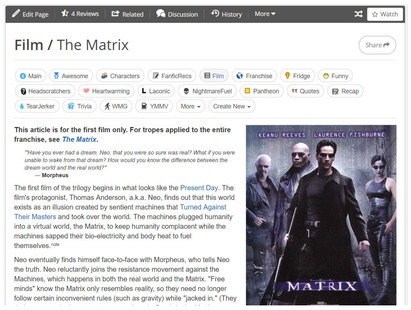
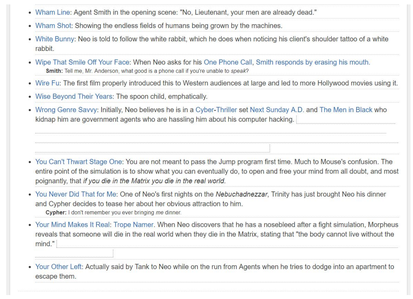
Predicting the movie popularity using user-identified tropes

Dennis Jeong (wonjeo@stanford.edu), Amy Xu (xuamyj@stanford.edu)
Stanford University

Predicting

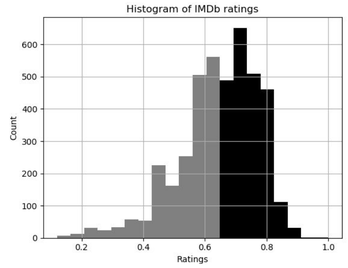
What makes a cinematic story feel compelling to an audience? We attempt to wrangle that question into an ML-friendly form and answer it using supervised and unsupervised learning. We use IMDb audience ratings data as a proxy for how compelling a movie is, and user-identified trope labels from TV Tropes to encapsulate story elements such as plot and character.

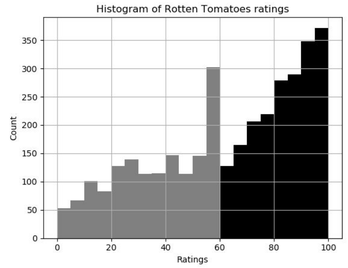




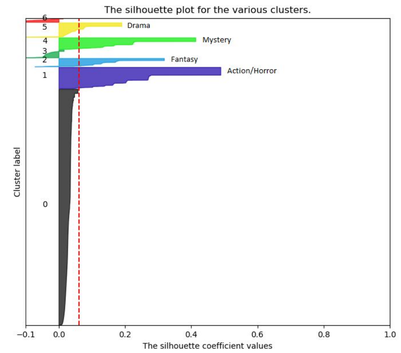
TV Tropes is an online collaborative environment where users label fictional works with common tropes, and descriptions of why each trope applies, creating a bipartite graph of tropes and works. We scraped the Linked Data Wrapper provided by DBTropes.org and built a parser for storing the movie-trope-comment relationships.

Data and Features





We used IMDb and Rotten Tomatoes ratings to generate binary labels for classifications. For IMDb, we used the median score as the cutoff. For Rotten Tomatoes, we used the built-in cutoff of 60% for being "fresh" or "rotten." We chose to use binary classification instead of linear regression since numerical rankings create arbitrary granularity within a measure that is inherently subjective.



We applied k-means to our data and found that k=7 generated the best silhouette score with four distinct clusters. These clusters roughly correspond with the categories of drama, mystery, fantasy, action/horror.

Models

Naive Bayes

$$p(y=1|x) \;=\; \frac{\prod_{i=1}^{n} p(x_i|y=1)p(y=1)}{\prod_{i=1}^{n} p(x_i|y=1)p(y=1) + \prod_{i=1}^{n} p(x_i|y=0)p(y=0)}$$

We chose to begin with the multinomial naive Bayes model because it was a fast and effective way of getting a baseline of test and training accuracy.
Naive Bayes assumes that each feature is independent of the other features, conditioned on the label.

Logistic Regression

$$p(y \mid x; \theta) = (h_\theta(x))^y \, (1 - h_\theta(x))^{1-y}$$

The naive Bayes model had promising results, so we decided to further pursue binary classification by applying logistic regression.
Logistic regression assumes:

1. The data is distributed as a Bernoulli variable.
2. The model predicts the mean of that random variable.

Results

	precis.	recall	accur.
Naive Bayes			
IMDb	0.65	0.65	0.64
Rotten Tomatoes	0.79	0.67	0.65
Log. Regression			
IMDb	0.57	0.67	0.64
Rotten Tomatoes	0.74	0.67	0.64

Table 1: Test results using only trope features for naive Bayes and logistic regression models.

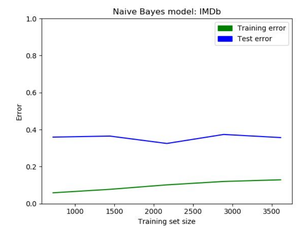
	precis.	recall	accur.
Naive Bayes			
IMDb	0.61	0.63	0.62
Rotten Tomatoes	0.77	0.65	0.63
Log. Regression			
IMDb	0.60	0.66	0.64
Rotten Tomatoes	0.70	0.64	0.61

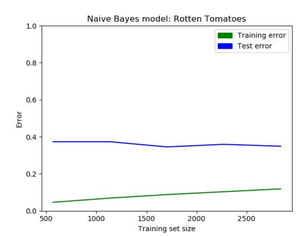
Table 2: Test results using both trope and comment features for naive Bayes and logistic regression models.

Our base feature set consisted of trope labels. We experimented with this feature set by adding text frequency elements and our k-means labels. We found that:

1. Logistic regression performed better with IMDb labels and worse with Rotten Tomatoes labels. This may be because the IMDb data fits better with the underlying assumptions of logistic regression.
2. Adding the text frequency elements led to worse performance overall, likely due to high variance.

Discussion and Future





We ran diagnostics to determine our next steps:

1. The gap between training and test error indicates overfitting, which we would like to address by adding regularization to our model and reducing the feature dimensions.
2. Test error decreases slightly as data set size increases for the Rotten Tomatoes labels. This indicates that more data could be helpful, but we are already using all the data that exists.