Suhail Sharafuddin

Amazon Reviews Analysis Report

Problem Statement

Many companies require the ability to analyze reviews. There are quite a few advantages to doing this which include developing customer loyalty and figuring out why people might prefer competitors' products.

I will be using a dataset that has Amazon reviews for SanDisk SD cards. This file has over 4000 rows of reviews. Other columns include user id, how many people found the review helpful, and star rating.

In this project, I will use models to guess the sentiment of the reviews using sentiment analysis. The models will guess whether the given review is positive, negative, or neutral, and give it a score as well.

Data Wrangling

First, I had to do the data wrangling. Data wrangling is an incredibly important step. It is the process of transforming the raw data into a more consumable format for machine learning. It involves making the data presentable and cleaning up errors.

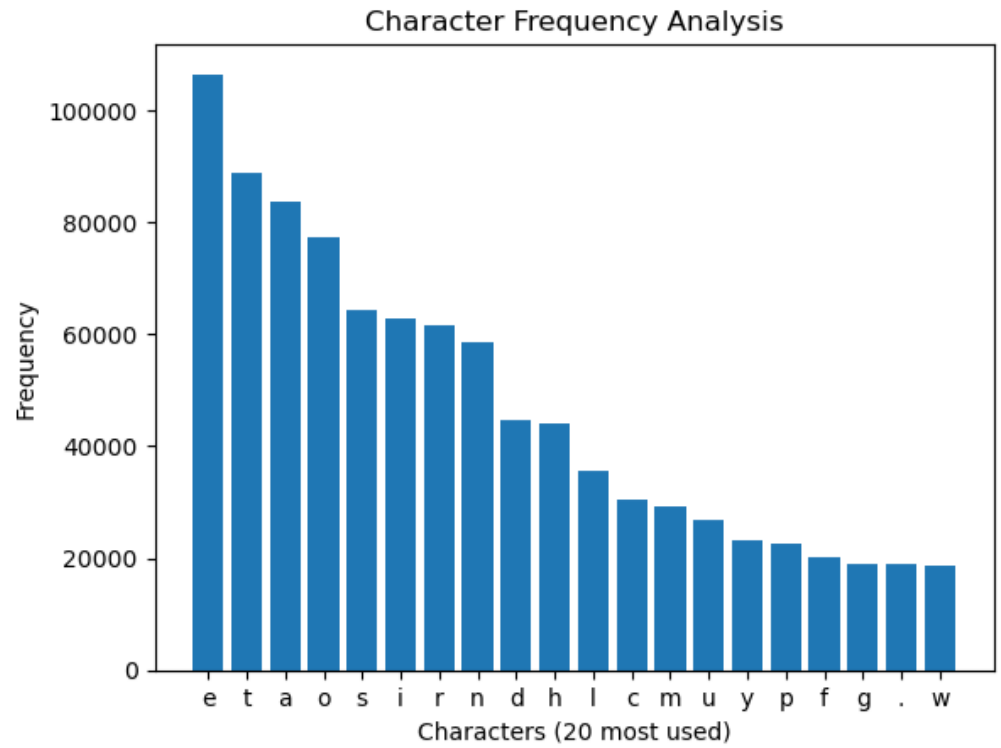
The dataset was already clean. There were only two missing values in different rows that I would have to worry about. One NAN value was in the Reviewer Name column, and the other one was in the Review column. I kept the NAN for the Reviewer Name column since the rest of the row was intact. For the Review column, I deleted that row. I did this because the Review is the most important part of the data, and a missing value for that column meant that the entire row was useless.

