Московский государственный технический университет им. Н.Э. Баумана

Факультет «Информатика и системы управления»

Кафедра «Системы обработки информации и управления»

Рубежный контроль №2

ИСПОЛНИТЕЛЬ:

Колпаков М. О.

Группа ИУ5-22М

_____

"__"_____ 2020 г.

Москва 2020

Рубежный контроль №2

Колпаков Максим Олегович, группа ИУ5-22М. Вариант №1.

Задание

Необходимо решить задачу классификации текстов на основе любого выбранного Вами да
бинарной или многоклассовой. Целевой признак из выбранного Вами датасета может име
является задача анализа тональности текста.

Необходимо сформировать признаки на основе `CountVectorizer` или `TfidfVectorizer`.

В качестве классификаторов необходимо использовать один из классификаторов, не отно
(например, `LogisticRegression`), а также Multinomial Naive Bayes (MNB), Complement Naive B
каждого метода необходимо оценить качество классификации с помощью хотя бы одной м
(например, `accuracy`).

Сделайте выводы о том, какой классификатор осуществляет более качественную классиф

+ Code + Text

Решение

Загрузка и предобработка данных

```
from sklearn.datasets import fetch_20newsgroups
from sklearn.feature_extraction.text import TfidfVectorizer
```

```
newsgroups_train = fetch_20newsgroups(subset='train', remove=('headers', 'footers'))
newsgroups_test = fetch_20newsgroups(subset='test', remove=('headers', 'footers'))
```

```
vectorizer = TfidfVectorizer()
vectorizer.fit(newsgroups_train.data + newsgroups_test.data)
```

```
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)
```

```
X_train = vectorizer.transform(newsgroups_train.data)
X_test = vectorizer.transform(newsgroups_test.data)
```

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

```
y_test = newsgroups_test.target
```

Обучение моделей

```
from sklearn.metrics import accuracy_score

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

from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import MultinomialNB, ComplementNB, BernoulliNB

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
```

```
test(MultinomialNB())
```

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

```
test(ComplementNB())
```

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

```
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 много дисбаланса классов.