Bài giảng môn học:

Học Máy (Machine Learning)

CHƯƠNG 3:
HỌC CÓ GIÁM SÁT (tiếp)
(Supervised Learning)

Giảng viên: Đặng Văn Nam

Email: dangvannam@humg.edu.vn

Nội dung bài học – Phần 03

1. NLP và bài toán phân lớp văn bản
2. Thuật toán Naïve bayes
3. Phân loại comment trên twitter
4. Bài tập Thực hành

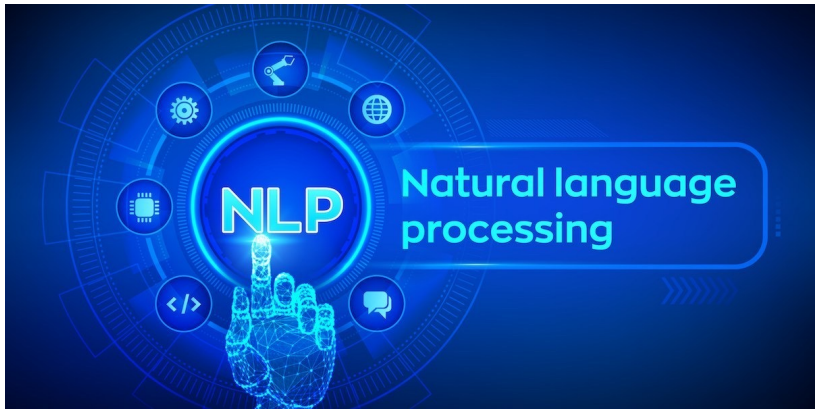
1. NLP và bài toán phân lớp văn bản

1. Giới thiệu về NLP

- **Xử lý ngôn ngữ tự nhiên – NLP (Natural Language Processing)** là một nhánh của Trí tuệ nhân tạo, tập trung vào việc nghiên cứu sự tương tác giữa máy tính và ngôn ngữ tự nhiên của con người, dưới dạng tiếng nói (speech) hoặc văn bản (text).
- Mục tiêu của NLP là giúp máy tính hiểu và thực hiện hiệu quả những nhiệm vụ liên quan đến ngôn ngữ của con người như: tương tác giữa người và máy, cải thiện hiệu quả giao tiếp giữa con người với con người, hoặc đơn giản là nâng cao hiệu quả xử lý văn bản và lời nói.

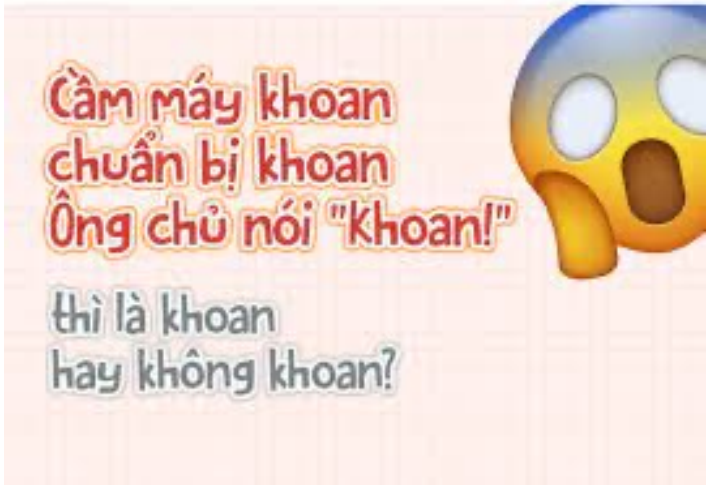
Xử lý ngôn ngữ tự nhiên được chia ra thành hai nhánh lớn, bao gồm:

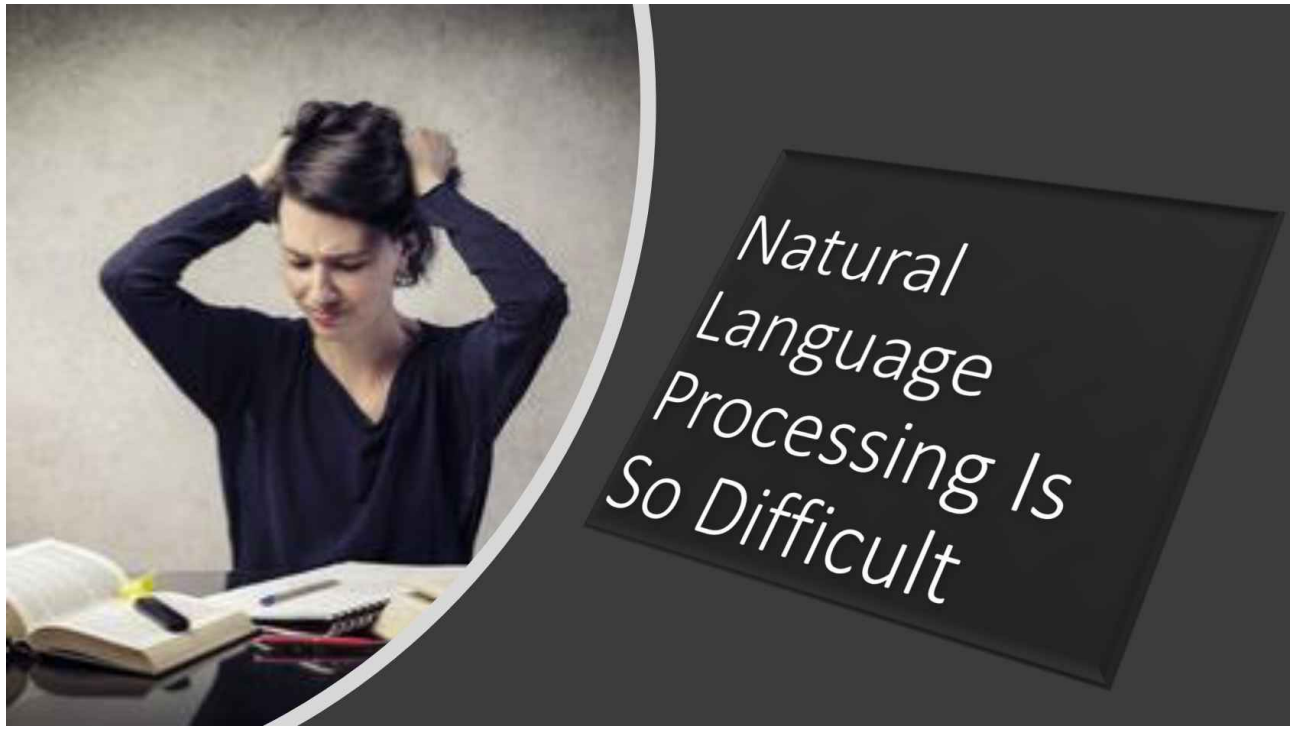
- Xử lý Tiếng nói (Speech processing)
- **Xử lý Văn bản (Text processing).**



1. Giới thiệu về NLP

- **NLP là một lĩnh vực khó?:**
  - Tính nhập nhằng của ngôn ngữ (Ambiguity)
  - Tri thức nền (Background knowledge)





