	19 cond 20 symi 21 frad 22 rad: 23 tex 24 per: 25 area 26 smod 27 com 28 cond 29 cond 30 symi 31 frad 32 Unna	ius_worst ture_worst imeter_wo a_worst othness_w pactness_ cavity_wo cave poir metry_wor ctal_dime amed: 32 float64(3	se _se e nts_se ension_se t st orst worst _worst orst nts_worst rst ension_worst	0 non-null	float64			
<pre>In [6]: Out[6]:</pre>	# to che		nnamed columr	1				
In [7]:	<pre># to rem df.drop(# to rem</pre>	ove the during the overthe in		<i>is not requi</i> , inplace= Tr u	•	1)		
In [9]:	# to ched f.colum Index(['column') 'column')	"id", axis ck if col ns diagnosis area_mear concave p radius_se compactne fractal_c	s=1,inplace=1 lumns are rem s', 'radius_m n', 'smoothne points_mean', e', 'texture_ ess_se', 'cor dimension_se'	moved mean', 'textuess_mean', 'cometry_metry_metry_se', 'perimencavity_se', 'radius_wo	compactness_m nean', 'fract eter_se', 'ar 'concave poi orst', 'textu	nean', 'cond :al_dimensio :ea_se', 'sm !nts_se', 's !re_worst',	avity_mean', n_mean', noothness_se',	
Out[10]:	<pre># to che type(df. pandas.co # to seg 1 = list print(1)</pre>	perimeter compactne symmetry ype='objeck the da columns) ore.index regate the da columns	r_worst', 'aress_worst', 'aress_worst', 'fracect') atatype of contacts xes.base.Index the columns (mans)	rea_worst', ' concavity_wo actal_dimensi olumns ex	smoothness_worst', 'concaton_worst'],	vorst', ave points_w		smoothness_mea
In [12]:	n', 'commension_i tness_se s_worst', worst', t']	pactness_ mean', 'r ', 'conca , 'textur 'concavit	_mean', 'conc radius_se', ' avity_se', 'c re_worst', 'p	cavity_mean', texture_se', concave point perimeter_wor concave point	'concave po 'perimeter_ :s_se', 'symm st', 'area_w :s_worst', 's	pints_mean', _se', 'area_ netry_se', ' worst', 'smo	'symmetry_me se', 'smoothne fractal_dimen othness_worst	an', 'fractal_di ess_se', 'compac sion_se', 'radiu ', 'compactness_ _dimension_wors
In [13]:	print(fe	_se = 1[1 _worst = atures_me _mean', '	12:23] 1[23:] ean) 'texture_mear					an', 'compactnes imension_mean']
	'concave	e_se', 'p points_s atures_wo	perimeter_se' se', 'symmetr orst)	ry_se', 'frac	ctal_dimensio	on_se', 'rad	ius_worst', '	<pre>'concavity_se', texture_worst'] oncavity_worst',</pre>
In [16]: Out[16]:	<pre>'concave # to get df.head(</pre>	points_w first 5)	vorst', 'symn	netry_worst', a <i>Frame</i>	'fractal_di	mension_wor	st']	s_mean concavity_mea
	1 2 3	M M M	20.57 17 19.69 21 11.42 20	7.77 13. 1.25 13. 0.38 7	2.80 1001.0 2.90 1326.0 0.00 1203.0 7.58 386.1 5.10 1297.0	0.00 0.10 0.10	8474 C 0960 C 4250 C	0.27760 0.300 0.07864 0.086 0.15990 0.197 0.28390 0.241 0.13280 0.198
In [18]:	n df["diag array(['l	nosis"].u M', 'B'], ualise th		patients of	Malignant a			gnant, B= Benig
	<matplot< td=""><td></td><td>subplots.Ax</td><td>· ·</td><td></td><td>570></td><td></td><td></td></matplot<>		subplots.Ax	· ·		570>		
	150 - 100 - 50 -	М	diagnosis	В				
In [19]: Out[19]:	df["diag B 357 M 212	nosis"].\	t of values (value_counts(()				
In [20]: Out[20]:	# to che df.shape (569, 31)	hape (rows ar	nd columns) i	in the DataFr	rame		
In [21]: Out[21]:	rad	ibe() lius_mean	texture_mean pe	erimeter_mean 569.000000	area_mean smo	othness_mean 569.000000 0.096360	569.00000	
	std min 25% 50% 75%	14.127292 3.524049 6.981000 11.700000 13.370000 15.780000 28.110000	19.289649 4.301036 9.710000 16.170000 18.840000 21.800000 39.280000	24.298981	654.889104 351.914129 143.500000 420.300000 551.100000 782.700000	0.096360 0.014064 0.052630 0.086370 0.095870 0.105300 0.163400	0.10434: 0.05281: 0.01938(0.06492(0.09263(0.13040(0.34540(3 0.079720 0 0.000000 0 0.029560 0 0.061540 0 0.130700
In [22]: Out[22]:	<pre>8 rows × 30 # len fu len(df.c)</pre>	columns	39.280000 o check the I			J.163400	0.345400	o.426800
In [23]: Out[23]:			Plot -relation amo	ong the data	features			
×1•	pe	radius_mea texture_mea erimeter_mea area_mea	an 1.000000 an 0.323782 an 0.997855 an 0.987357	0.323782 1.000000 0.329533 0.321086 -0.023389	0.997855 0.329533 1.000000 0.986507 0.207278	0.987357 0.321086 0.986507 1.000000 0.177028	0.170581 -0.023389 0.207278 0.177028 1.000000	0.506124 0.236702 0.556936 0.498502 0.659123
	compa co concave syı	actness_mea ncavity_mea e points_mea mmetry_mea nension_mea	an 0.506124 an 0.676764 an 0.822529 an 0.147741 an -0.311631	0.236702 0.302418 0.293464 0.071401 -0.076437	0.556936 0.716136 0.850977 0.183027 -0.261477	0.498502 0.685983 0.823269 0.151293 -0.283110	0.659123 0.521984 0.553695 0.557775 0.584792	1.000000 0.883121 0.831135 0.602641 0.565369
		radius_s texture_s perimeter_s area_s moothness_s mpactness_s concavity_s	-0.097317 se	0.275869 0.386358 0.281673 0.259845 0.006614 0.191975 0.143293	0.691765 -0.086761 0.693135 0.744983 -0.202694 0.250744 0.228082	0.732562 -0.066280 0.726628 0.800086 -0.166777 0.212583 0.207660	0.301467 0.068406 0.296092 0.246552 0.332375 0.318943 0.248396	0.497473 0.046205 0.548905 0.455653 0.135299 0.738722 0.570517
	fractal_	concavity_s ave points_s symmetry_s dimension_s radius_wors texture_wors	se 0.376169 se -0.104321 se -0.042641 st 0.969539	0.143293 0.163851 0.009127 0.054458 0.352573 0.912045	0.228082 0.407217 -0.081629 -0.005523 0.969476 0.303038	0.207660 0.372320 -0.072497 -0.019887 0.962746 0.287489	0.248396 0.380676 0.200774 0.283607 0.213120 0.036072	0.570517 0.642262 0.229977 0.507318 0.535315 0.248133
	smoo compa co	area_wors area_wors othness_wors actness_wors ncavity_wors	st 0.941082 st 0.119616 st 0.413463 st 0.526911	0.358040 0.343546 0.077503 0.277830 0.301025 0.295316	0.970387 0.941550 0.150549 0.455774 0.563879 0.771241	0.959120 0.959213 0.123523 0.390410 0.512606 0.722017	0.238853 0.206718 0.805324 0.472468 0.434926 0.503053	0.590210 0.509604 0.565541 0.865809 0.816275 0.815573
In [24]:	fractal_dim 30 rows × 3	e the siz	st 0.007066	0.105008 0.119205	0.189115 0.051019	0.143570 0.003738	0.394309 0.499316	0.510223 0.687382
Out[24]:	<pre>sns.heat <matplot. pre="" r="" te<=""></matplot.></pre>	map(corr) lib.axes. radius_mean - exture_mean - meter_mean -	subplots.Ax				related featu	res)
	compac conc concave p symi fractal_dime	area_mean - hness_mean - ctness_mean - cavity_mean - points_mean - metry_mean - rasion_mean - radius_se - texture_se - perimeter_se - area_se -				١	- 0.8 - 0.6 - 0.4	
