Predicting the 85th Academy Awards

Stephen Barber, Kasey Le, Sean O'Donnell

December 13, 2012

As the oldest awards ceremony in the media, the Academy Awards is responsible for bestowing the penultimate achievement in cinema, the Oscar, with which comes distinction and prestige. This immense potential for positive exposure is a marketer's dream and so every year, studios will spend millions to promote their films during Oscar Season, enough to draw accusations that perhaps the Academy Awards is more a promotional scheme than recognizing superior quality film. Using a training set composed of every film to have ever been nominated for an Oscar in four target categories, we aim to develop an effective learning algorithm to reproduce the results of past Oscars, predict the results of the upcoming 85th Academy Awards (to be held February 2013), and attempt to determine the features that contribute the most to an Academy Award victory.

1 Data and Features

We wished to use data that could easily be collected for any movie within the last century, which limited us to more conventional and standardized film attributes. Since we needed qualitative critique that would be available for any movie that could have been nominated for an Academy Award, we decided to focus on aggregated ratings, such as from audience and critics on Rotten Tomatoes and IMDb. Our choice of training data is determined by a master list of all movies that have ever earned at least a nomination in the Best Film, Best Director, Best Actor, or Best Actress categories of the Academy Awards.

1.1 Data Set Generation

IMDB and Rotten Tomatoes (RT) served as our primary sources of feature information; first we aggregated a list of every film nominated for an Oscar in the (colloquial) categories of Best Film, Best Director, Best Actor, and Best Actress, and then we used these film titles and their respective release years to search for and acquire respective film features, using the unofficial RT API and our own custom web-scraping script for IMDb. We have aggregated, in all, the IMDb and RT (both audience and critic) ratings, release month, writer, director, cast, MPAA rating, year, run time, and genre feature data from our master list of collected film titles, which is nearly 1000 films in total. All collected data has been discretized and binned accordingly, and we use extensive binary labeling to indicate the presence of an actor/actress in a movie, as well as labeling our known Academy Award victories for each film, resulting in a sparse set of 4408 features.

2 Testing Methods

For testing, we began using hold-out cross validation where we trained on movies released before 2000 and tested on movies released after 2000. For our specific problem, this division of data made sense as the real-life application of our algorithm would be to predict future Academy Award nominations. We also wanted to take in account for any kind of growth or evolution in the judging process from year to year. Furthermore, since nominations are selected from a pool of movies all released in the same year, it makes sense that year is not just a feature of the movie but also a way to group movies that are in competition with each other.

For robustness, we also begin using k-fold cross validation with 10 subsets. In general, we found that our models had the same success under either validation method. However, since we were often testing multiple models with multiple variations, we found k-fold cross validation too computationally expensive to be of much use. Thus, all stated results will be from hold-out cross validation.

To better evaluate our data, we also opted to use Matthews correlation coefficient as our measure of performance since we were performing binary classifica-

tions. Additionally, given that our data is skewed with a small fraction of positive examples, we wanted to put less emphasis on accuracy and instead use a balanced measure of precision and recall. The coefficient returns a value between -1 and +1 with +1 indicating a perfect prediction, 0 indicating a random prediction, and -1 indicating a completely opposite prediction.

3 Learning Algorithms

We began by using quick and dirty implementations of the Gaussian Naive Bayes Classifier, for simplicity and speed, and a SVM Linear and Polynomial Classifier (2-9 degrees) for robustness and flexibility of implementation. Since we had no prior knowledge or assumptions about the distribution of our data, we wanted to cast a wide net and work with generic models to start.

We had some initial success using polynomial SVMs and thus we decided to also implement Logistic Regression as an alternative objective function optimization classifier. The Gaussian Naive Bayes did not perform as well as the SVM, so we began to use the Multinomial Naive Bayes model instead which assumes less about the distribution of the data.

The results of these initial tests can be seen in the first four columns of each category graph in the results section (4).

3.1 SVM Exploration

In the first runs through our data, we found that polynomial SVMs performed the best out of all the basic models. Thus, we decided to explore SVMs further and try to optimize their performance over our data. Although we tried to reduce the range of values of each feature during our data set generation, we had to allow enough values to maintain distinct years and ratings. Thus, our feature vector was uneven with the union of these features and our many binary vectors. As a result, we tried to normalize our feature vectors to assure that no feature was intrinsically weighted more based solely on its wider range of values. However, this did not improve performance, so we reverted back to our original vectors.

Next, we experimented with different types of kernels to try to better model the distribution of our data. We applied the Gaussian radial basis function as a kernel to see how normal our data was. This resulted in much worse performance of the SVM, definitively concluding that our data is not normal. As further evidence, Gaussian Naive Bayes performed the worst out of all our models and significantly poorer than Multinomial Naive Bayes. We ended up using a polynomial kernel for our SVM and optimized the degree for each category.

3.2 Feature and Model Selection

After discovering that each of the categories had varied success with each of the different classifiers, we decided to separately optimize the features and models used for each category. Motivated by the fact that our feature set was over four times the size of our training set, we implemented feature selection using backwards search. We counted the binary vectors for cast, writers, and directors as one feature each, resulting in a total of 11 features. We wrapped our feature selection around all our models including logistic regression, SVM with polynomial kernels of degree 4 and 5, Gaussian Naive Bayes, and Multinomial Naive Bayes. Although computationally expensive, we ran the process over all models, enabling us to simultaneously select the best features and model for each category. We used hold-out cross validation and Matthews Correlation Coefficient as our score to measure the performance of each iteration. This resulted in the following optimal model and features for each category:

Best Actress: SVM with a polynomial kernel of degree 4 without audience rating
Best Actor: Multinomial Naive Bayes without director, cast, critic rating, and runtime
Best Director: Logistic Regression without audience rating, release month, and runtime
Best Film: Logistic Regression without audience rating, cast, and genre

With feature and model selection, we found improvement in the Matthews Correlation Coefficient in all four categories. It was interesting to see that the best model changed from SVM to either Logistic Regression or Multinomial Naive Bayes in three of the four categories as we reduced the dimension of our feature vectors.

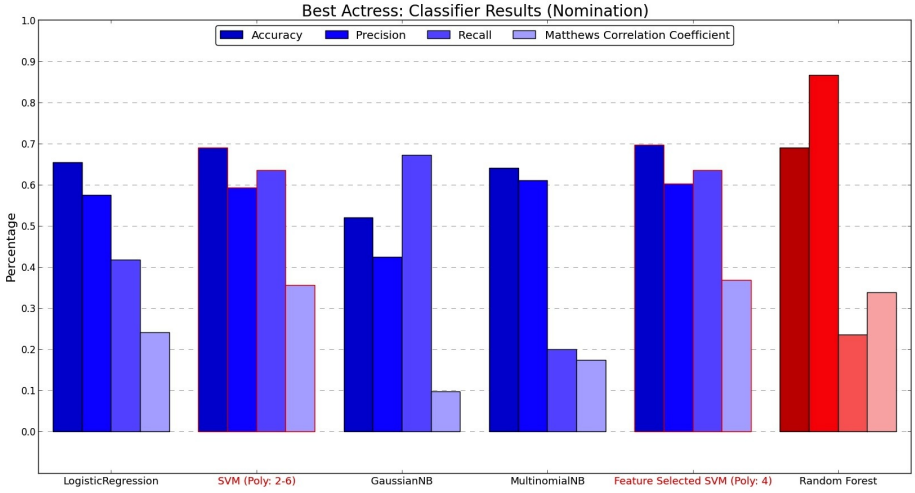
For curiosity's sake, we also did an ablative analysis of the features to identify the ones with the most predictive power. The two most predictive features were MPAA rating and the Rotten Tomatoes critic ratings for Best Film, IMBd rating and cast for Best Director, director and genre for Best Actress, and release month and genre for Best Actor. Genre was overwhelmingly

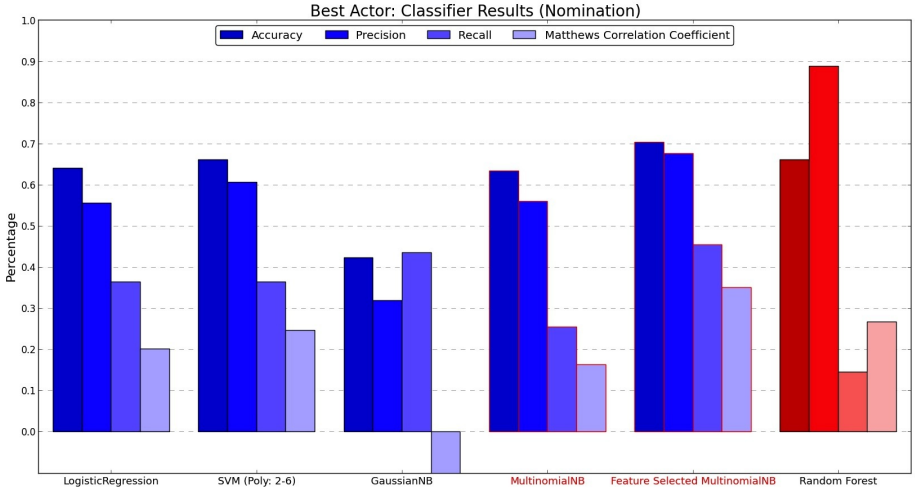
the most significant factor for Best Actress and Actor.

