Московский государственный технический университет им. Н.Э. Баумана
Кафедра «Системы обработки информации и управления»

Лабораторная работа №2
по дисциплине
«Технологии машинного обучения»
на тему
«Обработка пропусков в данных, кодирование категориальных признаков, масштабирование данных»

Выполнил:
студент группы ИУ5-61Б
Агличеев М. С.

Москва —2022 г.

1. 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.

1.0.1. 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.

[89]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
sns.set(style="ticks")
```

1.1. Загрузка и первичный анализ данных

Используем датасет по супергеройским фильмам Most popular superhero TV shows.

[90]:
```
# Будем использовать только обучающую выборку
data = pd.read_csv('data/superheroes.csv', sep=",", thousands=',')
```

[91]:
```
# размер набора данных
data.shape
```

[91]:
```
(750, 8)
```

[92]:
```
# преобразование "object" в действительные типы колонок
data[["runtime", "imdb_votes"]] = data[["runtime", "imdb_votes"]].apply(pd.
 ↪to_numeric)

# типы колонок
data.dtypes
```

[92]:
```
show_title            object
imdb_rating           object
release_year          object
runtime              float64
```

```
genre                 object
parental_guideline    object
imdb_votes           float64
synopsis              object
dtype: object
```

[93]:
```
# проверим есть ли пропущенные значения
data.isnull().sum()
```

[93]:
```
show_title             0
imdb_rating           15
release_year           0
runtime              106
genre                  0
parental_guideline   169
imdb_votes            36
synopsis               0
dtype: int64
```

[94]:
```
# Первые 5 строк датасета
data.head()
```

[94]:

	show_title	imdb_rating	release_year	runtime	genre	parental_guideline	imdb_votes	synopsis
0	Peacemaker	8.5	2022-	40.0	Action, Adventure, Comedy	TV-MA	60116.0	Picking up where The Suicide Squad (2021) left…
1	The Legend of Vox Machina	8.6	2022-	30.0	Animation, Action, Adventure	TV-MA	13128.0	In a desperate attempt to pay off a mounting b…
2	Daredevil	8.6	2015-2018	54.0	Action, Crime, Drama	TV-MA	410433.0	A blind lawyer by day, vigilante by night. Mat…
3	The Boys	8.7	2019-	60.0	Action, Crime, Drama	TV-MA	347831.0	A group of vigilantes set out to take down cor…
4	Raising Dion	7.2	2019-	50.0	Drama, Sci-Fi	TV-G	13375.0	A widowed single mom discovers that her son ha…

[95]:
```
total_count = data.shape[0]
print('Всего строк: {}'.format(total_count))
```

```
Всего строк: 750
```

2. Обработка пропусков в данных

2.1. 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.

[96]:
```
# Удаление колонок, содержащих пустые значения
data_new_1 = data.dropna(axis=1, how='any')
(data.shape, data_new_1.shape)
```

[96]:
```
((750, 8), (750, 4))
```

[97]:
```
# Удаление строк, содержащих пустые значения
data_new_2 = data.dropna(axis=0, how='any')
(data.shape, data_new_2.shape)
```

[97]:
```
((750, 8), (530, 8))
```

[98]:
```
data[10:15]
```

[98]:

	show_title	imdb_rating	release_year	runtime	genre	parental_guideline	imdb_votes	synopsis
10	The Umbrella Academy	8	2019-	60.0	Action, Adventure, Comedy	TV-14	196483.0	A family of former child heroes, now grown apa…
11	Loki	8.3	2021-	NaN	Action, Adventure, Fantasy	TV-14	268311.0	The mercurial villain Loki resumes his role as…
12	Agents of S.H.I.E.L.D.	7.5	2013-2020	45.0	Action, Adventure, Drama	TV-PG	211951.0	The missions of the Strategic Homeland Interve…
13	What If…?	7.5	2021-	32.0	Animation, Action, Adventure	TV-14	89330.0	Exploring pivotal moments from the Marvel Cine…
14	Gotham	7.8	2014-2019	42.0	Action, Crime, Drama	TV-14	223106.0	The story behind Detective James Gordon's rise…

[99]:
```
# 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[10:15]
```

[99]:
```
                show_title imdb_rating release_year  runtime  \
10    The Umbrella Academy           8       2019-      60.0
11                    Loki         8.3       2021-       0.0
12  Agents of S.H.I.E.L.D.         7.5    2013-2020     45.0
13            What If…?         7.5       2021-      32.0
14                 Gotham         7.8    2014-2019     42.0

                          genre parental_guideline  imdb_votes  \
10     Action, Adventure, Comedy              TV-14    196483.0
11    Action, Adventure, Fantasy              TV-14    268311.0
12      Action, Adventure, Drama              TV-PG    211951.0
13  Animation, Action, Adventure              TV-14     89330.0
14          Action, Crime, Drama              TV-14    223106.0

                                             synopsis
10  A family of former child heroes, now grown apa…
11  The mercurial villain Loki resumes his role as…
12  The missions of the Strategic Homeland Interve…
13  Exploring pivotal moments from the Marvel Cine…
14  The story behind Detective James Gordon's rise…
```

2.2. “Внедрение значений” - импьютация (imputation)

2.2.1. Обработка пропусков в числовых данных

[100]:
```
# 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 106, 14.13%.
Колонка imdb_votes. Тип данных float64. Количество пустых значений 36, 4.8%.
```

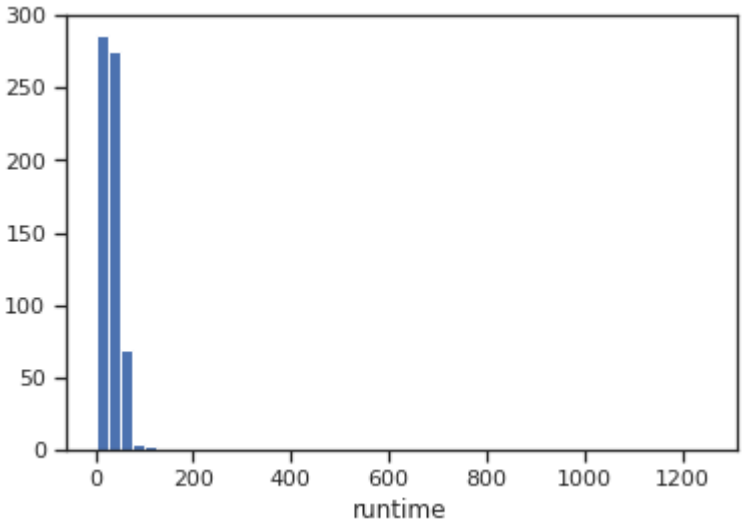
[101]:
```
# Фильтр по колонкам с пропущенными значениями
data_num = data[num_cols]
data_num
```

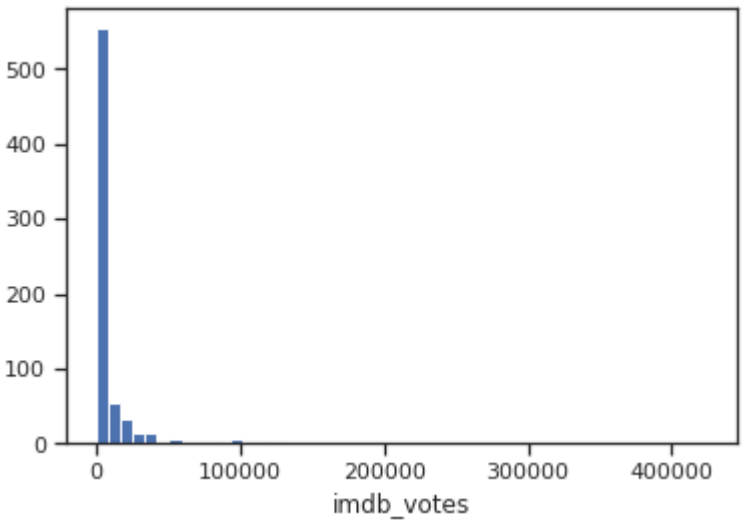
[101]:
```
     runtime  imdb_votes
0       40.0     60116.0
1       30.0     13128.0
2       54.0    410433.0
```