	com (conca s fractal_di ! te	oothness_se - pactness_se - concavity_se - ve points_se - ymmetry_se - imension_se - radius_worst - exture_worst - area_worst -			×		- 0.2 - 0.0	
	compac conc concave p symi	hness_worst - tness_worst - cavity_worst - points_worst - metry_worst - ension_worst -	idius_mean - iture_mean - ieter_mean - area_mean - ness_mean - avity_mean - avity_mean - avity_mean - avity_mean - avity_mean -	radius se texture se perimeter se area se noothness se pactness se contactness se	swe points se - symmetry se - dimension se - radius worst - texture worst - area worst - othness worst -	y worst - ts worst - v wor	0.2	
In [25]: In [26]: Out[26]:		e numerio nosis"] =	c vales to the diagnos	tactal dimension of the categorical map({"M	al variables	(M= Maligna	nean compactness) s_mean concavity_mea
	diagnos	ie radine i		-	ican arca_mcan	31110011111633_11	ican compactness	D.27760 0.300
	diagnos 0 1 2 3 4	1 1 1 1	19.69 21 11.42 20	7.77 13. 1.25 13. 0.38 7	2.80 1001.0 2.90 1326.0 0.00 1203.0 7.58 386.1 5.10 1297.0	0.00 0.10 0.10	8474 C 0960 C 4250 C	0.07864 0.086 0.15990 0.197 0.28390 0.241 0.13280 0.198
In [27]: Out[27]: In [28]:	0 1 2 3 4 5 rows × 31 df["diag array([1, # to dro X = df.d	1 1 1 1 1 1 columns nosis"].u , 0], dty p diagnos rop("diag	20.57 17 19.69 21 11.42 20 20.29 14	7.77 13: 1.25 13: 0.38 7' 1.34 13:	2.90 1326.0 0.00 1203.0 7.58 386.1	0.00 0.10 0.10	8474 C 0960 C 4250 C	0.15990 0.197 0.28390 0.243
Out[27]:	0 1 2 3 4 5 rows × 31 df["diag array([1, # to dro X = df.d X.head() radius_ 0 1	1 1 1 1 1 1 columns nosis"].u , 0], dty p diagnos rop("diag	20.57 17 19.69 21 11.42 20 20.29 14 unique() ype=int64) sis column gnosis", axis=	7.77 13. 1.25 13. 1.38 7 1.34 13. =1) eter_mean area_r 122.80 10 132.90 13	2.90 1326.0 0.00 1203.0 7.58 386.1 5.10 1297.0	0.00 0.11 0.14	8474 C 0960 C 4250 C	0.15990 0.197 0.28390 0.241 0.13280 0.198
Out[27]: In [28]: Out[28]: In [29]:	0 1 2 3 4 5 rows × 31 df["diag array([1 # to dro X = df.d X.head() radius_ 0 1 2 3 4 5 rows × 30 y = df["y.head()	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	20.57 17 19.69 21 11.42 20 20.29 14 unique() ype=int64) sis column gnosis", axis= ure_mean perime 10.38 17.77 21.25 20.38 14.34	7.77 13: 1.25 13: 0.38 7 1.34 13: 1.22.80 10 1.32.90 13 1.30.00 13 77.58 3	2.90 1326.0 0.00 1203.0 7.58 386.1 5.10 1297.0 mean smoothnes 001.0 326.0	0.01 0.11 0.11 0.11 0.11840 0.08474	actness_mean cor 0.27760 0.07864	0.15990 0.197 0.28390 0.241 0.13280 0.198 ncavity_mean conpoints_r 0.3001 0.1 0.0869 0.0
Out[27]: In [28]: Out[28]:	0 1 2 3 4 5 rows × 31 df["diag array([1, # to dro X = df.d X.head() radius_ 0 1 2 3 4 5 rows × 30 y = df[" y.head() 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1 1 columns nosis"]. U , 0], dty p diagnos rop("diag mean textu 17.99 20.57 19.69 11.42 20.29 0 columns diagnosis diagnosis agnosis, regate th	20.57 17 19.69 21 11.42 20 20.29 14 unique() ype=int64) sis column gnosis", axis= ure_mean perime 10.38 17.77 21.25 20.38 14.34 dtype: int64	7.77 13: 1.25 13: 0.38 7 1.34 13: 1.34 13: 1.32.90 13: 1.30.00 13: 77.58 3: 1.35.10 13:	2.90 1326.0 0.00 1203.0 7.58 386.1 5.10 1297.0 mean smoothnes 001.0 326.0 203.0 386.1 297.0	0.00 0.11 0.11 0.11 0.11 0.11 0.11 0.11	actness_mean col 0.27760 0.07864 0.15990 0.28390 0.13280	0.15990 0.1970 0.28390 0.2410 0.13280 0.1980 0.3001 0.1 0.0869 0.0 0.1974 0.1 0.2414 0.1
Out[27]: In [28]: Out[28]: Out[29]: In [30]: In [31]: Out[31]:	0 1 2 3 4 5 rows × 31 df["diag array([1 # to dro X = df.d X.head() radius_ 0 1 2 3 4 5 rows × 30 y = df[" y.head() 0 1 1 1 1 1 1 Name: dia # to seg model = from skl X_train, df.shape (569, 31	1 1 1 1 1 1 1 1columns nosis"].u , 0], dty p diagnos rop("diag mean textu 17.99 20.57 19.69 11.42 20.29 0 columns diagnosis diagnosis agnosis, regate th 30%). earn.mode X_test,y_	20.57 17 19.69 21 11.42 20 20.29 14 unique() ype=int64) sis column gnosis", axis= ure_mean perime 10.38 17.77 21.25 20.38 14.34 dtype: int64	7.77 13: 1.25 13: 1.38 7: 1.34 13: 1.34 13: 1.32.90 13: 1.30.00 13: 77.58 3: 1.35.10 13: 1	2.90 1326.0 0.00 1203.0 7.58 386.1 5.10 1297.0 mean smoothnes 001.0 326.0 203.0 386.1 297.0	0.00 0.11 0.11 0.11 0.11 0.11 0.11 0.11	actness_mean col 0.27760 0.07864 0.15990 0.28390 0.13280	0.15990 0.195 0.28390 0.241 0.13280 0.198 0.3001 0.1 0.0869 0.0 0.1974 0.1 0.2414 0.1 0.1980 0.1
Out[27]: In [28]: Out[28]: In [29]: Out[29]: In [31]: Out[31]: In [32]: Out[32]: Out[32]:	0 1 2 3 4 5 rows × 31 df["diag array([1] # to dro X = df.d X.head() radius_ 0 1 2 3 4 5 rows × 30 y = df[" y.head() 0 1 1 2 1 3 1 4 1 Name: dia # to seg model = from skl X_train, df.shape (569, 31 X_train. (398, 30 X_test.s (171, 30	1 1 1 1 1 1 1 1 1 1 1 1 columns nosis"]. , 0], dty p diagnos rop("diag mean textu 17.99 20.57 19.69 11.42 20.29 0 columns diagnosis diagnosis agnosis, regate th 30%). earn.mode X_test,y_) shape) hape)	20.57 17 19.69 21 11.42 20 20.29 14 unique() ype=int64) sis column gnosis", axis= ure_mean perime 10.38 17.77 21.25 20.38 14.34 dtype: int64 the dataset in el_selection	7.77 13: 1.25 13: 1.38 7: 1.34 13: 1.34 13: 1.32.90 13: 1.30.00 13: 77.58 3: 1.35.10 13: 1	2.90 1326.0 0.00 1203.0 7.58 386.1 5.10 1297.0 mean smoothnes 001.0 326.0 203.0 386.1 297.0	0.00 0.11 0.11 0.11 0.11 0.11 0.11 0.11	actness_mean col 0.27760 0.07864 0.15990 0.28390 0.13280	0.15990 0.195 0.28390 0.241 0.13280 0.198 0.3001 0.1 0.0869 0.0 0.1974 0.1 0.2414 0.1 0.1980 0.1
Out[27]: In [28]: Out[28]: Out[28]: In [30]: Out[29]: In [31]: Out[31]: In [32]: Out[32]: In [33]: Out[33]: Out[33]: Out[34]: Out[34]: Out[35]:	0 1 2 3 4 5 rows × 31 df["diag array([1 # to dro X = df.d X.head() radius_ 0 1 2 3 4 5 rows × 30 y = df[" y.head() 0 1 1 2 1 3 1 4 1 Name: dia # to seg model = from skl X_train, df.shape (569, 31 X_train, (398, 30 X_test.s (171, 30 y_train. (398,) y_test.s	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	20.57 17 19.69 21 11.42 20 20.29 14 unique() ype=int64) sis column gnosis", axis= ure_mean perime 10.38 17.77 21.25 20.38 14.34 dtype: int64 the dataset in el_selection	7.77 13: 1.25 13: 1.38 7: 1.34 13: 1.34 13: 1.32.90 13: 1.30.00 13: 77.58 3: 1.35.10 13: 1	2.90 1326.0 0.00 1203.0 7.58 386.1 5.10 1297.0 mean smoothnes 001.0 326.0 203.0 386.1 297.0	0.00 0.11 0.11 0.11 0.11 0.11 0.11 0.11	actness_mean col 0.27760 0.07864 0.15990 0.28390 0.13280	0.15990 0.195 0.28390 0.241 0.13280 0.198 0.3001 0.1 0.0869 0.0 0.1974 0.1 0.2414 0.1 0.1980 0.1
