Classifying Amazon Reviews Using Zero Shot Classification

By Maya Luther (24915012)

Introduction

For this project, I used a Large Language Model (LLM) from Hugging Face to evaluate online reviews of an Amazon shoe seller and determine whether they were positive, negative, or neutral. E-commerce is a field rich in data and has many opportunities for analysis such as this. The use of a LLM on online reviews makes sense, as these reviews are written by humans in a natural language and need to be processed before large data analyses. The dataset contains two features consisting of a numerical score from 0-4 and the review itself as a string. My aim was to use a Zero-Shot Classifier model to assess the language used in the review and predict whether the review was positive, negative, or neutral. This serves as important feedback for the Amazon seller and can help address weak points in their e-commerce strategy.

Data Exploration

Data Collection

This dataset comes from the Hugging Face database. It has 10,000 rows of data and was updated June 5, 2023 by Hugging Face user Peter Szilvasi.

Features

	labels	text	text_length
0	3	Good shoe for office work. They will scuff ver...	65
1	1	I have had the Patricia II wedge in black for ...	1145
2	1	Width not right and size too small if width ha...	136
3	0	I received these shoes and they weren't the sa...	277
4	2	They began to split alone the mesh material af...	95
...	...	...	...
89995	2	I wear a size 7 in all my shoes but this one w...	111
89996	3	Love the sunglasses. Love the look. Love the P...	354
89997	0	Were comfortable the 1st time, but seem to get...	136
89998	1	Hurts my feet. Like wearing razor blades. The ...	281
89999	3	Good sandals for around the pool but not a pai...	184

Fig 1: Training dataset in a Pandas Dataframe.

	labels	text_length
count	90000.000000	90000.000000
mean	2.000367	174.725967
std	1.414634	228.961303
min	0.000000	0.000000
25%	1.000000	45.000000
50%	2.000000	104.000000
75%	3.000000	220.000000
max	4.000000	9231.000000

Fig 2: Description of training dataset.

Visualizations

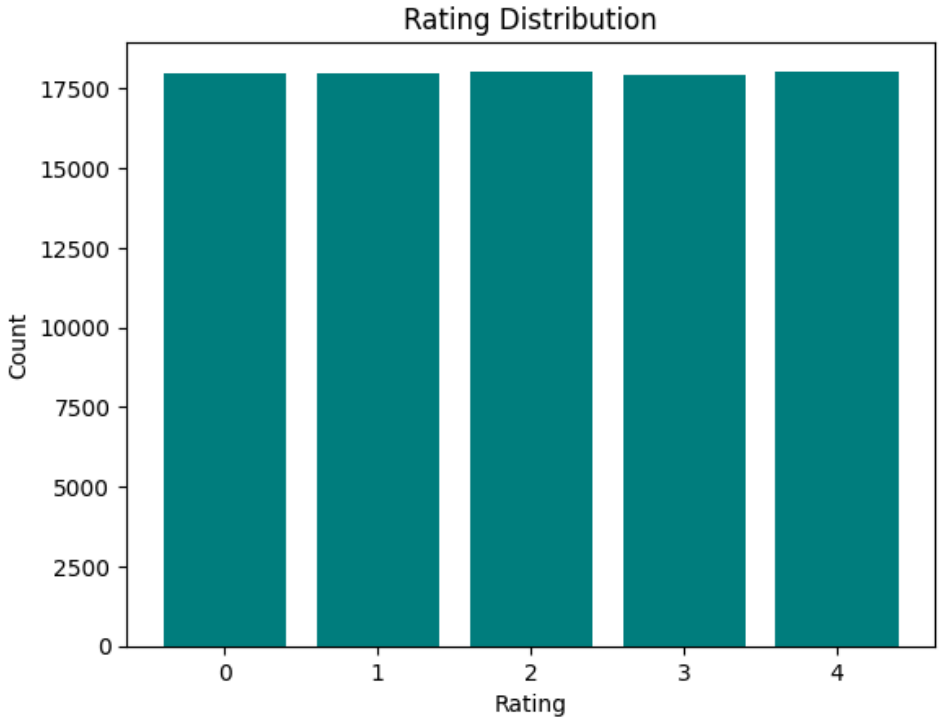

Fig 3: Counts of each review score.