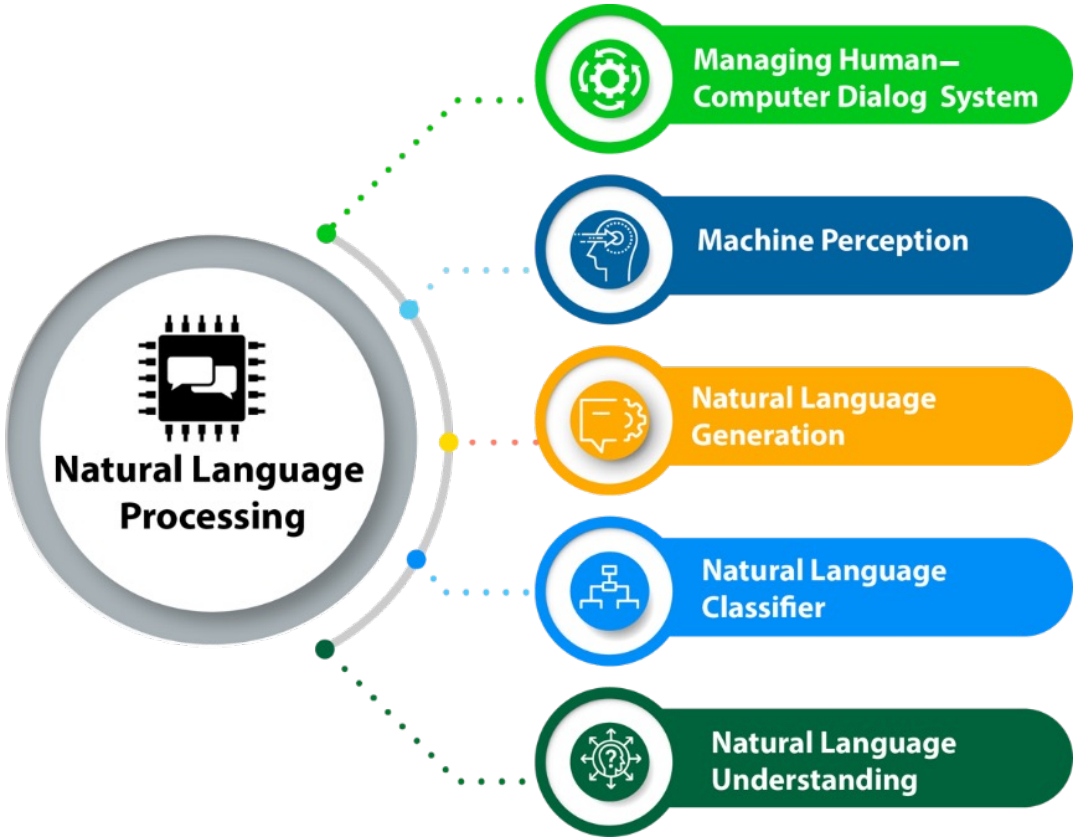
1. Giới thiệu về NLP

- **Một số ứng dụng quan trọng của NLP:**

1.Nhận dạng tiếng nói (Speech To Text) chuyển đổi ngôn ngữ từ dạng tiếng nói sang dạng văn bản, thường được ứng dụng trong các chương trình điều khiển qua giọng nói.

2.Tổng hợp tiếng nói (Text to Speech) chuyển đổi ngôn ngữ từ dạng văn bản sang tiếng nói, thường được dùng trong đọc văn bản tự động.

3. Truy xuất thông tin (Information Retrieval) có nhiệm vụ tìm các tài liệu dưới dạng không có cấu trúc (thường là văn bản) đáp ứng nhu cầu về thông tin từ những nguồn tổng hợp lớn. Những công cụ tìm kiếm như Google, Yahoo, hoặc Bing search…cho phép tiếp nhận một câu truy vấn dưới dạng ngôn ngữ tự nhiên làm đầu vào và cho ra một danh sách các tài liệu được sắp xếp theo mức độ phù hợp.

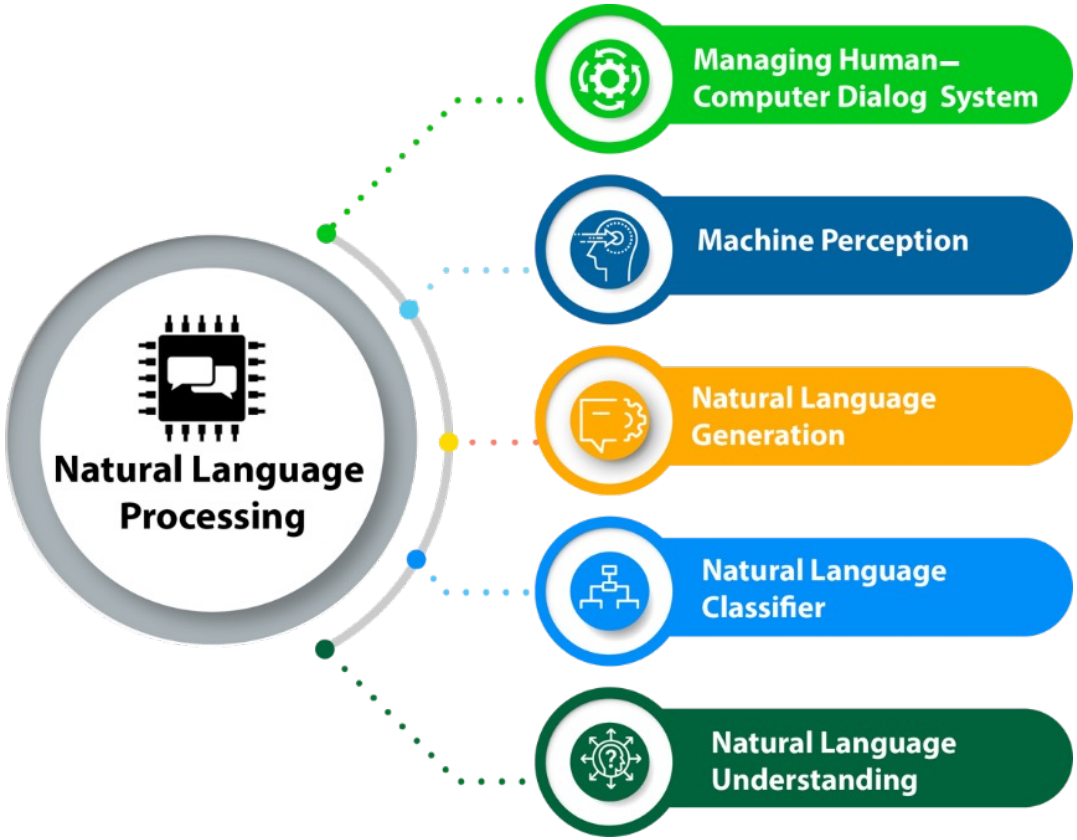


1. Giới thiệu về NLP

4. Trích chọn thông tin (Information Extraction) nhận diện một số loại thực thể được xác định trước, mối quan hệ giữa các thực thể và các sự kiện trong văn bản ngôn ngữ tự nhiên. Khác với truy xuất thông tin trả về một danh sách các văn bản hợp lệ thì trích chọn thông tin trả về chính xác thông tin mà người dùng cần. Những thông tin này có thể là về con người, địa điểm, tổ chức, ngày tháng, hoặc thậm chí tên công ty, mẫu sản phẩm hay giá cả.

5. Trả lời câu hỏi (Question Answering) có khả năng tự động trả lời câu hỏi của con người ở dạng ngôn ngữ tự nhiên bằng cách truy xuất thông tin từ một tập hợp tài liệu.

6. Tóm tắt văn bản tự động (Automatic Text Summarization) là bài toán thu gọn văn bản đầu vào để cho ra một bản tóm tắt ngắn gọn với những nội dung quan trọng nhất của văn bản gốc.

