Text Classification

 (https://github.com/learn-co-curriculum/dsc-text-classification)  (https://github.com/learn-co-curriculum/dsc-text-classification/issues/new)

Introduction

In this lesson, we'll discuss the general process for setting up text datasets for classification problems.

Objectives

You will be able to:

- List the steps for classifying text data

Classification for Text Data

For the final lab of this section, we'll use everything we've learned so far to build a classifier that works well with text data. As you've probably guessed, text data is significantly harder to work with than most traditional datasets, because of the sheer amount of preprocessing needed to get the data into a format acceptable to a machine learning algorithm.

The main challenge in working with text data isn't just the preprocessing -- its the number of decisions you have to make about how you'll clean and structure the data. In a traditional dataset full of numerical and categorical features, the preprocessing steps are fairly straightforward. Generally, we normalize the numeric data, check for and deal with multicollinearity, convert categorical data to numerical format through one-hot encoding, and so forth. Although the steps themselves may not be easy, there's generally little ambiguity about *what needs to be done*. Text data is a bit more ambiguous. Let's examine some of the decisions we generally need to make when working with text data.

Cleaning and Preprocessing Text Data

Once we have our data, the fun part begins. We'll need to begin by preprocessing and cleaning our text data. As you've seen throughout this section, preprocessing text data is a bit more challenging than working with more traditional data types because there's no clear-cut answer for exactly what sort of preprocessing and cleaning we need to do. When working with traditional datasets, our goals are generally pretty clear for this stage -- normalize and clean our numerical data, convert categorical data to a numeric format, check for and deal with multicollinearity, etc. The steps we take are largely dependent on what the data already looks like when we get a hold of it. Text data is different -- if we

inspect a raw text dataset, we'll generally see that it only has one dimension -- the actual text, in the form of a string. This could be anything from a tweet to a full novel. This means that we need to make some decisions about how to preprocess our data. Before we can begin cleaning and preprocessing our text data, we need to make some decisions about things such as:

- Do we remove stop words or not?
- Do we stem or lemmatize our text data, or leave the words as is?
- Is basic tokenization enough, or do we need to support special edge cases through the use of regex?
- Do we use the entire vocabulary, or just limit the model to a subset of the most frequently used words? If so, how many?
- Do we engineer other features, such as bigrams, or POS tags, or Mutual Information Scores?
- What sort of vectorization should we use in our model? Boolean Vectorization? Count Vectorization? TF-IDF? More advanced vectorization strategies such as Word2Vec?

These are all questions that we'll need to think about pretty much anytime we begin working with text data.

Feature Engineering

Another common decision point when working with text data is exactly what features to include in the dataset. As we saw in a previous lab, NLTK makes it quite easy to do things like generate part-of-speech tags for words, or create word or character-level n-grams. In general, there's no great answer for exactly which features will improve the performance of your model, and which won't. This means that your best bet is to experiment, and treat the entire project as an iterative process! When working with text data, don't be afraid to try modeling on alternative forms of the text data, such as bigrams or n-grams. Similarly, we encourage you to explore how adding in additional features such as POS tags or mutual information scores affect the overall model performance. Sometimes, it has a great effect on performance. Other times, not much. Either way, you won't know until you try!

Summary

In this lesson, we discussed the challenges that come with working with text data for classification, and the types of decisions we should be ready to make when cleaning and preprocessing a dataset!