# KỸ THUẬT LỌC TIN NHẮN RÁC SMS

## Khai phá dữ liệu

Nguyễn Chí Dũng - 20002040 (Nhóm trưởng)
Đinh Tiến Dũng – 20002039
Nguyễn Hoàng Giang – 20002048
Nguyễn Thị Khánh Linh - 20002066

K65 Khoa học dữ liệu
Khoa Toán - Cơ - Tin học
Trường Đại học Khoa học Tự nhiên, ĐHQGHN

Tháng 05/2023

# Outline

# Table of Contents

# Xác định vấn đề

## Hiện trạng tin nhắn rác SMS tại Việt Nam

- Theo một báo cáo vào năm 2015 của BKAV, một hãng bảo mật thông tin, có 13,9 triệu tin nhắn rác được gửi đến người dùng điện thoại di động mỗi ngày ở Việt Nam và cứ hai người thì có một người nhận được tin nhắn rác
- Spam SMS gây ra tổn thất tài chính bằng cách phản ứng lại chúng. Người dùng có thể vô tình gọi đến các số giá cước cao cấp hoặc đăng ký các dịch vụ đắt tiền bằng cách trả lời các tin nhắn này.

# Xác định vấn đề

## Kỹ thuật lọc tin nhắn rác SMS

- Hơn nữa, chúng ta có thể gặp phải một số rủi ro khi truy cập các trang đường link trong Spam SMS, bị mất hoặc bị đánh cắp dữ liệu, hình ảnh trong điện thoại thậm trí tổn hại nặng hơn thế. Các nhà khai thác mạng di động cũng bị thiệt hại về tài chính vì họ có thể mất người dùng hoặc chi nhiều hơn cho việc ngăn chặn thư rác.
- Trong khi đó, có ít nhất 120 triệu người dùng di động tích cực trên thị trường theo một báo cáo của Bộ Thông tin và Truyền thông, được công bố vào năm 2015.

# Bộ dữ liệu sử dụng

## SMS Spam Collection

- SMS Spam Collection là bộ dữ liệu được sử dụng rộng rãi trong lĩnh vực xử lý ngôn ngữ tự nhiên và phân loại tin nhắn rác. Bộ dữ liệu này được thu thập tại Anh và chứa hơn 5572 tin nhắn SMS bằng tiếng Anh, trong đó có 4825 tin nhắn thường (ham) và 747 tin nhắn rác (spam).

- Bộ dữ liệu gồm các tệp văn bản, chứa thông tin về một tin nhắn, bao gồm cả nội dung của tin nhắn, loại tin nhắn (thông thường hoặc tin rác) và một số thông tin khác như thời gian gửi và số điện thoại của người gửi.

| | |
|---|---|
| ham | Today am going to college so am not able to atten the class. |
| ham | I'm in class. Will holla later |
| ham | Easy ah?sen got selected means its good.. |
| ham | Mmm thats better now i got a roast down me! i'd b better if i had a few drinks down me 2! Good indian? |
| spam | We know someone who you know that fancies you. Call 09058097218 to find out who. POBox 6 LS15HB 150p |
| ham | Come round it's . |
| ham | Do 1 thing! Change that sentence into: "Because i want 2 concentrate in my educational career im leaving here.." |
| spam | 1000's flirting NOW! Txt GIRL or BLOKE & ur NAME & AGE eg GIRL ZOE 18 to 8007 to join and get chatting! |
| ham | I walked an hour 2 c u! doesn't that show I care y wont u believe im serious? |
| spam | 18 days to Euro2004 kickoff! U will be kept informed of all the latest news and results daily. Unsubscribe send GET EURO STOP to 83222. |

# Bộ dữ liệu sử dụng

## SMS Spam Collection

- Được sử dụng để huấn luyện các mô hình phân loại tin nhắn là tin nhắn rác hoặc tin nhắn thường.

- Bộ dữ liệu SMS Spam Collection chỉ chứa các tin nhắn tiếng Anh và không phải là một bộ dữ liệu đại diện cho mọi loại tin nhắn rác. Do đó khi sử dụng bộ dữ liệu này, cần xem xét kỹ càng để đảm bảo tính khả dụng của nó đối với ứng dụng cụ thể.

- Bộ dữ liệu này bao gồm 2 cột, trong đó:

+ Cột đầu tiên: Ứng với 2 nhãn: spam hoặc ham.
+ Cột còn lại: Chứa các tin nhắn tiếng Anh.

# Table of Contents

# Tiền xử lý dữ liệu

## Quy trình

Gồm 4 giai đoạn chính:

1. Data Cleaning: Làm sạch dữ liệu
2. EDA (Exploratory Data Analysis): Phân tích khai phá dữ liệu
3. Text Preproccessing: Tiền xử lý văn bản
4. Text Embedding: Nhúng văn bản

## Data Cleaning

- Data Cleaning là quá trình làm sạch và chuẩn hóa dữ liệu nhằm đảm bảo tính chính xác và đầy đủ của dữ liệu. Trong quá trình này, ta sẽ xác định và loại bỏ các giá trị dữ liệu lỗi, thiếu sót, trùng lặp, không chính xác hoặc không hợp lệ.

- Khi dữ liệu đã được làm sạch, nó sẽ trở nên đáng tin cậy hơn và được sử dụng để phân tích, đưa ra các quyết định có cơ sở.

# Data Cleaning

Trong project này, quy trình làm sạch dữ liệu trải qua các công đoạn sau:

- Kiểm tra thông tin bộ dữ liệu. Sau đó loại bỏ đi các trường trống (nếu có) và format lại tên các trường.
- Mã hóa nhãn thành các giá trị nằm trong (0, n-1); với n là số nhãn, bộ dữ liệu có 2 nhãn "ham: 0", "spam: 1".
- Kiểm tra các giá trị trống (null) và các giá trị trùng lặp (duplicate).

Bộ dữ liệu này có 403 giá trị trùng.

$\rightarrow$ Sử dụng Drop duplicate để loại bỏ các giá trị trùng.

## Exploratory Data Analysis

Với EDA, người phân tích sẽ khám phá và phân tích dữ liệu để hiểu và tìm ra mối quan hệ, xu hướng và đặc điểm của các biến. Từ đó có thể hiểu dữ liệu một cách sâu sắc hơn và đưa ra những phát hiện, giả định ban đầu cho các bước tiếp theo của quy trình phân tích dữ liệu.

