PART 4:

From this project, we have learned a lot about the process of building a classifier and some of the challenges that come with its construction. Although we were fortunate to have the YouTube comments on LMFAO, Katy Perry, Shakira, Eminem, and Psy music videos, we still had to clean the data. The process of data cleaning and the overall preprocessing of the data was a tedious step, but necessary to call and train our classifiers for each artist's YouTube comments. However, as highlighted in the *TubeSpam: Comment Spam Filtering on YouTube* paper authored by Alberto et. al., the business model of YouTube requires that any spam filtering performed by a well-trained classifier on its site must be very good at differentiating Spam vs Authentic content in order to avoid disenfranchising users. Therefore, in an effort to improve our knowledge on whether or not the created classifier's are good enough to move into production, we implement the LIME methodology to provide feature explanations on our Spam filtering classifiers.

The classifiers trained for each Artist's spam filtering used either the Random Forest model or the Multinomial Naive Bayes approach. Following preprocessing, we apply the vectorizer to tokenize the text, learn the vocabulary, and measure word frequency simultaneously. At this point, the model is then fitted and transformed.

Following vectorization and transformation, we apply the classification technique of our choice. Our group had the choice of applying Decision Trees, a Logistic Regression and Support Vector Machines, among others, but ultimately leveraged the Multinomial Naive Bayes approach and Random Forest approaches for our project. To examine the quality of our Multinomial Naive Bayes or Random Forest classifier, we calculate the accuracy and the Spam Caught rate for each model before applying LIME. Although these calculations help us gauge the quality of the predictions generated by our classifiers (i.e., Spam vs. Not Spam), we still do not know the factors driving the predictions of our model, which is why we need to apply the LIME methodology.

After importing all necessary libraries, we take our vectorized data and the classifier we just trained and use sklearn's make_pipeline function and implement predict_proba on a raw text list to accommodate the fact that LIME explainers assume classifiers act on raw text while sklearn assumes classifiers act on vectorized text. What is returned is a decimal representing the probability that a comment under the CONTENT column is classified as not-spam (spam=1, not-spam=0).

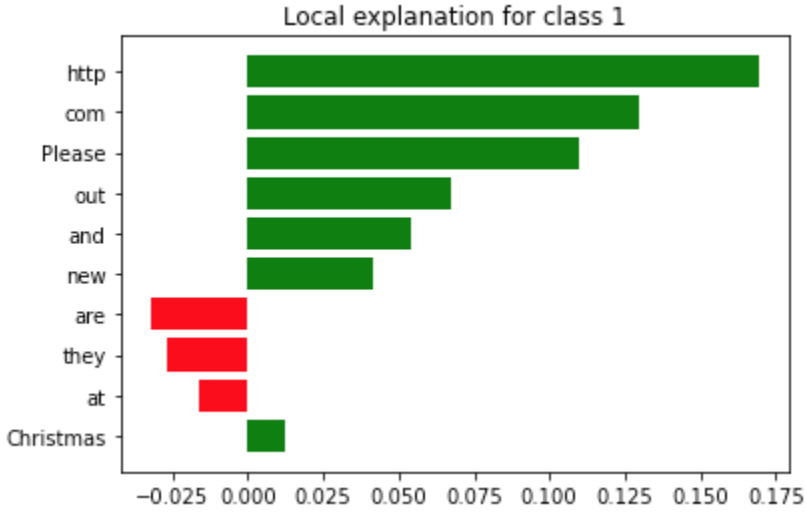
The next step for implementing LIME is to create an explainer object. Due to the fact that we are filtering YouTube comments, we use the LimeTextExplainer to provide a representative set of factor explanations so the user can obtain a Global Understanding of the model. In theory, this approach runs contrary to one of our desired characteristics of a classifier which is that an explanation must be locally faithful. However, the LIME method attempts to explain the model globally. In its entirety, this necessary step within the implementation of LIME accommodates the conflict between the goal of the LIME method and the desired characteristics of an explainer.

Finally, we test the LIME model on our processed data (Psy, Eminem, Shakira, LMFAO, or Katy Perry) by calling the the explainer.explain_instance function on our CONTENT column, the c.predict_proba soft classifier, and the num_features (user specified) to update the explainer

