Импорт библиотек

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn import svm, tree
from sklearn.tree import DecisionTreeRegressor
from sklearn.metrics import mean_squared_error
from sklearn.metrics import r2_score
from operator import itemgetter
```

EDA

```
df = pd.read_csv("heart.csv")
```

```
df.head()
```

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

   ca  thal  target
0   2     3       0
1   0     3       0
2   0     3       0
3   1     3       0
4   3     2       0
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1025 entries, 0 to 1024
Data columns (total 14 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   age       1025 non-null   int64
```

```
 1   sex       1025 non-null   int64
 2   cp        1025 non-null   int64
 3   trestbps  1025 non-null   int64
 4   chol      1025 non-null   int64
 5   fbs       1025 non-null   int64
 6   restecg   1025 non-null   int64
 7   thalach   1025 non-null   int64
 8   exang     1025 non-null   int64
 9   oldpeak   1025 non-null   float64
 10  slope     1025 non-null   int64
 11  ca        1025 non-null   int64
 12  thal      1025 non-null   int64
 13  target    1025 non-null   int64
dtypes: float64(1), int64(13)
memory usage: 112.2 KB
```

Содержание датасета:

- age - возраст
- sex - пол (1 - мужчина, 0 - женщина)
- cp - тип боли в груди
- trestbps - артериальное давление в покое
- chol - сывороточный холестерин
- fbs - уровень сахара в крови натощак
- restecg - результаты экг в покое
- thalach - максимально достигнутая чсс
- exang - стенокардия при физической нагрузке
- oldpeak - депрессия ST сегмента при физической нагрузке
- slope - наклон пикового st сегмента при нагрузке
- ca - количество крупных сосудов (0-3), окрашенных при флюороскопии
- thal - 0 = нормальный, 1 = фиксированный дефект, 2 = обратимый дефект
- target - наличие болезни

```
df.corr()
```

```
                age       sex        cp  trestbps      chol
fbs  \
age        1.000000 -0.103240 -0.071966  0.271121  0.219823  0.121243

sex       -0.103240  1.000000 -0.041119 -0.078974 -0.198258  0.027200

cp        -0.071966 -0.041119  1.000000  0.038177 -0.081641  0.079294

trestbps   0.271121 -0.078974  0.038177  1.000000  0.127977  0.181767

chol       0.219823 -0.198258 -0.081641  0.127977  1.000000  0.026917
```

```
fbs       0.121243  0.027200  0.079294  0.181767  0.026917  1.000000
restecg  -0.132696 -0.055117  0.043581 -0.123794 -0.147410 -0.104051
thalach  -0.390227 -0.049365  0.306839 -0.039264 -0.021772 -0.008866
exang     0.088163  0.139157 -0.401513  0.061197  0.067382  0.049261
oldpeak   0.208137  0.084687 -0.174733  0.187434  0.064880  0.010859
slope    -0.169105 -0.026666  0.131633 -0.120445 -0.014248 -0.061902
ca        0.271551  0.111729 -0.176206  0.104554  0.074259  0.137156
thal      0.072297  0.198424 -0.163341  0.059276  0.100244 -0.042177
target   -0.229324 -0.279501  0.434854 -0.138772 -0.099966 -0.041164

           restecg   thalach     exang   oldpeak     slope
ca  \
age      -0.132696 -0.390227  0.088163  0.208137 -0.169105  0.271551
sex      -0.055117 -0.049365  0.139157  0.084687 -0.026666  0.111729
cp        0.043581  0.306839 -0.401513 -0.174733  0.131633 -0.176206
trestbps -0.123794 -0.039264  0.061197  0.187434 -0.120445  0.104554
chol     -0.147410 -0.021772  0.067382  0.064880 -0.014248  0.074259
fbs      -0.104051 -0.008866  0.049261  0.010859 -0.061902  0.137156
restecg   1.000000  0.048411 -0.065606 -0.050114  0.086086 -0.078072
thalach   0.048411  1.000000 -0.380281 -0.349796  0.395308 -0.207888
exang    -0.065606 -0.380281  1.000000  0.310844 -0.267335  0.107849
oldpeak  -0.050114 -0.349796  0.310844  1.000000 -0.575189  0.221816
slope     0.086086  0.395308 -0.267335 -0.575189  1.000000 -0.073440
ca       -0.078072 -0.207888  0.107849  0.221816 -0.073440  1.000000
thal     -0.020504 -0.098068  0.197201  0.202672 -0.094090  0.149014
```

```
target    0.134468  0.422895 -0.438029 -0.438441  0.345512 -0.382085


              thal    target
age       0.072297 -0.229324
sex       0.198424 -0.279501
cp       -0.163341  0.434854
trestbps  0.059276 -0.138772
chol      0.100244 -0.099966
fbs      -0.042177 -0.041164
restecg  -0.020504  0.134468
thalach  -0.098068  0.422895
exang     0.197201 -0.438029
oldpeak   0.202672 -0.438441
slope    -0.094090  0.345512
ca        0.149014 -0.382085
thal      1.000000 -0.337838
target   -0.337838  1.000000

df.describe().T

           count        mean         std    min    25%    50%    75%
max
age       1025.0   54.434146   9.072290   29.0   48.0   56.0   61.0
77.0
sex       1025.0    0.695610   0.460373    0.0    0.0    1.0    1.0
1.0
cp        1025.0    0.942439   1.029641    0.0    0.0    1.0    2.0
3.0
trestbps  1025.0  131.611707  17.516718   94.0  120.0  130.0  140.0
200.0
chol      1025.0  246.000000  51.592510  126.0  211.0  240.0  275.0
564.0
fbs       1025.0    0.149268   0.356527    0.0    0.0    0.0    0.0
1.0
restecg   1025.0    0.529756   0.527878    0.0    0.0    1.0    1.0
2.0
thalach   1025.0  149.114146  23.005724   71.0  132.0  152.0  166.0
202.0
exang     1025.0    0.336585   0.472772    0.0    0.0    0.0    1.0
1.0
oldpeak   1025.0    1.071512   1.175053    0.0    0.0    0.8    1.8
6.2
slope     1025.0    1.385366   0.617755    0.0    1.0    1.0    2.0
2.0
ca        1025.0    0.754146   1.030798    0.0    0.0    0.0    1.0
4.0
thal      1025.0    2.323902   0.620660    0.0    2.0    2.0    3.0
3.0
```

```
target    1025.0    0.513171   0.500070    0.0    0.0    1.0    1.0
1.0
```

