Star Rating Sentiment Analysis and Topic Modeling for Amazon Reviews

Suzanne Zhen
Department of Computer
& Information Sciences
Fordham University
New York, NY

Abstract— **Sentiment analysis is a popular technique used by businesses to monitor brand and product performance. It is useful for detecting positive or negative sentiments toward a product based on customer feedbacks. In most cases, classification of sentiment is categorized into two or three classes. In this paper, the target classes are defined by the total number of star ratings present in Amazon reviews, on a scale of one to five. As a global leader in the ecommerce industry, Amazon considers customer satisfaction to be one of its core values. Therefore, it will not only be beneficial to extract sentiment polarity of the reviews, but also to identify the aspects being discussed for low rating reviews. Some examples of aspects include shipping, specific product qualities, prices, and packaging. This project applies text mining techniques, both supervised and unsupervised, to dissect the sentiments (predict star rating) toward Amazon's Grocery and Gourmet Food Department as well as to identify topics within negative sentiments.**

Introduction

In recent years, Natural Language Processing (NLP) has become an active area of research for extracting insightful information from text data. NLP is a computerized approach to understand the human language in its natural speech or text form. It encompasses numerous techniques to parse, clean, and interpret texts [1]. One major application of NLP is Sentiment Analysis, the computational study of people's opinions, attitude, and emotions toward an entity. Sentiment Analysis is a classification approach to detect the polarity and subjectivity in opinions regarding an event, topic, or experiences [2]. In this digital world with abundant options, product reviews play an important role in consumer decision behavior. A product with many positive reviews can provide justification product quality. Therefore, extracting sentiment information from product reviews is valuable for both businesses and consumers to gauge market response and take necessary actions for improvement. In addition, clustering techniques can be applied to conclude aspects or topics being discussed in the reviews. This is the process of opinion mining or opinion summarization [3].

Machine Learning Background

A. Multinomial Naïve Bayes

Naïve Bayes is a supervised classification algorithm based on the Bayes Theorem. It measures the posterior probability of the sample being in a given class based on the likelihood and prior. The likelihood is defined as the conditional probability of a sample's features given a class: $P(x_1, x_2, x_3, \ldots, x_n|c)$. The prior is the probability of the class of all samples: $P(c)$. The goal of Naïve Bayes is to find the maximum probability of being in a specific class given the features of a sample and assuming independence between features.

$$C_{map} = \underset{c \in C}{\operatorname{argmax}}\, P(x_1, x_2, x_3, \ldots, x_n|c)P(c)$$

Multinomial Naïve Bayes (MNB) is used for this specific problem since there are five classes. In addition, it is less computationally expensive since the number of parameters to be learned is drastically reduced by assuming feature independence. Classification is made based on the probability of each star rating for each review [4].

B. Stochastic Gradient Descent Regression

Stochastic Gradient Descent (SGD) is a popular technique used in various machine learning algorithms. Gradient descent is an iterative method to optimize the loss function by reaching the lowest point on the slope [5]. Since gradient descent iterates through each datapoint, it can become computationally expensive if the dataset is large and sparse with many features, which is a common problem in text mining problems. Therefore, SGD is used by randomly picking one data point at each iteration to reduce the computational expense. In this project, SGD will be used alongside regression to obtain a numeric prediction of the star rating for each review. Regression is represented by the following formula where w is the parameter to be learned:

$$Y = w_0 + w_1x_1 + w_2x_2 + \ldots + w_nx_n$$

C. *Long Short-Term Memory*

Long Short-Term Memory, or LSTM, is a type of recurrent neural network in which there are connections between nodes of one state with those of others [10]. These connection across states allow the network to learn sequential patterns and captures the semantics within sentences. The LSTM unit structure has three gates: Input, Output, and Forget Gate. As new sequential data is being passed through the model, the memory cell becomes updated by adding new data and partially forgetting history data. LSTM is very useful for large sequence of text due to its ability to retain important words or tokens [6]. In this project, word embedding will be used with LSTM to perform the sentiment classification.

