Text Classification

Fatema Nagori

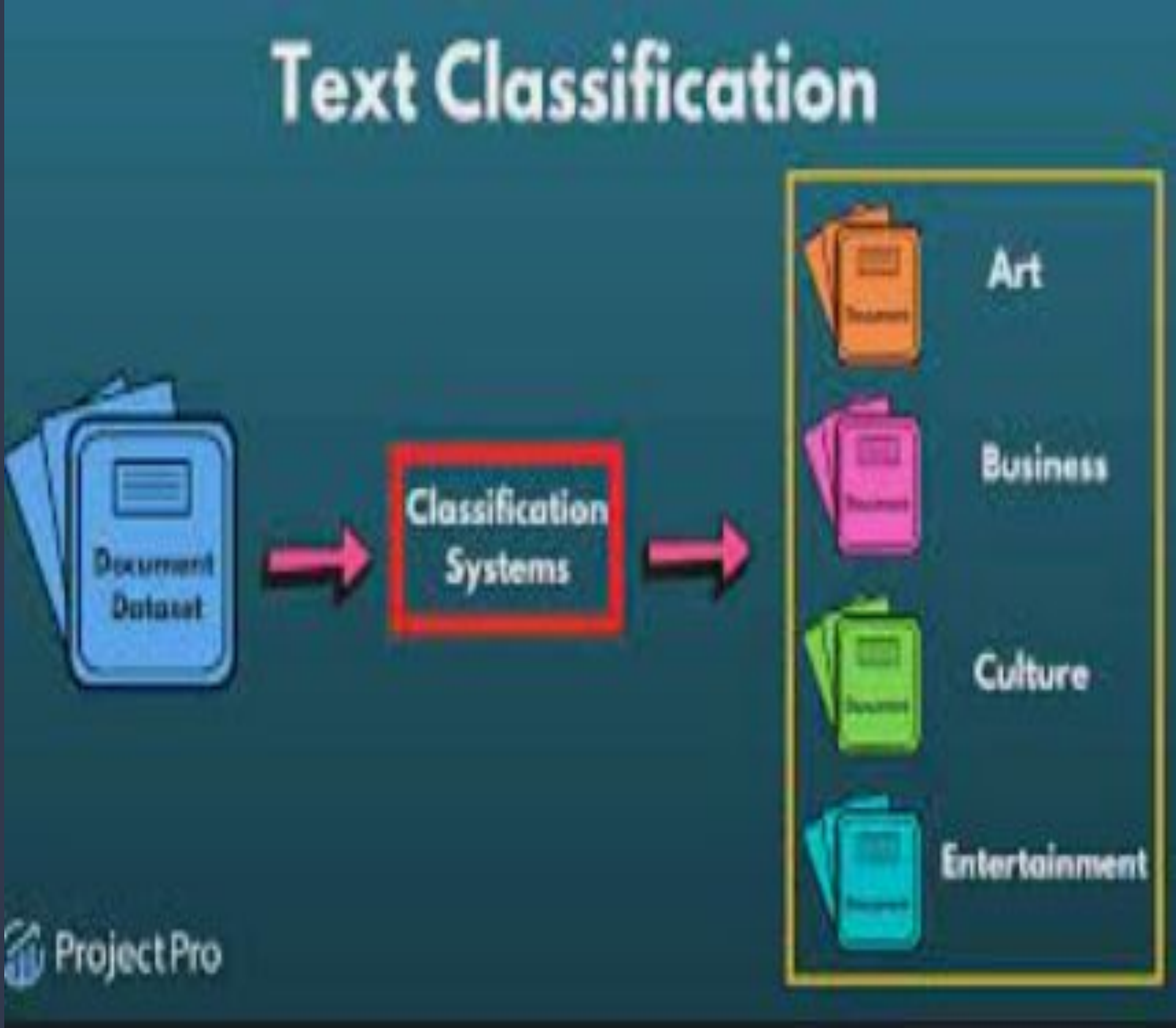

TABLE OF CONTENTS:

- Introduction
- Design
- Implementation
- Test
- Result
- Conclusion
- Enhancement
- References

Introduction

In machine learning, text classification is a supervised learning task that involves training a model to automatically assign one or more predefined categories or labels to a given text document.

Text classification is a widely used technique in a variety of applications, such as sentiment analysis, spam filtering, topic modeling, and content recommendation. The accuracy and performance of text classification models can be improved by using various techniques, such as data pre-processing, feature engineering, model optimization, and ensemble methods.

