The evaluation process encountered significant challenges due to inconsistencies in tagging conventions and inaccuracies from poorly tagged data, which led to smaller than expected sample sizes.

During the pre-processing phase, I produced multiple versions of the review texts for my opinion mining process. To determine the most effective string combination, I assessed each version by applying a scoring formula: (number of feature matches to tagged data multiplied by accuracy), run by the `show\_optimum\_string\_variables` function. The optimal combinations identified were the \*\*Lemmatized Review String\*\* for feature extraction and the \*\*Soft Filtered Review String\*\* for sentiment analysis.

Furthermore, I conducted a grid search to fine-tune similarity filters for my word2vec and GloVe product feature models. A threshold of 0.25 proved to yield good results in terms of feature relevance.

Below, I provide a direct comparison of my two feature extraction models. The dependency model surpassed the POS model in terms of accuracy, recall, and F1 score, although it did not perform as well in precision. This was close to in line with my expectations.

Similarly, I compared two sentiment detection models: SentiWordNet and VADER, across three sample files. Contrary to my expectations, VADER did not surpass SentiWordNet in recall, accuracy, or F1 score, although it did in precision. It appears that the strengths of VADER's model, typically suited for social media posts, did not extend effectively to this set of customer reviews.

My attempts to build a multi-feature ML classifier yielded disappointing results. The confusion matrix showed only 7 feature matches with the tagged data, an insufficient sample to derive reliable conclusions. I attribute this to the considerable sparsity in my feature matrix and the limited sample size.