```
3       60.0    347831.0
4       50.0     13375.0
..       …          …
745     30.0       820.0
746     25.0       647.0
747      NaN        62.0
748     27.0        77.0
749     20.0        94.0

[750 rows x 2 columns]
```

[102]:
```
# Гистограмма по признакам
for col in data_num:
    plt.hist(data[col], 50)
    plt.xlabel(col)
    plt.show()
```





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

[103]:
```
data_num_runtime = data_num[['runtime']]
data_num_runtime[10:15]
```

[103]:
```
    runtime
10     60.0
11      NaN
12     45.0
13     32.0
14     42.0
```

[104]:
```
from sklearn.impute import SimpleImputer
from sklearn.impute import MissingIndicator
```

[105]:
```
# Фильтр для проверки заполнения пустых значений
indicator = MissingIndicator()
mask_missing_values_only = indicator.fit_transform(data_num_runtime)
mask_missing_values_only
```

[105]:
```
array([[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],
       [ True],
       [False],
       [False],
```

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

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

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

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

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

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

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

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

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

С помощью класса SimpleImputer можно проводить импьютацию различными показателями центра распределения

[106]:
```
strategies=['mean', 'median', 'most_frequent']
```

[107]:
```
def test_num_impute(strategy_param):
    imp_num = SimpleImputer(strategy=strategy_param)
    data_num_imp = imp_num.fit_transform(data_num_runtime)
    return data_num_imp[mask_missing_values_only]
```

[108]:
```
strategies[0], test_num_impute(strategies[0])
```

[108]:
```
('mean',
 array([34.73291925, 34.73291925, 34.73291925, 34.73291925, 34.73291925,
        34.73291925, 34.73291925, 34.73291925, 34.73291925, 34.73291925,
        34.73291925, 34.73291925, 34.73291925, 34.73291925, 34.73291925,
        34.73291925, 34.73291925, 34.73291925, 34.73291925, 34.73291925,
        34.73291925, 34.73291925, 34.73291925, 34.73291925, 34.73291925,
        34.73291925, 34.73291925, 34.73291925, 34.73291925, 34.73291925,
        34.73291925, 34.73291925, 34.73291925, 34.73291925, 34.73291925,
        34.73291925, 34.73291925, 34.73291925, 34.73291925, 34.73291925,
        34.73291925, 34.73291925, 34.73291925, 34.73291925, 34.73291925,
        34.73291925, 34.73291925, 34.73291925, 34.73291925, 34.73291925,
        34.73291925, 34.73291925, 34.73291925, 34.73291925, 34.73291925,
        34.73291925, 34.73291925, 34.73291925, 34.73291925, 34.73291925,
        34.73291925, 34.73291925, 34.73291925, 34.73291925, 34.73291925,
        34.73291925, 34.73291925, 34.73291925, 34.73291925, 34.73291925,
        34.73291925, 34.73291925, 34.73291925, 34.73291925, 34.73291925,
```