Out[27]: In [28]: Out[28]: Out[28]: Out[29]: In [30]: Out[31]: Out[31]: Out[32]: In [32]: Out[32]: Out[32]: Out[33]: Out[33]: Out[34]: Out[34]: Out[34]: Out[34]: Out[36]:	0 1 2 3 4 5 rows × 31 df["diag array([1 # to dro X = df.d X.head() radius_ 0 1 2 3 4 5 rows × 30 y = df[" y.head() 0 1 1 2 1 3 1 4 1 Name: dia # to seg model = from skl X_train, df.shape (569, 31 X_train, df.shape (569, 31 X_train, df.shape (569, 31) X_train. (398, 30) X_test.s (171, 30) y_train. (398,) y_train. radiu 184 1 rows × 30	1 1 1 1 1 1 columns nosis"]. nosis"	20.57 17 19.69 21 11.42 20 20.29 14 unique() ype=int64) sis column gnosis", axis= ure_mean perime 10.38 17.77 21.25 20.38 14.34 s"] dtype: int64 he dataset in el_selection _train, y_test	7.77 13: 1.25 13: 1.38 7 1.34 13: 1.34 13: 1.22.80 10: 1.32.90 13: 1.30.00 13: 77.58 3: 1.35.10 13: meter_mean area import train t = train_tes 98.92	2.90 1326.0 0.00 1203.0 7.58 386.1 5.10 1297.0 mean smoothnes 001.0 326.0 203.0 386.1 297.0 a_mean smoothi 710.6	0.00 0.11 0.11 0.11 0.11 0.11 0.11 0.11	actness_mean cor 0.27760 0.07864 0.15990 0.28390 0.13280	0.15990 0.197 0.28390 0.243 0.13280 0.198 0.3001 0.1 0.0869 0.0 0.1974 0.1 0.2414 0.1 0.1980 0.1 0.05375 concavity_mean points_t 0.005375
Dut[27]: In [28]: Out[28]: Out[28]: In [30]: Out[31]: In [32]: Out[32]: In [32]: Out[32]: In [33]: Out[33]: In [36]: Out[36]: In [37]:	0 1 2 3 4 5 rows × 31 df["diag array([1 # to dro X = df.d X.head() radius_ 0 1 2 3 4 5 rows × 30 y = df[" y.head() 0 1 1 2 1 3 1 4 1 Name: dia # to seg model = from skl X_train, df.shape (569, 31 X_train, (398, 30 X_test.s (171, 30 y_train. (398,) y_test.s (171,) X_train. radiu 184 1 rows × 30 Standard deviation has nega # to stal from skl ss = Stal	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	20.57 17 19.69 21 11.42 20 20.29 14 unique() ype=int64) sis column gnosis", axis= ure_mean perime 10.38 17.77 21.25 20.38 14.34 s"] dtype: int64 he dataset in el_selection _train, y_test the features processing in the selection _train, y_test the features processing in ler()	### ### ### ### ### ### ### ### ### ##	2.90 1326.0 0.00 1203.0 7.58 386.1 5.10 1297.0 mean smoothnes 001.0 326.0 203.0 386.1 297.0 a_mean smooth 710.6 in such a mar s the data. S	0.00 0.11 0.11 0.11 0.11 0.11 0.11 0.11	actness_mean col 0.27760 0.07864 0.15990 0.28390 0.13280 mpactness_mean col 0.1052	0.15990 0.1970 0.28390 0.2470 0.13280 0.1980 0.3001 0.1 0.0869 0.0 0.1974 0.1 0.2414 0.1 0.1980 0.1 1 = 70% testing
Out [27]: In [28]: Out [28]: Out [29]: Out [29]: In [30]: In [31]: Out [31]: In [32]: Out [32]: In [32]: Out [32]: In [36]: Out [36]: In [37]: In [37]: In [37]: In [38]: In [38]: In [38]:	0 1 2 3 4 5 rows × 31 df["diag array([1, # to dro X = df.d X.head() radius_ 0 1 2 3 4 5 rows × 30 y = df[" y.head() 0 1 1 2 1 3 1 4 1 Name: dia # to seg model = from skl X_train, df.shape (569, 31 X_train. (398, 30 X_test.s (171, 30 y_train. (398,) y_test.s (171,) X_train. radiu 184 1 rows × 30 Standard deviatio has nega # to staf from skl ss = Sta X_train. radiu 184 1 rows × 30 Standard deviatio has nega # to staf from skl ss = Sta X_train. radiu 184 1 rows × 30 Standard deviatio has nega	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	20.57 17 19.69 21 11.42 20 20.29 12 unique() ype=int64) sis column gnosis", axis= ure_mean perime 10.38 17.77 21.25 20.38 14.34 s"] dtype: int64 he dataset in el_selection _train, y_test the features processing in ler() _transform(X_ sform(X_test) 27.41, 0.24845 44, 0.41763	meter_mean area_r 122.80 10 132.90 13 130.00 13 77.58 3 135.10 13 meter_mean area 122.80 10 132.90 13 130.00 13 77.58 3 135.10 13 meter_mean area simport train t = train_tes sof dataset mport Standar _train) 2699, 0.2710 394], 5846, 1.5625 3432],	2.90 1326.0 0.00 1203.0 7.58 386.1 5.10 1297.0 mean smoothnes 001.0 326.0 203.0 386.1 297.0 a_mean smooths rtst_split st_split(x,y, dtest_split(x,y, data she data. She data	0.00 0.11 0.11 0.11 0.11 0.11 0.11 0.11	actness_mean col 0.27760 0.07864 0.15990 0.28390 0.13280 chas mean as cion is useful	0.15990 0.1970 0.28390 0.243 0.13280 0.1980 0.3001 0.1 0.0869 0.0 0.1974 0.1 0.2414 0.1 0.1980 0.1 0.05375 0 and standard
Out [27]: In [28]: Out [28]: Out [29]: Out [29]: In [30]: In [31]: Out [31]: In [32]: Out [32]: In [32]: Out [32]: In [36]: Out [36]: In [37]: In [37]: In [37]: In [38]: In [38]: In [38]:	0 1 2 3 4 5 rows × 31 df ["diag array([1] # to dro X = df.d X.head() radius_ 0 1 2 3 4 5 rows × 30 y = df ["y.head() 0 1 1 2 1 3 1 4 1 Name: dia # to seg model = from skl X_train, df.shape (569, 31 X_train. (398, 30 X_test.s (171, 30 y_train. (398,) y_test.s (171,) X_train. radiu 184 1 rows × 30 Standard deviation has nega # to sta from skl ss = Sta X_train. radiu 184 1 rows × 30 Standard deviation has nega # to sta from skl ss = Sta X_train X_test = X_train x_test = X_train array([[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	20.57 17 19.69 21 11.42 20 20.29 12 unique() ype=int64) sis column gnosis", axis= ure_mean perime 10.38 17.77 21.25 20.38 14.34 s"] dtype: int64 he dataset in el_selection _train, y_test the features processing in ler() _transform(X_ sform(X_test) 27.41, 0.75972 741, 0.24845	######################################	2.90 1326.0 0.00 1203.0 7.58 386.1 5.10 1297.0 mean smoothnes 001.0 326.0 203.0 386.1 297.0 a_mean smooth 710.6 in such a mar s the data. S cts_split st_split(x,y, 74864,, 74864,, 74864,, 74864,, 74864,, 74864,, 74864,,	0.00 0.11 0.12 0.13 0.14 0.19 0.1840 0.08474 0.10960 0.14250 0.10030 0.10105866, 0.68294841, 1.6437431 , 1.17311909, 0.08602275,	actness_mean col 0.27760 0.07864 0.15990 0.28390 0.13280 mpactness_mean col 0.1052	0.15990 0.1970 0.28390 0.243 0.13280 0.1980 0.3001 0.1 0.0869 0.0 0.1974 0.1 0.2414 0.1 0.1980 0.1 0.05375 0 and standard