Exploratory Data Analysis

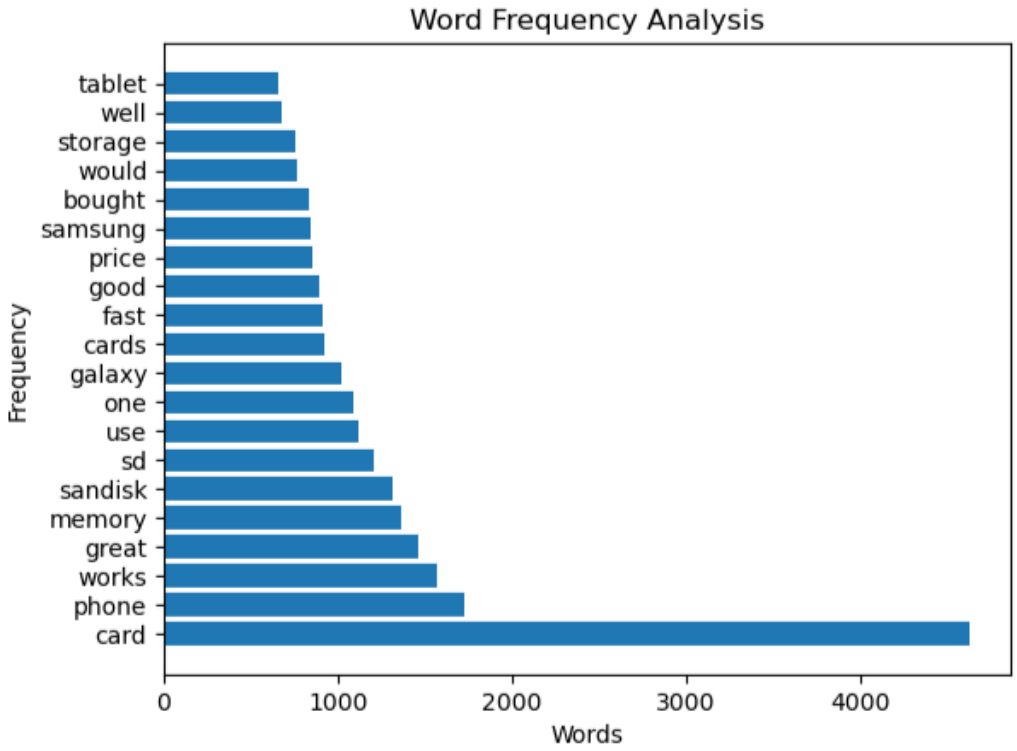
Exploratory Data Analysis is getting a feel for your data, usually by visualization. It helps summarize the main characteristics of the dataset.

Before I dove into the analysis itself, I made sure to concatenate the reviews as one string. This would help with easier analysis.

I plotted the 20 most used characters in the string. If I plotted all characters, then the graph would become too muddled. I also took out the spaces, as spaces were by far the most used character. As shown below, 'e','t','a', and 'o' were the most frequent characters.



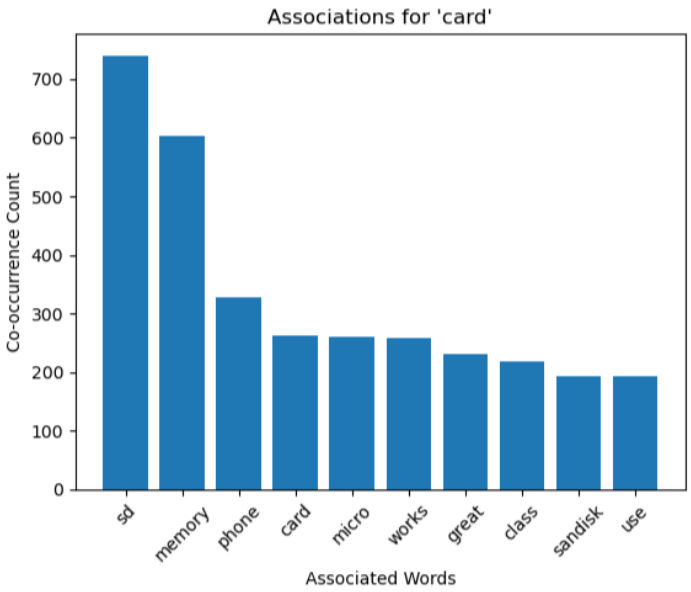
Next, I looked at the frequency of the 20 most used words. I plotted it horizontally since it would be easier to read. 'card' was much higher than any word, with 'phone','works','great', and 'memory' being incredibly frequent as well. This was when I realized that the overall sentiment would most likely be positive.

