Analyse IMDB Score with Data Mining Algorithms

Abhishek Vyas
M tech Computer Science
Abhishek19086@iiitd.ac.in
Indraprastha Institute of Information Technology Delhi
Data Mining

Abstract
Entertainment is one of the important part of our life and movies makes it more entertaining. Now days, movies are not only source of entertainment but also a big source of income for film companies as well as many industries related to film industries. Many productions houses are investing in eagerness of good outcome from a movie they are investing on. If we are able to predict the success or failure of any movies on the basis of some data resources and on the basis of technical analysis or with help of certain factors than it may save many from huge losses and may help many to earn much than they predict. And we are trying to do same here with the help of certain data mining algorithms.

1 Introduction

1.1 Background

A successful movie not only win heart of their audience but also fill bank accounts of many who does a big investment on it. Many factors may result either success or failure of a movie. A good actor and experienced director may help production house to earn high profit on box office but it is not always do same to get good IMDB score.

1.2 Data Description

The dataset is taken from Kaggle website. It contains 28 variables for 5043 movies, spanning across 100 years.Here "imdb_score" is the response variable while the other 27 variables are possible predictors.

The original dataset has been replaced in Kaggle, here's the link for the original dataset from Dataworld:
https://data.world/data-society/imdb-5000-movie-dataset

1.3 Problem Statement

On the basis of IMDB movies data, our task is to predict right class of imdb score on the basis of remaining attributes of the data. It is encouraging to analyse various trends data and to find attributes affecting the IMDB score most or least on basis of various attributes. In this project we trying to achieve same by using data mining algorithems.

2 Data Exploration

2.1 Remove Duplicates

IN our DataSet we check and remove duplicate rows. After removing duplicates rows we have row count 4998 which was 5043 earlier.We also check if any row have all null values but no such row found in data.

2.2 Remove Null Values

For removing null values from our dataset we find percentage of null values in each attributes. We need to handle three type of cases here:-

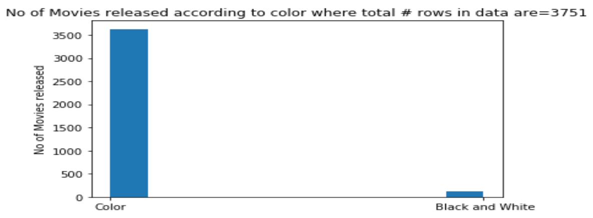
Type1(attributes with *high% of null values* as *gross* 31.21%, *budget* 17.39%,*aspect_ratio* 11.67%, *content_rating* 10.75% **):-** In such case we remove all such rows from which have null values for any of given attributes. Due to this 24% of rows are deleted.

Type2(Null in Numeric Attributes):- Few attributes as 'gross' ,'num_of_facebook_likes', 'budget' have zeroes and null values. In such case first we replace All zeros with NA/Null and than replace all NA values with mean value of given attribute.

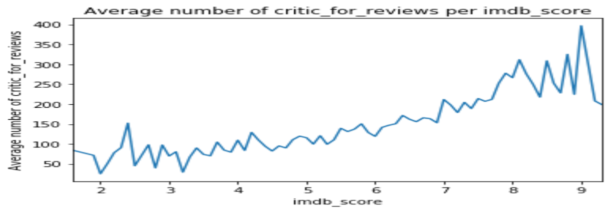
Type3(Null in Non-Numeric Attributes**):-**Few attribute as `movie_title` ,` diector_name`, ` actor_1_name' have non-numeric values. In such case we replace Null values with most frequent Item in given attributes.

2.3 Visualise Trends in Data using graphs

We try find trends of different in data by making many graphs using matplotlib few trends are as follows:-



Here we can see in color attributes above 95% of row values are same such attributes are nearby constant. Same trend in followed in attributes as `country` where most are USA and `language` which mostly have English value.



As in given graph we can see the dependency of "critic_for_reviews_per_imdb" on IMDB Score.

In same way we see different trends and concluded that attributes that do not help in classification can be deleted.