Разделим выборку

```
y = df['target']
x = df.drop(['target'], axis = 1)

scaler = MinMaxScaler()
scaled_data = scaler.fit_transform(x)

x_train, x_test, y_train, y_test = train_test_split(scaled_data, y,
test_size = 0.2, random_state = 0)
```

Логистическая регрессия

```
lr = LogisticRegression()
lr_prediction = lr.fit(x_train, y_train).predict(x_test)
```

SVM

```
svr = svm.SVR()
svr_prediction = svr.fit(x_train, y_train).predict(x_test)
```

Дерево решений

```
dt = DecisionTreeRegressor(random_state=0)
dt_prediction = dt.fit(x_train, y_train).predict(x_test)
```

Оценка качества моделей

```
print("Linear regression: ", mean_squared_error(y_test,
lr_prediction))
print("SVM: ", mean_squared_error(y_test, svr_prediction))
print("Decision tree: ", mean_squared_error(y_test, dt_prediction))

Linear regression:  0.14146341463414633
SVM:  0.04264875882252695
Decision tree:  0.0

print("Linear regression: ", r2_score(y_test, lr_prediction))
print("SVM: ", r2_score(y_test, svr_prediction))
print("Decision tree: ", r2_score(y_test, dt_prediction))

Linear regression:  0.43305359526988374
SVM:  0.8290755207403495
Decision tree:  1.0
```

