Individual Project Final Paper

Matthew J. Bierman

Arizona State University

CIS 450, Dr. Tamuchin McCreless

Introduction

Last year, in the United States and Canada, the movie industry realized over $4.5 billion in box office sales (The Numbers, 2021). The movie industry spends vast amounts of money producing, promoting, and releasing these movies; yet most of the movies released fail to generate significant revenues at the box office. Like with many things, The 80/20 Rule (or Pareto Principle) applies. This rule states that in general, 80% of the consequences come from 20% of the causes (Laoyan, 2021). In this context, this means that 20% of the movies released should account for 80% of the total box office revenues (about $3.6 billion in 2021). In fact, only the top 8% (31) of all movies released in 2021 accounted for 80% of the total revenue. This means that 92% (376) of the movies released had to fight over the remaining 20%. While some of these remaining films will generate positive cash flows for their investors, there are many that will not. Given this, studios will want to know in advance how their movies will perform at the box office. Early detection of performance will allow them to make faster decisions about adjustments to their release strategy, which may help them increase their overall financial performance. The question then becomes: How?

Film studios and distributors use financial analysis and forecasting to determine the viability of a project prior to production. Using all types of subjective metrics, they can estimate a film's performance fairly well and have a good track record of producing profitable movies; that is why they are still in business. However, there is always a wildly uncontrollable variable, human behavior. Just because a studio has put the brightest stars and a lot of marketing dollars into a movie, it does not always directly translate to return on investment (ROI). By nature, human beings are irrational and do not always make the most logical decisions. Despite the best efforts of a studio, a movie may bomb for any number of reasons. Maybe people did not like the trailer. Maybe the film's star committed some unlikable act right before the movie opened. Whatever the reason, it would be helpful if the studios had some insight into this irrational behavior. An indicator that could tell them whether or not their film would be one of the top 80% of revenue generators or the bottom 20%.

Preliminary Analysis

By analyzing social media data from Twitter, the goal is to build a prediction engine that hopefully can predict whether a movie will be one of the top 80% of revenue generators or the bottom 20%. Movie studios have a good idea which movies will be at the very top of the box office revenues list. They also have a good idea which movies will end up at the bottom. But there is a vast amount of middle ground between these points. I am hoping to provide some clarification in this area. My analysis started by scraping tweets about films released during 2021 using snscrape. Snscrape is a Python package that allows you to do just that (Desai, 2021). Extracting data is one thing, but how did I determine the criteria for what I am evaluating? I used North American box office data for 2021 pulled from *The Numbers* (The Numbers, 2021). This website has a complete listing of all films released within the U.S. and Canada, along with their respective box office performance. Using The 80/20 Rule, I considered the top films that generated 80% of the box office revenue in 2021 (31 films) as my positive set. I pulled my negative outcome variables from the remaining films. To filter down to the appropriate time window (before the release of the movie), I will use the 30-day period prior to the film's release (release date – 31 (the day prior to release+ 30 days)). I will also need to filter down to only movie-relevant data and will accomplish this by pulling the most popular movie-specific hashtag for each film from Twitter (Twitter, 2022). Using this, I scraped tweet data from Twitter to build my corpus.

In building my corpus, my first step was determining which movies compose each class. I pulled the 2021 box office data for North America from *thenumbers.com* using a simple copy and paste into an Excel spreadsheet. Since the web scraping assignment was later in the course, I did not know how to scrape this data using Python, but eventually learned. In fact, this was my submission for that assignment. However, due to my lack of knowledge at the time, I just copied the table into Excel. This data set included the rank (in terms of revenue), the release date of the film, the distributor, the genre, the 2021 gross revenue, and the gross tickets sold.

Next, I performed a few simple calculations to get the basis for my classification. First, was to sum the total box office revenues for 2021, which amounted to $4,512,603,979. Next, I calculated 80% of this sum ($3,610,083,183) to determine the cutoff point. I then created a column for cumulative total by row, so I could determine which film would be the cutoff point. Then, I created a column with a binary variable to identify which films were part of the top 80% (1 = positive, 0 = negative). I then needed to determine how to handle the remaining films that were not part of the top 80%, which accounted for 375 films out of a total 409. I decided to use the next 69 films as my negative set, which makes the total films analyzed 100. For all films, I calculated the start date and end date for the Twitter data I would be collecting and added these to individual columns.

With the films that would compose both the positive and negative classes identified, I started researching the most appropriate search string to use in Twitter. I determined that the best way to ensure that I pulled mostly relevant data was to use the most commonly used hashtag that was specific to each film. Using only the title of the film did not seem like a good idea since many films use commonly used words in their titles, which would bring in all sorts of irrelevant data. An example of this, is the use of the hashtag "#FreeGuy" instead of the title *Free Guy*, which would have pulled in all Tweets that contained these very common words. With the basis for my classification set now completed, I separated the positive and negative corpuses into separate CSV files containing only the data I needed for my search string (hashtag, start date, end date).

The next part was pretty fun and challenging. I had no idea how to do social media scraping prior to this assignment but wanted to build a tool to do just that. What I ended up creating was a program that reads both of the CSV files I created (POS_list and NEG_list), scraping the relevant Tweets from Twitter, then turning these into individual text files. This entailed using os.listdir to iterate through the files in the folder, creating a search string in a format specific to snscrape (example: "#SpidermanNoWayHome lang: en since:2021-11-16 until:2021-12-16), and appending each of these strings to a list. I then iterated through this list of search strings to scrape the appropriate Tweets using snscrape, which I then stored in a new list. This list of Tweets was then stripped of all urls, which are very common in Tweets, using RegEx. This was done to make sure that the Tweet data contained only relevant text and emojis. The last step was to create a folder for each corpus (POS, NEG) and dump a text file for each tweet into the appropriate folder. With all of this done, my corpus was complete.

