Amazon Pet Supplies Product Reviews Sentiment Analysis using Logistic and distilBERT models

Final Report

Amanda (Man Kuei) Chen
MS in Business Analytics

DATS 6312_11: Natural Language Processing
Professor: Amir Jafari, PhD

Introduction

As a business person with a marketing background, I would like to explore the usage of NLP techniques to address common marketing problems – customer service team spending too much time on reading each products review might delay the messages that could be more urgent, leading to more complains. Therefore, to mitigate this problem, I built a sentiment classifier to help customer service team prioritize their tasks by categorizing the sentiment of the incoming product reviews. This sentiment classifier can detect whether the reviews are positive or negative. I used a pre-trained model to fine tune the amazon pets product reviews dataset. Therefore, this model might be ideally for retailers in pet supplies industry.

This project encompasses two parts – Part1. Exploratory Data Analysis, which helps us know more about this dataset and gain insights about what customers talk about in their reviews; Part2. Sentiment Analysis using a baseline model and a fine tune pre trained model.

Data understanding

Data: Amazon pets product reviews
Size: 1.2GB
Year: 2013-10 ~ 2018-10
Records: 5,816,874 rows and 3 columns
Source: https://cseweb.ucsd.edu/~jmcauley/pdfs/emnlp19a.pdf

	overall	reviewText	year_month
0	1.0	I was not happy with product would like to ret...	2016-12
1	1.0	This cd is scratched and it constantly skips. ...	2016-12
2	4.0	It works just fine and repeats when I'm not he...	2016-12

Data preparation

For Part 1, firstly, I imported libraries and loaded the amazon pets review dataset, and then performed text preprocessing for feature extraction by using NLTK packages to remove special characters, stopwords and lemmatize. For sentiment analysis, it's supposed to be careful when removing the stopwords because if you remove negative words like "NOT" and then train the model, then the model would be very bad. As a result, it's better to customize a stopword list. However, I was doing feature extraction, so in this case I ignored this step. I also ignored the contradiction expansion for the same reason. Then, I took 10 percent of the original dataset to do exploratory data analysis. I generated top 10 most common words for each class (Fig1&Fig2) and bigram for each class (Fig3&Fig4). I tried tri-gram but it doesn't return any useful outcomes. According to these results, we can see from bigram, "year old" is the 6th on the top 10 positive word list. It provides the idea that sellers might want to categorize their products by age since their customers like to talk about these two words and it's reasonable to think if the products are

categorized by age, it would be easier for customers to navigate specific products. I would like to note that in the Figure 4, the results returning positive bigrams like" would recommend" and " work well" are supposed to be negative bigrams, but because I removed the stopwords based on the word list from NLTK corpus, it returned otherwise.