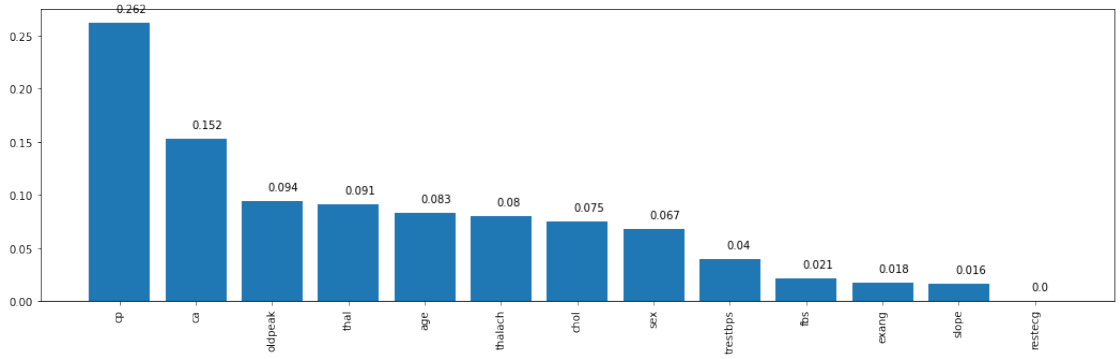
Важность признаков

```
list(zip(x.columns.values, dt.feature_importances_))
```

```
[('age', 0.08327324581520917),
 ('sex', 0.06744438766554109),
 ('cp', 0.2620208903056958),
 ('trestbps', 0.03996351167432581),
 ('chol', 0.07458044754488027),
 ('fbs', 0.0213110818816126),
 ('restecg', 0.0),
 ('thalach', 0.07967578364816712),
 ('exang', 0.017589775586969354),
 ('oldpeak', 0.09444809393622243),
 ('slope', 0.01591128449308529),
 ('ca', 0.1524340266116806),
 ('thal', 0.09134747083661057)]
```

```
def draw_feature_importances(tree_model, X_dataset, figsize=(18,5)):
    # Sorting the values of the importance of features in descending
order
    list_to_sort = list(zip(X_dataset.columns.values,
tree_model.feature_importances_))
    sorted_list = sorted(list_to_sort, key=itemgetter(1), reverse =
True)
    # Features names
    labels = [x for x,_ in sorted_list]
    # Features importance
    data = [x for _,x in sorted_list]
    # Graph output
    fig, ax = plt.subplots(figsize=figsize)
    ind = np.arange(len(labels))
    plt.bar(ind, data)
    plt.xticks(ind, labels, rotation='vertical')
    # Values output
    for a,b in zip(ind, data):
        plt.text(a-0.05, b+0.01, str(round(b,3)))
    plt.show()
    return labels, data

dt_fl, dt_fd = draw_feature_importances(dt, x)
```



