ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

ĐỒ ÁN MÔN TÍNH TOÁN ĐA PHƯƠNG TIỆN

ĐỀ TÀI: PHÂN TÍCH BÌNH LUẬN CỦA MỘT TRANG THƯƠNG MẠI ĐIỆN TỬ

GVPT: Lê Đình Duy
Phạm Nguyễn Trường An

Lớp: CS114.K21
18521060 – Trịnh Hưng Long
18521062 – Hà Văn Luân
18521274 – Lỡ Đình Phương

TPHCM, ngày 6 tháng 8 năm 2020

MỤC LỤC

1 Tổng quan

1.1 Thông tin nhóm

STT	Mã số sinh viên	Họ và Tên
1	18521060	Trịnh Hưng Long
2	18521062	Hà Văn Luân
3	18521274	Lỡ Đình Phương

1.2 Giới thiệu đề tài

- Hiện nay nhu cầu mua hàng qua mạng của người dùng ngày càng trở nên phát triển mạnh hơn do những lợi ích mà nó mang lại như tiện lợi, chi phí rẻ, có nhiều chương trình khuyến mãi hấp dẫn, có thể ngồi ở nhà để xem sản phẩm mà không cần phải đến tận nơi để xem, … Tuy nhiên, việc mua hàng qua mạng cũng có những nhược điểm, trong đó có việc người dùng không thể tận mắt đánh giá sản phẩm của mình như mua trực tiếp tại các cửa hàng được. Vì vậy, các mục bình luận về sản phẩm của những người dùng đã sử dụng qua sản phẩm đóng vai trò quan trọng trong việc đánh giá chất lượng các sản phẩm tương ứng, các bình luận chủ yếu gồm 3 loại: tích cực, trung tính hoặc tiêu cực. Tuy nhiên, số lượng các bình luận trên các trang thương mại điện tử rất nhiều, gây khó khăn khi đánh giá từng bình luận bằng tay, vì vậy, các thuật toán máy học sẽ hỗ trợ đắc lực cho việc phân loại này.
- Để minh chứng cho đề tài này, chúng em sẽ tập trung phân loại các sản phẩm là "điện thoại" từ các bình luận trên một trang thương mại điện tử cụ thể đó là trang “*Thế Giới Di Động*”, hỗ trợ cho việc mua sắm điện thoại của người tiêu dùng ngày nay.
- Ngôn ngữ sử dụng: Python.

2 Đồ án

2.1 Mô tả đồ án

2.1.1 Ý tưởng

- Đưa một đoạn bình luận của 1 sản phẩm bất kỳ và đưa ra kết quả dự đoán bình luận (là tích cực, tiêu cực hoặc trung tính).

2.2.2 Input & Output

- Input: Một bình luận về sản phẩm điện thoại của khách hàng bằng tiếng Việt.
- Output: Bình luận đó là tiêu cực, trung tính hay tích cực (3 class).

2.2 Chuẩn bị dữ liệu *(Prepare Dataset)*

- Crawl data (các bình luận sản phẩm trên trang) từ trang thương mại điện tử: '*https://www.thegioididong.com/dtdd*' *(Thế Giới Di Động)* bằng thư viện BeautifulSoup – 1 package Python dùng để phân tích cú pháp các tài liệu HTML và XML.
- Ta thu về được 4679 bình luận (*file new_data_machinelearning.json*) với 4 features ('TYPE', 'NAME', 'COMMENT', 'RATING COMMENT') các nhãn được gán tự động theo các tiêu chí với:

✓ Các bình luận có số lượng đạt 5 sao là 5 sẽ cho nhãn bằng 1.

✓ Các bình luận có số lượng đạt 5 sao là 4 và 3 sẽ cho nhãn bằng 0.

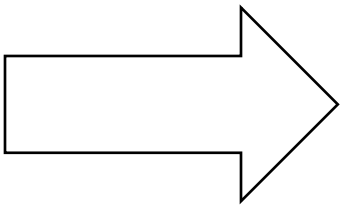
✓ Các bình luận có số lượng đạt 5 sao là 1 và 2 sẽ cho nhãn bằng -1.

- Sau đó, tụi em sẽ thực hiện kiểm tra lại các nhãn các bình luận của bộ data trên bằng tay (với mỗi bạn kiểm tra xấp sỉ khoảng 1500 bình luận), thu gọn và làm cân bằng bộ data trên -> thu về được 3000 bình luận (1000 bình luận

tích cực, 1000 bình luận trung tính và 1000 bình luận tiêu cực) – *file new_data-1-1000 (1).json*.

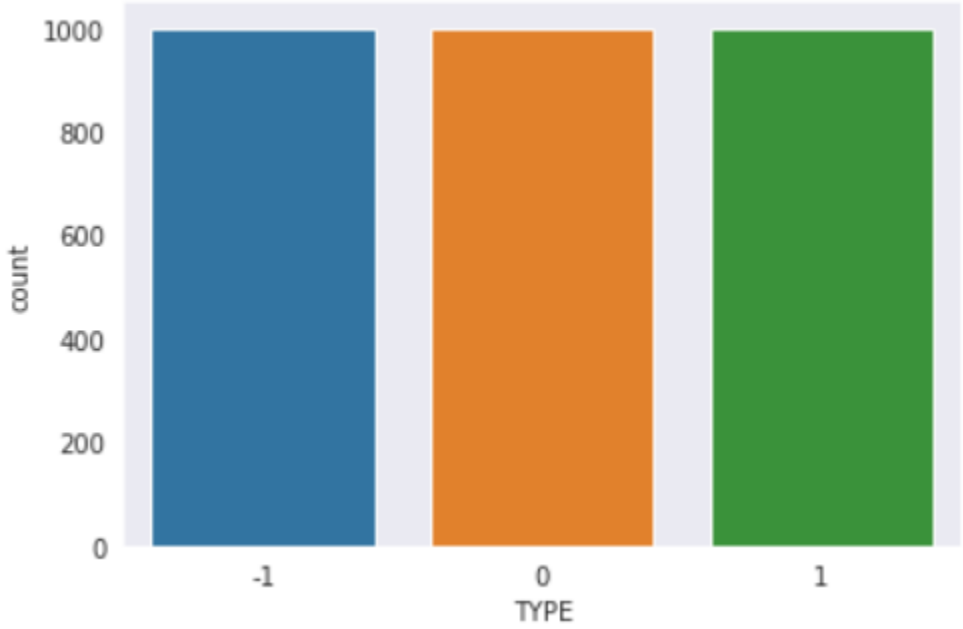
*Link chứa 2 file *new_data_machinelearning.json* và *new_data-1-1000 (1).json*:

https://drive.google.com/drive/folders/1CZYJ0TWsKcO502sDbPkW68Y_vTXCu8hw



File chứa 4679 bình luận sau khi thực hiện crawl data với các nhãn được gán tự động theo các tiêu chí trên.

