Московский государственный технический университет им. Н.Э. Баумана
Кафедра «Системы обработки информации и управления»

Лабораторная работа по дисциплине «Технологии машинного обучения» на тему «Рубежный контроль №2»

Выполнил:
студент группы ИУ5-64Б
Зубков А. Д.

Москва — 2020 г.

1. Рубежный контроль №2

Зубков Андрей, ИУ5-64, Вариант №6, Задача №1

2. 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 признаки на основе CountVectorizer или TfidfVectorizer.

В качестве классификаторов необходимо использовать два классификатора, не относящихся к наивным Байесовским методам (например, LogisticRegression, LinearSVC), а также Multinomial Naive Bayes (MNB), Complement Naive Bayes (CNB), Bernoulli Naive Bayes.

Для каждого метода необходимо оценить качество классификации с помощью хотя бы одной метрики качества классификации (например, Accuracy).

Сделате выводы о том, какой классификатор осуществляет более качественную классификацию на Вашем наборе данных.

3. Решение

Подключим необходимые библиотеки и загрузим набор данных

```
[1]: from sklearn.linear_model import LogisticRegression
     from sklearn.svm import LinearSVC
     from sklearn.naive_bayes import MultinomialNB, ComplementNB, BernoulliNB
     from sklearn.metrics import accuracy_score
     from sklearn.datasets import fetch_20newsgroups
     from sklearn.feature_extraction.text import TfidfVectorizer
     import pandas as pd
     import seaborn as sns
     import matplotlib.pyplot as plt

     %matplotlib inline

     # Устанавливаем тип графиков
     sns.set(style="ticks")

     # Для лучшего качествоа графиков
     from IPython.display import set_matplotlib_formats
     set_matplotlib_formats("retina")

     # Устанавливаем ширину экрана для отчета
     pd.set_option("display.width", 70)

     # Загружаем данные
     data_train = fetch_20newsgroups(subset='train', remove=('headers',␣
      ↪'footers'))
```

```
data_test = fetch_20newsgroups(subset='test', remove=('headers',␣
 ↪'footers'))
```

[2]:
```
data_train.target.shape
```

[2]:
```
(11314,)
```

[3]:
```
data_train.data[:3]
```

[3]:
```
['I was wondering if anyone out there could enlighten me on this car I␣
 ↪saw\nthe
other day. It was a 2-door sports car, looked to be from the late 60s/
 ↪\nearly
70s. It was called a Bricklin. The doors were really small. In␣
 ↪addition,\nthe
front bumper was separate from the rest of the body. This is \nall I know.␣
 ↪If
anyone can tellme a model name, engine specs, years\nof production, where␣
 ↪this
car is made, history, or whatever info you\nhave on this funky looking car,
please e-mail.',
 "A fair number of brave souls who upgraded their SI clock oscillator
have\nshared their experiences for this poll. Please send a brief message
detailing\nyour experiences with the procedure. Top speed attained, CPU␣
 ↪rated
speed,\nadd on cards and adapters, heat sinks, hour of usage per day,␣
 ↪floppy
disk\nfunctionality with 800 and 1.4 m floppies are especially requested.
 ↪\n\nI
will be summarizing in the next two days, so please add to the
network\nknowledge base if you have done the clock upgrade and haven't␣
 ↪answered
this\npoll. Thanks.",
 'well folks, my mac plus finally gave up the ghost this weekend␣
 ↪after\nstarting
life as a 512k way back in 1985.  sooo, i\'m in the market for a\nnew␣
 ↪machine a
bit sooner than i intended to be…\n\ni\'m looking into picking up a␣
 ↪powerbook
160 or maybe 180 and have a bunch\nof questions that (hopefully) somebody␣
 ↪can
answer:\n\n* does anybody know any dirt on when the next round of
powerbook\nintroductions are expected?  i\'d heard the 185c was supposed␣
 ↪to make
an\nappearence "this summer" but haven\'t heard anymore on it - and since
i\ndon\'t have access to macleak, i was wondering if anybody out there␣
 ↪had\nmore
info…\n\n* has anybody heard rumors about price drops to the powerbook line
like the\nones the duo\'s just went through recently?\n\n* what\'s the
impression of the display on the 180?  i could probably swing\na 180 if i␣
 ↪got
the 80Mb disk rather than the 120, but i don\'t really have\na feel for␣
 ↪how much
"better" the display is (yea, it looks great in the\nstore, but is that all
"wow" or is it really that good?).  could i solicit\nsome opinions of␣
 ↪people who
use the 160 and 180 day-to-day on if its worth\ntaking the disk size and␣
 ↪money
hit to get the active display?  (i realize\nthis is a real subjective␣
 ↪question,
but i\'ve only played around with the\nmachines in a computer store␣
 ↪breifly and
figured the opinions of somebody\nwho actually uses the machine daily might
prove helpful).\n\n* how well does hellcats perform?  ;)\n\nthanks a bunch␣
 ↪in
advance for any info - if you could email, i\'ll post a\nsummary (news␣
 ↪reading
time is at a premium with finals just around the\ncorner… :( )\n--\nTom␣
 ↪Willis
\\  twillis@ecn.purdue.edu    \\    Purdue Electrical Engineering']
```

