| 0 842302 | nosis radius_mean te | exture_mean perin | | mean smoothness_me | | | | texture_worst per | erimeter_worst |
|--|--|--|---|---|--------------------|-----------|--------------------|-------------------|---------------------------|
| 842302 842517 84300903 84348301 | M 20.57 M 19.69 M 11.42 | 10.38 17.77 21.25 20.38 | 132.90 13 130.00 13 | 326.0 0.10s 203.0 0.10s 386.1 0.14s | 0.078 960 0.159 | 990 0.197 | 0.07017 0.12790 | 23.41 | 158.80 152.50 98.87 |
| 4 84358402 5 rows × 33 column | M 20.29 | 14.34 | | 297.0 0.100 | | | | | 152.20 |
| df.shape (569, 33) | | | | | | | | | |
| df.isnull().sum id diagnosis | o 0 | | | | | | | | |
| radius_mean texture_mean perimeter_mean area_mean smoothness_mear | 0 0 0 0 | | | | | | | | |
| smootnness_mear compactness_mea concavity_mean concave points_ symmetry_mean | an 0 0 | | | | | | | | |
| fractal_dimensi radius_se texture_se perimeter_se area_se | ion_mean 0 0 0 0 | | | | | | | | |
| smoothness_se compactness_se concavity_se concave points_ | 0 | | | | | | | | |
| <pre>symmetry_se fractal_dimensi radius_worst texture_worst perimeter_worst</pre> | 9 9 | | | | | | | | |
| area_worst smoothness_wors compactness_wor concavity_worst concave points_ | rst 0 t 0 | | | | | | | | |
| symmetry_worst fractal_dimensi Unnamed: 32 dtype: int64 | 0 | | | | | | | | |
| df.diagnosis.va B 357 M 212 | | | | | | | | | |
| df.hist() array([[<axessu< td=""><td>s, dtype: int64 ubplot:title={'cen</td><td>ter':'id'}>,</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></axessu<> | s, dtype: int64 ubplot:title={'cen | ter':'id'}>, | | | | | | | |
| <axessu <axessu <axessu< td=""><td>ubplot:title={'cen ubplot:title={'cen ubplot:title={'cen ubplot:title={'cen ubplot:title={'cen</td><td>nter':'texture_ nter':'perimete nter':'area_mea</td><td>_mean'}>, er_mean'}>, an'}>,</td><td></td><td></td><td></td><td></td><td></td><td></td></axessu<></axessu </axessu | ubplot:title={'cen ubplot:title={'cen ubplot:title={'cen ubplot:title={'cen ubplot:title={'cen | nter':'texture_ nter':'perimete nter':'area_mea | _mean'}>, er_mean'}>, an'}>, | | | | | | |
| [<axessı <axessı <axessı <axessı< td=""><td>ubplot:title={'cen ubplot:title={'cen ubplot:title={'cen ubplot:title={'cen</td><td>nter':'compactn nter':'concavit nter':'concave nter':'symmetry</td><td>ness_mean'}>, ty_mean'}>, points_mean'}> y_mean'}>,</td><td></td><td></td><td></td><td></td><td></td><td></td></axessı<></axessı </axessı </axessı | ubplot:title={'cen ubplot:title={'cen ubplot:title={'cen ubplot:title={'cen | nter':'compactn nter':'concavit nter':'concave nter':'symmetry | ness_mean'}>, ty_mean'}>, points_mean'}> y_mean'}>, | | | | | | |
| <axessı [<axessı <axessı <axessı< td=""><td>ubplot:title={'cen ubplot:title={'cen ubplot:title={'cen ubplot:title={'cen ubplot:title={'cen</td><td>nter':'radius_s nter':'texture_ nter':'perimete nter':'area_se'</td><td>se'}>], _se'}>, er_se'}>, '}>,</td><td>1`}>,</td><td></td><td></td><td></td><td></td><td></td></axessı<></axessı </axessı </axessı | ubplot:title={'cen ubplot:title={'cen ubplot:title={'cen ubplot:title={'cen ubplot:title={'cen | nter':'radius_s nter':'texture_ nter':'perimete nter':'area_se' | se'}>], _se'}>, er_se'}>, '}>, | 1`}>, | | | | | |