File chứa 3000 bình luận sau khi thực hiện kiểm tra lại bằng tay các nhãn, cân bằng thu gọn lại (1000 tích cực, 1000 tiêu cực, 1000 trung tính).



Biểu đồ minh họa bộ data hoàn chỉnh

* Link colab:
https://colab.research.google.com/drive/1ApzLOcKpevmINQXFR9llo15pFWLjjMHa?usp=sharing

2.2 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 re
import underthesea
```

- Đưa đoạn text về chữ thường (lower):

```
def text_lowercase(text):
    return text.lower()
```

- Loại bỏ các con số có trong đoạn text:
  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 ‘’.

```
def remove_number(text):
    result = re.sub(r'\d', '', text)
    return result
```

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

```
def remove_punctuation(text):
    text = text.replace(",", " ").replace(".", " ") \
    .replace(";", " ").replace("“", " ") \
    .replace(":", " ").replace("”", " ") \
    .replace('"', " ").replace("'", " ") \
    .replace("!", " ").replace("?", " ") \
    .replace("-", " ").replace("?", " ")
    return text
```

- Loại bỏ các khoảng trắng thừa có trong đoạn text:
  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 " ".

```
def remove_whitespace(text):
    return  " ".join(text.split())
```

- Loại bỏ các kí tự giống nhau liên tiếp có trong đoạn text:
  VD: quaaaa -> qua, kkkkkkkk -> k,…

```
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* :

```
def VN_Tokenize(text,format='text'):
  return underthesea.word_tokenize(text)
```

- Loại bỏ stopwords tiếng việt có trong đoạn text:
  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.
  Trong quá trình tìm kiếm, chúng em thu thập được 1 file *vietnamese-stopwords-dash (1).txt* chứa một list những từ stopwords Tiếng Việt của tác giả *Lê Văn Duyệt* và chúng em có tinh chỉnh lại file cho phù hợp với đồ án lần này.
  *Link : *https://github.com/stopwords/vietnamese-stopwords/blob/master/vietnamese-stopwords-dash.txt*
  *Link chứa file *vietnamese-stopwords-dash (1).txt*: https://drive.google.com/file/d/1pVm9Friucg_1ghhUzEkceoHVgaFK9TJ4/view

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

```
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)
```

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

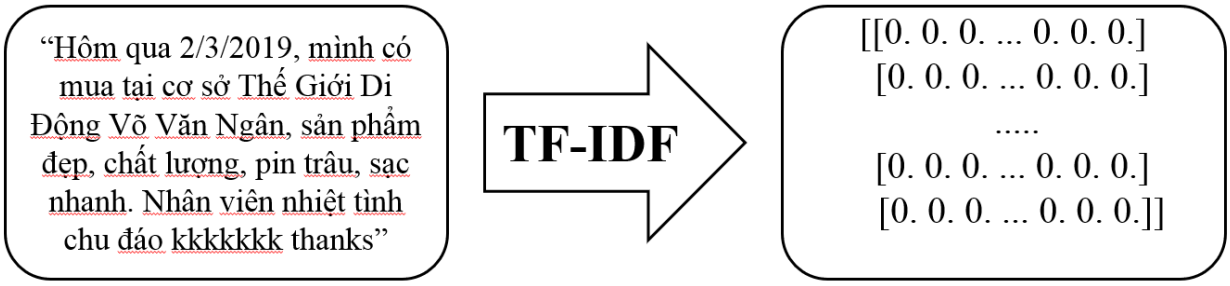
```
def Util(text):
    text = text.split()
    len_ = len(text)
    for i in range(0, len_):
      for k, v in replace_list.items():
        if (text[i]==k):
          text[i] = v
    return " ".join(text)
```

- Sau đó 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.



2.3 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 80/20.

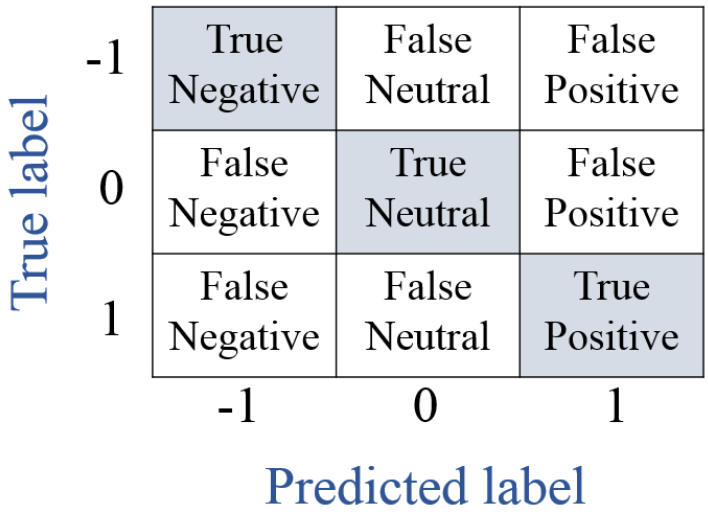
```
from sklearn.model_selection import train_test_split
X_train_1, X_test_1, Y_train_1, Y_test_1 = train_test_split(X_data_tfidf_1, Y_data_1, test_size=0.2)
```

- 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.*



Confusion Matrix

- ✓ Precision_1: là tỉ lệ số điểm True Negative trong số những điểm được phân loại là Negative.
- ✓ Precision_2: là tỉ lệ số điểm True Neutral trong số những điểm được phân loại là Neutral.
- ✓ Precision_3: là tỉ lệ số điểm True Positive trong số những điểm được phân loại là Positive.

$$\text{Precision\_1} = \frac{\text{True Negative}}{\text{True Negative + False Negative +False Negative}}$$

$$\text{Precision\_2} = \frac{\text{True Neutral}}{\text{True Neutral + False Neutral +False Neutral}}$$

$$\text{Precision\_3} = \frac{\text{True Positive}}{\text{True Positive + False Positive +False Positive}}$$

- ✓ Recall_1: là tỉ lệ số điểm True Negative trong số những điểm thực sự là Negative.
- ✓ Recall_2: là tỉ lệ số điểm True Neutral trong số những điểm thực sự là Neutral.

