Predicting IMDb Movie Rating using Machine Learning

Ananya Jain
ananya19408@iiitd.ac.in

Manasvi Singh
manasvi19369@iiitd.ac.in

Pritish Wadhwa
pritish19440@iiitd.ac.in

Yash Bhargava
yash19289@iiitd.ac.in

1. Abstract

Is it possible to predict the rating of a movie prior to its release or production? Every year countless movies are made and released worldwide. All these movies are given ratings by viewers throughout the globe.These ratings are combined together to form the IMDb ratings. IMDb rating is the singlemost influential factor in deciding any consumer's opinion and inherently the success of a movie.

With the machine learning techniques at our disposal, we aim to predict the seemingly unpredictable IMDb rating of any movie before its theatrical release. Successfully predicting IMDb rating is beneficial for both producers (from a financial standpoint) and consumers (from an entertainment standpoint) alike.
Link to the project: https://github.com/PritishWadhwa/IMDb-Movie-Rating-Predictor

2. Introduction

The main problem our team aim to tackle is to predict the IMDb rating of any movie prior to its release. We use a variety of features ranging from the movie overview, length of movie run time, the country where the movie was produced, the language of the movie, to the details about the lead actors, the movie director and even the details about the movie's key poster. With this information in our arsenal, we aim to use a number of machine learning algorithms to accomplish this uphill task.
We have made sure that features which are affected by the release of movies are not taken into account. These features include properties like popularity, revenue, user ratings among others.
We have tackled this problem as a regression task. We aim to predict the ratings as close as possible to the original IMDb ratings. One thing to note is that since the IMDb ratings are influenced by the viewer reviews, these ratings might fluctuate. But over time, these ratings are more or less constant. For the task at hand, we are using the IMDb ratings as reported in the dataset without actually verifying any small change there might have been in any of the movie ratings since the publication of the dataset.

3. Literature Survey

Oghina et al. in their paper "Predicting IMDB Movie Ratings Using Social Media" [3] addressed the task of predicting IMDb movie ratings using data collected from social media services. They identified qualitative and quantitative activity indicators for a movie in social media, and extracted two sets of surface and textual features. They trained various models and upon analysing them, they found that the fraction of the number of likes and dislikes on YouTube, combined with textual features from Twitter lead to the best performing model, with strong agreement with the observed ratings and high predictive performance.
Quader et al. in their paper "A machine learning approach to predict movie box-office success" [4] proposes a decision support system for movie investment sector using machine learning techniques. The system predicts an approximate success rate of a movie based on its profitability by analyzing historical data from different sources like IMDb, Rotten Tomatoes, Box Office Mojo and Metacritic. They discovered that budget, IMDb votes and number of screens are the most important features which play a vital role while predicting a movie's box-office success.
C. izmeci et. al in their article [5] explored the IMDb ratings by exploring matrix decomposition, regression analaysis and factorization machines on social media data.

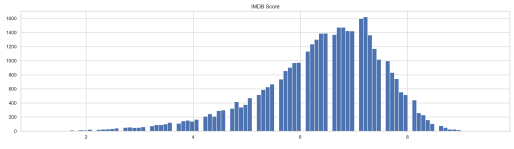

Figure 1. IMDb score distribution over the dataset

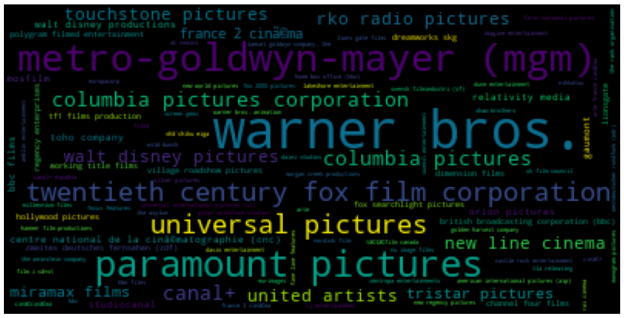

Figure 2. Word Cloud Depicting frequency of major production houses throughout the dataset

4. Dataset

We utilized open-source datasets, "The Movie Dataset" [1], and "Movie Genre from its Poster" [2] from Kaggle. We selected features like posters, synopses, cast, crew, runtime, genre among others as input features and IMDb score as the prediction objective. The data set after preprocessing have the fields mentioned in table 1.

4.1. Preprocessing

The data set had variables which were discarded like Adult, describing whether a movie is adult or not, because of false and highly skewed data. Budget had about 75 percent null or zero values and Tagline also had around 75 percent null or empty values and both of these were dropped. We tried but could not find replacement for budget data. Only alternatives in front of us were illegally scraping the data or dropping the column itself. We could not assign mean or median values to the empty filed as the movies on our dataset ranged over a large number of years. This might have lead to inclusion of false data in the dataset, thus we decided it better to leave out the feature itself.

