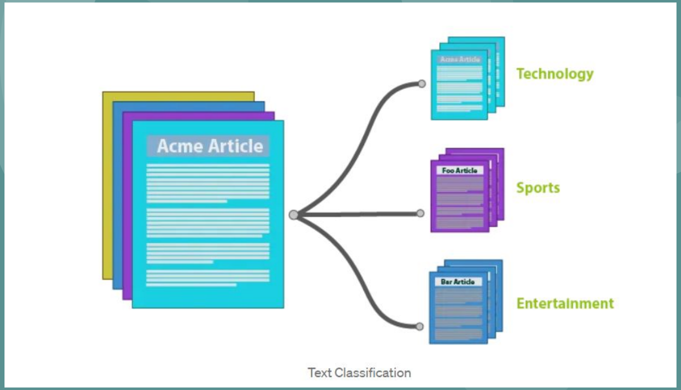
CS550 - Machine Learning and Business Intelligence

Text Classification



By: Maranata Muluneh

Instructor: Dr. Chang, Henry

2023

Table of Content

Introduction

Text classification, also known as text categorization, is a subfield of machine learning that involves classifying text documents into predefined categories based on their content. The goal of text classification is to automatically assign the appropriate category or label to a given text document, based on its topic, sentiment, author, or other relevant characteristics.

Introduction

Text classification has many real-world applications, such as spam filtering, sentiment analysis, news categorization, product review analysis, and content recommendation. By automating the process of categorizing large volumes of text, text classification can save time and resources, enable better decision-making, and improve the user experience.

Theory

Text Classification: Definition

- **Input:**

  A document d

  A fixed set of classed C = {c1, c2, …cj}

- **Output:** a predicted class c is element of C

Theory

Classification Methods: Hand coded rules

- Rules based on combinations of words or other features

  Spam: black- list address OR (“dollars” and “have been selected”)

- Accuracy can be high

  If rules carefully refined by expert

- But building and maintaining these rules is expensive

Theory

Classification Methods: Supervised Machine Learning

- **Input:**

A document d

A fixed set of classed $C = \{c1, c2, \ldots cj\}$

A training set of m hand-labeled documents (d1,c1)...(dm,cm)

- **Output:** a learned classifier $y: d \Rightarrow c$

Theory

Bayes Rule Applied to Documents and Classes

- For a document d and a class c

$$P(c \mid d) = \frac{P(d \mid c)P(c)}{P(d)}$$

Implementation

Project: Who is the real author of Hamlet?

Step 1: Please implement a Text Classifier
Test the Text Classifier to predict who the real author of Hamlet is

Implementation

Project: Who is the real author of Hamlet?

	Doc	Words	Author
Training	1	W1 W2 W3 W4 W5	C (Christopher Marlowe)
	2	W1 W1 W4 W3	C (Christopher Marlowe)
	3	W1 W2 W5	C (Christopher Marlowe)
	4	W5 W6 W1 W2 W3	W (William Stanley)
	5	W4 W5 W6	W (William Stanley)
	6	W4 W6 W3	F (Francis Bacon)
	7	W2 W2 W4 W3 W5 W5	F (Francis Bacon)
Test	8 (Hamlet)	W1 W4 W6 W5 W3	?

Implementation

- We can do the prediction of the above question in two ways:

  Manually applying the compare model

  Using Google Colab

Implementation (Manually)

- **P(C) : The probability of class C = 3/7**
- **P(W) : The probability of class W = 2/7**
- **P(F) : The probability of class F = 2/7**
- **P(W1|C): The probability that the word "W1" appears on the 3 class c documents**

= (count(W1, C) + 1) / (count(C)+|V|)

= (4+1) / (12+6) = 5/18

  - 4: how many times the word "W1" appear on the 3 class C documents.
  - 12: how many words in the 3 class C documents.
  - 6: number of vocabulary: (W1 W2 W3 W4 W5 W6)

Implementation (Manually)

- **P(W1|W) : The probability that the word "W1" appears on the 3 class W documents**

= (count(W1, W) + 1) / (count(W)+|V|)

= (1+1) / (8+6) = 2/14 = 1/7

- 1: how many times the word "W1" appear on the 2 class W documents.
- 8 : how many words in the 3 class W documents.
- 6: number of vocabulary: (W1 W2 W3 W4 W5 W6)

