Московский Государственный Технический Университет
им. Н.Э. Баумана

Отчет по лабораторной работе №3
по курсу
Технологии Машинного Обучения

Выполнила:
Костян Алина
ИУ5-53

_____

Проверил:
Гапанюк Ю.Е.

_____

Москва, 2019

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

Задание:

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

In [1]:

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.impute import SimpleImputer
from sklearn.impute import MissingIndicator
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
from sklearn.preprocessing import MinMaxScaler, StandardScaler, Normalizer
%matplotlib inline
sns.set(style="ticks")
```

In [2]:

```
data = pd.read_csv('train.csv', sep=",")
```

Размер датасета

In [3]:

```
data.shape
```

Out[3]:

```
(891, 12)
```

Типы данных колонок

In [4]:

```
data.dtypes
```

Out[4]:

```
PassengerId      int64
Survived         int64
Pclass           int64
Name            object
Sex             object
Age            float64
SibSp            int64
Parch            int64
Ticket          object
Fare           float64
Cabin           object
Embarked        object
dtype: object
```

Проверим были ли пропущены значения в каких-нибудь колонках

In [5]:

```
data.isnull().sum()
```

Out[5]:

```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
dtype: int64
```

Первые пять строк датасета

In [6]:

```
data.head()
```

Out[6]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500

In [7]:

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

```
Всего строк: 891
```

1. Обработка пропущенных данных

1.1. Простые стратегии - удаление или заполнение нулями

1.1.1. Удаление колонок содержащих пустые значения

In [8]:

```
data_new_1 = data.dropna(axis=1, how='any')
(data.shape, data_new_1.shape)
```

Out[8]:

```
((891, 12), (891, 9))
```

1.1.2. Удаление строк содержащих пустые значения

In [9]:

```
data_new_2 = data.dropna(axis=0, how='any')
(data.shape, data_new_2.shape)
```

Out[9]:

```
((891, 12), (183, 12))
```

1.1.2. Заполнение всех пропущенных значений нулями, что некорректно для категориальных знаичений

In [10]:

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

Out[10]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500

"Внедрение значений" - импьютация

1.2.1. Обработка пропусков в числовых данных

Выберем числовые колонки с пропущенными значениями

In [11]:

```
num_cols = [col for col in data.columns if (data[data[col].isnull()].shape[0] >
0 and (data[col].dtype=='float64' or

data[col].dtype=='int64'))]
for col in num_cols:
    print(f"Колонка {col}, количество пропусков {data[col].isnull().sum()} - {round
((data[col].isnull().sum()/total_count)*100,2)}%")
```

```
Колонка Age, количество пропусков 177 - 19.87%
```

Фильтр по колонкам с пропущенными значениями

In [12]:

```
data_num = data[num_cols]
data_num
```

Out[12]:

	Age
0	22.0
1	38.0
2	26.0
3	35.0
4	35.0
5	NaN
6	54.0
7	2.0
8	27.0
9	14.0
10	4.0
11	58.0
12	20.0
13	39.0
14	14.0
15	55.0
16	2.0
17	NaN
18	31.0
19	NaN
20	35.0
21	34.0
22	15.0
23	28.0
24	8.0
25	38.0
26	NaN
27	19.0
28	NaN
29	NaN
...	...
861	21.0
862	48.0
863	NaN
864	24.0
865	42.0

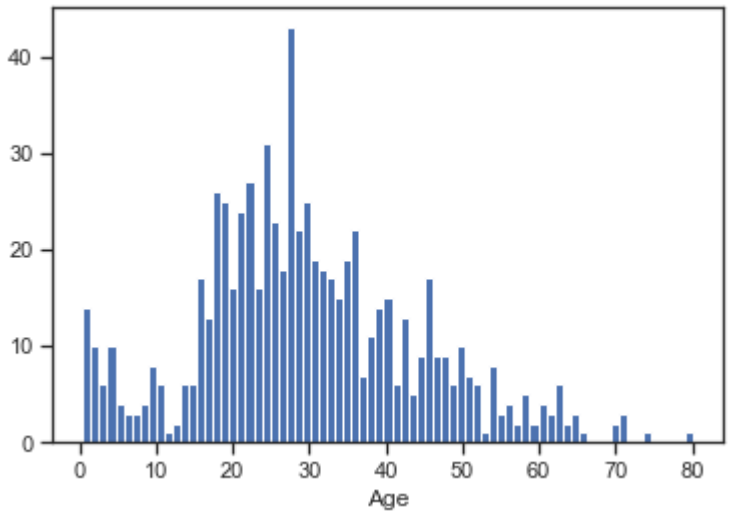
	Age
866	27.0
867	31.0
868	NaN
869	4.0
870	26.0
871	47.0
872	33.0
873	47.0
874	28.0
875	15.0
876	20.0
877	19.0
878	NaN
879	56.0
880	25.0
881	33.0
882	22.0
883	28.0
884	25.0
885	39.0
886	27.0
887	19.0
888	NaN
889	26.0
890	32.0

891 rows × 1 columns

In [13]:

```
for col in data_num:
    plt.hist(data[col], 75)
    plt.xlabel(col)
    plt.show()
```

```
c:\users\linalt\ml\.venv\lib\site-packages\numpy\lib\histograms.py:8
24: RuntimeWarning: invalid value encountered in greater_equal
  keep = (tmp_a >= first_edge)
c:\users\linalt\ml\.venv\lib\site-packages\numpy\lib\histograms.py:8
25: RuntimeWarning: invalid value encountered in less_equal
  keep &= (tmp_a <= last_edge)
```



In [14]:

```
data[data['Age'].isnull()]
```

Out[14]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	
5	6	0	3	Moran, Mr. James	male	NaN	0	0	330877	8
17	18	1	2	Williams, Mr. Charles Eugene	male	NaN	0	0	244373	13
19	20	1	3	Masselmani, Mrs. Fatima	female	NaN	0	0	2649	7
26	27	0	3	Emir, Mr. Farred Chehab	male	NaN	0	0	2631	7
28	29	1	3	O'Dwyer, Miss. Ellen "Nellie"	female	NaN	0	0	330959	7
29	30	0	3	Todoroff, Mr. Lalio	male	NaN	0	0	349216	7
31	32	1	1	Spencer, Mrs. William Augustus (Marie Eugenie)	female	NaN	1	0	PC 17569	146
32	33	1	3	Glynn, Miss. Mary Agatha	female	NaN	0	0	335677	7
36	37	1	3	Mamee, Mr. Hanna	male	NaN	0	0	2677	7
42	43	0	3	Kraeff, Mr. Theodor	male	NaN	0	0	349253	7
45	46	0	3	Rogers, Mr. William John	male	NaN	0	0	S.C./A.4. 23567	8
46	47	0	3	Lennon, Mr. Denis	male	NaN	1	0	370371	15
47	48	1	3	O'Driscoll, Miss. Bridget	female	NaN	0	0	14311	7
48	49	0	3	Samaan, Mr. Youssef	male	NaN	2	0	2662	21
55	56	1	1	Woolner, Mr. Hugh	male	NaN	0	0	19947	35
64	65	0	1	Stewart, Mr. Albert A	male	NaN	0	0	PC 17605	27
65	66	1	3	Moubarek, Master. Gerios	male	NaN	1	1	2661	15
76	77	0	3	Staneff, Mr. Ivan	male	NaN	0	0	349208	7
77	78	0	3	Moutal, Mr. Rahamin Haim	male	NaN	0	0	374746	8
82	83	1	3	McDermott, Miss. Brigdet Delia	female	NaN	0	0	330932	7

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	
87	88	0	3	Slocovski, Mr. Selman Francis	male	NaN	0	0	SOTON/OQ 392086	8
95	96	0	3	Shorney, Mr. Charles Joseph	male	NaN	0	0	374910	8
101	102	0	3	Petroff, Mr. Pastcho ("Pentcho")	male	NaN	0	0	349215	7
107	108	1	3	Moss, Mr. Albert Johan	male	NaN	0	0	312991	7
109	110	1	3	Moran, Miss. Bertha	female	NaN	1	0	371110	24
121	122	0	3	Moore, Mr. Leonard Charles	male	NaN	0	0	A4. 54510	8
126	127	0	3	McMahon, Mr. Martin	male	NaN	0	0	370372	7
128	129	1	3	Peter, Miss. Anna	female	NaN	1	1	2668	22
140	141	0	3	Boulos, Mrs. Joseph (Sultana)	female	NaN	0	2	2678	15
154	155	0	3	Olsen, Mr. Ole Martin	male	NaN	0	0	Fa 265302	7
...	...	...	...	...	...	...	...	...	...	
718	719	0	3	McEvoy, Mr. Michael	male	NaN	0	0	36568	15
727	728	1	3	Mannion, Miss. Margareth	female	NaN	0	0	36866	7
732	733	0	2	Knight, Mr. Robert J	male	NaN	0	0	239855	0
738	739	0	3	Ivanoff, Mr. Kanio	male	NaN	0	0	349201	7
739	740	0	3	Nankoff, Mr. Minko	male	NaN	0	0	349218	7
740	741	1	1	Hawksford, Mr. Walter James	male	NaN	0	0	16988	30
760	761	0	3	Garfirth, Mr. John	male	NaN	0	0	358585	14
766	767	0	1	Brewe, Dr. Arthur Jackson	male	NaN	0	0	112379	39
768	769	0	3	Moran, Mr. Daniel J	male	NaN	1	0	371110	24
773	774	0	3	Elias, Mr. Dibo	male	NaN	0	0	2674	7
776	777	0	3	Tobin, Mr. Roger	male	NaN	0	0	383121	7

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	
778	779	0	3	Kilgannon, Mr. Thomas J	male	NaN	0	0	36865	7
783	784	0	3	Johnston, Mr. Andrew G	male	NaN	1	2	W./C. 6607	23
790	791	0	3	Keane, Mr. Andrew "Andy"	male	NaN	0	0	12460	7
792	793	0	3	Sage, Miss. Stella Anna	female	NaN	8	2	CA. 2343	69
793	794	0	1	Hoyt, Mr. William Fisher	male	NaN	0	0	PC 17600	30
815	816	0	1	Fry, Mr. Richard	male	NaN	0	0	112058	B
825	826	0	3	Flynn, Mr. John	male	NaN	0	0	368323	6
826	827	0	3	Lam, Mr. Len	male	NaN	0	0	1601	56
828	829	1	3	McCormack, Mr. Thomas Joseph	male	NaN	0	0	367228	7
832	833	0	3	Saad, Mr. Amin	male	NaN	0	0	2671	7
837	838	0	3	Sirota, Mr. Maurice	male	NaN	0	0	392092	8
839	840	1	1	Marechal, Mr. Pierre	male	NaN	0	0	11774	29
846	847	0	3	Sage, Mr. Douglas Bullen	male	NaN	8	2	CA. 2343	69
849	850	1	1	Goldenberg, Mrs. Samuel L (Edwiga Grabowska)	female	NaN	1	0	17453	89
859	860	0	3	Razi, Mr. Raihed	male	NaN	0	0	2629	7
863	864	0	3	Sage, Miss. Dorothy Edith "Dolly"	female	NaN	8	2	CA. 2343	69
868	869	0	3	van Melkebeke, Mr. Philemon	male	NaN	0	0	345777	9
878	879	0	3	Laleff, Mr. Kristo	male	NaN	0	0	349217	7
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23

177 rows × 12 columns

In [15]:

```
missedValues = data[data['Age'].isnull()].index
missedValues
```

Out[15]:

```
Int64Index([  5,  17,  19,  26,  28,  29,  31,  32,  36,  42,
             ...
             832, 837, 839, 846, 849, 859, 863, 868, 878, 888],
            dtype='int64', length=177)
```

In [16]:

```
data[data.index.isin(missedValues)]
```

Out[16]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	
5	6	0	3	Moran, Mr. James	male	NaN	0	0	330877	8
17	18	1	2	Williams, Mr. Charles Eugene	male	NaN	0	0	244373	13
19	20	1	3	Masselmani, Mrs. Fatima	female	NaN	0	0	2649	7
26	27	0	3	Emir, Mr. Farred Chehab	male	NaN	0	0	2631	7
28	29	1	3	O'Dwyer, Miss. Ellen "Nellie"	female	NaN	0	0	330959	7
29	30	0	3	Todoroff, Mr. Lalio	male	NaN	0	0	349216	7
31	32	1	1	Spencer, Mrs. William Augustus (Marie Eugenie)	female	NaN	1	0	PC 17569	146
32	33	1	3	Glynn, Miss. Mary Agatha	female	NaN	0	0	335677	7
36	37	1	3	Mamee, Mr. Hanna	male	NaN	0	0	2677	7
42	43	0	3	Kraeff, Mr. Theodor	male	NaN	0	0	349253	7
45	46	0	3	Rogers, Mr. William John	male	NaN	0	0	S.C./A.4. 23567	8
46	47	0	3	Lennon, Mr. Denis	male	NaN	1	0	370371	15
47	48	1	3	O'Driscoll, Miss. Bridget	female	NaN	0	0	14311	7
48	49	0	3	Samaan, Mr. Youssef	male	NaN	2	0	2662	21
55	56	1	1	Woolner, Mr. Hugh	male	NaN	0	0	19947	35
64	65	0	1	Stewart, Mr. Albert A	male	NaN	0	0	PC 17605	27
65	66	1	3	Moubarek, Master. Gerios	male	NaN	1	1	2661	15
76	77	0	3	Staneff, Mr. Ivan	male	NaN	0	0	349208	7
77	78	0	3	Moutal, Mr. Rahamin Haim	male	NaN	0	0	374746	8
82	83	1	3	McDermott, Miss. Brigdet Delia	female	NaN	0	0	330932	7

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	
87	88	0	3	Slocovski, Mr. Selman Francis	male	NaN	0	0	SOTON/OQ 392086	8
95	96	0	3	Shorney, Mr. Charles Joseph	male	NaN	0	0	374910	8
101	102	0	3	Petroff, Mr. Pastcho ("Pentcho")	male	NaN	0	0	349215	7
107	108	1	3	Moss, Mr. Albert Johan	male	NaN	0	0	312991	7
109	110	1	3	Moran, Miss. Bertha	female	NaN	1	0	371110	24
121	122	0	3	Moore, Mr. Leonard Charles	male	NaN	0	0	A4. 54510	8
126	127	0	3	McMahon, Mr. Martin	male	NaN	0	0	370372	7
128	129	1	3	Peter, Miss. Anna	female	NaN	1	1	2668	22
140	141	0	3	Boulos, Mrs. Joseph (Sultana)	female	NaN	0	2	2678	15
154	155	0	3	Olsen, Mr. Ole Martin	male	NaN	0	0	Fa 265302	7
...	...	...	...	...	...	...	...	...	...	
718	719	0	3	McEvoy, Mr. Michael	male	NaN	0	0	36568	15
727	728	1	3	Mannion, Miss. Margareth	female	NaN	0	0	36866	7
732	733	0	2	Knight, Mr. Robert J	male	NaN	0	0	239855	0
738	739	0	3	Ivanoff, Mr. Kanio	male	NaN	0	0	349201	7
739	740	0	3	Nankoff, Mr. Minko	male	NaN	0	0	349218	7
740	741	1	1	Hawksford, Mr. Walter James	male	NaN	0	0	16988	30
760	761	0	3	Garfirth, Mr. John	male	NaN	0	0	358585	14
766	767	0	1	Brewe, Dr. Arthur Jackson	male	NaN	0	0	112379	39
768	769	0	3	Moran, Mr. Daniel J	male	NaN	1	0	371110	24
773	774	0	3	Elias, Mr. Dibo	male	NaN	0	0	2674	7
776	777	0	3	Tobin, Mr. Roger	male	NaN	0	0	383121	7

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	
778	779	0	3	Kilgannon, Mr. Thomas J	male	NaN	0	0	36865	7
783	784	0	3	Johnston, Mr. Andrew G	male	NaN	1	2	W./C. 6607	23
790	791	0	3	Keane, Mr. Andrew "Andy"	male	NaN	0	0	12460	7
792	793	0	3	Sage, Miss. Stella Anna	female	NaN	8	2	CA. 2343	69
793	794	0	1	Hoyt, Mr. William Fisher	male	NaN	0	0	PC 17600	30
815	816	0	1	Fry, Mr. Richard	male	NaN	0	0	112058	B
825	826	0	3	Flynn, Mr. John	male	NaN	0	0	368323	6
826	827	0	3	Lam, Mr. Len	male	NaN	0	0	1601	56
828	829	1	3	McCormack, Mr. Thomas Joseph	male	NaN	0	0	367228	7
832	833	0	3	Saad, Mr. Amin	male	NaN	0	0	2671	7
837	838	0	3	Sirota, Mr. Maurice	male	NaN	0	0	392092	8
839	840	1	1	Marechal, Mr. Pierre	male	NaN	0	0	11774	29
846	847	0	3	Sage, Mr. Douglas Bullen	male	NaN	8	2	CA. 2343	69
849	850	1	1	Goldenberg, Mrs. Samuel L (Edwiga Grabowska)	female	NaN	1	0	17453	89
859	860	0	3	Razi, Mr. Raihed	male	NaN	0	0	2629	7
863	864	0	3	Sage, Miss. Dorothy Edith "Dolly"	female	NaN	8	2	CA. 2343	69
868	869	0	3	van Melkebeke, Mr. Philemon	male	NaN	0	0	345777	9
878	879	0	3	Laleff, Mr. Kristo	male	NaN	0	0	349217	7
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23

177 rows × 12 columns

In [17]:

```
data_num[data_num.index.isin(missedValues)]['Age']
```

Out[17]:

```
5      NaN
17     NaN
19     NaN
26     NaN
28     NaN
29     NaN
31     NaN
32     NaN
36     NaN
42     NaN
45     NaN
46     NaN
47     NaN
48     NaN
55     NaN
64     NaN
65     NaN
76     NaN
77     NaN
82     NaN
87     NaN
95     NaN
101    NaN
107    NaN
109    NaN
121    NaN
126    NaN
128    NaN
140    NaN
154    NaN
       ..
718    NaN
727    NaN
732    NaN
738    NaN
739    NaN
740    NaN
760    NaN
766    NaN
768    NaN
773    NaN
776    NaN
778    NaN
783    NaN
790    NaN
792    NaN
793    NaN
815    NaN
825    NaN
826    NaN
828    NaN
832    NaN
837    NaN
839    NaN
846    NaN
849    NaN
859    NaN
863    NaN
868    NaN
```

```
878   NaN
888   NaN
Name: Age, Length: 177, dtype: float64
```

In [18]:

```
data_num_age=data_num[['Age']]
data_num_age.head()
```

Out[18]:

	Age
0	22.0
1	38.0
2	26.0
3	35.0
4	35.0

Фильтр для проверки заполнения пустых значений

In [19]:

```
indicator = MissingIndicator()
mask_missing_values_only = indicator.fit_transform(data_num_age)
mask_missing_values_only
```

Out[19]:

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

In [20]:

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

т.е. нашими стратегиями будут:

- Среднее значение
- Медиана
- Наиболее часто встречающаяся величина

Определим функцию для импьютации в которую будет отправляться название стратегии как аргумент

In [21]:

```
def test_num_impute(strat):
    # Определяем  сратегию
    imp_num = SimpleImputer(strategy=strat)
    data_num_imp = imp_num.fit_transform(data_num_age)
    return data_num_imp[mask_missing_values_only]
```

In [22]:

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

Out[22]:

```
('mean',
 array([29.69911765, 29.69911765, 29.69911765, 29.69911765, 29.69911
765,
        29.69911765, 29.69911765, 29.69911765, 29.69911765, 29.69911
765,
        29.69911765, 29.69911765, 29.69911765, 29.69911765, 29.69911
765,
        29.69911765, 29.69911765, 29.69911765, 29.69911765, 29.69911
765,
        29.69911765, 29.69911765, 29.69911765, 29.69911765, 29.69911
765,
        29.69911765, 29.69911765, 29.69911765, 29.69911765, 29.69911
765,
        29.69911765, 29.69911765, 29.69911765, 29.69911765, 29.69911
765,
        29.69911765, 29.69911765, 29.69911765, 29.69911765, 29.69911
765,
        29.69911765, 29.69911765, 29.69911765, 29.69911765, 29.69911
765,
        29.69911765, 29.69911765, 29.69911765, 29.69911765, 29.69911
765,
        29.69911765, 29.69911765, 29.69911765, 29.69911765, 29.69911
765,
        29.69911765, 29.69911765, 29.69911765, 29.69911765, 29.69911
765,
        29.69911765, 29.69911765, 29.69911765, 29.69911765, 29.69911
765,
        29.69911765, 29.69911765, 29.69911765, 29.69911765, 29.69911
765,
        29.69911765, 29.69911765, 29.69911765, 29.69911765, 29.69911
765,
        29.69911765, 29.69911765, 29.69911765, 29.69911765, 29.69911
765,
        29.69911765, 29.69911765, 29.69911765, 29.69911765, 29.69911
765,
        29.69911765, 29.69911765, 29.69911765, 29.69911765, 29.69911
765,
        29.69911765, 29.69911765, 29.69911765, 29.69911765, 29.69911
765,
        29.69911765, 29.69911765, 29.69911765, 29.69911765, 29.69911
765,
        29.69911765, 29.69911765, 29.69911765, 29.69911765, 29.69911
765,
        29.69911765, 29.69911765, 29.69911765, 29.69911765, 29.69911
765,
        29.69911765, 29.69911765, 29.69911765, 29.69911765, 29.69911
765,
        29.69911765, 29.69911765, 29.69911765, 29.69911765, 29.69911
765,
        29.69911765, 29.69911765, 29.69911765, 29.69911765, 29.69911
765,
        29.69911765, 29.69911765, 29.69911765, 29.69911765, 29.69911
765,
        29.69911765, 29.69911765, 29.69911765, 29.69911765, 29.69911
765,
        29.69911765, 29.69911765, 29.69911765, 29.69911765, 29.69911
765,
        29.69911765, 29.69911765, 29.69911765, 29.69911765, 29.69911
765,
        29.69911765, 29.69911765, 29.69911765, 29.69911765, 29.69911
765,
```

```
        29.69911765, 29.69911765, 29.69911765, 29.69911765, 29.69911
765,
        29.69911765, 29.69911765, 29.69911765, 29.69911765, 29.69911
765,
        29.69911765, 29.69911765, 29.69911765, 29.69911765, 29.69911
765,
        29.69911765, 29.69911765, 29.69911765, 29.69911765, 29.69911
765,
        29.69911765, 29.69911765, 29.69911765, 29.69911765, 29.69911
765,
        29.69911765, 29.69911765, 29.69911765, 29.69911765, 29.69911
765,
        29.69911765, 29.69911765]))
```

In [23]:

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

Out[23]:

```
('median',
 array([28., 28., 28., 28., 28., 28., 28., 28., 28., 28., 28., 28.,
28.,
        28., 28., 28., 28., 28., 28., 28., 28., 28., 28., 28., 28.,
28.,
        28., 28., 28., 28., 28., 28., 28., 28., 28., 28., 28., 28.,
28.,
        28., 28., 28., 28., 28., 28., 28., 28., 28., 28., 28., 28.,
28.,
        28., 28., 28., 28., 28., 28., 28., 28., 28., 28., 28., 28.,
28.,
        28., 28., 28., 28., 28., 28., 28., 28., 28., 28., 28., 28.,
28.,
        28., 28., 28., 28., 28., 28., 28., 28., 28., 28., 28., 28.,
28.,
        28., 28., 28., 28., 28., 28., 28., 28., 28., 28., 28., 28.,
28.,
        28., 28., 28., 28., 28., 28., 28., 28., 28., 28., 28., 28.,
28.,
        28., 28., 28., 28., 28., 28., 28., 28., 28., 28., 28., 28.,
28.,
        28., 28., 28., 28., 28., 28., 28., 28., 28., 28., 28., 28.,
28.,
        28., 28., 28., 28., 28., 28., 28., 28., 28., 28., 28., 28.,
28.,
        28., 28., 28., 28., 28., 28., 28., 28., 28., 28., 28., 28.,
28.,
        28., 28., 28., 28., 28., 28., 28., 28.]))
```

In [24]:

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

Out[24]:

```
('most_frequent',
 array([24., 24., 24., 24., 24., 24., 24., 24., 24., 24., 24., 24.,
24.,
        24., 24., 24., 24., 24., 24., 24., 24., 24., 24., 24., 24.,
24.,
        24., 24., 24., 24., 24., 24., 24., 24., 24., 24., 24., 24.,
24.,
        24., 24., 24., 24., 24., 24., 24., 24., 24., 24., 24., 24.,
24.,
        24., 24., 24., 24., 24., 24., 24., 24., 24., 24., 24., 24.,
24.,
        24., 24., 24., 24., 24., 24., 24., 24., 24., 24., 24., 24.,
24.,
        24., 24., 24., 24., 24., 24., 24., 24., 24., 24., 24., 24.,
24.,
        24., 24., 24., 24., 24., 24., 24., 24., 24., 24., 24., 24.,
24.,
        24., 24., 24., 24., 24., 24., 24., 24., 24., 24., 24., 24.,
24.,
        24., 24., 24., 24., 24., 24., 24., 24., 24., 24., 24., 24.,
24.,
        24., 24., 24., 24., 24., 24., 24., 24., 24., 24., 24., 24.,
24.,
        24., 24., 24., 24., 24., 24., 24., 24., 24., 24., 24., 24.,
24.,
        24., 24., 24., 24., 24., 24., 24., 24., 24., 24., 24., 24.,
24.,
        24., 24., 24., 24., 24., 24., 24., 24.]))
```

1.2.2. Обработка пропусков в категориальных данных

Выберем категорильные колонки с пропущенными значениями

In [25]:

```
cat_cols = [col for col in data.columns if (data[data[col].isnull()].shape[0] >
0 and data[col].dtype=='object')]
for col in cat_cols:
    print(f"Колонка {col}, количество пропусков {data[col].isnull().sum()} - {round
((data[col].isnull().sum()/total_count)*100,2)}%")
```

```
Колонка Cabin, количество пропусков 687 - 77.1%
Колонка Embarked, количество пропусков 2 - 0.22%
```

In [26]:

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

Out[26]:

	Cabin
0	NaN
1	C85
2	NaN
3	C123
4	NaN

Получим уникальныые значения для колонки

In [28]:

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

Out[28]:

```
array([nan, 'C85', 'C123', 'E46', 'G6', 'C103', 'D56', 'A6',
       'C23 C25 C27', 'B78', 'D33', 'B30', 'C52', 'B28', 'C83', 'F3
3',
       'F G73', 'E31', 'A5', 'D10 D12', 'D26', 'C110', 'B58 B60', 'E
101',
       'F E69', 'D47', 'B86', 'F2', 'C2', 'E33', 'B19', 'A7', 'C49',
'F4',
       'A32', 'B4', 'B80', 'A31', 'D36', 'D15', 'C93', 'C78', 'D35',
       'C87', 'B77', 'E67', 'B94', 'C125', 'C99', 'C118', 'D7', 'A1
9',
       'B49', 'D', 'C22 C26', 'C106', 'C65', 'E36', 'C54',
       'B57 B59 B63 B66', 'C7', 'E34', 'C32', 'B18', 'C124', 'C91',
'E40',
       'T', 'C128', 'D37', 'B35', 'E50', 'C82', 'B96 B98', 'E10', 'E
44',
       'A34', 'C104', 'C111', 'C92', 'E38', 'D21', 'E12', 'E63', 'A1
4',
       'B37', 'C30', 'D20', 'B79', 'E25', 'D46', 'B73', 'C95', 'B3
8',
       'B39', 'B22', 'C86', 'C70', 'A16', 'C101', 'C68', 'A10', 'E6
8',
       'B41', 'A20', 'D19', 'D50', 'D9', 'A23', 'B50', 'A26', 'D48',
       'E58', 'C126', 'B71', 'B51 B53 B55', 'D49', 'B5', 'B20', 'F G
63',
       'C62 C64', 'E24', 'C90', 'C45', 'E8', 'B101', 'D45', 'C46',
'D30',
       'E121', 'D11', 'E77', 'F38', 'B3', 'D6', 'B82 B84', 'D17', 'A
36',
       'B102', 'B69', 'E49', 'C47', 'D28', 'E17', 'A24', 'C50', 'B4
2',
       'C148'], dtype=object)
```

In [29]:

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

Out[29]:

```
(687, 1)
```

Импьютация наиболее частыми выражениями

In [30]:

```
imp2 = SimpleImputer(missing_values=np.nan, strategy='most_frequent')
data_imp2 = imp2.fit_transform(cat_temp_data)
data_imp2
```

Out[30]:

```
array([['B96 B98'],
       ['C85'],
       ['B96 B98'],
       ['C123'],
       ['B96 B98'],
       ['B96 B98'],
       ['E46'],
       ['B96 B98'],
       ['B96 B98'],
       ['B96 B98'],
       ['G6'],
       ['C103'],
       ['B96 B98'],
       ['B96 B98'],
       ['B96 B98'],
       ['B96 B98'],
       ['B96 B98'],
       ['B96 B98'],
       ['B96 B98'],
       ['B96 B98'],
       ['B96 B98'],
       ['D56'],
       ['B96 B98'],
       ['A6'],
       ['B96 B98'],
       ['B96 B98'],
       ['B96 B98'],
       ['C23 C25 C27'],
       ['B96 B98'],
       ['B96 B98'],
       ['B96 B98'],
       ['B78'],
       ['B96 B98'],
       ['B96 B98'],
       ['B96 B98'],
       ['B96 B98'],
       ['B96 B98'],
       ['B96 B98'],
       ['B96 B98'],
       ['B96 B98'],
       ['B96 B98'],
       ['B96 B98'],
       ['B96 B98'],
       ['B96 B98'],
       ['B96 B98'],
       ['B96 B98'],
       ['B96 B98'],
       ['B96 B98'],
       ['B96 B98'],
       ['B96 B98'],
       ['B96 B98'],
       ['B96 B98'],
       ['D33'],
       ['B96 B98'],
       ['B30'],
       ['C52'],
       ['B96 B98'],
       ['B96 B98'],
       ['B96 B98'],
```

```
['B96 B98'],
['B96 B98'],
['B28'],
['C83'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['F33'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['F G73'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['C23 C25 C27'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['E31'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['A5'],
['D10 D12'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['D26'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['C110'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B58 B60'],
['B96 B98'],
```

```
['B96 B98'],
['B96 B98'],
['B96 B98'],
['E101'],
['D26'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['F E69'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['D47'],
['C123'],
['B96 B98'],
['B86'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['F2'],
['B96 B98'],
['B96 B98'],
['C2'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['E33'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B19'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['A7'],
['B96 B98'],
['B96 B98'],
['C49'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
```

```
['B96 B98'],
['B96 B98'],
['F4'],
['B96 B98'],
['A32'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['F2'],
['B4'],
['B80'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['G6'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['A31'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['D36'],
['B96 B98'],
['B96 B98'],
['D15'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['C93'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['C83'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
```

```
['B96 B98'],
['B96 B98'],
['B96 B98'],
['C78'],
['B96 B98'],
['B96 B98'],
['D35'],
['B96 B98'],
['B96 B98'],
['G6'],
['C87'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B77'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['E67'],
['B94'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['C125'],
['C99'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['C118'],
['B96 B98'],
['D7'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['A19'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B49'],
['D'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['C22 C26'],
['C106'],
['B58 B60'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
```

```
['E101'],
['B96 B98'],
['C22 C26'],
['B96 B98'],
['C65'],
['B96 B98'],
['E36'],
['C54'],
['B57 B59 B63 B66'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['C7'],
['E34'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['C32'],
['B96 B98'],
['D'],
['B96 B98'],
['B18'],
['B96 B98'],
['C124'],
['C91'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['C2'],
['E40'],
['B96 B98'],
['T'],
['F2'],
['C23 C25 C27'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['F33'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['C128'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['E33'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
```

```
['B96 B98'],
['B96 B98'],
['D37'],
['B96 B98'],
['B96 B98'],
['B35'],
['E50'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['C82'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['D36'],
['G6'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['C78'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
```

```
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['E10'],
['C52'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['E44'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['C23 C25 C27'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['A34'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['C104'],
['B96 B98'],
['B96 B98'],
['C111'],
['C92'],
['B96 B98'],
['B96 B98'],
['E38'],
['D21'],
['B96 B98'],
['B96 B98'],
['E12'],
['B96 B98'],
['E63'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['D'],
['B96 B98'],
['A14'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B49'],
['B96 B98'],
```

```
['C93'],
['B37'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['C30'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['D20'],
['B96 B98'],
['C22 C26'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B79'],
['C65'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['E25'],
['B96 B98'],
['B96 B98'],
['D46'],
['F33'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B73'],
['B96 B98'],
['B96 B98'],
['B18'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['C95'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B38'],
['B96 B98'],
['B96 B98'],
['B39'],
['B22'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['C86'],
['B96 B98'],
['B96 B98'],
```

```
['B96 B98'],
['B96 B98'],
['B96 B98'],
['C70'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['A16'],
['B96 B98'],
['E67'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['C101'],
['E25'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['E44'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['C68'],
['B96 B98'],
['A10'],
['B96 B98'],
['E68'],
['B96 B98'],
['B41'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['D20'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['A20'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
```

```
['B96 B98'],
['C125'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['F4'],
['B96 B98'],
['B96 B98'],
['D19'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['D50'],
['B96 B98'],
['D9'],
['B96 B98'],
['B96 B98'],
['A23'],
['B96 B98'],
['B50'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B35'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['D33'],
['B96 B98'],
['A26'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['D48'],
['B96 B98'],
['B96 B98'],
['E58'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
```

```
['C126'],
['B96 B98'],
['B71'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B51 B53 B55'],
['B96 B98'],
['D49'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B5'],
['B20'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['C68'],
['F G63'],
['C62 C64'],
['E24'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['E24'],
['B96 B98'],
['B96 B98'],
['C90'],
['C124'],
['C126'],
['B96 B98'],
['B96 B98'],
['F G73'],
['C45'],
['E101'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['E8'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
```

