МГТУ им. Н. Э. Баумана, кафедра ИУ5

курс “Технологии машинного обучения”

Рубежный контроль №2

«Технологии использования и оценки моделей машинного обучения»

ВЫПОЛНИЛ:

Пученков Д.О.

Группа: ИУ5-61Б

Вариант: 20

ПРОВЕРИЛ:

Гапанюк Ю.Е.

Москва, 2020 г.

Задача №1:

Необходимо решить задачу классификации текстов на основе любого выбранного Вами датасета (кроме примера, который рассматривался в лекции). 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 два классификатора, не относящихся к наивным Байесовским методам (например, LogisticRegression, LinearSVC), 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.

Датасет: https://www.kaggle.com/andradaolteanu/rickmorty-scripts

Выполненная работа:

```
import pandas as pd
import numpy as np
import seaborn as sns
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.metrics import f1_score, precision_score
from sklearn.linear_model import LogisticRegression
import matplotlib.pyplot as plt
from sklearn.naive_bayes import MultinomialNB, ComplementNB, BernoulliNB
from sklearn.metrics import accuracy_score
from sklearn.svm import LinearSVC
from sklearn.feature_extraction.text import TfidfVectorizer

%matplotlib inline
sns.set(style="ticks")
```

```
data = pd.read_csv('../datasets/RickAndMortyScripts.csv')
data
```

	index	season no.	episode no.	episode name	name	line
0	0	1	1	Pilot	Rick	Morty! You gotta come on. Jus'... you gotta co...
1	1	1	1	Pilot	Morty	What, Rick? What's going on?
2	2	1	1	Pilot	Rick	I got a surprise for you, Morty.
3	3	1	1	Pilot	Morty	It's the middle of the night. What are you tal...
4	4	1	1	Pilot	Rick	Come on, I got a surprise for you. Come on, h...
...	...	...	...	...	...	...
1900	2483	3	7	Tales From the Citadel	Morty	That was amazing!
1901	2484	3	7	Tales From the Citadel	Rick	Got some of that mermaid puss!
1902	2485	3	7	Tales From the Citadel	Morty	I'm really hoping it wasn't a one-off thing an...
1903	2486	3	7	Tales From the Citadel	Rick	Pssh! Not at all, Morty. That place will never...
1904	2487	3	7	Tales From the Citadel	Morty	Whoo! Yeah! Yeaah! Ohhh, shit!

1905 rows × 6 columns

```
data = data.drop(columns = ['season no.', 'episode no.','episode name'])
```

```
data.columns
```

```
Index(['index', 'name', 'line'], dtype='object')
```

```
data['name'].value_counts()
```

```
Rick                    420
Morty                   347
Beth                    148
Jerry                   132
Summer                   97
Pickle Rick              77
Supernova                44
Cop Morty                34
All Ricks                32
Mr. Goldenfold           28
President                27
Cop Rick                 26
Testicle Monster A       26
Principal Vagina         25
Cornvelious Daniel       22
Snuffles                 22
Drunk Rick               21
Dr. Wong                 21
Agency Director          20
Alan                     19
Candidate Morty          18
Vance                    17
Scary Terry              17
Jessica                  16
Million Ants             15
All Mortys               15
Ice-T                    13
Morty 2                  13
All Summers              13
Riq IV                   13
Alien Doctor             12
Campaign Manager Morty   12
Lizard Morty             11
Cromulon                 10
Brad                     10
Slick                    10
Nathan                   10
Birdperson                9
Glasses Morty             9
Rick J-22                 9
Young Rick                9
Vet                       9
Morty 1                   8
Announcer                 8
Mrs. Pancakes             8
Teacher Rick              8
Narrator                  8
Summer 1                  7
Name: name, dtype: int64
```

```
data = data[data['name'].isin(['Rick', 'Morty', 'Beth', 'Jerry', 'Summer'])]
data
```

	index	name	line
0	0	Rick	Morty! You gotta come on. Jus'... you gotta co...
1	1	Morty	What, Rick? What's going on?
2	2	Rick	I got a surprise for you, Morty.
3	3	Morty	It's the middle of the night. What are you tal...
4	4	Rick	Come on, I got a surprise for you. Come on, h...
...	...	...	...
1900	2483	Morty	That was amazing!
1901	2484	Rick	Got some of that mermaid puss!
1902	2485	Morty	I'm really hoping it wasn't a one-off thing an...
1903	2486	Rick	Pssh! Not at all, Morty. That place will never...
1904	2487	Morty	Whoo! Yeah! Yeaah! Ohhh, shit!

1144 rows × 3 columns

```
X = data.drop('name', axis=1)
Y = data['name']
```

```
X
```

	index	line
0	0	Morty! You gotta come on. Jus'... you gotta co...
1	1	What, Rick? What's going on?
2	2	I got a surprise for you, Morty.
3	3	It's the middle of the night. What are you tal...
4	4	Come on, I got a surprise for you. Come on, h...
...	...	...
1900	2483	That was amazing!
1901	2484	Got some of that mermaid puss!
1902	2485	I'm really hoping it wasn't a one-off thing an...
1903	2486	Pssh! Not at all, Morty. That place will never...
1904	2487	Whoo! Yeah! Yeaah! Ohhh, shit!

1144 rows × 2 columns

```
Y
```

```
0        Rick
1       Morty
2        Rick
3       Morty
4        Rick
        ...
1900    Morty
1901     Rick
1902    Morty
1903     Rick
1904    Morty
Name: name, Length: 1144, dtype: object
```

```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.25, random_state=1)
print('{}, {}'.format(X_train.shape, X_test.shape))
print('{}, {}'.format(Y_train.shape, Y_test.shape))
```

```
(858, 2), (286, 2)
(858,), (286,)
```

```
vectorizer = TfidfVectorizer()
vectorizer.fit(X_train + X_test)
```

```
index    float64
line      object
dtype: object
```

```
X_train
```

	index	line
239	267	Ohh, man. Oh, geez! Ohh.
87	95	Aw, geez. Okay. I guess I can skip history. Wh...
494	566	Oww!
538	611	You have dropped so many balls, man. Do you ev...
2	2	I got a surprise for you, Morty.
...	...	...
1028	1276	Hey, what's wrong Morty? Oh, you're worried ab...
1277	1652	McNuggets?
1686	2123	I think... no matter what we put on there, we ...
259	288	Roll over.
1606	2040	Because my epidermis is laced with a nanofiber...

```
X_train_vec = vectorizer.transform(X_train['line'])
X_test_vec = vectorizer.transform(X_test['line'])
```

```
X_train_vec.shape
```

```
(858, 2)
```

```
def test(model):
    print(model)
    model.fit(X_train_vec, Y_train)
    print("accuracy:", accuracy_score(Y_test, model.predict(X_test_vec)))
```

```
test(LogisticRegression(solver='lbfgs', multi_class='auto'))
```

```
LogisticRegression()
accuracy: 0.3706293706293706
```

```
test(LinearSVC())
```

```
LinearSVC()
accuracy: 0.3706293706293706
```

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

```
MultinomialNB()
accuracy: 0.3706293706293706
```

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

```
ComplementNB()
accuracy: 0.13986013986013987
```

```
test(BernoulliNB())
```

```
BernoulliNB()
accuracy: 0.3706293706293706
```

Вывод: Хуже всех отработал алгоритм Complement Naive Bayes. Другие алгоритмы отработали на одном уровне, это связано с большим дисбалансом классов и, в случае устранения этого дисбаланса, результаты работы алгоритмов должны улучшиться.