Рубежный контроль №1

Шако Давиташвили, ИУ5И-65Б

Вариант 18

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? Для набора данных построить "парные диаграммы".

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

data = load_wine()

df = pd.DataFrame(data.data, columns=data.feature_names)
df['target'] = data.target

df.head()
```

```
   alcohol  malic_acid   ash  alcalinity_of_ash  magnesium
total_phenols  \
0    14.23        1.71  2.43               15.6      127.0
2.80
1    13.20        1.78  2.14               11.2      100.0
2.65
2    13.16        2.36  2.67               18.6      101.0
2.80
3    14.37        1.95  2.50               16.8      113.0
3.85
4    13.24        2.59  2.87               21.0      118.0
2.80

   flavanoids  nonflavanoid_phenols  proanthocyanins  color_intensity
hue  \
0        3.06                  0.28             2.29             5.64
1.04
1        2.76                  0.26             1.28             4.38
```

```
1.05
2        3.24                  0.30             2.81              5.68
1.03
3        3.49                  0.24             2.18              7.80
0.86
4        2.69                  0.39             1.82              4.32
1.04
```

```
   od280/od315_of_diluted_wines  proline  target
0                          3.92   1065.0       0
1                          3.40   1050.0       0
2                          3.17   1185.0       0
3                          3.45   1480.0       0
4                          2.93    735.0       0
```

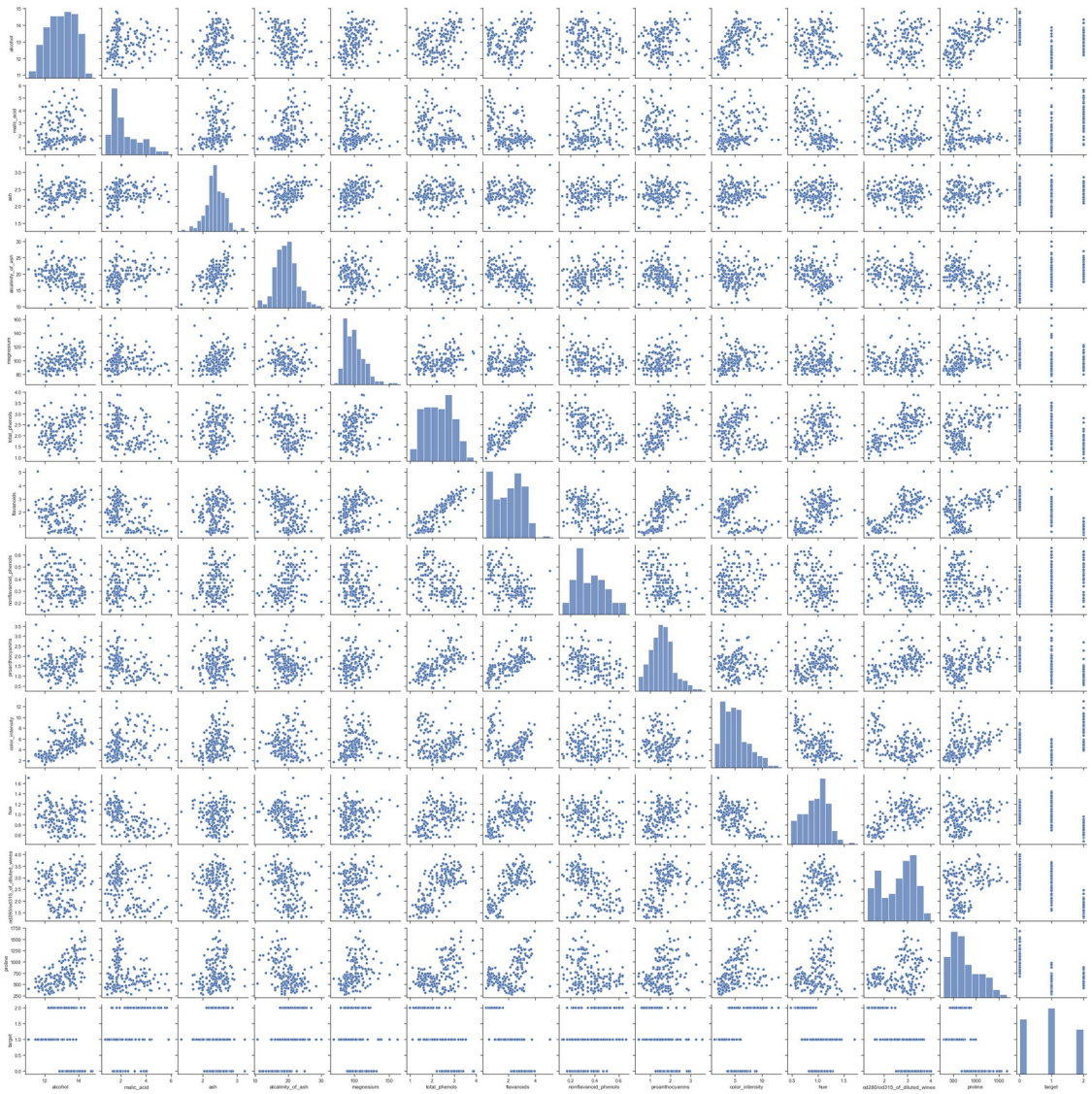
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 178 entries, 0 to 177
Data columns (total 14 columns):
 #   Column                        Non-Null Count  Dtype  
---  ------                        --------------  -----  
 0   alcohol                       178 non-null    float64
 1   malic_acid                    178 non-null    float64
 2   ash                           178 non-null    float64
 3   alcalinity_of_ash             178 non-null    float64
 4   magnesium                     178 non-null    float64
 5   total_phenols                 178 non-null    float64
 6   flavanoids                    178 non-null    float64
 7   nonflavanoid_phenols          178 non-null    float64
 8   proanthocyanins               178 non-null    float64
 9   color_intensity               178 non-null    float64
 10  hue                           178 non-null    float64
 11  od280/od315_of_diluted_wines  178 non-null    float64
 12  proline                       178 non-null    float64
 13  target                        178 non-null    int32  
dtypes: float64(13), int32(1)
memory usage: 18.9 KB
```