Now, I was ready to start building and testing potential models. For this, I decided to use Scikit Learn. I have some (brief) experience using this library from a previous course and wanted to explore it further. The first step in this part was reading in my corpora and transforming them into a dataframe using Pandas. During this step, each Tweet was assigned the appropriate class label (POS = 1, NEG = 0). Next, the dataframe was split into training and testing sets. I decided a 60/40 (train/test) split would be appropriate, so I could maximize the test data and hopefully get better insight into the accuracy of the resulting model. I then set up my Scikit Learn pipeline using the TFIDF Vectorizer and the Multinomial Naïve Baye's algorithm. I figured that Naïve Baye's was a good place to start and the only one that I have experience with. I also used the TFIDF Vectorizer for preprocessing. Then I used GridSearchCV from the Scikit Learn library to explore the various values for the different parameters (normalization, document frequency, etc.). GridSearchCV is a method within the Scikit Learn library that uses a cross-validated grid-search to optimize the parameters used during the machine learning process (Scikit Learn, 2022). Since each analysis is different, the optimal parameters for each analysis may be different. As such, iterating through a dictionary full of parameters is a necessary step to ensure the best results from the analysis. Running just this portion took overnight for my computer to process, as it iterated through each possible combination of parameters. After the optimal parameters were determined, the model was fitted using the training corpus. Once the model was built, it was tested on the test data, which resulted in the confusion matrix below.

	POS	**NEG**
POS	14341	211
NEG	881	9889

Recall	94.21%
Precision	98.55%
Accuracy	95.69%
F Measure	96.33%

As you can see from the results, the model shows promise. Not only did it have an accuracy of over 95%, but the other metrics are adequate as well. Of particular note is the F Score of 96.33%, which takes the precision and recall into account. At this stage, it seems that we have built a suitable model for further testing. How would it perform against previously unseen data?

To test my models, I have decided to use year-to-date box office revenues for 2022 (The Numbers, 2022). Following the same process outlined above for determining my positive and negative corpora, I generated the same Tweet data for each film in the 2022 testing set. I modified my tools slightly for this step to only evaluate one film at a time. With all of this built, I decided to test the first iteration's model on this data, which resulted in the confusion matrix below.

	POS	NEG
POS	15	16
NEG	0	0

Recall	100.00%
Precision	48.39%
Accuracy	48.39%
F Measure	65.22%

As you can see, the results are not good. Yes, it did a great job of identifying all examples from the positive class, but this is only because it classified every film as positive. Out of the 31 films that compose the 2022 test set (15 of which are positive and 16 of which are negative), it misclassified 51% of them. Admittedly, this is not a great start, but this was only the first iteration. In future iterations, I experimented with different algorithms, parameters, and data points. The way I see it, this single failure is not a bad thing and can be learned from. After all, the entire point of this analysis is to answer a question that had not previously been answered. Can social media sentiment from Twitter be used to differentiate between high performing films or not? Even if the totality of this experiment ends in a negative result, I still would have learned something. This is what is at the core of analysis and the scientific process: learning, not only from successes, but from failures as well.

Further Analysis

Continuing down the path on which I had already started, I decided to experiment with different data transformations, parameters, and machine learning algorithms. My first step was to look at the raw data and determine if any further transformation was necessary. After reviewing hundreds of raw tweets, a pattern emerged: most of the tweet data still contained URLs. Despite using RegEx to strip out the URLs from each tweet, it seems that quite a few of them still remained. This was undesirable in my corpus as I did not want the machine learning algorithms to look at the unique strings contained within the URLs and treat these as discriminant features. After multiple iterations of creating a more efficient URL stripper, I finally developed a reliable method for doing so. This was achieved by turning each tweet into a list, using the split operator, then iterating through each word in the list and searching for the string “http” in each word. Then, only words not containing “http” were recombined into a string using the join operator. This left only clean tweet data, free from all URLs. Now that my corpus contained cleaner data, I turned to parameter optimization.

Using GridSearchCV, I ran my current model through the pipeline with the now cleaner corpus. This did result in slightly adjusted parameters for the TFIDF vectorizer, which now had an optimal maximum document frequency of 20% (0.2) as opposed to the 10% (0.1) from the previous iteration. However, the parameters for the Naïve Baye’s classifier did not change during this round of parameter optimization. The resulting classifier was no better at predicting the desired classes (top 80% of revenue vs. bottom 20%). Pursuing this analysis further, I decided to experiment with different machine learning algorithms. Perhaps using a different algorithm would result in a more accurate classifier.

First, I used the LinearSVC classifier from the Scikit Learn library. The LinearSVC machine learning algorithm is a linear form of support vector machine learning (Scikit Learn, 2022) and was highly recommended by many articles about classification analysis. As such, I decided to give it a try. The GridSearchCV process proved to be too much for my computer to handle and I was unable to successfully run a full cross-validated grid-search on the LinearSVC pipeline. Instead, I manually toggled the parameters through multiple iterations before settling on a classifier with the greatest accuracy. This resulted in the confusion matrix below.

	POS	**NEG**
POS	10656	109
NEG	180	8024

Recall	98.34%
Precision	98.99%
Accuracy	98.48%
F Measure	98.66%

Versus the Naïve Baye's classifier, the LinearSVC scored much higher in every aspect. Most surprising is the change in recall, which increased by almost four percentage points. Additionally, the F Score increased by two percentage points. Surely if a relationship existed between Twitter sentiment and box office revenue, this classifier would help me find it.

	POS	**NEG**
POS	0	0
NEG	15	16

Recall	0.00%
Precision	N/A
Accuracy	51.61%
F Measure	N/A

The confusion matrix above is from the LinearSVC tested on the year-to-date box office data for 2022. As you can see, it did not do very well. Yes, the accuracy of the classifier was slightly better than a coin toss. But that is only because it classified every instance as negative and there are 16 films in the negative class. Not willing to give up just yet, I decided to try one more machine learning algorithm.

During my research, the second-most recommended machine learning algorithm for classification analysis was gradient boosting. Gradient boosting is a modified decision tree algorithm, that optimizes a loss function by iterating through multiple splits, referred to as "weak learners" (Brownlee, 2016). In the Scikit Learn library, this can be found under the sklearn.ensemble.GradientBoostingClassifier class method (Scikit Learn, 2022). Much like with the LinearSVC algorithm, my computer was unable to effectively run the GridSearchCV optimizer with gradient boosting. Once again, I toggled the parameters through multiple iterations to arrive at the optimal outcome. This resulted in the confusion matrix below.

	POS	**NEG**
POS	14156	210
NEG	1437	9489

Recall	90.78%
Precision	98.54%
Accuracy	93.49%
F Measure	94.50%

The results were not much different from the Naïve Baye's algorithm. However, given that the accuracy and F Score were both slightly better, I still tested it on the 2022 year-to-date box office revenue data.