Post release attributes like IMDb_id, popularity, revenue,

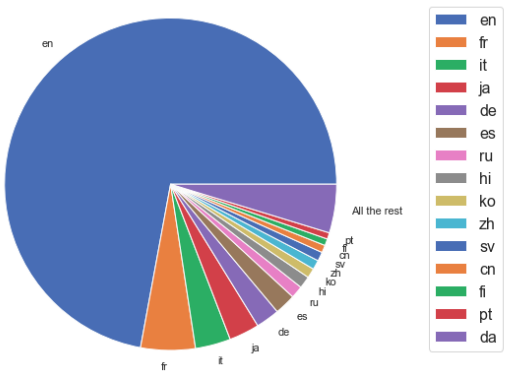

Figure 3. Language Distribution of the movies

Feature Name	Description
year	Year of release of the movie
runtime	Duration of the movie
director	The director of the movie
actors	The actor and actresses in the movie
original_language	The main language spoken
original_title	The title of the movies
overview	The summary of the movie
production_companies	The company producing the movie
production_countries	country of main production
keywords	The main themes in the movie
genre	The different genres of the movies
num_faces	Number of faces in the poster
saturation	Saturation level
hue	Hue level of the movie poster
brightness_sd	standard deviation of brightness level
saturation_sd	standard deviation of saturation level
hue_sd	standard deviation of hue level
green	green colour level in the poster
green_sd	standard deviation of green colour level
red	red colour level in the poster
red_sd	standard deviation of red colour level
blue_sd	standard deviation of blue colour level

Table 1. Features

vote_count, homepage, status, IMDb link were discarded, spoken language was also discarded since it was found redundant and non informative. We did binning on release date feature into release year. Production company and country column also required further formatting and were

converted into the desired typed; after processing they were stored as pipe separated values in the intermediate preprocessed stage. We manually extracted 13 visual features which are the mean and standard deviation of red, green, blue, hue, saturation, brightness, as well as the number of human faces using the openCV library. For movie overview and title, we used spaCy for tokenization and nltk for stop words and only kept words that appeared in more than 75 and 10 movies respectively. For cast and crew, we extracted the director and top three actors(in the order of the credits) for each movie, and kept directors involved in at least 10 movies and actors involved in at least 20 movies in our dataset.

Our prediction attribute IMDb_score was subsequently separated from the dataset and made as the labels. After all preprocessing steps, we have a total of 21782 data points and 3561 features.

Figure 4. Word Cloud Depicting frequncy of words in the movie overview

4.2. Feature Selection

We used SkLearn's SelectKBest technique to evaluate the best features in our dataset, we realised Adult to be the feature with the least score hence dropped it. We also realised the poster data did not act as strong features due to their lower scores in the analysis but nevertheless, they were kept in the final dataset as when combined they were substantial enough.

We plotted a correlation matrix and visualized it using a heatmap. We realised our data is not very highly correlated, and dropped the features with strong positive or negative correlation. Irrespective, we found that brightness had a really high correlation with the three primary colours and thus as a consequence it was removed.

We used the Random Forest Classifier to find the features of importance and validated the output we obtained from SelectKBest technique.

4.3. Hot Encoding Data

The textual data is converted to Bag of words format and stop words are removed, analysis is done on the basis of frequency.The Bag of word format is then hot encoded.

Feature	Score
director	2259479
genres	2228322
runtime	27871
original_language	23777
green	6883
red	5974
brightness	4983
blue	4826
saturation	4117
saturation_sd	2374
hue_sd	1517
hue	981
blue_sd	364
release_date	327
num_faces	213
brightness_sd	199
green_sd	180
red_sd	171
adult	31

Figure 5. Feature Scores calculated using SelectKBest

4.4. Preparation of Training and Testing Data

We divided out dataset into training, validation and testing set with a split of 70:15:15, setting the random seed as 0 and hence shuffling the dataset to prevent any unfortunate split and patterned split.

