Column	3 84348301 4 84358402 5 843786 6 844359 7 84458202 8 844981 9 84501001 10 rows × 32 col	M M M M M M M M M M M M M M M M M M M	17.99 20.57 19.69 11.42 20.29 12.45 18.25 13.71 13.00 12.46	10.38 17.77 21.25 20.38 14.34 15.70 19.98 20.83 21.82 24.04	122.80 132.90 130.00 77.58 135.10 82.57 119.60 90.20 87.50 83.97	1001.0 1326.0 1203.0 386.1 1297.0 477.1 1040.0 577.9 519.8 475.9	0.11840 0.08474 0.10960 0.14250 0.10030 0.12780 0.09463 0.11890 0.12730 0.11860	0.27760 0.07864 0.15990 0.28390 0.13280 0.17000 0.10900 0.16450 0.19320 0.23960	0.30010 0.08690 0.19740 0.24140 0.19800 0.15780 0.11270 0.09366 0.18590 0.22730	concave points_mean  0.14710  0.07017  0.12790  0.10520  0.10430  0.08089  0.07400  0.05985  0.09353  0.08543
The content	df.tail()  id d  564 926424  565 926682  566 926954  567 927241  568 92751  5 rows × 32 columnumber of references	liagnosis radi M M M M B	21.56 20.13 16.60 20.60 7.76	22.39 28.25 28.08 29.33	142.00 131.20 108.30 140.10	1479.0 1261.0 858.1 1265.0	0.11100 0.09780 0.08455 0.11780	0.11590 0.10340 0.10230 0.27700	0.24390 0.14400 0.09251 0.35140	0.13890 0.09791 0.05302 0.15200
Column	#describe ma df.describe(  count 5.690000 mean 3.037183 std 1.250206 min 8.670000 25% 8.692180 50% 9.060240 75% 8.813129 max 9.113205	id radius_ 0e+02 569.00 3e+07 14.13 5e+08 3.53 0e+03 6.90 0e+05 11.70 0e+05 13.3 0e+06 15.70 5e+08 28.1	mean texture  00000 569.0  27292 19.0  24049 4.0  81000 9.0  00000 16.0  70000 18.0  80000 21.0	289649 9 301036 2 710000 4 170000 7 340000 8	59.000000 569 91.969033 654 24.298981 351 43.790000 143 75.170000 420 36.240000 551 04.100000 782	0.000000 0.889104 0.914129 0.500000 0.300000 0.100000	569.000000 0.096360 0.014064 0.052630 0.086370 0.095870 0.105300	569.000000 0.104341 0.052813 0.019380 0.064920 0.092630 0.130400	569.000000 5 0.088799 0.079720 0.000000 0.029560 0.061540 0.130700	ints_mean 59 669.000000 0.048919 0.038803 0.000000 0.020310 0.033500 0.074000
The content	df.info() <class #="" 'panda="" <="" column="" columns="" data="" rangeindex:="" state="" td=""><td>as.core.fra 569 entries (total 32  is mean _mean er_mean an ess_mean ty_mean points_mea y_mean _dimension_ se _se er_se ess_se ty_se points_se y_se _dimension_ worst _worst er_worst rst ess_worst rst ess_worst rst ess_worst ty_worst points_wor y_worst _dimension_ t64(30), in : 142.4+ KB</td><td>me.DataFram (, 0 to 568 (columns): Non 569 569 569 569 569 569 569 569 569 569</td><td>Null Count non-null non-null</td><td>int64 object float64 float64</td><td></td><td></td><td></td><td></td><td></td></class>	as.core.fra 569 entries (total 32  is mean _mean er_mean an ess_mean ty_mean points_mea y_mean _dimension_ se _se er_se ess_se ty_se points_se y_se _dimension_ worst _worst er_worst rst ess_worst rst ess_worst rst ess_worst ty_worst points_wor y_worst _dimension_ t64(30), in : 142.4+ KB	me.DataFram (, 0 to 568 (columns): Non 569 569 569 569 569 569 569 569 569 569	Null Count non-null	int64 object float64					
The property of the content of the c	texture_mean perimeter_mean smoothness_meconcavity_mean concave points symmetry_mean fractal_dimentadius_se texture_se perimeter_se area_se smoothness_se concavity_se concave points symmetry_se fractal_dimentadius_worst texture_worst perimeter_worst smoothness_we concavity_worst smoothness_we concavity_worst symmetry_worst symmetry_wo	an ean mean an ts_mean n nsion_mean  e se ts_se nsion_se t rst orst worst rst ts_worst rst ts_worst st nsion_worst	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0							
The process of the control of the co	<pre>df.replace({ df.head()</pre>	liagnosis radi  1  1  1  1	us_mean text  17.99  20.57  19.69  11.42	ure_mean peri 10.38 17.77 21.25 20.38	imeter_mean a 122.80 132.90 130.00 77.58	1001.0 1326.0 1203.0 386.1	0.11840 0.08474 0.10960 0.14250	0.27760 0.07864 0.15990 0.28390	0.3001 0.0869 0.1974 0.2414	0.14710 0.07017 0.12790 0.10520
Part	diagnosis  0 2.654 1 3.683 2 rows × 31 columbreast_cance	id rad  4382e+07  1805e+07  umns  r=df.drop('	lius_mean tex 12.146524 17.462830	17.914762 21.604906	78.075406	462.790196	0.092478	0.080085	0.046058	0.025717
