Московский государственный технический университет им. Н.Э. Баумана.

Факультет «Информатика и управление»

Кафедра ИУ5. Курс «Технологии машинного обучения»

Отчет по лабораторной работе №3

Выполнила:		Проверил:
студентка группы ИУ5-61		преподаватель каф. ИУ5
Абросимова Надежда		Гапанюк Ю.Е.
Подпись и дата:		Подпись и дата:

г. Москва, 2019 г.

Задание

Выбрать набор данных (датасет), содержащий категориальные признаки и пропуски в данных. Для выполнения следующих пунктов можно использовать несколько различных наборов данных (один для обработки пропусков, другой для категориальных признаков и т.д.)
Для выбранного датасета (датасетов) на основе материалов лекции решить следующие задачи:
обработку пропусков в данных;
кодирование категориальных признаков;
масштабирование данных.

Текст программы

```
import numpy as np
import pandas as pd
 import seaborn as sns
import sklearn
import matplotlib.pyplot as plt
 %matplotlib inline
sns.set(style="ticks")
data=pd.read_csv('ToyProducts.csv', sep=",")
data.shape
Out[4]:
(10000, 17)
data.dtypes
Out[5]:
uniq_id                                      object
product_name                                 object
manufacturer                                 object
price                                        object
number_available_in_stock                    object
number_of_reviews                            object
number_of_answered_questions                float64
average_review_rating                        object
amazon_category_and_sub_category             object
customers_who_bought_this_item_also_bought   object
description                                  object
product_information                          object
product_description                          object
items_customers_buy_after_viewing_this_item  object
customer_questions_and_answers               object
customer_reviews                             object
sellers                                      object
dtype: object
data.isnull().sum()
Out[6]:
uniq_id                                         0
product_name                                    0
manufacturer                                    7
price                                        1435
number_available_in_stock                    2500
number_of_reviews                              18
number_of_answered_questions                  765
```

```
average_review_rating                              18
amazon_category_and_sub_category                  690
customers_who_bought_this_item_also_bought       1062
description                                       651
product_information                                58
product_description                               651
items_customers_buy_after_viewing_this_item      3065
customer_questions_and_answers                   9086
customer_reviews                                   21
sellers                                          3082
dtype: int64
total_count=data.shape[0]
print('Всего строк:{}'.format(total_count))
Всего строк:10000
In [9]:
#Обработка пропусков
#Удаление колонок, содержащих пустые значения
data_new1=data.dropna(axis=1, how='any')
(data.shape, data_new1.shape)
Out[9]:
((10000, 17), (10000, 2))
In [10]:
#Удаление строк, содержащих пустые значения
data_new2=data.dropna(axis=0, how='any')
(data.shape, data_new2.shape)
Out[10]:
((10000, 17), (511, 17))
In [11]:
#Заполнение пропущенных значений нулями
data_new3=data.fillna(0)
# Выберем числовые колонки с пропущенными значениями
# Цикл по колонкам датасета
num_cols = []
for col in data.columns:
    # Количество пустых значений
    temp_null_count = data[data[col].isnull()].shape[0]
    dt = str(data[col].dtype)
    if temp_null_count>0 and (dt=='float64' or dt=='int64'):
        num_cols.append(col)
        temp_perc = round((temp_null_count / total_count) * 100.0, 2)
        print('Колонка {}. Тип данных {}. Количество пустых значений {}, {}%.
'.format(col, dt, temp_null_count, temp_perc))
Колонка number_of_answered_questions. Тип данных float64. Количество пустых з
начений 765, 7.65%.
In [13]:
# Фильтр по колонкам с пропущенными значениями
data_num = data[num_cols]

# Фильтр по пустым значениям поля
data[data['number_of_answered_questions'].isnull()]
```

```
# Запоминаем индексы строк с пустыми значениями
flt_index = data[data['number_of_answered_questions'].isnull()].index
flt_index
```

Out[15]:

```
Int64Index([ 128,  199,  200,  201,  202,  203,  204,  205,  206,  207,
             ...
             9797, 9798, 9799, 9957, 9958, 9959, 9960, 9961, 9962, 9963],
            dtype='int64', length=765)
```

In [16]:

```
# Проверяем что выводятся нужные строки
data[data.index.isin(flt_index)]
```

	uniq_id	product_name	manufacturer	price	number_available_in_stock	number_
128	aedf496c4f0594f1814f301db907ffad	Kato N Gauge Train Set Case (Kato PlaRail Mode...	Kato	£11.04	39 new	2
199	159b1371be56ec94a1568647669416b3	Smasha Ballz Ninjaaah	Smasha-Ballz	£15.84	6 new	23
200	eb85d6369c891422a89137b0008f1818	Moomins - 6.5 Inch Moominpappa Soft Toy - 20056	Moomins	£7.29	2 new	1
201	5e9618d43e14edff1c4bb5cce3d1d2d2	Classic Cuddly Paddington Bear by Rainbow Desi...	Paddington Bear	£14.60	21 new	41

```
data_num_number_of_answered_questions = data_num[['number_of_answered_questions']]
data_num_number_of_answered_questions.head()
```

	number_of_answered_questions
0	1.0
1	1.0
2	2.0
3	2.0
4	2.0

:

```
from sklearn.impute import SimpleImputer
from sklearn.impute import MissingIndicator
# Фильтр для проверки заполнения пустых значений
indicator = MissingIndicator()
mask_missing_values_only = indicator.fit_transform(data_num_number_of_answered_questions)
mask_missing_values_only
```

