****Московский государственный технический университет им. Н.Э. Баумана

Кафедра «Системы обработки информации и управления»

Лабораторная работа №9

по дисциплине

«Методы машинного обучения»

на тему

# **«Классификация текста»**

Выполнил:

студент группы ИУ5-22М

Джин Шуо

Москва — 2024 г.

**1. Цель лабораторной работы**

изучение методов классификации текстов.

**2. Задание**

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

Способ 1. На основе CountVectorizer или TfidfVectorizer.

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

Сравните качество полученных моделей.

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

**3. текст программы**

import numpy as np

from sklearn.datasets import fetch\_20newsgroups

from sklearn.feature\_extraction.text import CountVectorizer, TfidfVectorizer

from sklearn.model\_selection import train\_test\_split

from sklearn.naive\_bayes import MultinomialNB

from sklearn.metrics import accuracy\_score

newsgroups\_data = fetch\_20newsgroups(subset='all')

X, y = newsgroups\_data.data, newsgroups\_data.target

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# CountVectorizer

count\_vect = CountVectorizer()

X\_train\_counts = count\_vect.fit\_transform(X\_train)

X\_test\_counts = count\_vect.transform(X\_test)

clf\_count = MultinomialNB()

clf\_count.fit(X\_train\_counts, y\_train)

y\_pred\_count = clf\_count.predict(X\_test\_counts)

accuracy\_count = accuracy\_score(y\_test, y\_pred\_count)

print(f"CountVectorizer 准确率: {accuracy\_count}")

# TfidfVectorizer

tfidf\_vect = TfidfVectorizer()

X\_train\_tfidf = tfidf\_vect.fit\_transform(X\_train)

X\_test\_tfidf = tfidf\_vect.transform(X\_test)

clf\_tfidf = MultinomialNB()

clf\_tfidf.fit(X\_train\_tfidf, y\_train)

y\_pred\_tfidf = clf\_tfidf.predict(X\_test\_tfidf)

accuracy\_tfidf = accuracy\_score(y\_test, y\_pred\_tfidf)

print(f"TfidfVectorizer 准确率: {accuracy\_tfidf}")

import os

import numpy as np

from sklearn.datasets import fetch\_20newsgroups

from sklearn.model\_selection import train\_test\_split

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import accuracy\_score

# GloVe

def load\_glove\_model(glove\_file):

glove\_model = {}

with open(glove\_file, 'r', encoding='utf-8') as f:

for line in f:

split\_line = line.split()

word = split\_line[0]

embedding = np.array(split\_line[1:], dtype=np.float64)

glove\_model[word] = embedding

return glove\_model

glove\_model = load\_glove\_model('glove.6B.300d.txt')

newsgroups\_data = fetch\_20newsgroups(subset='all')

X, y = newsgroups\_data.data, newsgroups\_data.target

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

def text\_to\_glove\_vector(text, model):

words = text.split()

word\_vecs = [model[word] for word in words if word in model]

if len(word\_vecs) == 0:

return np.zeros(300)

else:

return np.mean(word\_vecs, axis=0)

X\_train\_vec = np.array([text\_to\_glove\_vector(text, glove\_model) for text in X\_train])

X\_test\_vec = np.array([text\_to\_glove\_vector(text, glove\_model) for text in X\_test])

clf\_glove = RandomForestClassifier(n\_estimators=100, random\_state=42)

clf\_glove.fit(X\_train\_vec, y\_train)

y\_pred\_glove = clf\_glove.predict(X\_test\_vec)

accuracy\_glove = accuracy\_score(y\_test, y\_pred\_glove)