```
        34.73291925, 34.73291925, 34.73291925, 34.73291925, 34.73291925,
        34.73291925, 34.73291925, 34.73291925, 34.73291925, 34.73291925,
        34.73291925, 34.73291925, 34.73291925, 34.73291925, 34.73291925,
        34.73291925, 34.73291925, 34.73291925, 34.73291925, 34.73291925,
        34.73291925, 34.73291925, 34.73291925, 34.73291925, 34.73291925,
        34.73291925, 34.73291925, 34.73291925, 34.73291925, 34.73291925,
        34.73291925]))
```

[109]:
```
strategies[1], test_num_impute(strategies[1])
```

[109]:
```
('median',
 array([30., 30., 30., 30., 30., 30., 30., 30., 30., 30., 30., 30., 30.,
        30., 30., 30., 30., 30., 30., 30., 30., 30., 30., 30., 30., 30.,
        30., 30., 30., 30., 30., 30., 30., 30., 30., 30., 30., 30., 30.,
        30., 30., 30., 30., 30., 30., 30., 30., 30., 30., 30., 30., 30.,
        30., 30., 30., 30., 30., 30., 30., 30., 30., 30., 30., 30., 30.,
        30., 30., 30., 30., 30., 30., 30., 30., 30., 30., 30., 30., 30.,
        30., 30., 30., 30., 30., 30., 30., 30., 30., 30., 30., 30., 30.,
        30., 30., 30., 30., 30., 30., 30., 30., 30., 30., 30., 30., 30.,
        30., 30.]))
```

[110]:
```
strategies[2], test_num_impute(strategies[2])
```

[110]:
```
('most_frequent',
 array([30., 30., 30., 30., 30., 30., 30., 30., 30., 30., 30., 30., 30.,
        30., 30., 30., 30., 30., 30., 30., 30., 30., 30., 30., 30., 30.,
        30., 30., 30., 30., 30., 30., 30., 30., 30., 30., 30., 30., 30.,
        30., 30., 30., 30., 30., 30., 30., 30., 30., 30., 30., 30., 30.,
        30., 30., 30., 30., 30., 30., 30., 30., 30., 30., 30., 30., 30.,
        30., 30., 30., 30., 30., 30., 30., 30., 30., 30., 30., 30., 30.,
        30., 30., 30., 30., 30., 30., 30., 30., 30., 30., 30., 30., 30.,
        30., 30., 30., 30., 30., 30., 30., 30., 30., 30., 30., 30., 30.,
        30., 30.]))
```

[111]:
```
# 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]
```

[112]:
```
data[['imdb_votes']].describe()
```

[112]:
```
           imdb_votes
count      714.000000
mean     13576.459384
std      41723.725840
min         14.000000
25%        546.250000
50%       1758.500000
75%       6901.000000
max     423527.000000
```

[113]:
```
test_num_impute_col(data, 'imdb_votes', strategies[0])
```

[113]:
```
('imdb_votes', 'mean', 36, 13576.459383753501, 13576.459383753501)
```

[114]:
```
test_num_impute_col(data, 'imdb_votes', strategies[1])
```

[114]:
```
('imdb_votes', 'median', 36, 1758.5, 1758.5)
```

[115]:
```
test_num_impute_col(data, 'imdb_votes', strategies[2])
```

[115]:
```
('imdb_votes', 'most_frequent', 36, 133.0, 133.0)
```

2.2.2. Обработка пропусков в категориальных данных

[116]:
```
# 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 imdb_rating. Тип данных object. Количество пустых значений 15, 2.0%.
Колонка parental_guideline. Тип данных object. Количество пустых значений␣
 ↪169,
22.53%.
```

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”.

[117]:
```
cat_temp_data = data[['parental_guideline']]
cat_temp_data[15:20]
```

[117]:
```
   parental_guideline
15              TV-14
16              TV-MA
17              TV-MA
18              TV-14
19                NaN
```

[118]:
```
cat_temp_data['parental_guideline'].unique()
```

[118]:
```
array(['TV-MA', 'TV-G', 'TV-PG', 'TV-14', nan, 'TV-Y7-FV', 'TV-Y7',
       'TV-Y', 'Not Rated', 'R', 'PG-13'], dtype=object)
```

[119]:
```
cat_temp_data[cat_temp_data['parental_guideline'].isnull()].shape
```

[119]:
```
(169, 1)
```

[120]:
```
# Импьютация наиболее частыми значениями
imp2 = SimpleImputer(missing_values=np.nan, strategy='most_frequent')
data_imp2 = imp2.fit_transform(cat_temp_data)
data_imp2
```

[120]:
```
array([['TV-MA'],
       ['TV-MA'],
       ['TV-MA'],
       ['TV-MA'],
       ['TV-G'],
       ['TV-MA'],
       ['TV-PG'],
       ['TV-MA'],
       ['TV-PG'],
       ['TV-14'],
       ['TV-14'],
       ['TV-14'],
       ['TV-PG'],
       ['TV-14'],
       ['TV-14'],
       ['TV-14'],
       ['TV-MA'],
       ['TV-MA'],
       ['TV-14'],
       ['TV-Y7'],
       ['TV-MA'],
       ['TV-Y7-FV'],
       ['TV-Y7-FV'],
       ['TV-Y7'],
       ['TV-14'],
       ['TV-PG'],
       ['TV-PG'],
       ['TV-14'],
       ['TV-Y7'],
       ['TV-MA'],
```

```
['TV-PG'],
['TV-MA'],
['TV-MA'],
['TV-14'],
['TV-PG'],
['TV-14'],
['TV-Y7'],
['TV-14'],
['TV-PG'],
['TV-PG'],
['TV-MA'],
['TV-14'],
['TV-PG'],
['TV-MA'],
['TV-MA'],
['TV-Y7'],
['TV-Y7'],
['TV-MA'],
['TV-PG'],
['TV-PG'],
['TV-PG'],
['TV-PG'],
['TV-14'],
['TV-MA'],
['TV-MA'],
['TV-PG'],
['TV-G'],
['TV-PG'],
['TV-Y7'],
['TV-G'],
['TV-PG'],
['TV-PG'],
['TV-Y7'],
['TV-MA'],
['TV-PG'],
['TV-PG'],
['TV-14'],
['TV-PG'],
['TV-MA'],
['TV-G'],
['TV-Y7'],
['TV-PG'],
['TV-PG'],
['TV-MA'],
['TV-PG'],
['TV-G'],
['TV-Y'],
['TV-G'],
['TV-14'],
['TV-PG'],
```

```
['TV-G'],
['TV-PG'],
['TV-Y7'],
['TV-14'],
['TV-PG'],
['TV-14'],
['TV-PG'],
['TV-Y7-FV'],
['TV-Y7'],
['TV-Y7'],
['TV-Y'],
['TV-MA'],
['TV-PG'],
['TV-G'],
['TV-MA'],
['TV-14'],
['TV-PG'],
['TV-14'],
['TV-14'],
['TV-14'],
['TV-14'],
['TV-Y7'],
['TV-14'],
['TV-Y7'],
['TV-PG'],
['TV-Y7-FV'],
['TV-MA'],
['TV-Y7-FV'],
['TV-MA'],
['TV-PG'],
['TV-Y7'],
['TV-PG'],
['TV-PG'],
['TV-Y7-FV'],
['TV-PG'],
['TV-G'],
['TV-14'],
['TV-Y'],
['TV-Y7'],
['TV-Y7'],
['TV-MA'],
['TV-PG'],
['TV-PG'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7-FV'],
['TV-14'],
['TV-PG'],
['TV-PG'],
['TV-14'],
```

```
['TV-Y7'],
['TV-G'],
['TV-MA'],
['TV-G'],
['Not Rated'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-14'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-G'],
['TV-Y7'],
['TV-Y7-FV'],
['TV-Y7'],
['TV-Y7'],
['TV-MA'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7-FV'],
['TV-PG'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-MA'],
['TV-PG'],
['TV-PG'],
['TV-Y7'],
['TV-Y7-FV'],
['TV-Y7'],
['TV-14'],
['TV-PG'],
['TV-Y7'],
['TV-Y7'],
['TV-MA'],
['TV-MA'],
['TV-Y7'],
['TV-Y7'],
['TV-PG'],
['TV-Y7-FV'],
['TV-Y'],
['TV-PG'],
['TV-Y7-FV'],
['TV-PG'],
['TV-14'],
['TV-Y7'],
['TV-PG'],
```