Dut[27]: In [28]: Out[28]: Out[28]: In [30]: In [31]: Out[31]: In [32]: Out[32]: In [32]: Out[32]: In [33]: Out[33]: In [34]: Out[34]: In [35]: Out[36]: In [37]: In [37]: In [37]: In [39]: In [39]: In [39]: In [39]: In [39]:	0 1 2 3 4 5 rows × 31 df["diag array([1 # to dro X = df.d X.head() radius_ 0 1 2 3 4 5 rows × 30 y = df["y.head() 0 1 1 2 1 3 1 4 1 Name: dia # to seg model = from skl X_train, df.shape (569, 31 X_train. (398, 30 X_test.s (171, 30 y_train. (398,) y_test.s (171,) X_train. radiu 184 1 rows × 30 Standard deviation has nega # to staf from skl x_train x_test = X_train x_test = X_train array([[[[[[[[[[[[[[[[[[[[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	20.57 17 19.69 21 11.42 20 20.29 12 20.29 12 20.29 12 20.29 12 20.29 12 20.29 12 20.29 12 20.29 12 20.29 12 20.29 12 20.29 12 20.29 12 20.29 12 20.29 12 20.29 12 20.29 12 20.29 12 20.29 12 20.29 12 20.38 12 20.38 12 20.38 14.34 20.20 12 20.38 14.34 20.20 12 20.38 14.34 20.20 12 20.38 14.34 20.20 12 20.38 14.34 20.20 12 20.38 14.34 20.20 12 20.38 14.34 20.20 12 20.38 14.34 20.20 12 20.38 14.34 20.20 12 20.38 14.34 20.20 12 20.38 14.34 20.20 12 20.38 14.34 20.20 12 20.38 14.34 20.20 12 20.20 12 20.20 12 20.38 14.34 20.20 12 20.20	######################################	2.90 1326.0 0.00 1203.0 7.58 386.1 5.10 1297.0 mean smoothnes 001.0 326.0 203.0 386.1 297.0 d test models n_test_split st_split(x,y, 710.6 in such a mar s the data. S rdScaler d test_split st_split(x,y, 710.6 in such a mar s the data. S rdScaler	0.0 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1	actness_mean col 0.27760 0.07864 0.15990 0.28390 0.13280 inpactness_mean col 0.1052	0.15990 0.1970 0.28390 0.243 0.13280 0.1980 0.3001 0.1 0.0869 0.0 0.1974 0.1 0.2414 0.1 0.1980 0.1 0.05375 0 and standard
Dut[27]: In [28]: Out[28]: Out[28]: In [30]: In [31]: Out[31]: In [32]: Out[32]: In [32]: Out[32]: In [33]: Out[33]: In [34]: Out[34]: In [35]: Out[36]: In [37]: In [37]: In [37]: In [39]: In [39]: In [39]: In [39]: In [39]:	0 1 2 3 4 5 rows × 31 df["diag array([1 # to dro X = df.d X.head() radius_ 0 1 2 3 4 5 rows × 30 y = df[" y.head() 0 1 1 2 1 3 1 4 1 Name: dia # to seg mode1 = from skl X_train, df.shape (569, 31 X_train. (398, 30 X_test.s (171, 30 Y_train. (398,) y_test.s (171,) X_train. radiu 184 1 rows × 30 Standard deviatio has nega # to sta from skl ss = Sta X_train x_test = X_train array([[[[[[[[[[[[[[[[[[[[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	20.57 17 19.69 21 11.42 20 20.29 14 unique() ype=int64) sis column gnosis", axis= ure_mean perime 10.38 17.77 21.25 20.38 14.34 s"] dtype: int64 he dataset in el_selection _train, y_test the features processing in ler() _transform(X_test) 22.41 It transform(X_sform(X_test) 27.4, 0.75972 73, 0.77376 26.59.6, -0.24616 26.70.7376 27.71, 0.75972 27.72, 0.77576 27.73, 0.77376 27.74, 0.75972 27.74, 0.75972 27.75, -0.24845 27.75, -0.24845 28.70, -0.33775 29.71, 0.7257 29.72, 0.7257 20.73, 0.77376 20.75, -0.24815 20.77, 0.7257 20.77,	### ### ### ### ### ### ### ### ### ##	2.90 1326.0 0.00 1203.0 7.58 386.1 5.10 1297.0 mean smoothnes 001.0 326.0 203.0 386.1 297.0 d test models 1.5t_split(X,y, 1.5t_st_split(X,y, 1.5t_st_split(X,y, 1.5t_st_split(X,y, 1.5t_split(X,y, 1.5t_spli	0.00 0.11 0.11 0.11 0.11 0.11 0.11 0.11	actness_mean col 0.27760 0.07864 0.15990 0.28390 0.13280 Apactness_mean col 0.1052 That mean as as a sion is useful	0.15990 0.1970 0.28390 0.243 0.13280 0.1980 0.3001 0.1 0.0869 0.0 0.1974 0.1 0.2414 0.1 0.1980 0.1 0.05375 0 and standard
Dut [27]: In [28]: In [28]: Out [28]: In [30]: In [31]: In [31]: In [32]: In [32]: In [33]: In [34]: In [35]: In [36]: In [36]: In [37]: In [37]: In [39]: In [40]: Out [40]: Out [40]:	0 1 2 3 4 5 rows × 31 df["diag array([1 # to dro X = df.d X.head() radius_ 0 1 2 3 4 5 rows × 30 y = df["y.head() 0 1 1 2 1 3 1 4 1 Name: dia # to seg model = from skl X_train, (398, 30 X_train. (398, 30 X_test.s (171, 30 y_train. (398,) y_test.s (171,) X_train. radiu 184 1 rows × 30 Standard deviatio has nega # to sta from skl ss = Sta X_train array([[[[[[[[[[[[[[[[[[[[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	20.57 17 19.69 21 11.42 20 20.29 12 unique() ype=int64) sis column gnosis", axis= ure_mean perime 10.38 17.77 21.25 20.38 14.34 s"] dtype: int64 he dataset in el_selection _train, y_test the features processing in ler() _train, y_test 22.41 It transform(X_ ss"] attrain, y_test your approach in the selection _train, y_test 22.41 It transform(X_ ssform(X_test) 23.41 A	### A 13. 1.25 13. 1	2.90	0.00 0.11 0.11 0.11 0.11 0.11 0.11 0.11	actness_mean col 0.27760 0.07864 0.15990 0.28390 0.13280 Apactness_mean col 0.1052 That mean as as a sion is useful	0.15990 0.1970 0.28390 0.243 0.13280 0.1980 0.3001 0.1 0.0869 0.0 0.1974 0.1 0.2414 0.1 0.1980 0.1 0.05375 0 and standard
Dut [27]: In [28]: In [29]: In [30]: In [30]: In [31]: In [31]: In [32]: In [32]: In [32]: In [33]: In [34]: In [35]: In [36]: In [36]: In [37]: In [37]: In [37]: In [40]: In [4	0 1 2 3 4 5 rows × 31 df["diag array([1 # to dro X = df.d X.head() radius_ 0 1 2 3 4 5 rows × 30 y = df[" y.head() 0 1 1 2 1 3 1 4 5 rows × 30 X_train. (398, 30 X_train.	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	20.57 17 19.69 21 11.42 20 21.1.42 20 20.29 12 2	meter_mean area_r 122.80 10 132.90 13 130.00 13 77.58 13 135.10 13 meter_mean area_r 122.80 10 132.90 13 135.10 13 meter_mean area_r 124.80 10 132.90 13 135.10 13 136.40 135.10 13 136.41 136.41 137.41 138 138.41	2.90 1326.0 0.00 1203.0 7.58 386.1 5.10 1297.0 mean smoothnes 001.0 326.0 203.0 386.1 297.0 d test models 1.5t_split(X,y, 1.	0.00 0.11 0.12 0.12 0.13 0.14 0.14 0.10960 0.14250 0.10030 ness_mean cor 0.09057 nner that it standardizat 0.10105866, 0.68294841, 1.17311909, 0.08602275, 0.26822331, 0.99419176, 1.46387893, 0.99419176, 1.46387893, 0.99419176, 1.46387893, 0.99419176, 1.46387893, 0.99419176,	actness_mean col 0.27760 0.07864 0.15990 0.28390 0.13280 Ctraining mode 0.3) Chas mean as cion is useful	0.15990 0.1970 0.28390 0.243 0.13280 0.1980 0.3001 0.1 0.0869 0.0 0.1974 0.1 0.2414 0.1 0.1980 0.1 0.05375 0 and standard
