Movie Review Data Analysis
Final Project - IST 664

Christopher Fredrick
Evan Helig
Mikayla Scott

Introduction

This paper describes a classification analysis of sentiment in movie reviews using different machine learning techniques. This analysis demonstrates various classification techniques to explore and classify the dataset. The data cleaning process involved several phases of cleaning which included converting the reviews to lowercase, removing punctuation, tokenizing, removing stop words, and joining the remaining words into strings suitable for modeling.

Exploratory data analysis was conducted to better understand the dataset. The dataset consisted of 156,060 rows, with 7,072 negative, 27,273 somewhat negative, 79,582 neutral, 32,927 somewhat positive, and 9,206 positive reviews. The most frequent words across the entire dataset were analyzed, with "movie" and "film" being among the most recurring words. Bag of words and parts of speech were also analyzed to identify features for modeling.

This report will discuss data collection and cleaning, walk through exploratory data analysis techniques, show the modeling approach for the dataset, discuss the results of each of the models, share general conclusions about the study and provide recommendations for further research.

Data Collection and Cleaning

The data that was used for this experiment is a free dataset provided by Kaggle.com, the dataset can be found under the competition name, ‘Sentiment Analysis on Movie Reviews”. The reviews were scraped from Rotten Tomatoes, and are rated on a scale from zero to four. Zero is considered to be extremely negative, four is considered to be extremely positive and two is neutral. The sentiment for each review was produced by using a crowd-sourcing tool, Amazon Mechanical Turk (AMT). The dataset itself is composed of two main files, a training file and a test file, both of which are stored as tab-separated values (TSV) files. The files have identical columns (phraseID, senteceID, Phrase), however; the training file includes a sentiment column, and the test file does not. Because the test file does not have a labeled sentiment associated with the phrase in the set, it was not used for this experiment. The training set was then split into a training and testing set, where 30% of the data was reserved for testing models and the other 70% was used for training.

For the data cleaning process there were several phases that the review phrases had to go through in order to consider the reviews clean for modeling. The first step was to make all the text in the ‘Phrase’ column lower case. Once converted to lowercase, punctuation was removed from the phrases. Lowercase, non-punctuation phrase strings were then tokenized. The tokenization of the strings allowed for stop words to be removed. The stop words that were used for this experiment was the standard stop word list provided by the Natural Language Toolkit (NLTK) package. Once stop words were removed, the cleaned tokenized lists of remaining words were joined back together into strings. The cleaned strings were then ready for modeling.

Data Analysis

Before modeling, exploratory data analysis was conducted to better understand the dataset. The findings from that analysis will be discussed in this section.

The train file was composed of 156,060 rows. Of those rows there were 7,0722 negative (0) reviews, 27,273 somewhat negative (1), 79,582 neutral (2), 32,927 somewhat positive (3), and 9,206 positive (4) reviews. See the distribution in the chart below:

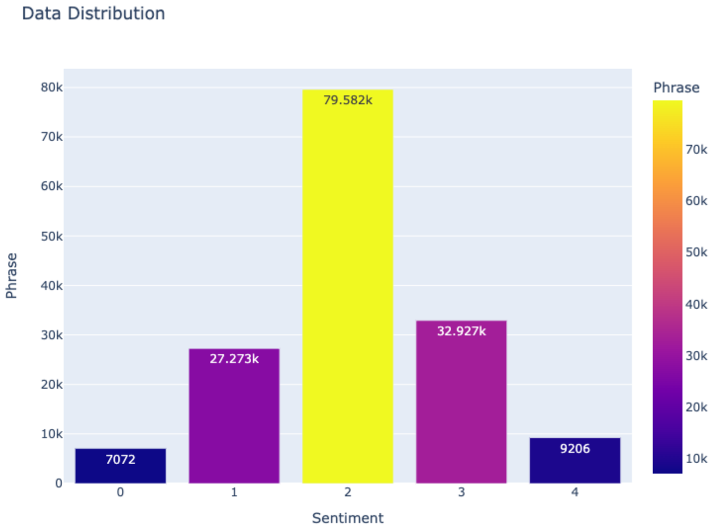


Figure 1: Data Distribution, Count of Phrases by Sentiment

The most frequent words across the entire dataset were then analyzed. Before stop word removal the dataset had 1,077,113 words, after stop word removal the dataset had 635,173 words remaining. The most frequent terms were generated using the cleaned text. Movie, and film were among the top recurring words within the dataset, both having more than 5,000 occurrences; other words like director, and plot were mentioned around 1,000 times. See the 50 most frequent words across the entire set in the figure below:

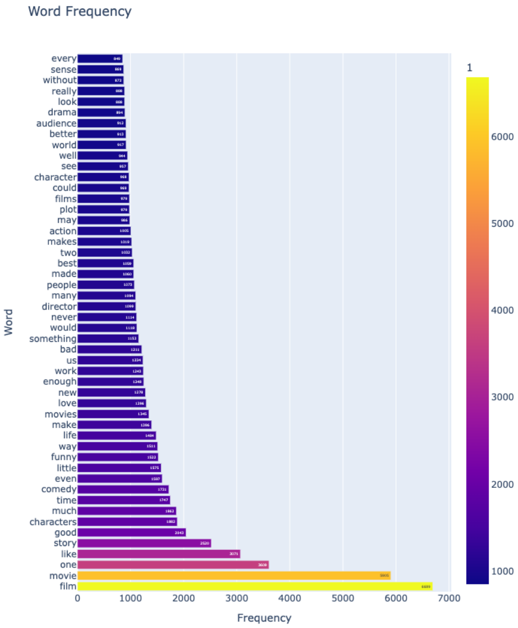


Figure 1: Data Distribution, Count of Phrases by Sentiment

Bag of words were then generated to identify features for modeling. Utilizing the NLTK bag of words function, wordlists were created for each of the sentiment categories (0-4). The 25 most common words in each of the categories are listed in the table below:

Negative		***Somewhat Negative***		***Neutral***		***Somewhat Positive***		***Positive***	
movie	760	movie	1379	film	2158	film	1844	film	936
film	480	film	1271	movie	1894	movie	1296	movie	576
bad	434	like	913	one	1283	good	859	one	499
like	324	one	720	like	1110	one	823	best	370
one	284	much	594	story	942	story	661	funny	345
characters	167	story	528	much	695	funny	629	good	289
comedy	165	little	490	time	681	like	578	performances	253
even	164	characters	455	life	647	characters	497	story	236
minutes	161	bad	427	characters	614	love	491	comedy	231
story	153	even	407	movies	571	comedy	463	great	220
would	146	time	389	little	566	life	455	performance	219
time	145	plot	356	way	560	enough	450	work	197
worst	145	way	356	comedy	542	time	421	love	196
dull	134	would	335	even	521	us	410	movies	170
plot	130	never	335	new	513	make	404	entertaining	168
action	124	good	321	make	505	even	397	well	168
way	120	comedy	320	us	487	way	389	year	165
dialogue	113	many	291	people	484	new	376	fun	165
much	111	nothing	287	good	483	best	360	films	155
little	109	could	286	two	457	great	343	characters	149
nothing	109	better	281	director	454	makes	343	life	147
long	109	make	279	love	446	something	337	like	146
really	109	enough	268	world	429	much	336	cast	134

Table 1: Bag of Words by Sentiment

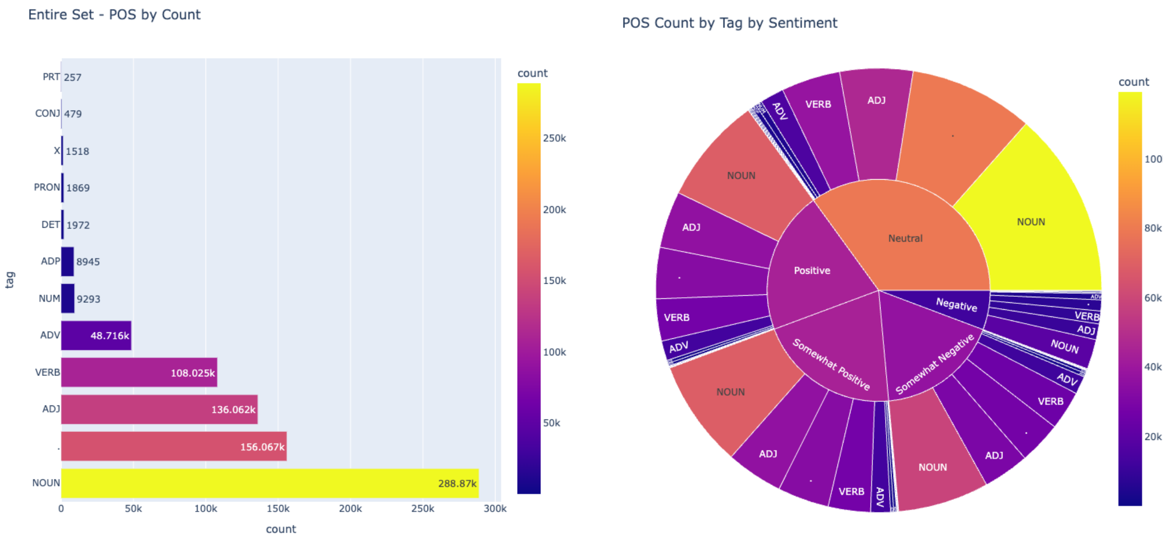
Parts of speech (POS) were identified for the whole corpus and then aggregated by sentiment classification. The POS library identifies several different types of speech. See the table below for the parts of speech and the corresponding tags:

Tag	***Part of Speech***
ADJ	Adjective
ADP	Adposition
ADV	Adverb
CONJ	Conjunction
DET	Determiner, article
NOUN	Noun
NUM	Numeral
PRON	Pronoun

.	Punctuation
X	other

Table 2: POS Tags

Tags were assigned via the NLTK POS functions. For each of the sentiment classification categories the cleaned text was used to create a frequency distribution of all the words in the scripts by part of speech. A chart with the different parts of speech for the entire corpus as well as a breakdown of the parts of speech by sentiment can be seen in the figures below.



Figures 2 & 3: Part of Speech Statistics

Modeling

NLTK

Nltk section

NLTK's Naive Bayes model was used on three different feature sets for comparison. For every model, a 70/30 train/test split was used. The first feature set involved scores derived from a subjectivity sentiment lexicon. Each word in a review was assigned one of four sentiment values: weak positive, strong positive, weak negative, strong negative. The weak words were counted once while the strong words were counted twice. The positive and negative numbers were then summed respectively to produce an output of two values for every review (Ex: {'positivecount': 4, 'negativecount':6}). The second feature set consisted of all adjectives and adverbs discovered by using NLTK's POS tagging function. For this feature set, the actual word and the part of speech tag were used as inputs for the model (Ex: {'manipulative': 'JJ', 'aggressive': 'JJ'}). For the final feature set, there was a combination of sentiment analysis and POS tagging. Every word in the review was assigned three values: the word itself, the part of speech, and the sentiment. This was the most detailed feature set(Ex: {'word': 'good', 'POS': 'JJ', 'Polarity': 'positive'}).

Sci-Kit - SVM

There were four different Support Vector Machine (SVM) models that were trained and tested for this study. Each of the models were trained and tested on the same partitioned training and testing set that originated from the training movie reviews set. Each of the four models were trained and tested using a different vectorizer. One model was vectorized using just count frequency of unigrams, the other was trained using term frequency-inverse document frequency (TFIDF), the last model that used unigrams was trained with the count frequency with the binary variable set to true. The fourth model was trained with the frequency of bigrams. Read about the results of the models in the *Sci-Kit SVM* section.

Sci-Kit - MNB

There was one Multinomial Naive Bayes (MNB) model that was trained and tested for this study. The model was trained using the same training and testing set that was used for the SVM models. The model was trained using reviews that were vectorized based on unigram frequency. Read more about the results of the MNB model in the *Sci-Kit - MNB* section.

Results

NLTK

Of the three feature sets used in NLTK's Naive Bayes model, the feature set using subjectivity sentiment lexicon performed the best. This model had an accuracy of 32.7% as opposed to approximately 31% for the other two sets. Diving into the feature importance rankings provided a clearer picture of what the models were struggling with. Among the most important features are reviews with either very high or very low positive and negative counts. The model did well with classifying reviews that were very positive or very negative. The problems occurred with smaller positive and negative counts. A positive count of 1 or 2 may be found classified in any of the groups.

There were similar trends in the remaining two feature sets. Reviews with words associated with very positive or very negative reviews were classified more accurately, while reviews with a more subtle word choice were often misclassified.

Sci-Kit - SVM

There were four Support Vector Machine models trained and tested on the movie review training set. Each of the SVM models were trained using a different vectorizer, four total. Unigram count frequency produced 62% accuracy. Some of the most negative words that were picked out from the model were cesspool, stinks and pompous, other positive words include stunning, flawless and phenomenal. The model trained with a term frequency-inverse document frequency (TFIDF) vectorizer also had 62% accuracy, and picked out a lot of the same terms that the unigram count frequency model picked out. The third model, trained with a gram 12 count vectorizer performed the worst out of all the models, at 58% accuracy. The gram 12 count vectorizer works with bigrams, instead of just single words. Some of the bigrams that the model deciphered for the negative category include time stinker, awful movie, and major waste, some of the positive bigrams that it identified were best film, superb performance, and extremely funny. The final SVM model that was trained was trained using unigrams but this time the binary variable was set to true, which set all non zero counts to one. This model had the best accuracy amongst all the SVM models at 63%.