```
['TV-Y7'],
['TV-G'],
['TV-Y7'],
['TV-MA'],
['TV-Y7'],
['TV-14'],
['TV-Y7'],
['TV-MA'],
['TV-PG'],
['TV-Y'],
['TV-MA'],
['TV-Y7'],
['TV-Y7'],
['TV-PG'],
['TV-PG'],
['TV-Y7'],
['TV-PG'],
['TV-Y7'],
['TV-MA'],
['TV-Y7-FV'],
['TV-Y7'],
['TV-14'],
['TV-G'],
['TV-PG'],
['TV-G'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y'],
['TV-G'],
['TV-14'],
['TV-Y7'],
['TV-MA'],
['TV-Y7'],
['TV-PG'],
['TV-14'],
['TV-Y7'],
['TV-Y7-FV'],
['TV-MA'],
['TV-G'],
['TV-Y7-FV'],
['TV-Y7'],
['TV-G'],
['TV-Y7'],
['TV-MA'],
['TV-Y7'],
['TV-PG'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
```

```
['TV-14'],
['TV-14'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-14'],
['TV-MA'],
['TV-14'],
['TV-Y7'],
['TV-Y7'],
['TV-Y'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-14'],
['TV-G'],
['TV-14'],
['TV-G'],
['TV-PG'],
['TV-PG'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-PG'],
['TV-MA'],
['TV-PG'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-PG'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-14'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-PG'],
['TV-Y7'],
['TV-MA'],
['TV-Y7'],
['TV-Y7'],
['TV-G'],
['TV-Y7'],
['TV-PG'],
['TV-G'],
```

```
['TV-Y7'],
['TV-PG'],
['TV-Y7'],
['TV-MA'],
['TV-Y7'],
['TV-Y7'],
['TV-14'],
['TV-Y7'],
['TV-PG'],
['TV-Y7'],
['TV-Y7-FV'],
['TV-Y7-FV'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7-FV'],
['TV-Y7'],
['TV-14'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-14'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-PG'],
['TV-Y7-FV'],
['R'],
['TV-Y7'],
['TV-G'],
['TV-MA'],
['TV-Y7'],
['TV-Y7'],
['TV-Y'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-PG'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
```

```
['TV-Y7'],
['TV-PG'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-14'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-14'],
['TV-Y7'],
['TV-PG'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-PG'],
['TV-Y7'],
['TV-G'],
['TV-Y'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-14'],
['TV-G'],
['TV-Y7'],
['TV-G'],
['R'],
['TV-Y7'],
['TV-Y7'],
['TV-PG'],
['TV-MA'],
['TV-Y7-FV'],
['TV-MA'],
['TV-G'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-PG'],
['TV-Y7'],
['TV-14'],
['Not Rated'],
['TV-Y'],
['TV-MA'],
['TV-14'],
['TV-Y7'],
['TV-Y7'],
```

```
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-PG'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-G'],
['TV-Y'],
['TV-G'],
['TV-Y7'],
['TV-Y7-FV'],
['TV-Y7'],
['TV-Y7'],
['TV-G'],
['TV-Y7'],
['TV-Y7'],
['TV-G'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-G'],
['TV-Y7'],
['TV-PG'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-PG'],
['TV-Y'],
['TV-Y7-FV'],
['TV-Y7'],
['TV-Y7'],
['TV-PG'],
['TV-Y7'],
['TV-Y7-FV'],
['TV-G'],
['TV-Y7'],
['TV-Y7'],
['TV-Y'],
['TV-G'],
['TV-PG'],
```

```
['TV-Y7'],
['TV-Y7'],
['TV-Y'],
['Not Rated'],
['TV-14'],
['TV-Y7-FV'],
['TV-Y7'],
['TV-Y7'],
['TV-Y'],
['TV-MA'],
['TV-Y7-FV'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7-FV'],
['TV-Y'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-G'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-PG'],
['TV-G'],
['TV-PG'],
['TV-Y7'],
['TV-Y7'],
['TV-G'],
['TV-Y7'],
['TV-Y7'],
['TV-PG'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7-FV'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['Not Rated'],
['TV-PG'],
```

```
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-PG'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7-FV'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-MA'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-14'],
['TV-Y7'],
['TV-Y7'],
['TV-PG'],
['TV-PG'],
['TV-Y7'],
['TV-Y7'],
['TV-PG'],
['TV-G'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7-FV'],
['TV-Y7'],
['TV-14'],
['TV-G'],
['TV-Y7'],
['TV-PG'],
['TV-Y7'],
['TV-14'],
['TV-Y'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
```

```
['TV-Y7'],
['TV-14'],
['TV-Y7'],
['PG-13'],
['TV-Y7-FV'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-G'],
['TV-Y7'],
['TV-Y7-FV'],
['TV-Y'],
['TV-Y7-FV'],
['TV-PG'],
['TV-G'],
['TV-Y'],
['TV-Y7'],
['TV-14'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-14'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-MA'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7-FV'],
['TV-PG'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7-FV'],
['TV-G'],
['TV-PG'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7-FV'],
['TV-Y7'],
['TV-Y7-FV'],
['TV-Y7'],
['TV-Y7'],
```

```
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-PG'],
['TV-Y7'],
['TV-PG'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-PG'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-G'],
['TV-Y7'],
['TV-Y7'],
['TV-PG'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-PG'],
['TV-Y7'],
['TV-G'],
['TV-Y7'],
['TV-PG'],
['TV-PG'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-MA'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7-FV'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-MA'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-MA'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
```

```
['TV-Y7'],
['TV-PG'],
['TV-MA'],
['TV-Y7'],
['TV-PG'],
['TV-Y7'],
['TV-MA'],
['TV-Y7'],
['TV-G'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-G'],
['TV-Y7'],
['TV-Y7'],
['TV-PG'],
['TV-Y7-FV'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7-FV'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7-FV'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7-FV'],
['TV-Y7'],
['TV-PG'],
['TV-Y7'],
['TV-MA'],
['TV-Y'],
['TV-Y7'],
```

```
['TV-Y7'],
['TV-Y7'],
['TV-MA'],
['TV-PG'],
['TV-PG'],
['TV-Y7'],
['TV-Y7'],
['TV-Y'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-PG'],
['TV-14'],
['TV-MA'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-PG'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-PG'],
['TV-PG'],
['TV-Y7'],
['TV-PG'],
['TV-G'],
['TV-Y7'],
['TV-Y7'],
['TV-PG'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7-FV'],
['TV-Y7'],
['TV-G'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-14'],
['TV-14'],
['TV-Y7'],
['TV-Y7'],
['TV-Y'],
['TV-Y7'],
['TV-Y7'],
```

