Predicting Academy Award Winners

Firas Abuzaid
fabuzaid@stanford.edu

Emily Cheng
emcheng@stanford.edu

Omosola Odetunde
omosolao@stanford.edu

December 12, 2014

1 Introduction

Sometimes a dark horse comes along; other times, the winner is the favored nominee. Every year, film critics and media buffs across the nation make their predictions about the winners of the Academy Awards. Some of these critics lambaste the Awards saying that the award selection is formulaic and the same types of films win each year. With this opinion (and its opposition) in mind, we wanted to investigate the predictability of the Oscars. Are there in fact predictive features of a nominee which make him, her, or it more likely to ultimately win? We sought to explore this question for our final project.

2 Task Definition

In this project, we aimed to build an effective set of models to predict the results of past Academy Award winners given a list of nominees in each category. We built 24 separate models for each of the 24 Academy Award categories. Upon constructing the models, we evaluated the accuracy of our models in predicting the last five years of Academy Award winners.

For our prediction tasks (per award category), our inputs were a set of feature vectors, each representing one of the nominees within the given category. Because our features were mostly categorical (e.g. genre), we used a sparse feature vector of indicator variables (e.g. "directorIs_StevenSpielberg=1"). Our model calculated a probability for each nominee of it being the winner in its category, and the nominee with the highest probability was returned as the winner. The output of our model was a value of either 0 or 1: 1 if the nominee was predicted to be the winner of its category, and 0 otherwise.

3 Literature Review

We gained inspiration and insight from several sources in the course of our research, but found the two articles *Applying discrete choice models to predict Academy Award winners* and *Identifying Predictive Structures in Relational Data Using Multiple Instance Learning* to be the most informative for our project.

In *Applying discrete choice models to predict Academy Award winners*, Pardoe applies discrete choice models in modeling Academy Award, which is complementary to our approach [2]. This paper distinguishes between who *will* win and who *should* win; the former is the actual goal of our prediction task. As a result, some expected useful information (e.g. critic reviews), may actual add noise to our predictions. In addition, the paper discusses how different categories (eg. Best Actor vs Best Actress) had different predictors, and furthermore, how the changes in parameter estimates did not always follow the same trends. This knowledge was useful when applied to our task. It suggested that as we narrow down our feature set, we may arrive at entirely different feature vectors for each of our award categories.

In *Identifying Predictive Structures in Relational Data Using Multiple Instance Learning*, McGovern and Jensen take a different approach than ours, using Multiple Instance Learning (MIL) to approach this task [3]. MIL is used in their research to identify predictive structures in the large IMDB database. McGovern states that MIL can detect complex relations and structures that cannot be represented by simply flattening the relational data into feature vector format. This drawback is worthwhile to note as we evaluate the efficacy our models since our models uses these feature vectors; however, it is also relevant to note that McGovern built a complex relational database to solve this task, whereas we are working with scraped data stored in JSON format that does not maintain the relationship structure between films.

4 Data

All data for our project was collected from IMDB.com. As IMDB.com does not have a public API, we wrote a scraper to pull both our nominee/winner data as well as our individual features from the site. Our data set contains information for all Academy Awards from 1980 until 2014.

We partitioned our data into training and test sets. For our training set, we used nominated films from 1980 until 2009. For our test set, we used nominated films from 2010 to 2014.

5 Models & Algorithms

5.1 Baseline

In our baseline implementation, we used random selection in order to determine the winner in a given category. We wanted to ensure that any improvements in our accuracies performed better than random. For each category and year pair, we selected one of the nominees as the winner in the category uniformly at random.

5.1.1 Baseline Results

Our baseline approach resulted in an average classification accuracy of 20.87% across 100 runs. This was a reasonable result as the average number of nominees in a given category within our data set is 5 and the probability of randomly selecting the correct winner of these 5 nominees is 20%.

5.2 Oracle

For our oracle, we consulted the 2014 predictions as published online by "Yahoo Movies" experts [1]. These predictions seemed to be a good oracle for our research, as they were high in accuracy, but also human-achieved. The experts' performance seemed to be a good and potentially achievable goal for our models.

