ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA: CÔNG NGHỆ THÔNG TIN

-----2022-----

BÁO CÁO

Project 2: Sentiment analysis

LỚP : **Ứng dụng dữ liệu lớn - 19_21**

SINH VIÊN THỰC HIỆN:

1712778 - Thống A Thảo

19120151 - Nguyễn Trí Tuệ

19120543 - Hoàng Mạnh Khiêm

19120683 - Thái Trung Tín

Bảng đóng góp của thành viên nhóm:

Công việc	Người nhận	Phần trăm hoàn thành (%)
Yêu cầu 1, báo cáo	19120151 - Nguyễn Trí Tuệ	100
Yêu cầu 2, báo cáo	19120543 - Hoàng Mạnh Khiêm	100
Yêu cầu 3, báo cáo	19120683 - Thái Trung Tín	100
Yêu cầu 4, báo cáo	1712778 - Thống A Thảo	100

1. Project 2: Sentiment Analysis

Project được nhóm chia làm 2 file .iypnb:

- Project_2_Sentiment_analysis_Collect_Preprocess_data.ipynb: Thu thập và tiền xử lý dữ liệu. https://colab.research.google.com/drive/1ntwZCwQkT1ndjnstRWmzdG1dxI5myGWi?usp=sharing
- Project_2_Sentiment_analysis_Analysis_Evaluate_results.ipynb: Biểu diễn dữ liệu bằng các bộ embedding và sử dụng mô hình để phân tích, đánh giá. https://colab.research.google.com/drive/1E0i6c5LZ1cUHIbKRZdPVOzXeAgRsmZvY?usp=sharing

1.1. Thu thập dữ liệu

Dữ liệu được thu thập từ đánh giá về ứng dụng Facebook trên Google Play[1]. Nhóm đã tiến hành thu thập 20000 đánh giá để phục vụ đồ án. Do đã có sẵn thư viện `google-play-scraper` nên nhóm đã sử dụng luôn thư viện hỗ trợ này để thu thập đánh từ Google Play.

Truyền vào id của trang web (...id=**com.facebook.katana**&hl=en&gl=US) thiết lập các tham số về ngôn ngữ, khu vực, số lượng đánh giá muốn thu thập rồi khởi chạy hàm để thu thập đánh giá.

[1] https://play.google.com/store/apps/details?id=com.facebook.katana&hl=en&gl=US

```
[ ]  review_list, continuation_token = reviews(
         'com.facebook.katana',
         lang= 'en',
         country= 'us',
         sort= Sort.MOST_RELEVANT,
         count= 20000,
     )

     review_list, _ = reviews(
         'com.facebook.katana',
         continuation_token= continuation_token
     )
```

Lưu dữ liệu vào DataFrame để chuẩn bị cho phần tiền xử lý.

```
[ ]  review_content = []
     for review in review_list:
         review_content.append(review['content'])

     raw_df = pd.DataFrame()
     raw_df['Content'] = np.array(review_content)
     raw_df.head()
```

Dữ liệu đánh giá thu thập được như hình bên dưới.

	Content
0	News feed r so annoying...not responding properly...
1	Where do I begin? I don't know why you include a data saver option in the app while it is worthless and never saves any data. Still videos load on the highest quality and to reduce it i have to click twice on a lower quality. Not kidding twice or the the video will load anyway. I wonder if you are intentinally not fixing this to help network providers in my country charge us for more. It is 2021 and internet should be free for all but in my country we are still struggling for more quota!
2	i think u should add the options for folders for the featured photos. I mean, when I want to add photos in my featured photos from my gallery, it's hard to pick from the old up to present pics that I want to add in the featured photos. Like, im gonna wait for minutes just to scroll the other photo. What i wanted to say is u should add the options like "gallery" "camera" "album 1" or something like that for us to access easily the photos that we only wanted to display. Thank u!!
3	horrible! can't scroll on pages
4	The mobile app is all but dead after the Android 12 update. Nothing updates ; newsfeed, notifications, even messenger. I have completely uninstalled both, cleared cache and reinstalled. Woke up 7 hours later with the same entry at the top of my feed, no notifications from any of my groups and it won't refresh on command. Note 20 Ultra, unlocked, on VZW.

