Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования «Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)» (МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ Информатика и системы управления и искусственный интеллект

КАФЕДРА Системы обработки информации и управления

Рубежный контроль №2 по курсу «Методы машинного обучения в автоматизированных системах обработки информации и управления»

Подготовили:

У Жун

ИУ5И-25М

08.05.2024

Проверил:

Гапанюк Ю. Е.

2024 г.

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.

В качестве классификаторов необходимо использовать два классификатора по варианту для Вашей группы:

Группа	Классификатор №1	Классификатор №2
ИУ5И-25М	SVC	LogisticRegression

Для каждого метода необходимо оценить качество классификации. Сделайте вывод о том, какой вариант векторизации признаков в паре с каким классификатором показал лучшее качество.

Набор данных 20 Newsgroups является одним из классических наборов данных для задачи классификации текста. Он содержит новостные сообщения, собранные с 20 различных новостных групп Usenet в период с 1995 по 1997 год. Набор данных включает сообщения по различным темам, таким как компьютерная графика, медицина, религия и атеизм.

Основные характеристики набора данных 20 Newsgroups:

1. Количество категорий: 20
2. Количество документов: около 20,000
3. Распределение классов: Набор данных включает новости из различных категорий, таких как наука, религия, компьютерная графика и т. д.
4. Язык: В основном на английском языке.

Загрузите набор данных 20 Newsgroups и выполните начальное исследование

```
: from sklearn.datasets import fetch_20newsgroups

категории = ['alt.atheism', 'soc.religion.christian', 'comp.graphics', 'sci.med']
новости_обучение = fetch_20newsgroups(subset='train', categories=категории, shuffle=True, random_state=42)

print("Целевые классы набора данных:", новости_обучение.target_names)
print("\nКоличество образцов:", len(новости_обучение.data))
print("\nНекоторые примеры образцов:\n")
for i in range(3):
    print("Класс:", новости_обучение.target_names[новости_обучение.target[i]])
    print("Текст:", новости_обучение.data[i])
    print("\n----------------------------------------\n")
```

```
Целевые классы набора данных: ['alt.atheism', 'comp.graphics', 'sci.med', 'soc.religion.christian']

Количество образцов: 2257

Некоторые примеры образцов:

Класс: comp.graphics
Текст: From: sd345@city.ac.uk (Michael Collier)
Subject: Converting images to HP LaserJet III?
Nntp-Posting-Host: hampton
Organization: The City University
Lines: 14

Does anyone know of a good way (standard PC application/PD utility) to
convert tif/img/tga files into LaserJet III format.  We would also like to
do the same, converting to HPGL (HP plotter) files.

Please email any response.
```

```
Please email any response.

Is this the correct group?

Thanks in advance.  Michael.
--
Michael Collier (Programmer)                The Computer Unit,
Email: M.P.Collier@uk.ac.city               The City University,
Tel: 071 477-8000 x3769                     London,
Fax: 071 477-8565                           EC1V 0HB.


----------------------------------------

Класс: comp.graphics
Текст: From: ani@ms.uky.edu (Aniruddha B. Deglurkar)
Subject: help: Splitting a trimming region along a mesh
Organization: University Of Kentucky, Dept. of Math Sciences
Lines: 28
```

```
Hi,

I have a problem, I hope some of the 'gurus' can help me solve.

Background of the problem:
I have a rectangular mesh in the uv domain, i.e  the mesh is a
mapping of a 3d Bezier patch into 2d. The area in this domain
which is inside a trimming loop had to be rendered. The trimming
loop is a set of 2d Bezier curve segments.
For the sake of notation: the mesh is made up of cells.

My problem is this :
The trimming area has to be split up into individual smaller
cells bounded by the trimming curve segments. If a cell
is wholly inside the area...then it is output as a whole ,
else it is trivially rejected.

Does any body know how thiss can be done, or is there any algo.
somewhere for doing this.
```

