Final Report

Sophie Lancaster
Notre Dame, United States
slancas1@nd.edu

Alex Ayala
Notre Dame, United States
aayala4@nd.edu

Aidan Lewis
Notre Dame, United States
alewis9@nd.edu

I. Abstract

For this project we aimed to see if we could predict the nominees and winner of the Best Picture category of the Oscars. We did this by collecting data for multiple movies from a variety of sources for two different years, 2016 and 2017. We used the 2016 data to train our model and create a list of optimized weights for each of the different types of data. We then used these optimized weights in conjunction with the 2017 data to predict that 2017 nominees and winner. With the genetic algorithm method we were able to predict up to 7 out of the 9 2017 nominees correctly but were not able to predict the winner. We then used a random forest classifier to benchmark our results and were able to predict up to 8 out of the 9 2017 nominees. We were pleased with these results but also identified future areas for growth. For example by training our model on more years of previous data and by including more data sources we could hopefully improve the accuracy of our predictions.

II. Introduction

The main goal of the project is to utilize popular data collected from Twitter alongside other data sources to help predict the nominees and winner of the Best Picture category of the Oscars. With this project we will collect and analyze Twitter data for different movies and try to get a sense for not only public opinion but also the momentum that a movie may or may not be building throughout the movie awards' season. One of the main objectives of this project is to see if popular buzz about a movie (measured through Twitter data) contributes to this momentum. Furthermore, other contributing factors such as critics' reviews from Metacritic, user reviews, release date, previous trends in nominees and winners for both the Oscars and other award shows, lifetime gross, and the number of theaters a movie showed in will be considered when evaluating the momentum of the film. Overall, with this project we hope to gain a better understanding of some of the potential contributing factors for movie award show nominees and winners.

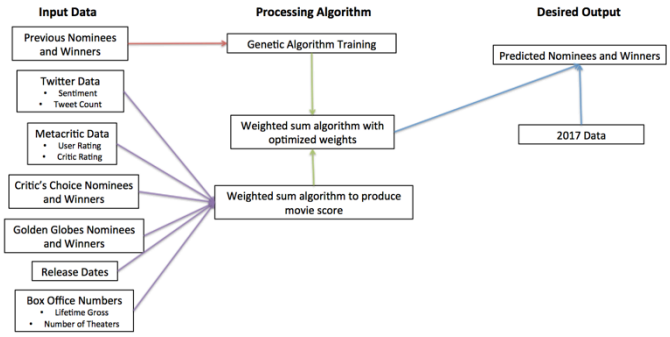

Figure 1: Project Overview Diagram

III. Related Work

The study of machine learning is becoming more prevalent in a world that is overwhelmed with the amount of data that we have access to. Today we are able to obtain data from virtually every platform and every person. The main idea behind machine learning prediction models is to gather the appropriate data, transform this data, choose a machine learning algorithm, and finally train, test and evaluate the model. With this project we hope to follow a very similar process, which can be seen more clearly in the diagram above. Many prediction models exist across a number of fields that make use of the machine learning process and algorithms. This includes things such as frameworks for sport prediction[1], customer behavior prediction[2], and even applications for predicting relapses in cancer[3]. When looking into what prediction models existed similar to the one we plan to make we came across an application that predicted the 2017 Oscar Winners using

[1] https://www.sciencedirect.com/science/article/pii/S2210832717301485
[2] https://www.altocloud.com/blog/how-machine-learning-can-be-used-to-predict-customer-behaviour
[3] https://www.nature.com/articles/s41598-017-07408-0

BigML[4], which is a powerful machine learning interface. The group that created this application went through a very similar process as the one explained above with data collection and transformation followed by the modeling and prediction. This model was able to predict three out of eight categories correctly with a 67% confidence in the best director category. With our project we hope to be able to achieve results similar to, if not better than, this group was able to achieve with their model.

IV. Data Collection

In order to create an initial list of movies to collect data for we used the feature of Metacritic that allowed us to specify the year and a threshold score. By doing this search we got a list of 2016 and 2017 movies that had a Metacritic score of 65 or higher. We were able to get these movies by scraping different HTML pages on Metacritic's website. This initial filtering process yielded a list of 336 movies from 2016 and a list of 383 movies from 2017. The remaining data was collected using these movie lists. An explanation of how each type of data was collected can be seen below.

