МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
им. Н.Э. Баумана

Кафедра «Систем обработки информации и управления»

ОТЧЕТ

Лабораторная работа №3
по курсу «Методы машинного обучения»

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

ИСПОЛНИТЕЛЬ: Семенова Е. В.

группа ИУ5-23М ______________
подпись

"__"_______2019 г.

Москва - 2019

Задание:

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

1) обработку пропусков в данных;

2) кодирование категориальных признаков;

3) масштабирование данных.

In [1]:

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

In [2]:

```
data = pd.read_csv('datasets/titanic.csv', sep=",")
```

In [3]:

```
# размер набора данных
data.shape
```

Out[3]:

```
(418, 11)
```

In [4]:

```
# типы колонок
data.dtypes
```

Out[4]:

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

In [5]:

```
# пропущенные значения
data.isnull().sum()
```

Out[5]:

```
PassengerId      0
Pclass           0
Name             0
Sex              0
Age             86
SibSp            0
Parch            0
Ticket           0
Fare             1
Cabin          327
Embarked         0
dtype: int64
```

In [6]:

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

Out[6]:

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	E
0	892	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	NaN	
1	893	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	NaN	
2	894	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	NaN	
3	895	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	NaN	
4	896	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	NaN	

In [7]:

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

```
Всего строк: 418
```

Обработка пропусков в данных

Удаление

In [8]:

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

Out[8]:

```
((418, 11), (418, 8))
```

In [9]:

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

Out[9]:

```
((418, 11), (87, 11))
```

In [10]:

```
data.head()
```

Out[10]:

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	E
0	892	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	NaN	
1	893	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	NaN	
2	894	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	NaN	
3	895	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	NaN	
4	896	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	NaN	

In [11]:

```
# Заполнение всех пропущенных значений нулями
data_new_3 = data.fillna({'age':0})
data_new_3.head()
```

Out[11]:

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Emb
0	892	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	NaN	
1	893	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	NaN	
2	894	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	NaN	
3	895	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	NaN	
4	896	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	NaN	

Внедрение значений

In [12]:

```
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 Age. Тип данных float64. Количество пустых значений 86, 20.57%.
Колонка Fare. Тип данных float64. Количество пустых значений 1, 0.24%.
```

In [13]:

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

Out[13]:

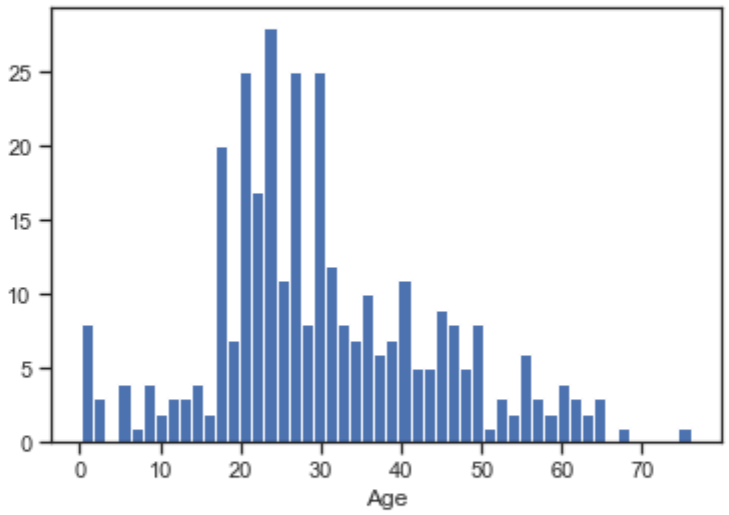
	Age	Fare
0	34.5	7.8292
1	47.0	7.0000
2	62.0	9.6875
3	27.0	8.6625
4	22.0	12.2875
5	14.0	9.2250
6	30.0	7.6292
7	26.0	29.0000
8	18.0	7.2292
9	21.0	24.1500
10	NaN	7.8958
11	46.0	26.0000
12	23.0	82.2667
13	63.0	26.0000
14	47.0	61.1750
15	24.0	27.7208
16	35.0	12.3500
17	21.0	7.2250
18	27.0	7.9250
19	45.0	7.2250
20	55.0	59.4000
21	9.0	3.1708
22	NaN	31.6833
23	21.0	61.3792
24	48.0	262.3750
25	50.0	14.5000
26	22.0	61.9792
27	22.5	7.2250
28	41.0	30.5000
29	NaN	21.6792
...	...	...
388	21.0	7.7500
389	6.0	21.0750
390	23.0	93.5000
391	51.0	39.4000
392	13.0	20.2500

