Kavin Elamurugan, Nick Abadiotakis, Cole Kandel

CS301

Professor Pantelis

4/22/2021

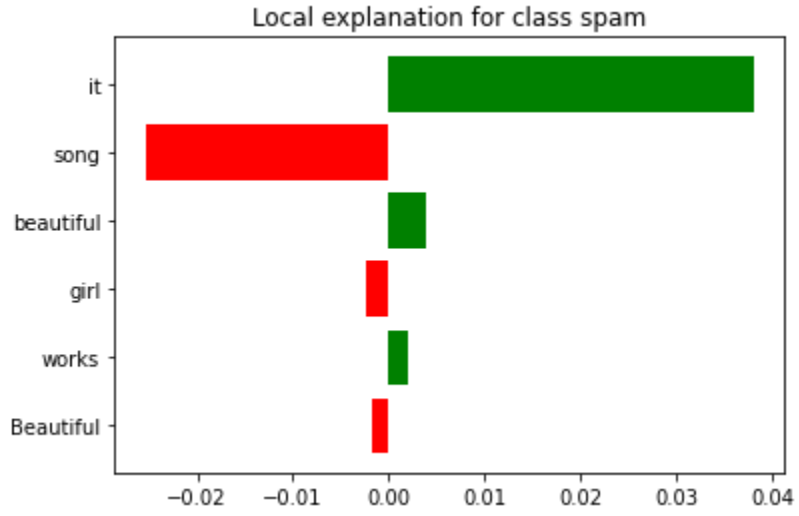
Final Project

We chose to use a 2-class subset imported from the Youtube dataset, spam and ham, and for classification we decided to use the random forest classifier. In order to do this we used the tfidf vectorizer for the text data that Converts a collection of raw documents to a matrix of TF-IDF features. When using the random forest classifier, although it may seem difficult to the naked eye to understand what is happening with the code, we know that it is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. We then used the sklearn f1_score metric to determine the F score, a weighted average of the precision and recall(result relevancy and number of relevant results), where an F1 score reaches its best value at 1 and worst score at 0. We found ours to be 0.9908256880733944, a very high F-score, indicating that Multinomial Naive Bayes overfits this dataset by learning irrelevant stuff, such as headers.

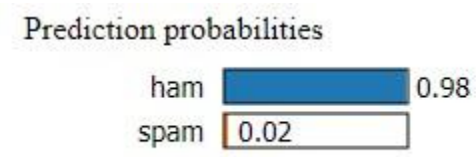
Next, we used LIME to predict the models accuracy on a real document. In order to do this we use sklearn's pipeline, and implement predict_proba and then print out the probability predictions. We then create an explainer object and pass the class_names as an argument in order to have a more understandable display. After that we can generate an explanation with 6 features for an arbitrary document in the test set. We print out the Document ID(83) and Probability of Spam(0.022). This means that the model found the probability the document was spam to be 2.2%, therefore there is a 97.8% probability the document is ham . We could also print out a list

of weighted features in the document, with a negative value meaning ham and positive meaning spam, which weighed 'it' 0.038121474784150934, 'song' -0.025428885902265864, 'beautiful', 0.0038647980489139736 , 'girl', -0.002341759687761373, 'works', 0.0020451377464024098, 'Beautiful', -0.0016872794850452469. These weighted features are a linear model, which approximates the behaviour of the random forest classifier in the vicinity of the test example. We then thought to remove ‘song’ and ‘beautiful’ from the document, theoretically meaning that the model would predict more towards the spam class by about .054, which was the sum of those features weights. After running another prediction, the probability increased from .022 to .078, a difference of .054, which was our prediction.

We also exported our explanations as a html page, rendered in the document with images showing the local explanation for class spam for each feature,



as well as the prediction probabilities for the document as a whole,



the prediction probabilities for each feature,

as well as the visualization of the original document with the words in the explanations highlighted.

As a conclusion, we learned about using LIME and all its different tools that are offered about how it is useful to implement classifiers on documents. We also learned about random forests and how this explainer works for any classifier you may want to use, as long as you implement predict_proba into your model.