5.1) The dataset used was downloaded from kaggle as a .csv file and read into my python file. This data set had a total of 5000 entries which is of a reasonable size to get accurate predictions from our model. There were no empty rows in the column reviews.text, meaning all 5000 data entries were used in the model. The dataset is considered unlabeled since there is no clear indication if the customer gave a positive or negative review. There is a column to suggest if the customer would recommend it, but this doesn't necessarily mean that they liked or disliked it. Also, there is a numerical rating column, but this too may be misleading as a customer may leave a neutral review and still give the product five stars. Therefore the data was treated as unlabeled.

5.2) The first step to cleaning the data was to drop any missing rows from the column reviews.text column. This could have affected the accuracy in our results so was removed. Then the text was passed through the nlp model. After passing through this, tokens were cleaned by using .lower(), .strip() and .is_stop() to convert all non-stopwords to lowercase and to strip any whitespace in the words too. Once this was completed, the cleaned tokens were then joined together again and passed through the nlp model.

5.3) To evaluate the results, we will need to use visual inspection. Numerical methods would be preferred, however we don't have ground truths to compare our results too. Using the visual inspection method, we see that the majority of results are predicted correctly. The positive reviews are returned positive, the negative ones are returned as negative and the model even returns neutral reviews as neutral, which was expected. However, there is one review up for contention which is the last one. This is returned as neutral, however, "would recommend to a friend" is generally considered to be a positive review, although it's clear to see how this might be misleading. Overall, the model seems to have good accuracy for determining sentiment on product reviews.

5.4) The main strength of this model is its prediction accuracy. From a quick visual inspection, we see that the model does really well with predicting sentiment. However, there are a few things that could be improved. For example, the model currently takes a few seconds to run. Although this isn't a major impediment, it's still possible to reduce the running time and get the same results. This could be done by passing through fewer results into the model which will greatly reduce run time.

It would also be good to label our data with ground truths. This would allow us to accurately examine the model's outputs and the error in our model. Doing this would take some time, but it would improve the results we return.

The model also struggles with determining context in short sentences. This is evident with the last sample review we passed through it. To improve our results here, it would be suggested to use a different model type such as Long-Short term memory or a transformer so context can be more accurately predicted.