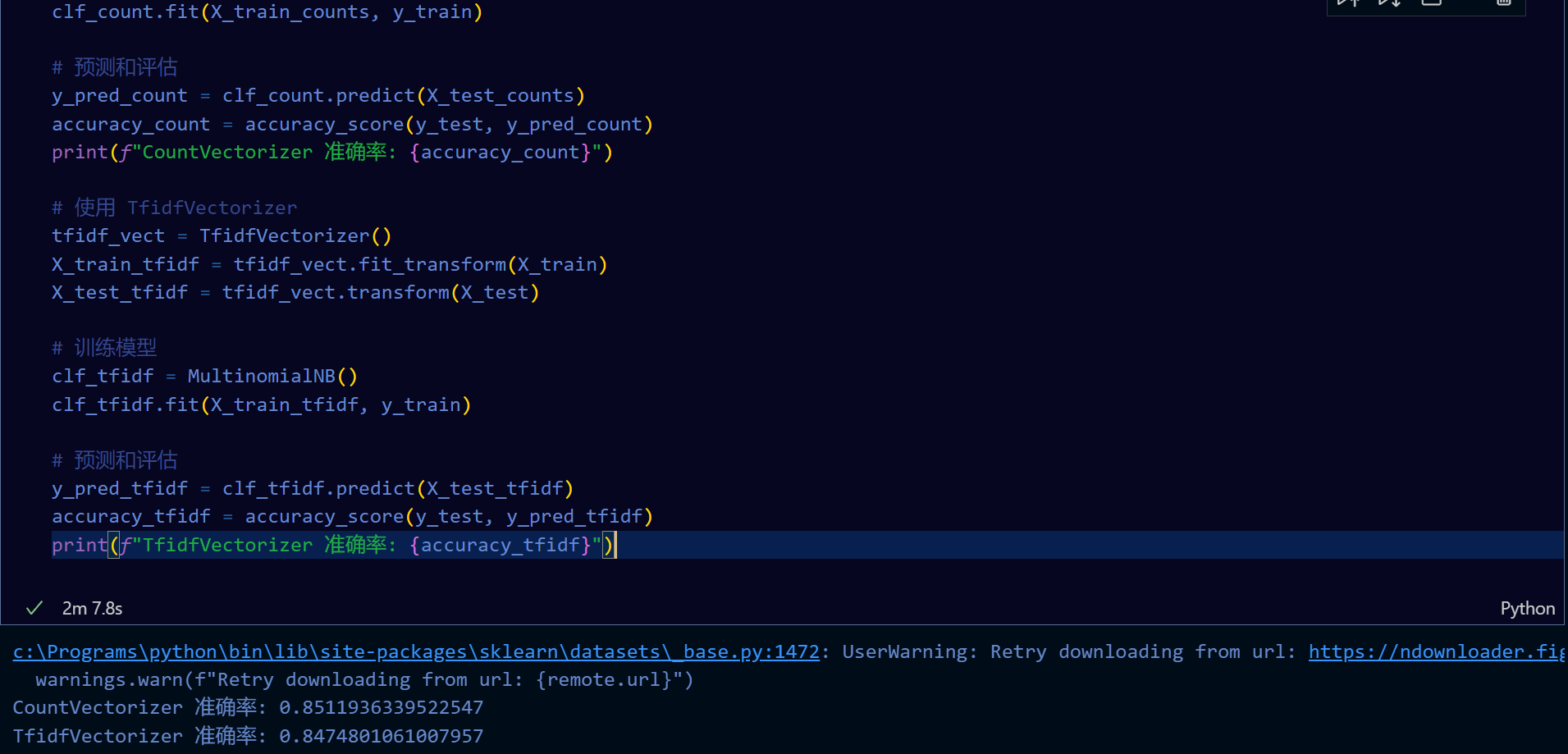
print(f"GloVe 准确率: {accuracy\_glove}")