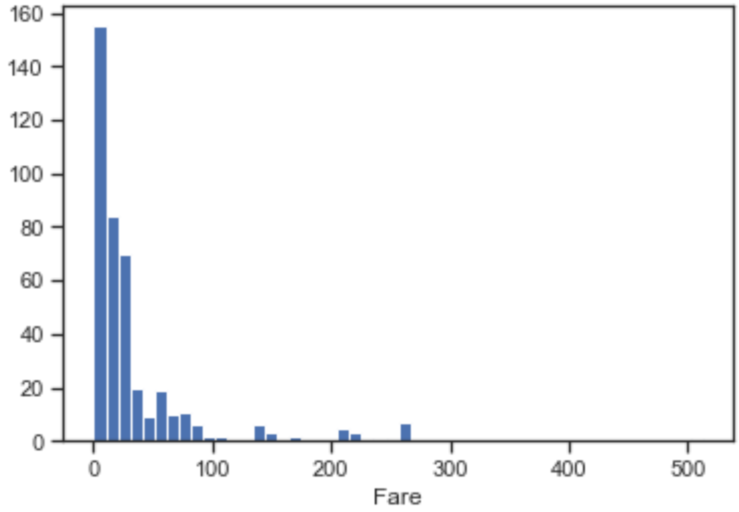
	Age	Fare
393	47.0	10.5000
394	29.0	22.0250
395	18.0	60.0000
396	24.0	7.2500
397	48.0	79.2000
398	22.0	7.7750
399	31.0	7.7333
400	30.0	164.8667
401	38.0	21.0000
402	22.0	59.4000
403	17.0	47.1000
404	43.0	27.7208
405	20.0	13.8625
406	23.0	10.5000
407	50.0	211.5000
408	NaN	7.7208
409	3.0	13.7750
410	NaN	7.7500
411	37.0	90.0000
412	28.0	7.7750
413	NaN	8.0500
414	39.0	108.9000
415	38.5	7.2500
416	NaN	8.0500
417	NaN	22.3583

418 rows × 2 columns

In [14]:

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





In [15]:

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

Out[15]:

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	C
10	902	3	Ilieff, Mr. Ylio	male	NaN	0	0	349220	7.8958	
22	914	1	Flegenheim, Mrs. Alfred (Antoinette)	female	NaN	0	0	PC 17598	31.6833	
29	921	3	Samaan, Mr. Elias	male	NaN	2	0	2662	21.6792	
33	925	3	Johnston, Mrs. Andrew G (Elizabeth Lily" Watson)"	female	NaN	1	2	W./C. 6607	23.4500	
36	928	3	Roth, Miss. Sarah A	female	NaN	0	0	342712	8.0500	
39	931	3	Hee, Mr. Ling	male	NaN	0	0	1601	56.4958	
41	933	1	Franklin, Mr. Thomas Parham	male	NaN	0	0	113778	26.5500	
47	939	3	Shaughnessy, Mr. Patrick	male	NaN	0	0	370374	7.7500	
54	946	2	Mangiavacchi, Mr. Serafino Emilio	male	NaN	0	0	SC/A.3 2861	15.5792	
58	950	3	Davison, Mr. Thomas Henry	male	NaN	1	0	386525	16.1000	
65	957	2	Corey, Mrs. Percy C (Mary Phyllis Elizabeth Mi...	female	NaN	0	0	F.C.C. 13534	21.0000	
76	968	3	Miles, Mr. Frank	male	NaN	0	0	359306	8.0500	
83	975	3	Demetri, Mr. Marinko	male	NaN	0	0	349238	7.8958	
84	976	2	Lamb, Mr. John Joseph	male	NaN	0	0	240261	10.7083	
85	977	3	Khalil, Mr. Betros	male	NaN	1	0	2660	14.4542	
88	980	3	O'Donoghue, Ms. Bridget	female	NaN	0	0	364856	7.7500	
91	983	3	Pedersen, Mr. Olaf	male	NaN	0	0	345498	7.7750	
93	985	3	Guest, Mr. Robert	male	NaN	0	0	376563	8.0500	
102	994	3	Foley, Mr. William	male	NaN	0	0	365235	7.7500	
107	999	3	Ryan, Mr. Edward	male	NaN	0	0	383162	7.7500	
108	1000	3	Willer, Mr. Aaron (Abi Weller")"	male	NaN	0	0	3410	8.7125	