1.2. Tiền xử lý dữ liệu và gán nhãn cho dữ liệu

1.1.1. Tách câu

Dữ liệu sau khi được thu thập sẽ gồm nhiều câu được ngăn cách với nhau. Do đó ta sẽ tiến hành tách đánh giá thành các câu đơn. Sử dụng thư viện `sent_tokenize` từ `nltk` để hỗ trợ tách đánh giá thành các câu đơn.

```
[ ]  raw_df['Content'] = raw_df['Content'].apply(lambda x: sent_tokenize(x))
```

```
[ ]  reviews_df = raw_df.apply(lambda x: x.explode()).reset_index().iloc[:, 1:]
     reviews_df.head()
```

	Content
0	News feed r so annoying...not responding properly...
1	Where do I begin?
2	I don't know why you include a data saver option in the app while it is worthless and never saves any data.
3	Still videos load on the highest quality and to reduce it i have to click twice on a lower quality.
4	Not kidding twice or the the video will load anyway.

1.1.2. In thường toàn bộ

Chuyển toàn bộ đánh giá về chung một format để giúp việc biểu diễn dữ liệu hiệu quả hơn.

```
[ ]  reviews_df['Content'] = reviews_df['Content'].str.lower().astype(str)
     reviews_df.head()
```

	Content
0	news feed r so annoying...not responding properly...
1	where do i begin?
2	i don't know why you include a data saver option in the app while it is worthless and never saves any data.
3	still videos load on the highest quality and to reduce it i have to click twice on a lower quality.
4	not kidding twice or the the video will load anyway.

1.1.3. Thay thế các từ viết tắt

Thay thế một số từ viết tắt thường thấy và bên cạnh đó là một số từ viết tắt thấy được trong dữ liệu.

```
[ ]  abbreviations = {
          "i\'m": 'i am',
          "im": 'i am',
          "don\'t": 'do not',
          "\'t": ' not',
          "\'s": ' is',
          "\'ll": ' will',
          "\'ve": ' have',
          "ive": 'i have',
          "\'d": ' would',
          "\'re": ' are',
          "cant": 'can not',
          "dont": 'do not',
          "idk": 'i do not know',
          "cannot": 'can not',
          "fb": 'facebook',
          "pls": 'please',
          "smh": 'shaking my head',
          "btw": 'by the way',
          "cuz": 'because'
      }
```

```
[ ]  reviews_df['Content'] = reviews_df['Content'].replace(to_replace= abbreviations.keys(), value= abbreviations.values(), regex= True)
     reviews_df.head()
```

	Content
0	news feed r so annoying...not responding properly...
1	where do i begin?
2	i do not know why you include a data saver option in the app while it is worthless and never saves any data.
3	still videos load on the highest quality and to reduce it i have to click twice on a lower quality.
4	not kidding twice or the the video will load anyway.

1.1.4. Loại bỏ các chữ số

Các chữ số đứng riêng lẻ không có nhiều ý nghĩa trong việc nhận định đánh giá sẽ được loại bỏ.

```
[ ]  def remove_digit(text):
         result = re.sub(r'\d+', '', text)
         return ' '.join(result.split())
```

```
[ ]  reviews_df['Content'] = reviews_df['Content'].apply(lambda text: remove_digit(text))
     reviews_df.head()
```

	Content
0	news feed r so annoying...not responding properly...
1	where do i begin?
2	i do not know why you include a data saver option in the app while it is worthless and never saves any data.
3	still videos load on the highest quality and to reduce it i have to click twice on a lower quality.
4	not kidding twice or the the video will load anyway.