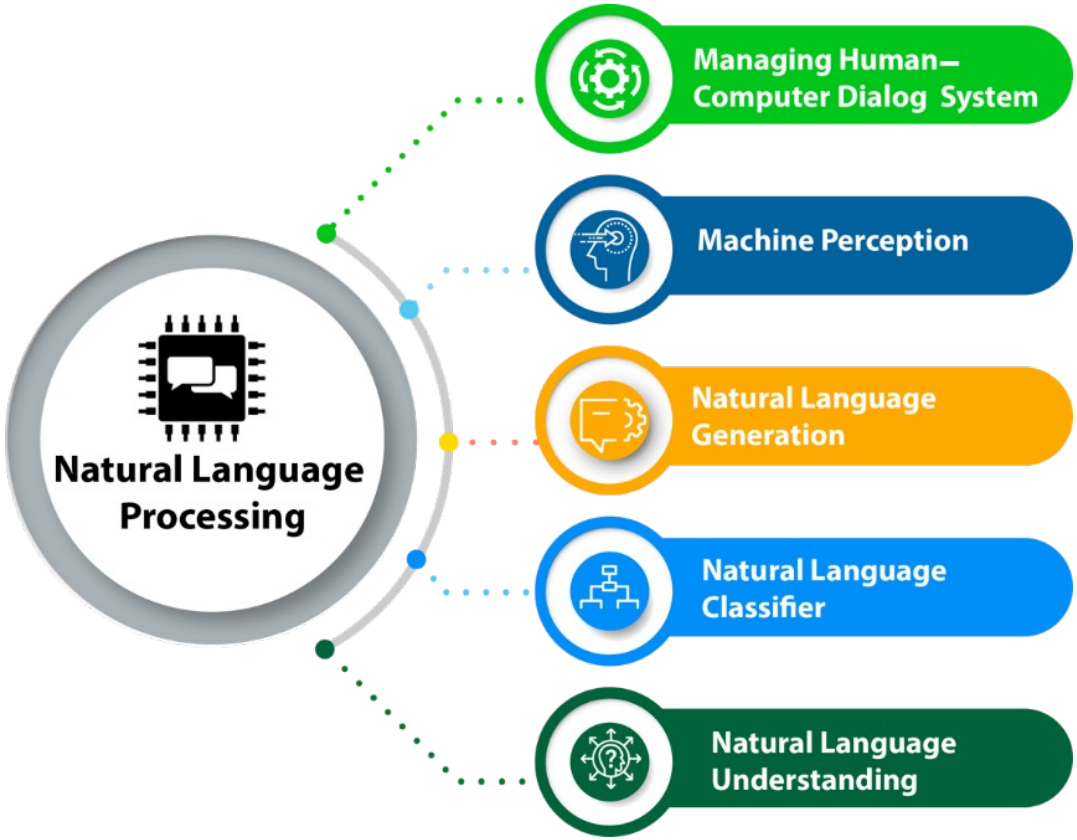


1. Giới thiệu về NLP

7. Chatbot là việc chương trình máy tính có khả năng trò chuyện (chat), hỏi đáp với con người qua hình thức hội thoại dưới dạng văn bản (text). Chatbot thường được sử dụng trong ứng dụng hỗ trợ khách hàng, giúp người dùng tìm kiếm thông tin sản phẩm, hoặc giải đáp thắc mắc.

8. Dịch máy (Machine Translation) là việc sử dụng máy tính để tự động hóa một phần hoặc toàn bộ quá trình dịch từ ngôn ngữ này sang ngôn ngữ khác.

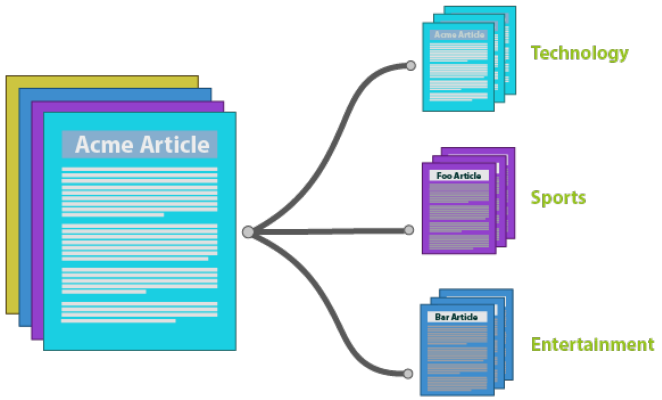
9. Kiểm lỗi chính tả tự động là việc sử dụng máy tính để tự động phát hiện các lỗi chính tả trong văn bản (lỗi từ vựng, lỗi ngữ pháp, lỗi ngữ nghĩa) và đưa ra gợi ý cách chỉnh sửa lỗi.

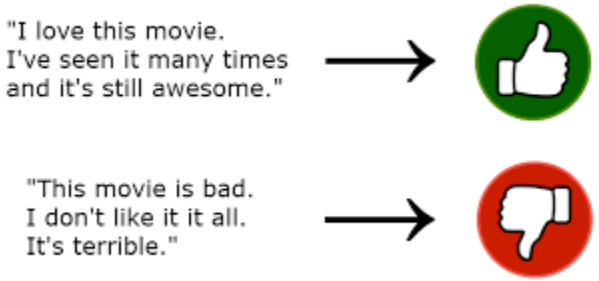


1. NLP và bài toán phân lớp văn bản

- Trong các ứng dụng của NLP, bài toán phân lớp văn bản là một trong những bài toán quan trọng và kinh điển nhất.
- Mục tiêu của một hệ thống phân lớp văn bản là nó có thể tự động phân lớp một văn bản cho trước, để xác định xem văn bản đó thuộc lớp nào.

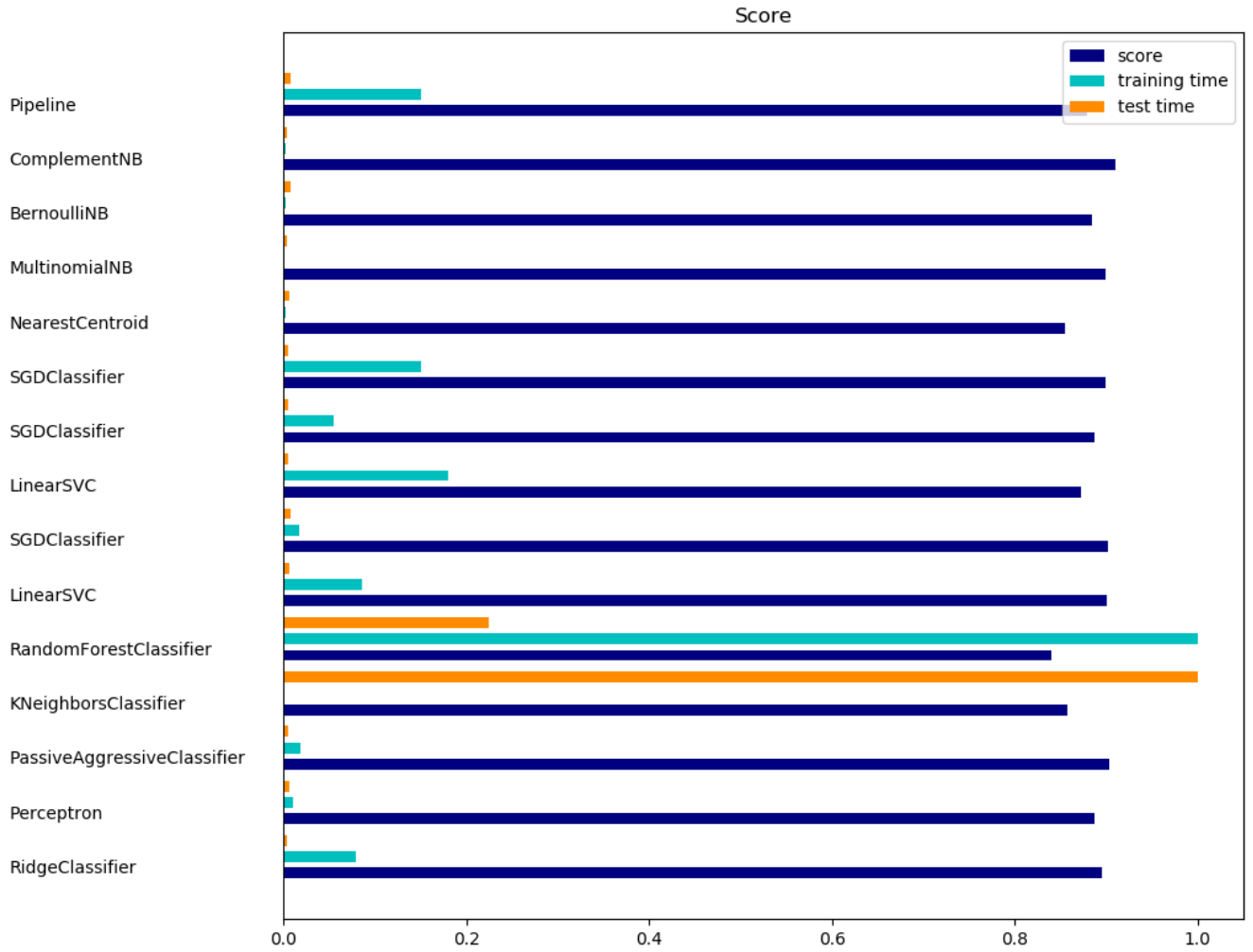
- Các ứng dụng của phân lớp văn bản rất đa dạng như: Hiểu được ý nghĩa, đánh giá, bình luận của người dùng; Lọc email rác; Phân tích cảm xúc; Phân lớp tin tức các bài báo điện tử…