5.2.1 Oracle Results

The movie experts predicted the 2014 award winners, given the nominees, with an accuracy rate of 87.5%, which shows there is certainly predictability in the winners.

5.3 Multinominal & Bernoulli Naïve Bayes

The first algorithm we implemented was Multinominal Naïve Bayes, which calculates the likelihood of a particular film winning the Academy Award for a given category. The Naïve Bayes model relies on the assumption that the value of a particular feature is independent of the value of other features. Although this assumption contradicts our basic intuition for this problem, the Naïve Bayes approach has historically outperformed these low expectations; therefore, we considered it to be a good first choice for constructing our initial predictor.

In our implementation, we trained our Multinominal Naïve Bayes model on our training data, using the feature sets described in "Feature Extraction" below.

We applied our trained Multinomial Naïve Bayes model to our test data in order to calculate the probability of each nominee in a particular category being the winner. We then selected the film with the highest probability as our predicted winner.

As a concrete example our our Naïve Bayes results (log probabilities), we include below output from running Multinominal Naïve Bayes on the category Best Performance by an Actor in a Supporting Role for the 2014 Academy Awards.

```
{'Best Performance by an Actor in a Supporting Role' :  [
'The Wolf of Wall Street':  -2.53176234628,
'American Hustle':  -1.00595799588,
'12 Years a Slave':  -4.25495853286,
'Captain Phillips':  -2.70875362182,
'Dallas Buyers Club':  -3.12255620033] }
```

In this example, based on our classifier, we would select 'American Hustle' for 'Best Performance by an Actor in a Supporting Role'.

We chose a Multinomial model, rather than a Bernoulli model, for our initial implementation, because we wanted to capture the frequency of particular features. In particular, this is necessary to model actors, actresses, directors, writers, and films who have won multiple awards in the past. However, we ultimately decided that a Multinomial model was not necessary given our final feature set, since all of the features were indicator features. It made more sense to use a Bernoulli Naïve Bayes model, which treats features as binary variables.

5.4 Logistic Regression

For our second algorithm, we implemented Logistic Regression, another machine learning algorithm which is also commonly used for binary classification. Unlike the Naïve Bayes approach, Logistic Regression attempts to learn a fitted curve – typical, a sigmoid curve – to distinguish between positive and negative training examples. The regression coefficients of the curve are learned using Maximum Likelihood Estimation. We used an L2 regularization in our implementation to combat over-fitting during training.

However, similar to Naïve Bayes, Logistic Regression also relies on a feature vector to generate a particular probability, and it also takes into consideration the multiplicity of different features. Logistic Regression was a sound choice for us as our second model as it can be used to effectively predict binary categorical outputs.

5.5 SVM

We utilized SVM as a third supervised algorithm which we thought might improve classification accuracy. We experimented with several types of kernels, including a radial-based function and a polynomial function; however, we settled on a linear kernel as it provided the best classification results. We used L2 regularization, just as we did for Logistic Regression. The SVM implementation we used provided us additional flexibility over Logistic Regression as it permitted us to experiment with the shape of the decision boundary used in our predictions.

5.6 Unsupervised Learning with K-Means

Finally, we wanted to explore whether our collected data had any hidden or latent structure amongst the nominees and categories, and learn a predictor without using the labels in our data set. To determine this, we implemented the K-Means algorithm, and we parametrized it two clusters: C_{winner} for nominees who won their nominations, and C_{loser} for nominees who didn't win. We hypothesized that, if we were able to cluster our training data into these two separate clusters (without using any labels), then we could make predictions on our test set simply by computing the nominees which had the minimum distance to C_{winner}.

To seed the initial iteration of our K-Means algorithm, we used k-means++, rather than a random initialization, which has been shown to increase the likelihood of avoiding poor clusterings [4].

5.7 Prediction

For all of our models, except for our baseline and oracle, we used the computed probabilities produced during testing. For each category, we computed the probability that each of the nominees was a winner.

