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

Мы научимся обрабатывать пропуски в данных для количественных (числовых) и категориальных признаков и масштабировать данные. Также мы научимся преобразовывать категориальные признаки в числовые.

В чем состоит проблема?

- Если в данных есть пропуски, то большинство алгоритмов машинного обучения не будут с ними работать. Даже корреляционная матрица не будет строиться корректно.
- Большинство алгоритмов машинного обучения требуют явного перекодирования категориальных признаков в числовые. Даже если алгоритм не требует этого явно, такое перекодирование возможно стоит попробовать, чтобы повысить качество модели.
- Большинство алгоритмов показывает лучшее качество на масштабированных признаках, в особенности алгоритмы, использующие методы градиентного спуска.

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

Загрузка и первичный анализ данных

Используем данные из соревнования House Prices: Advanced Regression Techniques.

```
# Будем использовать только обучающую выборку
data = pd.read_csv('imdb.csv', sep=';')

# размер набора данных
data.shape
```

```
(1000, 8)
```

```
# типы колонок
data.dtypes
```

```
title             object
director          object
release_year      object
runtime          float64
```

```
genre            object
rating          float64
metascore       float64
gross           float64
dtype: object
```

```
# проверим есть ли пропущенные значения
data.isnull().sum()
```

```
title             5
director          7
release_year      6
runtime           5
genre             5
rating            4
metascore       156
gross           188
dtype: int64
```

```
# Первые 5 строк датасета
data.head()
```

```
                     title              director release_year
runtime  \
0  The Shawshank Redemption        Frank Darabont         1994
142.0
1             The Godfather  Francis Ford Coppola         1972
175.0
2           The Dark Knight     Christopher Nolan         2008
152.0
3          Schindler's List      Steven Spielberg         1993
195.0
4              12 Angry Men          Sidney Lumet         1957
96.0

                       genre  rating  metascore   gross
0                      Drama     9.3       82.0   28.34
1              Crime, Drama      9.2      100.0  134.97
2       Action, Crime, Drama     9.0       84.0  534.86
3  Biography, Drama, History     9.0       95.0   96.90
4              Crime, Drama      9.0       97.0    4.36
```

```
total_count = data.shape[0]
print('Всего строк: {}'.format(total_count))
```

```
Всего строк: 1000
```

Обработка пропусков в данных

Простые стратегии - удаление или заполнение нулями

Удаление колонок, содержащих пустые значения `res = data.dropna(axis=1, how='any')`

Удаление строк, содержащих пустые значения `res = data.dropna(axis=0, how='any')`

Документация

Удаление может производиться для группы строк или колонок.

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

```
((1000, 8), (1000, 0))
```

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

```
((1000, 8), (717, 8))
```

```
data.head()
```

```
                        title              director release_year
runtime  \
0  The Shawshank Redemption        Frank Darabont         1994
142.0
1             The Godfather  Francis Ford Coppola         1972
175.0
2           The Dark Knight     Christopher Nolan         2008
152.0
3          Schindler's List      Steven Spielberg         1993
195.0
4              12 Angry Men          Sidney Lumet         1957
96.0

                        genre  rating  metascore   gross
0                       Drama     9.3       82.0   28.34
1               Crime, Drama     9.2      100.0  134.97
2       Action, Crime, Drama     9.0       84.0  534.86
3  Biography, Drama, History     9.0       95.0   96.90
4               Crime, Drama     9.0       97.0    4.36
```

```
# Заполнение всех пропущенных значений нулями
# В данном случае это некорректно, так как нулями заполняются в том
числе категориальные колонки
```

```
data_new_3 = data.fillna(0)
data_new_3.head()
```

```
                      title              director release_year
runtime  \
0  The Shawshank Redemption        Frank Darabont         1994
142.0
1             The Godfather  Francis Ford Coppola         1972
175.0
2           The Dark Knight     Christopher Nolan         2008
152.0
3          Schindler's List      Steven Spielberg         1993
195.0
4              12 Angry Men          Sidney Lumet         1957
96.0

                       genre  rating  metascore   gross
0                      Drama     9.3       82.0   28.34
1               Crime, Drama     9.2      100.0  134.97
2       Action, Crime, Drama     9.0       84.0  534.86
3  Biography, Drama, History     9.0       95.0   96.90
4               Crime, Drama     9.0       97.0    4.36
```

"Внедрение значений" - импьютация (imputation)

Обработка пропусков в числовых данных

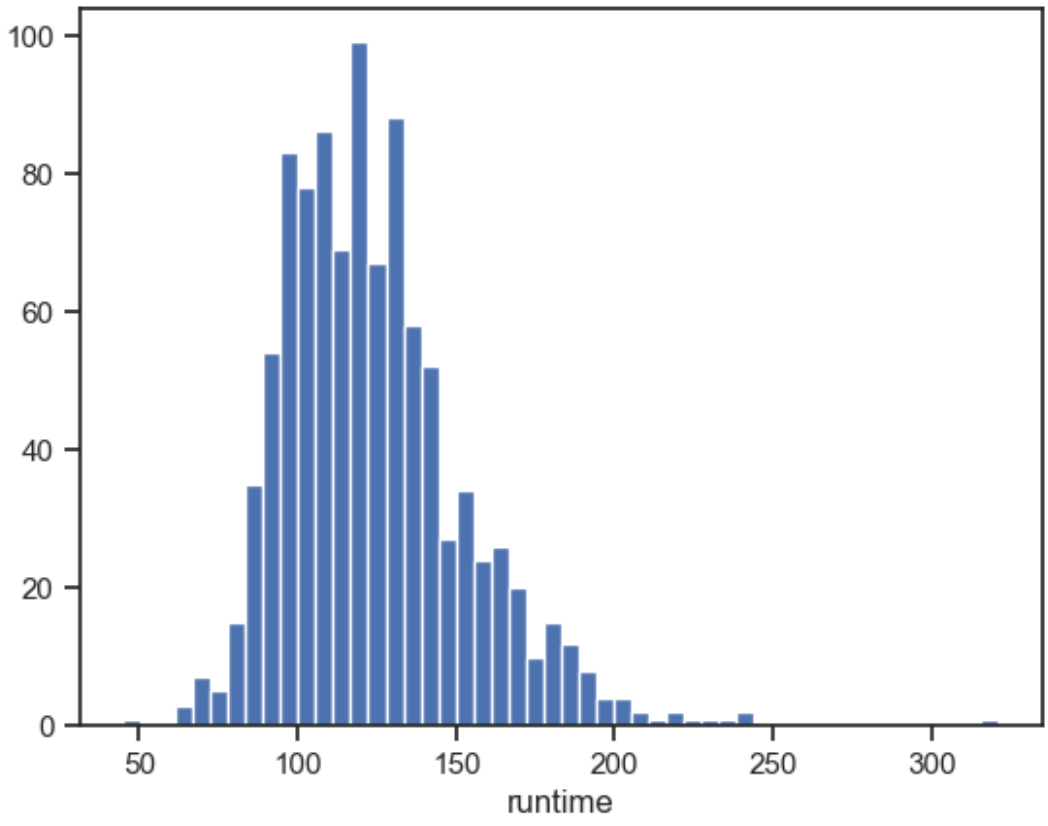
```
# Выберем числовые колонки с пропущенными значениями
# Цикл по колонкам датасета
num_cols = []
for col in data.columns:
    # Количество пустых значений
    temp_null_count = data[data[col].isnull()].shape[0]
    dt = str(data[col].dtype)
    if temp_null_count>0 and (dt=='float64' or dt=='int64'):
        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))
```

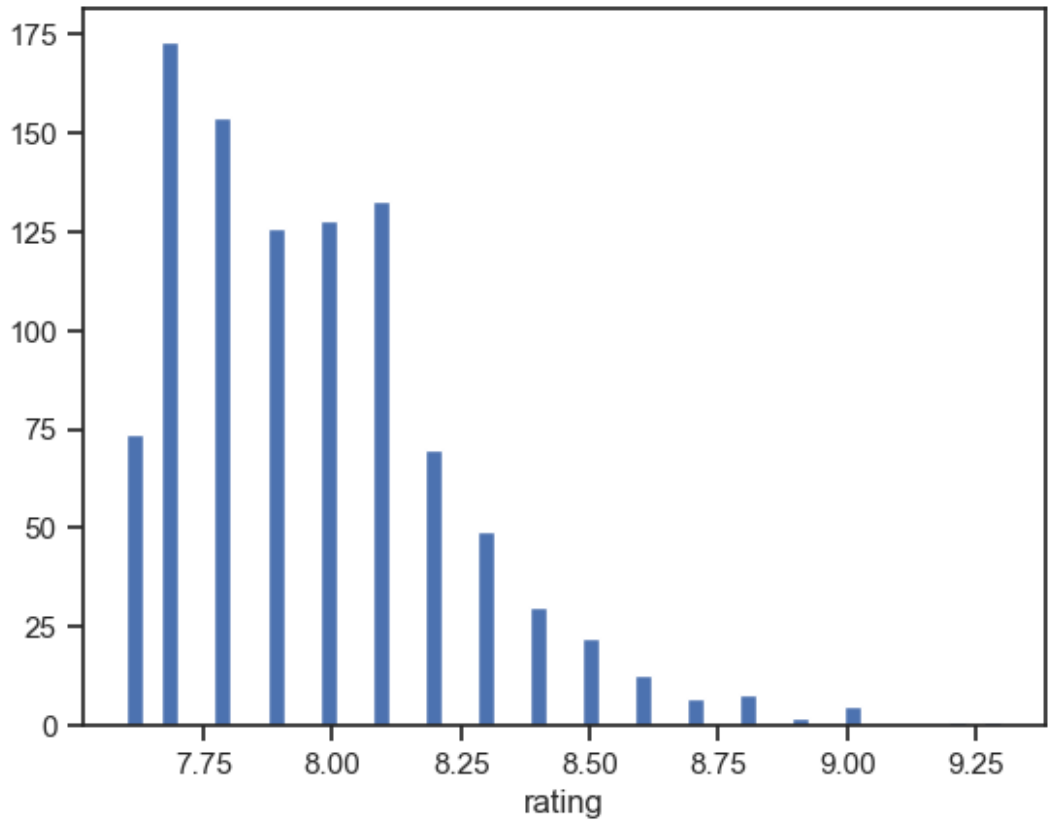
```
Колонка runtime. Тип данных float64. Количество пустых значений 5,
0.5%.
Колонка rating. Тип данных float64. Количество пустых значений 4,
0.4%.
Колонка metascore. Тип данных float64. Количество пустых значений 156,
15.6%.
Колонка gross. Тип данных float64. Количество пустых значений 188,
18.8%.
```

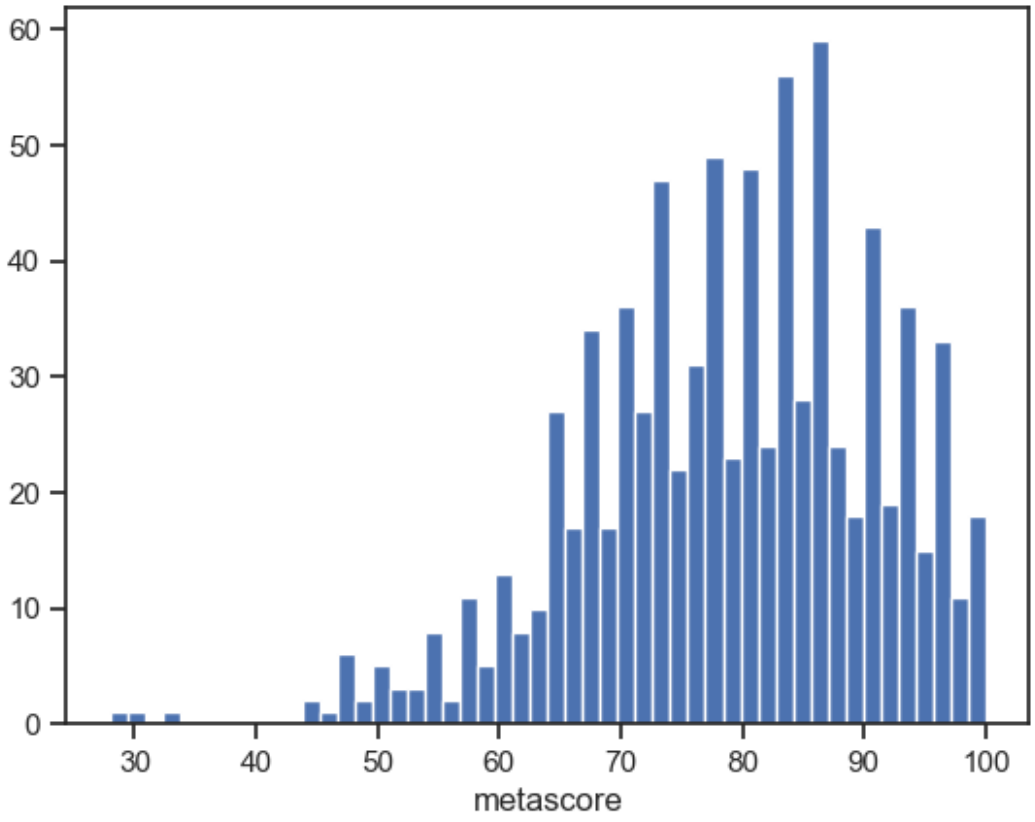
```
# Фильтр по колонкам с пропущенными значениями
data_num = data[num_cols]
data_num
```

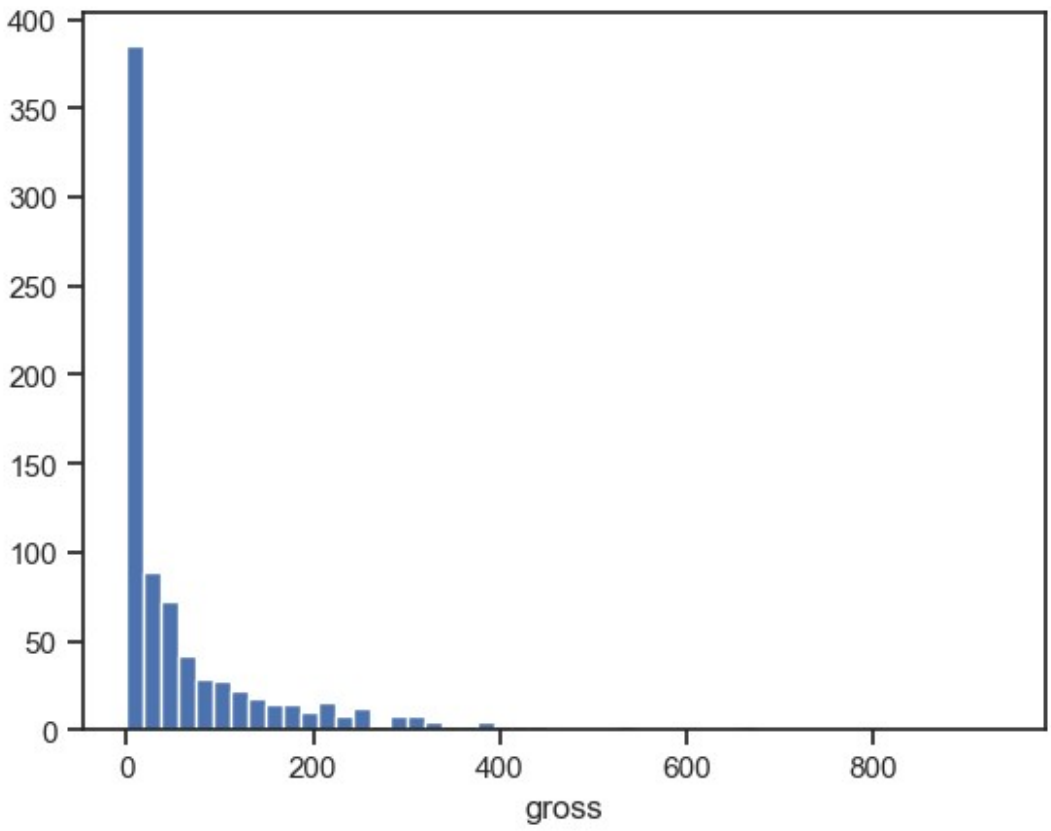
```
     runtime  rating  metascore   gross
0      142.0     9.3       82.0   28.34
1      175.0     9.2      100.0  134.97
2      152.0     9.0       84.0  534.86
3      195.0     9.0       95.0   96.90
4       96.0     9.0       97.0    4.36
..       ...     ...        ...     ...
995    133.0     7.6       76.0    6.17
996    105.0     7.6       87.0   35.81
997     98.0     7.6       77.0   37.71
998     71.0     7.6       87.0     NaN
999    113.0     7.6        NaN     NaN

[1000 rows x 4 columns]
```

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




175
150
125
100
75
50
25
0
7.75
8.00
8.25
8.50
8.75
9.00
9.25
rating


60
50
40
30
20
10
0
30
40
50
60
70
80
90
100
metascore



Будем использовать встроенные средства импьютации библиотеки scikit-learn - https://scikit-learn.org/stable/modules/impute.html

```
data_num_MasVnrArea = data_num[['metascore']]
data_num_MasVnrArea.head()
```

```
   metascore
0       82.0
1      100.0
2       84.0
3       95.0
4       97.0
```

```
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_MasVnrArea)
mask_missing_values_only
```

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

```
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [ True],
        [ True],
        [False],
        [False],
        [False],
        [False],
        [False],
        [ True],
        [ True],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [ True],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
```

```
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [ True],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [ True],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [ True],
        [False],
        [ True],
        [ True],
        [ True],
        [ True],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
```

```
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [ True],
        [False],
        [ True],
        [False],
        [False],
        [ True],
        [False],
        [ True],
        [ True],
        [ True],
        [ True],
        [ True],
        [ True],
        [ True],
        [ True],
        [ True],
        [ True],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
```

```
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [ True],
        [False],
        [False],
        [ True],
        [ True],
        [False],
        [False],
        [False],
        [False],
        [False],
        [ True],
        [ True],
        [ True],
        [ True],
        [ True],
        [False],
        [False],
        [False],
        [ True],
        [ True],
        [False],
```

```
        [ True],
        [False],
        [ True],
        [ True],
        [ True],
        [False],
        [ True],
        [ True],
        [ True],
        [ True],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
```

```
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [ True],
       [ True],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [ True],
       [False],
       [ True],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [ True],
       [False],
       [ True],
       [False],
       [False],
       [ True],
       [False],
       [False],
       [ True],
       [False],
       [ True],
       [False],
       [False],
```

```
       [False],
       [False],
       [False],
       [False],
       [False],
       [ True],
       [ True],
       [ True],
       [False],
       [False],
       [False],
       [ True],
       [ True],
       [ True],
       [ True],
       [ True],
       [ True],
       [ True],
       [False],
       [ True],
       [ True],
       [False],
       [ True],
       [ True],
       [ True],
       [ True],
       [False],
       [ True],
       [ True],
       [ True],
       [ True],
       [ True],
       [ True],
       [False],
       [ True],
       [ True],
       [False],
       [ True],
       [ True],
       [ True],
       [False],
       [ True],
       [ True],
       [ True],
       [ True],
       [ True],
       [False],
       [False],
       [False],
```

```
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [ True],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
```

```
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [ True],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [ True],
       [False],
       [False],
       [False],
       [False],
       [False],
       [ True],
       [False],
       [False],
       [False],
       [ True],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [ True],
       [False],
       [False],
       [False],
       [False],
       [False],
       [ True],
       [ True],
       [False],
       [ True],
```

```
       [ True],
       [False],
       [False],
       [ True],
       [ True],
       [False],
       [False],
       [False],
       [False],
       [ True],
       [False],
       [ True],
       [False],
       [False],
       [ True],
       [ True],
       [ True],
       [False],
       [ True],
       [ True],
       [False],
       [ True],
       [ True],
       [ True],
       [ True],
       [ True],
       [ True],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
```

```
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [ True],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [ True],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [ True],
       [False],
       [False],
       [False],
```

```
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [ True],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [ True],
       [ True],
       [ True],
       [False],
       [False],
       [False],
       [False],
       [ True],
       [False],
       [ True],
       [ True],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [ True],
       [ True],
       [False],
       [False],
       [ True],
       [ True],
       [ True],
       [False],
       [ True],
       [False],
       [False],
```

```
       [False],
       [ True],
       [False],
       [ True],
       [ True],
       [ True],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
```

```
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
```

```
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [ True],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [ True],
        [False],
        [ True],
        [False],
        [ True],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [ True],
        [False],
        [False],
        [False],
        [False],
```

```
       [False],
       [False],
       [False],
       [False],
       [False],
       [ True],
       [False],
       [ True],
       [False],
       [ True],
       [False],
       [ True],
       [ True],
       [ True],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
```

```
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
```

```
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [ True],
        [False],
        [ True],
        [False],
        [False],
        [ True],
        [False],
        [False],
        [False],
        [ True],
        [False],
        [False],
        [False],
        [False],
        [False],
        [ True],
        [False],
        [False],
        [ True],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [ True],
```

```
        [False],
        [False],
        [False],
        [ True],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [ True],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [ True],
        [False],
        [ True],
        [ True],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [ True],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
```

```
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
```

```
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [ True]])
```