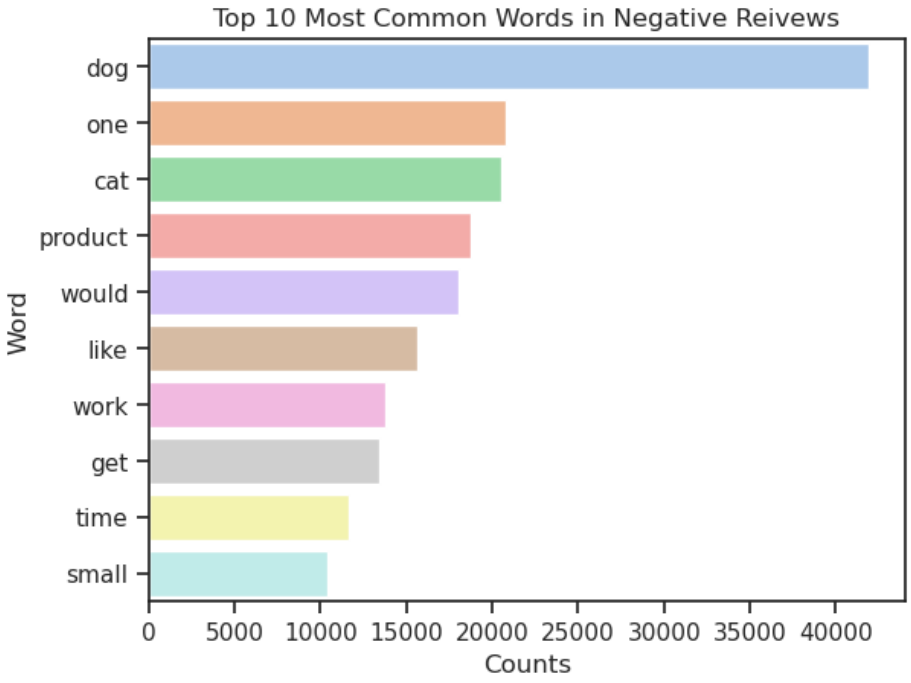

Fig1

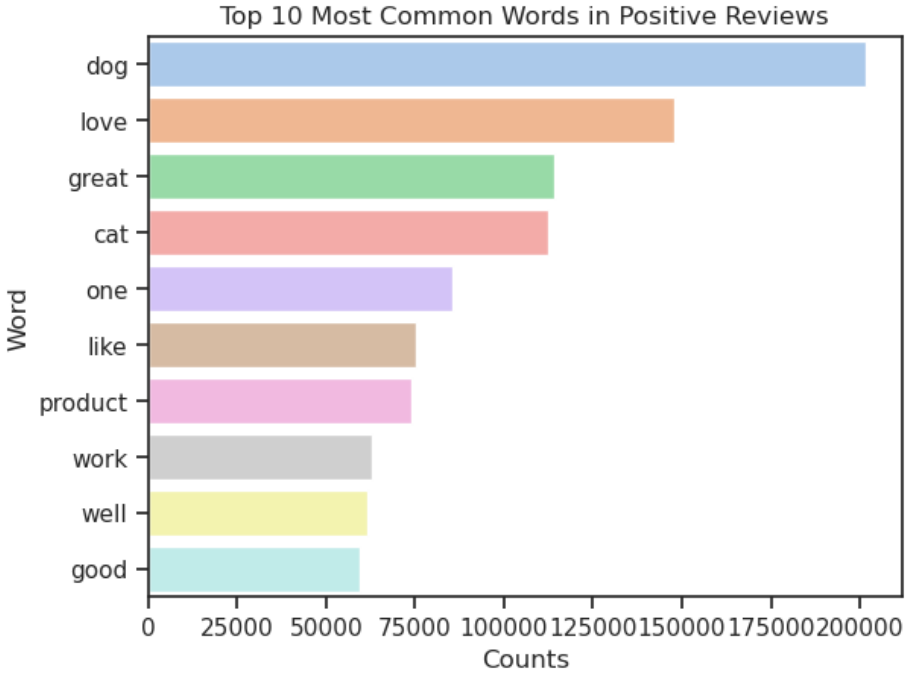

Fig2

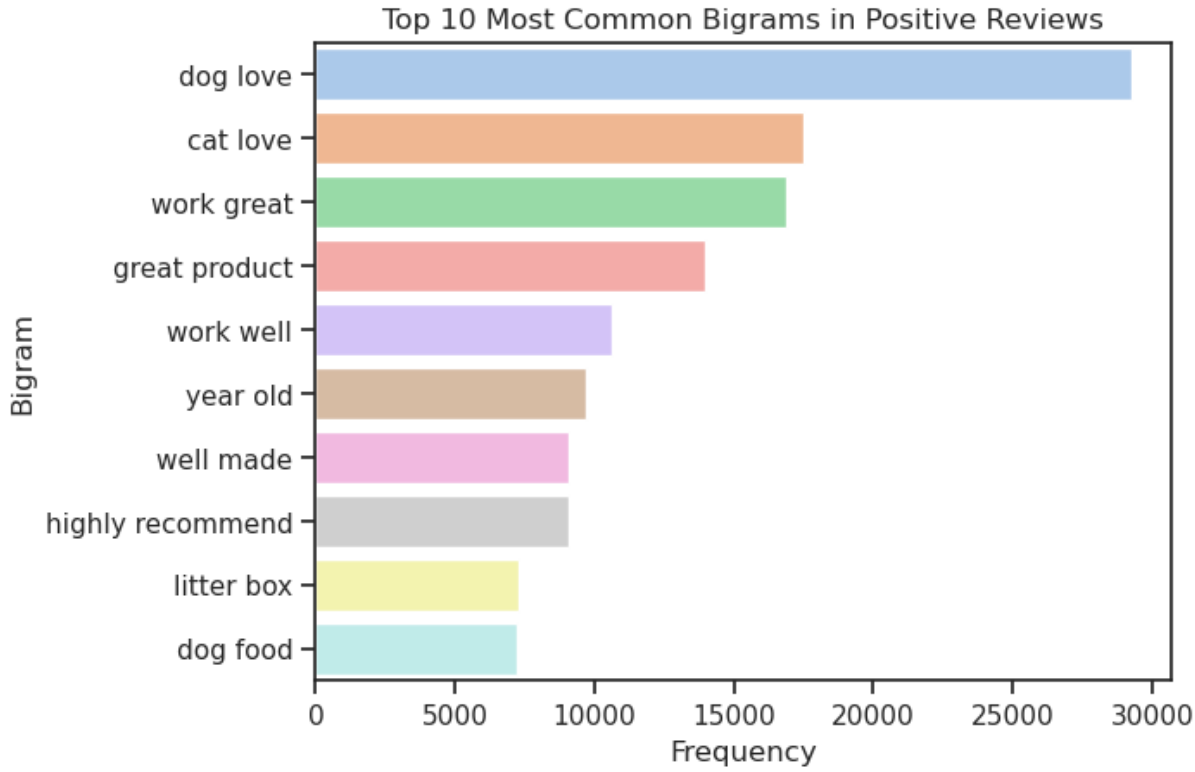
Top 10 Most Common Bigrams in Positive Reviews
dog love
cat love