TWITTER DATA

To collect the necessary data, the GetOldTweets-Python library was utilized. Originally, the intention was to utilize Internet Archive's collection of tweets. This collection featured several million tweets per day from Twitter. Utilizing this data source, however, resulted in multiple challenges. First, the data from each day of collection required between one and two gigabytes of storage. Considering that we hoped to gather this Twitter data from every day in 2016 (our training set) and 2017 (our test set), the resulting memory requirement would likely be over a terabyte. Despite gaining access to extra storage through Professor Wang's SS Lab machine, this amount of memory was extreme. In addition, very few of these tweets actually concerned the movies we were intending to gather data on, meaning much of this space would be wasted. Further, while the data contained several million tweets per day, it was not comprised of every tweet from that day. In fact, this dataset contained only a fraction of the hundred of millions of tweets a day, meaning we would miss most of the tweets we were searching for. During midterm presentations, another group mentioned use of the GetOldTweets-Python library, which we explored, and decided would be a better fit for our task. This library allowed us to search for tweets containing a specific search term between any two dates. A shortcoming of Twitter's API is that the Standard Tier includes access to only tweets from the most recent 7 days. Since we needed data from 2016 and 2017, this restrictiveness made the API unusable. In contrast, GetOldTweets-Python allowed access to much further back dates, ensuring that we could gather data on movies released in recent years.

Another challenge faced in data collection was in determining the search term for each movie. While simply looking for a movie's title would work for most movies, movies with more common names posed problems. For example, the title of the movie *Get Out*, a best picture nominee in 2017, is simply a common phrase. Searching for the term "get out" results in some tweets discussing the movie, but the vast majority are using it in an unrelated fashion. Consequently, searching for this term would lead to a misleadingly high number of tweets associated with *Get Out*. To solve this problem, we adopted an idea from another group in the midterm presentations; this group decided to search for a movie's official hashtag to better filter out unrelated tweets.

To collect the training data set, any tweet from March 1, 2016 to March 1, 2017 containing a hashtag associated with one of the 336 2016 movies was collected. An initial decision faced was whether to collect Twitter data starting from each movie's release date, or from immediately after the previous Oscars for every movie. It was decided that we would collect data starting from the previous years Oscars, in case we needed to use Twitter information to determine a movie's pre-release popularity. In the end, however, this pre-release information was not factored into any of our models. To gain the tweet count data, the total number of tweets for each movie was calculated for each day from this data. This tweet count data was stored such that each movie had a set of 365 tweet count values associated with it, one for every day of data collection. To obtain tweet sentiment data, the Python module TextBlob was utilized. Given a text input, TextBlob returns a score between -1 (bad/sad) and 1 (good/happy) representing the detected sentiment of the tweet. The sentiment of each tweet was analyzed using this method, and a per day average sentiment was calculated for each movie. A challenge faced was performing this analysis in a reasonable amount of time. Initially, the intention was to perform the sentiment analysis on each tweet for a movie, and then take the average sentiment per day and write it to one file with every movie and its respective tweet sentiments. However, performing this

[4] https://dzone.com/articles/predicting-the-2017-oscar-winners

analysis for one movie at a time took around 20-30 minutes per movie, meaning that for the 336 2016 movies and 383 2017 movies, it would take over two weeks to complete. Taking this approach would have significantly slowed our group's progress. As a solution, we took a MapReduce-inspired approach. Instead of performing the analysis sequentially and writing to the same, one file, we simultaneously ran the analysis function on every movie and wrote every movie's results to its own file. After the analysis had completed, another function wrote the contents from all these different, movie-specific files to one master file that now contained every movie's average per day tweet sentiment. Now instead of taking two weeks, this process took about 30 minutes, so essentially the time it took to run the analysis for one movie.