Implementation (Manually)

- **P(W1|F) : The probability that the word "W1" appears on the 2 class F documents**

= (count(W1, F) + 1) / (count(F)+|V|)

= (0+1) / (9+6) = 1/15

  - 0: how many times the word "W1" appear on the 2 class F documents.
  - 9: how many words in the 3 class W documents.
  - 6 : number of vocabulary: (W1 W2 W3 W4 W5 W6)

Implementation (Manually)

- **P(W3|C) : The probability that the word "W3" appears on the 3 class C documents**

= (count(W3, C) + 1) / (count(C)+|V|)

= (2+1) / (12+6) = 3/18 = 1/6

  - 2: how many times the word "W3" appear on the 3 class C documents.
  - 12 : how many words in the 3 class C documents.
  - 6 : number of vocabulary: (W1 W2 W3 W4 W5 W6)

Implementation (Manually)

- **P(W3|W) : The probability that the word "W3" appears on the 3 class W documents**

= (count(W3, W) + 1) / (count(W)+|V|)

= (1+1) / (8+6) = 2/14 = 1/7

  - 1: how many times the word "W3" appear on the 2 class W documents.
  - 8 : how many words in the 3 class W documents.
  - 6: number of vocabulary: (W1 W2 W3 W4 W5 W6)

Implementation (Manually)

- **P(W3|F) : The probability that the word "W3" appears on the 2 class F documents**

= (count(W3, F) + 1) / (count(F)+|V|)

= (2+1) / (9+6) = 3/15 = 1/5

- 2: how many times the word "W3" appear on the 2 class F documents.
- 9: how many words in the 3 class F documents.
- 6: number of vocabulary: (W1 W2 W3 W4 W5 W6)

Implementation (Manually)

- **P(W4|C) : The probability that the word "W4" appears on the 3 class C documents**

= (count(W4, C) + 1) / (count(C)+|V|)

= (2+1) / (12+6) = 3/18 = 1/6

  - 2: how many times the word "W4" appear on the 3 class C documents.
  - 12 : how many words in the 3 class C documents.
  - 6 : number of vocabulary: (W1 W2 W3 W4 W5 W6)

Implementation (Manually)

- **P(W4|W) : The probability that the word "W4" appears on the 3 class W documents**

= (count(W4, W) + 1) / (count(W)+|V|)

= (1+1) / (8+6) = 2/14 = 1/7

- 1: how many times the word "W4" appear on the 2 class W documents.
- 8 : how many words in the 3 class W documents.
- 6: number of vocabulary: (W1 W2 W3 W4 W5 W6)

Implementation (Manually)

- **P(W4|F) : The probability that the word "W4" appears on the 2 class F documents**

= (count(W4, F) + 1) / (count(F)+|V|)

= (2+1) / (9+6) = 3/15

  - 2: how many times the word "W4" appear on the 2 class F documents.
  - 9: how many words in the 3 class F documents.
  - 6: number of vocabulary: (W1 W2 W3 W4 W5 W6)

Implementation (Manually)

- **P(W5|C): The probability that the word "W5" appears on the 3 class C documents**

= (count(W5, C) + 1) / (count(C)+|V|)

= (2+1) / (12+6) = 3/18 = 1/6

  - 2: how many times the word "W5" appear on the 3 class C documents.
  - 12 : how many words in the 3 class C documents.
  - 6 : number of vocabulary: (W1 W2 W3 W4 W5 W6)

Implementation (Manually)

- **P(W5|W): The probability that the word "W5" appears on the 3 class W documents**

= (count(W5, W) + 1) / (count(W)+|V|)

= (2+1) / (8+6) = 3/14

- 2: how many times the word "W5" appear on the 2 class W documents.
- 8 : how many words in the 3 class W documents.
- 6: number of vocabulary: (W1 W2 W3 W4 W5 W6)

Implementation (Manually)

- **P(W5|F): The probability that the word "W5" appears on the 2 class F documents**

= (count(W5, F) + 1) / (count(F)+|V|)

= (2+1) / (9+6) = 3/15

- 2: how many times the word "W5" appear on the 2 class F documents.
- 9: how many words in the 3 class F documents.
- 6: number of vocabulary: (W1 W2 W3 W4 W5 W6)

Implementation (Manually)

- **P(W6|C): The probability that the word "W6" appears on the 3 class C documents**

