

Sentimental Analysis of Amazon Reviews

Sagar Punn
Computer Science
Ryerson University
Toronto, Canada
sagar.punn@ryerson.ca

Adam Sorrenti
Computer Science
Ryerson University
Toronto, Canada
adam.sorrenti@ryerson.ca

Eisa Keramatinejad
Computer Science
Ryerson University
Toronto, Canada
eisa.keramati@ryerson.ca

Abstract—Final report of an application level project using machine learning classification algorithms in sentimental analysis and comparing results between algorithms.

Index Terms—NLP (Natural Language), Natural Language Toolkit (NLTK), BOW (Bag of Words), TFIDF (Term Frequency Inverse Document Frequency), Negation (NEG), K Nearest Neighbours (KNN), Principal Component Analysis (PCA), Stemming, Filtering

I. INTRODUCTION

Sentiment analysis is the interpretation and classification of text-based data. The point of this analysis is to categorize each data-point into a class that represents its quality (positive, negative, etc.). Sentiment analysis focuses on the polarity, emotions, and intentions of authors. Classic sentiment analysis consists of the following steps: preprocessing, training, feature extraction, and classification. The method used in this paper will follow the classical approach while applying sentimental analysis on Amazon reviews.

A. Challenges

Due to the nature of the Amazon reviews, the five-star rating system is good, but not perfect, an indication of the positivity/negativity sentiment of reviews. Outliers became apparent when looking at the three-star reviews. These reviews were ultimately considered as inherent noise of sentiment analysis of Amazon reviews. The second challenge which is common with most NLP-related projects is having a huge body of features causing high dimensionality. When dealing with text data and using the common preprocessing models such as a bag of words or TFIDF, where each feature represents a word, this raises questions about their relevance. The main problems faced were finding solutions and compromises for the dimensionality of our feature set.

B. Prior work

Reference [11], is an example of prior work performed on similar Amazon reviews looking at sentiment analysis. Some major differences in methodologies include: First, in the processing of the dataset, despite the original dataset being unbalanced, the prior project the data was intentionally balanced so that the number of positive reviews is equal to the negative ones. For the methodology discussed here, the data was kept unbalanced in order to more accurately train our model in accordance with the real-world distribution of review sentiment. The accuracy levels achieved here actually surpassed that seen in the prior works by approximately 15% while maintaining a model true to the practical zeitgeist of Amazon review sentiment.

C. Overview

Elaborating further on the rest of the project, different methods of text preprocessing such as pure BOW, TFIDF, stemming, and filtering were used. The project started with a simple BOW representation, which provided more than 62 thousand features and finished off with a TFIDF stemmed and filtered version that reduced the number of features to only eight hundred (an approximately 98.7% reduction). The main machine learning algorithms used were multinomial Naive Bayes, logistic regression, support vector machines, k nearest neighbors clustering algorithm, and decision trees. For the implementation of these algorithms, the python scikit learn and NLTK libraries were mainly used. Aside from attaining high levels of accuracy and performant precision/recall in almost all of the methods and variations, one of the main achievements of this project was after revising the preprocessing model and reducing the number of features while maintaining or improving performance metrics within a margin of error. Furthermore, removing the three-star reviews significantly improved performance

metrics. This post-processing step supported the hypothesis that three-star reviews acted as an inherent noise and made it difficult for our model to learn the binary decision bound despite the extensive progress made on feature selection and model tuning.

II. PROBLEM STATEMENT

More rigorously, the problem faced implied classifying an Amazon review with a positive or negative sentiment, exclusively. For example, given the following review: 'Super comfortable and extremely lightweight. Great for crossfit!' Using machine learning and natural language processing is a must to identify whether the review implies a positive or negative sentiment. This entails breaking each review down into its most relevant components and then applying machine learning techniques to assign weighted sentiment scores to the samples. By the implementation of data processing techniques, which will be discussed in more detail later, the example review given previously was tokenized into the following list of sentimentally relevant words: [super, comfort, extrem, great]. Training your model on a refined list of the most relevant tokens is central to the problem of sentiment analysis and will result in a more robust model.