On their own, this set of tweet counts and average tweet sentiments for every day for each movie was not necessarily useful. All tweet count would indicate was which movies were most popular, but these are not necessarily the movies that the Academy will like best. To extract a meaningful metric from this data, a ratio was calculated to determine a movie's "buzz" immediately before the Oscar nominations were announced. Since the Oscars are the last major movie award show of the season, many award shows precede it. Conveniently, the announcement of Oscar nominations generally come during a very active time during movie awards show season, where there is much discussion on the possible nominees and winners for different award shows. The intention of this ratio was to determine how much a movie was being discussed during this time, with the hope that this would indicate the likelihood of a movie being an Oscar best picture candidate. For tweet count, the ratio was calculated by dividing the average number of tweets per day about a movie in the week before the announcement of Oscar nominations by the average number of tweets per day about the same movie during its week of release. By dividing by the average number of tweets during the movie's week of release, the data was essentially normalized. We did not care if a movie had a lot of tweets about it at all times; all this indicated was popularity, which is not necessarily related to award season success. Instead, we wanted to identify movies that were being talked about more during the week before the Oscar nominations announcement than during their initial week of release. It would be expected that a movie would be most discussed and relevant during its week of release; for it to be more discussed in the lead up to the Oscar nominations announcement could mean that a movie was in fact a likely candidate for the Oscars. The same ratio was calculated for tweet sentiment in order to determine if popular sentiment about a movie was higher during the awards season, possibly indicating that a movie was having a successful awards season, and could be an Oscar contender.

BOX OFFICE NUMBERS

A website called Box Office Mojo[5] was used to collect the lifetime gross of the movie and the number of theaters the movie showed in. There was no good way to scrape this website for the desired information so the numbers for each movie were entered into a box office number text file manually. If no data was available for a given movie on Box Office Mojo then the value NULL was used for both data entries. These NULL values were dealt with once the data was fed into the different processing algorithms and is discussed later in the paper. For lifetime gross, the number was very large compared to others, so to normalize this value we divided by one billion for the genetic algorithm and one hundred million for the random forest classifier. This is necessary especially for the genetic algorithm as the mutations and initial random selection of constants have the same random selection range for every factor. If we did not do this normalization, Disney and Marvel movies, which tend to dominate the box office would have scores that are almost entirely based on box office numbers and nothing else. This could either give them extremely high scores or extremely low scores based on our movie scoring algorithm.

METACRITIC SCORES

We used a user built Metacritic API by maracalencc[6] in combination with a Python script to gather both average user review scores and average critic review scores. In the event that the API could not find a user/critic review score, a value of NULL was used for the two entries.

RELEASE DATES

We used the same Metacritic API and Python script mentioned above to gather release dates for each movie. In the genetic algorithm, this date is converted to a numeric value which is computed as the number of days since January 1 of the Oscar year divided by 365 to normalize. For the random forest classifier, the month number was used instead (i.e. 12 for December, 11 for November, etc.) because the classifier does not work as well on continuous data.

---

[5] http://www.boxofficemojo.com/

[6] https://market.mashape.com/marcalencc/metacritic

PREVIOUS WINNERS AND NOMINEES

After collecting all of the data from the sources mentioned above the data was aggregated into a CSV file containing the final formatted data from all sources. In order to account for the data from the other award shows two more entries were added to the CSV and were used to indicate which of the movies had been nominated and won the Critics' Choice Movie Awards and the Golden Globes. If the movie had been nominated for one of these awards then a one was put in that spot for that movie otherwise a zero was added.

V. Movie Scoring Algorithm

In order to effectively "rank" the movies, we assign a score to the movies based on the following equation:

$$\begin{aligned} MovieScore = {} & (a * UserReview) + (b * CriticRating) \\ & + (c * SentimentScore) \\ & + (d * DomesticGross) \\ & + (e * TheatersReleased) \\ & + (f * ReleaseDate) \\ & + (g * GoldenGlobe) \\ & + (h * CriticsChoice) \end{aligned}$$

Eq. 1

Effectively, each component used in assigning a score to the movie is given a weight which gets optimized via the genetic algorithm. The predicted winner and nominees as such would have the highest assigned score.

VI. Genetic Algorithm

The training algorithm we used is a genetic algorithm[7] which trains on 2016 Oscars data, and consists of the following steps:

1. **Initial Population:** An initial set of 1000 parameter sets for the fitness function will be randomly generated to make the "population" of parameters.
2. **Fitness Function:** After assigning a score based on the algorithm above to each movie, the algorithm will sort by that movie score. The fitness function in this case gives the parameter set a higher fitness the closer the actual winner and nominees are to the front of the list with a slightly higher weight on the winner. The fitness also gets lowered if the predicted winner and/or nominees are not the true winner/nominees.
3. **Parent Selection:** Select two parents from the population using a weighted random selection based on the fitness score to generate children parameter sets.
4. **Child Generation:** Each pair of parent parameter sets generates two children. We perform a uniform crossover with the parameters of the parents to generate the two children parameter sets.
5. **Mutation:** At each generation of a new child, the child has a 20% chance to introduce a mutation which offsets the value of randomly chosen parameters by a random number between -3 and 3.
6. **Replacement:** Repeat steps 3-5 until the number of children generated is 30% of the original population. Replace the 30% of parameter sets in the population with the lowest fitness scores with the children.
7. **Optimization:** Repeat steps 2-6 until the population is fit enough (i.e. a convergence). We had the algorithm run for 700 generations to ensure convergence.

This algorithm allows us to optimize the weights we use for our predictor since a parameter set is "rewarded" for better matching the true results of the past Oscar award show.

VII. Random Forest Classifier

Initially, the plan was to utilize only the genetic algorithm for predicting the Oscar nominees and winners. However, it became clear later that this would provide no point of comparison for the genetic algorithm, which could be very problematic if the genetic algorithm underperformed. Using Sckit-Learn, three different machine learning classifiers were tested as possibilities: multilayer perceptron (MLP) classifier, decision tree classifier, and random forest classifier. The MLP classifier is a neural network model that learns a function by training on a dataset (the 2016 movie information, in our case), and then utilizes this function to predict output on new data. A decision tree classifier is a supervised learning method that repeatedly divides the training data into smaller subsets based off of splits in the data for various features. When a subset of the data eventually is all of the same output class, the splitting ends and a leaf node of this specific class is created. This process results in the construction of a tree, and the testing data is run through this same tree. When an instance of testing data reaches a leaf node, it is given the class of that node. The random forest classifier

[7] https://towardsdatascience.com/introduction-to-genetic-algorithms-including-example-code-e396e98d8bf3

creates a specified number of decision trees (1000, in our case), and trains these on different sub-samples of the training data. When testing data is run on a random forest classifier, the resulting classifications from these decision trees is averaged for the final prediction. Due to its large sample of trees, random forest classifiers can have superior accuracy over a normal decision tree, and its property of training on sub-samples of the data can help control for over-fitting.

To test the predictive capabilities of the Twitter data, the first trials were conducted with only the tweet count ratio and box office revenue. Box office revenue was included to control for movies whose tweet count ratios were unreliable since they were calculated based off a very low number of tweets. For example, if a movie was tweeted about once during it's week of release, and two times during the week before the Oscar nominations announcement, the calculated tweet count ratio of 2 would be very high, even though this difference was likely due to chance. The inclusion of box office data ensured that movies that barely anyone had seen would be disregarded, even if they had a high tweet count ratio. To ensure accuracy, 100 iterations of fitting and training were performed for each model. For a later trial, the process remained same, except release date was added as a third feature. To develop a more successful predictive model, a final trial was performed where more features were now added to the model, including Metacritic critics and user score, whether a movie was a Golden Globe best picture nominee, whether a movie was a Critics' Choice best picture nominee, the number of theaters a movie was played in, and the tweet sentiment ratio. In each of these trials, all three classifiers were run on this data and compared in terms of accuracy.

VIII. Results and Evaluation

PRELIMINARY RESULTS

In order to test our approach for measuring the sentiment and popularity surrounding a given movie during awards season, we collected Twitter data for the week leading up to the 2018 Oscars. Using the Tweepy Search API, we gathered tweets that contained the title of some of the strongest contenders for Best Picture (*Three Billboards Outside Ebbing, Missouri*, *Dunkirk*, *Lady Bird*, and *Call Me By Your Name*) as well as for the Best Picture winner, *The Shape of Water*. These tweets were time stamped from a week before the event, up to several minutes before the Best Picture announcement. We grouped the tweets by hour, and for each hour recorded the number of tweets mentioning each movie. Further, we used the TextBlob Python library to run a sentiment analysis on each tweet, and recorded the average tweet sentiment for each movie for each hour in the week leading up to the Oscars. By collecting data in this way, we were able to track over time the buzz and sentiment surrounding a movie, and see if this seemed to be related to the outcomes on Oscars' night.