We selected the highest probability nominee as the winner in the particular category.

6 Feature Extraction & Error Analysis

6.1 First Pass

For our initial feature set, we chose film data which was consistently available on IMDB.com and seemed as if it might be predictive. We ended up selecting the following features: MPAA rating, film length, actors, director, producer, writer, and genre. An example film nominee feature vector is as below:

`'Gravity': {'genres|Sci-Fi,Thriller': 1.0, 'producers|Warner Bros.': 1.0, 'length=90': 1.0, 'actors|Sandra Bullock,George Clooney': 1.0}`

6.1.1 Results

With this feature set, we achieved an accuracy of around 20.8% for our Logistic Regression model and 33.0% for our Naïve Bayes model. Naïve Bayes showed a significant increase over our baseline implementation; however, Logistic Regression did not. There were several issues that we noticed with our initial feature extraction pass, so we decided to tackle these issues in order to address our low accuracies.

6.1.2 Error Analysis

An analysis of our data revealed the main issue with our feature selection: our features were too specific, creating a very sparse feature set. The majority of features in our test set were not seen in our training set, resulting in very short feature vectors which retained very little information. Examining the feature vectors in the training and test sets, we noted that features such as 'length=113' and 'genres—Biography,Drama,History' were very specific, especially given that, oftentimes, Oscar-worthy films can have new breakout actors, or other unique, distinguishing qualities. To provide a concrete example, the film "Her" had a feature length of 1: `'Her': 'rating=R': 1.0` in our test set, because the actors, producers, writers, and directors never appeared in our training set.

The weight vectors also helped inform our feature selection decisions. All the rating features were negative except for 'rating=Approved', an antiquted and obsolete rating that appeared in our training set (which goes back to 1980), but has no predictive relevance for our test set from 2010 to 2014. In fact, 'rating=R' was the most negative weight in the weight vector, and a manual analysis of our films showed that many ultimate winners were rated R, meaning it is likely this feature hurt the accuracy of our model. Concrete examples include: `'rating=Approved': 0.221104, 'rating=PG-13': -0.335083, 'rating=R': -0.500413`

6.2 Second Pass

In order to help address the problem of sparsity, we began discretizing our features. As an example, for film length (minutes), rather than having a specific feature 'length=113', we discretized it into several buckets: `'length<60', 'length=60-120', 'length=120-180, 'length=180+'`

We also separated each actor into a feature, rather than grouping all actors in a film into a single feature as we had originally done (e.g. {`'actors|Sandra Bullock,George Clooney': 1.0`}). This was similarly done for genres, writers, producers, and directors.

Lastly, we removed movie ratings after our analysis suggested that this feature did not, in fact, have any predictive power in our model. In sum, here is an example of a revised feature vector after our second pass:

`'Inception': {'genres|Action': 1.0, 'genres|Thriller': 1.0, 'genres|Mystery': 1.0,`

‘genres|Sci-Fi’: 1.0, ‘producers|Legendary Pictures’: 1.0, ‘producers|Warner Bros.’: 1.0, ‘length180+’: 1.0, ‘actors|Leonardo DiCaprio’: 1.0}

6.2.1 Results

After our second pass of feature extraction, we achieved accuracies of around 27.0% for our Logistic Regression model and 33.0% for our Naïve Bayes model. Naïve Bayes did not show a significant improvement from the previous pass; however, Logistic Regression did.

6.2.2 Error Analysis

For the error analysis of our second pass, we noticed a few areas of our feature set which could be further improved.

During our analysis, we confirmed that a majority of weights of our feature vector had reasonable predictive value and were in line with what we'd expect for notable Hollywood actors: ‘actors|Morgan Freeman’: 0.6215, ‘actors|Meryl Streep’: 0.3739. Additionally, certain genres – particularly genres with niche appeal – had high feature weights: ‘genres|Mystery’: 0.395271, ‘genres|War’: 0.388670

In contrast, we noticed that many film features had unexpected weights (some genres, directors, etc.). For example, genre|Drama had a very negative weight, even though many Oscar winners are often dramas – the same held true for genre|Comedy. Specifically, we saw the following unexpected negative weights: ‘genres|Drama’: -0.692709, ‘genres|Comedy’: -0.276933.

