Predicting IMDb scores of movies based on their written reviews on Rotten Tomatoes

Introduction:

We are going to build a model that can predict IMDb-ratings for movies based on their written reviews. It can take any written review as an input and then predicts an IMDb-rating for that. This is especially useful for situations where you hear someone tell you about a new movie, but you don't have access to IMDb. The model can also be used for reviews of other things than movies, because it's based on the positivity of words.

The report has the following structure:

Logically Introduction introduces the task. Then in Problem Formulation the problem is formulated and properties of the data used are discussed. The Methods section talks about the machine learning methods used and the different properties of these methods. Used features and splitting the dataset is also discussed in this section. In Results and Conclusion are the results that we got from using the models and conclusion about which model is the best and about the pros and cons of the used methods.

Problem Formulation:

The IMDb dataset contains movie names, ids and their IMDb grades, plus many other non-important statistics. The rotten tomatoes dataset contains data about movies and reviews written for those movies. The data which we will be using was collected from kaggle.com. Each review marked as either positive or negative.
The model associates each word with a TF-IDF value, which signifies the importance of the

word in a review. Using this and the knowledge of which reviews are positive, and which negative and the know IMDb scores, the model has a multidimensional input based on which it can predict the IMDb score based on text reviews.

The datapoints of a review consist of IMDb scores and matrices containing the combined positivity and TF-IDF of words in the review. So, the program will be a supervised learning model with the IMDb grade working as the label of the result. For the input labels, the positivity of each review can be represented as a binary number, between (0,1) and the TF-IDF score will be a continuous variable between zero and 1. IMDb grade will be a continuous variable between 0 and 10.

Methods:

The number of datapoints will be roughly 7000 movies, each with their own IMDb grade and with roughly 100 reviews each. Since the data comes from two different datasets the data needs to first be pre-processed. We need to first remove all the movies that don't appear in either of the databases. After this we need to combine the databases into one singular data set, from which we can filter away all the statistics that we will not be using. And finally, because the data contains a lot less high/low rated movies and more positive reviews than negative, the data was up sampled so that all of these would be as balanced as possible. The dataset was combined from an IMDb movie dataset, and a rotten tomatoes movie dataset, which both were acquired from Kaggle.

We believe that the words contained in a movie review will have strong correlation with the grade that the film has received. For example, if all the reviews for a movie contain the word "good", then it is likely that the grade that the film has received will also be good. Therefore, we also believe that the positivity of each word will make for good feature vectors, since we can relatively easily estimate it, because there are more reviews that contain it classified as either positive or negative.

Chosen methods:

The methods used in this project are Logistic Regression, Linear Regression and Multi-layer Perceptron.

We constructed two frameworks which both use two different models. This is because we first needed to classify reviews as either positive or negative and then use that as an additional feature for the other model that predicts ratings.

For the first framework, we used Logistic Regression to teach the model how to classify reviews as either positive or negative and then Logical Regression to predict IMDb-ratings. We used these because Logistic Regression is good for a binary outcome, and Linear Regression for continuous.

In the second framework we used MLP for both, because it can do both. The library used was Keras from TensorFlow because it was significantly faster than MLPClassifier or MLPRegressor.

For both models the main loss function we will be using is the mean square error. We chose this because it is commonly used with regression methods, and since IMDb grades are continuous it should work well. We also considered Huber Loss but decided against it because we concluded that extreme outliers wouldn't be a major issue. Also, for clearer comparisons we will be using the coefficient of determination R^2, since it is an intuitive parameter to look at when determining how well the model works.

For splitting the data between the training set and test set, we decided to opt for an 80/20 approach. After some research we concluded that 80% training data and 20% test data seems to be a commonly used approach that works well. The upsampled data was used in the splitting to guarantee a balanced set. Regarding the size of the data this would mean that the training data would be roughly 5600 movies and test data roughly 1400 movies, with each movie having on average about 100 reviews.

Results and Conclusion

The linear regression model doesn't seem to be overfitting too much since the training MSE is 0.98 while the test MSE is 1.01, which means that the model performs similarly between data it has already seen and data that is has never seen.

The Multi-layer Perceptron on the other hand produced a MSE of 0.34 with test data and 0.30 with training data, meaning that overfitting is also not a big issue in this model.

For coefficient of determination R^2 the Linear Regression model got 0.72 and MLP got 0.91. The closer R^2 is to 1, generally the better the model is.

Since the Multi-layer Perceptron model was better according to every used indicator, it can be said that it was the better model and therefore the chosen method.

Since the MSE test error we got for the chosen Multi-layer Perceptron was 0.34, meaning that the model could predict the IMDb score on average within ±0.5 points of the actual grade.

In conclusion we have determined that written reviews and IMDb ratings for movies seem to be heavily correlated. This correlation is semi-linear as it can be decently predicted with a linear model, although better results can be obtained by using more sophisticated models, such as MLP. Our final model seems to be quite accurate at predicting the ratings, but there is still room for improvement, especially for predicting movies whose rating falls within the average of about a 6±1. This is because movies with average ratings have both positive and negative comments, so the models have trouble determining what kind of grade a singular comment from these movies suggests. Whereas highly positive or highly negative comments are much easier to distinguish. The performance of the models could be improved for example by using language analyzing models that consider combinations of words to determine the positivity or negativity, instead of the current analysis that determines it for a single word at a time.

