PDF-версия

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, системы управления и искусственный интеллект

КАФЕДРА Системы обработки информации и управления

Рубежный контроль № 2

По курсу
«Методы машинного обучения в АСОИУ»

Выполнил:
Студент группы ИУ5-22М
Кириллов Д.С.

Проверил:
Гапанюк Ю.Е.

2024 г.

РК-2 ММО в АСОИУ (Методы машинного обучения в АСОИУ)

ИУ5-22М Кириллов Д.С. Вариант 7

20.05.2024

Задание

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.

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

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

Группа	Классификатор №1	Классификатор №2
ИУ5-22М, ИУ5И-22М	RandomForestClassifier	LogisticRegression

Ход работы

Подготовка датасета

Взял датасет, который является результатом парсинга сайтов промышленных компаний из РФ. Цель классификации - определить область деятельности компании по тексту на ее сайте и сайтах, связанных с сайтом компании гиперссылками.

К сожалению, датасет совсем небольшой - 79 строк, 9 из которых зашумлены. Кроме того, парсинг различных страниц, каждая из которых структурирована по-своему не всегда эффективен, т.к. нельзя стопроцентно выудить смысловую нагрузку. Это все отразилось на результатах моделей.

In [ ]:
```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import    RandomForestClassifier
from sklearn.model_selection import cross_val_predict
from sklearn.metrics import classification_report

%matplotlib inline

sns.set(style="ticks")
```

In [ ]:
```python
data_folder = '../data/'
df = pd.read_csv(data_folder + '79_rows_text_depth_3.csv')
print("размер:", df.shape)
print("\nколонки:\n", df.dtypes)
```

```
размер: (77, 10)

колонки:
 №                                   int64
ИНН                                int64
Наименование организации          object
Полное наименование организации   object
Сайт                              object
Индустрия                         object
Даты                              object
Телефоны                          object
Текст                             object
Документы                         object
dtype: object
```

In [ ]:
```python
df.head()
```

Out[ ]:

	№	ИНН	Наименование организации	Полное наименование организации	Сайт	Индустрия	Даты	Телефоны	Т
0	17	5051000880	АО "ЩЛЗ"	АКЦИОНЕРНОЕ ОБЩЕСТВО "ЩЕРБИНСКИЙ ЛИФТОСТРОИТЕЛ...	www.shlz.ru	Машиностроение		8-800-350-30-50 8-800-350-30-50 8-800-350-30-5...	компан компа награ, отзывы исто
1	26	7724075162	ФГБУ "НМИЦ ОНКОЛОГИИ ИМ. Н.Н. БЛОХИНА МИНЗДРАВ...	ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ УЧРЕЖДЕН...	www.ronc.ru	Медицинская промышленность	30.8.2022 30.8.2022 30.8.2022 30.8.2022 5.11.2...	8-499-324-24-24 8-499-324-25-94 8-499-324-24-2...	фгбу н онкологии н бло минздра
2	55	7720605108	ООО "ФАБРИКА ВЕНТИЛЯЦИИ ГАЛВЕНТ"	ОБЩЕСТВО С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ "ФАБР...	www.ventilyacia.ru	Лёгкая промышленность	6.4.2020 6.1.2020	8-495-790-76-98 8-495-790-76-98 8-495-790-76-9...	к сожале ваш браузе поддержи ja
3	80	7724190750	АО "СМЕРФИТ КАППА РУС"	АКЦИОНЕРНОЕ ОБЩЕСТВО "СМЕРФИТ КАППА РУС"	www.smurfitkappa.com/ru	Целлюлозно-бумажная промышленность	3.1.2022 13.1.2022 13.1.2022 13.1.2022 13.1.20...		перей основн содержим страницыинн
4	83	7701165130	ООО "НПП "СПЕЦКАБЕЛЬ"	ОБЩЕСТВО С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ НПП "...	www.spcable.ru	Лёгкая промышленность	14.3.2017 20.12.2016		проду кат сертифи продажа

Удалю данные, которые для исследования мне не нужны - все кроме текста сайта и метки области промышленности.