This occurred because most Oscar-nominated films in our data set were dramas. Though most winners were dramas, most nominees which did not win were also dramas and the number of losing nominees far exceeds the number of winning nominees. As a result, our models learned a negative weight for genre|Drama, though genre|Drama does not actually provide us significant information about whether or not a film will win.

6.3 Third Pass

In our third pass, we scraped and added additional features which we expected to have predictive power.

We focused on features which indicated previous accolades that an award had received. Specifically, we added features for BAFTA, Golden Globe, Writers Guild, Screen Actors Guild, and Critic's Choice nominations and wins, data scraped from IMDb. We were careful to only include awards in our feature vector if the award ceremony occurs prior to the Academy Awards. We also added a feature representing the number of other Oscar nominations the nominee received. An example film nominee feature vector is as below:

‘Inception’: {‘num_other_oscar_nominations’: 4.0, ‘num_bafta_nominations’: 4.0, ‘num_bafta_awards’: 3.0, ‘num_golden_globe_nominations’: 4.0, ‘genres|Action’: 1.0, ‘genres|Thriller’:1.0, ‘genres|Mystery’: 1.0, ‘genres|Sci-Fi’: 1.0, ‘producers|Legendary Pictures’: 1.0, ‘producers|Warner Bros.’: 1.0, ‘length180+’: 1.0, ‘actors|Leonardo DiCaprio’: 1.0}

6.3.1 Results

At this stage, all of our features were indicator variables. As a result, Multinomial Naïve Bayes was no longer necessary and we changed to Bernoulli Naïve Bayes where everything is binarized.

Our Bernoulli Naïve Bayes model achieved an accuracy of 41.7% and our Logistic Regression model achieved an accuracy of 50%.

We removed the Drama and Comedy genres, which suffered from the issues previously mentioned. After removing these two categories, we saw an accuracy increase from 37.0% to 41.7% in our Naïve Bayes model, and an increase to 50% in our Logistic Regression model.

Further, we added in two features for each of the additional award ceremonies - one representing the number of nominations and one representing the number of wins. After adding these features, our accuracies improved over the Second Pass values, exhibiting a leap from 27% to 37% and 33% to 42% accuracy in our Naive Bayes model and Logistic Regression model, respectively.

6.3.2 Error Analysis

An initial pass of our weight vector after training shows that the awards features were the strongest predictor of a film's success of winning. The most positively weighted feature was: `'num_golden_globe_awards=4': 0.845475` and the most negatively weighted feature was: `'num_sag_awards=0': -0.566302`, which logically reflects the real-world scenario where winning many awards is a good indicator of future success and winning no awards is, conversely, a bad one. This jump in accuracy for both algorithms can be attributed to this modification of our model.

However, given our model is more then 30% away from achieving the accuracy of the oracle, it is important to analyze the data to find examples that could give insight as to why this is the case. Error analysis of specific incorrect classifications shows that in the majority of misclassified films, the 2nd highest weighted film, that is, our next best guess for the winner of the award out of the nominees, would have been the correct prediction.

One example is the Best Performance by an Actress in a Supporting Role Oscar award. The correct winner is 'Gravity' versus our prediction of 'American Hustle'. Here's an example:

Film Nominee	Log Probabilities
American Hustle	-1.02949080205
Gravity	-1.1817731833
12 Years a Slave	-1.3753770261
The Wolf of Wall Street	-2.71212986455
Nebraska	-4.34354796571

We then asked the question: what caused 'American Hustle' to not have surpassed 'Gravity' as our predicted winner? Our error analysis reveals that Gravity has the feature `'num_bafta_nominations=5'`, which has weight 0.294295, whereas American Hustle has the feature `'num_bafta_nominations=6'` with a much higher weight 0.685954. From our training data, we learned weights: 6 BAFTAs: 0.69, a much higher weight than 5 BAFTAs: 0.29. However, is one extra nomination a significant difference? Instead, it may again be useful to discretize the number of awards won, similar to how we discretized the length of the film, e.g. `'nominations|0', 'nominations|btwn1-3', 'nominations>=4'`. Or, at an extreme, we could introduce the following pair of binary features: `'bafta_nomination', 'no_bafta_nomination'`.