```
       ['TV-PG'],
       ['TV-PG'],
       ['TV-Y7'],
       ['TV-Y7'],
       ['TV-Y7'],
       ['TV-Y7'],
       ['TV-PG'],
       ['TV-PG'],
       ['TV-Y7'],
       ['TV-Y7'],
       ['TV-PG'],
       ['TV-PG'],
       ['TV-PG'],
       ['TV-Y'],
       ['TV-Y7'],
       ['TV-Y7'],
       ['TV-Y7'],
       ['TV-Y7'],
       ['TV-PG'],
       ['TV-Y7']], dtype=object)
```

[121]:
```
# Пустые значения отсутствуют
np.unique(data_imp2)
```

[121]:
```
array(['Not Rated', 'PG-13', 'R', 'TV-14', 'TV-G', 'TV-MA', 'TV-PG',
       'TV-Y', 'TV-Y7', 'TV-Y7-FV'], dtype=object)
```

[122]:
```
# Импьютация константой
imp3 = SimpleImputer(missing_values=np.nan, strategy='constant',␣
 ↪fill_value='Not Rated')
data_imp3 = imp3.fit_transform(cat_temp_data)
data_imp3
```

[122]:
```
array([['TV-MA'],
       ['TV-MA'],
       ['TV-MA'],
       ['TV-MA'],
       ['TV-G'],
       ['TV-MA'],
       ['TV-PG'],
       ['TV-MA'],
       ['TV-PG'],
       ['TV-14'],
       ['TV-14'],
       ['TV-14'],
       ['TV-PG'],
       ['TV-14'],
       ['TV-14'],
       ['TV-14'],
       ['TV-MA'],
       ['TV-MA'],
```

```
['TV-14'],
['Not Rated'],
['TV-MA'],
['TV-Y7-FV'],
['TV-Y7-FV'],
['TV-Y7'],
['TV-14'],
['TV-PG'],
['TV-PG'],
['TV-14'],
['Not Rated'],
['TV-MA'],
['TV-PG'],
['TV-MA'],
['TV-MA'],
['TV-14'],
['TV-PG'],
['TV-14'],
['Not Rated'],
['TV-14'],
['TV-PG'],
['TV-PG'],
['TV-MA'],
['TV-14'],
['TV-PG'],
['TV-MA'],
['TV-MA'],
['TV-Y7'],
['Not Rated'],
['TV-MA'],
['TV-PG'],
['TV-PG'],
['TV-PG'],
['TV-PG'],
['TV-14'],
['TV-MA'],
['TV-MA'],
['TV-PG'],
['TV-G'],
['TV-PG'],
['TV-Y7'],
['TV-G'],
['TV-PG'],
['TV-PG'],
['Not Rated'],
['TV-MA'],
['TV-PG'],
['TV-PG'],
['TV-14'],
['TV-PG'],
```

```
['TV-MA'],
['TV-G'],
['Not Rated'],
['TV-PG'],
['TV-PG'],
['TV-MA'],
['TV-PG'],
['TV-G'],
['TV-Y'],
['TV-G'],
['TV-14'],
['TV-PG'],
['TV-G'],
['TV-PG'],
['Not Rated'],
['TV-14'],
['TV-PG'],
['TV-14'],
['TV-PG'],
['TV-Y7-FV'],
['TV-Y7'],
['TV-Y7'],
['TV-Y'],
['TV-MA'],
['TV-PG'],
['TV-G'],
['TV-MA'],
['TV-14'],
['TV-PG'],
['TV-14'],
['TV-14'],
['TV-14'],
['TV-14'],
['Not Rated'],
['TV-14'],
['TV-Y7'],
['TV-PG'],
['TV-Y7-FV'],
['TV-MA'],
['TV-Y7-FV'],
['TV-MA'],
['TV-PG'],
['TV-Y7'],
['TV-PG'],
['TV-PG'],
['TV-Y7-FV'],
['TV-PG'],
['TV-G'],
['TV-14'],
['TV-Y'],
```

```
['TV-Y7'],
['TV-Y7'],
['TV-MA'],
['TV-PG'],
['TV-PG'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7-FV'],
['TV-14'],
['TV-PG'],
['TV-PG'],
['TV-14'],
['TV-Y7'],
['TV-G'],
['TV-MA'],
['TV-G'],
['Not Rated'],
['TV-Y7'],
['TV-Y7'],
['Not Rated'],
['TV-Y7'],
['TV-14'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-G'],
['TV-Y7'],
['TV-Y7-FV'],
['TV-Y7'],
['TV-Y7'],
['TV-MA'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7-FV'],
['TV-PG'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-MA'],
['TV-PG'],
['TV-PG'],
['TV-Y7'],
['TV-Y7-FV'],
['Not Rated'],
['TV-14'],
['TV-PG'],
['TV-Y7'],
['TV-Y7'],
['TV-MA'],
```

```
['TV-MA'],
['TV-Y7'],
['TV-Y7'],
['TV-PG'],
['TV-Y7-FV'],
['TV-Y'],
['TV-PG'],
['TV-Y7-FV'],
['TV-PG'],
['TV-14'],
['TV-Y7'],
['TV-PG'],
['Not Rated'],
['TV-G'],
['Not Rated'],
['TV-MA'],
['Not Rated'],
['TV-14'],
['TV-Y7'],
['TV-MA'],
['TV-PG'],
['TV-Y'],
['TV-MA'],
['Not Rated'],
['TV-Y7'],
['TV-PG'],
['TV-PG'],
['TV-Y7'],
['TV-PG'],
['TV-Y7'],
['TV-MA'],
['TV-Y7-FV'],
['TV-Y7'],
['TV-14'],
['TV-G'],
['TV-PG'],
['TV-G'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y'],
['TV-G'],
['TV-14'],
['Not Rated'],
['TV-MA'],
['TV-Y7'],
['TV-PG'],
['TV-14'],
['TV-Y7'],
['TV-Y7-FV'],
```

```
['TV-MA'],
['TV-G'],
['TV-Y7-FV'],
['TV-Y7'],
['TV-G'],
['TV-Y7'],
['TV-MA'],
['Not Rated'],
['TV-PG'],
['TV-Y7'],
['Not Rated'],
['TV-Y7'],
['TV-14'],
['TV-14'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-14'],
['TV-MA'],
['TV-14'],
['TV-Y7'],
['Not Rated'],
['TV-Y'],
['TV-Y7'],
['Not Rated'],
['TV-Y7'],
['TV-14'],
['TV-G'],
['TV-14'],
['TV-G'],
['TV-PG'],
['TV-PG'],
['TV-Y7'],
['Not Rated'],
['TV-Y7'],
['TV-PG'],
['TV-MA'],
['TV-PG'],
['TV-Y7'],
['Not Rated'],
['Not Rated'],
['Not Rated'],
['TV-Y7'],
['TV-Y7'],
['TV-PG'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-14'],
['TV-Y7'],
```

