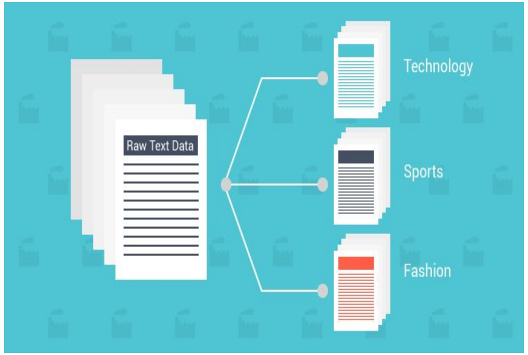
Text Classification

CS550 - Machine Learning and Business Intelligence



Submitted by: Divya Pandey(19665)
Instructor: Dr. Henry Chang

Table of Content

Introduction

★ Text classification is one of the fundamental tasks in natural language processing with broad applications such as sentiment analysis, topic labeling, spam detection, and intent detection.

★ Text classifiers can be used to organize, structure, and categorize pretty much any kind of text – from documents, medical studies and files, and all over the web.

Design

Text Classification: definition

- *Input*:
  - a document d
  - a fixed set of classes $C = \{c_1, c_2, \dots, c_J\}$
- *Output*: a predicted class $c \in C$

Design

- Training
  - Priors:

```
P(X) = The probability of a class X
     = Number of class X / total number of classes
     = Nx / N
```

Note:

- P(c)= The probability of class c =3/4 (i.e., 3 c-classes / total classes)
- P(j)= The probability of class j = 1/4

Design

- Conditional probabilities:

```
P(w|x) = If a document belongs to class x,
         the probability that the document has word w.
       = The probability that the word w appears on the class x document.
       = (count(w, x) + 1) / (count(x)+|V|)
```

Note:

- Original definition of P(w|x) = count(w, x) / count (x)
  - count(w, x): how many times the word w appears on the x class documents.
  - count(x): how many words on the x class documents.
- |V|: number of vocaculary = number of different words
- Tunable knobs (i.e., parameters) of Naive Bayes
  - 1 and |V| are used for Laplace Smoothing to prevent the possibility of letting P(w|x) have value of 0 or 1.
    - Other values can be used to replace 1 and |V|.

Design

Bayes' Rule Applied to Documents and Classes

- For a document *d* and a class *c*

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

Design

$P(C)$ = The probability of Author C = 3/7 (i.e., 3 C- Authors / total Authors)

$P(W)$ = The probability of Author W = 2/7 (i.e., 2 W- Authors / total Authors)

$P(F)$ = The probability of Author F = 2/7 (i.e., 2 F- Authors / total Authors)

$P(W1|C) = (count(W1, C) + 1) / (count(C)+|V|) = (4+1) / (12+6) = 5/18$

$P(W1|W) = (count(W1, W) + 1) / (count(W)+|V|) = (1+1) / (8+6) = 2/14$

$P(W1|F) = (count(w1, F) + 1) / (count(F)+|V|) = (0+1)/(9+6) = 1/15$

$P(W3|C) = (count(W3, C) + 1) / (count(C)+|V|) = (2+1)/(12+6) = 3/18$

Design

- $P(W3|W) = (count(W3, W) + 1) / (count(W)+|V|) = (1+1)/(8+6) = 2/14$
- $P(W3|F) = (count(W3, F) + 1) / (count(F)+|V|) = (2+1) / (9+6) = 3/15$
- $P(W4|C) = (count(W4, C) + 1) / (count(C)+|V|) = (2+1) / (12+6) = 3/18$
- $P(W4|W) = (count(W4, W) + 1) / (count(W)+|V|) = (1+1)/(8+6) = 2/14$
- $P(W4|F) = (count(W4, F) + 1) / (count(F)+|V|) = (2+1) / (9+6) = 3/15$
- $P(W5|C) = (count(W5, C) + 1) / (count(C)+|V|) = (2+1) / (12+6) = 3/18$
- $P(W5|W) = (count(W5, W) + 1) / (count(W)+|V|) = (2+1)/(8+6) = 3/14$

Design

- $P(W5|F) = (count(W5, F) + 1) / (count(F)+|V|) = (2+1) / (9+6) = 3/15$
- $P(W6|C) = (count(W6, C) + 1) / (count(C)+|V|) = (0+1)/(12+6) = 1/18$
- $P(W6|W) = (count(W6, W) + 1) / (count(W)+|V|) = (2+1)/(8+6) = 3/14$
- $P(W6|F) = (count(W6, F) + 1) / (count(F)+|V|) = (1+1)/(9+6) = 2/15$

Design

A. The probability of d8 (i.e., document 8) belonging to Author C

$P(C|d8) = P(C) * P(d8|C) / P(d8)$

Applying Bayes Theorem $= P(C) * P(W1 \cap W4 \cap W6 \cap W5 \cap W3 \mid C) / P(d8)$

Applying Naive Bayes Theorem $\propto (P(C) * P(W1|C) * P(W4|C) * P(W6|C) * P(W5|C) * P(W3|C)) / P(d8)$

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

Applying Compare Model $P(C|d8) \propto P(C) * P(W1|C) * P(W4|C) * P(W6|C) * P(W5|C) * P(W3|C) = 3/7 * 5/18 * 3/18 * 1/18 * 3/18 * 3/18 = 0.00003061924$