So, while this demonstrates that further potential improvements could be made for our models, analysis we do have confirmation that our models are somewhat accurate; by and large, we are either predicting the correct winner, or our 'runner-up' prediction is in fact the correct winner.

6.4 Final Feature Set

6.4.1 Indicator Features on Awards

The value 'x' in these features indicate either the number of nominations or the number of wins the Oscar nominee has received for a particular award. For example, in our data set, the film *Terms of Endearment* received 4 Golden Globe awards. As a result, it has 'num_golden_globe_awards=4: 1.0' in its feature vector.

- $\mathbb{1}$[num_golden_globe_nominations = x], $\mathbb{1}$[num_golden_globe_awards = x]
- $\mathbb{1}$[num_bafta_nominations = x], $\mathbb{1}$[num_bafta_awards = x]
- $\mathbb{1}$[num_critics_choice_nominations = x], $\mathbb{1}$[num_critics_choice_awards = x]

- $\mathbb{1}$[num_sag_nominations = x],, $\mathbb{1}$[num_sag_awards = x]
- $\mathbb{1}$[num_other_oscar_nominations = x]

6.5 Additional Indicator Features

Other than film length and title, the features listed below can represent several indicator features for a given film. For example, *Terms of Endearment*, our example from before, has three actors which were scraped while collecting our data on the film. Therefore, the film has three individual indicator features in its representative feature vector (`'actors|Debra Winger': 1`, `'actors|Shirley MacLaine': 1`, `'actors|Jack Nicholson': 1`).

- $\mathbb{1}$`[length` $\leq$ `180]` (length of film is less than or equal to 180 minutes)
- $\mathbb{1}$[title = x] (where x is the title of the nominated film)
- $\mathbb{1}$[producers = x]
- $\mathbb{1}$[actors = x]
- $\mathbb{1}$[recipients = x]
- $\mathbb{1}$[writers = x]
- $\mathbb{1}$[directors = x]

