1. Рубежный контроль №2

Лещев Артем Олегович, группа ИУ5-24М. Вариант №1.

1.1. Задание

Необходимо решить задачу классификации текстов на основе любого выбранного Вами датасета. 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 один из классификаторов, не относящихся к наивным Байесовским методам (например, `LogisticRegression`), 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`).

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

1.2. Решение

1.2.1. Загрузка и предобработка данных

[1]:
```
from sklearn.datasets import fetch_20newsgroups
from sklearn.feature_extraction.text import TfidfVectorizer
```

[2]:
```
newsgroups_train = fetch_20newsgroups(subset='train', remove=('headers',␣
↪'footers'))
newsgroups_test = fetch_20newsgroups(subset='test', remove=('headers',␣
↪'footers'))
```

[3]:
```
vectorizer = TfidfVectorizer()
vectorizer.fit(newsgroups_train.data + newsgroups_test.data)
```

[3]:
```
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)
```

[4]:
```
X_train = vectorizer.transform(newsgroups_train.data)
X_test = vectorizer.transform(newsgroups_test.data)

y_train = newsgroups_train.target
y_test = newsgroups_test.target
```

1.2.2. Обучение моделей

[5]:
```
from sklearn.metrics import accuracy_score
```

[6]:
```
def test(model):
    print(model)
    model.fit(X_train, y_train)
    print("accuracy:", accuracy_score(y_test, model.predict(X_test)))
```

[7]:
```
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import MultinomialNB, ComplementNB, BernoulliNB
```

[8]:
```
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
```

[9]:
```
test(MultinomialNB())
```

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

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

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

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

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

1.2.3. Вывод

Метод Complement Naive Bayes, ожидаемо, лучше всего решает поставленную задачу многоклассовой классификации в условиях дисбаланса классов.