	POS	NEG
POS	0	0
NEG	15	16

Recall	0.00%
Precision	N/A
Accuracy	51.61%
F Measure	N/A

Much like with the LinearSVC, the gradient boosting method classified all results as negative. While it may have shown promise with the 2021 test data, it did not return the desired result with the 2022 data. It was at this stage that I decided to change up my approach and try another line of analysis.

A Shift in Direction

Analyzing Twitter data that I had extracted and classified myself proved to be a fruitless venture. At least within the context of using it as a means to discriminate between movies that represented the top 80% of box office revenue generators and the bottom 20%. My initial hypothesis was that Twitter data could be used prior to movie release to determine class membership. Given the results thus far, I declared this hypothesis dead and failed to reject the null hypothesis that this data could not be used for the purposes of classification. However, this does not mean that I was ready to give up.

What other potential relationships may exist between movies in the top 80% of revenue generators and the bottom 20%? This is the question that I moved forward with. I already had other data points to work with from my existing box office data. This included data such as the film's release date, distributor, genre. Maybe there were relationships to be found among these. However, I was still not willing to give up on the social media aspect of this analysis. Perhaps there was another way of using social media data to establish relationships between these films.

Rather than building a classifier that discriminated between classes of box office revenue, I decided to build a sentiment analysis classifier which could discriminate between positive and negative Tweets. To do so, I would need a large amount of Twitter data in which the individual Tweets were classified as either positive or negative. This meant either extracting Twitter data and classifying it manually or using a dataset that already existed. I opted for the latter.

To build my sentiment classifier, I extracted the Sentiment140 dataset from Kaggle. This dataset has been used several times over for this exact purpose and has proven to be a reliable starting point for many analyses. It contains 1.6 million Tweets that have been classified into one of three possible categories (0 = negative, 2 = neutral, 4 = positive) (ΜΑΡΙΟΣ ΜΙΧΑΗΛΙΔΗΣ KAZANOVA, 2017). In addition to the classification category, it also includes data such as the unique identifier of each Tweet, the date of the Tweet, the user that created the Tweet, and the text of the tweet itself. Using this dataset, I built classifiers using all three of the same algorithms I used for my first analysis. This resulted in the three confusion matrices below.

Naïve Baye's

	POS	**NEG**
POS	257192	62806

NEG	77902	242100

Recall	76.75%
Precision	80.37%
Accuracy	78.01%
F Measure	78.52%

LinearSVC

	POS	**NEG**
POS	258493	61505
NEG	71539	248463

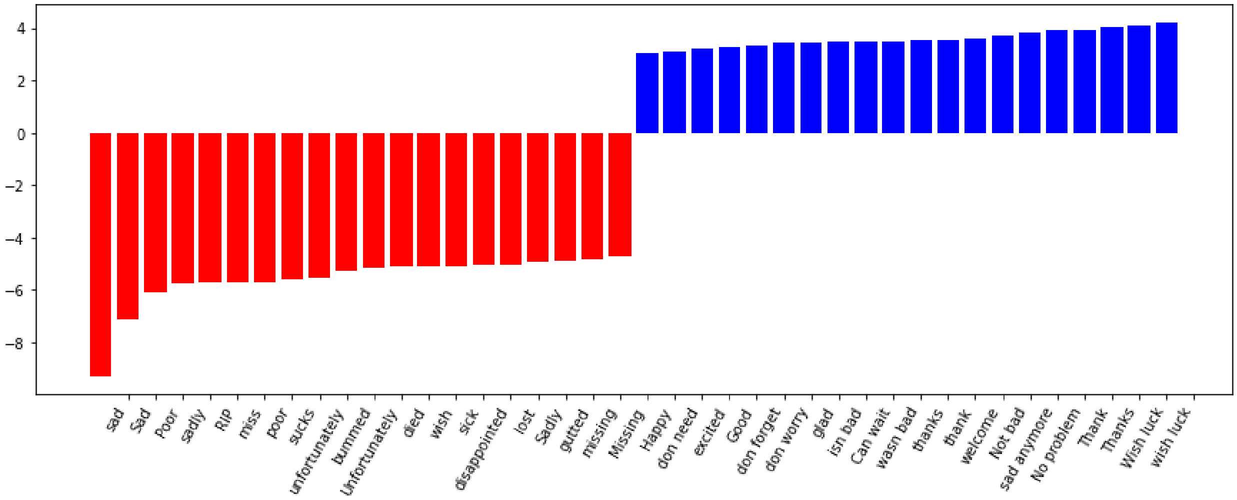
Recall	78.32%
Precision	80.78%
Accuracy	79.21%
F Measure	79.53%