work great
great product
work well
year old
well made
highly recommend
litter box
dog food
Bigram
0
5000
10000
15000
20000
25000
30000
Frequency

Fig3

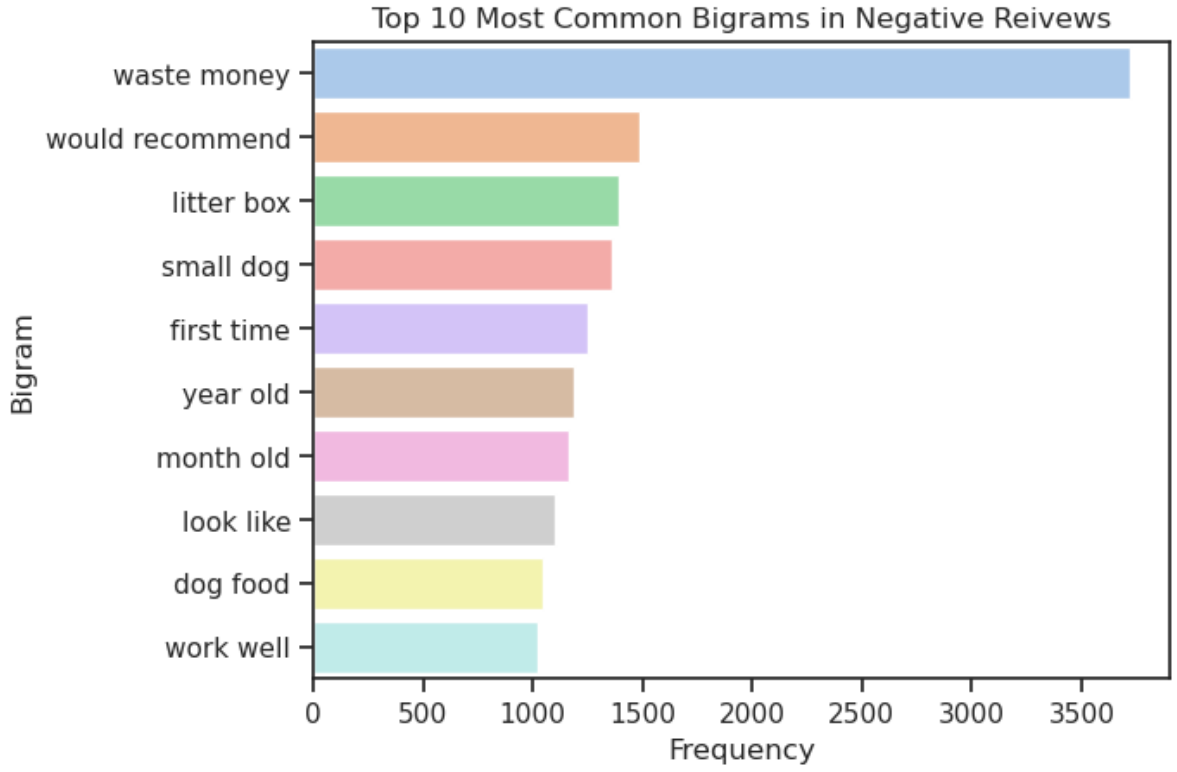
Top 10 Most Common Bigrams in Negative Reivews
waste money
would recommend
litter box
small dog
first time
year old
month old
look like
dog food
work well
Bigram
0
500
1000
1500
2000
2500
3000
3500
Frequency

