ĐHQG TP. Hồ Chí Minh - Trường ĐH Công Nghệ Thông Tin

-oOo-

ĐỒ ÁN CUỐI KỲ

PHÂN TÍCH CẢM XÚC BẰNG CÁCH SỬ DỤNG DỮ LIỆU ĐÁNH GIÁ SẢN PHẨM SHOPEE

Môn học:	Tư duy tính toán – CS117.M21.KHCL
Giảng viên:	Ngô Đức Thành
Sinh viên thực hiện:	Hoàng Gia Huy – 19521607
	Đoàn Tấn Phát – 20520269
	Bùi Thị Bích Hậu -19521483

TP. Hồ Chí Minh, 11 tháng 06 năm 2022

MỤC LỤC

I. GIỚI THIỆU:

1.1. TỔNG QUAN:

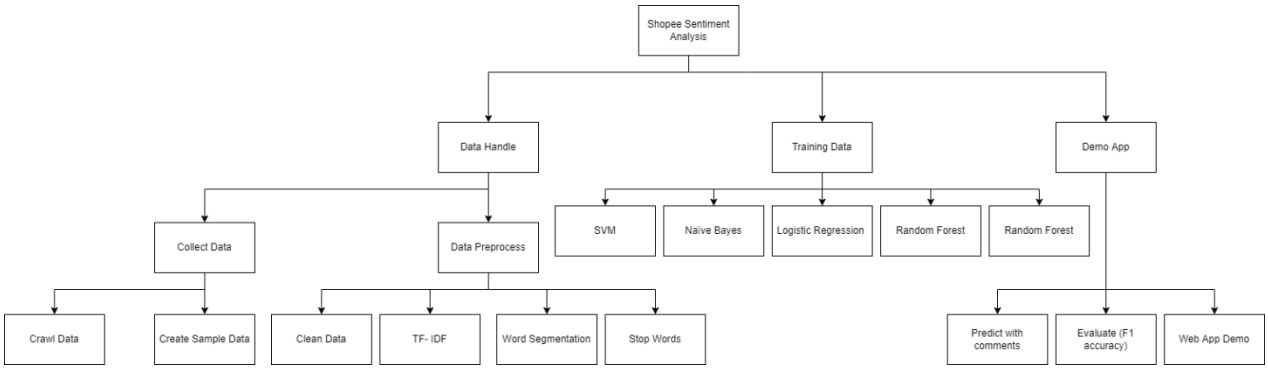
- Ngày nay, với sự phát triển vượt bậc của khoa học và công nghệ, đặc biệt là sự bùng nổ của Internet với các phương tiện truyền thông xã hội, thương mại điện tử,... cho phép mọi người không những chia sẻ thông tin trên Internet mà còn thể hiện thái độ, quan điểm của mình đối với các sản phẩm, dịch vụ và các vấn đề xã hội khác trong cuộc sống.
- Việc thu thập và xem xét các thông tin phản hồi đó của khách hàng là một cách để giúp cho các doanh nghiệp hiểu được điểm mạnh, điểm yếu trong sản phẩm, dịch vụ của mình; đồng thời nhanh chóng nắm bắt được tâm lý và nhu cầu khách hàng để mang đến cho họ sản phẩm và dịch vụ hoàn hảo nhất.

1.2. NHIỆM VỤ CỦA BÀI TOÁN:

- Với nhu cầu thị trường và sự phát triển của ngành công nghệ thông tin hiện nay, việc xây dựng một mô hình tự động đánh giá và phân loại các câu bình luận, phê bình của người tiêu dùng đóng vai trò rất quan trọng, nhằm giúp:

- Người dùng sử dụng nó có thể tìm kiếm, tham khảo trước khi đưa ra quyết định sử dụng một sản phẩm hay dịch vụ nào đó.
- Các nhà cung cấp dịch vụ cũng có thể sử dụng những nguồn thông tin này để đánh giá về sản phẩm của mình, từ đó có thể đưa ra những cải tiến phù hợp hơn với người dùng, mang lại lợi nhuận cao hơn, tránh các rủi ro đáng tiếc xảy ra.

1.3. SƠ ĐỒ HIERARCHY:



Cấu trúc giải quyết cho bài toán:

Để giải quyết bài toán nhóm, đã phân chia bài toán thành cấu trúc gồm 3 phần chính:

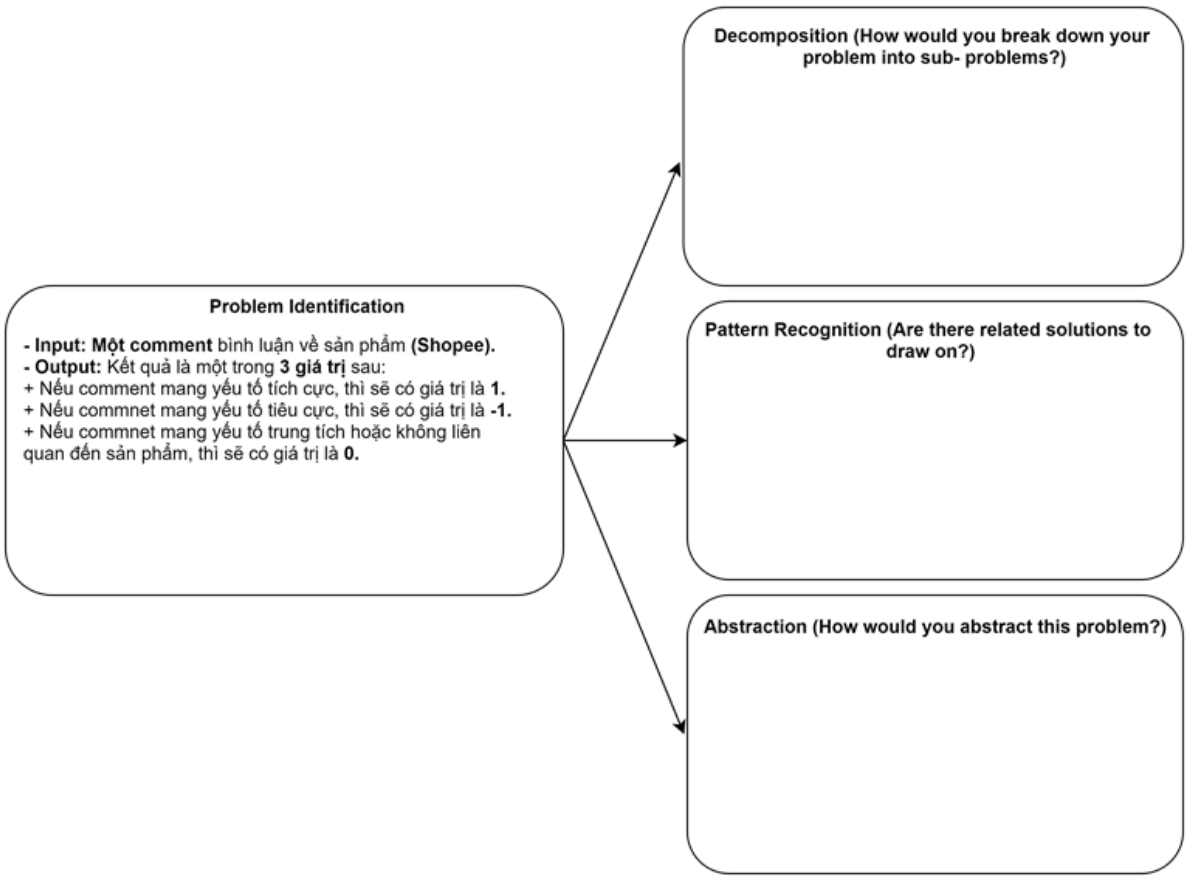
- Data
- Training
- Demo

Với 3 phần trên nhóm thấy rằng có thể áp dụng các pattern của việc giải quyết một bài toán máy học (cụ thể là bài toán phân loại). Sau khi phân chia bài toán thành 3 vấn đề nhỏ hơn, chúng em sẽ thực hiện giải quyết vấn đề bằng các bước tương tự khi giải quyết một vấn đề máy học (các công đoạn để giải quyết một bài toán máy học điển hình)

