A Comparative Study of NLP Models for Sentiment Analysis on Amazon Pet Product Reviews

Kiyoon Jeong *kj629*
Tejas Shetty *trs389*
Sanjana Jangnure *sbj286*
Young Jun Do *yd482*

1 Introduction

A review of a product helps the customers to understand the quality of a product, and also influences their decision whether to buy the product. In this research, we analyze customer reviews on amazon pet products and find the important factors behind those reviews that are labeled as is. This paper also makes a comparison between several machine learning models for analyzing the sentiment of those customer reviews to the ratings they gave with the reviews. A manual approach on such a large-scale dataset is inefficient and unproductive.

It would be more efficient to utilize machine learning and teach computers to predict ratings from text reviews. With meaningful results, this research could help sellers (companies) to improve their products, and also help customers make a better decision. In this paper, after preprocessing the data by converting to lower case, tokenizing, and removing English stop words, we compare two feature extraction methods count vectorizer and term frequency(TF) and inverse document frequency(IDF) using logistic regression. Then, we analyze the sentiment with the Naive Bayes classifier. Lastly, we compare those results with results generated by Bidirectional Encoder Representation(BERT).

2 Literature Review

Sentiment analysis can be used in various datasets, including social media comments, movie reviews, Stanford sentiment treebank, academic paper review, Amazon product reviews and so on. To search for similar problems, we decided to focus on sentiment analysis on comparably short texts that also contain ratings. Thus, when we focused on solutions people used for amazon product reviews and IMDB movie reviews, there were several models that were used for good performance: Naive Bayes, Logistic Regression, Support Vector Machines, and Deep Learning. In the following, these researchers are reviewed in terms of pre-processing techniques, feature extraction methods, machine learning models, and evaluation metrics.

Sanjay Dey [1] studies a comparative study of support vector machine (SVM) and Naive Bayes classifier for sentiment analysis on amazon product reviews, but they only use TF-IDF for feature extraction, and only use binary classification for the comparison of the two models. Results show that support vector machine can polarize the feedback of Amazon products with a higher accuracy rate.

Sara Ashour Aljuhani [2] studies a comparison of sentiment analysis methods on amazon reviews of mobile phones. They use bag-of-words, TF-IDF, logistic regression, Naive Bayes and convolutional neural network(CNN) and compare the results. Their results show that CNN with the word2vec method achieved the best results with 79.60% accuracy.

Shivaji Alaparthi [4] compares the unsupervised lexicon-based model (Sent WordNet), logistic regression, Long Short-Term Memory(LSTM), and BERT to investigate the relative effectiveness of sentiment analysis techniques. Their results show that BERT achieved the highest accuracy, precision, recall and f1 score.

3 Dataset

The dataset contains the customer review text with accompanying metadata. A link to the dataset can be found at [0] The data is present on AWS S3 in the amazon-reviews-pds bucket in the US East region. Each line in the data files corresponds to an individual review (tab delimited, with no quote and escape characters). The dataset contains samples in English and French. We are working with the English dataset. We also have a parquet version of the dataset that can be used.

The data is a collection of reviews written in the Amazon.com marketplace and associated metadata from 1995 until 2015. This is intended to facilitate study into the properties (and the evolution) of customer reviews potentially including how people evaluate and express their experiences with respect to products at scale. (130M+ customer reviews). We decided

to filter the "pet products" from the whole dataset, to improve processing time. The Pet Products data (compressed version) in itself is around 500 MB in size. Amazon provides a compressed gz file which can be extracted to get the tsv file. Below is a description of the metadata of the dataset

Column name	Description
Marketplace	2 letter country code of the marketplace where the review was written
Customer_id	Random identifier that can be used to aggregate reviews written by a single author
Review_id	The unique ID of the review
Product_id	The unique Product ID the review pertains to
Product_parent	Random identifier that can be used to aggregate reviews for the same product
Product_title	Title of the product
Product_category	Broad product category that can be used to group reviews(also used to group the dataset into coherent parts)
star_rating	The 1-5 star rating of the review
helpful_votes	No of helpful votes
verified_purchase	The review is on a verified purchase
review_headline	Title of the review
review_body	The review text
review_date	The date the review was written

Table 1: Dataset Description

4 NLP Methods Used

4.1 Preprocessing

Before we use this dataset, the portion of 5-star reviews were about 5 times bigger than the other star reviews. In multi-label classification, using the unbalanced data might cause a severe problem, so we pooled 150,000 records from each labels. In our model, the 'document' and 'label' would be used as feature and output column respectively. Among our data columns, 'Review' and 'Review_headline' columns are combined and used as 'document' column, and 'Star_rating' is used as output column.