Рис.1

Выполните предварительную обработку текста и извлечение признаков

Используйте CountVectorizer и TfidfVectorizer для преобразования текста в векторы признаков.

```python
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer

# Инициализация CountVectorizer и TfidfVectorizer
count_vectorizer = CountVectorizer(stop_words='english')
tfidf_vectorizer = TfidfVectorizer(stop_words='english')

# Преобразование обучающих данных в признаковые векторы
X_count = count_vectorizer.fit_transform(новости_обучение.data)
X_tfidf = tfidf_vectorizer.fit_transform(новости_обучение.data)

# Вывод формы признаковых векторов
print("Форма признакового вектора CountVectorizer:", X_count.shape)
print("Форма признакового вектора TfidfVectorizer:", X_tfidf.shape)
```

```
Форма признакового вектора CountVectorizer: (2257, 35482)
Форма признакового вектора TfidfVectorizer: (2257, 35482)
```

Рис. 2

Подготовьте помеченные данные, а затем разделите набор данных на обучающий и тестовый наборы

```
from sklearn.model_selection import train_test_split

# Подготовка меток классов
y = новости_обучение.target

# Разделение набора данных на обучающий и тестовый
X_count_train, X_count_test, y_train, y_test = train_test_split(X_count, y, test_size=0.2, random_state=42)
X_tfidf_train, X_tfidf_test, y_train, y_test = train_test_split(X_tfidf, y, test_size=0.2, random_state=42)
```

Рис.3

Используя классификаторы SVC и LogisticRegression для классификации векторов признаков соответственно, классификаторы будут обучены и их производительность будет оценена на тестовом наборе

```
from sklearn.svm import SVC
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report

# Инициализация классификатора SVC
svc_classifier = SVC()
# Обучение классификатора SVC и предсказание
svc_classifier.fit(X_count_train, y_train)
svc_count_predictions = svc_classifier.predict(X_count_test)
svc_count_report = classification_report(y_test, svc_count_predictions)

# Инициализация классификатора LogisticRegression
lr_classifier = LogisticRegression(max_iter=1000)
# Обучение классификатора LogisticRegression и предсказание
lr_classifier.fit(X_tfidf_train, y_train)
lr_tfidf_predictions = lr_classifier.predict(X_tfidf_test)
lr_tfidf_report = classification_report(y_test, lr_tfidf_predictions)

# Вывод отчета о классификации
print("Отчет о классификации классификатора SVC (признаки CountVectorizer):\n", svc_count_report)
print("\nОтчет о классификации классификатора LogisticRegression (признаки TfidfVectorizer):\n", lr_
```

Отчет о классификации классификатора SVC (признаки CountVectorizer):

	precision	recall	f1-score	support
0	0.99	0.87	0.93	86
1	0.77	0.96	0.85	107
2	0.93	0.86	0.90	132
3	0.96	0.91	0.93	127
accuracy			0.90	452
macro avg	0.91	0.90	0.90	452
weighted avg	0.91	0.90	0.90	452

Отчет о классификации классификатора LogisticRegression (признаки TfidfVectorizer):

	precision	recall	f1-score	support
0	0.98	0.92	0.95	86
1	0.91	1.00	0.95	107
2	0.98	0.95	0.97	132
3	0.96	0.94	0.95	127
accuracy			0.96	452
macro avg	0.96	0.95	0.95	452
weighted avg	0.96	0.96	0.96	452

Рис.4

На основании отчета о результатах классификации можно сделать следующие выводы:

Для классификатора SVC, использующего признаки CountVectorizer, точность составляет 0,90, в то время как точность классификатора LogisticRegression, использующего признаки TfidfVectorizer, равна 0,96. Таким образом, сочетание признаков TfidfVectorizer с классификатором LogisticRegression работает лучше.

По другим показателям (например, recall, F1 score и т. д.) комбинация признаков TfidfVectorizer с классификатором LogisticRegression также работает лучше.

В целом, комбинация функций TfidfVectorizer и классификатора LogisticRegression имеет лучшие показатели для этой задачи.