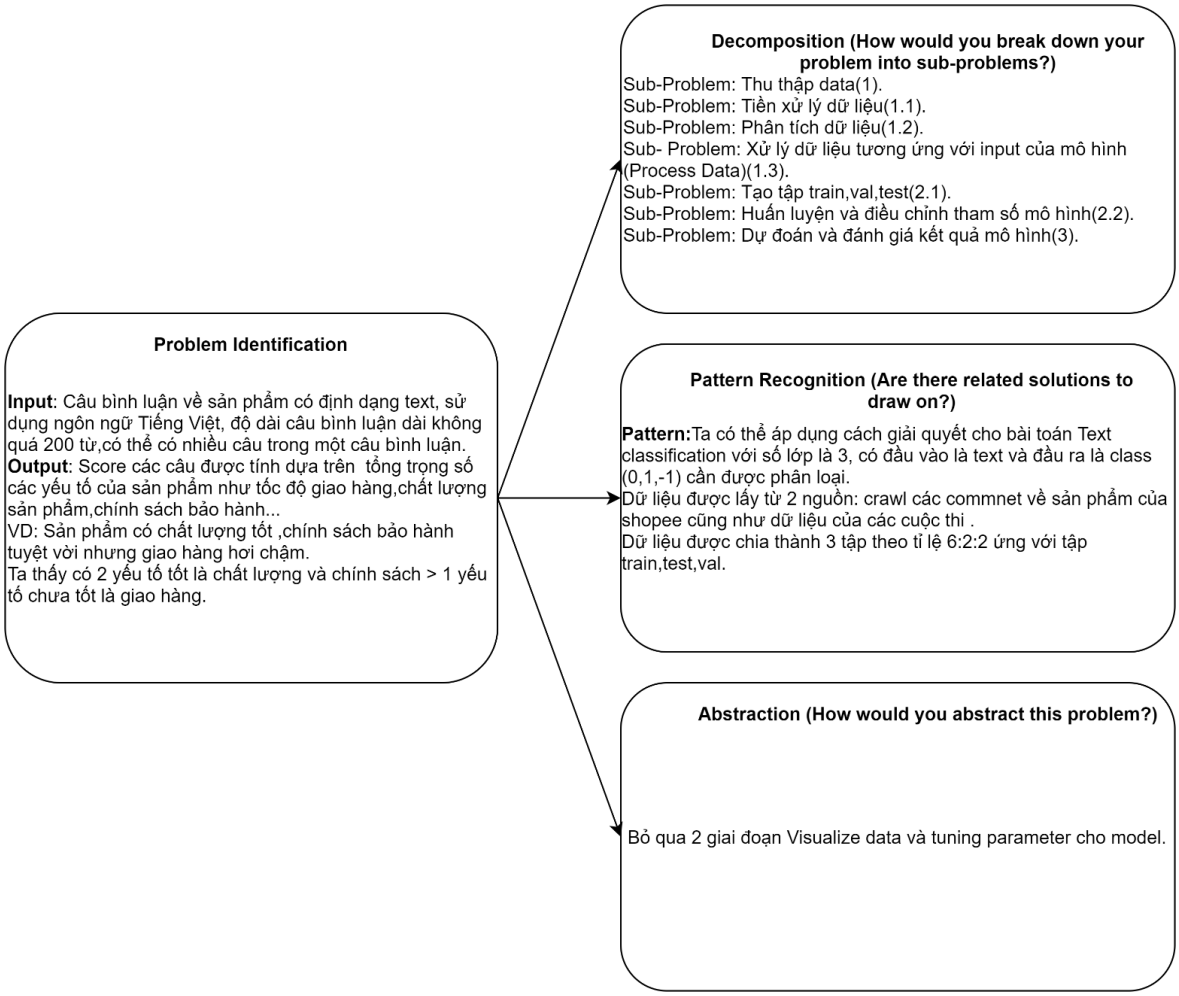
Ngoài ra ở mỗi bước thực hiện, sẽ có một số bước nhỏ kèm theo (rõ hơn ở phần Graphic Organizer). Các vấn đề ở node trên tuy chưa là một vấn đề quen đơn giản nhưng ta có thể sử dụng các pattern recognition về bài toán text classification để giải quyết (các phần ở các node lá là các công đoạn được xem là cơ bản trong các bài toán có sử dụng máy học).

1.4. GRAPHIC ORGANIZER:

Iteration 1:

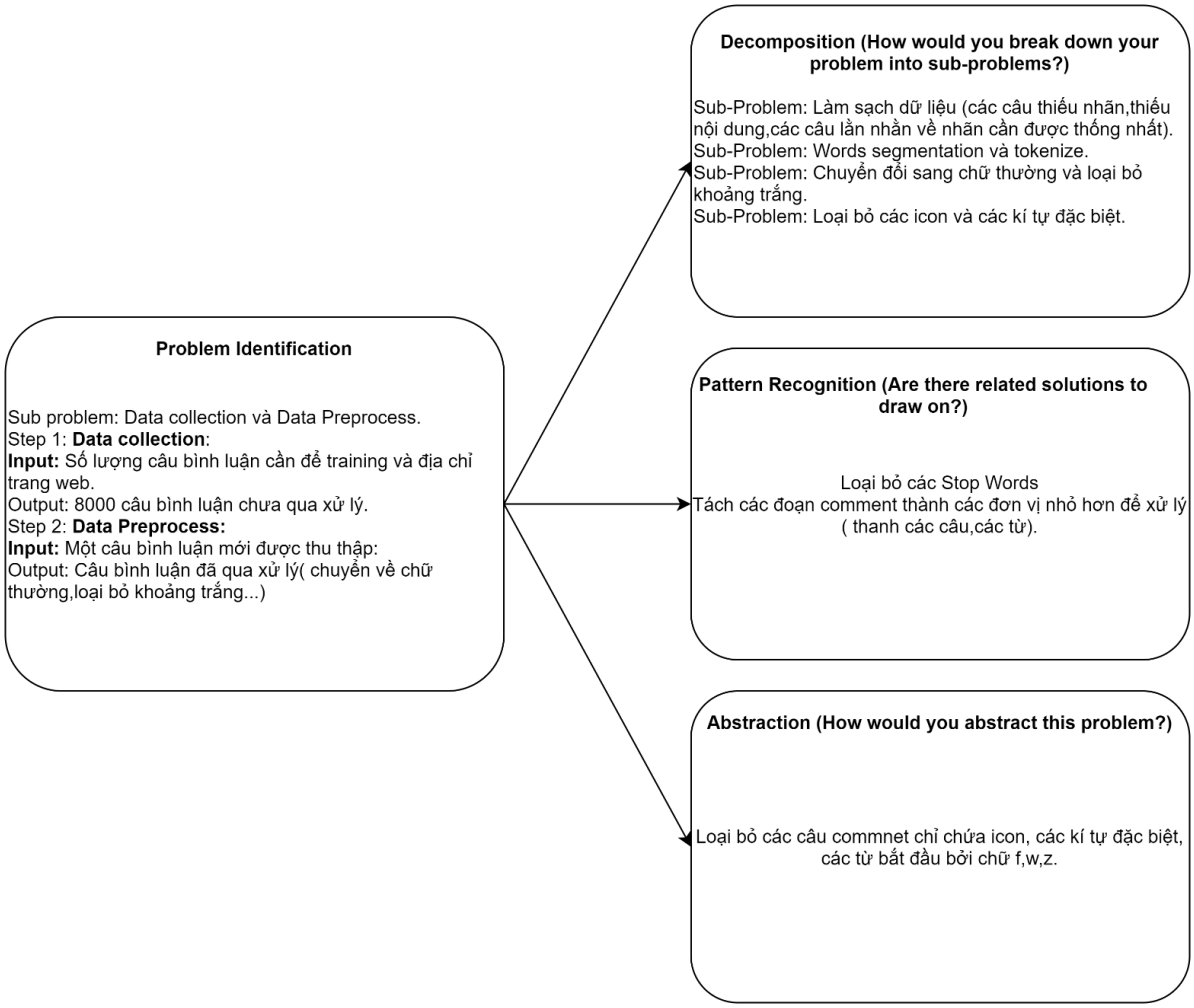


Iteration 2:

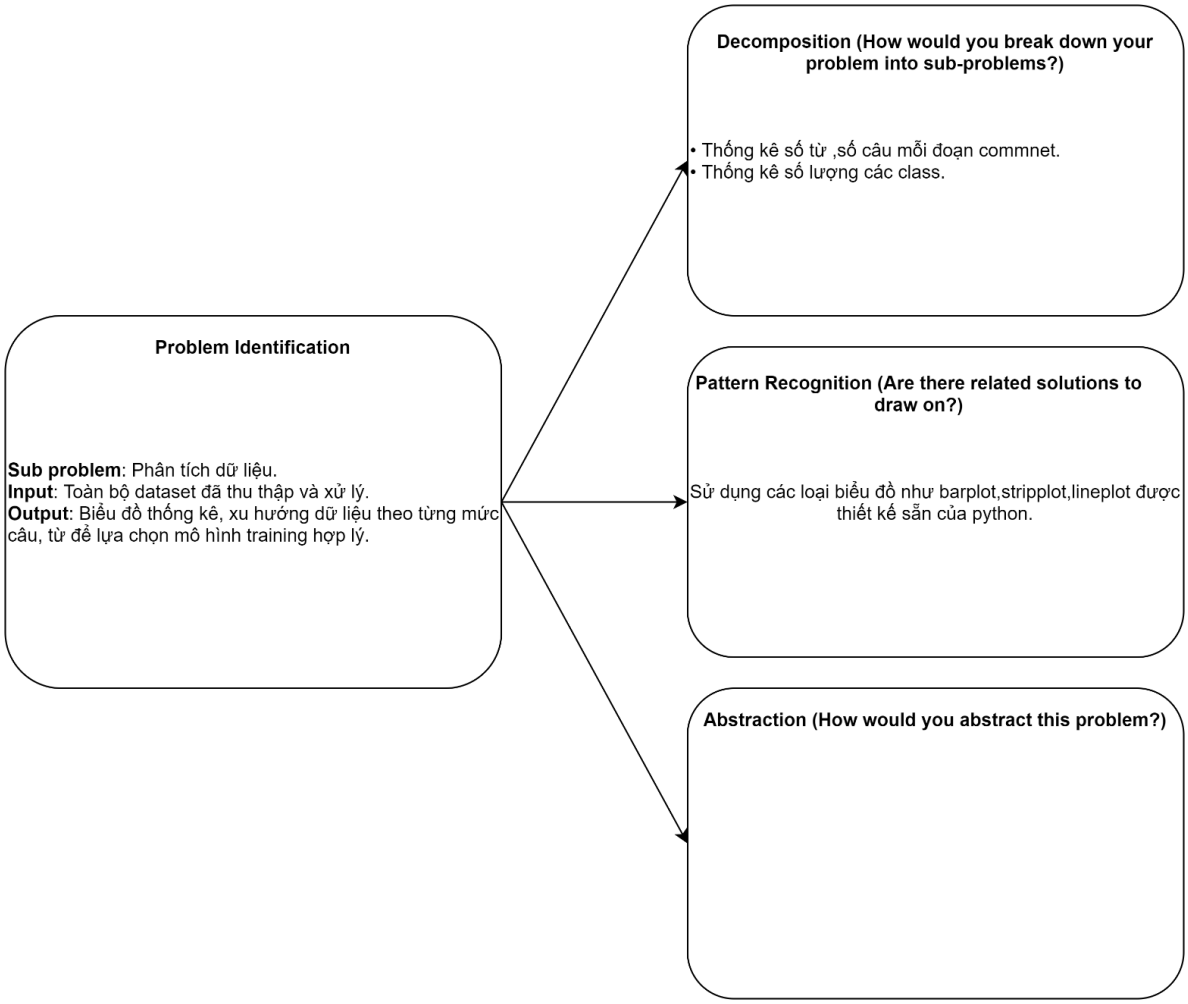

Problem Identification
Input: Câu bình luận về sản phẩm có định dạng text, sử dụng ngôn ngữ Tiếng Việt, độ dài câu bình luận dài không quá 200 từ,có thể có nhiều câu trong một câu bình luận.
Output: Score các câu được tính dựa trên tổng trọng số các yếu tố của sản phẩm như tốc độ giao hàng,chất lượng sản phẩm,chính sách bảo hành...
VD: Sản phẩm có chất lượng tốt ,chính sách bảo hành tuyệt vời nhưng giao hàng hơi chậm.
Ta thấy có 2 yếu tố tốt là chất lượng và chính sách > 1 yếu tố chưa tốt là giao hàng.
Decomposition (How would you break down your problem into sub-problems?)
Sub-Problem: Thu thập data(1).
Sub-Problem: Tiền xử lý dữ liệu(1.1).
Sub-Problem: Phân tích dữ liệu(1.2).
Sub- Problem: Xử lý dữ liệu tương ứng với input của mô hình (Process Data)(1.3).
Sub-Problem: Tạo tập train,val,test(2.1).
Sub-Problem: Huấn luyện và điều chỉnh tham số mô hình(2.2).
Sub-Problem: Dự đoán và đánh giá kết quả mô hình(3).
Pattern Recognition (Are there related solutions to draw on?)
Pattern:Ta có thể áp dụng cách giải quyết cho bài toán Text classification với số lớp là 3, có đầu vào là text và đầu ra là class (0,1,-1) cần được phân loại.
Dữ liệu được lấy từ 2 nguồn: crawl các commnet về sản phẩm của shopee cũng như dữ liệu của các cuộc thi .
Dữ liệu được chia thành 3 tập theo tỉ lệ 6:2:2 ứng với tập train,test,val.
Abstraction (How would you abstract this problem?)
Bỏ qua 2 giai đoạn Visualize data và tuning parameter cho model.

-

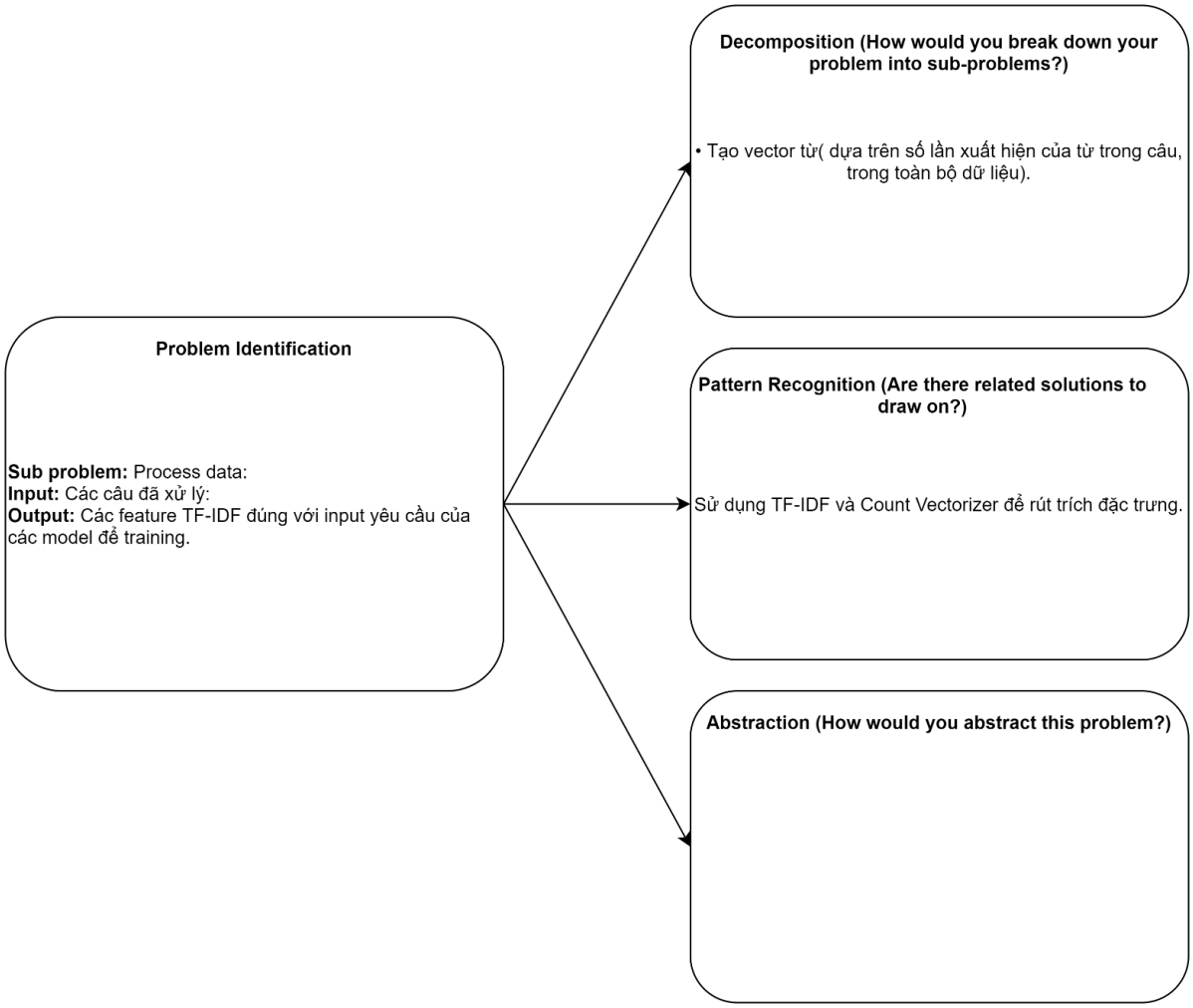
Iteration 3:


Problem Identification
Sub problem: Data collection và Data Preprocess.
Step 1: Data collection:
Input: Số lượng câu bình luận cần để training và địa chỉ trang web.
Output: 8000 câu bình luận chưa qua xử lý.
Step 2: Data Preprocess:
Input: Một câu bình luận mới được thu thập:
Output: Câu bình luận đã qua xử lý(chuyển về chữ thường,loại bỏ khoảng trắng...)
Decomposition (How would you break down your problem into sub-problems?)
Sub-Problem: Làm sạch dữ liệu (các câu thiếu nhãn,thiếu nội dung,các câu lằn nhằn về nhãn cần được thống nhất).
Sub-Problem: Words segmentation và tokenize.
Sub-Problem: Chuyển đổi sang chữ thường và loại bỏ khoảng trắng.
Sub-Problem: Loại bỏ các icon và các kí tự đặc biệt.
Pattern Recognition (Are there related solutions to draw on?)
Loại bỏ các Stop Words
Tách các đoạn comment thành các đơn vị nhỏ hơn để xử lý (thanh các câu,các từ).
Abstraction (How would you abstract this problem?)
Loại bỏ các câu commnet chỉ chứa icon, các kí tự đặc biệt, các từ bắt đầu bởi chữ f,w,z.

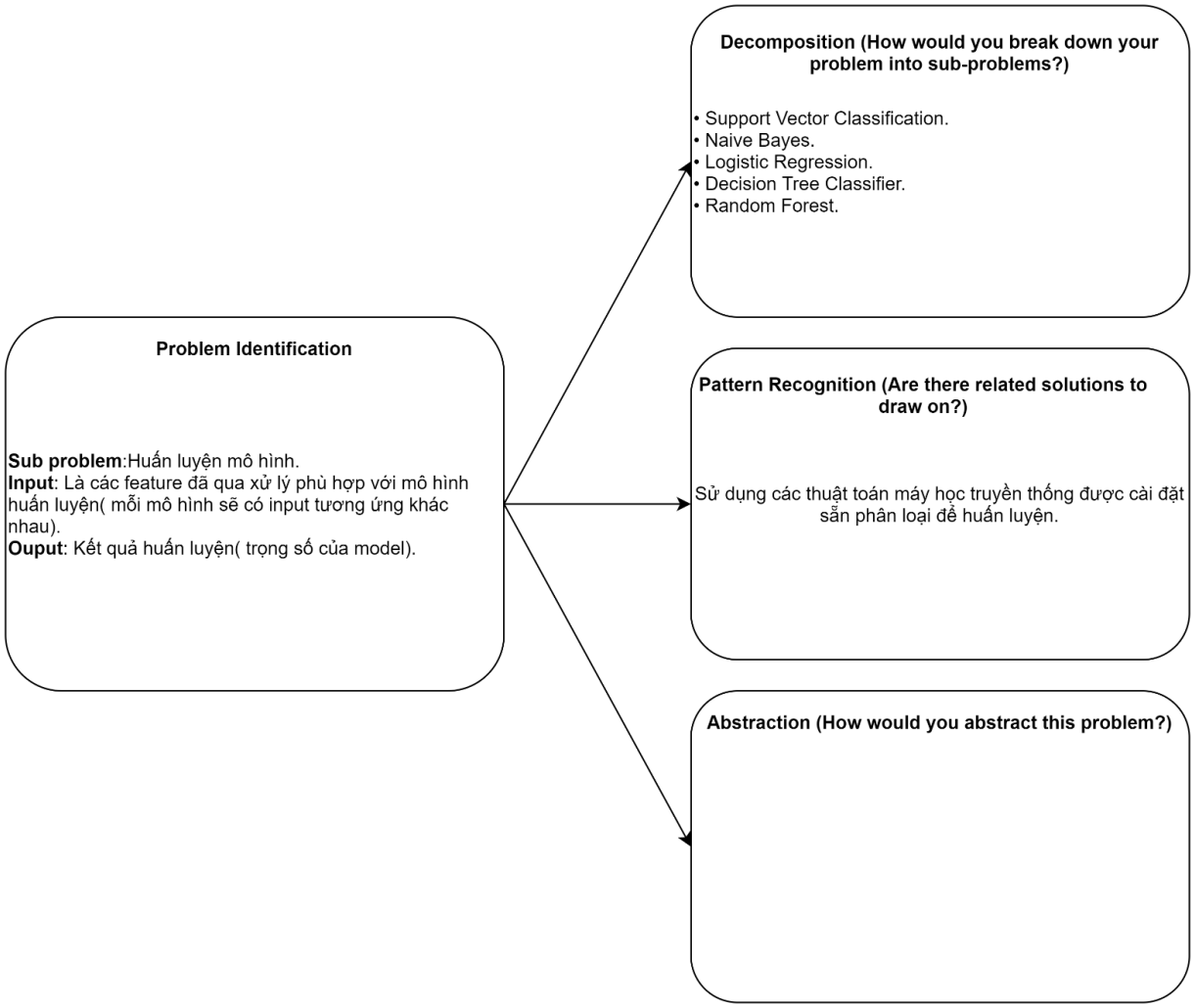
Iteration 4:


Problem Identification
Sub problem: Phân tích dữ liệu.
Input: Toàn bộ dataset đã thu thập và xử lý.
Output: Biểu đồ thống kê, xu hướng dữ liệu theo từng mức câu, từ để lựa chọn mô hình training hợp lý.
Decomposition (How would you break down your problem into sub-problems?)
• Thống kê số từ ,số câu mỗi đoạn commnet.
• Thống kê số lượng các class.
Pattern Recognition (Are there related solutions to draw on?)
Sử dụng các loại biểu đồ như barplot,stripplot,lineplot được thiết kế sẵn của python.
Abstraction (How would you abstract this problem?)

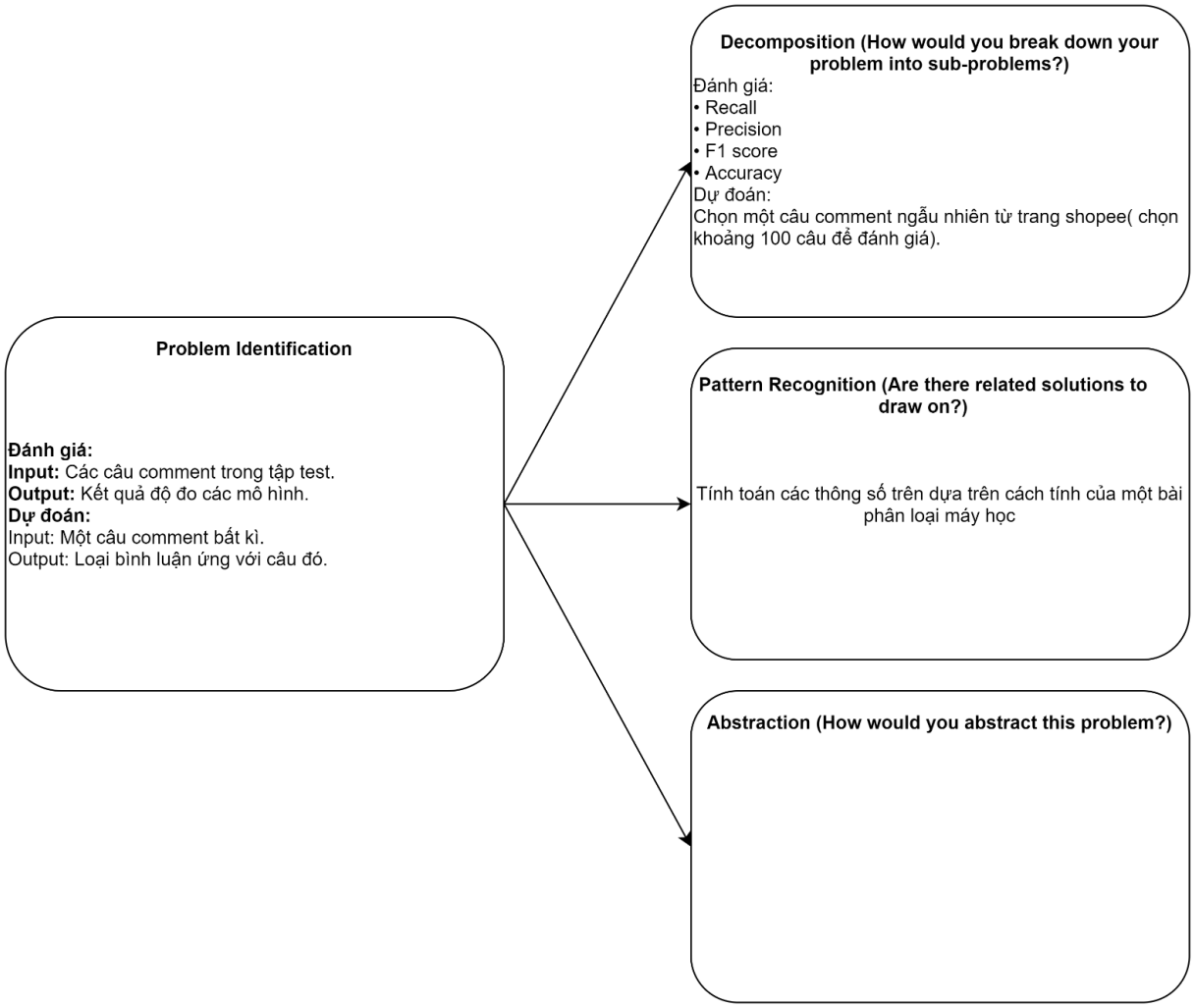
Iteration 5:


Problem Identification
Sub problem: Process data:
Input: Các câu đã xử lý:
Output: Các feature TF-IDF đúng với input yêu cầu của các model để training.
Decomposition (How would you break down your problem into sub-problems?)
• Tạo vector từ(dựa trên số lần xuất hiện của từ trong câu, trong toàn bộ dữ liệu).
Pattern Recognition (Are there related solutions to draw on?)
Sử dụng TF-IDF và Count Vectorizer để rút trích đặc trưng.
Abstraction (How would you abstract this problem?)

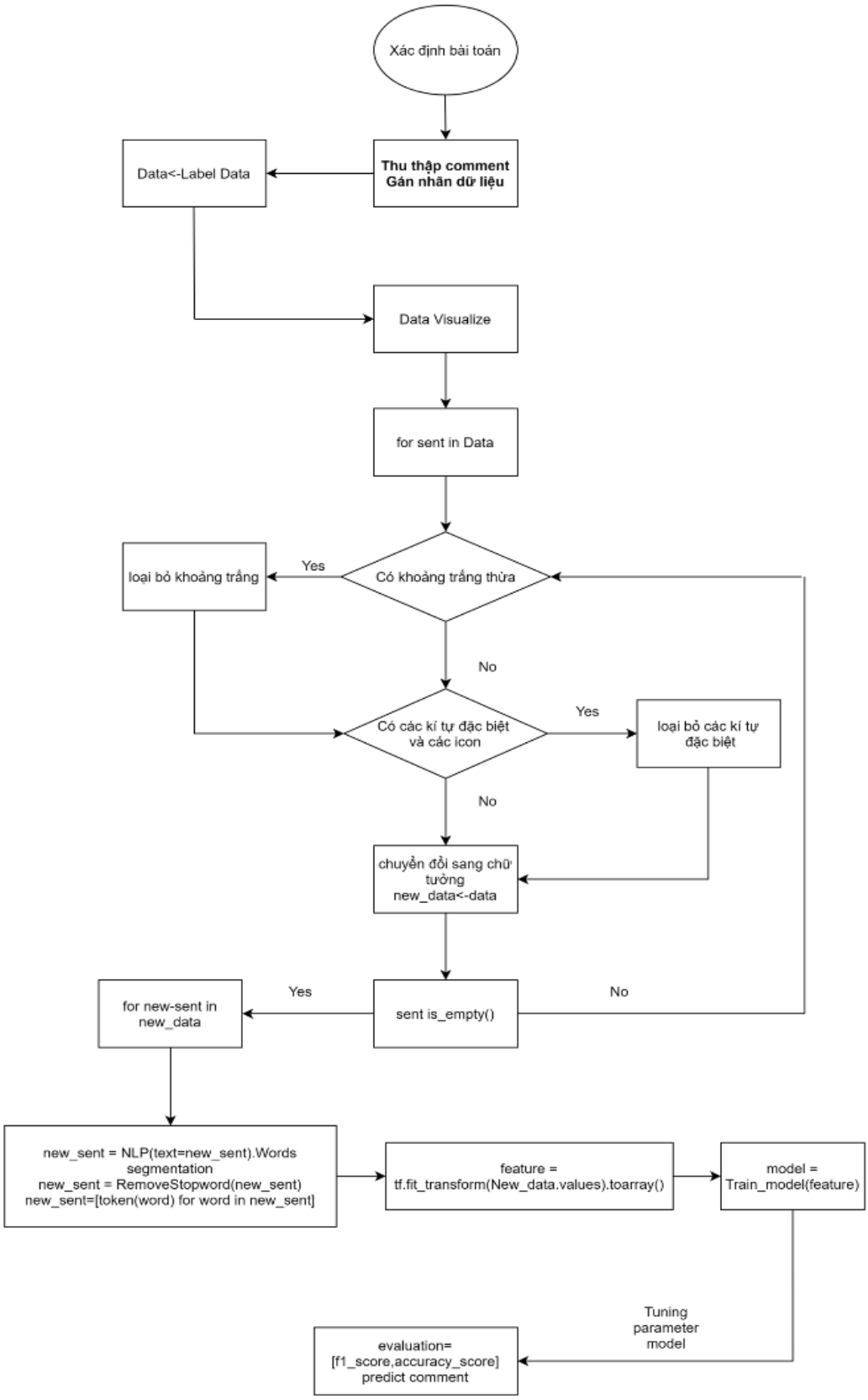
Iteration 6:


Problem Identification
Sub problem:Huấn luyện mô hình.
Input: Là các feature đã qua xử lý phù hợp với mô hình huấn luyện(mỗi mô hình sẽ có input tương ứng khác nhau).
Ouput: Kết quả huấn luyện(trọng số của model).
Decomposition (How would you break down your problem into sub-problems?)
• Support Vector Classification.
• Naive Bayes.
• Logistic Regression.
• Decision Tree Classifier.
• Random Forest.
Pattern Recognition (Are there related solutions to draw on?)
Sử dụng các thuật toán máy học truyền thống được cài đặt sẵn phân loại để huấn luyện.
Abstraction (How would you abstract this problem?)

Iteration 7:


Problem Identification
Đánh giá:
Input: Các câu comment trong tập test.
Output: Kết quả độ đo các mô hình.
Dự đoán:
Input: Một câu comment bất kì.
Output: Loại bình luận ứng với câu đó.
Decomposition (How would you break down your problem into sub-problems?)
Đánh giá:
• Recall
• Precision
• F1 score
• Accuracy
Dự đoán:
Chọn một câu comment ngẫu nhiên từ trang shopee(chọn khoảng 100 câu để đánh giá).
Pattern Recognition (Are there related solutions to draw on?)
Tính toán các thông số trên dựa trên cách tính của một bài phân loại máy học
Abstraction (How would you abstract this problem?)

1.4. FLOWCHART:



II. BÀI TOÁN:

2.1. Xác định input và output của bài toán:

- Input:

- Một input duy nhất dưới định dạng text.
- Input đó là một hay nhiều câu comments/reviews ngắn của một người về một sản phẩm/dịch vụ trong ngôn ngữ tiếng Việt có dấu, được nhập trên cùng một hàng.
- Input không bao gồm các từ ngữ tiếng Anh, các con số và các kí tự đặc biệt như dấu câu và emojis.
- Input chỉ có thể chứa tối đa 200 từ.(vì quá 200 từ thường là spam)

- Output:

- Một output duy nhất cũng dưới định dạng dạng text.
- Output chỉ có thể là 1 trong 3 giá trị: -1, 0, 1.

"-1" - Negative: Thể hiện rằng comment/reviews của người dùng mang hàm ý phê bình, chê bai, góp ý về sản phẩm/dịch vụ.

"0" - Neutral: Thể hiện rằng comment/reviews của người dùng mang hàm ý trung tính. Output ở dạng này thường xuất hiện khi:

+ Người dùng không nêu rõ ý kiến của mình ở input.

+ Người dùng vừa thể hiện tích tiêu cực và tích cực ở 2 phần khác nhau của input.

+ Comments/Reviews của người dùng khó phân tích hoặc không liên quan đến sản phẩm/dịch vụ. (đa phần là các tin nhắn spam, hoặc là các đường link mà không phải là bình luận đánh giá,…).

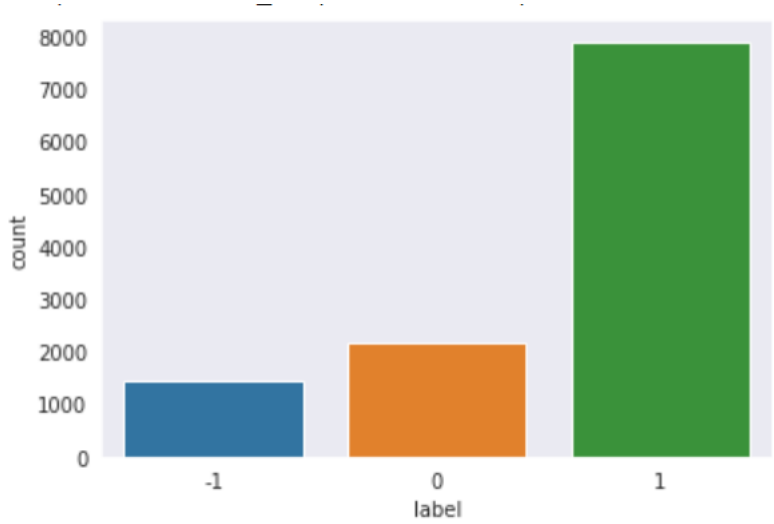
"1" - Positive: Thể hiện rằng comment/reviews của người dùng mang hàm ý ủng hộ, đánh giá cáo về sản phẩm/dịch vụ.

2.2. Chuẩn bị dữ liệu (Prepare Dataset):

- Crawl Data: Đầu tiên, dữ liệu sẽ được thu thập bằng cách crawl data từ các câu bình luận, đánh giá của khách hàng về sản phẩm/dịch vụ ở trang bán hàng online shopee sử dụng thư viện BeautifulSoup – 1 package của Python dùng để phân tích cú pháp các tài liệu HTML và XML.

➔ Kết quả, nhóm em thu về được ~11000 bình luận (file FinalData.csv) với các thông tin ('label, 'text') và được 1 thành viên trong nhóm dán nhãn thủ công.

- Phân phối dữ liệu:



1 7897

0 2150

-1 1457

- Link Colab:
https://colab.research.google.com/drive/1iHeTCzlso1NaPKKQT3K59Lamhh-VkqGU?usp=sharing

2.3. Tiền xử lý dữ liệu (Data Preprocessing):

- Import các thư viện cần thiết hỗ trợ cho việc xử lí dữ liệu:

Thư viện re (Regular Expression) dùng để so khớp các chuỗi hoặc một tập các chuỗi.

Thư viện underthesea được phát triển bởi nhóm nghiên cứu xử lý ngôn ngữ tự nhiên tiếng Việt của tác giả chính là Vũ Anh.

```
import numpy as np
import re
```

- Đưa về kiểu chữ thường:

```
#Đưa về kiểu chữ thường
def text_lowercase(text):
    return text.lower()
```

- Loại bỏ các con số: re.sub: 1 phương thức có tác dụng so khớp và thay thế chuỗi so khớp được. Với: r'\d' tương ứng với bất kỳ chữ số thập phân Unicode nào [0-9].

➔ Thay thế các chữ số tìm được bằng ‘’.

```
#Loại bỏ các con số
def remove_number(text):
    result = re.sub(r'\d+', '', text)
    return result
```

- Loại bỏ các dấu câu: Dùng hàm .replace có sẵn trong python để thay thế các dấu câu tìm được bằng “ ”.

```
#Loại bỏ các dấu câu
def remove_punctuation(text):
    text = text.replace(",", " ").replace(".", " ") \
    .replace(";", " ").replace("“", " ") \
    .replace(":", " ").replace("”", " ") \
    .replace('"', " ").replace("'", " ") \
    .replace("!", " ").replace("?", " ") \
    .replace("-", " ").replace("?", " ")
    return text
```

- Loại bỏ khoảng trắng thừa: Dùng hàm .split() để chuyển chuỗi text thành một list cắt theo separator (dấu phân tách) – separator để trống mặc định là khoảng cách. Sau đó dùng hàm .join() để chuyển list về chuỗi - các phần tử cách nhau bởi một khoảng cách “ ”.

```
#Loại bỏ khoảng trắng thừa
def remove_whitespace(text):
    return  " ".join(text.split())
```

- Loại bỏ các chữ cái giống nhau liên tiếp:

```
# Bỏ bớt các chữ cái giống nhau liên tiếp
def remove_similarletter(text):
    text = re.sub(r'([A-Z])\1+', lambda m: m.group(1).upper(), text, flags=re.IGNORECASE)
    return text
```

- Tách từ tiếng Việt sử dụng thư viện underthesea có sẵn:

```
#Tách từ tiếng Việt sử dụng thư viện underthesea có sẵn
def VN_Tokenize(text,format='text'):
  return underthesea.word_tokenize(text)
```

