TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN, ĐHQG-HCM
KHOA CÔNG NGHỆ THÔNG TIN

ỨNG DỤNG DỮ LIỆU LỚN

Báo cáo Đồ án 2: Khai thác ý kiến

Lớp 19_21

Giảng viên Nguyễn Ngọc Thảo

~

19120615 Hùng Ngọc Phát

19120621 Lê Minh Phục

19120633 Nguyễn Anh Quốc

19120689 Lại Khánh Toàn

TP.HCM, Tháng 09/2022

Mục lục

Phân công

MSSV	Tên	Công việc
19120615	Hùng Ngọc Phát	EDA, tiền xử lý dữ liệu, gán nhãn sơ bộ
19120621	Lê Minh Phục	Vectorize và xây dựng mô hình
19120633	Nguyễn Anh Quốc	Thu thập dữ liệu và tiền xử lý dữ liệu
19120689	Lại Khánh Toàn	

1 Về các tệp tin mã nguồn

Thư mục **source** chứa các tệp tin mã nguồn mà nhóm đã viết, bao gồm các tệp tin sau:

- **1-crawl-gplay.ipynb**: crawl data từ Google Play.
- **1-crawl-facebook.ipynb**: crawl data từ Facebook.
- **2-preprocess-eda.ipynb**: thực hiện preprocess, EDA và label sơ bộ dữ liệu.
  Dữ liệu sau đó sẽ được nhóm đánh nhãn thủ công.
- **3-modeling.ipynb**: vector hoá bằng TF-IDF, fasttext, PhoBERT và xây dựng mô hình phân lớp.

Ngoài ra trong thư mục **source** cũng chứa 2 thư mục con là **data** (chứa **raw** data đã được crawl về, và **clean** data có được sau khi chạy file *2-preprocess-eda.ipynb*) và **dict** (chứa các file từ điển, teencode, stopword, … để preprocess dữ liệu text).

Để chạy các tập tin đánh số thứ tự 1 và 2, thầy cô vui lòng cài các thư viện sau đây:

- scikit-learn
- google-play-scraper
- pandas
- numpy
- matplotlib
- seaborn

Đối với file đánh số thứ tự 3, nhóm khuyến khích chạy trên môi trường Colab hoặc môi trường có thể sử dụng được CUDA vì inference bằng PhoBERT tốn rất nhiều thời gian nếu chạy trên CPU thông thường.

Để chạy file **3-modeling.ipynb**, thầy cô vui lòng:

- Mở notebook bằng Colab
- Upload file "*data/labeled_data.csv*" vào *"/content*" trên Colab.
  Nghĩa là ta sẽ có file *"/content/labeled_data.csv"*.
- Chạy tất cả các cell như bình thường.

2 Thu thập và tiền xử lý dữ liệu

2.1 Thu thập

Dữ liệu của nhóm được thu thập từ 2 nguồn.

Nguồn thứ nhất là từ các comment của trò chơi Subway Surfer trên Google Play. Nhóm đã sử dụng thư viện hỗ trợ google-play-scraper để quá trình thu thập diễn ra nhanh hơn. Dữ liệu được crawl là các *comment tiếng Việt* đánh giá 1 sao và 5 sao của trò chơi trên.

Nguồn thứ 2 là các bài viết review cửa hàng CellphoneS và MobileCity trên Facebook.

Tổng số mẫu dữ liệu là 6466.

