Phase 4 Part B: Summary

We created a Decision Tree classifier, a Gradient Boosting classifier, and a Random Forest classifier. We used a TF-IDF vectorizer to vectorize all the review_texts in our dataset because none of these models take strings as input. We then split our dataset into training and testing sets with a 75:25 split ratio. We first generated the models without using any sampling strategies, which meant that we trained the models on an imbalanced dataset. As indicated in the previous phases, our dataset is highly skewed toward positive (particularly 5 star) reviews. Our training and testing results were quite poor. The Gradient Boosting classifier took the most time to construct by a significant margin, but it performed the best across all scores.

No Sampling Results

Model	Accuracy	Precision	Recall	Time Taken to Construct Model
Decision Tree	0.4844	0.4875	0.4844	4.3144 mins
Gradient Boosting	**0.5986**	**0.5410**	**0.5986**	18.7993 hours
Random Forest	0.5664	0.4803	0.5664	5.95 mins

We retrained the models after adding an additional step to the training/testing process to reduce the dataset's imbalance: undersample the dataset's majority classes and oversample its majority classes. Excluding the precision scores, all scores decreased to nearly the same value; the time required to construct each model practically remained unchanged. The precision scores of the Gradient Boosting and Random Forest classifiers increased, meaning these models were now more consistent when making predictions; however their predictions, on the whole, were still incorrect.

With Sampling Results

Model	Accuracy	Precision	Recall	Time Taken to Construct Model
Decision Tree	0.2732	0.4947	0.2732	4.3507 mins
Gradient Boosting	**0.2875**	0.6381	**0.2875**	17.7269 hours
Random Forest	0.2744	**0.7309**	0.2744	4.3279 mins

The sampling approaches helped capture these models' true predictive abilities. About half of their predictions were correct when they did not use sampling techniques, but this was most likely due to the dataset's imbalanced skew. Because the data was heavily skewed towards positive ratings (there were so many positive reviews in the dataset's distribution), the first models were more likely to want to classify a review as positive; they did not truly "learn" what made each class of review different.

We attribute most of these models' poor performances on how they are not NLP models.