Also, classifying the document into good or bad reviews could be a good measure to analyze. Thus, we create new dataset that classify 1,2,3 star rating as bad (0) and 4,5 as good (1). We pooled 500,000 records from each label for this binary dataset. The last preprocess step is conversion to lower case.

4.2 Machine Learning Models - Naive Bayes and TF-IDF

First, we divide our works into three part. The first part is building a logistic regression model and Naive Bayes model that could predict the 'Star-Rating' of the review. As we mentioned above, the size of our dataset is very huge, so, we choose a faster tool, Spark, which is a framework based on Hadoop technologies which provides far more flexibility than traditional Hadoop.

Also, we skipped the cross-validation part, which is not needed for the large dataset. We mainly used 'pyspark' library to build these models. Pyspark machine learning library has lots of features, such as regex tokenizer, stop-words removal and count vectorizer. In data pre-processing part, we included two steps, tokenization with regex, and stop-words removal using NLTK library. The spark machine learning pipelines API is similar to Scikit-learn library.

First, we build the count-vectorizer model. In the pipeline, In the pipeline, we included four stages, regex tokenizer, stop words remover, count vector, and label. We set minimum document frequency as 5, and restrict the vocabulary size as 10000 for our count vectorizer. The ratio of our training set and test set is 7:3. For our logistic regression model, we set our regularization parameter to 0.3. The Multi-class Classification Evaluator is used to evaluate our multi-class data and binary data. Also, we calculate the precision, recall, and F1- score using the label and prediction output.

Next, we built the TF-IDF model. In this pipeline, 4 stages are included, regex tokenizer, stop-words removal, hashing TF-IDF, and logistic regression. The other settings are same with tokenizer model. NaiveBayes model is very simple. We used smoothing for our model as per. [5]

4.3 Deep Learning Models -BERT

The second part is building a deep learning model using a BERT. The Spark-NLP library, which is developed by John-SnowLabs, provides lots of deep learning tools including BERT. However, the prerequisites to run this library are too strict and hard to satisfy, hence, we chose to run our model using GPU.

Torch is used for GPU setup, and transformer library is used to pre-train our model using BERT. The cost of building BERT model is very huge that only 25% of our dataset was used. The ratio of our training set and test set is $7:3$. In the training set, we used 80% of data for model training, and 20% for model validation.

Before the training, tokenization is held on each document. The size of review is mostly short, so maximum length is set to 64. We set the number of epoch as 4 because it doesn't affect the model accuracy significantly. Even though we use

25% of dataset, it is still huge amount of data so that 4 epochs is more than enough and use small learning rate (1e-5) to avoid undesirable divergent behavior.

After testing a model, we created predicted label array and get precision, recall, and f-score using scikit-learn in binary case. In multi-class case, we created a table showing accuracy of each labels. The model is largely built on understandings from [6].

4.4 Sentiment Analysis

The last part is extracting top 100 important words that could be used for sentiment analysis. TF-IDF score is used for determining the importance of word. We set the minimum number of document frequency as 100 because TF-IDF tends to give a high score on rare words.

For each label, we get the feature names list of corresponding documents, then transform the list into a single string. By testing this string in the TF-IDF vectorizer model, we could get each word's TF-IDF score.

5 RESULTS

5.1 Binary Classification Results

- We split the data from 5 classes to 2 classes as follows: Bad: Rating 1 to 3, Good: Rating 4 and 5.
- We are using precision, recall and F1-score as evaluation metrics because they give a better understanding when data is not equally distributed.
- Based on the results we can see that BERT performs well even after training it on only 25% of the dataset. Also, all the three metrics precision, recall and F1-score are the same for BERT because the number of False-Positives and number of False-Negatives is the same.

5.2 Results for Binary Classes

Below is a summary of the Precision, Recall and F1-score obtained for binary classification of the reviews.

