PHÂN LOẠI THƯ ĐIỆN TỬ

Báo cáo thực hành lab 01

Nguyễn Minh Vũ Phùng Hoài Thi

Đại học Khoa học Tự nhiên, ĐHQG-HCM

Trí tuệ nhân tạo cho an ninh thông tin
Ngày 18 tháng 10 năm 2024

Tổng quan

Đánh giá công việc

Yêu cầu	Phụ trách	Mức độ hoàn thành
Đọc dữ liệu	Vũ	100%
Tiền xử lý dữ liệu	Vũ	100%
Mô hình	Vũ	100%
Cài đặt chức năng 1	Thi	100%
Cài đặt chức năng 2	Thi	100%
Viết báo cáo	Vũ, Thi	100%
Làm slide	Vũ, Thi	100%

Hình 1: Bảng đánh giá công việc

Bộ dữ liệu Enron-Spam

Bộ dữ liệu Enron-Spam được chia thành 2 file csv chính gồm:

- File **train.csv** chứa 27.284 mails dùng để huấn luyện.
- File **val.csv** chứa 3.084 mails dùng để kiểm thử.

Đọc dữ liệu trong Python

Nhóm sử dụng thư viện Pandas để đọc dữ liệu từ file định dạng bảng như csv. Nếu lượng dữ liệu trở nên lớn hơn (khoảng hàng triệu) thì pandas sẽ không xử lý được. Khi đó, ta sẽ dùng thư viện Polars để thay thế cho Pandas.

Nội dung dữ liệu

Yêu cầu dữ liệu đầu vào phải gồm có 2 đặc trưng quan trọng: Subject (Tiêu đề) và Message (Nội dung). Các đặc trưng không quan trọng (Message ID, split) sẽ được loại bỏ.

Unnamed: 0	Message ID	Subject	Message	Spam/Ham	split
0	0	christmas ...	NaN	ham	0.038415
1	1	vastar res...	gary , pro...	ham	0.696509
2	2	calpine da...	- calpine ...	ham	0.587792
3	3	re : issue...	fyi - see ...	ham	-0.055438
5	5	mcmullen g...	jackie , s...	ham	-0.419658

Hình 2: Dữ liệu Enron-Spam

Xử lý dữ liệu

Do dữ liệu chuẩn bị đã được tiền xử lý bằng các phương pháp như loại bỏ từ dừng, đưa về từ gốc,... nên nhóm tiến hành các bước sau:

1. Đối với dữ liệu rỗng (Nan/Null) $\rightarrow$ chuyển đổi về chuỗi kí tự rỗng. Không nên loại bỏ vì làm mất dữ liệu giảm hiệu suất mô hình.
2. Đối với dữ liệu có nội dung trùng lặp với nhau $\rightarrow$ Loại bỏ để dữ liệu unique.
3. Gộp nội dung của 2 đặc trưng Subject và Message thành 1 đặc trưng duy nhất, thuận tiện để vec-tơ hoá văn bản.
4. Tạo 1 cột nhãn đầu ra với các giá trị nhị phân thay thế cho nhãn Spam/Ham ban đầu.

Xử lý dữ liệu

Spam/Ham	Text	spam
ham	christmas tree farm ...	0
ham	vastar resources , i...	0
ham	calpine daily gas no...	0

Hình 3: Dữ liệu sau khi xử lý

CountVectorizer

CountVectorizer là một công cụ của thư viện scikit-learn áp dụng phương pháp mô hình túi từ Bag-of-Words (BoW) để vec-tơ hoá văn bản. Trong đó, BoW biểu diễn văn bản bằng cách đếm số lần xuất hiện của từ.

CountVectorizer

Ví dụ, ta có các văn bản sau: [*"Trời hôm nay mưa", "Hôm nay tôi đi học", "Trời mưa nhưng tôi vẫn mặc áo mưa đi học"*] . Khi triển khai *CountVectorizer* ta sẽ nhận được các vec-tơ có dạng như bên dưới:

	áo	đi	hôm	học	mưa	mặc	nay	nhưng	tôi	trời	vẫn
docs[0]	0	0	1	0	1	0	1	0	0	1	0
docs[1]	0	1	1	1	0	0	1	0	1	0	0
docs[2]	1	1	0	1	2	1	0	1	1	1	1

Hình 4: Ví dụ về CountVectorizer

Cài đặt CountVectorizer

Để *CountVectorizer* hoạt động hiệu quả hơn, nhóm đã điều chỉnh tham số $ngram\_range = (1, 2)$ để tăng số lượng đặc trưng thay vì chỉ có từ đơn thì bây giờ sẽ có thêm 2 từ liên tiếp nhau xuất hiện.

```
#Chia tập huấn luyện
x_train, x_val, y_train, y_val = train_data.Text,
val_data.Text, train_data.spam, val_data.spam
#Mô hình CountVectorizer
cv = CountVectorizer(ngram_range=(1,2))
#Biến đổi dữ liệu
x_train_cv = cv.fit_transform(x_train)
x_val_cv = cv.transform(x_val)
```

Hình 5: Cài đặt Vectorizer

Multinominal Naive Bayes

Ta cần vec-tơ hoá dữ liệu để mô hình có thể xử lý và một trong những cách tiếp cận cơ bản là sử dụng ý tưởng của Bag of Words (BoW). Khi này, mỗi văn bản được thể hiện dưới dạng một vec-tơ đặc trưng và với mỗi thành phần thứ *i* trong vec-tơ có độ dài *l* chính là số lượng từ đó có trong từ điển.

Multinominal Naive Bayes

Nhóm sử dụng mô hình *Multinominal Naive Bayes* với $\alpha = 0.1$ vì *alpha* nhỏ sẽ giảm ảnh hưởng của các từ hiếm mà không làm mất thông tin. Bên cạnh đó, các từ phổ biến sẽ chiếm ưu thế hơn trong việc dự đoán.

Pipeline

Nhóm kết hợp 2 mô hình trên thành một pipeline hoàn chỉnh và huấn luyện trên tập dữ liệu đã xử lý ở hình 3:

```
pipeline = Pipeline([
            ('vect', CountVectorizer()),
            ('mnb', MultinomialNB())
])
grid_search = GridSearchCV(pipeline, parameters)
complete_pipeline =  grid_search.fit(x_train, y_train)
```

Hình 6: Huấn luyện mô hình

Dự đoán

Nhóm dự đoán trên tập val và đạt kết quả $\approx 99.45\%$.

Class	Precision	Recall	F1-score	Support
0	0.9951	0.9944	0.9948	1433
1	0.9939	0.9947	0.9943	1318
Accuracy			0.9945	2751
Macro avg	0.9945	0.9946	0.9945	2751
Weighted avg	0.9945	0.9945	0.9945	2751

Hình 7: Bảng báo cáo phân loại

Demo

Tài liệu tham khảo

- Vangelis Metsis, Ion Androutsopoulos, and Geogios P. *Spam filtering with naive bayes-which naive bayes?*. Third conference on email and anti-spam (CEAS), 2006.
- Vũ Hữu Tiệp. *Machine learning cơ bản*. Nhà xuất bản Khoa học và Kỹ thuật, 2018.

The End