= (count(W6, C) + 1) / (count(C)+|V|)

= (0+1) / (12+6) = 1/18

- 0: how many times the word "W6" appear on the 3 class C documents.
- 12 : how many words in the 3 class C documents.
- 6 : number of vocabulary: (W1 W2 W3 W4 W5 W6)

Implementation (Manually)

- **P(W6|W): The probability that the word "W6" appears on the 2 class W documents**

= (count(W6, W) + 1) / (count(W)+|V|)

= (2+1) / (8+6) = 3/14

- 2: how many times the word "W6" appear on the 2 class W documents.
- 8 : how many words in the 3 class W documents.
- 6: number of vocabulary: (W1 W2 W3 W4 W5 W6)

Implementation (Manually)

- **P(W6|F) : The probability that the word "W6" appears on the 2 class F documents**

= (count(W6, F) + 1) / (count(F)+|V|)

= (1+1) / (9+6) = 2/15

- 1: how many times the word "W6" appear on the 2 class F documents.
- 9: how many words in the 3 class F documents.
- 6: number of vocabulary: (W1 W2 W3 W4 W5 W6)

Implementation (Manually)

P(C|d8) : P(C) * P(W1|C) * P(W4|C)* P(W6|C) * P(W5|C) * P(W3|C)

= ((3/7) * (5/18)* (1/6)* (1/18) *(1/6) *(1/6))

= 0.00003061924 , approx 0.00003

= 3/7: prior : P(C)

= There are 5 words in d8 : W1 W4 W6 W5 W3

- Each word "W1" has P(W1|C) = 5/18
- The word "W4" has P(W4|C) =3/18 = 1/6
- The word "W6" has P(W6|C) = 1/18
- The word "W5" has P(W5|C) = 3/18 = 1/6
- The word "W3" has P(W3|C) = 3/18 = 1/6

Implementation (Manually)

$P(W|d8) = P(W) * P(W1|W) * P(W4|W) * P(W6|W) * P(W5|W) * P(W3|W)$

$= (2/7 * 2/14 * 2/14 * 3/14 * 3/14 * 2/14)$

$= 0.00003824936$, approx 0.00004

= 2/7: prior : P(W)

= There are 5 words in d8 : W1 W4 W6 W5 W3

- Each word "W1" has $P(W1|W) = 2/14$
- The word "W4" has $P(W4|W) = 2/14$
- The word "W6" has $P(W6|W) = 3/14$
- The word "W5" has $P(W5|W) = 3/14$
- The word "W3" has $P(W3|W) = 2/14$
-

Implementation (Manually)

P(F|d8) = P(F) * P(W1|F) * P(W4|F)* P(W6|F) * P(W5|F) * P(W3|F)

=((2/7) * (1/15)*(3/15) * (2/15) * (3/15) * (3/15))

= 0.00002031746 , approx 0.00002

= 2/7:prior : P(F)

= There are 5 words in d8 : W1 W4 W6 W5 W3

- Each word "W1" has P(W1|F) = 1/15
- The word "W4" has P(W4|F)= 3/15
- The word "W6" has P(W6|F)= 2/15
- The word "W5" has P(W5|F)= 3/15
- The word "W3" has P(W3|F) = 3/15

Implementation (Manually)

Does d8 belong to C or W or F?

According to the numbers from the probability calculations, Document 8 should belong to class W Since it has the highest probability calculation.

Implementation (Colab)

