Московский государственный технический университет им. Н.Э. Баумана
Кафедра «Системы обработки информации и управления»

Лабораторная работа №3
по дисциплине
«Методы машинного обучения»

Выполнил:
студент группыИУ5И-23М

Лю Цзычжан

Москва — 2025 г.

Цель лабораторной работы: изучение методов классификации текстов.

Требования к отчету:

Отчет по лабораторной работе должен содержать:

1. титульный лист;
2. описание задания;
3. текст программы;
4. экранные формы с примерами выполнения программы.

Задание:

Для произвольного набора данных, предназначенного для классификации текстов, решите задачу классификации текста двумя способами:

1. Способ 1. На основе CountVectorizer или TfidfVectorizer.
2. Способ 2. На основе моделей word2vec или Glove или fastText.
3. Сравните качество полученных моделей.

Для поиска наборов данных в поисковой системе можно использовать ключевые слова "datasets for text classification".

Способ 1. На основе CountVectorizer или TfidfVectorizer.

```
from sklearn.datasets import fetch_20newsgroups
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import LogisticRegression
from sklearn.pipeline import Pipeline
from sklearn.metrics import classification_report

# 加载数据
categories = ['rec.sport.baseball', 'sci.med', 'talk.politics.misc']
data = fetch_20newsgroups(subset='train', categories=categories,
remove=('headers', 'footers', 'quotes'))
```

```
test = fetch_20newsgroups(subset='test', categories=categories,
remove=('headers', 'footers', 'quotes'))

# 构建管道：TF-IDF + 逻辑回归
pipe1 = Pipeline([
    ('tfidf', TfidfVectorizer(max_features=500)),
    ('clf', LogisticRegression(max_iter=100))
])

pipe1.fit(data.data, data.target)
y_pred1 = pipe1.predict(test.data)

print(classification_report(test.target, y_pred1, target_names=categories))
```

```
(venv) PS E:\BMSTU\一下\jqxx\MMO\np5> python -u "e:\BMSTU\一下\jqxx\MMO\np5\lab6.py"
                  precision    recall  f1-score   support

rec.sport.baseball       0.80      0.84      0.82       397
         sci.med       0.81      0.80      0.81       396
talk.politics.misc       0.76      0.72      0.74       310

        accuracy                           0.79      1103
       macro avg       0.79      0.79      0.79      1103
    weighted avg       0.79      0.79      0.79      1103
```

Способ 2. На основе моделей word2vec или Glove или fastText.

```
import nltk
for resource in ['punkt', 'punkt_tab']:
    try:
        nltk.data.find(f'tokenizers/{resource}')
    except LookupError:
        nltk.download(resource)

import numpy as np
import gensim.downloader as api
from nltk.tokenize import word_tokenize
from sklearn.datasets import fetch_20newsgroups
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report

nltk.download('punkt')

# 加载 GloVe 词向量
w2v = api.load('glove-wiki-gigaword-100')

# 加载数据
```

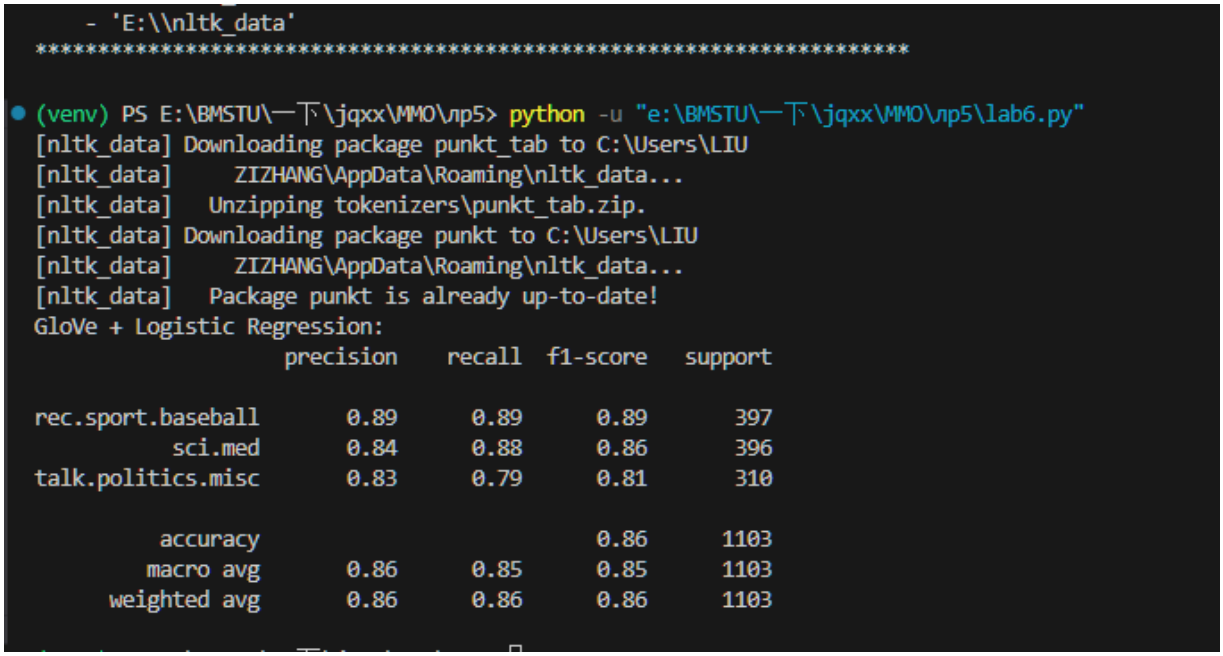
```
categories = ['rec.sport.baseball', 'sci.med', 'talk.politics.misc']
data = fetch_20newsgroups(subset='train', categories=categories,
remove=('headers', 'footers', 'quotes'))
test = fetch_20newsgroups(subset='test', categories=categories,
remove=('headers', 'footers', 'quotes'))

# 文本转向量：平均词向量
def vectorize(texts):
    vectors = []
    for text in texts:
        words = word_tokenize(text.lower())
        word_vecs = [w2v[word] for word in words if word in w2v]
        if word_vecs:
            vectors.append(np.mean(word_vecs, axis=0))
        else:
            vectors.append(np.zeros(100))  # GloVe 是 100 维
    return np.array(vectors)

# 转换训练和测试集
X_train = vectorize(data.data)
X_test = vectorize(test.data)

# 训练分类器
clf = LogisticRegression(max_iter=1000)
clf.fit(X_train, data.target)
y_pred = clf.predict(X_test)

# 输出分类结果
print("GloVe + Logistic Regression:")
print(classification_report(test.target, y_pred, target_names=categories))
```


- 'E:\\nltk_data'
****
(venv) PS E:\BMSTU\一下\jqxx\MMO\np5> python -u "e:\BMSTU\一下\jqxx\MMO\np5\lab6.py"
[nltk_data] Downloading package punkt_tab to C:\Users\LIU
[nltk_data] ZIZHANG\AppData\Roaming\nltk_data...
[nltk_data] Unzipping tokenizers\punkt_tab.zip.
[nltk_data] Downloading package punkt to C:\Users\LIU
[nltk_data] ZIZHANG\AppData\Roaming\nltk_data...
[nltk_data] Package punkt is already up-to-date!
GloVe + Logistic Regression:
precision recall f1-score support
rec.sport.baseball 0.89 0.89 0.89 397
sci.med 0.84 0.88 0.86 396
talk.politics.misc 0.83 0.79 0.81 310
accuracy 0.86 1103
macro avg 0.86 0.85 0.85 1103
weighted avg 0.86 0.86 0.86 1103