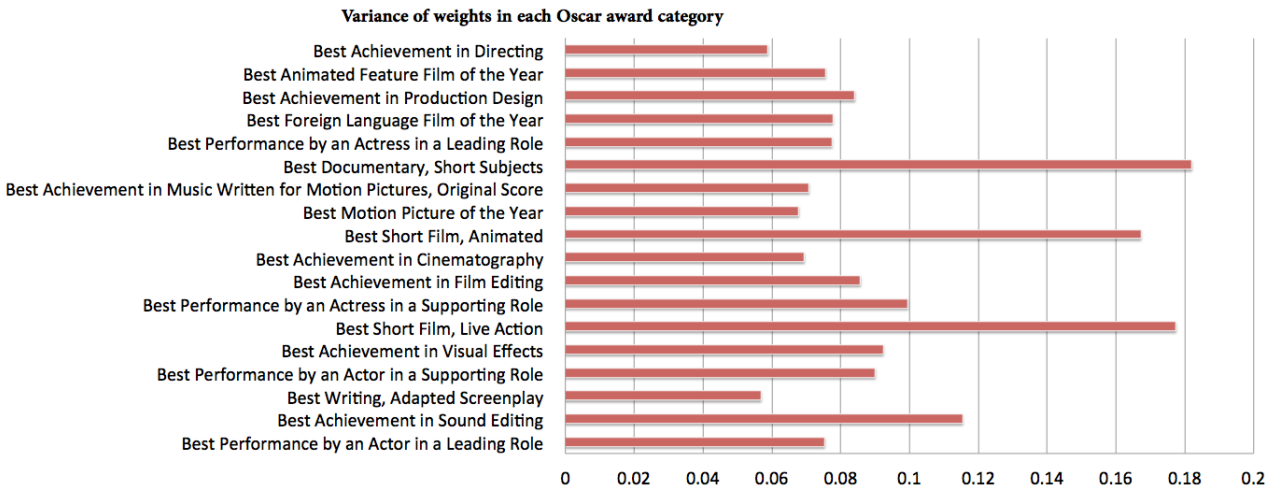

Figure 1: The variance in feature weights measured for our Logistic Regression models for each of the four categories. Interestingly, the variance is much higher for the more esoteric categories, such as "Best Documentary, Short Subjects" and "Best short Film, Live Action". In contrast, the categories that typically receive more nominees and are more noteworthy to the public (e.g. "Best Motion Picture of the Year") had a relatively low variance – our Logistic Regression predictors were less confident for these particular awards.

7 Challenges

7.1 Feature Selection

Our main challenge in our project lay in exploring and narrowing down our feature set: determining which features were useful for our model, given that there is a huge range of film features. We iterated through feature set development (as detailed above), continuously performing error analysis on our results and adjusting or improving the set of features based on our analysis.

We began using a set of basic film details (MPAA rating, film length, actors, director, producer, writer, genre) and trained and tested our algorithms using this set. Through in-depth error analysis, we pinpointed features which should be added and other features which were hurting our performance.

This process included discretizing certain features, such as length of film, instead of treating each possible value as a separate feature, adding details about previous award nominations and wins, as well as pruning features with significantly skewed weights, which did not actually provide predictive power.

As a more specific example of the latter, it turned out that `genre|Drama` was very negatively weighted after learning our weights from the training set, since the majority of nominees fell into this genre and most nominees are not the winner of a category. During error analysis, however, we identified this problem and noted that `genre|Drama` did not actually provide significant predictability about whether or not a given film would win. As a result, we removed it from our feature set.

Our error analysis steps were extremely necessary and beneficial in expanding and redefining a predictive set of features.

7.2 Temporal Data

Another challenge arose from a phenomenon specific to our particular task and our data set: the possible time variation in predictive features (mentioned in Pardoe [2]). For example, the number of previous nominations for a given actor has become more predictive for a film's success than in the past. We began with an available data set containing Academy Award nominees and recipients from 1927 through 2014. We attempted to address some issues of this temporal-data problem by pruning our data set to a set of years which were most influential in predicting results in our test set (2010-2014). We ultimately found that data prior to 1980 did not provide significant predictive ability for our test set and that these years actually negatively impacted our performance.

For future work, it would certainly be interesting to perform feature selection on data sets from different time periods (possibly partitioning our data by decades). We might then be able to gain some interesting insights about the types of features which were the most predictive in different time periods and whether or not the set of most predictive features has changed over time.

We additionally had to deal with challenges related to the possible importance of the prior year's winners in determining the important features which would be predictive for the following year's winners. This was a phenomenon that we wanted to capture in our model. To elucidate, it has been suggested by some that the most different film from the previous winner is most likely to win, reflecting the idea that the Academy desires novelty. This is a particular challenge we were not able to address during the course of our project; however, it serves as a very interesting topic to address in future research.

8 Results

The results of our models after each pass of feature set development is listed above in "Feature Extraction & Error Analysis"; for our supervised learning experiments, a succinct summarization of our results is provided below in Figure 2.

