HUST

ĐẠI HỌC BÁCH KHOA HÀ NỘI

HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

ONE LOVE. ONE FUTURE.

ĐẠI HỌC
BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY
OF SCIENCE AND TECHNOLOGY

Consumer complaint classification

GVHD: TS. Thân Quang Khoát

Nhóm 21
Nguyễn Hà Phú Thịnh
Mai Thu Hiền
Trần Thị Như Quỳnh
Lê Đức Anh Duy
Nguyễn Duy Khánh
Nguyễn Thùy Dương

ONE LOVE. ONE FUTURE.

Mục lục

1. Giới thiệu đề tài

Các sàn thương mại điện tử như Shopee, Lazada, Tiki, và Amazon ngày càng phát triển và thu hút lượng lớn người tiêu dùng.
=> các sàn này cũng đối mặt với nhiều phàn nàn và khiếu nại từ khách hàng về chất lượng sản phẩm, dịch vụ giao hàng, và chăm sóc khách hàng.

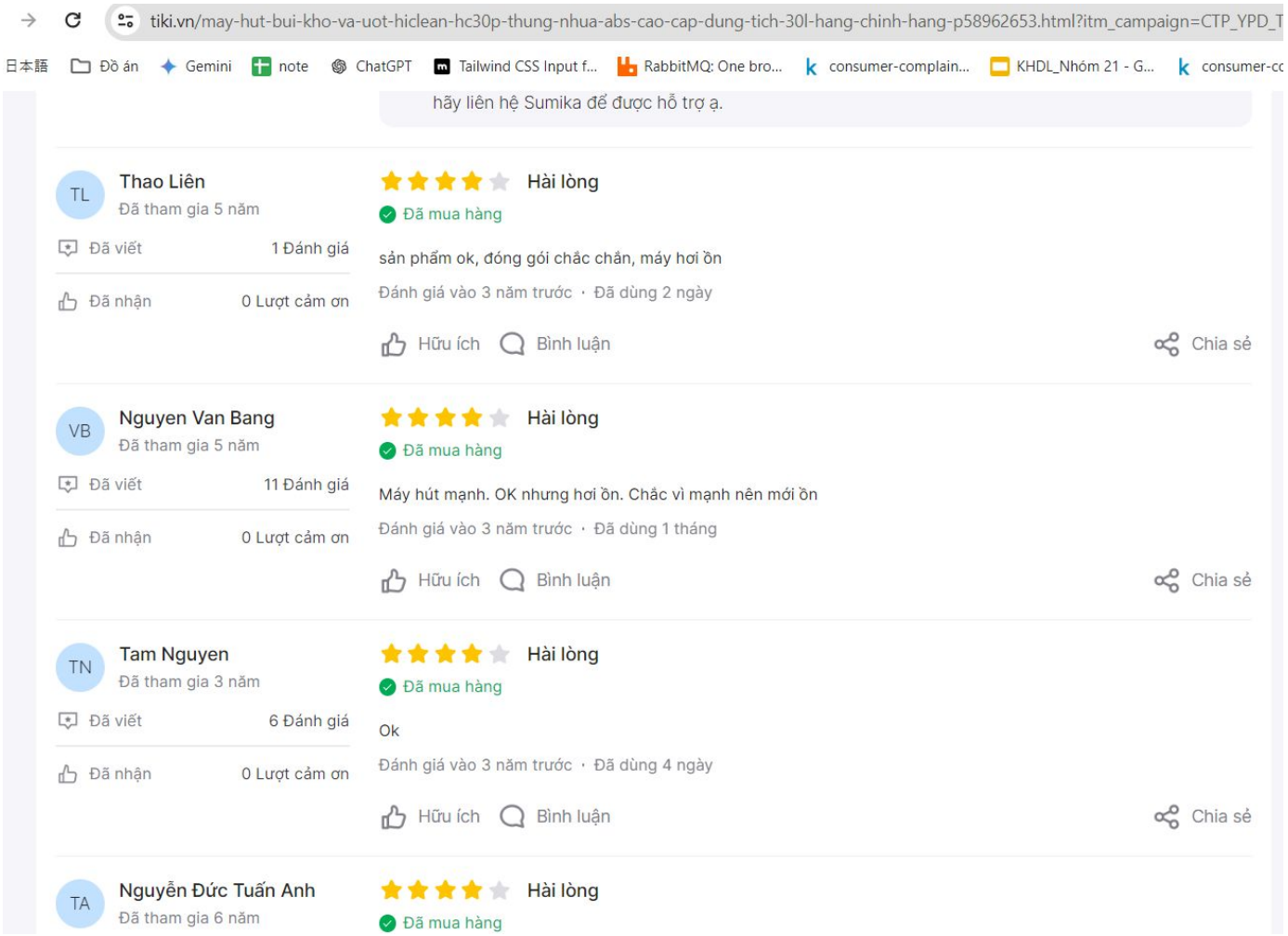
Việc phân loại và xử lý hiệu quả các phàn nàn này không chỉ giúp cải thiện trải nghiệm khách hàng mà còn giúp các doanh nghiệp hiểu rõ hơn về các vấn đề cần khắc phục.

2. Đặt vấn đề

Phân loại các phàn nàn vào các nhóm cụ thể như chất lượng sản phẩm, dịch vụ giao hàng, chăm sóc khách hàng, v.v.