Gradient Boosting

	POS	**NEG**
POS	149491	170507
NEG	42479	277523

Recall	77.87%
Precision	46.72%
Accuracy	66.72%
F Measure	58.40%

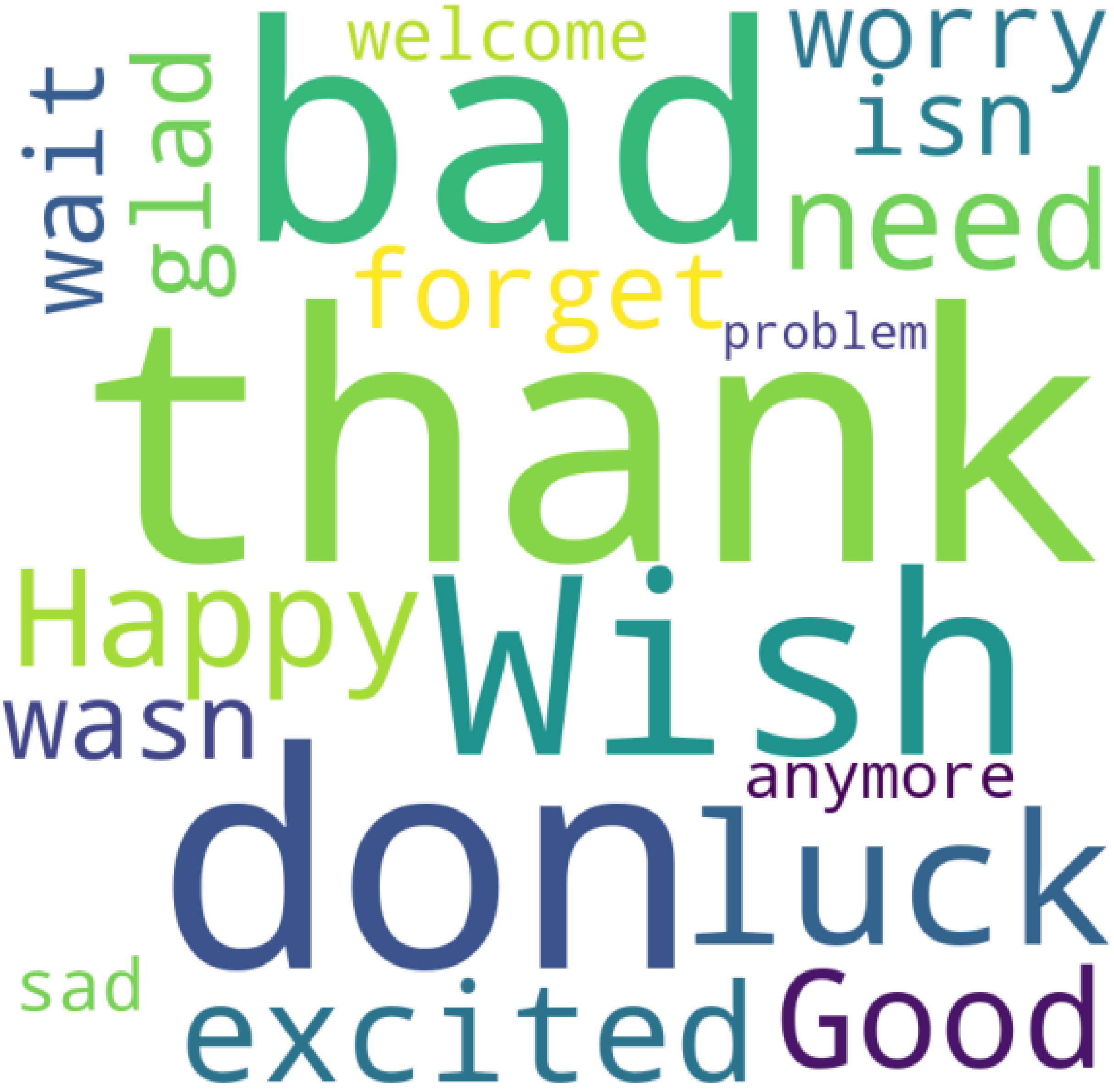
Given that the LinearSVC resulted in the model with the best metrics overall, I decided to deploy this classifier for my analysis. Prior to doing so, I wanted to know exactly what features were important in determining whether a Tweet fell into the positive class (4) or the negative class (0). I extracted the top features for each class, which are depicted below. Prior to vectorization, the words were stemmed to avoid redundancy during the machine learning process.



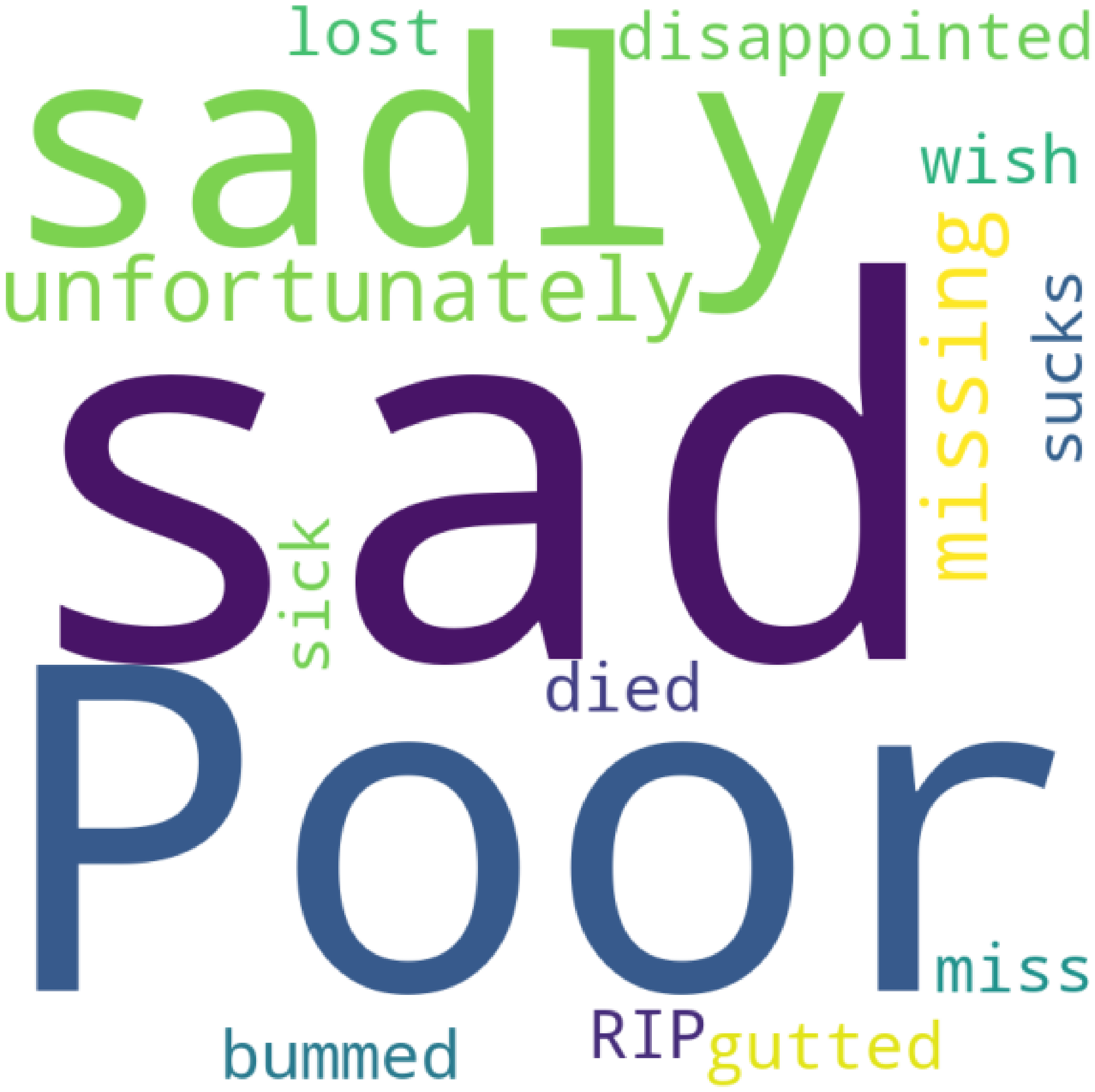
This chart depicts the top 20 features in each class, ranked by their relative weight.

