Amazon Product Review Sentiment Analysis Report: PDF composed in Python with reportlab library

Dataset Description

This report analyses the sentiment of customer reviews for Amazon products dated between 2010 and 2018, sourced from bestbuy.com and amazon.com. The dataset is retrieved from the file named 1429_1.csv (48.99MB) from this page on Kaggle.com and contains 34,660 reviews.

All columns except reviews.text were dropped, since they will not be used for the current scope of sentiment distribution analysis. In future work, other columns would be required in order to draw correlations and predictive analytics between aspects of the dataset which may be interrelated.

Preprocessing Steps

Leveraging spaCy, pandas, and inbuilt Python string manipulation functions, the following preprocessing steps were applied to the review text column:

- Text cleaning: Removed stop words, punctuation, and converted to lowercase.

Performance Considerations

- In order to reduce duration of preprocessing, the preprocessing function was refactored from an approach of using a lambda function to apply to every review one by one, to a pipeline/text stream-based function using spaCy's pipe functionality. Batch processing occurs, as well as parallelisation, thanks to spaCy's parameters inside the nlp.pipe(texts, batch_size, n_processes) function. Thereby, reviews are bunched together to be processed in batches, and the pipe will create as many functions as it can - limited by the number of CPU cores.

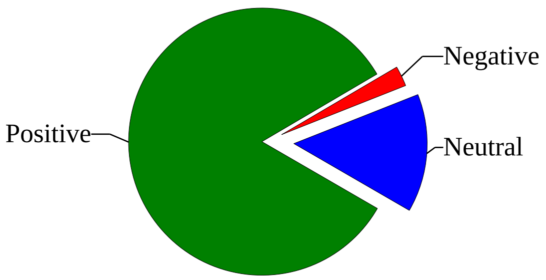
- The bottleneck in preprocessing comes from the fact that there is a single nlp instance which must share its logical resources despite the parallelisation, and it is likely that creating worker functions with individual nlp instances would reduce runtime greatly. This latter approach is applied to sentiment analysis further down.

Sentiment Analysis Results

The sentiment analysis identified 28847 positive reviews, 850 negative reviews, and 4962 neutral reviews.

- spaCy is used in pipeline with a spacytextblob component engaged, and it is via TextBlob attributes that the sentiment and polarity of the reviews is judged. These counts are highly likely to change if the model could see the entire sentence in every case, e.g. without preprocessing, which could impact the meaning of the sentences in the reviews. It is also likely that a beefier language model would disagree with the small English spaCy language model used here, which lacks proper vectors between representations of word meanings for reasons of capacity limitation.

- There is an artistic licence in counting neutral reviews, since polarity is a continuous scalar floating point value in the range -1 to 1, meaning you are unlikely to find reviews which have a score of a perfect 0. Widening or narrowing the bounds for what is considered basically neutral will naturally alter the counts of the sentiment labels of the reviews dataset.



Sample Reviews

Positive review sample: *great for beginner or experienced person. Bought as a gift and she loves it*

Negative review sample: *I really like this tablet. I would have given 5 stars but sometimes you have to push start several times after you unlock the screen and it is a little annoying.*

Neutral review sample: *This product so far has not disappointed. My children love to use it and I like the ability to monitor control what content they see with ease.*

Commentary on Accuracy

- The positive review seems an adequate categorisation. The negative review is not that negative, it is close to as positive as can be, save for one small detail. The neutral review sounds a lot more positive than neutral, perhaps the confounding word is 'disappointed', but the model should be able to see the whole context of the token and state that this is a positive review.

- It is debatable whether reviews are being accurately categorised to an adequate standard for executive decision making. There are plenty of nuances in speech, including sarcasm and humour, which it is hard for such a small language model to pick up on.