2.4 Remove Attributes

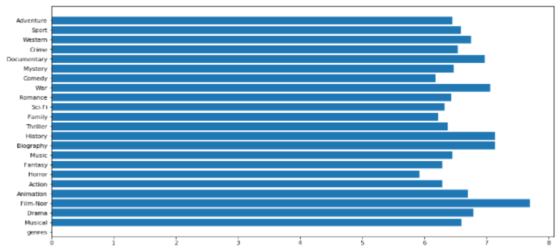
Some attributes as ‘color’,’country’,’langauge’ are nearby constant as most of the row in column have values so we delete such attributes.

Some attributes as ‘movie_title’ **,** ‘aspect_ratio’**,** ‘facenumber_in_poster’**,** ‘movie_imdb_link’ do not affect IMDB Score So we remove all these attributes.

Attributes as ‘movie_title’,’director_name’ and all such attributes as actors name which have high count of unique values do not help much is classification so remove such attributes before applying classification model.

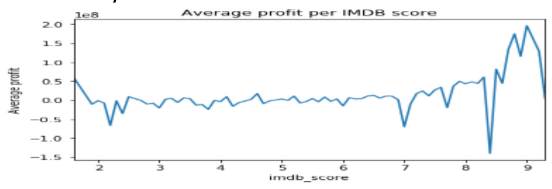
2.4 Split Genres

As ‘Genre’ attribute have many genres so we split these genres and make column for each genre and try to find its effect on IMDB Score. We found that most of the genres have IMDB Score in range 6 to 8. And these attributes are not affecting to IMDB Score. So we need not to add these attributes for classification.

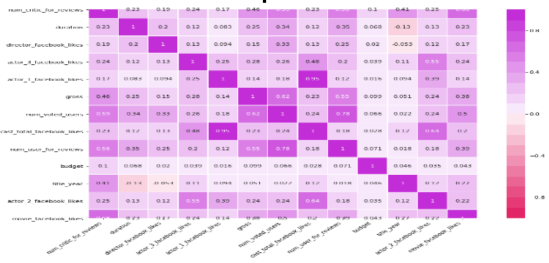


2.4 Add Attributes

We add “Profit” and “Profit_percentage” attributes whose values is calculated with help of Gross and Budget .We find that average profit and IMDB Score are directly related to each other.



2.5 Correlation Heat Map



From Attributes Correlation Heatmap we can see that few attributes are highly correlated to each other. We remove one of two highly co-related attributes, add some attributes using few other attributes so that correlation can be removed and we get new attributes.Eg.

```
others_facebook_likes=sum(actor_2_facebook_likes+actor_3_facebook_likes)
```

```
num_user_per_critic=num_user_for_reviews/num_critic_for_reviews
```

2.5 Create Discrete Classes for IMDB score

As IMDB Score is a continuous attribute in range from 1 to 10.We want to predict the class of movies on basis of various attributes so we classify IMDB score in different classes as

“A+” ([8,10], “A”[7-8), “B”[6-7), “C”[5-6), “D”[0-5)

3 Data Encoding and splitting

We use label encoder to encode pre-processed data so that it can be used by various model as they want values to be integer.

Than we split the encoded training data into X_train and y_Train in ration of 7:3

4 Algorithm Implementation

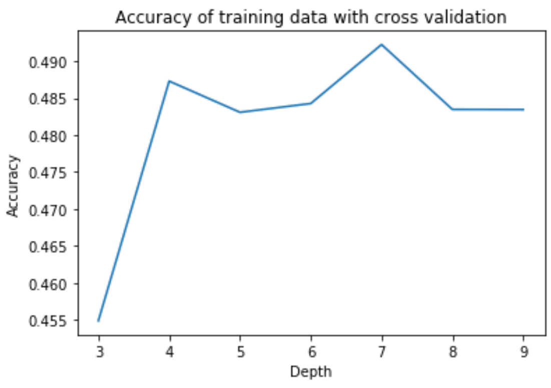
4.1 Decision Tree Classifier

We use DecisionTreeClassifier from sklearn.tree and use “gini” as criterion to predict the class of IMDB score.