1.1.5. Loại bỏ các dấu câu

Loại bỏ các punctuation ra khỏi dữ liệu để hỗ trợ biểu diễn dữ liệu. Tránh trường hợp coi 2 từ 'Help' và 'Help!' là khác nhau.

```
[ ]  reviews_df['Content'] = reviews_df['Content'].apply(lambda text: re.sub(r'\W+', ' ', text))
     reviews_df.head()
```

	Content
0	news feed r so annoying not responding properly
1	where do i begin
2	i do not know why you include a data saver option in the app while it is worthless and never saves any data
3	still videos load on the highest quality and to reduce it i have to click twice on a lower quality
4	not kidding twice or the the video will load anyway

1.1.6. Loại bỏ các emoji

Trong các đánh giá và bình luận luôn thường thấy các emoji. Các emoji này có thể có nghĩa hoặc vô nghĩa từ đó có 2 cái xử lý đó là thay thế emoji thành văn bản hoặc xóa bỏ hoàn toàn emoji. Để cho đơn giản các emoji sẽ được loại bỏ ra khỏi dữ liệu.

Xây dựng các pattern biểu diễn emoji.

```
[ ]  def remove_emoji(string):
         emoji = re.compile("["
                            u"\U0001F600-\U0001F64F"  # emoticons
                            u"\U0001F300-\U0001F5FF"  # symbols & pictographs
                            u"\U0001F680-\U0001F6FF"  # transport & map symbols
                            u"\U0001F1E0-\U0001F1FF"  # flags (iOS)
                            u"\U00002500-\U00002BEF"  # chinese char
                            u"\U00002702-\U000027B0"
                            u"\U00002702-\U000027B0"
                            u"\U000024C2-\U0001F251"
                            u"\U0001f926-\U0001f937"
                            u"\U00010000-\U0010ffff"
                            u"\u2640-\u2642"
                            u"\u2600-\u2B55"
                            u"\u200d"
                            u"\u23cf"
                            u"\u23e9"
                            u"\u231a"
                            u"\ufe0f"  # dingbats
                            u"\u3030"
                            "]+", flags=re.UNICODE)
         return emoji.sub(r'', string)
```

Loại bỏ các emoji có pattern tương tự trong dữ liệu.

```
[ ]  reviews_df['Content'] = reviews_df['Content'].apply(lambda text: remove_emoji(text))
     reviews_df.head()
```

	Content
0	news feed r so annoying not responding properly
1	where do i begin
2	i do not know why you include a data saver option in the app while it is worthless and never saves any data
3	still videos load on the highest quality and to reduce it i have to click twice on a lower quality
4	not kidding twice or the the video will load anyway

1.1.7. Loại bỏ Stopwords

Các Stopwords như *i, me, my, myself, out*, v.v. không có nhiều ý nghĩa trong việc nhận định đánh giá. Nên sẽ loại bỏ các từ này đi.

```
[ ] print(stopwords.words('english'))

['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', "you're", "you've", "you'll", "you'd", 'your', 'yours', 'yourself', 'yourselves', 'h

[ ] reviews_df['Content'] = reviews_df['Content'].apply(lambda text: " ".join([word for word in str(text).split() if word not in set(stopwords.words('englis
    reviews_df.head()
```

	Content
0	news feed r annoying responding properly
1	begin
2	know include data saver option app worthless never saves data
3	still videos load highest quality reduce click twice lower quality
4	kidding twice video load anyway

1.1.8. Lemmatization

Trong tiếng anh một từ có thể có nhiều dạng từ loại khác nhau ví dụ như *walks*, *walked*, *walking* đều là biến thể của từ *walk*. Do đó để việc biểu diễn dữ liệu liệu hiệu quả hơn ta sẽ tiến hành đưa các dạng từ loại về nguyên mẫu của chúng.

Có 2 kỹ thuật để thực hiện nhiệm vụ trên là Steamming và Lemmatization. Steamming có tốc độ cao như do chỉ đơn thuần là loại bỏ hậu tố nên sẽ gặp khó khăn với từ bất quy tắc. Do đó nhóm sử dụng kỹ thuật Lemmatization, đây là kỹ thuật sẽ thực hiện việc tra cứu từ trong cơ sở dữ liệu để xác định nguyên mẫu của từ đó.