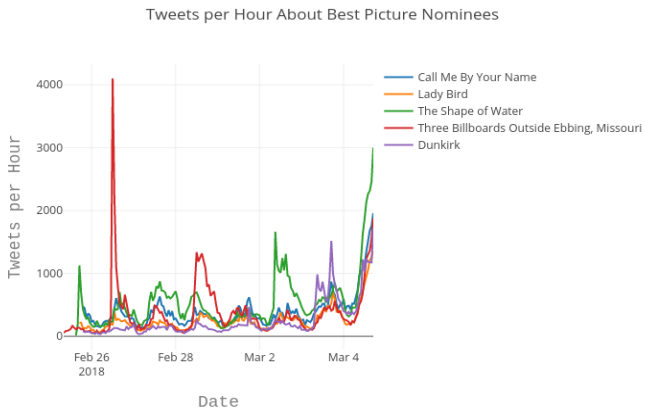

Figure 2: The number of tweets per hour mentioning five different Best Picture nominees

In Figure 2, the tweets per hour about each Best Picture contender is graphed over the week prior to the Oscars. Many movies had spikes in Twitter mentions throughout the week, including *Three Billboards Outside Ebbing, Missouri* on February 26th, *The Shape of Water* on March 2nd, and *Dunkirk* on March 3rd, but in general, *The Shape of Water* was the most discussed film in the week leading up to the Oscars. *The Shape of Water* had several spikes in mentions throughout the week, and was especially talked up about on the day of the Oscars and during the ceremony (before the award was announced). This would seem to be an indication that people were expecting *The Shape of Water* to be the strongest contender for Best Picture.

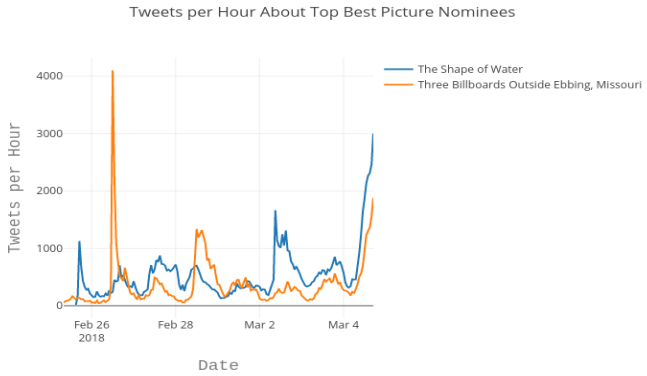

Figure 3: The number of tweets per hour mentioning the two leading contenders for Best Picture

In Figure 3, again tweets per hour is plotted against date during the week prior to the Oscars, but this time the only two movies graphed are *The Shape of Water* (the eventual Best Picture winner) and *Three Billboards Outside Ebbing, Missouri*, which was considered its toughest competition in the Best Picture category, since *Three Billboards Outside Ebbing, Missouri* won Best Drama Motion Picture at the 2018 Golden Globes, over *The Shape of Water*. However, it is clear from this graph that leading up to the Oscars, it was *The Shape of Water* that was receiving more buzz. The one gigantic spike in the popularity of *Three Billboards Outside Ebbing, Missouri* on February 26th was actually unrelated to Best Picture discussions: a company in Philadelphia, inspired by the movie, put up three billboards asking Lebron James to come play for the Philadelphia 76ers. Ignoring this outlier, *The Shape of Water* was more popular in every hour of the last three days leading up to the awards ceremony, so it had more "buzz" than its closest rival prior to the Oscars.

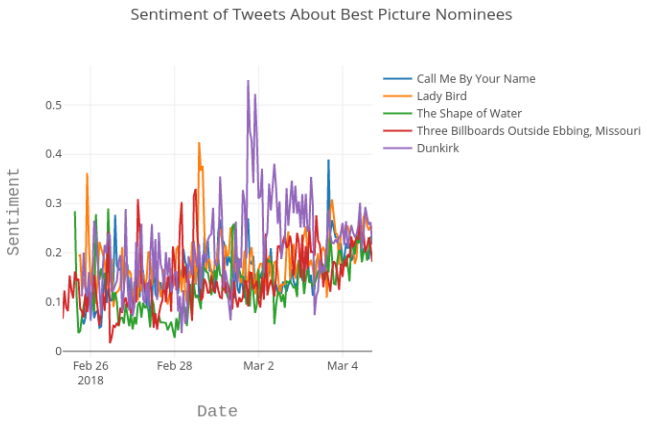

Figure 4: The average sentiment of tweets mentioning five different Best Picture nominees

