Midterm Competition Writeup

Junting Lyu

October 2024

1 Feature Extraction

1.1 Sentiment

I derived a sentiment score from the original text to reflect the emotional tone of the review using TextBlob. I assume that reviews with high sentiment scores are more likely to be related to a high rating, vise versa. The sentiment analysis is extremely time consuming since it needs to perform analysis on every single data. I used VADER and multiprocessing to accelerate this procedure.

1.2 Time (Year/Month/Season)

This feature aims to reflect how the reviewer's preference or criterion changing over time. The rating distribution might be different for newer and older movies. Also the release of a good movie could result in a trend of high rating across several months/seasons, vise versa.

1.3 Helpfulness

This feature indicates how much the users find the reviews helpful. I assume that a higher helpfulness rate implies that the sentiment of the comment would be highly correlated with the scores. And it could also make the distribution of scores more condensed.

2 Processing Methods

2.1 Standard Scaling

Since I used distance-based algorithm KNN, it's important to ensure that all features have the same scale. So I used StandardScale to scale each feature to have a mean of 0 and a standard deviation of 1, preventing features with large scales to dominate the model's prediction.

2.2 Dimensionality Reduction (PCA)

I used PCA to reduce the complexity of the feature space, ensuring the model to focus on the most informative components while discarding noise. PCA was applied to reduce the data to 3 principal components, which also provides a simpler feature space and improves the efficiency of computation.

2.3 KNN

I used KNN for classification to predict the final score. The hyper parameter k is tested in the range 3 to 63. I think KNN would be suitable for this task because it doesn't assume a linear decision boundary, given that we use real-world data whose relationships could be complex.

3 Potential Improvement

After several runs I realized that the score distribution in the training set is highly skewed towards 5, which resulted in overfitting. I tried resampling the training set to ensure the scores distributed evenly, but there wasn't enough time to restart from sentiment analysis.