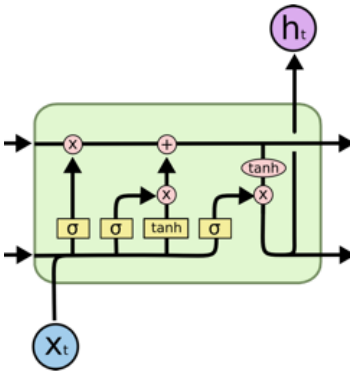

Figure 1: A single LSTM cell, σ represents a sigmoid function

D. *Latent Dirichlet Allocation*

Latent Dirichlet Allocation (LDA) is an unsupervised probabilistic model for discrete data such as text corpora. It is used to discover the topics discussed within a text document. LDA assumes that a text corpus has K number of topics. Each topic has its own distribution of words. After choosing the number of topics, each word, *w*, in the review is then randomly assigned to a topic, *t*. Based on the probability of words in the review that are assigned to topic *t,* and the probability of *w* given the topic, assignment of words to a particular topic is made [7]. In this project, LDA will be used to find the topics being discussed in low rating reviews: star ratings 1 and 2.

Experiment

A. *Dataset*

The dataset being used is Amazon Review Data published by UCSD [8]. A subset of the data from one category (Grocery and Gourmet Food) will be used to build the models. This subset of the data contains 1,143,860 total reviews from 41,320 products, with each of the products having at least 5 reviews (5-core). There are two files: Review file in CSV format and Metadata file in JSON format. In order to identify the subcategory of the reviews, the two files are combined together based on ASIN (ID of the product). Since the category field contains a list of categories in increasing specificity, for example: [Grocery & Gourmet Food, Snack Foods, Bars, Nut Bars], the second item in the list will be extracted as 'Subcategory' (i.e. Snack Foods). There is a total of 26 subcategories within the Grocery & Gourmet Food category and the rating distribution per subcategory will be analyzed. The dataset contains reviews posted from 2000 to 2018. Reviews prior to 2013 were removed since they were obsolete.

B. *Data Cleaning and Preprocessing*

The column "reviewText" will be used to perform the analysis. Prior to applying vector transformation to the data, several preprocessing steps need to take place. Two rounds of data cleaning were conducted. The first round includes tokenization (breaking down the texts into sentences or words), lowercasing of characters, and removal of punctuations, special characters, repeating characters, and numbers. After the first round of cleaning, 55% of the vocabulary was removed. The second round of cleaning include Part of Speech (POS) tagging, lemmatization (transformation to root words), and stop words removal. POS tagging is the process of assigning one of the parts of speech to each word in the review [9]. Some examples of parts of speech include nouns, verbs, adjective, adverb, and pronouns. After identifying which part of speech the word belongs to, lemmatization, the process of removing inflectional endings of a word to return the base form, known as lemma. POS tagging was performed prior to lemmatization in order to create the transformation more accurately. Last but not least, extremely common words that appear in every text corpus, namely stop words, were removed. Examples of stop words include "be", "the", "can", etc. It is important to do the stop words removal after lemmatization so the different inflections of the same word would be captured. The negative words "no", "not", and "don't" were excluded from the stop words list since I wanted to capture the negative notations for low star rating reviews. A final step was to remove reviews that became empty after all the cleaning steps.

C. *Exploratory Data Analysis, Data Imbalance, and Train-Test Split*

After the data preprocessing step, exploratory data analysis was performed to gain some initial understanding of the dataset. First, I explored the

statistics of star ratings for each subcategory to identify subcategories that are underperforming. All subcategories have a mean rating above 4.0 except for Alcoholic Beverages. At 25 percentile, subcategories that have relatively low ratings include: Alcoholic Beverages, Prepared Foods, Breads & Bakery, Snack Foods, and Soups and Stocks. At 50 and 75 percentiles, all subcategories have 5 start rating except Alcoholic Beverages.

