Рубежный контроль №1

Нищук Роман Сергеевич, ИУ5-62Б, Вариант 17. Задание 3, Датасет №1.

Задача №3.

Для заданного набора данных произведите масштабирование данных (для одного признака) и преобразование категориальных признаков в количественные двумя способами **(label encoding, one hot encoding)** для одного признака. Какие методы Вы использовали для решения задачи и почему?

Дополнение для ИУ5-62Б

Для произвольной колонки данных построить гистограмму

In [24]:

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import *
```

Датасет

In [25]:

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

In [26]:

```
# первые строки
data.head()
```

Out[26]:

	Unnamed: 0	ID	Name	Age	Photo	Nationality	Fl
0	0	158023	L. Messi	31	https://cdn.sofifa.org/players/4/19/158023.png	Argentina	https://cdn.sofifa.org/flags/52.p
1	1	20801	Cristiano Ronaldo	33	https://cdn.sofifa.org/players/4/19/20801.png	Portugal	https://cdn.sofifa.org/flags/38.p
2	2	190871	Neymar Jr	26	https://cdn.sofifa.org/players/4/19/190871.png	Brazil	https://cdn.sofifa.org/flags/54.p
3	3	193080	De Gea	27	https://cdn.sofifa.org/players/4/19/193080.png	Spain	https://cdn.sofifa.org/flags/45.p
4	4	192985	K. De Bruyne	27	https://cdn.sofifa.org/players/4/19/192985.png	Belgium	https://cdn.sofifa.org/flags/7.p

5 rows × 89 columns

In [27]:

```
# типы колонок
data.dtypes
```

Out[27]:

```
Unnamed: 0           int64
ID                   int64
Name                object
Age                  int64
Photo               object
                    ...
GKHandling         float64
GKKicking          float64
GKPositioning      float64