| <axessı <axessı <axessı [<axessı< td=""><td>ubplot:title={'cen ubplot:title={'cen ubplot:title={'cen ubplot:title={'cen</td><td>nter':'smoothne nter':'compactn nter':'concavit nter':'concave</td><td>ess_se'}>, ness_se'}>, ty_se'}>], points_se'}>,</td><td></td><td></td><td></td><td></td><td></td><td></td></axessı<></axessı </axessı </axessı | ubplot:title={'cen ubplot:title={'cen ubplot:title={'cen ubplot:title={'cen | nter':'smoothne nter':'compactn nter':'concavit nter':'concave | ess_se'}>, ness_se'}>, ty_se'}>], points_se'}>, | | | | | | |
| <axessı <axessı <axessı <axessı< td=""><td>ubplot:title={'cen ubplot:title={'cen ubplot:title={'cen ubplot:title={'cen ubplot:title={'cen</td><td>nter':'fractal_ nter':'radius_w nter':'texture_ nter':'perimete</td><td>_dimension_se'} worst'}>, _worst'}>, er_worst'}>],</td><td>.>,</td><td></td><td></td><td></td><td></td><td></td></axessı<></axessı </axessı </axessı | ubplot:title={'cen ubplot:title={'cen ubplot:title={'cen ubplot:title={'cen ubplot:title={'cen | nter':'fractal_ nter':'radius_w nter':'texture_ nter':'perimete | _dimension_se'} worst'}>, _worst'}>, er_worst'}>], | .>, | | | | | |
| [<axessı <axessı <axessı <axessı< td=""><td>ubplot:title={'cen ubplot:title={'cen ubplot:title={'cen ubplot:title={'cen ubplot:title={'cen</td><td>nter':'area_wor nter':'smoothne nter':'compactn nter':'concavit</td><td>rst'}>, ess_worst'}>, ness_worst'}>, ty_worst'}>,</td><td>>.</td><td></td><td></td><td></td><td></td><td></td></axessı<></axessı </axessı </axessı | ubplot:title={'cen ubplot:title={'cen ubplot:title={'cen ubplot:title={'cen ubplot:title={'cen | nter':'area_wor nter':'smoothne nter':'compactn nter':'concavit | rst'}>, ess_worst'}>, ness_worst'}>, ty_worst'}>, | >. | | | | | |
| <axessı [<axessı <axessı< td=""><td>ubplot:title={'cen ubplot:title={'cen ubplot:title={'cen ubplot:title={'cen ubplot:>, <axessub< td=""><td>nter':'symmetry nter':'fractal_ nter':'Unnamed:</td><td>y_worst'}>], _dimension_wors : 32'}>, <axess< td=""><td>st'}>, Subplot:>,</td><td></td><td></td><td></td><td></td><td></td></axess<></td></axessub<></td></axessı<></axessı </axessı | ubplot:title={'cen ubplot:title={'cen ubplot:title={'cen ubplot:title={'cen ubplot:>, <axessub< td=""><td>nter':'symmetry nter':'fractal_ nter':'Unnamed:</td><td>y_worst'}>], _dimension_wors : 32'}>, <axess< td=""><td>st'}>, Subplot:>,</td><td></td><td></td><td></td><td></td><td></td></axess<></td></axessub<> | nter':'symmetry nter':'fractal_ nter':'Unnamed: | y_worst'}>], _dimension_wors : 32'}>, <axess< td=""><td>st'}>, Subplot:>,</td><td></td><td></td><td></td><td></td><td></td></axess<> | st'}>, Subplot:>, | | | | | |
| id 500 - | radius_metæx | ture_n peair met | ter_m aae a_ns | enamothness_mea | ın | | | | |
| 100 Otextur | 200 100 - | 100 for | 100 - | 250 | | | | | |
| concave p | 250 - 250 - | 250 - 1 | 2000.45.00 | 250 - 250 mpælistneter wors | : | | | | |
| 200 | 250 | 100 | 100 | 100 | | | | | |
| 0alrea_v | www.modo.thmess.mwpoz | astnessowcas | tegrwave poir | tsowoetty wors | t | | | | |
| 2000 | 100 - 200 6 | 100 | 100 | 200 worsty wors | t | | | | |
| fractal dimer 250 0.1 | 100 - 200 6 8 2 0 0 0 0 0 0 0 0 0 0 0 1 | 100 | 100 | 200 | t | | | | |
| fractalOdimer 250 0.1 x=df.iloc[:, 2: y=df.diagnosis from sklearn.pr Scaler=Standard | 0.05 0.05 0.2 0 1 0.2 0 1 | 100 - 1 0 | 1 0.00 0. | 200 | t | | | | |