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

$$\text{Recall\_1} = \frac{\text{True Negative}}{\text{True Negative + False Neutral +False Positive}}$$

$$\text{Recall\_2} = \frac{\text{True Neutral}}{\text{True Neutral + False Negative +False Positive}}$$

$$\text{Recall\_3} = \frac{\text{True Positive}}{\text{True Positive + False Neutral +False Negative}}$$

- ✓ F1-Score_1 là hàm harmonic mean của Precision_1 và Recall_1.
- ✓ F1-Score_2 là hàm harmonic mean của Precision_2 và Recall_2.
- ✓ F1-Score_3 là hàm harmonic mean của Precision_3 và Recall_3.

$$\text{F1-Score\_1} = \frac{2 \text{ x } (\text{Precision\_1} + \text{Recall\_1})}{\text{Precision\_1} + \text{Recall\_1}}$$

$$\text{F1-Score\_2} = \frac{2 \text{ x } (\text{Precision\_2} + \text{Recall\_2})}{\text{Precision\_2} + \text{Recall\_2}}$$

$$\text{F1-Score\_3} = \frac{2 \text{ x } (\text{Precision\_3} + \text{Recall\_3})}{\text{Precision\_3} + \text{Recall\_3}}$$

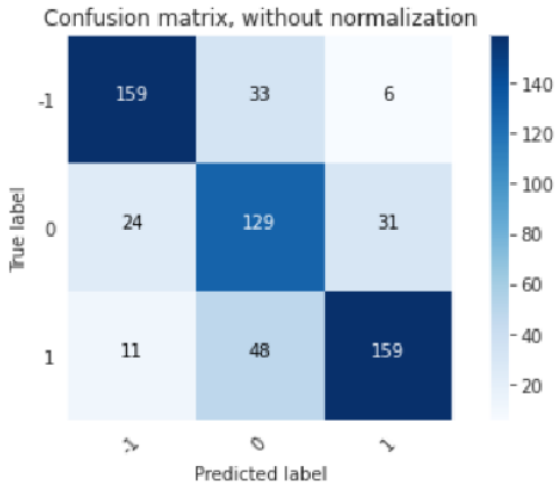
$$\textbf{F1-Score} = \frac{\text{F1-Score\_1} + \text{F1-Score\_2} + \text{F1-Score\_3}}{3}$$

- 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.
- Chúng em nhắm tới 4 model:
  - ❖ SVC.
  - ❖ Multinomial Naïve Bayes.
  - ❖ Logistic Regression.
  - ❖ Random Forest.

```
from sklearn.naive_bayes import MultinomialNB
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.svm import LinearSVC
```

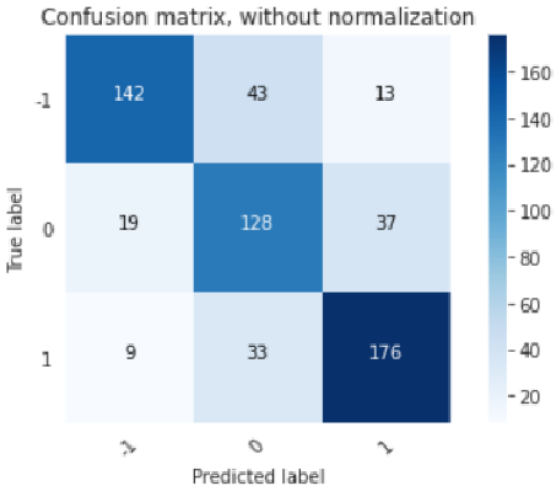
- Kết quả thu về được:
  * Link colab:
  https://colab.research.google.com/drive/1BfdrT8tM_aTOLOyD9v6PaEhh5bsrcRpq?usp=sharing&fbclid=IwAR3S2uA5G9e6tA0PJcqDjX6J02yUQf05DikwE-XehcliDH6JHCjknZvAcsQ#scrollTo=PsOD6b2EqC3Y

Model SVC



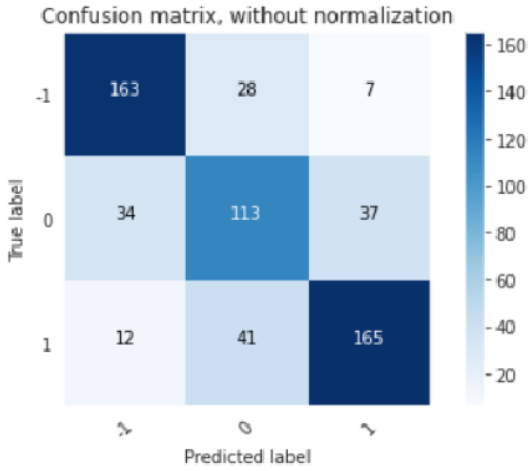

```
Model SVC
Train score:  0.9079166666666667
Test score:  0.745
F1 score:  0.7447209222834282
```

Multinomial Naïve Bayes



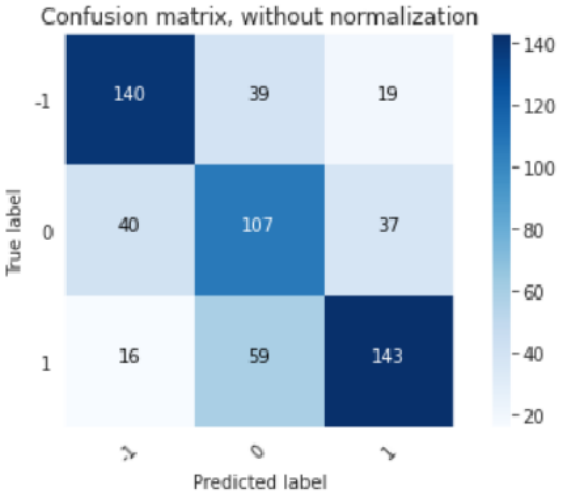

```
Model MultinomialNB
Train score:  0.8495833333333334
Test score:  0.7433333333333333
F1 score:  0.7414419125535217
```

Logistic Regression




```
Model LogisticRegression
Train score:  0.8554166666666667
Test score:  0.735
F1 score:  0.7304342878113371
```

Random Forest




```
Model RandomForestClassifier
Train score:  0.99125
Test score:  0.65
F1 score:  0.6488799173754572
```

- **Nhận xét:**
  F1 Score của model SVC ~ 0.7447
  F1 Score của model Multinomial Naïve Bayes ~ 0.7414
  F1 Score của model Logistic Regression ~ 0.7304
  F1 Score của model Random Forest ~ 0.6489
  ➔ *F1 Score* của model SVC đạt giá trị cao nhất, chọn model này xây dựng ứng dụng.

