**Software Implementation:**

The implementation for this program comes in two parts. We decided to use a Naive Bayes methodology, which starts with a training method and then uses that data to predict classifications onto a testing dataset. We use a bag of words model in which we consider each tweet to be represented as a bag of independent words, meaning that we will ignore the position that the words appear and only deal with the frequency that they appear.

Using a unigram model, the program first calls the training function found in training.py, which goes through the training data given and creates a bag of words model. Because each tweet is pre labeled as either sarcasm or not sarcasm, we can treat each tweet as a list of words using the NLTK TweetTokenizer. This method returns a list of sarcastic words and their frequencies, a list of non-sarcastic words and their frequencies, as well as totals for both sarcastic and non sarcastic words.

After that, during the classification phase, the program calculates the probability for each tweet to be sarcastic or non-sarcastic based on the probabilities that were developed from the training set. The program develops a posterior probability for each tweet, and based on which is higher, classifies the tweet as either sarcastic or non-sarcastic. In order to avoid zero probabilities, we have a smoothing parameter in place, which has been optimized to give the best results.

**Installation and running:**

To install the software, first clone this git repository: <https://github.com/shail4221/Classification-Competition.git>

Once finished, you can run the program inside by running *python classify.py*. This will train the data on the train.jsonl file, run the classification on the test.jsonl file, and output the results into answer.txt

**Description of contribution:**

Both members worked together mostly equally, with Xuechen focusing primarily on the training side of the program and Shail focusing on the classification side. Both team members worked together to refine and test the program for accuracy.