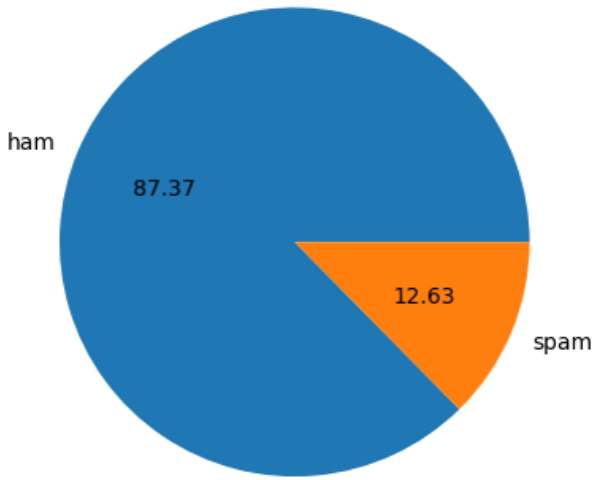
## Exploratory Data Analysis

Kiểm tra số lượng dữ liệu thuộc về 2 nhãn "ham" và "spam"

Nhãn 0 (ham) có 4516 giá trị

Nhãn 1 (spam) có 653 giá trị

$\rightarrow$ Bộ dữ liệu không cân bằng (Do bản chất dữ liệu trong thực tế là tin nhắn thường nhiều hơn tin nhắn rác).

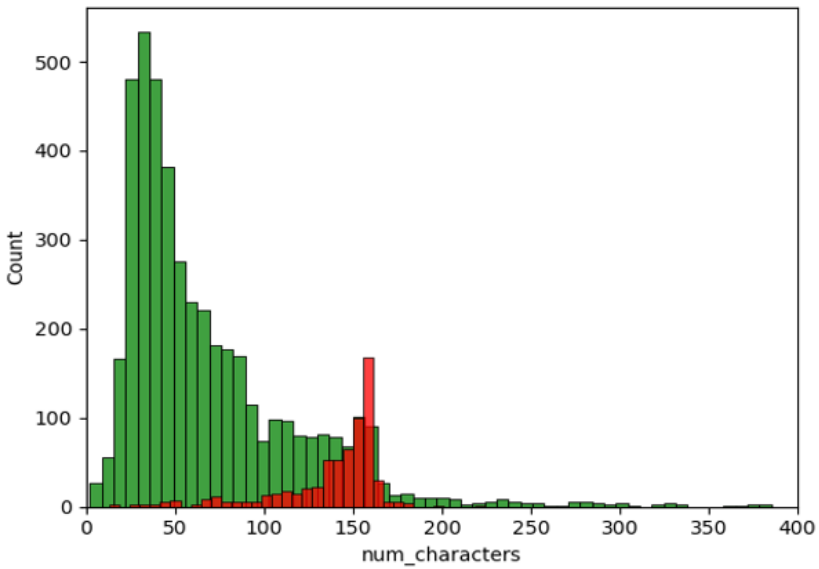


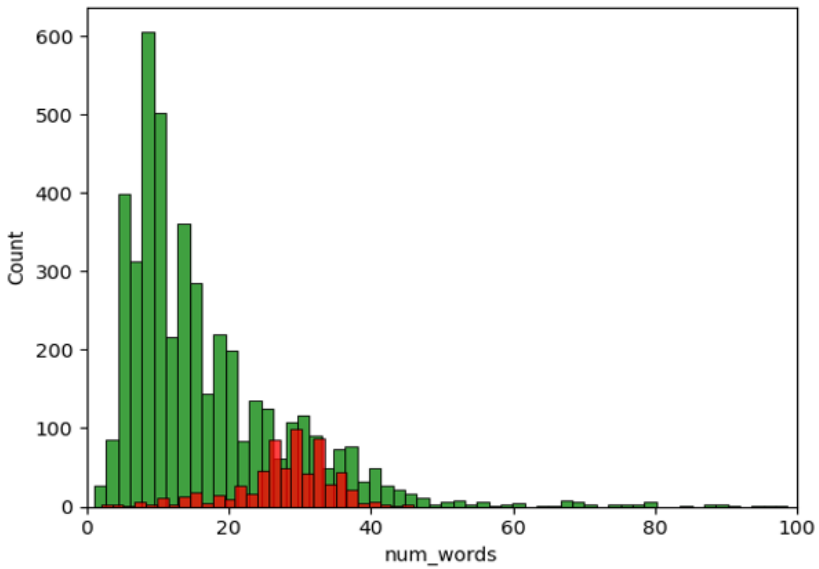
## Exploratory Data Analysis

Sử dụng thư viện 'nltk' (natural language toolkit) phân tích:

1. Số ký tự, số từ trong tin nhắn rác nhiều hơn

2. Tần suất xuất hiện của tin nhắn thường nhiều hơn tin nhắn rác



## Text Preprocessing

1. Tokenization: Tách một cụm từ, câu, đoạn văn thành các đơn vị nhỏ - tokens.

2. Loại bỏ các ký tự đặc biệt và dấu câu.

3. Chuyển đổi chữ hoa thành chữ thường Ví dụ, "Hello" và "hello" sẽ được coi là cùng một từ trong quá trình phân tích văn bản.