| fractalodimer 250 0.1 x=df.iloc[:, 2: y=df.diagnosis from sklearn.pr Scaler=Standard x_Scaled=Scaler x_Scaled array([[1.097 2.750 | 100 100 100 100 100 100 100 100 | 100 100 100 100 100 100 100 100 100 100 | ler #scaling | 7613, | Ċ. | | | | |
| fractalOdimer 250 0.1 x=df.iloc[:, 2: y=df.diagnosis from sklearn.pr Scaler=Standard x_Scaled=Scaler x_Scaled array([[1.097 2.750 [1.829 -0.243 [1.579 1.152 | nside manual. B2 0.05 0.05 0.2 0 1 :32].values reprocessing import dScaler() r.fit_transform(x) | 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 | 1 0.00 0. | 7613, 843 , | | | | | |
| fractalodimer 250 0.1 x=df.iloc[:, 2: y=df.diagnosis from sklearn.pr Scaler=Standard x_Scaled=Scaler x_Scaled array([[1.097 2.750 [1.829 -0.243 [1.579 1.152 , [0.702 -1.104 [1.838 1.919 | reprocessing impordscaler() r.fit_transform(x) 706398, -2.0733350 62224, 1.9370146 982061, -0.3536324 988811, 0.4561869 9255 , 0.2013912 228425, 2.0455738 454895, -0.3184091 334103, 2.3364571 908301, 2.2196352 | 1, 1.26993369 11, 1.68595471 19], 1.56650313 11], 0.67267578 6], 0.9, 1.98252415 | ler #scaling 1 0.00 0. 1 0.00 0. 1 0.00 0. | 7613, 843, 0035, 6869, 8549, | | | | | |
| fractalodimer 250 0.1 x=df.iloc[:, 2: y=df.iloc[:, 2: y=df.diagnosis from sklearn.pr Scaler=Standard x_Scaled=Scaler x_Scaled array([[1.097 2.750 [1.829 -0.243 [1.579 1.152 , [0.702 -1.104 [1.838 1.919 [-1.808 | reprocessing impordscaler() r.fit_transform(x) 706398, -2.0733350 62224, 1.9370146 682061, -0.3536324 682061, -0.3536324 682061, 0.2811899 682811, 0.4561869 62255 , 0.2013912 6228425, 2.0455738 63495, -0.3184091 6334103, 2.3364571 | 1, 1.26993369 1, 1.68595471 19], 1.56650313 1], 0.67267578 6], 0.67267578 6], 9, 1.98252415 8], 44, -1.81438851 | ler #scaling 1 0.00 0. 1 0.00 0. 1 0.00 0. | 7613, 843, 0035, 6869, 8549, | | | | | |
| fractalodimer 250 0.1 x=df.iloc[:, 2: y=df.diagnosis from sklearn.pr Scaler=Standard x_Scaled=Scaler x_Scaled array([[1.097 2.750 [1.829 -0.243 [1.579 1.152 , [0.702 -1.104 [1.838 1.919 [-1.808 -0.048 x_train.shape (426, 30) | reprocessing impordscaler() r.fit_transform(x) 706398, -2.0733350 62224, 1.9370146 982061, -0.3536324 988811, 0.4561869 9255, 0.2013912 228425, 2.0455738 454895, -0.3184091 334103, 2.3364571 908301, 2.2196352 340125, 1.2217920 | 1, 1.26993369 1, 1.68595471 19], 1.56650313 1], 0.67267578 6], 0.67267578 6], 9, 1.98252415 8], 44, -1.81438851 | ler #scaling 1 0.00 0. 1 0.00 0. 1 0.00 0. | 7613, 843, 0035, 6869, 8549, | | | | | |
| fractalodimer 250 0.1 x=df.iloc[:, 2: y=df.iloc[:, 2: y=df.diagnosis from sklearn.pr Scaler=Standard x_Scaled=Scaler x_Scaled array([[1.097 2.750 [1.829 -0.243 [1.579 1.152 , [0.702 -1.104 [1.838 1.919 [-1.808 -0.048 x_train.shape (426, 30) y_train.shape (426,) | reprocessing impordscaler() r.fit_transform(x) 706398, -2.0733350 62224, 1.9370146 982061, -0.3536324 988811, 0.4561869 9255, 0.2013912 228425, 2.0455738 454895, -0.3184091 334103, 2.3364571 908301, 2.2196352 340125, 1.2217920 | 1, 1.26993369 1, 1.68595471 19], 1.56650313 1], 0.67267578 6], 0.67267578 6], 9, 1.98252415 8], 44, -1.81438851 | ler #scaling 1 0.00 0. 1 0.00 0. 1 0.00 0. | 7613, 843, 0035, 6869, 8549, | | | | | |