	CountVectorizer (lr)	TF-IDF (lr)	Naive Bayes	BERT(25% of the dataset
Precision	0.8802	0.8678	0.809	0.9392
Recall	0.8835	0.8725	0.8709	0.9392
F1-Score	0.8818	0.8701	0.8388	0.9392

Table 2: Binary classification results

5.3 Multiclass Classification Results

- In this dataset we have 5 labels each for reviews going from 1 to 5.
- The evaluation metric used here is Accuracy because the dataset was divided into 5 equal classes, each class with 150,000 records. So since the dataset is equally distributed across all the 5 classes, we can use Accuracy as a measure.
- Even with multiclass we observe that BERT performs better than other 3 models.

Below is a summary of the average accuracy metric for multiclass classification.

	CountVectorizer (lr)	TF-IDF (lr)	Naive Bayes	BERT(25% of the dataset
Accuracy	0.579	0.5568	0.526	0.6599

Table 3: Multiclass classification results

5.4 Comparison of Binary and Multiclass results

- It is very evident that BERT performs well in both the cases but we also observe that overall, Binary classifiers give better results because in binary class 2 getting classified as class 1 is no more an error. Similarly, for other classes and hence the performance seems to be better with binary classification.
- Another point to be considered is that customers tend to be in a dilemma when rating a neutral or a slightly bad review and slightly good review. Some customers end up giving rating 4 for the same type of review whereas some customers might give a rating 3. Because of this disparity in the dataset, we can see that Class 2 (57% accuracy), Class 3(60% accuracy), Class 4 (60% accuracy) have lower accuracy than Class 1 (70% accuracy) and Class 5 (80% accuracy). Hence the overall accuracy reduces.
- The BERT Model is 25% of the total dataset so the number of reviews per class are not exactly distributed but overall, the classes have almost the same number of records and the main dataset is perfectly distributed.
- Below, we have mentioned class wise results for the BERT model.

Class	Accuracy
Class 1	3958/5640 = 0.7017
Class 2	3227/5614 = 0.5748
Class 3	3417/5619 = 0.6081
Class 4	3385/5592 = 0.6053
Class 5	4583/5660 = 0.8097

Table 4: Accuracy across classes - BERT

5.5 Important Words for every Class

- Using TF-IDF Vectorizer, we found the top 100 words that are important in every Class of Star Rating. This information helps the sellers know the reasons for every star rating.
- Eventually help them attract the customers based on their needs and improve the quality of products by adding features that occur most frequently in star ratings 1 and 2. Also know the attributes or the specific products that they need to concentrate on to improve the ratings.

5.5.1 Important 100 words in Class 1

```
['glued' 'dimensions' 'crack' 'corners' 'stronger' 'raw' 'passed'
 'willing' 'obvious' 'beef' 'assembled' 'manual' 'video' 'removing'
 'refill' 'measured' 'wild' 'beautiful' 'ad' 'lowest' 'stitching' 'stiff'
 'stain' 'bully' 'harder' '2014' 'blade' 'wrap' 'bills' 'tall' 'kids'
 'reviewer' 'oz' 'print' 'temp' 'arrive' 'cleaner' 'wrapped' 'son'
 'ruined' 'rice' 'clasp' 'posted' 'numerous' 'shop' 'page' 'pig' 'hook'
 'tanks' 'current' 'fat' 'sealed' 'burn' 'lightweight' 'bend' 'durability'
 'follow' 'outdoor' 'kibble' 'string' 'types' 'grain' 'vibrate' 'english'
 'duck' 'tug' 'guinea' 'heart' 'shelf' 'indestructible' 'asking' 'choke'
 'refuse' 'tossed' 'directed' 'advise' 'instantly' 'amazing' 'worry'
 'feature' 'sat' 'applying' 'colors' 'vibration' 'crappy' 'danger' '17'
 'lethargic' 'kitchen' 'covers' 'butter' 'talking' 'refunded' 'terribly'
 'beds' 'smelling' 'duty' 'vets' 'writing' 'recent']
```