3. Thu thập dữ liệu

Các đánh giá của người dùng về sản phẩm trên Tiki.vn



3. Thu thập dữ liệu

Cách thức thu thập: Lấy dữ liệu từ phản hồi (response) khi gửi yêu cầu (request) thông qua API của Tiki.

product-id list → product-review list

Lấy product_id list theo từng danh mục thông qua API :

https://tiki.vn/api/personalish/v1/blocks/listings',%20headers=header,%20params=params

Lấy product_review list list theo từng sản phẩm thông qua API :

https://tiki.vn/api/v2/reviews/{product_id}

3. Thu thập dữ liệu

Dữ liệu product-review bao gồm : id, title, content, rating

```
id,title,content,rating
,Cực kì hài lòng,"Lần đầu mua máy đọc sách.
Mất rất nhiều thời gian suy nghĩ chọn PPW5 hay OA3.Cuối cùng quyết định mua luôn OA3 bản 32Gb.
Món quà tặng tuyệt vời cho cô em gái mới thi lên lớp 10.

Sản phẩm có vỏ nhôm cao cấp,nhẹ,độ hoàn thiện cao...giao hàng nhanh.Sản phẩm đáng để mua vì đầu tư cho tri thức,cho giáo dục.",5
,Rất không hài lòng,"Mình đặt bản 32gb, màu gold; bên bán hàng giao màu graphite mà không hỏi ý kiến bên mua. Dịch vụ quá tệ.",1
,Cực kì hài lòng,Máy xài thích! Cầm rất thoải mái và chắc tay,5
,Cực kì hài lòng,Hàng mới nguyên seal. Shop tư vấn rất nhiệt tình.,5
,Hài lòng,,4
,Cực kì hài lòng,,5
,Cực kì hài lòng,,5
,Rất không hài lòng,Sp khác review inbox trên YouTube và hướng dẫn chính hãng,1
,Cực kì hài lòng,,5
,Hài lòng,"mua trên tiki đắt hơn mua từ nhà phân phối 200k. Giá tiki ghi là 7350k, giá hóa đơn 7139k",4
,Cực kì hài lòng,,5
,Cực kì hài lòng,Sản phẩm như mô tả!,5
,Cực kì hài lòng,,5
,Cực kì hài lòng,service rat tot. giao hang cung nhanh!!,5
,Cực kì hài lòng,,5
,Cực kì hài lòng,,5
,Cực kì hài lòng,,5
,Cực kì hài lòng,,5
,Cực kì hài lòng,,5
,Cực kì hài lòng,,5
```

4. Xử lý dữ liệu

- Chỉ lấy reviews phàn nàn (rating <= 3)
- Loại bỏ trường id, title, chỉ giữ lại trường content
- Gán nhãn content vào các nhãn Quality, Shipping, Packing, Service

content	Quality	Service	Shipping	Packing
chán lắm mau pin	0	-1.0	-1	-1.0
hàng màu trắng cửa hàng giao màu đen	-1	0.0	-1	-1.0
hiệu quả chuột	0	-1.0	-1	-1.0
cắm điện đèn sạc usb mô tả	0	-1.0	-1	-1.0
tệ giao nhầm hàng yc giao nhất	-1	0.0	-1	-1.0
ổ lỗi cắm phích lỏng lẻo không chặt 1 sao	0	-1.0	-1	-1.0
giao găng thực phẩm cục khuyếch đại wifi mercusy	-1	0.0	-1	-1.0
miếng dán hình giới thiệu	0	-1.0	-1	-1.0
sản phẩm kém chất lượng đầu ống đầu vòi không ...	0	-1.0	-1	-1.0
tam tiền	0	-1.0	-1	-1.0