For K-Means, we adjusted our set of features by removing the indicator features regarding the nominees' genres, actors, producers, directors, and writers; this is because K-Means is sensitive to feature sparsity. In this reduced feature space, the only features which remain pertain to the awards and nominations that the

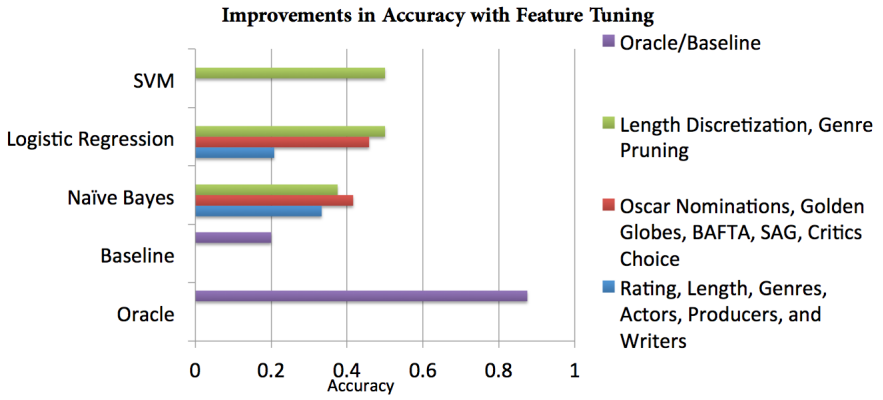

Figure 2: The improvement we saw by tuning our futures was substantial, as both Logistic Regression and Naive Bayes improved to perform much better than our baseline. The test set for these experiments was restricted to the nominations in 2014 for a more direct comparison with our oracle.

film received in previous film award ceremonies (e.g. BAFTA, SAG, Critics' Choice). (The only exception to this is the indicator feature related to the length of the film, which is also included.) Our results for K-Means are shown below in Figure 3.