Dut [27]: In [28]: In [28]: Out [28]: In [30]: In [31]: In [31]: In [31]: In [32]: In [32]: In [32]: In [33]: In [34]: In [35]: In [36]: In [37]: In [37]: In [37]: In [41]: In [42]: In [42]: In [42]: In [43]: In [44]: In [44]: In [44]:	0 1 2 3 4 5 rows × 31 df["diag array([1 # to dro X = df.d X.head() radius_ 0 1 2 3 4 5 rows × 30 Y = df[" y.head() 0 1 1 2 1 3 4 1 Name: dia # to seg model = from skl X_train. (398, 30 X_train. (398, 30 X_test.s (171, 30 Y_train. (398,) Y_train. (398,) Y_train. (398,) Y_train. (398,) Y_train. (398,) Y_train. (398,) X_test.s (171,) X_train. radiu 184 1 rows × 30 Standard deviation has nega # to stan from skl rows x 30 Standard deviation has nega # to stan from skl rows x 30 X_test.s (171,) X_train. I I I I I I I I I I I I I	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	20.57 17 19.69 21 11.42 20 21.42 20 22.29 14 unique() ype=int64) sis column gnosis", axis= ure_mean perime 10.38 17.77 21.25 20.38 14.34 dtype: int64 the dataset in el_selection _train, y_test 22.41 It transform(X_ seform(X_test) 274, 0.75972 773, 0.77376 774, 0.7486 366, -0.24616 367, -0.33775 366, 0.02492 3675, -0.22796 3687, -0.33775 366, 0.02492 3775, -0.0757 3784, -1.06178 3784, -1.06178 389, -0.46731 3784, -1.06178 389, -0.46731 399, -0.46731 399, -0.4799	meter_mean area_r 122.80 10 132.90 13 130.00 13 77.58 3 135.10 13 meter_mean area_r 123.90 13 130.00 13 77.58 3 135.10 13 14 15 16 17 18 19 19 19 19 19 19 19 19 19	2.90	0.00 0.11 0.11 0.11 0.11 0.11 0.11 0.11	actness_mean col	0.15990 0.1970 0.28390 0.243 0.13280 0.1980 0.3001 0.1 0.0869 0.0 0.1974 0.1 0.2414 0.1 0.1980 0.1 0.05375 0 and standard
Dut [27]: In [28]: In [29]: In [30]: In [31]: In [31]: In [31]: In [32]: In [32]: In [32]: In [33]: In [34]: In [35]: In [36]: In [37]: In [37]: In [37]: In [41]: In [42]: In [42]: In [42]: In [42]: In [43]: In [44]: In [42]: In [44]: In [46]: In [4	0 1 2 3 4 5 rows × 31 df["diag array([1 # to dro X = df.d X.head() radius_ 0 1 2 3 4 5 rows × 30 Y = df[" Y.head() 0 1 1 2 3 4 5 rows × 30 Y = df[" Y.head() 0 1 1 2 1 3 1 4 1 Name: dia # to seg model = from skl X_train, df.shape (569, 31 X_train. (398, 30 X_test.s (171, 30 Y_train. (398,) Y_test.s (171,) X_train. (398,) Y_test.s (171,) X_train. radiu 184 1 rows × 30 Standard deviation has nega # to sta from skl ss = Sta X_train array([[[[[[[[[[[[[[[[[[[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	20.57 17 19.69 21 11.42 20 21.42 20 220.29 14 unique() ype=int64) sis column gnosis", axis= are_mean perime 10.38 17.77 21.25 20.38 14.34 s"] dtype: int64 he dataset in el_selection _train, y_test 22.41 It transform(X_ ss"] dtype: int64 he dataset in el_selection _train, y_test 27, 0.77376 273, 0.77376 273, 0.77376 274, 0.24845 44, 0.41763 265, -0.2929 265, -0.2707 682, -0.13006 265, -0.2929 265, -0.2707 682, -0.1306 265, -0.2929 265, -0.2707 682, -0.1306 265, -0.2929 265, -0.2707 273, 0.77376 274, 0.75972 275, -0.07257 276, -0.07257 2773, 0.77376 2774, 0.24845 2775, -0.07257 2788, -1.03546 2775, -0.07257 2788, -1.03546 2775, -0.07257 2788, -1.03546 2775, -0.07257 2788, -1.03546 2775, -0.07257 2789, -0.1306	meter_mean area_r 122.80 10 132.90 13 130.00 13 77.58 3 135.10 13 import train t = train_tes import standar train) 2699, 0.2710 394], 6846, 1.5625 3432], 6447, 6594], 6846, 1.5625 3432], 6470, 6994], 6846, 1.5625 3432], 6470, 6994], 6846, 1.5625 3432], 6470, 6994], 6846, 1.5625 3432], 6470, 6994], 6846, 1.5625 3432], 6470, 6994], 6846, 1.5625 6871,	2.90	0.00 0.11 0.11 0.11 0.11 0.11 0.11 0.11	actness_mean col 0.27760 0.07864 0.15990 0.28390 0.13280 ftraining mode 0.3) firaining mode 0.3) actness_mean col 0.07864 0.15990 0.28390 0.13280 actness_mean col 0.1052	0.15990 0.1970 0.28390 0.243 0.13280 0.1980 0.3001 0.1 0.0869 0.0 0.1974 0.1 0.2414 0.1 0.1980 0.1 0.05375 0 and standard
Dut [27]: In [28]: In [29]: Out [29]: In [30]: In [31]: In [31]: In [32]: In [32]: In [32]: In [33]: In [34]: In [36]: In [37]: In [37]: In [37]: In [41]: In [42]: Out [40]: Out [40]: Out [40]: Out [40]:	0 1 2 3 4 5rows × 33 df["diag array([1 # to dro X = df.d X.head() radius_ 0 1 2 3 4 5rows × 30 y = df[" y.head() 0 1 1 2 1 3 1 4 5rows × 30 y = df[" y.head() 0 1 1 2 1 3 1 4 1 Name: dia # to seg model = from skl X_train, (398, 30 X_train. (398, 30 X_test.s (171, 30 y_train. (398,) y_test.s (171,) X_train. radiu 184 1 rows × 30 Standard deviation has nega # to stan from skl ss = Sta X_train x_test = X_train array([[[[[[[[[[[[[[[[[[[[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	20.57 17 20.57 17 20.69 27 21.42 20 20.29 14 20.29 14 20.29 14 20.29 16 20.29 16 20.29 16 20.29 16 20.29 16 20.29 16 20.29 16 20.29 16 20.29 16 20.29 16 20.29 16 20.29 16 20.29 16 20.29 16 20.20 16 20.	meter_mean area_r 122.80 10 132.90 13 130.00 13 77.58 3 135.10 13 import train t = train_tes import standar train) 2699, 0.2710 394], 6846, 1.5625 3432], 6447, 6594], 6846, 1.5625 3432], 6470, 6994], 6846, 1.5625 3432], 6470, 6994], 6846, 1.5625 3432], 6470, 6994], 6846, 1.5625 3432], 6470, 6994], 6846, 1.5625 3432], 6470, 6994], 6846, 1.5625 6871,	2.90	0.00 0.11 0.11 0.11 0.11 0.11 0.11 0.11	actness_mean col 0.27760 0.07864 0.15990 0.28390 0.13280 ftraining mode 0.3) firaining mode 0.3) actness_mean col 0.07864 0.15990 0.28390 0.13280 actness_mean col 0.1052	0.15990 0.1970 0.28390 0.243 0.13280 0.1980 0.3001 0.1 0.0869 0.0 0.1974 0.1 0.2414 0.1 0.1980 0.1 0.05375 0 and standard
Dut [27]: In [28]: In [29]: In [30]: In [31]: In [31]: In [31]: In [32]: In [32]: In [32]: In [33]: In [34]: In [35]: In [36]: In [37]: In [37]: In [37]: In [41]: In [42]: In [42]: In [42]: In [42]: In [43]: In [44]: In [42]: In [44]: In [46]: In [4	0 1 2 3 4 5 rows × 31 df["diag array([1 # to dro X = df.d X.head() radius_ 0 1 2 3 4 5 rows × 30 y = df["y,head() 0 1 1 2 1 3 1 4 1 Name: dia # to seg model = h from skl. X_train, df.shape (569, 31 X_train. (398, 30 X_test.s (171, 30 y_train. (398,) y_test.s (171,) X_train. radiu 184 1 rows × 30 Standard deviation has nega # to stal from skl. ss = Sta X_train X_test = X_train array([[[[[[[[[[[[[[[[[[[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	20.57 17 20.57 17 20.59 21 20.69 21 21.42 26 20.29 14 22 20.29 14 22 20.29 14 22 20.29 14 22 20.29 14 22 20.29 14 22 20.29 14 23 24 25 26 26 27 28 28 28 28 28 28 28 28 28 28 28 28 28	meter_mean area_r 122.80	2.90 1326.0 2.90 120.00 1203.0 7.58 386.1 5.10 1297.0 2.326.0 2.03.0 2.	0.00 0.11 0.11 0.11 0.12 0.13 0.14250 0.10030 0.14250 0.10030 0.14250 0.10030 0.14250 0.10030 0.14250 0.10030 0.14250 0.10030 0.14250 0.10030 0.14250 0.10030 0.14250 0.10030 0.14250 0.10030 0.14250 0.10030 0.14250 0.10030 0.14250 0.10030 0.14250 0.10030 0.14250 0.10030 0.14250 0.10030 0.14250 0.10030	actness_mean col 0.27760 0.07864 0.1599 0.28390 0.13280 **Training mode** 0.3) **Training mode** 0.1052 **That mean as ion is useful**	0.15990 0.1970 0.28390 0.243 0.13280 0.1980 0.3001 0.1 0.0869 0.0 0.1974 0.1 0.2414 0.1 0.1980 0.1 0.05375 0 and standard