DESIGN

1. Define the problem and determine the categories or labels to be assigned to text documents.
2. Collect and preprocess the text data.
3. 3.Split the data into training, validation, and testing sets.
4. Extract features from the preprocessed text data.
5. Choose an appropriate machine learning model.
6. Train and optimize the model.
7. Evaluate the model's performance using appropriate metrics.
8. Deploy the model for use in classifying new, unseen text data.

It's important to note that the design process is iterative and may require revisiting earlier steps to optimize the model's performance.

IMPLEMENTATION

Who is the real author of Hamlet?

PROCESS

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

	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	?

TRAINING

Training

Priors:

P(X) = The probability of a class X

= Number of class X / total number of classes

= N_X / N

-

  Note:

  ○ P(C)= The probability of class C =3/7 (i.e., 3 c-classes / total classes)

  ○ P(W)= The probability of class W = 2/7

  ○ P(F)= The probability of class F = 2/7

CONDITIONAL PROBABILITY

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.

CONDITIONAL PROBABILITY

The Test only has 6 words: W1 W2 W3 W4 W5 W6

- P(W1|C)
  = The probability that the word "W1" appears on the 6 class C documents.
  = (4+1) / (12+6) = 5/18
  Note:
  - 4: how many times the word "W1" appear on the 6 class C documents.
  - 12: how many words in the 6 class C documents.
  - 6: number of vocabulary: W1 W2 W3 W4 W5 W6
- P(W1|W) = (1+1) / (8+6) = 1/7
- P(W1|F) = (0+1) / (9+6) = 1/15
- P(W3|C) = (2+1) / (12+6) = 1/3
- P(W3|W) = (1+1) / (8+6) = 1/7
- P(W3|F) = (2+1) / (9+6) = 1/5

CONDITIONAL PROBABILITY

- $P(W4|C) = (2+1) / (12+6) =1/6$
- $P(W4|W) = (1+1) / (8+6) =1/7$
- $P(W4|F) = (2+1) / (9+6) =1/5$
- $P(W5|C) = (2+1) / (12+6) =1/6$
- $P(W5|W) = (2+1) / (8+6) =3/14$
- $P(W5|F) = (2+1) / (9+6) =1/5$
- $P(W6|C) = (0+1) / (12+6) =1/18$
- $P(W6|W) = (1+1) / (8+6) =1/7$
- $P(W6|F) = (1+1) / (9+6) =2/15$
- $P(C|D8) = (0+1) / (8+6) =$
- $P(W|D8) = (0+1) / (8+6) =$
- $P(D|D8) = (0+1) / (8+6) =$

TEST

Decide whether d8 (i.e., document 8) belongs to class c or class w or class f.

- Step 1: Analysis

The probability that the document d8 belongs to class C, W or F

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

==> Applying Bayes Theorm

$= P(c) * P(W1 \cap W4 \cap W6 \cap W5 \cap W3 |c) / P(d8)$

==> Applying Naive Bayes Theorm

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

==> Applying Compare Model

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

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

$\cong 0.00003061924$

TEST

- $P(W|D8) = P(w) * P(d8|w) / P(d8)$

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

$$= 2/7 * 1/7 * 1/7* 1/7* 1/7* 3/14$$

$$= 2/7 * (1/7)^4 * 3/14$$

$$\cong 0.00002549957$$

- $P(F|D8) = P(F) * P(d8|F) / P(d8)$

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

$$=2/7 * 1/15 * 1/5 * 2/15* 1/5* 1/5$$

$$=2/7 *(1/5)^4 * 2/15$$

$$\cong 0.00002031746$$

Conclusion

- Document 8 should belong to the class c

Enhancement

Text classification is a powerful tool for processing and analyzing large amounts of textual data. To enhance the accuracy and effectiveness of a text classifier, several techniques can be used, including feature engineering, model selection, data preprocessing, data augmentation, regularization, and ensemble methods. The selection of appropriate techniques depends on the specific task at hand and the characteristics of the data. By combining these techniques and optimizing hyperparameters, it is possible to build highly accurate text classifiers that can be used for a variety of applications, including spam filtering, sentiment analysis, topic modeling, and content recommendation. Ultimately, the key to enhancing a text classifier is to continuously iterate and experiment with different approaches until the desired level of accuracy and performance is achieved.

References

- CHAT GPT
- Prof Chang’s Material