```
['Not Rated'],
['TV-Y7'],
['TV-Y7'],
['TV-PG'],
['Not Rated'],
['TV-MA'],
['Not Rated'],
['TV-Y7'],
['TV-G'],
['TV-Y7'],
['TV-PG'],
['TV-G'],
['Not Rated'],
['TV-PG'],
['TV-Y7'],
['TV-MA'],
['TV-Y7'],
['TV-Y7'],
['TV-14'],
['TV-Y7'],
['TV-PG'],
['Not Rated'],
['TV-Y7-FV'],
['TV-Y7-FV'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7-FV'],
['TV-Y7'],
['TV-14'],
['Not Rated'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-14'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['Not Rated'],
['TV-Y7'],
['TV-PG'],
['TV-Y7-FV'],
['R'],
['TV-Y7'],
['TV-G'],
['TV-MA'],
['Not Rated'],
['TV-Y7'],
['TV-Y'],
['TV-Y7'],
```

```
['TV-Y7'],
['TV-Y7'],
['Not Rated'],
['TV-PG'],
['TV-Y7'],
['TV-Y7'],
['Not Rated'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['Not Rated'],
['TV-PG'],
['TV-Y7'],
['Not Rated'],
['Not Rated'],
['TV-14'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['Not Rated'],
['Not Rated'],
['TV-14'],
['TV-Y7'],
['TV-PG'],
['Not Rated'],
['Not Rated'],
['Not Rated'],
['TV-PG'],
['TV-Y7'],
['TV-G'],
['TV-Y'],
['Not Rated'],
['TV-Y7'],
['TV-Y7'],
['Not Rated'],
['TV-14'],
['TV-G'],
['TV-Y7'],
['TV-G'],
['R'],
['TV-Y7'],
['TV-Y7'],
['TV-PG'],
['TV-MA'],
['TV-Y7-FV'],
['TV-MA'],
['TV-G'],
```

```
['TV-Y7'],
['TV-Y7'],
['Not Rated'],
['TV-PG'],
['TV-Y7'],
['TV-14'],
['Not Rated'],
['TV-Y'],
['TV-MA'],
['TV-14'],
['TV-Y7'],
['Not Rated'],
['TV-Y7'],
['TV-Y7'],
['Not Rated'],
['TV-Y7'],
['TV-Y7'],
['Not Rated'],
['TV-PG'],
['Not Rated'],
['TV-Y7'],
['TV-Y7'],
['TV-G'],
['TV-Y'],
['TV-G'],
['Not Rated'],
['TV-Y7-FV'],
['Not Rated'],
['TV-Y7'],
['TV-G'],
['TV-Y7'],
['TV-Y7'],
['TV-G'],
['Not Rated'],
['TV-Y7'],
['TV-Y7'],
['TV-G'],
['TV-Y7'],
['TV-PG'],
['TV-Y7'],
['Not Rated'],
['Not Rated'],
['Not Rated'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-PG'],
['TV-Y'],
```

```
['TV-Y7-FV'],
['Not Rated'],
['TV-Y7'],
['TV-PG'],
['Not Rated'],
['TV-Y7-FV'],
['TV-G'],
['Not Rated'],
['TV-Y7'],
['TV-Y'],
['TV-G'],
['TV-PG'],
['TV-Y7'],
['Not Rated'],
['TV-Y'],
['Not Rated'],
['TV-14'],
['TV-Y7-FV'],
['TV-Y7'],
['TV-Y7'],
['TV-Y'],
['TV-MA'],
['TV-Y7-FV'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7-FV'],
['TV-Y'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['Not Rated'],
['TV-Y7'],
['TV-Y7'],
['Not Rated'],
['TV-Y7'],
['TV-Y7'],
['TV-G'],
['Not Rated'],
['TV-Y7'],
['Not Rated'],
['Not Rated'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-PG'],
['TV-G'],
['TV-PG'],
['Not Rated'],
['TV-Y7'],
```

```
['TV-G'],
['TV-Y7'],
['Not Rated'],
['TV-PG'],
['TV-Y7'],
['Not Rated'],
['TV-Y7-FV'],
['TV-Y7'],
['Not Rated'],
['TV-Y7'],
['Not Rated'],
['TV-PG'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-PG'],
['Not Rated'],
['Not Rated'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7-FV'],
['TV-Y7'],
['Not Rated'],
['TV-Y7'],
['TV-MA'],
['Not Rated'],
['TV-Y7'],
['Not Rated'],
['TV-Y7'],
['Not Rated'],
['TV-14'],
['Not Rated'],
['TV-Y7'],
['TV-PG'],
['TV-PG'],
['Not Rated'],
['TV-Y7'],
['TV-PG'],
['TV-G'],
['TV-Y7'],
['TV-Y7'],
['Not Rated'],
['Not Rated'],
['TV-Y7-FV'],
['Not Rated'],
['TV-14'],
['TV-G'],
['TV-Y7'],
['TV-PG'],
```

```
['TV-Y7'],
['TV-14'],
['TV-Y'],
['Not Rated'],
['Not Rated'],
['Not Rated'],
['TV-Y7'],
['Not Rated'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['Not Rated'],
['TV-Y7'],
['TV-14'],
['Not Rated'],
['PG-13'],
['TV-Y7-FV'],
['TV-Y7'],
['Not Rated'],
['Not Rated'],
['TV-G'],
['Not Rated'],
['TV-Y7-FV'],
['TV-Y'],
['TV-Y7-FV'],
['TV-PG'],
['TV-G'],
['TV-Y'],
['Not Rated'],
['TV-14'],
['Not Rated'],
['TV-Y7'],
['Not Rated'],
['TV-14'],
['Not Rated'],
['TV-Y7'],
['TV-Y7'],
['Not Rated'],
['TV-MA'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['Not Rated'],
['TV-Y7'],
['TV-Y7-FV'],
['TV-PG'],
['Not Rated'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
```

```
['TV-Y7-FV'],
['TV-G'],
['TV-PG'],
['Not Rated'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7-FV'],
['Not Rated'],
['TV-Y7-FV'],
['Not Rated'],
['Not Rated'],
['TV-Y7'],
['Not Rated'],
['Not Rated'],
['TV-PG'],
['TV-Y7'],
['TV-PG'],
['Not Rated'],
['Not Rated'],
['Not Rated'],
['TV-PG'],
['Not Rated'],
['Not Rated'],
['Not Rated'],
['TV-Y7'],
['TV-G'],
['Not Rated'],
['Not Rated'],
['TV-PG'],
['TV-Y7'],
['Not Rated'],
['Not Rated'],
['TV-Y7'],
['Not Rated'],
['TV-PG'],
['Not Rated'],
['TV-G'],
['Not Rated'],
['TV-PG'],
['TV-PG'],
['Not Rated'],
['Not Rated'],
['TV-Y7'],
['Not Rated'],
['TV-Y7'],
['TV-MA'],
['TV-Y7'],
['TV-Y7'],
['TV-Y7-FV'],
```