```
['B5'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B101'],
['B96 B98'],
['B96 B98'],
['D45'],
['C46'],
['B57 B59 B63 B66'],
['B96 B98'],
['B96 B98'],
['B22'],
['B96 B98'],
['B96 B98'],
['D30'],
['B96 B98'],
['B96 B98'],
['E121'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B77'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['D11'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['E77'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['F38'],
['B96 B98'],
['B96 B98'],
['B3'],
['B96 B98'],
['B20'],
['D6'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B82 B84'],
['B96 B98'],
```

```
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['D17'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['A36'],
['B96 B98'],
['B96 B98'],
['E8'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B102'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B69'],
['B96 B98'],
['B96 B98'],
['E121'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B28'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['E49'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['C47'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['C92'],
['B96 B98'],
['B96 B98'],
```

```
['B96 B98'],
['D28'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['E17'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['D17'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['A24'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['D35'],
['B51 B53 B55'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['C50'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B96 B98'],
['B42'],
['B96 B98'],
['C148'],
['B96 B98']], dtype=object)
```

Проверим, что пустые значения отсутствуют

In [31]:

```
np.unique(data_imp2)
```

Out[31]:

```
array(['A10', 'A14', 'A16', 'A19', 'A20', 'A23', 'A24', 'A26', 'A3
1',
       'A32', 'A34', 'A36', 'A5', 'A6', 'A7', 'B101', 'B102', 'B18',
       'B19', 'B20', 'B22', 'B28', 'B3', 'B30', 'B35', 'B37', 'B38',
       'B39', 'B4', 'B41', 'B42', 'B49', 'B5', 'B50', 'B51 B53 B55',
       'B57 B59 B63 B66', 'B58 B60', 'B69', 'B71', 'B73', 'B77', 'B7
8',
       'B79', 'B80', 'B82 B84', 'B86', 'B94', 'B96 B98', 'C101', 'C1
03',
       'C104', 'C106', 'C110', 'C111', 'C118', 'C123', 'C124', 'C12
5',
       'C126', 'C128', 'C148', 'C2', 'C22 C26', 'C23 C25 C27', 'C3
0',
       'C32', 'C45', 'C46', 'C47', 'C49', 'C50', 'C52', 'C54', 'C62
C64',
       'C65', 'C68', 'C7', 'C70', 'C78', 'C82', 'C83', 'C85', 'C86',
       'C87', 'C90', 'C91', 'C92', 'C93', 'C95', 'C99', 'D', 'D10 D1
2',
       'D11', 'D15', 'D17', 'D19', 'D20', 'D21', 'D26', 'D28', 'D3
0',
       'D33', 'D35', 'D36', 'D37', 'D45', 'D46', 'D47', 'D48', 'D4
9',
       'D50', 'D56', 'D6', 'D7', 'D9', 'E10', 'E101', 'E12', 'E121',
       'E17', 'E24', 'E25', 'E31', 'E33', 'E34', 'E36', 'E38', 'E4
0',
       'E44', 'E46', 'E49', 'E50', 'E58', 'E63', 'E67', 'E68', 'E7
7',
       'E8', 'F E69', 'F G63', 'F G73', 'F2', 'F33', 'F38', 'F4', 'G
6',
       'T'], dtype=object)
```

Импьютация константой

In [32]:

```
imp3 = SimpleImputer(missing_values=np.nan, strategy='constant', fill_value='MyC
onst')
data_imp3 = imp3.fit_transform(cat_temp_data)
data_imp3
```

Out[32]:

```
array([['MyConst'],
       ['C85'],
       ['MyConst'],
       ['C123'],
       ['MyConst'],
       ['MyConst'],
       ['E46'],
       ['MyConst'],
       ['MyConst'],
       ['MyConst'],
       ['G6'],
       ['C103'],
       ['MyConst'],
       ['MyConst'],
       ['MyConst'],
       ['MyConst'],
       ['MyConst'],
       ['MyConst'],
       ['MyConst'],
       ['MyConst'],
       ['MyConst'],
       ['D56'],
       ['MyConst'],
       ['A6'],
       ['MyConst'],
       ['MyConst'],
       ['MyConst'],
       ['C23 C25 C27'],
       ['MyConst'],
       ['MyConst'],
       ['MyConst'],
       ['B78'],
       ['MyConst'],
       ['MyConst'],
       ['MyConst'],
       ['MyConst'],
       ['MyConst'],
       ['MyConst'],
       ['MyConst'],
       ['MyConst'],
       ['MyConst'],
       ['MyConst'],
       ['MyConst'],
       ['MyConst'],
       ['MyConst'],
       ['MyConst'],
       ['MyConst'],
       ['MyConst'],
       ['MyConst'],
       ['MyConst'],
       ['MyConst'],
       ['D33'],
       ['MyConst'],
       ['B30'],
       ['C52'],
       ['MyConst'],
       ['MyConst'],
       ['MyConst'],
```

```
['MyConst'],
['MyConst'],
['B28'],
['C83'],
['MyConst'],
['MyConst'],
['MyConst'],
['F33'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['F G73'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['C23 C25 C27'],
['MyConst'],
['MyConst'],
['MyConst'],
['E31'],
['MyConst'],
['MyConst'],
['MyConst'],
['A5'],
['D10 D12'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['D26'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['C110'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['B58 B60'],
['MyConst'],
```

```
['MyConst'],
['MyConst'],
['MyConst'],
['E101'],
['D26'],
['MyConst'],
['MyConst'],
['MyConst'],
['F E69'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['D47'],
['C123'],
['MyConst'],
['B86'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['F2'],
['MyConst'],
['MyConst'],
['C2'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['E33'],
['MyConst'],
['MyConst'],
['MyConst'],
['B19'],
['MyConst'],
['MyConst'],
['MyConst'],
['A7'],
['MyConst'],
['MyConst'],
['C49'],
['MyConst'],
['MyConst'],
['MyConst'],
```

```
['MyConst'],
['MyConst'],
['F4'],
['MyConst'],
['A32'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['F2'],
['B4'],
['B80'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['G6'],
['MyConst'],
['MyConst'],
['MyConst'],
['A31'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['D36'],
['MyConst'],
['MyConst'],
['D15'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['C93'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['C83'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
```

```
['MyConst'],
['MyConst'],
['MyConst'],
['C78'],
['MyConst'],
['MyConst'],
['D35'],
['MyConst'],
['MyConst'],
['G6'],
['C87'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['B77'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['E67'],
['B94'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['C125'],
['C99'],
['MyConst'],
['MyConst'],
['MyConst'],
['C118'],
['MyConst'],
['D7'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['A19'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['B49'],
['D'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['C22 C26'],
['C106'],
['B58 B60'],
['MyConst'],
['MyConst'],
['MyConst'],
```

```
['E101'],
['MyConst'],
['C22 C26'],
['MyConst'],
['C65'],
['MyConst'],
['E36'],
['C54'],
['B57 B59 B63 B66'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['C7'],
['E34'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['C32'],
['MyConst'],
['D'],
['MyConst'],
['B18'],
['MyConst'],
['C124'],
['C91'],
['MyConst'],
['MyConst'],
['MyConst'],
['C2'],
['E40'],
['MyConst'],
['T'],
['F2'],
['C23 C25 C27'],
['MyConst'],
['MyConst'],
['MyConst'],
['F33'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['C128'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['E33'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
```

```
['MyConst'],
['MyConst'],
['D37'],
['MyConst'],
['MyConst'],
['B35'],
['E50'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['C82'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['B96 B98'],
['MyConst'],
['MyConst'],
['D36'],
['G6'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['C78'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
```

```
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['E10'],
['C52'],
['MyConst'],
['MyConst'],
['MyConst'],
['E44'],
['B96 B98'],
['MyConst'],
['MyConst'],
['C23 C25 C27'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['A34'],
['MyConst'],
['MyConst'],
['MyConst'],
['C104'],
['MyConst'],
['MyConst'],
['C111'],
['C92'],
['MyConst'],
['MyConst'],
['E38'],
['D21'],
['MyConst'],
['MyConst'],
['E12'],
['MyConst'],
['E63'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['D'],
['MyConst'],
['A14'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['B49'],
['MyConst'],
```

```
['C93'],
['B37'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['C30'],
['MyConst'],
['MyConst'],
['MyConst'],
['D20'],
['MyConst'],
['C22 C26'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['B79'],
['C65'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['E25'],
['MyConst'],
['MyConst'],
['D46'],
['F33'],
['MyConst'],
['MyConst'],
['MyConst'],
['B73'],
['MyConst'],
['MyConst'],
['B18'],
['MyConst'],
['MyConst'],
['MyConst'],
['C95'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['B38'],
['MyConst'],
['MyConst'],
['B39'],
['B22'],
['MyConst'],
['MyConst'],
['MyConst'],
['C86'],
['MyConst'],
['MyConst'],
```

