

EXPLORING MACHINE LEARNING APPROACHES FOR IMDb MOVIE RATING PREDICTION AND RECOMMENDATION

*Daneshwari Mulimani**, *M. S. Jevoor*** & *Shreya Kulkarni****

DOI: <https://doi.org/10.5281/zenodo.21207828>

ABSTRACT

In the film industry, movie ratings are now a vital predictor of box office success. For personalized recommendation systems, which rely on reliable and effective models to produce accurate results, accurate prediction of these ratings is also essential. Using the IMDb dataset, this study investigates the use of machine learning techniques in movie rating analysis and recommendation systems to forecast movie rating categories and produce customized movie recommendations. A brief comparison with deep learning architectures is also provided. The results of this study provide a useful resource for future developments in movie rating prediction, assisting practitioners and researchers in enhancing recommendation systems' efficacy.

KEYWORDS: TF-IDF, Deep Learning, Movie Rating, Machine Learning.

Introduction

This paper tackles the important issue of predicting movie ratings effectively and producing personalized recommendations in both the film industry and online streaming. With an immense amount of movie data available, especially on websites such as IMDb, machine learning techniques can be utilized to produce relevant insights and even predictions over ratings that significantly enhance the user experience by providing tailored content recommendations. But selecting the most efficient approach is complicated by the diversity of data types involved - numerical ratings versus textual metadata - and the complexity of individual user preferences. The objective of the study is to evaluate and compare traditional machine learning techniques, in particular, a classification model with TF-IDF vectorization along with cosine similarity for content.

Overview

The data preprocessing will be done with the complete IMDb movie data set. The following process is followed to classify the huge data according to each category like genre, casting, director, etc.

Data cleaning and content analysis: It performs categorical encoding using One-Hot Encoding (OHE). OHE creates binary columns for each category, which converts the categorical variables (genre, director, cast) into numerical form extracted from the sample data set which will be shown as below.

- Genre: Action → [1,0], Comedy → [0,1]
- Director: Rajmouli → [1,0], Priyadarshan → [0,1]

The next process will be text cleaning for review text to make it usable for sentiment analysis or embeddings. By lowercasing, removing punctuation, removing stop words, and Stemming/Lemmatization process, we finally get data in the form of Cleaned reviews:

- Bahubali → "mind bend visual stun"
- SeethaRamam → "heartbreak love story"
- Avathar → "epic space journey emotional depth"

The next process involves generating sentiment scores to reviews using tools like VADER or TextBlob.

- Bahubali → 0.85 (positive)

* **Dr. Daneshwari Mulimani**, Assistant Professor, Department of Computer Science, BDEA's S.B. Arts and K.C.P. Science College, Vijayapur.

** **M. S. Jevoor**, Assistant Professor, Department of Computer Applications, BDEA's S.B. Arts and K.C.P. Science College, Vijayapur.

*** **Shreya Kulkarni**, Student, Department of Computer Science (IV Sem), BDEA's S.B. Arts and K.C.P. Science College, Vijayapur.

- SeethaRamam → 0.70 (positive but less intense)
- Avathar → 0.80 (positive)

Feature Engineering: This step involves Numerical Features for a movie with Runtime, year, sentiment score. Along with Categorical Features like Encoded genre, director, cast. Later it extracts the Derived Features from the IMDb data set for Director popularity (based on average IMDb rating) and actor frequency.

Comparison with Model Approaches

Linear Regression

Linear regression models the relationship between independent variables (features like runtime, year, sentiment, genre, director, cast) and the dependent variable (IMDb rating).

- Rating= $\beta_0+\beta_1\cdot\text{Runtime}+\beta_2\cdot\text{Sentiment}+\dots+\epsilon$
- β_0 : Intercept
- β_i : Coefficients (weights) learned by the model
- ϵ : Error term

With larger datasets, performance will vary - linear regression is a good baseline model. This model can be interpreted; coefficients represent the contribution of each feature towards the rating. Ensemble approaches and Deep Learning are generally superior to Linear Regression for predicting highly non-linear relationships.

The process of a typical Machine Learning Environment for IMDb Movie Rating Prediction can be represented by this workflow, beginning at Data Collection (IMDb metadata, reviews, external features), followed by Data Preprocessing (cleaning, normalization, sentiment analysis), Feature Extraction (cast, director, genre, reviews), and finally using ML Models (regression, decision trees, deep learning, NLP). The workflow concludes with Evaluation Metrics (RMSE, MAE, R^2) and Integration into a Recommendation System.

