Task 21 - Sentiment Analysis Capstone

Note to the reviewer:

The task explicitly states that I should create the file ‘sentiment_analysis.py’ for this task. It’s my belief that using Jupyter Notebook would be more suitable, given the need for reporting on my findings, etc. I also feel it would be good to use visualisation for this task. However, since ‘.py’ is specified, not ‘.ipynb’, I have used Visual Studio Code.

This has led me to embrace the functionality made possible with VSC, resulting in the menu screen UI I’ve created.
I feel this approach has satisfied the brief I’ve been given, but I acknowledge this wouldn’t be the correct approach if the task was to run bulk sentiment analysis, or to store or export the results.

Table of Contents

1. Introduction

The Dataset

I chose to use the smallest of the three datasets. This is because, for the purpose of the task, I only needed a handful of reviews. Using the smaller of the three has the benefit of putting less strain on my computer. Nevertheless, this dataset has over 34000 lines.

The dataset is presented in the form of a CSV file, and contains information relating to Amazon products and their user reviews.

For this task I'm asked to isolate the column named 'review.text'.

Other potentially useful columns are:

- 'reviews.doRecommend' - a boolean True or False, and;
- "reviews.rating" - a user-input score between 1 and 5.

These columns could be used to further verify the accuracy of the model's sentiment prediction.

There are also columns for reviewer location, though these appear to be blank in this dataset. This could be an interesting thing to investigate, to see if location affects user satisfaction of specific products.

Preprocessing steps

For the purposes of assessing the model's prediction accuracy, the following preprocessing steps have been taken:

- Import dataset into a Pandas DataFrame
- Remove rows that contain no review text
- Isolate the reviews column

Chosen review texts are then:

- Converted to NLP objects
- Tokenized
- Lemmatized

Stop words are then removed, along with punctuation marks and whitespaces.
The resulting list of lemmatized tokens are then joined into a string, making all lowercase.

2. Evaluation of Results

Introduction

In this section we compare the original and processed texts from 10 randomly chosen indexes.
We view the polarity score as output by the program and assess the accuracy of this score against the intended sentiment of the reviewer.

Text has been processed as follows:

- Stop words removed
- Punctuation and whitespaces removed
- Tokenization
- Lemmatization

Findings

More often than not, the model produces what I feel is an accurate score.
Some of the results are quite intriguing, with fairly neutral-looking tokens producing a very positive score, and vice versa.
From this small selection from the dataset:

- In 8 of 10 tests, the sentiment output by the program matches that as assessed by a human (me) - allowing for a small tolerance.
- In only 2 of 10, the sentiment seems inaccurate.

Review 3453

Original text: The charger won't stay securely connected to the Paperwhite and wiggles. It will only charge when held at a particular angle.
Tokens only: charger will stay securely connected paperwhite wiggle charge hold particular angle
Polarity value: 0.283
Description: Somewhat positive
Assessment:

- I would say this score is fairly generous. I'm not sure which lemmas lead it to give a value above 0.000.
- In reality, having to keep a charging cable set at a particular angle would be a real inconvenience and might cause the user to return the item.
- Lemmatizing the text has changed 'won't' into 'will', which changes the meaning considerably, if perhaps not the sentiment value.

Review 6666

Original text: This tablet is a good buy for the price. Allows quick and easy access to the internet. Great product for a 1st time user.
Tokens only: tablet good buy price allow quick easy access internet great product 1st time user
Polarity value: 0.567
Description: Very positive

Assessment:

- I would say this score is quite accurate.

Review 777

Original text: this item work just as I expected it to, great product!
Tokens only: item work expect great product
Polarity value: 0.800
Description: Very positive
Assessment:

- I would say this score is quite accurate.