The next step was to look into the distribution of review length. The average words per review is 35 words with majority of the review having 50 words or less (see figure 2). About 93% of the reviews have 100 words or less. A box plot of review length by star rating was created to investigate the relationship between number of words by ratings (see figure 3). At 75 percentile, low rating reviews (1 and 2) have approximately 60 words, neutral to slightly positive reviews (3 and 4) have approximately 55 words, and positive reviews (5) have approximately 40 words. In general, negative reviews have more words.

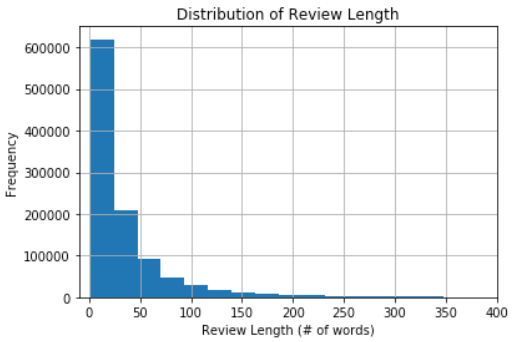

Figure 2: Length distribution of reviews

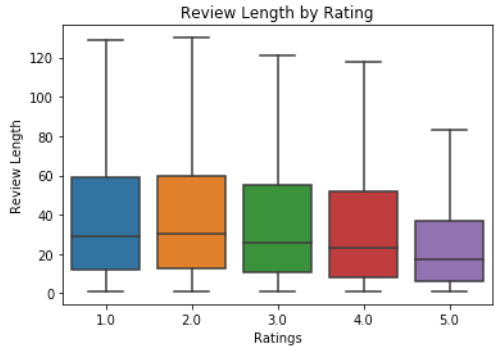

Figure 3: Length distribution of reviews by rating

Last but not least, the distribution of ratings within the dataset was examined. There was a clear imbalance in ratings since a large portion of the reviews have positive ratings (see figure 4). The star ratings have the following distribution: 72% 5-star, 13% 4-star, 7% 3-star, 4% 2-star, and 4% 1-star. In order to avoid bias during the modeling phase, the data imbalance issue was address by down-sampling. Down-sampling is a technique to reduce the amount of training samples in the majority classes to a achieve a balanced dataset. In this case, a subset of 20,000 reviews were randomly selected from each rating to create a dataset of 100,000 samples to be used for modeling. Subsequently, this dataset was split by 80% for training and 20% for testing.

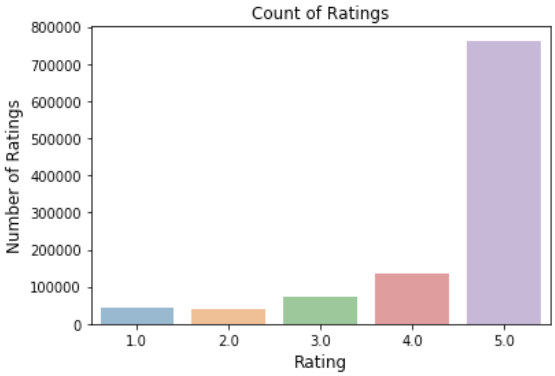

Figure 4: Ratings distribution in dataset

D. Feature extraction

Once the modeling data was created, the words in each review will be vectorized to create a document-term matrix. The document term matrix transformation is necessary before passing the data through machine learning algorithms, which require well defined fixed-length inputs and outputs. There are numerous techniques to implement this transformation. Three of the most common techniques will be used in this project. They are Bag or Words (BoW), Term Frequency- Inverse Document frequency (TF-IDF), and Word Embedding.