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	C
111	1003	3	Shine, Miss. Ellen Natalia	female	NaN	0	0	330968	7.7792	
116	1008	3	Thomas, Mr. John	male	NaN	0	0	2681	6.4375	
121	1013	3	Kiernan, Mr. John	male	NaN	1	0	367227	7.7500	
124	1016	3	Kennedy, Mr. John	male	NaN	0	0	368783	7.7500	
127	1019	3	McCoy, Miss. Alicia	female	NaN	2	0	367226	23.2500	
132	1024	3	Lefebre, Mrs. Frank (Frances)	female	NaN	0	4	4133	25.4667	
133	1025	3	Thomas, Mr. Charles P	male	NaN	1	0	2621	6.4375	
146	1038	1	Hilliard, Mr. Herbert Henry	male	NaN	0	0	17463	51.8625	
148	1040	1	Crafton, Mr. John Bertram	male	NaN	0	0	113791	26.5500	
...	...	...	...	...	...	...	...	...	...	
268	1160	3	Howard, Miss. May Elizabeth	female	NaN	0	0	A. 2. 39186	8.0500	
271	1163	3	Fox, Mr. Patrick	male	NaN	0	0	368573	7.7500	
273	1165	3	Lennon, Miss. Mary	female	NaN	1	0	370371	15.5000	
274	1166	3	Saade, Mr. Jean Nassr	male	NaN	0	0	2676	7.2250	
282	1174	3	Fleming, Miss. Honora	female	NaN	0	0	364859	7.7500	
286	1178	3	Franklin, Mr. Charles (Charles Fardon)	male	NaN	0	0	SOTON/O.Q. 3101314	7.2500	
288	1180	3	Mardirosian, Mr. Sarkis	male	NaN	0	0	2655	7.2292	F
289	1181	3	Ford, Mr. Arthur	male	NaN	0	0	A/5 1478	8.0500	
290	1182	1	Rheims, Mr. George Alexander Lucien	male	NaN	0	0	PC 17607	39.6000	
292	1184	3	Nasr, Mr. Mustafa	male	NaN	0	0	2652	7.2292	
297	1189	3	Samaan, Mr. Hanna	male	NaN	2	0	2662	21.6792	
301	1193	2	Malachard, Mr. Noel	male	NaN	0	0	237735	15.0458	
304	1196	3	McCarthy, Miss. Catherine Katie""	female	NaN	0	0	383123	7.7500	