С помощью класса SimpleImputer можно проводить импьютацию различными показателями центра распределения

```
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_MasVnrArea)
    return data_num_imp[mask_missing_values_only]

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

```
('mean',
 array([78.97274882, 78.97274882, 78.97274882, 78.97274882,
78.97274882,
        78.97274882, 78.97274882, 78.97274882, 78.97274882,
78.97274882,
        78.97274882, 78.97274882, 78.97274882, 78.97274882,
78.97274882,
        78.97274882, 78.97274882, 78.97274882, 78.97274882,
78.97274882,
        78.97274882, 78.97274882, 78.97274882, 78.97274882,
78.97274882,
        78.97274882, 78.97274882, 78.97274882, 78.97274882,
78.97274882,
        78.97274882, 78.97274882, 78.97274882, 78.97274882,
78.97274882,
        78.97274882, 78.97274882, 78.97274882, 78.97274882,
78.97274882,
        78.97274882, 78.97274882, 78.97274882, 78.97274882,
78.97274882,
        78.97274882, 78.97274882, 78.97274882, 78.97274882,
78.97274882,
```

```
        78.97274882, 78.97274882, 78.97274882, 78.97274882,
78.97274882,
        78.97274882, 78.97274882, 78.97274882, 78.97274882,
78.97274882,
        78.97274882, 78.97274882, 78.97274882, 78.97274882,
78.97274882,
        78.97274882, 78.97274882, 78.97274882, 78.97274882,
78.97274882,
        78.97274882, 78.97274882, 78.97274882, 78.97274882,
78.97274882,
        78.97274882, 78.97274882, 78.97274882, 78.97274882,
78.97274882,
        78.97274882, 78.97274882, 78.97274882, 78.97274882,
78.97274882,
        78.97274882, 78.97274882, 78.97274882, 78.97274882,
78.97274882,
        78.97274882, 78.97274882, 78.97274882, 78.97274882,
78.97274882,
        78.97274882, 78.97274882, 78.97274882, 78.97274882,
78.97274882,
        78.97274882, 78.97274882, 78.97274882, 78.97274882,
78.97274882,
        78.97274882, 78.97274882, 78.97274882, 78.97274882,
78.97274882,
        78.97274882, 78.97274882, 78.97274882, 78.97274882,
78.97274882,
        78.97274882, 78.97274882, 78.97274882, 78.97274882,
78.97274882,
        78.97274882, 78.97274882, 78.97274882, 78.97274882,
78.97274882,
        78.97274882, 78.97274882, 78.97274882, 78.97274882,
78.97274882,
        78.97274882, 78.97274882, 78.97274882, 78.97274882,
78.97274882,
        78.97274882, 78.97274882, 78.97274882, 78.97274882,
78.97274882,
        78.97274882, 78.97274882, 78.97274882, 78.97274882,
78.97274882,
        78.97274882, 78.97274882, 78.97274882, 78.97274882,
78.97274882,
        78.97274882, 78.97274882, 78.97274882, 78.97274882,
78.97274882,
        78.97274882]))
```

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

```
('median',
 array([80., 80., 80., 80., 80., 80., 80., 80., 80., 80., 80., 80.,
80.,
        80., 80., 80., 80., 80., 80., 80., 80., 80., 80., 80., 80.,
```

```
80.,
        80., 80., 80., 80., 80., 80., 80., 80., 80., 80., 80., 80.,
80.,
        80., 80., 80., 80., 80., 80., 80., 80., 80., 80., 80., 80.,
80.,
        80., 80., 80., 80., 80., 80., 80., 80., 80., 80., 80., 80.,
80.,
        80., 80., 80., 80., 80., 80., 80., 80., 80., 80., 80., 80.,
80.,
        80., 80., 80., 80., 80., 80., 80., 80., 80., 80., 80., 80.,
80.,
        80., 80., 80., 80., 80., 80., 80., 80., 80., 80., 80., 80.,
80.,
        80., 80., 80., 80., 80., 80., 80., 80., 80., 80., 80., 80.,
80.,
        80., 80., 80., 80., 80., 80., 80., 80., 80., 80., 80., 80.,
80.,
        80., 80., 80., 80., 80., 80., 80., 80., 80., 80., 80., 80.,
80.,
        80., 80., 80., 80., 80., 80., 80., 80., 80., 80., 80., 80.,
80.]))
```

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

```
('most_frequent',
 array([86., 86., 86., 86., 86., 86., 86., 86., 86., 86., 86., 86.,
86.,
        86., 86., 86., 86., 86., 86., 86., 86., 86., 86., 86., 86.,
86.,
        86., 86., 86., 86., 86., 86., 86., 86., 86., 86., 86., 86.,
86.,
        86., 86., 86., 86., 86., 86., 86., 86., 86., 86., 86., 86.,
86.,
        86., 86., 86., 86., 86., 86., 86., 86., 86., 86., 86., 86.,
86.,
        86., 86., 86., 86., 86., 86., 86., 86., 86., 86., 86., 86.,
86.,
        86., 86., 86., 86., 86., 86., 86., 86., 86., 86., 86., 86.,
86.,
        86., 86., 86., 86., 86., 86., 86., 86., 86., 86., 86., 86.,
86.,
        86., 86., 86., 86., 86., 86., 86., 86., 86., 86., 86., 86.,
86.,
        86., 86., 86., 86., 86., 86., 86., 86., 86., 86., 86., 86.,
86.,
        86., 86., 86., 86., 86., 86., 86., 86., 86., 86., 86., 86.,
86.,
        86., 86., 86., 86., 86., 86., 86., 86., 86., 86., 86., 86.,
86.]))
```

```
# 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[['gross']].describe()
```

```
            gross
count  812.000000
mean    71.131404
std    117.171987
min      0.000000
25%      3.180000
50%     23.250000
75%     85.100000
max    936.660000
```

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

```
('gross', 'mean', 188, 71.1314039408867, 71.1314039408867)
```

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

```
('gross', 'median', 188, 23.25, 23.25)
```

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

```
('gross', 'most_frequent', 188, 0.01, 0.01)
```

Обработка пропусков в категориальных данных

```
# 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 title. Тип данных object. Количество пустых значений 5, 0.5%.
Колонка director. Тип данных object. Количество пустых значений 7,
0.7%.
Колонка release_year. Тип данных object. Количество пустых значений 6,
0.6%.
Колонка genre. Тип данных object. Количество пустых значений 5, 0.5%.
```

Какие из этих колонок Вы бы выбрали или не выбрали для построения модели?

Класс SimpleImputer можно использовать для категориальных признаков со стратегиями "most_frequent" или "constant".

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

```
  release_year
0         1994
1         1972
2         2008
3         1993
4         1957
```

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

```
array(['1994', '1972', '2008', '1993', '1957', '2003', '1974', '2023',
       '2010', '2001', '1999', '1966', '2002', '2021', '2022', '2014',
       '1990', '1975', '1980', '2020', '1991', '1995', '1998', '1977',
       '1997', '1954', '1946', '1962', '1985', '2006', '2000', '1979',
       '2019', '1942', '1960', '2011', '1968', '1988', nan, '1931',
       'II 2018', '1981', '2018', '2012', '1986', 'I 2019', '1984',
       'I 2017', '2016', '2009', '1964', '1950', '1940', '1963',
'1987',
       '1983', '2004', '1971', '1958', '1992', '1959', '1941', '1952',
       '1927', '1973', '1961', '1944', '2007', '1948', '2017', '1921',
       '2013', '1989', '1982', '2005', '1976', 'I 2020', '1939',
'1965',
       '2015', '1955', '1969', '1953', '1928', 'I 2015', '1996',
'1978',
       'I 2013', '1967', '1949', '1934', '1951', '1926', '1925',
'1937',
       'III 2016', 'I 2014', '1920', '1947', 'I 2004', '1945', 'II
2016',
       '1956', '1922', 'I 2011', '1933', '1970', '1938', 'I 2001',
       'I 2008', 'I 2010', 'I 2006', 'I 2007', '1932', '1943', '1935',
       'II 2022', 'I 1995', 'I 2016', 'III 2018', 'I 1985'],
dtype=object)
```

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

```
(6, 1)
```

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

```
array([['1994'],
       ['1972'],
       ['2008'],
       ['1993'],
       ['1957'],
       ['2003'],
       ['1974'],
       ['2023'],
       ['1994'],
       ['2010'],
       ['2001'],
       ['1999'],
       ['1994'],
       ['1966'],
       ['2002'],
       ['2021'],
       ['2022'],
       ['2014'],
       ['1990'],
       ['1999'],
       ['1975'],
       ['1980'],
       ['2022'],
       ['2020'],
       ['1999'],
       ['1991'],
       ['1995'],
       ['1998'],
       ['1991'],
       ['1977'],
       ['2001'],
       ['2002'],
       ['1997'],
       ['1954'],
       ['1946'],
       ['1962'],
       ['2022'],
       ['1985'],
       ['2006'],
       ['2000'],
       ['1979'],
```

```
       ['2019'],
       ['2006'],
       ['2014'],
       ['1994'],
       ['1994'],
       ['1995'],
       ['1942'],
       ['1998'],
       ['1960'],
       ['2002'],
       ['2011'],
       ['1968'],
       ['1988'],
       ['1954'],
       ['1988'],
       ['2004'],
       ['1931'],
       ['II 2018'],
       ['1981'],
       ['2018'],
       ['2019'],
       ['2012'],
       ['1979'],
       ['1986'],
       ['I 2019'],
       ['2012'],
       ['2018'],
       ['1980'],
       ['2003'],
       ['2000'],
       ['1984'],
       ['1985'],
       ['I 2017'],
       ['2016'],
       ['2008'],
       ['2009'],
       ['2006'],
       ['1964'],
       ['1981'],
       ['2018'],
       ['1950'],
       ['1957'],
       ['1957'],
       ['1940'],
       ['1963'],
       ['2019'],
       ['2021'],
       ['2019'],
       ['2021'],
```

```
       ['2022'],
       ['1995'],
       ['2009'],
       ['1987'],
       ['1983'],
       ['1999'],
       ['2000'],
       ['1997'],
       ['2004'],
       ['1971'],
       ['1958'],
       ['1995'],
       ['1968'],
       ['1992'],
       ['1984'],
       ['2012'],
       ['1995'],
       ['2020'],
       ['2009'],
       ['1983'],
       ['2001'],
       ['2010'],
       ['1962'],
       ['2010'],
       ['1997'],
       ['1959'],
       ['1962'],
       ['1941'],
       ['1952'],
       ['1927'],
       ['1973'],
       ['2011'],
       ['1931'],
       ['1960'],
       ['2016'],
       ['1952'],
       ['1961'],
       ['2022'],
       ['1944'],
       ['2007'],
       ['1948'],
       ['2017'],
       ['2022'],
       ['2022'],
       ['2019'],
       ['1921'],
       ['2021'],
       ['2018'],
       ['2013'],
```

```
       ['2023'],
       ['1989'],
       ['2013'],
       ['2021'],
       ['2007'],
       ['1993'],
       ['2010'],
       ['2007'],
       ['1982'],
       ['2005'],
       ['1976'],
       ['2019'],
       ['1995'],
       ['2000'],
       ['2004'],
       ['1992'],
       ['1998'],
       ['2018'],
       ['2003'],
       ['1997'],
       ['1999'],
       ['2003'],
       ['1988'],
       ['2001'],
       ['I 2020'],
       ['2004'],
       ['1939'],
       ['1974'],
       ['2005'],
       ['2006'],
       ['1963'],
       ['1959'],
       ['1975'],
       ['1980'],
       ['2020'],
       ['2009'],
       ['1965'],
       ['1985'],
       ['2019'],
       ['2022'],
       ['1954'],
       ['1950'],
       ['2017'],
       ['2018'],
       ['2012'],
       ['1950'],
       ['1948'],
       ['1961'],
       ['1942'],
```

```
       ['2015'],
       ['2018'],
       ['2011'],
       ['2019'],
       ['1955'],
       ['1969'],
       ['1953'],
       ['1997'],
       ['2019'],
       ['2004'],
       ['1953'],
       ['2018'],
       ['1928'],
       ['2005'],
       ['2004'],
       ['1975'],
       ['1952'],
       ['2017'],
       ['2012'],
       ['2018'],
       ['2013'],
       ['1975'],
       ['2014'],
       ['2013'],
       ['2003'],
       ['1982'],
       ['1984'],
       ['2015'],
       ['2014'],
       ['2019'],
       ['2011'],
       ['2017'],
       ['2002'],
       ['1998'],
       ['2016'],
       ['1986'],
       ['2007'],
       ['2011'],
       ['1973'],
       ['I 2015'],
       ['2016'],
       ['1976'],
       ['1954'],
       ['1996'],
       ['1996'],
       ['1939'],
       ['2017'],
       ['1989'],
       ['2007'],
```

```
       ['1978'],
       ['2001'],
       ['2003'],
       ['I 2015'],
       ['1995'],
       ['I 2013'],
       ['2013'],
       ['I 2015'],
       ['1965'],
       ['1986'],
       ['2010'],
       ['1998'],
       ['2011'],
       ['2019'],
       ['2008'],
       ['2004'],
       ['1975'],
       ['2000'],
       ['1995'],
       ['2009'],
       ['1988'],
       ['1980'],
       ['1967'],
       ['1999'],
       ['2014'],
       ['2004'],
       ['1979'],
       ['2016'],
       ['1976'],
       ['1959'],
       ['1984'],
       ['1957'],
       ['1957'],
       ['1966'],
       ['1993'],
       ['2004'],
       ['2000'],
       ['1976'],
       ['1940'],
       ['1997'],
       ['1962'],
       ['1960'],
       ['1949'],
       ['1973'],
       ['1946'],
       ['2004'],
       ['2004'],
       ['2009'],
       ['1974'],
```

```
       ['1959'],
       ['1994'],
       ['1998'],
       ['2014'],
       ['1934'],
       ['1982'],
       ['2015'],
       ['1957'],
       ['1948'],
       ['2001'],
       ['2000'],
       ['1995'],
       ['1966'],
       ['1993'],
       ['1939'],
       ['2014'],
       ['1955'],
       ['1966'],
       ['2001'],
       ['2012'],
       ['1958'],
       ['1951'],
       ['1949'],
       ['1926'],
       ['2015'],
       ['1925'],
       ['1957'],
       ['2012'],
       ['2015'],
       ['1927'],
       ['1978'],
       ['2008'],
       ['1955'],
       ['2006'],
       ['1937'],
       ['1991'],
       ['1957'],
       ['III 2016'],
       ['2007'],
       ['1960'],
       ['2013'],
       ['1986'],
       ['2019'],
       ['1988'],
       ['1975'],
       ['2003'],
       ['2013'],
       ['2012'],
       ['2000'],
```

```
       ['1996'],
       ['1928'],
       ['2010'],
       ['2012'],
       ['2005'],
       ['1998'],
       ['1999'],
       ['1971'],
       ['2021'],
       ['2017'],
       ['2014'],
       ['2016'],
       ['1988'],
       ['2010'],
       ['1987'],
       ['2012'],
       ['2016'],
       ['2015'],
       ['2001'],
       ['2004'],
       ['2013'],
       ['2016'],
       ['1967'],
       ['2006'],
       ['I 2015'],
       ['2014'],
       ['2005'],
       ['2021'],
       ['1990'],
       ['1999'],
       ['2008'],
       ['1992'],
       ['2020'],
       ['1993'],
       ['1989'],
       ['1975'],
       ['1974'],
       ['1992'],
       ['2006'],
       ['1995'],
       ['2006'],
       ['2003'],
       ['1991'],
       ['1991'],
       ['1952'],
       ['1968'],
       ['2007'],
       ['1988'],
       ['1968'],
```

```
       ['1997'],
       ['2016'],
       ['2004'],
       ['1979'],
       ['1969'],
       ['2010'],
       ['2003'],
       ['1971'],
       ['2016'],
       ['1962'],
       ['2008'],
       ['1996'],
       ['1994'],
       ['1953'],
       ['1973'],
       ['I 2014'],
       ['1979'],
       ['1971'],
       ['1963'],
       ['1977'],
       ['1984'],
       ['1941'],
       ['1972'],
       ['1986'],
       ['1959'],
       ['1982'],
       ['2005'],
       ['1998'],
       ['1955'],
       ['1982'],
       ['1960'],
       ['1982'],
       ['1987'],
       ['1966'],
       ['2002'],
       ['1999'],
       ['2015'],
       ['1959'],
       ['2020'],
       ['1958'],
       ['1975'],
       ['1920'],
       ['1961'],
       ['1963'],
       ['1947'],
       ['1967'],
       ['2007'],
       ['1957'],
       ['2013'],
```

```
       ['2004'],
       ['1972'],
       ['1953'],
       ['1996'],
       ['2007'],
       ['2014'],
       ['I 2004'],
       ['2001'],
       ['1972'],
       ['1945'],
       ['1983'],
       ['1995'],
       ['2008'],
       ['2007'],
       ['2016'],
       ['1954'],
       ['1998'],
       ['2020'],
       ['2018'],
       ['1987'],
       ['1962'],
       ['2003'],
       ['2014'],
       ['2010'],
       ['1940'],
       ['1949'],
       ['1962'],
       ['1962'],
       ['1998'],
       ['2004'],
       ['1945'],
       ['2004'],
       ['1963'],
       ['2006'],
       ['2013'],
       ['2009'],
       ['1994'],
       ['2001'],
       ['1997'],
       ['2023'],
       ['2009'],
       ['2022'],
       ['2021'],
       ['II 2016'],
       ['2009'],
       ['2019'],
       ['2008'],
       ['2001'],
       ['2004'],
```

```
       ['2009'],
       ['1993'],
       ['1997'],
       ['1990'],
       ['2014'],
       ['2012'],
       ['2014'],
       ['2004'],
       ['2000'],
       ['2019'],
       ['2002'],
       ['2004'],
       ['2004'],
       ['1963'],
       ['1982'],
       ['2001'],
       ['1948'],
       ['2012'],
       ['2008'],
       ['2006'],
       ['1980'],
       ['2003'],
       ['2013'],
       ['2009'],
       ['2019'],
       ['I 2017'],
       ['1976'],
       ['1999'],
       ['1956'],
       ['I 2014'],
       ['1984'],
       ['1993'],
       ['2007'],
       ['1993'],
       ['1985'],
       ['1995'],
       ['1993'],
       ['2013'],
       ['1964'],
       ['1960'],
       ['2008'],
       ['1951'],
       ['1956'],
       ['2000'],
       ['2008'],
       ['1969'],
       ['1967'],
       ['1965'],
       ['2014'],
       ['1922'],
```