Sử dụng thư viện `WordNetLemmatizer` từ `nltk.stem` thực hiện

```
[ ] lemmatizer = WordNetLemmatizer()
    wordnet_map = {"N": wordnet.NOUN, "V": wordnet.VERB, "J": wordnet.ADJ, "R": wordnet.ADV}
    def lemmatize_words(text):
        pos_tagged_text = nltk.pos_tag(text.split())
        return " ".join([lemmatizer.lemmatize(word, wordnet_map.get(pos[0], wordnet.NOUN)) for word, pos in pos_tagged_text])

[ ] reviews_df['Content'] = reviews_df['Content'].apply(lambda text: lemmatize_words(text))
    reviews_df.head()
```

	Content
0	news feed r annoy respond properly
1	begin
2	know include data saver option app worthless never save data
3	still videos load high quality reduce click twice low quality
4	kid twice video load anyway

1.1.9. Sửa các lỗi chính tả

Việc đánh giá hay bình luận có lỗi chính tả là điều hoàn toàn bình thường. Để khắc phục vấn đề này ta sẽ tiến hành sửa các lỗi chính tả nhờ thư viện `SpellChecker`.

```
[ ]  spell = SpellChecker()
     def correct_spellings(text):
         corrected_text = []
         misspelled_words = spell.unknown(text.split())
         for word in text.split():
             if word in misspelled_words:
                 correct_word = spell.correction(word)
                 if correct_word != None:
                     corrected_text.append(correct_word)
             else:
                 corrected_text.append(word)
         return " ".join(corrected_text)
```

```
[ ]  reviews_df['Content'] = reviews_df['Content'].apply(lambda text: correct_spellings(text))
     reviews_df.head()
```

	Content
0	news feed i annoy respond properly
1	begin
2	know include data saver option app worthless never save data
3	still videos load high quality reduce click twice low quality
4	kid twice video load anyway

1.1.10. Loại bỏ từ đơn vô nghĩa

Các từ đơn đứng một mình chỉ có 1 ký tự có thể có sẵn trong bình luận hoặc xuất hiện sau các bước tiền xử lý bên trên không có nghĩa nên sẽ được loại bỏ.

```
[ ]  def remove_single_word(text):
         # Loại bỏ từ đơn ở giữa câu
         result = re.sub(r'\s+[a-zA-Z]\s+', ' ', text)
         result = ' '.join(result.split())
         while True:
             result = re.sub(r'\s+[a-zA-Z]\s+', ' ', result)
             result = ' '.join(result.split())
             if len(re.findall(r'\s+[a-zA-Z]\s+', result)) ==  0: break;


         # Loại bỏ từ đơn ở đầu câu
         result = re.sub(r'^[a-zA-Z]\s+', '', result)

         # Loại bỏ từ đơn ở cuối câu
         result = re.sub(r'\s+[a-zA-Z]$', '', result)

         return result
```

```
[ ]  reviews_df['Content'] = reviews_df['Content'].apply(lambda text: remove_single_word(text))
     reviews_df.head()
```

1.1.11. Loại bỏ ô dữ liệu trống

Sau các bước tiền xử lý bên trên có thể sẽ xuất hiện các dòng dữ liệu rỗng. Do đó, ta sẽ tiến hành loại bỏ các dòng này đi.

Trước khi loại bỏ các dòng trống.

```
[ ]  reviews_df.info()

     <class 'pandas.core.frame.DataFrame'>
     RangeIndex: 72102 entries, 0 to 72101
     Data columns (total 1 columns):
      #   Column   Non-Null Count  Dtype
     ---  ------   --------------  -----
      0   Content  72102 non-null  object
     dtypes: object(1)
     memory usage: 563.4+ KB
```

```
[ ]  reviews_df['Content'] = reviews_df['Content'].replace('', np.nan, regex= True)
     reviews_df.isna().sum()

     Content    1672
     dtype: int64
```

Sau khi loại bỏ các dòng trống.

```
[ ]  reviews_df = reviews_df.dropna().reset_index().iloc[:, 1:]
     reviews_df.isna().sum()

     Content    0
     dtype: int64

[ ]  reviews_df.info()

     <class 'pandas.core.frame.DataFrame'>
     RangeIndex: 70430 entries, 0 to 70429
     Data columns (total 1 columns):
      #   Column   Non-Null Count  Dtype
     ---  ------   --------------  -----
      0   Content  70430 non-null  object
     dtypes: object(1)
     memory usage: 550.4+ KB
```