4.5. Data Normalization

Normalization is a scaling technique where the values are centred around the mean and have a unit standard deviation. Hence the mean of the feature in consideration becomes zero and the distribution obtained has a standard deviation of one. The formula followed for the same is:

$$\hat{x_i} = \frac{x_i - \mu}{\sigma}$$

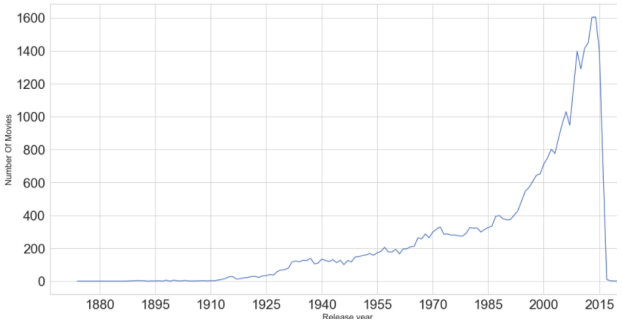

Figure 6. Number of movies vs. Release year distribution

where μ is mean & σ the standard deviation of the feature.

5. Methodology

For the prediction of IMDb scores, we approached the from two ways; one as a regression problem, and other as a classification problem. The models put to use are:

5.1. Regression

We used the features extracted for our task to predict the IMDb ratings, and applied the following regression models:

5.1.1 Linear Regression

Standard linear regression was used as one of our baseline models. Furthermore other regularisation techniques were also used to make the model perform better.

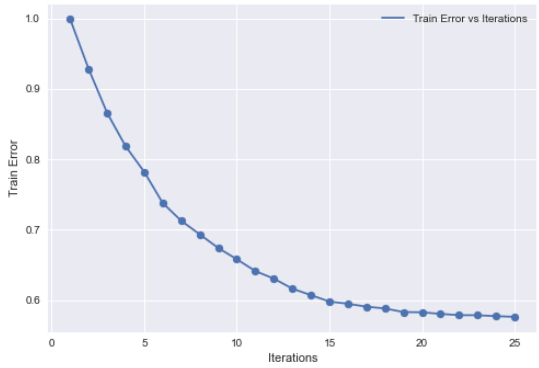

Figure 7. Train Error v.s. Iteration for Ridge Regression

5.1.2 Ridge Regression

Ridge introduces penalty term to shrink model parameters. However it doesn't eliminate parameters and thus model includes all predictors which makes model interpretability difficult. Using GridSearchCV we tuned alpha to be 57. It performed the best among all regression models.

5.1.3 Lasso Regression

Lasso regularization uses shrinkage to form sparse model which helps solve multi-collinearity and aids feature selection. Since it can shrink some model parameters to exactly 0 unlike ridge , it makes a much more interpretable model than ridge. Using GridSearchCV we tuned alpha as 0.002.

5.1.4 ElasticNet Regression

ElasticNet is a linear regression model with combined L1 and L2 priors as regularizer. This allows learning a sparse model like Lasso while still maintaining regularization properties of Ridge. Using ElasticNetCV we tuned alpha and the l1 ratio to be 0.00667 and 0.25 respectively.

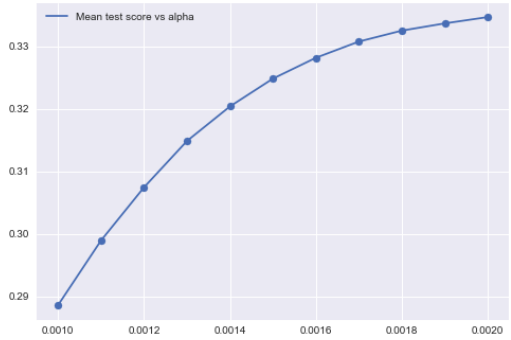

Figure 8. Lasso alpha-avg score plot using GridSearchCV

Model	R^2	**MSE**
Linear Regression		
Training Set	0.5951	0.4824
Testing Set	0.3462	0.7739
Lasso Regression		
Training Set	0.3860	0.7314
Testing Set	0.3860	0.7268
Ridge Regression		
Training Set	0.5175	0.5748
Testing Set	0.4236	0.6823
ElasticNet Regression		
Training Set	0.3811	0.7374
Testing Set	0.3821	0.7314
Support Vector Regression		
Training Set	0.4081	0.7605
Testing Set	0.3087	0.8843

Table 2. Results of Regression Models

5.1.5 Support Vector Regression

Support Vector Regression defines epsilon to identify the error acceptable in our model and C to indicate the allowed tolerance, to identify the best hyper plane to fit the data.We used GridSearchCV to tune hyperparamers C and epsilon on the validation set.They were as follows, C: 0.2138 and Epsilon: 1.02938.

5.2. Classification

The problem was also modelled as a classification problem. The IMDB ratings we had were continuous values between 0 and 10. We rounded them off so that we can convert them in various buckets. For the sake of this problem, we made 3 types of buckets. In the first one, we rounded off the data to nearest multiple of 2 so that we could get 6 buckets(0, 2, 4, 8 and 10). For the second one, we rounded off the data to nearest multiple of 1, so that we could obtain 11 buckets(0, 1, 2, ..., 10). For the last one, we rounded off the data to

nearest multiple of 0.5, so that we could get 21 buckets(0, 0.5, 1, ..., 9.5, 10). The models were trained on all three buckets and their results were observed.