Визуализация дерева решений

```
tree.plot_tree(dt)
```

```
[Text(171.00775862068966, 206.56799999999998, 'X[2] <= 0.167\nmse =
0.25\nsamples = 820\nvalue = 0.511'),
 Text(106.78965517241379, 184.824, 'X[11] <= 0.125\nmse = 0.186\
nsamples = 397\nvalue = 0.247'),
 Text(75.04137931034482, 163.07999999999998, 'X[12] <= 0.833\nmse =
0.25\nsamples = 169\nvalue = 0.503'),
 Text(46.179310344827584, 141.336, 'X[8] <= 0.5\nmse = 0.191\nsamples
= 97\nvalue = 0.742'),
 Text(23.089655172413792, 119.592, 'X[4] <= 0.435\nmse = 0.1\nsamples
= 62\nvalue = 0.887'),
 Text(11.544827586206896, 97.848, 'X[7] <= 0.195\nmse = 0.05\nsamples
= 57\nvalue = 0.947'),
 Text(5.772413793103448, 76.10399999999998, 'mse = 0.0\nsamples = 3\
nvalue = 0.0'),
 Text(17.317241379310346, 76.10399999999998, 'mse = 0.0\nsamples = 54\
nvalue = 1.0'),
 Text(34.63448275862069, 97.848, 'X[0] <= 0.677\nmse = 0.16\nsamples =
5\nvalue = 0.2'),
 Text(28.86206896551724, 76.10399999999998, 'mse = 0.0\nsamples = 4\
nvalue = 0.0'),
 Text(40.40689655172414, 76.10399999999998, 'mse = 0.0\nsamples = 1\
nvalue = 1.0'),
 Text(69.26896551724138, 119.592, 'X[7] <= 0.573\nmse = 0.25\nsamples
= 35\nvalue = 0.486'),
 Text(57.72413793103448, 97.848, 'X[3] <= 0.198\nmse = 0.109\nsamples
= 16\nvalue = 0.125'),
 Text(51.95172413793103, 76.10399999999998, 'mse = 0.0\nsamples = 2\
nvalue = 1.0'),
 Text(63.49655172413793, 76.10399999999998, 'mse = 0.0\nsamples = 14\
nvalue = 0.0'),
 Text(80.81379310344828, 97.848, 'X[12] <= 0.333\nmse = 0.166\nsamples
= 19\nvalue = 0.789'),
 Text(75.04137931034482, 76.10399999999998, 'mse = 0.0\nsamples = 3\
nvalue = 0.0'),
```

```
 Text(86.58620689655172, 76.10399999999998, 'X[9] <= 0.161\nmse =
0.059\nsamples = 16\nvalue = 0.938'),
 Text(80.81379310344828, 54.360000000000014, 'mse = 0.0\nsamples = 13\
nvalue = 1.0'),
 Text(92.35862068965517, 54.360000000000014, 'X[1] <= 0.5\nmse =
0.222\nsamples = 3\nvalue = 0.667'),
 Text(86.58620689655172, 32.615999999999985, 'mse = 0.0\nsamples = 1\
nvalue = 0.0'),
 Text(98.13103448275862, 32.615999999999985, 'mse = 0.0\nsamples = 2\
nvalue = 1.0'),
 Text(103.90344827586206, 141.336, 'X[9] <= 0.105\nmse = 0.148\
nsamples = 72\nvalue = 0.181'),
 Text(98.13103448275862, 119.592, 'X[0] <= 0.271\nmse = 0.25\nsamples
= 25\nvalue = 0.52'),
 Text(92.35862068965517, 97.848, 'mse = 0.0\nsamples = 8\nvalue =
0.0'),
 Text(103.90344827586206, 97.848, 'X[4] <= 0.255\nmse = 0.18\nsamples
= 17\nvalue = 0.765'),
 Text(98.13103448275862, 76.10399999999998, 'mse = 0.0\nsamples = 13\
nvalue = 1.0'),
 Text(109.67586206896551, 76.10399999999998, 'mse = 0.0\nsamples = 4\
nvalue = 0.0'),
 Text(109.67586206896551, 119.592, 'mse = 0.0\nsamples = 47\nvalue =
0.0'),
 Text(138.53793103448277, 163.07999999999998, 'X[9] <= 0.073\nmse =
0.054\nsamples = 228\nvalue = 0.057'),
 Text(126.99310344827586, 141.336, 'X[1] <= 0.5\nmse = 0.143\nsamples
= 58\nvalue = 0.172'),
 Text(121.22068965517241, 119.592, 'mse = 0.0\nsamples = 5\nvalue =
1.0'),
 Text(132.7655172413793, 119.592, 'X[7] <= 0.271\nmse = 0.085\nsamples
= 53\nvalue = 0.094'),
 Text(126.99310344827586, 97.848, 'mse = 0.0\nsamples = 3\nvalue =
1.0'),
