Jon Rippe  
CSCE A415  
HW #2.1

## Problem 2: Spam Classifier

### Data Manipulation

The dataset was imported into a pandas DataFrame for easy manipulation (2). There was no missing data, so the entire set was used. The set was then converted to lowercase and stripped of English stop words, non-letters, and accents (6).

### Naïve Bayes Classification

Train/Test Score

90/10 0.9826164874551973

0.9689964157706095

80/20 0.9838565022421525

0.9688789237668161

70/30 0.9833133971291865

0.96872009569378

60/40 0.9848362494392104

0.9673844773441005

50/50 0.983776022972003

0.9642498205312275

40/60 0.9833133971291865

0.960047846889952

30/70 0.9817995385798515

0.9540374263009485

20/80 0.9800807537012114

0.9405114401076716

10/90 0.9759720837487537

0.9121036889332006

I attempted two approaches to classification:

1. A simple count of the words in each message.
2. A weighted count of the words in each message based on message length and average message length using a term-frequency transformer.

Each method was run through a series of train/test splits to determine accuracy (8).

### Results

Using a term-frequency transformer creates a less aggressive model, allowing more spam through the filter but providing zero false positives (9 – confMatrix). There may be a way to fine-tune the transformer to get better results.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | Predicted Class | | | |
|  |  | Model #1 | | Model #2 | |
|  |  | Ham | Spam | Ham | Spam |
| Actual  Class | Ham | 1197 | 9 | 1206 | 0 |
| Spam | 15 | 172 | 41 | 146 |

Confusion Matrix for an instance of each model.