Environment : Colab, Tensorflow 2

Programming Language: Python

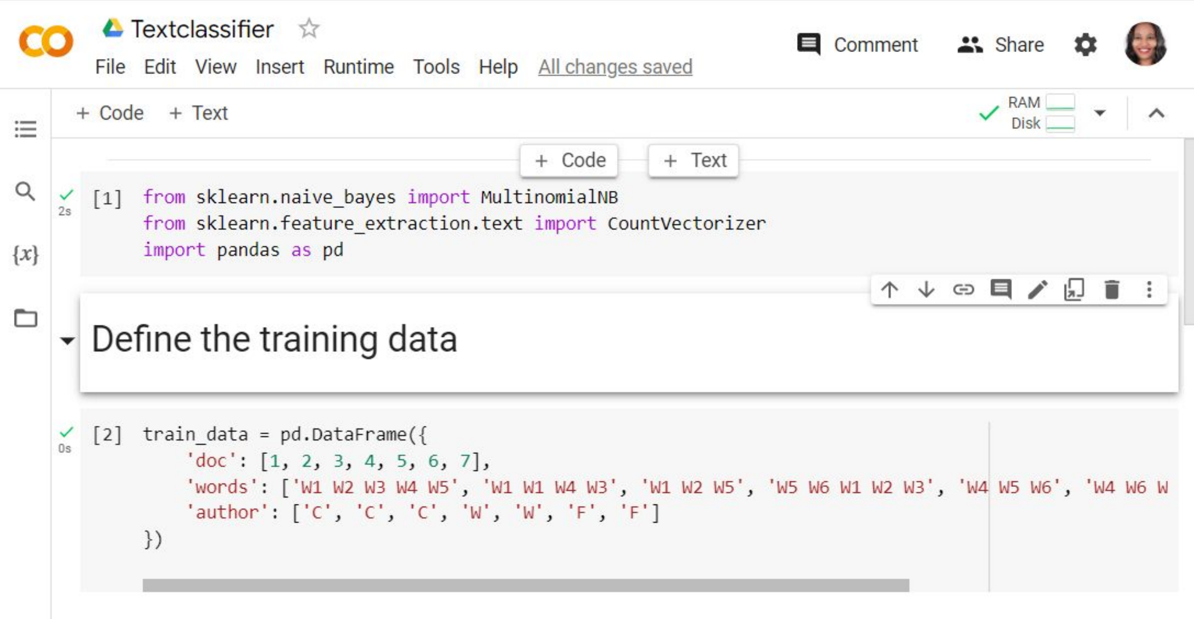
Libraries

```
from sklearn.naive_bayes import
MultinomialNB
from sklearn.feature_extraction.text import
CountVectorizer
import pandas as pd
```

Implementation

- Here's implementation of the text classifier using Colab:

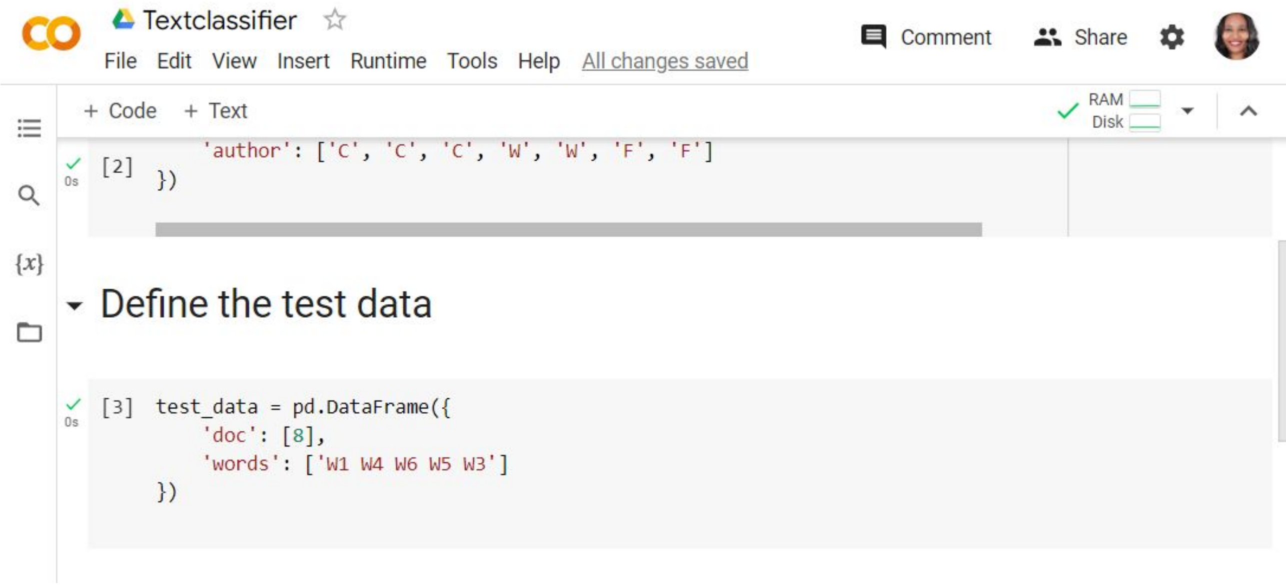
Step 1: Define the training data



Implementation

- Here's implementation of the text classifier using Colab:

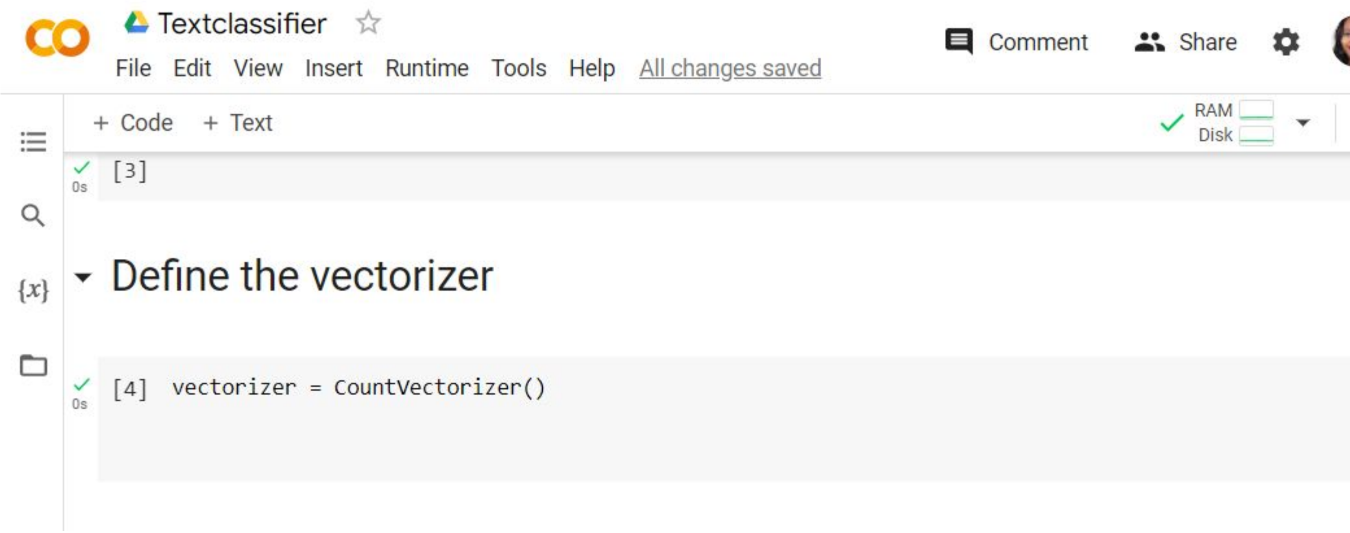
Step 2: Define the test data



Implementation

- Here's implementation of the text classifier using Colab:

Step 3: Define the Vectorizer



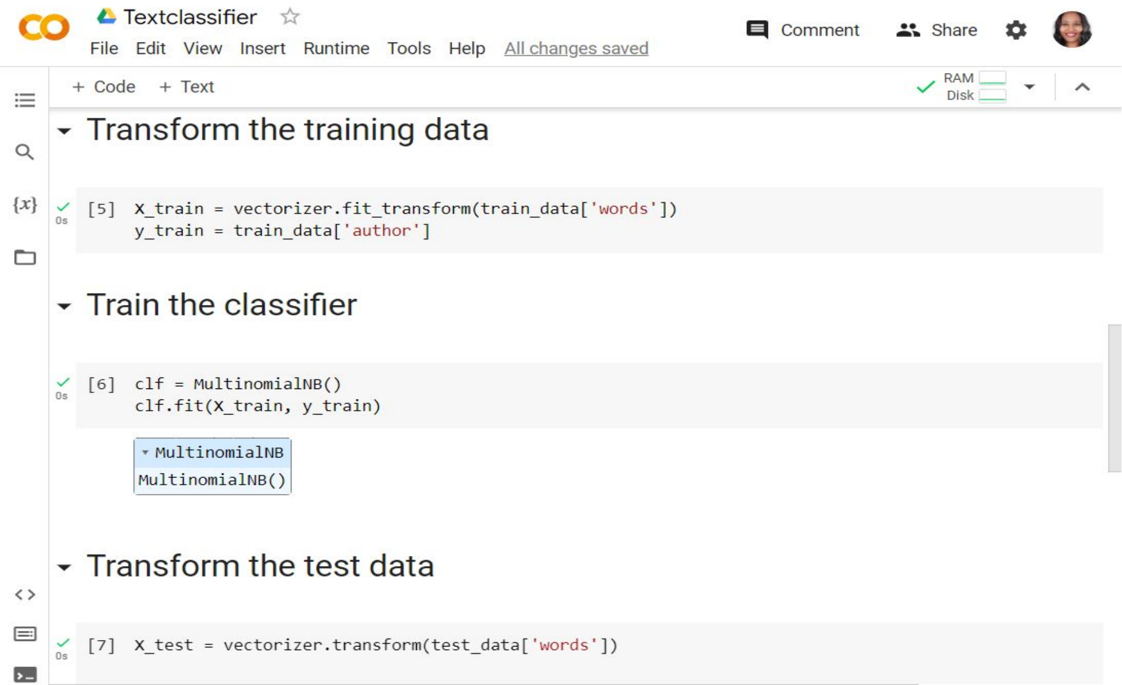
Implementation

- Here's implementation of the text classifier using Colab:

Step 4: Transform the training data

Step 5: Train the classifier

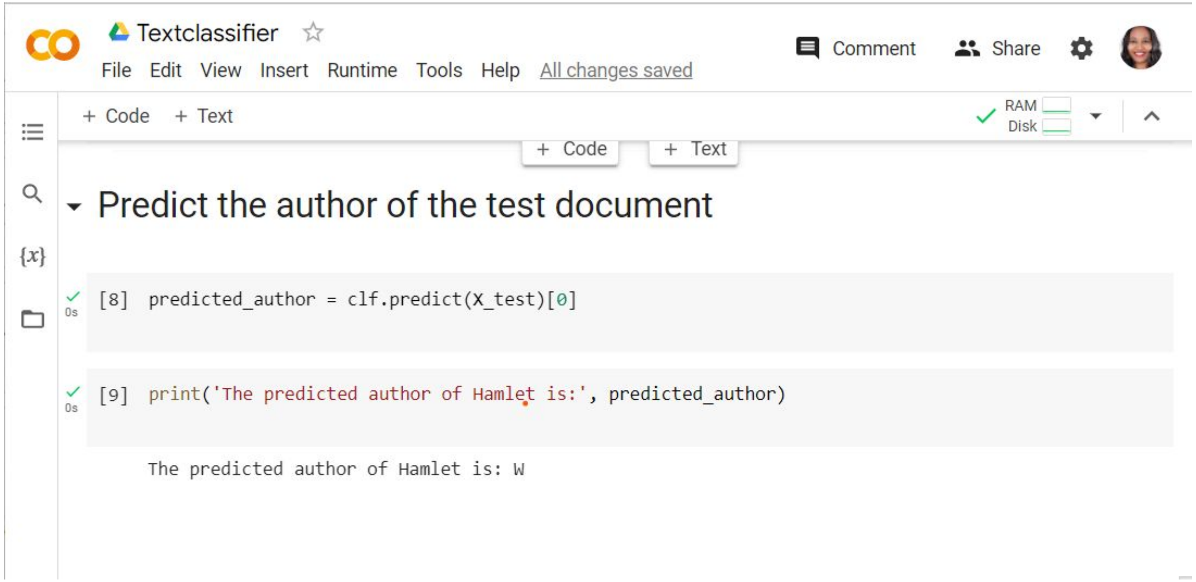
Step 6: Transform the test data



Implementation

- Here's implementation of the text classifier using Colab:

Step 7: Predict the author of the test document



Conclusion

Text classification is a powerful technique that allows us to automatically categorize and classify text data. It has a wide range of applications, including sentiment analysis, spam filtering, and topic modeling.

Overall, text classification is a valuable tool for analyzing and processing large amounts of text data, but it requires careful planning and implementation to be effective.

References

- Shaikh, J. (2017, October 30). Machine Learning, NLP: Text classification using scikit-learn, python and NLTK. Medium. Retrieved March 7, 2023, from https://towardsdatascience.com/machine-learning-nlp-text-classification-using-scikit-learn-python-and-nltk-c52b92a7c73a

- San Francisco Bay University. SFBU. (n.d.). Retrieved March 7, 2023, from https://hc.labnet.sfbu.edu/~henry/sfbu/course/mllib/naive_bayes/hw/q12/2021_summer/Text_Classifier.ipynb