2.2 Tiền xử lý

Từ dữ liệu crawl về, đầu tiên nhóm sẽ tiến hành tiền xử lý cơ bản:

- Bỏ các ký tự đặc biệt (dấu câu, emoji, biểu tượng, chữ tiếng Trung, ...)
- Bỏ HTML
- Bỏ URL
- Bỏ các số
- Chuẩn hoá unicode NFC

Sau đó nhóm sẽ tiến hành tiền xử lý nâng cao hơn, kết hợp với EDA

1. Tokenize theo âm tiết. Vì dữ liệu comment rất "bẩn" nên các công cụ word-based tokenize phổ biến cho tiếng Việt không thể hoạt động chính xác được.
2. Lấy ra 100 token phổ biến nhất trong các comment mà vừa không có trong tự điển tiếng Việt và từ điển 3000 từ phổ biến nhất trong tiếng Anh (mục đích để không bỏ sót các từ phổ thông như amazing, good, bad, …). Những từ không có trong từ điển thường là teencode.
3. Tạo rule replace thủ công cho 100 token trên, sau đó replace vào dữ liệu comment.
   Rule replace thủ công được nhóm viết ở file *dict/replace_rules.csv*.
   Ngoài ra nhóm cũng kết hợp với một bộ dataset teencode có sẵn trên mạng ở trang xltiengviet là file *dict/teen_code.tsv*.
   Các từ teencode khiếm nhã (từ 2 dataset trên) sẽ được nhóm gán lại thành nhãn *PROFANE*. Các từ khiếm nhã không nằm trong danh sách ở trên sẽ được giữ nguyên (thường mang tính tiêu cực, có thể giúp mô hình dễ dàng nhận diện đó là câu tiêu cực, hơn nữa nhóm không muốn ghi các từ đó vào trong bài nộp).
4. Sau khi xử lý teencode, nhóm sẽ bỏ đi các từ còn lại trong câu mà không nằm trong từ điển.
5. Cuối cùng, các câu có dưới 3 từ; hoặc có 50% số từ trong câu không phải là tiếng Việt sẽ được bỏ đi.

2.3 Gán nhãn nhanh

Cuối cùng, nhóm gán nhãn nhanh bằng thư viện **underthesea** của nhóm tác giả *undertheseanlp*.

Nhóm chạy mô hình SVM đã được train sẵn trên 2 dataset one_star và five_star. Với mỗi dataset, nhóm tiến hành đánh giá sơ bộ "tính không chắc chắn" của mô hình trên.

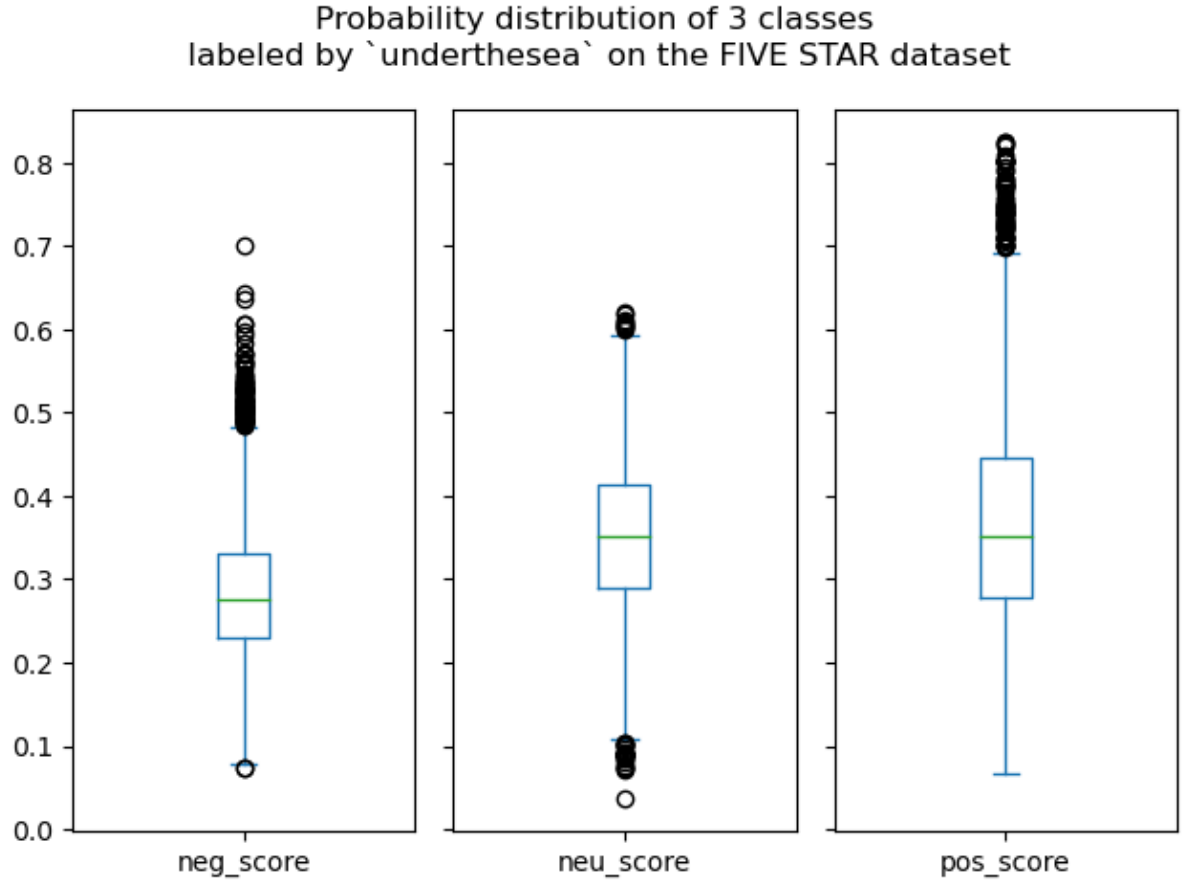
Mô hình này cho ra 3 class là negative, neutral và positive.