4. Loại bỏ các stopword để tập trung vào những từ mang thông tin quan trọng.

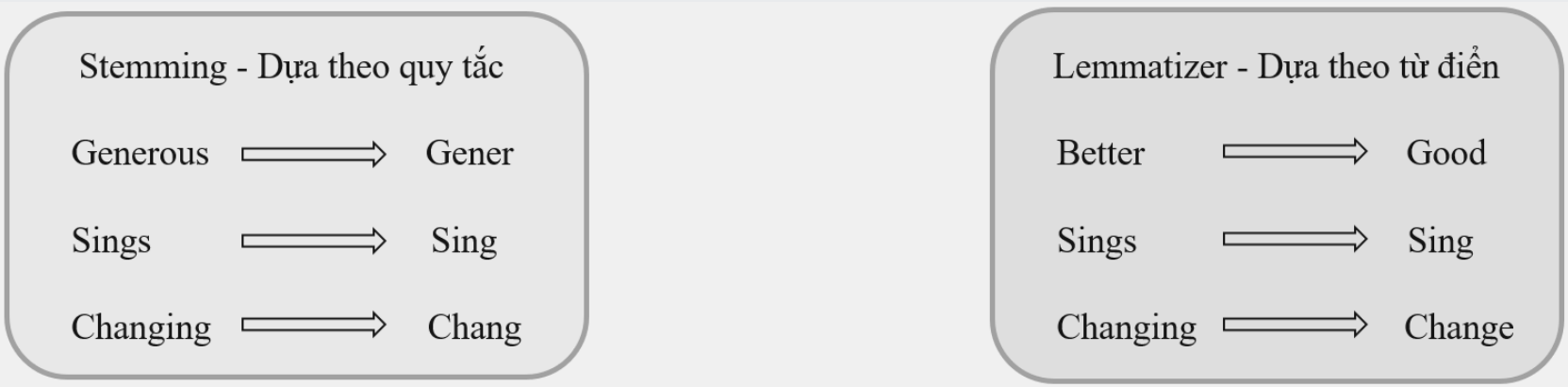
5. Chuẩn hóa các từ đồng nghĩa bằng cách đưa về các từ gốc (stemming, lemmatizer).

6. Kiểm tra các từ thường xuyên xuất hiện trong văn bản.

## Chuẩn hóa các từ về dạng gốc

Stemming: Quá trình chuyển đổi các từ trong văn bản về dạng từ gốc bằng cách loại bỏ các hậu tố. Ví dụ: Từ "running" sẽ được chuyển thành từ "run" thông qua việc loại bỏ hậu tố "ing".

Lemmatizer: Quá trình chuyển đổi các từ trong văn bản về dạng từ điển gốc (lemma) bằng cách sử dụng thông tin về ngữ cảnh và từ điển. Ví dụ: "better" sẽ được chuyển đổi về từ "good".

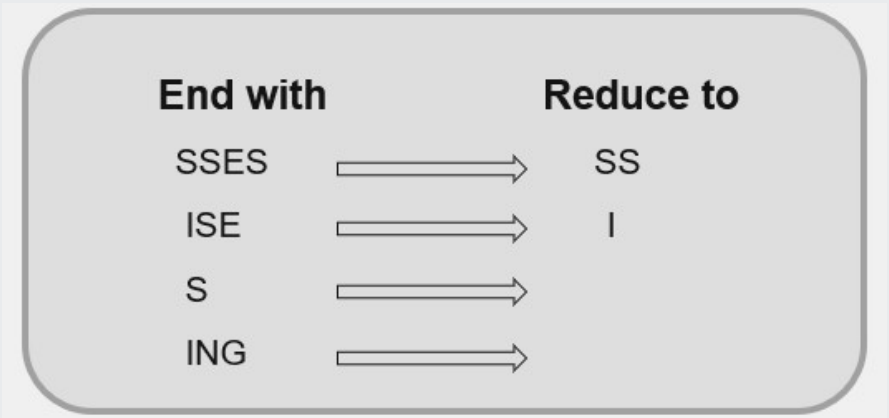


## Porter's stemmer

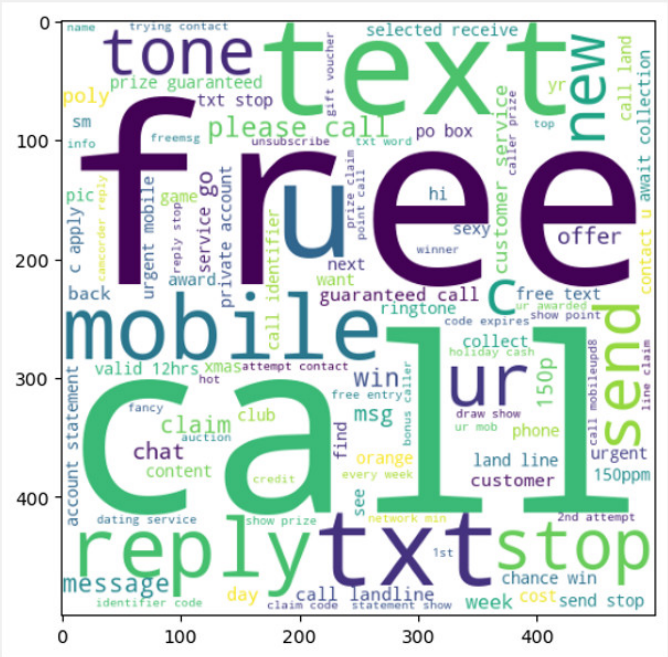
Một trong những phương pháp chuyển từ về dạng gốc phổ biến nhất được đề xuất vào năm 1980, dựa trên quy tắc các hậu tố được tạo thành từ sự kết hợp của các hậu tố nhỏ hơn và đơn giản hơn.

Chỉ áp dụng được trong tiếng Anh, đầu ra của thuật toán không nhất thiết phải là một từ có nghĩa.

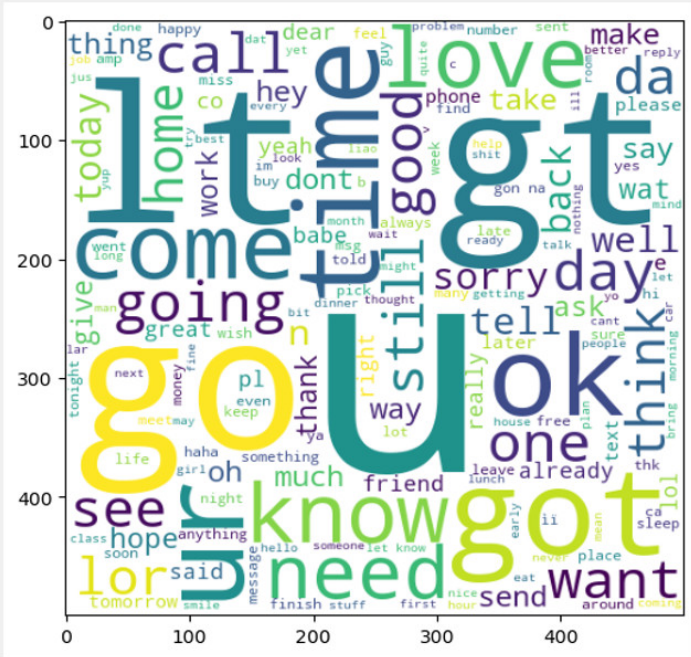
Một số quy tắc của Porter's stemmer:



# Tiền xử lý dữ liệu

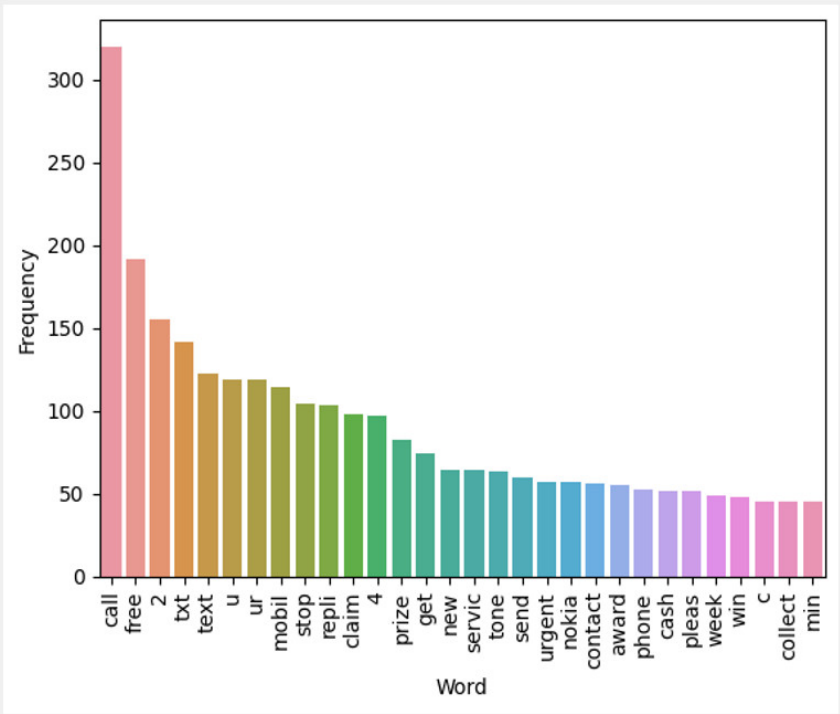



Spam

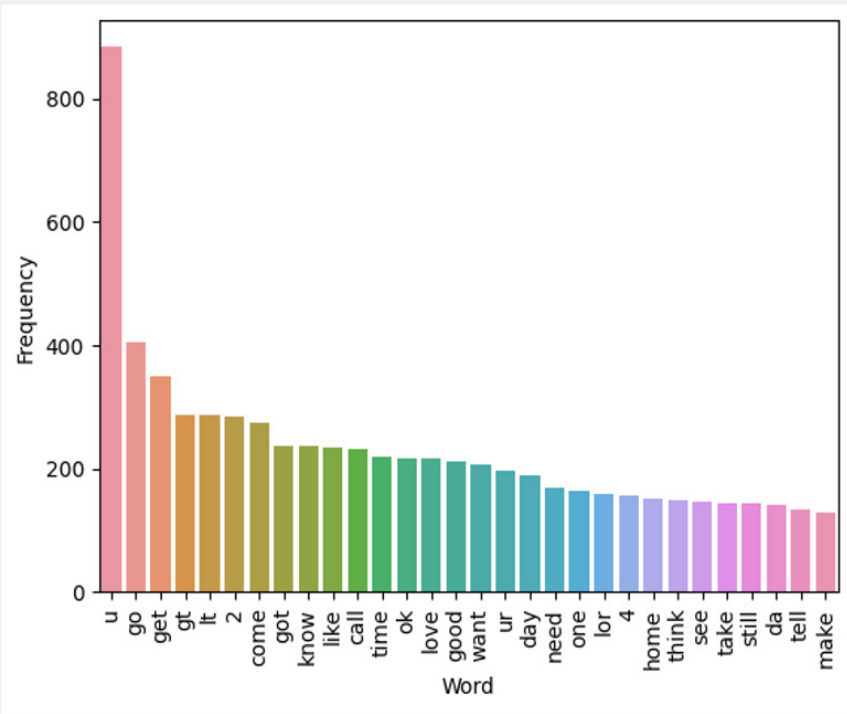



Ham

# Tiền xử lý dữ liệu




Spam




Ham

## Text Embedding

1. Text embedding là quá trình biểu diễn một từ, cụm từ hoặc văn bản thành vector số trong không gian đa chiều dựa trên mối quan hệ ngữ nghĩa giữa chúng.

2. Các vectơ số này được tạo ra từ các thuật toán biểu diễn từ, ví dụ như các thuật toán: Word2Vec, GloVe, FastText, TF-IDF.

3. Các vector số được xây dựng để có khoảng cách và hình dạng tương tự giữa các từ mang nghĩa tương đồng. Nhờ vậy, các từ được biểu diễn bằng các vector số này có thể được sử dụng để phân tích ngữ nghĩa của câu hoặc văn bản.

## Thuật toán TF-IDF

1. TF-IDF là viết tắt của Term Frequency-Inverse Document Frequency.

2. Phương pháp đánh giá độ quan trọng của một từ dựa trên tần suất xuất hiện trong văn bản và tần suất xuất hiện trong bộ dữ liệu.

3. Term Frequency (Tần suất xuất hiện của từ): Chỉ số này cho biết tần suất xuất hiện của một từ trong một tài liệu văn bản cụ thể.

4. Inverse Document Frequency (Nghịch đảo tần suất tài liệu): Chỉ số này cho biết tần suất xuất hiện của một từ trong cả bộ dữ liệu.

## Thuật toán TF-IDF

$$w_{i,j} = tf_{i,j} * \log(\frac{N}{df_i})$$

i: Từ cần tính giá trị TF-IDF

j: Tài liệu cần tính giá trị TF-IDF

N: Tổng số tài liệu trong bộ dữ liệu

$tf_{i,j}$: Tần suất xuất hiện từ i trong tài liệu j

$df_i$: Tổng số tài liệu chứa từ i

**Ví dụ:**

**Câu 1:** "This movie is very scary and long"

**Câu 2:** "This movie is not scary and is slow"

**Câu 3**: "This movie is spooky and good"

Xét câu 2, có các từ: "This", "movie", "is", "not", "scary", "and", "is", "slow"

$$TF(This) = \frac{1}{8}$$

$$IDF(This) = \log(\frac{3}{3}) = 0$$

$$TF\text{-}IDF(This) = \frac{1}{8}.0 = 0$$

→ Từ "This" có độ quan trọng bằng 0 trong bộ dữ liệu trên.