```
['MyConst'],
['MyConst'],
['MyConst'],
['C70'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['A16'],
['MyConst'],
['E67'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['C101'],
['E25'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['E44'],
['MyConst'],
['MyConst'],
['MyConst'],
['C68'],
['MyConst'],
['A10'],
['MyConst'],
['E68'],
['MyConst'],
['B41'],
['MyConst'],
['MyConst'],
['MyConst'],
['D20'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['A20'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
```

```
['MyConst'],
['C125'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['F4'],
['MyConst'],
['MyConst'],
['D19'],
['MyConst'],
['MyConst'],
['MyConst'],
['D50'],
['MyConst'],
['D9'],
['MyConst'],
['MyConst'],
['A23'],
['MyConst'],
['B50'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['B35'],
['MyConst'],
['MyConst'],
['MyConst'],
['D33'],
['MyConst'],
['A26'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['D48'],
['MyConst'],
['MyConst'],
['E58'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
```

```
['C126'],
['MyConst'],
['B71'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['B51 B53 B55'],
['MyConst'],
['D49'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['B5'],
['B20'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['C68'],
['F G63'],
['C62 C64'],
['E24'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['E24'],
['MyConst'],
['MyConst'],
['C90'],
['C124'],
['C126'],
['MyConst'],
['MyConst'],
['F G73'],
['C45'],
['E101'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['E8'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
```

```
['B5'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['B101'],
['MyConst'],
['MyConst'],
['D45'],
['C46'],
['B57 B59 B63 B66'],
['MyConst'],
['MyConst'],
['B22'],
['MyConst'],
['MyConst'],
['D30'],
['MyConst'],
['MyConst'],
['E121'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['B77'],
['MyConst'],
['MyConst'],
['MyConst'],
['B96 B98'],
['MyConst'],
['D11'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['E77'],
['MyConst'],
['MyConst'],
['MyConst'],
['F38'],
['MyConst'],
['MyConst'],
['B3'],
['MyConst'],
['B20'],
['D6'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['B82 B84'],
['MyConst'],
```

```
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['D17'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['B96 B98'],
['MyConst'],
['MyConst'],
['MyConst'],
['A36'],
['MyConst'],
['MyConst'],
['E8'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['B102'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['B69'],
['MyConst'],
['MyConst'],
['E121'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['B28'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['E49'],
['MyConst'],
['MyConst'],
['MyConst'],
['C47'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['C92'],
['MyConst'],
['MyConst'],
```

```
['MyConst'],
['D28'],
['MyConst'],
['MyConst'],
['MyConst'],
['E17'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['D17'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['A24'],
['MyConst'],
['MyConst'],
['MyConst'],
['D35'],
['B51 B53 B55'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['C50'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['MyConst'],
['B42'],
['MyConst'],
['C148'],
['MyConst']], dtype=object)
```

In [33]:

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

Out[33]:

```
array(['A10', 'A14', 'A16', 'A19', 'A20', 'A23', 'A24', 'A26', 'A3
1',
       'A32', 'A34', 'A36', 'A5', 'A6', 'A7', 'B101', 'B102', 'B18',
       'B19', 'B20', 'B22', 'B28', 'B3', 'B30', 'B35', 'B37', 'B38',
       'B39', 'B4', 'B41', 'B42', 'B49', 'B5', 'B50', 'B51 B53 B55',
       'B57 B59 B63 B66', 'B58 B60', 'B69', 'B71', 'B73', 'B77', 'B7
8',
       'B79', 'B80', 'B82 B84', 'B86', 'B94', 'B96 B98', 'C101', 'C1
03',
       'C104', 'C106', 'C110', 'C111', 'C118', 'C123', 'C124', 'C12
5',
       'C126', 'C128', 'C148', 'C2', 'C22 C26', 'C23 C25 C27', 'C3
0',
       'C32', 'C45', 'C46', 'C47', 'C49', 'C50', 'C52', 'C54', 'C62
C64',
       'C65', 'C68', 'C7', 'C70', 'C78', 'C82', 'C83', 'C85', 'C86',
       'C87', 'C90', 'C91', 'C92', 'C93', 'C95', 'C99', 'D', 'D10 D1
2',
       'D11', 'D15', 'D17', 'D19', 'D20', 'D21', 'D26', 'D28', 'D3
0',
       'D33', 'D35', 'D36', 'D37', 'D45', 'D46', 'D47', 'D48', 'D4
9',
       'D50', 'D56', 'D6', 'D7', 'D9', 'E10', 'E101', 'E12', 'E121',
       'E17', 'E24', 'E25', 'E31', 'E33', 'E34', 'E36', 'E38', 'E4
0',
       'E44', 'E46', 'E49', 'E50', 'E58', 'E63', 'E67', 'E68', 'E7
7',
       'E8', 'F E69', 'F G63', 'F G73', 'F2', 'F33', 'F38', 'F4', 'G
6',
       'MyConst', 'T'], dtype=object)
```

In [34]:

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

Out[34]:

```
687
```

2. Форматирование категориальных признаков в числовые

Рассмотрим набор, в котором мы заменили пропущенные значения на саме частое значение

In [35]:

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

Out[35]:

	Cabin
0	B96 B98
1	C85
2	B96 B98
3	C123
4	B96 B98
5	B96 B98
6	E46
7	B96 B98
8	B96 B98
9	B96 B98
10	G6
11	C103
12	B96 B98
13	B96 B98
14	B96 B98
15	B96 B98
16	B96 B98
17	B96 B98
18	B96 B98
19	B96 B98
20	B96 B98
21	D56
22	B96 B98
23	A6
24	B96 B98
25	B96 B98
26	B96 B98
27	C23 C25 C27
28	B96 B98
29	B96 B98
...	...
861	B96 B98
862	D17
863	B96 B98
864	B96 B98
865	B96 B98

	Cabin
866	B96 B98
867	A24
868	B96 B98
869	B96 B98
870	B96 B98
871	D35
872	B51 B53 B55
873	B96 B98
874	B96 B98
875	B96 B98
876	B96 B98
877	B96 B98
878	B96 B98
879	C50
880	B96 B98
881	B96 B98
882	B96 B98
883	B96 B98
884	B96 B98
885	B96 B98
886	B96 B98
887	B42
888	B96 B98
889	C148
890	B96 B98

891 rows × 1 columns

2.1 Label encoding - кодирование целыми значениями

In [36]:

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

In [37]:

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

Out[37]:

```
array(['B96 B98', 'C85', 'C123', 'E46', 'G6', 'C103', 'D56', 'A6',
       'C23 C25 C27', 'B78', 'D33', 'B30', 'C52', 'B28', 'C83', 'F3
3',
       'F G73', 'E31', 'A5', 'D10 D12', 'D26', 'C110', 'B58 B60', 'E
101',
       'F E69', 'D47', 'B86', 'F2', 'C2', 'E33', 'B19', 'A7', 'C49',
'F4',
       'A32', 'B4', 'B80', 'A31', 'D36', 'D15', 'C93', 'C78', 'D35',
       'C87', 'B77', 'E67', 'B94', 'C125', 'C99', 'C118', 'D7', 'A1
9',
       'B49', 'D', 'C22 C26', 'C106', 'C65', 'E36', 'C54',
       'B57 B59 B63 B66', 'C7', 'E34', 'C32', 'B18', 'C124', 'C91',
'E40',
       'T', 'C128', 'D37', 'B35', 'E50', 'C82', 'E10', 'E44', 'A34',
       'C104', 'C111', 'C92', 'E38', 'D21', 'E12', 'E63', 'A14', 'B3
7',
       'C30', 'D20', 'B79', 'E25', 'D46', 'B73', 'C95', 'B38', 'B3
9',
       'B22', 'C86', 'C70', 'A16', 'C101', 'C68', 'A10', 'E68', 'B4
1',
       'A20', 'D19', 'D50', 'D9', 'A23', 'B50', 'A26', 'D48', 'E58',
       'C126', 'B71', 'B51 B53 B55', 'D49', 'B5', 'B20', 'F G63',
       'C62 C64', 'E24', 'C90', 'C45', 'E8', 'B101', 'D45', 'C46',
'D30',
       'E121', 'D11', 'E77', 'F38', 'B3', 'D6', 'B82 B84', 'D17', 'A
36',
       'B102', 'B69', 'E49', 'C47', 'D28', 'E17', 'A24', 'C50', 'B4
2',
       'C148'], dtype=object)
```

In [38]:

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

Out[38]:

```
array([  0,   1,   2,   3,   4,   5,   6,   7,   8,   9,  10,  11,
12,
        13,  14,  15,  16,  17,  18,  19,  20,  21,  22,  23,  24,
25,
        26,  27,  28,  29,  30,  31,  32,  33,  34,  35,  36,  37,
38,
        39,  40,  41,  42,  43,  44,  45,  46,  47,  48,  49,  50,
51,
        52,  53,  54,  55,  56,  57,  58,  59,  60,  61,  62,  63,
64,
        65,  66,  67,  68,  69,  70,  71,  72,  73,  74,  75,  76,
77,
        78,  79,  80,  81,  82,  83,  84,  85,  86,  87,  88,  89,
90,
        91,  92,  93,  94,  95,  96,  97,  98,  99, 100, 101, 102, 1
03,
       104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 1
16,
       117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 1
29,
       130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 1
42,
       143, 144, 145, 146])
```

In [39]:

```
le.inverse_transform([  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])
```

Out[39]:

```
array(['A10', 'A14', 'A16', 'A19', 'A20', 'A23', 'A24', 'A26', 'A3
1',
       'A32', 'A34', 'A36', 'A5', 'A6', 'A7', 'B101', 'B102', 'B18',
       'B19', 'B20', 'B22', 'B28', 'B3', 'B30', 'B35', 'B37', 'B38',
       'B39', 'B4', 'B41', 'B42', 'B49', 'B5', 'B50', 'B51 B53 B55',
       'B57 B59 B63 B66', 'B58 B60', 'B69', 'B71', 'B73', 'B77', 'B7
8',
       'B79', 'B80', 'B82 B84', 'B86', 'B94', 'B96 B98', 'C101', 'C1
03',
       'C104', 'C106', 'C110', 'C111', 'C118', 'C123', 'C124', 'C12
5',
       'C126', 'C128', 'C148', 'C2', 'C22 C26', 'C23 C25 C27', 'C3
0',
       'C32', 'C45', 'C46', 'C47', 'C49', 'C50', 'C52', 'C54', 'C62
C64',
       'C65', 'C68', 'C7', 'C70', 'C78', 'C82', 'C83', 'C85', 'C86',
       'C87', 'C90', 'C91', 'C92', 'C93', 'C95', 'C99', 'D', 'D10 D1
2',
       'D11', 'D15', 'D17', 'D19', 'D20', 'D21', 'D26', 'D28', 'D3
0',
       'D33', 'D35', 'D36', 'D37', 'D45', 'D46', 'D47', 'D48', 'D4
9',
       'D50', 'D56', 'D6', 'D7', 'D9', 'E10', 'E101', 'E12', 'E121',
       'E17', 'E24', 'E25', 'E31', 'E33', 'E34', 'E36', 'E38', 'E4
0',
       'E44', 'E46', 'E49', 'E50', 'E58', 'E63', 'E67', 'E68', 'E7
7',
       'E8', 'F E69', 'F G63', 'F G73', 'F2', 'F33', 'F38', 'F4', 'G
6',
       'T'], dtype=object)
```

2.2. One-hot encoding - Кодирование наборами бинарных значений

In [40]:

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

In [41]:

```
cat_enc.shape
```

Out[41]:

```
(891, 1)
```

In [42]:

```
cat_enc_ohe.shape
```

Out[42]:

```
(891, 147)
```

In [43]:

```
cat_enc_ohe
```

Out[43]:

```
<891x147 sparse matrix of type '<class 'numpy.float64'>'
        with 891 stored elements in Compressed Sparse Row format>
```

In [44]:

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

Out[44]:

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

In [45]:

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

Out[45]:

	Cabin
0	B96 B98
1	C85
2	B96 B98
3	C123
4	B96 B98
5	B96 B98
6	E46
7	B96 B98
8	B96 B98
9	B96 B98

Pandas get_dummies - быстрый вариант one-hot кодирования

In [46]:

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

Out[46]:

	Cabin_A10	Cabin_A14	Cabin_A16	Cabin_A19	Cabin_A20	Cabin_A23	Cabin_A24	Cabin_A
0	0	0	0	0	0	0	0	
1	0	0	0	0	0	0	0	
2	0	0	0	0	0	0	0	
3	0	0	0	0	0	0	0	
4	0	0	0	0	0	0	0	

5 rows × 147 columns

dummy_na - параметр, который создает отдельный столбец для NaNов, если False, то игнорирует NaNы

In [47]:

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

### 3. Масштабированние данных
```

Out[47]:

	Cabin_A10	Cabin_A14	Cabin_A16	Cabin_A19	Cabin_A20	Cabin_A23	Cabin_A24	Cabin_A
0	0	0	0	0	0	0	0	
1	0	0	0	0	0	0	0	
2	0	0	0	0	0	0	0	
3	0	0	0	0	0	0	0	
4	0	0	0	0	0	0	0	

5 rows × 148 columns

3. Заполним пропуски и закодируем пропуски в нашей выборке

In [48]:

```
data[data['Age'].isnull()]
```

Out[48]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	
5	6	0	3	Moran, Mr. James	male	NaN	0	0	330877	8
17	18	1	2	Williams, Mr. Charles Eugene	male	NaN	0	0	244373	13
19	20	1	3	Masselmani, Mrs. Fatima	female	NaN	0	0	2649	7
26	27	0	3	Emir, Mr. Farred Chehab	male	NaN	0	0	2631	7
28	29	1	3	O'Dwyer, Miss. Ellen "Nellie"	female	NaN	0	0	330959	7
29	30	0	3	Todoroff, Mr. Lalio	male	NaN	0	0	349216	7
31	32	1	1	Spencer, Mrs. William Augustus (Marie Eugenie)	female	NaN	1	0	PC 17569	146
32	33	1	3	Glynn, Miss. Mary Agatha	female	NaN	0	0	335677	7
36	37	1	3	Mamee, Mr. Hanna	male	NaN	0	0	2677	7
42	43	0	3	Kraeff, Mr. Theodor	male	NaN	0	0	349253	7
45	46	0	3	Rogers, Mr. William John	male	NaN	0	0	S.C./A.4. 23567	8
46	47	0	3	Lennon, Mr. Denis	male	NaN	1	0	370371	15
47	48	1	3	O'Driscoll, Miss. Bridget	female	NaN	0	0	14311	7
48	49	0	3	Samaan, Mr. Youssef	male	NaN	2	0	2662	21
55	56	1	1	Woolner, Mr. Hugh	male	NaN	0	0	19947	35
64	65	0	1	Stewart, Mr. Albert A	male	NaN	0	0	PC 17605	27
65	66	1	3	Moubarek, Master. Gerios	male	NaN	1	1	2661	15
76	77	0	3	Staneff, Mr. Ivan	male	NaN	0	0	349208	7
77	78	0	3	Moutal, Mr. Rahamin Haim	male	NaN	0	0	374746	8
82	83	1	3	McDermott, Miss. Brigdet Delia	female	NaN	0	0	330932	7

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	
87	88	0	3	Slocovski, Mr. Selman Francis	male	NaN	0	0	SOTON/OQ 392086	8
95	96	0	3	Shorney, Mr. Charles Joseph	male	NaN	0	0	374910	8
101	102	0	3	Petroff, Mr. Pastcho ("Pentcho")	male	NaN	0	0	349215	7
107	108	1	3	Moss, Mr. Albert Johan	male	NaN	0	0	312991	7
109	110	1	3	Moran, Miss. Bertha	female	NaN	1	0	371110	24
121	122	0	3	Moore, Mr. Leonard Charles	male	NaN	0	0	A4. 54510	8
126	127	0	3	McMahon, Mr. Martin	male	NaN	0	0	370372	7
128	129	1	3	Peter, Miss. Anna	female	NaN	1	1	2668	22
140	141	0	3	Boulos, Mrs. Joseph (Sultana)	female	NaN	0	2	2678	15
154	155	0	3	Olsen, Mr. Ole Martin	male	NaN	0	0	Fa 265302	7
...	...	...	...	...	...	...	...	...	...	
718	719	0	3	McEvoy, Mr. Michael	male	NaN	0	0	36568	15
727	728	1	3	Mannion, Miss. Margareth	female	NaN	0	0	36866	7
732	733	0	2	Knight, Mr. Robert J	male	NaN	0	0	239855	0
738	739	0	3	Ivanoff, Mr. Kanio	male	NaN	0	0	349201	7
739	740	0	3	Nankoff, Mr. Minko	male	NaN	0	0	349218	7
740	741	1	1	Hawksford, Mr. Walter James	male	NaN	0	0	16988	30
760	761	0	3	Garfirth, Mr. John	male	NaN	0	0	358585	14
766	767	0	1	Brewe, Dr. Arthur Jackson	male	NaN	0	0	112379	39
768	769	0	3	Moran, Mr. Daniel J	male	NaN	1	0	371110	24
773	774	0	3	Elias, Mr. Dibo	male	NaN	0	0	2674	7
776	777	0	3	Tobin, Mr. Roger	male	NaN	0	0	383121	7