Удалю пропуски и NaN

In [ ]:
```python
df = df.filter(['Индустрия','Текст'], axis=1)

print("\nПропущенные значения, %:")
for index, value in df.isnull().sum().get(lambda x: x > 0).items():
  print("{:25s} {: ^10} {:>10.3f}%\t{:s}".format(index, value, value*100/df.shape[0], str(df[index].dtype)))

data_no_score = df.dropna(subset=["Текст"])
print()
print("Очистка от строк, где Текст = NaN:")
print("Было %d значений. Стало %d. Было удалено %d." % (df.shape[0], data_no_score.shape[0], df.shape[0] - data_no_s

df = data_no_score
```

```
Пропущенные значения, %:
Текст                         7           9.091%        object

Очистка от строк, где Текст = NaN:
Было 77 значений. Стало 70. Было удалено 7.
```

In [ ]:
```python
for industry in df['Индустрия'].unique():
  print(industry, '-', str(df[df['Индустрия'] == industry].count()[0]))
```

```
Машиностроение - 4
Медицинская промышленность - 4
Лёгкая промышленность - 12
Целлюлозно-бумажная промышленность - 7
Электротехническая промышленность - 21
Пищевая промышленность - 2
Текстильная промышленность - 6
Энергетическая промышленность - 2
Деревообрабатывающая  - 3
Металлообработка - 8
Авиационно-космическая промышленность - 1
```

Векторизация признаков

По заданию необходимо сформировать два варианта векторизации признаков - на основе CountVectorizer и на основе TfidfVectorizer.

Сначала сформируем общий словарь для обучения моделей из обучающей и тестовой выборки. В словаре будут слова и столбца "Текст".

In [ ]:
```python
vocab_list = df['Текст']
vocab_list.head()
```

Out[ ]:
```
0     компания о компании награды и отзывы история ...
1     фгбу нмиц онкологии им н н блохина минздрава ...
2     к сожалению ваш браузер не поддерживает javas...
3     перейти к основному содержимому страницыинвес...
4     продукция каталог сертификаты продажа для кли...
Name: Текст, dtype: object
```

CountVectorizer

In [ ]:
```python
count_vectorizer = CountVectorizer()
count_vectorizer.fit(vocab_list)
count_vectorizer_vocab = count_vectorizer.vocabulary_
print('Количество сформированных признаков - {}'.format(len(count_vectorizer_vocab)))
```

```
Количество сформированных признаков - 19724
```

Посмотрим на некоторые из слов сформированного с помощью CountVectorizer словаря:

In [ ]:
```python
for word in list(count_vectorizer_vocab)[1:10]:
  print(f'{word}: {count_vectorizer_vocab[word]}')
```

```
компании: 7838
награды: 9682
отзывы: 11462
история: 7292
сертификаты: 15816
разрешения: 14696
на: 9647
лифты: 8668
устройства: 18277
```

TfidfVectorizer

In [ ]:
```python
tfidf_vectorizer = TfidfVectorizer()
tfidf_vectorizer.fit(vocab_list)
tfidf_vectorizer_vocab = tfidf_vectorizer.vocabulary_
print('Количество сформированных признаков - {}'.format(len(tfidf_vectorizer_vocab)))
print()

for word in list(tfidf_vectorizer_vocab)[1:10]:
  print(f'{word}: {tfidf_vectorizer_vocab[word]}')
```

```
Количество сформированных признаков - 19724

компании: 7838
награды: 9682
отзывы: 11462
история: 7292
сертификаты: 15816
разрешения: 14696
на: 9647
лифты: 8668
устройства: 18277
```

Слова и их количество совпадают у методов CountVectorizer и TfidfVectorizer.

Создание тестовой и тренировочной выборок

In [ ]:
```python
X = df['Текст']
y = df['Индустрия']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

print(*map(lambda x: x.shape, [X_train, X_test, y_train, y_test]))
```

```
(49,) (21,) (49,) (21,)
```

Обучение моделей-классификаторов

По моему варианту необходимо использовать методы:

- RandomForestClassifier
- LogisticRegression

Проверим данные методы совместно с рассмотренными выше вариантами векторизации.

