Рубежный контроль №2 по курсу "Методы машинного обучения"

Корнеева Анна Павловна, ИУ5-21М

Вариант №1. Классификация текстов на основе методов наивного Байеса.

In [1]:

```
import numpy as np
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics import accuracy_score, balanced_accuracy_score
from sklearn.metrics import precision_score, recall_score, f1_score
from sklearn.metrics import classification_report, confusion_matrix
from sklearn.datasets import fetch_20newsgroups
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import MultinomialNB, ComplementNB, BernoulliNB
import spacy
```

In [2]:

```
def get_train_test(target_names):
    '''
    Преобразует корпус текстов в матрицу признаков при помощи TfidfVectorizer.
    Предобработка текстов - приведение к нижнему регистру, удаление стоп-слов,
    удаление редких и незначащих, т.е. частотных, слов, лемматизация.
    '''
    # получение обучающего и тестового датасетов
    newsgroups_train = fetch_20newsgroups(
        subset='train', remove=('headers', 'footers', 'quotes'), categories=target_names
)
    newsgroups_test = fetch_20newsgroups(
        subset='test', remove=('headers', 'footers', 'quotes'), categories=target_names)

    # преобразование датасетов в корпуса текстов
    train_data_raw = newsgroups_train['data']
    test_data_raw = newsgroups_test['data']

    nlp = spacy.load('en_core_web_sm', disable=['parser', 'ner'])

    def lemmatize_text(text):
        '''
        Производит лемматизацию токенов.
        '''
        processed = nlp(text)
        return ' '.join([token.lemma_ for token in processed])

    # лемматизация
    train_data = [lemmatize_text(text.lower()) for text in train_data_raw]
    test_data = [lemmatize_text(text.lower()) for text in test_data_raw]

    # преобразование корпусов текстов в матрицы признаков
    vectorizer = TfidfVectorizer(stop_words='english', min_df=0.001, max_df=0.5)
    X_train = vectorizer.fit_transform(train_data)
    X_test = vectorizer.transform(test_data)

    # метки для обучающих и тестовых данных
    y_train = newsgroups_train['target']
    y_test = newsgroups_test['target']

    return X_train, X_test, y_train, y_test
```

In [3]:

```
target_names = ['comp.graphics', 'rec.sport.baseball', 'soc.religion.christian', 'talk.po
```

```
litics.mideast']
%time X_train, X_test, y_train, y_test = get_train_test(target_names)
```

```
Wall time: 1min 30s
```

In [4]:

```
def test_classifiers(classifiers, X_train, X_test, y_train, y_test, target_names):
    '''
    Оценивает качество работы классификаторов на тестовой выборке.
    '''
    for clf_class in classifiers:
        clf = clf_class()
        clf.fit(X_train, y_train)
        y_pred = clf.predict(X_test)

        report = classification_report(y_test, y_pred, target_names=target_names)

        print(f'Classifier {clf}')
        print(report)
        print('-' * 63)
```

Все классификаторы будем использовать со стандартными гиперпараметрами.

In [5]:

```
classifiers = [LogisticRegression, MultinomialNB, ComplementNB, BernoulliNB]
%time test_classifiers(classifiers, X_train, X_test, y_train, y_test, target_names)
```

```
Classifier LogisticRegression()
                        precision    recall  f1-score   support

         comp.graphics       0.93      0.95      0.94       389
     rec.sport.baseball       0.86      0.94      0.90       397
soc.religion.christian       0.92      0.90      0.91       398
 talk.politics.mideast       0.96      0.86      0.91       376

              accuracy                           0.91      1560
             macro avg       0.92      0.91      0.91      1560
          weighted avg       0.92      0.91      0.91      1560

---------------------------------------------------------------
Classifier MultinomialNB()
                        precision    recall  f1-score   support

         comp.graphics       0.94      0.94      0.94       389
     rec.sport.baseball       0.96      0.87      0.91       397
soc.religion.christian       0.79      0.97      0.87       398
 talk.politics.mideast       0.98      0.84      0.91       376

              accuracy                           0.91      1560
             macro avg       0.92      0.91      0.91      1560
          weighted avg       0.92      0.91      0.91      1560

---------------------------------------------------------------
Classifier ComplementNB()
                        precision    recall  f1-score   support

         comp.graphics       0.86      0.97      0.91       389
     rec.sport.baseball       0.95      0.89      0.92       397
soc.religion.christian       0.90      0.93      0.91       398
 talk.politics.mideast       0.98      0.88      0.93       376

              accuracy                           0.92      1560
             macro avg       0.92      0.92      0.92      1560
          weighted avg       0.92      0.92      0.92      1560

---------------------------------------------------------------
Classifier BernoulliNB()
                        precision    recall  f1-score   support
```

```
         comp.graphics       0.84      0.93      0.88       389
    rec.sport.baseball       0.64      0.94      0.76       397
soc.religion.christian       0.83      0.70      0.76       398
 talk.politics.mideast       0.97      0.54      0.70       376

              accuracy                           0.78      1560
             macro avg       0.82      0.78      0.77      1560
          weighted avg       0.82      0.78      0.78      1560

------------------------------------------------------------
Wall time: 708 ms
```

Классы в исследуемом датасете сбалансированы**.**

Наибольшая точность **(Accuracy) - 0.92 -** получается при использовании метода **Complement Naive Bayes.** Чуть меньшая **- 0.91 -** и примерно одинаковая точность у **Ligistic Regression** и **Multinomial Naive Bayes.** Наконец**,** худшая **- 0.78 -** получается при использовании **Bernoulli Naive Bayes,** причем точность значительно хуже**,** чем у остальных методов**.**
Аналогичная ситуация при оценивании **f1-score** по различным классам**:** у **ComplementNB** и **Logistic Regression** значение **f1-score** для всех классов не менее **0.9,** у **MultinomialNB** значение метрики для одного из классов уменьшилось до **0.87,** а для **BernoulliNB** все значения **f1-score** лежат в диапазоне от **0.70** до **0.88.**

Для достижения лучших результатов можно воспользоваться следующими методами**:**

- более точная предобработка данных**;**
- подбор гиперпараметров моделей**.**