Bag of Words: BoW is the simplest form of feature extraction commonly used in Natural Language Processing and Information Retrieval. A BoW is a representation of text that describes the occurrence of words within a document. It provides a vocabulary of known words as well as the frequency of the known words. It is referred to as a bag because any information regarding word order or semantics will be discarded. This technique emphasizes the occurrence and frequency of words, not where they are in the document [9]. After the transformation, each word in the corpus will represent a feature or column in the matrix while each review will be represented as a sample. The words in each review become one-hot representations of their respective frequency in the review. See below for an example of the transformation. For this project, the matrix

created will be limited to only the top 3,000 words with the highest weights to avoid a large sparse matrix.

```
Review 1: "This coffee tastes great and the
package looks great. "

Review 2: "This dress looks great."

They will be transformed into the following
vector form:
```

	this	coffee	tastes	great	and	the	package	looks	dress
Review 1	1	1	1	2	1	1	1	1	0
Review 2	1	0	0	1	0	0	0	1	1

Figure 5: BoW vector transformation example

Term Frequency- Inverse Document frequency: TF-IDF is another popular technique used in Natural Language Processing for text feature extraction. One problem with the BoW model was that highly frequent words have higher scores and begin to dominate in the matrix. These frequent words may not contribute much to the information content to clearly distinguish the words used within each star rating category. TF-IDF gives more weight to domain specific or rarer words by combining two measures: term frequency (TF) and Inverse Document Frequency (IDF). Term frequency is defined as the frequency of the word in the review and Inverse Document Frequency defined as the log of the total number of reviews divided by the number of reviews with the term *t*. Words that have a high occurrence in all the documents will obtain a low IDF score [9]. See below for an example of the transformation. For this project, bi-grams, or pair of consecutive words, will be used along with the top 3000 words with the highest weights will be used to create the matrix. Bi-grams were used to retain certain semantics between words to reflect the actual sentiment. For example, "not good" has completely opposite meaning than "not" and "good" individually.

```
Review 1: "This coffee tastes great and the
package looks great. "

Review 2: "This dress looks great."

TF ("great") = (2/9)
IDF("great") = log (2/2) = log (1) = 0
TF-IDF ("great") = (2/9) *0 = 0

TF ("coffee") = (1/9)
IDF("coffee") = log (2/1) = 0.3
TF-IDF ("coffee") = (1/9) *0.3 = 0.03
```

Word Embedding: word embedding is a learned representation for text where words that have the same meaning have a similar representation. Each individual word is represented as real-valued vectors in a predefined vector space. Each word is mapped to a vector and the vector values are learned in a way similar to a neural network [11]. Word embeddings solves the problem of sparsity commonly encountered with BoW and TF-IDF by predefined dimensions. Words that are similar in meaning will have closer projections in the vector space and similarity between words is measured through cosine similarity. There are numerous types of word embeddings trained using different algorithms. Some of the most popular embeddings are Word2Vec, GloVe, and Fasttext. In this project, a pre-trained Global Vectors for Word Representation (GloVe) model based on Twitter messages with 100 dimensions will be used. The GloVe embedding contains word vectors for 1 million words and each word in the reviews was assigned a real-value vector if exist in the pre-trained model. The embedding was then passed through a LSTM model to perform the ratings classification.

E. Experiment setup

a) Sentiment Analysis (star rating predictions)

Three different types of model were used to classify or predict the sentiment or star ratings: Multinomial Naïve Bayes, SGD Regressor, and LSTM. The setup of the experiment is as follows:

- BoW transformation with MNB (1 Laplace soothing) and SGD Regressor (l2 regularization)
- TF-IDF transformation with MNB (1 Laplace soothing) and SGD Regressor (l2 regularization)
- GloVe embedding with LSTM (100 units, 20% dropout, and Adam optimizer)

b) Topic modeling

Low rating reviews (ratings 1 and 2) from the entire dataset was used to the unsupervised LDA model. After tokenization of the reviews, a dictionary and document term matrix were created to map the words in the correct format for modeling. Clusters or topics were created using the LDA library form the Gensim package. One metric to evaluate the model is coherence score (see result section). The optimal number of topics was found by iterating the model with different number of topics and choosing the number of topics with highest coherence score.

Results Evaluation

For sentiment or star rating classification, the classification report as well as the confusion matrix

were used to evaluate the models. Metrics used for classification evaluation include Accuracy, Precision, Recall, and F1 Score. For star rating prediction using the SGD regressor, the R2 and Mean Square Error (MSE) were used to evaluate the model. For LDA topic modeling, the coherence score was used for evaluation. Please see below for the definitions of these metrics.

