The Movie Quality Predictor

Yi Zhang, Ruohong Zhang, Junhan Liu, Guixing Lin

Motivation

Nowadays, people put more focus on working towards producing movies of the highest quality in the film industry to achieve commercial success, but very few studies have been done on how to make high quality films as opposed to how to achieve box office success. Aspired to discover the secret formula of producing the high quality movies, our team is going to explore multiple features of film production and advance people's understanding of this topic with machine learning models. We believe our endeavors will benefit many people especially film producers, directors, companies or students who are seeking the magic formula to craft high quality films.

Dataset

IMDB database and Wikipedia serve as our main data sources. We queried the IMDB database for all qualified movies by the following filters:

- Year > 1976
- Country = USA
- Number of Rating Votes > 500

As a result we got a total of 8942 movies with ratings ranging from 1.1 to 9.5. We further filtered out examples that have many missing data (movies that don't have box office, budget, and actor data) to improve training performance and are left with 4476 instances.

IMDb rating is widely used by critics and consumers as an indicator of movie quality relatively independent of commercial success. Therefore we rely directly on IMDB ratings to generate our movie quality label. We used two ways to label/classify instances according to their ratings 1) divide ratings into four quartiles representing four levels,

a) A: rating >= 7.1, first quartile
b) B: rating >= 6.4 second quartile
c) C: rating >= 5.7 third quartile
d) D: rating < 5.7 fourth quartile

and 2) dividing ratings into two groups, below and above top 40% line.

a) A: rating >= 6.7, top 40%, high quality
b) B: rating < 6.7, lower 60, low quality

For each movie, we collected the following attributes:

a) Year (numerical)
b) Main genre (nominal)
c) Number of genres (numerical)
d) Length (numerical)
e) Number of languages (numerical)
f) Box office (numerical)
g) Budget (numerical)

h) Award Index of top three actors/actresses (numerical)
i) Award Index of director (numerical)

Among these attributes, a) to e) can be obtained directly from the IMDB database. For attributes f) to i), we wrote a python script to collect them from Wikipedia page. Most of these attributes listed above are straightforward, probably except the last two. The original intention of the last two attributes is to take the number of awards owned by directors/casts into account. However, considering the difficulty of counting the exact number of awards, we decided to approximate it by the frequencies of some award related keywords in the Wikipedia page of directors/casts. Keywords considered for this purpose includes "oscar", "golden globe", "prize", and "awards". Although such an approximation sacrificed accuracy to some extent, the compromise did bring us the practicability to take this attribute into consideration.

Training and Testing

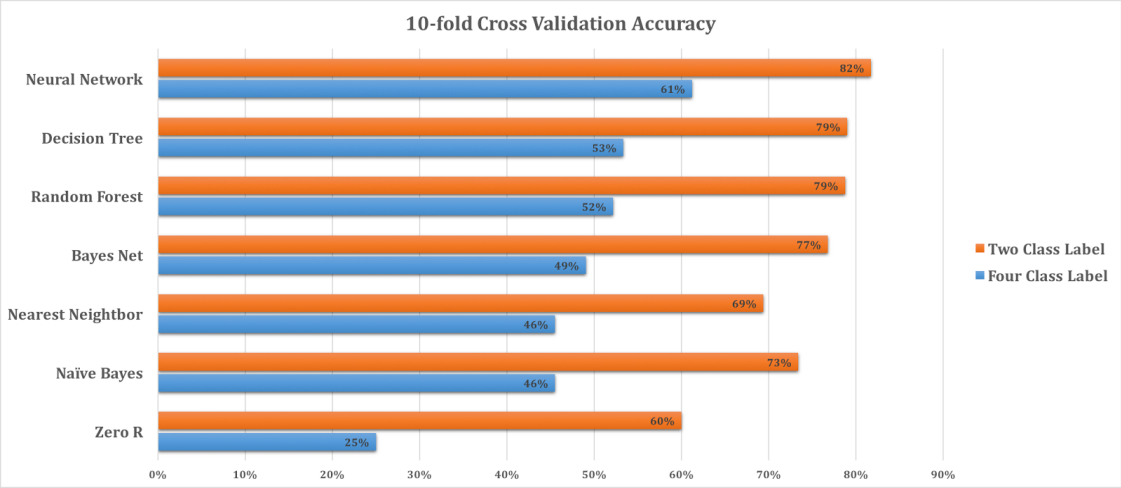
We tried a broad range of models in order to find the one that works best for our project, including Decision Tree (J48), IBK (30 nearest neighbors), Naive Bayes Classifier, Bayes Net, and Neural Network (Multilayer Perceptron). Considering we only have about five thousands data left after multiple filters, we are not able to divide our data into training set and validation set while maintaining a reasonable amount of data for both sets. Instead, we decided to use 10-fold cross validation as our measurement of success.

