Emotion Classification Report

- Hyperparameters: max_features = 500, stop_words = ‘english, and n_jobs = 1
- Dimensionality of the dataset is its size
- max_features=500: Limits the number of features (i.e., terms) to the top 500 that appear most frequently in the corpus. This helps in reducing the dimensionality of the feature space, focusing on the most relevant terms and improving computational efficiency.
- stop_words='english': Excludes English stop words (commonly used words such as "the", "is", "in", which do not add much meaning to the text) from the analysis. Removing stop words can help in focusing on more meaningful terms for text analysis and machine learning tasks.
-

Naïve Bayes

- Best cross-validation accuracy: 0.92
- Time taken for GridSearchCV: 77 minutes
- 'classifier__alpha': [0.01, 0.1, 1.0, 10.0, 100.0].
- “alpha” in this case means smoothing. A smaller smoothening has the algorithm trust the training data more. This can lead to overfitting if the data is too noisy and if it trusts it too closely. A larger smoothening has the algorithm trust the priors more, leading to less overfitting.
- Originally created a model that resulted in the Grid search being unable to find the sentiment module. Different hyperparameters were unable to be used in such an approach. We then created another model where we used param_grid to take in the different hyperparameters, and the Grid search was able to properly operate.

Logistic Regression

- Best accuracy: 0.92
- Time taken for GridSearchCV: 79 minutes
- Had five configurations with the similar set of hyperparameters, but each one's clf_max_iter with it increasing from config 1 through config 5. What was found was only a very miniscule 0.01 difference in a decrease from the accuracy score of config 1, 0.928%, wich each other one, which was 0.927%.

Decision Tree

- The main issue was finding the right hyperparameters so that the search wouldn't take a monstrously long time.

Random Forest

- 'max_depth': [10, 10, 10, 10], to prevent the desicion trees from taking up too much space when searching and to prevent overfitting
- 'max_features': ['sqrt', 'log2'],
-
- Incorrectly used 'auto' as a hyperparameter for 'max_features' set. Also didn't utilize n_jobs, which meant the algorithm used one core of the CPU, rendering the search time much slower.

Grid Search (Decision Tree)

- decisiontree__max_depth': [10, 50, 100, 200]
  `decisiontree__min_samples_split': [2, 50, 100]`
  `decisiontree__min_samples_leaf': [1, 20, 50]`

- max_depth here was especially fine-tuned here with min_samples_split and min_samples_leaf to prevent overfitting the model given the size of the dataset.

Grid Search (Cat Boost)

- catboost__learning_rate': [0.01, 0.05, 0.1, 0.2]
- The learning rate refers to the step size shrinkage