Review 29998

Original text: much faster and stable than the fire tv stick... definitely worth the upgrade
Tokens only: fast stable fire tv stick definitely worth upgrade
Polarity value: 0.250
Description: Somewhat positive
Assessment:

- I would say this score is quite reserved.
- The original text implies that the product is comparatively fast and stable, essentially, better than its rival products.
- The model is perhaps being misled by the inclusion of ‘fire stick’, perhaps believing the reviewer thinks that product is better, and believing the reviewer is suggesting people upgrade from this product, not to it.

Review 12345

Original text: Best tablet on the market for the price without a doubt
Tokens only: good tablet market price doubt
Polarity value: 0.700
Description: Very positive
Assessment:

- I would say this is accurate when going on the lemmas alone.
- However, when looking at the original text, this value seems a little reserved.
- I believe the inclusion of the word ‘doubt’ and the exclusion of ‘without a’ has skewed the result.

Review 34567

Original text: Very easy to use. My bf loves it. Good price.recommend it
Tokens only: easy use bf love good price.recommend
Polarity value: 0.544
Description: Very positive
Assessment:

- I would say this score is quite accurate.
- It’s surprising to see the “.” has remained, despite the preprocessing. However, it doesn’t seem to have compromised the accuracy of the analysis.

Review 17356

Original text: I definitely recommend this tablet is so perfect for children.
Tokens only: definitely recommend tablet perfect child
Polarity value: 0.500
Description: Very positive
Assessment:

- I would say this score is accurate, if a little reserved.
- I would expect the inclusion of “perfect” to elevate the score.

Review 29347

Original text: This is the best of both worlds. A reasonably priced BT speaker and an Amazon Alexa enabled device. I would highly recommend to anyone looking for a BT speaker and considering an Alexa enabled device.
Tokens only: good world reasonably price bt speaker amazon alexa enable device highly recommend look bt speaker consider alexa enable device
Polarity value: 0.353
Description: Very positive
Assessment:

- Similar to the previous review: I feel this is accurate, if a little reserved.

Review 32198

Original text: Bought to play my PlayStation Vue tv, love the Alexus feature overall a great thing for steaming.
Tokens only: buy play playstation vue tv love alexus feature overall great thing steaming
Polarity value: 0.433
Description: Very positive
Assessment:

- This feels quite accurate to me.

Review 109

Original text: Works as we thought it should. NO PROBLEMS. Good buying experience.
Tokens only: work think problems good buying experience
Polarity value: 0.700
Description: Very positive
Assessment:

- This feels quite accurate to me.

3. The Effect of Stop Words

Introduction

In this section we compare and assess polarity values for text, both with and without stop words, to better understand the effects of this process.

The tokens in this section have not been lemmatized.

Review[0]:
With stop words 0.325
Without 0.050
The removal of the stop word ‘not’ changes the fairly positive sentiment of ‘product so far has not disappointed’ to ‘product far disappointed’.

Review[1]:
With stop words 0.800
Without 0.800

Review[2]:
With stop words 0.600
Without 0.600

Review[999]:
With stop words 0.633
Without 0.633

Review[20000]:
With stop words -0.385
Without -0.457
The reason for this discrepancy is less clear cut than review [0]. It seems to simply be the lack of nuance given by the stop words that affects these scores.

Review[30000]:
With stop words 0.550
Without 0.550

Summary

It’s fair to say that, from this small dataset, stop words seem to add valuable nuance to longer texts, such as reviews, and removing them does affect the accuracy of the outcome.

4. The Effect of Lemmatization

Introduction

In this section we compare and assess polarity values for text, both with and without lemmatization of the tokens, to better understand the effects of this process.

Stop words have been removed for this section.