III. DATASET

The dataset [7] used includes 233.1 million Amazon reviews spanning from May 1996 to October 2018 and over a total of thirty unique e-commerce categories. For this project, approximately 140 thousand reviews were randomly sampled from each category proportional to the size of that category. This allowed for an optimal representation of the dataset within reasonable computational restraints. The resulting dataset is unbalanced in favour of positive reviews and the decision was made to preserve this aspect of the data to better represent the nature of the sentimental analysis of Amazon reviews. Although, preserving the unbalanced nature of the data implied taking precautions when analyzing performance metrics which will be noted later. Furthermore, the structure of the dataset followed the form of 'Summary and Review', 'Rating', 'Overall', where 'Summary and Review' is the title of a given review concatenated with the full review text, 'Rating' is either zero or one corresponding to positive or negative sentiment, respectively, and where 'Overall' is the original star rating from one to five. How exactly the ratings were assigned to the sample will be discussed in further detail in the methods and models section. Following a BOW and TFIDF approach,

the initial dimensions of the features were approximately 64 thousand degrees, well out of the acceptable range and feature selection techniques were employed to refine this. Of course, we labelled the data with a binary, positive or negative sentimental value, in accordance with the provided star ratings from the source.

IV. METHODS AND MODELS

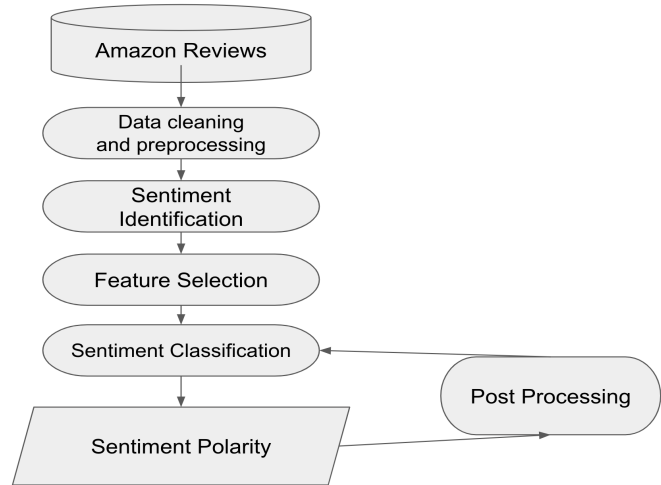


Fig. 1. Sentimental analysis process.

Fig. 1 is the sequence of steps that were taken in order to tackle the classification of sentiments. Initially preprocessing is applied on the raw dataset after which sentiment labelling/identification took place. Then a variety of experimentation is done to get multiple versions of the dataset to put inside the model. After each iteration of dataset variation post-processing methods were applied in order to gain better results and then the process was repeated on every dataset variation.

A. Data Preprocessing

From the original dataset, only review text (concatenation of summary and body text) and overall ratings were considered. Since our methodology is concerned with binary classification, overall ratings (from 1-5) were converted to 0 (1 and 2 stars) or 1 (4 or 5 stars) representing negative or positive sentiment respectively. For 3-star reviews, ambiguity arises in terms of sentiment and since the goal of this report is to apply binary classification methods, use of the neutral class is not utilized, hence 3 stars are given 0/1 label randomly.

B. Features Selection

In this case, BOW and TFIDF approach is employed where, for BOW, the dataset is converted into a matrix where features are all words in our dataset's dictionary and row contains word frequency in a review. In contrast, for each word within a review, TFIDF contains a normalized count. For both BOW and TFIDF, some baseline conditions were set where the former removes stop words and the latter only considers words with a minimum of 50 document frequency. These two baseline approaches are applied on top of the following variations of the dataset:

1) *Regular*: This is untouched review without any filtering applied.

2) *Stemmed*: In this case, Porter Stemmer is used to stemming the original review text where stemmer removes morphological affixes from words, leaving only the word stem.