- Loại bỏ các stopwords tiếng Việt sử dụng bộ stopwords có sẵn:
Stopwords là những từ xuất hiện rất nhiều trong các bài viết, các đoạn text nhưng lại không hề liên quan gì đến nội dung và ý nghĩa của bài viết, gây mơ hồ, làm quá trình máy học, phân loại giảm đi độ chính xác.

```
a_lô
a_ha
ai
ai_ai
ai_nấy
ai_đó
alô
amen
anh
anh_ấy
ba
ba_ba
ba_bản
ba_cùng
ba_họ
ba_ngày
ba_ngôi
ba_tăng
bao_giờ
bao_lâu
bao_nhiêu
bao_nả
bay_biến
biết
biết_bao
biết_bao_nhiêu
biết_chắc
biết_chừng_nào
biết_mình
biết_mấy
biết_thế
biết_trước
biết_được
buổi
buổi_làm
buổi_mới
buổi_ngày
buổi_sớm
bà
bà_ấy
bài
bài_bác
bài_bỏ
bài_cái
bác
bán
bán_cấp
bán_dạ
bán_thế
bây_bẩy
bây_chừ
bây_giờ
bây_nhiêu
bèn
béng
bên
bên_bị
bên_có
bên_cạnh
bông
bước
bước_khỏi
bước_tới
bước_đi
cho_tới_khi
cho_về
cho_ăn
cho_đang
cho_được
cho_đến
cho_đến_khi
cho_đến_nỗi
choa
chui_cha
chung
chung_cho
chung_chung
chung_cuộc
chung_cục
chung_nhau
chung_qui
chung_quy
chung_quy_lại
chung_ái
chuyển
chuyển_tự
chuyển_đạt
chuyện
chuẩn_bị
chành_chạnh
chí_chết
chùn_chùn
chùn_chũn
chú
chú_dẫn
chú_khách
dào
dì
dưới
dưới_nước
dạ
dạ_bán
dạ_con
dạ_dài
dạ_dạ
dạ_khách
dần_dà
dần_dần
dầu_sao
dẫn
dẫu
dẫu_mà
dẫu_rằng
dẫu_sao
em
em_em
giá_trị
giá_trị_thực_tế
giờ
giờ_lâu
giờ_này
giờ_đi
giờ_đây
giờ_đến
giữ
giữ_lấy
giữ_ý
giữa
ngõi_trệt
ngộ_nhỡ
nhau
nhiên_hậu
nhiệt_liệt
nhung_nhăng
nhà
nhà_chung
nhà_khó
nhà_làm
nhà_ngoài
nhà_ngươi
nhà_tôi
nhà_việc
nhóm
nhón_nhén
nhất_loạt
nhất_luật
nhất_là
nhất_mực
nhất_nhất
nhất_quyết
nhất_sinh
nhất_thiết
nhất_thì
nhất_tâm
nhất_tề
nhất_đán
nhất_định
nhận_biết
nhận_họ
nhận_làm
```

- Chuẩn hóa dữ liệu: Việc chuẩn hóa là một công đoạn hết sức cần thiết, vì bộ data

```
#Loại bỏ các stopwords tiếng Việt sử dụng bộ stopwords có sẵn lấy từ
def remove_VN_stopwords(text):
    file_stopwords = pd.read_csv("vietnamese-stopwords-dash (1).txt", encoding = 'UTF-8')
    file_stopwords.columns = ["Stop_words"]

    VN_stopword = []
    for i in file_stopwords["Stop_words"]:
        VN_stopword.append(i)

    text_token = VN_Tokenize(text)
    result = [word for word in text_token if word not in VN_stopword]
    return " ".join(result)
```

chúng em thu thập là các bình luận khá là thông thường, ngẫu hứng (dữ liệu chưa sạch) trên trang thương mại điện tử, việc xuất hiện các teencode, viết tắt,… là một chuyện hết sức bình thường. Trong lúc thu thập dữ liệu, chúng em thu thập được một dict chứa các teencode, viết tắt,… Trong quá trình xử lí dữ liệu, sẽ thực hiện tìm trong các bình luận nếu chứa các từ giống với key của phần tử trong replace_list ,ta gán giá trị từ đó bằng value của key tương ứng.

```
replace_list = {
      'ship': 'vận chuyển', 'shop': 'cửa hàng', 'sho': 'cửa hàng', 'm': 'mình', 'mik': 'mình', 'ko': 'không' , 'k': 'không', 'kh': 'không'
      , 'khong': 'không', 'kg': 'không', 'khg': 'không', 'tl': 'trả lời','rep': 'trả lời', 'r': 'rồi', 'fb': 'facebook', 'face': 'faceook'
      , 'thanks': 'cảm ơn', 'thank': 'cảm ơn', 'tks': 'cảm ơn', 'tk': 'cảm ơn', 'ok': 'tốt', 'oki': 'tốt', 'okie': 'tốt', 'sp': 'sản phẩm'
      ,'dc': 'được', 'vs': 'với', 'đt': 'điện thoại', 'thjk': 'thích', 'thik': 'thích', 'qá': 'quá', 'trể': 'trễ', 'bgjo': 'bao giờ'
      , 'h': 'giờ', 'qa': 'quá', 'dep': 'đẹp', 'xau': 'xấu', 'ib': 'nhắn tin','cute': 'dễ thương', 'sz': 'size', 'good': 'tốt', 'god': 'tốt'
      , 'bt': 'bình thường','sz': 'cỡ', 'size': 'cỡ', 'đx': 'được', 'dk': 'được', 'dc': 'được', 'đk': 'được','đc': 'được'
      , 'authentic': 'chuẩn chính hãng', 'aut': 'chuẩn chính hãng', 'auth ': 'chuẩn chính hãng', 'thick': 'thích','gud': 'tốt', 'god': 'tốt'
      , 'wel done': 'tốt', 'good': 'tốt', 'gút': 'tốt','sấu': 'xấu', 'gut': 'tốt', 'tot': 'tốt', 'nice': 'tốt', 'perfect': 'rất tốt'
      , 'bt': 'bình thường','time': 'thời gian', 'qá': 'quá', 'ship': 'giao hàng','product': 'sản phẩm', 'quality': 'chất lượng', 'chat': 'chất'
      , 'excelent': 'hoàn hảo', 'bad': 'tệ', 'sad': 'tệ','beautiful': 'đẹp', 'tl': 'trả lời', 'r': 'rồi', 'order': 'đặt hàng'
      ,'chất lg': 'chất lượng', 'sd': 'sử dụng', 'dt': 'điện thoại', 'nt': 'nhắn tin', 'tl': 'trả lời', 'sài': 'xài', 'bjo': 'bao giờ',
      'thik': 'thích', 'sop': 'cửa hàng', 'fb': 'facebook', 'face': 'facebook', 'very': 'rất','dep': 'đẹp', 'xau': 'xấu', 'iu': 'yêu'
      , 'fake': 'giả mạo', 'trl': 'trả lời', '><': 'tiêu cực','por': 'tệ', 'poor': 'tệ', 'ib': 'nhắn tin', 'rep': 'trả lời', 'fback': 'feedback'
      , 'fedback': 'feedback','bin': 'pin', 'cx': 'cũng', 'nch': 'nói chuyện', 'ntn': 'như thế nào', 'vde': 'vấn đề'
  }
```

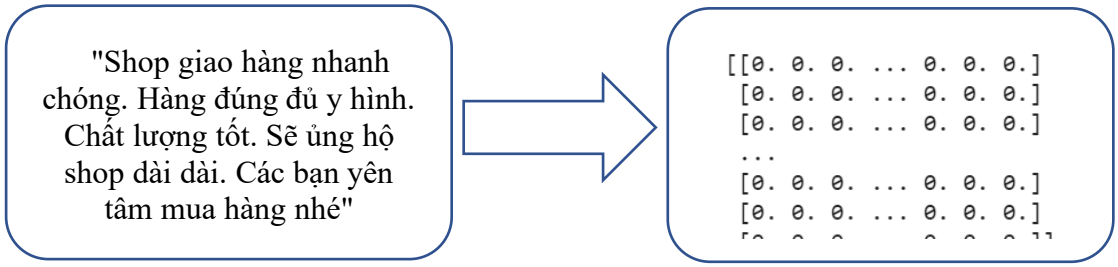
replace_list() thu thập được

```
text = text.split()
len_ = len(text)
for i in range(0, len_):
  for k, v in replace_list.items():
    if (text[i]==k):
      text[i] = v
return " ".join(text)
```

- Cuối cùng chúng em tổng hợp các hàm về một hàm xử lý dữ liệu (Text_PreProcessing_noutil) để thuận tiện cho việc sử dụng:

```
def Text_PreProcessing_noutil(data):
  result_1 = []
  for i in data:
    i = str(i)
    text = text_lowercase(i)
    text = Util(text)
    text = remove_similarletter(text)
    text = remove_number(text)
    text = remove_punctuation(text)
    text = remove_whitespace(text)
    text = remove_VN_stopwords(text)
    result_1.append(text)
  return result_1
```

TF-IDF (Term Frequency - Inverse Document Frequency):
TF: tần số xuất hiện của 1 từ trong 1 văn bản
IDF: tần số nghịch của 1 từ trong một tập các văn bản
Kĩ thuật TF-IDF dùng để tính toán mức độ quan trọng của từ trong một văn bản.
TfidfVectorizer dùng để chuyển đổi dữ liệu văn bản sang ma trận các features TF-IDF.



2.4. Xây dựng và Huấn luyện model (Choosing and Training model):

- Train_test_split: Chia dữ liệu (dataset) thành train set và test set để huấn luyện và thử nghiệm trên tập dữ liệu thu thập được theo tỉ lệ train/test ứng với 70/30.

```
X_train_1, X_test_1, Y_train_1, Y_test_1 = train_test_split(X_data_tfidf_1, Y_data_1, test_size=0.3)
```

- Đánh giá model:

```
from sklearn.metrics import accuracy_score
from sklearn.metrics import f1_score
```

- Cách tính độ chính xác của model bằng score thông thường chỉ cho ta biết phần trăm dữ liệu được phân loại đúng mà không chỉ ra được dữ liệu được phân loại như thế nào, nên ta sử dụng một ma trận được gọi là confusion matrix.
- Confusion matrix giúp ta có cái nhìn chi tiết hơn trong quá trình chọn lọc model dựa trên tập dữ liệu có sẵn.
- Để đánh giá chất lượng của model, ta sử dụng khái niệm F1-score, khái niệm này dựa trên 2 khái niệm khác là Precision và Recall.
- Bài toán lần này có 3 class (tích cực, tiêu cực và trung tính) nên sẽ có True/False Positive, True/False Negative, True/False Neutral.

True label \ Predicted label	-1	0	1
-1	**True Negative**	False Neutral	False Positive
0	False Negative	**True Neutral**	False Positive
1	False Negative	False Neutral	**True Positive**

- Precision_1: là tỉ lệ số điểm True Negative trong số những điểm được phân loại là Negative.