Red = Negative

Blue = Positive



This Word Cloud is shows the top features of the positive class.



This Word Cloud shows the top features of the negative class.

Using this classifier, along with the Tweet scraper I had already built, I gathered a sample of 1,000 Tweets from each movie, which were classified as either positive (4) or negative (0). This data was then aggregated into four data points (Total Positive Tweets, Total Negative Tweets, Total Tweets, and Percentage of Positive Tweets), which were added to the existing dataset. With this new batch of social media data collected, I conducted further analysis on the films released in 2021, hoping to find some interesting patterns.

I summarized the data by class (top 80% of revenue generators (1) and bottom 20% (0)) and started to analyze this data for patterns. The first thing I looked at was the percentage of positive Tweets in each class. In the positive class (1), the Tweets were 70.26% positive. However, in the negative class (0), the Tweets were 70.88% positive. I was sure that this meant there was no significant difference between classes, but still ran an ANOVA analysis to confirm this, resulting in the tables below.

SUMMARY

Groups	*Count*	*Sum*	*Average*	*Variance*
Top 80% Percent Positive	31	21.9534	0.7082	0.0097
Bottom 20% Percent Positive	69	50.4457	0.7311	0.0119

ANOVA

Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Between Groups	0.0112	1	0.0112	0.9990	0.3200	3.9381
Within Groups	1.1024	98	0.0112			
Total	1.1137	99				

Given that both the P-Value was greater than the Alpha value of 0.05 and the value of F was not greater than F Critical, I could not reject the null hypothesis that the means of both populations are equal.

Just for the sake of curiosity, I also analyzed the average percentage of positive Tweets within each class. While averaging a percentage may not be considered ideal in many instances, it can sometimes be another indicator of potential differences between classes. With that said, the average percentage of positive Tweets within the positive class (1) was 70.82%, while the average percentage of positive Tweets within the negative class (0) was 73.11%. Do more people actually view the lower-performing movies more favorably than the top performers? This would seem to be what this indicates. That is when I noticed something else interesting.

I had already established that there were far more films in the negative class than in the positive class. This point was the main driver of this analysis; that 90%+ of all films released had to fight over only 20% of total box office revenue. The implications for what this meant within the context of social media sentiment analysis would soon become apparent. In total, there were 27,683 Tweets from the positive class and 48,752 Tweets from the negative class. When divided by the total number of films in each class (positive = 31, negative = 69), this averages to 893 Tweets per movie for the positive class and 707 Tweets for the negative class. This is a difference of 26.39% between classes, which is further exemplified when looking at just the positive Tweets. The average number of positive Tweets for films in the positive class is 627. The average number of positive Tweets for films in the negative class is 501. This is a difference of 25.28%.

To drive this point even further, 87.10% of films in the positive class had at least 1,000 tweets (27 out of 31). Opposed to this, only 56.52% of films in the negative class had at least 1,000 Tweets (39 out of 69). This final revelation is when it finally hit me. I had been missing a critical data point all along. Vital information could be gleaned not just from the sentiment contained within the Tweet data but from the amount of Twitter activity around each film.

Now knowing that there were significant differences in the level of Twitter activity around films in each class, I needed a way to quantify this. I chose to standardize the Twitter data in Excel to make them directly comparable among classes. This way, when I analyzed a data point such as the number of positive Tweets that a film received, the relative volume of these Tweets would also be taken into account. Data that were standardized for each film include the sum of positive Tweets, the sum of Negative Tweets, the sum of Total Tweets, and the percentage of positive Tweets. Running an ANOVA on the standardized number of positive Tweets resulted in the tables below.

SUMMARY

Groups	*Count*	*Sum*	*Average*	*Variance*
POS Top 80% (Standardized)	31	10.3427	0.3336	0.6803
POS Bottom 20% (Standardized)	69	-10.3427	-0.1499	1.0969

