Рубежный контроль №2 по курсу "Технологии машинного обучения"

Выполнила студентка 3 курса Курганова Александра, ИУ5-63

Вариант №1, датасет: "Text Classification"

Задание: необходимо решить задачу классификации текстов на основе выбранного датасета. классификация может быть бинарной или многоклассовой. целевой признак из выбранного датасета может иметь любой физический смысл. 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 выбранном наборе данных.

In [435]:
```
import pandas as pd
import numpy as np
import warnings
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer
from sklearn.pipeline import Pipeline
from sklearn.neighbors import KNeighborsRegressor, KNeighborsClassifier
from sklearn.metrics import accuracy_score
from sklearn.naive_bayes import GaussianNB, MultinomialNB, ComplementNB, BernoulliNB
```

первичный анализ датасета

датасет о спаме: категория текста и сам текст

In [436]:
```
data = pd.read_csv("spam.csv")
```

In [437]:
```
data.head()
```

Out[437]:

	Category	Message
0	ham	Go until jurong point, crazy.. Available only ...
1	ham	Ok lar... Joking wif u oni...
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...
3	ham	U dun say so early hor... U c already then say...
4	ham	Nah I don't think he goes to usf, he lives aro...

In [438]:
```
data.shape
```

Out[438]:
```
(5572, 2)
```

In [439]:
```
data.dtypes
```

Out[439]:
```
Category    object
Message     object
dtype: object
```

In [440]:
```
data.isnull().sum()
```

Out[440]:
```
Category    0
Message     0
dtype: int64
```

In [441]:
```
xtrain, xtest, ytrain, ytest = train_test_split(data['Message'], data['Category'], test_size=0.3, random_state=2)
```

In [442]:
```
def fit_and_predict_and_score(v, c):
    model = Pipeline(
        [("vectorizer", v),
         ("classifier", c)])
    model.fit(xtrain, ytrain)
    ypredtrain = model.predict(xtrain)
    ypredtest = model.predict(xtest)
    # оценивание качество модели классификации
    print('train accuracy(%) {}'.format(accuracy_score(ytrain, ypredtrain) * 100))
    print('test accuracy(%) {}'.format(accuracy_score(ytest, ypredtest) * 100))
```

logistic regression

In [443]:
```
fit_and_predict_and_score(TfidfVectorizer(), LogisticRegression(C=15))
```

```
train accuracy(%) 99.97435897435898
test accuracy(%) 97.78708133971293
```

метод MultinomialNB

In [444]:
```
fit_and_predict_and_score(TfidfVectorizer(), MultinomialNB())
```

```
train accuracy(%) 97.17948717948718
test accuracy(%) 95.15550239234449
```

метод ComplementNB

In [445]:
```
fit_and_predict_and_score(TfidfVectorizer(), ComplementNB())
```

```
train accuracy(%) 98.58974358974359
test accuracy(%) 96.94976076555024
```

метод BernoulliNB

In [446]:
```
fit_and_predict_and_score(TfidfVectorizer(), BernoulliNB())
```

```
train accuracy(%) 98.92307692307692
test accuracy(%) 96.88995215311004
```

можно сделать выводы, что:

- из классификаторов, не относящаяся к наивным Байесовским методам линейная регрессия работает лучше всего (97.78708133971293),
- из наивных Байесовских методов (Multinomial Naive Bayes (MNB), Complement Naive Bayes (CNB), Bernoulli Naive Bayes) лучше всего работает Complement Naive Bayes (CNB) (96.94976076555024).