 Text(138.53793103448277, 97.848, 'X[3] <= 0.142\nmse = 0.038\nsamples
= 50\nvalue = 0.04'),
 Text(132.7655172413793, 76.10399999999998, 'X[7] <= 0.615\nmse =
0.24\nsamples = 5\nvalue = 0.4'),
 Text(126.99310344827586, 54.360000000000014, 'mse = 0.0\nsamples = 2\
nvalue = 1.0'),
 Text(138.53793103448277, 54.360000000000014, 'mse = 0.0\nsamples = 3\
nvalue = 0.0'),
 Text(144.3103448275862, 76.10399999999998, 'mse = 0.0\nsamples = 45\
nvalue = 0.0'),
 Text(150.08275862068965, 141.336, 'X[4] <= 0.4\nmse = 0.017\nsamples
= 170\nvalue = 0.018'),
 Text(144.3103448275862, 119.592, 'mse = 0.0\nsamples = 145\nvalue =
0.0'),
 Text(155.8551724137931, 119.592, 'X[4] <= 0.413\nmse = 0.106\nsamples
= 25\nvalue = 0.12'),
```

```
 Text(150.08275862068965, 97.848, 'mse = 0.0\nsamples = 3\nvalue =
1.0'),
 Text(161.62758620689655, 97.848, 'mse = 0.0\nsamples = 22\nvalue =
0.0'),
 Text(235.2258620689655, 184.824, 'X[0] <= 0.573\nmse = 0.183\nsamples
= 423\nvalue = 0.759'),
 Text(199.14827586206897, 163.07999999999998, 'X[9] <= 0.573\nmse =
0.108\nsamples = 253\nvalue = 0.877'),
 Text(193.37586206896552, 141.336, 'X[7] <= 0.546\nmse = 0.088\
nsamples = 246\nvalue = 0.902'),
 Text(178.94482758620688, 119.592, 'X[11] <= 0.5\nmse = 0.243\nsamples
= 31\nvalue = 0.581'),
 Text(173.17241379310343, 97.848, 'X[5] <= 0.5\nmse = 0.188\nsamples =
24\nvalue = 0.75'),
 Text(167.4, 76.10399999999998, 'X[12] <= 0.833\nmse = 0.09\nsamples =
20\nvalue = 0.9'),
 Text(161.62758620689655, 54.360000000000014, 'mse = 0.0\nsamples =
18\nvalue = 1.0'),
 Text(173.17241379310343, 54.360000000000014, 'mse = 0.0\nsamples = 2\
nvalue = 0.0'),
 Text(178.94482758620688, 76.10399999999998, 'mse = 0.0\nsamples = 4\
nvalue = 0.0'),
 Text(184.71724137931034, 97.848, 'mse = 0.0\nsamples = 7\nvalue =
0.0'),
 Text(207.80689655172412, 119.592, 'X[3] <= 0.83\nmse = 0.049\nsamples
= 215\nvalue = 0.949'),
 Text(202.03448275862067, 97.848, 'X[3] <= 0.16\nmse = 0.036\nsamples
= 212\nvalue = 0.962'),
 Text(190.4896551724138, 76.10399999999998, 'X[10] <= 0.25\nmse =
0.125\nsamples = 34\nvalue = 0.853'),
 Text(184.71724137931034, 54.360000000000014, 'mse = 0.0\nsamples = 2\
nvalue = 0.0'),
 Text(196.26206896551724, 54.360000000000014, 'X[7] <= 0.622\nmse =
0.085\nsamples = 32\nvalue = 0.906'),
 Text(190.4896551724138, 32.615999999999985, 'X[10] <= 0.75\nmse =
0.245\nsamples = 7\nvalue = 0.571'),
 Text(184.71724137931034, 10.872000000000014, 'mse = 0.0\nsamples = 4\
nvalue = 1.0'),
 Text(196.26206896551724, 10.872000000000014, 'mse = 0.0\nsamples = 3\
nvalue = 0.0'),
 Text(202.03448275862067, 32.615999999999985, 'mse = 0.0\nsamples =
25\nvalue = 1.0'),
 Text(213.57931034482758, 76.10399999999998, 'X[12] <= 0.833\nmse =
0.017\nsamples = 178\nvalue = 0.983'),
 Text(207.80689655172412, 54.360000000000014, 'mse = 0.0\nsamples =
154\nvalue = 1.0'),
 Text(219.35172413793103, 54.360000000000014, 'X[11] <= 0.125\nmse =
0.109\nsamples = 24\nvalue = 0.875'),
 Text(213.57931034482758, 32.615999999999985, 'mse = 0.0\nsamples =
21\nvalue = 1.0'),
```

```
 Text(225.12413793103448, 32.615999999999985, 'mse = 0.0\nsamples = 3\
nvalue = 0.0'),
