

```
In [1]: import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import precision_score, balanced_accuracy_score
```

```
In [2]: data = pd.read_csv("heart.csv")
```

```
In [3]: data.head()
```

```
Out[3]:
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	52	1	0	125	212	0	1	168	0	1.0	2	2	3	0
1	53	1	0	140	203	1	0	155	1	3.1	0	0	3	0
2	70	1	0	145	174	0	1	125	1	2.6	0	0	3	0
3	61	1	0	148	203	0	1	161	0	0.0	2	1	3	0
4	62	0	0	138	294	1	1	106	0	1.9	1	3	2	0

Предобработка данных

```
In [4]: data.dtypes
```

```
Out[4]: age      int64
sex      int64
cp       int64
trestbps int64
chol     int64
fbs      int64
restecg  int64
thalach  int64
exang    int64
```

```
Out[4]: age          int64
sex          int64
cp           int64
trestbps     int64
chol         int64
fbs          int64
restecg      int64
thalach      int64
exang        int64
oldpeak      float64
slope        int64
ca           int64
thal         int64
target       int64
dtype: object
```

```
In [5]: data.shape
```

```
Out[5]: (1025, 14)
```

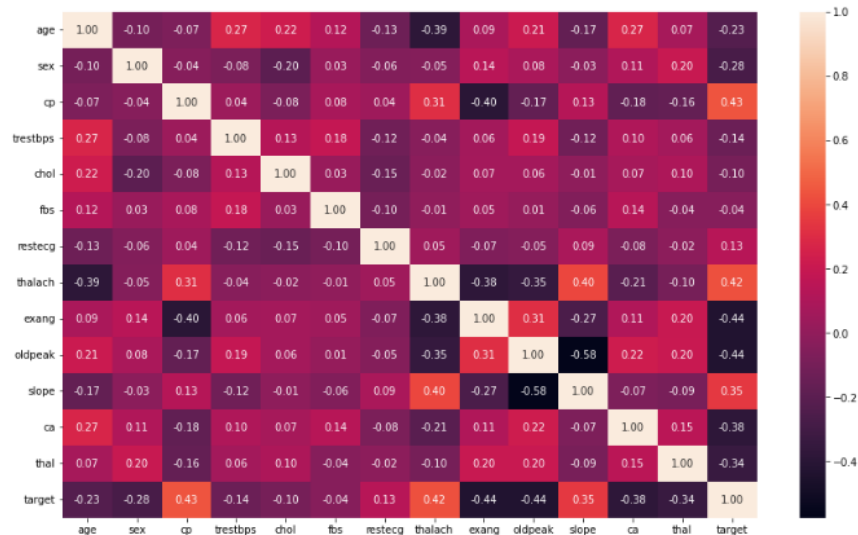
```
In [6]: data.isnull().sum()
```

```
Out[6]: age      0
sex      0
cp       0
trestbps 0
chol     0
fbs      0
restecg  0
thalach  0
exang    0
oldpeak  0
slope    0
ca       0
thal     0
target   0
dtype: int64
```

target: int64

```
In [7]: fig, ax = plt.subplots(figsize=(15,9))
sns.heatmap(data.corr(method="pearson"), ax=ax, annot=True, fmt=".2f")
```

Out[7]: <AxesSubplot:>



```
In [8]: target="target"
xArray = data.drop(target, axis=1)
```

target: int64

```
In [8]: target="target"
xArray = data.drop(target, axis=1)
yArray = data[target]
trainX, testX, trainY, testY = train_test_split(xArray, yArray, test_size=0.2, random_state=1)
```

Логистическая регрессия

```
In [9]: LR = LogisticRegression()
LR.fit(trainX, trainY)
```

C:\Users\79634\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\linear_model_logistic.py:444: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
<https://scikit-learn.org/stable/modules/preprocessing.html>
Please also refer to the documentation for alternative solver options:
https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
n_iter_i = _check_optimize_result