Precision_1 = TNeg/(TNeg + FNeg + FNeg)

- Precision_2: là tỉ lệ số điểm True Neutral trong số những điểm được phân loại là Neutral.

Precision_2 = TNeu/(TNeu + FNeu + FNeu)

- Precision_3: là tỉ lệ số điểm True Positive trong số những điểm được phân loại là Positive.

Precision_3 = TPo/(TPo + FPo + FPo)

- Recall_1: là tỉ lệ số điểm True Negative trong số những điểm thực sự là Negative.

Recal_1 = TNeg/(TNeg + FNeu + FPo)

- Recall_2: là tỉ lệ số điểm True Neutral trong số những điểm thực sự là Neutral.

Recal_2 = TNeu/(TNeu + FNeg + FPo)

- Recall_3: là tỉ lệ số điểm True Positive trong số những điểm thực sự là Positive.

Recal_3 = TPo/(TPo + FNeu + FNeg)

- F1-Score_1 là hàm harmonic mean của Precision_1 và Recall_1.

F1-Score_1 = 2 x (Precision_1 x Recall_1)/(Precision_1 + Recall_1)

- F1-Score_2 là hàm harmonic mean của Precision_2 và Recall_2.

F1-Score_2 = 2 x (Precision_2 x Recall_2)/(Precision_2 + Recall_2)

- F1-Score_3 là hàm harmonic mean của Precision_3 và Recall_3.

F1-Score_3 = 2 x (Precision_3 x Recall_3)/(Precision_3 + Recall_3)

➔ F1-Score = (F1-Score_1 + F1-Score_2 + F1-Score_3)/3

- Tiến hành thử nghiệm với các model khác nhau để tìm được một model tốt nhất, phù hợp nhất cho đồ án lần này thông qua F1_Score trong quá trình training model.

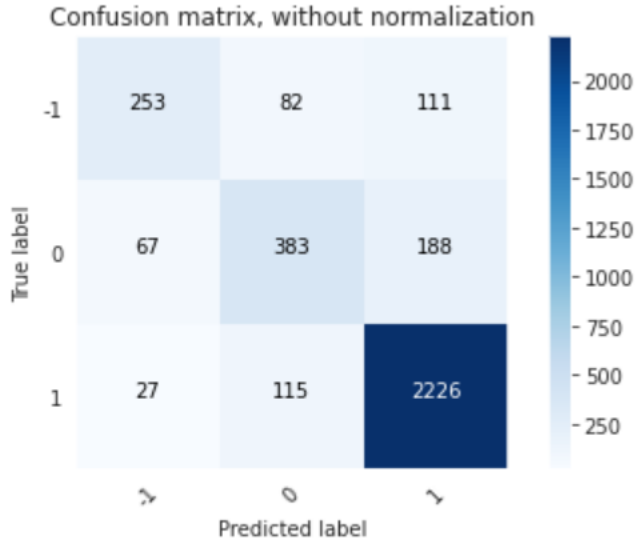
```
from sklearn.svm import SVC
from sklearn.pipeline import Pipeline
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics import classification_report
```

2.5. KẾT QUẢ:

- Mô hình SVM:

```
Model SVC
Train score:  0.9119473422752111
Test score:  0.8290845886442642
F1 score:  0.7256181028929145

Confusion matrix, without normalization
[[ 253   82  111]
 [  67  383  188]
 [  27  115 2226]]
```

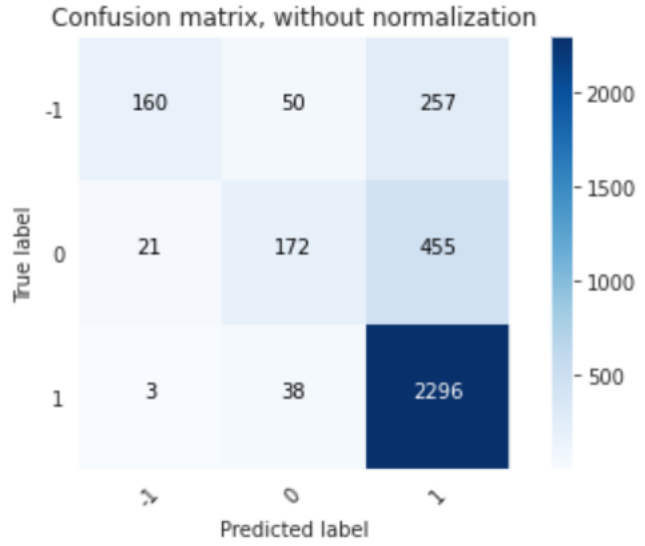


	precision	recall	f1-score	support
-1	0.73	0.57	0.64	446
0	0.66	0.60	0.63	638
1	0.88	0.94	0.91	2368
accuracy			0.83	3452
macro avg	0.76	0.70	0.73	3452
weighted avg	0.82	0.83	0.82	3452

- Mô hình Multinominal Naïve Bayes:

```
Model MultinomialNB
Train score:  0.8181818181818182
Test score:  0.761297798377752
F1 score:  0.5765089194569053
```

```
Confusion matrix, without normalization
[[ 160   50  257]
 [  21  172  455]
 [   3   38 2296]]
```

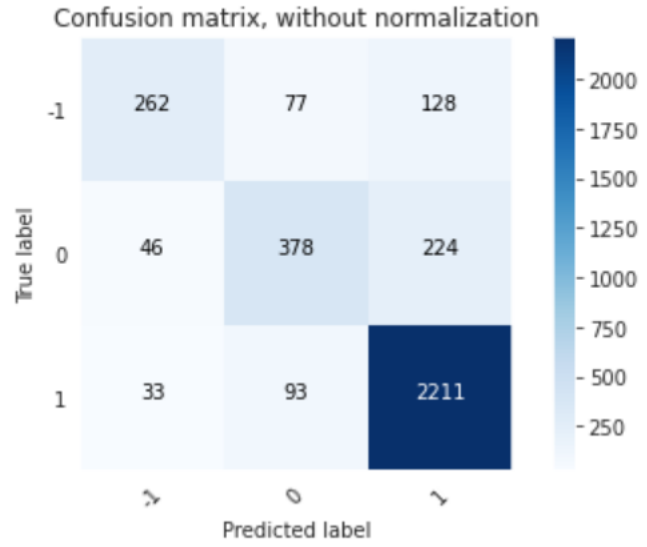


	precision	recall	f1-score	support
-1	0.87	0.34	0.49	467
0	0.66	0.27	0.38	648
1	0.76	0.98	0.86	2337
accuracy			0.76	3452
macro avg	0.76	0.53	0.58	3452
weighted avg	0.76	0.76	0.72	3452

- Mô hình Logistic Regression:

```
Model LogisticRegression
Train score:  0.8970442126179831
Test score:  0.8258980301274623
F1 score:  0.7276902848294521
```

```
Confusion matrix, without normalization
[[ 262   77  128]
 [  46  378  224]
 [  33   93 2211]]
```

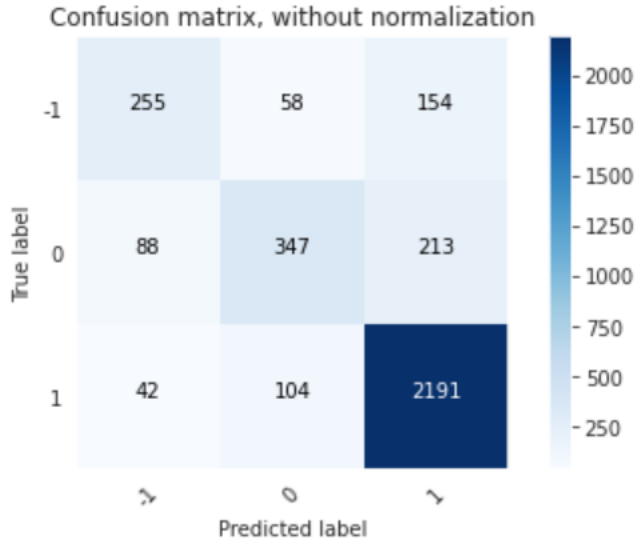


	precision	recall	f1-score	support
-1	0.77	0.56	0.65	467
0	0.69	0.58	0.63	648
1	0.86	0.95	0.90	2337
accuracy			0.83	3452
macro avg	0.77	0.70	0.73	3452
weighted avg	0.82	0.83	0.82	3452

- Mô hình Random Forest:

```
Model RandomForestClassifier
Train score:  0.9846000993541977
Test score:  0.8090961761297798
F1 score:  0.6978726237627962
```

```
Confusion matrix, without normalization
[[ 255   58  154]
 [  88  347  213]
 [  42  104 2191]]
```

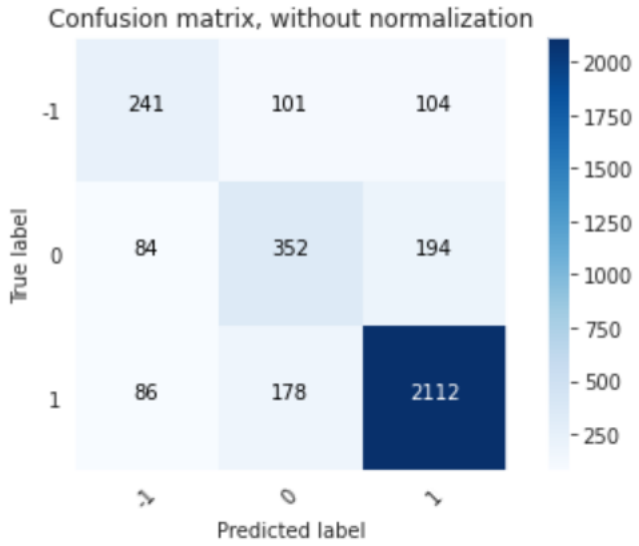


	precision	recall	f1-score	support
-1	0.66	0.55	0.60	467
0	0.68	0.54	0.60	648
1	0.86	0.94	0.90	2337
accuracy			0.81	3452
macro avg	0.73	0.67	0.70	3452
weighted avg	0.80	0.81	0.80	3452