Dut [27]: In [28]: In [28]: In [29]: In [30]: In [31]: In [31]: In [31]: In [32]: In [32]: In [32]: In [34]: In [35]: In [36]: In [36]: In [37]: In [37]: In [47]: In [48]: In [4	0 1 2 3 4 5 rows × 31 df["diag array([1 # to dro X = df.d X.head() radius_ 0 1 2 3 4 5 rows × 30 y = df["y.head() 0 1 1 2 1 3 4 1 Name: dia # to seg model = h from skl. X_train, df.shape (569, 31 X_train. (398, 30 X_train. (398, 30 X_train. (398,) y_test.s (171, 30 y_train. (398,) y_test.s (171,) X_train. radiu 184 1 rows × 30 Standard deviatio has nega # to sta from skl. ss = Sta X_train x_test = X_train array([[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	20.57 17 19.69 21 19.69 21 11.42 20 20.29 14 20.29 14 20.29 14 20.29 14 20.29 14 20.29 14 20.29 14 20.29 14 20.29 14 20.29 14 20.29 14 20.29 14 20.29 14 20.29 14 20.29 16 20.38 17.77 2.25 20.38	meter_mean area_r 122.80 10 132.90 13 130.00 13 77.58 13 135.10 13 meter_mean area_r 122.80 10 132.90 13 130.00 13 77.58 13 135.10 13 meter_mean area_r 98.92 ms the data is standardizes for dataset area_r 122.80 10 132.90 13 130.00 13 77.58 13 135.10 13 meter_mean area_r 98.92 ms the data is standardizes sof dataset area_r 122.80 10 132.90 13 130.00 13 135.10 13 135.10 13 136.90 1, 0.27 136.91 1, 0.27 136.91 1, 0.27 136.91 1, 0.27 136.91 1, 0.30 136.91 1	2.90 1326.0 0.00 1203.0 7.58 386.1 5.10 1297.0 mean smoothnes 001.0 326.0 203.0 386.1 297.0 d test models n_test_split st_split(x,y, 36143,, 36441,, 36441,, 36441,, 36844,, 36844,, 36844,, 36844,, 36844,, 36844,, 36844,, 36844,, 36844,, 36844,, 36844,, 36848,, 368	0.00 0.11 0.11 0.11 0.12 0.13 0.14250 0.10030 0.14250 0.10030 0.14250 0.10030 0.14250 0.10030 0.14250 0.10030 0.14250 0.10030 0.14250 0.10030 0.14250 0.10030 0.14250 0.10030 0.14250 0.10030 0.14250 0.10030 0.14250 0.10030 0.14250 0.10030 0.14250 0.10030 0.14250 0.10030 0.14250 0.10030 0.14250 0.10030	actness_mean col 0.27760 0.07864 0.1599 0.28390 0.13280 **Training mode** 0.3) **Training mode** 0.1052 **That mean as ion is useful**	1 = 70% testing Concavity_mean Con
Dut [27]: In [28]: In [28]: In [29]: In [30]: In [31]: In [31]: In [31]: In [32]: In [32]: In [32]: In [34]: In [35]: In [36]: In [36]: In [37]: In [37]: In [47]: In [48]: In [4	0 1 2 3 4 5 rows × 31 df["diag array([1 # to dro X = df.d X.head() radius_ 0 1 2 3 4 5 rows × 30 y = df["y.head() 0 1 1 2 1 3 1 4 1 Name: dia # to seg model = from skl X_train, df.shape (569, 31 X_train. (398, 30 X_test.s (171, 30 y_train. (398,) Y_test.s (171,) X_train. (398,) Y_test.s (171,) X_train. array([[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	20.57 17 19.69 21 19.69 21 11.42 20 20.29 14 unique() ye=int64) sis column gnosis', axis= lre_mean perime 10.38 17.77 21.25 20.38 14.34 dtype: int64 the dataset in ell_selection train,y_test 22.41 It transform(X_test) 27.4, 0.75972 77.4, 0.75972 77.4, 0.75972 77.4, 0.75972 77.4, 0.74848 the dataset in ell_selection train,y_test 22.41 It transform(X_test) 26.66, 0.02492 26.77, 0.7776 26.77, 0.27767 26.82, 0.13006 26.93, 0.02492 26.93, 0.13626 26.94, 0.13626 26.95, 0.2992 26.95, 0.2992 26.95, 0.2992 26.95, 0.2992 26.95, 0.13066 26.97, 0.13066 26.99, 0.2992 26.99, 0.1066 26.99, 0.2992 26.99, 0.1066 26.9	### ### ### ### ### ### ### ### ### ##	2.90 1326.0 0.00 1203.0 7.58 386.1 5.10 1297.0 mean smoothnes 001.0 326.0 203.0 336.1 297.0 d test models st_split(X,y, 36143,, 37.6441,, 37.6441,, 37.6441,, 38.825,	0.10105866, 0.08474 0.10960 0.14250 0.10030 0.14250 0.10030 0.10030 0.10030 0.101030	actness_mean col 0.27760 0.07864 0.1599 0.28390 0.13280 **Training mode** 0.3) **Training mode** 0.1052 **That mean as ion is useful**	1 = 70% testing 1
Dut [27]: In [28]: In [28]: Dut [28]: Dut [28]: In [30]: In [30]: In [30]: In [31]: In [31]: In [32]: In [33]: In [34]: In [35]: In [36]: In [36]: In [36]: In [37]: In [47]: In [48]: In [49]: In [49]: In [49]: In [49]: In [40]: In [40]: In [40]: In [41]: In [42]: In [42]: In [43]: In [44]: In [44]: In [44]: In [44]: In [46]: In [46]: In [46]: In [47]: In [48]:	0 1 2 3 4 5 rows × 31 df["diag array([1 # to dro X = df.d X head() radius_ 0 1 2 3 4 5 rows × 30 y = df["y, head() 0 1 1 2 1 3 4 1 Name: dia # to seg model = from skl X_train, df.shape (569, 31 X_train. (398,) y_train. (398,) X_train. x_test = X_train array([[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	20.57 17 19.69 21 19.69 21 11.42 20 20.29 14 unique() ype=int64) sis column gnosis", axis= 10.38 17.77 21.25 20.38 14.34 dtype: int64 the dataset in el_selection	### ### ### ### ### ### ### ### ### ##	2.90 1326.0 0.00 1203.0 7.58 386.1 5.10 1297.0 mean smoothnes 001.0 326.0 203.0 336.1 297.0 d test models st_split(X,y, 36143,, 37.6441,, 37.6441,, 37.6441,, 38.825,	0.10105866, 0.68294841, 1.6437431, 1.17311909, 0.08602275, 0.26822331, 2.56845161, 0.08902993, 0.99419176, 1.46387893, 0.99419	actness_mean col 0.27760 0.07864 0.15990 0.28390 0.13280 "training mode 0.33) "training mode 0.33) "training mode 0.1052 Thas mean as ion is useful	1 = 70% testing 1
Dut [27]: In [28]: In [28]: Out [28]: Out [28]: In [30]: In [30]: In [30]: In [31]: In [32]: In [32]: In [32]: In [33]: In [34]: In [35]: In [36]: In [47]: In [47]: In [48]: In [48]: Out [40]: In [47]: In [47]: In [48]: In	0 1 2 3 4 5 rows × 33 df["diag array([1] # to dro,	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	20.57 17 19.69 21 19.69 21 11.42 20 20.29 14 20.29 14 20.29 14 20.29 14 20.29 14 20.29 16 20.29 16 20.20 16	### ### ### ### ### ### ### ### ### ##	2.90 1326.0 0.00 1203.0 7.58 386.1 5.10 1297.0 mean smoothnes 001.0 326.0 203.0 386.1 297.0 a.mean smooth 710.6 in such a mar steep split st_split(X,y, 384.1,, 4864,, 48825,, 488441,, 488441,, 488441,, 48864,, 48825,, 48884,, 48884,, 48884,, 48894,, 48894,, 48994,, 48994,, 48994,, 48994,, 48994,, 48994,, 48994,, 48994,, 48994,, 48994,, 48994,, 48994,, 48994,, 48994,, 48994,, 48994,, 48999	0.10105866, 0.104250 0.10030 0.10030 0.1	actness_mean col 0.27760 0.07864 0.15990 0.28390 0.13280 "training mode 0.33) "training mode 0.33) "training mode 0.1052 Thas mean as ion is useful	accuracy of mo concavity_mean points_t 0.3081 0.0089 0.01 0.1980 0.1980 0.1980 accuracy of mo concavity_mean points_t 0.05375 0 and standard for data which