Sources:

Data:

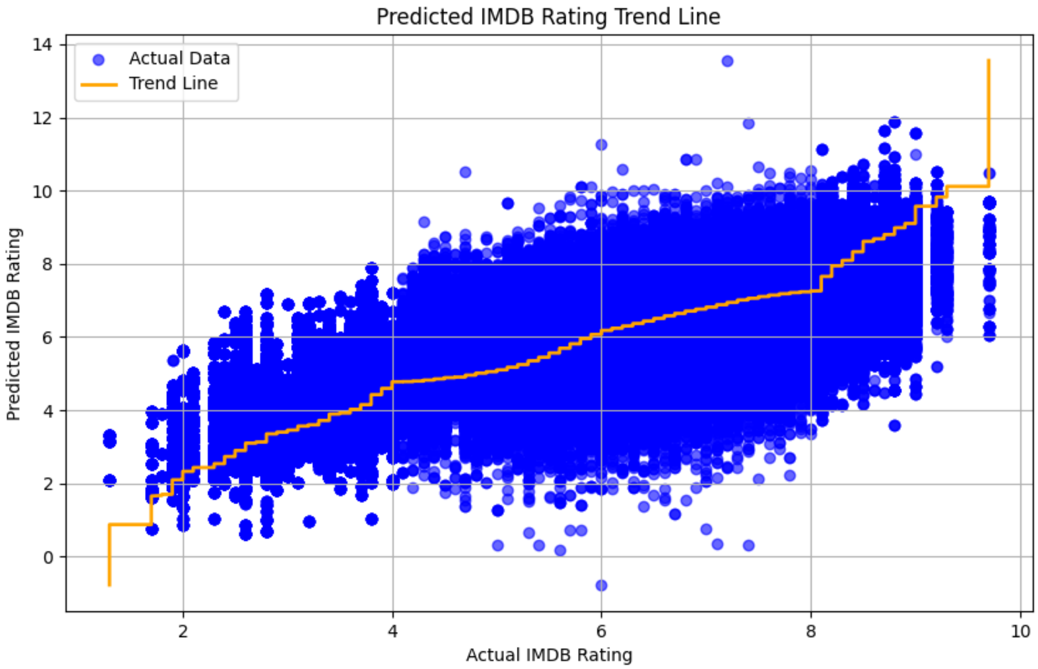
https://www.kaggle.com/datasets/andrezaza/clapper-massive-rotten-tomatoes-movies-and-reviews

https://www.kaggle.com/datasets/amanbarthwal/imdb-movies-data

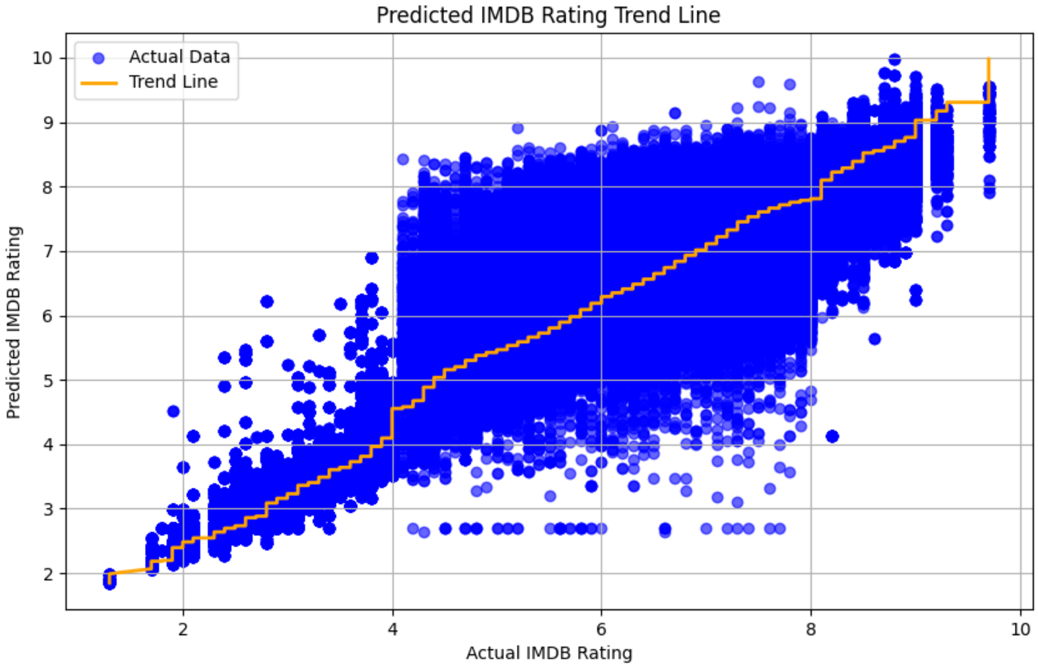
Appendix:

ChatGPT was used in generating most of the machine learning code.

Linear Regression and Logistic Regression



Multi-layer Perceptron


Predicted IMDB Rating Trend Line
Actual Data
Trend Line
Predicted IMDB Rating
Actual IMDB Rating