Юриков Егор ИУ5-65Б Вариант 17 Задача №3 Датасет №1

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets import load_iris
import pandas as pd
from sklearn.preprocessing import MinMaxScaler, StandardScaler,
LabelEncoder, OneHotEncoder
import seaborn as sns
```

```
iris = load_iris()
data = pd.DataFrame(data= np.c_[iris['data'], iris['target']],
columns= iris['feature_names'] + ['target'])
data
```

```
     sepal length (cm)  sepal width (cm)  petal length (cm)  petal
width (cm)
0                  5.1               3.5                1.4
0.2  \
1                  4.9               3.0                1.4
0.2
2                  4.7               3.2                1.3
0.2
3                  4.6               3.1                1.5
0.2
4                  5.0               3.6                1.4
0.2
..                 ...               ...                ...
...
145                6.7               3.0                5.2
2.3
146                6.3               2.5                5.0
1.9
147                6.5               3.0                5.2
2.0
148                6.2               3.4                5.4
2.3
149                5.9               3.0                5.1
1.8

     target
0       0.0
1       0.0
2       0.0
3       0.0
4       0.0
..      ...
145     2.0
146     2.0
147     2.0
```

```
148     2.0
149     2.0

[150 rows x 5 columns]
```

```
print('max = ', max(data['sepal length (cm)']), '\nmin = ',
min(data['sepal length (cm)']))
```

```
max =  7.9
min =  4.3
```

```
mms = MinMaxScaler()
min_max_data = mms.fit_transform(data[['sepal length (cm)']])
print('max = ', max(min_max_data), '\nmin = ', min(min_max_data))
```

```
max =  [1.]
min =  [0.]
```

```
sts = StandardScaler()
st_scaler_data = sts.fit_transform(data[['sepal length (cm)']])
print('max = ', max(st_scaler_data), '\nmin = ', min(st_scaler_data))
```

```
max =  [2.4920192]
min =  [-1.87002413]
```

```
names = list(iris.target_names)
mas = []
for item in data['target']:
    mas.append(names[int(item)])
data['target_names'] = mas
data = data.drop(columns=['target'])
```

```
le = LabelEncoder()
le_encoded = le.fit_transform(data['target_names'])
data['encoded_target'] = le_encoded
data
```

```
     sepal length (cm)  sepal width (cm)  petal length (cm)  petal
width (cm)
0                  5.1               3.5                1.4
0.2  \
1                  4.9               3.0                1.4
0.2
2                  4.7               3.2                1.3
0.2
3                  4.6               3.1                1.5
0.2
4                  5.0               3.6                1.4
0.2
..                 ...               ...                ...
...
145                6.7               3.0                5.2
```

```
2.3
146                 6.3               2.5                5.0
1.9
147                 6.5               3.0                5.2
2.0
148                 6.2               3.4                5.4
2.3
149                 5.9               3.0                5.1
1.8

    target_names  encoded_target
0         setosa               0
1         setosa               0
2         setosa               0
3         setosa               0
4         setosa               0
..           ...             ...
145    virginica               2
146    virginica               2
147    virginica               2
148    virginica               2
149    virginica               2

