**Project Report for Stage 4**

**“Los Ancianos”**

William Hanson

CS 487

New Mexico State University

Las Cruces, NM USA

[billh@nmsu.edu](mailto:billh@nmsu.edu)

Sanford “Jay” Johnston

CS 487

New Mexico State University

Las Cruces, NM USA

[sanford.johnston@gmail.com](mailto:sanford.johnston@gmail.com)

Eloy “Frank” Macha

CS 519

New Mexico State University

Las Cruces, NM USA

[efmacha@nmsu.edu](mailto:efmacha@nmsu.edu)

# Introduction and Motivation

It’s no secret that the internet has become one of the most important ways that people use to both perform research and purchase goods and services. One of the key tools that potential buyers use is online reviews. An item or movie with an overwhelming number of positive reviews is typically considered a high-quality items and tends to attract the attention of potential buyers, while one with mostly negative reviews will be much less likely to be considered. Similarly, online reviews are one way for a company to receive user feedback on products. As noted in a recent Forbes Technology Council post, 97% of customers use online reviews as a way to find businesses. [1]

However, it is not uncommon to see reviews where the “star” or numerical rating does not match the actual review feedback. A 1-star (poor) rating may have glowing positive feedback, and vice versa. [2] These mis-marked feedback comments make the task of discovering and properly evaluating user feedback more difficult, possibly resulting in faulty purchasing decisions.

Similarly, there is a large problem with fake reviews – either reviews generated *en masse* by bots, reviews for other products (e.g., you are looking at a hammer, but the review is for a Bluetooth headset), or companies willing to outright pay for fake positive reviews designed to boost sales or negative reviews to sabotage competitors. As reported by CBS News, a study by the website Fakespot.com noted that 30% of Amazon reviews are either fake or unreliable.[3]

# Problem Definition

With many thousands, or even millions of online reviews on sites like Amazon.com, IMDB.com, yelp.com and others, an accurate review will influence purchase decisions, movie-watching decisions, or other engagement decisions. Mis-marked, fake or misleading reviews can have significant implications to those involved in the decision, and a manual review to validate these review ratings would be untenable for most businesses with significant numbers of products for sale. The problem at hand is essentially a classification task: How can we gauge whether a given review is positive or negative?

# Proposed Solution

The challenge we are attempting to solve is to accurately classify online review texts as either positive or negative. In other words, we are performing sentiment analysis with an aim at classifying review ratings.

Our proposed approach to solve this is as follows:

1. Obtain a curated dataset of text sentences, classified as positive or negative based on its sentiments. The dataset needs to be significant (> 10k). For our project, the “IMDB50K” dataset meets these requirements.[[1]](#footnote-1) This dataset contains 50,000 highly polar (positive/negative) movie reviews, already split between training and testing subsets.
2. Pre-process review text to develop a bag-of-words (or bag-of-ngrams) representation. This is a common approach to sentiment analysis and is our tentative choice due to straightforward implementation and a wide variety of available tools. The process of developing the bag-of-words includes tasks such as vectorization and tokenization, stemming or lemmatization, developing stop words, parts of speech tagging, etc. Part of the task is also deciding which tools are most appropriate and which actually add accuracy to our classifiers. For example, vectorizer tools can be found in sklearn, NLTK, and Keras, among others. It should be noted that other techniques exist, including continuous bag-of words (e.g., word2vec), Topic modeling approaches such as Latent Semantic Analysis, and lexicon-based approaches such as using SentiWordNet. At this point, we have not finalized the exact set of tools, and may add more candidates than have been identified.
3. Develop classifiers for this dataset using supervised learning approaches that are commonly used for sentiment analysis. These include Multinomial Naïve Bayes, Logistic Regression, Support Vector Machines and K-neighbors, along with neural network approaches such as Multi-Layer Perceptron, LSTM, and Convolutional Neural Networks. Depending on what levels of accuracy we achieve, either a single classifier will be picked, or we may develop an ensemble classifier based on the above approaches.
4. Test and refine against the IMDB dataset, and potentially against new datasets.
5. Analyze the results of the testing using various metrics, including accuracy, f1, precision, recall, ROC-AUC, as appropriate.

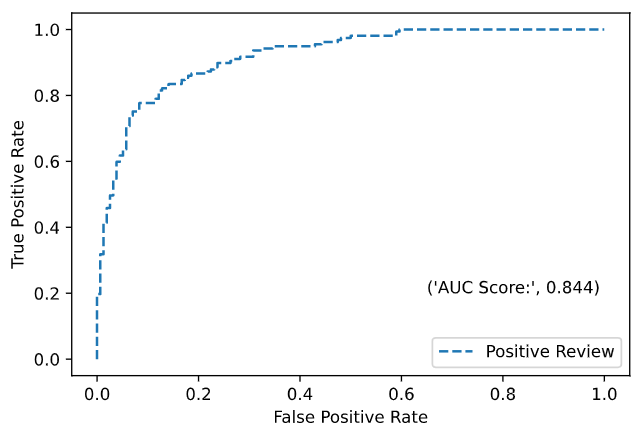
# Current Progress

To date, we have made considerable progress towards our goals. This includes generation of a custom stop word list, initial analysis of the dataset and generating various descriptive charts and figures (included at Appendix A). Several approaches to pre-processing the data using tokenization and vectorization were successfully accomplished. We have also produced three working classifiers: Logistic Regression, Multi-layer Perceptron, and Multinomial Naïve Bayes. All three classifiers produced solid results as shown in the table below:

|  |  |  |
| --- | --- | --- |
|  | Test Accuracy | Test f1 |
| Log Reg | 0.87 | 0.87 |
| MLP | 0.86 | 0.86 |
| MNB | 0.84 | 0.83 |

Table : Preliminary Classifier Performance

The ROC curve for Multinomial Naïve Bayes is illustrative of current classifier performance.