3) *Filtered*: Each review is filtered to only contain positive and negative words using an opinion lexicon list [3]. Before employing the filtering process, reviews are passed through the Mark Negation method [4] which appends NEG on words between negation and punctuation mark. Furthermore, all words with NEG are considered as one single word in order to reduce noise. This process significantly reduced the number of features and BOW and TFIDF are applied at the end.

4) *Filtered Stemmed*: Here, reviews are filtered (as explained above) first and then stemmed. Finally, BOW and TFIDF are applied.

The reason these variations were created was due to large amounts of features produced by the regular model where a lot of noise seems to be generated (discussed in the Results section). Our hypothesis is that only positive/negative words have an effect on the overall sentiment of the review, therefore filtering was done. Also, stemming is employed where this report hypothesized that reducing multiple words to their root element would help in the reduction of features and removing noise.

C. Dataset Split

Dataset is split into 60:20:20 ratio where training accounts for 60% and both validation and testing account for 20%. This was done to ensure the hyper-parameters of the models are tuned based on valid set results and final accuracies are based on the test set. Furthermore, before making final predictions on the test set, the training and validation set combined in order to take advantage of the full 80% of allocated training data.

D. Models

Following are the classification models that were used in this report:

1) *Logistic Regression*: Baseline version with max iterations set in range 1000-10000 in order for model to converge.

2) *Multinomial Naive Bayes*: Baseline version used.

3) *Support Vector Machine*: Baseline version used with max iterations set in the range of 1000-20000 and, if failed to converge, the dual formulation parameter was set to false.

4) *K Nearest Neighbours*: Three variations of this model were used where the number of neighbours was set to 1, 3, and 5.

5) *Decision Trees*: First baseline version is employed after which criterion of split is set to entropy with depth of tree being 3. Last variation just consisted of the depth of the tree set to 5.

E. Post Processing

After implementing the main variations and algorithms, a method was developed to capture the mislabelled reviews. By looking at some of the mislabelled reviews it was found out that the majority of them consisted of former three-star reviews which were randomly assigned a positive/negative label during the preprocessing stage. For instance, a review that was written with satisfaction and was overall a positive one was predicted correctly however in the preprocessing stage randomization step label it a negative review. After this observation, we had a hypothesis that the three-star reviews were noisier than we thought and have almost no contribution to the learning process, and the Extreme version was devised. In the extreme version training data only consists of one and five-star reviews. By omitting the rest of the reviews the amount of noise in the training process is minimized and only the reviews that were surely positive or negative were given to the model.

F. Performance Metric

The metrics used implements functions for assessing prediction error/success and visualizations to better identify upward performance trends while training. Again, all tuning is performed in relation to the validation set to ensure the integrity of the model's predictive capabilities. The following performance metric was used and helped guide the next steps:

- 1) Accuracy Score
- 2) Precision & Recall
- 3) F1 Score

Due to the unbalanced nature of the dataset, with reviews tending to positive sentiment in general, special attention was paid to the metrics resulting from precision and recall. Accuracies, while useful for showing final performance, did not show the whole picture when tuning models and hyper-parameters. A precision-recall curve also showed the trade-off between precision and recall for different thresholds. A high area under the curve represents both high recall and high precision, where high precision relates to a low false-positive rate, and high recall relates to a low false-negative rate. Whereas, F1 score can be interpreted as a weighted average of the precision and recall, where an F1 score reaches its best value at one and worst score at zero.

V. RESULTS AND DISCUSSIONS

Variations	Logistic	Multinomial	SVM
Regular (BOW)	90.903	89.173	89.749
Regular (TFIDF)	91.357	88.853	91.192
STEM (BOW)	90.114	88.347	89.159
STEM (TFIDF)	90.623	88.264	90.827
FILTERED (BOW)	89.452	86.679	89.232
FILTERED (TFIDF)	89.277	88.433	89.242
FILTERED STEMMED (BOW)	89.328	86.228	89.084
FILTERED STEMMED (TFIDF)	89.277	88.322	89.17

Fig. 2. Accuracies in percent of test set predictions.