Fig4

Moreover, line charts (Fig5~Fig9) give some flavors of seasonality. For example, whether the number of sentiment reviews is correlated to holiday seasons. From year 2014 to 2017, the number of positive reviews increased in every February, every June and every November except for 2015 while it decreased in every March and August. It might just because of the peak season and off-peak seasons. However, it's worth to make a strategic plan for these particular months. It's impressive that the number of negative reviews was relatively low consistently during these five years.

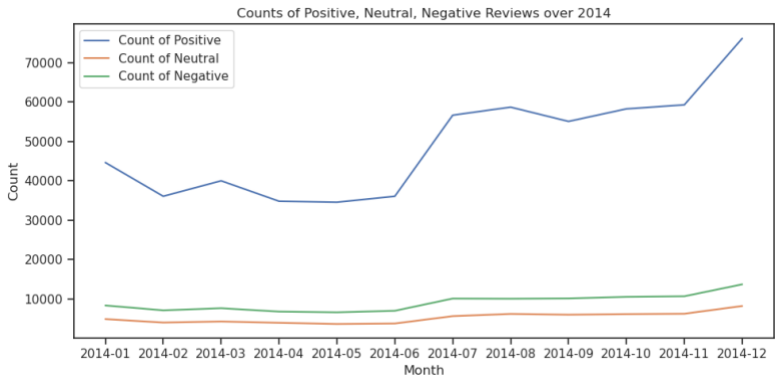

Fig5

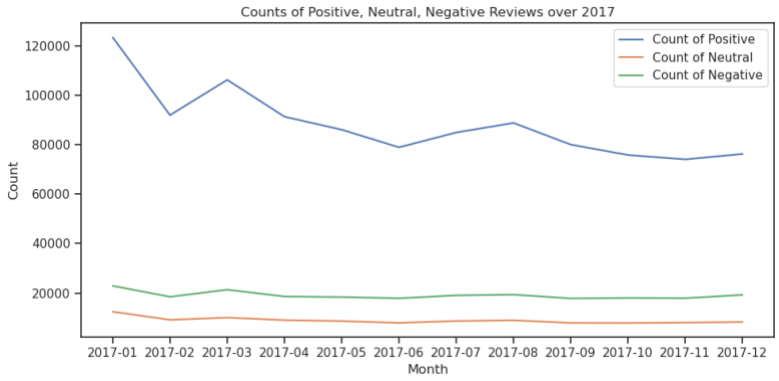

Fig6

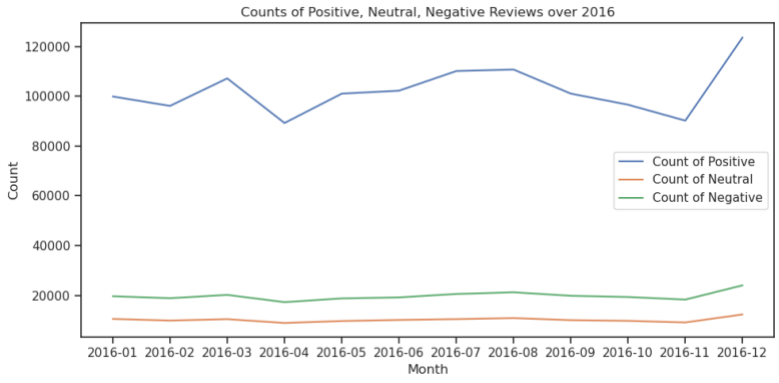

Fig7

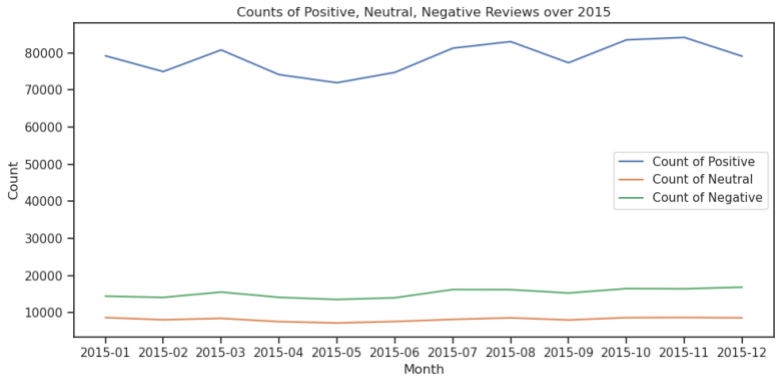

Fig8

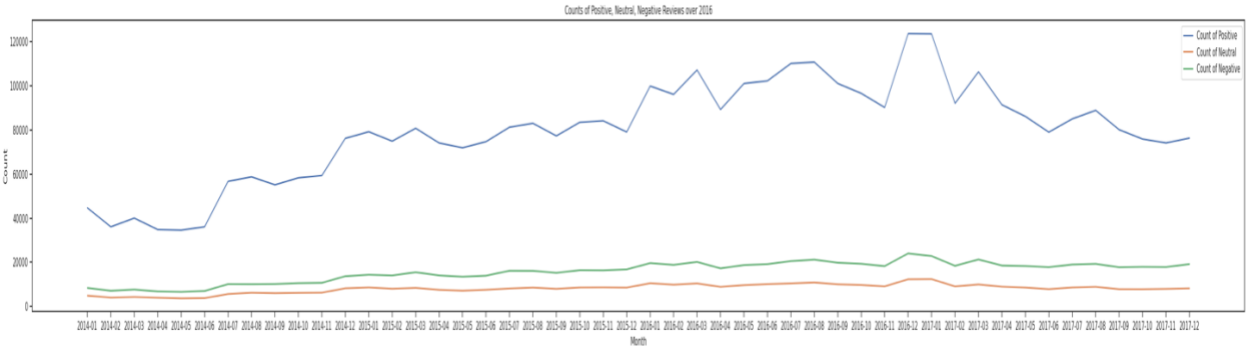

Fig9

Further, W2V is a static embedding technique. For example, unlike the technique that is used in a distilBERT model, it provides a functionality that tell you which are similar words with a particular word without contextual aware. I tested with the word" husky", the outcomes returned other dog breeds. To a limited degree, it might be able to provide insights for SEO on the product contents.