In Figure 4, the average sentiment per hour of tweets is graphed over the course of the week. *The Shape of Water*, while very talked about (as shown in Figures 2 and 3), was not generally talked about as positively as many of the other movies. What this data suggests is that the movie that is predicted to win Best Picture is not necessarily a movie people saw or particularly liked. *The Shape of Water* had a pretty strange concept (a woman falls in love with a fish-man), so naturally it would be a little more divisive in terms of popular opinion. However, it is not popular opinion that wins a movie an Oscar for Best Picture, as the people who vote on this award are accomplished people in the world of cinema, not your average moviegoer. From these initial results, it was evident that tweet count could serve to be an important feature, while the usefulness of tweet sentiment was much more questionable.

GENETIC ALGORITHM

After running the genetic algorithm a number of times on the 2016 data, we came up with the following constants which had the best results for predicting the 2016 data:

User Review: -2.04529602987
Critic Review: 3.58677554026
Sentiment: 3.60802131784
Domestic Gross: -6.3138135618
Number of Theaters: 0.0148526973376
Tweet Count: -1.31409237727
Release Date: 9.47295805662
Golden Globes: 8.21434314202
Critic's Choice: 7.09437124458

This set of constants predicted the correct winner for 2016, *Moonlight,* and predicted 7 of the 8 other nominees *(La La Land, Manchester by the Sea, Hell or High Water, Hacksaw Ridge, Arrival, Lion,* and *Fences)* with the only miss being *Hidden Figures.* When running the movie scoring algorithm on these constants on the 2017 data, the predicted winner was *Dunkirk* (which was not the winner but a nominee) and the 5 of the other 8 nominees were also predicted (*Lady Bird, The Shape of Water, Call Me By Your Name, Get Out,* and *Three Billboards Outside Ebbing, Missouri).* The only missed movie predictions were *The Post, Phantom Thread,* and *Darkest Hour* (with *The Post* and *Phantom Thread* being in the top 20).

For comparison's sake, we also came up with the following constants which did not predict the Best Picture winner but still predicted most of the nominees:

User Review: -8.68062878462
Critic Review: 0.655583750887
Sentiment: 0.272055593674
Domestic Gross: -9.55888213033
Number of Theaters: 0.0211656468326
Tweet Count: 3.46734740166
Release Date: 8.09562388019
Golden Globes: 7.84450702783
Critic's Choice: 9.93304840944

This set of constants predicted *La La Land* as the 2016 winner, but also predicted 7 of the 8 other nominees (*Moonlight, Hacksaw Ridge, Manchester by the Sea, Lion, Hell or High Water, Arrival, and Fences*) and had the missed nominee, *Hidden Figures*, as the eleventh ranked movie. One of the reasons why *Hidden*

Figures and *La La Land* were ranked higher compared to the other set is the importance of tweet count. *La La Land* had won one of the Best Picture awards at the Golden Globes and was nominated for a total of 13 Oscars compared to *Moonlight's* 8. Furthermore, *Hidden Figures* was very popular due to its theme of African American women working in the sciences around the time of desegregation.

When running the constants on the 2017 data, *Dunkirk* was incorrectly predicted as the winner (even though it is a nominee), and 6 of the other 8 nominees were correctly predicted (*Three Billboards Outside Ebbing, Missouri, The Shape of Water, Get Out, Lady Bird, Call Me By Your Name,* and *Darkest Hour*). Additionally, one of the two missing nominees, *The Post* was also ranked in the top 20 with these constants.

RANDOM FOREST CLASSIFIER

Training on the tweet count ratios and box office revenues of 2016 movies, and testing on the tweet count ratios and box office revenues of 2017 movies, the random forest classifier and decision tree performed best. Each predicted five of the nine best picture nominees (*The Shape of Water*, *Three Billboards Outside Ebbing, Missouri*, *Darkest Hour*, *The Post*, *Lady Bird*) on all 100 iterations. However, both models predicted four other best picture nominees that were incorrect (*Molly's Game*, *Wonder*, *Coco*, *The Lego Batman Movie*) on all 100 iterations, and another incorrect nominee (*Wind River*) on 95 iterations. The MLP classifier performed terribly, as it predicted 333 of the 383 total 2017 movies to be a best picture nominee during at least one iteration. To its credit, the two movies it picked most during these iterations as a best picture nominee were *The Shape of Water* and *Lady Bird*, which were in fact nominees. The model's lack of consistency here though made it easily the worst of the three models.