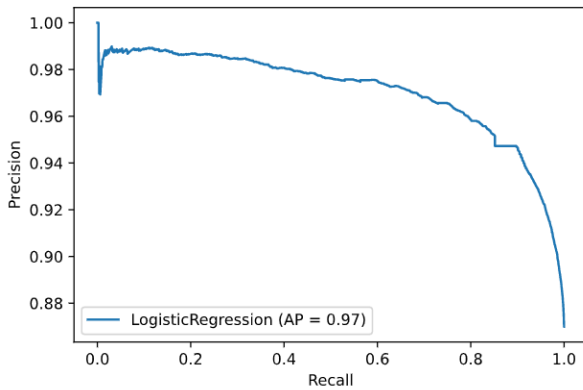


Fig. 3. Precision-Recall curve for logistic regression on Filtered-Stemmed-BOW samples.

In the beginning stages, BOW or TFIDF models were mainly used to pre-process the text and to vectorize it. This led to a huge number of features and made the dataset high dimensional. As the model was trained on this data, we achieved high accuracy on the training set but the accuracy on the validation and the testing set

were lower. This difference indicates overfitting. Since the model kept overfitting to the training data, it was clear there was an unnecessary amount of complexity, therefore variance. This was most significant in the baseline version of SVM, KNN, and Decision Tree with slightly less variance in Logistic with BOW without any filtering.

Variations	Baseline	C=Entr, D=3	D=5
Regular (Freq)	87.003	87.43	88.181
Regular (TFIDF)	86.686	87.699	88.143
STEM (Freq)	86.779	87.585	87.954
STEM (TFIDF)	86.087	87.554	87.919
FILTERED (Freq)	86.238	86.982	87.172
FILTERED (TFIDF)	86.628	86.982	87.203
FILTERED STEMMED (Freq)	86.149	86.982	87.509
FILTERED STEMMED (TFIDF)	86.593	86.982	87.516

Fig. 4. Accuracies in percent of test set for a decision tree classifier.

Variations	K=1	K=3	K=5
Regular (Freq)	86.48	88.078	88.436
Regular (TFIDF)	87.365	87.933	86.524
STEM (Freq)	84.812	87.754	88.037
STEM (TFIDF)	86.173	87.613	87.451
FILTERED (Freq)	83.696	85.949	86.586
FILTERED (TFIDF)	83.589	86.211	87.403
FILTERED STEMMED (Freq)	83.886	85.591	86.834
FILTERED STEMMED (TFIDF)	83.531	85.46	87.261

Fig. 5. Accuracies in percent of test set for a KNN classifier.

For KNN and Decision Tree, after tuning hyperparameters, the variance was reduced regardless of the kind of dataset used. Similarly, there was high variance present in the regular BOW dataset throughout Logistic, Multinomial NB, and SVM, but after applying the filtering process, variance dropped significantly, however, at the cost of a slight reduction in the accuracies.

This high variance might have been caused by the high number of features. After implementing the filter and stemmed version of the previous preprocessing stage which lowered the number of features to 1.3% of the original number, the complexity of the model decreased and the results stated that the model was closer to the goal of having low bias and low variance.

Filtered-Stemmed All Reviews	Train	Validation	Test
Baseline (max_iter:10000)	90.19	89.594	89.328
solver=liblinear, penalty=l1	90.1	89.614	89.352
solver=newton-cg, penalty =l2	90.197	89.604	89.332

Filtered-Stemmed 1&5 Reviews	Train	Validation	Test
Baseline (max_iter:10000)	96.047	95.694	95.517
solver=liblinear, penalty=l1	96.055	95.67	95.56
solver=newton-cg, penalty =l2	96.046	95.694	95.521

Fig. 6. Accuracies in percent of train/valid/test sets for a modified logistic regression classifier on one-five star reviews.

