Natural Language Processing Final Project

Amazon Customer Reviews on Electronic Products

Jenny Tsai (G49487749)

Fall 2022

I. Introduction

Amazon has been one of the biggest players in the e-commerce market. According to Amazon's latest quarterly revenue report in 2021, the online stores alone has generated $66.08 billion, contributing to Amazon's biggest-ever Black Friday and Cyber Monday driving growth for Q4 (Insider Intelligence, 2022).

As customer reviews and ratings on the online products have proven to have a great impact on product sales (Spiegel Research Center, 2017), this project aims to explore and analyze the customer reviews on Amazon electronic products and *find the words/Natural Language Processing (NLP) models that can best predict product recommendation and ratings*. The results of the project are expected to be utilized in several domains, such as product improvement (by comparing the key words in positive reviews vs. negative reviews), marketing (craft the advertisement using the most impactful words in positive reviews), and product outlook (by predicting ratings based on a small sample of existing reviews).

II. Description of the Dataset

This project used the Amazon electronic product customer reviews dataset downloaded from Kaggle.com. The dataset including reviews from 2014 to 2018, with a total of 5,000 reviews and 24 features including product brand, category, rating, recommendation, etc. In this project, the primary feature used was the text from customer reviews, and the target variables were (1) recommendation (binary, whether or not the customer recommends the product) and (2) review ratings (multiclass, 1 to 5). The reason why I chose these two variables as target variables was that customer would recommend the product without giving a top rating, so it would be interesting to look at both variables to see if they yield different results.

III. NLP Models and Algorithms

This project used both classical and modern approach to solving the NLP problem stated earlier in the Introduction section. First, some rule-based analyses, such as frequency and Word Cloud were used to visualize the review distribution on the target variables. Next, a logistic regression was run using the odds of the reviews to predict recommendation. The logistic model is a statistical model that models the probability of an event taking place by having the log-odds for the event be a linear combination of one or more independent variables (Wikipedia). To make results more interpretable, SHAP was applied to the logistic model to find the most important features on recommendation. SHAP is a mathematical method based on the concepts of game theory to explain the predictions of machine learning models.

It calculates the contribution of each feature to the prediction and thus can determine the most important features and their influence on the model prediction (Bagheri, 2022; please refer to this article for more mathematical details behind SHAP).

To compare with results from the classical approach, two neural network models were implemented - Multi-Layer Perceptron (MLP) and Long Short-Term Memory (LSTM). MLP is a basic neural network which consists of at least three layers: an input layer, a hidden layer, and an output layer. Each neuron, except for the ones in the input layer, is transformed by a nonlinear activation function then uses backpropagation for training. It can distinguish data that is not linearly separable.

On the other hand, LSTM is a Recurrent Neural Network (RNN) that can not only process single input, but also sequence data (such as text, time-series data). Its recurrent nature (feeding previous outputs to the next step) connects weights and biases in the network change once per episode of training, thus provides both "long-term" and "short-term" memory for the model.

The algorisms were developed using the python packages including sklearn (logistic regression), wordcloud, SHAP, and PyTorch (MLP and LSTM). Specifically, the algorisms for logistic regression, wordcloud, and SHAP were developed based on their respective official website, while the algorithms for MLP and LSTM were modified from the lecture codes by Professor Amir Jafari.

IV. Experimental Setup & Hyper-Parameter

After data pre-processing (e.g., lower-case, remove punctuation and stop words, etc.), the dataset was split into train set and test set in a ratio of 70/30. There are a couple of things worth noting in pre-processing. First, the text was not lemmatized because the tense of each word could imply specific sentiment towards the product. For example, past tense or subjunctive mood imply regrets or negative sentiment (e.g., I wish I had never bought this product). Second, besides the regular stop words from nltk package, a custom set of stop words that contains product names (e.g., Alexa, kindle, etc) were removed from the original text.

To evaluate the performance of each model, accuracy score was chosen as the primary criterion with f1-score as a supplementary criterion. For the neural network models (MLP and LSTM), pretrained embeddings (gloVe) and model (transformer) were used to save training time and hopefully can help to boost the overall model performance. For the LSTM model, a linear learning rate scheduler was used to decay the learning rate by linearly changing small multiplicative factor until the number of epoch reaches total number of iterations. To avoid overfitting, the data was pre-processed (clean) and more train data was used (70%) to better detect the relationship between features and outcome variables. Also, dropout layers were added to the MLP model.

V. Results

i. Frequency and Word Cloud

As mentioned earlier, the first part of the analyses used the classical approach. By running frequency on the two target variables (see Figure 1 and Figure 2), we can see that most of the reviews were positive (i.e., over 93% of the reviews has a rating of 4 or 5, and over 95% of the reviews recommends the product). Therefore, the data has imbalanced class issue that can potentially decrease model performance.

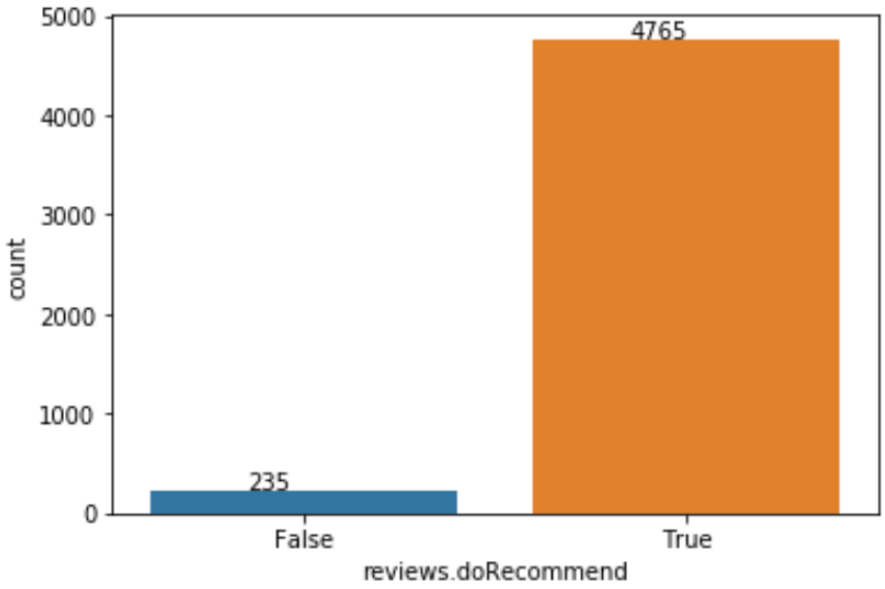

Figure 1. Number of reviews by recommendation.

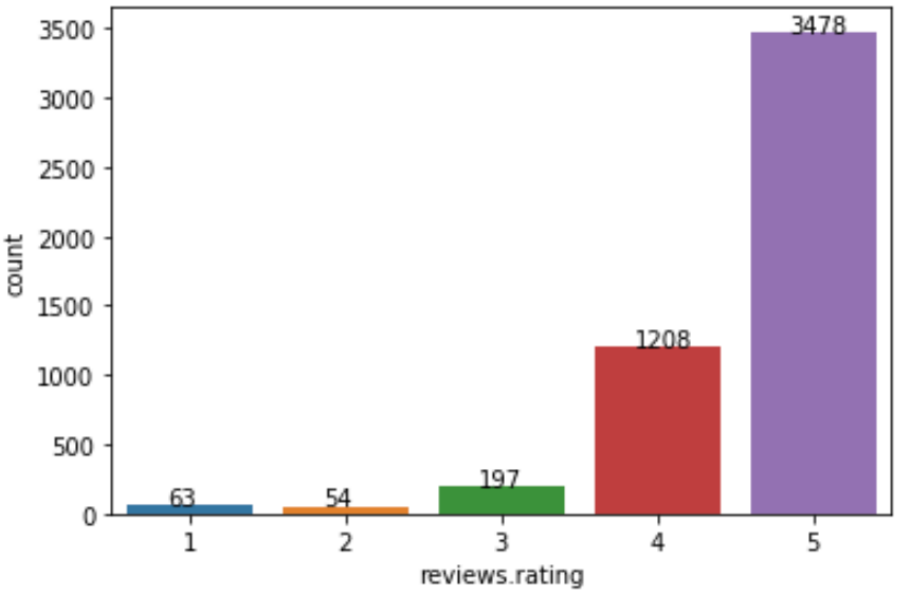

Figure 2. Number of reviews by rating.

Next, reviews with a rating of 4 or 5 were categorized as positive, while reviews with a rating of 1 or 2 were categorized as negative. In the Word Cloud for the positive reviews (see Figure 3), the most frequent words stood out including ‘great’, ‘love’, ‘easy (to) use’. Also, ‘price’ and ‘’works’ might imply the products were offered at a reasonable price and worked in the way as the customers expected.

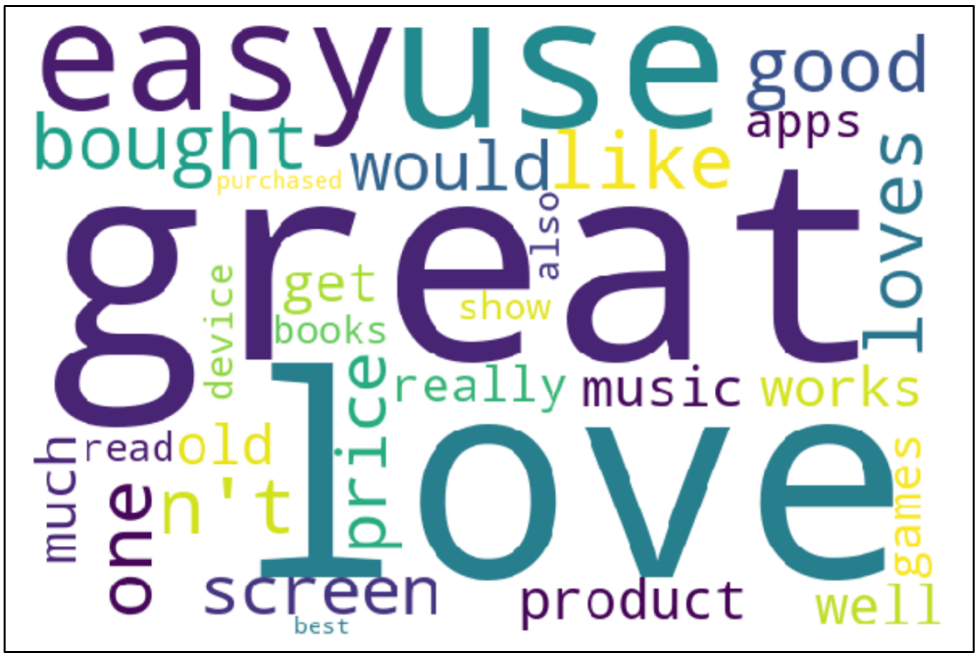

Figure 3. Word cloud for positive reviews (rating 4 or 5).

On the other hand, the word cloud on negative reviews seemed to show more words with negation or regrets, such as “n’t (not)”, “would”, “could”, “returned”, ”junk”. It’s also likely that customers complained about the functionality and durability when they mentioned “use”, “(not) work”, and “last”. Specifically, they might not be satisfied with the “screen”, “charge”, and “apps” aspect of the product.

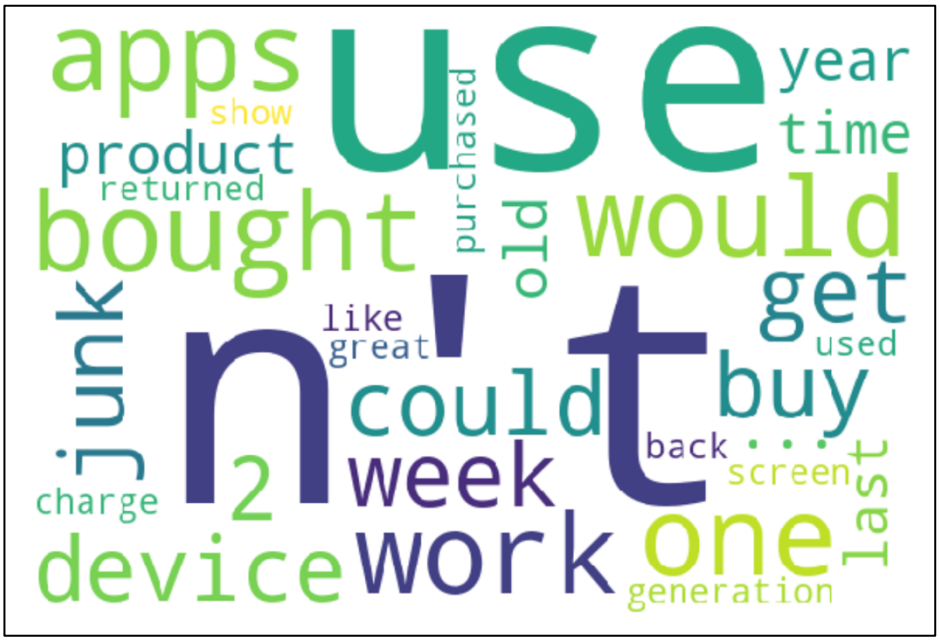

Figure 4. Word cloud for the negative reviews (rating of 1 or 2).

ii. Logistic Regression with SHAP

The next step was to run a logistic regression on the binary target variable – recommendation, where 1 is Recommend and 0 is Not Recommend. By looking at the performance scores and the confusion matrix from the logistic regression (see Table 1 and Table 2), we can see that although the accuracy score appeared to be high (0.96), the f-1 score and recall for label 0 (Not Recommend) are zero, meaning that the model cannot distinguish well between true negative and false negative (0 true negative), which is likely due to the imbalanced class of the outcome variable (i.e., most reviews are positive).

Results from Logistic Regression:

	precision	recall	f1-score	support
0	1.00	0.00	0.00	61
1	0.96	1.00	0.98	1439
accuracy			0.96	1500
macro avg	0.98	0.50	0.49	1500
weighted avg	0.96	0.96	0.94	1500

Table 1. Results from Logistic regression on recommendation.

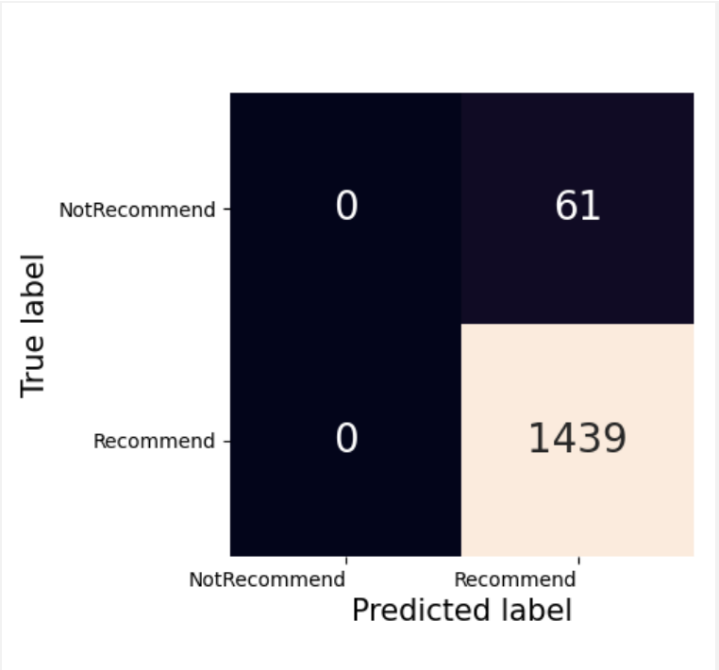

Table 2. Confusion matrix from logistic regression model.

Although logistic model didn't perform well, it's still worth taking a look at the SHAP results from the model, which is shown in Figure 5. The most important features/words with positive sentiment are mostly consistent with the results from the frequency analysis (Word Cloud), such as 'love', 'great', 'easy'. However, SHAP seemed to find some different important words with negative sentiment, such as 'ca(n't)' and 'time', which possibly imply that the product didn't work in such a way that meet the customers' expectations and appeared to waste their time.

Since the results from logistic regression appeared to be highly affected by the imbalance class nature of the dataset (and addressing imbalance class issue is not the focus of this project), let's move on to the results from the modern approach – neural networks to see it can outperform the traditional model.

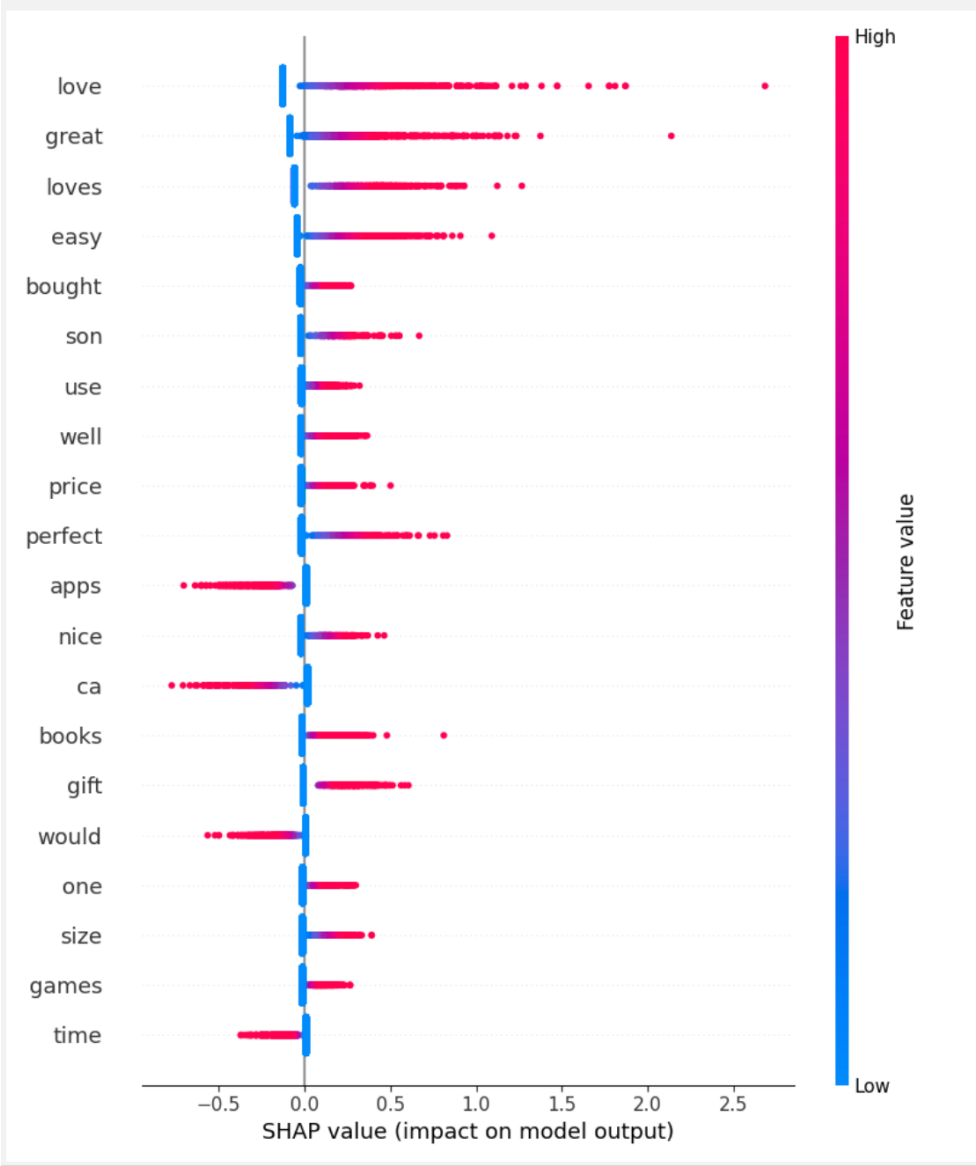

Figure 5. Importance of feature by SHAP value.

iii. MLP Model

As mentioned earlier, pretrained embeddings – 50 dimension gloVe - were used for the MLP model in this project, and these embeddings were frozen for training. The neural network design was adopted from Professor's lecture codes, which consists of four layers: linear, ReLu activation function, batch normalization, and finally a dropout layer. Once again, the model was used to predict the binary target – recommendation.

After training with a learning rate of 1e-2, loss function being cross-entropy, and Adam being the optimizer, the results from the MLP model applied on the test set has an accuracy score of 95.93 (see Figure 6). Compared with the logistic model, MLP has a similarly high accuracy score and seemed to do a better job at predicting true negative (6 correct true negative, see confusion matrix in Figure 6).

```
Epoch 0 | Train Loss 0.00948, Train Acc 79.86 - Test Loss 0.67656, Test Acc 82.93
The model has been saved!
Epoch 1 | Train Loss 0.00716, Train Acc 95.31 - Test Loss 0.40942, Test Acc 95.67
The model has been saved!
Epoch 2 | Train Loss 0.00438, Train Acc 95.66 - Test Loss 0.20036, Test Acc 95.67
Epoch 3 | Train Loss 0.00256, Train Acc 95.77 - Test Loss 0.16527, Test Acc 95.67
Epoch 4 | Train Loss 0.00184, Train Acc 96.09 - Test Loss 0.18404, Test Acc 95.73
The model has been saved!
Epoch 5 | Train Loss 0.00147, Train Acc 96.14 - Test Loss 0.21150, Test Acc 95.80
The model has been saved!
Epoch 6 | Train Loss 0.00106, Train Acc 96.11 - Test Loss 0.22905, Test Acc 95.80
Epoch 7 | Train Loss 0.00088, Train Acc 96.29 - Test Loss 0.22939, Test Acc 95.80
Epoch 8 | Train Loss 0.00055, Train Acc 96.54 - Test Loss 0.23452, Test Acc 95.93
The model has been saved!
Epoch 9 | Train Loss 0.00037, Train Acc 97.09 - Test Loss 0.23851, Test Acc 95.87
The accuracy on the test set is 95.93
The confusion matrix is
[[   6   61]
 [   0 1433]]
```