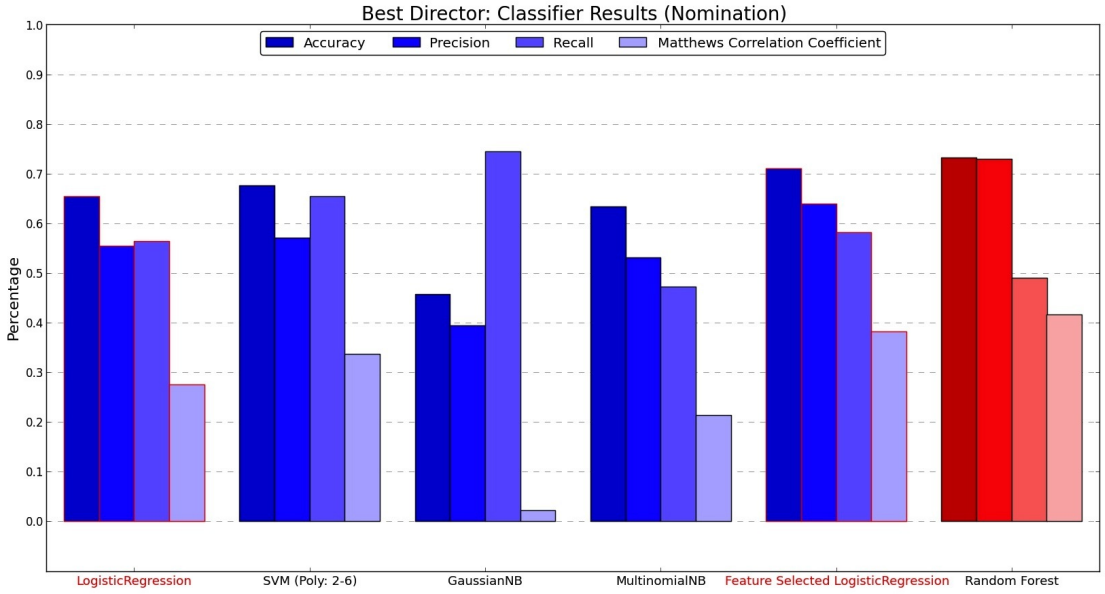
3.3 Semi-Supervised Learning and Random Forests

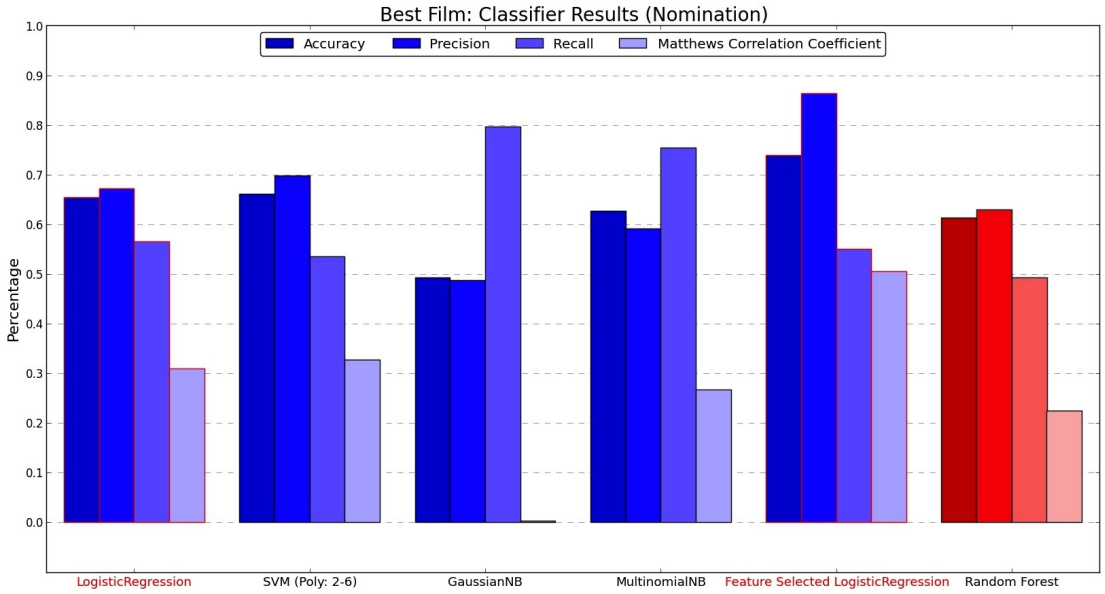
In an attempt to potentially better model the underlying distribution of data (which has not be well-modeled by the Gaussian distribution), we attempted a semi-supervised Naive Bayes classifier using EM (Nigam et. al, 2006), and Random Forests, a favorite of Kaggle contestants looking for an effective algorithm that does not require much dataset tuning. We trained our semi-supervised classifier on a small fraction of our labeled dataset, labeled a larger fraction of an unlabeled dataset, then trained our classifier on the union of our pre-labeled and classifier-labeled dataset, repeating these two steps until convergence. However, it preliminary results showed that it performed no better than our Multinomial Naive Bayes classifier, so we discontinued its use - perhaps our dataset was too fundamentally different from the text dataset used by Nigam, to have benefitted.