- Mô hình Decision Tree:

```
Model DecisionTree
Train score:  0.9898161947342276
Test score:  0.783603707995365
F1 score:  0.6677627732018866
```

```
Confusion matrix, without normalization
[[ 241  101  104]
 [  84  352  194]
 [  86  178 2112]]
```



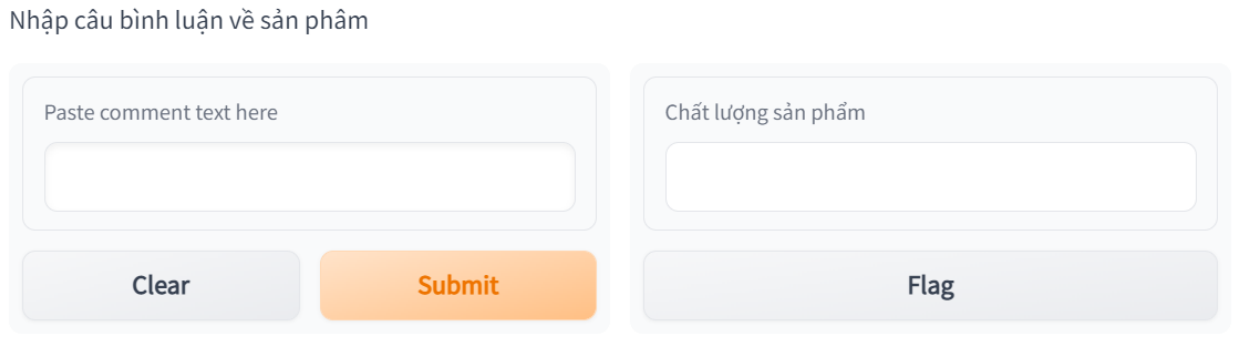
	precision	recall	f1-score	support
-1	0.59	0.54	0.56	446
0	0.56	0.56	0.56	630
1	0.88	0.89	0.88	2376
accuracy			0.78	3452
macro avg	0.67	0.66	0.67	3452
weighted avg	0.78	0.78	0.78	3452

Bảng so sánh kết quả chạy của các mô hình:

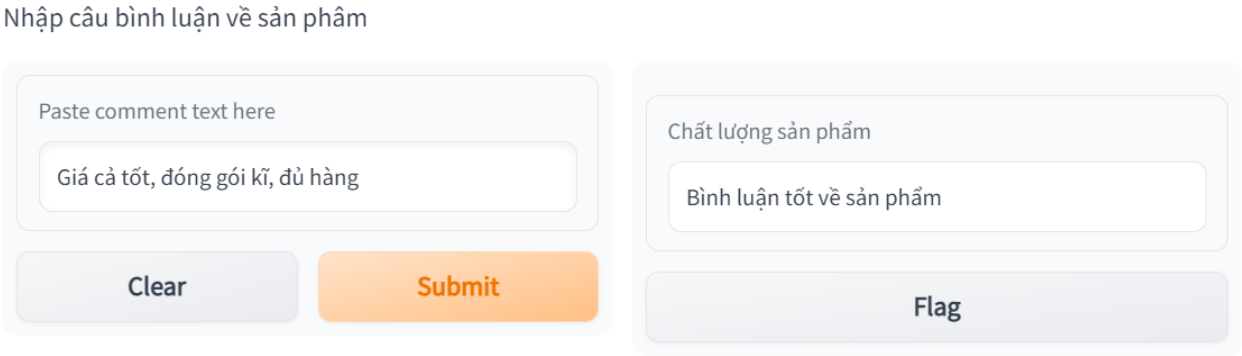
Kết quả	SVM	Naïve Bayes	Logistic Regression	Random Forest	Decision Tree
Recall	0.70	0.53	0.70	0.68	0.66
Precision	0.76	0.76	0.77	0.73	0.67
F1_score	0.73	0.58	0.73	0.70	0.67
Accuracy	0.83	0.76	0.83	0.81	0.78

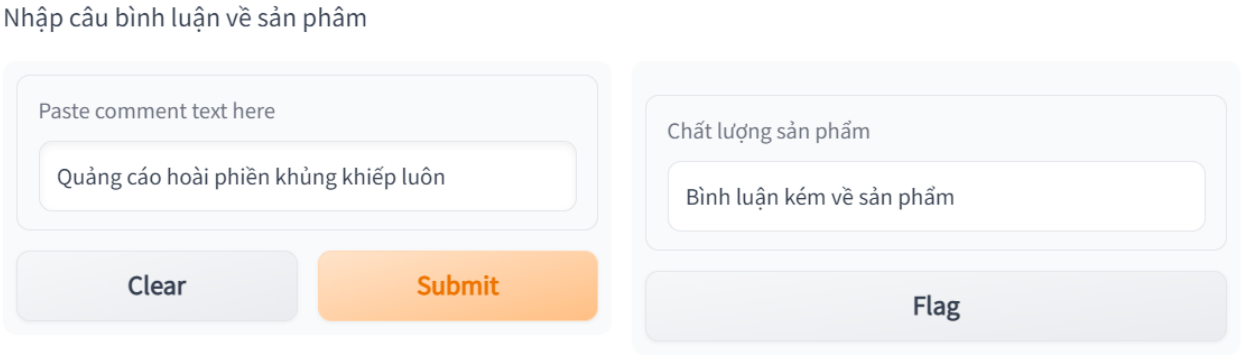
III. ỨNG DỤNG DEMO:

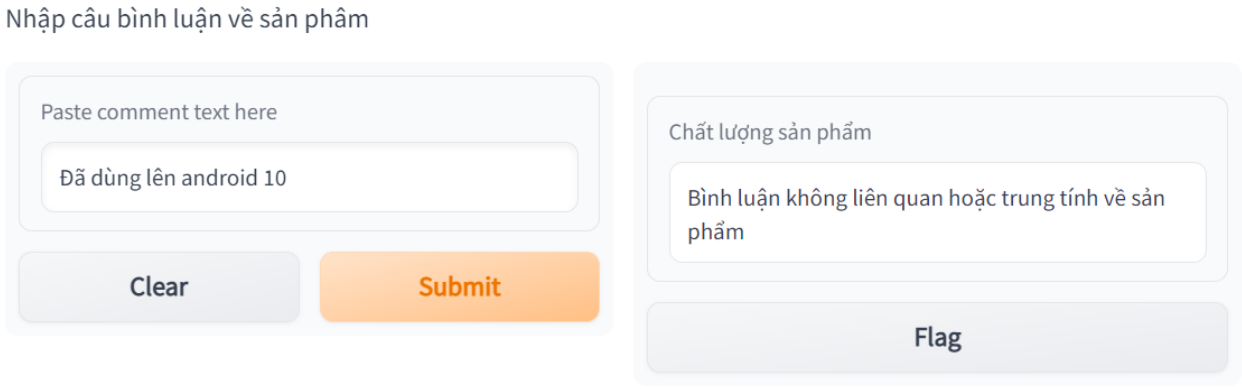
- Giao diện thông thường của Web:



- Một số bình luận mẫu và kết quả trả về:







IV. BẢNG PHÂN CÔNG:

Người Nhận	MSSV	Công việc được phân công	Đánh giá tỉ lệ hoàn thành
Hoàng Gia Huy	19521607	Thực hiện crawl data, thu thập data, label các nhãn, tổng hợp data hoàn chỉnh, xây dựng demo, tiền xử lý data.	100%
Bùi Thị Bích Hậu	19521483	Label data, format báo cáo, debug code,làm poster.	100%
Đoàn Tấn Phát	20520269	Thực hiện đánh giá, viết báo cáo, training model, tiền xử lý data.	100%

V. TÀI LIỆU THAM KHẢO:

1. https://machinelearningcoban.com/2017/08/31/evaluation/#truefalsepositivenegative
2. https://www.digitalocean.com/community/tutorials/how-to-scrape-webpages-with-beautiful-soup-and-python-3
3. https://blog.vietnamlab.vn/2019/08/04/xay-dung-1-model-machine-learningdon-gian-de-giai-quyet-bai-toan-phan-loai-sac-thai-binh-luan-trong-tiengviet/
4. https://codetudau.com/bag-of-words-tf-idf-xu-ly-ngon-ngu-tunhien/index.html
5.https://scikitlearn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html
6. https://towardsdatascience.com/how-sklearns-tf-idf-is-different-from-thestandard-tf-idf-275fa582e73d
7. https://machinelearningmastery.com/precision-recall-and-f-measure-forimbalancedclassification/?fbclid=IwAR3NT5Ui6YmU4i8lDTCt9sTekHNjyWg4-vn4HSto8aZg5OP5yVhxHupyVpc#:~:text=Once%20precision%20and%20recall%20have
8. https://realpython.com/python-gui-tkinter/
9. https://docs.python.org/3/library/re.html
10. https://github.com/undertheseanlp/word_tokenize
11. Code được tham khảo từ nhóm:

https://github.com/Long-1234kfgkl/CS114.K21/blob/master/BaoCaoCuoiKy_CS114.K21/Main.ipynb

https://viblo.asia/p/phan-tich-phan-hoi-khach-hang-hieu-qua-voi-machine-learningvietnamese-sentiment-analysis-Eb85opXOK2G

Working With Text Data — scikit-learn 0.24.2 documentation

CS114.K21/Text_Classification.ipynb at master · ThuanPhong0126/CS114.K21 (github.com)