Московский государственный технический университет имени Н.Э.Баумана

Кафедра «Системы обработки информации и управления»

О Т Ч Е Т

Лабораторная работа №3

по курсу «Методы машинного обучения»

« «Обработка пропусков в данных, кодирование категориальных признаков, масштабирование данных.»

Исполнитель: **Соболева Е.Д.**

группа ИУ5-11М

Проверил: **Гапанюк Ю.Е.**

Москва, 2020

Цель лабораторной работы:

изучение способов предварительной обработки данных для дальнейшего формирования моделей.

Задание:

Выбрать набор данных (датасет), содержащий категориальные признаки и пропуски в данных. Для выполне ния следующих пунктов можно использовать несколько различных наборов данных (один для обработки пр опусков, 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.

In [154]:

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
sns.set(style="ticks")
```

In [155]:

```
# Набор 1
data = pd.read_csv('googleplaystore.csv', sep=",")
# Набор 2
data1 = pd.read_csv('googleplaystore_user_reviews.csv', sep=",")
```

In [156]:

```
# Набор 1
data.shape
```

Out[156]:

```
(10841, 13)
```

In [157]:

```
# Набор 2
data1.shape
```

Out[157]:

```
(64295, 5)
```

In [158]:

```
# Набор 1
data.dtypes
```

Out[158]:

```
App                object
Category           object
Rating            float64
Reviews            object
Size               object
Installs           object
Type               object
Price              object
Content Rating     object
Genres             object
Last Updated       object
Current Ver        object
Android Ver        object
dtype: object
```

In [159]:

```
# Набор 2
data1.dtypes
```

Out[159]:

```
App                        object
Translated_Review          object
Sentiment                  object
Sentiment_Polarity        float64
Sentiment_Subjectivity    float64
dtype: object
```

In [160]:

```
# пропуски в наборе 1
data.isnull().sum()
```

Out[160]:

```
App                0
Category           0
Rating          1474
Reviews            0
Size               0
Installs           0
Type               1
Price              0
Content Rating     1
Genres             0
Last Updated       0
Current Ver        8
Android Ver        3
dtype: int64
```

In [161]:

```
# пропуски в наборе 2
data1.isnull().sum()
```

Out[161]:

```
App                          0
Translated_Review        26868
Sentiment                26863
Sentiment_Polarity       26863
Sentiment_Subjectivity   26863
dtype: int64
```

In [349]:

```
# Набор 1
data.head()
```

Out[349]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	C
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1	159	19M	10,000+	Free	0	Eve
1	Coloring book moana	ART_AND_DESIGN	3.9	967	14M	500,000+	Free	0	Eve
2	U Launcher Lite – FREE Live Cool Themes, Hide ...	ART_AND_DESIGN	4.7	87510	8.7M	5,000,000+	Free	0	Eve
3	Sketch - Draw & Paint	ART_AND_DESIGN	4.5	215644	25M	50,000,000+	Free	0	
4	Pixel Draw - Number Art Coloring Book	ART_AND_DESIGN	4.3	967	2.8M	100,000+	Free	0	Eve

In [350]:

```
# Набор 2
data1.head()
```

Out[350]:

	App	Translated_Review	Sentiment	Sentiment_Polarity	Sentiment_Subjectivity
0	10 Best Foods for You	I like eat delicious food. That's I'm cooking ...	Positive	1.00	0.533333
1	10 Best Foods for You	This help eating healthy exercise regular basis	Positive	0.25	0.288462
2	10 Best Foods for You	NaN	NaN	NaN	NaN
3	10 Best Foods for You	Works great especially going grocery store	Positive	0.40	0.875000
4	10 Best Foods for You	Best idea us	Positive	1.00	0.300000

In [351]:

```
total_count = data.shape[0]
print('Строки в наборе 1: {}'.format(total_count))
```

```
Строки в наборе 1: 10841
```

In [352]:

```
total_count1 = data1.shape[0]
print('Строки в наборе 2: {}'.format(total_count1))
```

```
Строки в наборе 2: 64295
```

Обработка пропусков

In [165]:

```
# Удаление колонок, содержащих пустые значения в наборе 1
data_new_1 = data.dropna(axis=1, how='any')
(data.shape, data_new_1.shape)
```

Out[165]:

```
((10841, 13), (10841, 8))
```

In [166]:

```
# Удаление колонок, содержащих пустые значения в наборе 2
data_new_11 = data1.dropna(axis=1, how='any')
(data1.shape, data_new_11.shape)
```

Out[166]:

```
((64295, 5), (64295, 1))
```

In [167]:

```
# Удаление строк, содержащих пустые значения в наборе 1
data_new_2 = data.dropna(axis=0, how='any')
(data.shape, data_new_2.shape)
```

Out[167]:

```
((10841, 13), (9360, 13))
```

In [168]:

```
data.head()
```

Out[168]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	C
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1	159	19M	10,000+	Free	0	Eve
1	Coloring book moana	ART_AND_DESIGN	3.9	967	14M	500,000+	Free	0	Eve
2	U Launcher Lite – FREE Live Cool Themes, Hide ...	ART_AND_DESIGN	4.7	87510	8.7M	5,000,000+	Free	0	Eve
3	Sketch - Draw & Paint	ART_AND_DESIGN	4.5	215644	25M	50,000,000+	Free	0	
4	Pixel Draw - Number Art Coloring Book	ART_AND_DESIGN	4.3	967	2.8M	100,000+	Free	0	Eve

In [169]:

```
# Удаление строк, содержащих пустые значения в наборе 2
data_new_21 = data1.dropna(axis=0, how='any')
(data1.shape, data_new_21.shape)
```

Out[169]:

```
((64295, 5), (37427, 5))
```

In [170]:

```
data1.head()
```

Out[170]:

	App	Translated_Review	Sentiment	Sentiment_Polarity	Sentiment_Subjectivity
0	10 Best Foods for You	I like eat delicious food. That's I'm cooking ...	Positive	1.00	0.533333
1	10 Best Foods for You	This help eating healthy exercise regular basis	Positive	0.25	0.288462
2	10 Best Foods for You	NaN	NaN	NaN	NaN
3	10 Best Foods for You	Works great especially going grocery store	Positive	0.40	0.875000
4	10 Best Foods for You	Best idea us	Positive	1.00	0.300000

In [171]:

```
# Заполнение всех пропущенных значений нулями в наборе 1
data_new_3 = data.fillna(0)
data_new_3.head()
```

Out[171]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	C
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1	159	19M	10,000+	Free	0	Eve
1	Coloring book moana	ART_AND_DESIGN	3.9	967	14M	500,000+	Free	0	Eve
2	U Launcher Lite – FREE Live Cool Themes, Hide ...	ART_AND_DESIGN	4.7	87510	8.7M	5,000,000+	Free	0	Eve
3	Sketch - Draw & Paint	ART_AND_DESIGN	4.5	215644	25M	50,000,000+	Free	0	
4	Pixel Draw - Number Art Coloring Book	ART_AND_DESIGN	4.3	967	2.8M	100,000+	Free	0	Eve

In [172]:

```
# Заполнение всех пропущенных значений нулями в наборе 1
data_new_31 = data1.fillna(0)
data_new_31.head()
```

Out[172]:

	App	Translated_Review	Sentiment	Sentiment_Polarity	Sentiment_Subjectivity
0	10 Best Foods for You	I like eat delicious food. That's I'm cooking ...	Positive	1.00	0.533333
1	10 Best Foods for You	This help eating healthy exercise regular basis	Positive	0.25	0.288462
2	10 Best Foods for You	0	0	0.00	0.000000
3	10 Best Foods for You	Works great especially going grocery store	Positive	0.40	0.875000
4	10 Best Foods for You	Best idea us	Positive	1.00	0.300000

Импьютация

Числовые данные

In [173]:

```
# Выберем числовые колонки с пропущенными значениями
# Цикл по колонкам датасета набора 1
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 {}, {}%.'.f
ormat(col, dt, temp_null_count, temp_perc))
```

```
Колонка Rating. Тип данных float64. Количество пустых значений 1474,
13.6%.
```

In [174]:

```
# Выберем числовые колонки с пропущенными значениями
# Цикл по колонкам датасета набора 2
num_cols1 = []
for col in data1.columns:
    # Количество пустых значений
    temp_null_count1 = data1[data1[col].isnull()].shape[0]
    dt1 = str(data1[col].dtype)
    if temp_null_count1>0 and (dt1=='float64' or dt1=='int64'):
        num_cols1.append(col)
        temp_perc1 = round((temp_null_count1 / total_count1) * 100.0, 2)
        print('Колонка {}. Тип данных {}. Количество пустых значений {}, {}%.'.f
ormat(col, dt1, temp_null_count1, temp_perc1))
```

```
Колонка Sentiment_Polarity. Тип данных float64. Количество пустых зн
ачений 26863, 41.78%.
Колонка Sentiment_Subjectivity. Тип данных float64. Количество пусты
х значений 26863, 41.78%.
```

In [175]:

```
# Фильтр по колонкам с пропущенными значениями  набора 1
data_num = data[num_cols]
data_num
```

Out[175]:

	Rating
0	4.1
1	3.9
2	4.7
3	4.5
4	4.3
...	...
10836	4.5
10837	5.0
10838	NaN
10839	4.5
10840	4.5

10841 rows × 1 columns

In [176]:

```
# Фильтр по колонкам с пропущенными значениями  набора 2
data_num1 = data1[num_cols1]
data_num1
```

Out[176]:

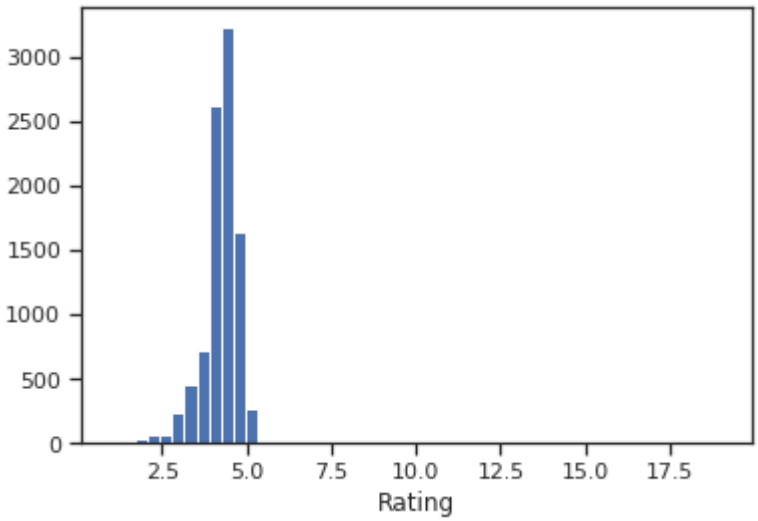
	Sentiment_Polarity	Sentiment_Subjectivity
0	1.00	0.533333
1	0.25	0.288462
2	NaN	NaN
3	0.40	0.875000
4	1.00	0.300000
...	...	...
64290	NaN	NaN
64291	NaN	NaN
64292	NaN	NaN
64293	NaN	NaN
64294	NaN	NaN

64295 rows × 2 columns

In [177]:

```
# Гистограмма по признакам  набора 1 - Rating
for col in data_num:
    plt.hist(data[col], 50)
    plt.xlabel(col)
    plt.show()
```

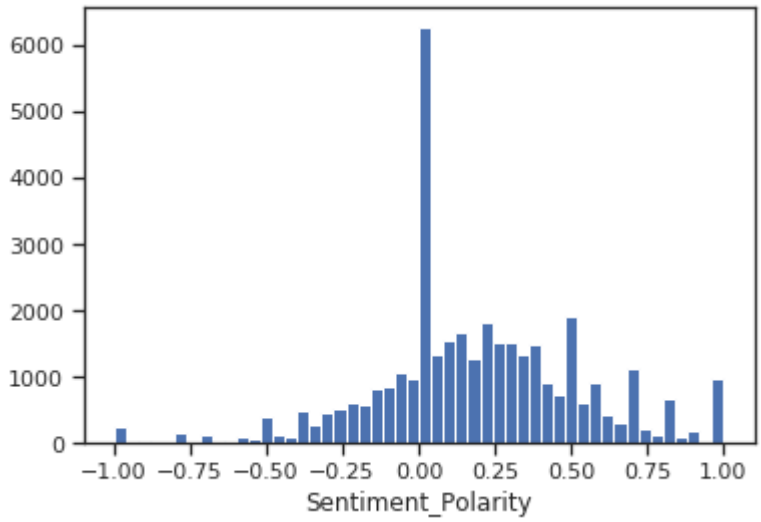
```
/home/lisobol/tensorflow_env/my_tensorflow/lib/python3.7/site-packag
es/numpy/lib/histograms.py:839: RuntimeWarning: invalid value encoun
tered in greater_equal
  keep = (tmp_a >= first_edge)
/home/lisobol/tensorflow_env/my_tensorflow/lib/python3.7/site-packag
es/numpy/lib/histograms.py:840: RuntimeWarning: invalid value encoun
tered in less_equal
  keep &= (tmp_a <= last_edge)
```

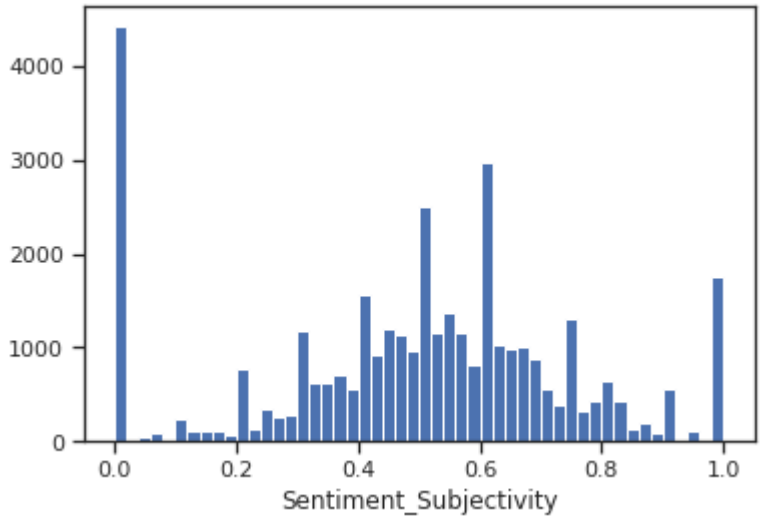


In [354]:

```
# Гистограмма по признакам  набора 1: Sentiment_Polarity, Sentiment_subjectivity
for col in data_num1:
    plt.hist(data1[col], 50)
    plt.xlabel(col)
    plt.show()
```

```
/home/lisobol/tensorflow_env/my_tensorflow/lib/python3.7/site-packag
es/numpy/lib/histograms.py:839: RuntimeWarning: invalid value encoun
tered in greater_equal
  keep = (tmp_a >= first_edge)
/home/lisobol/tensorflow_env/my_tensorflow/lib/python3.7/site-packag
es/numpy/lib/histograms.py:840: RuntimeWarning: invalid value encoun
tered in less_equal
  keep &= (tmp_a <= last_edge)
```





In [179]:

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

Out[179]:

	App	Category	Rating	Reviews	Size	Inst
23	Mcqueen Coloring pages	ART_AND_DESIGN	NaN	61	7.0M	100,0
113	Wrinkles and rejuvenation	BEAUTY	NaN	182	5.7M	100,0
123	Manicure - nail design	BEAUTY	NaN	119	3.7M	50,0
126	Skin Care and Natural Beauty	BEAUTY	NaN	654	7.4M	100,0
129	Secrets of beauty, youth and health	BEAUTY	NaN	77	2.9M	10,0
...	...	...	...	...	...	
10824	Cardio-FR	MEDICAL	NaN	67	82M	10,0
10825	Naruto & Boruto FR	SOCIAL	NaN	7	7.7M	1
10831	payermonstationnement.fr	MAPS_AND_NAVIGATION	NaN	38	9.8M	5,0
10835	FR Forms	BUSINESS	NaN	0	9.6M	
10838	Parkinson Exercices FR	MEDICAL	NaN	3	9.5M	1,0

1474 rows × 13 columns

In [180]:

```
# Фильтр по пустым значениям поля Sentiment_Polarity
data1[data1['Sentiment_Polarity'].isnull()]
```

Out[180]:

	App	Translated_Review	Sentiment	Sentiment_Polarity	Sentiment_Subjectivit
2	10 Best Foods for You	NaN	NaN	NaN	Nal
7	10 Best Foods for You	NaN	NaN	NaN	Nal
15	10 Best Foods for You	NaN	NaN	NaN	Nal
102	10 Best Foods for You	NaN	NaN	NaN	Nal
107	10 Best Foods for You	NaN	NaN	NaN	Nal
...	...	...	...	...	.
64290	Houzz Interior Design Ideas	NaN	NaN	NaN	Nal
64291	Houzz Interior Design Ideas	NaN	NaN	NaN	Nal
64292	Houzz Interior Design Ideas	NaN	NaN	NaN	Nal
64293	Houzz Interior Design Ideas	NaN	NaN	NaN	Nal
64294	Houzz Interior Design Ideas	NaN	NaN	NaN	Nal

26863 rows × 5 columns

In [181]:

```
# Фильтр по пустым значениям поля Sentiment_Subjectivity
data1[data1['Sentiment_Subjectivity'].isnull()]
```

Out[181]:

	App	Translated_Review	Sentiment	Sentiment_Polarity	Sentiment_Subjectivit
2	10 Best Foods for You	NaN	NaN	NaN	Nal
7	10 Best Foods for You	NaN	NaN	NaN	Nal
15	10 Best Foods for You	NaN	NaN	NaN	Nal
102	10 Best Foods for You	NaN	NaN	NaN	Nal
107	10 Best Foods for You	NaN	NaN	NaN	Nal
...	...	...	...	...	.
64290	Houzz Interior Design Ideas	NaN	NaN	NaN	Nal
64291	Houzz Interior Design Ideas	NaN	NaN	NaN	Nal
64292	Houzz Interior Design Ideas	NaN	NaN	NaN	Nal
64293	Houzz Interior Design Ideas	NaN	NaN	NaN	Nal
64294	Houzz Interior Design Ideas	NaN	NaN	NaN	Nal

26863 rows × 5 columns

In [182]:

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

Out[182]:

```
Int64Index([   23,   113,   123,   126,   129,   130,   134,   163,
180,
              185,
             ...
             10816, 10818, 10821, 10822, 10823, 10824, 10825, 10831,
10835,
             10838],
            dtype='int64', length=1474)
```

In [183]:

```
# Запоминаем индексы строк с пустыми значениями поля Sentiment_Polarity
flt_index1 = data1[data1['Sentiment_Polarity'].isnull()].index
flt_index1
```

Out[183]:

```
Int64Index([    2,     7,    15,   102,   107,   115,   362,   368,
405,
              407,
             ...
             64285, 64286, 64287, 64288, 64289, 64290, 64291, 64292,
64293,
             64294],
            dtype='int64', length=26863)
```

In [184]:

```
# Запоминаем индексы строк с пустыми значениями поляSentiment_Subjectivity
flt_index11 = data1[data1['Sentiment_Subjectivity'].isnull()].index
flt_index11
```

Out[184]:

```
Int64Index([    2,     7,    15,   102,   107,   115,   362,   368,
405,
              407,
             ...
             64285, 64286, 64287, 64288, 64289, 64290, 64291, 64292,
64293,
             64294],
            dtype='int64', length=26863)
```

In [185]:

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

Out[185]:

	App	Category	Rating	Reviews	Size	Inst
23	Mcqueen Coloring pages	ART_AND_DESIGN	NaN	61	7.0M	100,0
113	Wrinkles and rejuvenation	BEAUTY	NaN	182	5.7M	100,0
123	Manicure - nail design	BEAUTY	NaN	119	3.7M	50,0
126	Skin Care and Natural Beauty	BEAUTY	NaN	654	7.4M	100,0
129	Secrets of beauty, youth and health	BEAUTY	NaN	77	2.9M	10,0
...	...	...	...	...	...	
10824	Cardio-FR	MEDICAL	NaN	67	82M	10,0
10825	Naruto & Boruto FR	SOCIAL	NaN	7	7.7M	1
10831	payermonstationnement.fr	MAPS_AND_NAVIGATION	NaN	38	9.8M	5,0
10835	FR Forms	BUSINESS	NaN	0	9.6M	
10838	Parkinson Exercices FR	MEDICAL	NaN	3	9.5M	1,0

1474 rows × 13 columns

In [186]:

```
# Проверяем что выводятся нужные строки Sentiment_Polarity
data1[data1.index.isin(flt_index1)]
```

Out[186]:

	App	Translated_Review	Sentiment	Sentiment_Polarity	Sentiment_Subjectivit
2	10 Best Foods for You	NaN	NaN	NaN	Nal
7	10 Best Foods for You	NaN	NaN	NaN	Nal
15	10 Best Foods for You	NaN	NaN	NaN	Nal
102	10 Best Foods for You	NaN	NaN	NaN	Nal
107	10 Best Foods for You	NaN	NaN	NaN	Nal
...	...	...	...	...	.
64290	Houzz Interior Design Ideas	NaN	NaN	NaN	Nal
64291	Houzz Interior Design Ideas	NaN	NaN	NaN	Nal
64292	Houzz Interior Design Ideas	NaN	NaN	NaN	Nal
64293	Houzz Interior Design Ideas	NaN	NaN	NaN	Nal
64294	Houzz Interior Design Ideas	NaN	NaN	NaN	Nal

26863 rows × 5 columns

In [187]:

```
# Проверяем что выводятся нужные строки Sentiment_Subjectivity
data1[data1.index.isin(flt_index11)]
```

Out[187]:

	App	Translated_Review	Sentiment	Sentiment_Polarity	Sentiment_Subjectivit
2	10 Best Foods for You	NaN	NaN	NaN	Nal
7	10 Best Foods for You	NaN	NaN	NaN	Nal
15	10 Best Foods for You	NaN	NaN	NaN	Nal
102	10 Best Foods for You	NaN	NaN	NaN	Nal
107	10 Best Foods for You	NaN	NaN	NaN	Nal
...	...	...	...	...	.
64290	Houzz Interior Design Ideas	NaN	NaN	NaN	Nal
64291	Houzz Interior Design Ideas	NaN	NaN	NaN	Nal
64292	Houzz Interior Design Ideas	NaN	NaN	NaN	Nal
64293	Houzz Interior Design Ideas	NaN	NaN	NaN	Nal
64294	Houzz Interior Design Ideas	NaN	NaN	NaN	Nal

26863 rows × 5 columns

In [188]:

```
# фильтр по колонке Rating
data_num[data_num.index.isin(flt_index)]['Rating']
```

Out[188]:

```
23      NaN
113     NaN
123     NaN
126     NaN
129     NaN
         ..
10824   NaN
10825   NaN
10831   NaN
10835   NaN
10838   NaN
Name: Rating, Length: 1474, dtype: float64
```

In [189]:

```
# фильтр по колонке Sentiment_Polarity
data_num1[data_num1.index.isin(flt_index1)]['Sentiment_Polarity']
```

Out[189]:

```
2       NaN
7       NaN
15      NaN
102     NaN
107     NaN
         ..
64290   NaN
64291   NaN
64292   NaN
64293   NaN
64294   NaN
Name: Sentiment_Polarity, Length: 26863, dtype: float64
```

In [190]:

```
# фильтр по колонке Sentiment_Subjectivity
data_num1[data_num1.index.isin(flt_index1)]['Sentiment_Subjectivity']
```

Out[190]:

```
2       NaN
7       NaN
15      NaN
102     NaN
107     NaN
         ..
64290   NaN
64291   NaN
64292   NaN
64293   NaN
64294   NaN
Name: Sentiment_Subjectivity, Length: 26863, dtype: float64
```

In [191]:

```
data_num_Rating = data_num[['Rating']]
data_num_Rating.head()
```

Out[191]:

	Rating
0	4.1
1	3.9
2	4.7
3	4.5
4	4.3

In [192]:

```
data_num_SPol = data_num1[['Sentiment_Polarity']]
data_num_SPol.head()
```

Out[192]:

	Sentiment_Polarity
0	1.00
1	0.25
2	NaN
3	0.40
4	1.00

In [193]:

```
data_num_SSub = data_num1[['Sentiment_Subjectivity']]
data_num_SSub.head()
```

Out[193]:

	Sentiment_Subjectivity
0	0.533333
1	0.288462
2	NaN
3	0.875000
4	0.300000

In [194]:

```
from sklearn.impute import SimpleImputer
from sklearn.impute import MissingIndicator
```

In [195]:

```
# Фильтр для проверки заполнения пустых значений
indicator = MissingIndicator()
mask_missing_values_only = indicator.fit_transform(data_num_Rating)
mask_missing_values_only
```

Out[195]:

```
array([[False],
       [False],
       [False],
       ...,
       [ True],
       [False],
       [False]])
```

In [196]:

```
# Фильтр для проверки заполнения пустых значений
indicator = MissingIndicator()
mask_missing_values_only1 = indicator.fit_transform(data_num_SPol)
mask_missing_values_only1
```

Out[196]:

```
array([[False],
       [False],
       [ True],
       ...,
       [ True],
       [ True],
       [ True]])
```

In [197]:

```
# Фильтр для проверки заполнения пустых значений
indicator = MissingIndicator()
mask_missing_values_only11 = indicator.fit_transform(data_num_SSub)
mask_missing_values_only11
```

Out[197]:

```
array([[False],
       [False],
       [ True],
       ...,
       [ True],
       [ True],
       [ True]])
```

In [198]:

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

In [355]:

```
# Rating
def test_num_impute(strategy_param):
    imp_num = SimpleImputer(strategy=strategy_param)
    data_num_imp = imp_num.fit_transform(data_num_Rating)
    return data_num_imp[mask_missing_values_only]
```

In [200]:

```
# Sentiment_Polarity
def test_num_impute1(strategy_param):
    imp_num = SimpleImputer(strategy=strategy_param)
    data_num_imp = imp_num.fit_transform(data_num_SPol)
    return data_num_imp[mask_missing_values_only1]
```

In [201]:

```
# Sentiment_Subjectivity
def test_num_impute11(strategy_param):
    imp_num = SimpleImputer(strategy=strategy_param)
    data_num_imp = imp_num.fit_transform(data_num_SSub)
    return data_num_imp[mask_missing_values_only11]
```

In [202]:

```
strategies[0], test_num_impute(strategies[0])
```

Out[202]:

```
('mean',
 array([4.19333832, 4.19333832, 4.19333832, ..., 4.19333832, 4.19333
832,
        4.19333832]))
```

In [203]:

```
strategies[0], test_num_impute1(strategies[0])
```

Out[203]:

```
('mean',
 array([0.18214631, 0.18214631, 0.18214631, ..., 0.18214631, 0.18214
631,
        0.18214631]))
```

In [204]:

```
# Sentiment_Subjectivity
strategies[0], test_num_impute11(strategies[0])
```

Out[204]:

```
('mean',
 array([0.49270393, 0.49270393, 0.49270393, ..., 0.49270393, 0.49270
393,
        0.49270393]))
```

In [205]:

```
strategies[1], test_num_impute(strategies[1])
```

Out[205]:

```
('median', array([4.3, 4.3, 4.3, ..., 4.3, 4.3, 4.3]))
```

In [206]:

```
strategies[1], test_num_impute1(strategies[1])
```

Out[206]:

```
('median', array([0.15, 0.15, 0.15, ..., 0.15, 0.15, 0.15]))
```

In [207]:

```
# Sentiment_Subjectivity
strategies[1], test_num_impute11(strategies[1])
```

Out[207]:

```
('median',
 array([0.51428571, 0.51428571, 0.51428571, ..., 0.51428571, 0.51428
571,
        0.51428571]))
```

In [208]:

```
strategies[2], test_num_impute(strategies[2])
```

Out[208]:

```
('most_frequent', array([4.4, 4.4, 4.4, ..., 4.4, 4.4, 4.4]))
```

In [209]:

```
strategies[2], test_num_impute1(strategies[2])
```

Out[209]:

```
('most_frequent', array([0., 0., 0., ..., 0., 0., 0.]))
```

In [210]:

```
# Sentiment_Subjectivity
strategies[2], test_num_impute11(strategies[2])
```

Out[210]:

```
('most_frequent', array([0., 0., 0., ..., 0., 0., 0.]))
```

In [211]:

```
# 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]
```

In [212]:

```
# Более сложная функция, которая позволяет задавать колонку и вид импьютации
def test_num_impute_col1(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_only1]

    return column, strategy_param, filled_data.size, filled_data[0], filled_data
[filled_data.size-1]
```

In [213]:

```
# Sentiment_Subjectivity
# Более сложная функция, которая позволяет задавать колонку и вид импьютации
def test_num_impute_col11(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_only11]

    return column, strategy_param, filled_data.size, filled_data[0], filled_data
[filled_data.size-1]
```

In [214]:

```
data[['Rating']].describe()
```

Out[214]:

	Rating
count	9367.000000
mean	4.193338
std	0.537431
min	1.000000
25%	4.000000
50%	4.300000
75%	4.500000
max	19.000000

In [215]:

```
data1[['Sentiment_Polarity']].describe()
```

Out[215]:

	Sentiment_Polarity
count	37432.000000
mean	0.182146
std	0.351301
min	-1.000000
25%	0.000000
50%	0.150000
75%	0.400000
max	1.000000

In [216]:

```
data1[['Sentiment_Subjectivity']].describe()
```

Out[216]:

	Sentiment_Subjectivity
count	37432.000000
mean	0.492704
std	0.259949
min	0.000000
25%	0.357143
50%	0.514286
75%	0.650000
max	1.000000

In [217]:

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

Out[217]:

```
('Rating', 'mean', 1474, 4.193338315362443, 4.193338315362443)
```

In [218]:

```
test_num_impute_col1(data1, 'Sentiment_Polarity', strategies[0])
```

Out[218]:

```
('Sentiment_Polarity', 'mean', 26863, 0.18214631382977464, 0.1821463
1382977464)
```

In [219]:

```
test_num_impute_col11(data1, 'Sentiment_Subjectivity', strategies[0])
```

Out[219]:

```
('Sentiment_Subjectivity',
 'mean',
 26863,
 0.49270392839557814,
 0.49270392839557814)
```

In [220]:

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

Out[220]:

```
('Rating', 'median', 1474, 4.3, 4.3)
```

In [221]:

```
test_num_impute_col1(data1, 'Sentiment_Polarity', strategies[1])
```

Out[221]:

```
('Sentiment_Polarity', 'median', 26863, 0.15, 0.15)
```

In [311]:

```
test_num_impute_col1(data1, 'Sentiment_Subjectivity', strategies[2])
```

Out[311]:

```
('Sentiment_Subjectivity', 'most_frequent', 26863, 0.0, 0.0)
```

In [223]:

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

Out[223]:

```
('Rating', 'most_frequent', 1474, 4.4, 4.4)
```

In [224]:

```
test_num_impute_col1(data1, 'Sentiment_Polarity', strategies[2])
```

Out[224]:

```
('Sentiment_Polarity', 'most_frequent', 26863, 0.0, 0.0)
```

In [312]:

```
test_num_impute_col1(data1, 'Sentiment_Subjectivity', strategies[2])
```

Out[312]:

```
('Sentiment_Subjectivity', 'most_frequent', 26863, 0.0, 0.0)
```

Обработка пропусков в категориальных данных

In [226]:

```
# Выберем категориальные колонки с пропущенными значениями
# Цикл по колонкам датасета
data5 = pd.read_csv('covid_19_data.csv', sep=",")
total_count5 = data5.shape[0]
cat_cols5 = []
for col in data5.columns:
    # Количество пустых значений
    temp_null_count5 = data5[data5[col].isnull()].shape[0]
    dt5 = str(data5[col].dtype)
    if temp_null_count5>0 and (dt5=='object'):
        cat_cols5.append(col)
        temp_perc5 = round((temp_null_count5 / total_count5) * 100.0, 2)
        print('Колонка {}. Тип данных {}. Количество пустых значений {}, {}%.'.f
ormat(col, dt5, temp_null_count5, temp_perc5))
```

```
Колонка Province/State. Тип данных object. Количество пустых значени
й 1815, 36.78%.
```

In [227]:

```
# Выберем категориальные колонки с пропущенными значениями
# Цикл по колонкам датасета

cat_cols1 = []
for col in data1.columns:
    # Количество пустых значений
    temp_null_count1 = data1[data1[col].isnull()].shape[0]
    dt1 = str(data1[col].dtype)
    if temp_null_count1>0 and (dt1=='object'):
        cat_cols1.append(col)
        temp_perc1 = round((temp_null_count1 / total_count1) * 100.0, 2)
        print('Колонка {}. Тип данных {}. Количество пустых значений {}, {}%.'.f
ormat(col, dt1, temp_null_count1, temp_perc1))
```

```
Колонка Translated_Review. Тип данных object. Количество пустых знач
ений 26868, 41.79%.
Колонка Sentiment. Тип данных object. Количество пустых значений 268
63, 41.78%.
```

In [231]:

```
cat_temp_data = data5[['Province/State']]
cat_temp_data.head()
```

Out[231]:

	Province/State
0	Anhui
1	Beijing
2	Chongqing
3	Fujian
4	Gansu

In [232]:

```
cat_temp_data1 = data1[['Translated_Review']]
cat_temp_data1.head()
```

Out[232]:

	Translated_Review
0	I like eat delicious food. That's I'm cooking ...
1	This help eating healthy exercise regular basis
2	NaN
3	Works great especially going grocery store
4	Best idea us

In [233]:

```
cat_temp_data11 = data1[['Sentiment']]
cat_temp_data11.head()
```

Out[233]:

	Sentiment
0	Positive
1	Positive
2	NaN
3	Positive
4	Positive

In [234]:

```
cat_temp_data['Province/State'].unique()
```

Out[234]:

```
array(['Anhui', 'Beijing', 'Chongqing', 'Fujian', 'Gansu', 'Guangdon
g',
       'Guangxi', 'Guizhou', 'Hainan', 'Hebei', 'Heilongjiang', 'Hen
an',
       'Hong Kong', 'Hubei', 'Hunan', 'Inner Mongolia', 'Jiangsu',
       'Jiangxi', 'Jilin', 'Liaoning', 'Macau', 'Ningxia', 'Qingha
i',
       'Shaanxi', 'Shandong', 'Shanghai', 'Shanxi', 'Sichuan', 'Taiw
an',
       'Tianjin', 'Tibet', 'Washington', 'Xinjiang', 'Yunnan', 'Zhej
iang',
       nan, 'Chicago', 'Illinois', 'California', 'Arizona', 'Ontari
o',
       'New South Wales', 'Victoria', 'British Columbia', 'Bavaria',
       'Queensland', 'Chicago, IL', 'South Australia', 'Boston, MA',
       'Los Angeles, CA', 'Orange, CA', 'Santa Clara, CA', 'Seattle,
WA',
       'Tempe, AZ', 'San Benito, CA', 'Toronto, ON', 'London, ON',
       'Madison, WI', 'Cruise Ship', 'Diamond Princess cruise ship',
       'San Diego County, CA', 'San Antonio, TX', 'Ashland, NE',
       'Travis, CA', 'From Diamond Princess', 'Lackland, TX', 'Non
e',
       'Humboldt County, CA', 'Sacramento County, CA',
       'Omaha, NE (From Diamond Princess)',
       'Travis, CA (From Diamond Princess)',
       'Lackland, TX (From Diamond Princess)',
       'Unassigned Location (From Diamond Princess)', ' Montreal, Q
C',
       'Western Australia', 'Portland, OR', 'Snohomish County, WA',
       'Providence, RI', 'King County, WA', 'Cook County, IL', 'Tasm
ania',
       'Grafton County, NH', 'Hillsborough, FL', 'New York City, N
Y',
       'Placer County, CA', 'San Mateo, CA', 'Sarasota, FL',
       'Sonoma County, CA', 'Umatilla, OR', 'Fulton County, GA',
       'Washington County, OR', ' Norfolk County, MA', 'Berkeley, C
A',
       'Maricopa County, AZ', 'Wake County, NC', 'Westchester Count
y, NY',
       'Orange County, CA', 'Northern Territory',
       'Contra Costa County, CA', 'Bergen County, NJ',
       'Harris County, TX', 'San Francisco County, CA',
       'Clark County, NV', 'Fort Bend County, TX', 'Grant County, W
A',
       'Queens County, NY', 'Santa Rosa County, FL',
       'Williamson County, TN', 'New York County, NY',
       'Unassigned Location, WA', 'Montgomery County, MD',
       'Suffolk County, MA', 'Denver County, CO', 'Summit County, C
O',
       'Calgary, Alberta', 'Chatham County, NC', 'Delaware County, P
A',
       'Douglas County, NE', 'Fayette County, KY', 'Floyd County, G
A',
       'Marion County, IN', 'Middlesex County, MA', 'Nassau County,
NY',
       'Norwell County, MA', 'Ramsey County, MN', 'Washoe County, N
V',
       'Wayne County, PA', 'Yolo County, CA', 'Santa Clara County, C
A',
```

```
       'Grand Princess Cruise Ship', 'Douglas County, CO',
       'Providence County, RI', 'Alameda County, CA',
       'Broward County, FL', 'Fairfield County, CT', 'Lee County, F
L',
       'Pinal County, AZ', 'Rockland County, NY', 'Saratoga County,
NY',
       'Edmonton, Alberta', 'Charleston County, SC', 'Clark County,
WA',
       'Cobb County, GA', 'Davis County, UT', 'El Paso County, CO',
       'Honolulu County, HI', 'Jackson County, OR ',
       'Jefferson County, WA', 'Kershaw County, SC', 'Klamath Count
y, OR',
       'Madera County, CA', 'Pierce County, WA', 'Plymouth County, M
A',
       'Santa Cruz County, CA', 'Tulsa County, OK',
       'Montgomery County, TX', 'Norfolk County, MA',
       'Montgomery County, PA', 'Fairfax County, VA',
       'Rockingham County, NH', 'Washington, D.C.',
       'Berkshire County, MA', 'Davidson County, TN',
       'Douglas County, OR', 'Fresno County, CA', 'Harford County, M
D',
       'Hendricks County, IN', 'Hudson County, NJ', 'Johnson County,
KS',
       'Kittitas County, WA', 'Manatee County, FL', 'Marion County,
OR',
       'Okaloosa County, FL', 'Polk County, GA', 'Riverside County,
CA',
       'Shelby County, TN', 'Spokane County, WA', 'St. Louis County,
MO',
       'Suffolk County, NY', 'Ulster County, NY',
       'Unassigned Location, VT', 'Unknown Location, MA',
       'Volusia County, FL', 'Alberta', 'Quebec', 'Johnson County, I
A',
       'Harrison County, KY', 'Bennington County, VT',
       'Carver County, MN', 'Charlotte County, FL', 'Cherokee Count
y, GA',
       'Collin County, TX', 'Jefferson County, KY',
       'Jefferson Parish, LA', 'Shasta County, CA',
       'Spartanburg County, SC', 'New York', 'Massachusetts',
       'Grand Princess', 'Georgia', 'Colorado', 'Florida', 'New Jers
ey',
       'Oregon', 'Texas', 'Pennsylvania', 'Iowa', 'Maryland',
       'North Carolina', 'South Carolina', 'Tennessee', 'Virginia',
       'Indiana', 'Kentucky', 'District of Columbia', 'Nevada',
       'New Hampshire', 'Minnesota', 'Nebraska', 'Ohio', 'Rhode Isla
nd',
       'Wisconsin', 'Connecticut', 'Hawaii', 'Oklahoma', 'Utah', 'Ka
nsas',
       'Louisiana', 'Missouri', 'Vermont', 'Alaska', 'Arkansas',
       'Delaware', 'Idaho', 'Maine', 'Michigan', 'Mississippi', 'Mon
tana',
       'New Mexico', 'North Dakota', 'South Dakota', 'West Virgini
a',
       'Wyoming', 'France', 'UK', 'Denmark', 'Faroe Islands', 'St Ma
rtin',
       'Channel Islands', 'New Brunswick', 'Saint Barthelemy',
       'Gibraltar'], dtype=object)
```

In [235]:

```
cat_temp_data1['Translated_Review'].unique()
```

Out[235]:

```
array(['I like eat delicious food. That\'s I\'m cooking food myself,
case "10 Best Foods" helps lot, also "Best Before (Shelf Life)"',
       'This help eating healthy exercise regular basis', nan, ...,
       'Dumb app, I wanted post property rent give option. Website w
ork. Waste time space phone.',
       'I property business got link SMS happy performance still guy
s need raise bar guys Cheers',
       'Useless app, I searched flats kondapur, Hyderabad . None num
ber reachable I know flats unavailable would keep posts active'],
      dtype=object)
```

In [236]:

```
cat_temp_data11['Sentiment'].unique()
```

Out[236]:

```
array(['Positive', nan, 'Neutral', 'Negative'], dtype=object)
```

In [242]:

```
cat_temp_data[cat_temp_data['Province/State'].isnull()].shape
```

Out[242]:

```
(1815, 1)
```

In [243]:

```
cat_temp_data1[cat_temp_data1['Translated_Review'].isnull()].shape
```

Out[243]:

```
(26868, 1)
```

In [244]:

```
cat_temp_data11[cat_temp_data11['Sentiment'].isnull()].shape
```

Out[244]:

```
(26863, 1)
```

In [245]:

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

Out[245]:

```
array([['Anhui'],
       ['Beijing'],
       ['Chongqing'],
       ...,
       ['West Virginia'],
       ['Wyoming'],
       ['Gansu']], dtype=object)
```

In [ ]:

```
# Импьютация наиболее частыми значениями
imp2 = SimpleImputer(missing_values=np.nan, strategy='most_frequent')
data_imp21 = imp2.fit_transform(cat_temp_data1)
data_imp21
```

In [247]:

```
# Импьютация наиболее частыми значениями
imp2 = SimpleImputer(missing_values=np.nan, strategy='most_frequent')
data_imp211 = imp2.fit_transform(cat_temp_data11)
data_imp211
```

Out[247]:

```
array([['Positive'],
       ['Positive'],
       ['Positive'],
       ...,
       ['Positive'],
       ['Positive'],
       ['Positive']], dtype=object)
```

In [248]:

```
# Пустые значения отсутствуют
np.unique(data_imp2)
```

Out[248]:

```
array([' Montreal, QC', ' Norfolk County, MA', 'Alameda County, CA',
       'Alaska', 'Alberta', 'Anhui', 'Arizona', 'Arkansas', 'Ashlan
d, NE',
       'Bavaria', 'Beijing', 'Bennington County, VT', 'Bergen Count
y, NJ',
       'Berkeley, CA', 'Berkshire County, MA', 'Boston, MA',
       'British Columbia', 'Broward County, FL', 'Calgary, Alberta',
       'California', 'Carver County, MN', 'Channel Islands',
       'Charleston County, SC', 'Charlotte County, FL',
       'Chatham County, NC', 'Cherokee County, GA', 'Chicago',
       'Chicago, IL', 'Chongqing', 'Clark County, NV', 'Clark Count
y, WA',
       'Cobb County, GA', 'Collin County, TX', 'Colorado', 'Connecti
cut',
       'Contra Costa County, CA', 'Cook County, IL', 'Cruise Ship',
       'Davidson County, TN', 'Davis County, UT', 'Delaware',
       'Delaware County, PA', 'Denmark', 'Denver County, CO',
       'Diamond Princess cruise ship', 'District of Columbia',
       'Douglas County, CO', 'Douglas County, NE', 'Douglas County,
OR',
       'Edmonton, Alberta', 'El Paso County, CO', 'Fairfax County, V
A',
       'Fairfield County, CT', 'Faroe Islands', 'Fayette County, K
Y',
       'Florida', 'Floyd County, GA', 'Fort Bend County, TX', 'Franc
e',
       'Fresno County, CA', 'From Diamond Princess', 'Fujian',
       'Fulton County, GA', 'Gansu', 'Georgia', 'Gibraltar',
       'Grafton County, NH', 'Grand Princess',
       'Grand Princess Cruise Ship', 'Grant County, WA', 'Guangdon
g',
       'Guangxi', 'Guizhou', 'Hainan', 'Harford County, MD',
       'Harris County, TX', 'Harrison County, KY', 'Hawaii', 'Hebe
i',
       'Heilongjiang', 'Henan', 'Hendricks County, IN',
       'Hillsborough, FL', 'Hong Kong', 'Honolulu County, HI', 'Hube
i',
       'Hudson County, NJ', 'Humboldt County, CA', 'Hunan', 'Idaho',
       'Illinois', 'Indiana', 'Inner Mongolia', 'Iowa',
       'Jackson County, OR ', 'Jefferson County, KY',
       'Jefferson County, WA', 'Jefferson Parish, LA', 'Jiangsu',
       'Jiangxi', 'Jilin', 'Johnson County, IA', 'Johnson County, K
S',
       'Kansas', 'Kentucky', 'Kershaw County, SC', 'King County, W
A',
       'Kittitas County, WA', 'Klamath County, OR', 'Lackland, TX',
       'Lackland, TX (From Diamond Princess)', 'Lee County, FL',
       'Liaoning', 'London, ON', 'Los Angeles, CA', 'Louisiana', 'Ma
cau',
       'Madera County, CA', 'Madison, WI', 'Maine', 'Manatee County,
FL',
       'Maricopa County, AZ', 'Marion County, IN', 'Marion County, O
R',
       'Maryland', 'Massachusetts', 'Michigan', 'Middlesex County, M
A',
       'Minnesota', 'Mississippi', 'Missouri', 'Montana',
       'Montgomery County, MD', 'Montgomery County, PA',
       'Montgomery County, TX', 'Nassau County, NY', 'Nebraska', 'Ne
vada',
```