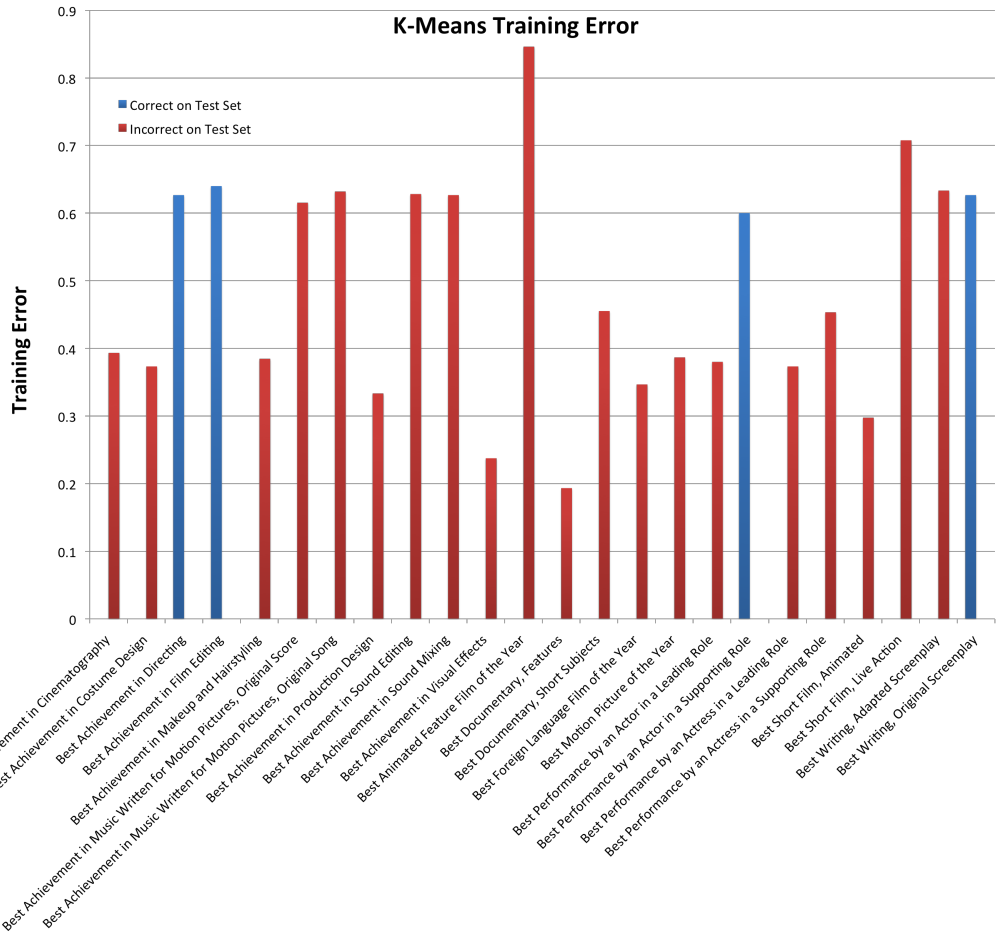

Figure 3: Results for our K-Means algorithm for each of the 24 categories. Similar to the results in Figure 2, the test set for these experiments was restricted to the nominations in 2014. Only four categories in the test were correctly predicted.

As we can see, despite our feature reduction strategy, K-Means performed quite poorly, with an accuracy of 16.67% on the 2014 test set. Increasing the test set to its original range did not yield any improvement; on the contrary, the accuracy was only 2.5% for the the full range of nominations between 2010 and 2014. Moreover, we can see that, in the few instances where K-Means did correctly predict the result, this was not based on accurate modeling of the hidden structure in our data set, as the training error for all of the categories is quite high, especially for the four categories that were correctly predicted.

9 Conclusion

As we demonstrated in our results, we were able to develop feature sets and build models which performed significantly better than our baseline. We saw more than a twofold improvement in performance between our baseline and our Third Pass feature set. There is still work to be done to achieve accuracies close to our oracle; however, our performance still indicates a notable degree of predictability in determining the winners of Academy Awards.

We saw that, with more advanced and intelligently designed features, our Logistic Regression and SVM algorithms outperformed Naïve Bayes, while our K-Means algorithm did not perform well at all, even with the reduced feature set. This intuitively makes sense if we consider the sparsity of our data set; since we are only training on 20 or so years of data, we are certainly susceptible to outliers, which can vary from algorithm to algorithm. For SVM, Logistic Regression, and Naïve Bayes, we can combat this by using regularization (in the case of SVM and Logistic Regression) or smoothing (in the case of Naïve Bayes). However, K-Means is particularly vulnerable to outliers in the data set.

Fundamentally, because of this sparsity, the task of predicting Academy Award winners is certainly a difficult machine learning problem. For future work, we'd like to explore various techniques for addressing outliers for the K-Means algorithm [5], which could potentially help its performance. Furthermore, we can also try to take advantage of the large corpus of critic's reviews and articles to improve our classifier. Specifically, we can investigate the use of text mining, sentiment analysis, and other various NLP techniques to determine the "positive" sentiment for each nominee, which, hypothetically, could signal an increased likelihood of winning a particular nomination. This is certainly an interesting question that we could further explore in a subsequent research project.

References

[1] Adams, T. 2014 Oscar Predictions: Our Picks in Every Category (2014.) https://movies.yahoo.com/blogs/movie-talk/2014-oscar-predictions-183733830.html

[2] Pardoe, I., & Simonton, D. K. (2008). Applying discrete choice models to predict Academy Award winners. Journal of the Royal Statistical Society: Series A (Statistics in Society), 171, 375–394. http://iainpardoe.com/research/08jrssa/oscars.pdf

[3] McGovern J. and Jensen D. (2003.) Identifying Predictive Structures in Relational Data Using Multiple Instance Learning. Knowledge Discovery Laboratory, Univ. of Massachusetts Amherst https://kdl.cs.umass.edu/papers/mcgovern-jensen-icml2003.pdf

[4] Arthur, D. and Vassilvitskii, S. (2007). "Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms". Society for Industrial and Applied Mathematics Philadelphia, PA, USA. pp. 1027–1035.

[5] Hautamaki V, Cherednichenko S, Karkkainen I, Kinnunen T, Franti P (2005) Improving K-means by outlier removal. In: Kalviainen H, Parkkinen J, Kaarna A (eds) Image analysis, lecture notes in computer science, vol 3540. Springer, Berlin/Heidelberg, pp 978–987