```
Out[9]: LogisticRegression
```

```
In [11]: BAS_LR = balanced_accuracy_score(testY, LR.predict(testX))
PS_LR = precision_score(testY, LR.predict(testX))
```

```
In [12]: print ("Метрика balanced_accuracy_score: {}".format(BAS_LR))
print ("Метрика precision_score: {}".format(PS_LR))
```

Метрика balanced_accuracy_score: 0.8142679663608563
Метрика precision_score: 0.7522123893805309

Градиентный бустинг

```
In [14]: from catboost import CatBoostClassifier
```

```
In [27]: exang="exang"
X_Array = data.drop(exang, axis=1)
y_Array = data[exang]
trainX, testX, trainY, testY = train_test_split(X_Array, y_Array, test_size=0.2, random_state=1)
```

```
In [30]: model2 = CatBoostClassifier(iterations=200,
                                   learning_rate=1,
                                   min_data_in_leaf=1)
```

```
In [32]: model2.fit(trainX, trainY)
```

```
59:  learn: 0.0024372    total: 199ms  remaining: 464ms
60:  learn: 0.0022911    total: 202ms  remaining: 460ms
61:  learn: 0.0022260    total: 206ms  remaining: 458ms
62:  learn: 0.0022253    total: 231ms  remaining: 502ms
63:  learn: 0.0022253    total: 234ms  remaining: 497ms
64:  learn: 0.0022253    total: 237ms  remaining: 492ms
65:  learn: 0.0022252    total: 240ms  remaining: 487ms
66:  learn: 0.0022252    total: 285ms  remaining: 567ms
67:  learn: 0.0022157    total: 289ms  remaining: 561ms
68:  learn: 0.0022155    total: 292ms  remaining: 554ms
69:  learn: 0.0022155    total: 295ms  remaining: 547ms
70:  learn: 0.0022155    total: 300ms  remaining: 545ms
71:  learn: 0.0022155    total: 305ms  remaining: 542ms
72:  learn: 0.0021900    total: 308ms  remaining: 536ms
73:  learn: 0.0021360    total: 311ms  remaining: 530ms
74:  learn: 0.0021360    total: 315ms  remaining: 525ms
75:  learn: 0.0021359    total: 321ms  remaining: 524ms
76:  learn: 0.0020872    total: 356ms  remaining: 568ms
77:  learn: 0.0020868    total: 359ms  remaining: 562ms
78:  learn: 0.0020676    total: 364ms  remaining: 557ms
```

jupyter Zonova RK2 Last Checkpoint: час назад (autosaved)



Logout

File Edit View Insert Cell Kernel Help

Trusted

Python 3 (ipykernel)

Run

```
77:  learn: 0.0020868    total: 359ms  remaining: 562ms
78:  learn: 0.0020676    total: 364ms  remaining: 557ms
```

```
In [33]: print(model2.score(trainX, trainY))
```

1.0

```
In [34]: preds_2 = model2.predict(testX)
preds_2
```

```
Out[34]: array([1, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 1,
1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0,
1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0,
1, 1, 0, 0, 0, 1, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1,
0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0,
0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0,
0, 0, 0, 0, 1, 0, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 1,
1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1,
1, 0, 0, 0, 1, 0, 1]) dtype=int64)
```

```
In [35]: BAS_LR = balanced_accuracy_score(testY, preds_2)
PS_LR = precision_score(testY, preds_2)
```

```
In [36]: print("Метрика balanced_accuracy_score: {}".format(BAS_LR))
print("Метрика precision_score: {}".format(PS_LR))
```

Метрика balanced_accuracy_score: 1.0
Метрика precision_score: 1.0

```
In [37]: ac_LR = accuracy_score(testY, preds_2)
```

```
In [38]: print("Метрика accuracy_score: {}".format(ac_LR))
```

Метрика accuracy_score: 1.0