1 : bình luận tích cực

0: bình luận tiêu cực

-1: tương ứng với ô rỗng

4. Xử lý dữ liệu

Kết quả sau khi xử lý: data ít, mất cân bằng giữa cái categories

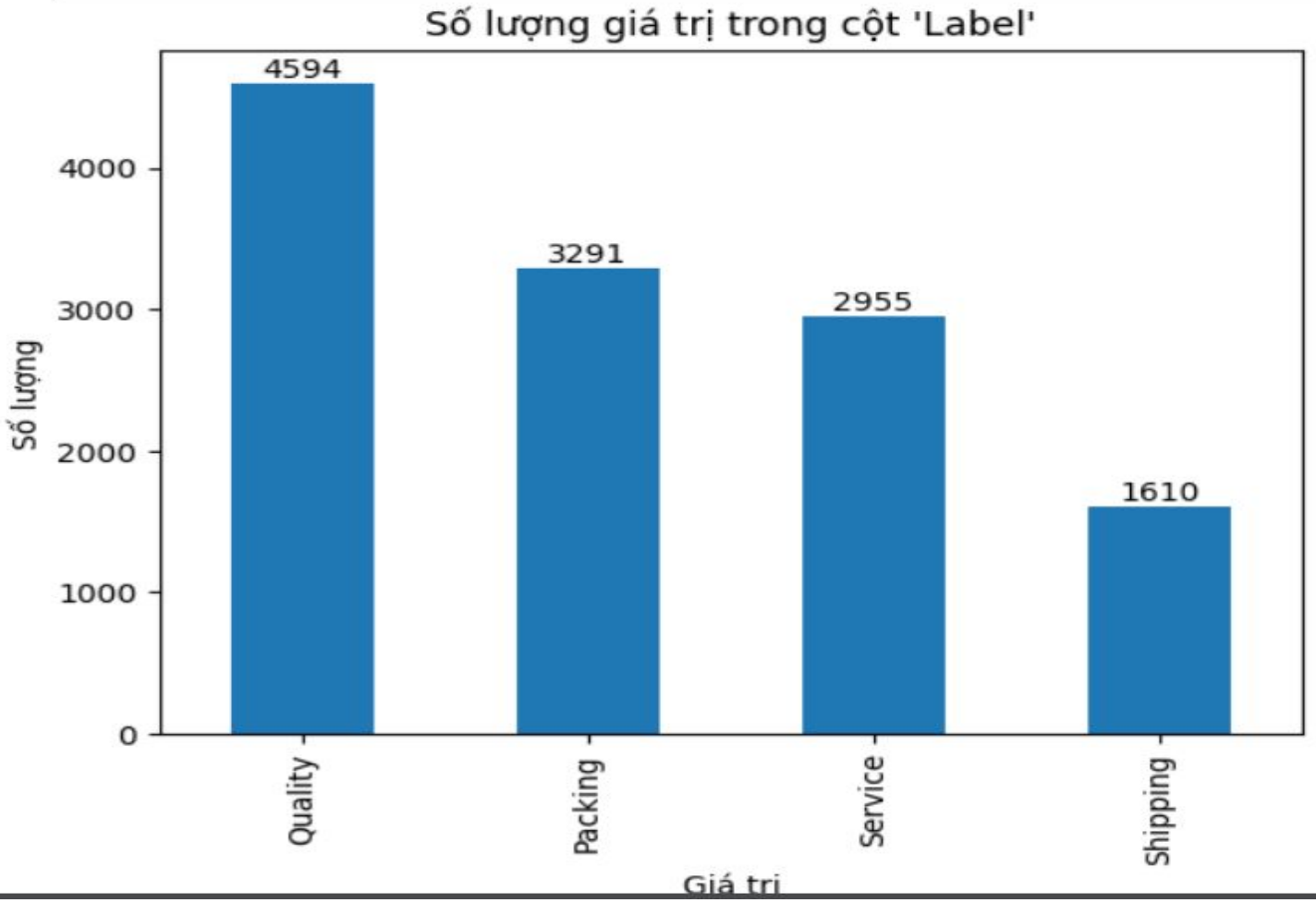
5. Tiền xử lý dữ liệu

+ Chuẩn hóa unicode
+ Chuẩn hóa dấu câu
+ Đưa về chữ viết thường
+ Chuẩn hóa câu:
    - Xóa khoảng trắng liên tiếp
    - Xóa ký tự không hợp lệ: @, #,....emoji
    - Xóa stopword : “rất”, “các”, “những”
    - Thay thế từ viết tắt: “sp”(sản phẩm), “k”(không),...

+ Mở rộng dữ liệu: Thay thế từ đồng nghĩa (Synonym replacement)

5. Tiền xử lý dữ liệu

Sau tiền xử lý : **12450** records



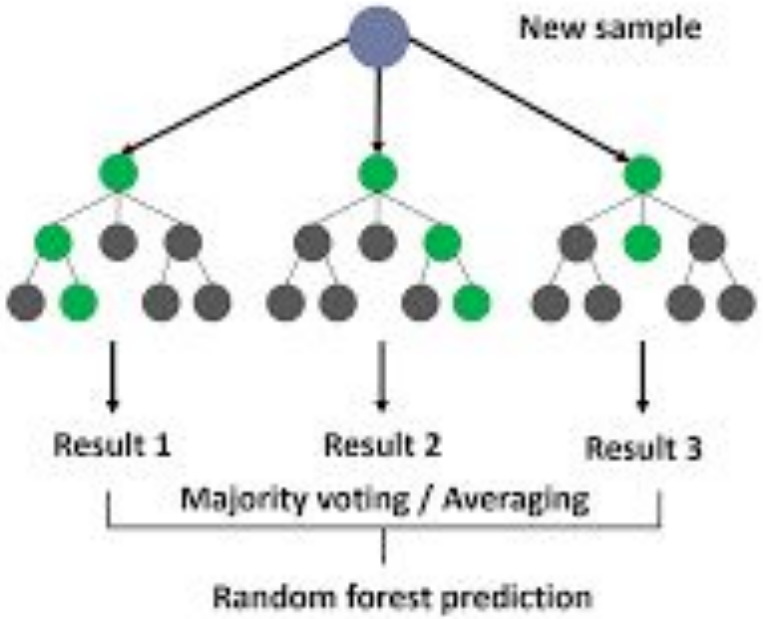
Mô hình học máy

1. Random Forest
2. Support Vector Machine
3. Logistic Regression

6. Mô hình

Random Forest

Trong trường hợp của Random Forest, mô hình này kết hợp nhiều cây quyết định (decision trees) để đạt được hiệu suất tốt hơn



6. Mô hình

Ưu điểm

- Giảm độ biến động và tránh overfitting: việc sử dụng nhiều cây quyết định và lựa chọn đặc trưng ngẫu nhiên, Random Forest thường ít bị overfitting so với một cây quyết định đơn lẻ
- Độ chính xác cao: Random Forest thường đạt được độ chính xác cao nhờ vào phương pháp tổng hợp
- Xử lý tốt các dữ liệu thiếu và không cân bằng nhờ vào phương pháp bootstrap
- Ít nhạy cảm với các outliers: Việc kết hợp nhiều cây quyết định giúp giảm ảnh hưởng của các outliers

