James Podeszfinski, Avinash Kumar, Mike Jeong

Professor Pantelis Monogioudis

CS 301 Section 004

April 9 2021

Summary

For the purposes of this paper, we will be talking about the tubespam dataset as provided by Professor Monogioudis in the project description. The classifier that we used for part 3 of the project was the Random Forest classifier. We felt that this would be a good classifier to use since we already had experience with it from the previous step, and so it would be a good choice for us to implement it again for this part of the project. We also decided that the Random Forest classifier was highly effective and easy to use, which also helped our decision.

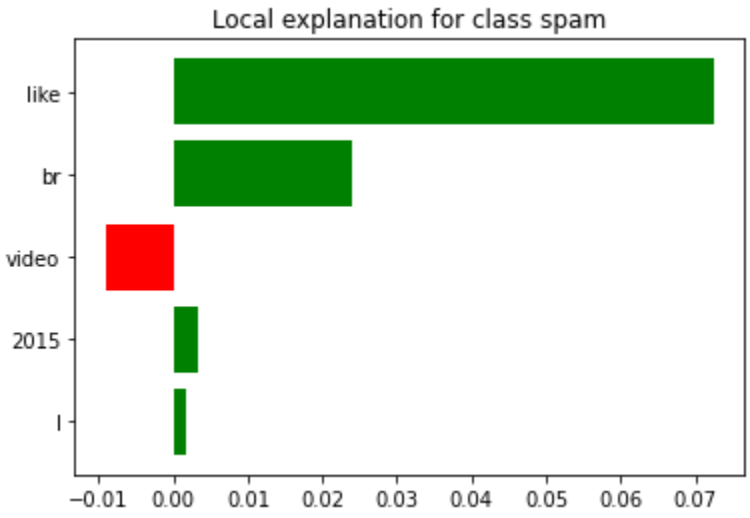
To start off, the first thing that we did as a group to solve this part of the project was to successfully parse the csv files embedded in the tubespam dataset. In order to do this, we wrote code that parsed through two of the csv files, the ones labeled “Youtube01-Psy.csv” and “Youtube02-KatyPerry.csv” and we used their contents to fill our training dataset. If the comment corresponded to spam, we appended the contents and the classifier of that example into the list spam. If the inverse was true, we appended the contents and the classifier to the “not_spam” list. We combined the contents of “spam” and “not_spam’ into a list called “file_in” and then we parsed through “file_in” to append appropriately to our lists that we used for the training of our model. After this, we used a third csv file, the one labeled “Youtube03-LMFAO.csv” as something that we could test the results of our training on, and we appended that contents of that file into our training datasets.

After we successfully prepared testing and training datasets, the next step was to import the sklearn library in Python, and use the Random Forest library that it contained within it. We used this library to classify whether a comment in the "Youtube03-LMFAO.csv" file was spam or not spam. After writing the appropriate code for this part, we were able to determine that the model we had implemented had an "f-score" of 90.9%, which is rather high for a Random Forest model. Due to this, we were able to determine that our model of implementing the Random Forest classifier was successful and that part 3 of the project had been completed.

After we were sure that our use of the Random Forest classifier was working to determine whether a comment was or was not a spam comment, we then decided to implement the LIME explainer, to try and get more insight as to what was going on inside of our classifier. We imported the lime explainer from the LimeTextExplainer library, and continued to work on our model. We chose to use a random line from the "Youtube03-LMFAO.csv" file, and the line contained the comment "2015
 I like video". The comment is listed inside of the "Youtube03-LMFAO.csv" as not a spam comment. We decided that this comment would be a good comment to use for testing purposes since it appears like a spam comment, but it is in fact not. We in a way tried to deceive the Random Forest classifier to see if it would wrongly write this off as spam, and we also decided that this would be a good comment to have the LIME Explainer show us what the Random Forest classifier was thinking in regards to how the Random Forest classifier makes decisions.

After implementing the LIME explainer we got the following results. The Random Forest model correctly predicted that this comment was not a spam comment, and the results from the LIME explainer helped us to understand how the model was thinking. From the training datasets, the Random Forest model that we used was able to determine that the words "like", "br",

“2015”, and “I” were not indicative of spam, and the only word that threw off this thinking was the use of the word “video”. However, due to this only throwing off the model slightly, it was still able to detect that the comment was in fact not spam and it estimated that there was a 90% chance of it not being spam. Even after removing the words “br” and “2015” from being used as a reference, the Random Forest model was still able to correctly determine that the comment was not spam.



This is a chart taken from our project, detailing the thoughts of the Random Forest classifier. If a word has a bar in green, that means that this word is indicative of not being used in a spam comment. If the bar is in red, that means that the word is indicative of being used in a spam comment. As you can see, the Random Forest classifier has classified the words of “like”, “br”, “2015”, and “I” as being indicative of a comment not being spam, and the only word to throw off this thinking is the use of the word “video”, but even that has a relatively little impact on how confident the classifier is in making a decision, only throwing the classifier off by less than 10%.

All in all, Random Forest as a classifier seems to be highly effective. It is very simple to implement, but also gives accurate results in a timely manner. The strength of Random Forest likely comes from its ability to combine a bunch of weak learning methods in order to create a strong prediction, by almost taking a vote of the number of methods that decided whether something is true or false. As a project, we as a group definitely felt that this experience helped us to learn about how classifiers worked and also allowed us to glimpse into how the “thought processing” of classifiers works and we thoroughly enjoyed working on it.