| fractalodimer 250 0.1 x=df.iloc[:, 2: y=df.iloc[:, 2: y=df.diagnosis from sklearn.pr Scaler=Standard x_Scaled=Scaler x_Scaled array([[1.097 2.750 [1.829 -0.243 [1.579 1.152 , [0.702 -1.104 [1.838 1.919 [-1.808 -0.048 x_train.shape (426, 30) y_train.shape (426,) x_test.shape (143, 30) | reprocessing impordscaler() r.fit_transform(x) 706398, -2.0733350 62224, 1.9370146 982061, -0.3536324 988811, 0.4561869 9255, 0.2013912 228425, 2.0455738 454895, -0.3184091 334103, 2.3364571 908301, 2.2196352 340125, 1.2217920 | 1, 1.26993369 1, 1.68595471 19], 1.56650313 1], 0.67267578 6], 0.67267578 6], 9, 1.98252415 8], 44, -1.81438851 | ler #scaling 1 0.00 0. 1 0.00 0. 1 0.00 0. | 7613, 843, 0035, 6869, 8549, | | | | | |
| fractalodimer 250 0.1 x=df.iloc[:, 2: y=df.diagnosis from sklearn.pr Scaler=Standard x_Scaled=Scaler x_Scaled array([[1.097 2.750 [1.829 -0.243 [1.579 1.152 , [0.702 -1.104 [1.838 1.919 [-1.808 -0.048 x_train.shape (426, 30) y_train.shape (426,) x_test.shape (143, 30) y_test.shape (143,) | 100 | 1, 1.26993369 1, 1.68595471 19], 1.56650313 1], 0.67267578 6], 0.67267578 6], 9, 1.98252415 8], 44, -1.81438851 | ler #scaling 1 0.00 0. 1 0.00 0. 1 0.00 0. | 7613, 843, 0035, 6869, 8549, | | | | | |
| fractalodimer 250 0.1 x=df.iloc[:, 2: y=df.diagnosis from sklearn.pr Scaler=Standard x_Scaled=Scaler x_Scaled array([[1.097 2.750 [1.829 -0.243 [1.579 1.152 , [0.702 -1.104 [1.838 1.919 [-1.808 -0.048 x_train.shape (426, 30) y_train.shape (426,) x_test.shape (143, 30) y_train.value_d 0 266 1 160 | 100 | 1, 1.26993369 1, 1.68595471 19], 1.56650313 1], 0.67267578 6], 0.67267578 6], 9, 1.98252415 8], 44, -1.81438851 | ler #scaling 1 0.00 0. 1 0.00 0. 1 0.00 0. | 7613, 843, 0035, 6869, 8549, | | | | | |
| fractalodimer 250 o.1 x=df.iloc[:, 2:y=df.diagnosis from sklearn.pr Scaler=Standard x_Scaled=Scaler x_Scaled array([[1.097 | 100 100 100 100 100 100 100 100 | 1, 1.26993369 1, 1.68595471 19], 1.56650313 1], 0.67267578 6], 0.67267578 6], 9, 1.98252415 8], 44, -1.81438851 | ler #scaling 1 0.00 0. 1 0.00 0. 1 0.00 0. | 7613, 843, 0035, 6869, 8549, | | | | | |
| fractalodimer 250 0.1 x=df.iloc[:, 2: y=df.diagnosis from sklearn.pr Scaler=Standard x_Scaled=Scaler x_Scaled array([[1.097 | 100 100 100 100 100 100 100 100 100 100 | 1, 1.26993369 1, 1.68595471 9], 1.56650313 1], 0.67267578 6], 9, 1.98252415 8], 44, -1.81438851 | ler #scaling 1 0.00 0. 1 0.00 0. 1 0.00 0. | 7613, 843, 0035, 6869, 8549, | | | | | |
| fractalodimer 250 0.1 x=df.iloc[:, 2:y=df.diagnosis from sklearn.pr Scaler=Standard X_Scaled=Scaler X_Scaled array([[1.097 | 100 | 1, 1.26993369 1, 1.68595471 9], 1.56650313 1], 0.67267578 6], 9, 1.98252415 8], 44, -1.81438851 | ler #scaling 1 0.00 0. 1 0.00 0. 1 0.00 0. | 7613, 843, 0035, 6869, 8549, | | | | | |
| ### Train. shape (143, 30) y_train. shape (143, 30) | 100 | 1, 1.26993369 1, 1.68595471 9], 1.56650313 1], 1, 0.67267578 6], 9, 1.98252415 18], 4, -1.81438851 19]]) | 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, | 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, | | | | | |
| fractal of the property of the | 100 100 100 100 100 100 100 100 | 1, 1.26993369 11, 1.68595471 9], 1.56650313 11], 1, 0.67267578 6], 9, 1.98252415 18], 14, -1.81438851 19]]) | 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, | 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, | | | | | |
