**Project Report for Stage 3**

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 with an overwhelming number of positive reviews is typically considered a high-quality product 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]

Accordingly, sellers pay a lot of attention to online reviews. When a buyer posts a very positive review, the seller may reach out in appreciation to the buyer. And similarly, when a buy posts a very negative review, the seller will often reach out to the buyer to apologize and attempt to rectify the situation.[[1]](#footnote-1) By providing an additional feedback mechanism, reviews can be used to guide product redesign efforts, or even whether to continue selling that product.[2] Therefore, being able to accurately locate and assess both positive and negative reviews is important for business.

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. [3] 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.[4]

# 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. A mis-marked, fake or misleading review can have significant implications to those involved in the decision, and a manual review to validate these mismarked reviews would be untenable for most businesses with significant numbers of products for sale. The problem at hand is essentially an anomaly detection task: How reliable is the review for a given product, movie or experience?

# Proposed Solution

The challenge we are attempting to solve is to accurately classify online review texts and to discover if the sentiment of the review text matches the review classification. In other words, we are facing a sentiment analysis and classification challenge.

As review ratings range in the 0 – 5 out of 5 stars, 1 – 10 out of 10 points, or some other arbretary scale, we will look normalizing the ratings to fit a positive (4+ out of 5, or 7+ out of 10) and negative (3- out of 5, 5- out of 10) characteristic. By going with this approach, our scope is more targeted in validating that indeed the indicated positive review (in stars or points) matches the given review text, or not.

A possible approach in solving this is as follows:

. Obtain a curated dataset of text sentences, classified as positive or negative based on its sentiments

. The dataset needs to be significant (> 10k). Curate the dataset to ensure it will be of similar nature as that of which you’ll be testing against.

. Identify a number of sentiment analysis algorithms, such as Naïve Bayes, Sci-Kit’s TF-IDF amongst others (Positive/Negative, Logistic Regression, SVN, KNN, etc)

. Generate the program to ingest the dataset and run against the selected algorithms

. Test against the curated dataset, and potentially against new datasets.

. Analyze the results of the testing, for accuracy, precision, ROC Curves, Precission vs Recall, etc

This approach should help us confirm back what type of rating (positive/negative) is for any given review.

**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] Lei Zhang, Xuening Chu, & Deyi Xue. 2018. “Identification of the To-be-improved Product Features Based on Online Reviews for Product Redesign.” *International Journal of Production Research* 57, 8 (Sep 2018), 2464-2479. DOI: <https://doi.org/10.1080/00207543.2018.1521019>

[3] 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>

[4] 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/>

1. One of the authors bought two separate items online recently. Both arrived inoperative. After posting negative reviews, the author was offered both a refund and a replacement item. [↑](#footnote-ref-1)