```
['TV-Y7'],
['Not Rated'],
['TV-Y7'],
['TV-MA'],
['TV-Y7'],
['Not Rated'],
['TV-Y7'],
['Not Rated'],
['TV-MA'],
['TV-Y7'],
['Not Rated'],
['Not Rated'],
['TV-Y7'],
['TV-PG'],
['TV-MA'],
['TV-Y7'],
['TV-PG'],
['TV-Y7'],
['TV-MA'],
['TV-Y7'],
['TV-G'],
['Not Rated'],
['Not Rated'],
['Not Rated'],
['TV-Y7'],
['TV-G'],
['Not Rated'],
['TV-Y7'],
['TV-PG'],
['TV-Y7-FV'],
['TV-Y7'],
['TV-Y7'],
['Not Rated'],
['Not Rated'],
['Not Rated'],
['Not Rated'],
['TV-Y'],
['Not Rated'],
['Not Rated'],
['TV-Y7-FV'],
['TV-Y7'],
['Not Rated'],
['TV-Y7'],
['TV-Y'],
['TV-Y7'],
['TV-Y7'],
['Not Rated'],
['Not Rated'],
['Not Rated'],
['TV-Y7-FV'],
```

```
['Not Rated'],
['TV-Y7'],
['Not Rated'],
['Not Rated'],
['Not Rated'],
['TV-Y7-FV'],
['Not Rated'],
['TV-PG'],
['Not Rated'],
['TV-MA'],
['TV-Y'],
['Not Rated'],
['Not Rated'],
['TV-Y7'],
['TV-MA'],
['TV-PG'],
['TV-PG'],
['Not Rated'],
['Not Rated'],
['TV-Y'],
['TV-Y7'],
['Not Rated'],
['TV-Y7'],
['TV-PG'],
['TV-14'],
['TV-MA'],
['TV-Y7'],
['Not Rated'],
['Not Rated'],
['Not Rated'],
['TV-PG'],
['Not Rated'],
['Not Rated'],
['Not Rated'],
['Not Rated'],
['TV-PG'],
['TV-PG'],
['TV-Y7'],
['TV-PG'],
['TV-G'],
['Not Rated'],
['Not Rated'],
['TV-PG'],
['TV-Y7'],
['Not Rated'],
['Not Rated'],
['Not Rated'],
['TV-Y7-FV'],
['Not Rated'],
['TV-G'],
```

```
       ['TV-Y7'],
       ['Not Rated'],
       ['TV-Y7'],
       ['TV-Y7'],
       ['Not Rated'],
       ['TV-14'],
       ['TV-14'],
       ['Not Rated'],
       ['TV-Y7'],
       ['TV-Y'],
       ['TV-Y7'],
       ['Not Rated'],
       ['TV-PG'],
       ['TV-PG'],
       ['TV-Y7'],
       ['Not Rated'],
       ['TV-Y7'],
       ['Not Rated'],
       ['TV-PG'],
       ['TV-PG'],
       ['TV-Y7'],
       ['Not Rated'],
       ['TV-PG'],
       ['TV-PG'],
       ['TV-PG'],
       ['TV-Y'],
       ['Not Rated'],
       ['Not Rated'],
       ['TV-Y7'],
       ['TV-Y7'],
       ['TV-PG'],
       ['Not Rated']], dtype=object)
```

[123]:
```
np.unique(data_imp3)
```

[123]:
```
array(['Not Rated', 'PG-13', 'R', 'TV-14', 'TV-G', 'TV-MA', 'TV-PG',
       'TV-Y', 'TV-Y7', 'TV-Y7-FV'], dtype=object)
```

[124]:
```
data_imp3[data_imp3=='Not Rated'].size
```

[124]:
```
173
```

3. Преобразование категориальных признаков в числовые

[125]:
```
cat_enc = pd.DataFrame({'parental_guideline':data_imp2.T[0]})
cat_enc
```

[125]:
```
    parental_guideline
0                TV-MA
1                TV-MA
2                TV-MA
3                TV-MA
4                 TV-G
..                  …
745              TV-Y7
746              TV-Y7
747              TV-Y7
748              TV-PG
749              TV-Y7

[750 rows x 1 columns]
```

3.1. Кодирование категорий целочисленными значениями - label encoding

[126]:
```
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
```

[127]:
```
le = LabelEncoder()
cat_enc_le = le.fit_transform(cat_enc['parental_guideline'])
```

[128]:
```
cat_enc['parental_guideline'].unique()
```

[128]:
```
array(['TV-MA', 'TV-G', 'TV-PG', 'TV-14', 'TV-Y7', 'TV-Y7-FV', 'TV-Y',
       'Not Rated', 'R', 'PG-13'], dtype=object)
```

[129]:
```
np.unique(cat_enc_le)
```

[129]:
```
array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

[130]:
```
le.inverse_transform([0, 1, 2, 3])
```

[130]:
```
array(['Not Rated', 'PG-13', 'R', 'TV-14'], dtype=object)
```

3.2. Кодирование категорий наборами бинарных значений - one-hot encoding

[131]:
```
ohe = OneHotEncoder()
cat_enc_ohe = ohe.fit_transform(cat_enc[['parental_guideline']])
```

[132]:
```
cat_enc.shape
```

[132]:
```
(750, 1)
```

[133]:
```
cat_enc_ohe.shape
```

[133]:
```
(750, 10)
```

[134]:
```
cat_enc_ohe
```

[134]:
```
<750x10 sparse matrix of type '<class 'numpy.float64'>'
        with 750 stored elements in Compressed Sparse Row format>
```

[135]:
```
cat_enc_ohe.todense()[0:10]
```

[135]:
```
matrix([[0., 0., 0., 0., 0., 1., 0., 0., 0., 0.],
        [0., 0., 0., 0., 0., 1., 0., 0., 0., 0.],
        [0., 0., 0., 0., 0., 1., 0., 0., 0., 0.],
        [0., 0., 0., 0., 0., 1., 0., 0., 0., 0.],
        [0., 0., 0., 0., 1., 0., 0., 0., 0., 0.],
        [0., 0., 0., 0., 0., 1., 0., 0., 0., 0.],
        [0., 0., 0., 0., 0., 0., 1., 0., 0., 0.],
        [0., 0., 0., 0., 0., 1., 0., 0., 0., 0.],
        [0., 0., 0., 0., 0., 0., 1., 0., 0., 0.],
        [0., 0., 0., 1., 0., 0., 0., 0., 0., 0.]])
```

[136]:
```
cat_enc.head(10)
```

[136]:

	parental_guideline
0	TV-MA
1	TV-MA
2	TV-MA
3	TV-MA
4	TV-G
5	TV-MA
6	TV-PG
7	TV-MA
8	TV-PG
9	TV-14