Accuracy: accuracy is the most common measure for classification problems. It is the percentage of the outcomes correctly classified by the classifier [9].

$$Accuracy = \frac{\#\ of\ Correct\ Prediction}{Total\ Sample}$$

Precision: precision is the percentage of positive samples correctly classified by the classifier out of all the positive samples classified by the classifier. It measures the exactness of a classifier [9]. It is an important measure when detecting deadly diseases but not as important in this case.

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

Recall: recall is the percentage of positive samples correctly classified by the classifier out of all actual the positive samples [9]. Since it measures the sensitivity of the classifier and captures the class correctly predicted by the classifier, it will be the measure used to evaluate the models in this case.

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

F1 Score: F1 score is the harmonic mean between Precision and Recall. It is used when a balance between Precision and Recall is needed [9]. It is an important measure when there is an uneven class distribution. Since the class imbalance has been addressed by down-sampling previously, it will not be as important of a measure in this case.

$$F1\ Score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

R^2: R^2 or the coefficient of determination is commonly used as an evaluation metric in regression problems. It measures the proportion of the variance in the dependent variable explained by regression model [12]. The value of R^2 is between 0 and 1.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(\mathrm{yi} - \hat{y})^2}{\sum_{i=1}^{n}(\mathrm{yi} - \bar{y})^2}$$

Mean Square Error (MSE): MSE is another commonly used evaluation metric in regression problems. It represents the average of the squared difference between the actual and predicted values [12]. It measures the variance of the residuals and is defined as:

$$MSE = \frac{1}{N}\sum_{i=1}^{n}(\mathrm{yi} - \hat{y})^2$$

Coherence Score: coherence score is a measure to evaluate topics in LDA models. It measures the degree if semantic similarity between high scoring words within the topic. If the words within the topic support each other, the coherence score for the topic is high [13]. It is used to decide the optimum number of topics to be extracted using LDA. The value of coherence score is between 0 and 1 and defined as:

$$Coherence\ Score = \sum_{i<j}(Wi, Wj)$$

Wi and Wj are the top words of the topic.

Results

For sentiment analysis (star rating prediction), there was a general trend for all three classification models. More extreme ratings such as 1-star and 5-star have higher scores than ratings in between. The model that achieved the highest average recall score was GloVe with LSTM (53%), followed by TFIDF with MNB (51%), then BoW with MNB (50%). For GloVe with LSTM, recall reached 61% for 1-star rating and 70% for 5-star rating but hovers around 45% for 2-4-star ratings. Both BoW with MNB and TFIDF with MNB have similar patterns (see figure 6 and 7). Compared to BoW with MNB, TFIDF with MNB was able to achieve 1% higher average recall. Although the overall recall was not very high, when looking at the confusion matrix of these classifiers, there was a distinct trend along the diagonal (see figure 6, 7, and 8). This indicates that many of the predictions were missed by one class from the true label. The average recall was then re-calculated with plus and minus one class for each rating and the resulting average recalls were 82% for BoW with MNB, 82% for TFIDF with MNB, and 86% for GloVe with LSTM.

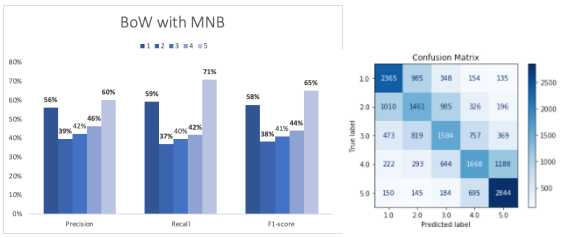

Figure 6: Classification report and confusion matrix of BoW transformation with MNB

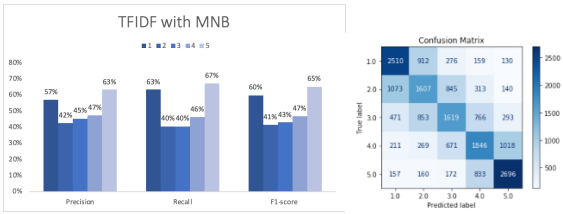

Figure 7: Classification report and confusion matrix of TFIDF transformation with MNB