Напишем вспомогательную функцию

In [ ]:
```python
def train_and_score(vectorizer, classifier, X_train, y_train, X_test, y_test):
  X_train_vec = vectorizer.fit_transform(X_train)
  X_test_vec = vectorizer.transform(X_test)

  classifier.fit(X_train_vec, y_train)
  y_pred = classifier.predict(X_test_vec)

  clr = classification_report(y_test, y_pred, zero_division=True, digits=6)

  print('Метод векторизации: {}'.format(vectorizer))
  print('Метод классификации: {}'.format(classifier))
  print('Оценка точности:\n', clr)
  return clr, vectorizer, classifier

def test(vectorizer, classifier):
  docs = [
    'лифты сертификаты на устройства безопасности лифтов', # Электротехническая промышленность
    'цикл производства от бумаги до упаковки', # Целлюлозно-бумажная промышленность
    'продажа красной икры рыбных консервов свежемороженой и вяленой рыбы оптом по всей россии', # Пищевая промышленн
    'добыча рыбы на сахалине камчатке курильских островах', # Пищевая промышленность
  ]
  correct = [
    'Электротехническая промышленность',
    'Целлюлозно-бумажная промышленность',
    'Пищевая промышленность',
    'Пищевая промышленность',
  ]

  X_vec = vectorizer.transform(docs)
  y_pred = classifier.predict(X_vec)

  print()
  print('Ответ модели\t\t\tПравильный ответ')
  for pred, answer in zip(y_pred, correct):
    print(f'{pred}\t{answer}')
```

CountVectorizer, RandomForestClassifier

In [ ]:
```python
cv_rf_clr, cv1, rf1 = train_and_score(CountVectorizer(), RandomForestClassifier(), X_train, y_train, X_test, y_test)
test(cv1, rf1)
```

```
Метод векторизации: CountVectorizer()
Метод классификации: RandomForestClassifier()
Оценка точности:
                                    precision    recall  f1-score   support

             Лёгкая промышленность   1.000000  0.250000  0.400000         4
                    Машиностроение   1.000000  0.000000  0.000000         1
        Медицинская промышленность   1.000000  0.000000  0.000000         1
                  Металлообработка   1.000000  0.000000  0.000000         3
        Текстильная промышленность   1.000000  0.000000  0.000000         4
Целлюлозно-бумажная промышленность   1.000000  1.000000  1.000000         1
 Электротехническая промышленность   0.400000  0.857143  0.545455         7
     Энергетическая промышленность   0.000000  1.000000  0.000000         0

                          accuracy                       0.380952        21
                         macro avg   0.800000  0.388393  0.243182        21
                      weighted avg   0.800000  0.380952  0.305628        21


Ответ модели                    Правильный ответ
Энергетическая промышленность   Электротехническая промышленность
Энергетическая промышленность   Целлюлозно-бумажная промышленность
Энергетическая промышленность   Пищевая промышленность
Энергетическая промышленность   Пищевая промышленность
```

TfidfVectorizer, RandomForestClassifier

In [ ]:
```python
tfv_rf_clr, tfv2, rf2 = train_and_score(TfidfVectorizer(), RandomForestClassifier(), X_train, y_train, X_test, y_tes
test(tfv2, rf2)
```

```
Метод векторизации: TfidfVectorizer()
Метод классификации: RandomForestClassifier()
Оценка точности:
                                    precision    recall  f1-score   support

             Лёгкая промышленность   1.000000  0.000000  0.000000         4
                    Машиностроение   1.000000  0.000000  0.000000         1
        Медицинская промышленность   1.000000  0.000000  0.000000         1
                  Металлообработка   1.000000  0.000000  0.000000         3
        Текстильная промышленность   1.000000  0.000000  0.000000         4
Целлюлозно-бумажная промышленность   1.000000  1.000000  1.000000         1
 Электротехническая промышленность   0.315789  0.857143  0.461538         7
     Энергетическая промышленность   0.000000  1.000000  0.000000         0

                          accuracy                       0.333333        21
                         macro avg   0.789474  0.357143  0.182692        21
                      weighted avg   0.771930  0.333333  0.201465        21


Ответ модели                    Правильный ответ
Энергетическая промышленность   Электротехническая промышленность
Энергетическая промышленность   Целлюлозно-бумажная промышленность
Энергетическая промышленность   Пищевая промышленность
Энергетическая промышленность   Пищевая промышленность
```

