МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

им. Н.Э. Баумана

Факультет «Информатика и системы управления»

Кафедра «Систем обработки информации и управления»

ОТЧЕТ

Рубежный контроль№__2__

по дисциплине «Методы машинного обучения

»

Тема: «Методы обработки данных»

ИСПОЛНИТЕЛЬ: ____Ван Тяньшо____

ФИО

группа ИУ5И-23М _____

подпись

"28"___05___2024 г.

"__"_____2024 г

Москва - 2024

Задание

Необходимо подготовить отчет по рубежному контролю и разместить его в Вашем репозитории. Вы можете использовать титульный лист, или в начале ноутбука в текстовой ячейке указать Ваши Ф.И.О. и группу.

Тема: Методы обработки текстов.

Решение задачи классификации текстов.

Необходимо решить задачу классификации текстов на основе любого выбранного Вами датасета (кроме примера, который рассматривался в лекции). Классификация может быть бинарной или многоклассовой. Целевой признак из выбранного Вами датасета может иметь любой физический смысл, примером является задача анализа тональности текста.

Необходимо сформировать два варианта векторизации признаков - на основе CountVectorizer и на основе TfidfVectorizer.

Мой наборов данных：20 Newsgroups

Код задачи

```
# 安装必要的库
!pip install scikit-learn

# 导入必要的库
import pandas as pd
from sklearn.datasets import fetch_20newsgroups
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.svm import LinearSVC
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, accuracy_score

# 下载20类新闻组数据集
newsgroups = fetch_20newsgroups(subset='all', shuffle=True, random_state=42)
X, y = newsgroups.data, newsgroups.target

# 划分训练集和测试集
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# 使用CountVectorizer进行特征提取
count_vectorizer = CountVectorizer(stop_words='english')
X_train_count = count_vectorizer.fit_transform(X_train)
X_test_count = count_vectorizer.transform(X_test)

# 使用TfidfVectorizer进行特征提取
tfidf_vectorizer = TfidfVectorizer(stop_words='english')
X_train_tfidf = tfidf_vectorizer.fit_transform(X_train)
X_test_tfidf = tfidf_vectorizer.transform(X_test)

# 定义分类器
classifiers = {
    'LinearSVC': LinearSVC(),
    'LogisticRegression': LogisticRegression(max_iter=1000)
}

# 评估每个分类器的性能
for vec_name, (X_train_vec, X_test_vec) in zip(['CountVectorizer', 'TfidfVectorizer'], [(X_train_count, X_test_count), (X_train_tfidf, X_test_tfidf)]):
    for clf_name, clf in classifiers.items():
        clf.fit(X_train_vec, y_train)
        y_pred = clf.predict(X_test_vec)

        print(f"Results for {vec_name} with {clf_name}:")
        print(classification_report(y_test, y_pred))
        print(f"Accuracy: {accuracy_score(y_test, y_pred)}\n")
```

Рисунок 1

Результат выполнения задачи:

1.Results for CountVectorizer with LinearSVC:

```
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.10/dist-packages (1.2.2)
Requirement already satisfied: numpy>=1.17.3 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (1.25.2)
Requirement already satisfied: scipy>=1.3.2 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (1.11.4)
Requirement already satisfied: joblib>=1.1.1 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (1.4.2)
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (3.5.0)
/usr/local/lib/python3.10/dist-packages/sklearn/svm/_base.py:1244: ConvergenceWarning: Liblinear failed to converge, increase the number of iterations.
  warnings.warn(
Results for CountVectorizer with LinearSVC:
              precision    recall  f1-score   support

           0       0.89      0.88      0.89       151
           1       0.80      0.81      0.80       202
           2       0.86      0.85      0.85       195
           3       0.70      0.73      0.72       183
           4       0.83      0.83      0.83       205
           5       0.88      0.88      0.88       215
           6       0.84      0.82      0.83       193
           7       0.91      0.94      0.92       196
           8       0.95      0.96      0.95       168
           9       0.96      0.95      0.95       211
          10       0.97      0.98      0.97       198
          11       0.98      0.96      0.97       201
          12       0.85      0.80      0.82       202
          13       0.90      0.96      0.93       194
          14       0.95      0.96      0.96       189
          15       0.92      0.98      0.95       202
          16       0.93      0.94      0.93       188
          17       0.98      0.95      0.97       182
          18       0.95      0.88      0.91       159
          19       0.87      0.82      0.84       136

    accuracy                           0.90      3770
   macro avg       0.90      0.89      0.89      3770
weighted avg       0.90      0.90      0.90      3770

Accuracy: 0.8954907161803713
```

Рисунок 2

2.Results for CountVectorizer with LogisticRegression:

Results for CountVectorizer with LogisticRegression:

	precision	recall	f1-score	support
0	0.89	0.89	0.89	151
1	0.78	0.84	0.81	202
2	0.88	0.82	0.85	195
3	0.72	0.73	0.73	183
4	0.85	0.85	0.85	205
5	0.88	0.86	0.87	215
6	0.83	0.85	0.84	193
7	0.89	0.94	0.92	196
8	0.97	0.93	0.95	168
9	0.94	0.97	0.96	211
10	0.97	0.96	0.97	198
11	0.98	0.94	0.96	201
12	0.85	0.81	0.83	202
13	0.91	0.95	0.93	194
14	0.94	0.94	0.94	189
15	0.91	0.99	0.95	202
16	0.93	0.94	0.94	188
17	0.99	0.96	0.97	182
18	0.92	0.88	0.90	159
19	0.87	0.81	0.84	136
accuracy			0.89	3770
macro avg	0.90	0.89	0.89	3770
weighted avg	0.90	0.89	0.89	3770

Accuracy: 0.8949602122015915

Рисунок 3

3. Results for TfidfVectorizer with LinearSVC

Results for TfidfVectorizer with LinearSVC:

	precision	recall	f1-score	support
0	0.93	0.93	0.93	151
1	0.83	0.90	0.86	202
2	0.92	0.89	0.90	195
3	0.79	0.80	0.79	183
4	0.90	0.92	0.91	205
5	0.92	0.91	0.92	215
6	0.88	0.85	0.86	193
7	0.93	0.94	0.94	196
8	0.97	0.95	0.96	168
9	0.99	1.00	0.99	211
10	0.97	0.99	0.98	198
11	0.98	0.98	0.98	201
12	0.92	0.88	0.90	202
13	0.95	0.96	0.95	194
14	0.96	0.98	0.97	189
15	0.96	0.98	0.97	202
16	0.95	0.95	0.95	188
17	0.99	0.99	0.99	182
18	0.96	0.91	0.93	159
19	0.92	0.88	0.90	136
accuracy			0.93	3770
macro avg	0.93	0.93	0.93	3770
weighted avg	0.93	0.93	0.93	3770

Accuracy: 0.9305039787798408

Рисунок 4

4. Results for TfidfVectorizer with LogisticRegression:

Results for TfidfVectorizer with LogisticRegression:

	precision	recall	f1-score	support
0	0.89	0.90	0.89	151
1	0.79	0.87	0.83	202
2	0.83	0.83	0.83	195
3	0.72	0.76	0.74	183
4	0.90	0.86	0.88	205
5	0.89	0.85	0.87	215
6	0.84	0.83	0.83	193
7	0.91	0.94	0.93	196
8	0.97	0.93	0.95	168
9	0.97	0.97	0.97	211
10	0.96	0.97	0.97	198
11	0.98	0.95	0.96	201
12	0.87	0.87	0.87	202
13	0.96	0.96	0.96	194
14	0.91	0.98	0.94	189
15	0.92	0.98	0.95	202
16	0.93	0.93	0.93	188
17	0.99	0.98	0.99	182
18	0.93	0.86	0.89	159
19	0.89	0.73	0.80	136
accuracy			0.90	3770
macro avg	0.90	0.90	0.90	3770
weighted avg	0.90	0.90	0.90	3770

Accuracy: 0.9007957559681697

Рисунок 5

5.Оценка метода

	Precision	Recall	F1-Score	Accuracy
CountVectorizer + LinearSVC	0.9	0.9	0.9	0.895
CountVectorizer + Logistic Regression	0.9	0.89	0.89	0.895
TfidfVectorizer + LinearSVC	0.93	0.93	0.93	0.93
TfidfVectorizer +Logistic Regression	0.90	0.90	0.90	0.901

Вывод

Как видно из приведенных выше результатов, производительность различных комбинаций методов векторизации объектов (CountVectorizer и TfidfVectorizer) и классификаторов (LinearSVC и LogisticRegression) в задачах классификации различна.

Комбинация CountVectorizer и LinearSVC показала хорошие результаты, достигнув показателя точности 0,90.

Комбинация CountVectorizer и LogisticRegression также показала хорошие результаты, но они были немного ниже, чем у LinearSVC, с точностью 0,895.

Комбинация TfidfVectorizer и LinearSVC показала наилучшие

результаты среди всех комбинаций, достигнув показателя точности 0,93.

Комбинация TfidfVectorizer и LogisticRegression также показала хорошие результаты с точностью 0,90, но немного ниже, чем LinearSVC.

Таким образом, комбинация TfidfVectorizer и LinearSVC демонстрирует наилучшие характеристики классификации с высочайшей точностью (0,930) и высокой точностью воспроизведения, а также показателем F1.