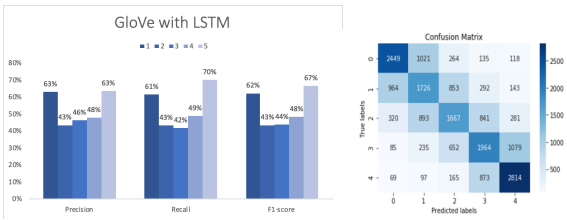

Figure 8: Classification report and confusion matrix of GloVe embedding with LSTM

When training GloVe with LSTM for 80 epochs, the training and validation accuracy increased dramatically from the 13th epoch to the 25th epoch (see figure 9). While the training accuracy continued to increase after the 25th epoch, validation accuracy stayed consistent all the way to the 80th epoch. This indicates that the model has become overfitted after the 25th epoch. The training and validation loss indicate the same pattern since the training loss continued to decrease but the validation loss continued to increase.

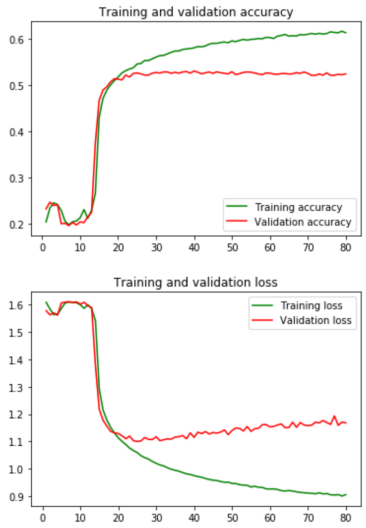

Figure 9: Training and validation accuracy and loss for GloVe embedding with LSTM

On the regression problem, TFIDF with SGD Regressor was able to achieve a better result than BoW with SGD Regressor. TFIDF with SGD Regressor had an R^2 of 0.51 and MSE of 0.99 while BoW with SGD Regressor had an R^2 of 0.37 and MSE of 1.26. As can be observed from the graphs of predicted ratings versus actual ratings for the first 500 testing samples (see figure 10 and 11), BoW with SGD Regressor had some extreme prediction past the rating of 5 while TFIDF with SGD Regressor did not have that many extreme predictions and the predicted ratings were more aligned with the actual ratings. In general, the TFIDF feature extraction method was able to capture more sentiment information than BoW.

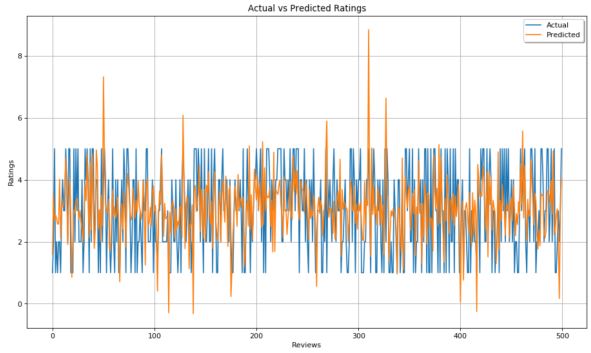

Figure 10: Actual versus predicted regression results of BoW with SGD Regressor

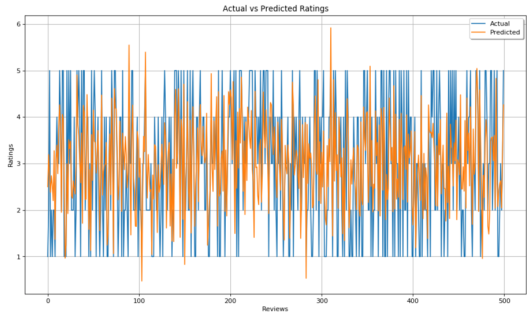

Figure 11: Actual versus predicted regression results of TFIDF with SGD Regressor