```
       ['1962'],
       ['1994'],
       ['2016'],
       ['1971'],
       ['2019'],
       ['1946'],
       ['2002'],
       ['1968'],
       ['I 2011'],
       ['1933'],
       ['2018'],
       ['1983'],
       ['1956'],
       ['2000'],
       ['1993'],
       ['1951'],
       ['2013'],
       ['1970'],
       ['1970'],
       ['1946'],
       ['2009'],
       ['1940'],
       ['1958'],
       ['1944'],
       ['2004'],
       ['1970'],
       ['2012'],
       ['2002'],
       ['2003'],
       ['1934'],
       ['1986'],
       ['1950'],
       ['1973'],
       ['1938'],
       ['2010'],
       ['2017'],
       ['1945'],
       ['1944'],
       ['1967'],
       ['1950'],
       ['1941'],
       ['2004'],
       ['2005'],
       ['2014'],
       ['1980'],
       ['2019'],
       ['1947'],
       ['2000'],
       ['1958'],
```

```
       ['2019'],
       ['1945'],
       ['2007'],
       ['1925'],
       ['1959'],
       ['1953'],
       ['2009'],
       ['1939'],
       ['2007'],
       ['2003'],
       ['1946'],
       ['I 2001'],
       ['2018'],
       ['1961'],
       ['2018'],
       ['I 2015'],
       ['2016'],
       ['2022'],
       ['2022'],
       ['1993'],
       ['2006'],
       ['2017'],
       ['1993'],
       ['2012'],
       ['1985'],
       ['2019'],
       ['I 2017'],
       ['2004'],
       ['2022'],
       ['I 2011'],
       ['2005'],
       ['2015'],
       ['2007'],
       ['2017'],
       ['2022'],
       ['I 2013'],
       ['2015'],
       ['1984'],
       ['2019'],
       ['2019'],
       ['2011'],
       ['1993'],
       ['1987'],
       ['2010'],
       ['2018'],
       ['2000'],
       ['1986'],
       ['2012'],
       ['2016'],
```

```
       ['2015'],
       ['2006'],
       ['2014'],
       ['1993'],
       ['2016'],
       ['2007'],
       ['2007'],
       ['1984'],
       ['2008'],
       ['2016'],
       ['1987'],
       ['I 2008'],
       ['2000'],
       ['1989'],
       ['2021'],
       ['1971'],
       ['1976'],
       ['2014'],
       ['2012'],
       ['2013'],
       ['2016'],
       ['2016'],
       ['2013'],
       ['1997'],
       ['1990'],
       ['2003'],
       ['2014'],
       ['2022'],
       ['1969'],
       ['1991'],
       ['2010'],
       ['1964'],
       ['2005'],
       ['1955'],
       ['2015'],
       ['1974'],
       ['1989'],
       ['1990'],
       ['2005'],
       ['2009'],
       ['1973'],
       ['1983'],
       ['I 2010'],
       ['1988'],
       ['1982'],
       ['I 2006'],
       ['2017'],
       ['2009'],
       ['1989'],
```

```
       ['2014'],
       ['1993'],
       ['I 2014'],
       ['2016'],
       ['1999'],
       ['2008'],
       ['1994'],
       ['1968'],
       ['1951'],
       ['2013'],
       ['1978'],
       ['1964'],
       ['1979'],
       ['1989'],
       ['1973'],
       ['1938'],
       ['1972'],
       ['1999'],
       ['1978'],
       ['1967'],
       ['I 2007'],
       ['1972'],
       ['1932'],
       ['1999'],
       ['1979'],
       ['2006'],
       ['2002'],
       ['1995'],
       ['1944'],
       ['2008'],
       ['1979'],
       ['1996'],
       ['2017'],
       ['2005'],
       ['1931'],
       ['1944'],
       ['1939'],
       ['1948'],
       ['2013'],
       ['2015'],
       ['1940'],
       ['1993'],
       ['1960'],
       ['1986'],
       ['1993'],
       ['1975'],
       ['2009'],
       ['1943'],
       ['2009'],
```

```
       ['1935'],
       ['1964'],
       ['2016'],
       ['1961'],
       ['1989'],
       ['1994'],
       ['1938'],
       ['2001'],
       ['2004'],
       ['2003'],
       ['1989'],
       ['2011'],
       ['1950'],
       ['2004'],
       ['1972'],
       ['2013'],
       ['2012'],
       ['1933'],
       ['1962'],
       ['2001'],
       ['1997'],
       ['2016'],
       ['1935'],
       ['2017'],
       ['2012'],
       ['2022'],
       ['2018'],
       ['2022'],
       ['1985'],
       ['II 2022'],
       ['2014'],
       ['2005'],
       ['2009'],
       ['2001'],
       ['2012'],
       ['2007'],
       ['2001'],
       ['2012'],
       ['1992'],
       ['2013'],
       ['1974'],
       ['2003'],
       ['1987'],
       ['2010'],
       ['2005'],
       ['1992'],
       ['1996'],
       ['2017'],
       ['2011'],
```

```
       ['2014'],
       ['2009'],
       ['2007'],
       ['1980'],
       ['2001'],
       ['1990'],
       ['2001'],
       ['1986'],
       ['1988'],
       ['2009'],
       ['1992'],
       ['1967'],
       ['1997'],
       ['2002'],
       ['2009'],
       ['2014'],
       ['1993'],
       ['2010'],
       ['2004'],
       ['2014'],
       ['1972'],
       ['1982'],
       ['1995'],
       ['1982'],
       ['2014'],
       ['2004'],
       ['2012'],
       ['I 1995'],
       ['2011'],
       ['2019'],
       ['1978'],
       ['2020'],
       ['2004'],
       ['2000'],
       ['I 2004'],
       ['1993'],
       ['1960'],
       ['1971'],
       ['2013'],
       ['1997'],
       ['2020'],
       ['1997'],
       ['2002'],
       ['1999'],
       ['1993'],
       ['1987'],
       ['1985'],
       ['2019'],
       ['1989'],
```

```
       ['2008'],
       ['2015'],
       ['2013'],
       ['1991'],
       ['2004'],
       ['1971'],
       ['1973'],
       ['2011'],
       ['1998'],
       ['1999'],
       ['2014'],
       ['1967'],
       ['2006'],
       ['2002'],
       ['1994'],
       ['1962'],
       ['2006'],
       ['I 2010'],
       ['1986'],
       ['1964'],
       ['1980'],
       ['1999'],
       ['1967'],
       ['1982'],
       ['2013'],
       ['1992'],
       ['2000'],
       ['1990'],
       ['2006'],
       ['2013'],
       ['I 2004'],
       ['2018'],
       ['2000'],
       ['1998'],
       ['1967'],
       ['1987'],
       ['1992'],
       ['2016'],
       ['1973'],
       ['2010'],
       ['2009'],
       ['1973'],
       ['1951'],
       ['1997'],
       ['2015'],
       ['2004'],
       ['2015'],
       ['1952'],
       ['1997'],
```

```
       ['1960'],
       ['2006'],
       ['1993'],
       ['1996'],
       ['1940'],
       ['2013'],
       ['1991'],
       ['1964'],
       ['2001'],
       ['1992'],
       ['1988'],
       ['1956'],
       ['2014'],
       ['2001'],
       ['2016'],
       ['1966'],
       ['1991'],
       ['1947'],
       ['2012'],
       ['2003'],
       ['2003'],
       ['2015'],
       ['1962'],
       ['1932'],
       ['1955'],
       ['2004'],
       ['1938'],
       ['1986'],
       ['1954'],
       ['2008'],
       ['1997'],
       ['1948'],
       ['2016'],
       ['2010'],
       ['1990'],
       ['2001'],
       ['2003'],
       ['2005'],
       ['2006'],
       ['2002'],
       ['2006'],
       ['2002'],
       ['2017'],
       ['1962'],
       ['1975'],
       ['1997'],
       ['1985'],
       ['1992'],
       ['2013'],
```

```
       ['1983'],
       ['2007'],
       ['2022'],
       ['2001'],
       ['2019'],
       ['2000'],
       ['1997'],
       ['2007'],
       ['2001'],
       ['2006'],
       ['2009'],
       ['2018'],
       ['2017'],
       ['I 2016'],
       ['1987'],
       ['2010'],
       ['2015'],
       ['1992'],
       ['1998'],
       ['2004'],
       ['1999'],
       ['2011'],
       ['2018'],
       ['2004'],
       ['2006'],
       ['2005'],
       ['2007'],
       ['2010'],
       ['2008'],
       ['2002'],
       ['II 2016'],
       ['2009'],
       ['2007'],
       ['2001'],
       ['1963'],
       ['1998'],
       ['1986'],
       ['1987'],
       ['2014'],
       ['2021'],
       ['1998'],
       ['2010'],
       ['III 2018'],
       ['2011'],
       ['1981'],
       ['1998'],
       ['2007'],
       ['2011'],
       ['2008'],
```

```
       ['1967'],
       ['1973'],
       ['2012'],
       ['2019'],
       ['2014'],
       ['2001'],
       ['2005'],
       ['1995'],
       ['I 1985'],
       ['1954'],
       ['1995'],
       ['1955'],
       ['1937'],
       ['1953'],
       ['2011'],
       ['1991'],
       ['2003'],
       ['2006'],
       ['2006'],
       ['2002'],
       ['1974'],
       ['2007'],
       ['2004'],
       ['1996'],
       ['2013'],
       ['1933'],
       ['2009']], dtype=object)
```

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

```
array(['1920', '1921', '1922', '1925', '1926', '1927', '1928', '1931',
       '1932', '1933', '1934', '1935', '1937', '1938', '1939', '1940',
       '1941', '1942', '1943', '1944', '1945', '1946', '1947', '1948',
       '1949', '1950', '1951', '1952', '1953', '1954', '1955', '1956',
       '1957', '1958', '1959', '1960', '1961', '1962', '1963', '1964',
       '1965', '1966', '1967', '1968', '1969', '1970', '1971', '1972',
       '1973', '1974', '1975', '1976', '1977', '1978', '1979', '1980',
       '1981', '1982', '1983', '1984', '1985', '1986', '1987', '1988',
       '1989', '1990', '1991', '1992', '1993', '1994', '1995', '1996',
       '1997', '1998', '1999', '2000', '2001', '2002', '2003', '2004',
       '2005', '2006', '2007', '2008', '2009', '2010', '2011', '2012',
       '2013', '2014', '2015', '2016', '2017', '2018', '2019', '2020',
       '2021', '2022', '2023', 'I 1985', 'I 1995', 'I 2001', 'I 2004',
       'I 2006', 'I 2007', 'I 2008', 'I 2010', 'I 2011', 'I 2013',
       'I 2014', 'I 2015', 'I 2016', 'I 2017', 'I 2019', 'I 2020',
       'II 2016', 'II 2018', 'II 2022', 'III 2016', 'III 2018'],
      dtype=object)
```

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

```
array([['1994'],
       ['1972'],
       ['2008'],
       ['1993'],
       ['1957'],
       ['2003'],
       ['1974'],
       ['2023'],
       ['1994'],
       ['2010'],
       ['2001'],
       ['1999'],
       ['1994'],
       ['1966'],
       ['2002'],
       ['2021'],
       ['2022'],
       ['2014'],
       ['1990'],
       ['1999'],
       ['1975'],
       ['1980'],
       ['2022'],
       ['2020'],
       ['1999'],
       ['1991'],
       ['1995'],
       ['1998'],
       ['1991'],
       ['1977'],
       ['2001'],
       ['2002'],
       ['1997'],
       ['1954'],
       ['1946'],
       ['1962'],
       ['2022'],
       ['1985'],
       ['2006'],
       ['2000'],
       ['1979'],
       ['2019'],
       ['2006'],
       ['2014'],
```

```
       ['1994'],
       ['1994'],
       ['1995'],
       ['1942'],
       ['1998'],
       ['1960'],
       ['2002'],
       ['2011'],
       ['1968'],
       ['1988'],
       ['1954'],
       ['1988'],
       ['NA'],
       ['1931'],
       ['II 2018'],
       ['1981'],
       ['2018'],
       ['2019'],
       ['2012'],
       ['1979'],
       ['1986'],
       ['I 2019'],
       ['2012'],
       ['2018'],
       ['1980'],
       ['2003'],
       ['2000'],
       ['1984'],
       ['1985'],
       ['I 2017'],
       ['2016'],
       ['2008'],
       ['2009'],
       ['2006'],
       ['1964'],
       ['1981'],
       ['2018'],
       ['1950'],
       ['1957'],
       ['1957'],
       ['1940'],
       ['1963'],
       ['2019'],
       ['2021'],
       ['2019'],
       ['2021'],
       ['2022'],
       ['1995'],
       ['2009'],
```

```
       ['1987'],
       ['1983'],
       ['1999'],
       ['2000'],
       ['1997'],
       ['2004'],
       ['1971'],
       ['1958'],
       ['1995'],
       ['1968'],
       ['1992'],
       ['1984'],
       ['2012'],
       ['1995'],
       ['2020'],
       ['2009'],
       ['1983'],
       ['2001'],
       ['2010'],
       ['1962'],
       ['2010'],
       ['1997'],
       ['1959'],
       ['1962'],
       ['1941'],
       ['1952'],
       ['1927'],
       ['1973'],
       ['2011'],
       ['1931'],
       ['1960'],
       ['2016'],
       ['1952'],
       ['1961'],
       ['2022'],
       ['1944'],
       ['2007'],
       ['1948'],
       ['2017'],
       ['2022'],
       ['2022'],
       ['2019'],
       ['1921'],
       ['2021'],
       ['2018'],
       ['2013'],
       ['2023'],
       ['1989'],
       ['2013'],
```

```
       ['2021'],
       ['2007'],
       ['1993'],
       ['2010'],
       ['2007'],
       ['1982'],
       ['2005'],
       ['1976'],
       ['2019'],
       ['1995'],
       ['2000'],
       ['2004'],
       ['1992'],
       ['1998'],
       ['2018'],
       ['2003'],
       ['1997'],
       ['1999'],
       ['2003'],
       ['1988'],
       ['2001'],
       ['I 2020'],
       ['2004'],
       ['1939'],
       ['1974'],
       ['2005'],
       ['2006'],
       ['1963'],
       ['1959'],
       ['1975'],
       ['1980'],
       ['2020'],
       ['2009'],
       ['1965'],
       ['1985'],
       ['2019'],
       ['2022'],
       ['1954'],
       ['1950'],
       ['2017'],
       ['2018'],
       ['2012'],
       ['1950'],
       ['1948'],
       ['1961'],
       ['1942'],
       ['2015'],
       ['2018'],
       ['2011'],
```

```
       ['2019'],
       ['1955'],
       ['1969'],
       ['1953'],
       ['1997'],
       ['2019'],
       ['2004'],
       ['1953'],
       ['2018'],
       ['1928'],
       ['2005'],
       ['NA'],
       ['1975'],
       ['1952'],
       ['2017'],
       ['2012'],
       ['2018'],
       ['2013'],
       ['1975'],
       ['2014'],
       ['2013'],
       ['2003'],
       ['1982'],
       ['1984'],
       ['2015'],
       ['2014'],
       ['2019'],
       ['2011'],
       ['2017'],
       ['2002'],
       ['1998'],
       ['2016'],
       ['1986'],
       ['2007'],
       ['2011'],
       ['1973'],
       ['I 2015'],
       ['2016'],
       ['1976'],
       ['1954'],
       ['1996'],
       ['1996'],
       ['1939'],
       ['2017'],
       ['1989'],
       ['2007'],
       ['1978'],
       ['2001'],
       ['2003'],
```

```
       ['I 2015'],
       ['1995'],
       ['I 2013'],
       ['2013'],
       ['I 2015'],
       ['1965'],
       ['1986'],
       ['2010'],
       ['1998'],
       ['2011'],
       ['2019'],
       ['2008'],
       ['2004'],
       ['1975'],
       ['2000'],
       ['1995'],
       ['2009'],
       ['1988'],
       ['1980'],
       ['1967'],
       ['1999'],
       ['2014'],
       ['2004'],
       ['1979'],
       ['2016'],
       ['1976'],
       ['1959'],
       ['1984'],
       ['1957'],
       ['1957'],
       ['1966'],
       ['1993'],
       ['2004'],
       ['2000'],
       ['1976'],
       ['1940'],
       ['1997'],
       ['1962'],
       ['1960'],
       ['1949'],
       ['1973'],
       ['1946'],
       ['NA'],
       ['NA'],
       ['2009'],
       ['1974'],
       ['1959'],
       ['1994'],
       ['1998'],
```

```
       ['2014'],
       ['1934'],
       ['1982'],
       ['2015'],
       ['1957'],
       ['1948'],
       ['2001'],
       ['2000'],
       ['1995'],
       ['1966'],
       ['1993'],
       ['1939'],
       ['2014'],
       ['1955'],
       ['1966'],
       ['2001'],
       ['2012'],
       ['1958'],
       ['1951'],
       ['1949'],
       ['1926'],
       ['2015'],
       ['1925'],
       ['1957'],
       ['2012'],
       ['2015'],
       ['1927'],
       ['1978'],
       ['2008'],
       ['1955'],
       ['2006'],
       ['1937'],
       ['1991'],
       ['1957'],
       ['III 2016'],
       ['2007'],
       ['1960'],
       ['2013'],
       ['1986'],
       ['2019'],
       ['1988'],
       ['1975'],
       ['2003'],
       ['2013'],
       ['2012'],
       ['2000'],
       ['1996'],
       ['1928'],
       ['2010'],
```

```
       ['2012'],
       ['2005'],
       ['1998'],
       ['1999'],
       ['1971'],
       ['2021'],
       ['2017'],
       ['2014'],
       ['2016'],
       ['1988'],
       ['2010'],
       ['1987'],
       ['2012'],
       ['2016'],
       ['2015'],
       ['2001'],
       ['2004'],
       ['2013'],
       ['2016'],
       ['1967'],
       ['2006'],
       ['I 2015'],
       ['2014'],
       ['2005'],
       ['2021'],
       ['1990'],
       ['1999'],
       ['2008'],
       ['1992'],
       ['2020'],
       ['1993'],
       ['1989'],
       ['1975'],
       ['1974'],
       ['1992'],
       ['2006'],
       ['1995'],
       ['2006'],
       ['2003'],
       ['1991'],
       ['1991'],
       ['1952'],
       ['1968'],
       ['2007'],
       ['1988'],
       ['1968'],
       ['1997'],
       ['2016'],
       ['2004'],
```

```
       ['1979'],
       ['1969'],
       ['2010'],
       ['2003'],
       ['1971'],
       ['2016'],
       ['1962'],
       ['2008'],
       ['1996'],
       ['1994'],
       ['1953'],
       ['1973'],
       ['I 2014'],
       ['1979'],
       ['1971'],
       ['1963'],
       ['1977'],
       ['1984'],
       ['1941'],
       ['1972'],
       ['1986'],
       ['1959'],
       ['1982'],
       ['2005'],
       ['1998'],
       ['1955'],
       ['1982'],
       ['1960'],
       ['1982'],
       ['1987'],
       ['1966'],
       ['2002'],
       ['1999'],
       ['2015'],
       ['1959'],
       ['2020'],
       ['1958'],
       ['1975'],
       ['1920'],
       ['1961'],
       ['1963'],
       ['1947'],
       ['1967'],
       ['2007'],
       ['1957'],
       ['2013'],
       ['2004'],
       ['1972'],
       ['1953'],
```

```
       ['1996'],
       ['2007'],
       ['2014'],
       ['I 2004'],
       ['2001'],
       ['1972'],
       ['1945'],
       ['1983'],
       ['1995'],
       ['2008'],
       ['2007'],
       ['2016'],
       ['1954'],
       ['1998'],
       ['2020'],
       ['2018'],
       ['1987'],
       ['1962'],
       ['2003'],
       ['2014'],
       ['2010'],
       ['1940'],
       ['1949'],
       ['1962'],
       ['1962'],
       ['1998'],
       ['2004'],
       ['1945'],
       ['2004'],
       ['1963'],
       ['2006'],
       ['2013'],
       ['2009'],
       ['1994'],
       ['2001'],
       ['1997'],
       ['2023'],
       ['2009'],
       ['2022'],
       ['2021'],
       ['II 2016'],
       ['2009'],
       ['2019'],
       ['2008'],
       ['2001'],
       ['2004'],
       ['2009'],
       ['1993'],
       ['1997'],
```

```
       ['1990'],
       ['2014'],
       ['2012'],
       ['2014'],
       ['2004'],
       ['2000'],
       ['2019'],
       ['2002'],
       ['NA'],
       ['NA'],
       ['1963'],
       ['1982'],
       ['2001'],
       ['1948'],
       ['2012'],
       ['2008'],
       ['2006'],
       ['1980'],
       ['2003'],
       ['2013'],
       ['2009'],
       ['2019'],
       ['I 2017'],
       ['1976'],
       ['1999'],
       ['1956'],
       ['I 2014'],
       ['1984'],
       ['1993'],
       ['2007'],
       ['1993'],
       ['1985'],
       ['1995'],
       ['1993'],
       ['2013'],
       ['1964'],
       ['1960'],
       ['2008'],
       ['1951'],
       ['1956'],
       ['2000'],
       ['2008'],
       ['1969'],
       ['1967'],
       ['1965'],
       ['2014'],
       ['1922'],
       ['1962'],
       ['1994'],
       ['2016'],
```

```
       ['1971'],
       ['2019'],
       ['1946'],
       ['2002'],
       ['1968'],
       ['I 2011'],
       ['1933'],
       ['2018'],
       ['1983'],
       ['1956'],
       ['2000'],
       ['1993'],
       ['1951'],
       ['2013'],
       ['1970'],
       ['1970'],
       ['1946'],
       ['2009'],
       ['1940'],
       ['1958'],
       ['1944'],
       ['2004'],
       ['1970'],
       ['2012'],
       ['2002'],
       ['2003'],
       ['1934'],
       ['1986'],
       ['1950'],
       ['1973'],
       ['1938'],
       ['2010'],
       ['2017'],
       ['1945'],
       ['1944'],
       ['1967'],
       ['1950'],
       ['1941'],
       ['2004'],
       ['2005'],
       ['2014'],
       ['1980'],
       ['2019'],
       ['1947'],
       ['2000'],
       ['1958'],
       ['2019'],
       ['1945'],
       ['2007'],
```

```
       ['1925'],
       ['1959'],
       ['1953'],
       ['2009'],
       ['1939'],
       ['2007'],
       ['2003'],
       ['1946'],
       ['I 2001'],
       ['2018'],
       ['1961'],
       ['2018'],
       ['I 2015'],
       ['2016'],
       ['2022'],
       ['2022'],
       ['1993'],
       ['2006'],
       ['2017'],
       ['1993'],
       ['2012'],
       ['1985'],
       ['2019'],
       ['I 2017'],
       ['2004'],
       ['2022'],
       ['I 2011'],
       ['2005'],
       ['2015'],
       ['2007'],
       ['2017'],
       ['2022'],
       ['I 2013'],
       ['2015'],
       ['1984'],
       ['2019'],
       ['2019'],
       ['2011'],
       ['1993'],
       ['1987'],
       ['2010'],
       ['2018'],
       ['2000'],
       ['1986'],
       ['2012'],
       ['2016'],
       ['2015'],
       ['2006'],
       ['2014'],
```

```
       ['1993'],
       ['2016'],
       ['2007'],
       ['2007'],
       ['1984'],
       ['2008'],
       ['2016'],
       ['1987'],
       ['I 2008'],
       ['2000'],
       ['1989'],
       ['2021'],
       ['1971'],
       ['1976'],
       ['2014'],
       ['2012'],
       ['2013'],
       ['2016'],
       ['2016'],
       ['2013'],
       ['1997'],
       ['1990'],
       ['2003'],
       ['2014'],
       ['2022'],
       ['1969'],
       ['1991'],
       ['2010'],
       ['1964'],
       ['2005'],
       ['1955'],
       ['2015'],
       ['1974'],
       ['1989'],
       ['1990'],
       ['2005'],
       ['2009'],
       ['1973'],
       ['1983'],
       ['I 2010'],
       ['1988'],
       ['1982'],
       ['I 2006'],
       ['2017'],
       ['2009'],
       ['1989'],
       ['2014'],
       ['1993'],
       ['I 2014'],
```

```
       ['2016'],
       ['1999'],
       ['2008'],
       ['1994'],
       ['1968'],
       ['1951'],
       ['2013'],
       ['1978'],
       ['1964'],
       ['1979'],
       ['1989'],
       ['1973'],
       ['1938'],
       ['1972'],
       ['1999'],
       ['1978'],
       ['1967'],
       ['I 2007'],
       ['1972'],
       ['1932'],
       ['1999'],
       ['1979'],
       ['2006'],
       ['2002'],
       ['1995'],
       ['1944'],
       ['2008'],
       ['1979'],
       ['1996'],
       ['2017'],
       ['2005'],
       ['1931'],
       ['1944'],
       ['1939'],
       ['1948'],
       ['2013'],
       ['2015'],
       ['1940'],
       ['1993'],
       ['1960'],
       ['1986'],
       ['1993'],
       ['1975'],
       ['2009'],
       ['1943'],
       ['2009'],
       ['1935'],
       ['1964'],
       ['2016'],
```

```
       ['1961'],
       ['1989'],
       ['1994'],
       ['1938'],
       ['2001'],
       ['2004'],
       ['2003'],
       ['1989'],
       ['2011'],
       ['1950'],
       ['2004'],
       ['1972'],
       ['2013'],
       ['2012'],
       ['1933'],
       ['1962'],
       ['2001'],
       ['1997'],
       ['2016'],
       ['1935'],
       ['2017'],
       ['2012'],
       ['2022'],
       ['2018'],
       ['2022'],
       ['1985'],
       ['II 2022'],
       ['2014'],
       ['2005'],
       ['2009'],
       ['2001'],
       ['2012'],
       ['2007'],
       ['2001'],
       ['2012'],
       ['1992'],
       ['2013'],
       ['1974'],
       ['2003'],
       ['1987'],
       ['2010'],
       ['2005'],
       ['1992'],
       ['1996'],
       ['2017'],
       ['2011'],
       ['2014'],
       ['2009'],
       ['2007'],
```

```
       ['1980'],
       ['2001'],
       ['1990'],
       ['2001'],
       ['1986'],
       ['1988'],
       ['2009'],
       ['1992'],
       ['1967'],
       ['1997'],
       ['2002'],
       ['2009'],
       ['2014'],
       ['1993'],
       ['2010'],
       ['2004'],
       ['2014'],
       ['1972'],
       ['1982'],
       ['1995'],
       ['1982'],
       ['2014'],
       ['2004'],
       ['2012'],
       ['I 1995'],
       ['2011'],
       ['2019'],
       ['1978'],
       ['2020'],
       ['2004'],
       ['2000'],
       ['I 2004'],
       ['1993'],
       ['1960'],
       ['1971'],
       ['2013'],
       ['1997'],
       ['2020'],
       ['1997'],
       ['2002'],
       ['1999'],
       ['1993'],
       ['1987'],
       ['1985'],
       ['2019'],
       ['1989'],
       ['2008'],
       ['2015'],
       ['2013'],
```

```
       ['1991'],
       ['2004'],
       ['1971'],
       ['1973'],
       ['2011'],
       ['1998'],
       ['1999'],
       ['2014'],
       ['1967'],
       ['2006'],
       ['2002'],
       ['1994'],
       ['1962'],
       ['2006'],
       ['I 2010'],
       ['1986'],
       ['1964'],
       ['1980'],
       ['1999'],
       ['1967'],
       ['1982'],
       ['2013'],
       ['1992'],
       ['2000'],
       ['1990'],
       ['2006'],
       ['2013'],
       ['I 2004'],
       ['2018'],
       ['2000'],
       ['1998'],
       ['1967'],
       ['1987'],
       ['1992'],
       ['2016'],
       ['1973'],
       ['2010'],
       ['2009'],
       ['1973'],
       ['1951'],
       ['1997'],
       ['2015'],
       ['2004'],
       ['2015'],
       ['1952'],
       ['1997'],
       ['1960'],
       ['2006'],
       ['1993'],
```

```
       ['1996'],
       ['1940'],
       ['2013'],
       ['1991'],
       ['1964'],
       ['2001'],
       ['1992'],
       ['1988'],
       ['1956'],
       ['2014'],
       ['2001'],
       ['2016'],
       ['1966'],
       ['1991'],
       ['1947'],
       ['2012'],
       ['2003'],
       ['2003'],
       ['2015'],
       ['1962'],
       ['1932'],
       ['1955'],
       ['2004'],
       ['1938'],
       ['1986'],
       ['1954'],
       ['2008'],
       ['1997'],
       ['1948'],
       ['2016'],
       ['2010'],
       ['1990'],
       ['2001'],
       ['2003'],
       ['2005'],
       ['2006'],
       ['2002'],
       ['2006'],
       ['2002'],
       ['2017'],
       ['1962'],
       ['1975'],
       ['1997'],
       ['1985'],
       ['1992'],
       ['2013'],
       ['1983'],
       ['2007'],
       ['2022'],
```

```
       ['2001'],
       ['2019'],
       ['2000'],
       ['1997'],
       ['2007'],
       ['2001'],
       ['2006'],
       ['2009'],
       ['2018'],
       ['2017'],
       ['I 2016'],
       ['1987'],
       ['2010'],
       ['2015'],
       ['1992'],
       ['1998'],
       ['2004'],
       ['1999'],
       ['2011'],
       ['2018'],
       ['2004'],
       ['2006'],
       ['2005'],
       ['2007'],
       ['2010'],
       ['2008'],
       ['2002'],
       ['II 2016'],
       ['2009'],
       ['2007'],
       ['2001'],
       ['1963'],
       ['1998'],
       ['1986'],
       ['1987'],
       ['2014'],
       ['2021'],
       ['1998'],
       ['2010'],
       ['III 2018'],
       ['2011'],
       ['1981'],
       ['1998'],
       ['2007'],
       ['2011'],
       ['2008'],
       ['1967'],
       ['1973'],
       ['2012'],
```

```
       ['2019'],
       ['2014'],
       ['2001'],
       ['2005'],
       ['1995'],
       ['I 1985'],
       ['1954'],
       ['1995'],
       ['1955'],
       ['1937'],
       ['1953'],
       ['2011'],
       ['1991'],
       ['2003'],
       ['2006'],
       ['2006'],
       ['2002'],
       ['1974'],
       ['2007'],
       ['2004'],
       ['1996'],
       ['2013'],
       ['1933'],
       ['2009']], dtype=object)
```

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

```
array(['1920', '1921', '1922', '1925', '1926', '1927', '1928', '1931',
       '1932', '1933', '1934', '1935', '1937', '1938', '1939', '1940',
       '1941', '1942', '1943', '1944', '1945', '1946', '1947', '1948',
       '1949', '1950', '1951', '1952', '1953', '1954', '1955', '1956',
       '1957', '1958', '1959', '1960', '1961', '1962', '1963', '1964',
       '1965', '1966', '1967', '1968', '1969', '1970', '1971', '1972',
       '1973', '1974', '1975', '1976', '1977', '1978', '1979', '1980',
       '1981', '1982', '1983', '1984', '1985', '1986', '1987', '1988',
       '1989', '1990', '1991', '1992', '1993', '1994', '1995', '1996',
       '1997', '1998', '1999', '2000', '2001', '2002', '2003', '2004',
       '2005', '2006', '2007', '2008', '2009', '2010', '2011', '2012',
       '2013', '2014', '2015', '2016', '2017', '2018', '2019', '2020',
       '2021', '2022', '2023', 'I 1985', 'I 1995', 'I 2001', 'I 2004',
       'I 2006', 'I 2007', 'I 2008', 'I 2010', 'I 2011', 'I 2013',
       'I 2014', 'I 2015', 'I 2016', 'I 2017', 'I 2019', 'I 2020',
       'II 2016', 'II 2018', 'II 2022', 'III 2016', 'III 2018', 'NA'],
      dtype=object)
```

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

```
6
```

Преобразование категориальных признаков в числовые

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

```
       c1
0    1994
1    1972
2    2008
3    1993
4    1957
..    ...
995  2004
996  1996
997  2013
998  1933
999  2009

[1000 rows x 1 columns]
```

Кодирование категорий целочисленными значениями (label encoding)

В этом случае уникальные значения категориального признака кодируются целыми числами.

В scikit-learn для такого кодирования используется два класса :

- LabelEncoder - который ориентирован на применение к одному признаку. Этот класс прежде всего предназначен для кодирования целевого признака, но может быть также использован для последовательного кодирования отдельных нецелевых признаков.
- OrdinalEncoder - который ориентирован на применение к матрице объект-признак, то есть для кодирования матрицы нецелевых признаков.

Использование LabelEncoder

```
from sklearn.preprocessing import LabelEncoder

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

```
array(['1994', '1972', '2008', '1993', '1957', '2003', '1974', '2023',
       '2010', '2001', '1999', '1966', '2002', '2021', '2022', '2014',
       '1990', '1975', '1980', '2020', '1991', '1995', '1998', '1977',
       '1997', '1954', '1946', '1962', '1985', '2006', '2000', '1979',
       '2019', '1942', '1960', '2011', '1968', '1988', '2004', '1931',
       'II 2018', '1981', '2018', '2012', '1986', 'I 2019', '1984',
```

```
       'I 2017', '2016', '2009', '1964', '1950', '1940', '1963',
'1987',
       '1983', '1971', '1958', '1992', '1959', '1941', '1952', '1927',
       '1973', '1961', '1944', '2007', '1948', '2017', '1921', '2013',
       '1989', '1982', '2005', '1976', 'I 2020', '1939', '1965',
'2015',
       '1955', '1969', '1953', '1928', 'I 2015', '1996', '1978', 'I
2013',
       '1967', '1949', '1934', '1951', '1926', '1925', '1937', 'III
2016',
       'I 2014', '1920', '1947', 'I 2004', '1945', 'II 2016', '1956',
       '1922', 'I 2011', '1933', '1970', '1938', 'I 2001', 'I 2008',
       'I 2010', 'I 2006', 'I 2007', '1932', '1943', '1935', 'II
2022',
       'I 1995', 'I 2016', 'III 2018', 'I 1985'], dtype=object)
```

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

```
# Наименования категорий в соответствии с порядковыми номерами

# Свойство называется classes, потому что предполагается что мы решаем

# задачу классификации и каждое значение категории соответствует
# какому-либо классу целевого признака
```

```
le.classes_
```

```
array(['1920', '1921', '1922', '1925', '1926', '1927', '1928', '1931',
       '1932', '1933', '1934', '1935', '1937', '1938', '1939', '1940',
       '1941', '1942', '1943', '1944', '1945', '1946', '1947', '1948',
       '1949', '1950', '1951', '1952', '1953', '1954', '1955', '1956',
       '1957', '1958', '1959', '1960', '1961', '1962', '1963', '1964',
       '1965', '1966', '1967', '1968', '1969', '1970', '1971', '1972',
       '1973', '1974', '1975', '1976', '1977', '1978', '1979', '1980',
       '1981', '1982', '1983', '1984', '1985', '1986', '1987', '1988',
       '1989', '1990', '1991', '1992', '1993', '1994', '1995', '1996',
       '1997', '1998', '1999', '2000', '2001', '2002', '2003', '2004',
       '2005', '2006', '2007', '2008', '2009', '2010', '2011', '2012',
       '2013', '2014', '2015', '2016', '2017', '2018', '2019', '2020',
       '2021', '2022', '2023', 'I 1985', 'I 1995', 'I 2001', 'I 2004',
       'I 2006', 'I 2007', 'I 2008', 'I 2010', 'I 2011', 'I 2013',
       'I 2014', 'I 2015', 'I 2016', 'I 2017', 'I 2019', 'I 2020',
       'II 2016', 'II 2018', 'II 2022', 'III 2016', 'III 2018'],
      dtype=object)
```

```
cat_enc_le
```

```
array([ 69,  47,  83,  68,  32,  78,  49,  98,  69,  85,  76,  74,
69,
        41,  77,  96,  97,  89,  65,  74,  50,  55,  97,  95,  74,
```

```
66,
      70,  73,  66,  52,  76,  77,  72,  29,  21,  37,  97,  60,
81,
      75,  54,  94,  81,  89,  69,  69,  70,  17,  73,  35,  77,
86,
      43,  63,  29,  63,  79,   7, 116,  56,  93,  94,  87,  54,
61,
     113,  87,  93,  55,  78,  75,  59,  60, 112,  91,  83,  84,
81,
      39,  56,  93,  25,  32,  32,  15,  38,  94,  96,  94,  96,
97,
      70,  84,  62,  58,  74,  75,  72,  79,  46,  33,  70,  43,
67,
      59,  87,  70,  95,  84,  58,  76,  85,  37,  85,  72,  34,
37,
      16,  27,   5,  48,  86,   7,  35,  91,  27,  36,  97,  19,
82,
      23,  92,  97,  97,  94,   1,  96,  93,  88,  98,  64,  88,
96,
      82,  68,  85,  82,  57,  80,  51,  94,  70,  75,  79,  67,
73,
      93,  78,  72,  74,  78,  63,  76, 114,  79,  14,  49,  80,
81,
      38,  34,  50,  55,  95,  84,  40,  60,  94,  97,  29,  25,
92,
      93,  87,  25,  23,  36,  17,  90,  93,  86,  94,  30,  44,
28,
      72,  94,  79,  28,  93,   6,  80,  79,  50,  27,  92,  87,
93,
      88,  50,  89,  88,  78,  57,  59,  90,  89,  94,  86,  92,
77,
      73,  91,  61,  82,  86,  48, 110,  91,  51,  29,  71,  71,
14,
      92,  64,  82,  53,  76,  78, 110,  70, 108,  88, 110,  40,
61,
      85,  73,  86,  94,  83,  79,  50,  75,  70,  84,  63,  55,
42,
      74,  89,  79,  54,  91,  51,  34,  59,  32,  32,  41,  68,
79,
      75,  51,  15,  72,  37,  35,  24,  48,  21,  79,  79,  84,
49,
      34,  69,  73,  89,  10,  57,  90,  32,  23,  76,  75,  70,
41,
      68,  14,  89,  30,  41,  76,  87,  33,  26,  24,   4,  90,
3,
      32,  87,  90,   5,  53,  83,  30,  81,  12,  66,  32, 118,
82,
      35,  88,  61,  94,  63,  50,  78,  88,  87,  75,  71,   6,
85,
```

```
        87,  80,  73,  74,  46,  96,  92,  89,  91,  63,  85,  62,
87,
        91,  90,  76,  79,  88,  91,  42,  81, 110,  89,  80,  96,
65,
        74,  83,  67,  95,  68,  64,  50,  49,  67,  81,  70,  81,
78,
        66,  66,  27,  43,  82,  63,  43,  72,  91,  79,  54,  44,
85,
        78,  46,  91,  37,  83,  71,  69,  28,  48, 109,  54,  46,
38,
        52,  59,  16,  47,  61,  34,  57,  80,  73,  30,  57,  35,
57,
        62,  41,  77,  74,  90,  34,  95,  33,  50,   0,  36,  38,
22,
        42,  82,  32,  88,  79,  47,  28,  71,  82,  89, 102,  76,
47,
        20,  58,  70,  83,  82,  91,  29,  73,  95,  93,  62,  37,
78,
        89,  85,  15,  24,  37,  37,  73,  79,  20,  79,  38,  81,
88,
        84,  69,  76,  72,  98,  84,  97,  96, 115,  84,  94,  83,
76,
        79,  84,  68,  72,  65,  89,  87,  89,  79,  75,  94,  77,
79,
        79,  38,  57,  76,  23,  87,  83,  81,  55,  78,  88,  84,
94,
       112,  51,  74,  31, 109,  59,  68,  82,  68,  60,  70,  68,
88,
        39,  35,  83,  26,  31,  75,  83,  44,  42,  40,  89,   2,
37,
        69,  91,  46,  94,  21,  77,  43, 107,   9,  93,  58,  31,
75,
        68,  26,  88,  45,  45,  21,  84,  15,  33,  19,  79,  45,
87,
        77,  78,  10,  61,  25,  48,  13,  85,  92,  20,  19,  42,
25,
        16,  79,  80,  89,  55,  94,  22,  75,  33,  94,  20,  82,
3,
        34,  28,  84,  14,  82,  78,  21, 101,  93,  36,  93, 110,
91,
        97,  97,  68,  81,  92,  68,  87,  60,  94, 112,  79,  97,
107,
        80,  90,  82,  92,  97, 108,  90,  59,  94,  94,  86,  68,
62,
        85,  93,  75,  61,  87,  91,  90,  81,  89,  68,  91,  82,
82,
        59,  83,  91,  62, 105,  75,  64,  96,  46,  51,  89,  87,
88,
        91,  91,  88,  72,  65,  78,  89,  97,  44,  66,  85,  39,
```

```
80,
        30,  90,  49,  64,  65,  80,  84,  48,  58, 106,  63,  57,
103,
        92,  84,  64,  89,  68, 109,  91,  74,  83,  69,  43,  26,
88,
        53,  39,  54,  64,  48,  13,  47,  74,  53,  42, 104,  47,
8,
        74,  54,  81,  77,  70,  19,  83,  54,  71,  92,  80,   7,
19,
        14,  23,  88,  90,  15,  68,  35,  61,  68,  50,  84,  18,
84,
        11,  39,  91,  36,  64,  69,  13,  76,  79,  78,  64,  86,
25,
        79,  47,  88,  87,   9,  37,  76,  72,  91,  11,  92,  87,
97,
        93,  97,  60, 117,  89,  80,  84,  76,  87,  82,  76,  87,
67,
        88,  49,  78,  62,  85,  80,  67,  71,  92,  86,  89,  84,
82,
        55,  76,  65,  76,  61,  63,  84,  67,  42,  72,  77,  84,
89,
        68,  85,  79,  89,  47,  57,  70,  57,  89,  79,  87, 100,
86,
        94,  53,  95,  79,  75, 102,  68,  35,  46,  88,  72,  95,
72,
        77,  74,  68,  62,  60,  94,  64,  83,  90,  88,  66,  79,
46,
        48,  86,  73,  74,  89,  42,  81,  77,  69,  37,  81, 106,
61,
        39,  55,  74,  42,  57,  88,  67,  75,  65,  81,  88, 102,
93,
        75,  73,  42,  62,  67,  91,  48,  85,  84,  48,  26,  72,
90,
        79,  90,  27,  72,  35,  81,  68,  71,  15,  88,  66,  39,
76,
        67,  63,  31,  89,  76,  91,  41,  66,  22,  87,  78,  78,
90,
        37,   8,  30,  79,  13,  61,  29,  83,  72,  23,  91,  85,
65,
        76,  78,  80,  81,  77,  81,  77,  92,  37,  50,  72,  60,
67,
        88,  58,  82,  97,  76,  94,  75,  72,  82,  76,  81,  84,
93,
        92, 111,  62,  85,  90,  67,  73,  79,  74,  86,  93,  79,
81,
        80,  82,  85,  83,  77, 115,  84,  82,  76,  38,  73,  61,
62,
        89,  96,  73,  85, 119,  86,  56,  73,  82,  86,  83,  42,
48,
        87,  94,  89,  76,  80,  70,  99,  29,  70,  30,  12,  28,
```

```
86,
        66,  78,  81,  81,  77,  49,  82,  79,  71,  88,   9,  84])

np.unique(cat_enc_le)

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,
103,
       104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115,
116,
       117, 118, 119])

# В этом примере видно, что перед кодированием
# уникальные значения признака сортируются в лексикографическом порядке
le.inverse_transform([0, 1, 2, 3])

array(['1920', '1921', '1922', '1925'], dtype=object)
```