3.3. Pandas get_dummies - быстрый вариант one-hot кодирования

[137]:
```
pd.get_dummies(cat_enc).head()
```

[137]:

	parental_guideline_Not Rated	parental_guideline_PG-13	parental_guideline_R	parental_guideline_TV-14	parental_guideline_TV-G
0	0	0	0	0	0
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0
4	0	0			

```
4                     0                           0                           1

   parental_guideline_TV-MA  parental_guideline_TV-PG  \
0                         1                         0
1                         1                         0
2                         1                         0
3                         1                         0
4                         0                         0

   parental_guideline_TV-Y  parental_guideline_TV-Y7  \
0                        0                         0
1                        0                         0
2                        0                         0
3                        0                         0
4                        0                         0

   parental_guideline_TV-Y7-FV
0                            0
1                            0
2                            0
3                            0
4                            0
```

[138]:
```
pd.get_dummies(cat_temp_data, dummy_na=True).head()
```

[138]:
```
   parental_guideline_Not Rated  parental_guideline_PG-13  \
0                             0                         0
1                             0                         0
2                             0                         0
3                             0                         0
4                             0                         0

   parental_guideline_R  parental_guideline_TV-14  parental_guideline_TV-G␣
↪ \
0                     0                         0                         0
1                     0                         0                         0
2                     0                         0                         0
3                     0                         0                         0
4                     0                         0                         1

   parental_guideline_TV-MA  parental_guideline_TV-PG  \
0                         1                         0
1                         1                         0
2                         1                         0
3                         1                         0
4                         0                         0

   parental_guideline_TV-Y  parental_guideline_TV-Y7  \
0                        0                         0
1                        0                         0
```

```
2                          0                          0
3                          0                          0
4                          0                          0

   parental_guideline_TV-Y7-FV  parental_guideline_nan
0                            0                       0
1                            0                       0
2                            0                       0
3                            0                       0
4                            0                       0
```

[ ]:

4. 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:

$$x = \frac{x - min(X)}{max(X) - min(X)}$$

В этом случае значения лежат в диапазоне от 0 до 1. - Масштабирование данных на основе Z-оценки:

$$x = \frac{x - AVG(X)}{\sigma(X)}$$

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

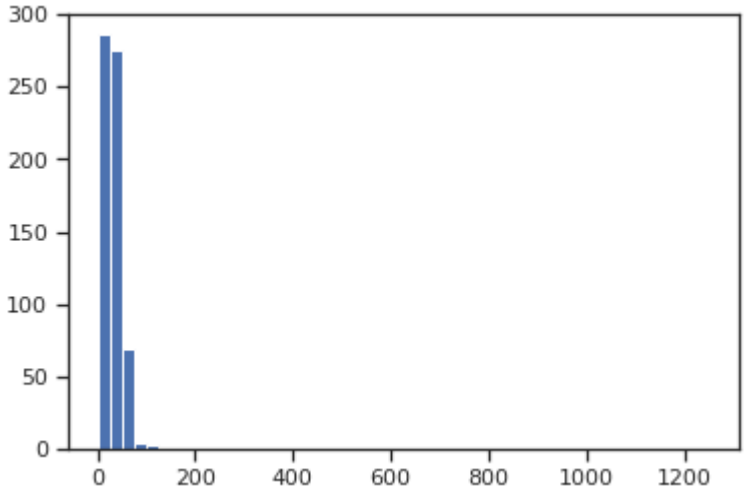
где X - матрица объект-признак, $AVG(X)$ - среднее значение, σ - среднеквадратичное отклонение.

[139]:
```
from sklearn.preprocessing import MinMaxScaler, StandardScaler, Normalizer
```

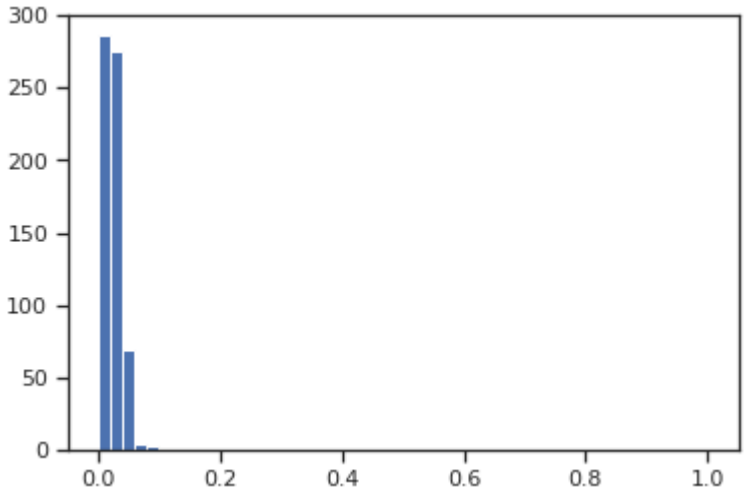
4.1. MinMax масштабирование

[140]:
```
sc1 = MinMaxScaler()
sc1_data = sc1.fit_transform(data[['runtime']])
```

[141]:
```
plt.hist(data['runtime'], 50)
plt.show()
```



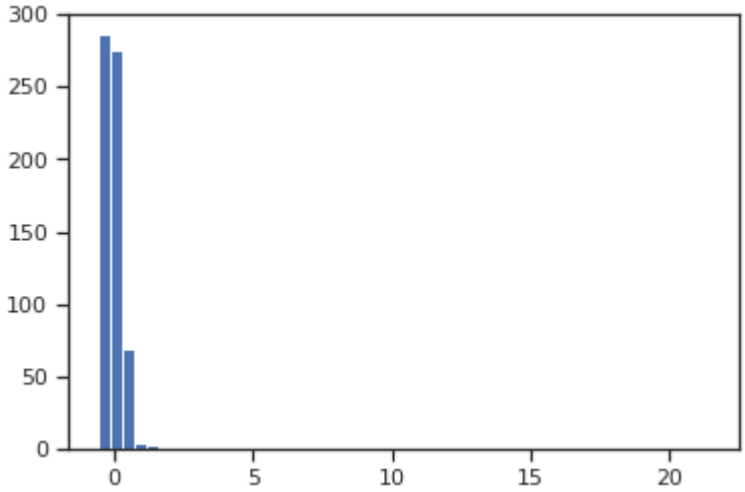
[142]:
```
plt.hist(sc1_data, 50)
plt.show()
```



4.2. Масштабирование данных на основе Z-оценки - StandardScaler

[144]:
```
sc2 = StandardScaler()
sc2_data = sc2.fit_transform(data[['runtime']])
```

[145]:
```
plt.hist(sc2_data, 50)
plt.show()
```



5. 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