is for the target attribute. According to the observations facenumber_in_poster and title_year are least correlated. So, We can drop these features.

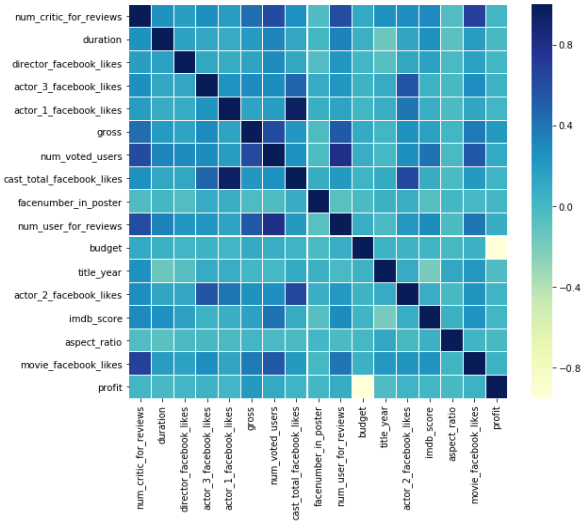

Fig. 2. "Correlation Heat Map"

D. Encoding the Data

The data is encoded for applying classification algorithms by using Label Encoder. The IMDB score is converted into the classes by using ranges like 0-2,2-4 and so on.

III. Techniques Used

A. K-nearest Neighbour

K-nearest Neighbour is used to classifying data by looking at the k number of nearest data point's class labels. The encoded data is split into the train-test data set with a ratio of 80:20 respectively. Here, the value for k is unknown. So, It is implemented with multiple values of k. The graph is drawn to show accuracy. The maximum accuracy achieved by kNN is 0.63 with the k equals to 18.

B. Decision Tree Classifier

Decision Tree with different height is implemented by using cross-validation. The training accuracy and test accuracy for different heights are shown in the graph. The Accuracy achieved by the decision tree is 0.66 which is better than the kNN model. The best tree would be with height 7 or 8.

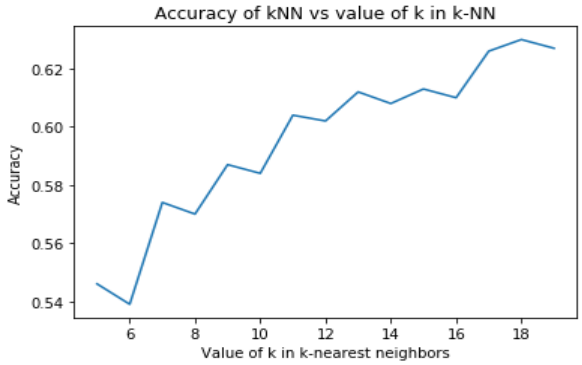

Fig. 3. "Accuracy of kNN"

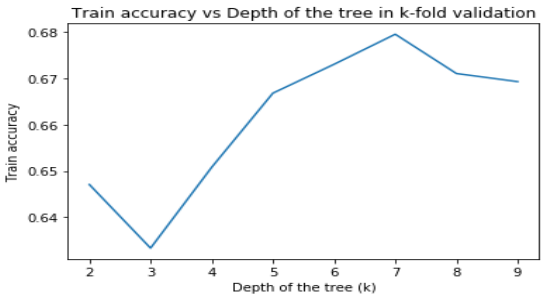

Fig. 4. "Train accuracy of Decision Tree"

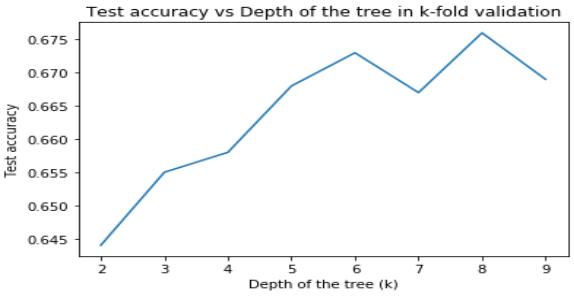

Fig. 5. "Test accuracy of Decision Tree"

IV. Results and Inferences

K-NN gives the accuracy of 0.63 with k equals to the 18 and the decision tree give the accuracy of 0.66 with the depth of the tree equals to 8. So, a comparatively Decision tree performs better for this problem than that of K-NN.

REFERENCES

[1] https://www.kaggle.com/carolzhangdc/imdb-5000