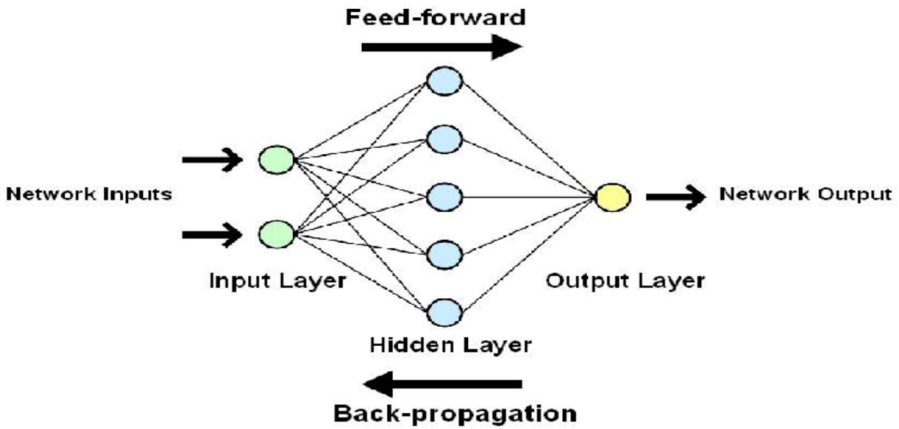
# Thuật toán TF-IDF

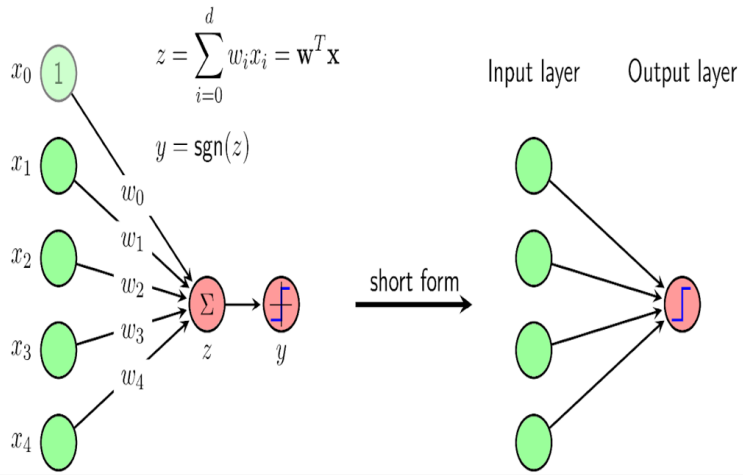
| Term | Câu 1 | Câu 2 | Câu 3 | IDF | TF-IDF (Câu 1) | TF-IDF (Câu 2) | TF-IDF (Câu 3) |
|---|---|---|---|---|---|---|---|
| This | 1 | 1 | 1 | 0.00 | 0.000 | 0.000 | 0.000 |
| movie | 1 | 1 | 1 | 0.00 | 0.000 | 0.000 | 0.000 |
| is | 1 | 2 | 1 | 0.00 | 0.000 | 0.000 | 0.000 |
| very | 1 | 0 | 0 | 0.48 | 0.068 | 0.000 | 0.000 |
| scary | 1 | 1 | 0 | 0.18 | 0.025 | 0.022 | 0.000 |
| and | 1 | 1 | 1 | 0.00 | 0.000 | 0.000 | 0.000 |
| long | 1 | 0 | 0 | 0.48 | 0.068 | 0.000 | 0.000 |
| not | 0 | 1 | 0 | 0.48 | 0.000 | 0.060 | 0.000 |
| slow | 0 | 1 | 0 | 0.48 | 0.000 | 0.060 | 0.000 |
| spooky | 0 | 0 | 1 | 0.48 | 0.000 | 0.000 | 0.080 |
| good | 0 | 0 | 1 | 0.48 | 0.000 | 0.000 | 0.080 |

# Table of Contents

# Multi Layer Perceptron

Artificial Neuron NetWork (mạng nơ-ron) là một mô hình được xây dựng dựa trên cách hoạt động bộ não con người với các neuron được kết nối với nhau. ANN gồm 3 thành phần chính: input layer, output layer, hidden layers



## Activation function

$$sigmoid(z) = \frac{1}{[1 + exp(-z)]}$$

$$tanh(z) = \frac{[1 + exp(z)] - [1 + exp(-z)]}{[1 + exp(z)] + [1 + exp(-z)]}$$

$$RELU : f(z) = \max(0, z)$$

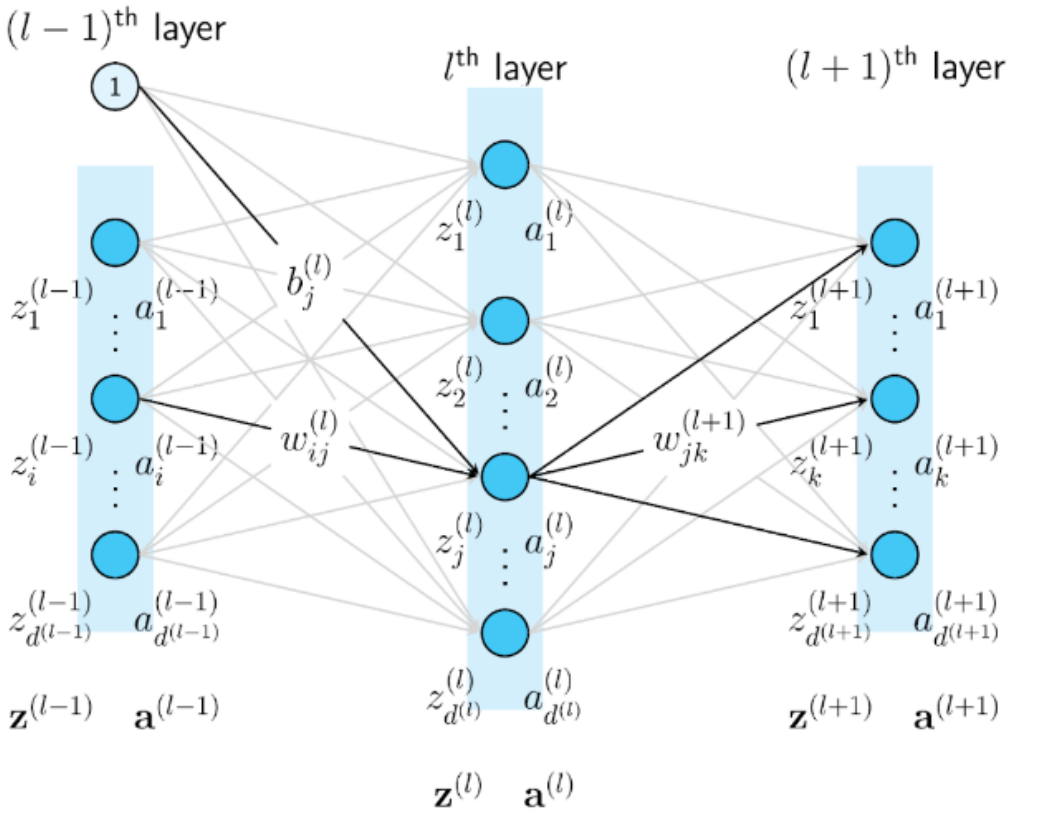
- Nhược điểm của $sigmoid(z)$ và $tanh(z)$: Với $z \gg 1$ thì$f'(z) \approx 0$.
- ReLU : Với $z > 0, f'(z) = 1$ và $z < 0, f'(z) = 0$.