1. NLP và bài toán phân lớp văn bản

```
2034 documents - 3.980MB (training set)
1353 documents - 2.867MB (test set)
```



https://scikit-learn.org/stable/auto_examples/text/plot_document_classification_20newsgroups.html#sphx-glr-auto-examples-text-plot-document-classification-20newsgroups-py

2. THUẬT TOÁN NAÏVE BAYES

Thuật toán Naïve Bayes

- Naive Bayes Classifiers (NBC) là một trong những thuật toán tiêu biểu cho bài toán phân lớp dựa trên lý thuyết xác suất áp dụng định lý Bayes.
- Định lý Bayes cho phép chúng ta có thể tính toán một xác suất chưa biết dựa vào các xác suất có điều kiện khác. Với công thức tổng quát tính xác suất của biến cố A với điều kiện biến cố B_k xảy ra trước (hay được gọi là xác suất hậu nghiệm):

Ví dụ: "xác suất trời sẽ mưa nếu trời có mây là bao nhiêu?" là một ví dụ về xác suất có điều kiện.

- Với P(A) > 0 và $\{B_1, B_2, \dots, B_n\}$ là một hệ đầy đủ các biến cố thỏa mãn tổng xác suất của hệ bằng 1 ($\sum_{k=1}^{n} P(B_k) = 1$) và từng đôi một xung khắc ($P(B_i \cap B_j) = 0$).

Naïve Bayes Classifier

$$P(A|B) = \frac{P(B|A)\ P(A)}{P(B)}$$

Thomas Bayes
1702 - 1761

Ví dụ công thức Naïve Bayes

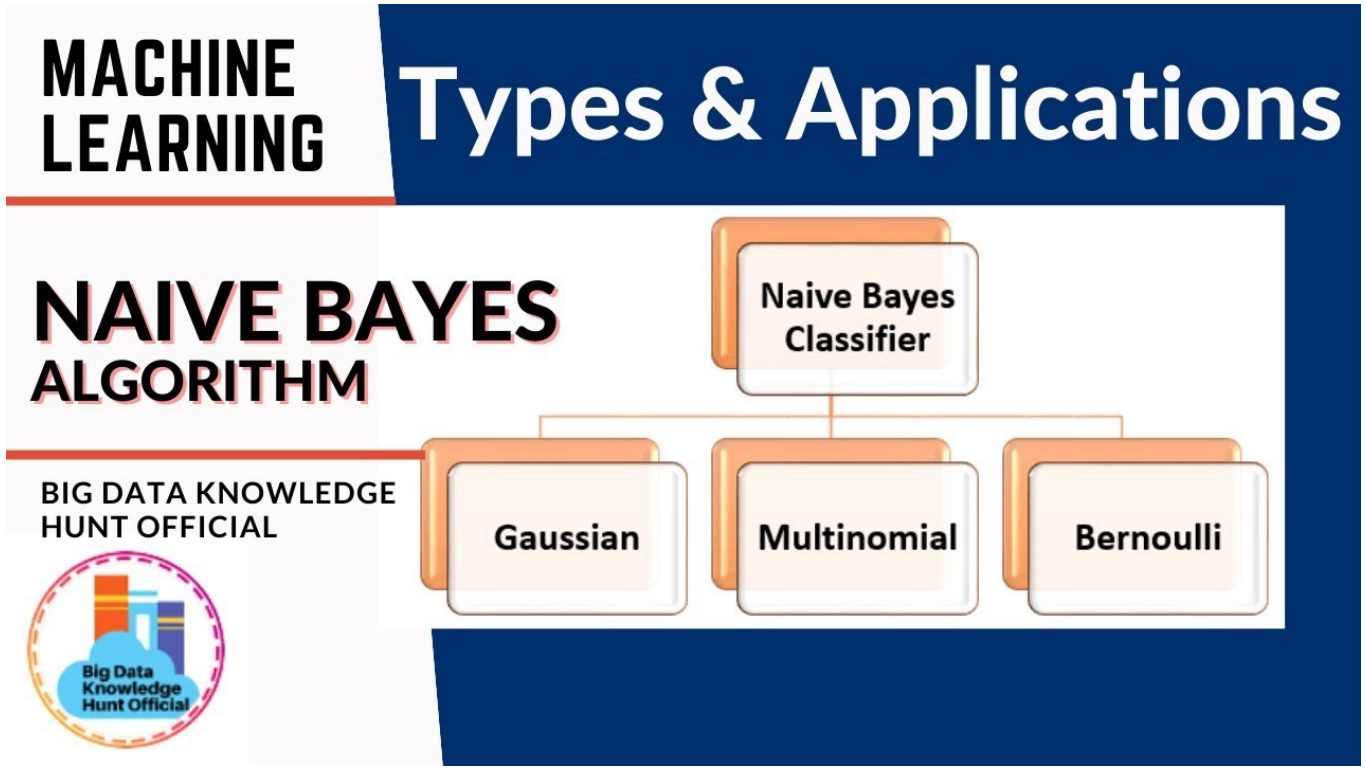
- Một bệnh viện phải làm xét nghiệm với một số lượng lớn bệnh nhân và thấy rằng có 0.1% bị mắc bệnh còn 99.9% là khoẻ. Để biết rằng việc xét nghiệm là đúng người ta cho một người khoẻ xét nghiệm thì xác suất âm tính 99%. Nếu xét nghiệm trên một người bị bệnh thì xác suất dương tính là 98%. Chọn ngẫu nhiên một người và thấy rằng người này dương tính, tìm xác suất người này bị mắc bệnh?
- +(-) là sự kiện người này dương (âm tính). A(Ā) là người này khoẻ (mắc bệnh).
- Theo dữ kiện, ta có:

  P(A) = 0.999;
  P(Ā) = 0.001;
  P(+|A) = 0.01;
  P(−|A) = 0.99;
  P(+|Ā) = 0.98;
  P(−|Ā) = 0.02

$$P(\overline{A}|+) = \frac{P(+|\overline{A}).P(\overline{A})}{P(+)} = \frac{0.98\times0.001}{0.98\times0.001+0.01\times0.999} = 0.0893$$

Thuật toán Naïve Bayes

Trong công thức Naïve Bayes, Giả thiết các chiều dữ liệu là độc lập với nhau

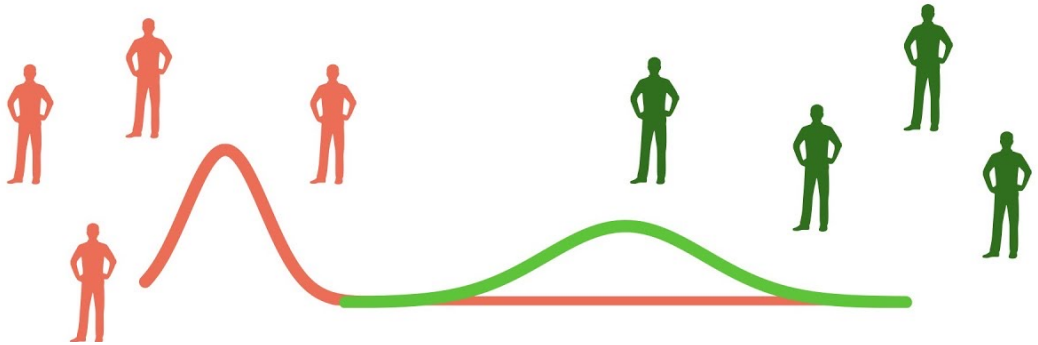


Có 3 loại phân bố xác suất phổ biến là:

- Gausian naïve bayes
- multinomial naïve bayes
- Bernoulli naïve bayes

Thuật toán Naïve Bayes

Guassian Naive Bayes....



sử dụng chủ yếu trong loại dữ liệu mà các thành phần là các biến liên tục. Với mỗi chiều dữ liệu I và một nhãn c, x_i tuân theo một phân phối chuẩn có kỳ vọng μ và phương sai σ

$$P(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Thuật toán Naïve Bayes

Multinomial naïve bayes: Mô hình này được sử dụng chủ yếu bài toán phân loại văn bản và vector đặc trưng được xây dựng dựa trên ý tưởng Bag of Words (BoW)

Multinomial Naïve Bayes Classifier

$$c_{MAP} = \underset{c \in C}{\operatorname{argmax}} P(x_1, x_2, \ldots, x_n \mid c) P(c)$$

$$c_{NB} = \underset{c \in C}{\operatorname{argmax}} P(c_j) \prod_{x \in X} P(x \mid c)$$

Bernoulli naïve bayes: Mô hình này được áp dụng cho các loại dữ liệu mà mỗi thành phần là một giá trị nhị phân (0|1).

The Bernoulli distribution

$$p(x) = P[X = x] = \begin{cases} q = 1 - p & x = 0 \\ p & x = 1 \end{cases}$$

3. Phân lớp comment trên Twitter sử dụng Naïve Bayes

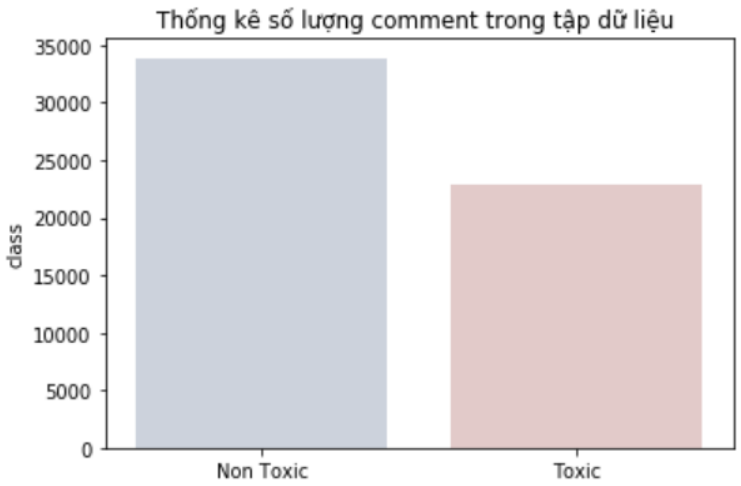
Ứng dụng Naïve Bayes trong phân văn bản

- Xây dựng mô hình học máy phân lớp văn bản thành 2 lớp: 1-Toxic 0-Non Toxic sử dụng thuật toán Naïve Bayes

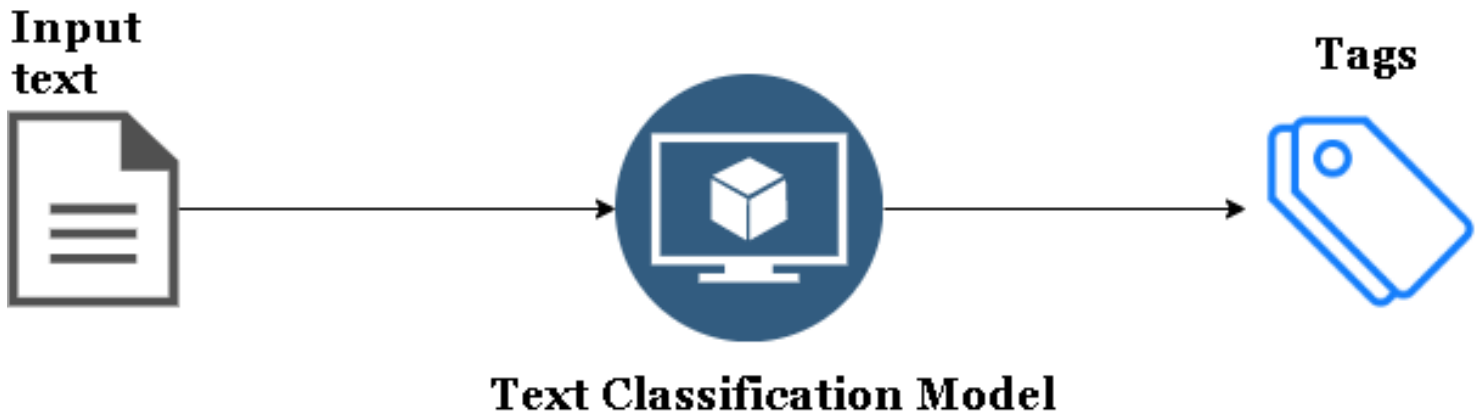
	A	
1	class	tweet
2	0	!!! RT @mayasolovely: As a woman you shouldn't complain about cleaning up your house. & as a man you should
3	1	!!!!! RT @mleew17: boy dats cold...tyga dwn bad for cuffin dat hoe in the 1st place!!
4	1	!!!!!!! RT @UrKindOfBrand Dawg!!!! RT @80sbaby4life: You ever fuck a bitch and she start to cry? You be confus
5	1	!!!!!!!!! RT @C_G_Anderson: @viva_based she look like a tranny
6	1	!!!!!!!!!!!!! RT @ShenikaRoberts: The shit you hear about me might be true or it might be faker than the bitch who told
7	1	!!!!!!!!!!!!!!!!!"@T_Madison_x: The shit just blows me..claim you so faithful and down for somebody but still fucking
8	1	!!!!!!"@__BrighterDays: I can not just sit up and HATE on another bitch .. I got too much shit going on!"
9	1	!!!!“@selfiequeenbri: cause I'm tired of you big bitches coming for us skinny girls!!”
10	1	" & you might not get ya bitch back & thats that "
11	1	" @rhythmixx_ :hobbies include: fighting Mariam"
12	1	" Keeks is a bitch she curves everyone " lol I walked into a conversation like this. Smh
13	1	" Murda Gang bitch its Gang Land "
14	1	" So hoes that smoke are losers ? " yea ... go on IG
15	1	" bad bitches is the only thing that i like "
16	1	" bitch get up off me "
17	1	" bitch nigga miss me with it "

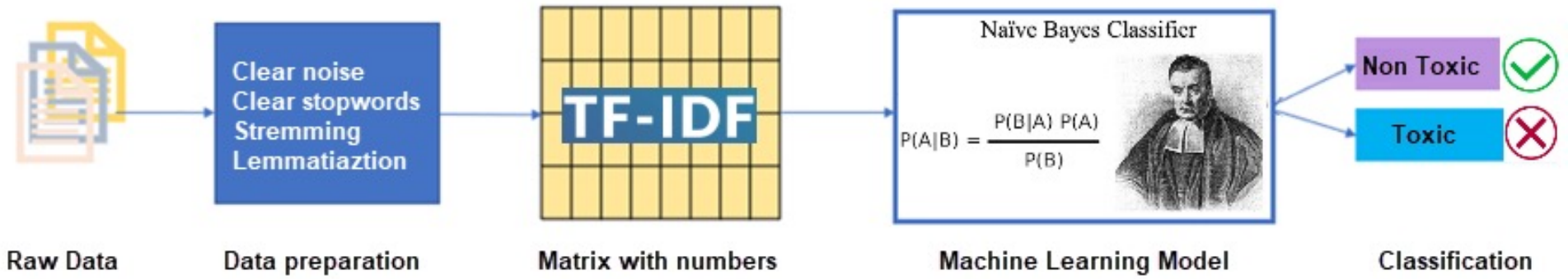
Data_NLP

```
Thông kê tập dữ liệu:
class    56700
tweet    56700
dtype: int64
----------------------------------
Thông kê số lượng comment theo lớp:
0    33847
1    22853
Name: class, dtype: int64
```



