ИУ5-22М Киричков Е.Е. РК2

Задача классификации текстов

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 два варианта векторизации признаков - на основе CountVectorizer и на основе TfidfVectorizer.

В качестве классификаторов необходимо использовать два классификатора по варианту для Вашей группы: ИУ5-22М - RandomForestClassifier, LogisticRegression

Ввод [1]:
```
import pandas as pd
import time
from sklearn.ensemble import RandomForestClassifier
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
```

Ввод [2]:
```
# Загрузка данных
train_data = pd.read_csv('data/Corona_NLP_train.csv', encoding='latin1')
test_data = pd.read_csv('data/Corona_NLP_test.csv', encoding='latin1')
```

Ввод [3]:
```
train_data.head()
```

Out[3]:

	UserName	ScreenName	Location	TweetAt	OriginalTweet	Sentiment
0	3799	48751	London	16-03-2020	@MeNyrbie @Phil_Gahan @Chrisitv https://t.co/i...	Neutral
1	3800	48752	UK	16-03-2020	advice Talk to your neighbours family to excha...	Positive
2	3801	48753	Vagabonds	16-03-2020	Coronavirus Australia: Woolworths to give elde...	Positive
3	3802	48754	NaN	16-03-2020	My food stock is not the only one which is emp...	Positive
4	3803	48755	NaN	16-03-2020	Me, ready to go at supermarket during the #COV...	Extremely Negative

Ввод [4]:
```
train_data.shape
```

Out[4]:
```
(41157, 6)
```

Ввод [5]:
```
test_data.head()
```

Out[5]:

	UserName	ScreenName	Location	TweetAt	OriginalTweet	Sentiment
0	1	44953	NYC	02-03-2020	TRENDING: New Yorkers encounter empty supermar...	Extremely Negative
1	2	44954	Seattle, WA	02-03-2020	When I couldn't find hand sanitizer at Fred Me...	Positive
2	3	44955	NaN	02-03-2020	Find out how you can protect yourself and love...	Extremely Positive
3	4	44956	Chicagoland	02-03-2020	#Panic buying hits #NewYork City as anxious sh...	Negative
4	5	44957	Melbourne, Victoria	03-03-2020	#toiletpaper #dunnypaper #coronavirus #coronav...	Neutral

Ввод [6]:
```
test_data.shape
```

Out[6]:
```
(3798, 6)
```

Ввод [7]:
```
X_train = train_data['OriginalTweet']
y_train = train_data['Sentiment']

X_test = test_data['OriginalTweet']
y_test = test_data['Sentiment']
```

Ввод [8]:
```
def check_missing(data, name):
    missing = data.isnull().sum()
    print(f'У {name} {missing} пропущенных строк')
```

Ввод [9]:
```
check_missing(train_data, 'train_data')
check_missing(test_data, 'test_data')
check_missing(X_train, 'X_train')
check_missing(X_test, 'X_test')
check_missing(y_train, 'y_train')
check_missing(y_test, 'y_test')
```

```
У train_data UserName            0
ScreenName          0
Location         8590
TweetAt             0
OriginalTweet       0
Sentiment           0
dtype: int64 пропущенных строк
У test_data UserName           0
ScreenName          0
Location          834
TweetAt             0
OriginalTweet       0
Sentiment           0
dtype: int64 пропущенных строк
У X_train 0 пропущенных строк
У X_test 0 пропущенных строк
У y_train 0 пропущенных строк
У y_test 0 пропущенных строк
```

Ввод [10]:
```
# Векторизация с помощью CountVectorizer
count_vect = CountVectorizer()
X_train_counts = count_vect.fit_transform(X_train)
X_test_counts = count_vect.transform(X_test)

# Векторизация с помощью TfidfVectorizer
tfidf_vect = TfidfVectorizer()
X_train_tfidf = tfidf_vect.fit_transform(X_train)
X_test_tfidf = tfidf_vect.transform(X_test)
```

Ввод [11]:
```
def evaluate_model(vectorizer_name, vectorizer_train, vectorizer_test, model,
    start_time = time.time()
    obj_model = model
    obj_model.fit(vectorizer_train, y_train)
    predictions = obj_model.predict(vectorizer_test)

    accuracy = accuracy_score(y_test, predictions)
    duration = (time.time() - start_time) / 60

    print(f'Точность {vectorizer_name} + {model_name}: {accuracy:.4f}, время

# Для CountVectorizer
evaluate_model('CountVectorizer', X_train_counts, X_test_counts, RandomForest
evaluate_model('CountVectorizer', X_train_counts, X_test_counts, LogisticRegr

# Для TfidfVectorizer
evaluate_model('TfidfVectorizer', X_train_tfidf, X_test_tfidf, RandomForestCl
evaluate_model('TfidfVectorizer', X_train_tfidf, X_test_tfidf, LogisticRegres
```

```
Точность CountVectorizer + RandomForestClassifier: 0.4489, время обучения кл
ассификатора: 1.27 мин.
Точность CountVectorizer + LogisticRegression: 0.6087, время обучения класси
фикатора: 0.46 мин.
Точность TfidfVectorizer + RandomForestClassifier: 0.4342, время обучения кл
ассификатора: 1.09 мин.
Точность TfidfVectorizer + LogisticRegression: 0.5658, время обучения класси
фикатора: 0.18 мин.
```

Лучшее качество показала модель LogisticRegression с векторизацией CountVectorizer