Feedforward:

Quá trình tính *predicted output* $\widehat{y}$ với input $\mathbf{x}$:

$$
\begin{aligned}
\mathbf{a}^{(0)} &= \mathbf{x} \\
z_i^{(l)} &= \mathbf{w}_i^{(l)T}\mathbf{a}^{(l-1)} + b_i^{(l)} \\
\mathbf{z}^{(l)} &= \mathbf{W}^{(l)T}\mathbf{a}^{(l-1)} + \mathbf{b}^{(l)}, \;\; l = 1, 2, \ldots, L \\
\mathbf{a}^{(l)} &= f(\mathbf{z}^{(l)}), \;\; l = 1, 2, \ldots, L \\
\hat{\mathbf{y}} &= \mathbf{a}^{(L)}
\end{aligned}
$$

# Multi Layer Perceptron




$$\mathbf{W}^{(l)} \in \mathbb{R}^{d^{(l-1)} \times d^{(l)}}$$

$$\mathbf{b}^{(l)} \in \mathbb{R}^{d^{(l)} \times 1}$$

$$z_j^{(l)} = \mathbf{w}_j^{(l)T} \mathbf{a}^{(l-1)} + b_j^{(l)}$$

$$\mathbf{z}^{(l)} = \mathbf{W}^{(l)T} \mathbf{a}^{(l-1)} + \mathbf{b}^{(l)}$$

$$\mathbf{a}^{(l)} = f(\mathbf{z}^{(l)})$$

# Multi Layer Perceptron

Giả sử J(W, b, X, Y) là một hàm mất mát của bài toán, trong đó W, b là tập hợp tất cả các ma trận trọng số giữa các layers và biases của mỗi layer. Để có thể áp dụng Backpropagation (Gradient Descent), chúng ta cần tính được:

$$\frac{\partial J}{\partial \mathbf{W}^{(l)}}; \frac{\partial J}{\partial \mathbf{b}^{(l)}}, \quad l = 1, 2, \ldots, L$$

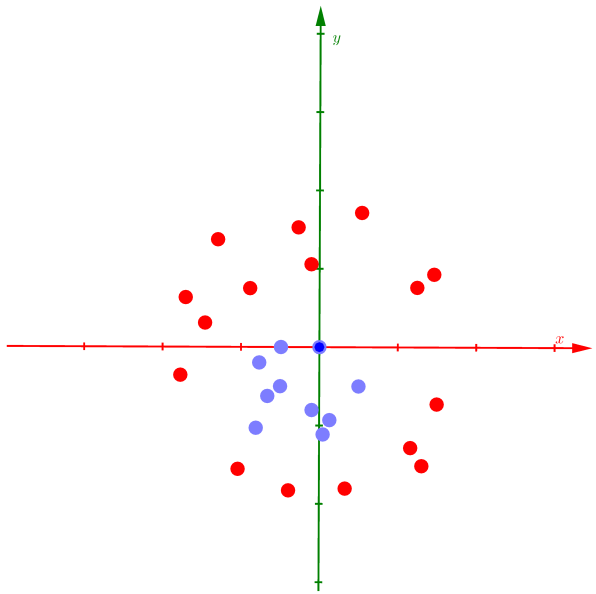
Tại mỗi bước lặp GD, ta cập nhật lại các trọng số $W^{(l)}$ và bias $b^{(l)}$ :

$$w_{ij}^{(I)} = w_{ij}^{(I)} - \alpha \frac{\partial \boldsymbol{J}(\boldsymbol{W}, \boldsymbol{b}, \boldsymbol{X}, \boldsymbol{Y})}{\partial w_{ij}^{(I)}} = w_{ij}^{(I)} - \alpha \frac{\partial \boldsymbol{J}(\boldsymbol{W}, \boldsymbol{b})}{\partial w_{ij}^{(I)}}$$

$$b_{i}^{(I)} = b_{i}^{(I)} - \alpha \frac{\partial \boldsymbol{J}(\boldsymbol{W}, \boldsymbol{b}, \boldsymbol{X}, \boldsymbol{Y})}{\partial b_{i}^{(I)}} = b_{i}^{(I)} - \alpha \frac{\partial \boldsymbol{J}(\boldsymbol{W}, \boldsymbol{b})}{\partial b_{i}^{(I)}}$$

## Support Vector Machine

Trường hợp dữ liệu tách được tuyến tính gần như tách được tuyến tính, ta có Hard Margin & Soft Margin SVM. Xét trường hợp dữ liệu thực sự không tách được tuyến tính như trong hình $\leftrightarrow$ không sử dụng được Hard Soft Margin SVM.