Review[0]:
Without lemmatization -0.050
- product far disappointed children love use like ability monitor control content ease

With lemmatization 0.300
- product far disappoint child love use like ability monitor control content ease

Review[1]:
Without lemmatization 0.800
- great beginner experienced person bought gift loves

With lemmatization 0.700
- great beginner experienced person buy gift love

Review[2]:
Without lemmatization 0.600
- inexpensive tablet use learn step nabi thrilled learn skype

With lemmatization 0.600
- inexpensive tablet use learn step nabi thrilled learn skype

Review[999]:
Without lemmatization 0.633
- great tablet price fast works great great sound volume great pixelation screen size

With lemmatization 0.633
- great tablet price fast work great great sound volume great pixelation screen size

Review[20000]:
Without lemmatization -0.457
- reading bit easier paperwhite navigating drove nuts touch screen annoying limited functionality tried like sitting unused use expensive fire

With lemmatization -0.235
- read bit easy paperwhite navigate drive nuts touch screen annoying limited functionality try like sit unused use expensive fire

Review[777]:
Without lemmatization 0.350
- item work expected great product

With lemmatization 0.800
- item work expect great product

Summary

As with the previous trial, there are occasionally discrepancies between the polarities here. It's interesting to note that, for the most part, the reviews that had large discrepancies in the stop word trial also have large discrepancies in this trial. And that lemmatization brings the values closer to those with all stop words included.

It's very interesting to see how small differences can lead to large discrepancies in polarity. This is most notable in review 777, where the simple change from 'expected' to 'expect' increased the sentiment value by 0.450.

5. Similarity in Review Pairs

Introduction

As per the task instructions, here I demonstrate that I can use the .similarity function to compare pairs of reviews.

Select an index for comparison: 0
Select an index for comparison: 1111

Review 1 Lemmas: product far disappoint child love use like ability monitor control content ease
Review 2 Lemmas: thank perfect little kid love tablet good quality fast
Similarity value: 0.709
Description: Very similar

Select an index for comparison: 11111
Select an index for comparison: 22222

Review 1 Lemmas: purchase tablet 9 year old daughter extra christmas present love start great starter tablet reading game lag gaming space perfect house highly recommend time tablet starter especially price
Review 2 Lemmas: nice sound speaker add app listen music
Similarity value: 0.569
Description: Very similar

Select an index for comparison: 0
Select an index for comparison: 34659

Review 1 Lemmas: product far disappoint child love use like ability monitor control content ease
Review 2 Lemmas: spite fact good thing amazon anthing get love fire find greedy wall charger come kindle ok people usb port plug take charger think amazon thing right let purchase kindle charger free credit buy
Similarity value: 0.793
Description: Very similar

Select an index for comparison: 26475
Select an index for comparison: 9825

Review 1 Lemmas: honestly difference iphone
Review 2 Lemmas: niece love tablet long learn app download app visit wifi etc thank
Similarity value: 0.598
Description: Very similar

6. Insights Into the Model's Strengths and Limitations

Strengths

spaCy's en_core_web_sm provides decent linguistic analysis, capturing essential syntax and meaning in the product reviews. Although it has fewer parameters than the larger model (en_core_web_md), limiting its understanding of nuance to a degree, it is more lightweight, making it suitable for scenarios with limited computational resources.

TextBlob's is quick and straightforward to use for initial sentiment analysis, allowing users, regardless of coding experience, to gain insights into sentiment patterns in product reviews quickly.

Combining NLP with spaCy's en_core_web_sm and TextBlob enhances the adaptability of the model, enabling effective sentiment analysis across different writing styles. This combined approach benefits from spaCy's linguistic insights and TextBlob's simplicity.

Limitations

The combined model, like any sentiment analysis model, struggled with handling ambiguous language. Ambiguity in product reviews can lead to challenges in accurately categorising sentiments.

TextBlob's simplicity may limit its ability to capture complex language structures or nuanced sentiments that require deeper analysis. The model may struggle with highly intricate expressions present in certain product reviews.

The model only takes into account the reviews text. Deeper analysis would require additional inputs, like the boolean recommendation True/False and star ratings that are within the dataset.

Conclusion

The combined approach using NLP, TextBlob, and spaCy's en_core_web_sm gives a good balance between linguistic analysis and ease of use. It’s easy to use and offers a good degree of accuracy with only simple data inputs.