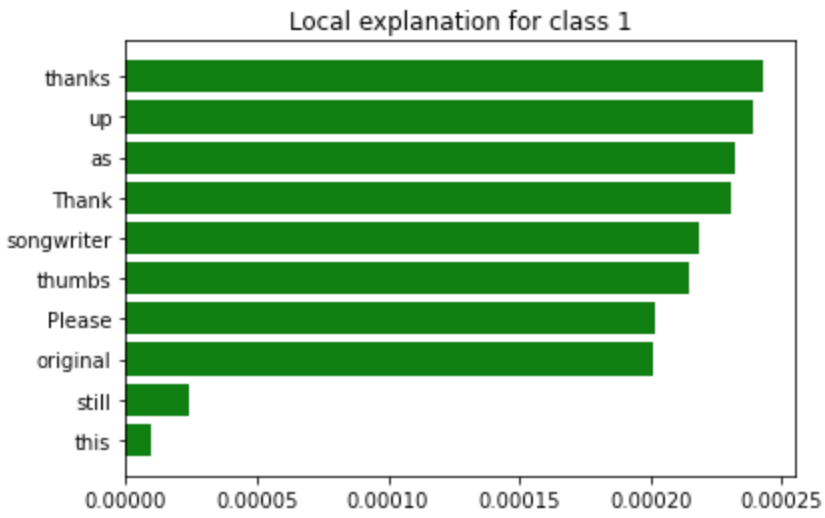
used by the LIME methodology. Converting this explainer to a list returns our feature explanations for our classifier. In this case, we are presented with the most heavily weighted words used to predict whether a YouTube comment is classified as Spam or Not Spam. Analyzing these feature explanations provides us with additional insight into whether our classifier's are sufficient to move into production.

Looking at the below feature explanations for each Artist, we may consider editing our code to exclude stopwords, to switch the classification method used, or to incorporate Lemmatization or Stemming of user comments. On that note, we suspect that the classifier's we created could be improved and that YouTube's team of developers likely have a Spam-filtering classifier that easily outperforms our own. However, we believe that the provided feature explanations validate the capacity of our classifier's to filter spam.

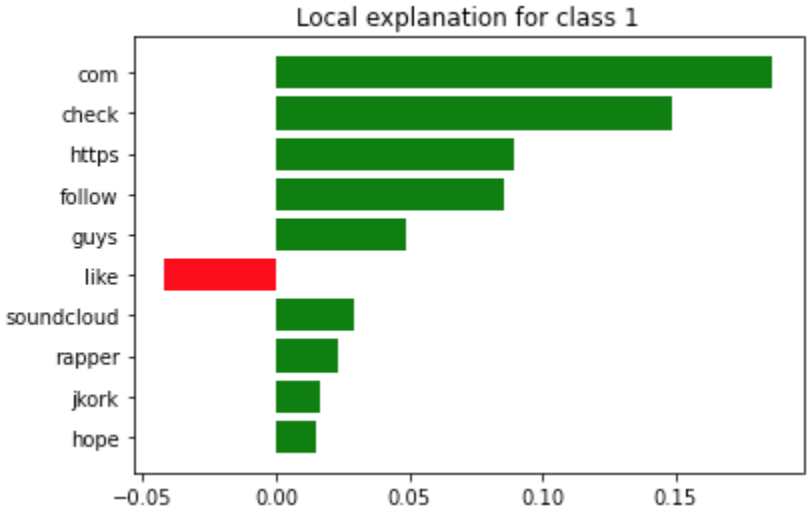
Psy Feature Explanations:



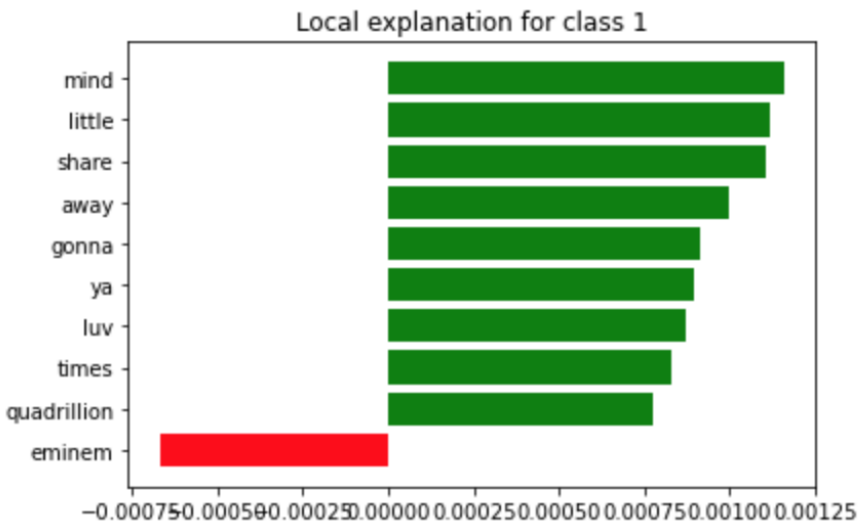
Shakira Feature Explanations:



Katy Perry Feature Explanations:



Eminem Feature Explanations:


Local explanation for class 1
mind
little
share
away
gonna
ya
luv
times
quadrillion
eminem
−0.00075
−0.00050
−0.00025
0.00000
0.00025
0.00050
0.00075
0.00100
0.00125