Out[21]:

```
array([[False],
```

```
       [False],
       [False],
       ...,
       [False],
       [False],
       [False]])
```

In [22]:

```
strategies=['mean', 'median','most_frequent']

def test_num_impute(strategy_param): imp_num =
SimpleImputer(strategy=strategy_param) data_num_imp =
imp_num.fit_transform(data_num_number_of_answered_questions) return
data_num_imp[mask_missing_values_only]

strategies[0], test_num_impute(strategies[0])
strategies[1], test_num_impute(strategies[1])
strategies[2], test_num_impute(strategies[2])

# Более сложная функция, которая позволяет задавать колонку и вид
импьютации def test_num_impute_col(dataset, column, strategy_param):
temp_data = dataset[[column]] indicator = MissingIndicator()
mask_missing_values_only = indicator.fit_transform(temp_data) imp_num =
SimpleImputer(strategy=strategy_param) data_num_imp =
imp_num.fit_transform(temp_data) filled_data =
data_num_imp[mask_missing_values_only] return column, strategy_param,
filled_data.size, filled_data[0], filled_data[filled_data.size-1]
data[['number_of_answered_questions']].describe()
```

	number_of_answered_questions
count	9235.000000
mean	1.834976
std	2.517268
min	1.000000
25%	1.000000
50%	1.000000
75%	2.000000
max	39.000000

```
test_num_impute_col(data, 'number_of_answered_questions', strategies[0])
```

Out[29]:

```
('number_of_answered_questions',
 'mean',
 765,
 1.8349756361667569,
 1.8349756361667569)
```

In [30]:

```
test_num_impute_col(data, 'number_of_answered_questions', strategies[1])
```

Out[30]:

```
('number_of_answered_questions', 'median', 765, 1.0, 1.0)
```

In [31]:

```
test_num_impute_col(data, 'number_of_answered_questions', strategies[2])
```

Out[31]:

```
('number_of_answered_questions', 'most_frequent', 765, 1.0, 1.0)
```

In [ ]:

Обработка пропусков в категориальных данных

In [32]:

```
# Выберем категориальные колонки с пропущенными значениями
# Цикл по колонкам датасета
cat_cols = []
for col in data.columns:
    # Количество пустых значений
    temp_null_count = data[data[col].isnull()].shape[0]
    dt = str(data[col].dtype)
    if temp_null_count>0 and (dt=='object'):
        cat_cols.append(col)
        temp_perc = round((temp_null_count / total_count) * 100.0, 2)
        print('Колонка {}. Тип данных {}. Количество пустых значений {}, {}%.
'.format(col, dt, temp_null_count, temp_perc))
```

```
Колонка manufacturer. Тип данных object. Количество пустых значений 7, 0.07%.
Колонка price. Тип данных object. Количество пустых значений 1435, 14.35%.
Колонка number_available_in_stock. Тип данных object. Количество пустых значе
ний 2500, 25.0%.
Колонка number_of_reviews. Тип данных object. Количество пустых значений 18,
0.18%.
Колонка average_review_rating. Тип данных object. Количество пустых значений
18, 0.18%.
Колонка amazon_category_and_sub_category. Тип данных object. Количество пусты
х значений 690, 6.9%.
Колонка customers_who_bought_this_item_also_bought. Тип данных object. Количе
ство пустых значений 1062, 10.62%.
Колонка description. Тип данных object. Количество пустых значений 651, 6.51%
.
Колонка product_information. Тип данных object. Количество пустых значений 58
, 0.58%.
Колонка product_description. Тип данных object. Количество пустых значений 65
1, 6.51%.
Колонка items_customers_buy_after_viewing_this_item. Тип данных object. Колич
ество пустых значений 3065, 30.65%.
Колонка customer_questions_and_answers. Тип данных object. Количество пустых
значений 9086, 90.86%.
Колонка customer_reviews. Тип данных object. Количество пустых значений 21, 0
.21%.
Колонка sellers. Тип данных object. Количество пустых значений 3082, 30.82%.
```

```
cat_temp_data = data[['number_of_reviews']]
cat_temp_data.head()
```

```
cat_temp_data['number_of_reviews'].unique()
```

```
cat_temp_data[cat_temp_data['number_of_reviews'].isnull()].shape
```

Out[35]:

```
(18, 1)
```

```
# Импьютация наиболее частыми значениями
imp2 = SimpleImputer(missing_values=np.nan, strategy='most_frequent')
data_imp2 = imp2.fit_transform(cat_temp_data)
data_imp2
```

```
# Импьютация константой
imp3 = SimpleImputer(missing_values=np.nan, strategy='constant', fill_valu
e='!!!!')
data_imp3 = imp3.fit_transform(cat_temp_data)
data_imp3
```

Преобразование категориальных признаков в числовые

```
cat_enc = pd.DataFrame({'c1':data_imp2.T[0]})
le = LabelEncoder()
cat_enc_le = le.fit_transform(cat_enc['c1'])
ohe = OneHotEncoder()
cat_enc_ohe = ohe.fit_transform(cat_enc[['c1']])
```

Масштабирование данных

```
from sklearn.preprocessing import MinMaxScaler, StandardScaler, Normalizer
sc1 = MinMaxScaler()
sc1_data = sc1.fit_transform(data[['number_of_answered_questions']])
plt.hist(data['number_of_answered_questions'], 50)
plt.show()
```