2.4 Thiết kế giao diện cho người dùng

- Tiến hành import Tkinter – một package trong python có chứa module Tk hỗ trợ cho việc lập trình GUI:

```
import tkinter as tk
test=tk.Tk()
```

- .title("tên giao diện") : đặt tên cho giao diện lập trình.
- Để thiết kế một giao diện hoàn thiện, ta sử dụng đến các khái niệm canvas, frame, label, text và button,… ngay trong thư viện tkinter:

❖ Canvas: dùng để vẽ các hình học không gian nhất định (hình chữ nhật, hình vuông, hình tròn,…) cho giao diện:

- .Canvas (parameter, height= “chiều cao mong muốn”, width= “chiều rộng mong muốn”).
- .pack() để hiển thị lên giao diện.
- Ta tạo một hình chữ nhật với chiều cao = 700 và chiều rộng = 800 làm một khung chính cho toàn bộ giao diện.

```
HEIGHT = 700
WIDTH = 800
canvas=tk.Canvas(test, height=HEIGHT,width=WIDTH)
canvas.pack()
```

❖ Frame: dùng để tạo khung hỗ trợ thuận tiện cho việc đưa các label, entry, text, button,… lên giao diện dễ dàng:

- .Frame(parameter, bg=”màu background mong muốn”).
- Hàm .place: vị trí mong muốn trên giao diện (tính theo tọa độ).
- .place(relx=”hoành độ điểm xuất phát”, rely=”tung độ điểm xuất phát”, relwidth=”chiều rộng mong muốn ”,relheight=”chiều cao mong muốn ”.

```
frame1=tk.Frame(test,bg="#11b2f2",bd=5)
frame1.place(relx=0.04,rely=0.1,relwidth=0.5,relheight=0.5)
```

❖ Text: tạo một Text trên giao diện:

- .Text(parameter, font=(‘Tên front’, cỡ chữ)).

```
entry =tk.Text(frame1,font=('Times New Roman',14))
entry.focus_get()
entry.place(relwidth=1,relheight=1)
entry.pack()
```

❖ Button: tạo một nút bấm trên giao diện:

- .Button(parameter, text=”tên nút”, font=(“tên font”,cỡ chữ), command=”một hàm ta truyền vào để thực hiện sau khi thao tác trên nút bấm này”).

```
button1=tk.Button(frame3,text='XÓA',font=('Times New Roman',10),command=clearTextInput)
button1.place(relwidth=1,relheight=1)
```

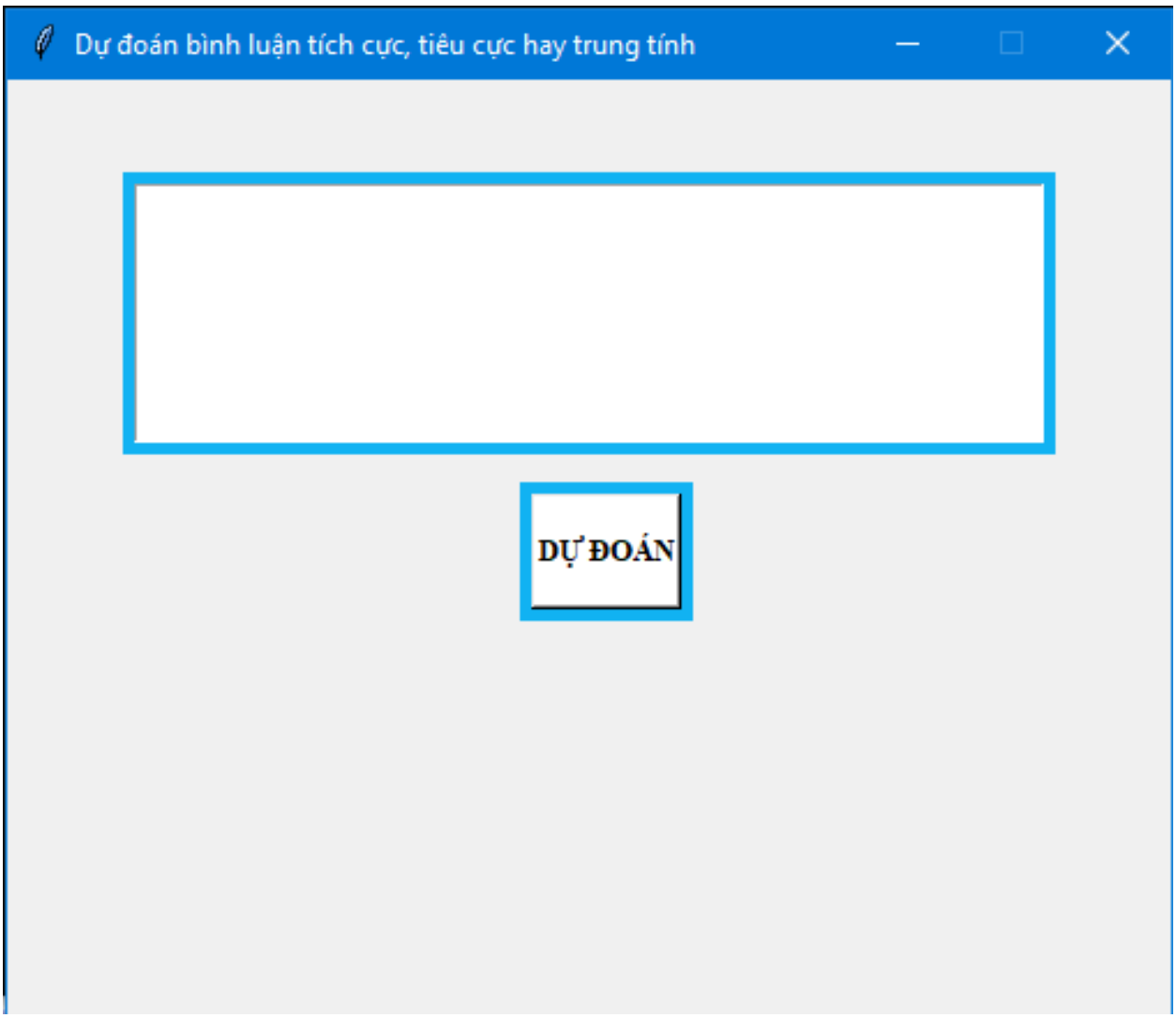
❖ Label: tạo một nhãn trên giao diện (thường được dùng để hiển thị kết quả):