Noise trade-off was also an issue, as the first set of results were reviewed, a post-processing part was implemented in which mislabeled reviews were studied. This study revealed that a huge number of mislabeled reviews were those that originally had 3-star ratings. Since in the preprocessing stage each 3-star review was assigned a random label of positive or negative, in a lot of cases the model had predicted a 3-star rating review written with overall positivity and content to be a positive review but since it was randomly assigned a false truth label of negative, it was counted as an error. This observation suggested that the model could not fully rely on the 3-star reviews on the training and contributing more to the noise. In the next stage, the 3-star reviews were removed from the dataset and only the one and five-star reviews were passed on to the training stage. This decision to only look at the extreme sides of each end (1 and 5 stars) provided a higher accuracy (cite the table with extreme) that proved our hypothesis about the three-star reviews.

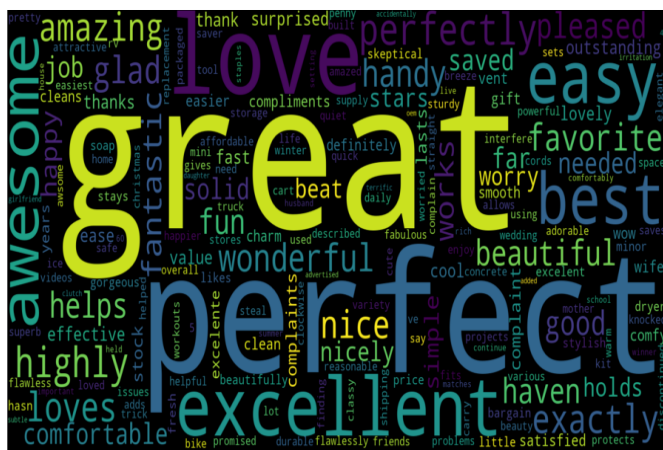


Fig. 7. Word cloud of most sentimentally indicative words.

Fig. 7 shows a word cloud representation of the most

indicative features. The size of each word/feature has a direct correlation with the weight assigned to that word by the classification algorithm. For this specific word cloud, we used TFIDF stemmed filtered variation with a logistic regression algorithm. It clearly states that the most indicative words in the project are the positive words and this reflects the nature of the dataset.

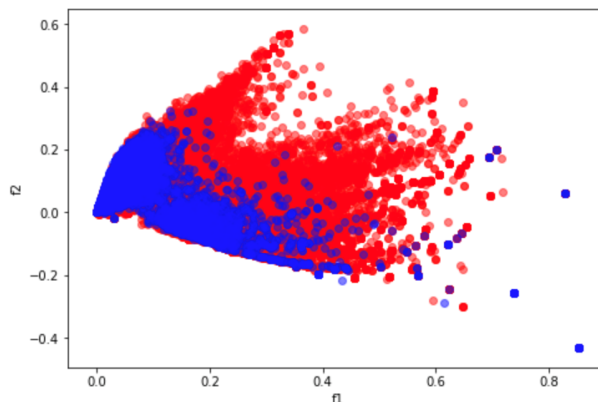


Fig. 8. 2-Dimensional principal component analysis Filtered-Stemmed-BOW features.

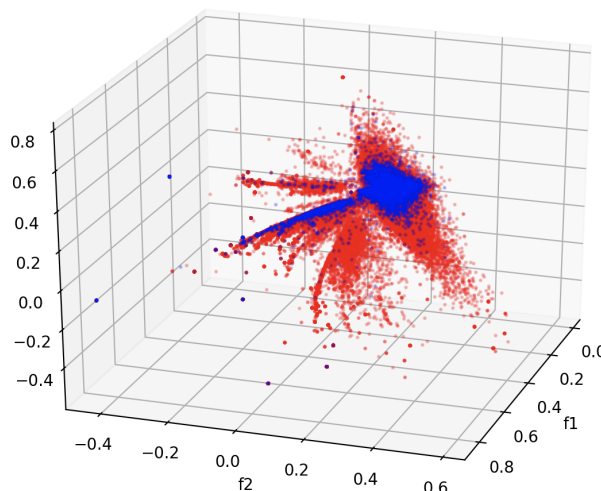


Fig. 9. 3-Dimensional principal component analysis Filtered-Stemmed-BOW features.