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	C
312	1204	3	Sadowitz, Mr. Harry	male	NaN	0	0	LP 1588	7.5750	
332	1224	3	Thomas, Mr. Tannous	male	NaN	0	0	2684	7.2250	
339	1231	3	Betros, Master. Seman	male	NaN	0	0	2622	7.2292	
342	1234	3	Sage, Mr. John George	male	NaN	1	9	CA. 2343	69.5500	
344	1236	3	van Billiard, Master. James William	male	NaN	1	1	A/5. 851	14.5000	
357	1249	3	Lockyer, Mr. Edward	male	NaN	0	0	1222	7.8792	
358	1250	3	O'Keefe, Mr. Patrick	male	NaN	0	0	368402	7.7500	
365	1257	3	Sage, Mrs. John (Annie Bullen)	female	NaN	1	9	CA. 2343	69.5500	
366	1258	3	Caram, Mr. Joseph	male	NaN	1	0	2689	14.4583	
380	1272	3	O'Connor, Mr. Patrick	male	NaN	0	0	366713	7.7500	
382	1274	3	Risien, Mrs. Samuel (Emma)	female	NaN	0	0	364498	14.5000	
384	1276	2	Wheeler, Mr. Edwin Frederick""	male	NaN	0	0	SC/PARIS 2159	12.8750	
408	1300	3	Riordan, Miss. Johanna Hannah""	female	NaN	0	0	334915	7.7208	
410	1302	3	Naughton, Miss. Hannah	female	NaN	0	0	365237	7.7500	
413	1305	3	Spector, Mr. Woolf	male	NaN	0	0	A.5. 3236	8.0500	
416	1308	3	Ware, Mr. Frederick	male	NaN	0	0	359309	8.0500	
417	1309	3	Peter, Master. Michael J	male	NaN	1	1	2668	22.3583	

86 rows × 11 columns

In [16]:

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

Out[16]:

```
Int64Index([ 10,  22,  29,  33,  36,  39,  41,  47,  54,  58,  65,
76,  83,
             84,  85,  88,  91,  93, 102, 107, 108, 111, 116, 121, 1
24, 127,
            132, 133, 146, 148, 151, 160, 163, 168, 170, 173, 183, 1
88, 191,
            199, 200, 205, 211, 216, 219, 225, 227, 233, 243, 244, 2
49, 255,
            256, 265, 266, 267, 268, 271, 273, 274, 282, 286, 288, 2
89, 290,
            292, 297, 301, 304, 312, 332, 339, 342, 344, 357, 358, 3
65, 366,
            380, 382, 384, 408, 410, 413, 416, 417],
           dtype='int64')
```

In [17]:

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

Out[17]:

```
10     NaN
22     NaN
29     NaN
33     NaN
36     NaN
39     NaN
41     NaN
47     NaN
54     NaN
58     NaN
65     NaN
76     NaN
83     NaN
84     NaN
85     NaN
88     NaN
91     NaN
93     NaN
102    NaN
107    NaN
108    NaN
111    NaN
116    NaN
121    NaN
124    NaN
127    NaN
132    NaN
133    NaN
146    NaN
148    NaN
        ..
268    NaN
271    NaN
273    NaN
274    NaN
282    NaN
286    NaN
288    NaN
289    NaN
290    NaN
292    NaN
297    NaN
301    NaN
304    NaN
312    NaN
332    NaN
339    NaN
342    NaN
344    NaN
357    NaN
358    NaN
365    NaN
366    NaN
380    NaN
382    NaN
384    NaN
408    NaN
410    NaN
413    NaN
```

```
416    NaN
417    NaN
Name: Age, Length: 86, dtype: float64
```

In [18]:

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

Out[18]:

	Age
0	34.5
1	47.0
2	62.0
3	27.0
4	22.0

In [25]:

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

In [26]:

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

In [31]:

```
def test_num_impute_col(dataset, column, strategy_param):
    temp_data = dataset[[column]]

    indicator = MissingIndicator()
    mask_missing_values_only = indicator.fit_transform(temp_data)

    imp_num = SimpleImputer(strategy=strategy_param)
    data_num_imp = imp_num.fit_transform(temp_data)

    filled_data = data_num_imp[mask_missing_values_only]

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

In [33]:

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

Out[33]:

```
('Age',
 'most_frequent',
 86,
 21.0,
 21.0,
 array([21., 21., 21., 21., 21., 21., 21., 21., 21., 21., 21., 21.,
21.,
        21., 21., 21., 21., 21., 21., 21., 21., 21., 21., 21., 21.,
21.,
        21., 21., 21., 21., 21., 21., 21., 21., 21., 21., 21., 21.,
21.,
        21., 21., 21., 21., 21., 21., 21., 21., 21., 21., 21., 21.,
21.,
        21., 21., 21., 21., 21., 21., 21., 21., 21., 21., 21., 21.,
21.,
        21., 21., 21., 21., 21., 21., 21., 21., 21., 21., 21., 21.,
21.,
        21., 21., 21., 21., 21., 21., 21., 21.]))
```

Преобразование категориальных признаков в числовые

Кодирование категорий целочисленными значениями

In [34]:

```
data = pd.read_csv('datasets/titanic.csv', sep=",")
data.head()
```

Out[34]:

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Emb
0	892	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	NaN	
1	893	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	NaN	
2	894	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	NaN	
3	895	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	NaN	
4	896	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	NaN	

In [36]:

```
data["Embarked"].unique()
```

Out[36]:

```
array(['Q', 'S', 'C'], dtype=object)
```

In [37]:

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

In [38]:

```
le = LabelEncoder()
emb_le = le.fit_transform(data['Embarked'])
```

In [39]:

```
np.unique(emb_le)
```

Out[39]:

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

In [40]:

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

Out[40]:

```
array(['C', 'Q', 'S'], dtype=object)
```

Масштабирование данных

In [41]:

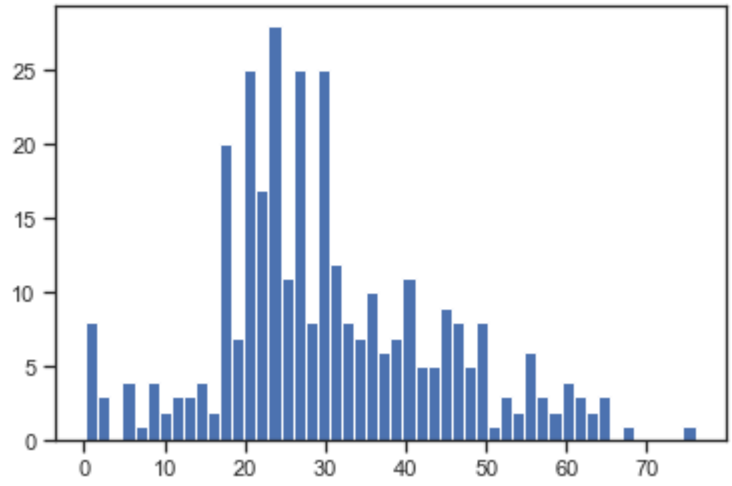
```
from sklearn.preprocessing import MinMaxScaler, StandardScaler, Normalizer
```

In [43]:

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

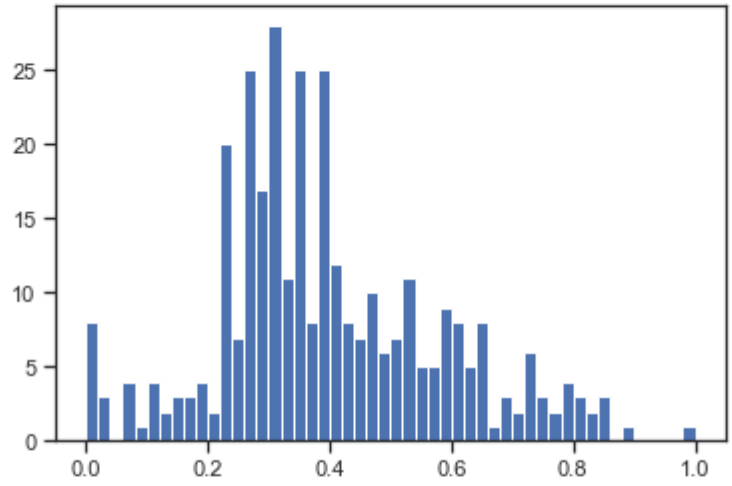
In [44]:

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



In [45]:

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



25
20
15
10
5
0
0.0
0.2
0.4
0.6
0.8
1.0