In Class Activity

Text classification with Naïve Bayesian Model – Roadmap Activity 4B

So, at the end of last class, what I really heard everybody saying was, we want a more complicated example. We can always make things more complicated!

What about “nonsense” words? What happens when there is no matching word found… that is… the user provided a nonsense word such as “Sharknado” or no word at all?

This is what the big kids call a “zero vector” when talking about vectorized text.

Given a zero vector (or a very sparse vector if the nonsense word is accompanied by other words not in the vocabulary), the Naive Bayes classifier will still make a prediction.

However, this prediction will be based solely on the prior probabilities of each class (i.e., the overall distribution of classes in the training data) because the input does not provide any additional information to sway the prediction towards one class or another.

Essentially, the model will default to its "best guess" based on the most common class or a distribution it learned during training.

So, if most of the training documents went to “Customer Support” then the text will be assigned to customer support.

The practical outcome is that the prediction for a nonsense word input will not be meaningful or based on any real understanding of the input text. It highlights a limitation of such models when encountering out-of-vocabulary (OOV) words or phrases.

To improve handling of OOV words or enhance the model's ability to deal with unexpected inputs, you might consider techniques like word embeddings (which can capture semantic similarities between words, including those not seen during training) or incorporating a wider variety of preprocessing steps to better manage rare or unknown words.

Side note: It was also asked if there was any limit to the amount of text we could enter in to our script… and the short answer to that is that there is no *practical* limit.

Instructions

1. Students should visit http://tinyurl.com/shaferaicourse and download the files found in the **roadmap4** folder. (There are three files, the Jupyter notebook, a csv file, and an excel file.)
2. Run the new and improved script. Your instructor will highlight its new and improved features.
3. Break into groups of three. Working in groups of three use our classification script to find the data in the excel spreadsheet. You will need to clean that spreadsheet a bit, change its data format to csv, and also identify which columns should be used for the “document” and the “classification”

4. After you get the script running, examine the reports generated by the script. Use those to add new and improved custom “Stop Words”
5. Rerun your analysis and test it. Is your classification model better, or worse?
6. When you are done, every student should **find the relevant discussion to post to on canvas.**

   Share what your custom “stop words” were and if you were able to improve the accuracy of the model.