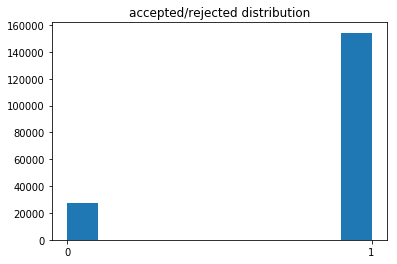
Problem Description:

Given essays to various submitted essays, can an algorithm be trained to predict whether or not an essay application will be accepted or rejected.

Experimental Results:



MLP Classifier

Word Embeddings & No preprocessing:

F1 score: 0.920

ROC AUC score: .501

Accuracy score: .854

Best parameters: {'activation': 'relu', 'hidden\_layer\_sizes': (10, 10), 'solver': 'lbfgs'}

Word Embeddings & Normalization preprocessing:

F1 score: 0.923

ROC AUC score: .499

Accuracy score: .849

Best parameters: {'mlpclassifier\_\_activation': 'relu', 'mlpclassifier\_\_hidden\_layer\_sizes': (100, 100), 'mlpclassifier\_\_solver': 'adam'}

Data Frame on back

TFIDF Vectorizer & KMEANS:

F1 score: 0.903

ROC AUC score: 0.582

Accuracy score: 0.819

MLP Regressor

TFIDF Vectorizer & KMEANS:

F1 score: 0.924

ROC AUC score: 0.623

Accuracy score: 0.848

Analysis of results:

The data given had a large imbalance, which can lead to scoring that may show an algorithm performed better than it did. As such running different algorithms and getting different scores can help to determine that actual performance of an algorithm. In the first run through each essay was given a vector representing the average of each of the word vectors for every word in the essay. This was done for both essays resulting in a data set of 182080 entries with 100 attributes. This was run through a MLPC which returned an f1 score of 92% and ROC AUC score of 50%. The ROC AUC score shows that even though the algorithm works very well, the same could be expected from a majority class predictor. A second approach was then taken using TF-IDF vectorizer with KMEANS clustering. This method was run through both a classification MLP and regressor. The classifier returned similar results to that of the embedding classifier. The improvement in ROC AUC is significant as that change is more meaningful than the change in f1 score, as the ROC AUC score shows that the classifier is slightly more effective than that of a majority class classifier by having more true positives than false positives. The regressor however did much better receiving the same score as the normalized word embeddings, but by getting a much better ROC AUC score. This could be because instead of assigning discrete class values like the classifier, the regressor assigns continuous values that could be analogous to a probability of an instance belonging to a specific class. When binned to be discrete, this probability can be considered to make better predictions.

Conclusion:

The data sets unbalanced nature shows the importance of understanding what is happening when an algorithm is learning a set of data, and what the score given at the end means in order to make any meaningful assertions. While the f1 score given for all of the algorithms run where in the 90s, the ROC AUC score shows that the predictions being made are the equivalent of majority class predictions. Only the TFIDF vectorized kmeans mlpc regressor was able to break fifty percent and achieve a greater amount of true positive predictions to false positive predictions. This difference is what made the vectorized approach the best predictor out of the methods used and shows how different methods of approaching a problem are important, and that exploring those methods is key to be able to find the best approach to solving a problem. With the correct evaluation metric to understand which of the approaches is best, it is possible to increase the effectiveness of the best method, and perhaps even understand why the best method outperforms other similar approaches which allows for much better understanding of the problem at hand.