[150 rows x 6 columns]
```

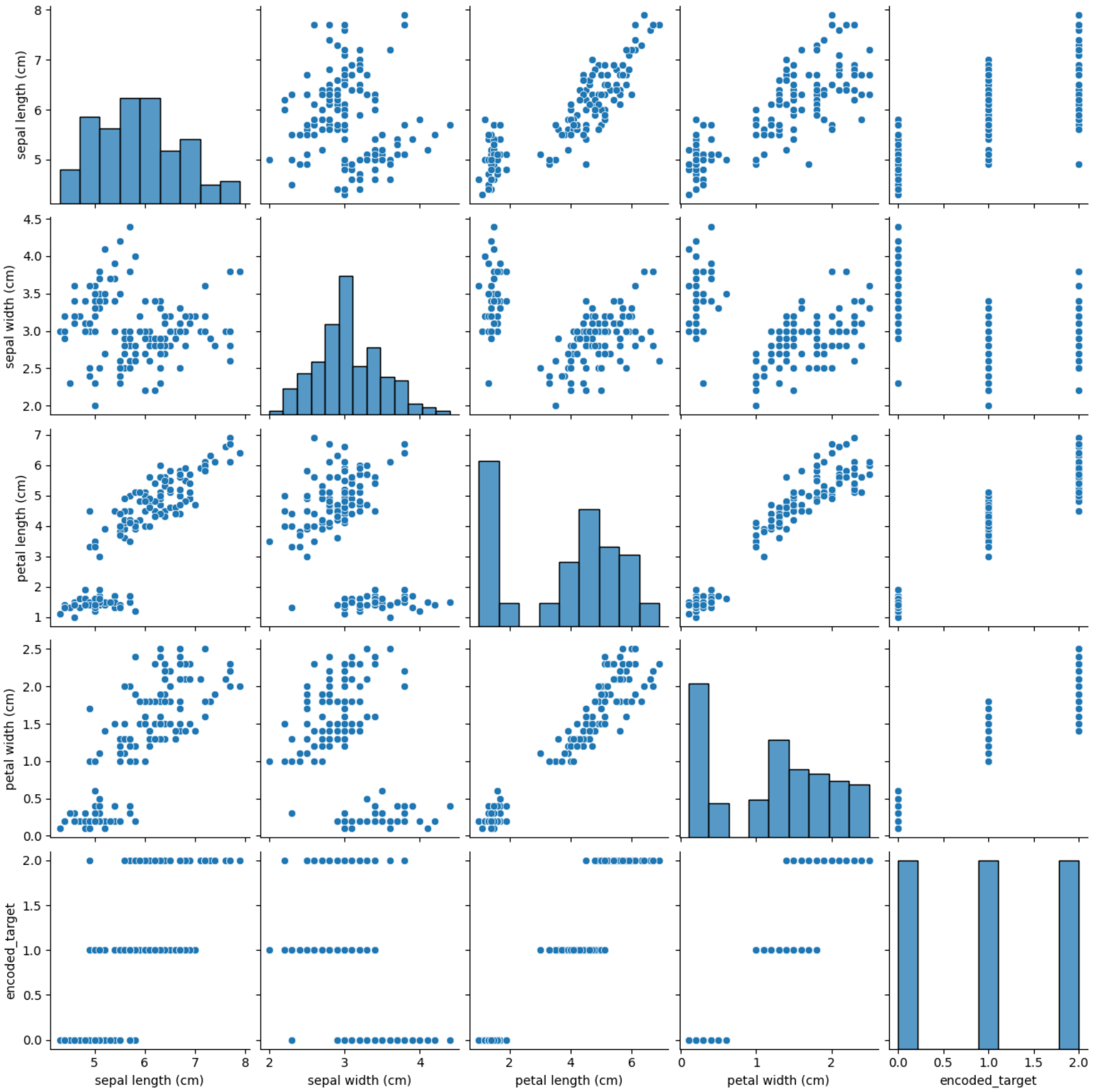
```
ohe = OneHotEncoder()
ohe_encoded = ohe.fit_transform(data[['target_names']])
data['encoded_setosa'] = list(ohe_encoded.toarray()[:, 0])
data['encoded_versicolor'] = list(ohe_encoded.toarray()[:, 1])
data['encoded_virginica'] = list(ohe_encoded.toarray()[:, 2])
data
```

```
     sepal length (cm)  sepal width (cm)  petal length (cm)  petal
width (cm)
0                   5.1               3.5                1.4
0.2  \
1                   4.9               3.0                1.4
0.2
2                   4.7               3.2                1.3
0.2
3                   4.6               3.1                1.5
0.2
4                   5.0               3.6                1.4
0.2
..                  ...               ...                ...
...
145                 6.7               3.0                5.2
2.3
146                 6.3               2.5                5.0
1.9
```

```
147                6.5               3.0                5.2
2.0
148                6.2               3.4                5.4
2.3
149                5.9               3.0                5.1
1.8

    target_names  encoded_target  encoded_setosa  encoded_versicolor

0         setosa               0             1.0
0.0  \
1         setosa               0             1.0                 0.0

2         setosa               0             1.0                 0.0

3         setosa               0             1.0                 0.0

4         setosa               0             1.0                 0.0

..           ...             ...             ...                 ...

145    virginica               2             0.0                 0.0

146    virginica               2             0.0                 0.0

147    virginica               2             0.0                 0.0

148    virginica               2             0.0                 0.0

149    virginica               2             0.0                 0.0


     encoded_virginica
0                  0.0
1                  0.0
2                  0.0
3                  0.0
4                  0.0
..                 ...
145                1.0
146                1.0
147                1.0
148                1.0
149                1.0

[150 rows x 9 columns]
```

```
sns.pairplot(data=data[['sepal length (cm)', 'sepal width (cm)',
'petal length (cm)', 'petal width (cm)', 'encoded_target']])
```

```
<seaborn.axisgrid.PairGrid at 0x1a68e046cd0>
```



Для решеения задачи использовался метод fit_transform(), потому что этот метод считает выборочные мат. ожидание и среднее квадратическое отклонение и преобразовывает признак согласно подсчитанным значениям.