To improve this model, the release date feature was added. Since most Oscar nominees are released late in the year, it was hoped that adding this feature would remove some of the false positives. Most notably, adding this feature drastically improved the MLP classifier, which predicted five of the best picture nominees on the majority of iterations (*The Shape of Water* (85 iterations), *Three Billboards Outside Ebbing, Missouri* (83 iterations), *Darkest Hour* (85 iterations), *The Post* (84 iterations), *Lady Bird* (85 iterations)), and had only one false negative (*Wind River* (81 iterations)). While the decision tree classifier performed similarly to how it performed without release date, the random forest classifier predicted the same five correct best picture nominees on all 100 iterations, but only predicted two false positives (*Wonder* on all 100 iterations and *Coco* on 99 iterations).

While just including tweet count ratio, box office revenue, and release date as features does not yield the most accurate model, it is impressive that by utilizing only these features, a model can be developed that predicts over 50% of the correct nominees and only suffers from one or two false positives out of 383 movies. This indicates the usefulness of the tweet count ratio in a predictive model. As for the missed nominees, there are several likely reasons. To start, two of the missed nominees, *Call Me By Your Name* and *Phantom Thread*, both had very high tweet count ratios, but had the two lowest box office revenues of the nominees. Although both movies still made around $20 million in box office revenue, which is not an insignificant number, they still made less than any of the best picture nominees in the training data. This shortcoming could likely be fixed with more training data. For the other two missed nominees, *Get Out* and *Dunkirk*, there was essentially the opposite problem. These two movies were far and away the most successful movies at the box office of the nominees, and unlike the other nominees, were initially very popular. Because of this, their tweet count ratios were very low, since they were very talked about during their week of release. This low tweet count ratio likely hurt their performance in this model, where tweet count ratio was heavily weighted.

For the final trial, all features were added in, including awards show data from the Golden Globes and Critics' Choice Awards. Once again, all three methods of predictive modeling were tried on the data. The MLP classifier performed very poorly, predicting 94 different movies to be a best picture nominee across the 100 iterations. The decision tree classifier performed much better, predicting 6 of 9 best picture nominees on all 100 iterations (*The Shape of Water*, *Three Billboards Outside Ebbing, Missouri*, *Darkest Hour*, *The Post*, *Lady Bird*, *Call Me By Your Name*) , and another best picture nominee (*Get Out*) on 32 iterations. However, it still had several false positives, including *The Big Sick* on all 100 iterations, *Shadowman* on 28 iterations, and *I am Jane Doe* on 19 iterations. The random forest classifier performed best, predicting 8 of the 9 nominees (all except *Phantom Thread*) on all 100 iterations, with only one false positive (*The Big Sick* on 82 iterations). This random forest classifier model was the most successful of any model tried. As for why *Phantom Thread* was not selected, it is likely because it was the only Oscar best picture nominee that was not nominated for best picture at either the Golden Globes or Critics' Choice Awards.

IX. Discussion and Limitations

One of the limitations of our model was the tweet sentiment ratio feature. Initially, we believed that Oscar nominated movies might be discussed with greater reverence on Twitter. However, tweet sentiment ratio was a feature that failed to improve any of our models. This is likely because popular sentiment about a movie and success at the Oscars are not necessarily related. The Oscars have a reputation for not always choosing crowd pleasing movies; instead, they often choose movies that are much more adored by critics than fans. Consequently, trying to gauge how the public felt about a movie through tweet sentiment was not very indicative of how well a movie would do during awards season.

Another limitation of our data is that only one year's worth of training data was utilized. Because of this small sample size for training, our models predicted based on patterns in the 2016 data. However, the patterns identified for 2016 don't necessarily represent the typical Oscar best picture nominee patterns. This is especially true if there are any outliers in the 2016 data, as without more data, these outliers would be considered the norm. Expanding the training data to include several other years would undoubtedly create a more balanced, generalizable model.

