Amazon Products Sentiment Analysis

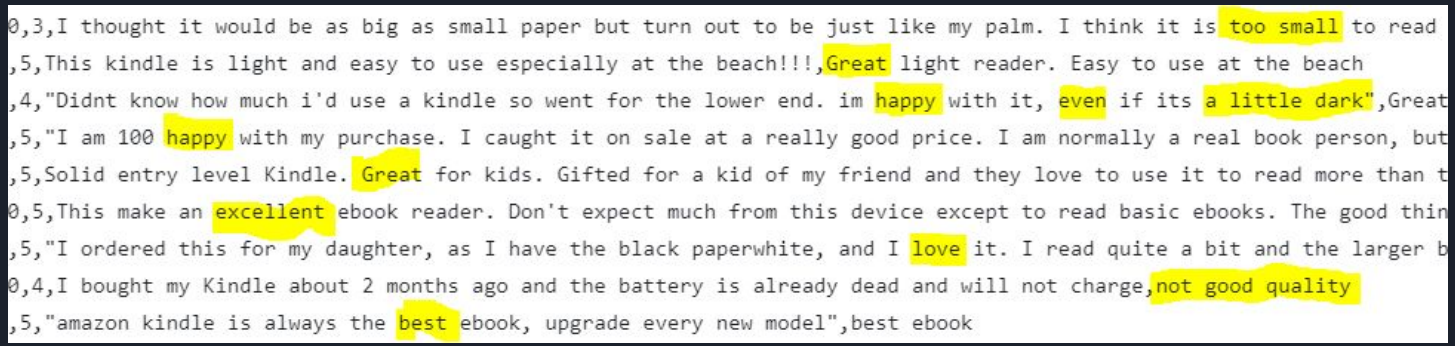
Ahsun Asad, Anisha Kishore, Khadija King

Background

- In today's age of social media, companies receive an overwhelming amount of information
- One major use case is customer product reviews: a large company can easily receive millions of product reviews in total
- These reviews can impact how a company prioritizes how to respond to customer complaints and analyze customer feedback
- Sentiment Analysis is the process of identifying and categorizing opinions to determine whether the user has a positive, neutral, or negative opinion based on their language compared to their ratings system

Database

- As of 2022, Amazon had over 250 million product reviews
- We used an Amazon product database that specifically looked at various Amazon products such as the Amazon Kindle, Echo, Fire Tablet, etc.
- The database included user reviews of these products rated 1-5 stars
- Our goal was to parse through the reviews, in order to create our Sentiment Analysis model
- Sample reviews we parsed through are below



Coding Process

- We standardized the data of Amazon product reviews prior to modeling it:
  - We initially cleaned up the model by removing unnecessary rows, and translating the text into numbers to turn the written reviews into something that could be analyzed my our model in a predictive manner.
  - From there, we padded our sequence for training and testing, which we did after normalizing the features. After this, we initialized our model with two hidden layers, then compiled and trained our model before finally evaluating the model and plotting the accuracy results.
- We looked at various reviews on a 5-star scale and made a judgment on what number of stars would correspond to a specific sentiment:
  - 4 stars or higher is positive
  - 3 stars is neutral
  - 2 stars or lower is negative
- The model would then seek out key words to determine what reviews had a specific sentiment, and predict whether a review was negative/neutral/positive

```
def assign_sentiment(rating):
    if rating >= 4:
        return 'positive'
    elif rating == 3:
        return 'neutral'
    else:
        return 'negative'
df['sentiment'] = df['Reviews_Ratings'].apply(assign_sentiment)
df['sentiment_num'] = df["sentiment"].map({'positive':1, 'negative':2, 'neutral':0 }).values
```

Predictive Behavior, Optimization, and Performance

- The goal of our model was to have 75% classification accuracy or 0.80 R-squared
- We ended up with 0.05% classification accuracy
- Looking back, part of the problem with our model was that we had been looking at every single word in a review for our sentiment analysis
- For example, the word “thought” could be present in a neutral, negative, and positive review (“I thought it was terrible vs. I thought it was amazing”)
- One way to optimize our model’s performance would be to have a stricter guideline for the key words our model is seeking out for predictive behavior

Conclusion

- With how prominent social media and online reviews are, Sentiment Analysis plays a key role in determining customer responses when companies have large volumes of data, such as millions of product reviews
- Although our model was limited to reviews of Amazon products, it could also be used across several different companies and scenarios
  - Various retailer product reviews over time
  - Breaking down which customer demographics have strong reactions to certain products
  - Prioritizing customer service issues
- We do plan on further working on the model and improving it over time to increase our classification accuracy and R-squared
- Once improved, the model would prove highly valuable for monitoring social media reactions and analyzing customer feedback