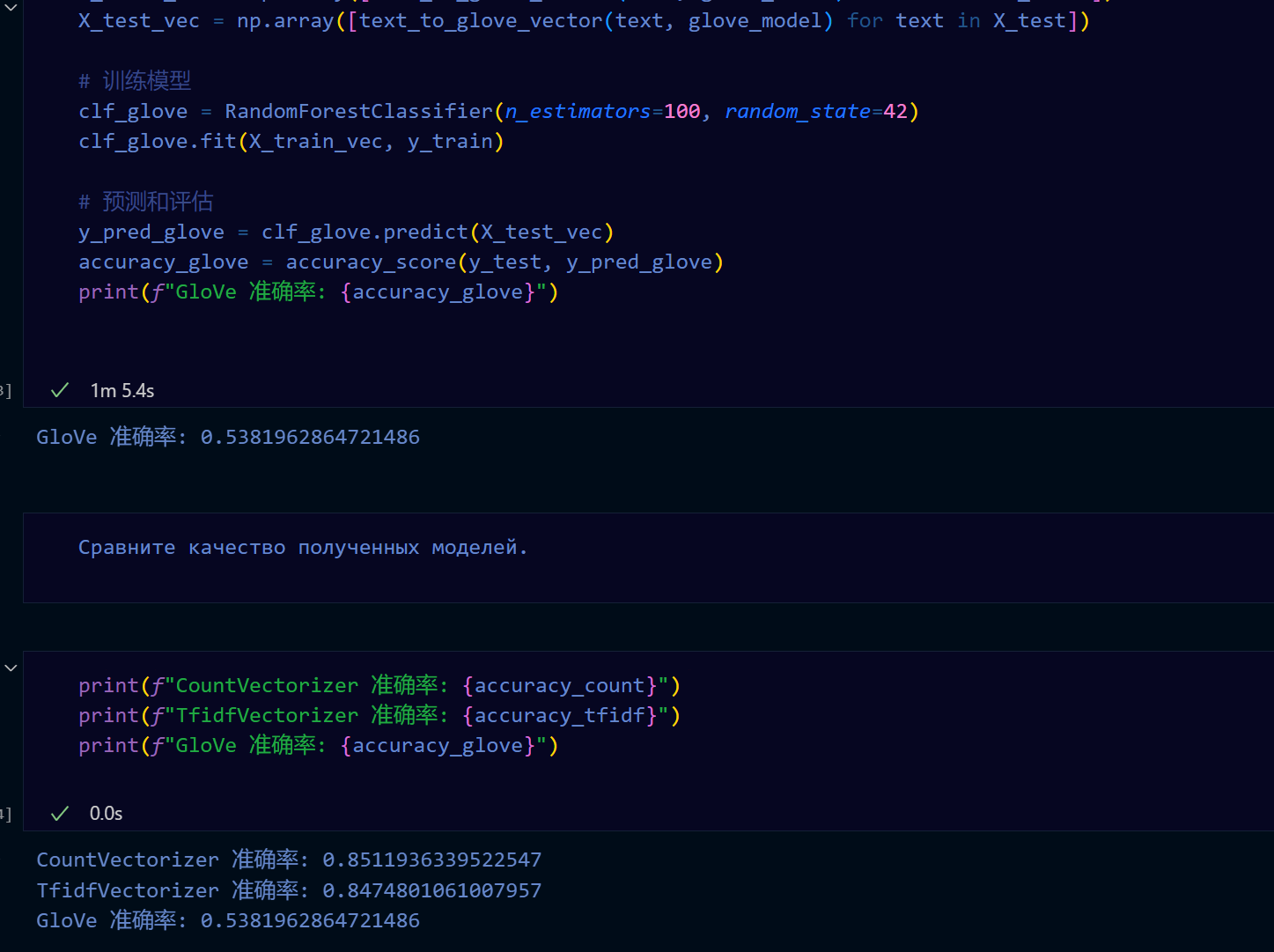
print(f"CountVectorizer 准确率: {accuracy\_count}")

print(f"TfidfVectorizer 准确率: {accuracy\_tfidf}")

print(f"GloVe 准确率: {accuracy\_glove}")

1. **экранные формы с примерами выполнения программы.**





**Список литературы**

[1] https://github.com/ugapanyuk/ml\_course\_2021/wiki/LAB\_MMO\_\_TEXT\_CLAS