4 Results









Red outline/name indicates best of 4 basic models, which is enhanced through feature selection. Compared against (red) Random Forest model to judge optimality of feature selection, considering Random Forest's strong feature handling.

4.1 Conclusions and Discussion

We were not able to produce as strong of a classifier as we would have liked, but the results of our models and feature selections still hold various implications for the Academy Awards nomination and selection process. Excluding different features for each Academy Awards category overall produced better results, yielding numerous insights. Some exclusions were intuitive in hindsight, such as runtime became a confounding feature for 2 of 4 categories. Some exclusions were surprising: audience rating confounded 3 of 4 categories, implying perhaps that audiences are maybe too selective, or maybe too undiscriminating, or even that judges makes selections differently from the average movie reviewer. Either way, it seems assuring that both cast list and genre hold little importance in nominating a Best Film - an Oscar worthy film seems to still need something more. Our difficulty in selecting an accurate model built upon a supposed dataset distribution could be explained by the sheer difficulty in winning an Academy Award in the first place: considering that the most Oscar worthy films tend to walk away with the most awards, and (for example) with 610 movies released in 2011 and only 9 Best Film nominations for 2011 movies, Oscar nominations likely follow some kind of power law distribution, which is more difficult to model with most well-established machine learning algorithms, at least compared to Gaussian and Multinomial distributions. Future work in improving an Academy Awards classifier would likely require even more careful, in-depth feature selection, with a well-designed neural network to more closely model any underlying distribution.

4.2 Predictions for the 85th Academy Awards

As a fun end to our project, we wanted to make predictions for the 85th Academy Awards being held on February 2013. Using the model and features we selected for the category of Best Film, we evaluated the likelihood of some major films from 2012 being nominated for or winning in the category. The following table lists some of the top films most likely to be nominated.

Film Title	Likelihood of Nomination
Life of Pi	70.8%
Les Miserables	65.4%
Zero Dark Thirty	65.0%
Wreck-it Ralph	54.3%
Skyfall	42.1%
Silver Linings Playbook	36.9%
The Sessions	36.8%
Moonrise Kingdom	36.2%
Mirror Mirror	35.4%
Looper	35.0%
The Amazing Spider-Man	34.3%
Argo	34.2%
The Dark Knight Rises	32.5%
Marvel's The Avengers	3.16%
Safety Not Guaranteed	3.13%
Beasts of the Southern Wild	2.98%
Playing for Keeps	27.0%
Rock of Ages	16.1%
Men in Black III	15.5%
Wrath of the Titans	14.4%
Lincoln	9.8%
Red Dawn	7.0%
Amour	5.1%

5 References

Kamal Nigam, Andrew McCallum, and Tom M. Mitchell. Semi-Supervised Text Classification Using EM. 2006. //pal.sri.com/CALOfiles/cstore/PAL-publications/calo/2005/semisup-em.pdf

MPAA. 2011 Theatrical Market Statistics. 2012. http://www.mpaa.org/Resources/5bec4ac9-a95e-443b-987b-bff6fb5455a9.pdf

Mathieu Blondel. Semi-supervised Naive Bayes in Python. Mathieus Log, June 21, 2010. http://www.mblondel.org/journal/2010/06/21/semi-supervised-naive-bayes-in-python/

Pedregosa et al. Scikit-learn: Machine Learning in Python. 2011. http://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html

Made extensive usage of: http://www.imdb.com/ oscars/nominations/