Рубежный контроль №1 по курсу «Методы машинного обучения»

Подготовил: Студент группы ИУ5-25М Клюкин Н. А. 16.05.2024

Вариант задания

Номер группы	Классификатор 1	Классификатор 2
ИУ-25М	SVC	LogisticRegression

Импорт библиотек

```
import numpy as np
import pandas as pd
from typing import Dict, Tuple
from scipy import stats
from IPython.display import Image
from sklearn.feature_extraction.text import CountVectorizer,
TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsRegressor,
KNeighborsClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import GridSearchCV, RandomizedSearchCV
from sklearn.metrics import accuracy_score, balanced_accuracy_score
from sklearn.metrics import precision_score, recall_score, f1_score,
classification_report
from sklearn.metrics import confusion_matrix
from sklearn.model_selection import cross_val_score
from sklearn.pipeline import Pipeline
from sklearn.metrics import mean_absolute_error, mean_squared_error,
mean_squared_log_error, median_absolute_error, r2_score
from sklearn.metrics import roc_curve, roc_auc_score
from sklearn.svm import SVC, NuSVC, LinearSVC, OneClassSVM, SVR,
NuSVR, LinearSVR
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
sns.set(style="ticks")
```

Подгрузка датасета и подготовительные действия

- Используем набор Depressive/Non-Depressive Tweets Data - Депрессивные/недепрессивные твиты в период с декабря 2019 года по декабрь 2020 года

```
# Загрузка данных
df = pd.read_csv("datasets/clean_tweet.csv")

# Удаление первой колонки
df = df.drop(df.columns[0], axis=1)
df = df.rename(columns={'sentiment': 'value'})

df.head()
```

	text	value
0	text	0
1	rising cases of covid does not alarm me rising...	1
2	please vote for chicagoindiaresolution marking...	0
3	wishing all of you eidaladha hazrat ibrahim as...	1
4	daily coronavirus cases in india top for first...	1

```
df.shape
```

```
(134348, 2)
```

Датасет слишком большой, а также содержит пропуски. Возьмём часть и проведем очистку данных

```
df = df.sample(frac=0.05)
df = df.dropna()

X_train, X_test, y_train, y_test = train_test_split(df['text'],
df['value'], test_size=0.2, random_state=42)
```

Задание 1. Сформировать два варианта векторизации признаков

```
count_vectorizer = CountVectorizer()
X_train_count = count_vectorizer.fit_transform(X_train)

tfidf_vectorizer = TfidfVectorizer()
X_train_tfidf = tfidf_vectorizer.fit_transform(X_train)
```

Задание 2. Решение задачи классификации текстов

2.1. Решение задачи классификации текстов с использованием классификатора SVC

```
svc_classifier_tfidf = SVC()
svc_classifier_count = SVC()

svc_classifier_tfidf.fit(X_train_tfidf, y_train)
```

```
SVC()
```

```
svc_classifier_count.fit(X_train_count, y_train)
```

```
SVC()
```

2.2. Решение задачи классификации текстов с использованием классификатора LogisticRegression

```
lr_classifier_tfidf = LogisticRegression()
lr_classifier_count = LogisticRegression()

lr_classifier_tfidf.fit(X_train_tfidf, y_train)
```

```
LogisticRegression()
```

```
lr_classifier_count.fit(X_train_count, y_train)
```

```
LogisticRegression()
```

Подведение результатов

```
X_test_tfidf = tfidf_vectorizer.transform(X_test)

svc_accuracy = svc_classifier_tfidf.score(X_test_tfidf, y_test)
lr_accuracy = lr_classifier_tfidf.score(X_test_tfidf, y_test)

print(f'Точность для метода векторизации TfidfVectorizer через
классификатор SVC = {svc_accuracy}')
print(f'Точность для метода векторизации TfidfVectorizer через
классификатор Logistic Regression = {lr_accuracy}')
```

```
Точность для метода векторизации TfidfVectorizer через классификатор
SVC = 0.8377976190476191
Точность для метода векторизации TfidfVectorizer через классификатор
Logistic Regression = 0.8333333333333334
```

```
X_test_count = count_vectorizer.transform(X_test)

svc_accuracy = svc_classifier_count.score(X_test_count, y_test)
lr_accuracy = lr_classifier_count.score(X_test_count, y_test)

print(f'Точность для метода векторизации CountVectorizer через
классификатор SVC = {svc_accuracy}')
print(f'Точность для метода векторизации CountVectorizer через
классификатор Logistic Regression = {lr_accuracy}')
```

```
Точность для метода векторизации CountVectorizer через классификатор
SVC = 0.8132440476190477
Точность для метода векторизации CountVectorizer через классификатор
Logistic Regression = 0.8616071428571429
```

Таким образом, можно заключить, что наилучшие результаты были получены с использованием метода векторизации CountVectorizer с классификатором Logistic Regression. Вероятно это связано с тем, что:

- Метод векторизации CountVectorizer подсчитывает количество вхождений слов в документе. В следствие чего на основе частоты встречи определенных слов можно делать вывод о депрессивности твитов
- В свою очередь, TfidfVectorizer использует метод TF-IDF для выделения важности слов в документе. Это позволяет выделять определенные слова или фразы могут быть связаны с депрессией. Я предполагал, что этот метод будет давать наилучшие результаты, однако он давал наиболее стабильные, что вполне коррелирует с особенностью его работы (выделять триггерные слова)
- Использование Logistic Regression позволило достичь наилучших результатов, поскольку он хорошо работает в задачах бинарной классификации (когда требуется оценить вероятность принадлежности объекта к одному из двух классов)