- Furthermore, there is additional useful data to be integrated to get the full picture: e.g., did the customer return the product? What was the title of the review? Does the user buy other products in the same category thereby making their opinion a more comparative one? There are ways that objectivity can also be analysed, using spacytextblob, and this could be incorporated for deeper value to be extracted from these reviews.

Insights

Based on the sentiment distribution, the high percentage of positive reviews suggests customer satisfaction. On the other hand, one can qualitatively notice some reviews which are falsely categorised. Further analysis of positive/negative review content can reveal valuable perspectives into customer preferences and areas for improvement, only if the confidence in the sentiment analysis component is high. One should consider all columns that might be relevant to informing future stock/procurement/delivery service improvement decisions to be made.

Review Similarity Example

Review 1: Not easy for elderly users cease of ads that pop up.

Review 2: Excellent tablet with nice screen. I wish Amazon would pre install the play store, this would have been perfect.

The similarity score between the selected reviews is: 0.23. The main purpose of this display is to show how spaCy has inbuilt methods that allow similarities between sentences to be estimated. There raises, however, a warning:

UserWarning: [W007] The model you're using has no word vectors loaded, so the result of the Doc.similarity method will be based on the tagger, parser and NER, which may not give useful similarity judgements. This may happen if you're using one of the small models, e.g. `en_core_web_sm`, which don't ship with word vectors and only use context-sensitive tensors. You can always add your own word vectors, or use one of the larger models instead if available.

...and this warning tells us we should be using a medium-sized or larger spaCy language model to accurately leverage insights from similar reviews (to extract key themes, understand commonalities between satisfied customers in order to maximise customer satisfaction in future, and more).

- **Generally**: Reviews with high similarity scores likely discuss similar themes, while low scores suggest diverse or contrasting opinions. It is fruitful to point out that the similarity scores are judged from the raw reviews.text column in this instance, rather than the cleaned text column since this may have cleaned out the nuance in the review.

- **Caution**: However, using user-entered text data is dangerous at the point of analysis. There may be typographical errors as seen in Review 1 above; there may be unfamiliar slang, abbreviations, or pop culture references. These all along with extra myriad factors that can confound a language model and cause its similarity score to stray from something a human would judge.

Model Strengths

Here are some strengths of the sentiment analysis model:

- **Identifies overall sentiment quickly**: The model distinguishes between positive, negative, and neutral reviews, aiding in understanding overall customer sentiment. With my modifications to make the preprocessing and sentiment analysis run quicker, this approach represents a decent quick and dirty first-pass attempt at analytics of free text, which is often the hardest data to analyse in a dataset.

- **Potential for customer insights**: Analysing the patterns or themes within positive and negative reviews can provide valuable insights into customer preferences and pain points. It would be wise to continue adding to the Natural Language Processing arsenal employed here, by extracting entities which are mentioned, perhaps taking a lemmatised approach, using a best-guess spelling corrector function, keeping polarity as a scalar rather than a categorical variable, and more.

Model Limitations

- **Difficulty in understanding nuance**: Sentiment analysis models may not always capture sarcasm or complex emotions. Additionally, the accuracy can be influenced by the dataset size and quality. It's recommended to manually review a sample of classified reviews to assess model performance, so a next step if trying to test this pipeline would be to compare human-labelled sentiments of reviews to what this model outputs for its best guess of sentiment.

- **Tradeoffs and the scientific approach**: It would be fruitful to try with bigger models, always accounting for a balance of runtime, memory usage, CPU effort, and budget to provide for these resources. In all cases, logging and timing data should be collected to add a degree of impartiality to improving the analysis pipeline.

Future Research Directions

- **Experiment with other spaCy Models**: Experiment with medium-sized spaCy models (e.g., en_core_web_md) to potentially improve the accuracy of sentiment analysis and similarity scores.

- **Evaluation**: Include quantitative evaluation metrics (accuracy, precision, recall, F1-score) to compare the performance of the model before and after optimisations and with different spaCy models. Use a manually labelled dataset for this purpose.