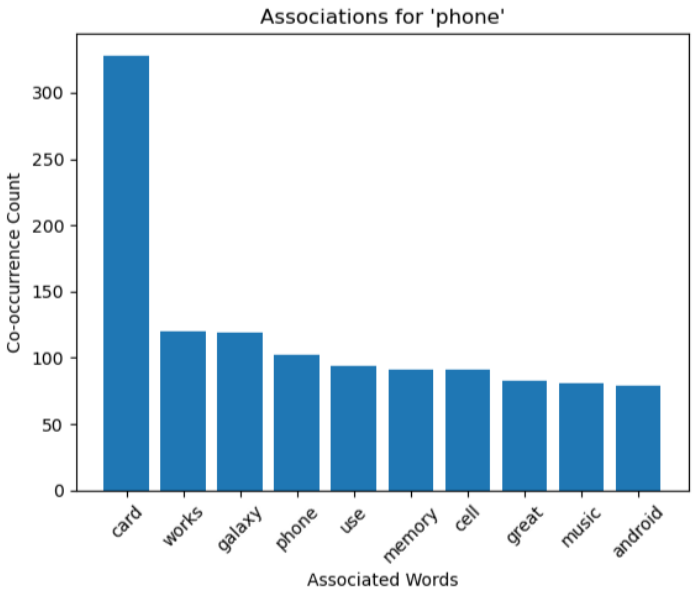


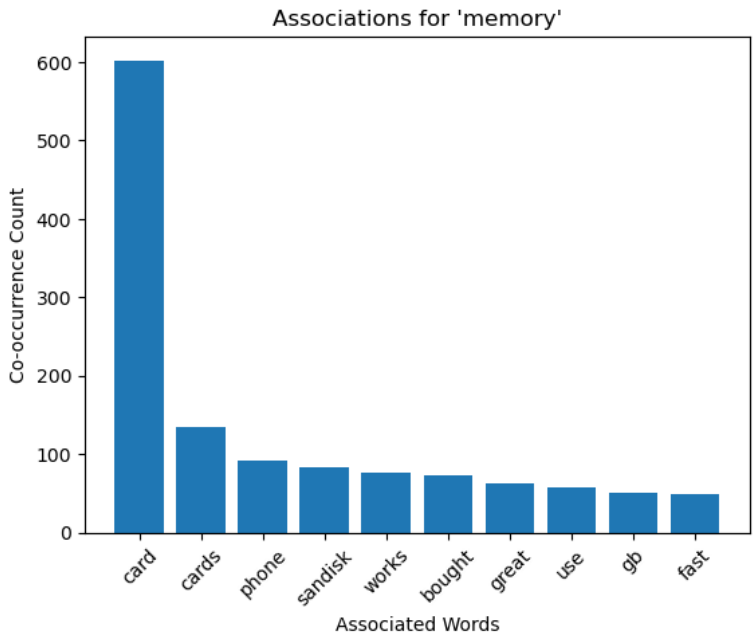
I created a word cloud from the word frequency as well.

Finally, I looked at the word associations for the most frequent words. These words were ‘card’, ‘phone’, and ‘memory’. I plotted the 10 most associated words.

Most of the associations were more technical such as 'card','phone','memory', and 'sd'. However, in terms of association when it relates to sentiment, it seemed positive. This was because they were heavily associated with 'great', 'fast', and 'works'.







Modeling

Next, I had to do the modeling. But, before going into the models, I made all the reviews into one big list. Then, I got three pretrained models from the Hugging Face Transformers library. The first model was based on the DistilBERT architecture. The second model was a variant of the DistilBERT model fine-tuned on the Stanford Sentiment Treebank (SST-2) dataset. The third model was a Distilbert model fine-tuned on English language product reviews by Julie Simon. I called these models model_1, model_2, and model_3 respectively.

The three models labeled every review with a sentiment, and a score of the review's po sitivity or negativity. For model_1 and model_3, LABEL_1 = positive and LABEL_0 = negative . I looked at a few individual reviews to see what the models said about them. The first review was:

```
'it works as expected. I should have sprung for the higher capacity.  I think its made a bit cheesier than the earlier versi
ons; the paint looks not as clean as before'
```

model_1, model_2, and model_3 scores for this review:

```
[{'label': 'LABEL_1', 'score': 0.5094072222709656}]
```

```
[{'label': 'NEGATIVE', 'score': 0.9990612864494324}]
```

```
[{'label': 'LABEL_1', 'score': 0.7103031873703003}]
```

model_1 is more neutral with its score. model_2 has an incredibly high negative score, even though the review was nowhere near that negative. model_3 has an incredibly high positive

score, even though the review was not that positive. However, I would say that the review is more positive than negative.

