CAP4770 INTRO TO DATA SCIENCE MAIN PROJECT
AMAZON FOOD REVIEW SENTIMENT ANALYSIS

Team Members: Josh Ngoboc, Raghav Rathi, Amol Sinha, Shreyas Kalikar

Description

Our project aims to create a method of categorising reviews as helpful based on their text. This method will be developed using an existing repository of Amazon reviews including text, ratings, and other metadata.

We have created a method of analysing sentiment around products from the text of their reviews. This allows for more complex information about items to be relayed to customers during shopping.

Dataset

This dataset contains Amazon reviews on exquisite meals. The data spans more than a decade, with all 102,000 reviews up to October 2018 included. Product and user information, ratings, and a plaintext review are all included in reviews. We also have feedback from all of Amazon's other categories. Amazon evaluations are frequently the most visible customer product reviews. As a frequent Amazon customer, we were intrigued by the idea of visualising the structure of a vast collection of Amazon reviews in order to become a more informed consumer and reviewer.

The data was collected between 1996 and 2018, with >100K reviews, >250K users, and >70K products represented.

Insights:

- User Reviews from May 1996 - Oct 2018.
- There are total 102K reviews
- 256,059 users
- 74,258 products
- 3345 users with > 50 reviews

Data Pre-Processing & Sentiment Analysis:

The Dataset contains features such as ProductID, ProfileName, Summary, Time, Score, Time etc.

They were obtained from multiple datasets of Amazon Review Data(2018):

https://nijianmo.github.io/amazon/index.html

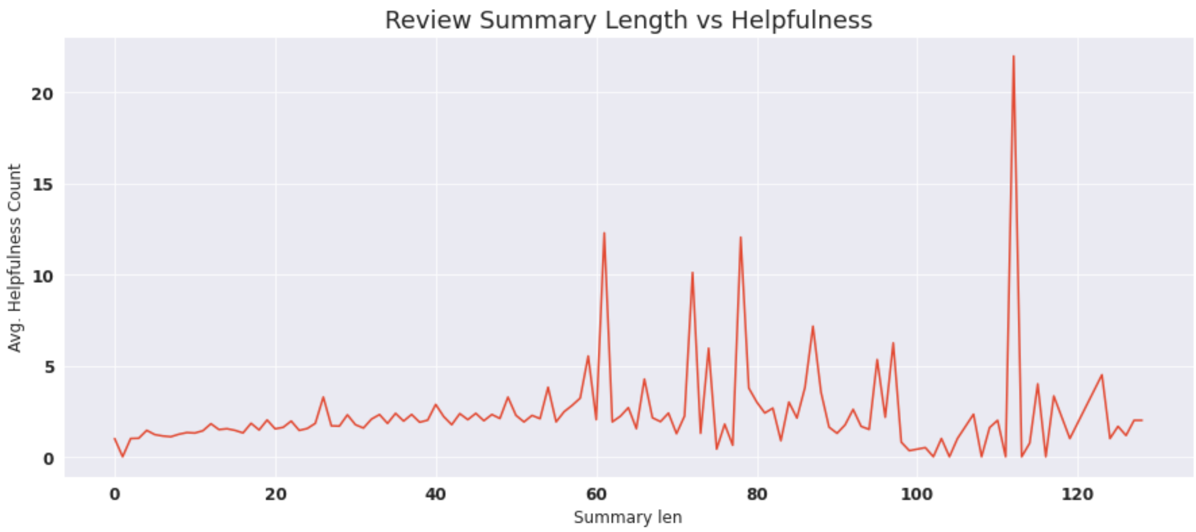
In the dataset, some value entries under the Summary and Profile_Name feature were missing. These were omitted from the main dataframe.

For other very small percentages of missing values, we have overlooked it because they have no effect bearing on our research or predictions, for rest considerable NULL data entries, we have replaced it with empty strings for the sake of calculation.

We also generated diagrams like word clouds to help determine themes between positive and negative reviews.

Hypothesis:

- **Overall product ratings closely correlate with the positive or negative sentiment of the text of their reviews.**
  To confirm our hypothesis, we have built a Feed-forward Neural Network model and trained it on the sentence embeddings to predict the sentiment and compared it with the real reviews, we have found out that higher rated products have a greater level of positive sentiment and vice versa.
- **Increased length of the review increases the helpfulness score.**
  To confirm or deny this hypothesis, we have done feature analysis to extract the helpfulness score and compare it with the length of the reviews. We find out that there is no strong relation of helpfulness score with the average length of review.



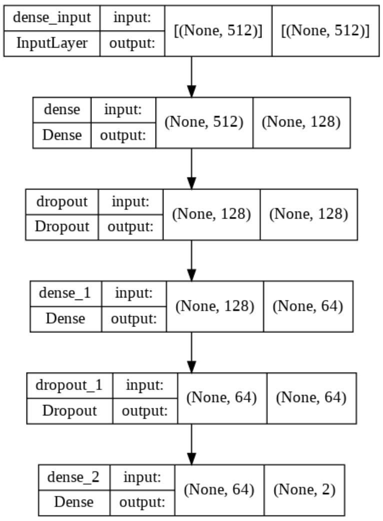
Model Used:

Built a Feed-forward Neural Network model and trained it on the sentence embeddings to predict the sentiment.
We are to create a model that extracts sentiment from user reviews. The review text will be transformed into fixed-length 512 dimension sentence embeddings using Universal Sentence Encoder, a pre-trained model.

We then prepare the data for training an FNN model to predict sentiment by preprocessing the review text.

- Using the score values to create a binary label.
- Sets for training and testing are separated.
- Downsample the predominant class to address class imbalance.
- Using Universal Sentence Encoder, convert the review text into embedded vectors.

Our model follows the structure in the diagram. On our test sets, we were able to achieve an accuracy of ~85% consistently.



dense_input | InputLayer | input: [(None, 512)] | output: [(None, 512)]
dense | Dense | input: (None, 512) | output: (None, 128)
dropout | Dropout | input: (None, 128) | output: (None, 128)
dense_1 | Dense | input: (None, 128) | output: (None, 64)
dropout_1 | Dropout | input: (None, 64) | output: (None, 64)
dense_2 | Dense | input: (None, 64) | output: (None, 2)