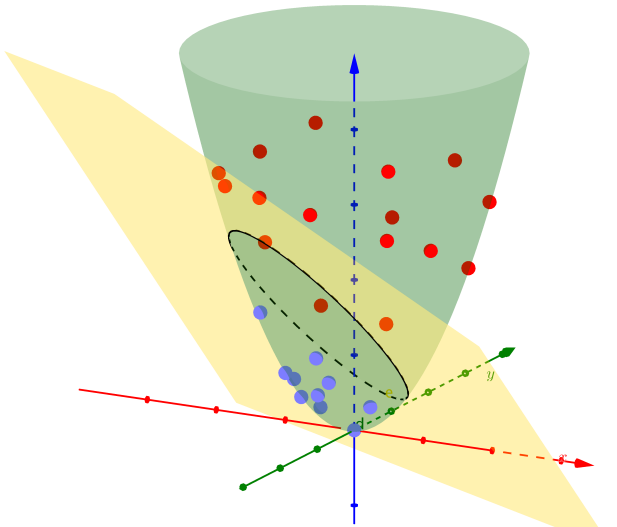
## Support Vector Machine

Nếu bổ sung chiều thứ ba là hàm số của hai chiều đã có

$$z = x^2 + y^2$$

Dữ liệu mới được phân bố trong không gian 3D: Những điểm xa tâm sẽ có giá trị z lớn hơn.

$\rightarrow$ Các lớp dữ liệu là tách được tuyến tính trong chiều thứ ba.



# Support Vector Machine

**Kernel Support Vector Machine**

•Tổng quát:

- Kernel SVM ⇔ Tìm hàm $\Phi(x)$ biến đổi dữ liệu X từ không gian đặc trưng ban đầu thành dữ liệu trong không gian mới.
- Trong ví dụ, $\Phi(\cdot)$ bổ sung một chiều dữ liệu mới (hay một đặc trưng mới), là hàm số của các đặc trưng đã biết.

•So sánh tương đối:

- Hàm Kernel $\Phi(\cdot)$ tương đồng với hàm ACTIVATION trong nhóm ρnương pháp dạng NEURAL NETWORKS.
- Cả hai hàm đều có mục đích chuyển đổi giữa bài toán phân loại phi tuyến và bài toán phân loại tuyến tính.

❖ Điểm khác biệt

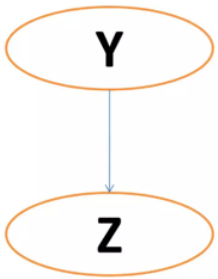
- Nhiệm vụ của activation function là phá vỡ tính tuyến tính của mô hình.
- Trong khi đó, hàm $\Phi(\cdot)$ biến dữ liệu không phân biệt tuyến tính thành phân biệt tuyến tính.

## Bayesian Belief Network

Mạng niềm tin Bayesian được định nghĩa bởi hai phần:

- Đồ thị có hướng không tuần hoàn (DAG - Directed Acyclic Graph) và một tập các bảng xác suất có điều kiện (CPT - Conditional Probability Table).
- Khi ta vẽ kết nổi của Y và Z như ở hình dưới thì Y ngay lập tức trở thành cha hay tiền nhân của Z còn Z là hậu duệ, con cháu của Y.

Mạng gồm các node thể hiện các biến và các link thể hiện sự phụ thuộc.



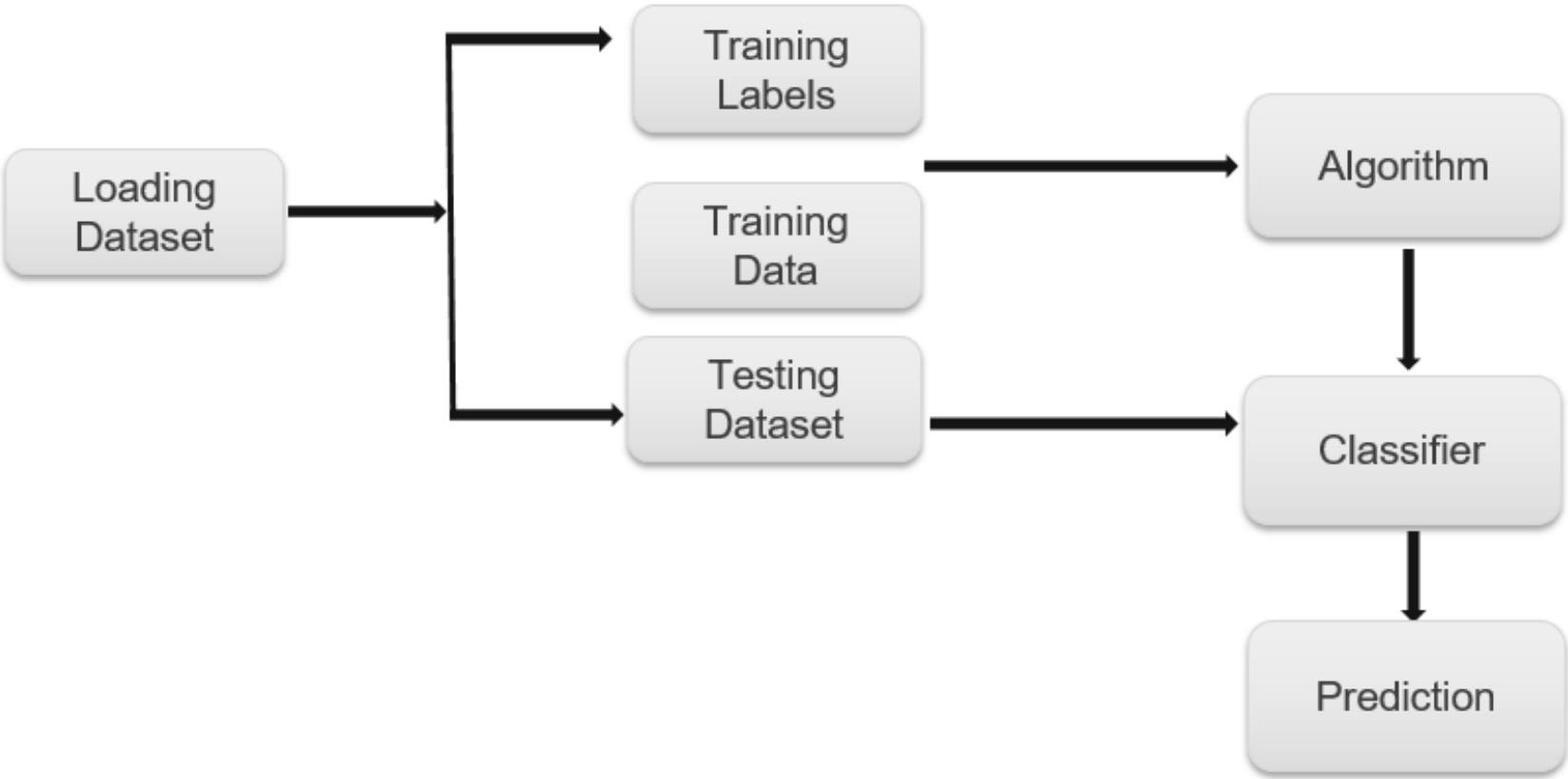
## Bayesian Belief Network

1.Chọn tập các biến liên quan $X_i$ mô tả domain.