In order to visualize the data, a scatter plot was created with points categorized with 2 colors (red for positive and blue for negative data points). Since there were a huge number of features involved with the model, PCA was applied. The scatter plot is based on the top two principal components after applying PCA. There was no clear decision boundary dividing the two classes which indicate the complexity of the model. After taking the top three components and creating a 3D scatter plot there

still was not any clear hyperplane that would give us two distinct classes.

VI. FUTURE WORK

If the project was to be continued there could be an effective and useful approach to handle the three-star reviews and that would be introducing another class such as a neutral review class, for classification of reviews that do not fit in any of the positive or negative classes. This multi-classification approach would need a much more complex preprocessing since in some instances the data needs to have a neutral label and currently, the ratings are the main source of input for assigning a truth label to the data points. Another way this project could be improved is if Neural Networks were used in addition to the main machine learning algorithms discussed in this paper. Traditionally, recurrent neural networks were used for natural language processing projects and machine classification problems that had text as input data. Another idea could be assigning a positive (or negative) label to all of the three-star reviews in the preprocessing stage.

VII. IMPLEMENTATION AND CODE

The dataset was retrieved from the authors of the Empirical Methods in Natural Language Processing (EMNLP) paper [7]. All initial coding was done on Jupyter Notebook [10] to make the analysis process efficient and segmented. Since a large amount of data would have been computationally intensive we used a proprietary algorithm to randomly sample from the larger dataset. During the process of creating multiple dataset variations, the NLTK [4] mark negation function was used to label negated words between punctuation which helped in the filtering process. Furthermore, a dictionary of opinion lexicon (retrieved from UIC database [3]) was used in order to identify the positive and negative word in reviews. For general data handling and manipulation Pandas [8] and Numpy [5] were used. Sklearn [6] was used to implement the main machine learning classifiers. Data visualization was done with the help of wordcloud [1] library as well as matplotlib [9]. Theoretical aspect of all algorithms was covered in CPS803 class at Ryerson University. The final code is deployed on Github for easy access [2].

REFERENCES

[1] Amueller, "amueller/wordcloud," GitHub. [Online]. Available: https://github.com/amueller/word_cloud.

- [2] A. Sorrenti, E. Keramatinejad, and S. Punn, "Sentimental Analysis of Amazon Reviews," 2020. [Online]. Available: <https://github.com/mbrotos/ML-Project>.
- [3] Bing Liu. "Opinion Mining." Invited contribution to Encyclopedia of Database Systems, 2008.
- [4] Bird, S., Klein, E., & Loper, E. (2009). Natural language processing with Python: analyzing text with the natural language toolkit. O'Reilly Media, Inc.
- [5] C. R. Harris, K. J. Millman, S. J. V. D. Walt, et al. "Array programming with NumPy," Nature, vol. 585, no. 7825, pp. 357–362, 2020.
- [6] Fabian Pedregosa, et al. "Scikit-learn: Machine Learning in Python". Journal of Machine Learning Research 12. 85(2011): 2825-2830.
- [7] Jianmo Ni, Jiacheng Li, Julian McAuley. "Justifying recommendations using distantly-labeled reviews and fine-grained aspects." Empirical Methods in Natural Language Processing (EMNLP), 2019.
- [8] J. Reback, W. McKinney, Jbrockmendel, et al. "pandas-dev/pandas: Pandas 1.1.4," Zenodo, 30-Oct-2020. [Online]. Available: <https://doi.org/10.5281/zenodo.3509134>.
- [9] J. D. Hunter, "Matplotlib: A 2D Graphics Environment", Computing in Science & Engineering, vol. 9, no. 3, pp. 90-95, 2007.
- [10] Kluyver, T. et al., 2016. Jupyter Notebooks – a publishing format for reproducible computational workflows. In F. Loizides & B. Schmidt, eds. Positioning and Power in Academic Publishing: Players, Agents and Agendas. pp. 87–90.
- [11] Rathor, A. S., Agarwal, A., & Dimri, P. "Comparative Study of Machine Learning Approaches for Amazon Reviews." 2018.