Dut [27]: In [28]: In [28]: In [29]: In [29]: In [30]: In [30]: In [31]: In [31]: In [32]: In [32]: In [34]: In [35]: In [36]: In [37]: In [37]: In [41]: In [42]: In [42]: In [42]: In [42]: In [43]: In [44]: In [44]: In [44]: In [44]: In [46]: In [4	0 1 2 3 4 5 rows × 33 df ["diag array([1 # to dro X = df.d X head() radius_ 0 1 2 3 4 5 rows × 30 y = df ["y, head() 0 1 1 2 1 3 1 4 5 rows × 30 y = df ["y, head() 0 1 1 2 1 3 1 4 1 Name: dia # to seg model = in from skl X_train. (398, 30 X_test.s (171, 30 y_train. (398, 30 X_test.s (171, 30 y_train. (398, 30 X_test.s (171, 30 y_train. (398,) y_test.s (171,) X_train. radiu 184 1 rows × 30 X_test.s (171,) X_train. (398,) y_test.s (171,) X_train array([[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	20.57 17 19.69 27 19.69 27 19.69 27 11.42 20 20.29 12 20.29 12 20.20 12 20.	meter_mean area_t 122.80 10 132.90 13 130.00 13 77.58 13 130.00 13 77.58 13 135.10 13 moto train and area_termin area_	2.90 1326.0 2.00 1203.0 3.00 1203.0 7.58 386.1 5.10 1297.0	0.0.0.0.1	actness_mean colors	accuracy of mo concavity_mean points_t 0.3081 0.0089 0.01 0.1980 0.1980 0.1980 accuracy of mo concavity_mean points_t 0.05375 0 and standard for data which
Dut [27]: In [28]: In [28]: In [29]: In [29]: In [30]: In [30]: In [31]: In [31]: In [32]: In [32]: In [34]: In [35]: In [36]: In [37]: In [37]: In [41]: In [42]: In [42]: In [42]: In [42]: In [43]: In [44]: In [44]: In [44]: In [44]: In [46]: In [4	0 1 2 3 4 5 rows × 33 df ["diag array([1] # to dro, X - df.d X - head() radius_ 0 1 2 3 4 5 rows × 30 y = df [", y - head() 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	20.57 17 19.69 27 19.69 27 19.69 27 19.69 27 11.42 20 20.29 12 20.29 12 20.20 12 20.	### ### ### ### ### ### ### ### ### ##	2.90 1326.0 2.00 1203.0 3.6.1 5.10 1297.0	0.00.0.1 0.1 0	### Accu Contact Contact	accuracy of mo concavity_mean points_t 0.3081 0.0089 0.01 0.1980 0.1980 0.1980 accuracy of mo concavity_mean points_t 0.05375 0 and standard for data which
Out [27]: In [28]: In [28]: Out [28]: Out [28]: Out [29]: Out [29]: In [30]: In [30]: In [30]: In [31]: In [32]: In [32]: In [32]: In [34]: In [34]: In [34]: In [37]: In [37]: In [47]: In [48]: Out [48]: Out [49]: In [49]:	0 1 2 3 4 5 rows × 33 df["diag array([1] # to dro X = df.d X head() radius_ 0 1 2 3 4 5 rows × 30 y = df[" y head() 0 1 1 2 1 3 1 4 1 Name: diag # to seg model = from skl X_train, df.shape (569, 31 X_train. (398, 30 X_test.s (171,) X_train. (398,) y_test.s (171,) X_train. (398,) Y_test.s (171,) X_train. (398,) Y_test.s (171,) X_train. (398,) Y_test.s (171,) X_train. (398,) Y_test.s (171,) X_train. (398,) Y_test.s (171,) X_train. (398,) Y_test.s (171,) X_train. (398,) Y_test.s (171,) X_train. (10 0 0 0 0 # origin Y_test # to train Tray([[] 0 0 0 0 0 # origin Y_test # to train 184 1 rows × 30 Standard devation har regal # to stain from skl # to train 184 1 rows × 30 Standard devation har regal # to stain from skl # to train 190 0 0 0 # origin Y_test 82 11 1368 10 0 10 0 0 0 0 # origin Y_test 82 11 1368 10 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	20.57 17 19.69 21 19.69 21 19.69 21 19.69 21 11.42 20 20.29 14 20.29 14 20.29 14 20.29 14 20.29 14 20.29 14 20.29 14 20.29 14 20.29 14 20.20 20.	### ### ### ### ### ### ### ### ### ##	2.90 1326.0 2.00 1203.0 3.6.1 5.10 1297.0	0.00.0.1 0.1 0	### Accu Contact Contact	accuracy of mo concavity_mean points_t 0.3081 0.0089 0.01 0.1980 0.1980 0.1980 accuracy of mo concavity_mean points_t 0.05375 0 and standard for data which
Dut [27]: Cn [28]: Cn [28]: Cn [29]: Cn [29]: Cn [30]: Cn [30]: Cn [30]: Cn [30]: Cn [31]: Cn [32]: Cn [32]: Cn [32]: Cn [34]: Cn [34]: Cn [36]: Cn [37]: Cn [37]: Cn [38]: Cn [40]: Cn [4	1 2 3 4 4 5 5 7 6 7 8 7 8 7 8 7 8 7 8 7 8 8 1 8 9 8 9 9 9 9 9 1 8 1 9 9 9 9 9 1 8 1 9 1 9	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	20.57 13 19.69 23 11.42 26 20.29 14 20.29 14 20.29 14 20.29 14 20.29 14 20.29 14 20.29 14 20.29 14 20.29 14 20.29 14 20.29 14 20.29 14 20.29 14 20.29 14 20.29 14 20.29 14 20.29 14 20.29 14 20.20 15 20.20 16 20.	meter_mean area_rates 2.77	2.90 1326.0 2.00 1203.0 2.00 1203.0 2.00 1203.0 7.58 386.1 5.10 1297.0 mean smoothnes 001.0 326.0 203.0 326.0 203.0 326.1 297.0 d test models 1.1 test split 1.2 split 1.3 split 1.4 split 1.4 split 1.5 split 1.6 split 1.7 spl	0.00.0.1 0.1 0	### Accu Contact Contact	accuracy of mo concavity_mean points_t 0.3081 0.0089 0.01 0.1980 0.1980 0.1980 accuracy of mo concavity_mean points_t 0.05375 0 and standard for data which
Out [27]: Out [27]: In [28]: Out [28]: Out [28]: Out [29]: In [29]: Out [29]: In [30]: In [31]: In [31]: In [32]: In [32]: In [32]: In [33]: In [34]: In [35]: In [36]: In [47]: In [48]: Out [48]: Out [49]: In [49]: Out	0 1 2 3 4 5 rows × 33 df ["diag array([1 # to dro, X = df.d, X head() radius_ 0 1 2 3 4 5 rows × 30 y = df[" y head() 0 1 1 2 1 3 4 5 rows × 30 y = df[" y head() 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	20.57 17 19.69 27 19.69 27 11.42 27 20.29 14 20.29 14 20.29 14 20.29 14 20.29 14 20.29 14 20.20 15 20.38 17.77 21.25 20.38 17.77 21.25 20.38 14.34 2 20.38 14.34 2 20.39 14.34 2 20.39 14.34 2 20.38 14.34 2 20.39 1.31 2 20.30 1.30 1 20.30 1.30 1 20.30 1.30 1 20.30 1.30 1 20.30 1 20.30 1 20.30 1 20	meter_mean area 112.80	### ### ### ### ### ### ### ### ### ##	0.0 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1	actness_mean col 0.27760 0.2764 0.15990 0.28390 0.13280 con. con	accuracy of mo concavity_mean points_t 0.3081 0.0089 0.01 0.1980 0.1980 0.1980 accuracy of mo concavity_mean points_t 0.05375 0 and standard for data which
Dut [27]: In [28]: In [28]: In [28]: In [28]: In [29]: In [29]: In [30]: In [31]: In [31]: In [31]: In [31]: In [32]: In [32]: In [33]: In [34]: In [35]: In [36]: In [47]: In [48]: In [52]: In [53]: In [53]: In [53]: In [54]: In [54]: In [54]: In [55]: In [52]: In [52]: In [52]: In [52]: In [53]: In [53]: In [53]: In [54]: In [54]: In [55]: In [55]:	0 1 2 3 4 5 rows x 33 df ["diag array([1 # to dro, X = df.d()	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	20.57 17 19.69 27 11.42 27 11.42 27 11.42 27 11.42 27 11.42 27 11.42 27 11.42 27 11.43 27 11.44 27 11.45 27 11.46 27 11.47 27 11.48 27 11.48 27 11.49 28 11.	### ### ### ### ### ### ### ### ### ##	### ### ### ### ### ### ### ### ### ##	0.0 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1	actness_mean col 0.27760 0.2764 0.15990 0.28390 0.13280 con. con	1 = 70% testing 1