For LDA topic modeling, a total of 11 topics achieved the highest coherence score (see figure 12). Therefore, it is the optimal number of topics used to find the topics being discussed in low rating reviews. Instead of returning the actual topic, LDA returned the list of top words for each topic and each word's respective probability within the topic. See below for an example of the first six topics concluded by LDA. Based on the word mix in each topic, the final aspects were defined as follows: 0 - ingredients, 1 - taste and smell, 2 - soup and sauces, 3 - calorie, 4 - price, 5 - delivery, 6 - bland/dry food, 7 - unit of measure, 8 - packaging, 9 - snacks (chocolate/almonds/peanut butter), 10 - green tea. Please see figure 13 for a graphical representation of the topics in two-dimensional space. Although there were some overlaps in topics, 11 topics were still used due to the high coherence score.

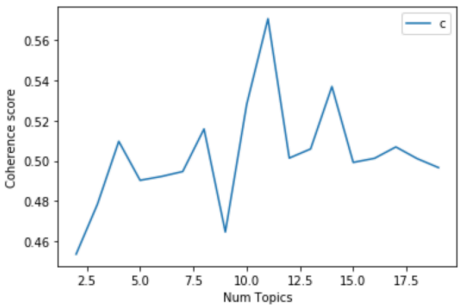

Figure 12: Coherence scores of LDA with different numbers of topics

```
[(0,
  '0.030*"product" + 0.027*"ingredient" + 0.
021*"not" + 0.021*"organic" + 0.018*"food" +
0.015*"label" + 0.012*"list" + 0.008*"health
" + 0.008*"company" + 0.008*"natural"'),
 (1,
  '0.154*"taste" + 0.081*"like" + 0.043*"not
" + 0.030*"bad" + 0.018*"smell" + 0.018*"awf
ul" + 0.016*"terrible" + 0.015*"horrible" +
0.014*"good" + 0.014*"away"'),
 (2,
  '0.022*"like" + 0.018*"not" + 0.017*"eat"
+ 0.013*"sauce" + 0.013*"try" + 0.011*"chees
e" + 0.010*"taste" + 0.010*"food" + 0.009*"s
oup" + 0.008*"chicken"'),
 (3,
  '0.050*"sugar" + 0.044*"oil" + 0.024*"coco
nut" + 0.019*"free" + 0.016*"add" + 0.015*"l
ow" + 0.015*"not" + 0.014*"milk" + 0.012*"fa
t" + 0.012*"calorie"'),
 (4,
  '0.039*"coffee" + 0.036*"not" + 0.017*"pri
ce" + 0.015*"taste" + 0.015*"like" + 0.014*"
try" + 0.013*"good" + 0.011*"bean" + 0.010*"
buy" + 0.010*"well"'),
 (5,
  '0.023*"product" + 0.016*"review" + 0.014*
"not" + 0.014*"no" + 0.013*"time" + 0.013*"o
rder" + 0.012*"date" + 0.009*"day" + 0.009*"
year" + 0.009*"bad"'),
```