6. Mô hình

Random Forest trong bài toán

```
# Huấn luyện Random Forest
rf_classifier = RandomForestClassifier(n_estimators=100, random_state=42)
rf_classifier.fit(X_train_tfidf, y_train)
```

- n_estimators=100: Số lượng cây quyết định (decision trees) trong rừng.
- random_state=42: Thiết lập seed cho bộ sinh số ngẫu nhiên để đảm bảo tính tái lập của kết quả
- X_train_tfidf: Dữ liệu huấn luyện đã được chuyển đổi thành ma trận TF-IDF
- y_train: Nhãn của dữ liệu huấn luyện

6. Mô hình

Random Forest trong bài toán

Đánh giá mô hình

- Accuracy: Cho biết tỷ lệ phần trăm các dự đoán đúng
- Precision: Cho biết tỷ lệ phần trăm các dự đoán đúng trong số các dự đoán được thực hiện cho mỗi lớp
- Recall: Cho biết tỷ lệ phần trăm các dự đoán đúng trong số các mẫu thực sự thuộc về mỗi lớp F1
- Score: Là trung bình điều hòa của precision và recall, cung cấp một cái nhìn cân bằng giữa hai chỉ số này

6. Mô hình

Random Forest trong bài toán

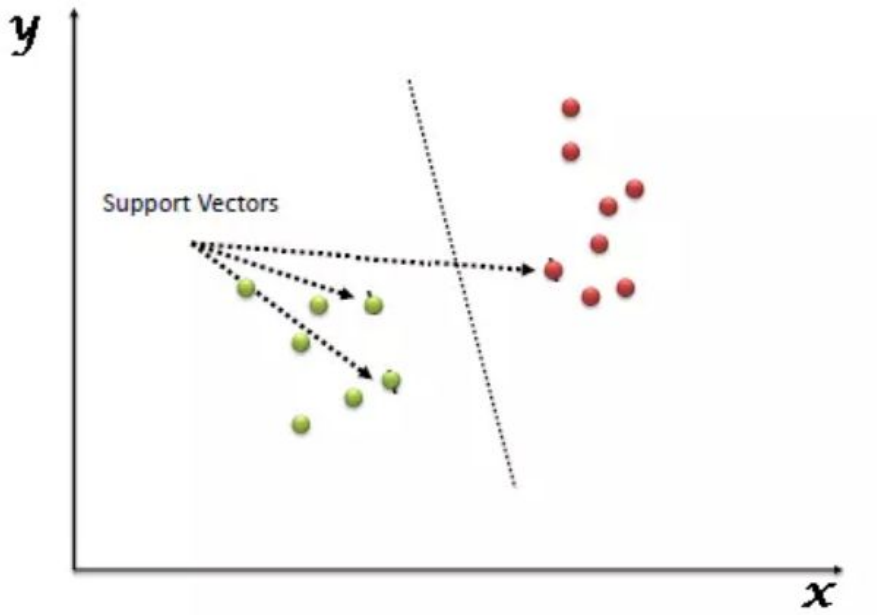
```
Kết quả Random Forest:
Độ chính xác: 0.9024096385542169
Độ chính xác (weighted): 0.9021775304094918
Độ thu hồi (weighted): 0.9024096385542169
Điểm F1 (weighted): 0.9021333667772969
```

Kết luận: Mô hình Random Forest đạt được kết quả đáng kể trong việc dự đoán với độ chính xác cao, đồng thời có khả năng giảm thiểu tình trạng overfitting so với các mô hình cây quyết định đơn lẻ nhờ vào phương pháp tổng hợp (ensemble)

6. Mô hình

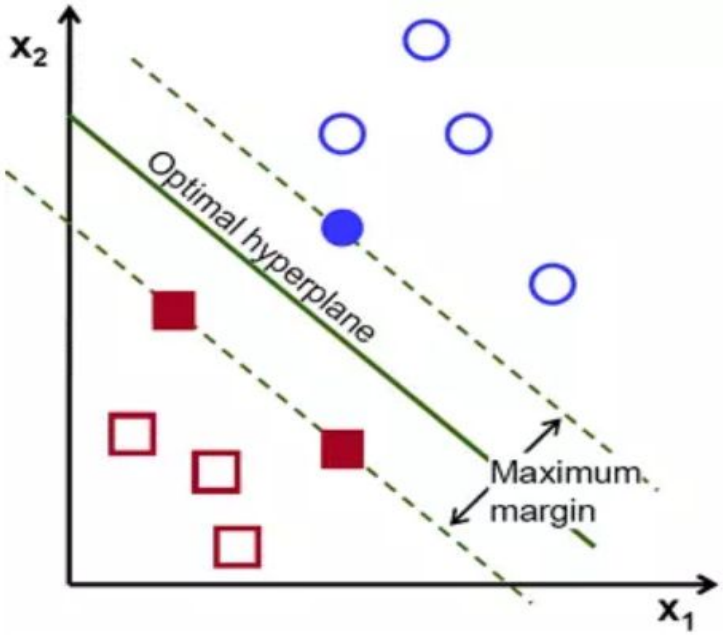
Support Vector Machine

SVM là một thuật toán giám sát được sử dụng chủ yếu cho việc phân loại. Trong thuật toán này, chúng ta vẽ đồ thị dữ liệu là các điểm trong n chiều với giá trị của mỗi tính năng sẽ là một phần liên kết. Sau đó chúng ta thực hiện tìm hyper-plane phân chia các lớp. Mục tiêu của SVM là tìm ra hyperplane tối ưu nhất để phân tách các lớp