The multilayer perceptron model worked very well in generally, with 81.72%, 61.33% for classification on 2 categories and 4 categories respectively using the default attributes. Then a few more trials were performed to test the precision of the model with different attributes. First, the number of layers (with default value a = (6+7)/2 = 6) was modified to 10 with precision of 81.72% and 61.22%, and to 20 with precision of 81.12%, and 60.70%. The change of hidden layer doesn't have big impact on the precision, which is counter-expectation because higher precision was expected with larger number of hidden layers. Second, the learning rate (with default value 0.3) was change to 0.6 with precision of 81.41% and 59.50%, to 1.0 with precision of 80.41% and 55.94%. There was a precision drop with the increase in learning rate. The reason was that with higher learning rate, there was a higher risk for the model not to converge. Last, the momentum (with default value 0.2) was changed to 0.5 with precision of 81.38%, 60.30%. The precision was lower because with higher momentum, the model may have higher risk of being stuck in local maximum.

The decision tree model(J48), the first planed model, was also tested. The precision for decision tree on 2 categories (78%) was close to that of the multilayer perceptron(80%), but the precision for 4 categories was not high (50% with default value) compared with MP(60%), partly because the relatively small number of datasets couldn't train an accurate model for more categories.

Result

Among all the models we tried, Multilayer Perceptron produced the best accuracy.
Considering that Multilayer Perceptron does not provide insight into internal structure of the model, we used Decision Tree (J48) instead to obtain some level of understanding of how our attributes affect the final result. By looking at the generated decision tree, we found that the award indices of directors and cast play the most important role in the classification, that being said, a high-quality team of directors and actors is the golden key to success for a movie.



Future Steps

After our experiments, there are a couple of future steps that we would consider:

1) Include more features that have an empirical impact on movie quality including special effect techniques, sound/videography effects, etc. Especially, as movies are usually made by many people with myriads of roles, finding a way to factor all of the human part of movie productions should be very effective.
2) We have seen that award indices of directors and actors have proven to be very effective. We intend to refine our way of calculating the award index of actors and directors. Right now we do a simple count of a few keywords, which may be a good representation of the award concept but does differentiate or give weight to different awards and nomination versus winning status as well as the date of award.
3) Expand our dataset to include more movies of different periods, regions and kinds as well as fill in missing data for the existing dataset. This will certainly increase both the performance and usability of our model.
4) Divide the label of models into more classes like 6 or even 10 classes. This will achieve a lower testing percentage as the Zero R base percentage will be really low, but it the classification result will be more useful as the user can infer more information from the result.
5) Experiment with more training methods. Neural network has proven to be the most effective. However, there are still many different settings of this model that should be test with in hope to achieve better testing results.

Contributions

Ruohong Zhang & Guixing Lin: wrote python scripts to collect data from IMDB, performed 10-fold cross validation under different various models and different parameters
Yi Zhang & Junhan Liu: wrote python scripts to collect data from Wikipedia, wrote final report and designed website.