Fig. 1: Important 100 words - Class 1

5.5.2 Important 100 words in Class 2

```
['miss' 'inexpensive' 'supplement' 'lick' 'appropriate' 'attachment'
 'mice' 'fetch' 'outer' 'pillow' 'rawhide' 'offer' 'ramp' 'gross'
 'incredibly' 'defeats' 'spaniel' 'measure' 'pushed' 'american' 'tips'
 'thinner' 'butter' 'stretch' 'strange' 'improved' 'tends' '3rd' 'trained'
 'attractive' 'swallow' 'shorter' 'ask' 'shelf' 'continues' 'unlike'
 'grown' 'sewn' 'stainless' 'refuses' 'eh' 'vacuum' 'bummer' 'ticks'
 'reasons' 'sucks' 'dose' 'answer' 'contains' 'killed' 'limited' 'general'
 'adjusted' 'prefers' 'applied' 'wears' 'scratches' 'stable' 'sleeping'
 'phone' 'tighten' 'sale' 'heavier' 'peanut' 'scratched' 'reaction'
 'worthless' 'chose' 'exchange' 'execution' 'pen' 'convenient' 'loop'
 'specifically' 'meat' 'storage' 'following' 'dusty' 'worried' 'boston'
 'waterproof' 'choking' 'required' 'contain' 'belt' 'comfort' 'pug' 'ship'
 'term' 'meal' 'tabs' 'betta' 'cap' 'plush' 'pricey' 'according' 'hazard'
 'dachshund' 'laying' 'charged']
```

Fig. 2: Important 100 words - Class 2

5.5.3 Important 100 words in Class 3

```
['dig' 'reviewer' 'scratched' 'humans' 'fancy' 'screen' 'treatment'
 'everyday' 'suitable' 'reaction' 'flap' 'refused' 'vs' 'offer' 'rock'
 'pro' 'begin' 'seams' 'chunks' 'worn' 'clasp' 'pass' '35' 'interesting'
 'towel' 'hood' 'helping' 'walked' 'squeaky' 'grab' 'discovered' 'funny'
 'carrying' 'chose' 'cap' 'challenge' 'shame' 'perch' 'closer' 'faster'
 'english' 'elastic' 'turning' 'plant' 'knowing' 'plush' 'jumping'
 'preferred' 'hardly' 'smallest' 'ship' 'rarely' 'source' 'scratches'
 'types' 'states' 'maltese' 'waterproof' 'trash' 'miracle' 'sweater'
 'owned' 'beginning' 'tired' 'orange' 'width' 'directly' 'wife' 'crack'
 'lift' 'complaints' 'comfortably' 'neighbors' 'seam' 'mice' 'general'
 'sticky' 'list' 'shedding' 'section' 'bunny' 'boston' 'haired' 'release'
 'squeeze' 'eater' 'feeling' 'shouldn' 'wild' 'sell' 'setup' 'walls'
 'prime' 'according' 'miss' 'scratcher' 'vacuum' '70' 'paint' 'improved']
```

Fig. 3: Important 100 words - Class 3

5.5.4 Important 100 words in Class 4

```
['adjustment' 'tennis' 'duck' 'plush' 'accidentally' 'application'
 'humans' 'sand' 'posts' 'proper' 'fleece' 'reviewer' 'noticeable' 'potty'
 'schnauzer' 'accident' 'boston' 'list' 'liner' 'miniature' 'sale' 'trust'
 'powerful' 'traveling' 'installation' 'rub' 'pair' 'dose' 'squirrels'
 'buckle' 'pond' 'squeeze' 'walls' 'scratcher' 'sun' 'ways' 'odd' 'dig'
 'younger' 'awkward' 'mistake' 'shiny' 'odors' 'thinks' 'upset' 'section'
 'approved' 'died' 'tracking' 'cracked' 'preferred' 'removable' 'lack'
 'tightly' 'ride' 'forget' 'shelf' 'miracle' 'filling' 'joint' 'ahead'
 'washable' 'dropped' 'wont' 'pockets' 'bored' 'bunny' 'pictured' 'prices'
 'challenge' 'world' 'strips' 'nylon' 'age' 'mention' 'personally'
 'flexible' 'feline' 'pillow' 'cap' 'ask' 'yr' 'reduce' 'program' 'bunch'
 'peanut' 'breeds' 'total' '45' 'tick' 'algae' 'carefully' 'liquid'
 'success' 'listed' 'vs' 'trash' 'dead' 'road' 'crack']
```