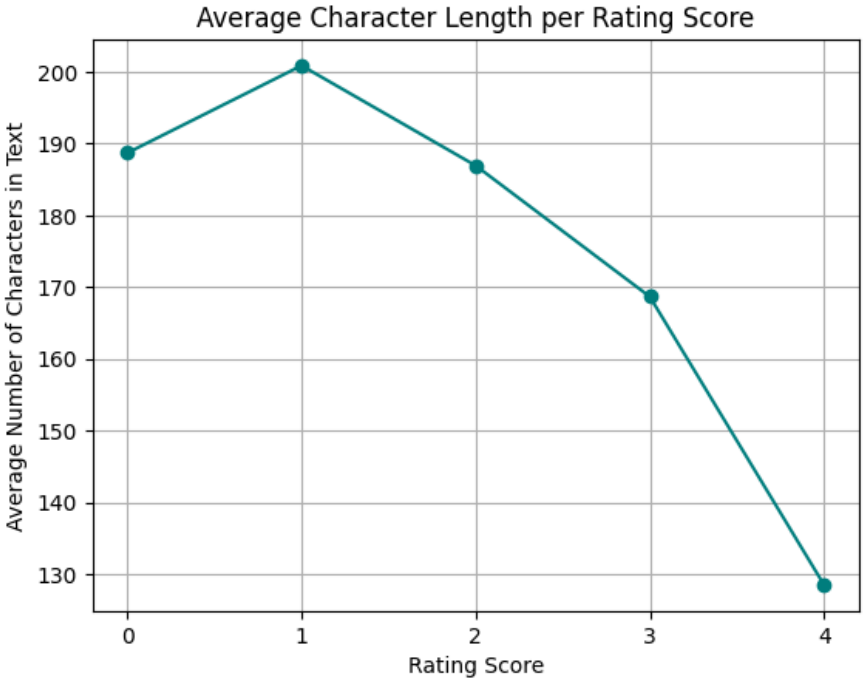

Fig 4: Line plot of relationship between the average length of a review and its score.

Methods

Pre-processing

The text in this dataset seems to have been webscraped, and thus contained unnecessary portions of text such as HTML tags. Anything between HTML brackets were removed, and the text was converted to lowercase. For the sake of reducing runtime when using the model, each word in the review was stemmed to its least redundant form. In future work, I would also remove filler words to reduce runtime.

About the Model

The model I used is one uploaded to Hugging Face by Facebook called BART. Its strengths are in text generation and text-based comprehension tasks. The particular instance I used has been trained on the MultiNLI dataset by NYU. This process uses NLI-based Zero Shot Text Classification, which works by giving the model a string along with possible class names, and generates a probability for how relevant each class name is to a string.

Zero-Shot Classification

In Zero-Shot Classification, the model can be used to recognize classes without requiring labeled examples for that specific class, though trained on supervised data. In the instance I used, I asked the model to identify the classes of “positive”, “negative”, and “neutral”. I allowed the model to interpret multiple classes as correct, however looking back in this case it may have been better to allow just one, as I ended up accepting the strongest class anyways. Allowing multiple classes to be correct would make more sense if the classes weren’t mutually exclusive.

Results

	True Label	Predicted Label	Confidence Score
0	negative	negative	0.998240
1	negative	negative	0.996514
2	negative	negative	0.123774
3	negative	negative	0.907510
4	neutral	negative	0.982493
...	...	...	...
195	negative	negative	0.819954
196	positive	positive	0.925454
197	negative	positive	0.998986
198	negative	negative	0.263888
199	negative	positive	0.026833

Fig 5: Comparison dataframe of the results of the classification model

Performance

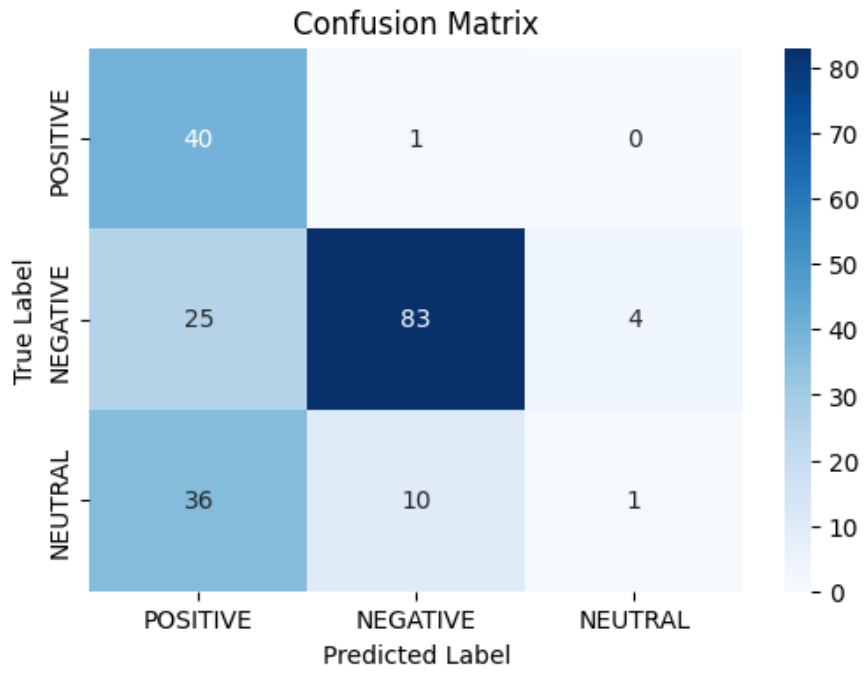

Fig 6: Confusion matrix

Conclusion

From the preview of the comparison dataframe, we can see that there are a few misclassifications even with a high confidence score, and others with low ones. With the accuracy of the model being calculated as 62%, we can conclude that it is a fairly accurate model. From the confusion matrix, we can see that the model does not seem to predict neutral reviews much at all, and would rather predict a review to be positive or negative. Its strength seems to be especially in predicting negative reviews, and predicted positive reviews seemed to be a bit varied. This could be because positive reviews address criticisms of the product despite being overall positive. In the end, I think that this is a fairly successful model, as negative reviews are of the most concern when it comes to review classification.

Recommendations

In a future iteration of this analysis, I would pre-process the text a bit more, and perhaps eliminate the “neutral” category altogether. This same model could also be used to look at what were the greatest faults of the products. I would do this by replacing the “positive”, “negative”, and “neutral” classes with words such as “poor quality”, “shipping”, or “expensive”, topics that customers may complain the most about.