| fractalodimer 250 oliver 250 fractalodimer 250 oliver 250 from sklearn.pr 2.750 scaler=Standard 2.750 [1.829 -0.243 [1.579 1.152, [0.702 -1.104 [1.838 1.919 [-1.808 -0.048 x_train.shape (426, 30) y_train.shape (426,) x_test.shape (143,) y_train.value_0 oliver 260 160 Name: diagnosis #using kernal=r from sklearn.sv model=SvC() model.score(x_t 0.9790209790209 model.predict(x array([1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, | 100 100 100 100 100 100 100 100 | 1, 1.26993369 11, 1.68595471 9], 1.56650313 11], 1, 0.67267578 6], 9, 1.98252415 18], 14, -1.81438851 19]]) | 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, | 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, | | | | | |
| fractalodimer 250 0.1 x=df.iloc[:, 2: y=df.diagnosis from sklearn.pr scaler=standard x_Scaled=Scaler x_Scaled array([[1.097 | 100 100 100 100 100 100 100 100 | 1, 1.26993369 11, 1.68595471 9], 1.56650313 11], 1, 0.67267578 6], 9, 1.98252415 18], 14, -1.81438851 19]]) | 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, | 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, | | | | | |
| fractalo dimer 250 fractalo dimer 250 0.1 x=df.iloc[:, 2: y=df.diagnosis from sklearn.pr Scaler=Standard X_Scaled=Scaler X_Scaled array([[1.097 | 100 | 1, 1, 26993369 11, 1.26993369 11, 1.68595471 9, 1.56650313 11], 1, 1.81438851 19]]) OnTreeClassific | 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, | 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, | | | | | |
| ractalodimer 250 0.1 x=df.iloc[:, 2: y=df.diagnosis from sklearn.pr Scaler=Standard x_Scaled=Scaler x_Scaled array([[1.097 | 100 | rt StandardScal 1, 1.26993369 11, 1.68595471 91, 1.56650313 11], 0.67267578 6], 9, 1.98252415 18, 4, -1.81438851 19]]) onTreeClassification of the contraction of t | 0, 1, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, | 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, | | | | | |
| ractalodiner 250 0.1 x=df.iloc[:, 2: y=df.diagnosis from sklearn.pr scaler=Standard x_Scaled=Scaler x_Scaled array([[1.097 | 100 100 100 100 100 100 100 100 | rt StandardScal 1, 1.26993369 11, 1.68595471 91, 1.56650313 11], 0.67267578 6], 9, 1.98252415 18, 4, -1.81438851 19]]) onTreeClassification of the contraction of t | 0, 1, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, | 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, | | | | | |
| #ractalogines **Tractalogines **Tractalogines **Trom sklearn.pr **Scaler=Standard **X_Scaled=Scaler **X_Scaled=Scaler **X_Scaled=Scaler **X_Scaled **array([[1.097 | 100 Tsidinwars.d. 32 0.05 0.00 0.05 0.2 0 1 :32].values reprocessing impordscaler() r.fit_transform(x) 206398, -2.0733350 206224, 1.9370146 2082061, -0.3536324 2082061, -0.3536324 2082061, -0.3536324 2082811, 0.4561869 20255 , 0.2013912 228425, 2.0455738 234403, 2.3364571 208301, 2.2196352 234103, 2.3364571 208301, 2.2196352 2313821, -0.7512066 counts() test,y_test) 29791 x_test) 1, 1, 0, 1, 0, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, | 100 1 0 1 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 1 0 1 1 1 0 1 1 1 1 | 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, | 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, | | | | | |
| fractalodinner 250 0.1 x=df.iloc[:, 2: y=df.diagnosis from sklearn.pr Scaler=Standard x_Scaled=Scaler x_Scaled array([[1.097 2.750 [1.829 -0.243 [1.579 | 1000 TSIGN MATERIAL BY TOURS OF THE TOUR STANDS TOUR OF THE TOUR STA | 1, 1.26993369 11, 1.68595471 9], 1.56650313 11], 0.67267578 6], 9, 1.98252415 8], 4, -1.81438851 9]]) OnTreeClassific port cross_val_freeClassifier(| 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, | 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, | | | | | |