:

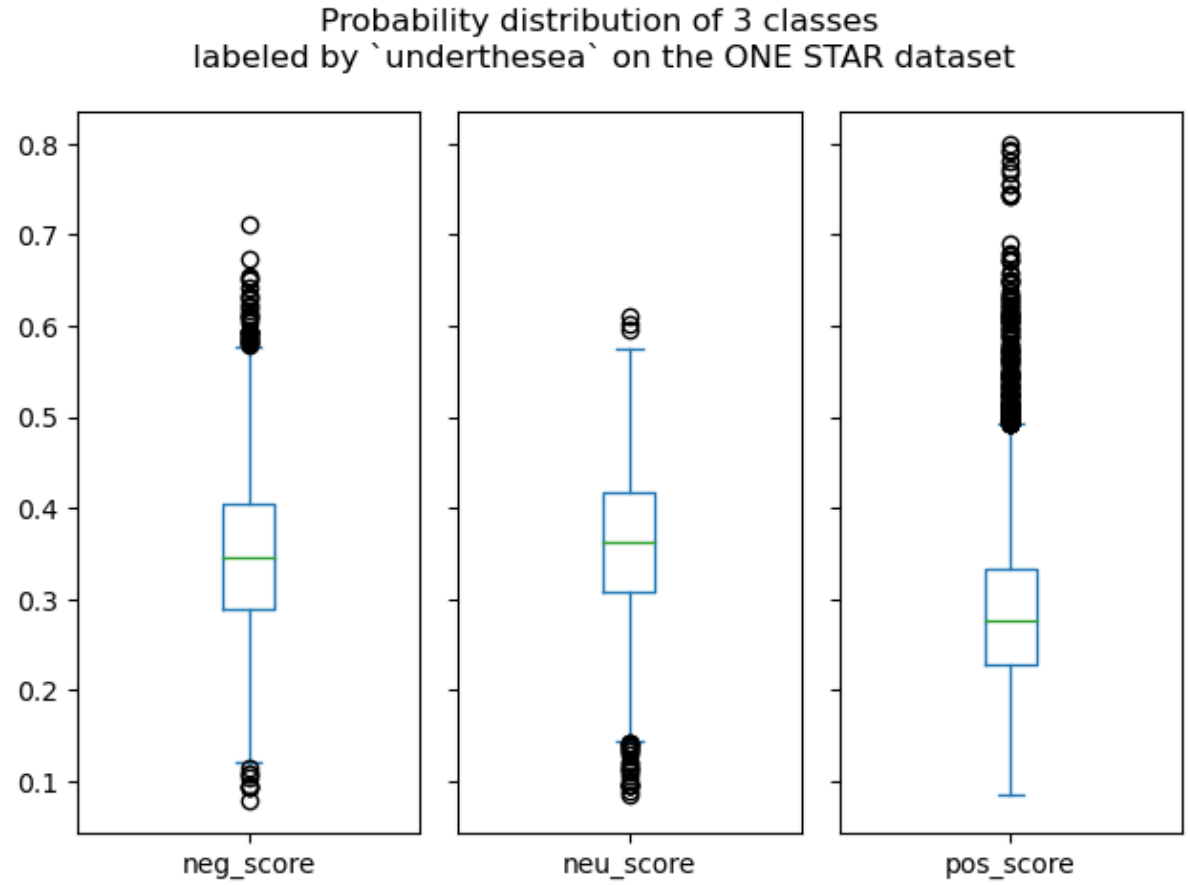
	tokens	neg_score	neu_score	pos_score
1070	game chơi vui cho sao	0.302839	0.466378	0.230783
3291	toi thay no rat hay it quang cao	0.260370	0.346050	0.393579
154	vô trang vay vốn nhận k free	0.320514	0.370013	0.309473
362	đang chơi thoát ra có tí mà đến km rồi mà vô l...	0.279092	0.422214	0.298694
2516	trò chơi này rất vui	0.272976	0.291544	0.435480