The second review was:

```
"This item was priced right and fit right into my cell phone without any issues. Does what it is suppose to do and I don't k
now what else I could say. Being a class 10 it is suppose to be plenty fast for cell phone use which I am sure speeds up cam
era use to some degree. Has worked well and the shipping was fast."
```

model_1, model_2, and model_3 scores for this review:

```
[{'label': 'LABEL_1', 'score': 0.5204753279685974}]
```

```
[{'label': 'POSITIVE', 'score': 0.9289764165878296}]
```

```
[{'label': 'LABEL_1', 'score': 0.968295156955719}]
```

model_1 is not able to be any higher than 55%, and that hurts it when looking at this review.
model_2 is correct and model_3 are correct.
It is interesting that model_3 has a higher score than model_2, as that is usually not the case. However, this review is quite positive, so it makes sense.

The third review was:

```
'Plenty of extra storage can load a lot of movies in high definition, music and a boat load of other stuff.'
```

model_1, model_2, and model_3 scores for this review:

```
[{'label': 'LABEL_1', 'score': 0.5050829648971558}]
```

```
[{'label': 'NEGATIVE', 'score': 0.994924783706665}]
```

```
[{'label': 'LABEL_1', 'score': 0.9126739501953125}]
```

Overall:
model_1 has the sentiment correct, but cannot handle extremes(very positive or very negative reviews).
model_2 is incorrect about the sentiment for this review. It rarely seems to capture reviews that are more neutral.
model_3 is correct about the sentiment and the score. It generally seems to handle extreme reviews better than model_1 and neutral reviews better than model_2.

Results

I decided to create three new reviews (one super positive, one super negative, and one neutral). This would allow me to see if the patterns I recognized for the models were valid. These were the reviews:

```
best_string = "This is the greatest sd card I have ever used, and I have used many throughout my lifetime. Enough said."
worst_string = "I really have no words for this. This piece of trash that they call a SD card is useless."
neutral_string = "It works fine enough. The file transfer speed could be faster though."
```

Here are the rankings of how accurate the models were for each of the reviews:

For the positive string, these are the models ranked:
1)model_2 = (sentiment = positive, score = 99.96%)
2)model_3 = (sentiment = positive, score = 97.22%)
3)model_1 = (sentiment = positive, score = 50.69%)

For the negative string, these are the models ranked:
1)model_2 = (sentiment = negative, score = 99.98%)
2)model_3 = (sentiment = negative, score = 99.96%)
3)model_1 = (sentiment = positive, score = 51.08%)

For the neutral string, these are the models ranked:
1)model_1 = (sentiment = positive, score = 52.66%)
2)model_3 = (sentiment = negative, score = 54.05%)
3)model_2 = (sentiment = positive, score = 99.87%)

Takeaways

So, my earlier assumptions about the three models were correct. This was:

model_1 can handle neutral reviews well, but not extremely positive or extremely negative reviews well.

model_2 can handle extremely negative and extremely positive reviews well, but not neutral reviews well.

model_3 can handle extremely positive, extremely negative, and neutral reviews well.

Ultimately, model 3 was the best model for sentiment analysis on these reviews. It was the only model that could handle the extremes, and the more neutral reviews.

Future Research

Since NLP is an incredibly popular topic recently, it was rewarding to learn more about it, and how to generally work with text data. Also, the fact that there were almost 5000 reviews was helpful.

For the models, most of the other columns were not important. Only the reviews were important for sentiment analysis.

Overall, these models worked well. However, I would like to try my own NLP models, as well as other pretrained models. My own models almost certainly would not be better than these models, but it would be interesting to see the differences. Other pretrained models would also be fun to play around with. More specifically, I would love to try other models from the Hugging Face Transformers library.