- .Lable(parameter,text=”tên nhãn”, font=(“tên font”,cỡ chữ), bg=” màu background mong muốn).

```
lable5=tk.Label(frame5,text="CHỦ ĐỀ:", font =('Times New Roman',10,"bold"),bg='white')
lable5.place(relwidth=0.25,relheight=1)
```

- <Tên giao diện>.mainloop():
  Dừng giao diện lại xem kết quả.

```
tk.mainloop()
```



Hình ảnh giao diện thiết kế được

2.5 Kết quả thực nghiệm

Thử nghiệm với 5 bình luận đánh giá sản phẩm bất kì với model SVC được chọn.

*Các bình luận dưới đây được lấy từ trang web tiki.vn.

Bình luận 1: Đầu tiên phải nói về khâu giao hàng, mình đặt hàng từ tối hôm trước thế là sang chiều hôm sau là đã giao tới rồi giao rất nhanh(giao hàng tiêu chuẩn). Hàng nguyên seal full box kèm phụ kiện như là cáp sạc nhanh Vooc, dây sạc và ốp lưng bằng nhựa. Trải nghiệm ban đầu về máy phải nói là máy rất mượt, cấu hình chơi game mạnh và đặc biệt là pin rất khoẻ. Do có sạc nhanh nên thời gian sạc cũng nhanh, pin vừa khoẻ mà sạc lại nhanh thì còn gì bằng :)). Máy bắt wifi tốt và màn hình 6.3 inch mang lại trải nghiệm tốt khi xem phim. Đối với ai chỉ cần chụp

hình thông thường selfie các kiểu thì sẽ thấy máy ảnh của realme 3 pro thật sự tốt, với giá 3tr750 thì đây là một chiếc điện thoại phải nói là thật sự xứng đáng để bỏ tiền mua. Dưới đây là một số tấm hình mình chụp bằng Realme 3 Pro

-Sản phẩm: Realme 3 Pro (128GB/6GB)

-Khách hàng: Hoàng Tấn Đức

-Kết quả:

Bình luận 2: Đây là lần thứ hai mình mua Iphone trên Tiki. Về sản phẩm thì mình tin tưởng Tiki, sản phẩm chính hãng Apple. Mã VN/A. Mình đang dùng thử, mới được 1 tuần. Hi vọng sản phẩm sẽ không có lỗi gì. Nhưng mình khá thất vọng vì việc giao hàng lần này. Thời gian giao hàng quá lâu. Nhân viên còn định gửi hàng cho cái quầy tạp hóa dưới tòa nhà mình ở, cầm hộ khi mình bảo chờ mình 10' (Vì giao hàng không có đúng ngày hẹn ban đầu nên mình phải vòng từ cty về)'.

-Sản phẩm: Iphone 7 Plus 128GB

-Khách hàng: Phạm Bá Hiếu

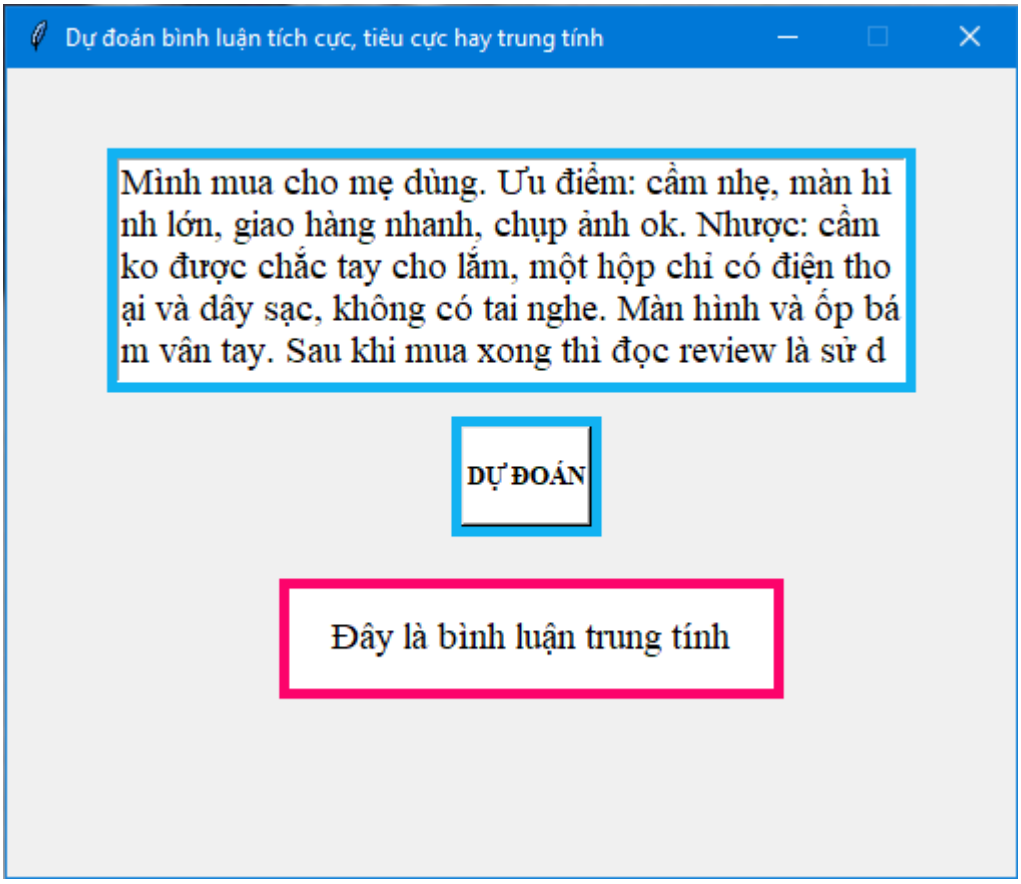
-Kết quả:

Bình luận 3: Mình mua cho mẹ dùng. Ưu điểm: cầm nhẹ, màn hình lớn, giao hàng nhanh, chụp ảnh ok. Nhược: cầm ko được chắc tay cho lắm, một hộp chỉ có điện thoại và dây sạc, không có tai nghe. Màn hình và ốp bám vân tay. Sau khi mua xong thì đọc review là sử dụng chip cũ snap450 cùi :)). Giá không phù hợp với cấu hình. Mua trả góp nên mình không huỷ được đơn hàng :(

-Sản phẩm: Samsung Galaxy A11

-Khách hàng: Dohuyen

-Kết quả:



Bình luận 4: Điện thoại phù hợp giá tiền thôi, mua cho bà để gọi điện với con cháu mà camera xấu tệ :)) , pin tuột nhanh, cấu hình tạm ổn, mình thấy mọi người hỏi chơi game được ko thì mình tải thử Free Fire cũg tạm hơi lag nhẹ nhưng vẫn chơi đc. mặt sau dễ bám vân tay,loa hơi nhỏ,nói chung tốt trog tầm giá, ngoài camara thì ko nói gì đc. Giao hàng nhanh.