Использование OrdinalEncoder

```
from sklearn.preprocessing import OrdinalEncoder

data_oe = data[['title', 'director', 'genre']]
data_oe.head()

                     title              director
genre
0  The Shawshank Redemption        Frank Darabont
Drama
1             The Godfather  Francis Ford Coppola                Crime,
Drama
2           The Dark Knight     Christopher Nolan       Action, Crime,
Drama
3          Schindler's List      Steven Spielberg  Biography, Drama,
History
4              12 Angry Men          Sidney Lumet                Crime,
Drama
```

```
imp4 = SimpleImputer(missing_values=np.nan, strategy='constant',
fill_value='NA')
data_oe_filled = imp4.fit_transform(data_oe)
data_oe_filled
```

```
array([['The Shawshank Redemption', 'Frank Darabont', 'Drama'],
       ['The Godfather', 'Francis Ford Coppola', 'Crime, Drama'],
       ['The Dark Knight', 'Christopher Nolan', 'Action, Crime,
Drama'],
       ...,
       ['Philomena', 'Stephen Frears', 'Biography, Comedy, Drama'],
       ['The Invisible Man', 'James Whale', 'Horror, Sci-Fi'],
       ['Cell 211', 'Daniel Monzón', 'NA']], dtype=object)
```

```
oe = OrdinalEncoder()
cat_enc_oe = oe.fit_transform(data_oe_filled)
cat_enc_oe
```

```
array([[864., 141., 133.],
       [756., 139., 120.],
       [728.,  77.,  20.],
       ...,
       [546., 475.,  80.],
       [785., 210., 188.],
       [162.,  92., 193.]])
```

```
# Уникальные значения 1 признака
np.unique(cat_enc_oe[:, 0])
```

```
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., 103., 104., 105., 106., 107., 108.,
109.,
       110., 111., 112., 113., 114., 115., 116., 117., 118., 119.,
```

```
120.,
       121., 122., 123., 124., 125., 126., 127., 128., 129., 130.,
131.,
       132., 133., 134., 135., 136., 137., 138., 139., 140., 141.,
142.,
       143., 144., 145., 146., 147., 148., 149., 150., 151., 152.,
153.,
       154., 155., 156., 157., 158., 159., 160., 161., 162., 163.,
164.,
       165., 166., 167., 168., 169., 170., 171., 172., 173., 174.,
175.,
       176., 177., 178., 179., 180., 181., 182., 183., 184., 185.,
186.,
       187., 188., 189., 190., 191., 192., 193., 194., 195., 196.,
197.,
       198., 199., 200., 201., 202., 203., 204., 205., 206., 207.,
208.,
       209., 210., 211., 212., 213., 214., 215., 216., 217., 218.,
219.,
       220., 221., 222., 223., 224., 225., 226., 227., 228., 229.,
230.,
       231., 232., 233., 234., 235., 236., 237., 238., 239., 240.,
241.,
       242., 243., 244., 245., 246., 247., 248., 249., 250., 251.,
252.,
       253., 254., 255., 256., 257., 258., 259., 260., 261., 262.,
263.,
       264., 265., 266., 267., 268., 269., 270., 271., 272., 273.,
274.,
       275., 276., 277., 278., 279., 280., 281., 282., 283., 284.,
285.,
       286., 287., 288., 289., 290., 291., 292., 293., 294., 295.,
296.,
       297., 298., 299., 300., 301., 302., 303., 304., 305., 306.,
307.,
       308., 309., 310., 311., 312., 313., 314., 315., 316., 317.,
318.,
       319., 320., 321., 322., 323., 324., 325., 326., 327., 328.,
329.,
       330., 331., 332., 333., 334., 335., 336., 337., 338., 339.,
340.,
       341., 342., 343., 344., 345., 346., 347., 348., 349., 350.,
351.,
       352., 353., 354., 355., 356., 357., 358., 359., 360., 361.,
362.,
       363., 364., 365., 366., 367., 368., 369., 370., 371., 372.,
373.,
       374., 375., 376., 377., 378., 379., 380., 381., 382., 383.,
384.,
```

```
       385., 386., 387., 388., 389., 390., 391., 392., 393., 394.,
395.,
       396., 397., 398., 399., 400., 401., 402., 403., 404., 405.,
406.,
       407., 408., 409., 410., 411., 412., 413., 414., 415., 416.,
417.,
       418., 419., 420., 421., 422., 423., 424., 425., 426., 427.,
428.,
       429., 430., 431., 432., 433., 434., 435., 436., 437., 438.,
439.,
       440., 441., 442., 443., 444., 445., 446., 447., 448., 449.,
450.,
       451., 452., 453., 454., 455., 456., 457., 458., 459., 460.,
461.,
       462., 463., 464., 465., 466., 467., 468., 469., 470., 471.,
472.,
       473., 474., 475., 476., 477., 478., 479., 480., 481., 482.,
483.,
       484., 485., 486., 487., 488., 489., 490., 491., 492., 493.,
494.,
       495., 496., 497., 498., 499., 500., 501., 502., 503., 504.,
505.,
       506., 507., 508., 509., 510., 511., 512., 513., 514., 515.,
516.,
       517., 518., 519., 520., 521., 522., 523., 524., 525., 526.,
527.,
       528., 529., 530., 531., 532., 533., 534., 535., 536., 537.,
538.,
       539., 540., 541., 542., 543., 544., 545., 546., 547., 548.,
549.,
       550., 551., 552., 553., 554., 555., 556., 557., 558., 559.,
560.,
       561., 562., 563., 564., 565., 566., 567., 568., 569., 570.,
571.,
       572., 573., 574., 575., 576., 577., 578., 579., 580., 581.,
582.,
       583., 584., 585., 586., 587., 588., 589., 590., 591., 592.,
593.,
       594., 595., 596., 597., 598., 599., 600., 601., 602., 603.,
604.,
       605., 606., 607., 608., 609., 610., 611., 612., 613., 614.,
615.,
       616., 617., 618., 619., 620., 621., 622., 623., 624., 625.,
626.,
       627., 628., 629., 630., 631., 632., 633., 634., 635., 636.,
637.,
       638., 639., 640., 641., 642., 643., 644., 645., 646., 647.,
648.,
       649., 650., 651., 652., 653., 654., 655., 656., 657., 658.,
```

```
659.,
       660., 661., 662., 663., 664., 665., 666., 667., 668., 669.,
670.,
       671., 672., 673., 674., 675., 676., 677., 678., 679., 680.,
681.,
       682., 683., 684., 685., 686., 687., 688., 689., 690., 691.,
692.,
       693., 694., 695., 696., 697., 698., 699., 700., 701., 702.,
703.,
       704., 705., 706., 707., 708., 709., 710., 711., 712., 713.,
714.,
       715., 716., 717., 718., 719., 720., 721., 722., 723., 724.,
725.,
       726., 727., 728., 729., 730., 731., 732., 733., 734., 735.,
736.,
       737., 738., 739., 740., 741., 742., 743., 744., 745., 746.,
747.,
       748., 749., 750., 751., 752., 753., 754., 755., 756., 757.,
758.,
       759., 760., 761., 762., 763., 764., 765., 766., 767., 768.,
769.,
       770., 771., 772., 773., 774., 775., 776., 777., 778., 779.,
780.,
       781., 782., 783., 784., 785., 786., 787., 788., 789., 790.,
791.,
       792., 793., 794., 795., 796., 797., 798., 799., 800., 801.,
802.,
       803., 804., 805., 806., 807., 808., 809., 810., 811., 812.,
813.,
       814., 815., 816., 817., 818., 819., 820., 821., 822., 823.,
824.,
       825., 826., 827., 828., 829., 830., 831., 832., 833., 834.,
835.,
       836., 837., 838., 839., 840., 841., 842., 843., 844., 845.,
846.,
       847., 848., 849., 850., 851., 852., 853., 854., 855., 856.,
857.,
       858., 859., 860., 861., 862., 863., 864., 865., 866., 867.,
868.,
       869., 870., 871., 872., 873., 874., 875., 876., 877., 878.,
879.,
       880., 881., 882., 883., 884., 885., 886., 887., 888., 889.,
890.,
       891., 892., 893., 894., 895., 896., 897., 898., 899., 900.,
901.,
       902., 903., 904., 905., 906., 907., 908., 909., 910., 911.,
912.,
       913., 914., 915., 916., 917., 918., 919., 920., 921., 922.,
923.,
```

```
       924., 925., 926., 927., 928., 929., 930., 931., 932., 933.,
934.,
       935., 936., 937., 938., 939., 940., 941., 942., 943., 944.,
945.,
       946., 947., 948., 949., 950., 951., 952., 953., 954., 955.,
956.,
       957., 958., 959., 960., 961., 962., 963., 964., 965., 966.,
967.,
       968., 969., 970., 971., 972., 973., 974., 975., 976., 977.,
978.,
       979., 980., 981., 982., 983., 984., 985., 986., 987., 988.,
989.,
       990.])
```

```
# Уникальные значения 2 признака
np.unique(cat_enc_oe[:, 1])
```

```
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., 103., 104., 105., 106., 107., 108.,
109.,
       110., 111., 112., 113., 114., 115., 116., 117., 118., 119.,
120.,
       121., 122., 123., 124., 125., 126., 127., 128., 129., 130.,
131.,
       132., 133., 134., 135., 136., 137., 138., 139., 140., 141.,
142.,
       143., 144., 145., 146., 147., 148., 149., 150., 151., 152.,
153.,
       154., 155., 156., 157., 158., 159., 160., 161., 162., 163.,
164.,
       165., 166., 167., 168., 169., 170., 171., 172., 173., 174.,
175.,
       176., 177., 178., 179., 180., 181., 182., 183., 184., 185.,
```

```
186.,
       187., 188., 189., 190., 191., 192., 193., 194., 195., 196.,
197.,
       198., 199., 200., 201., 202., 203., 204., 205., 206., 207.,
208.,
       209., 210., 211., 212., 213., 214., 215., 216., 217., 218.,
219.,
       220., 221., 222., 223., 224., 225., 226., 227., 228., 229.,
230.,
       231., 232., 233., 234., 235., 236., 237., 238., 239., 240.,
241.,
       242., 243., 244., 245., 246., 247., 248., 249., 250., 251.,
252.,
       253., 254., 255., 256., 257., 258., 259., 260., 261., 262.,
263.,
       264., 265., 266., 267., 268., 269., 270., 271., 272., 273.,
274.,
       275., 276., 277., 278., 279., 280., 281., 282., 283., 284.,
285.,
       286., 287., 288., 289., 290., 291., 292., 293., 294., 295.,
296.,
       297., 298., 299., 300., 301., 302., 303., 304., 305., 306.,
307.,
       308., 309., 310., 311., 312., 313., 314., 315., 316., 317.,
318.,
       319., 320., 321., 322., 323., 324., 325., 326., 327., 328.,
329.,
       330., 331., 332., 333., 334., 335., 336., 337., 338., 339.,
340.,
       341., 342., 343., 344., 345., 346., 347., 348., 349., 350.,
351.,
       352., 353., 354., 355., 356., 357., 358., 359., 360., 361.,
362.,
       363., 364., 365., 366., 367., 368., 369., 370., 371., 372.,
373.,
       374., 375., 376., 377., 378., 379., 380., 381., 382., 383.,
384.,
       385., 386., 387., 388., 389., 390., 391., 392., 393., 394.,
395.,
       396., 397., 398., 399., 400., 401., 402., 403., 404., 405.,
406.,
       407., 408., 409., 410., 411., 412., 413., 414., 415., 416.,
417.,
       418., 419., 420., 421., 422., 423., 424., 425., 426., 427.,
428.,
       429., 430., 431., 432., 433., 434., 435., 436., 437., 438.,
439.,
       440., 441., 442., 443., 444., 445., 446., 447., 448., 449.,
450.,
```

```
       451., 452., 453., 454., 455., 456., 457., 458., 459., 460.,
461.,
       462., 463., 464., 465., 466., 467., 468., 469., 470., 471.,
472.,
       473., 474., 475., 476., 477., 478., 479., 480., 481., 482.,
483.,
       484., 485., 486., 487., 488., 489., 490., 491., 492., 493.,
494.,
       495., 496., 497., 498., 499., 500., 501., 502., 503., 504.,
505.,
       506., 507., 508., 509., 510., 511., 512., 513., 514., 515.,
516.,
       517., 518., 519., 520., 521., 522., 523., 524., 525., 526.,
527.,
       528., 529., 530., 531., 532., 533., 534., 535., 536., 537.,
538.,
       539., 540., 541., 542., 543., 544., 545., 546., 547., 548.,
549.,
       550., 551., 552., 553., 554., 555., 556., 557.])
```

```
# Уникальные значения 3 признака
np.unique(cat_enc_oe[:, 2])
```

```
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., 103., 104., 105., 106., 107., 108.,
109.,
       110., 111., 112., 113., 114., 115., 116., 117., 118., 119.,
120.,
       121., 122., 123., 124., 125., 126., 127., 128., 129., 130.,
131.,
       132., 133., 134., 135., 136., 137., 138., 139., 140., 141.,
142.,
       143., 144., 145., 146., 147., 148., 149., 150., 151., 152.,
```

```
153.,
       154., 155., 156., 157., 158., 159., 160., 161., 162., 163.,
164.,
       165., 166., 167., 168., 169., 170., 171., 172., 173., 174.,
175.,
       176., 177., 178., 179., 180., 181., 182., 183., 184., 185.,
186.,
       187., 188., 189., 190., 191., 192., 193., 194., 195.])

# Наименования категорий в соответствии с порядковыми номерами
oe.categories_

[array(['12 Angry Men', '12 Monkeys', '12 Years a Slave', '1917',
        '2001: A Space Odyssey', '21 Grams', '25th Hour', '3 Idiots',
        '3-Iron', '300', '3:10 to Yuma', '4 Months, 3 Weeks and 2
Days',
        '50/50', '500 Days of Summer', '777 Charlie', '8½', 'A
Beautifuld',
        'A Bronx Tale', 'A Christmas Story', 'A Clockwork Orange',
        'A Few Good Men', 'A Fistful of Dollars', 'A Man Called Ove',
        'A Man for All Seasons', 'A Night at the Opera', 'A Prophet',
        'A Separation', 'A Silent Voice: The Movie', 'A Star Is Born',
        'A Streetcar Named Desire', 'A Taxi Driver',
        'A Very Long Engagement', 'A Wednesday',
        'A Woman Under the Influence', 'About Elly', 'About Time',
        'Ace in the Hole', "Adam's Apples", 'Adaptation.', 'After
Hours',
        'After the Wedding', 'Aftersun', 'Aguirre, the Wrath of God',
        'Airlift', 'Airplane!', 'Akira', 'Aladdin', 'Alien', 'Aliens',
        'All About Eve', 'All About My Mother',
        'All Quiet on the Western Front', 'All That Jazz',
        "All the President's Men", 'Almost Famous', 'Amadeus',
        'American Beauty', 'American Gangster', 'American History X',
        'American Psycho', 'Amores Perros', 'Amour', 'Amélie',
'Anand',
        'Anatomy of a Murder', 'And Your Mother Too', 'Andaz Apna
Apna',
        'Andhadhun', 'Andrei Rublev', 'Annie Hall', 'Another Round',
        'Apocalypse Now', 'Apocalypto', 'Apollo 13', 'Argo',
'Arrival',
        'Arsenic and Old Lace', 'Article 15', 'As Good as It Gets',
        'Asuran', 'Atonement', 'Autumn Sonata', 'Avatar',
        'Avatar: The Way of Water', 'Avengers: Endgame',
        'Avengers: Infinity War', 'Awakenings',
        'Ayla: The Daughter of War', 'Baahubali 2: The Conclusion',
        'Baahubali: The Beginning', 'Baby', 'Back to the Future',
        'Back to the Future Part II', 'Badhaai Ho', 'Badlands',
        'Bajrangi Bhaijaan', 'Barfi!', 'Barry Lyndon', 'Barton Fink',
        'Batman Begins', 'Batman: Mask of the Phantasm',
        'Battleship Potemkin', 'Beasts of No Nation',
```

```
        'Beauty and the Beast', 'Before Midnight', 'Before Sunrise',
        'Before Sunset', 'Being John Malkovich', 'Being There',
        'Belle de Jour', 'Ben-Hur', 'Bhaag Milkha Bhaag',
        'Bicycle Thieves', 'Big Fish', 'Big Hero 6', 'Billy Elliot',
        'Birdman or The Unexpected Virtue of Ignorance', 'Black',
        'Black Book', 'Black Cat, White Cat', 'Black Hawk Down',
        'Black Narcissus', 'Black Swan', 'Blade Runner',
        'Blade Runner 2049', 'Blazing Saddles', 'Blood Diamond',
        'Blue Is the Warmest Colour', 'Blue Velvet', 'Bohemian
Rhapsody',
        'Bonnie and Clyde', 'Boogie Nights', 'Bound by Honor',
'Boyhood',
        'Boyz n the Hood', 'Braveheart', 'Brazil', 'Breaking the
Waves',
        'Breathless', 'Brief Encounter', 'Bringing Up Baby',
        'Brokeback Mountain', 'Butch Cassidy and the Sundance Kid',
        'C.R.A.Z.Y.', 'CODA', 'Cabaret', 'Call Me by Your Name',
        'Cape Fear', 'Capernaum', 'Captain America: Civil War',
        'Captain America: The Winter Soldier', 'Captain Fantastic',
        'Captain Phillips', "Carlito's Way", 'Carry On, Munna Bhai',
        'Casablanca', 'Casino', 'Casino Royale', 'Cast Away',
        'Castle in the Sky', 'Cat on a Hot Tin Roof',
        'Catch Me If You Can', 'Cell 211', 'Central Station',
'Changeling',
        'Charade', 'Chhichhore', 'Children of Heaven', 'Children of
Men',
        'Chinatown', 'Chungking Express', 'Cinderella Man',
        'Cinema Paradiso', 'Citizen Kane', 'City Lights', 'City of
God',
        'Clerks', 'Coco', 'Come and See', 'Confessions', 'Control',
        'Cool Hand Luke', 'Coraline', 'Cowboy Bebop: The Movie',
'Crash',
        'Cries & Whispers', 'Crimes and Misdemeanors',
        'Crouching Tiger, Hidden Dragon', 'Dallas Buyers Club',
        'Dancer in the Dark', 'Dances with Wolves', 'Dangal', 'Dark
City',
        'Dark Waters', 'Das Boot', 'Dawn of the Dead',
        'Dawn of the Planet of the Apes', 'Days of Heaven',
        'Dead Poets Society', 'Deadpool', 'Deadpool 2', 'Deliverance',
        'Demon Slayer the Movie: Mugen Train', 'Departures', 'Dersu
Uzala',
        'Despicable Me', 'Detachment', 'Dev.D', 'Diabolique',
        'Dial M for Murder', 'Die Hard', 'Die Hard with a Vengeance',
        'Dil Chahta Hai', 'Dilwale Dulhania Le Jayenge', 'Dirty
Harry',
        'District 9', 'Django Unchained', 'Do the Right Thing',
        'Doctor Zhivago', 'Dog Day Afternoon', 'Dogville', 'Donnie
Brasco',
        'Donnie Darko', 'Double Indemnity', 'Down by Law', 'Downfall',
```

```
        'Dr. Strangelove or: How I Learned to Stop Worrying and Love
the Bomb',
        'Dreams', 'Drishyam', 'Drishyam 2', 'Drive', 'Duck Soup',
'Dune',
        'Dunkirk', 'E.T. the Extra-Terrestrial', 'East of Eden',
        'Eastern Promises', 'Ed Wood', 'Edge of Tomorrow',
        'Edward Scissorhands', 'Elevator to the Gallows', 'Elite
Squad',
        'Elite Squad 2: The Enemy Within', 'Empire of the Sun',
        'End of Watch', 'English Vinglish', 'Enter the Dragon',
        'Eternal Sunshine of the Spotlessd',
        'Everything Everywhere All at Once', "Everything's Gonna Be
Great",
        'Evil', 'Evil Dead II', 'Ex Machina', 'Fanny and Alexander',
        'Fantasia', 'Fantastic Mr. Fox', 'Fantastic Planet',