GKReflexes         float64
Release Clause      object
Length: 89, dtype: object
```

Масштабирование

In [28]:

```
# Статистика датасета
data.describe()
```

Out[28]:

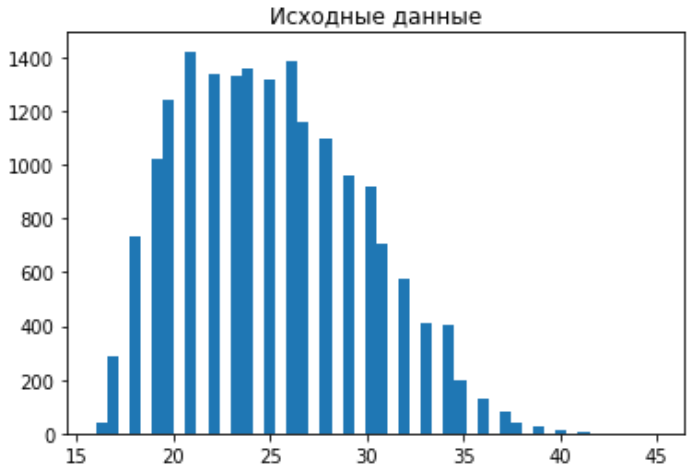
	Unnamed: 0	ID	Age	Overall	Potential	Special	International Reputation	Weak Foot	
count	18207.000000	18207.000000	18207.000000	18207.000000	18207.000000	18207.000000	18159.000000	18159.000000	18
mean	9103.000000	214298.338606	25.122206	66.238699	71.307299	1597.809908	1.113222	2.947299	
std	5256.052511	29965.244204	4.669943	6.908930	6.136496	272.586016	0.394031	0.660456	
min	0.000000	16.000000	16.000000	46.000000	48.000000	731.000000	1.000000	1.000000	
25%	4551.500000	200315.500000	21.000000	62.000000	67.000000	1457.000000	1.000000	3.000000	
50%	9103.000000	221759.000000	25.000000	66.000000	71.000000	1635.000000	1.000000	3.000000	
75%	13654.500000	236529.500000	28.000000	71.000000	75.000000	1787.000000	1.000000	3.000000	
max	18206.000000	246620.000000	45.000000	94.000000	95.000000	2346.000000	5.000000	5.000000	

8 rows × 44 columns

Исходные данные - значения лежат в диапазоне от **1** до **4043**

In [31]:

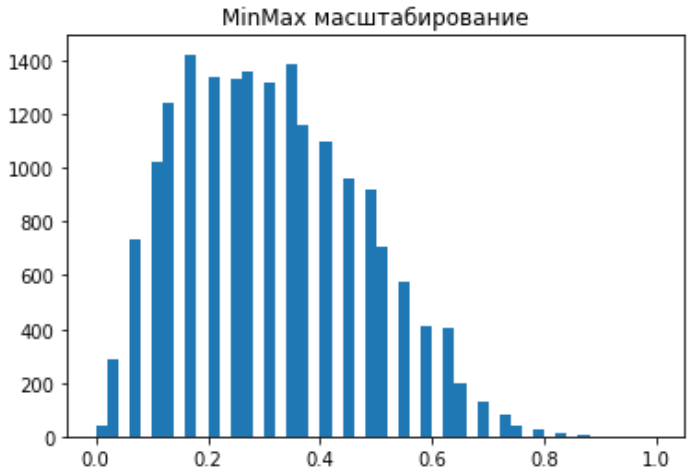
```
plt.hist(data['Age'], 50)
plt.title("Исходные данные")
plt.show()
```



Масштабирование на основе **MinMax** - значения лежат в диапазоне от **0** до **1**

In [32]:

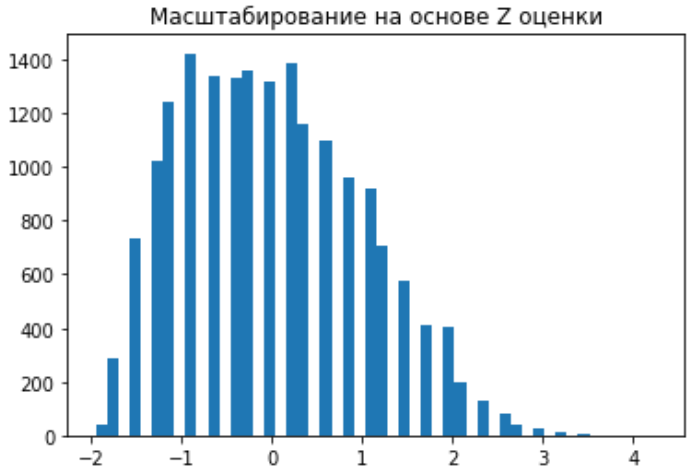
```
sc1 = MinMaxScaler()
sc1_data = sc1.fit_transform(data[['Age']])
plt.hist(sc1_data, 50)
plt.title("MinMax масштабирование")
plt.show()
```



Z оценка - значения лежат в диапазоне от -3 до 3

In [33]:

```
sc2 = StandardScaler()
sc2_data = sc2.fit_transform(data[['Age']])
plt.hist(sc2_data, 50)
plt.title("Масштабирование на основе Z оценки")
plt.show()
```



Преобразование категориальных признаков в количественные

Label encoding

In [34]:

```
# обработка пропусков с заменой на "Unknown"
imp2 = SimpleImputer(missing_values=np.nan, strategy='constant', fill_value='Unknown')
data['Nationality'] = imp2.fit_transform(data[['Nationality']])

#Уникальные типы
types = data['Nationality']
types.unique()
```

Out[34]:

```
array(['Argentina', 'Portugal', 'Brazil', 'Spain', 'Belgium', 'Croatia',
```

```
array(['Argentina', 'Portugal', 'Brazil', 'Spain', 'Belgium', 'Croatia',
       'Uruguay', 'Slovenia', 'Poland', 'Germany', 'France', 'England',
       'Italy', 'Egypt', 'Colombia', 'Denmark', 'Gabon', 'Wales',
       'Senegal', 'Costa Rica', 'Slovakia', 'Netherlands',
       'Bosnia Herzegovina', 'Morocco', 'Serbia', 'Algeria', 'Austria',
       'Greece', 'Chile', 'Sweden', 'Korea Republic', 'Finland', 'Guinea',
       'Montenegro', 'Armenia', 'Switzerland', 'Norway', 'Czech Republic',
       'Scotland', 'Ghana', 'Central African Rep.', 'DR Congo',
       'Ivory Coast', 'Russia', 'Ukraine', 'Iceland', 'Mexico', 'Jamaica',
       'Albania', 'Venezuela', 'Japan', 'Turkey', 'Ecuador', 'Paraguay',
       'Mali', 'Nigeria', 'Cameroon', 'Dominican Republic', 'Israel',
       'Kenya', 'Hungary', 'Republic of Ireland', 'Romania',
       'United States', 'Cape Verde', 'Australia', 'Peru', 'Togo',
       'Syria', 'Zimbabwe', 'Angola', 'Burkina Faso', 'Iran', 'Estonia',
       'Tunisia', 'Equatorial Guinea', 'New Zealand', 'FYR Macedonia',
       'United Arab Emirates', 'China PR', 'Guinea Bissau', 'Bulgaria',
       'Kosovo', 'South Africa', 'Madagascar', 'Georgia', 'Tanzania',
       'Gambia', 'Cuba', 'Belarus', 'Uzbekistan', 'Benin', 'Congo',
       'Mozambique', 'Honduras', 'Canada', 'Northern Ireland', 'Cyprus',
       'Saudi Arabia', 'Curacao', 'Moldova', 'Bolivia',
       'Trinidad & Tobago', 'Sierra Leone', 'Zambia', 'Chad',
       'Philippines', 'Haiti', 'Comoros', 'Libya', 'Panama',
       'São Tomé & Príncipe', 'Eritrea', 'Oman', 'Iraq', 'Burundi',
       'Fiji', 'New Caledonia', 'Lithuania', 'Luxembourg', 'Korea DPR',
       'Liechtenstein', 'St Kitts Nevis', 'Latvia', 'Suriname', 'Uganda',
       'El Salvador', 'Bermuda', 'Kuwait', 'Antigua & Barbuda',
       'Thailand', 'Mauritius', 'Guatemala', 'Liberia', 'Kazakhstan',
       'Niger', 'Mauritania', 'Montserrat', 'Namibia', 'Azerbaijan',
       'Guam', 'Faroe Islands', 'India', 'Nicaragua', 'Barbados',
       'Lebanon', 'Palestine', 'Guyana', 'Sudan', 'St Lucia', 'Ethiopia',
       'Puerto Rico', 'Grenada', 'Jordan', 'Rwanda', 'Qatar',
       'Afghanistan', 'Hong Kong', 'Andorra', 'Malta', 'Belize',
       'South Sudan', 'Indonesia', 'Botswana'], dtype=object)
```

In [35]:

```
#label encoding
le = LabelEncoder()
data_le = le.fit_transform(types)
```

Результат

In [36]:

```
np.unique(data_le)
```

Out[36]:

```
array([  0,   1,   2,   3,   4,   5,   6,   7,   8,   9,  10,  11,  12,
        13,  14,  15,  16,  17,  18,  19,  20,  21,  22,  23,  24,  25,
        26,  27,  28,  29,  30,  31,  32,  33,  34,  35,  36,  37,  38,
        39,  40,  41,  42,  43,  44,  45,  46,  47,  48,  49,  50,  51,
        52,  53,  54,  55,  56,  57,  58,  59,  60,  61,  62,  63,  64,
        65,  66,  67,  68,  69,  70,  71,  72,  73,  74,  75,  76,  77,
        78,  79,  80,  81,  82,  83,  84,  85,  86,  87,  88,  89,  90,
        91,  92,  93,  94,  95,  96,  97,  98,  99, 100, 101, 102, 103,
       104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116,
       117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129,
       130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142,
       143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155,
       156, 157, 158, 159, 160, 161, 162, 163])
```

Обратное преобразование

In [37]:

```
le.inverse_transform(data_le)
```

Out[37]:

```
array(['Argentina', 'Portugal', 'Brazil', ..., 'England', 'England',
       'England'], dtype=object)
```

England '], dtype=object)

One hot encoding

In [38]:

```
pd.get_dummies(data['Nationality']).head()
```

Out[38]:

	Afghanistan	Albania	Algeria	Andorra	Angola	Antigua & Barbuda	Argentina	Armenia	Australia	Austria	...	Uganda	Ukraine	U Emi
0	0	0	0	0	0	0	1	0	0	0	...	0	0	
1	0	0	0	0	0	0	0	0	0	0	...	0	0	
2	0	0	0	0	0	0	0	0	0	0	...	0	0	
3	0	0	0	0	0	0	0	0	0	0	...	0	0	
4	0	0	0	0	0	0	0	0	0	0	...	0	0	

5 rows × 164 columns

Гистограмма

Построим гистрограмму, которая позволит оценить плотность вероятности распределения данных.

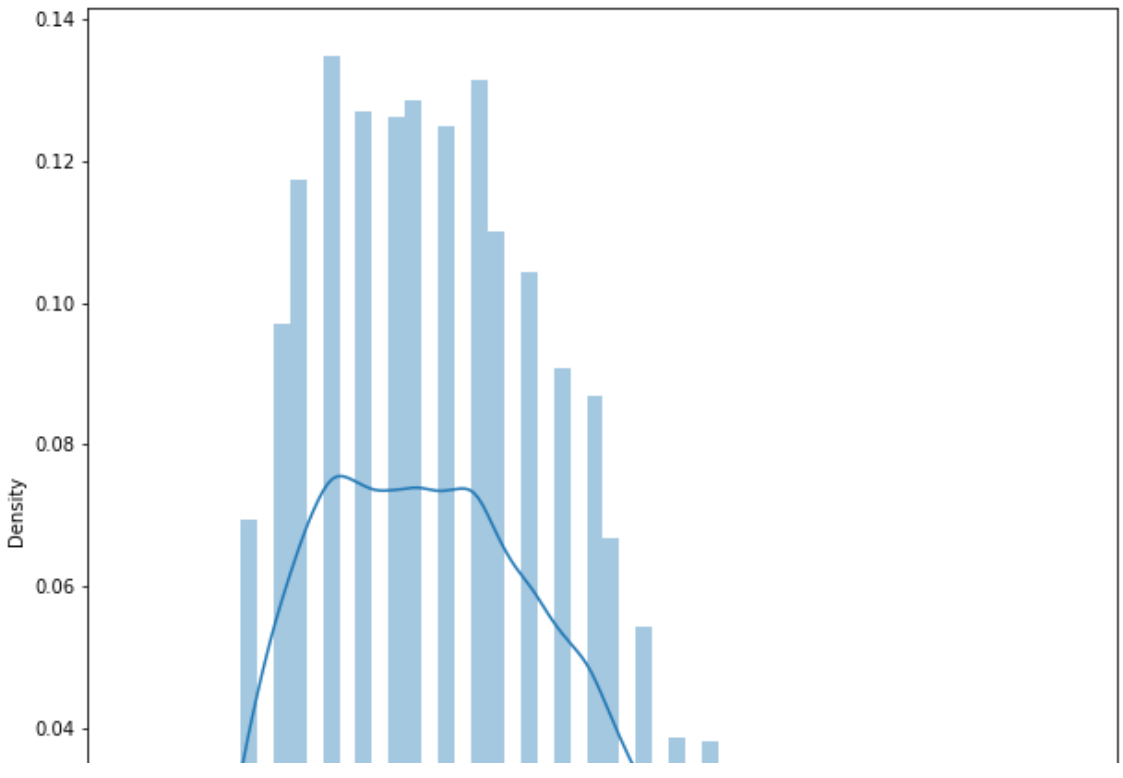
In [39]:

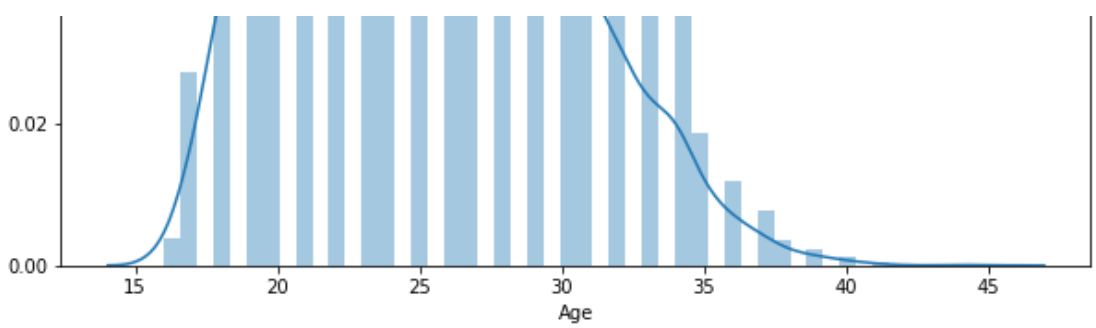
```
fig, ax = plt.subplots(figsize=(10,10))
sns.distplot(data['Age'])
```

```
c:\users\r3d\appdata\local\programs\python\python38\lib\site-packages\seaborn\distributio
ns.py:2557: FutureWarning: `distplot` is a deprecated function and will be removed in a f
uture version. Please adapt your code to use either `displot` (a figure-level function wi
th similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[39]:

```
<AxesSubplot:xlabel='Age', ylabel='Density'>
```





In [ ]: