

Báo cáo đồ án 1

Các thành viên:
21120205 - Nguyễn Tạ Bảo
21120511 - Lê Nguyễn

Trường Đại học Khoa học Tự nhiên VNU-HCM
Khoa Công nghệ thông tin
Môn: Trí tuệ nhân tạo cho an ninh thông tin

Mục lục

Mức độ
hoàn
thành

Dữ liệu

Mô hình

Thử
nghiệm
thực tế

1 Mức độ hoàn thành

2 Dữ liệu

3 Mô hình

4 Thử nghiệm thực tế

Mục lục

Mức độ
hoàn
thành

Dữ liệu

Mô hình

Thử
nghiệm
thực tế

1 Mức độ hoàn thành

2 Dữ liệu

3 Mô hình

4 Thử nghiệm thực tế

Mục lục

Mức độ
hoàn
thành

Dữ liệu

Mô hình

Thử
nghiệm
thực tế

1 Mức độ hoàn thành

2 Dữ liệu

3 Mô hình

4 Thử nghiệm thực tế

Dữ liệu gồm các cột là Message ID, Subject, Message, Spam/Ham, Split. Trong đó Spam/Ham là label để chỉ ra một email là rác hoặc không, còn cột Subject và Message dùng để chỉ nội dung và tiêu đề của email. Phía dưới là 5 dòng đầu của tập dữ liệu trong file `train.csv`.

Message ID	Subject	Message	Spam/Ham
0	christmas tree farm pictures	NaN	ham
1	vastar resources , inc .	gary , production fr ...	ham
2	calpine daily gas nomination	- calpine daily gas ...	ham
3	re : issue	fyi - see note below ...	ham
5	mcmullen gas for 11 / 99	jackie , since the i ...	ham

Bảng 2: 5 dòng đầu của tập dữ liệu trong file `train.csv`, phần Message đã được bỏ bớt ký tự (ngoài ra cũng đã loại bỏ một vài cột của dữ liệu).

Loại bỏ các phần có dữ liệu bị hư (NaN) và thay thế bằng chuỗi rỗng.

Message ID	Subject	Message	Spam/Ham
0	christmas tree farm pictures		ham
1	vastar resources , inc .	gary , production fr ...	ham
2	calpine daily gas nomination	- calpine daily gas ...	ham
3	re : issue	fyi - see note below ...	ham
5	mcmullen gas for 11 / 99	jackie , since the i ...	ham

Bảng 2: 5 dòng dữ liệu đầu sau khi xử lý

Gộp cột Subject và Message để tạo thành một cột duy nhất Text, điều này sẽ giúp mô hình dễ xử lý hơn, khi ta có một văn bản duy nhất và đưa thành một vector đặc trưng

Message ID	Text
0	christmas tree farm pictures ...
1	vastar resources , inc . gary ...
2	calpine daily gas nomination - ...
3	re : issue fyi - see note belo ...
5	mcmullen gas for 11 / 99 jacki ...

Bảng 2: 5 dòng dữ liệu đầu sau khi gộp lại thành cột Text

1 Mức độ hoàn thành

2 Dữ liệu

3 Mô hình

4 Thử nghiệm thực tế

Vector đặc trưng

Mức độ
hoàn
thành

Dữ liệu

Mô hình

Thử
nghiệm
thực tế

Trích xuất vector đặc trưng của các văn bản (cột Text) bằng TF-IDF.

```
tfidf = TfidfVectorizer(max_features=5000)
X_train_tfidf = tfidf.fit_transform(X_train)
X_test_tfidf = tfidf.transform(X_test)
```

```
[48] ✓ 0.0s

... Matrix: [[0.          0.          0.          ... 0.          0.          0.          0.          ]
 [0.03584193 0.3378409 0.          ... 0.          0.          0.          0.          ]
 [0.          0.          0.          ... 0.          0.          0.          0.          ]
 [0.          0.          0.          ... 0.          0.          0.          0.          ]
 [0.          0.          0.          ... 0.          0.          0.          0.          ]]
Number of features: 5000
```

Hình 1: Các vectord đặc trưng của 5 văn bản đầu tiên

Huấn luyện mô hình

Mức độ
hoàn
thành

Dữ liệu

Mô hình

Thử
nghiệm
thực tế

Sau khi đã có vector đặc trưng, nhóm huấn luyện mô hình Multinomial Naive Bayes để phân loại.

```
# Huấn luyện mô hình Naive Bayes  
model = MultinomialNB()  
model.fit(X_train_tfidf, Y_train)
```

```
# Dự đoán nhãn của tập kiểm thử  
Y_pred = model.predict(X_test_tfidf)
```