        'Farewell My Concubine', 'Fargo', "Ferris Bueller's Day Off",
        'Fiddler on the Roof', 'Fight Club', 'Finding Nemo',
        'Finding Neverland', 'Fireworks', 'First Blood',
'Fitzcarraldo',
        'Flipped', 'For a Few Dollars More', 'Ford v Ferrari',
        'Forrest Gump', 'Frankenstein', 'Freaks', 'Fried Green
Tomatoes',
        'From Here to Eternity', 'Frost/Nixon', 'Full Metal Jacket',
        'G.O.R.A.', 'Gandhi', 'Gangs of Wasseypur', 'Gaslight',
'Gattaca',
        'Get Out', 'Ghost in the Shell', 'Ghostbusters', 'Gladiator',
        'Glengarry Glen Ross', 'Glory', 'Goldfinger', 'Gone Baby
Gone',
        'Gone Girl', 'Gone with the Wind', 'Good Bye Lenin!',
        'Good Will Hunting', 'Goodbye, Children', 'Goodfellas',
        'Gran Torino', 'Grave of the Fireflies', 'Gravity', 'Green
Book',
        'Groundhog Day', 'Guardians of the Galaxy',
        'Guardians of the Galaxy Vol. 2', 'Guardians of the Galaxy
Vol. 3',
        "Guess Who's Coming to Dinner", 'Gully Boy', "Hachi: A Dog's
Tale",
        'Hacksaw Ridge', 'Haider', 'Halloween', 'Hamilton', 'Hamlet',
        'Hannah and Her Sisters', 'Happiness', 'Happy Together',
        'Harakiri', 'Hard Boiled', 'Harold and Maude',
        'Harry Potter and the Deathly Hallows: Part 1',
        'Harry Potter and the Deathly Hallows: Part 2',
        'Harry Potter and the Goblet of Fire',
        'Harry Potter and the Prisoner of Azkaban',
        "Harry Potter and the Sorcerer's Stone", 'Harvey', 'Head-On',
        'Heat', 'Hedwig and the Angry Inch', 'Hell or High Water',
'Her',
        'Hera Pheri', 'Hero', 'Hidden Figures', 'High Noon',
        'High and Low', 'Hindi Medium', 'Hiroshima Mon Amour',
```

```
        'His Girl Friday', 'Home Alone', 'Hot Fuzz', 'Hotel Rwanda',
        'How to Train Your Dragon', 'How to Train Your Dragon 2',
        "Howl's Moving Castle", 'Hunt for the Wilderpeople', 'I Am
Sam',
        'I Remember', 'I Saw the Devil', 'I, Daniel Blake', 'Ikiru',
        'In America', 'In Bruges', 'In Cold Blood', 'In a Lonely
Place',
        'In the Heat of the Night', 'In the Mood for Love',
        'In the Name of the Father', 'Incendies', 'Inception',
        'Indiana Jones and the Last Crusade',
        'Indiana Jones and the Raiders of the Lost Ark',
        'Infernal Affairs', 'Inglourious Basterds', 'Inherit the
Wind',
        'Inside Man', 'Inside Out', 'Interstellar',
        'Invasion of the Body Snatchers', 'Ip Man', 'Iron Man',
        'Isle of Dogs', 'It Happened One Night', "It's a Wonderful
Life",
        "Ivan's Childhood", 'JFK', 'Jab We Met', 'Jai Bhim', 'Jaws',
        'Jean de Florette', 'John Wick: Chapter 4', 'Joint Security
Area',
        'Jojo Rabbit', 'Joker', 'Judgment at Nuremberg', 'Jules and
Jim',
        'Jurassic Park', 'K.G.F: Chapter 1', 'K.G.F: Chapter 2',
        'Kagemusha', 'Kahaani', 'Kaithi', 'Kal Ho Naa Ho', 'Kantara',
        'Key Largo', 'Kick-Ass', "Kiki's Delivery Service",
        'Kill Bill: Vol. 1', 'Kill Bill: Vol. 2',
        'Kind Hearts and Coronets', 'King Kong',
        'Kingsman: The Secret Service', 'Klaus', 'Knives Out',
        "Knockin' on Heaven's Door", 'Kramer vs. Kramer',
        'Kubo and the Two Strings', 'Kung Fu Hustle', 'Kung Fu Panda',
        "L'Avventura", 'L.A. Confidential', 'La Dolce Vita', 'La La
Land',
        'La haine', 'La strada', 'Lagaan: Once Upon a Time in India',
        'Land ofe', 'Laura', 'Lawrence of Arabia', 'Le Samouraï',
        'Let the Right One In', "Let's Go! India", 'Lethal Weapon',
        'Letters from Iwo Jima', 'Life Is Beautiful', 'Life of Brian',
        'Life of Pi', 'Like Father, Like Son', 'Like Stars on Earth',
        'Lilya 4-Ever', 'Lion', 'Little Miss Sunshine', 'Little
Women',
        'Lock, Stock and Two Smoking Barrels', 'Logan',
        'Lost in Translation', 'Love and Death', 'Loving Vincent',
        'Lucky Number Slevin', 'Léon: The Professional', 'M',
        'M.S. Dhoni: The Untold Story', 'Mad Max: Fury Road',
'Magnolia',
        'Malcolm X', 'Man on Fire', 'Manchester by the Sea',
'Manhattan',
        'Marriage Story', 'Marty', 'Mary Poppins', 'Mary and Max',
        'Masaan', 'Match Point', 'Me and Earl and the Dying Girl',
        'Memento', 'Memories of Murder', 'Metropolis', 'Midnight
```

```
Cowboy',
        'Midnight in Paris', 'Mildred Pierce', 'Millennium Actress',
        "Miller's Crossing", 'Million Dollar Baby', 'Minority Report',
        'Miracle in Cell No. 7', 'Miracle on 34th Street', 'Mirror',
        'Misery', 'Mission: Impossible - Fallout', 'Mississippi
Burning',
        'Moana', 'Modern Times', 'Mommy', 'Moneyball', 'Monsters,
Inc.',
        'Monty Python and the Holy Grail', 'Moon', 'Moonrise Kingdom',
        'Mother', 'Mr. Nobody', 'Mr. Smith Goes to Washington',
'Mulan',
        'Mulholland Drive', 'Munna Bhai M.B.B.S.', 'My Cousin Vinny',
        'My Fair Lady', 'My Father and My Son', 'My Left Foot',
        'My Life as a Zucchini', 'My Name Is Khan', 'My Neighbor
Totoro',
        'My Sassy Girl', 'Mystic River', 'NA', 'Naked',
        'Nausicaä of the Valley of the Wind', 'Nebraska',
        'Neon Genesis Evangelion: The End of Evangelion', 'Network',
        'Night of the Living Dead', 'Night on Earth', 'Nightcrawler',
        'Nights of Cabiria', 'Nine Queens', 'Ninja Scroll',
        'No Country for Old Men', "No Man's Land", 'Nobody Knows',
        'North by Northwest', 'Nosferatu', 'Nostalghia', 'Notorious',
        'O Brother, Where Art Thou?', 'OMG: Oh My God!', "Ocean's
Eleven",
        'October Sky', 'Office Space', 'Oldboy', 'On the Waterfront',
        'Once', 'Once Upon a Time in America',
        'Once Upon a Time in Anatolia', 'Once Upon a Time in
Hollywood',
        'Once Upon a Time in the West', 'Once Were Warriors',
        "One Flew Over the Cuckoo's Nest", 'Open Your Eyes',
        'Ordinary People', 'Out of the Past', 'PK', 'Paan Singh
Tomar',
        'Pad Man', 'Paddington 2', "Pan's Labyrinth", 'Paper Moon',
        'Papillon', 'Paprika', 'Parasite', 'Paris, Texas',
        'Pather Panchali', 'Paths of Glory', 'Patton', 'Perfect Blue',
        'Perfect Strangers', 'Persepolis', 'Persona', 'Philadelphia',
        'Philomena', 'Pink', 'Pink Floyd: The Wall',
        'Pirates of the Caribbean: The Curse of the Black Pearl',
        'Planet of the Apes', 'Platoon', 'Ponyo', 'Porco Rosso',
        'Portrait of a Lady on Fire', 'Predator', 'Pride',
        'Pride & Prejudice', 'Primal Fear', 'Princess Mononoke',
        'Prisoners', 'Psycho', 'Pulp Fiction',
        'Puss in Boots: The Last Wish', 'Queen', 'Quo Vadis, Aida?',
'RRR',
        'Raatchasan', 'Raging Bull', 'Rain Man', 'Raise the Red
Lantern',
        'Ran', 'Rang De Basanti', 'Rangasthalam 1985', 'Rashomon',
        'Ratatouille', 'Ray', 'Rear Window', 'Rebecca',
        'Rebel Without a Cause', 'Red River', 'Remember the Titans',
```

```
        'Requiem for a Dream', 'Reservoir Dogs', 'Rififi', 'Rio
Bravo',
        'Road to Perdition', 'RoboCop', 'Rocketry: The Nambi Effect',
        'Rocky', 'Rogue One: A Star Wars Story', 'Roma', 'Roman
Holiday',
        'Rome, Open City', 'Room', 'Rope', "Rosemary's Baby",
        'Run Lola Run', 'Rush', 'Rushmore', 'Sabrina', 'Sanjuro',
        'Sardar Udham', 'Sarfarosh', 'Saving Private Ryan', 'Saw',
        'Scarface', 'Scent of a Woman', "Schindler's List", 'Se7en',
        'Searching', 'Secrets & Lies', 'Sense and Sensibility',
'Serenity',
        'Serpico', 'Seven Samurai', 'Shadow of a Doubt',
        'Shaun of the Dead', 'Sherlock Jr.', 'Shershaah', 'Shine',
        'Sholay', 'Shoplifters', 'Short Cuts', 'Short Term 12',
        'Shutter Island', 'Sicario', 'Silver Linings Playbook', 'Sin
City',
        'Sing Street', "Singin' in the Rain", 'Sita Ramam', 'Skyfall',
        'Sleuth', 'Sling Blade', 'Slumdog Millionaire', 'Snatch',
        'Snow White and the Seven Dwarfs', 'Solaris', 'Some Like It
Hot',
        'Song of the Sea', 'Soorarai Pottru', 'Soul', 'Sound of
Metal',
        'South Park: Bigger, Longer & Uncut', 'Spartacus', 'Special
26',
        'Spider-Man: Across the Spider-Verse',
        'Spider-Man: Into the Spider-Verse', 'Spider-Man: No Way
Home',
        'Spirited Away', 'Spotlight',
        'Spring, Summer, Fall, Winter... and Spring', 'Stagecoach',
        'Stalag 17', 'Stalker', 'Stand by Me', 'Star Trek',
        'Star Trek II: The Wrath of Khan', 'Star Trek Into Darkness',
        'Star Wars: Episode III - Revenge of the Sith',
        'Star Wars: Episode IV - A New Hope',
        'Star Wars: Episode V - The Empire Strikes Back',
        'Star Wars: Episode VI - Return of the Jedi',
        'Star Wars: Episode VII - The Force Awakens', 'Stardust',
        'Straight Outta Compton', 'Strangers on a Train',
        "Sullivan's Travels", 'Sunrise', 'Sunset Blvd.', 'Super 30',
        'Superbad', 'Swades', 'Sweet Smell of Success',
        'Tae Guk Gi: The Brotherhood of War', 'Taken', 'Talk to Her',
        'Talvar', 'Tangerines', 'Tangled', 'Taste of Cherry',
        'Taxi Driver', 'Terminator 2: Judgment Day', 'The 400 Blows',
        'The Adventures of Robin Hood', 'The African Queen',
        'The Apartment', 'The Artist', 'The Asphalt Jungle',
        'The Avengers', 'The Bandit', 'The Banshees of Inisherin',
        'The Batman', 'The Battle of Algiers', 'The Best Offer',
        'The Best Years of Our Lives', 'The Big Heat', 'The Big
Lebowski',
        'The Big Short', 'The Big Sleep', 'The Birds', 'The Blind
```

```
Side',
        'The Blues Brothers', 'The Boondock Saints', 'The Bourne
Identity',
        'The Bourne Supremacy', 'The Bourne Ultimatum',
        'The Boy in the Striped Pajamas', 'The Breadwinner',
        'The Breakfast Club', 'The Breath', 'The Bride of
Frankenstein',
        'The Bridge on the River Kwai', 'The Bridges of Madison
County',
        'The Broken Circle Breakdown', 'The Butterfly Effect',
        'The Cabinet of Dr. Caligari', 'The Caine Mutiny',
        'The Celebration', 'The Chaser', 'The Chorus', 'The Circus',
        'The Color Purple', 'The Conformist', 'The Conversation',
        'The Count of Monte Cristo', 'The Curious Case of Benjamin
Button',
        'The Dark Knight', 'The Dark Knight Rises',
        'The Day of the Jackal', 'The Day the Earth Stood Still',
        'The Deer Hunter', 'The Departed', 'The Dirty Dozen',
        'The Discreet Charm of the Bourgeoisie',
        'The Diving Bell and the Butterfly',
        'The Double Life of Véronique', 'The Edge of Heaven',
        'The Elephant Man', 'The Exorcist', 'The Experiment',
        'The Exterminating Angel', 'The Fall', 'The Father',
        'The Fault in Our Stars', 'The Fifth Element', 'The Fighter',
        'The Fly', 'The French Connection', 'The Fugitive', 'The
Game',
        'The General', 'The Gentlemen', 'The Girl Who Leapt Through
Time',
        'The Girl with the Dragon Tattoo', 'The Godfather',
        'The Godfather Part II', 'The Gold Rush',
        'The Good, the Bad and the Ugly', 'The Goonies', 'The
Graduate',
        'The Grand Budapest Hotel', 'The Grand Illusion',
        'The Grapes of Wrath', 'The Great Beauty', 'The Great
Dictator',
        'The Great Escape', 'The Green Mile', 'The Handmaiden',
        'The Hangover', 'The Hateful Eight', 'The Help',
        'The Hidden Fortress', 'The Hobbit: An Unexpected Journey',
        'The Hobbit: The Desolation of Smaug', 'The Holy Mountain',
        'The Hunt', 'The Hustler', 'The Imitation Game', 'The
Incredibles',
        'The Innocents', 'The Insider', 'The Intouchables',
        'The Invisible Guest', 'The Invisible Man', 'The Irishman',
        'The Iron Giant', 'The Jungle Book', 'The Kid', 'The Killer',
        'The Killing', 'The Killing Fields', 'The King of Comedy',
        "The King's Speech", 'The Lady Vanishes', 'The Last Emperor',
        'The Last King of Scotland', 'The Last Picture Show',
        'The Last Samurai', 'The Last of the Mohicans',
        'The Legend of 1900', 'The Lego Movie', 'The Leopard',
```

```
        'The Lion King', 'The Lion in Winter', 'The Little Prince',
        'The Lives of Others', 'The Longest Day',
        'The Lord of the Rings: The Fellowship of the Ring',
        'The Lord of the Rings: The Return of the King',
        'The Lord of the Rings: The Two Towers', 'The Lost Weekend',
        'The Lunchbox', 'The Machinist', 'The Magdalene Sisters',
        'The Magnificent Seven', 'The Maltese Falcon',
        'The Man Who Shot Liberty Valance', 'The Man Who Would Be
King',
        'The Man from Earth', 'The Man from Nowhere',
        'The Manchurian Candidate', 'The Martian', 'The Matrix',
        'The Message', 'The Mitchells vs the Machines',
        'The Motorcycle Diaries', 'The Muppet Christmas Carol',
        'The Name of the Rose', 'The Night of the Hunter',
        'The Nightmare Before Christmas', 'The Notebook', 'The
Others',
        'The Outlaw Josey Wales', 'The Passion of Joan of Arc', 'The
Past',
        'The Perks of Being a Wallflower', 'The Philadelphia Story',
        'The Pianist', 'The Postman', 'The Prestige', 'The Princess
Bride',
        'The Purple Rose of Cairo', 'The Pursuit of Happyness',
        'The Quiet Man', 'The Raid 2', 'The Raid: Redemption',
        'The Red Circle', 'The Red Shoes', 'The Remains of the Day',
        'The Return', 'The Revenant', 'The Right Stuff',
        'The Road Warrior', 'The Royal Tenenbaums',
        'The Rules of the Game', 'The Sacrifice', 'The Salesman',
        'The Sandlot', 'The Sea Inside', 'The Searchers',
        'The Secret in Their Eyes', 'The Seventh Seal',
        'The Shawshank Redemption', 'The Shining',
        'The Shop Around the Corner', 'The Silence of the Lambs',
        'The Sixth Sense', 'The Skin I Live In', 'The Social Network',
        'The Sound of Music', 'The Sting', 'The Straight Story',
        'The Taking of Pelham One Two Three',
        'The Tale of The Princess Kaguya', 'The Ten Commandments',
        'The Terminator', 'The Theory of Everything', 'The Thin Man',
        'The Thin Red Line', 'The Thing', 'The Third Man',
        'The Treasure of the Sierra Madre', 'The Trial of the Chicago
7',
        'The Triplets of Belleville', 'The Truman Show',
        'The Umbrellas of Cherbourg', 'The Untouchables',
        'The Usual Suspects', 'The Vanishing', 'The Verdict',
        'The Virgin Spring', 'The Wages of Fear', 'The Way He Looks',
        'The Whale', 'The White Ribbon', 'The Wild Bunch',
        'The Wild Pear Tree', 'The Wind Rises', 'The Wizard of Oz',
        'The Wolf of Wall Street', "The World's Fastest Indian",
        'The Worst Person in the World', 'The Wrestler',
        'There Will Be Blood', 'Thirteen Lives', 'This Is England',
        'This Is Spinal Tap', 'Thor: Ragnarok',
```

```
        'Three Billboards Outside Ebbing, Missouri', 'Three Colors:
Blue',
        'Three Colors: Red', 'Throne of Blood', 'Through a Glass
Darkly',
        'Time of the Gypsies', 'Titanic', 'To Be or Not to Be',
        'To Have and Have Not', 'To Kill a Mockingbird', 'Togo',
        'Tokyo Godfathers', 'Tokyo Story', 'Tombstone',
        'Top Gun: Maverick', 'Touch of Evil', 'Toy Story', 'Toy Story
2',
        'Toy Story 3', 'Toy Story 4', 'Training Day', 'Trainspotting',
        'True Grit', 'True Romance', 'Tumbbad', 'Udaan', 'Umberto D.',
        'Underground', 'Unforgiven', 'United 93', 'Up',
        'Uri: The Surgical Strike', 'V for Vendetta', 'Veer Zaara',
        'Vertigo', 'Vicky Donor', 'Vikram', 'Vikram Vedha', 'Vivre Sa
Vie',
        'Vizontele', 'WALL·E', 'Wait Until Dark', 'Waking Life',
        'Walk the Line', 'Warrior', 'Watchmen',
        'What Ever Happened to Baby Jane?', 'What We Do in the
Shadows',
        "What's Eating Gilbert Grape", 'When Harry Met Sally...',
        'When Marnie Was There', 'Whiplash', 'Whisper of the Heart',
        'White Heat', 'Who Framed Roger Rabbit',
        "Who's Afraid of Virginia Woolf?", 'Wild Strawberries',
        'Wild Tales', 'Willy Wonka & the Chocolate Factory', 'Wind
River',
        'Wings of Desire', 'Winter Light', 'Winter Sleep',
        'Witness for the Prosecution', 'Wolf Children', 'Wolfwalkers',
        'Wonder', 'X-Men: Days of Future Past', 'X-Men: First Class',
        'Yi Yi: A One and a Two...', 'Yojimbo',
        "You Can't Take It with You", 'Young Frankenstein', 'Your
Name.',
        'Z', "Zack Snyder's Justice League", 'Zelig',
        'Zindagi Na Milegi Dobara', 'Zodiac', 'Zootopia', 'Zulu'],
       dtype=object),
 array(['Aamir Khan', 'Aaron Sorkin', 'Abbas Kiarostami',
        'Abdellatif Kechiche', 'Abhishek Pathak', 'Adam Elliot',
        'Adam McKay', 'Aditya Chopra', 'Aditya Dhar', 'Akira
Kurosawa',
        'Alain Resnais', 'Alan Parker', 'Alejandro Amenábar',
        'Alejandro G. Iñárritu', 'Alejandro Jodorowsky', 'Alex
Garland',