The models struggled in classifying the same categories across the board. For the extremely negative category (0), three out of four of the models would classify the reviews in the extremely negative category (0) as reviews that belonged in the negative (1) category. The same thing happened with the extremely positive (4) and positive (3) categories, three out of four of the models classified the extremely positive (4) reviews as positive (3) reviews. The categories positive (3) and negative (1) were most often misclassified as neutral (2). The neutral (2) category had the best accuracy across the models. The model that was trained on bigrams struggled across all the categories, the model classified a large majority of the reviews as neutral across all five categories.

Sci-Kit - MNB

The Naive Bayes model achieved an overall accuracy of 61%. Some of the extremely negative words that the model picked up on were dull, mess and old, positive includes director, time and drama. The MNB model results showed similar results as the SVM models described above. For the extremely positive (4) and extremely negative (0) categories, the reviews were most often misclassified as positive (3) when the review was extremely positive (4) and negative (1) when the review was extremely negative (0). For the positive (3) and negative (1) categories the reviews were most often misclassified as neutral (2). The neutral (2) category had the best accuracy for this model.

Model Name	***Accuracy***
Subjectivity Sentiment Lexicon	32.7%
POS Tags - Adjective/Adverb	31.3%
Sentiment x POS Tag	31.2%
SVM (unigram count vectorizer)	62.4%
SVM (unigram TFIDF vectorizer)	62.4%
SVM (gram12 count vectorizer)	57.8%
SVM (unigram bool vectorizer)	62.5%
MNB (unigram count vectorizer)	60.6%

Table 3: Accuracy Across All Models

Conclusion

This project aimed to analyze sentiment in movie reviews using machine learning techniques. The dataset used for this analysis was the “Sentiment Analysis on Movie Reviews” dataset provided by Kaggle. This dataset contained 156,060 reviews that were categorized into five different sentiment categories: negative, somewhat negative, neutral, somewhat positive, and positive.

Bag-of-words and POS tagging were also used to identify the most frequent words and parts of speech associated with each sentiment category. Bag-of-words analysis revealed that certain words were strongly associated with particular sentiment categories. For example, words like “bad” and “disappointing” were commonly associated with negative reviews, while words like “funny” and “entertaining” were associated with positive reviews. POS tagging analysis revealed that certain parts of speech were more commonly associated with certain sentiment categories. For example, adjectives and adverbs were commonly used in negative reviews, while nouns and verbs were more commonly used in positive reviews.

After data cleaning and exploratory data analysis, different classification models were used to classify the sentiment of the movie reviews: Naive Bayes, Multinomial Naive Bayes and Support Vector Machine models. Two different Python libraries were used, from the Natural Language Toolkit (NLTK)the Naive Bayes package was utilized and from SKLearn Multinomial Naive Bayes and Support Vector Machine libraries were trained and tested on the same dataset. There were a total of eight models trained for this analysis. The model that performed the best out of all the models was the SVM model trained on features tokenized by a boolean count vectorizer. The accuracy of this model was 63%.

There were continuous struggles across all the models, most often seen between the extremely negative (0) and negative categories (1) and the extremely positive (4) and positive (3) categories. Extreme reviews were most often misclassified as the sentiment classification below it. In result of the continuous misclassification across the models, for continuation of analysis of the dataset a possible solution could be to reduce the number of categories. Combination of the third and fourth categories and the zero and first categories could eliminate some of the misclassification, as there are a lot of similarities between each of the classes. Another thing that could increase the accuracy of the models is to increase the number of reviews for the dataset. There are around 156,000 reviews in the original dataset, however; this number contains duplicates of each of the reviews. After removal of duplicates there are only 8,529 reviews that remain in the dataset. Adding more reviews for the models to train on could also increase accuracy. If additional review data is not available for continuous research, either undersampling or oversampling techniques should be considered to evenly distribute the dataset.

This project demonstrates the potential of machine learning techniques for analyzing sentiment in movie reviews and provides a foundation and recommendations for future research in this area.

Team Breakdown

Group Members	Work Completed
Mikayla Scott	Data Collection and Cleaning, Data Analysis, Modeling Sci-Kit-SVM, Modeling Sci-Kit MNB, Results Sci-Kit-SVM, Results Sci-Kit MNB, Conclusion
Christopher Fredrick	Data Prep, Feature Function Creation, NLTK

	Modeling and Analysis
Evan Helig	Introduction, Data Insight, Data Analysis, Conclusion