5.2.1 Logistic Regression

We used logistic regression on all three bucketize data and chose suitable hyper parameters by analysing accuracy and precision for various hyper parameters.

5.2.2 Naive Bayes

We used Gaussian, Multinomial, Bernoulli Naive Bayes.For all three buckets, Bernoulli Naive Bayes performed best.On all buckets,bucket with 6 categories has best results.

5.2.3 Random Forest Classifier

We performed Random Forest Classification setting the number of trees as 100.For classification ,we used 6, 11 and 21 buckets to identify our predicted IMDB values.On training the model was seen to over fit the training data.

5.2.4 K-Nearest Neighbours

We performed K-Nearest neighbours on the three buckets and found optimal value of K by analysing value of error v/s value of K.

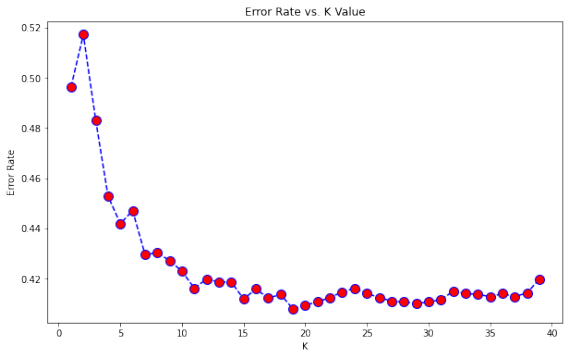

Figure 9. Error rate vs K for 6 bucket classification for KNN

5.2.5 Decision Trees

Decision Trees was performed on the three buckets and GridSearchCV was used to find best hyper parameter. To better its performance, we used Random Forest.

5.2.6 Support Vector Machines

SVM was the most successful classifier we used. Various kernels like *rbf*, *poly*, *linear* and *sigmoid* and values of the parameter C were varied. Variations available for SVM like NuSVC and LinearSVC were also tried. For all the three bucket distribution, best values were obtained with the *poly* kernel and C = 1.0

5.2.7 Artificial Neural Network

ANN was also used to classify the data points for all three types of buckets. Various architectures, booth deep and shallow with varying number of neurons for each layer were experimented with. Despite this, it performed very poorly as compared to SVM.

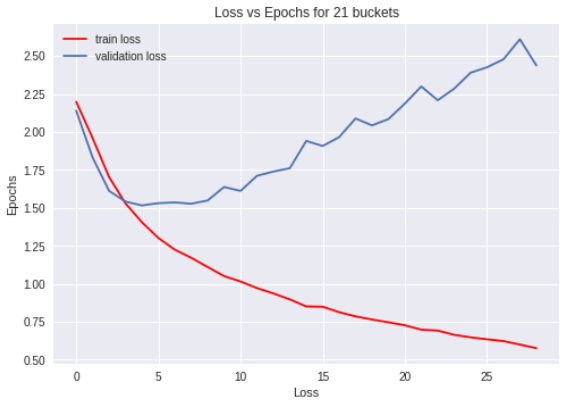

Figure 10. Loss vs Epoch plot for MLP 21 bucket classification

6. Results and analysis

We evaluate our working on this problem statement through the following:

6.1. Metrics

We have used Mean Squared Error (MSE) and R^2 metrics. MSE is used for calculating loss function. R^2, the coefficient of determination is a measurement of proportion of variance. For classification we have used Accuracy, weighted Precision, Recall and F1 to measure the performance of our models. While accuracy gives us a measure of how accurate our predictions are, precision indicates the number of true predictions to all predicted true predictions. Recall indicates the number of true predictions to all actual true prediction and F1 helps score the precision and recall cumulatively.

6.2. Analysis

A number of models were trained for the task for both Regression and Classification. For regression the best performing model was Ridge. We infer that most of the features selected after processing data are important as Ridge outperforms other linear regressors. For classification, SVM was the star performer. Basic neural networks were also tried but SVM outperformed them all. This might be because of SVM's ability of extrapolating the features to higher dimensions thus making the classification more prominent.