Парные диаграммы

```
sns.pairplot(df)
```

```
<seaborn.axisgrid.PairGrid at 0x2533ead4ac0>
```

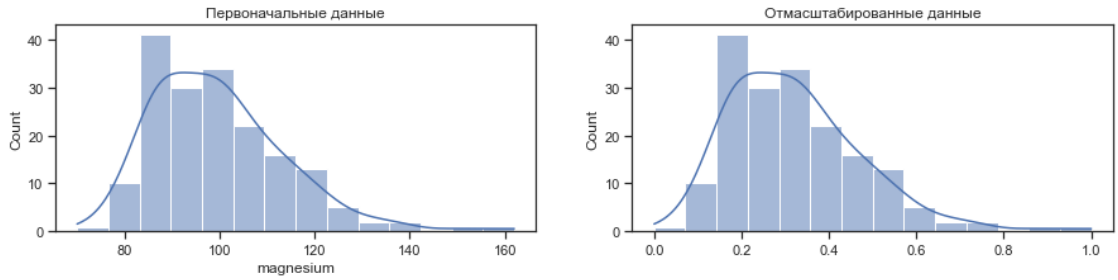


Масштабирование данных

```
fig, ax = plt.subplots(1, 2, figsize = (15, 3))
sns.histplot(df['magnesium'], ax = ax[0], kde = True, legend = False)
ax[0].set_title("Первоначальные данные")

sc = MinMaxScaler()
scaled_data = sc.fit_transform(df[['magnesium']])

sns.histplot(scaled_data, ax = ax[1], kde = True, legend = False)
ax[1].set_title("Отмасштабированные данные")
plt.show()
```



Кодирование категориальных признаков

Label Encoding

В данном датасете все признаки выражены числовыми значениями, поэтому выберем другой датасет для решения задачи кодирования.

```
data = pd.read_csv("data.csv")
```