-Sản phẩm: Vsmart Star 3

-Khách hàng: Nguyễn Thị Tâm Như

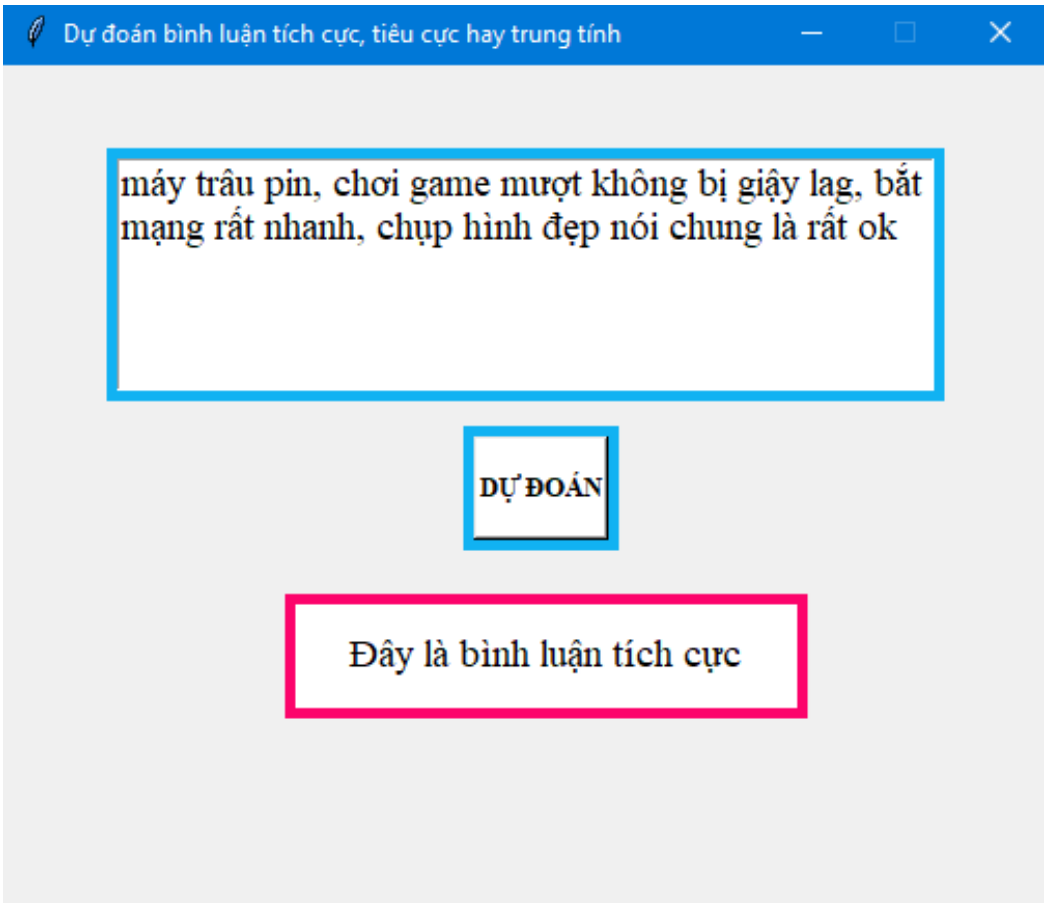
-Kết quả:

Bình luận 5: máy trâu pin, chơi game mượt không bị giậy lag, bắt mạng rất nhanh, chụp hình đẹp nói chung là rất ok

-Sản phẩm: OPPO A5s

-Khách hàng: Lê Hữu Quốc Minh

-Kết quả:


Dự đoán bình luận tích cực, tiêu cực hay trung tính
máy trâu pin, chơi game mượt không bị giậy lag, bắt mạng rất nhanh, chụp hình đẹp nói chung là rất ok
DỰ ĐOÁN
Đây là bình luận tích cực