Ứng dụng Naïve Bayes trong phân văn bản





Ứng dụng Naïve Bayes trong phân văn bản

- **1. Chuẩn bị dữ liệu:**

STT	Comments ban đầu	Comments đã xử lý
1	!!!!!!!!!!!!!!!!!!"@T_Madison_x: The shit just blows me..claim you so faithful and down for somebody but still fucking with hoes! 😂😂😂"	shit blow faithful somebody still fuck hoe
2	" got ya bitch tip toeing on my hardwood floors " 😂 http://t.co/cOU2WQ5L4q	get ya bitch tip toe hardwood floor
3	"@Dunderball: I'm an early bird and I'm a night owl, so I'm wise and have worms."	early bird night owl wise worm
4	"@Tmacc_GFG: “@VoiceOfDStreetz: "@Tmacc_GFG: “@tizzimarie: No slushes 😥”hoes nasty anyway fam"😴”them hoes taste like meds"🍇🍇🍼	slush hoe nasty anyway fam hoe taste like meds
5	beÃ°ÂŸÂ’Â“ Ã°ÂŸÂ’Â“@user @user @user @user @user @user @user @user @user @user @user	

Ứng dụng Naïve Bayes trong phân văn bản

- **1. Chuẩn bị dữ liệu:**

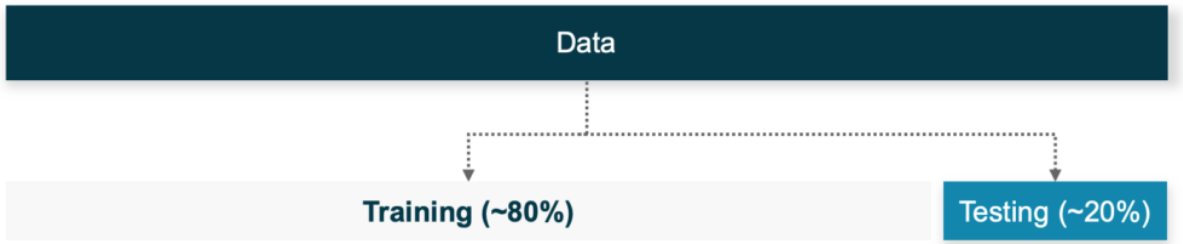
```
#Thống kê số row null
#trong tập dữ liệu sau xử lý:
data_finish.isnull().sum()
```

```
class        0
tweet_ok    43
```

```
#Thực hiện xóa tất cả các row có phần tử NaN
data_NLP = data_finish.dropna()
data_NLP.info()
```

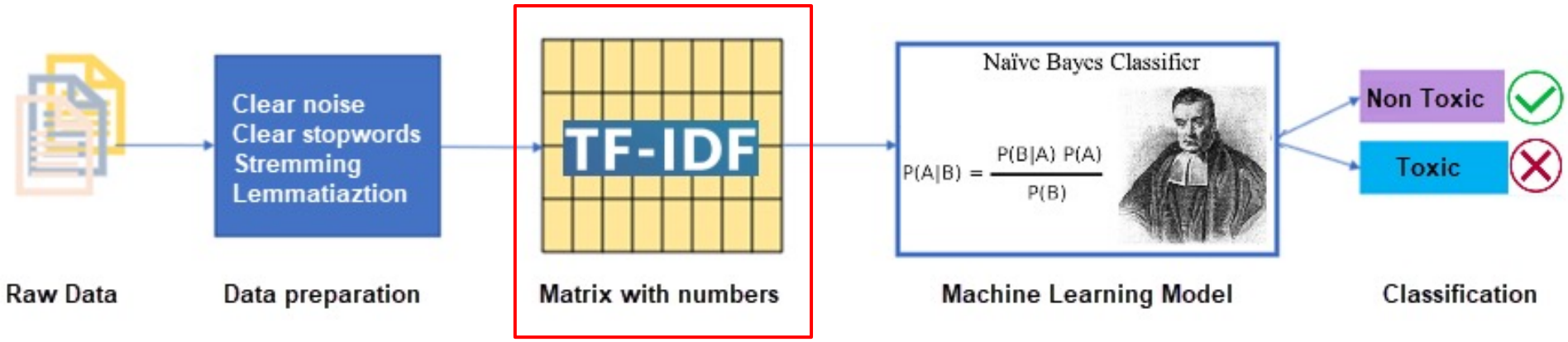
```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 56657 entries, 0 to 56699
Data columns (total 2 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   class     56657 non-null  int64
 1   tweet_ok  56657 non-null  object
dtypes: int64(1), object(1)
memory usage: 1.3+ MB
```

class	tweet	tweet_ok
0	!!! RT @mayasolovely: As a woman you shouldn't complain about cleaning up your house. & as a man you should always take the trash out...	woman complain clean house man always take trash
1	!!!!! RT @mleew17: boy dats cold...tyga dwn bad for cuffin dat hoe in the 1st place!!	boy dats dwn bad cuffin dat hoe st place
1	!!!!!!! RT @UrKindOfBrand Dawg!!!! RT @80sbaby4life: You ever fuck a bitch and she start to cry? You be confused as shit	dawg ever fuck bitch start cry confuse shit
1	!!!!!!!!! RT @C_G_Anderson: @viva_based she look like a tranny	look like tranny
1	!!!!!!!!!!!! RT @ShenikaRoberts: The shit you hear about me might be true or it might be faker than the bitch who told it to ya 	shit hear might true might faker bitch tell ya
1	!!!!!!!!!!!!!!!!!!"@T_Madison_x: The shit just blows me..claim you so faithful and down for somebody but still fucking with hoes! 😂😂😂"	shit blow faithful somebody still fuck hoe



Ứng dụng Naïve Bayes trong phân văn bản

- **2. Trích chọn đặc trưng TF-IDF**



Ứng dụng Naïve Bayes trong phân văn bản

- **2. Trích chọn đặc trưng TF-IDF**

- Thuật ngữ **TF-IDF (Term Frequency – Inverse Document Frequency)** là một phương thức thống kê được biết đến rộng rãi để xác định độ quan trọng của một từ trong đoạn văn bản trong một tập nhiều đoạn văn bản khác nhau.

- TF-IDF xác định trọng số của một từ trong văn bản thu được qua thống kê thể hiện mức độ quan trọng của từ này trong một văn bản, mà bản thân văn bản đang xét nằm trong một tập hợp các văn bản. Giá trị TF-IDF cao thể hiện độ quan trọng cao và nó phụ thuộc vào số lần từ xuất hiện trong văn bản nhưng bù lại bởi tần suất của từ đó trong tập dữ liệu.

Ứng dụng Naïve Bayes trong phân văn bản

- **2. Trích chọn đặc trưng TF-IDF**

  - *TF (Term Frequency)* – Tần suất xuất hiện của từ, là số lần từ xuất hiện trong văn bản. Vì các văn bản có thể có độ dài ngắn khác nhau nên một số từ có thể xuất hiện nhiều lần trong một văn bản dài hơn là một văn bản ngắn

$$TF(t,d) = \frac{f(t,d)}{\max\{f(w,d):w \in d\}}$$

Trong ɑo:

- TF(t, d) - Tần suất xuất hiện của từ t trong văn bản d.
- f(t,d) - Số lần xuất hiện của từ trong văn bản d.
- max{f(w,d): w ∈ d} – Số lần xuất hiện của từ có số lần xuất hiện nhiều nhất trong văn bản d.

  - *IDF (Inverse Document Frequency)* – Nghịch đảo tần suất của văn bản, giúp đánh giá tầm quan trọng của một từ. Khi tính tần số xuất hiện TF thì các từ đều được coi là quan trọng như nhau. Tuy nhiên có một số từ thường được được sử dụng nhiều nhưng không quan trọng để thể hiện ý nghĩa của đoạn văn. Vì vậy ta cần giảm đi mức độ quan trọng của những từ đó bằng cách sử dụng IDF:

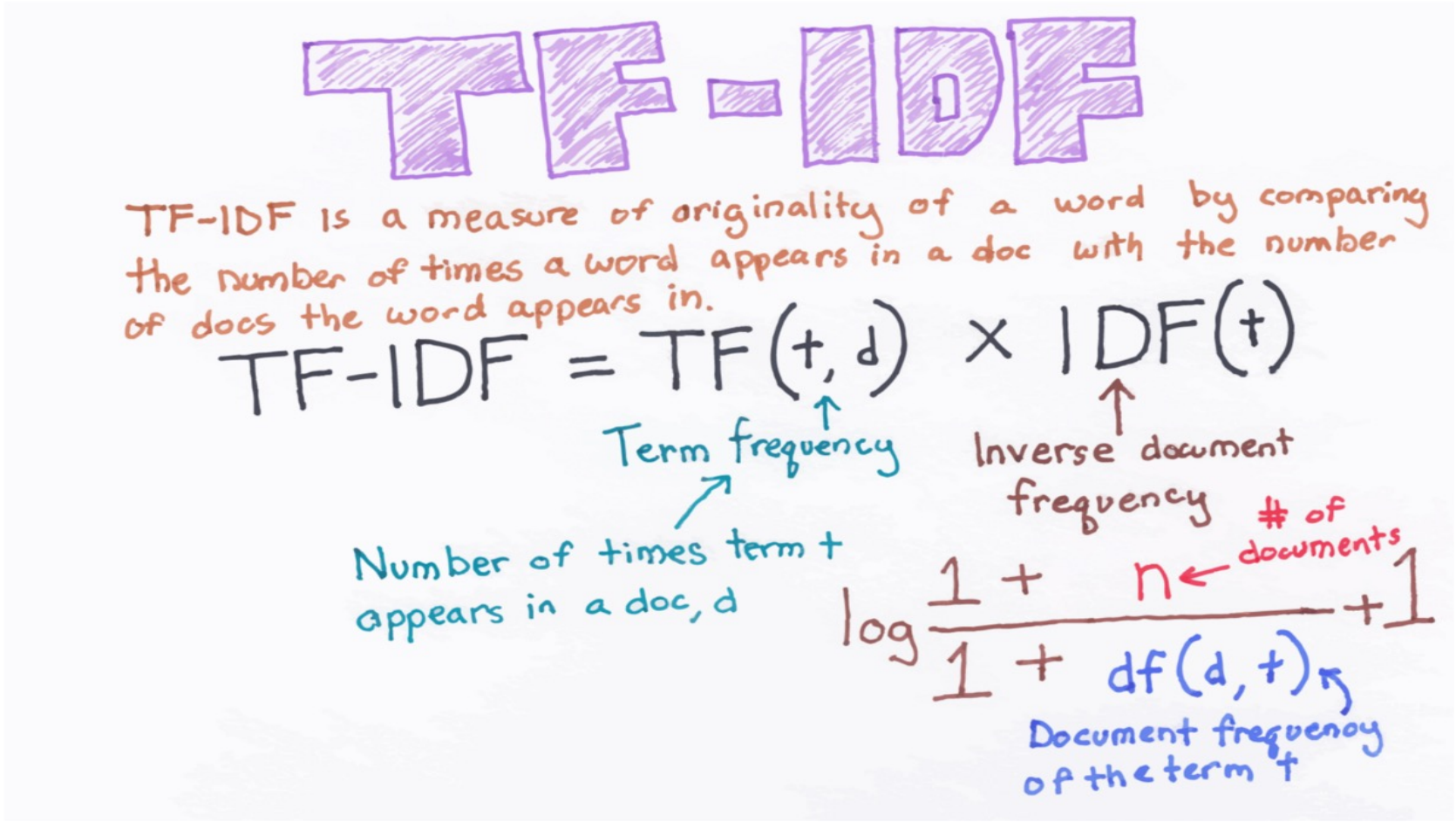
$$IDF(t,D) = log\frac{|D|}{|\{d \in D : t \in d\}|}$$

Trong đó:

- IDF(t, D) – Giá trị IDF của từ t trong tập văn bản D.
- |D| - Tổng số văn bản trong tập D.
- |{d ∈ D : t ∈ d}| – Thể hiện số văn bản trong tập D có chứa từ t.

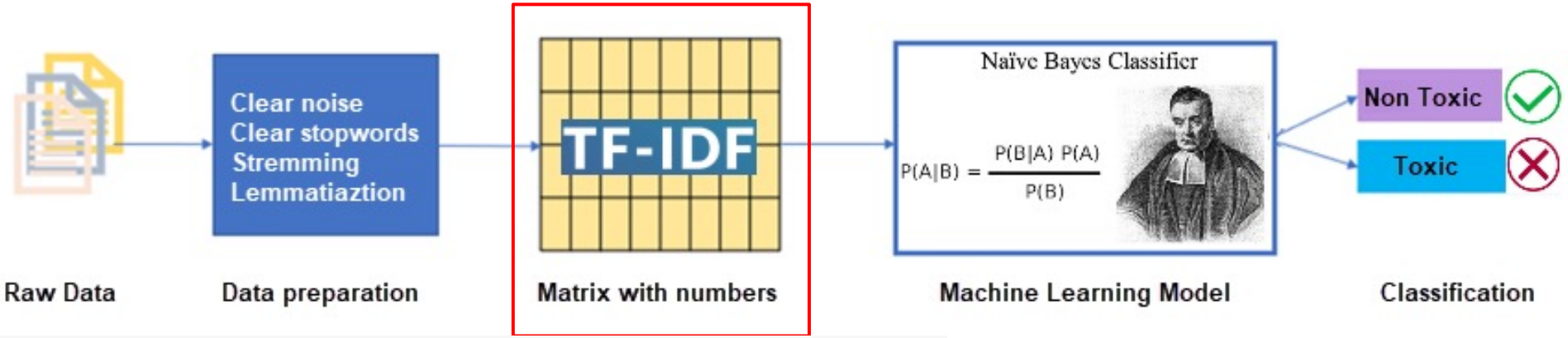
Ứng dụng Naïve Bayes trong phân văn bản

- **2. Trích chọn đặc trưng TF-IDF**



Ứng dụng Naïve Bayes trong phân văn bản

- **2. Trích chọn đặc trưng TF-IDF**



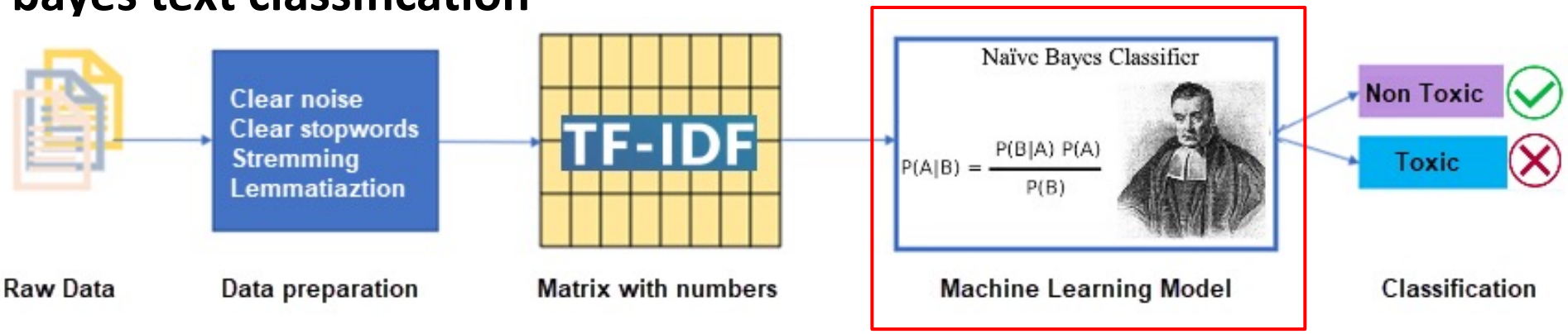

```python
# Tính TF-IDF cho tập dữ liệu
from sklearn.feature_extraction.text import TfidfVectorizer
#Convert a collection of raw documents to a matrix of TF-IDF features.
vector = TfidfVectorizer(analyzer='word',
                         max_features=20000,
                         stop_words = 'english')

vector.fit(data_NLP['tweet_ok'])
xtrain_tfidf = vector.transform(train_x)
xtest_tfidf = vector.transform(test_x)
print('Kết quả Vector hóa tập Train sang dạng số:')
print(xtrain_tfidf.data)
print(xtrain_tfidf.shape)
print('Kết quả Vector hóa tập Test sang dạng số:')
print(xtest_tfidf.data)
print(xtest_tfidf.shape)
```

```
Kết quả Vector hóa tập Train sang dạng số:
[0.54907788 0.53160745 0.64490852 ... 0.33670852 0.13308591 0.26849291]
(45325, 20000)
Kết quả Vector hóa tập Test sang dạng số:
[0.73386444 0.51887036 0.43842506 ... 0.58960476 0.39331306 0.52415607]
(11332, 20000)
```

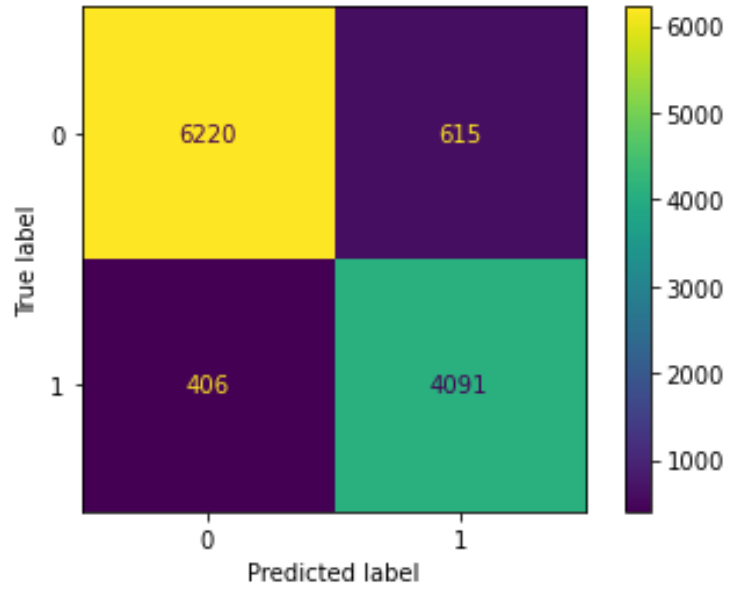
Ứng dụng Naïve Bayes trong phân văn bản

- **3. Naïve bayes text classification**

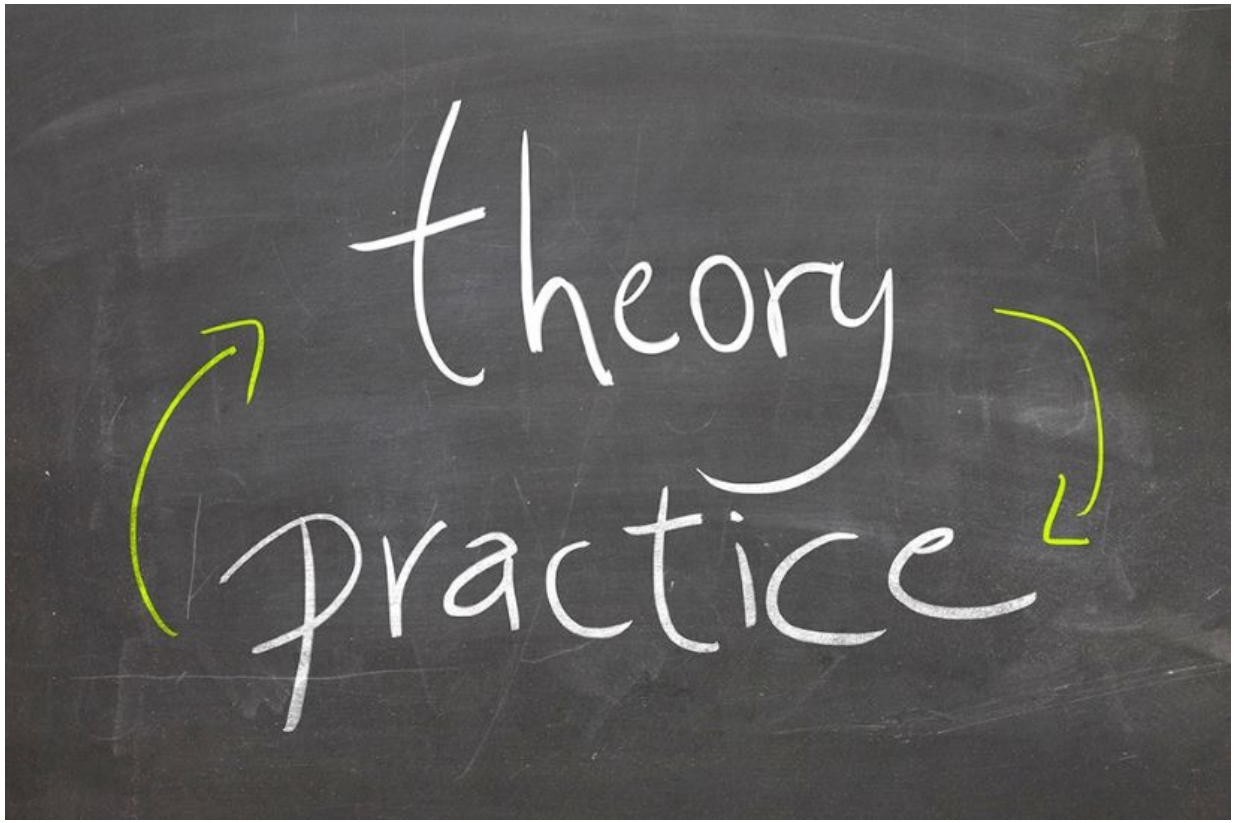



```
1  from sklearn.metrics import accuracy_score
2  #Dự đoán và tính đoán độ chính xác của model trên tập Test:
3  y_pred = MultiNB.predict(X_test_tfidf)
4
5  acc1 = round(accuracy_score(y_test, y_pred)*100, 2)
6  print('1.Độ chính xác của mô hình trên tập Test: ', acc1, '%')
7
8  acc2 = accuracy_score(y_test, y_pred, normalize=False)
9  print('2.Tổng số mẫu dự đoán đúng:', acc2, ' /', len(y_test))
10 print('3.Tổng số mẫu dự đoán sai:', len(y_test) - acc2, ' /', len(y_test))
```

```
1.Độ chính xác của mô hình trên tập Test:  90.99 %
2.Tổng số mẫu dự đoán đúng: 10311  / 11332
3.Tổng số mẫu dự đoán sai: 1021  / 11332
```



Học viên theo dõi và thực hành theo các bước trong file Jupyter notebook




HANOI UNIVERSITY OF MINING AND GEOLOGY
KHOA
CÔNG NGHỆ THÔNG TIN
FACULTY OF INFORMATION TECHNOLOGY

Thank you!