Predicting IMDb Movies with KNN and Linear Regression

Luis Damy

Department of Artificial Intelligence
Universidade Federal de São Paulo
São José dos Campos, SP, Brasil
damy.luis@unifesp.br

Abstract— **The Film industry has been one of the most important focal points of entertainment since the early ages. One of the reasons for a sudden increase in movies critics is due to the arrival of the internet which brought large amounts of random data. Thus, the field of machine learning research has become increasingly popular for conducting studies applied to predictions, such as movie ratings in the IMDB database. In this work, the IMDB dataset was used to predict movie ratings using the KNN algorithms, which achieved an accuracy of 78%, as well as the linear regression algorithm with 62%.**

***Keywords*— IMDB, Machine Learning, movie prediction, movie rating.**

I. Introduction

The film industry has expanded globally, providing audiences with the opportunity of watch a premiere and share their opinions within seconds. This scenario brings challenges such as the ability to produce high-quality films that resonate with audiences in order to gain popularity across various genres. For the purpose of achieving a convergence of data among global audiences, a platform for critiques and reviews, including ratings, actor information, director details, summaries, and production costs, known as the Internet Movie Database (IMDB) [1] has been created and so far, it's the one of the most popular.

There's a myriad of studies on predicting movies. In some of these, data mining is used through algorithms. This enables to gather information efficiently and determine which prediction models such as KNN, Decision Trees, Logistic Regression, MLP, Naive Bayes, SMO, Linear Regression, are most suitable.

II. Related Work

Many other researchers have used a lot machine learning techniques to predict success rates. The usage of data from films instead of social networks are being used for analysis, and it was discovered that logistic regression brings an accuracy of 84% [5]. Pramod, Abhisht, and Geetha demonstrate that fuzzy logic contains high accuracy for categorizing predictions [6]. The usage of machine learning are being applied to synthesize datasets in order to build an efficient structure for prediction using IMDB [7]. Other machine learning algorithms are being used to predict the success rate of a film [8]. Depending on numerous film attributes are applied in mathematical models as a way of determine a film success.

Another study highlights the divisions in the domestic and international film markets from Russia, in which international films are head from 2002 to 2014 [1]. This work also distinguishes three factors behind a successful film: cost, popular actresses and actors, directors, and audience opinions. Ultimately, the model concludes that these sanctions and the movie's budget have a significantly impact on the success of the film.

III. Methodology

In this study, data classification was used as being one of the most common forms of machine learning techniques as shown in Figure 1.

The steps are described below:

- Data extraction
- Data preprocessing

- Application of machine learning techniques
- Comparison of results from different algorithms

The supervised classification method is divided into several groups. The dataset is split into training and testing groups.

The former group is being used for modeling, training the algorithm using a percentage of the extracted and shaped data. The latter is then applied to perform the test and obtain the result based on the accuracy in comparison to the test set.

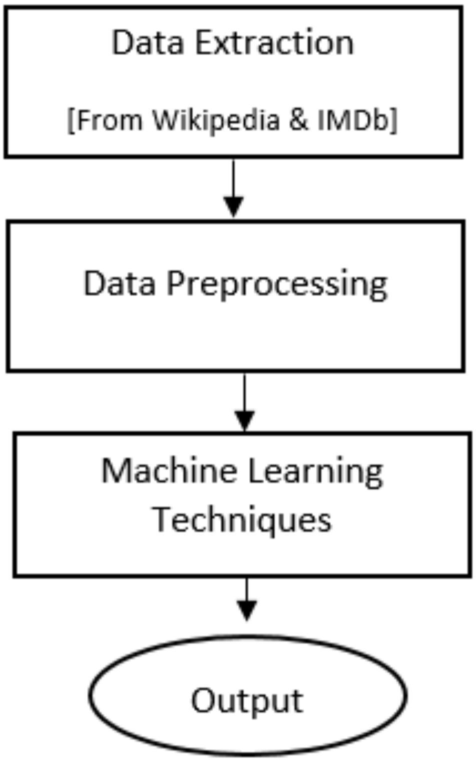


Fig.1 Study methodology workflow

For a correct analysis of the dataset, the data must be formatted and cleaned. This phase is called preprocessing.

In this study, unnecessary fields such as director names, actors, language, etc., were removed. Then, the table was separated into relevant items for the study.

TABLE I

Index	Columns
0	Genre
1	Duration
2	Rating
3	Critical Review
4	User Review
5	User votes
6	Reviews votes
7	Amount Movie likes
8	Amount Director likes
9	Amount Actor likes
10	Parental Rating (PR)

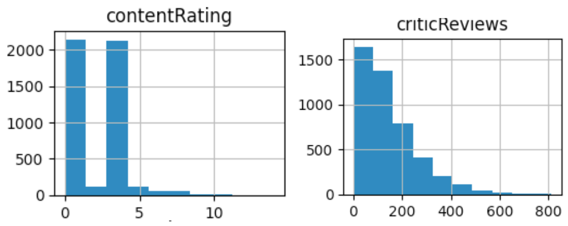
A. *The Movie Dataset*

The data extraction was carried out by reading the IMDB.csv file, obtained from the IMDB website [1]. The dataset used contains approximately 5000 rows and 30 columns of information initially.

B. *Data Extraction*

Through file reading functions provided in Pandas' library, it was possible to extract and filter the relevant information. Subsequently, the preprocessing step was performed, removing columns such as id.

Additionally, the filtered information from the initial file was dumped into another file called IMDB_likes_review.csv. Through this second file, a distribution of information was carried out via histograms represented in Figure 2.



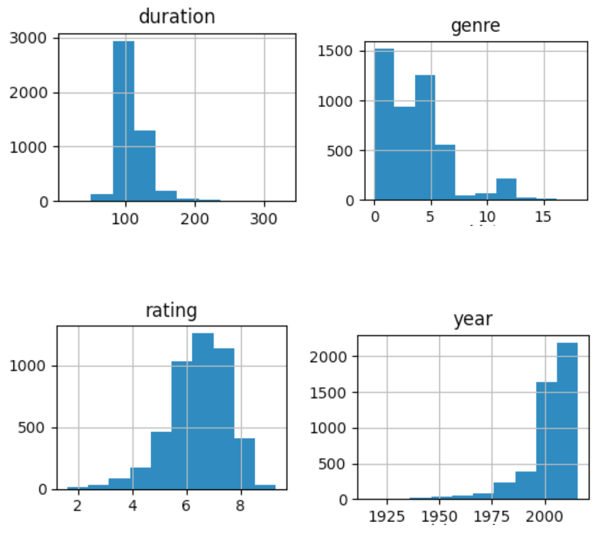


Fig.2 Preprocessing statistic data

Throughout this process the relevant information required for the utilization of machine learning techniques has been achieved. The pertinent data gathered afterwards was about 4500 movies.

IV. Experimental Results

A. Linear Regression

The linear regression model in statistics is a model with one explored variable. That is, in a two-dimensional graph, there are example points and one single dependent variable as well as an independent variable. This method operates by attempting to find the linear function, a straight line, that most accurately predicts the values of the dependent variable as a function of the independent variable.

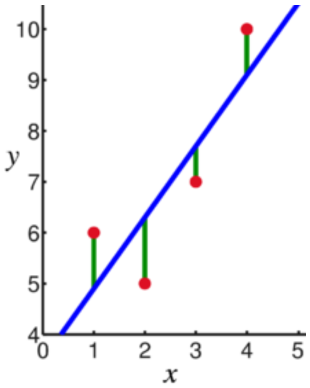


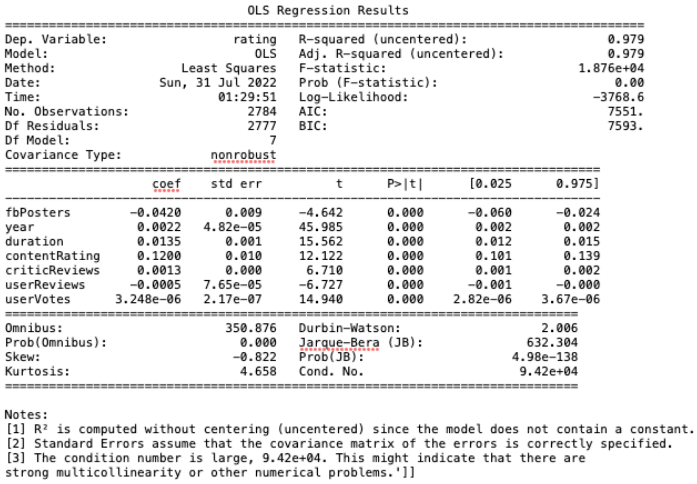
Fig.3 Linear Regression Explanation

The linear regression algorithm used in the study was Ordinary Least Squares (OLS) as it is one of the simplest methods to perform tests. This method focuses on minimizing the sum of squared differences observed in the dataset and from the predicted values.

Based upon the training the data in conjunct with the train_test_split function available in sklearn model_selection, it was possible to conduct training on the training data responsible for 40% of the total.

Additionally, through the statsmodels.api library, it was possible to use the "sm" model containing the OLS function capable of conducting the test, passing the training x and y values as arguments. In them, it is known that y contains the original vote values.

The results obtained were saved using the pre-built function, fit(). Then, a summary of this information was created and written into a file named "OLS_result.txt".


```
                            OLS Regression Results                            
=======================================================================================
Dep. Variable:                 rating   R-squared (uncentered):                   0.979
Model:                            OLS   Adj. R-squared (uncentered):              0.979
Method:                 Least Squares   F-statistic:                          1.876e+04
Date:                Sun, 31 Jul 2022   Prob (F-statistic):                        0.00
Time:                        01:29:51   Log-Likelihood:                         -3768.6
No. Observations:                2784   AIC:                                      7551.
Df Residuals:                    2777   BIC:                                      7593.
Df Model:                           7                                                  
Covariance Type:            nonrobust                                                  
=================================================================================
                    coef    std err          t      P>|t|      [0.025      0.975]
---------------------------------------------------------------------------------
fbPosters        -0.0420      0.009     -4.642      0.000      -0.060      -0.024
year              0.0022   4.82e-05     45.985      0.000       0.002       0.002
duration          0.0135      0.001     15.562      0.000       0.012       0.015
contentRating     0.1200      0.010     12.122      0.000       0.101       0.139
criticReviews     0.0013      0.000      6.710      0.000       0.001       0.002
userReviews      -0.0005   7.65e-05     -6.727      0.000      -0.001      -0.000
userVotes      3.248e-06   2.17e-07     14.940      0.000    2.82e-06    3.67e-06
==============================================================================
Omnibus:                      350.876   Durbin-Watson:                   2.006
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              632.304
Skew:                          -0.822   Prob(JB):                    4.98e-138
Kurtosis:                       4.658   Cond. No.                     9.42e+04
==============================================================================

Notes:
[1] R² is computed without centering (uncentered) since the model does not contain a constant.
[2] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[3] The condition number is large, 9.42e+04. This might indicate that there are
strong multicollinearity or other numerical problems.']]
```

Fig.4 OLS results file

Based on the data obtained from the File (Fig. 4), it is possible to demonstrate mathematically that there was an accuracy of 62%. This value was calculated based on the results of predicting tests of type X and by calculating the result with the accuracy of real cases of type Y. The difference founded between them was of 0.620286. Additionally, the execution time of the regression algorithm was 0.0016 seconds.

B. K-Nearest Neighbours

The KNN Algorithm is a supervised non-parametric learning method used for classification and regression. It applies a technique in which it assigns weights to the contributors of its "neighbors", representing the values of points closest to it. Thus, the nearest neighbors contribute to a average which is greater than those who are near by (Figure 5). Once the value of K is assigned, the neighbor that falls within the distance of this assignment will be assigned the category of the nearest.

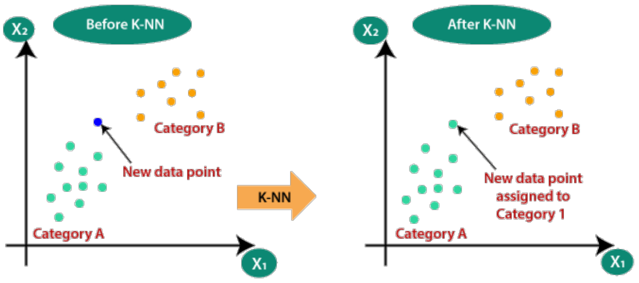


Fig.5 KNN Explanation

In this algorithm, it was done another filter was performed. In particular the IMDB ratings were splinted into a scale, ranging from 0 to 10 separated into 2.5 intervals, corresponding to the actual ratings. Additionally, a category grid was created to correspond to the film quality divided in Table 2.

TABLE II

Grade Criteria		
IMDB Rating	**Category**	**Grade**
0.0 – 2.4	0.0	Very Bad
2.5 – 4.9	2.5	Bad
5.0 – 7.4	5.0	Good
7.5 – 9.9	7.5	Very Good
10.0	10.0	-

Furthermore, an evaluation was made based on the value of K regarding its accuracy. The range for this scenario was set between 10 to 100 (Figure 6).

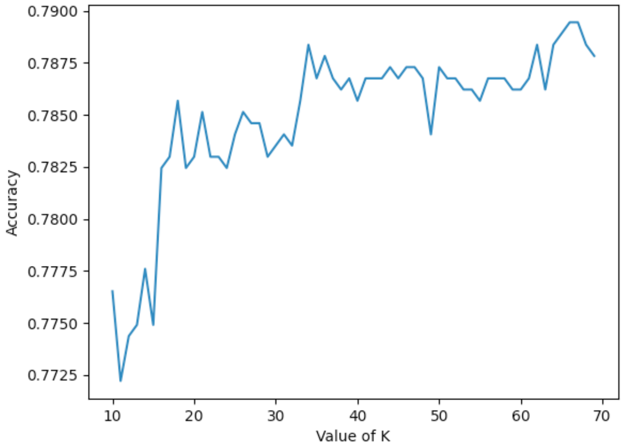


Fig.6 Comparison among K' values

Thus, it is noticeable that the peak point happened when the value of K corresponds to the highest accuracy is around 65, with an accuracy of approximately 79%. However, the execution time was around 4.741 seconds.

V. Conclusions

The results obtained in the study are contained in Table III. Although there was significantly better performance compared to the KNN algorithm, it had a much longer execution time than the Linear Regression.

TABLE III

Algorithm	% Accuracy	Execution Time
Linear Regression	62.02	0.0016 sec
KNN	78.95	4.741 sec

VI. Discussion and Future Work

The influence of changing the sample size on both tested algorithms was analyzed. After analysis, there was no significant change in the variation of the test_size variable when training the data in the train_test_split function.

Furthermore, varying the value given for K also did not show significant improvements beyond the limits defined in Figure 4. This corresponds to machine learning regarding the obtained data.

REFERENCES

[1] Internet Movie Data Base. URL: https://www.imdb.com/

[2] Warda Ruheen Bristi, Zakia Zaman, Nishat Sultana, “Predicting IMDb Rating of Movies by Machine Learning techniques” URL:https://www.researchgate.net/publication/338369572_Predicting_IMDb_Rating_of_Movies_by_Machine_Learning_Techniques

[3] Wikipedia KNN Definition. URL: https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm

[4] Wikipedia Linear Regression. URL: https://en.wikipedia.org/wiki/Linear_regression

[5] M. H. Latif and H. Afzal, “Prediction of movies popularity using machine learning techniques,” *International Journal of Computer Science and Network Security (IJCSNS)*, vol. 16, no. 8, p. 127, 2016.

[6] S. Pramod, A. Joshi, and A. Mary, “Prediction of movie success for real world movie dataset,” *Int. J. of Advance Res., Ideas and Innovations in Technol*, vol. 3, no. 3, 2017.

[7] Medium Web, “Predicting IMDb Ratings of New Movies” URL: https://medium.com/web-mining-is688-spring-2021/predicting-imdb-ratings-of-new-movies-2b39459fee9a

[8] TowardsDataScience Web, “Predicting IMDb Movie Ratings using Supervised Machine Learning” URL: https://towardsdatascience.com/predicting-imdb-movie-ratings-using-supervised-machine-learning-f3b126ab2ddb

[9] GitHub Web, “Predicting IMDb Ratings with Linear Regression” URL: https://github.com/josephpcowell/cowell_proj_2

[10] GitHub Web, “K-Nearest-Neighbor-IMDB-Project” URL: https://github.com/Alex-Schlee2/K-Nearest-Neighbors-IMDB-Project

[11] Analyticsvidhaya Web, “Predicting Movie Genres using NLP URL: https://www.analyticsvidhya.com/blog/2019/04/predicting-movie-genres-nlp-multi-label-classification/\

[12] Medium Web, “IMDB Movie Genre tag prediction” URL: https://www.analyticsvidhya.com/blog/2019/04/predicting-movie-genres-nlp-multi-label-classification/

[13] GitHub Web, “Python-IMDb-Ratings-Prediction” URL: https://github.com/akashagte/Python-IMDb-Ratings-Prediction

[14] Kaggle Web, “IMDB-Perform Sentiment Analysis with scikit-learn”