Dữ liệu mẫu sau khi label sơ bộ trong dataset one_star

Với mỗi dataset, nhóm vẽ boxplot của 3 cột neg_score, neu_score và pos_score.



Với tập các comment 5 sao, ta thấy "confidence" của lớp negative không cao. Vậy ta sẽ giữ lại các comment 5* bằng cách lấy ra tập con của tập trên sao cho *pos_score > 0.2* và *neg_score < 0.3*. 2 con số trên đã được điều chỉnh sao cho data sau khi filter có ít nhất 3000 comment.



Tập one_star "nhiễu" hơn đáng kể so với tập five_star khi confidence của lớp negative không vượt quá mức 0.6, trong khi khoảng giá trị của pos_score lại khá rộng.

Nhóm sẽ chọn ra các comment có *pos_score < 0.5* và *neg_score > 0.34*.

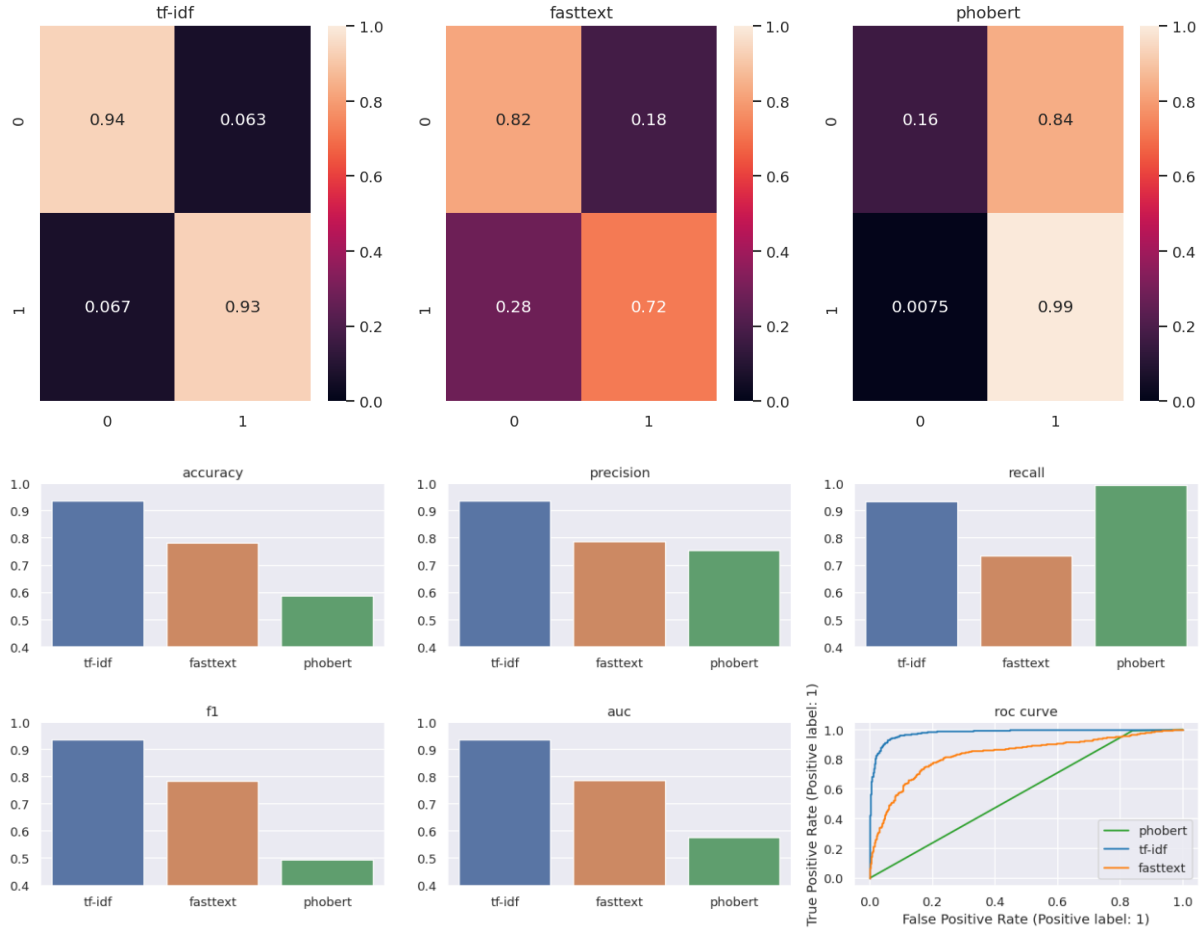
Cuối cùng, nhóm lưu data đã được đánh nhãn sơ bộ xuống đĩa, và cả nhóm sẽ phối hợp kiểm tra nhãn của các câu đã được gán.

3 Huấn luyện mô hình phân lớp

Phương pháp embedding	**Train embedding trên data riêng?**	**Mô hình phân lớp**
