РК2 Фадеев А.А. ИУ5-64

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

Данный вариант выполняется на основе материалов лекции.

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).

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

In [13]:

```
import numpy as np
import pandas as pd
from typing import Dict, Tuple
from scipy import stats
from sklearn.naive_bayes import GaussianNB, MultinomialNB, ComplementNB, BernoulliNB
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, balanced_accuracy_score
from sklearn.metrics import precision_score, recall_score, f1_score, classification_rep
ort
from sklearn.pipeline import Pipeline
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
sns.set(style="ticks")
```

In [14]:

```
data = pd.read_csv('./data/Question_Classification_Dataset.csv')
data.head()
```

Out[14]:

	Unnamed: 0	Questions	Category0	Category1	Category2
0	0	How did serfdom develop in and then leave Russ...	DESCRIPTION	DESC	manner
1	1	What films featured the character Popeye Doyle ?	ENTITY	ENTY	cremat
2	2	How can I find a list of celebrities ' real na...	DESCRIPTION	DESC	manner
3	3	What fowl grabs the spotlight after the Chines...	ENTITY	ENTY	animal
4	4	What is the full form of .com ?	ABBREVIATION	ABBR	exp

In [15]:

```
x_train, x_test, y_train, y_test = train_test_split(data['Questions'], data['Category0'
], test_size=0.5, random_state=1)
```

In [16]:

```
def accuracy_score_for_classes(
    y_true: np.ndarray,
    y_pred: np.ndarray) -> Dict[int, float]:
    """
    Вычисление метрики accuracy для каждого класса
    y_true - истинные значения классов
    y_pred - предсказанные значения классов
    Возвращает словарь: ключ - метка класса,
    значение - Accuracy для данного класса
    """
    # Для удобства фильтрации сформируем Pandas DataFrame
    d = {'t': y_true, 'p': y_pred}
    df = pd.DataFrame(data=d)
    # Метки классов
    classes = np.unique(y_true)
    # Результирующий словарь
    res = dict()
    # Перебор меток классов
    for c in classes:
        # отфильтруем данные, которые соответствуют
        # текущей метке класса в истинных значениях
        temp_data_flt = df[df['t']==c]
        # расчет accuracy для заданной метки класса
        temp_acc = accuracy_score(
            temp_data_flt['t'].values,
            temp_data_flt['p'].values)
        # сохранение результата в словарь
        res[c] = temp_acc
    return res

def print_accuracy_score_for_classes(
    y_true: np.ndarray,
    y_pred: np.ndarray):
    """
    Вывод метрики accuracy для каждого класса
    """
    accs = accuracy_score_for_classes(y_true, y_pred)
    if len(accs)>0:
        print('Метка \t Accuracy')
    for i in accs:
        print('{} \t {}'.format(i, accs[i]))
```

In [17]:

```
def sentiment(v, c):
    model = Pipeline(
        [("vectorizer", v),
         ("classifier", c)])
    model.fit(x_train, y_train)
    y_pred = model.predict(x_test)
    print_accuracy_score_for_classes(y_test, y_pred)
```

In [18]:

```
# Классификация с использованием логистической регресии
sentiment(TfidfVectorizer(), LogisticRegression(C=5.0, solver='lbfgs'))
```

```
c:\program files\python37\lib\site-packages\sklearn\linear_model\logistic.
py:469: FutureWarning: Default multi_class will be changed to 'auto' in 0.
22. Specify the multi_class option to silence this warning.
  "this warning.", FutureWarning)

Метка    Accuracy
ABBREVIATION     0.6470588235294118
DESCRIPTION      0.8071672354948806
ENTITY   0.7996768982229402
HUMAN    0.7781456953642384
LOCATION         0.872093023255814
NUMERIC          0.8653421633554084
```

In [19]:

```
sentiment(CountVectorizer(), MultinomialNB())
```

```
Метка    Accuracy
ABBREVIATION     0.0
DESCRIPTION      0.7047781569965871
ENTITY   0.7625201938610663
HUMAN    0.8294701986754967
LOCATION         0.8395348837209302
NUMERIC          0.7262693156732892
```

In [20]:

```
sentiment(TfidfVectorizer(), MultinomialNB())
```

```
Метка    Accuracy
ABBREVIATION     0.0
DESCRIPTION      0.7303754266211604
ENTITY   0.7512116316639742
HUMAN    0.8509933774834437
LOCATION         0.6976744186046512
NUMERIC          0.7064017660044151
```

In [21]:

```
#ComplementNB -развитие MNB, хорошо подходит для наборов данных с сильным дисбалансом к
лассов
sentiment(CountVectorizer(), ComplementNB())
```

```
Метка    Accuracy
ABBREVIATION     0.7058823529411765
DESCRIPTION      0.6382252559726962
ENTITY   0.6462035541195477
HUMAN    0.8410596026490066
LOCATION         0.9232558139534883
NUMERIC          0.8454746136865342
```

In [22]:

```
sentiment(TfidfVectorizer(), ComplementNB())
```

```
Метка    Accuracy
ABBREVIATION     0.7352941176470589
DESCRIPTION      0.6501706484641638
ENTITY   0.6429725363489499
HUMAN    0.847682119205298
LOCATION         0.9023255813953488
NUMERIC          0.8520971302428256
```

In [23]:

```
sentiment(CountVectorizer(binary=True), BernoulliNB())
```

```
Метка    Accuracy
ABBREVIATION     0.0
DESCRIPTION      0.810580204778157
ENTITY   0.7883683360258481
HUMAN    0.7682119205298014
LOCATION         0.30697674418604654
NUMERIC          0.5011037527593819
```

Для решение задачи классификации вопросов более качественно сработал метод Complement Naive Bayes (CNB).