3 Tổng kết

*** Link chứa toàn bộ Project của nhóm em:**

https://drive.google.com/drive/folders/1ZxcGq8rJJ41-5erTpiQt1NtPBo85mPiy

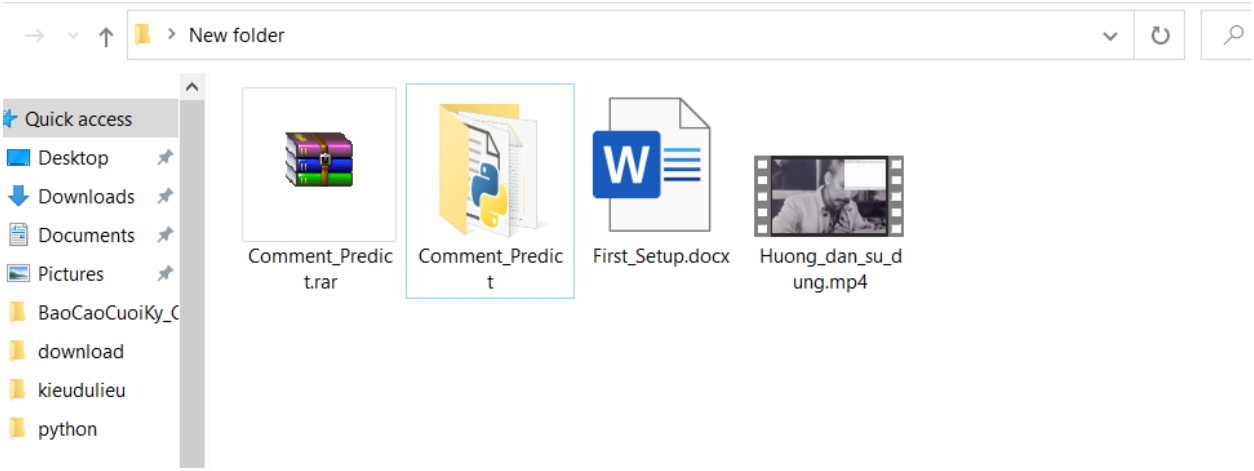
3.1 Những điểm hạn chế

- Model SVC khá nhanh, nhẹ nhàng, dùng tốt trong những trường hợp bình luận thông thường nhưng vẫn còn phân tích sai trong những trường hợp bình luận khó và phức tạp về ý nghĩa.
- Các nhãn trong bộ data có thể gán chưa được chuẩn và việc xuất hiện thêm nhiều từ mới có thể gây nhầm lẫn trong việc phân loại các bình luận.
- Việc chuẩn hóa vẫn chưa xử lí được các trường hợp bình luận lỗi font, sai chính tả, thiếu chữ,… (VD: ah, mu, mụa, giạo hàng, sạt pin, kiêủ,…) gây nhiễu trong quá trình huấn luyện.
- Mặc dù việc dùng TF-IDF để trích xuất vector đặc trưng đã có để ý đến thứ tự sắp xếp các từ trong câu nhưng TF-IDF vẫn chủ yếu là để đánh trọng số những từ quan trọng là chính nên có thể sai trong những trường hợp bình luận kiểu như: “không gây thất vọng” bị xếp vào nhóm tiêu cực.
- Giải pháp:
  Sử dụng word embedding (Word2vec) và LSTM để cải thiện.
  Thu thập thêm data.

3.2 Hướng dẫn sử dụng source code

* Link chứa file *Comment_Predict.rar*:
https://drive.google.com/file/d/1jiV4jrL4f0xjDKhSmAkhSjLg7b-Nq5Ef/view

Bước 1: Tải file Comment_Predict.rar được cung cấp về máy và giải nén.



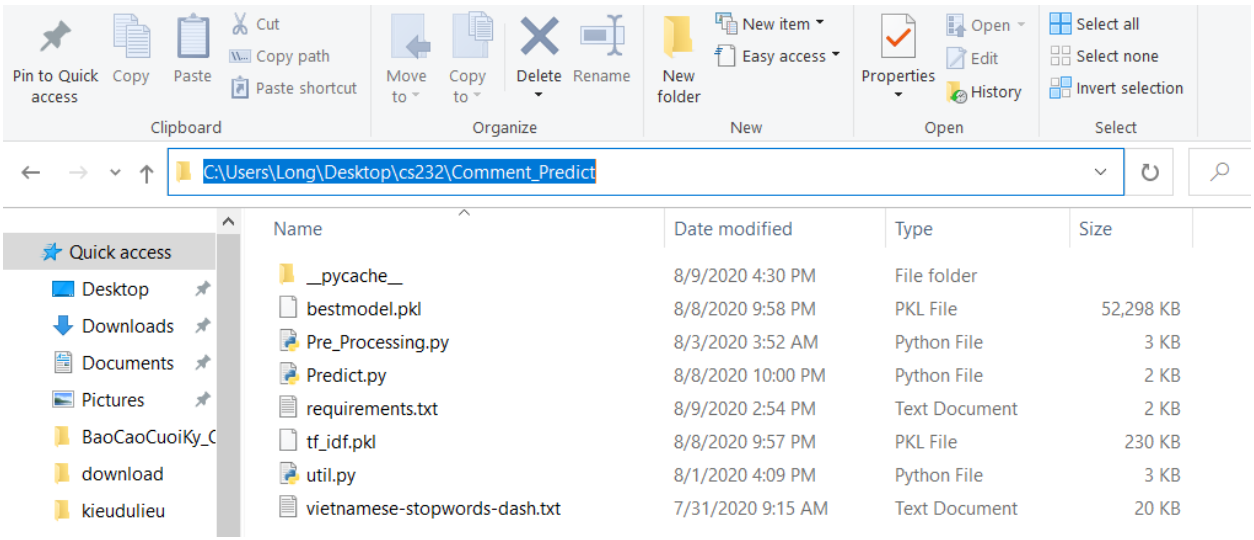
File Comment_Predict: chứa toàn bộ source code của đồ án lần này.

File First_Setup.docx: hướng dẫn install các thư viện cần thiết trước khi chạy chương trình.

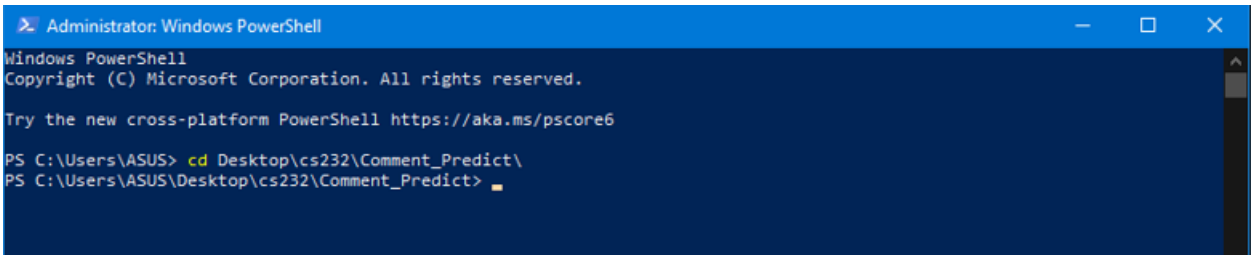
File Huong_dan_su_dung.mp4: Demo hướng dẫn sử dụng.

Bước 2: Để chạy chương trình ta tiến hành bật CMD hoặc PowerShell.

