CA assignment 3

Features

Bag of words: BOW is a vocabulary of known words and a measure of the presence of known words in turn helps us convert text sentences into numeric vectors. We used CountVectorizer () which creates a matrix of the same. We added this feature to reduce the complexity of sending whole text with numeric values

TF-IDF: finds meaning of sentences consisting of words and cancels out the incapabilities of Bag of Words technique which helps in text classification or for helping a machine read words in numbers. We used (TfidfVectorizer(min_df=2, max_df=0.75, ngram_range=(1,1))) to do the same.

ngram_range : tuple (min_n, max_n), default= (1, 1)
The lower and upper boundary of the range of n-values for different n-grams to be extracted. All values of n such that min_n <= n <= max_n will be used. For example, a ngram_range of (1, 1) means only unigrams, (1, 2) means unigrams and bigrams, and (2, 2) means only bigrams. Only applies if analyzer is not callable.

These features represent the argumentative knowledge of the text given to us.

Models

AdaBooster: Uses an Ensemble Method. Used mainly to classify the text. We used the model with paratmeters(n_estimators=5000, learning_rate=1) and gave us f1 score of **0.72**

NaiveBayes: a probabilistic classifier which uses probabilistic distribution to classify the texts. We got a score of around **0.68**

RandomForestClassifer: basically, a set of decision trees from a randomly selected subset of the training set. And votes for the final decision. We got a score of **0.75**

SVM: classifies or separates data using hyperplanes and does not require any parameter tuning.We got a score of **0.789**

Out of all the models **SVM** model gave the best result for our given corpus

F-1 Scores using different models

Model	F-1 Score
AdaBooster	0.72
NaiveBayes	0.68
RandomForestClassifer	0.745
SVM	0.789

Hyperparameter

We used the **GridSearchCV** method to perform the Gridsearch and perform cross validation on the model

It takes the parameters as follows:

1. Model to be used: clf
2. param_grid = {'C':[0.1, 1, 10, 100, 1000],'gamma':[1, 0.1, 0.01, 0.001, 0.0001], 'kernel':['linear','rbf']}
3. refit = True
4. verbose=3,
5. cv = 5,
6. return_train_score=True,
7. n_jobs=-1

Using GridSearch we found the best parameters for our SVM model (We print the best parameters out at the end of the code as well)

SVC (C=10, gamma=0.0001, verbose=True, kernel="rbf")

And the best feature which gave the good score was when we used BOW (bag of words) Parameter