Support Vector Machine

Margin là khoảng cách giữa hyperplane đến 2 điểm dữ liệu gần nhất tương ứng với các phân lớp. Phương pháp SVM luôn cố gắng cực đại hóa margin này, từ đó thu được một siêu phẳng tạo khoảng cách xa nhất so với 2 quả táo và lê. Nhờ vậy, SVM có thể giảm thiểu việc phân lớp sai (misclassification) đối với điểm dữ liệu mới đưa vào



6. Mô hình

SVM trong bài toán

```
# Huấn luyện SVM
svm_classifier = SVC(kernel='linear', random_state=42)
svm_classifier.fit(X_train_tfidf, y_train)
```

- kernel='linear': sử dụng kernel tuyến tính (linear kernel). SVM sẽ cố gắng tìm siêu phẳng phân chia các lớp trong không gian đặc trưng
- random_state=42: Giữ cho kết quả huấn luyện có thể tái lặp lại bằng cách cố định hạt giống ngẫu nhiên
- svm_classifier.fit(X_train_tfidf, y_train): Huấn luyện mô hình trên dữ liệu huấn luyện đã được chuyển đổi thành ma trận TF-IDF
- X_train_tfidf: Ma trận TF-IDF của dữ liệu huấn luyện
- y_train: Nhãn của dữ liệu huấn luyện

6. Mô hình

SVM trong bài toán

Dự đoán trên tập kiểm tra:

- svm_y_pred = svm_classifier.predict(X_test_tfidf): Dự đoán nhãn của dữ liệu kiểm tra sử dụng mô hình SVM đã huấn luyện
- X_test_tfidf: Ma trận TF-IDF của dữ liệu kiểm tra

Tính toán các giá trị chỉ số đánh giá:

- Accuracy: Tính tỷ lệ phần trăm các dự đoán đúng trên tổng số dự đoán
- Precision: Tính tỷ lệ phần trăm các dự đoán đúng trong số các dự đoán được thực hiện
- Recall: Tính tỷ lệ phần trăm các dự đoán đúng trong số các mẫu thực sự thuộc về mỗi lớp
- F1 Score: Tính trung bình điều hòa giữa precision và recall

6. Mô hình

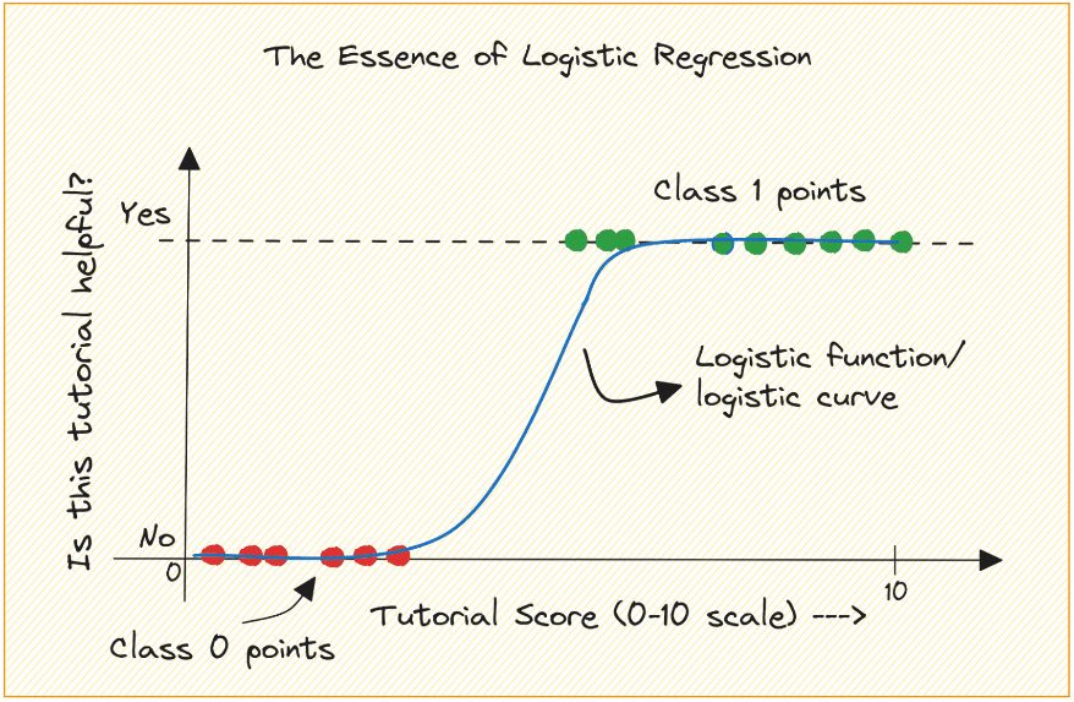
SVM trong bài toán

```
Kết quả SVM:
Độ chính xác: 0.901884534848938
Độ chính xác (weighted): 0.9018189076707184
Độ thu hồi (weighted): 0.901884534848938
Điểm F1 (weighted): 0.9005963660948356
```

Kết luận: SVM là một phương pháp hiệu quả cho bài toán phân lớp dữ liệu. Nó là một công cụ đắc lực cho bài toán phân tích văn bản.

Logistic Regression

Logistic Regression là một trong những mô hình phân loại cơ bản trong machine learning, được sử dụng rộng rãi trong các bài toán phân loại hai lớp



Logistic Regression

Logistic Regression sử dụng hàm logistic (hay sigmoid function) để biểu diễn xác suất thuộc vào một lớp cụ thể, được tính dựa trên tổ hợp tuyến tính của các biến đầu vào

$$P(y = 1|x) = \frac{1}{1+e^{-z}}$$

6. Mô hình

Ưu điểm

- Ít tài nguyên tính toán: Logistic Regression cần ít tài nguyên tính toán so với các mô hình phức tạp hơn như Neural Networks hay Random Forests, nên nó phù hợp cho các tập dữ liệu lớn

- Tính linh hoạt trong việc điều chỉnh tham số: Logistic Regression cho phép điều chỉnh các tham số như hệ số điều chỉnh (regularization) để kiểm soát overfitting

- Khả năng hiển thị tầm quan trọng của đặc trưng: Bằng cách xem xét hệ số của mỗi biến đầu vào, ta có thể đánh giá được mức độ ảnh hưởng của từng biến đến kết quả phân loại

- Cho kết quả xác suất: Logistic Regression không chỉ dự đoán lớp của một mẫu, mà còn cung cấp một ước lượng xác suất

6. Mô hình

Logistic Regression trong bài toán

```
# Huấn luyện Logistic Regression
lr_classifier = LogisticRegression(random_state=42)
lr_classifier.fit(X_train_tfidf, y_train)
```

- lr_classifier.fit(X_train_tfidf, y_train): Huấn luyện mô hình trên dữ liệu huấn luyện đã được chuyển đổi thành ma trận TF-IDF.
- X_train_tfidf: Ma trận TF-IDF của dữ liệu huấn luyện.
- y_train: Nhãn của dữ liệu huấn luyện.

6. Mô hình

Logistic Regression trong bài toán

Dự đoán trên tập kiểm tra:

- lr_y_pred = lr_classifier.predict(X_test_tfidf): Dự đoán nhãn của dữ liệu kiểm tra sử dụng mô hình Logistic Regression đã huấn luyện.
- X_test_tfidf: Ma trận TF-IDF của dữ liệu kiểm tra.

Tính toán các giá trị chỉ số đánh giá:

- Accuracy: Tính tỷ lệ phần trăm các dự đoán đúng trên tổng số dự đoán.
- Precision: Tính tỷ lệ phần trăm các dự đoán đúng trong số các dự đoán được thực hiện.
- Recall: Tính tỷ lệ phần trăm các dự đoán đúng trong số các mẫu thực sự thuộc về mỗi lớp.
- F1 Score: Tính trung bình điều hòa giữa precision và recall.

6. Mô hình

Logistic Regression trong bài toán

```
Kết quả Logistic Regression:
Độ chính xác: 0.8755608734669459
Độ chính xác (weighted): 0.8769734638837902
Độ thu hồi (weighted): 0.8755608734669459
Điểm F1 (weighted): 0.8725429330133847
```

Kết luận: Logistic Regression là một mô hình phân loại đơn giản, hiệu quả nhưng cũng có một số hạn chế (không thể mô hình hóa mối quan hệ phức tạp giữa các biến đầu vào và đầu ra, giả định tuyến tính)

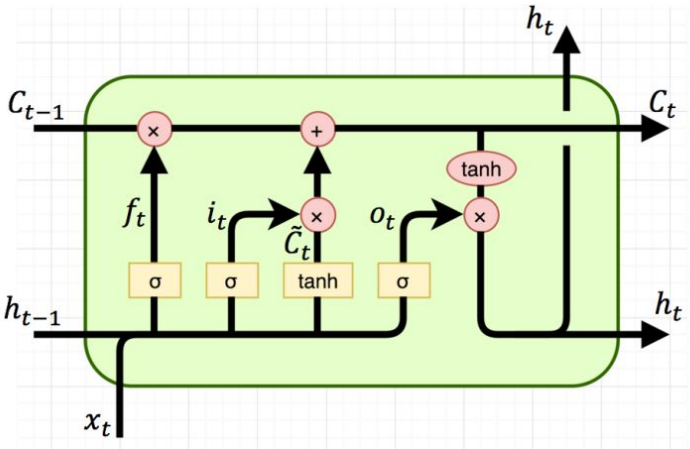
Mô hình học sâu

1. Long Short Term Memory (LSTM)
2. Gated Recurrent Unit (GRU)
3. Convolutional Neural Network (CNN)

Long Short Term Memory (LSTM)

Mạng nơ-ron hồi quy dài hạn (LSTM) là một kiến trúc mạng nơ-ron nhân tạo (ANN) được thiết kế để xử lý các chuỗi dữ liệu phụ thuộc thời gian

- **Khả năng học tập các chuỗi dài:** LSTM có thể học tập các chuỗi dài hơn nhiều so với RNN truyền thống do khả năng lưu trữ thông tin lâu dài
- **Khả năng xử lý các chuỗi phức tạp:** LSTM có thể xử lý các chuỗi phức tạp hơn so với RNN truyền thống do khả năng học tập các mối quan hệ phụ thuộc thời gian phức tạp.
- **Khả năng chống nhiễu:** LSTM có khả năng chống nhiễu tốt hơn so với RNN truyền thống do khả năng quên đi thông tin không quan trọng.



