Tut 6: Sử dụng Support Vector Machines trong việc phân loại text

- **Tác giả: Mai Đức Trung.**
- **Nhóm nghiên cứu về AI Đại học Bách Khoa Thành phố Hồ Chí Minh.**

1. Chia tập train và test:

```
from sklearn.datasets import fetch_20newsgroups
twenty_train = fetch_20newsgroups(subset='train', random_state=21)
train_label = twenty_train.target
twenty_test = fetch_20newsgroups(subset='test', random_state=21)
test_label = twenty_test.target
len(twenty_train.data), len(twenty_test.data), len(test_label)
```

- Trong đó, fetch_20newsgroups chính là tập dữ liệu ví dụ mà ta sẽ thực hiện, TfidfVectorizer là thư viện để tạo vector tf-idf mà ta đã làm quen trong Lab 1.

2. Trích xuất các features từ tệp văn bản

- Để thực hiện học máy trên các tài liệu văn bản, đầu tiên chúng ta cần phải biến các nội dung văn bản vào vectơ số học.
- Tiền xử lý text, tokenizing và lọc các stopwords được bao gồm trong một thành phần cấp cao mà có thể xây dựng một từ điển của các features và chuyển đổi text về vector.

```
from sklearn.feature_extraction.text import CountVectorizer
count_vect = CountVectorizer()
```

```
X_train_counts = count_vect.fit_transform(twenty_train.data)
X_train_counts.shapecount_vect.vocabulary_.get('for')
```

- CountVectorizer hỗ trợ số lượng N-grams của các từ hoặc ký tự liên tiếp. Khi được sử dụng, bộ vectơ đã tạo ra một từ điển của các features có chỉ mục:

```
ngram_count_vect = CountVectorizer(ngram_range=(1, 5))
XX_train_counts = ngram_count_vect.fit_transform(twenty_train.data)
XX_train_counts.shape
ngram_count_vect.vocabulary_.get('algorithm for')
```

3. Từ số lần xuất hiện đến frequency:

- Các tài liệu dài hơn sẽ có giá trị số trung bình cao hơn tài liệu ngắn hơn, mặc dù chúng có thể nói về cùng một chủ đề. Vấn đề này đã được nhắc trong buổi training số 1, ta sẽ đi tính vector tf-idf:

```
from sklearn.feature_extraction.text import TfidfVectorizer
tfidf_vectorizer = TfidfVectorizer()
X_train_tfidf = tfidf_vectorizer.fit_transform(twenty_train.data)
```

4. Xây dựng Pineline

- Import các thư viện cần thiết:

```
from sklearn.pipeline import Pipeline
from sklearn.linear_model import SGDClassifier
from sklearn.svm import SVC, LinearSVC, NuSVC
```

- Tạo pineline và fit model vào:

```
text_clf = Pipeline([('vect', TfidfVectorizer()),('clf', LinearSVC()),])
text_clf.fit(twenty_train.data, twenty_train.target)
```

- Kiểm tra bằng metric

```
from sklearn import metrics
print(metrics.classification_report(twenty_test.target, predicted,
```

target_names=twenty_test.target_names))

	precision	recall	f1-score	support
alt.atheism	0.82	0.80	0.81	319
comp.graphics	0.76	0.80	0.78	389
comp.os.ms-windows.misc	0.77	0.73	0.75	394
comp.sys.ibm.pc.hardware	0.71	0.76	0.74	392
comp.sys.mac.hardware	0.84	0.86	0.85	385
comp.windows.x	0.87	0.76	0.81	395
misc.forsale	0.83	0.91	0.87	390
rec.autos	0.92	0.91	0.91	396
rec.motorcycles	0.95	0.95	0.95	398
rec.sport.baseball	0.92	0.95	0.93	397
rec.sport.hockey	0.96	0.98	0.97	399
sci.crypt	0.93	0.94	0.93	396
sci.electronics	0.81	0.79	0.80	393
sci.med	0.90	0.87	0.88	396
sci.space	0.90	0.93	0.92	394
soc.religion.christian	0.84	0.93	0.88	398
talk.politics.guns	0.75	0.92	0.82	364
talk.politics.mideast	0.97	0.89	0.93	376
talk.politics.misc	0.82	0.62	0.71	310
talk.religion.misc	0.75	0.61	0.68	251
avg / total	0.85	0.85	0.85	7532

4. Parameter tuning sử dụng grid search

- Import thư viện và tạo paramater:

  from sklearn.model_selection import GridSearchCV

  parameters = {'vect__ngram_range': [(1, 1), (1, 2)],

  'vect__use_idf': (True, False),

  'clf__C': (1.0, 0.1, 1e-2, 1e-3),}

- Khởi tạo model GridSearchCV:

  gs_clf = GridSearchCV(text_clf, parameters, n_jobs=-1)

- Fit training data vào:

  gs_clf = gs_clf.fit(twenty_train.data, twenty_train.target)

- Lúc này, ta đã có model và có thể sử dụng để dự đoán:

  twenty_train.target_names[gs_clf.predict(['God is love'])[0]]

  output: 'soc.religion.christian'

- Ta cũng có thể print report ra để kiểm tra:

```
from sklearn import metrics

print(metrics.classification_report(twenty_test.target, predicted,

   target_names=twenty_test.target_names))
```

	precision	recall	f1-score	support
alt.atheism	0.83	0.79	0.81	319
comp.graphics	0.74	0.80	0.77	389
comp.os.ms-windows.misc	0.77	0.77	0.77	394
comp.sys.ibm.pc.hardware	0.73	0.76	0.74	392
comp.sys.mac.hardware	0.83	0.86	0.85	385
comp.windows.x	0.87	0.76	0.81	395
misc.forsale	0.84	0.91	0.87	390
rec.autos	0.94	0.91	0.92	396
rec.motorcycles	0.96	0.97	0.96	398
rec.sport.baseball	0.91	0.94	0.93	397
rec.sport.hockey	0.95	0.98	0.97	399
sci.crypt	0.93	0.95	0.94	396
sci.electronics	0.82	0.78	0.80	393
sci.med	0.90	0.86	0.88	396
sci.space	0.89	0.93	0.91	394
soc.religion.christian	0.85	0.94	0.89	398
talk.politics.guns	0.76	0.93	0.84	364
talk.politics.mideast	0.96	0.90	0.93	376
talk.politics.misc	0.84	0.64	0.73	310
talk.religion.misc	0.79	0.63	0.70	251
avg / total	0.86	0.86	0.86	7532