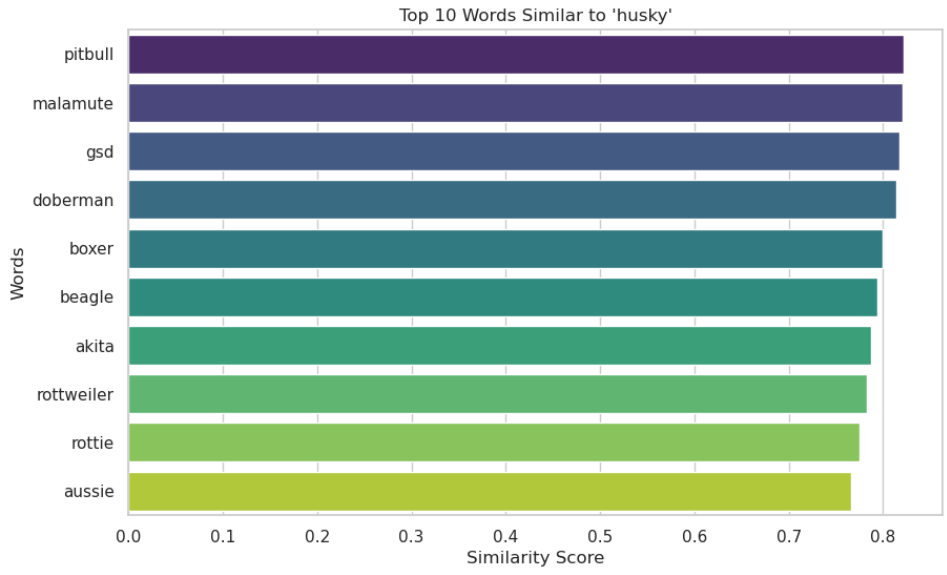

Fig10

Modeling

For Part 2, I used the raw text data without preprocessing, and classified the target column to positive and negative classes by assigned rating score above 3 as positive class and rating score below 3 as negative class. At the same time, I dropped the null values and all the rating score equal to 3, which are 456,403 records and is around 7% of the original dataset, and I wouldn't discuss the sentiment of neutral in this project. Then, I embedded texts with TF-IDF technique and trained with the logistic regression model as my baseline model because it's a simple and interpretable and serves as a good benchmark for classification problems. After that, I prefer to use DistilBERT to fine-tune my dataset because it's compact, easy to use, and has 95% of BERT's performance at the same time.

Logistic Regression Model	DistlBERT
1. The output can be interpreted as probabilities, making it easier to understand the impact of each variable on the outcome. 2. Lower risk of overfitting compared to more complex models.	DistilBERT is a small, fast, cheap and light Transformer model trained by distilling BERT base. It has 40% less parameters than bert-base-uncased, runs 60% faster while preserving over 95% of BERT's performances as measured on the GLUE language understanding benchmark.

When I trained the logistic model, I found that the F1 score for negative class is very low compared to the positive class. As a result, in order to increase the accuracy of the negative class (table1), I kept all negative reviews and randomly selected the same number of reviews from the positive class to make both classes even. I lost around 3.2M records, from 5M to 1.8M, but I have a better model with 0.92 accuracy. Besides, 1.8M records are not too bad.

	F1-Score	Accuracy
0	0.71	
1	0.95	
		0.92

Table1. Imbalanced classes

	F1-Score	Accuracy
0	0.92	
1	0.92	
		0.92

Table2. Balanced classes

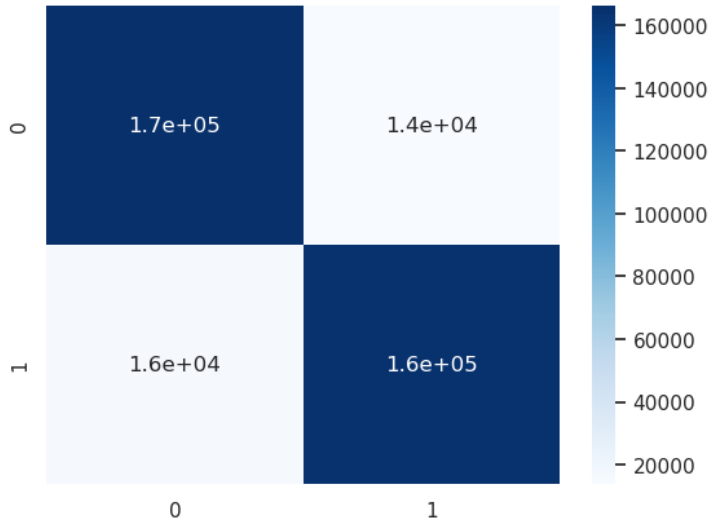

Fig11

After having a 0.92 accuracy as my benchmark, I fine tuned with my data with distilBERT, configuring the batch size 32, epoch 8, learning rate at 3e-5, and optimizer Adam. I experimented from batch size 16 to batch size 32 and epoch from 2 to 8.