Kết quả mô hình

Mức độ
hoàn
thành

Dữ liệu

Mô hình

Thử
nghiệm
thực tế

```
# Đánh giá mô hình bằng các độ đo Accuracy, Precision, Recall, F1
accuracy = accuracy_score(Y_test, Y_pred)
precision = precision_score(Y_test, Y_pred , pos_label='spam')
recall = recall_score(Y_test, Y_pred, pos_label='spam')
f1 = f1_score(Y_test, Y_pred , pos_label='spam')

print(f"Accuracy: {accuracy}")
print(f"Precision: {precision}")
print(f"Recall: {recall}")
print(f"F1 Score: {f1}")
```

[52]

✓ 0.0s

```
... Accuracy: 0.9818417639429312
Precision: 0.9753943217665615
Recall: 0.9891234804862444
F1 Score: 0.9822109275730623
```

Hình 2: Kết quả khi dùng mô hình Multinomial Naive Bayes

1 Mức độ hoàn thành

2 Dữ liệu

3 Mô hình

4 Thử nghiệm thực tế

Thử nghiệm thực tế

Mức độ
hoàn
thành

Dữ liệu

Mô hình

Thử
nghiệm
thực tế

```
def preprocess_text(text, vectorizer):
    # Giả sử sử dụng cùng vectorizer (TF-IDF)
    return vectorizer.transform([text])

def predict_email(model, vectorizer):
    subject = input("Nhập tiêu đề email: ")
    message = input("Nhập nội dung email: ")

    # Ghép subject và message
    email_text = subject + " " + message

    print(f"Subject: {subject}\nMessage: {message}")

    # Tiến xử lý giống như đã làm với tập dữ liệu ban đầu
    email_vectorized = preprocess_text(email_text, vectorizer)

    # Dự đoán
    prediction = model.predict(email_vectorized)

    # Trả về kết quả
    print("Dự đoán: ", "Spam" if prediction[0] == 'spam' else "Ham")

# Ví dụ sử dụng:
predict_email(model, tfidf) # model là mô hình đã huấn luyện, tfidf là vectorizer
```

[53] ✓ 7.1s

```
... Subject: Who are you
      Message: reqjrkqljewnrlkjqnjwre
      Dự đoán: Spam
```

Hình 3: Thử nghiệm thực tế dựa trên input của người dùng

Thử nghiệm thực tế

Mức độ
hoàn
thành

Dữ liệu

Mô hình

Thử
nghiệm
thực tế

```
def evaluate_on_csv(model, vectorizer, file_path):
    # Đọc file CSV
    data = pd.read_csv(file_path)
    # Thay thế NaN bằng chuỗi rỗng
    data['Subject'] = data['Subject'].fillna('')
    data['Message'] = data['Message'].fillna('')

    # Kết hợp 'subject' và 'message' thành một chuỗi
    data['Text'] = data['Subject'] + " " + data['Message']

    # Tiền xử lý dữ liệu
    X = vectorizer.transform(data['Text']) # Tiền xử lý giống với quá trình huấn luyện
    y_true = data['Spam/Ham'] # Nhân thực tế (spam hoặc ham)

    # Dự đoán
    y_pred = model.predict(X)

    # Đánh giá kết quả
    accuracy = accuracy_score(y_true, y_pred)
    precision = precision_score(y_true, y_pred, pos_label='spam')
    recall = recall_score(y_true, y_pred, pos_label='spam')
    f1 = f1_score(y_true, y_pred, pos_label='spam')

    print(f"Accuracy: {accuracy}")
    print(f"Precision: {precision}")
    print(f"Recall: {recall}")
    print(f"F1 Score: {f1}")

    # Ví dụ sử dụng:
    evaluate_on_csv(model, tfidf, "val.csv") # model là mô hình đã huấn luyện, tfidf là vectorizer
```

[54] ✓ 0.9s

```
... Accuracy: 0.9818417639429312
Precision: 0.9753943217665615
Recall: 0.9891234804862444
F1 Score: 0.9822189275738623
```

Hình 3: Kết quả của mô hình khi đánh giá trên tập val

Spam Classifier

Email Subject

IELTS Academic Test Preparation

Email Message

Hi Nguyen

We are excited to let you know that you can purchase this course for the special offer of \$99 USD. This offer is valid until Monday 14th October*.

Predict

Evaluate on CSV

Choose a CSV file



Drag and drop file here

Limit 200MB per file • CSV

Browse files



val.csv 4.2MB



Accuracy: 0.9818417639429312

Precision: 0.9753943217665615

Recall: 0.9891234804862444

F1 Score: 0.9822109275730623

Hình 3: Thử nghiệm thực tế bằng ứng dụng cho phép người dùng nhập hoặc đọc file CSV