Fig. 4: Important 100 words - Class 4

5.5.5 Important 100 words in Class 5

```
['prefers' 'throwing' 'beautifully' 'boys' 'returned' 'assembly' 'bully'
 'fix' 'pocket' 'played' 'seriously' 'slowly' 'dinner' 'infection'
 'sooner' 'undercoat' 'stuffed' 'china' 'track' 'rawhide' 'trainer' 'ride'
 'upset' 'algae' 'rinse' 'yummy' 'climb' 'walked' 'website' 'nature'
 'proof' 'bits' '75' 'humans' 'miniature' 'required' 'happened' 'closed'
 'reviewers' 'avoid' 'fold' 'steel' 'mouse' 'video' 'somewhat' 'levels'
 'tries' 'grocery' 'usual' 'cups' 'betta' 'belt' 'plant' 'straight'
 'annoying' 'complete' 'pig' 'professional' 'younger' 'stuffing' 'breeze'
 'appreciate' 'led' 'trash' 'picked' 'accurate' 'maintain' 'toss' 'scared'
 'sell' 'round' 'pop' 'joints' 'relief' 'household' 'tends' 'strap'
 'frequently' 'perch' 'protect' 'squeak' 'cuts' 'war' 'decent' 'carries'
 'blade' 'sprayed' 'fabulous' 'directly' 'sister' 'setup' 'breaking'
 'dane' 'surprise' 'peace' 'road' 'rope' 'promptly' 'hanging' 'apply']
```

Fig. 5: Important 100 words - Class 5

6 Conclusion

- Binary dataset performed better than multiclass dataset.This conclusion is as expected as multiclass problems have a greater chance of giving the wrong results. Also, in our case, the ratings 2,3 or 3,4 are tough to decipher even from a human perspective. The model thus, also gives lower accuracy for these rating categories.
- BERT Performed better that all other methods even with 25% of dataset. BERT being a more advanced model when compared to TF-IDF and Naive Bayes makes better classification decisions and thus gives better metrics overall.
- One of the applications of our model can be to label unlabeled reviews with our trained model. This will help in saving a shoppers decision making time. Instead of going through the entire review, they can get a gist from the generated rating.
- Sellers can use important words to improve their products, services and eventually sales. For example, if we see important words in class 5, we see that they are related to kitchen related products. Thus, such products for pets have a higher rating and are more likely to be bought by the customer.

References

[1] S. Dey, S. Wasif, D. S. Tonmoy, S. Sultana, J. Sarkar and M. Dey, "A Comparative Study of Support Vector Machine and Naive Bayes Classifier for Sentiment Analysis on Amazon Product Reviews," 2020 International Conference on Contemporary Computing and Applications (IC3A), Lucknow, India, 2020, pp. 217-220, doi: 10.1109/IC3A48958.2020.233300. (https://ieeexplore.ieee.org/abstract/document/9076924?casa_token=f1aRETYXpvIAAAAA:l5cO4U75McaLBwrjqcJwobPx3W8V768X7_ndtOZnS163D0j5v
WRZ1b)

[2] S. A. Aljuhani and N. S. Alghamdi, "A comparison of sentiment analysis methods on Amazon reviews of Mobile Phones," Int. J. Adv. Comput. Sci. Appl., vol. 10, no. 6, pp. 608–617, 2019, doi: 10.14569/ijacsa.2019.0100678 (https://thesai.org/Publications/ViewPaper?Volume=10Issue=6Code=IJACSASerialNo=78)

[3] Trivedi, K. (2019), "Multi-label text classification using BERT – The mighty transformer", available at: https://medium.com/huggingface/multi-label-text-classification-using-bert-themighty-transformer-69714fa3fb3d

[4] S. Alaparthi, M. Mishra, "Bidirectional Encoder Representations from Transformers (BERT): A sentiment analysis odyssey," Cornell University, arXiv: 2007.01127 (https://arxiv.org/abs/2007.01127)

[5] Susan Li, "Multi-Class Text Classification with PySpark", Towards Data Science, 2018.02.19 (https://towardsdatascience.com/multi-class-text-classification-with-pyspark-7d78d022ed35)

[6] Susan Li, "Multi Class Text Classification With Deep Learning Using BERT", Towards Data Science, 2020.08.02 (https://towardsdatascience.com/multi-class-text-classification-with-deep-learning-using-bert-b59ca2f5c613)

Amazon Pet Products Reviews Dataset (https://s3.amazonaws.com/amazon-reviews-pds/tsv/index.txt)