Московский государственный технический
университет им. Н. Э. Баумана
Факультет «Информатика и системы управления»

Кафедра «Системы обработки информации и управления»

Курс «Технологии машинного обучения»

Отчет по рубежному контролю №2
Технологии использования и оценки моделей машинного обучения

Группа: ИУ5-62Б

Студент: Селедкина А.С.

Преподаватель: Гапанюк Ю.Е.

Москва, 2020 г.

Вариант 15

Описание задания

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

Текст программы и примеры выполнения

Будем использовать датасет, содержащий отзывы на отели и оценку их тональности (happy — положительный, not happy — отрицательный): https://www.kaggle.com/harmanpreet93/hotelreviews.

```
import numpy as np
import pandas as pd
from typing import Dict, Tuple
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer
from sklearn.naive_bayes import GaussianNB, MultinomialNB, ComplementNB, BernoulliNB
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsRegressor, KNeighborsClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, balanced_accuracy_score
from sklearn.pipeline import Pipeline
from sklearn.svm import SVC, NuSVC, LinearSVC, OneClassSVM, SVR, NuSVR, LinearSVR
import matplotlib.pyplot as plt
%matplotlib inline
```

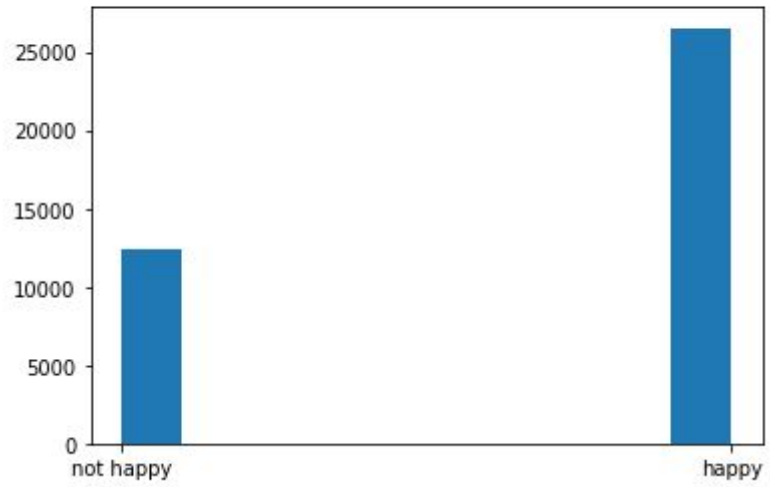
```
data = pd.read_csv('data/hotel-reviews.csv')
data.head()
```

	User_ID	Description	Browser_Used	Device_Used	Is_Response
0	id10326	The room was kind of clean but had a VERY stro...	Edge	Mobile	not happy
1	id10327	I stayed at the Crown Plaza April -- - April -...	Internet Explorer	Mobile	not happy
2	id10328	I booked this hotel through Hotwire at the low...	Mozilla	Tablet	not happy
3	id10329	Stayed here with husband and sons on the way t...	InternetExplorer	Desktop	happy
4	id10330	My girlfriends and I stayed here to celebrate ...	Edge	Tablet	not happy

```
data.shape
```

```
(38932, 5)
```

```
# Распределение классов целевого признака
plt.hist(data['Is_Response'])
plt.show()
```



Формирование признаков

```
# Сформируем общий словарь для обучения моделей из обучающей и тестовой выборки
vocab_list = data['Description'].tolist()
vocab_list[1:3]
```

```
["I stayed at the Crown Plaza April -- - April --, ----. The staff was friendly and attentive. The elevators are t
iny (about -' by -'). The food in the restaurant was delicious but priced a little on the high side. Of course thi
s is Washington DC. There is no pool and little for children to do. My room on the fifth floor had two comfortable
beds and plenty of space for one person. The TV is a little small by todays standards with a limited number of cha
nnels. There was a small bit of mold in the bathtub area that could have been removed with a little bleach. It app
eared the carpets were not vacummed every day. I reported a light bulb was burned out. It was never replaced. Ice
machines are on the odd numbered floors, but the one on my floor did not work. I encountered some staff in the ele
vator one evening and I mentioned the ice machine to them. Severel hours later a maid appeared at my door with ice
and two mints. I'm not sure how they knew what room I was in. That was a little unnerving! I would stay here again
for business, but would not come here on vacation.",
 'I booked this hotel through Hotwire at the lowest price I could find. When we got there the front desk manager g
ave us a """""""""""""""""""""""""""""""""smoking""""""""""""""""""""""""""""""""" room. I argued that I have a litt
le baby and I would not have booked the room had I known it was smoking. The manager would not hear anything furth
er and told me that Hotwire books the cheapest rooms that are available. So, from the get go I was very unhappy.\r
\nAfter a great deal of persuasion and discussion, I got a nonsmoking room. Thereafter the room had the most minim
al amenities. Besides the great location (near Dupont Circle), there was not much to say about this overpriced hot
el. The room was small and in OK condition. The bathroom was small with a tub. The bathroom amenities were also mi
nimal. We did not have a fridge or a microwave and had to again rent a fridge from the staff (to keep baby thing
s).\r\nThe parking costs $-- per day so it is best not to drive here. Also the breakfast is not included. The lobb
y is very small and feels old. The only thing available is coffee in the lobby which is decent.\r\nAll in all, poo
r service, minimal amenities, small rooms, small bathrooms, no view, but great location. Some distance from the me
tro (either McPherson station or Dupont station). Try to look for better if available.']
```

```
vocab_vect = CountVectorizer()
vocab_vect.fit(vocab_list)
corpus_vocab = vocab_vect.vocabulary_
print('Количество сформированных признаков - {}'.format(len(corpus_vocab)))
```

```
Количество сформированных признаков - 46016
```

```
# Первые 10 признаков
for i in list(corpus_vocab)[0:9]:
    print('{}={}'.format(i, corpus_vocab[i]))
```

```
the=40646
room=34587
was=44401
kind=22449
of=27927
clean=7665
but=5772
had=18401
very=43761
```

Решение задачи анализа тональности

```
x_train, x_test, y_train, y_test = train_test_split(data['Description'], data['Is_Response'],
                                                    test_size=0.3, random_state=1)
```

```python
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]))
```

```python
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)
```

```python
sentiment(CountVectorizer(), KNeighborsClassifier())
```

```
Метка    Accuracy
happy    0.9407025053506232
not happy        0.27535456248327533
```

```
sentiment(TfidfVectorizer(), LinearSVC())
```

```
Метка    Accuracy
happy    0.9223215409794788
not happy        0.7923468022477923
```

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

```
Метка    Accuracy
happy    0.9143900289563137
not happy        0.7511372758897511
```

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

```
Метка    Accuracy
happy    0.9925720760417978
not happy        0.36954776558736957
```

```
sentiment(CountVectorizer(), ComplementNB())
```

```
Метка    Accuracy
happy    0.9035628855596123
not happy        0.7647845865667647
```

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

```
Метка    Accuracy
happy    0.9714213773133576
not happy        0.5731870484345731
```

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

```
Метка    Accuracy
happy    0.857736371647992
not happy        0.6165373294086165
```

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

```
Метка    Accuracy
happy    0.857736371647992
not happy        0.6165373294086165
```

На выбранном наборе данных наиболее качественную классификацию осуществили LinearSVC, MultinomialNB и ComplementNB.