Design

B. The probability of document 8 belonging to Author W.

Applying Naive Bayes Theorem

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

Applying Compare Model

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

$= 2/7* 2/14 * 2/14 * 3/14 * 3/14 * 2/14 = 0.00003824936$

Design

C. **The probability of document 8 belonging to Author F.**

Applying Naive Bayes Theorem

$P(F|d8) \propto (P(F) * P(W1|F) * P(W4|F) * P(W6|F) * P(W5|F) * P(W3|F)) / P(d8)$

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

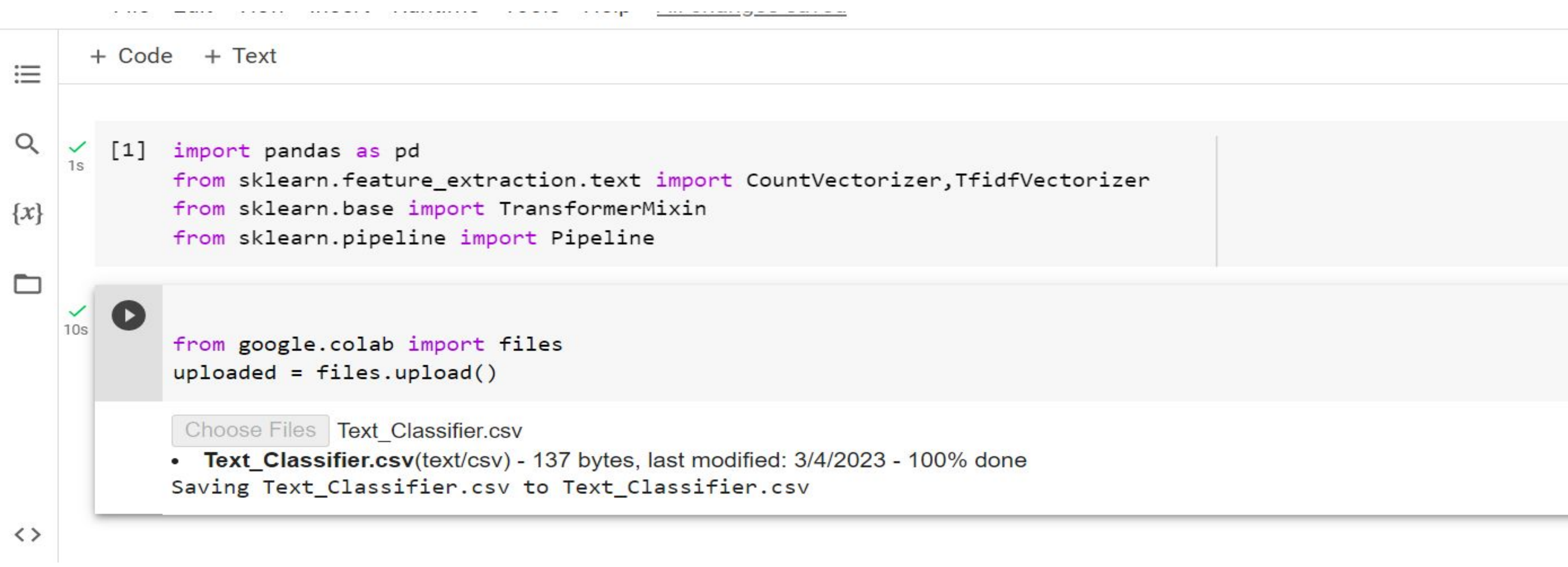
Applying Compare Model

$P(F|d8) \propto P(F) * P(W1|F) * P(W4|F) * P(W6|F) * P(W5|F) * P(W3|F)$

$= 2/7 * 1/15 * 3/15 * 2/15 * 3/15 * 3/15 =$ 0.00002031746

Implementation

★ Go to [Colab]



Implementation

- ★ Upload the **textclassifier.csv**
- ★ Run Python Code of text-classifier

Test

```
[3] data = pd.read_csv("Text_Classifier.csv")
    print(data)
```

```
   Doc                Words Author
0    1       w1 w2 w3 w4 w5      C
1    2          w1 w1 w4 w3      C
2    3             w1 w2 w5      C
3    4       w5 w6 w1 w2 w3      W
4    5             w4 w5 w6      W
5    6             w4 w6 w3      F
6    7    w2 w2 w4 w3 w5 w5      F
```

Test

textclassifier.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
[ ]                              multi_class='auto', n_jobs=None,
                                 penalty='l2', random_state=None,
                                 solver='lbfgs', tol=0.0001, verbose=0,
                                 warm_start=False))],
             verbose=False)
```

```
New_Value = ["w1 w4 w6 w5 w3"]
predicted1 = pipe.predict(New_Value) #New data
print(predicted1)
```

```
['W']
```

Enhancement Ideas

★ We can use pre-processing techniques such as tokenization, stemming, lemmatization, stop-word removal, and spell-checking can improve text classification accuracy by improving the quality of the text data.

Conclusion

★ Text classification is a core feature of Machine Learning that enables organizations to develop deep insights that inform future decisions.

★ Text classification algorithms can discover the many correlations between distinct parts of the text and the predicted output for a given text or input.

★ Thus, Predicted real Author of Hamlet is **William Stanley**

Github Link

https://github.com/divyapandey03/Machine-Learning/tree/main/Text%20Classification

References

★ Shaikh, J. (2017, October 30). *Machine Learning, NLP: Text classification using scikit-learn, python and NLTK*. Medium. Retrieved March 4, 2023, from https://towardsdatascience.com/machine-learning-nlp-text-classification-using-scikit-learn-python-and-nltk-c52b92a7c73a

★ *Text classifiers in Machine Learning: A practical guide*. RSS. (n.d.). Retrieved March 4, 2023, from https://levity.ai/blog/text-classifiers-in-machine-learning-a-practical-guide