*Figure 1: ROC Curve*

# Current Tasks

While great progress has been made, there is much to do. These tasks include:

1. Continue to assess and refine the existing bag-of-words approach, including using different combinations of tools.
2. Assess whether adding a semantic or lexicon approach will improve performance.
3. Complete creation of classifiers, to include SVM, K-Neighbors, LSTM, and Convolutional Neural Networks.
4. Decide whether to pick a single classifier or develop an ensemble approach.
5. Refine hyperparameters to produce best results.
6. If able, test the classifier developed for IMDB against a dataset covering similar subject matter, such as Rotten Tomatoes reviews or Amazon Movie Reviews.
7. Finalize project report.

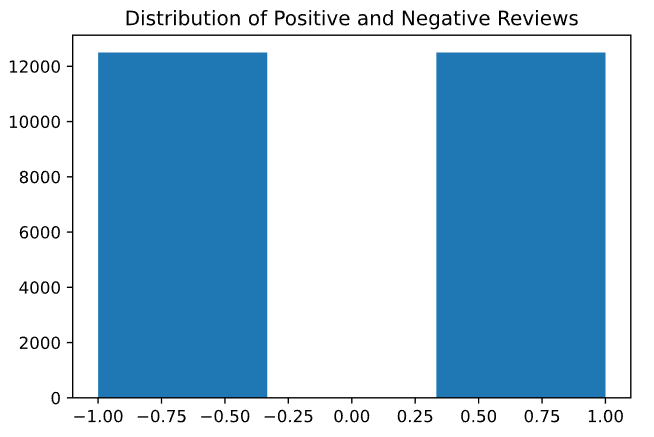
**REFERENCES**

[1] Cory Capoccia. 2018. “Online Reviews are the Best Thing That Ever Happened to Small Business.” *Forbes Technology Council*. April 11, 2018. [https://www.forbes.com/sites/forbestechcouncil/2018/04/11/online-reviews-are-the-best-thing-that-ever-happened-to-small-businesses/](https://www.forbes.com/sites/forbestechcouncil/2018/04/11/online-reviews-are-the-best-thing-that-ever-happened-to-small-businesses/#2eb45c63740a)

[2] Nishit Shrestha & Fatma Nasoz. 2019. “Deep Learning Sentiment Analysis of Amazon.com Reviews and Ratings.” International Journal of Soft Computing, Artificial Intelligence and Applications 8, 1 (Feb 2019), 1-15. <http://aircconline.com/ijscai/V8N1/8119ijscai01.pdf>

[3] Aimee Picchi. 2019. “Buyer Beware: Scourge of fake reviews hitting Amazon, Walmart and other major retailers.” CBS.com. <https://www.cbsnews.com/news/buyer-beware-a-scourge-of-fake-online-reviews-is-hitting-amazon-walmart-and-other-major-retailers/>

**Appendix A: Initial Charts and Figures**



IMDB Review Length Statistics and Boxplot

Max review length: 13700

Min review length: 52

Mean review length: 1324

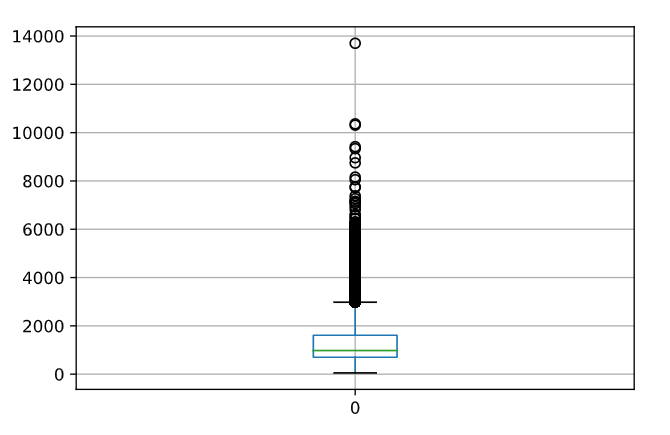
Median review length: 978

10th percentile review length: 512

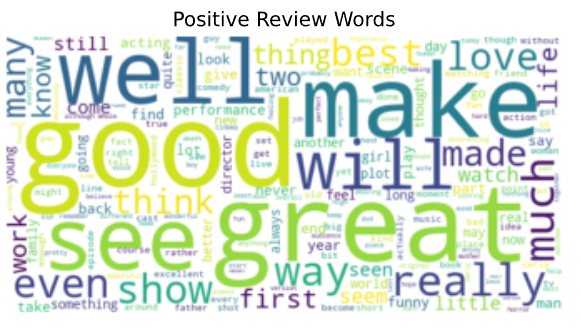
25th percentile review length: 702

75th percentile review length: 1613

90th percentile review length: 2614



Word Clouds for Positive and Negative Reviews





1. http://ai.stanford.edu/~amaas/data/sentiment/index.html [↑](#footnote-ref-1)