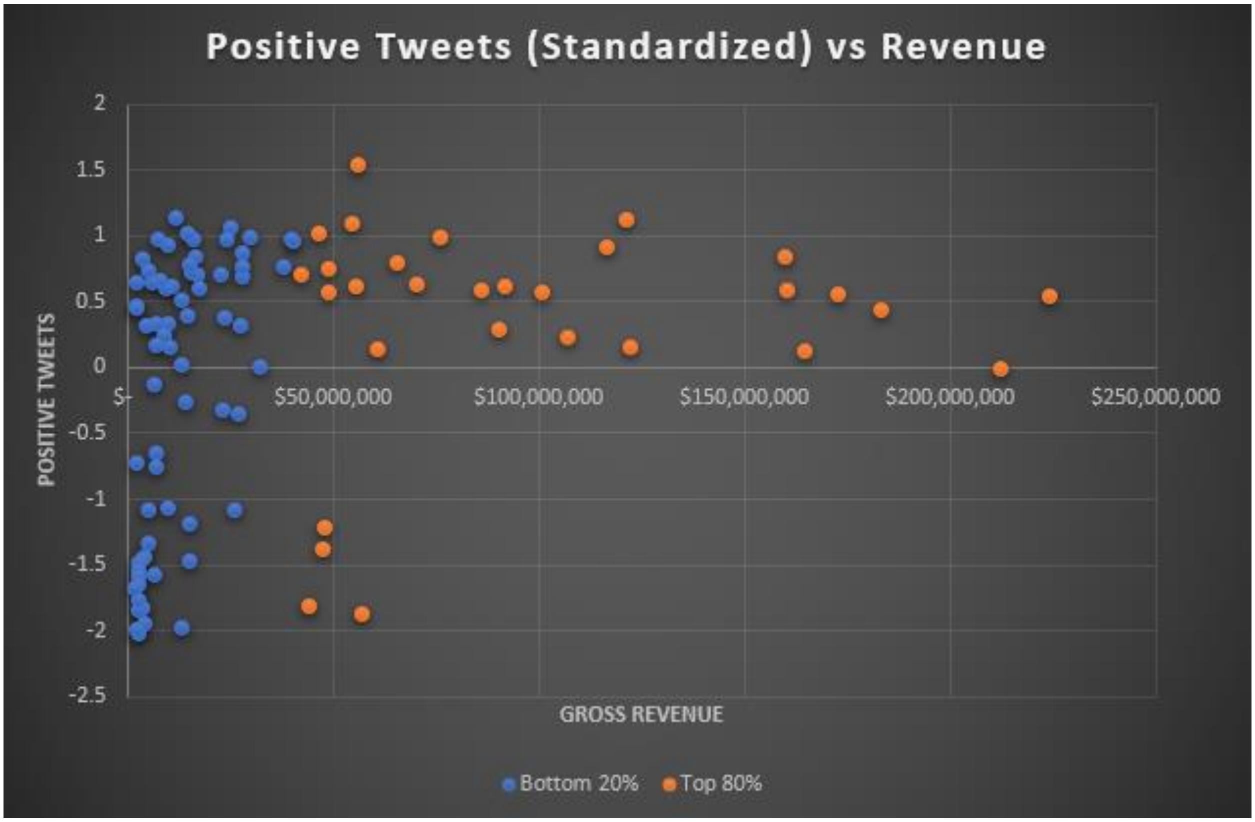
ANOVA

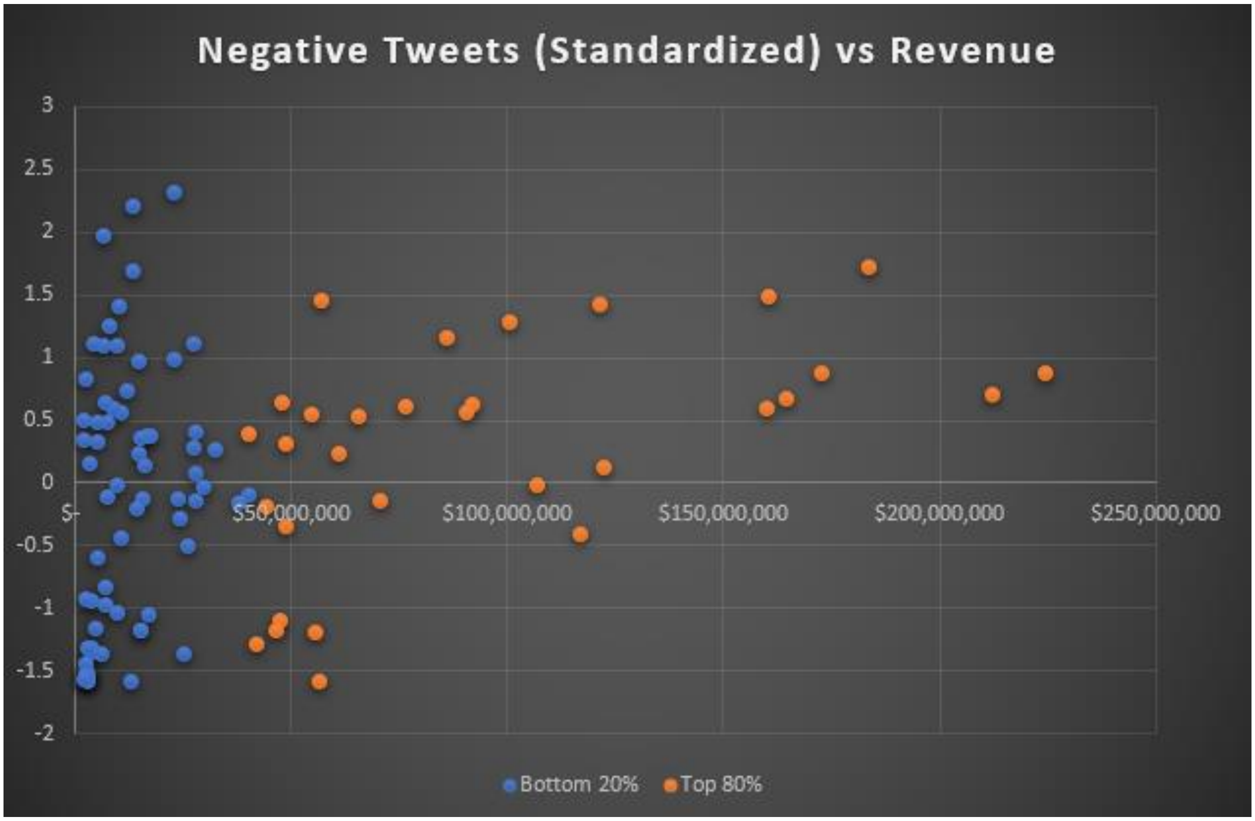
Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Between Groups	5.0010	1	5.0010	5.1590	0.0253	3.9381
Within Groups	94.9990	98	0.9694			
Total	100	99				

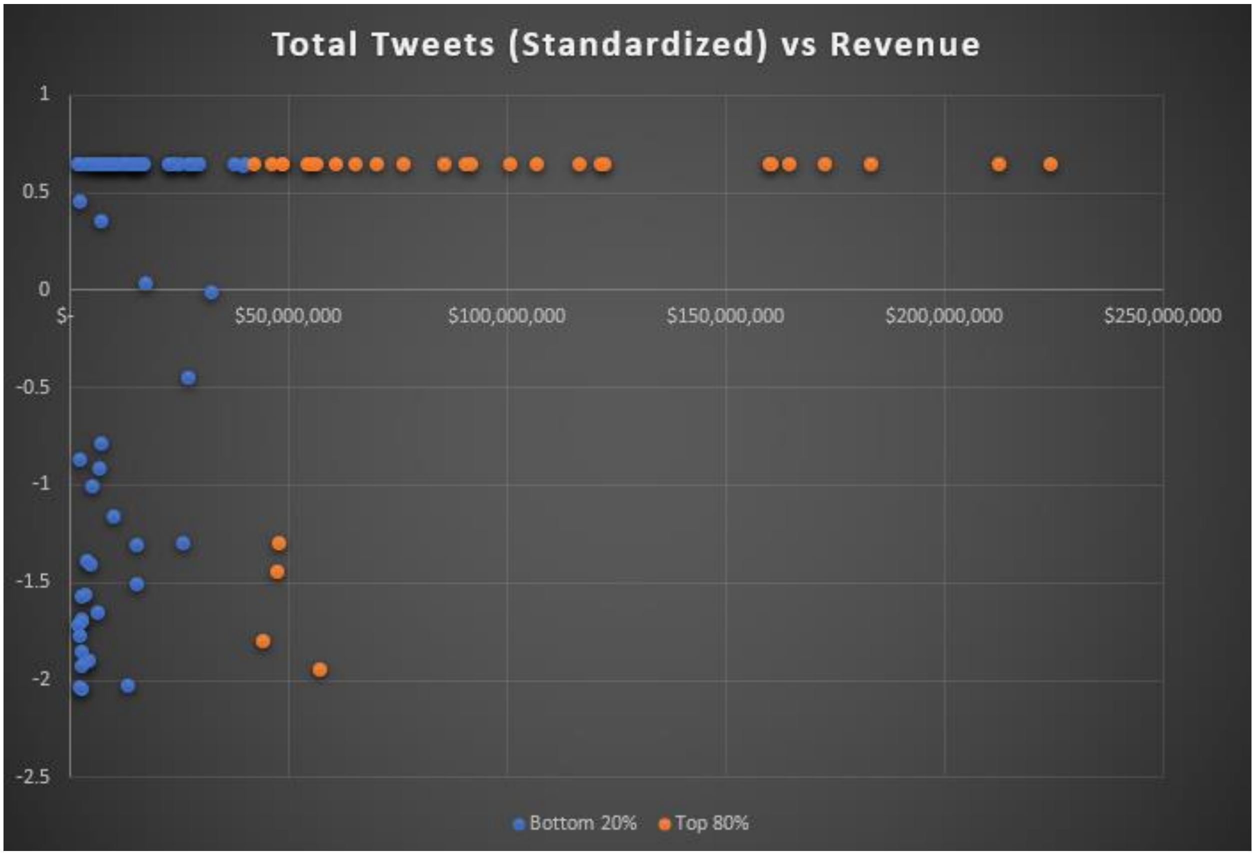
Not only was the value of F greater than F Critical, but the P-Value was less than the Alpha of 0.05. This shows that when the data is standardized to account for the differences in social media activity between classes, there is a statistically significant difference in the quantity of positive Tweets for films in each class.