6. Mô hình

Long Short Term Memory (LSTM)

```
model = tf.keras.Sequential([
    # This is how you need to set the Embedding layer when using pre-trained embeddings
    tf.keras.layers.Embedding(vocab_size+1, embedding_dim, input_length=maxlen, trainable=False),
    tf.keras.layers.Dropout(0.3),
    tf.keras.layers.Conv1D(64, 5, activation='relu'),
    tf.keras.layers.MaxPooling1D(pool_size=4),
    tf.keras.layers.LSTM(64),
    tf.keras.layers.Dense(4, activation='softmax')

])

model.compile(loss="sparse_categorical_crossentropy",
              optimizer='adam',
              metrics=['accuracy'])
```

- Lớp nhúng (Embedding Layer): Lớp này chuyển đổi các từ trong văn bản thành các vectơ biểu diễn
- Lớp tích chập 1D (1D Convolutional Layer): Lớp này sử dụng các bộ lọc để trích xuất các đặc trưng cục bộ từ chuỗi dữ liệu đầu vào.
- Lớp pooling 1D (1D Pooling Layer): Lớp này giảm kích thước của chuỗi dữ liệu đầu vào bằng cách thực hiện phép toán lấy giá trị tối đa (max pooling).
- Lớp kết nối đầy đủ (Fully Connected Layer): Lớp này kết nối tất cả các nơ-ron trong lớp LSTM với tất cả các nơ-ron trong lớp đầu ra.

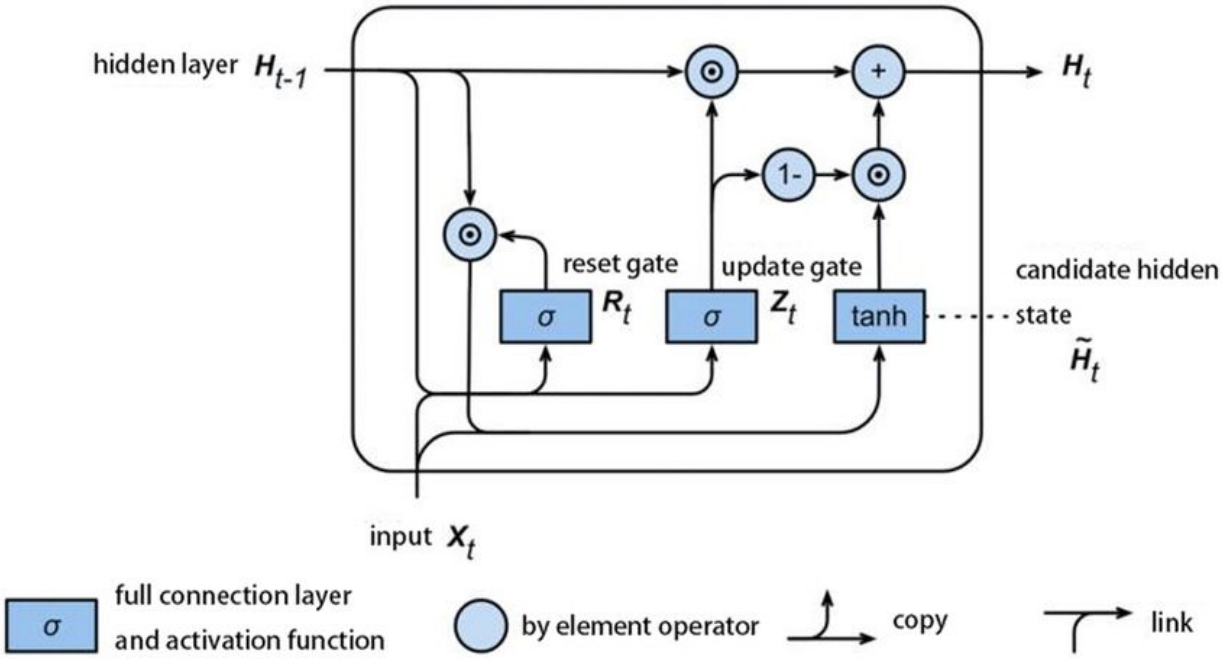
6. Mô hình

Long Short Term Memory (LSTM)

- Cấu trúc tương đối đơn giản, hiệu quả
- Hạn chế : việc lựa chọn tham số cho các thành phần có thể ảnh hưởng đến hiệu suất của mô hình

Gated Recurrent Unit (GRU)

Mạng GRU (Gated Recurrent Unit) là một biến thể của mạng LSTM (Long Short-Term Memory) được phát triển để khắc phục một số hạn chế của LSTM, đặc biệt là vấn đề biến mất gradient trong các mạng RNN (Recurrent Neural Network) sâu



6. Mô hình

Gated Recurrent Unit (GRU)

```
    model_gru = tf.keras.Sequential([
    tf.keras.layers.Embedding(vocab_size+1, embedding_dim, input_length=maxlen),
    tf.keras.layers.Dropout(0.3),
    tf.keras.layers.Bidirectional(tf.keras.layers.GRU(32)),
    tf.keras.layers.Dense(6, activation='relu'),
    tf.keras.layers.Dense(4, activation='softmax')
])

# Set the training parameters

    model_gru.compile(loss='sparse_categorical_crossentropy',
                      optimizer='adam',
                      metrics=['accuracy'])
```

- Lớp nhúng (Embedding Layer): Lớp này chuyển đổi các từ trong văn bản thành các vectơ biểu diễn
- Lớp GRU (Gated Recurrent Unit Layer): Lớp này là lớp chính của mô hình GRU
- Lớp kết nối đầy đủ (Fully Connected Layer): Lớp này kết nối tất cả các nơ-ron trong lớp GRU với tất cả các nơ-ron trong lớp đầu ra.

