Introduction

My goal is to learn basic concepts of Natural Language Processing and Data mining, the role that probability plays in language modelling. All code, images, and data for this project can be checked in Python notebook attached.

Project Description

This project involves five steps:

1. Data Pre-Processing

2. Language Modelling

3. Classification

4. Visualization

5. Evaluation of Results

Data Pre-Processing

Downloading Data

I use the open-source BBC Dataset. The dataset contains 2,225 articles from the BBC news. Each article is labelled with one of the following five classes: business, entertainment, politics, sport, and tech. I downloaded the dataset and wrote the script import-data. Python to output the data as a two-column data frame, with one row per article. The first column contained the document text, while the second column contained the class labels.

Splitting into Training and Testing

Next, I split the data frame into training and testing sets, randomly assigning 70% of the data to a “training” set and leaving the remaining 30% for “testing”. The purpose of this step is for classification. Classification models are fit to training data, which represents past observations, in order to predict the testing data, which would represent new observations. It is important to perform this step before creating a language model in order to avoid the mistake of training our model on data that is supposed to be unseen. Thus, I create two copies for each language model: one with 70% of the dataset, and another with only 30%.

Language Modelling

The goal of language modelling is to extract information from a corpus, or set of documents. One can engineer a variety of features from text, such as word length, or frequency of punctuation. The most descriptive and useful is term frequency, or the distribution of counts per word in each document. Because my corpus is quite large and it converted into binary format after removing all the stop words by Tfidfvectorizer.

Classification

After fitting a language model to the training and testing sets, I fit a multinomial Naïve Bayes classification model. Classification is the process of assigning a class to new data given a set of past entries. Our goal is to be able to read in a new document and, using a classification algorithm, predict what type of article the document is (business, sports, etc.).

Results

The table below shows the prediction accuracy of each language/classification model attempted.

```
[[150,   0,   3,   0,   3],
 [  0, 123,   1,   0,   2],
 [  7,   2, 110,   1,   0],
 [  0,   0,   0, 152,   0],
 [  5,   2,   1,   0, 106]],
```

Accuracy Score = 95.96%

Conclusion

I concluded that, with a 95.96% accuracy, the best way to classify the BBC dataset is to model it with naïve Bayes Bernoulli's Naïve Bayes classification algorithm. Thus, we see that a Natural Language Processing for classification can be improved with fitting different machine-learning algorithms, but that simply using an algorithm without paying attention to the quality of the language model proves futile.