****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 ИУ5-23М  
Чжэн Сяохуэй

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:

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".

**3. текст программы**

# 导入必要的库

from sklearn.datasets import fetch\_20newsgroups

from sklearn.feature\_extraction.text import TfidfVectorizer

from sklearn.linear\_model import LogisticRegression

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

from sklearn.metrics import accuracy\_score

import fasttext

import fasttext.util

# 下载20 Newsgroups数据集

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

X, y = newsgroups.data, newsgroups.target

# 数据集划分

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=42)

# 方法1：TfidfVectorizer + Logistic Regression

vectorizer = TfidfVectorizer(max\_features=10000)

X\_train\_tfidf = vectorizer.fit\_transform(X\_train)

X\_test\_tfidf = vectorizer.transform(X\_test)

# 训练逻辑回归模型

lr\_model = LogisticRegression(max\_iter=1000)

lr\_model.fit(X\_train\_tfidf, y\_train)

# 预测和评估

y\_pred\_tfidf = lr\_model.predict(X\_test\_tfidf)

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

print(f'TfidfVectorizer + Logistic Regression Accuracy: {accuracy\_tfidf:.4f}')

# 方法2：fastText

# 保存训练和测试数据到文件

train\_data = 'train.txt'

test\_data = 'test.txt'

with open(train\_data, 'w') as f:

    for text, label in zip(X\_train, y\_train):

        f.write('\_\_label\_\_{} {}\n'.format(label, text.replace("\n", " ")))

with open(test\_data, 'w') as f:

    for text, label in zip(X\_test, y\_test):

        f.write('\_\_label\_\_{} {}\n'.format(label, text.replace("\n", " ")))

# 训练fastText模型

# fasttext.util.download\_model('en', if\_exists='ignore')  # 下载预训练的英语词向量

model = fasttext.train\_supervised(input=train\_data, epoch=25, lr=1.0, wordNgrams=2, verbose=2)

# 评估fastText模型

def fasttext\_evaluate(model, test\_data):

    correct = 0

    total = 0

    with open(test\_data, 'r') as f:

        for line in f:

            label, text = line.split(' ', 1)

            label = label.replace('\_\_label\_\_', '').strip()

            pred\_label = model.predict(text.strip())[0][0].replace('\_\_label\_\_', '')

            if label == pred\_label:

                correct += 1

            total += 1

    return correct / total

accuracy\_fasttext = fasttext\_evaluate(model, test\_data)

print(f'fastText Accuracy: {accuracy\_fasttext:.4f}')

4. экранные формы с примерами выполнения программы.

TfidfVectorizer + Logistic Regression Accuracy: 0.8744

Read 3M words

Number of words: 305776

Number of labels: 20

Progress: 100.0% words/sec/thread: 1397325 lr: 0.000000 avg.loss: 0.239882 ETA: 0h 0m 0s

fastText Accuracy: 0.8571

**Список литературы**

[1] Гапанюк Ю. Е. COURSE\_MMO\_SPRING\_2024// GitHub. –– 2024. –– Режим доступа: https://github.com/ugapanyuk/courses\_current/wiki/COURSE\_MMO\_SPRING\_2024

[2] <https://www.kaggle.com/datasets>