```
       'New Brunswick', 'New Hampshire', 'New Jersey', 'New Mexico',
       'New South Wales', 'New York', 'New York City, NY',
       'New York County, NY', 'Ningxia', 'None', 'Norfolk County, M
A',
       'North Carolina', 'North Dakota', 'Northern Territory',
       'Norwell County, MA', 'Ohio', 'Okaloosa County, FL', 'Oklahom
a',
       'Omaha, NE (From Diamond Princess)', 'Ontario',
       'Orange County, CA', 'Orange, CA', 'Oregon', 'Pennsylvania',
       'Pierce County, WA', 'Pinal County, AZ', 'Placer County, CA',
       'Plymouth County, MA', 'Polk County, GA', 'Portland, OR',
       'Providence County, RI', 'Providence, RI', 'Qinghai', 'Quebe
c',
       'Queens County, NY', 'Queensland', 'Ramsey County, MN',
       'Rhode Island', 'Riverside County, CA', 'Rockingham County, N
H',
       'Rockland County, NY', 'Sacramento County, CA', 'Saint Barthe
lemy',
       'San Antonio, TX', 'San Benito, CA', 'San Diego County, CA',
       'San Francisco County, CA', 'San Mateo, CA',
       'Santa Clara County, CA', 'Santa Clara, CA',
       'Santa Cruz County, CA', 'Santa Rosa County, FL', 'Sarasota,
FL',
       'Saratoga County, NY', 'Seattle, WA', 'Shaanxi', 'Shandong',
       'Shanghai', 'Shanxi', 'Shasta County, CA', 'Shelby County, T
N',
       'Sichuan', 'Snohomish County, WA', 'Sonoma County, CA',
       'South Australia', 'South Carolina', 'South Dakota',
       'Spartanburg County, SC', 'Spokane County, WA', 'St Martin',
       'St. Louis County, MO', 'Suffolk County, MA', 'Suffolk Count
y, NY',
       'Summit County, CO', 'Taiwan', 'Tasmania', 'Tempe, AZ',
       'Tennessee', 'Texas', 'Tianjin', 'Tibet', 'Toronto, ON',
       'Travis, CA', 'Travis, CA (From Diamond Princess)',
       'Tulsa County, OK', 'UK', 'Ulster County, NY', 'Umatilla, O
R',
       'Unassigned Location (From Diamond Princess)',
       'Unassigned Location, VT', 'Unassigned Location, WA',
       'Unknown Location, MA', 'Utah', 'Vermont', 'Victoria', 'Virgi
nia',
       'Volusia County, FL', 'Wake County, NC', 'Washington',
       'Washington County, OR', 'Washington, D.C.', 'Washoe County,
NV',
       'Wayne County, PA', 'West Virginia', 'Westchester County, N
Y',
       'Western Australia', 'Williamson County, TN', 'Wisconsin',
       'Wyoming', 'Xinjiang', 'Yolo County, CA', 'Yunnan', 'Zhejian
g'],
      dtype=object)
```

In [249]:

```
# Пустые значения отсутствуют
np.unique(data_imp21)
```

Out[249]:

```
array(['!!!Dont waste time! Failed Samsung flagship phone galaxy s8,
Installed ,shows rotating circle internet download, keeps rotates fo
rever proper progress indication; finally shows failed download. Stu
pid game developers. Go NFS working good.',
       '"...Future Follow updated follow"...',
       '"An error occurred while loading the search results. Please
try again." And so it\'s already 2 days. The reinstallation did not
help',
       ..., '♡ Amazon',
       '❤❤ sometimes hands typing is not convenient to use, except t
his update on a version 10.19 a nice keyboard hands-on',
       '搵楼租楼 A lot of time, a lot of time management, easy to tak
e care of'],
      dtype=object)
```

In [250]:

```
# Пустые значения отсутствуют
np.unique(data_imp211)
```

Out[250]:

```
array(['Negative', 'Neutral', 'Positive'], dtype=object)
```

In [251]:

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

Out[251]:

```
array([['Anhui'],
       ['Beijing'],
       ['Chongqing'],
       ...,
       ['West Virginia'],
       ['Wyoming'],
       ['!!!']], dtype=object)
```

In [252]:

```
# Импьютация константой
imp3 = SimpleImputer(missing_values=np.nan, strategy='constant', fill_value=
'!!!')
data_imp31 = imp3.fit_transform(cat_temp_data1)
data_imp31
```

Out[252]:

```
array([['I like eat delicious food. That\'s I\'m cooking food mysel
f, case "10 Best Foods" helps lot, also "Best Before (Shelf Lif
e)"'],
       ['This help eating healthy exercise regular basis'],
       ['!!!'],
       ...,
       ['!!!'],
       ['!!!'],
       ['!!!']], dtype=object)
```

In [253]:

```
# Импьютация константой
imp3 = SimpleImputer(missing_values=np.nan, strategy='constant', fill_value=
'!!!')
data_imp311 = imp3.fit_transform(cat_temp_data11)
data_imp311
```

Out[253]:

```
array([['Positive'],
       ['Positive'],
       ['!!!'],
       ...,
       ['!!!'],
       ['!!!'],
       ['!!!']], dtype=object)
```

In [254]:

```
np.unique(data_imp3)
```

Out[254]:

```
array([' Montreal, QC', ' Norfolk County, MA', '!!!',
       'Alameda County, CA', 'Alaska', 'Alberta', 'Anhui', 'Arizon
a',
       'Arkansas', 'Ashland, NE', 'Bavaria', 'Beijing',
       'Bennington County, VT', 'Bergen County, NJ', 'Berkeley, CA',
       'Berkshire County, MA', 'Boston, MA', 'British Columbia',
       'Broward County, FL', 'Calgary, Alberta', 'California',
       'Carver County, MN', 'Channel Islands', 'Charleston County, S
C',
       'Charlotte County, FL', 'Chatham County, NC',
       'Cherokee County, GA', 'Chicago', 'Chicago, IL', 'Chongqing',
       'Clark County, NV', 'Clark County, WA', 'Cobb County, GA',
       'Collin County, TX', 'Colorado', 'Connecticut',
       'Contra Costa County, CA', 'Cook County, IL', 'Cruise Ship',
       'Davidson County, TN', 'Davis County, UT', 'Delaware',
       'Delaware County, PA', 'Denmark', 'Denver County, CO',
       'Diamond Princess cruise ship', 'District of Columbia',
       'Douglas County, CO', 'Douglas County, NE', 'Douglas County,
OR',
       'Edmonton, Alberta', 'El Paso County, CO', 'Fairfax County, V
A',
       'Fairfield County, CT', 'Faroe Islands', 'Fayette County, K
Y',
       'Florida', 'Floyd County, GA', 'Fort Bend County, TX', 'Franc
e',
       'Fresno County, CA', 'From Diamond Princess', 'Fujian',
       'Fulton County, GA', 'Gansu', 'Georgia', 'Gibraltar',
       'Grafton County, NH', 'Grand Princess',
       'Grand Princess Cruise Ship', 'Grant County, WA', 'Guangdon
g',
       'Guangxi', 'Guizhou', 'Hainan', 'Harford County, MD',
       'Harris County, TX', 'Harrison County, KY', 'Hawaii', 'Hebe
i',
       'Heilongjiang', 'Henan', 'Hendricks County, IN',
       'Hillsborough, FL', 'Hong Kong', 'Honolulu County, HI', 'Hube
i',
       'Hudson County, NJ', 'Humboldt County, CA', 'Hunan', 'Idaho',
       'Illinois', 'Indiana', 'Inner Mongolia', 'Iowa',
       'Jackson County, OR ', 'Jefferson County, KY',
       'Jefferson County, WA', 'Jefferson Parish, LA', 'Jiangsu',
       'Jiangxi', 'Jilin', 'Johnson County, IA', 'Johnson County, K
S',
       'Kansas', 'Kentucky', 'Kershaw County, SC', 'King County, W
A',
       'Kittitas County, WA', 'Klamath County, OR', 'Lackland, TX',
       'Lackland, TX (From Diamond Princess)', 'Lee County, FL',
       'Liaoning', 'London, ON', 'Los Angeles, CA', 'Louisiana', 'Ma
cau',
       'Madera County, CA', 'Madison, WI', 'Maine', 'Manatee County,
FL',
       'Maricopa County, AZ', 'Marion County, IN', 'Marion County, O
R',
       'Maryland', 'Massachusetts', 'Michigan', 'Middlesex County, M
A',
       'Minnesota', 'Mississippi', 'Missouri', 'Montana',
       'Montgomery County, MD', 'Montgomery County, PA',
       'Montgomery County, TX', 'Nassau County, NY', 'Nebraska', 'Ne
vada',
       'New Brunswick', 'New Hampshire', 'New Jersey', 'New Mexico',
```

```
       'New South Wales', 'New York', 'New York City, NY',
       'New York County, NY', 'Ningxia', 'None', 'Norfolk County, M
A',
       'North Carolina', 'North Dakota', 'Northern Territory',
       'Norwell County, MA', 'Ohio', 'Okaloosa County, FL', 'Oklahom
a',
       'Omaha, NE (From Diamond Princess)', 'Ontario',
       'Orange County, CA', 'Orange, CA', 'Oregon', 'Pennsylvania',
       'Pierce County, WA', 'Pinal County, AZ', 'Placer County, CA',
       'Plymouth County, MA', 'Polk County, GA', 'Portland, OR',
       'Providence County, RI', 'Providence, RI', 'Qinghai', 'Quebe
c',
       'Queens County, NY', 'Queensland', 'Ramsey County, MN',
       'Rhode Island', 'Riverside County, CA', 'Rockingham County, N
H',
       'Rockland County, NY', 'Sacramento County, CA', 'Saint Barthe
lemy',
       'San Antonio, TX', 'San Benito, CA', 'San Diego County, CA',
       'San Francisco County, CA', 'San Mateo, CA',
       'Santa Clara County, CA', 'Santa Clara, CA',
       'Santa Cruz County, CA', 'Santa Rosa County, FL', 'Sarasota,
FL',
       'Saratoga County, NY', 'Seattle, WA', 'Shaanxi', 'Shandong',
       'Shanghai', 'Shanxi', 'Shasta County, CA', 'Shelby County, T
N',
       'Sichuan', 'Snohomish County, WA', 'Sonoma County, CA',
       'South Australia', 'South Carolina', 'South Dakota',
       'Spartanburg County, SC', 'Spokane County, WA', 'St Martin',
       'St. Louis County, MO', 'Suffolk County, MA', 'Suffolk Count
y, NY',
       'Summit County, CO', 'Taiwan', 'Tasmania', 'Tempe, AZ',
       'Tennessee', 'Texas', 'Tianjin', 'Tibet', 'Toronto, ON',
       'Travis, CA', 'Travis, CA (From Diamond Princess)',
       'Tulsa County, OK', 'UK', 'Ulster County, NY', 'Umatilla, O
R',
       'Unassigned Location (From Diamond Princess)',
       'Unassigned Location, VT', 'Unassigned Location, WA',
       'Unknown Location, MA', 'Utah', 'Vermont', 'Victoria', 'Virgi
nia',
       'Volusia County, FL', 'Wake County, NC', 'Washington',
       'Washington County, OR', 'Washington, D.C.', 'Washoe County,
NV',
       'Wayne County, PA', 'West Virginia', 'Westchester County, N
Y',
       'Western Australia', 'Williamson County, TN', 'Wisconsin',
       'Wyoming', 'Xinjiang', 'Yolo County, CA', 'Yunnan', 'Zhejian
g'],
      dtype=object)
```

In [255]:

```
np.unique(data_imp31)
```

Out[255]:

```
array(['!!!',
       '!!!Dont waste time! Failed Samsung flagship phone galaxy s8,
Installed ,shows rotating circle internet download, keeps rotates fo
rever proper progress indication; finally shows failed download. Stu
pid game developers. Go NFS working good.',
       '"...Future Follow updated follow"...', ..., '♡ Amazon',
       '❤❤ sometimes hands typing is not convenient to use, except t
his update on a version 10.19 a nice keyboard hands-on',
       '搵楼租楼 A lot of time, a lot of time management, easy to tak
e care of'],
      dtype=object)
```

In [256]:

```
np.unique(data_imp311)
```

Out[256]:

```
array(['!!!', 'Negative', 'Neutral', 'Positive'], dtype=object)
```

In [257]:

```
data_imp3[data_imp3=='!!!'].size
```

Out[257]:

```
1815
```

In [258]:

```
data_imp31[data_imp31=='!!!'].size
```

Out[258]:

```
26868
```

In [259]:

```
data_imp311[data_imp311=='!!!'].size
```

Out[259]:

```
26863
```

In [260]:

```
data5.shape
```

Out[260]:

```
(4935, 8)
```

In [261]:

```
data1.shape
```

Out[261]:

```
(64295, 5)
```

Преобразование категориальных признаков в числовые

In [262]:

```
cat_enc = pd.DataFrame({'c1':data_imp2.T[0]})
cat_enc
```

Out[262]:

	c1
0	Anhui
1	Beijing
2	Chongqing
3	Fujian
4	Gansu
...	...
4930	Mississippi
4931	North Dakota
4932	West Virginia
4933	Wyoming
4934	Gansu

4935 rows × 1 columns

In [263]:

```
cat_enc1 = pd.DataFrame({'c1':data_imp21.T[0]})
cat_enc1
```

Out[263]:

	c1
0	I like eat delicious food. That's I'm cooking ...
1	This help eating healthy exercise regular basis
2	Good
3	Works great especially going grocery store
4	Best idea us
...	...
64290	Good
64291	Good
64292	Good
64293	Good
64294	Good

64295 rows × 1 columns

In [264]:

```
cat_enc11 = pd.DataFrame({'c1':data_imp211.T[0]})
cat_enc11
```

Out[264]:

	c1
0	Positive
1	Positive
2	Positive
3	Positive
4	Positive
...	...
64290	Positive
64291	Positive
64292	Positive
64293	Positive
64294	Positive

64295 rows × 1 columns

Кодирование категорий целочисленными значениями

In [265]:

```
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
```

In [266]:

```
le = LabelEncoder()
cat_enc_le = le.fit_transform(cat_enc['c1'])
```

In [267]:

```
le1 = LabelEncoder()
cat_enc_le1 = le1.fit_transform(cat_enc1['c1'])
```

In [268]:

```
le11 = LabelEncoder()
cat_enc_le11 = le11.fit_transform(cat_enc11['c1'])
```

In [269]:

```
cat_enc['c1'].unique()
```

Out[269]:

```
array(['Anhui', 'Beijing', 'Chongqing', 'Fujian', 'Gansu', 'Guangdon
g',
       'Guangxi', 'Guizhou', 'Hainan', 'Hebei', 'Heilongjiang', 'Hen
an',
       'Hong Kong', 'Hubei', 'Hunan', 'Inner Mongolia', 'Jiangsu',
       'Jiangxi', 'Jilin', 'Liaoning', 'Macau', 'Ningxia', 'Qingha
i',
       'Shaanxi', 'Shandong', 'Shanghai', 'Shanxi', 'Sichuan', 'Taiw
an',
       'Tianjin', 'Tibet', 'Washington', 'Xinjiang', 'Yunnan', 'Zhej
iang',
       'Chicago', 'Illinois', 'California', 'Arizona', 'Ontario',
       'New South Wales', 'Victoria', 'British Columbia', 'Bavaria',
       'Queensland', 'Chicago, IL', 'South Australia', 'Boston, MA',
       'Los Angeles, CA', 'Orange, CA', 'Santa Clara, CA', 'Seattle,
WA',
       'Tempe, AZ', 'San Benito, CA', 'Toronto, ON', 'London, ON',
       'Madison, WI', 'Cruise Ship', 'Diamond Princess cruise ship',
       'San Diego County, CA', 'San Antonio, TX', 'Ashland, NE',
       'Travis, CA', 'From Diamond Princess', 'Lackland, TX', 'Non
e',
       'Humboldt County, CA', 'Sacramento County, CA',
       'Omaha, NE (From Diamond Princess)',
       'Travis, CA (From Diamond Princess)',
       'Lackland, TX (From Diamond Princess)',
       'Unassigned Location (From Diamond Princess)', ' Montreal, Q
C',
       'Western Australia', 'Portland, OR', 'Snohomish County, WA',
       'Providence, RI', 'King County, WA', 'Cook County, IL', 'Tasm
ania',
       'Grafton County, NH', 'Hillsborough, FL', 'New York City, N
Y',
       'Placer County, CA', 'San Mateo, CA', 'Sarasota, FL',
       'Sonoma County, CA', 'Umatilla, OR', 'Fulton County, GA',
       'Washington County, OR', ' Norfolk County, MA', 'Berkeley, C
A',
       'Maricopa County, AZ', 'Wake County, NC', 'Westchester Count
y, NY',
       'Orange County, CA', 'Northern Territory',
       'Contra Costa County, CA', 'Bergen County, NJ',
       'Harris County, TX', 'San Francisco County, CA',
       'Clark County, NV', 'Fort Bend County, TX', 'Grant County, W
A',
       'Queens County, NY', 'Santa Rosa County, FL',
       'Williamson County, TN', 'New York County, NY',
       'Unassigned Location, WA', 'Montgomery County, MD',
       'Suffolk County, MA', 'Denver County, CO', 'Summit County, C
O',
       'Calgary, Alberta', 'Chatham County, NC', 'Delaware County, P
A',
       'Douglas County, NE', 'Fayette County, KY', 'Floyd County, G
A',
       'Marion County, IN', 'Middlesex County, MA', 'Nassau County,
NY',
       'Norwell County, MA', 'Ramsey County, MN', 'Washoe County, N
V',
       'Wayne County, PA', 'Yolo County, CA', 'Santa Clara County, C
A',
       'Grand Princess Cruise Ship', 'Douglas County, CO',
```

```
       'Providence County, RI', 'Alameda County, CA',
       'Broward County, FL', 'Fairfield County, CT', 'Lee County, F
L',
       'Pinal County, AZ', 'Rockland County, NY', 'Saratoga County,
NY',
       'Edmonton, Alberta', 'Charleston County, SC', 'Clark County,
WA',
       'Cobb County, GA', 'Davis County, UT', 'El Paso County, CO',
       'Honolulu County, HI', 'Jackson County, OR ',
       'Jefferson County, WA', 'Kershaw County, SC', 'Klamath Count
y, OR',
       'Madera County, CA', 'Pierce County, WA', 'Plymouth County, M
A',
       'Santa Cruz County, CA', 'Tulsa County, OK',
       'Montgomery County, TX', 'Norfolk County, MA',
       'Montgomery County, PA', 'Fairfax County, VA',
       'Rockingham County, NH', 'Washington, D.C.',
       'Berkshire County, MA', 'Davidson County, TN',
       'Douglas County, OR', 'Fresno County, CA', 'Harford County, M
D',
       'Hendricks County, IN', 'Hudson County, NJ', 'Johnson County,
KS',
       'Kittitas County, WA', 'Manatee County, FL', 'Marion County,
OR',
       'Okaloosa County, FL', 'Polk County, GA', 'Riverside County,
CA',
       'Shelby County, TN', 'Spokane County, WA', 'St. Louis County,
MO',
       'Suffolk County, NY', 'Ulster County, NY',
       'Unassigned Location, VT', 'Unknown Location, MA',
       'Volusia County, FL', 'Alberta', 'Quebec', 'Johnson County, I
A',
       'Harrison County, KY', 'Bennington County, VT',
       'Carver County, MN', 'Charlotte County, FL', 'Cherokee Count
y, GA',
       'Collin County, TX', 'Jefferson County, KY',
       'Jefferson Parish, LA', 'Shasta County, CA',
       'Spartanburg County, SC', 'New York', 'Massachusetts',
       'Grand Princess', 'Georgia', 'Colorado', 'Florida', 'New Jers
ey',
       'Oregon', 'Texas', 'Pennsylvania', 'Iowa', 'Maryland',
       'North Carolina', 'South Carolina', 'Tennessee', 'Virginia',
       'Indiana', 'Kentucky', 'District of Columbia', 'Nevada',
       'New Hampshire', 'Minnesota', 'Nebraska', 'Ohio', 'Rhode Isla
nd',
       'Wisconsin', 'Connecticut', 'Hawaii', 'Oklahoma', 'Utah', 'Ka
nsas',
       'Louisiana', 'Missouri', 'Vermont', 'Alaska', 'Arkansas',
       'Delaware', 'Idaho', 'Maine', 'Michigan', 'Mississippi', 'Mon
tana',
       'New Mexico', 'North Dakota', 'South Dakota', 'West Virgini
a',
       'Wyoming', 'France', 'UK', 'Denmark', 'Faroe Islands', 'St Ma
rtin',
       'Channel Islands', 'New Brunswick', 'Saint Barthelemy',
       'Gibraltar'], dtype=object)
```

In [270]:

```
cat_enc1['c1'].unique()
```

Out[270]:

```
array(['I like eat delicious food. That\'s I\'m cooking food myself,
case "10 Best Foods" helps lot, also "Best Before (Shelf Life)"',
       'This help eating healthy exercise regular basis', 'Good',
...,
       'Dumb app, I wanted post property rent give option. Website w
ork. Waste time space phone.',
       'I property business got link SMS happy performance still guy
s need raise bar guys Cheers',
       'Useless app, I searched flats kondapur, Hyderabad . None num
ber reachable I know flats unavailable would keep posts active'],
      dtype=object)
```

In [271]:

```
cat_enc11['c1'].unique()
```

Out[271]:

```
array(['Positive', 'Neutral', 'Negative'], dtype=object)
```

In [272]:

```
np.unique(cat_enc_le)
```

Out[272]:

```
array([  0,   1,   2,   3,   4,   5,   6,   7,   8,   9,  10,  11,
12,
        13,  14,  15,  16,  17,  18,  19,  20,  21,  22,  23,  24,
25,
        26,  27,  28,  29,  30,  31,  32,  33,  34,  35,  36,  37,
38,
        39,  40,  41,  42,  43,  44,  45,  46,  47,  48,  49,  50,
51,
        52,  53,  54,  55,  56,  57,  58,  59,  60,  61,  62,  63,
64,
        65,  66,  67,  68,  69,  70,  71,  72,  73,  74,  75,  76,
77,
        78,  79,  80,  81,  82,  83,  84,  85,  86,  87,  88,  89,
90,
        91,  92,  93,  94,  95,  96,  97,  98,  99, 100, 101, 102, 1
03,
       104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 1
16,
       117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 1
29,
       130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 1
42,
       143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 1
55,
       156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 1
68,
       169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 1
81,
       182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 1
94,
       195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 2
07,
       208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 2
20,
       221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 2
33,
       234, 235, 236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 2
46,
       247, 248, 249, 250])
```

In [273]:

```
np.unique(cat_enc_le1)
```

Out[273]:

```
array([    0,     1,     2, ..., 27991, 27992, 27993])
```

In [274]:

```
np.unique(cat_enc_le11)
```

Out[274]:

```
array([0, 1, 2])
```

In [275]:

```
le.inverse_transform([0, 1, 2])
```

Out[275]:

```
array([' Montreal, QC', ' Norfolk County, MA', 'Alameda County, C
A'],
      dtype=object)
```

In [276]:

```
le1.inverse_transform([0, 1, 2])
```

Out[276]:

```
array(['!!!Dont waste time! Failed Samsung flagship phone galaxy s8,
Installed ,shows rotating circle internet download, keeps rotates fo
rever proper progress indication; finally shows failed download. Stu
pid game developers. Go NFS working good.',
       '"...Future Follow updated follow"...',
       '"An error occurred while loading the search results. Please
try again." And so it\'s already 2 days. The reinstallation did not
help'],
      dtype=object)
```

In [277]:

```
le11.inverse_transform([0, 1, 2])
```

Out[277]:

```
array(['Negative', 'Neutral', 'Positive'], dtype=object)
```

Кодирование категорий наборами бинарных значений

In [278]:

```
ohe = OneHotEncoder()
cat_enc_ohe = ohe.fit_transform(cat_enc[['c1']])
```

In [279]:

```
ohe1 = OneHotEncoder()
cat_enc_ohe1 = ohe1.fit_transform(cat_enc1[['c1']])
```

In [280]:

```
ohe11 = OneHotEncoder()
cat_enc_ohe11 = ohe11.fit_transform(cat_enc11[['c1']])
```

In [281]:

```
cat_enc.shape
```

Out[281]:

```
(4935, 1)
```

In [282]:

```
cat_enc1.shape
```

Out[282]:

```
(64295, 1)
```

In [283]:

```
cat_enc11.shape
```

Out[283]:

```
(64295, 1)
```

In [284]:

```
cat_enc_ohe.shape
```

Out[284]:

```
(4935, 251)
```

In [285]:

```
cat_enc_ohe1.shape
```

Out[285]:

```
(64295, 27994)
```

In [286]:

```
cat_enc_ohe11.shape
```

Out[286]:

```
(64295, 3)
```

In [287]:

```
cat_enc_ohe
```

Out[287]:

```
<4935x251 sparse matrix of type '<class 'numpy.float64'>'
        with 4935 stored elements in Compressed Sparse Row format>
```

In [288]:

```
cat_enc_ohe1
```

Out[288]:

```
<64295x27994 sparse matrix of type '<class 'numpy.float64'>'
        with 64295 stored elements in Compressed Sparse Row format>
```

In [289]:

```
cat_enc_ohe1
```

Out[289]:

```
<64295x27994 sparse matrix of type '<class 'numpy.float64'>'
        with 64295 stored elements in Compressed Sparse Row format>
```

In [290]:

```
cat_enc_ohe.todense()[0:10]
```

Out[290]:

```
matrix([[0., 0., 0., ..., 0., 0., 0.],
        [0., 0., 0., ..., 0., 0., 0.],
        [0., 0., 0., ..., 0., 0., 0.],
        ...,
        [0., 0., 0., ..., 0., 0., 0.],
        [0., 0., 0., ..., 0., 0., 0.],
        [0., 0., 0., ..., 0., 0., 0.]])
```

In [291]:

```
cat_enc_ohe1[:45000].todense()[0:10]
```

Out[291]:

```
matrix([[0., 0., 0., ..., 0., 0., 0.],
        [0., 0., 0., ..., 0., 0., 0.],
        [0., 0., 0., ..., 0., 0., 0.],
        ...,
        [0., 0., 0., ..., 0., 0., 0.],
        [0., 0., 0., ..., 0., 0., 0.],
        [0., 0., 0., ..., 0., 0., 0.]])
```

In [292]:

```
cat_enc_ohe11[:45000].todense()[0:10]
```

Out[292]:

```
matrix([[0., 0., 1.],
        [0., 0., 1.],
        [0., 0., 1.],
        [0., 0., 1.],
        [0., 0., 1.],
        [0., 0., 1.],
        [0., 0., 1.],
        [0., 0., 1.],
        [0., 1., 0.],
        [0., 1., 0.]])
```

In [293]:

```
cat_enc.head(10)
```

Out[293]:

	c1
0	Anhui
1	Beijing
2	Chongqing
3	Fujian
4	Gansu
5	Guangdong
6	Guangxi
7	Guizhou
8	Hainan
9	Hebei

In [294]:

```
cat_enc1.head(10)
```

Out[294]:

	c1
0	I like eat delicious food. That's I'm cooking ...
1	This help eating healthy exercise regular basis
2	Good
3	Works great especially going grocery store
4	Best idea us
5	Best way
6	Amazing
7	Good
8	Looking forward app,
9	It helpful site ! It help foods get !

In [295]:

```
cat_enc11.head(10)
```

Out[295]:

	c1
0	Positive
1	Positive
2	Positive
3	Positive
4	Positive
5	Positive
6	Positive
7	Positive
8	Neutral
9	Neutral

Масштабирование данных

In [302]:

```
from sklearn.preprocessing import MinMaxScaler, StandardScaler, Normalizer
```

MinMax

In [ ]:

```
# data = pd.read_csv('googleplaystore.csv', sep=",")
strategies[0], test_num_impute(strategies[0])
sc1 = MinMaxScaler()
sc1_data = sc1.fit_transform(data[['Rating']])
```

In [322]:

```
strategies[0], test_num_impute1(strategies[0])
sc11 = MinMaxScaler()
sc1_data1 = sc11.fit_transform(data1[['Sentiment_Polarity']])
```

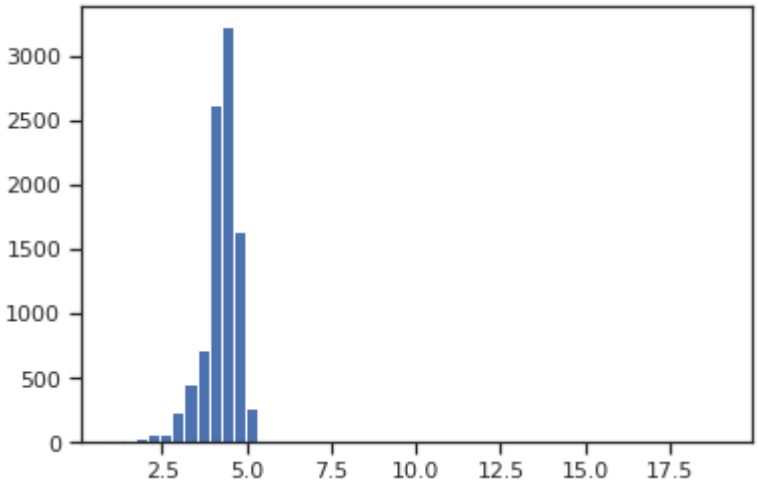
In [323]:

```
strategies[0], test_num_impute1(strategies[0])
sc111 = MinMaxScaler()
sc1_data11 = sc111.fit_transform(data1[['Sentiment_Subjectivity']])
```

In [324]:

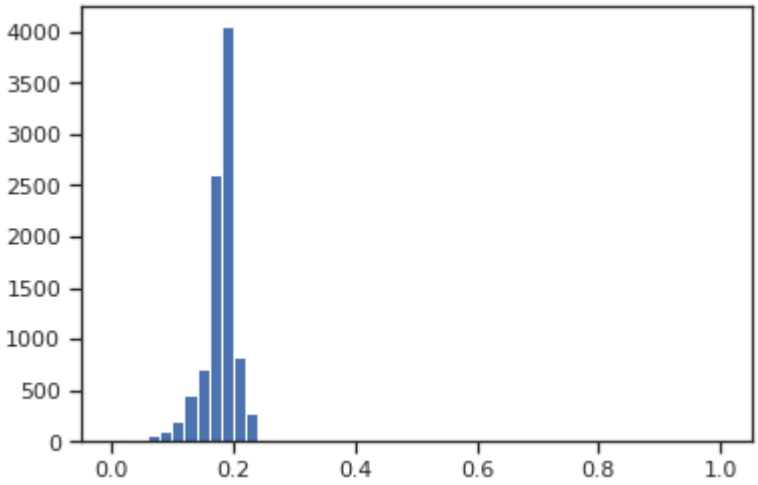
```
plt.hist(data['Rating'], 50)
plt.show()
```

```
/home/lisobol/tensorflow_env/my_tensorflow/lib/python3.7/site-packag
es/numpy/lib/histograms.py:839: RuntimeWarning: invalid value encoun
tered in greater_equal
  keep = (tmp_a >= first_edge)
/home/lisobol/tensorflow_env/my_tensorflow/lib/python3.7/site-packag
es/numpy/lib/histograms.py:840: RuntimeWarning: invalid value encoun
tered in less_equal
  keep &= (tmp_a <= last_edge)
```



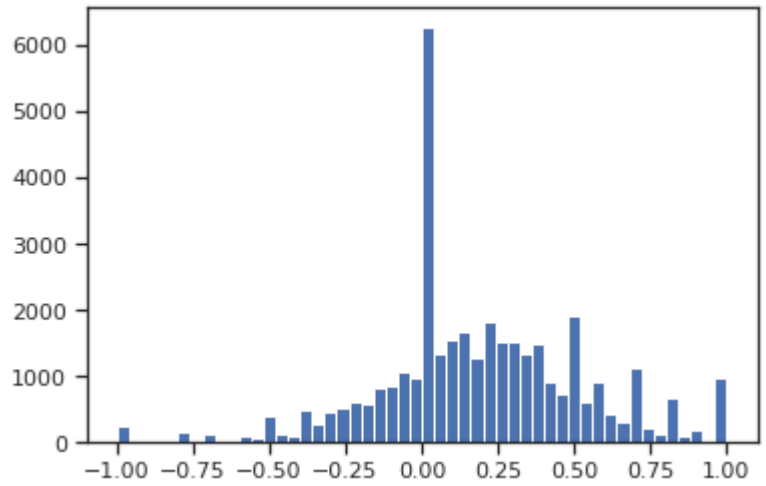
In [325]:

```
plt.hist(sc1_data, 50)
plt.show()
```



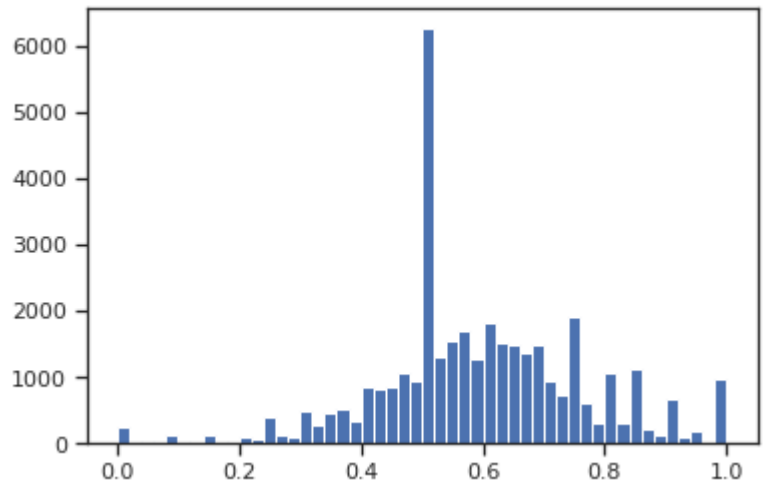
In [326]:

```
plt.hist(data1['Sentiment_Polarity'], 50)
plt.show()
```



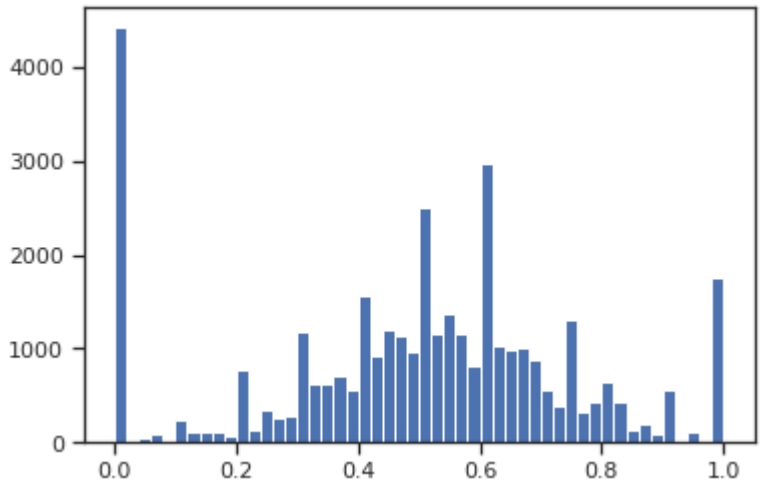
In [328]:

```
plt.hist(sc1_data1, 50)
plt.show()
```



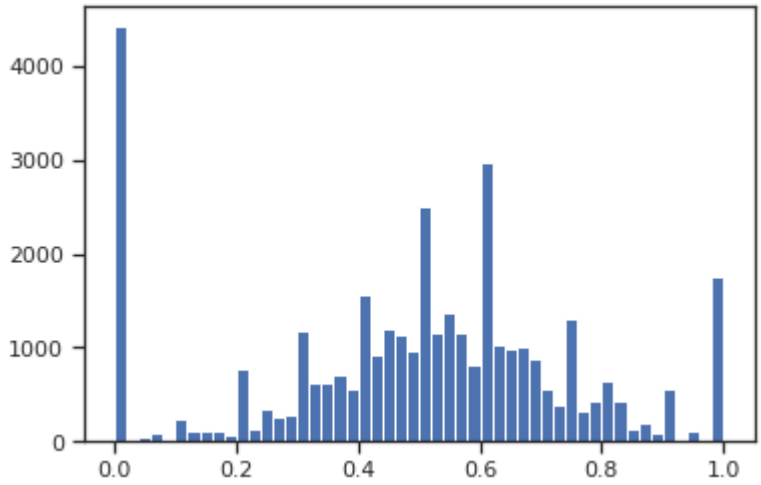
In [329]:

```
plt.hist(data1['Sentiment_Subjectivity'], 50)
plt.show()
```



In [330]:

```
plt.hist(sc1_data11, 50)
plt.show()
```



Z-оценка

In [331]:

```
sc2 = StandardScaler()
sc2_data = sc2.fit_transform(data[['Rating']])
```

In [332]:

```
sc21 = StandardScaler()
sc2_data1 = sc21.fit_transform(data1[['Sentiment_Polarity']])
```

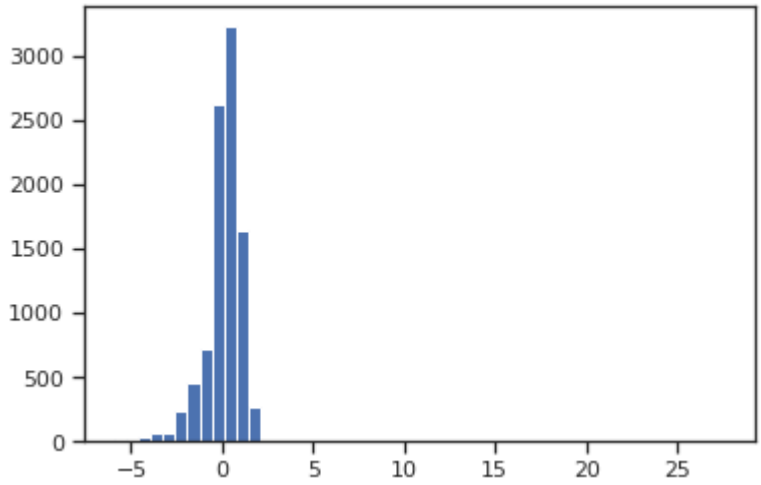
In [333]:

```
sc211 = StandardScaler()
sc2_data11 = sc211.fit_transform(data1[['Sentiment_Subjectivity']])
```

In [334]:

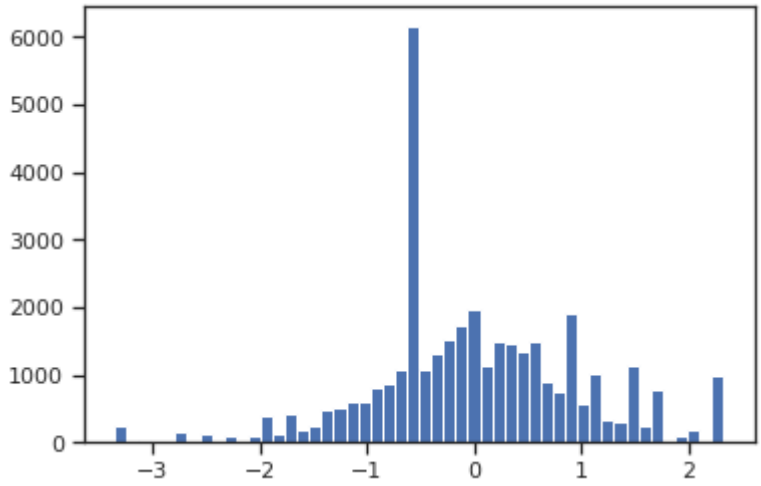
```
plt.hist(sc2_data, 50)
plt.show()
```

```
/home/lisobol/tensorflow_env/my_tensorflow/lib/python3.7/site-packag
es/numpy/lib/histograms.py:839: RuntimeWarning: invalid value encoun
tered in greater_equal
  keep = (tmp_a >= first_edge)
/home/lisobol/tensorflow_env/my_tensorflow/lib/python3.7/site-packag
es/numpy/lib/histograms.py:840: RuntimeWarning: invalid value encoun
tered in less_equal
  keep &= (tmp_a <= last_edge)
```



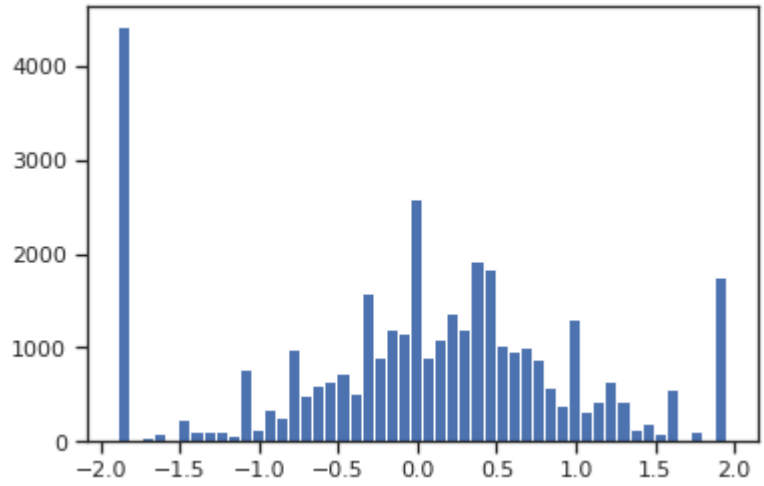
In [336]:

```
plt.hist(sc2_data1, 50)
plt.show()
```



In [337]:

```
plt.hist(sc2_data, 50)
plt.show()
```



Нормализация

In [340]:

```
sc3 = Normalizer()
sc3_data = sc3.fit_transform(data_new_2[['Rating']])
```

In [342]:

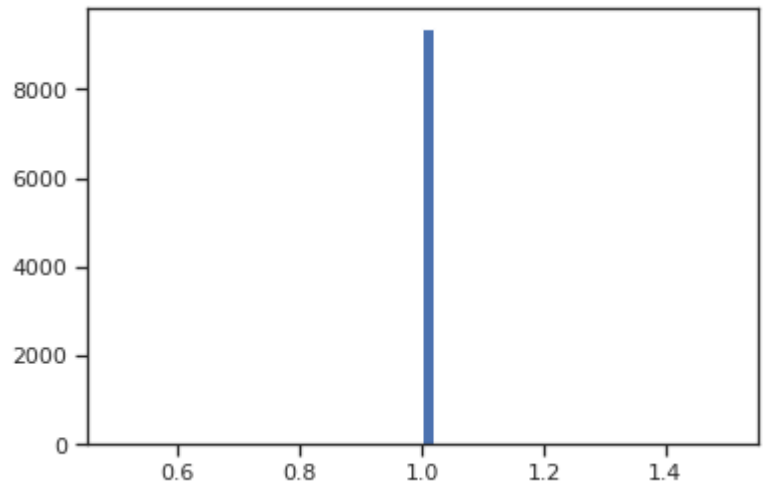
```
sc31 = StandardScaler()
sc3_data1 = sc31.fit_transform(data1[['Sentiment_Polarity']])
```

In [343]:

```
sc311 = StandardScaler()
sc3_data11 = sc311.fit_transform(data1[['Sentiment_Subjectivity']])
```

In [344]:

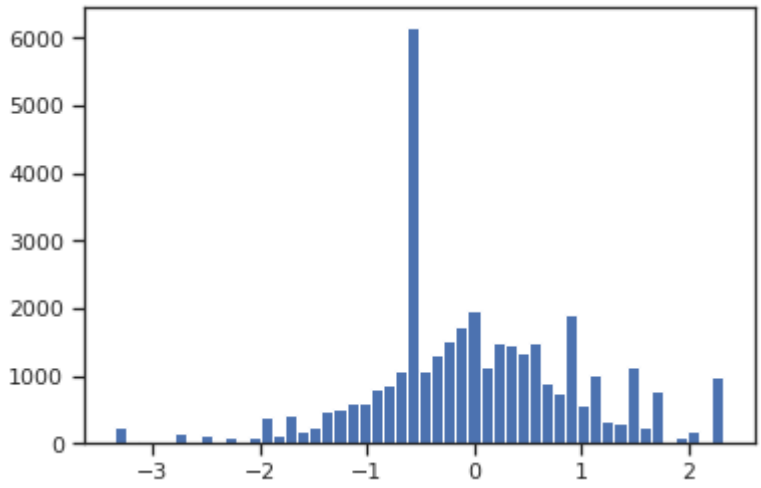
```
plt.hist(sc3_data, 50)
plt.show()
```



In [345]:

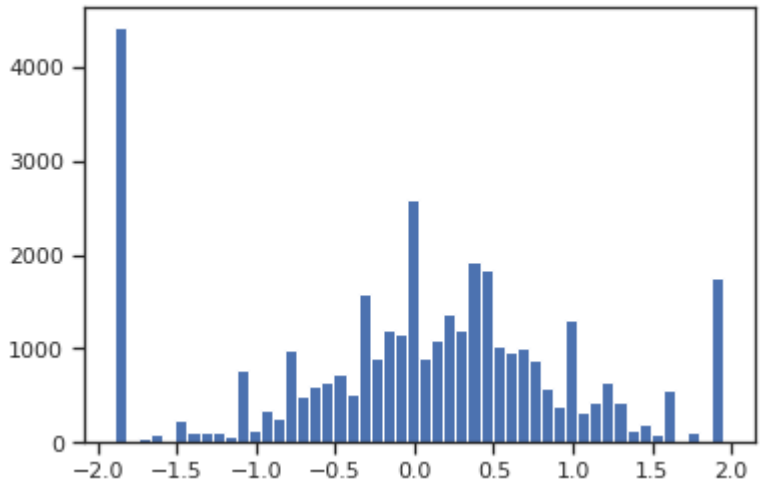
```
plt.hist(sc3_data1, 50)
plt.show()
```

```
/home/lisobol/tensorflow_env/my_tensorflow/lib/python3.7/site-packag
es/numpy/lib/histograms.py:839: RuntimeWarning: invalid value encoun
tered in greater_equal
  keep = (tmp_a >= first_edge)
/home/lisobol/tensorflow_env/my_tensorflow/lib/python3.7/site-packag
es/numpy/lib/histograms.py:840: RuntimeWarning: invalid value encoun
tered in less_equal
  keep &= (tmp_a <= last_edge)
```



In [346]:

```
plt.hist(sc3_data11, 50)
plt.show()
```



Вывод:

В процессе выполнения данной работы были изучены методы обработки пропусков в данных, кодирования категориальных признаков и масштабирования данных.