        'Alex Proyas', 'Alexander Mackendrick', 'Alexander Payne',
        'Alfonso Cuarón', 'Alfonso Gomez-Rejon', 'Alfred Hitchcock',
        'Amit Ravindernath Sharma', 'Anders Thomas Jensen',
        'Andrei Tarkovsky', 'Andrew Adamson', 'Andrew Davis', 'Andrew
Lau',
        'Andrew Niccol', 'Andrew Stanton', 'Andrey Zvyagintsev',
        'Aneesh Chaganty', 'Ang Lee', 'Aniruddha Roy Chowdhury',
        'Anthony Harvey', 'Anthony Russo', 'Antoine Fuqua',
```

```
        'Anton Corbijn', 'Anubhav Sinha', 'Anurag Basu', 'Anurag
Kashyap',
        'Arthur Penn', 'Asghar Farhadi', 'Ashutosh Gowariker',
        'Barry Levinson', 'Ben Affleck', 'Bennett Miller',
        'Bernardo Bertolucci', 'Billy Bob Thornton', 'Billy Wilder',
        'Boaz Yakin', 'Bob Clark', 'Bob Fosse', 'Bob Persichetti',
        'Bong Joon Ho', 'Brad Anderson', 'Brad Bird', 'Bradley
Cooper',
        'Brian De Palma', 'Brian Henson', 'Bryan Singer', 'Buster
Keaton',
        'Byron Howard', 'C. Prem Kumar', 'Cameron Crowe', 'Can Ulkay',
        'Carl Theodor Dreyer', 'Carol Reed', 'Cary Joji Fukunaga',
        'Cecil B. DeMille', 'Chad Stahelski', 'Charles Chaplin',
        'Charles Laughton', 'Charlotte Wells', 'Chris Columbus',
        'Christophe Barratier', 'Christopher McQuarrie',
        'Christopher Nolan', 'Claude Barras', 'Claude Berri',
        'Clint Eastwood', 'Clyde Bruckman', 'Costa-Gavras',
        'Cristian Mungiu', 'Curtis Hanson', 'Cy Endfield',
        'Céline Sciamma', 'DK Welchman', 'Damien Chazelle',
        'Damián Szifron', 'Dan Gilroy', 'Daniel Kwan', 'Daniel
Monzón',
        'Daniel Ribeiro', 'Danis Tanovic', 'Danny Boyle', 'Darius
Marder',
        'Darren Aronofsky', 'David Ayer', 'David Cronenberg',
        'David Fincher', 'David Lean', 'David Leitch', 'David Lynch',
        'David Mackenzie', 'David Mickey Evans', 'David O. Russell',
        'David Yates', 'Dean DeBlois', 'Delbert Mann', 'Denis
Villeneuve',
        'Destin Daniel Cretton', 'Don Hall', 'Don Siegel', 'Doug
Liman',
        'Duncan Jones', 'Edgar Wright', 'Edward Berger', 'Edward
Dmytryk',
        'Edward Yang', 'Edward Zwick', 'Elem Klimov', 'Elia Kazan',
        'Emir Kusturica', 'Eric Bress', 'Eric Radomski', 'Ericson
Core',
        'Ernst Lubitsch', 'Ethan Coen', 'F. Gary Gray', 'F.W. Murnau',
        'Fabián Bielinsky', 'Farhan Akhtar', 'Fatih Akin',
        'Federico Fellini', 'Felix van Groeningen', 'Fernando
Meirelles',
        'Florian Henckel von Donnersmarck', 'Florian Zeller',
        'Francis Ford Coppola', 'Frank Capra', 'Frank Darabont',
        'Frank Miller', 'Franklin J. Schaffner', 'François Truffaut',
        'Fred Zinnemann', 'Fritz Lang', 'Gabriele Muccino',
        'Gareth Edwards', 'Gareth Evans', 'Garth Davis', 'Gary
Trousdale',
        'Gauri Shinde', "Gavin O'Connor", 'Gayatri', 'George A.
Romero',
        'George Cukor', 'George Lucas', 'George Miller',
        'George P. Cosmatos', 'George Roy Hill', 'George Seaton',
```

```
        'George Sluizer', 'Gillo Pontecorvo', 'Giuseppe Tornatore',
        'Gore Verbinski', 'Greg Mottola', 'Gregory Hoblit', 'Greta
Gerwig',
        'Guillermo del Toro', 'Gus Van Sant', 'Guy Hamilton',
        'Guy Ritchie', 'Hal Ashby', 'Hannes Holm', 'Hanu Raghavapudi',
        'Harold Ramis', 'Haruo Sotozaki', 'Hayao Miyazaki',
        'Henri-Georges Clouzot', 'Henry Koster', 'Henry Selick',
        'Hideaki Anno', 'Hiromasa Yonebayashi', 'Howard Hawks',
        'Hrishikesh Mukherjee', 'Hun Jang', 'Imtiaz Ali', 'Ingmar
Bergman',
        'Irvin Kershner', 'Isao Takahata', 'Ivan Reitman',
        'J. Lee Thompson', 'J.J. Abrams', 'Jack Clayton',
        'Jaco Van Dormael', 'Jacques Audiard', 'Jacques Demy',
        'Jacques Tourneur', 'Jae-young Kwak', 'James Algar',
        'James Cameron', 'James Foley', 'James Gunn', 'James Ivory',
        'James L. Brooks', 'James Mangold', 'James Marsh',
        'James McTeigue', 'James Wan', 'James Whale', 'Jasmila
Zbanic',
        'Je-kyu Kang', 'Jean Cocteau', 'Jean Renoir',
        'Jean-Jacques Annaud', 'Jean-Luc Godard', 'Jean-Marc Vallée',
        'Jean-Pierre Jeunet', 'Jean-Pierre Melville', 'Jee-woon Kim',
        'Jeethu Joseph', 'Jemaine Clement', 'Jeong-beom Lee',
        'Jessie Nelson', 'Jim Abrahams', 'Jim Jarmusch', 'Jim
Sheridan',
        'Joachim Trier', 'Joaquim Dos Santos', 'Joe Wright', 'Joel
Coen',
        'Joel Crawford', 'John Boorman', 'John Cameron Mitchell',
        'John Carney', 'John Carpenter', 'John Cassavetes', 'John
Ford',
        'John Frankenheimer', 'John G. Avildsen', 'John Hughes',
        'John Huston', 'John Landis', 'John Lasseter', 'John Lee
Hancock',
        'John Mathew Matthan', 'John McTiernan', 'John Schlesinger',
        'John Singleton', 'John Sturges', 'John Woo', 'Jon Avnet',
        'Jon Favreau', 'Jon Watts', 'Jonathan Dayton', 'Jonathan
Demme',
        'Jonathan Levine', 'Jonathan Lynn', 'Jordan Peele',
        'Joseph Kosinski', 'Joseph L. Mankiewicz', 'Joseph Sargent',
        'Josh Boone', 'Josh Cooley', 'Joss Whedon', 'José Padilha',
        'Juan José Campanella', 'Jules Dassin', 'Julian Schnabel',
        'Kabir Khan', 'Kaige Chen', 'Kar-Wai Wong', 'Karan Johar',
        'Katsuhiro Ôtomo', 'Ken Annakin', 'Ken Loach', 'Kenneth
Branagh',
        'Kenneth Lonergan', 'Kevin Costner', 'Kevin Macdonald',
        'Kevin Reynolds', 'Kevin Smith', 'Kim Ki-duk', 'Kiranraj K',
        'Kore-eda Hirokazu', 'Krzysztof Kieslowski', 'Lana Wachowski',
        'Lars von Trier', 'Lasse Hallström', 'Lee Tamahori', 'Lee
Unkrich',
        'Lenny Abrahamson', 'Leo McCarey', 'Levent Semerci',
        'Lewis Milestone', 'Lokesh Kanagaraj', 'Louis Malle', 'Luc
```

```
Besson',
        'Luca Guadagnino', 'Luchino Visconti', 'Luis Buñuel',
        'Lukas Moodysson', 'M. Night Shyamalan', 'Madhavan',
        'Majid Majidi', 'Makoto Shinkai', 'Mamoru Hosoda', 'Mamoru
Oshii',
        'Marc Forster', 'Marc Webb', 'Mark Herman', 'Mark Osborne',
        'Martin Brest', 'Martin Campbell', 'Martin McDonagh',
        'Martin Scorsese', 'Martin Zandvliet', 'Mary Harron',
        'Masaki Kobayashi', 'Mathieu Kassovitz', 'Mathur Goswami',
        'Matt Reeves', 'Matt Ross', 'Matthew Vaughn', 'Matthew
Warchus',
        'Meghna Gulzar', 'Mehmet Ada Öztekin', 'Mel Brooks', 'Mel
Gibson',
        'Mel Stuart', 'Merian C. Cooper', 'Michael Cimino',
        'Michael Curtiz', 'Michael Haneke', 'Michael Mann',
        'Michael Powell', 'Michael Radford', 'Michael Rianda',
        'Michel Gondry', 'Michel Hazanavicius', 'Michelangelo
Antonioni',
        'Mikael Håfström', 'Mike Judge', 'Mike Leigh', 'Mike Newell',
        'Mike Nichols', 'Milos Forman', 'Morten Tyldum', 'Moustapha
Akkad',
        'NA', 'Na Hong-jin', 'Nadine Labaki', 'Naoko Yamada',
        'Nathan Greno', 'Neeraj Ghaywan', 'Neeraj Pandey',
        'Neill Blomkamp', 'Nicholas Meyer', 'Nicholas Ray',
        'Nick Cassavetes', 'Nicolas Winding Refn', 'Niels Arden
Oplev',
        'Nikkhil Advani', 'Nishikant Kamat', 'Nitesh Tiwari',
        'Nora Twomey', 'Norman Jewison', 'Nuri Bilge Ceylan',
        'Oliver Hirschbiegel', 'Oliver Stone', 'Olivier Nakache',
        'Oriol Paulo', 'Orson Welles', 'Otto Preminger', 'Paolo
Genovese',
        'Paolo Sorrentino', 'Park Chan-wook', 'Paul Greengrass',
        'Paul Haggis', 'Paul King', 'Paul McGuigan',
        'Paul Thomas Anderson', 'Paul Verhoeven', 'Pedro Almodóvar',
        'Penny Marshall', 'Pete Docter', 'Peter Bogdanovich',
        'Peter Farrelly', 'Peter Jackson', 'Peter Mullan', 'Peter
Weir',
        'Phil Lord', 'Philip Kaufman', 'Pierre Coffin', 'Pierre
Morel',
        'Prashanth Neel', 'Preston Sturges', 'Priyadarshan',
        'Quentin Tarantino', 'R. Balki', 'Rahi Anil Barve', 'Raja
Menon',
        'Rajkumar Hirani', 'Rajkumar Santoshi', 'Rakeysh Omprakash
Mehra',
        'Ram Kumar', 'Ramesh Sippy', 'Raoul Walsh', 'René Laloux',
        'Rian Johnson', 'Rich Moore', 'Richard Attenborough',
        'Richard Brooks', 'Richard Curtis', 'Richard Donner',
        'Richard Kelly', 'Richard Linklater', 'Richard Marquand',
        'Richard Schenkman', 'Ridley Scott', 'Rishab Shetty',
```

```
        'Ritesh Batra', 'Rob Reiner', 'Robert Aldrich', 'Robert
Altman',
        'Robert Benton', 'Robert Clouse', 'Robert De Niro', 'Robert
Hamer',
        'Robert Mulligan', 'Robert Redford', 'Robert Rossen',
        'Robert Stevenson', 'Robert Wiene', 'Robert Wise',
        'Robert Zemeckis', 'Roberto Benigni', 'Roberto Rossellini',
        'Roger Allers', 'Roger Donaldson', 'Roland Joffé',
        'Roman Polanski', 'Ron Clements', 'Ron Howard', 'S.S.
Rajamouli',
        'Saket Chaudhary', 'Sam Mendes', 'Sam Peckinpah', 'Sam Raimi',
        'Sam Wood', 'Sanjay Leela Bhansali', 'Satoshi Kon', 'Satyajit
Ray',
        'Scott Hicks', 'Sean Penn', 'Sergei Eisenstein', 'Sergio
Leone',
        'Sergio Pablos', 'Shane Meadows', 'Shimit Amin',
        "Shin'ichirô Watanabe", 'Shoojit Sircar', 'Sian Heder',
        'Sidney Lumet', 'Sofia Coppola', 'Spike Jonze', 'Spike Lee',
        'Sriram Raghavan', 'Stanley Donen', 'Stanley Kramer',
        'Stanley Kubrick', 'Stephen Chbosky', 'Stephen Chow',
        'Stephen Daldry', 'Stephen Frears', 'Steve McQueen',
        'Steven Soderbergh', 'Steven Spielberg', 'Stuart Rosenberg',
        'Sudha Kongara', 'Sujoy Ghosh', 'Sukumar', 'Susanne Bier',
        'Sylvain Chomet', 'T.J. Gnanavel', 'Taika Waititi',
        'Takeshi Kitano', 'Tarsem Singh', 'Tate Taylor', 'Taylor
Hackford',
        'Taylor Sheridan', 'Ted Kotcheff', 'Terence Young',
        'Terrence Malick', 'Terry George', 'Terry Gilliam', 'Terry
Jones',
        'Tetsuya Nakashima', 'Theodore Melfi', 'Thomas Jahn',
        'Thomas Kail', 'Thomas Vinterberg', 'Tigmanshu Dhulia',
        'Tim Burton', 'Tim Miller', 'Tod Browning', 'Todd Haynes',
        'Todd Phillips', 'Todd Solondz', 'Tom Hooper', 'Tom McCarthy',
        'Tom Tykwer', 'Tomas Alfredson', 'Tomm Moore', 'Tony
Bancroft',
        'Tony Kaye', 'Tony Scott', 'Travis Knight', 'Trey Parker',
        'Troy Duffy', 'Umesh Shukla', 'Vetrimaaran', 'Victor Fleming',
        'Vikas Bahl', 'Vikramaditya Motwane', 'Vincent Paronnaud',
        'Vishal Bhardwaj', 'Vishnuvardhan', 'Vittorio De Sica',
        'W.S. Van Dyke', 'Walter Salles', 'Werner Herzog', 'Wes
Anderson',
        'William Cottrell', 'William Friedkin', 'William Wyler',
        'Wilson Yip', 'Wim Wenders', 'Wolfgang Becker',
        'Wolfgang Petersen', 'Wolfgang Reitherman', 'Woody Allen',
        'Xavier Dolan', 'Yash Chopra', 'Yasujirô Ozu', 'Yavuz Turgul',
        'Yilmaz Erdogan', 'Yimou Zhang', 'Yoshiaki Kawajiri',
        'Yoshifumi Kondô', 'Yôjirô Takita', 'Zack Snyder',
        'Zaza Urushadze', 'Zoya Akhtar', 'Çagan Irmak', 'Ömer Faruk
Sorak',
```

```
        'Ömer Vargi'], dtype=object),
 array(['Action, Adventure', 'Action, Adventure, Biography',
        'Action, Adventure, Comedy', 'Action, Adventure, Crime',
        'Action, Adventure, Drama', 'Action, Adventure, Fantasy',
        'Action, Adventure, Horror', 'Action, Adventure, Mystery',
        'Action, Adventure, Romance', 'Action, Adventure, Sci-Fi',
        'Action, Adventure, Thriller', 'Action, Adventure, War',
        'Action, Biography, Crime', 'Action, Biography, Drama',
        'Action, Comedy', 'Action, Comedy, Crime',
        'Action, Comedy, Fantasy', 'Action, Comedy, Mystery',
        'Action, Comedy, Romance', 'Action, Crime, Comedy',
        'Action, Crime, Drama', 'Action, Crime, Mystery',
        'Action, Crime, Sci-Fi', 'Action, Crime, Thriller',
        'Action, Drama', 'Action, Drama, History',
        'Action, Drama, Mystery', 'Action, Drama, Sci-Fi',
        'Action, Drama, Sport', 'Action, Drama, Thriller',
        'Action, Drama, War', 'Action, Drama, Western',
        'Action, Mystery, Thriller', 'Action, Sci-Fi',
        'Action, Sci-Fi, Thriller', 'Action, Thriller',
        'Adventure, Biography, Drama', 'Adventure, Comedy, Crime',
        'Adventure, Comedy, Drama', 'Adventure, Comedy, Family',
        'Adventure, Comedy, Fantasy', 'Adventure, Comedy, Film-Noir',
        'Adventure, Comedy, Sci-Fi', 'Adventure, Drama',
        'Adventure, Drama, Family', 'Adventure, Drama, Fantasy',
        'Adventure, Drama, History', 'Adventure, Drama, Romance',
        'Adventure, Drama, Sci-Fi', 'Adventure, Drama, Thriller',
        'Adventure, Drama, War', 'Adventure, Drama, Western',
        'Adventure, Family, Fantasy', 'Adventure, Family, Sci-Fi',
        'Adventure, Fantasy', 'Adventure, Horror, Sci-Fi',
        'Adventure, Mystery, Thriller', 'Adventure, Sci-Fi',
        'Adventure, War', 'Adventure, Western',
        'Animation, Action, Adventure', 'Animation, Action, Crime',
        'Animation, Action, Drama', 'Animation, Adventure, Comedy',
        'Animation, Adventure, Drama', 'Animation, Adventure, Family',
        'Animation, Adventure, Sci-Fi', 'Animation, Biography, Drama',
        'Animation, Comedy, Drama', 'Animation, Comedy, Family',
        'Animation, Comedy, Fantasy', 'Animation, Crime, Drama',
        'Animation, Drama', 'Animation, Drama, Family',
        'Animation, Drama, Fantasy', 'Animation, Drama, Mystery',
        'Animation, Drama, War', 'Animation, Family, Fantasy',
        'Animation, Sci-Fi', 'Biography, Comedy, Crime',
        'Biography, Comedy, Drama', 'Biography, Crime, Drama',
        'Biography, Drama', 'Biography, Drama, Family',
        'Biography, Drama, History', 'Biography, Drama, Music',
        'Biography, Drama, Romance', 'Biography, Drama, Sport',
        'Biography, Drama, Thriller', 'Biography, Drama, War',
'Comedy',
        'Comedy, Crime', 'Comedy, Crime, Drama', 'Comedy, Crime,
Mystery',
```

```
        'Comedy, Crime, Romance', 'Comedy, Crime, Thriller',
        'Comedy, Drama', 'Comedy, Drama, Family', 'Comedy, Drama,
Fantasy',
        'Comedy, Drama, Music', 'Comedy, Drama, Musical',
        'Comedy, Drama, Romance', 'Comedy, Drama, Sci-Fi',
        'Comedy, Drama, Thriller', 'Comedy, Drama, War', 'Comedy,
Family',
        'Comedy, Family, Fantasy', 'Comedy, Family, Romance',
        'Comedy, Fantasy, Romance', 'Comedy, Horror', 'Comedy, Music',
        'Comedy, Music, Musical', 'Comedy, Music, Romance',
        'Comedy, Musical', 'Comedy, Musical, Romance',
        'Comedy, Mystery, Romance', 'Comedy, Romance',
        'Comedy, Romance, War', 'Comedy, War', 'Comedy, Western',
        'Crime, Drama', 'Crime, Drama, Fantasy', 'Crime, Drama, Film-
Noir',
        'Crime, Drama, Horror', 'Crime, Drama, Musical',
        'Crime, Drama, Mystery', 'Crime, Drama, Romance',
        'Crime, Drama, Thriller', 'Crime, Film-Noir, Mystery',
        'Crime, Film-Noir, Thriller', 'Crime, Mystery, Thriller',
        'Crime, Sci-Fi', 'Crime, Thriller', 'Drama', 'Drama, Family',
        'Drama, Family, Fantasy', 'Drama, Family, Musical',
        'Drama, Family, Sport', 'Drama, Fantasy', 'Drama, Fantasy,
Horror',
        'Drama, Fantasy, Music', 'Drama, Fantasy, Mystery',
        'Drama, Fantasy, Romance', 'Drama, Fantasy, War',
        'Drama, Film-Noir', 'Drama, Film-Noir, Mystery',
        'Drama, Film-Noir, Romance', 'Drama, History',
        'Drama, History, Thriller', 'Drama, History, War', 'Drama,
Horror',
        'Drama, Horror, Mystery', 'Drama, Horror, Sci-Fi',
        'Drama, Horror, Thriller', 'Drama, Music', 'Drama, Music,
Musical',
        'Drama, Music, Mystery', 'Drama, Music, Romance', 'Drama,
Musical',
        'Drama, Musical, Romance', 'Drama, Musical, Sport',
        'Drama, Mystery', 'Drama, Mystery, Romance',
        'Drama, Mystery, Sci-Fi', 'Drama, Mystery, Thriller',
        'Drama, Mystery, War', 'Drama, Romance', 'Drama, Romance, Sci-
Fi',
        'Drama, Romance, Thriller', 'Drama, Romance, War', 'Drama,
Sci-Fi',
        'Drama, Sci-Fi, Thriller', 'Drama, Sport',
        'Drama, Sport, Thriller', 'Drama, Thriller',
        'Drama, Thriller, War', 'Drama, Thriller, Western', 'Drama,
War',
        'Drama, Western', 'Family, Fantasy, Musical', 'Fantasy',
        'Fantasy, Horror', 'Fantasy, Mystery, Sci-Fi',
        'Film-Noir, Mystery, Thriller', 'Film-Noir, Thriller',
'Horror',
```

```
        'Horror, Mystery, Sci-Fi', 'Horror, Mystery, Thriller',
        'Horror, Sci-Fi', 'Horror, Thriller', 'Mystery, Romance,
Thriller',
        'Mystery, Sci-Fi, Thriller', 'Mystery, Thriller', 'NA',
'Thriller',
        'Western'], dtype=object)]

# Обратное преобразование
oe.inverse_transform(cat_enc_oe)

array([['The Shawshank Redemption', 'Frank Darabont', 'Drama'],
       ['The Godfather', 'Francis Ford Coppola', 'Crime, Drama'],
       ['The Dark Knight', 'Christopher Nolan', 'Action, Crime,
Drama'],
       ...,
       ['Philomena', 'Stephen Frears', 'Biography, Comedy, Drama'],
       ['The Invisible Man', 'James Whale', 'Horror, Sci-Fi'],
       ['Cell 211', 'Daniel Monzón', 'NA']], dtype=object)
```