Based on the results, it became clear that certain features were of greater importance than others. Two of the most important features were whether a movie was nominated for best picture at the Golden Globes or Critics' Choice Awards. In the genetic algorithm, a high weight was placed on this factor, and the accuracy of the two tree-based classifiers improved markedly upon the addition of this feature. The success of these features is understandable, since there is often significant overlap between the movies nominated for best picture at the Golden Globes and Critics' Choice Awards, and the movies nominated for best picture at the Oscars. Another important feature was box office revenue. While the movies nominated for best picture at the Oscars are not always box office smashes, they all still gross in the tens of millions of dollars at least. Many of the movies in the dataset were very small releases, and brought in less than a million dollars. These movies were essentially too small for Oscar consideration, and the box office revenue feature made sure to eliminate these from contention in our model. Release date was another critical feature; most Oscar nominated movies are released late in the year, since they are released with the intent of competing during award season. This feature was weighted highly in the genetic algorithm, and significantly improved the machine learning classifier models. A final influential feature was the tweet count ratio, which as discussed earlier, allowed us to gauge a movie's "buzz" and momentum during award season.

Apart from the factors that ended up being important for predicting the success of the movies there were a few features that ended up being less important. The first factor that had a low optimized weight constant was the number of theaters. We think that this factor was less significant because it was a redundant piece of information. The number of theaters that the movies showed in followed a very similar pattern to lifetime gross and because of this the genetic algorithm assigned a low weight to the number of theaters since it already had similar information from the domestic gross. Another factor that was less significant as determined by the genetic algorithm were the user reviews. This is a pretty understandable result because the reviews that user's leave can be biased and thus lead to random patterns. The final factor that ended up being less significant is the Tweet sentiment that was determined for each movie. This was an interesting result because when we initially started the project we thought that this was going to be our main piece of data. The explanation as to why this is the case has already been discussed.

With regards to the genetic algorithm, there were certain challenges that we encountered while training our model. We noticed while training that our model converged relatively early (at around 50 generations), but the numbers that were converged to did not always yield very good results. We helped to prevent this early convergence by increasing our mutation frequency from 5% to 20%, expanding the mutation factor range from [-0.2, 0.2] to [-3,3], utilizing a weighted random selection on the whole population rather than a subpopulation, and using a uniform crossover for child generation instead of the averages of the parents. All in all, this introduces more randomness into the algorithm to help slow down convergence to around 100-200 generations.

Another problem we encountered with the genetic algorithm was deciding on a fitness factor. While the fitness factor we had in place did a relatively good job with optimizing our constants, we were still never able to find a fitness factor that was able to give constants that guessed each nominee and winner correctly. Specifically, it was hard to decide how much to prioritize predicting the correct winner versus predicting correct nominees. For the future, we could probably do some more elaborate testing with the fitness function to help us decide how important guessing the winner versus the nominees is. An additional factor that could contribute to never guessing all of the nominees and

winner correctly is a need for more data sources. It may be that given our current data, it is actually impossible or extremely unlikely to find constants that would perfectly guess the winner and nominees on the 2017 data. Perhaps leveraging other data sources would fix this issue.

X. Future Work

While we were pleasantly surprised about how accurately our model was able to predict the nominees and winners, there are a couple future steps that could be taken to improve the accuracy. For example, we only chose to train the genetic algorithm and decision trees on one set of previous data, the 2016 movie data. In the future, it would be worthwhile to carry out the data collection process for movies from other previous years to see if this could help improve the accuracy of our model's prediction.

Another step we could take in the future would be to include additional data sources. One additional data source we could have included would be critics' year end best movie lists. These lists generally summarize the year in movies, and aggregating these lists from prominent movie critics would likely produce a list with many movies that were being considered for best picture at the Oscars. The advantage of using this over using an aggregation of critics movie scores is that it would also us to see how critics compare certain movies head-to-head. Further, adding data from more movie awards shows could also better our model. For instance, the random forest classifier missed one 2017 Oscar best picture nominee, *Phantom Thread*, likely because it was not nominated for best picture at either of the awards shows included in our data. However, *Phantom Thread* was nominated for best film at several smaller award shows, so including these award shows may have assisted our model. Finally, incorporating Google Trends data could also have improved our model. Google Trends would allow us to see how popular a movie was at any given point in time. This data could have been used in the same way Twitter data was used in order to determine which movies were most talked about during award season.