I have used the “Bert extractive summarizer” to help me perform text summarization.

To execute extractive summarizations, Bert's extractive summarizer makes use of the HuggingFace Pytorch transformers library. This is accomplished by first embedding the sentences and then performing a clustering technique to discover the sentences closest to the cluster's centroids.

After performing the sentiment analysis, I have separated my dataset based on different sentiments and the cyber security devices. I.e., for example- BlinkMini is a cyber security device for which we have performed sentiment analysis too. After that, I divided the BlinkMini data based on the sentiment of the review.

After separating the dataset based on sentiment, I have performed text summarization to understand why the users feel so positive or negative about that particular device. That is, what characteristics does that specific device have that people enjoy, and what features do not have that make the product less pleasant.

Before performing text summarization, I have performed a few preprocessing tasks. I.e., I have started off by removing special characters except (.), (!), because these special characters are added to represent an end of a sentence and by removing them the model wouldn’t understand where a particular review ends and hence displays all reviews as a summarized text. I then have removed all the emojis present in the reviews. Then, implemented contraction mapping to expand all the short forms such as ain’t to is not, etc.

After the preprocessing phase, I started to implement the text summarization. Since we had a lot of reviews, I used a few hyperparameters to generate perfectly short summarized reviews. I.e., I used the max\_length to be 200, min\_length to be 30, and num\_sentences = 20.