Проблемы использования LabelEncoder и OrdinalEncoder

Необходимо отметить, что LabelEncoder и OrdinalEncoder могут использоваться только для категориальных признаков в номинальных шкалах (для которых отсутствует порядок), например города, страны, названия рек и т.д.

Это связано с тем, что задать какой-либо порядок при кодировании с помощью LabelEncoder и OrdinalEncoder невозможно, они сортируют категории в лексикографическом порядке.

При этом кодирование целыми числами создает фиктивное отношение порядка (1 < 2 < 3 < ...) которого не было в исходных номинальных шкалах. Данное отношение порядка может негативно повлиять на построение модели машинного обучения.

Кодирование шкал порядка

Библиотека scikit-learn не предоставляет готового решения для кодирования шкал порядка, но можно воспользоваться функцией map для отдельных объектов Series.

```
# пример шкалы порядка 'small' < 'medium' < 'large'
sizes = ['small', 'medium', 'large', 'small', 'medium', 'large',
'small', 'medium', 'large']

pd_sizes = pd.DataFrame(data={'sizes':sizes})
pd_sizes

    sizes
0   small
1  medium
2   large
3   small
```

```
4  medium
5   large
6   small
7  medium
8   large
```

```
pd_sizes['sizes_codes'] = pd_sizes['sizes'].map({'small':1,
'medium':2, 'large':3})
pd_sizes
```

```
    sizes  sizes_codes
0   small            1
1  medium            2
2   large            3
3   small            1
4  medium            2
5   large            3
6   small            1
7  medium            2
8   large            3
```

```
pd_sizes['sizes_decoded'] = pd_sizes['sizes_codes'].map({1:'small',
2:'medium', 3:'large'})
pd_sizes
```

```
    sizes  sizes_codes sizes_decoded
0   small            1         small
1  medium            2        medium
2   large            3         large
3   small            1         small
4  medium            2        medium
5   large            3         large
6   small            1         small
7  medium            2        medium
8   large            3         large
```

Кодирование категорий наборами бинарных значений - one-hot encoding

В этом случае каждое уникальное значение признака становится новым отдельным признаком.

```
from sklearn.preprocessing import OneHotEncoder

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

cat_enc.shape
```

```
(1000, 1)
```

```
cat_enc_ohe.shape
```

```
(1000, 120)
```

```
cat_enc_ohe
```

```
<1000x120 sparse matrix of type '<class 'numpy.float64'>'
     with 1000 stored elements in Compressed Sparse Row format>
```

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

```
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.]])
```

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

```
     c1
0  1994
1  1972
2  2008
3  1993
4  1957
5  2003
6  1974
7  2023
8  1994
9  2010
```

Pandas get_dummies - быстрый вариант one-hot кодирования

```
pd.get_dummies(cat_enc).head()
```

```
   c1_1920  c1_1921  c1_1922  c1_1925  c1_1926  c1_1927  c1_1928
c1_1931  \
0    False    False    False    False    False    False    False
False
1    False    False    False    False    False    False    False
False
2    False    False    False    False    False    False    False
False
3    False    False    False    False    False    False    False
False
```

```
4    False    False    False    False    False    False    False
False

   c1_1932  c1_1933  ...  c1_I 2015  c1_I 2016  c1_I 2017  c1_I
2019  \
0    False    False  ...      False      False      False      False

1    False    False  ...      False      False      False      False

2    False    False  ...      False      False      False      False

3    False    False  ...      False      False      False      False

4    False    False  ...      False      False      False      False


   c1_I 2020  c1_II 2016  c1_II 2018  c1_II 2022  c1_III 2016  c1_III
2018
0      False       False       False       False        False
False
1      False       False       False       False        False
False
2      False       False       False       False        False
False
3      False       False       False       False        False
False
4      False       False       False       False        False
False

[5 rows x 120 columns]

pd.get_dummies(cat_temp_data, dummy_na=True).head()

   release_year_1920  release_year_1921  release_year_1922
release_year_1925  \
0              False              False              False
False
1              False              False              False
False
2              False              False              False
False
3              False              False              False
False
4              False              False              False
False

   release_year_1926  release_year_1927  release_year_1928
release_year_1931  \
0              False              False              False
False
1              False              False              False
```

```
False
2               False               False               False
False
3               False               False               False
False
4               False               False               False
False

   release_year_1932  release_year_1933  ...  release_year_I 2016  \
0              False              False  ...                False
1              False              False  ...                False
2              False              False  ...                False
3              False              False  ...                False
4              False              False  ...                False

   release_year_I 2017  release_year_I 2019  release_year_I 2020  \
0                False                False                False
1                False                False                False
2                False                False                False
3                False                False                False
4                False                False                False

   release_year_II 2016  release_year_II 2018  release_year_II 2022  \
0                 False                 False                 False
1                 False                 False                 False
2                 False                 False                 False
3                 False                 False                 False
4                 False                 False                 False

   release_year_III 2016  release_year_III 2018  release_year_nan
0                  False                  False             False
1                  False                  False             False
2                  False                  False             False
3                  False                  False             False
4                  False                  False             False

[5 rows x 121 columns]
```

Масштабирование данных

Термины "масштабирование" и "нормализация" часто используются как синонимы, но это неверно. Масштабирование предполагает изменение диапазона измерения величины, а нормализация - изменение распределения этой величины. В этом разделе рассматривается только масштабирование.

Если признаки лежат в различных диапазонах, то необходимо их нормализовать. Как правило, применяют два подхода:

- MinMax масштабирование:

$$x_{новый} = \frac{x_{старый} - min(X)}{max(X) - min(X)}$$

В этом случае значения лежат в диапазоне от 0 до 1.

- Масштабирование данных на основе Z-оценки:

$$x_{новый} = \frac{x_{старый} - AVG(X)}{\sigma(X)}$$

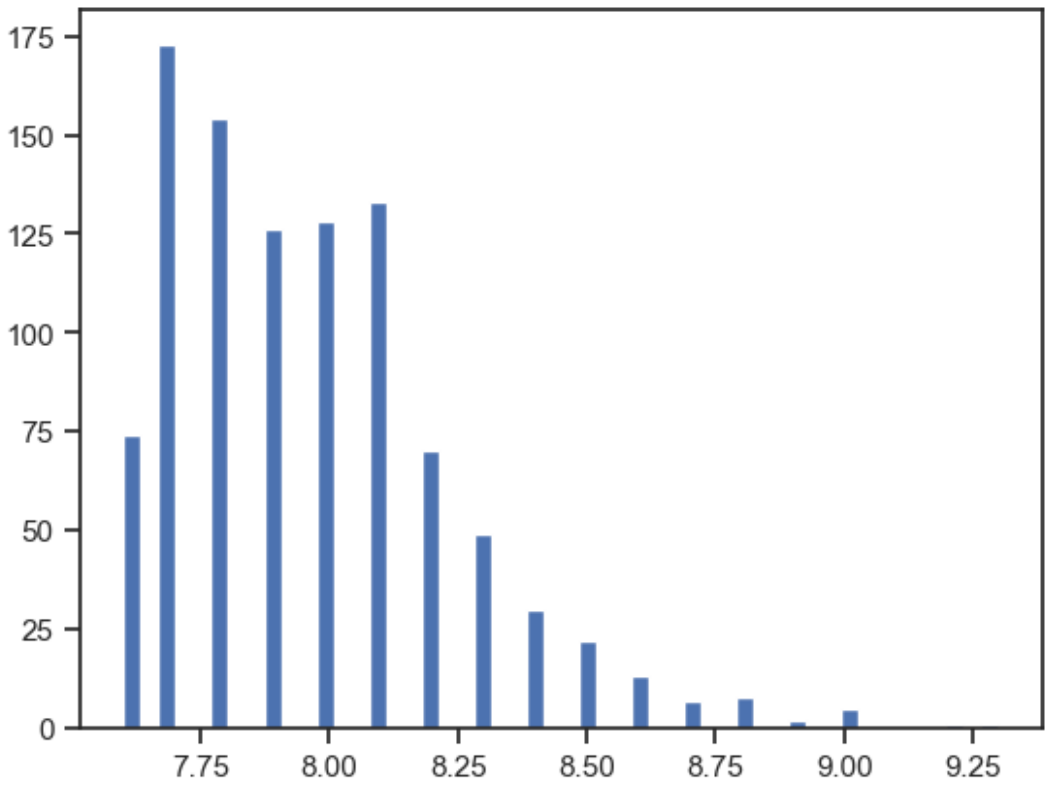
В этом случае большинство значений попадает в диапазон от -3 до 3.

где X - матрица объект-признак, $AVG(X)$ - среднее значение, σ - среднеквадратичное отклонение.

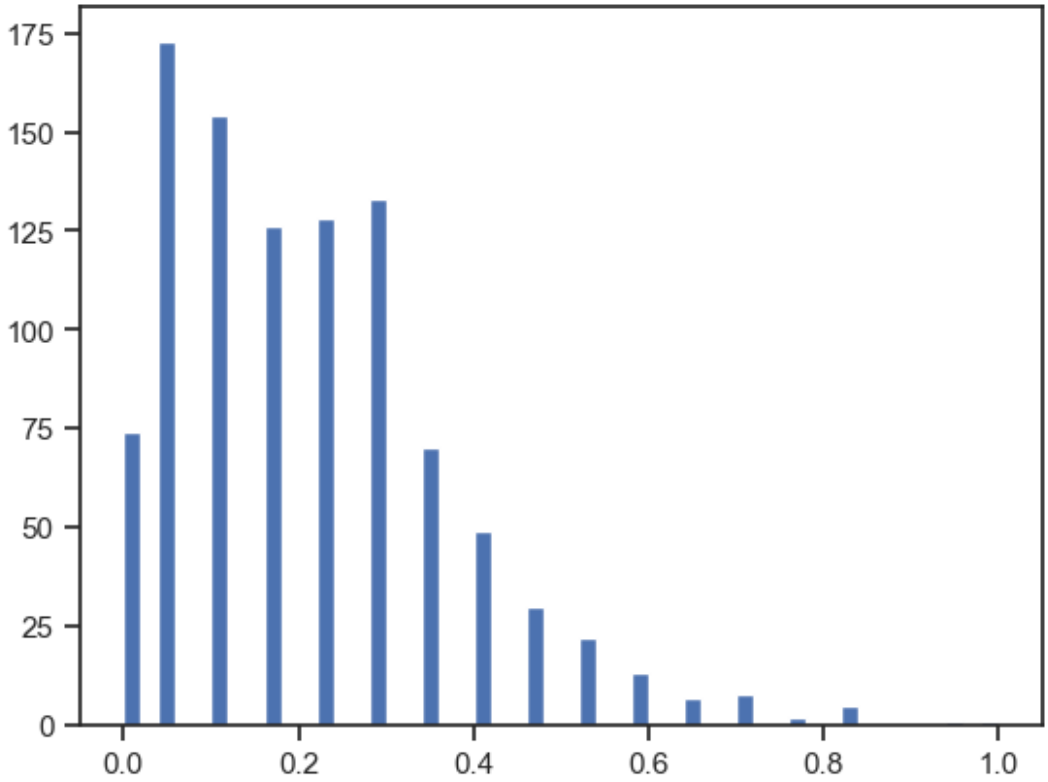
```
from sklearn.preprocessing import MinMaxScaler, StandardScaler,
Normalizer
```

MinMax масштабирование

```
sc1 = MinMaxScaler()
sc1_data = sc1.fit_transform(data[['rating']])

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

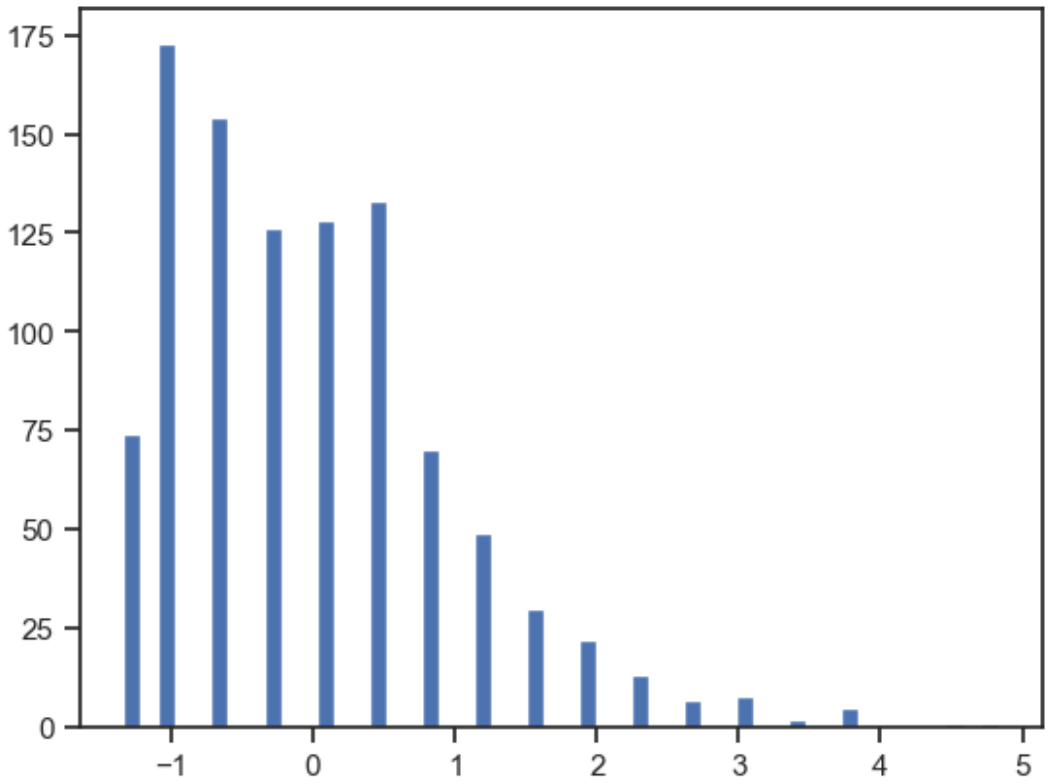


```
plt.hist(sc1_data, 50)
plt.show()
```



Масштабирование данных на основе Z-оценки - StandardScaler

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

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



Дополнительные источники

- Руководство scikit-learn по предобработке данных
- Kaggle Data Cleaning Challenge: Handling missing values (упражнения с пояснениями по обработке пропущенных значений и масштабированию признаков)
- Краткое руководство по категориальным признакам