CountVectorizer, LogisticRegression

In [ ]:
```python
cv_lr_clr, cv3, lr3 = train_and_score(CountVectorizer(), LogisticRegression(), X_train, y_train, X_test, y_test)
test(cv3, lr3)
```

```
Метод векторизации: CountVectorizer()
Метод классификации: LogisticRegression()
Оценка точности:
                                    precision    recall  f1-score   support

              Деревообрабатывающая   0.000000  1.000000  0.000000         0
             Лёгкая промышленность   0.250000  0.250000  0.250000         4
                    Машиностроение   1.000000  0.000000  0.000000         1
        Медицинская промышленность   1.000000  0.000000  0.000000         1
                  Металлообработка   1.000000  0.000000  0.000000         3
        Текстильная промышленность   1.000000  0.000000  0.000000         4
Целлюлозно-бумажная промышленность   1.000000  1.000000  1.000000         1
 Электротехническая промышленность   0.454545  0.714286  0.555556         7
     Энергетическая промышленность   0.000000  1.000000  0.000000         0

                          accuracy                       0.333333        21
                         macro avg   0.633838  0.440476  0.200617        21
                      weighted avg   0.675325  0.333333  0.280423        21


Ответ модели                    Правильный ответ
Энергетическая промышленность   Электротехническая промышленность
Энергетическая промышленность   Целлюлозно-бумажная промышленность
Энергетическая промышленность   Пищевая промышленность
Энергетическая промышленность   Пищевая промышленность
```

```
c:\Users\dturc\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\linear_model\_logistic.py:469: Conv
ergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
```

TfidfVectorizer, LogisticRegression

In [ ]:
```python
tfv_lr_clr, tfv4, lr4 = train_and_score(TfidfVectorizer(), RandomForestClassifier(), X_train, y_train, X_test, y_tes
test(tfv4, lr4)
```

```
Метод векторизации: TfidfVectorizer()
Метод классификации: RandomForestClassifier()
Оценка точности:
                                    precision    recall  f1-score   support

             Лёгкая промышленность   1.000000  0.250000  0.400000         4
                    Машиностроение   1.000000  0.000000  0.000000         1
        Медицинская промышленность   1.000000  0.000000  0.000000         1
                  Металлообработка   1.000000  0.000000  0.000000         3
        Текстильная промышленность   1.000000  0.000000  0.000000         4
Целлюлозно-бумажная промышленность   1.000000  1.000000  1.000000         1
 Электротехническая промышленность   0.400000  0.857143  0.545455         7
     Энергетическая промышленность   0.000000  1.000000  0.000000         0

                          accuracy                       0.380952        21
                         macro avg   0.800000  0.388393  0.243182        21
                      weighted avg   0.800000  0.380952  0.305628        21


Ответ модели                    Правильный ответ
Энергетическая промышленность   Электротехническая промышленность
Энергетическая промышленность   Целлюлозно-бумажная промышленность
Энергетическая промышленность   Пищевая промышленность
Энергетическая промышленность   Пищевая промышленность
```

Выводы

В ходе выполнения работы были сформированы два варианта векторизации признаков - на основе CountVectorizer и на основе TfidfVectorizer. Для каждого метода была произведена оценка качество классификации classification report.

К сожалению, невысокое качество датасета не позволило достичь значимых уровней достоверности результата в всех классификаторах. Таким образом, сделайть вывод о том, какой вариант векторизации признаков в паре с каким классификатором показал лучшее качество, не получится.

Однако, на этом же датасете, но со стеммингом для русского языка мне удалось достичь среднеклассовой точности 50%. А на нормальных датасетах точность классификации комбинации методов (CountVectorizer, TfidfVectorizer) x (RandomForestClassifier, LogisticRegression, MultinomialNB, SVC) и других показывают среднеклассовую точность выше 75%.