[4]:
```
vectorizer = TfidfVectorizer()
vectorizer.fit(data_train.data + data_test.data)
```

[4]:
```
TfidfVectorizer(analyzer='word', binary=False, decode_error='strict',
                dtype=<class 'numpy.float64'>, encoding='utf-8',
                input='content', lowercase=True, max_df=1.0,␣
 ↪max_features=None,
                min_df=1, ngram_range=(1, 1), norm='l2', preprocessor=None,
                smooth_idf=True, stop_words=None, strip_accents=None,
                sublinear_tf=False, token_pattern='(?u)\\b\\w\\w+\\b',
                tokenizer=None, use_idf=True, vocabulary=None)
```

[5]:
```
X_train = vectorizer.transform(data_train.data)
X_test = vectorizer.transform(data_test.data)

y_train = data_train.target
y_test = data_test.target
```

[6]:
```
X_train
```

[6]:
```
<11314x152843 sparse matrix of type '<class 'numpy.float64'>'
        with 1467517 stored elements in Compressed Sparse Row format>
```

[7]:
```
X_test
```

[7]:
```
<7532x152843 sparse matrix of type '<class 'numpy.float64'>'
        with 951914 stored elements in Compressed Sparse Row format>
```

```
[8]: def test(model):
         print(model)
         model.fit(X_train, y_train)
         print("accuracy:", accuracy_score(y_test, model.predict(X_test)))
```

```
[9]: test(LogisticRegression(solver='lbfgs', multi_class='auto'))
```

```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                   intercept_scaling=1, l1_ratio=None, max_iter=100,
                   multi_class='auto', n_jobs=None, penalty='l2',
                   random_state=None, solver='lbfgs', tol=0.0001, verbose=0,
                   warm_start=False)
accuracy: 0.774429102496017
```

```
[10]: test(LinearSVC())
```

```
LinearSVC(C=1.0, class_weight=None, dual=True, fit_intercept=True,
          intercept_scaling=1, loss='squared_hinge', max_iter=1000,
          multi_class='ovr', penalty='l2', random_state=None, tol=0.0001,
          verbose=0)
accuracy: 0.8048327137546468
```

```
[11]: test(MultinomialNB())
```

```
MultinomialNB(alpha=1.0, class_prior=None, fit_prior=True)
accuracy: 0.72623473181094
```

```
[12]: test(ComplementNB())
```

```
ComplementNB(alpha=1.0, class_prior=None, fit_prior=True, norm=False)
accuracy: 0.8089484864577802
```

```
[13]: test(BernoulliNB())
```

```
BernoulliNB(alpha=1.0, binarize=0.0, class_prior=None, fit_prior=True)
accuracy: 0.5371747211895911
```

Вывод Метод Complement Naive Bayes, ожидаемо, лучше всего решает поставленную задачу многоклассовой классификации в условиях дисбаланса классов, но LinearSVC также показал отличный результат и практический не уступил методу Complement Naive Bayes.