We find Accuracy of Classifier at different depth in both training data (using cross validation) and testing data:-

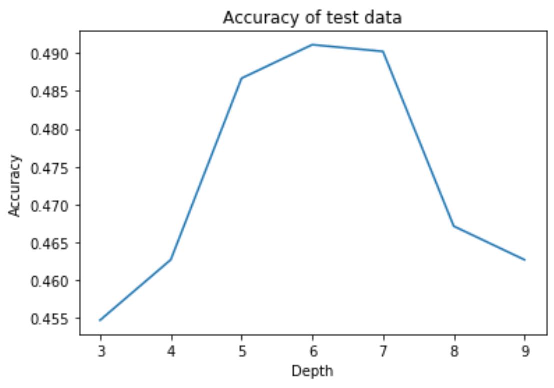
Cross Validation on training data at k =5

We get five accuracy for each depth so we take mean of five values to save accuracy of cross Validation at each depth



We get maximum mean on depth=7 and equal to 49%

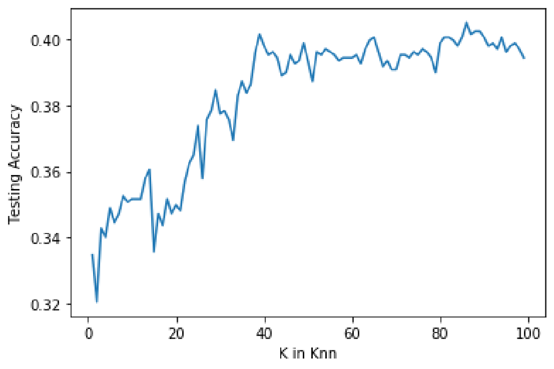
Accuracy on training data using gini citeria at different depth. Here we use X_train and y_train from training data to train our modal and than predict the IMDB class of of test data using our trained modal and than find accuracy of modal by using predicted class and actual classes of IMDB Score.



We get higest accuracy at depth 6 and it is 48.5%

4.2 KNN Classification

We use splitted test and train data that we split during pre-processing .We use sklearn for classifier and use default parameters as p=2 which is equivalent to use euclidian_ditance in it. Similarly Different default values have different meanings. Here we make Accuracy graph at different values of k from 1 to 100 and than use best value of k as per our data.



We get highest accuracy at k=82 which is 40.05%

4.3 Classification using Support Vector Machine

We use sklearn library to use svm classifier.
We train the svm modal using our training data
And predicted the values using that modal on test data. And than find Accuracy using real and predicted value of IMDB Class.
We get Accuracy =37.56%

4.3 Random Forest without LDA

From sklearn.ensemble we use Randomforest classifier and use parameter as max_depth=2 and random_state = 0.And in same way as before we train our modal than predict values of IMDB class using modal and than find Accuracy.

Accuracy= 42.09 %

4.3 Random Forest with LDA

LDA tries to reduce dimensions of the feature set while retaining the information that discriminates output classes. LDA tries to find a decision boundry around each cluster of a class.
From sklearn library use LinearDiscriminantAnalysis and pass parameter n_components = 1

Using LDA we fit the X_train and X_test ie the features of training and testing data and use this features to tain our Random forest model and than find accuracy as before.

Accuracy= 45.73%

5 Conclusion

Model	Accuracy
Decision Tree Classifier	48.5%
KNN Classification	40.5%
Support Vector Machine	37.56%
Random Forest without LDA	45.73%
Random Forest with LDA	45.73%

After applying the five models, we conclude that Decision Tree Classifier model give the highest accuracy .It predict IMDB Score class more accurately as compare to other classifier.
Our overall accuracy move in range of 35 to 48 it is quite low to use modal in real life.
In my inference the reason of low accuracy is less amount of data. In starting we have 5043 rows of data to train which remain to around 3500 after various pre-processing steps .
Large data set will be helpful for finding the accuracy.

References

https://scikit-learn.org/
https://www.kaggle.com/carolzhangdc/imdb-5000