Examples of word mix by topic

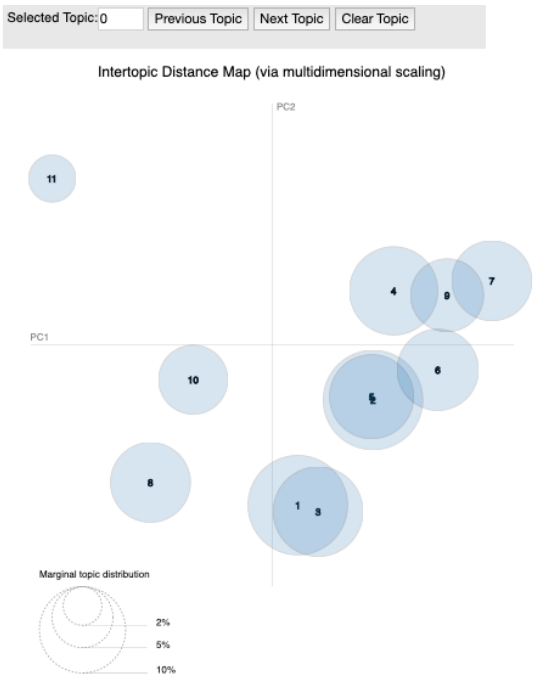

Figure 13: Topic clusters from LDA topic modeling

Conclusion and Future Work

In this project, sentiment analysis was performed to predict the star rating of Amazon reviews using both classification and regression methods and LDA topic modeling technique was used to identify underperforming products or aspects. A large amount of product reviews was used to train the supervised and unsupervised models using three types of feature extraction approaches. The basic theories and evaluation metrics behind the modeling techniques were explained. Based on the experimental results, word embeddings with neural network achieved the

highest performance with extreme rating reviews such as rating 1 and rating 5 having better predictive accuracy than reviews with other ratings. The optimal number of topics to describe the low rating reviews was 11 and certain topics agree with the initial findings from Exploratory Data Analysis. For example, the categories Soups & Stocks and Snack Foods had lower general ratings and they appeared in the low rating topics.

Some future works that can be done to improve the accuracy of the predictive models include removing outlier lengthy reviews and experimenting with different classification models such as Random Forests and Logistics Regression. An application-based product can be built using similar techniques to detect customer sentiments and identify areas of improvement promptly.

References

[1] Chowdhury, Gobinda G. "Natural language processing." *Annual review of information science and technology* 37.1 (2003): 51-89.

[2] Medhat, Walaa, Ahmed Hassan, and Hoda Korashy. "Sentiment analysis algorithms and applications: A survey." *Ain Shams engineering journal* 5.4 (2014): 1093-1113.

[3] M. Hu and B. Liu, "Mining and summarizing customer reviews," in *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04 (ACM, New York, NY, USA, 2004), pp. 168–177.

[4] Mubarok, Mohamad Syahrul, Adiwijaya, and Muhammad Dwi Aldhi. "Aspect-based sentiment analysis to review products using Naïve Bayes." *AIP Conference Proceedings*. Vol. 1867. No. 1. AIP Publishing LLC, 2017.

[5] Srinivasan, Aishwarya V. "Stochastic Gradient Descent - Clearly Explained !!" Medium, Towards Data Science, 7 Sept. 2019, towardsdatascience.com/stochastic-gradient-descent-clearly-explained-53d239905d31.

[6] "Understanding LSTM Networks." Understanding LSTM Networks -- Colah's Blog, colah.github.io/posts/2015-08-Understanding-LSTMs/.

[7] Huang, James, Stephanie Rogers, and Eunkwang Joo. "Improving restaurants by extracting subtopics from yelp reviews." *iConference 2014 (Social Media Expo)* (2014).

[8] Project, UCSD CSE Research. "Amazon Review Data (2018)." Amazon Review Data, nijianmo.github.io/amazon/index.html.

[9] Haque, Tanjim Ul, Nudrat Nawal Saber, and Faisal Muhammad Shah. "Sentiment analysis on large scale Amazon product reviews." 2018 IEEE international conference on innovative research and development (ICIRD). IEEE, 2018.

[10] Mukherjee, Anirban, et al. "Utilization of oversampling for multiclass sentiment analysis on amazon review dataset." 2019 IEEE 10th International Conference on Awareness Science and Technology (iCAST). IEEE, 2019.

[11] Giatsoglou, Maria, et al. "Sentiment analysis leveraging emotions and word embeddings." *Expert Systems with Applications* 69 (2017): 214-224.

[12] Chugh, Akshita. "MAE, MSE, RMSE, Coefficient of Determination, Adjusted R Squared - Which Metric Is Better?" Medium, Analytics Vidhya, 8 Dec. 2020, medium.com/analytics-vidhya/mae-mse-rmse-coefficient-of-determination-adjusted-r-squared-which-metric-is-better-cd0326a5697e.

[13] Topic Coherence To Evaluate Topic Models, qpleple.com/topic-coherence-to-evaluate-topic-models/.