Interpretation of Results and Further Questions

After a long journey and over multiple iterations, I had now established that there were statistically significant differences between classes. Now, I wanted the answers to two more questions: What was the extent of these differences? And could they be used to predict class status? To answer the first question, I looked at visual representations of the data. The charts below show the standardized data plotted against the total box office revenue generated by each film. The orange dots represent the positive class (top 80% of box office revenue for 2021) and the blue dots represent the negative class (bottom 20% of box office revenue for 2021). It should be noted that *Spider-Man: No Way Home* is excluded from these charts as it is a statistical anomaly and negatively affects the scale of these visuals. In 2021, it earned almost $600 million while all other films earned less than $225 million.




Negative Tweets (Standardized) vs Revenue
3
2.5
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
$-
$50,000,000
$100,000,000
$150,000,000
$200,000,000
$250,000,000
● Bottom 20%
● Top 80%


Total Tweets (Standardized) vs Revenue
1
0.5
0
-0.5
-1
-1.5
-2
-2.5
$-
$50,000,000
$100,000,000
$150,000,000
$200,000,000
$250,000,000
● Bottom 20%
● Top 80%

When displayed visually, some patterns began to emerge, and clusters began to form in my mind. There were significant differences between classes and all of these blue and orange dots were showing them to me. This is best exemplified in the graph that shows positive tweets (standardized) plotted against box office revenue. With the exception of a cluster of four films, all other films in the positive class were situated in the positive portion of the Y-axis. These were the four films I identified earlier as all having less than 1,000 Tweets. Additionally, the positive class had one more negative value, just below the zero point, with a value of -0.0154. Although this film (Venom: Let There Be Carnage) had 1,000 tweets, it had the lowest score in the class at 53.60% positive. This shows that both the quantity and the quality of the Tweet data was taken into account. The more I analyzed these charts and the patterns that emerged, I realized that there was an algorithm that does exactly what I was trying to do manually, decision trees.

Prior to building a decision tree, I decided to test my theory with an incredibly crude heuristic. Using the standardized positive Tweets for each film and a cutoff point of 0.5, I tested this theory on the films from both 2021 and 2022 year-to-date, resulting in the confusion matrices below.

2021 Box Office

	POS	**NEG**
POS	19	28
NEG	12	41

Recall	61.29%
Precision	40.43%
Accuracy	60.00%
F Measure	48.72%

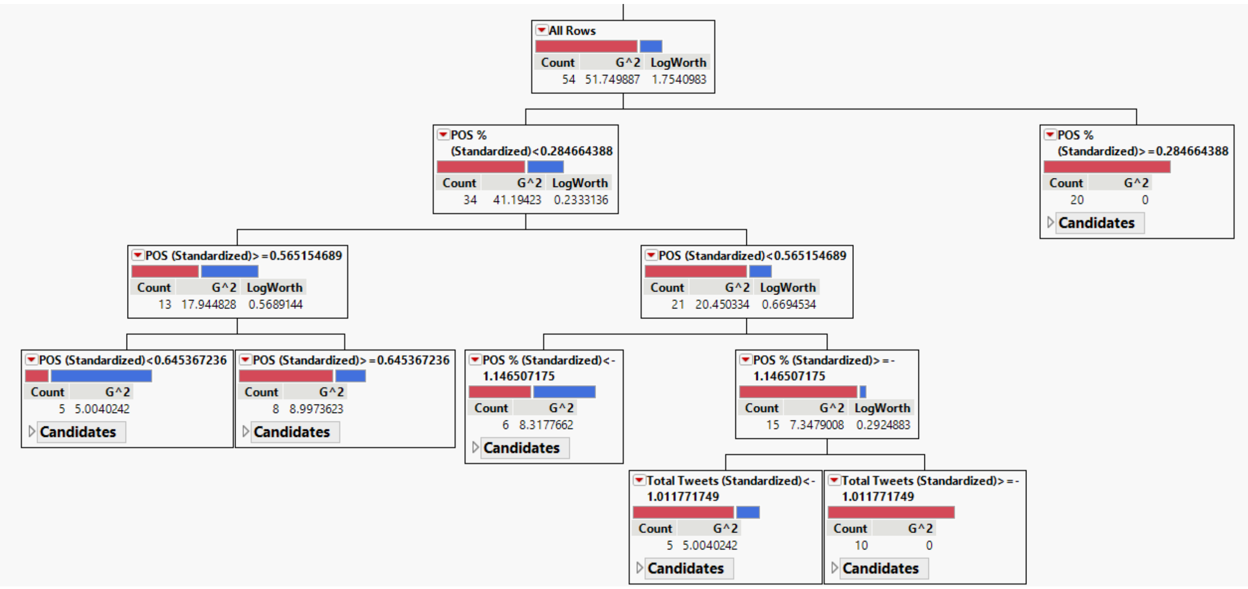
2022 Box Office

	POS	NEG
POS	12	10
NEG	3	6

Recall	80.00%
Precision	54.55%
Accuracy	58.06%
F Measure	64.86%

I was surprised to see that not only were the results positive, but that it had outperformed all other models that had been built prior to this. This incredibly crude method of discriminating between classes had the highest accuracy score of all the classifiers I had built. Not only this, but unlike the other classifiers, it actually worked on the 2022 year-to-date data as well. Although these metrics were far from ideal, they showed great promise. Given this, I decided to move forward with modeling a decision tree on this data in JMP. Surely if I could do this well with such a simple classifier, a machine could do even better.