1.3. Tạo embedding biểu diễn dữ liệu

1.3.1. TF-IDF

```
#TF-IDF
from sklearn.feature_extraction.text import TfidfVectorizer

vectorizer = TfidfVectorizer(use_idf=True)
train_vector = vectorizer.fit_transform(trainset.Content)
test_vector = vectorizer.transform(testset.Content)
```

1.3.2. fastText

Cài đặt fasttext

```
!wget https://github.com/facebookresearch/fastText/archive/v0.9.2.zip
!unzip v0.9.2.zip
%cd fastText-0.9.2
!make
!pip install .
```

Download pretrained model của fasttext

```
!./download_model.py en

Downloading https://dl.fbaipublicfiles.com/fasttext/vectors-crawl/cc.en.300.bin.gz
 (100.00%) [==================================================>]
```

Vector embedding cho cac reivews

```
ftEmbeddingVectors=[]
for i in range(len(trainset)):
  ftEmbeddingVectors.append(model.get_sentence_vector(trainset.iloc[i,0]))
Test_ftEmbeddingVectors=[]
for i in range(len(testset)):
  Test_ftEmbeddingVectors.append(model.get_sentence_vector(testset.iloc[i,0]))
```

1.3.3. BERT

Install mô hình lấy vector câu của transformers

```
!pip install -U sentence-transformers
```

Dùng mô hình bert base nli mean tokens tạo vector embedding

```
from sentence_transformers import SentenceTransformer
embeddings_train=[]
embeddings_test=[]
train_sentences=list(trainset.Content)
test_sentences=list(testset.Content)
model = SentenceTransformer('sentence-transformers/bert-base-nli-mean-tokens')
```

```
for i in range(len(test_sentences)):
  vect = model.encode(test_sentences[i])
  embeddings_test.append(vect)
  print(i)
```

```
for i in range(len(train_sentences)):
  vect = model.encode(train_sentences[i])
  embeddings_train.append(vect)
  print(i)
```

1.4. Xây dựng mô hình và đánh giá kết quả

Sau khi ta có các word embedding, ta tiếp hành phân lớp. Mô hình nhóm chọn thực hiện việc này là mô hình SVM.

Nhìn chung các bộ word embedding đều mang lại kết quả rất tốt. Dựa vào kết quả thu được TF-IDF đạt được độ chính xác tốt nhất. Dưới đây kết quả thu được của các bộ word embedding.

1.4.1. TF-IDF

```
accuracy =  0.9213888888888889
For Negative: precision = 0.9345203905801264
For Positive: precision = 0.9090909090909091
For Negative: recall = 0.905902004454343
For Positive: recall = 0.9368070953436807
For Negative: f1_score = 0.9199886909810574
For Positive: f1_score = 0.9227409227409228
```

1.4.2. fastText

```
accuracy =  0.8866666666666667
For Negative: precision = 0.8938706015891033
For Positive: precision = 0.8797606093579978
For Negative: recall = 0.8769487750556793
For Positive: recall = 0.8963414634146342
For Negative: f1_score = 0.8853288364249579
For Positive: f1_score = 0.8879736408566722
```

1.4.3. BERT

```
accuracy =  0.8866666666666667
For Negative: precision = 0.8522848034006376
For Positive: precision = 0.8882421420256111
For Negative: recall = 0.8930957683741648
For Positive: recall = 0.8458980044345898
For Negative: f1_score = 0.8722131593257204
For Positive: f1_score = 0.8665530948324817
```

2. Tham khảo

- https://www.kaggle.com/code/sudalairajkumar/getting-started-with-text-preprocessing
- https://pypi.org/project/google-play-scraper/
- https://github.com/JoMingyu/google-play-scraper
- https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html
- https://jalammar.github.io/a-visual-guide-to-using-bert-for-the-first-time/

- https://fasttext.cc/docs/en/crawl-vectors.html
- https://huggingface.co/sentence-transformers/bert-base-nli-mean-tokens