Evaluation

Because it's a balanced class, I would evaluate with accuracy score. The distilBERT has accuracy score at 0.98 and is so much better than baseline model which has the accuracy score 0.92. The training accuracy increases as the epoch increases, and hasn't seen any decrease yet, meaning it has the potential to run more epoch and reaches higher accuracy.

The distilBERT model has 0.96 the best training accuracy among 8 epochs:

```
Epoch 1/8
45133/45133 [==============================]- 1651s 36ms/step- loss: 0.2074- accuracy: 0.9146
Epoch 2/8
45133/45133 [==============================]- 1627s 36ms/step- loss: 0.1746- accuracy: 0.9303
Epoch 3/8
45133/45133 [==============================]- 1627s 36ms/step- loss: 0.1560- accuracy: 0.9390
Epoch 4/8
45133/45133 [==============================]- 1627s 36ms/step- loss: 0.1404- accuracy: 0.9463
Epoch 5/8
45133/45133 [==============================]- 1627s 36ms/step- loss: 0.1259- accuracy: 0.9528
Epoch 6/8
45133/45133 [==============================]- 1627s 36ms/step- loss: 0.1131- accuracy: 0.9581
Epoch 7/8
45133/45133 [==============================]- 1627s 36ms/step- loss: 0.1019- accuracy: 0.9630
Epoch 8/8
45133/45133 [==============================]- 1626s 36ms/step- loss: 0.0929- accuracy: 0.9666
```

And 0.98 testing accuracy with loss at 0.05:

11284/11284 [==============================]- 164s 14ms/step- loss: 0.0577- accuracy: 0.9819
loss: 0.05769692733883858
accuracy: 0.9819391965866089

Conclusion

Based on the results from EDA, we can gain insights about what are customers' needs when they shop for their fur family. However, the limitation is that the feature extraction gives much less insights as I imagined. Optimizing the usage of a dataset is a way to save the cost. So, business could, for example, redesign a UX to obtain ideal information they would like from customers and gain better insights from the data. Another limitation is that this dataset is quite obsolete and is not gathered by pet supplies categories, meaning we could not extract specific feedbacks from different animal products like turtle's products.

Model selections depended on the needs of business or preference. Even though the logistic model is a baseline model, the accuracy score is already good by industrial standard. However, the fine tuned distilBERT has so much better performance despite of its disadvantages in terms of time consumption and high cost. If the cost of training is the priority concern, then business can choose logistic model. Otherwise, fine tuned a pre-trained model gives a higher accuracy score since epoch2.

Summary

When doing text preprocessing, I realized that there are so many tokenization tools that can be used and each has its pros and cons. It would be interesting to further dive into this. However, I used regex to clean my text in this project because it's fast especially when I have a big dataset. It only took 1min and 12 sec to clean 1.2GB text data. Also, stopwords plays an important role when dealing with sentiment analysis. Either a customized list or a list from nltk package, it's worth to pay attention on the modified text and be aware of what tasks are performed. Imbalanced dataset is another issue as well. It's common in a real-world setting. So, gaining more knowledge to deal with this problem is important. Pre-trained model is more accurate but time consumed and expensive. I would love to experiment more different models in the near future.

Reference

https://huggingface.co/docs/transformers/model_doc/distilbert
https://github.com/amir-jafari/NLP
https://towardsdatascience.com/sentiment-analysis-with-python-part-2-4f71e7bde59a
https://www.kaggle.com/code/alexalex02/sentiment-analysis-distilbert-amazon-reviews/notebook#DistilBert
https://towardsdatascience.com/everything-you-need-to-know-about-albert-roberta-and-distilbert-11a74334b2da
https://towardsdatascience.com/baseline-models-your-guide-for-model-building-1ec3aa244b8d
https://towardsdatascience.com/word2vec-explained-49c52b4ccb71