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 300000 entries, 0 to 299999
Data columns (total 25 columns):
 #   Column  Non-Null Count   Dtype 
---  ------  --------------   ----- 
 0   id      300000 non-null  int64 
 1   bin_0   300000 non-null  int64 
 2   bin_1   300000 non-null  int64 
 3   bin_2   300000 non-null  int64 
 4   bin_3   300000 non-null  object
 5   bin_4   300000 non-null  object
 6   nom_0   300000 non-null  object
 7   nom_1   300000 non-null  object
 8   nom_2   300000 non-null  object
 9   nom_3   300000 non-null  object
 10  nom_4   300000 non-null  object
 11  nom_5   300000 non-null  object
 12  nom_6   300000 non-null  object
 13  nom_7   300000 non-null  object
 14  nom_8   300000 non-null  object
 15  nom_9   300000 non-null  object
 16  ord_0   300000 non-null  int64 
 17  ord_1   300000 non-null  object
 18  ord_2   300000 non-null  object
 19  ord_3   300000 non-null  object
 20  ord_4   300000 non-null  object
 21  ord_5   300000 non-null  object
 22  day     300000 non-null  int64 
 23  month   300000 non-null  int64 
 24  target  300000 non-null  int64 
```

```
dtypes: int64(8), object(17)
memory usage: 57.2+ MB
```

```
data.head()
```

```
   id  bin_0  bin_1  bin_2 bin_3 bin_4  nom_0      nom_1    nom_2
nom_3  \
0   0      0      0      0     T     Y  Green   Triangle    Snake
Finland
1   1      0      1      0     T     Y  Green  Trapezoid  Hamster
Russia
2   2      0      0      0     F     Y   Blue  Trapezoid     Lion
Russia
3   3      0      1      0     F     Y    Red  Trapezoid    Snake
Canada
4   4      0      0      0     F     N    Red  Trapezoid     Lion
Canada

   ...      nom_9 ord_0        ord_1        ord_2 ord_3 ord_4  ord_5
day  \
0  ...  2f4cb3d51     2  Grandmaster         Cold     h     D     kr
2
1  ...  f83c56c21     1  Grandmaster          Hot     a     A     bF
7
2  ...  ae6800dd0     1       Expert     Lava Hot     h     R     Jc
7
3  ...  8270f0d71     1  Grandmaster  Boiling Hot     i     D     kW
2
4  ...  b164b72a7     1  Grandmaster     Freezing     a     R     qP
7

  month target
0     2      0
1     8      0
2     2      0
3     1      1
4     8      0

[5 rows x 25 columns]
```

```
data.nom_0.unique()
```

```
array(['Green', 'Blue', 'Red'], dtype=object)
```

Используем столбец nom_0 для кодирования.

```
color = data[['nom_0']]
```

```
color
```

```
        nom_0
0       Green
```

```
1       Green
2        Blue
3         Red
4         Red
...       ...
299995    Red
299996  Green
299997   Blue
299998  Green
299999   Blue

[300000 rows x 1 columns]
```

```
le = LabelEncoder()
color_le = le.fit_transform(color['nom_0'])
```

```
np.unique(color_le)
```

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

```
le.classes_
```

```
array(['Blue', 'Green', 'Red'], dtype=object)
```

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

```
array(['Blue', 'Green', 'Red'], dtype=object)
```

One Hot Encoding

```
ohe = OneHotEncoder()
color_ohe = ohe.fit_transform(color[['nom_0']])
color.shape
```

```
(300000, 1)
```

```
color_ohe.shape
```

```
(300000, 3)
```

```
color_ohe
```

```
<300000x3 sparse matrix of type '<class 'numpy.float64'>'
     with 300000 stored elements in Compressed Sparse Row format>
```

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

```
matrix([[0., 1., 0.],
        [0., 1., 0.],
        [1., 0., 0.],
        [0., 0., 1.],
        [0., 0., 1.],
        [1., 0., 0.],
        [0., 1., 0.],
```

```
        [0., 0., 1.],
        [1., 0., 0.],
        [0., 0., 1.]])
```

```
color.head(10)
```

```
   nom_0
0  Green
1  Green
2   Blue
3    Red
4    Red
5   Blue
6  Green
7    Red
8   Blue
9    Red
```

```
pd.get_dummies(color).head()
```

```
   nom_0_Blue  nom_0_Green  nom_0_Red
0           0            1          0
1           0            1          0
2           1            0          0
3           0            0          1
4           0            0          1
```