Deep Learning Methods with CNN Process

The figure 1 represents a visual representation of the method (embedding →

Table 1: Sample Dataset Extraction

Title	Runtime	Year	Senti-ment	Genre_Action	Genre_Comedy	Direc-tor_RMouli	Direc-tor_Pri-yadar-shan	Cast_Leonar-do	Dr_Pop-ularity	IMDb Rating
Bahubali	148	2016	0.85	1	0	1	0	1	8.7	8.8
Seetha-Ramam	195	2023	0.70	0	1	0	1	1	7.8	7.8
Avathar	169	2000	0.80	1	0	1	0	0	8.7	8.6

Figure 1: Architecture of IMDb movie rating with Deep learning Method of CNN

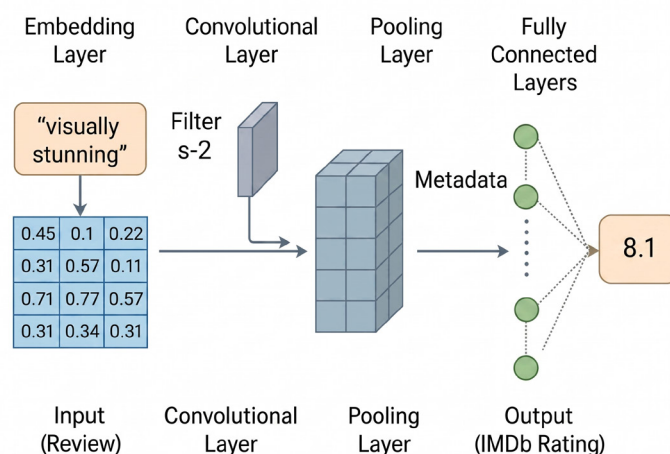


Table 2: Comparative study of Prediction review rate with three different Methods

Title	Actual Rating	Predicted (Linear)	Predicted (RF)	Predicted (TF-IDF)
Bahubali	8.8	8.5	8.7	8.8
SeethaRamam	7.8	7.5	7.7	7.9
Avathar	8.6	8.4	8.5	8.6

convolution → pooling → dense → score prediction). Each word inside the reviews is mapped as a dense vector (Word2Vec, GloVe, or BERT embeddings). Consequently, the review can have a matrix of word embeddings.

Pooling Layer: Max-pooling reduces the quantity of information that has been inputted through the filters by only taking the most critical records from each. This allows for a reduction in the dimensions of the data without losing the essential factors of the data.

Fully Connected Layers: After pooling has taken place, the pooled features visit the fully connected layers. Any metadata (runtime, year, sentiment score) could also be concatenated here.

Output (IMDb rating): The final dense layer produces an output score (for instance, 8.1) of the evaluation. CNNs [3] are good at detecting local semantic styles consisting of “heartbreaking love story” or “visually stunning”. CNNs are also more efficient than RNNs and can be more effortlessly parallelized. But, they are extremely tough to analyze to decide how the sentiment was determined within the assessment. CNNs can scale very well; however, their capability to remember local context is restrained.

Random Forest Method with TF-IDF

This method helps to frame a decision tree and adequate predictions. It's an ensemble method that builds many decision trees and averages their predictions. It handles nonlinear relationships between features (e.g., runtime × sentiment × director popularity). [4] This method provides feature importance scores, showing which factors drive ratings most. It is very robust against noise in user reviews and metadata.

The overall rating and prediction are analyzed with extended metadata (runtime, year, director, etc.) but also from textual movie reviews using embeddings like TF-IDF (Term Frequency–Inverse Document Frequency). It runs by combining metadata and text features into a single feature matrix. Later, we have to train a Random Forest Regressor. Scalability and data interpretability are very high compared with the CNN process. For the large data set, TF-IDF provides a high accuracy rate.

The expected output for the above sample will be generated by applying the three methods with respect to the review rates as follows.

Conclusion

Based on the study on [1] and [2], TF-IDF captures the meaningful words in reviews. Random Forest learns nonlinear interactions

between metadata and text features. Together, they provide a balanced model that uses both structured and unstructured data. TF-IDF with the random forest method is excellent for baseline models and smaller datasets - interpretable, fast, and robust. Whereas Deep Learning processes like CNN and RNN dominate when you have large-scale IMDb reviews and want to capture nuanced meaning in text.

References:

1. Warda Ruheen Bristi et al., “Predicting IMDb Rating of Movies by Machine Learning Techniques”, 2019, <https://www.imdb.com>
2. Manisha Valera, Dr. Rahul Mehta, “Advanced Deep Learning Models for Improving Movie Rating Predictions: A Benchmarking Study”, *BenchCouncil Transactions on Benchmarks, Standards and Evaluations* Volume 4, Issue 4, December 2024, 100200, <https://doi.org/10.1016/j.tbench.2025.100200>
3. Saikiran Gogineni et al., “Predicting IMDB Movie Rating Using Deep Learning”, *IEEE Xplore*: 10 July 2020, DOI: 10.1109/ICCSES48766.2020.9137994.
4. Anubhav Mathur, Snigdha Patil, “A comparison of Machine Learning techniques for predicting IMDb score of movies”, *International Journal of Advance Research, Ideas and Innovations in Technology*, 2023, www.IJARIIT.com.
5. Prof. Shubhangi Patil et al., “IMDB Movie Rating Prediction Using Machine Learning”, *International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)*, International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal Volume 3, Issue 12, May 2023.