Exercise 8 - Classification and Naive Bayes (part one)

First name: Brian

Last name: Schweigler

Matriculation number: 16-102-071

Q1: Apply the Naïve Bayes classifier to solve the authorship attribution problem related to the twelve disputed Federalist Papers (written by “Hamilton OR Madison”). You can use the 65 papers written by “Hamilton” (51) and “Madison” (14) to train your classifier and the disputed papers to evaluate your system. As features, you can use the following words: {“to”, “upon”, “would”}. For simplification, we consider only Hamilton or Madison as the possible authors of the disputed papers

General imports and solving the question:

In [1]:

```
%load_ext autoreload
%autoreload 2
%matplotlib inline

import matplotlib.pyplot as plt
import pandas as pd
import re
import numpy as np
import lxml.etree
import os
from scipy import stats
from sklearn.feature_extraction import text

np.random.seed(6)  # for reproducibility
df = pd.read_csv('Data/federalist-papersNew2.csv', index_col=0)
hamilton = df[df['AUTHOR'] == 'Hamilton']
madison = df[df['AUTHOR'] == 'Madison']

combined = pd.concat([hamilton, madison])
test_indices = [49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 62, 63]
test_set = df.loc[test_indices]
```

Essays where the author is known

In [2]:

```
df_known = df.loc[df['AUTHOR'].isin(('Hamilton', 'Madison'))]
print(df_known['AUTHOR'].value_counts())
```

```
Hamilton    51
Madison     14
Name: AUTHOR, dtype: int64
```

In [3]:

```
hamilton_short = hamilton[['what','to', 'would']]
madison_short = madison[['what','to', 'would']]
combined_short = combined[['what','to', 'would']]
```

Estimate probability of each word in vocabulary being used by Hamilton

In [4]:

```
fH = []
k = hamilton_short.sum(axis=0)
total_sum = sum(k)
for i in range(0, 3):
    prob = ((k[i] + 1) / (float(total_sum + len(hamilton_short))))
    fH.append(prob)
fH
```

Out[4]:

```
[0.02574430823117338, 0.8029772329246935, 0.1628721541155867]
```

Estimate probability of each word in vocabulary being used by Madison

In [5]:

```
fM = []
k = madison_short.sum(axis=0)
total_sum = sum(k)
for i in range(0, 3):
    prob = ((k[i] + 1) / float(total_sum + len(madison_short)))
    fM.append(prob)
fM
```

Out[5]:

```
[0.02979011509817197, 0.8544346648612051, 0.1083276912660799]
```

Compute ratio of these probabilities ('what', 'to', 'would')

In [6]:

```
fratio = [a / b for a, b in zip(fH, fM)]
fratio
```

Out[6]:

```
[0.8641896194873427, 0.939776048359566, 1.5035135726795097]
```

Compute prior probabilities

In [7]:

```
piH = len(hamilton_short) / float(len(combined))
piH
```

Out[7]:

```
0.7846153846153846
```

In [8]:

```
piM = len(madison_short) / float(len(combined))
piM
```

Out[8]:

```
0.2153846153846154
```

Next we iterate over disputed sets and try to figure out which author to attribute them to

In [9]:

```
h_count = 0
m_count = 0
for doc in range(0, len(test_set)):
    # Compute likelihood ratio for Naive Bayes model
    tmp = [np.power(a, b) for a, b in zip(fratio, test_set.iloc[doc])]
    tmp = np.prod(np.array(tmp))
    LR = tmp * (piH) / (piM)
    print(LR)
    if LR > 0.5:
        h_count = h_count + 1
        # print('Hamilton')
    else:
        m_count = m_count + 1
        # print('Madison')
```

```
3.642857142857143
3.642857142857143
3.642857142857143
3.642857142857143
3.642857142857143
3.642857142857143
3.642857142857143
3.4234698904527043
3.642857142857143
3.4234698904527043
3.642857142857143
3.642857142857143
```

In [10]:

```
print("Hamilton papers: " + str(h_count))
```

```python
print("Madison papers: " + str(m_count))
```

```
Hamilton papers: 12
Madison papers: 0
```

It seems like all disputed papers are attributed to Madison with this approach.

I am slightly unhappy though, as LR has a value larger than 1, should this be possible? Might have made a mistake somewhere