Marie   Mari	0 1 1 1 2 1 3 1 4 1 5 rows × 31 colu	17.99 20.57 19.69 11.42 20.29	10.38 17.77 21.25 20.38 14.34	122.8 132.9 130.0 77.5	0 1001.0 0 1326.0 0 1203.0 8 386.1	0.11 0.08 0.10 0.14	840 0.2 4474 0.6 960 0.2 250 0.2	27760 0.30 07864 0.08 15990 0.19 28390 0.24	points_mear 0.14710 669 0.07017 74 0.12790 14 0.10520	n symmetry_r 0 0.0.7 0.0.0 0 0.00 0 0.00 0 0.00 0 0.00
X_train, X_test, y_train, y_test=train_test_split(X, y, test_size=0.2, random_state=2, stratify=y)		0.57 9.69 1.42	17.77 21.25 20.38 14.34	132.90 130.00 77.58 135.10	0 1326.0 0 1203.0 8 386.1	0. 0. 0.	.08474 .10960 .14250			
Accuracy on training data: 0.000100700100700	3 12 4 20 564 22 565 20 566 10 567 20 568 compacts 0 1 2 3 4 564 565 566 567 568 fractal 0 1 2 2 3 4 564 565 566 567 568 perimete 0 1 2 2 3 4 564 565 566 567 568 fractal 0 1 2 2 3 4 564 565 566 567 568 fractal 0 1 2 2 3 4 564 565 566 567 568 fractal 0 1 2 2 3 4 564 565 566 567 568 fractal 0 1 2 1 3 1 4 1. 564 1 565 1 566 567 568 fractal 0 1 2 1 3 1 4 1. 564 1 565 1 566 567 568 fractal 0 1 1 2 1 3 1 4 1. 564 1 565 1 566 567 568 fractal 0 1 1 2 1 3 1 4 1. 564 1 565 1 566 567 568 fractal 0 1 1 2 1 3 1 4 1. 564 1 565 1 566 1 567 568 fractal 0 1 1 2 1 3 1 4 1. 564 1 565 1 566 1 567 568 fractal 0 1 1 1 2 1 3 1 4 1. 564 1 565 1 566 1 567 568 fractal 0 1 1 1 2 1 3 1 4 1. 564 1 565 1 566 1 567 568 fractal 0 1 1 1 2 1 3 1 4 1. 564 1 565 1 566 1 567 568 fractal 0 1 1 1 2 1 3 1 4 1. 564 1 565 1 566 1 567 568 fractal 0 1 1 1 2 1 3 1 4 1. 564 1 565 1 566 1 567 568 fractal 0 1 1 1 2 1 3 1 4 1. 564 1 565 1 566 1 567 568 fractal 0 1 1 1 2 1 3 1 4 1. 564 1 565 1 566 1 567 568 fractal 0 1 1 2 1 3 1 4 1. 564 1 565 1 566 1 567 568 fractal 0 1 1 2 1 3 1 4 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	1.56 0.13 6.60 0.60 7.76  ness_mean 0.27760 0.07864 0.15990 0.128390 0.13280 0.11590 0.10340 0.10230 0.27700 0.04362	22.39 28.25 28.08 29.33 24.54  CONCAVITY_M	131.20 108.30 140.10 47.92 ean concave 010 690 740 140 800 390 400 251 140 000 radius_worst 25.380 24.990 23.570 14.910 22.540 23.690 18.980 25.740 9.456 smoothness_worst 0.2654 0.1625 0.06 0.2430 0.2575 0.1625 0.06 0.2216 0.1628 0.1418 0.2650 0.0000 e: int64		0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0	11100 .09780 .08455 .11780 .05263 mean \2419 1812 .2069 .2597 1809       	atify=y)	Warning: lbfg	s failed to
Accuracy on training data: 0.9604395604395605  #accurcy of testing data X_test_prediction=model.predict(X_test) testing_data_accuracy_score(X_test_prediction,y_test) print('Accuracy on test data: ', testing_data_accuracy)  Accuracy on test data: 0.9385964912280702  #Making a Predictive system #changing the input_data to a numpy array input_data1=(20.29, 14.34,135.1,1297, 0.1003, 0.1328, 0.198, 0.1043, 0.1809, 0.05883, 0.7572, 0.7813, 5.438, 94.44, 0.01149, 0.02461, 0.01 input_data_numpy_array1=np.asarray(input_data1)  #reshape the np array as we are predicting for one instance input_data_reshaped1=input_data_numpy_array1.reshape(1, -1) input_data_reshaped1  array([[2.029e+01, 1.434e+01, 1.351e+02, 1.297e+03, 1.003e-01, 1.328e-01,	3 1: 4 20 564 2: 565 2: 566 1: 567 2: 568	1.56 0.13 6.60 0.13 6.60 0.60 0.60 7.76  ness_mean 0.27760 0.07864 0.15990 0.27760 0.02390 0.13280 0.11590 0.10340 0.10230 0.27700 0.04362	22.39 28.25 28.08 29.33 24.54  Concavity_m 0.36 0.08 0.19 0.24 0.14 0.09 0.35 0.06  mean 7871 7871 7871 7883 78667 79744 78883 79744 78884 79744 79744 79744 79744 79744 79744 79744 79745 79744 .	131.20 108.30 140.10 47.92 ean concave 010 690 740 140 800 396 400 251 140 000 radius_worst 25.38( 24.990 23.57( 14.91( 25.45( 23.690 18.98( 25.74( 9.45( smoothness_) 0.2 0.6 0.2 0.6 0.2 0.6 0.2 0.6 0.2 0.6 0.2 0.6 0.2430 0.2575 0.1625 0.2216 0.1628 0.1418 0.2650 0.0000  concave 0.1 0.1 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2		0   0   0   0   0   0   0   0   0   0	11100 09780 08455 11780 05263 mean \ 2419 1812 2069 2597 1809  1726 1752 1590 2397 1587 2001 2011 2012 2013 2014 2014 2015 2015 2016 2017 2	2,0.7813,5.438,9	4.44,0.01149,	0.02461,0.0