0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0], dtype=int64)

In [62]: tempResults = pd.DataFrame({'Algorithm':['Random Forest Classifier Method'], 'Accuracy':[rfc

In [63]: # to train (fit and transform) the model using Support Vector Classifier Algorithm

1, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0], dtype=int64)

In [67]: tempResults = pd.DataFrame({'Algorithm':['Support Vector Classifier Method'], 'Accuracy':[sv

In []: Hence , **as** the results shows Logistic Regression Method gives the best accuray.

In [60]: **from sklearn.metrics import** accuracy_score print(accuracy_score(y_test, y_pred))

In [61]: rfc_acc = accuracy_score(y_test, y_pred)

results = pd.concat([results, tempResults]) results = results[['Algorithm', 'Accuracy']]

Logistic Regression Method 0.988304

Decision tree Classifier Method 0.941520

Support Vector Classifier

0 Random Forest Classifier Method 0.964912

In [65]: **from sklearn.metrics import** accuracy_score print(accuracy_score(y_test, y_pred))

In [66]: svc_acc = accuracy_score(y_test, y_pred)

results = pd.concat([results, tempResults]) results = results[['Algorithm', 'Accuracy']]

Logistic Regression Method 0.988304

0 Decision tree Classifier Method 0.941520 **0** Random Forest Classifier Method 0.964912 **0** Support Vector Classifier Method 0.970760

Algorithm Accuracy

from sklearn import svm

svc.fit(X_train,y_train)

svc = svm.SVC()

In [64]: y_pred = svc.predict(X_test)

0.9707602339181286

0.9707602339181286

print(svc_acc)

c_acc]})

Out[67]:

y_pred

Algorithm Accuracy

0.9649122807017544

0.9649122807017544

print(rfc_acc)

_acc]})

results

Out[62]:

Out[63]: SVC()

Breast Cancer Prediction

warnings.filterwarnings("ignore")

import matplotlib.pyplot as plt
import seaborn as sns

М

Μ

М

М

М

In [4]: # to get all the columns present in dataset

dtype='object')

In [5]: # to get the detailed info of columns in dataset

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 569 entries, 0 to 568

Data columns (total 33 columns):

i mport warnings

import numpy as np import pandas as pd

In [2]: # importing libraries

import sklearn

df.head()

0 842302

1 842517

2 84300903

3 84348301

4 84358402

df.columns

df.info()

5 rows × 33 columns

Out[3]:

In [1]: # to ignore the warnings given by commands (if any).

to show only first 5 rows of the DataFrame

17.99

20.57

19.69

11.42

20.29

In [3]: # importing dataset and converting it into DataFrame
df = pd.read_csv("https://raw.githubusercontent.com/ingledarshan/AIML-B2/main/data.csv")

10.38

17.77

21.25

20.38

14.34

Out[4]: Index(['id', 'diagnosis', 'radius_mean', 'texture_mean', 'perimeter_mean', 'area_mean', 'smoothness_mean', 'compactness_mean', 'concavity_mean', 'concave points_mean', 'symmetry_mean', 'fractal_dimension_mean', 'radius_se', 'texture_se', 'perimeter_se', 'area_se', 'smoothness_se', 'compactness_se', 'concavity_se', 'concave points_se', 'symmetry_se', 'fractal_dimension_se', 'radius_worst', 'texture_worst', 'perimeter_worst', 'area_worst', 'smoothness_worst', 'compactness_worst', 'concavity_worst', 'concave points_worst', 'symmetry_worst', 'fractal_dimension_worst', 'Unnamed: 32'l,

'symmetry_worst', 'fractal_dimension_worst', 'Unnamed: 32'],

id diagnosis radius_mean texture_mean perimeter_mean area_mean smoothness_mean compactness_mean con

1001.0

1326.0

1203.0

386.1

1297.0

0.11840

0.08474

0.10960

0.14250

0.10030

0.27760

0.07864

0.15990

0.28390

0.13280

122.80

132.90

130.00

77.58

135.10