TF-IDF	Có	SVM (C=1)
fasttext	Có	SVM (C=1)
PhoBERT	Không finetune	Mô hình phân lớp tích hợp sẵn

Đối với TF-IDF và fasttext, nhóm train lại 2 mô hình embedding này trên data mà nhóm đã crawl ở trên, đồng thời thực hiện phân lớp bằng mô hình SVM.

Với PhoBERT, nhóm sử dụng embedding sẵn có và mô hình phân lớp cũng sẵn có, không finetuning lại trên dataset riêng.



Phương pháp embedding	**Nhận xét**
TF-IDF	Đơn giản và hiệu quả trên dataset của nhóm đã crawl. Các metrics đều dẫn đầu. TPR và TNR rất cao.
fasttext	Hiệu quả không tốt như TF-IDF trên dataset của nhóm. TPR và TNR thấp hơn TF-IDF, FPR và FNR cao.
PhoBERT	Không được finetune nên kết quả rất kém. ROC curve gần như ngang với random classifier. Có lẽ vì mô hình này được train trên ngữ liệu tiếng Việt "chuẩn" chứ không phải dữ liệu lộn xộn được lấy từ mạng xã hội (distribution shift) nên hiệu năng kém. Điểm bất thường là mô hình cho ra TPR cao nhưng TNR rất thấp.

HẾT