Dùng lệnh *cd* để truy cập đến đường dẫn nơi lưu file thực thi, file thực thi ở đây là Predict.py



Đường dẫn nơi lưu file thực thi (Predict.py)



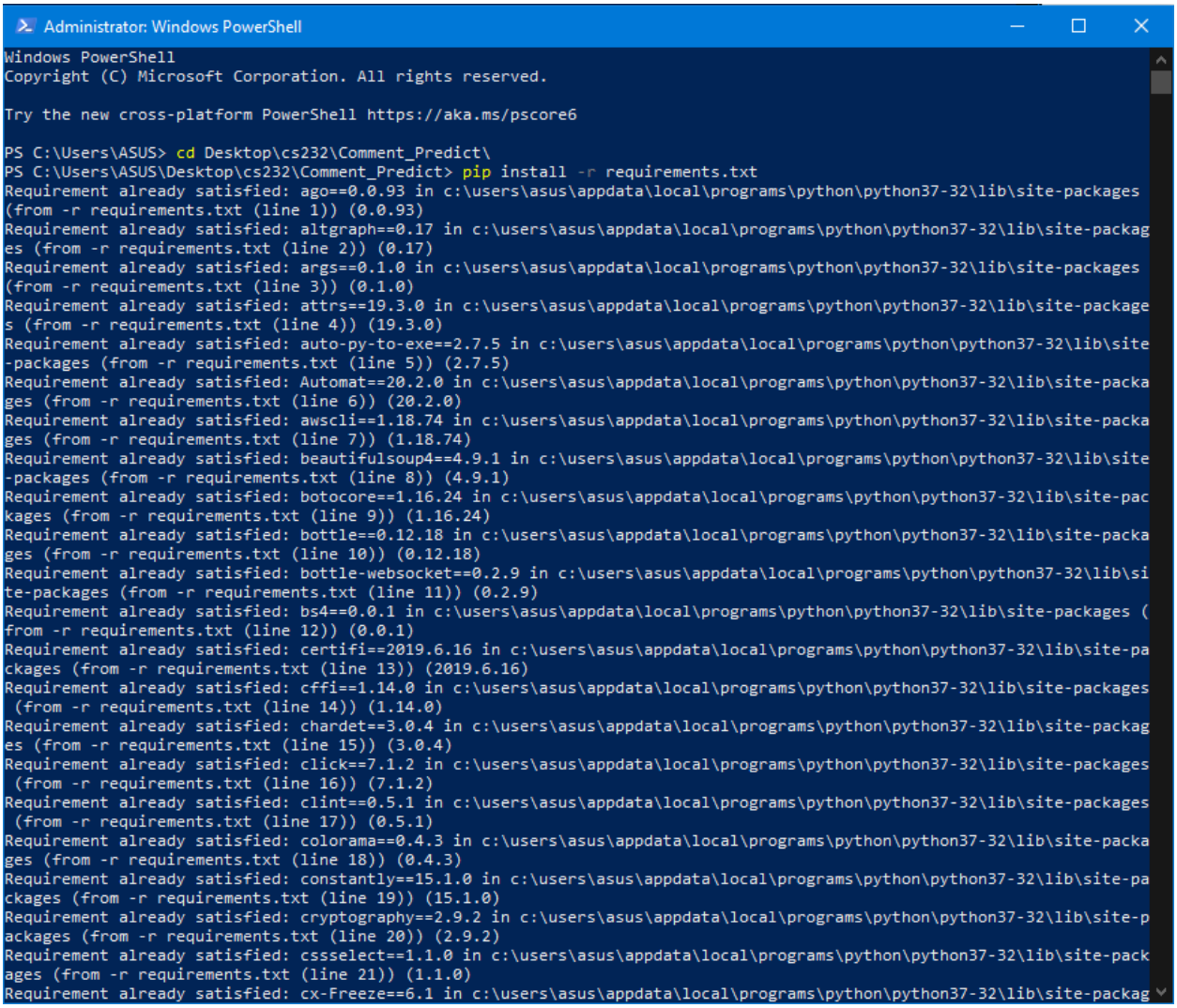
Bước 3: (Nếu không phải lần đầu thực thi bỏ qua bước này)

***Yêu cầu**: Python phiên bản 3.7

Dùng lệnh sau để cài đặt các thư viện cần thiết:

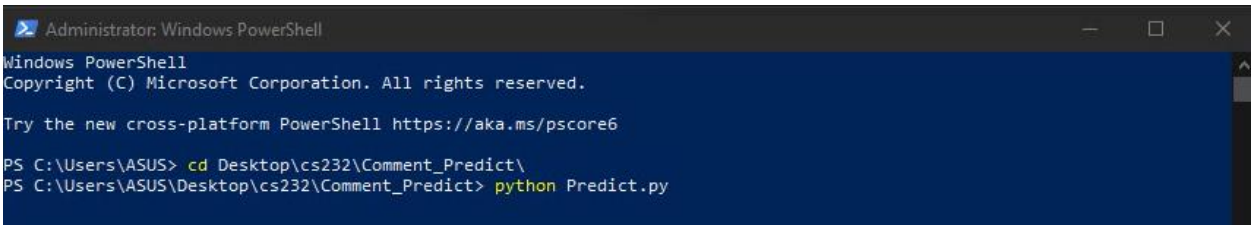
pip install -r requirements.txt

hoặc pip3 install -r requirements.txt



Bước 4: Nhập lệnh *python* <tênfile>.py để chạy chương trình.

python Predict.py



4 Tài liệu tham khảo

1. https://machinelearningcoban.com/2017/08/31/evaluation/#-truefalse-positivenegative
2. https://www.digitalocean.com/community/tutorials/how-to-scrape-web-pages-with-beautiful-soup-and-python-3
3. https://blog.vietnamlab.vn/2019/08/04/xay-dung-1-model-machine-learning-don-gian-de-giai-quyet-bai-toan-phan-loai-sac-thai-binh-luan-trong-tieng-viet/
4. https://codetudau.com/bag-of-words-tf-idf-xu-ly-ngon-ngu-tu-nhien/index.html
5. https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html
6. https://towardsdatascience.com/how-sklearns-tf-idf-is-different-from-the-standard-tf-idf-275fa582e73d
7. https://machinelearningmastery.com/precision-recall-and-f-measure-for-imbalanced-classification/?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

5 Bảng phân công

Mã số sinh viên	Họ và tên	Công việc được giao
18521060	Trịnh Hưng Long	Thực hiện crawl data từ trang web, tinh chỉnh lại các nhãn và tổng hợp bộ data hoàn chỉnh, chỉnh sửa file báo cáo và slide thuyết trình.
18521062	Hà Văn Luân	Thực hiện các bước tiền xử lí data, tinh chỉnh lại các nhãn thích hợp cho bộ dataset bằng tay, thu thập các từ cần chuẩn hóa trong bộ dataset.
18521274	Lỡ Đình Phương	Thực hiện đánh giá, training model, thiết kế giao diện ứng dụng và tinh chỉnh lại các nhãn thích hợp cho bộ dataset bằng tay.