After feeding the data into JMP, I performed the necessary preprocessing (analyzing distribution, correlations, collinearity, and dealing with outliers). Then, I ran a decision tree with the class identifier as the target variable. This resulted in the model below.


All Rows
Count 54 G^2 51.749887 LogWorth 1.7540983
POS % (Standardized)<0.284664388
Count 34 G^2 41.19423 LogWorth 0.2333136
POS % (Standardized)>=0.284664388
Count 20 G^2 0
Candidates
POS (Standardized)>=0.565154689
Count 13 G^2 17.944828 LogWorth 0.5689144
POS (Standardized)<0.565154689
Count 21 G^2 20.450334 LogWorth 0.6694534
POS (Standardized)<0.645367236
Count 5 G^2 5.0040242
Candidates
POS (Standardized)>=0.645367236
Count 8 G^2 8.9973623
Candidates
POS % (Standardized)<-1.146507175
Count 6 G^2 8.3177662
Candidates
POS % (Standardized)>=-1.146507175
Count 15 G^2 7.3479008 LogWorth 0.2924883
Total Tweets (Standardized)<-1.011771749
Count 5 G^2 5.0040242
Candidates
Total Tweets (Standardized)>=-1.011771749
Count 10 G^2 0
Candidates

The resulting decision tree model from JMP.

The decision tree model was run against the training data (below). The results were not great but showed some promise. This same model was then used on the testing set that was withheld. The results were less than ideal. Suffice it to say that I determined no further testing would be necessary on the 2022 year-to-date set.

2021 Training Set

	POS	NEG
POS	4	1
NEG	6	43

Recall	40.00%
Precision	80.00%
Accuracy	87.04%
F Measure	53.33%

2021 Testing Set

	POS	NEG
POS	2	2
NEG	8	23

Recall	20.00%
Precision	50.00%
Accuracy	71.43%
F Measure	28.57%

Conclusion

The initial goal was to discover whether or not social media data from Twitter could be used to predict the financial performance of movies. With the average cost of producing, distributing, and marketing a film estimated to be around $100 million (Mueller, 2021), studios need better and more timely insight into a film's financial performance. With less than 10% of all films released accounting for 80% of box office revenue, a better way to predict box office performance is necessary. Given that Twitter is such a widely used social media platform, I hypothesized that it would be possible to use data from Twitter to predict whether or not a film would be one of the few films that generates 80% of the revenue or one of the many films fighting over the remaining 20%. After a long and arduous journey, during which I learned many things, built many tools, and iterated through many models, the answer I came up with is "sort of".

I eventually proved that a relationship did exist between social media sentiment and class membership (top 80% or bottom 20%). There does appear to be a relationship between Twitter sentiment and box office revenue. A significant portion of this relationship is driven by the amount of activity on Twitter prior to a film's release. This is followed by the amount of positive sentiment contained within this activity.

Regarding whether or not this can be used to predict financial performance, I would say that prediction is plausible. Using an incredibly crude method, I was able to use a single data point to predict class membership with a fair level of accuracy. At this stage, I would not recommend that any movie studios use this as an indicator of performance until further study is done in this area.

With all of this said, this was an incredibly fun thought experiment and a great learning experience. I gained and used skills that I never had before, which I translated into tools that I never thought to build. It also provided me with a way to use the cumulative skills that I developed over the course of my education. Overall, I may have not built the next great predictive engine used by movie studios the world over, but I did have an incredible time trying.

References

Brownlee, J. (2016, September 9). *A Gentle Introduction to the Gradient Boosting Algorithm for Machine Learning*. Retrieved from Machine Learning Mastery: https://machinelearningmastery.com/gentle-introduction-gradient-boosting-algorithm-machine-learning/

Desai, R. (2021, June 21). *How to Scrape Millions of Tweets Using snscrape*. Retrieved from Medium: https://medium.com/dataseries/how-to-scrape-millions-of-tweets-using-snscrape-195ee3594721

Laoyan, S. (2021, August 6). *Understanding the Pareto principle (The 80/20 rule)*. Retrieved from asana: https://asana.com/resources/pareto-principle-80-20-rule

Mueller, A. (2021, December 1). *Why Movies Cost Do Much to Make*. Retrieved from Investopedia: https://www.investopedia.com/financial-edge/0611/why-movies-cost-so-much-to-make.aspx#:~:text=The%20average%20cost%20to%20produce,to%20right%20about%20%24100%20million.

Scikit Learn. (2022). *sklearn.ensemble.GradientBoostingClassifier*. Retrieved from Scikit Learn: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html

Scikit Learn. (2022). *sklearn.model_selection.GridSearchCV*. Retrieved from Scikit Learn: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html

Scikit Learn. (2022). *sklearn.svm.LinearSVC*. Retrieved from Scikit Learn: https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html

The Numbers. (2021). *North America (US and Canada) Domestic Movie Chart for 2021*. Retrieved from The Numbers: https://www.the-numbers.com/market/2021/top-grossing-movies

The Numbers. (2022). *North America (US and Canada) Domestic Movie Chart for 2022*. Retrieved from The Numbers: https://www.the-numbers.com/market/2022/top-grossing-movies

Twitter. (2022, June 29). *home*. Retrieved from Twitter: https://twitter.com/home?lang=en

ΜΑΡΙΟΣ ΜΙΧΑΗΛΙΔΗΣ KAZANOVA. (2017). *Sentiment140 dataset with 1.6 million tweets*. Retrieved from Kaggle: https://www.kaggle.com/datasets/kazanova/sentiment140