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	
778	779	0	3	Kilgannon, Mr. Thomas J	male	NaN	0	0	36865	7
783	784	0	3	Johnston, Mr. Andrew G	male	NaN	1	2	W./C. 6607	23
790	791	0	3	Keane, Mr. Andrew "Andy"	male	NaN	0	0	12460	7
792	793	0	3	Sage, Miss. Stella Anna	female	NaN	8	2	CA. 2343	69
793	794	0	1	Hoyt, Mr. William Fisher	male	NaN	0	0	PC 17600	30
815	816	0	1	Fry, Mr. Richard	male	NaN	0	0	112058	B
825	826	0	3	Flynn, Mr. John	male	NaN	0	0	368323	6
826	827	0	3	Lam, Mr. Len	male	NaN	0	0	1601	56
828	829	1	3	McCormack, Mr. Thomas Joseph	male	NaN	0	0	367228	7
832	833	0	3	Saad, Mr. Amin	male	NaN	0	0	2671	7
837	838	0	3	Sirota, Mr. Maurice	male	NaN	0	0	392092	8
839	840	1	1	Marechal, Mr. Pierre	male	NaN	0	0	11774	29
846	847	0	3	Sage, Mr. Douglas Bullen	male	NaN	8	2	CA. 2343	69
849	850	1	1	Goldenberg, Mrs. Samuel L (Edwiga Grabowska)	female	NaN	1	0	17453	89
859	860	0	3	Razi, Mr. Raihed	male	NaN	0	0	2629	7
863	864	0	3	Sage, Miss. Dorothy Edith "Dolly"	female	NaN	8	2	CA. 2343	69
868	869	0	3	van Melkebeke, Mr. Philemon	male	NaN	0	0	345777	9
878	879	0	3	Laleff, Mr. Kristo	male	NaN	0	0	349217	7
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23

177 rows × 12 columns

In [49]:

```
imp_num = SimpleImputer(strategy='mean')
data['Age'] = imp_num.fit_transform(data_num_age)
```

In [50]:

```
data['Age'].head(10)
```

Out[50]:

```
0    22.000000
1    38.000000
2    26.000000
3    35.000000
4    35.000000
5    29.699118
6    54.000000
7     2.000000
8    27.000000
9    14.000000
Name: Age, dtype: float64
```

In [51]:

```
data[data['Age'].isnull()]
```

Out[51]:

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Emba

4. Масштабирование

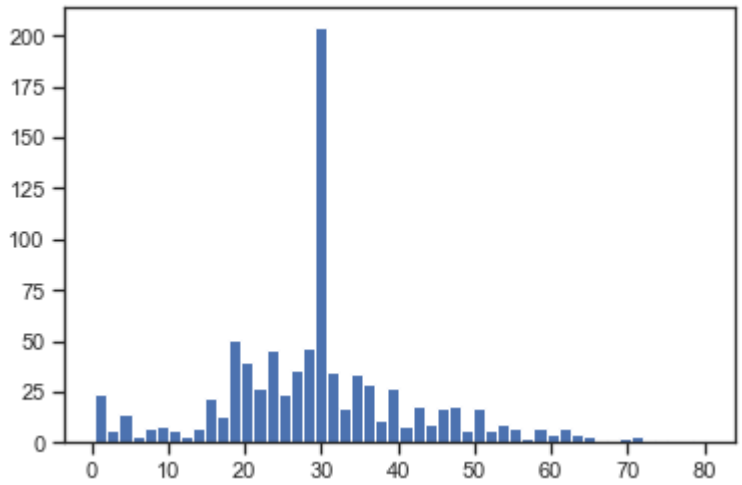
4.1. MinMax - масштабирование

In [52]:

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

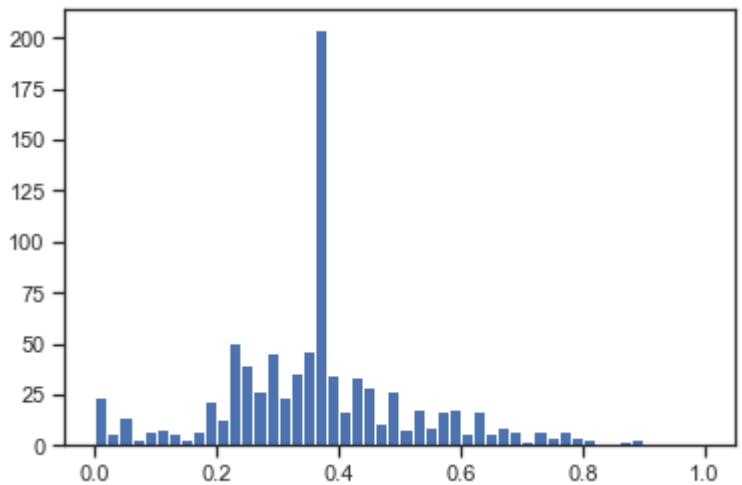
In [53]:

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



In [54]:

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



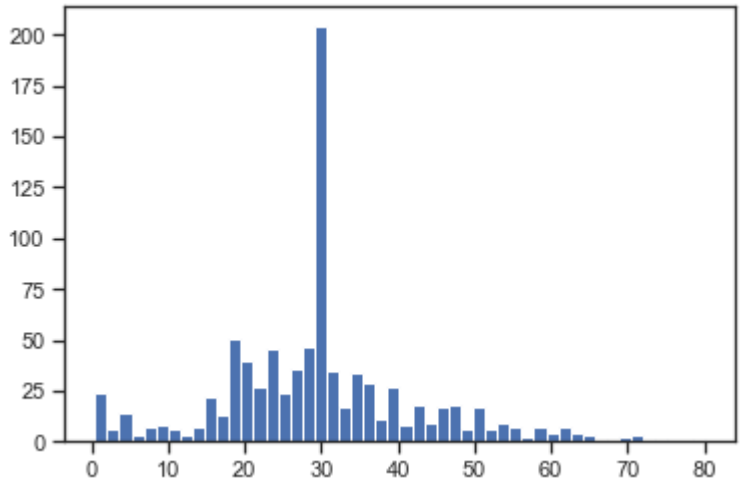
4.2. Z-оценка - StandartScaling

In [60]:

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

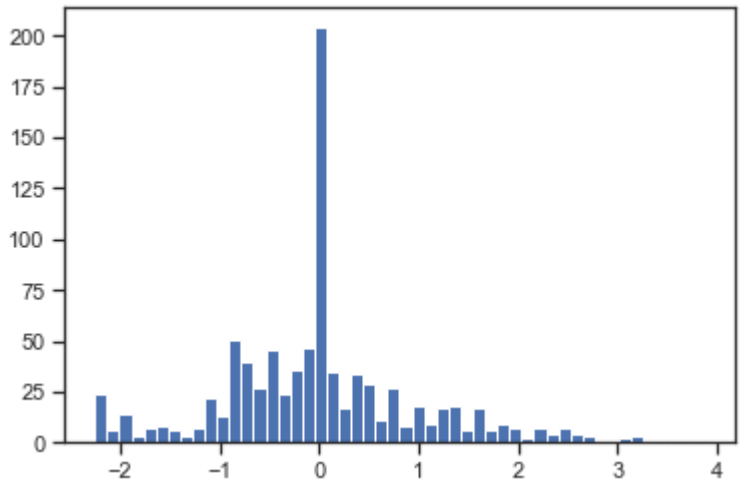
In [61]:

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



In [62]:

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



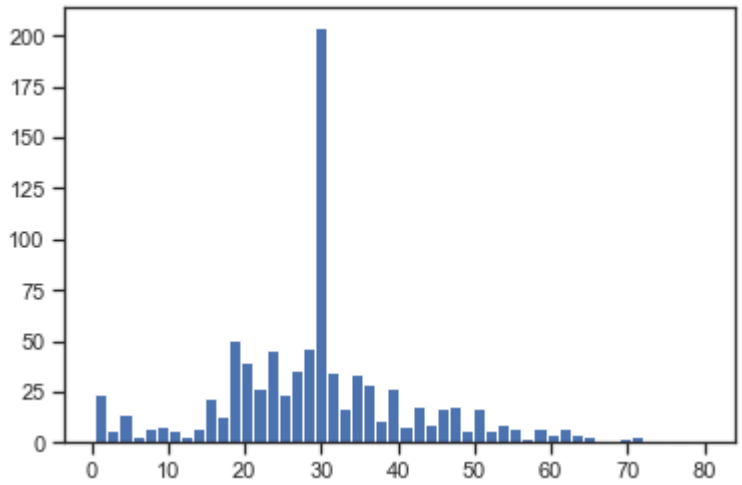
5. Нормализация данных

In [63]:

```
sc3 = Normalizer()
sc3_data = sc3.fit_transform(data[['Age']])
```

In [65]:

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



In [68]:

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