6. Mô hình

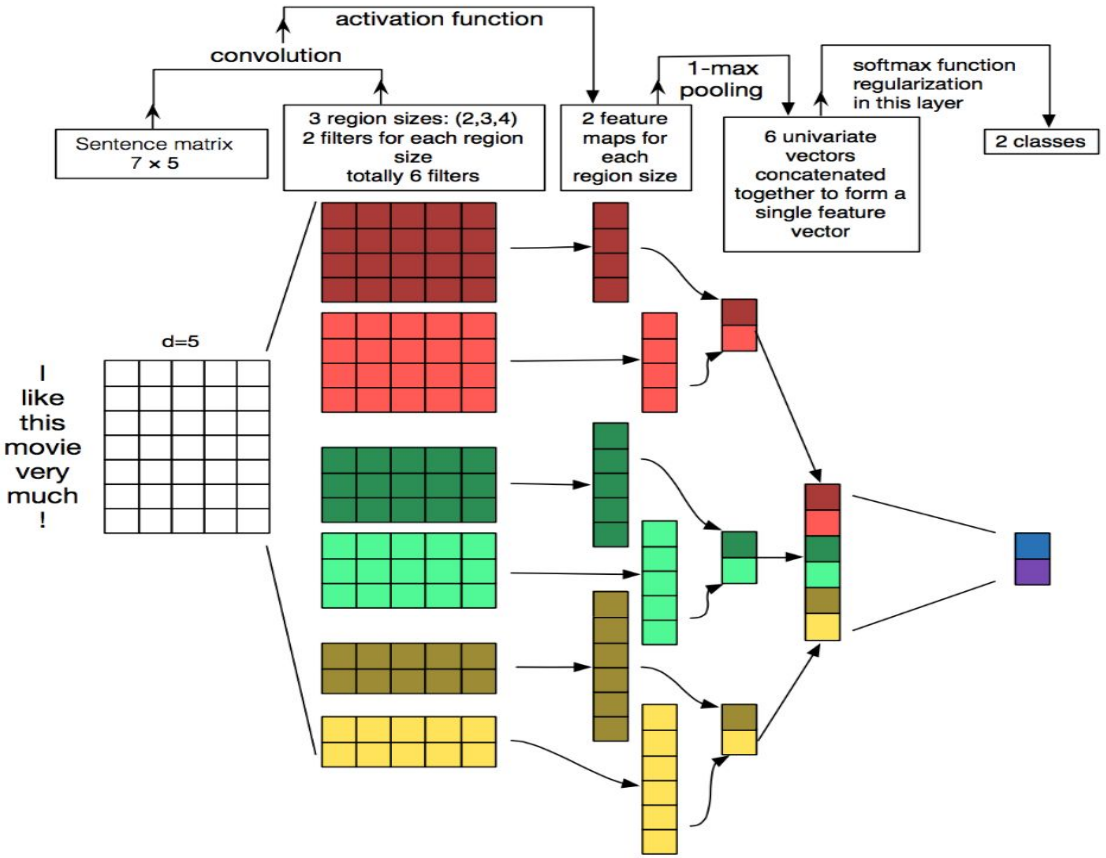
Gated Recurrent Unit (GRU)

- Cấu trúc đơn giản
- Mô hình học được các phụ thuộc thời gian dài trong văn bản

6. Mô hình

Convolutional Neural Network (CNN)

Mạng nơ-ron tích chập (CNN) là một loại mạng nơ-ron nhân tạo được sử dụng phổ biến trong lĩnh vực xử lý hình ảnh và thị giác máy tính



6. Mô hình

Convolutional Neural Network (CNN)

```
model = tf.keras.Sequential([
    # This is how you need to set the Embedding layer when using pre-trained embeddings
    tf.keras.layers.Embedding(vocab_size+1, embedding_dim, input_length=maxlen, trainable=False),
    tf.keras.layers.Dropout(0.3),
    tf.keras.layers.Conv1D(128, 5, activation='relu'),
    tf.keras.layers.GlobalMaxPooling1D(),
    tf.keras.layers.Dense(6, activation='relu'),
    tf.keras.layers.Dense(4, activation='softmax')

])

model.compile(loss="sparse_categorical_crossentropy",
              optimizer='adam',
              metrics=['accuracy'])
```

- Lớp nhúng (Embedding Layer): Lớp này chuyển đổi các từ trong văn bản thành các vectơ biểu diễn,
- Lớp tích chập 1D (1D Convolutional Layer): Lớp này sử dụng các bộ lọc để trích xuất các đặc trưng cục bộ từ chuỗi vectơ biểu diễn văn bản.
- Lớp pooling 1D (1D Pooling Layer): Lớp này giảm kích thước của chuỗi vectơ đầu ra bằng cách thực hiện phép toán lấy giá trị tối đa (max pooling) hoặc giá trị trung bình (average pooling) trong một cửa sổ nhất định.
- Lớp kết nối đầy đủ (Fully Connected Layer): Lớp này kết nối tất cả các nơ-ron trong lớp pooling cuối cùng với tất cả các nơ-ron trong lớp đầu ra.

7. Kết quả

Model	Accuracy	Precision	Recall	F1 Score
Random Forest	0.95872	0.95866	0.95872	0.95859
SVM	0.90188	0.90182	0.90188	0.9006
Logistic Regression	0.87556	0.87697	0.87556	0.87254

7. Kết quả

Mô hình	Accuracy Train	Accuracy Test
LSTM	0.7819	0.9094
GRU	0.6622	0.8974
CNN	0.7824	0.9177

HUST

THANK YOU !