Figure 6. MLP model training and performance from test set.

iv. LSTM Model with Transformer

The final model used transformer (BERT: bert-base-uncased) as the base model with a head layer of LSTM. Unlike the previous models using a binary target (recommendation), the target variable is review ratings, which is a multiclass variable ranging from 1 to 5. Therefore, the purpose of including this model was not to do an apple-to-apple comparison with the previous models. It's more like an attempt to try out a more powerful model to see how it performs when dealing with a more advanced classification problem (i.e., multiclass) even with a small sample. With this rationale in mind, the LSTM model design is of the simplest form, with only an input layer, a hidden layer, and a linear activation function. With a small learning rate of 5e-5 with linear scheduler, optimizer being Adam, loss function being cross-entropy, only 200 reviews were entered in the model. The results from the LSTM transformer-based model turned out not to be too bad with such a small sample size – the best accuracy score from the test set was 0.72 (see Figure 7).

Figure 7. Results from LSTM with transformer (N = 200).

VI. Summary and Conclusion

By exploring and analyzing the Amazon customer reviews on electronic products using the NLP classical approach, the project was able to find the most impactful words from positive and negative reviews, from which product insights were distilled. For example, factors that could impact customer recommendation are price, easiness to use, durability, and whether it works as expected. First three findings seemed to be straightforward for pricing and marketing experts, but the last finding (works as expected) could have some interesting implications. For instance, instead of presenting only the "bright side" of the product, providing a faithful product description that includes both pros and cons might have a better chance to win customer's heart.

Although the logistic regression model on predicting recommendation appeared to have a high accuracy score of 0.96 and didn't seem to perform well on distinguishing TP and TN due to imbalanced class issue, the SHAP results from the model were mostly consistent with the frequency analysis/word cloud. It's interesting to see how a classical approach (frequency) vs. a statistical approach (logit model) came to a similar conclusion like cross-validation.

With the modern approach using neural network, MLP has a similarly high accuracy score (~0.96) in predicting recommendation as the logistic regression model, and seemed to do a better job at classifying TP and TN. Finally, the transformer-based simple LSTM model using a small sample size did an okay job in predicting multiclass target (review ratings). Even though a good model without

enough data is useless (Jafari, 2022), in some pioneering stage such as launching a new product with only a few reviews, it is still insightful to project the product outlook with a okay model of some average performance score. In general, it looks like neural networks are less affected by the traditional imbalanced class issue. It is expected that the model performance can increase dramatically with a larger sample size and with a more sophisticated neural network design (e.g., adding more layers, finetuning the hyper-parameters).

I personally think neural network is “now and the future” in the NLP domain, however, the classical approach is still valuable in such a way that it can quickly provide intuitive insights that are consistent with that from the basic statistical models. Its interpretability is one of the greatest advantages, which is crucial when selling the ideas to non-technical audience.

This project is more like an EDA with basic model design for customer reviews classification-type-of problem. In the future, data scientists interested in this topic can not only can further refine the neural network model design to boost model performance, such as using a more sophisticated design, or include more features (e.g., whether it’s a verified purchase or a fake review). Additionally, it is recommended to explore the application of transformers in this area. For example, the transformer can provide the summary of top reviews vs. worst reviews so that the customers don’t have to go through each review to gain insights from the product. Another example of application is the Q&A section of the product. Instead of having real people answer the questions manually, transformer can use customer

reviews as context and pull out answers directly in response to the question. After all, pretrained neural networks can be very powerful as long as the users/data scientists know how to apply them properly.

References

Insider Intelligence (2022). *Amazon annual revenue breakdown by segment in 2022*. Retrieved from: https://www.insiderintelligence.com/insights/amazon-revenue/

Spiegel Research Center (2017). *How online reviews influence sales.* Retrieved from: https://spiegel.medill.northwestern.edu/wp-content/uploads/sites/2/2021/04/Spiegel_Online-Review_eBook_Jun2017_FINAL.pdf

Bagheri, Reza (2022). *Introduction to SHAP and their application in machine learning*. Retrieved from: https://towardsdatascience.com/introduction-to-shap-values-and-their-application-in-machine-learning-8003718e6827

Wikipedia. *Logistic regression.* Retrieved from: https://en.wikipedia.org/wiki/Logistic_regression

Jafari, Amir (2022). *Lectures in Natural Language Processing in Data Science, George Washington University.*