7. Conclusion

Our expectations from the project were to predict the IMDb rating of any movie using features that are available prior to its release and analyse the discriminatory power of features to analyse how different parameters impact the rating for any movie using various machine learning models.
We observed that features like length of a movie, the di-

Model	Accuracy	Precision	Recall	F1
Logistic Regression				
Training	0.79235	0.79859	0.79235	0.7854
Testing	0.6315	0.6144	0.6315	0.6165
Random Forest Classifier				
Training	1.0	1.0	1.0	1.0
Testing	0.6588	0.6588	0.6588	0.6182
K-Nearest Neighbours (K=18)				
Training	0.6157	0.6498	0.6157	0.5562
Testing	0.5862	0.5741	0.5862	0.5227
Decision Tree				
Training	0.66	0.65	0.66	0.64
Testing	0.65	0.63	0.65	0.63
Artificial Neural Network(MLP)				
Training	0.98	0.97	0.98	0.98
Testing	0.78	0.77	0.78	0.77
Support Vector Machine				
Training	0.9920	0.9847	0.9919	0.9882
Testing	0.9787	0.9721	0.9786	0.9753
Naive Bayes				
Training	0.6630	0.6769	0.6630	0.6669
Testing	0.5865	0.5977	0.5865	0.5899

Table 3. Results of Classification Models with 6 buckets

rector, movie genres are amongst the most important ones to make any kind of prediction. We also explored various features like the movie poster, the movie overview, the language of the movie and the country of production and found out that even though they play an important part in the predictions, when taken individually they are not as strong as the ones mentioned above.

7.1. Future Work

There are a lot of aspects which can be further explored. Use of deep learning model is one of them. Music of the movie along with its posters, can also play a huge part and this aspect can also be explored.

7.2. Learning

We all learnt how to curate a dataset, perform EDA on a large dataset, how to preprocess data, handle null values, create sparse matrices and train machine learning models.

8. Contribution

Each member played a crucial part in discussions, analysis & making this report. Ananya & Manasvi performed EDA & Preprocessing. Pritish & Yash curated dataset, feature selection & its analysis. Further Regression models were trained & explored by Ananya & Yash, while classification models were looked over by Manasvi & Pritish.

Model	Accuracy	Precision	Recall	F1
Logistic Regression				
Training	0.66047	0.6767	0.66047	0.6532
Testing	0.4323	0.4169	0.4323	0.4136
Random Forest Classifier				
Training	1.0	1.0	1.0	1.0
Testing	0.4596	0.4783	0.4596	0.3848
K-Nearest Neighbours (K=38)				
Training	0.4593	0.4502	0.4593	0.3949
Testing	0.4011	0.3582	0.4011	0.3356
Decision Tree				
Training	0.47	0.43	0.47	0.43
Testing	0.46	0.42	0.46	0.42
Artificial Neural Network				
Training	0.99	0.99	0.99	0.99
Testing	0.61	0.61	0.61	0.60
Support Vector Machine				
Training	0.9925	0.9872	0.9824	0.9766
Testing	0.9538	0.9412	0.9537	0.9470
Naive Bayes				
Training	0.5523	0.5515	0.5523	0.5493
Testing	0.4121	0.4025	0.4121	0.4042

Table 4. Results of Classification Models with 11 buckets

Model	Accuracy	Precision	Recall	F1
Logistic Regression				
Training	0.6421	0.6607	0.64	0.64
Testing	0.2270	0.2170	0.2270	0.2171
Random Forest Classifier				
Training	1.0	1.0	1.0	1.0
Testing	0.7261	0.7158	0.7261	0.6685
K-Nearest Neighbours (K=29)				
Training	0.3078	0.3078	0.2919	0.2601
Testing	0.2093	0.1968	0.2093	0.1809
Decision Tree				
Training	0.34	0.36	0.34	0.32
Testing	0.25	0.23	0.25	0.22
Artificial Neural Network				
Training	0.99	0.99	0.99	0.99
Testing	0.39	0.38	0.39	0.38
Support Vector Machine				
Training	0.9589	0.9587	0.9588	0.9519
Testing	0.8773	0.8572	0.8773	0.8655
Naive Bayes				
Training	0.2197	0.2097	0.2197	0.2121
Testing	0.2294	0.2173	0.2294	0.2215

Table 5. Results of Classification Models with 21 buckets

References

[1] . The Movies Dataset, 11 2017.

[2] . Movie Genre from its Poster, 05 2018.

[3] Andrei Oghina, Mathias Breuss, Manos Tsagkias, and Maarten Rijke. Predicting imdb movie ratings using social media. pages 503–507, 04 2012.

[4] Nahid Quader, Md. Osman Gani, Dipankar Chaki, and Md. Haider Ali. A machine learning approach to predict movie box-office success. In *2017 20th International Conference of Computer and Information Technology (ICCIT)*, pages 1–7, 2017.

[5] Chuan Sun. Predict Movie Rating, 2020.