<u>Documentation for argumentation mining:</u>

<u>Approach:</u>

According to [1], claims in sentences have a similar parse tree structure. Although [1] suggests a SVM with partial tree kernel for claim detection, it can be noted that Bag of Words (BoW) also gives a reasonable F1-score. Thus, we used parts of sentences (POS) and BoW as a feature and checked it in SVM as well as Naïve Bayes classifiers.

<u>Feature Engineering</u>:

First, we read the training set and test set, store their length and combine them into one corpus to apply various extraction techniques.

We then tokenized the sentence and removed the text of punctuation marks, white spaces, URL and stop words in each sentence.

Then we applied stemming to the tokens to get the root form of the word.

Finally, we used POS tagging that allocates parts of speech to each word, such as nouns, verb, adjectives etc.

We used this as a pre-processor and applied it in the sentences passed to the CountVectorizer function

<u>Classification</u>:

We split the entire corpus back to training and test sets from their respective lengths, without shuffling them. Thus, we extract back the original training and test sets from the corpus.

As, we were categorizing the data into two labels: 1 or 0, with 1 indicating a claim and 0 indicating not a claim, we can see that binary classification is required. Hence, Support Vector Machine (SVM) and Multinomial Naïve Bayes (MultinomialNB) were the two classifiers we tested. And we found that both the classifiers are very quick, gives a high score and has no overfitting problems. Naïve Bayes produced the highest F1 score of any of the combinations we tested (See Table 1).

MODEL	ACCURACY	PRECISION	RECALL	F1 SCORE
SVM (kernel = “linear”) + FE + CV	0.59	0.4	0.29	0.33
MultinomialNB + FE + CV	0.58	0.41	0.33	0.36
SVM (kernel = “linear”) + FE + TF-IDF	0.63	0.48	0.19	0.27
MultinomialNB + FE + TF-IDF	0.63	0.47	0.07	0.12
SVM (kernel = “linear”) + FE + Stemming + TF-IDF	0.67	0.63	0.24	0.35
MultinomialNB + FE + Stemming + CV	0.61	0.45	0.35	0.39
MultinomialNB + FE + Stemming + POS + CV	0.62	0.47	0.37	0.41

SVM (kernel = “linear”) + FE + Stemming + POS + CV	0.62	0.46	0.34	0.39
SVM (kernel = “linear”, gamma = ”scale”, C = 100.0) + FE + Stemming + POS + CV	0.62	0.48	0.47	0.47
MultinomialNB (alpha =0.1) + FE + Stemming + POS + CV	0.61	0.46	0.5	0.48

Table 1: Comparison of Various Classifiers + Features Combination

SVM = Support Vector Machine; FE = Feature Engineering such as removal of punctuation, spaces, URL and stop words; CV = CountVectorizer for Bag of Words; TF-IDF = Term Frequency – Inverse Document Frequency; POS – Part of Speech

We came up with the hyperparameters for both the classifiers by cross validation using GridSearchCV function. After cross validation, we found that Naïve Bayes yields slightly better f1-score than SVM but both give a better f1-score than 0.45.

Generating Output File:

After classification is completed, we output the evaluation result on the console.

Then we take the predicted values and map it to the text id and write them in the desired format to ***result.json*** file

Steps:

To run classifier:

python arg_mining.py -x <TRAINING SET> -y <TEST SET>

Output generated by the classifier in: **result.json**

To run eval.py:

python eval.py -t <TEST SET> -p result.json

References:

[1] M. Lippi and P. Torroni, “Context-Independent Claim Detection for Argument Mining,” Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, pp. 185–191, 2015.