 Text(213.57931034482758, 97.848, 'mse = 0.0\nsamples = 3\nvalue =
0.0'),
 Text(204.9206896551724, 141.336, 'mse = 0.0\nsamples = 7\nvalue =
0.0'),
 Text(271.3034482758621, 163.07999999999998, 'X[1] <= 0.5\nmse =
0.243\nsamples = 170\nvalue = 0.582'),
 Text(236.66896551724136, 141.336, 'X[0] <= 0.594\nmse = 0.099\
nsamples = 63\nvalue = 0.889'),
 Text(230.8965517241379, 119.592, 'mse = 0.0\nsamples = 2\nvalue =
0.0'),
 Text(242.44137931034481, 119.592, 'X[7] <= 0.263\nmse = 0.075\
nsamples = 61\nvalue = 0.918'),
 Text(230.8965517241379, 97.848, 'X[4] <= 0.216\nmse = 0.25\nsamples =
6\nvalue = 0.5'),
 Text(225.12413793103448, 76.10399999999998, 'mse = 0.0\nsamples = 3\
nvalue = 1.0'),
 Text(236.66896551724136, 76.10399999999998, 'mse = 0.0\nsamples = 3\
nvalue = 0.0'),
 Text(253.98620689655172, 97.848, 'X[0] <= 0.625\nmse = 0.035\nsamples
= 55\nvalue = 0.964'),
 Text(248.21379310344827, 76.10399999999998, 'X[11] <= 0.25\nmse =
0.188\nsamples = 8\nvalue = 0.75'),
 Text(242.44137931034481, 54.360000000000014, 'mse = 0.0\nsamples = 6\
nvalue = 1.0'),
 Text(253.98620689655172, 54.360000000000014, 'mse = 0.0\nsamples = 2\
nvalue = 0.0'),
 Text(259.7586206896552, 76.10399999999998, 'mse = 0.0\nsamples = 47\
nvalue = 1.0'),
 Text(305.93793103448274, 141.336, 'X[4] <= 0.273\nmse = 0.24\nsamples
= 107\nvalue = 0.402'),
 Text(294.39310344827584, 119.592, 'X[9] <= 0.387\nmse = 0.244\
nsamples = 66\nvalue = 0.576'),
 Text(288.6206896551724, 97.848, 'X[11] <= 0.125\nmse = 0.209\nsamples
= 54\nvalue = 0.704'),
 Text(271.3034482758621, 76.10399999999998, 'X[0] <= 0.76\nmse =
0.109\nsamples = 32\nvalue = 0.875'),
 Text(265.5310344827586, 54.360000000000014, 'mse = 0.0\nsamples = 26\
nvalue = 1.0'),
 Text(277.07586206896553, 54.360000000000014, 'X[9] <= 0.065\nmse =
0.222\nsamples = 6\nvalue = 0.333'),
 Text(271.3034482758621, 32.615999999999985, 'mse = 0.0\nsamples = 2\
nvalue = 1.0'),
 Text(282.84827586206893, 32.615999999999985, 'mse = 0.0\nsamples = 4\
nvalue = 0.0'),
 Text(305.93793103448274, 76.10399999999998, 'X[11] <= 0.625\nmse =
0.248\nsamples = 22\nvalue = 0.455'),
 Text(300.1655172413793, 54.360000000000014, 'X[5] <= 0.5\nmse =
0.188\nsamples = 16\nvalue = 0.25'),
```

```
 Text(294.39310344827584, 32.615999999999985, 'mse = 0.0\nsamples =
10\nvalue = 0.0'),
 Text(305.93793103448274, 32.615999999999985, 'X[12] <= 0.5\nmse =
0.222\nsamples = 6\nvalue = 0.667'),
 Text(300.1655172413793, 10.872000000000014, 'mse = 0.0\nsamples = 2\
nvalue = 0.0'),
 Text(311.7103448275862, 10.872000000000014, 'mse = 0.0\nsamples = 4\
nvalue = 1.0'),
 Text(311.7103448275862, 54.360000000000014, 'mse = 0.0\nsamples = 6\
nvalue = 1.0'),
 Text(300.1655172413793, 97.848, 'mse = 0.0\nsamples = 12\nvalue =
0.0'),
 Text(317.48275862068965, 119.592, 'X[3] <= 0.236\nmse = 0.107\
nsamples = 41\nvalue = 0.122'),
 Text(311.7103448275862, 97.848, 'mse = 0.0\nsamples = 3\nvalue =
1.0'),
 Text(323.2551724137931, 97.848, 'X[9] <= 0.573\nmse = 0.05\nsamples =
38\nvalue = 0.053'),
 Text(317.48275862068965, 76.10399999999998, 'mse = 0.0\nsamples = 36\
nvalue = 0.0'),
 Text(329.02758620689656, 76.10399999999998, 'mse = 0.0\nsamples = 2\
nvalue = 1.0')]
```