2. Chọn thứ tự cho các biến.

3. Khi vẫn còn biến
   - Chọn một biến X và thêm một node cho nó.
   - Đặt node cha của X - parent(X) thành một số tập tối thiểu các node hiện có sao cho thuộc tính độc lập có điều kiện được thỏa mãn.
   - Định nghĩa CPT cho X

# Xây dựng mô hình



# Xây dựng mô hình

**Sử dụng thư viện:** numpy, pandas, matplotlib, seaborn, sklearn, nltk.

**Bộ dữ liệu sau khi tiền xử lý:** 5169 tin nhắn (4516 giá trị ham, 653 giá trị spam)

- Bộ dữ liệu huấn luyện: 80% của bộ dữ liệu sau khi tiền xử lý
- Bộ dữ liệu kiểm định: 20% của bộ dữ liệu sau khi tiền xử lý

**Phương pháp trích xuất đặc trưng:** TF-IDF (Số đặc trưng: 3000)

**Đầu vào mô hình:** Dữ liệu dưới dạng vectơ số

**Đầu ra mô hình:** Không spam (0) hoặc spam (1)

**Mô hình phân loại:** Multi Layer Perceptron, Multinomial Naïve Bayes, Support Vector Machine

# Xây dựng mô hình

## Chỉ số đánh giá mô hình

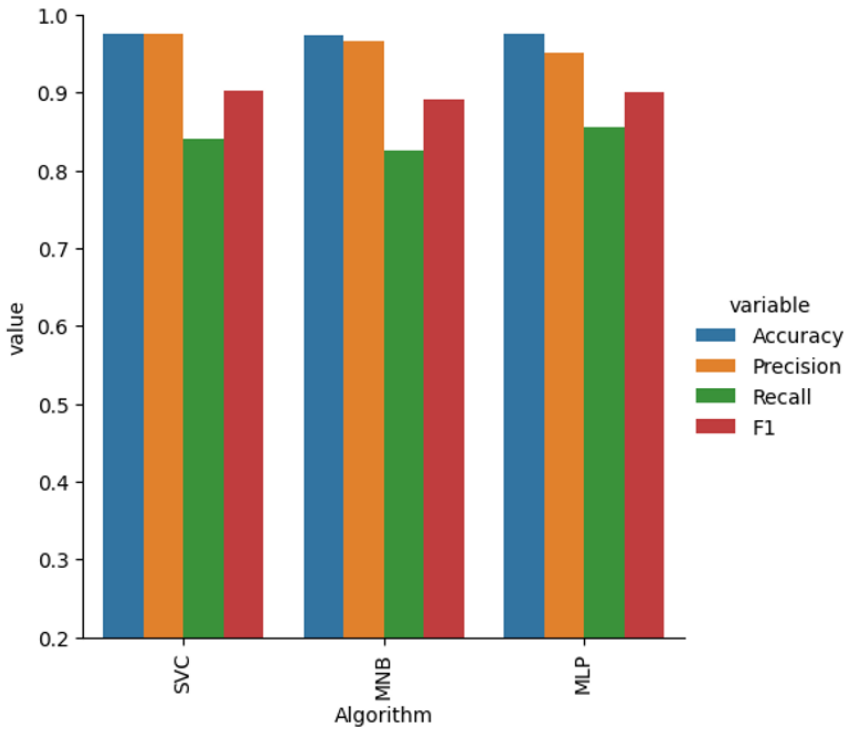
$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{Overall Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

$$\text{F1} = \frac{2 \text{ x } (\text{Precison x Recall})}{\text{Precison} + \text{Recall}}$$

| | Algorithm | Accuracy | Precision | Recall | F1 |
|---|---|---|---|---|---|
| 0 | SVC | 0.975822 | 0.974790 | 0.840580 | 0.902724 |
| 2 | MNB | 0.972921 | 0.966102 | 0.826087 | 0.890625 |
| 1 | MLP | 0.974855 | 0.951613 | 0.855072 | 0.900763 |



# Xây dựng mô hình

**Thời gian thực thi của các mô hình**

- Multinomial Naïve Bayes
- Support Vector Machine
- Multi Layer Perceptron

Thời gian chạy của hai mô hình Support Vector Machine và Multi Layer Perceptron chậm hơn từ 300 đến 800 lần so với Multinomial Naïve Bayes.

| | Algorithm | variable | value |
|---|---|---|---|
| 0 | MLP | Time | 33.107310 |
| 1 | SVC | Time | 12.959101 |
| 2 | MNB | Time | 0.036999 |

→ Mô hình Multinomial Naïve Bayes là phù hợp nhất.

# Kết luận và hướng phát triển

Mô hình của bọn em là sự kết hợp của thuật toán trích xuất đặc trưng TF-IDF và thuật toán phân loại Multinomial NB.

• Sau khi được huấn luyện trên bộ dữ liệu Spam SMS Collection, mô hình đã đưa ra kết quả khá tốt khi phát hiện và phân loại thành công tin nhắn rác.

• Tuy nhiên, vẫn còn một số trường hợp mà mô hình không phân loại chính xác được. Vì vậy trong tương lai nhóm em sẽ có thêm những cải tiến cho mô hình: Huấn luyện trên đa dạng các bộ dữ liệu về SMS, kết hợp với việc tối ưu các tham số và thuật toán giúp tăng tốc độ và hiệu suất của mô hình...

• Hạn chế trong bài nghiên cứu này là chúng em mới khai thác trên Spam SMS Tiếng Anh, chưa áp dụng với Spam SMS Tiếng Việt. Trong tương lai, chúng em sẽ phát triển cho các ngôn ngữ khác.