Selection 1	n [1]: n [2]: n [3]:	<pre>import num import par import ma import sec  df = pd.re</pre>	mpy as ny ndas as y tplotlib aborn as	pd .pyplot as	plt	tion	(Clas	sificat	ion)			
	[4]:	Passeng  0  1  2  3  4  5 rows × 28	1 22.0 2 38.0 3 26.0 4 35.0 5 35.0 columns	7.2500 0 71.2833 1 7.9250 1 53.1000 1	1 1 1 0 1 1	0 0 0	0 0	0 0 0 0 0 0 0 0	0 0 0	0 0 0 0 0 0 0 0	0 3 0 1 0 3 0 1	0 0 0
Column	n [5]:	Pass  O  1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29   1279  1280  1281  1282  1283  1284  1285  1286	()  engerid A  1 22 2 38 3 26 4 38 5 38 6 28 7 54 8 2 9 27 10 14 11 4 12 58 13 20 14 39 15 14 16 58 17 2 18 28 19 3 20 28 21 38 22 34 23 18 24 28 25 8 26 38 27 28 28 18 29 28 21 38 22 34 23 18 24 28 25 8 26 38 27 28 28 18 29 28 21 38 22 34 23 18 24 28 25 8 26 38 27 28 28 18 29 28 21 38 22 34 23 18 24 28 25 8 26 38 27 28 28 18 29 28 21 38 22 34 23 18 24 28 25 8 26 38 27 28 28 18 29 28 21 38 22 34 23 18 24 28 25 8 26 38 27 28 28 18 29 28 21 38 22 34 23 18 24 28 25 8 26 38 27 28 28 18 29 28 20 30 28 21 38 22 34 23 18 24 28 25 8 26 38 27 28 28 18 29 28 28 18 29 28 28 18 29 28 28 18 29 28 28 18 29 28 20 28 21 38 22 34 23 18 24 28 25 8 26 38 27 28 28 18 29 28 20 30 28 21 38 22 34 23 18 24 28 25 8 26 38 27 28 28 18 29 28 20 30 28 20 30 28 20 30	2.0 7.2500 8.0 71.2833 6.0 7.9250 5.0 53.1000 6.0 8.0500 8.0 51.8625 2.0 21.0750 7.0 11.1333 4.0 30.0708 4.0 16.7000 8.0 26.5500 0.0 8.0500 9.0 31.2750 4.0 7.8542 5.0 16.0000 2.0 29.1250 8.0 13.0000 1.0 18.0000 1.0 18.0000 6.0 26.0000 1.0 13.0000	0 1 1 1 0 0 0 0 0 0 1 1 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 1 0 0 0 1 1 1 0 0 0 1 1 0 0 1 1 0 0 0 1 1 0 0 0 1 1 0 0 0 1 1 0 0 0 1 1 0 0 0 1 1 0 0 0 1 1 0 0 0 1 1 0 0 0 1 1 0 0 0 1 1 0 0 0 0 1 1 0 0 0 0 1 1 0 0 0 0 1 1 0 0 0 0 1 1 0 0 0 0 1 1 0 0 0 0 1 1 0 0 0 0 1 1 0 0 0 0 1 1 0 0 0 0 1 1 0 0 0 0 1 0 0 0 0 1 1 0 0 0 0 1 1 0 0 0 0 0 1 1 0 0 0 0 0 1 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 1 0	1						3 1 3 1 3 1 3 3 1 3 3 1 3 3 2 3 1 3 3 2 3 1 3 3 3 1 3 3 1 3 3 1 1 3 3 1 1 3 3 1 1 3 3 1 1 1 3 1 1 1 3 1 1 1 1 3 1
Mathematical	[7]:	1288 1289 1290 1291 1292 1293 1294 1295 1296 1297 1298 1299 1300 1301 1302 1303 1304 1305 1306 1307 1308 1307 rows >  df = df.d 9','zero. ro.14','zero. df.descripted	1289 48 1290 22 1291 33 1292 36 1293 38 1294 22 1295 17 1296 43 1297 26 1298 23 1299 56 1300 28 1301 3 1302 28 1303 37 1304 28 1305 28 1306 38 1307 38 1308 28 1309 28 4 28 column  rop(['zeigero.15',	8.0 79.2000 2.0 7.7750 1.0 7.7333 0.0 164.8667 8.0 21.0000 2.0 59.4000 7.0 47.1000 3.0 27.7208 0.0 13.8625 3.0 10.5000 0.0 211.5000 8.0 7.7208 3.0 13.7750 7.7 90.0000 7.7 90.0000 8.0 7.7750 8.0 8.0500 9.0 108.9000 8.5 7.2500 8.0 22.3583 Ins  ro','zero.16','	1 0 0 1 0 1 0 0 0 0 0 1 1 1 1 1 0 0 0 0	1	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0	1 3 3 1 2 1 1 1 2 1 1 3 3 3 1 3 3 1 3 3 1 3 3 1 3 3 1 3 3 1 3 3 1 3 3 1 3 3 1 3 3 1 3 3 1 3 3 1 3 3 1 3 3 1 3 3 1 3 3 1 3 3 1 3 3 3 1 3 3 1 3 3 3 1 3 3 1 3 3 1 3 3 1 3 3 3 1 3 1 3 3 3 1 3 1 3 3 3 1 3 1 3 3 3 1 3 1 3 3 3 1 3 1 3 3 3 1 3 1 3 3 3 1 3 1 3 3 3 1 3 1 3 3 3 1 3 1 3 3 3 1 3 1 3 3 3 1 3 1 3 3 3 1 3 1 3 3 3 1 3 3 3 1 3 3 3 1 3 1 3 3 3 1 3 3 3 1 3 1 3 3 3 1 3 3 3 1 3 3 3 1 3 3 3 1 3 3 3 1 3 3 3 1 3 3 3 1 3 3 3 3 1 3 3 3 1 3 3 3 1 3 3 3 1 3 3 3 1 3 3 3 1 3 3 3 1 3 3 3 1 3 3 3 1 3 3 3 1 3 3 3 1 3 3 3 1 3 3 3 1 3 3 3 1 3 3 3 1 3 3 3 1 3 3 3 1 3 1 3 3 3 1 3 1 3 3 3 1 3 1 3 3 3 1 3 3 3 1 3 1 3 3 3 1 3 1 3 3 3 3 1 1 3 3 3 3 1 1 3 3 3 1 1 3 3 3 3 1 1 3 3 3 3 1 1 3 3 3 3 1 1 3 3 3 3 1 1 3 3 3 3 1 1 3 3 3 3 1 1 3 1 3 3 3 1 1 3 3 3 3 1 1 3 1 3 3 3 1 1 3 1 3 3 3 1 1 3 1 3 1 3 1 3 3 3 1 1 3 1 3 1 3 3 3 1 1 3 3 3 3 1 1 3 1 3 3 3 1 1 3 1
March   Marc		count 1309 mean 655 std 378 min 1 25% 328 50% 655 75% 982 max 1309	3.000000 1 5.000000 3.020061 1.000000 3.000000 5.000000	29.503186 12.905241 0.170000 22.000000 28.000000 35.000000	1309.0000 33.2810 51.7415 0.0000 7.8958 14.4542 31.2750	000 1309 086 (600 (600 (600 (600 (600 (600 (600 (	9.000000 0.355997 0.478997 0.000000 0.000000 0.000000	1309.000000 0.498854 1.041658 0.000000 0.000000 1.000000	1309.000000 0.385027 0.865560 0.000000 0.000000 0.000000	1309.000000 2.294882 0.837836 1.000000 2.000000 3.000000 3.000000	1307.000000 1.492731 0.814626 0.000000 1.000000 2.000000 2.000000	0.261 0.439 0.000 0.000 0.000 1.000
		Passeng  0  1  2  3  4  5  6  7  8  9  Embarked:  Pclass: 1 =  2urvived: 0  sibsp: no.	erid Age  1 22.0 2 38.0 3 26.0 4 35.0 5 35.0 6 28.0 7 54.0 8 2.0 9 27.0 10 14.0 c 0 = Cheri = 1st, 2 = 2 0 = No, 1 = 0 of sibling	7.2500 0 71.2833 1 7.9250 1 53.1000 1 8.0500 0 8.4583 0 51.8625 0 21.0750 0 11.1333 1 30.0708 1  bourg, 1 = Q 2nd, 3 = 3rd = Yes  Js / spouses	1 1 0 1 1 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1	0 0 0 0 0 1 2 0 wn, 2 =	3 1 3 1 3 3 1 3 2  Southamp	2.0 0.0 2.0 2.0 2.0 1.0 2.0 2.0 2.0 2.0	0 1 1 1 0 0 0 0			
Comparison   Com	[12]: [13]:	0 22.0 1 38.0 2 26.0 3 35.0 4 35.0 5 28.0 6 54.0 7 2.0 8 27.0 9 14.0 10 4.0 11 58.0 12 20.0 13 39.0 14 14.0 15 55.0 16 2.0 17 28.0 18 31.0 19 28.0 20 35.0 21 34.0 22 15.0 23 28.0 24 8.0 25 38.0 26 28.0 27 19.0 28 28.0 27 19.0 1282 51.0 1283 13.0 1284 47.0 1285 29.0 1280 6.0 1281 23.0 1282 51.0 1283 13.0 1284 47.0 1285 29.0 1288 48.0 1289 28.0 1290 31.0 1291 30.0 1292 38.0 1293 22.0 1294 17.0 1295 43.0 1297 23.0 1298 50.0 1299 28.0 1300 3.0 1301 28.0 1302 37.0 1303 28.0 1304 28.0 1307 28.0 1308 38.5 1307 28.0 1308 38.5 1307 28.0 1308 38.5 1307 28.0 1308 38.5 1307 28.0 1308 38.5 1307 28.0 1308 38.5 1307 28.0 1308 38.5 1307 28.0 1308 38.5 1307 28.0 1308 38.5	7.2500 71.2833 7.9250 53.1000 8.0500 8.4583 51.8625 21.0750 11.1333 30.0708 16.7000 26.5500 8.0500 31.2750 7.8542 16.0000 29.1250 13.0000 18.0000 7.2250 26.0000 13.0000 8.0292 35.5000 21.0750 31.3875 7.2250 263.0000 7.8792 7.8958 7.7500 21.0750 93.5000 39.4000 20.2500 10.5000 7.2500 7.7750 7.7333 164.8667 21.0000 59.4000 47.1000 27.7208 13.8625 10.5000 21.5000 7.7208 13.8750 7.7500 9.0000 7.7750 9.0000	0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 1 0	3 1 3 1 3 3 1 3 3 1 3 3 1 3 3 3 2 3 3 2 3 1 3 3 3 1 3 3 1 1 3 3 1 1 3 3 1 1 1 3 3 1 1 1 1 2 1 1 1 2 1 1 1 2 1 1 1 2 1 1 1 2 2 1 3 3 3 3	2.0 0.0 2.0 2.0 2.0 2.0 2.0 2.0 2.0 2.0		test_size=	O.2, random_	state=0)	
Logistic Regression    Comparison	[15]:	class Mode  def	els: _init(self.name elf.accus utput(seleturn f')	self, name, a = name racy = accu lf): Name of mod	accuracy uracy del is {	7):				accuracy}'		
22    print("Accoraty of this model is ", sourtOD)   Accoracy of this model is "0.1908090404690    24    from sklearn.matric@ import controlion matrix   25    confusion_matrix(y_met.lg_pred)   27    confusion_matrix(y_met.lg_pred)   28    confusion_matrix(y_met.lg_pred)   28    confusion_matrix(y_met.lg_pred)   29    confusion_matrix(y_met.lg_pred)   SVC (Simple Vector Classification)   From sklearn.www.import 800   20    clf = 6VC (gamma="a.od", kernel = "rapf")   21    clf.fit(X_train.y_train)   21    clf.fit(X_train.y_train)   clf. clf.fit(X_train.y_train)   clf. clf.fit(X_train.y_train)   clf. clf.fit(X_train.y_train)   clf. clf.fit(X_train.y_train)   clf. clf.fit(X_train.y_train)   clf. clf.fit(X_train.y_train)   clf.fit(X_train.y_train.y_train)   clf.fit(X_train.y_train.y_train.y_train.y_train.y_train.y_train.y_train.y_train.y_train.y_tra	[17]: [18]: [19]: [19]: [20]:	Logistic  from skle  clf = Log  clf.fit(X  LogisticRe  lg_pred =  from skle	= []  C Regres  arn.lines  isticRegres  train,y  egression  clf.preceern.metre	ession  ar_model in  ression(rar  train)  n(C=1.0, cl intercept multi_cla solver='l  dict(X_test	mport Londom_stallass_weit_scalings='ovr.bfgs',	ght=Nong=1, lar, n_j tol=0.	ne, dual 1_ratio= obs=None 0001, ve	=False, fi None, max_ , penalty=	it_interce _iter=1000 ='12', ran	pt=True, , dom_state=0,	s='ovr')	
<pre>[33]: array([[169, 8],</pre>	[23]: [24]: [25]: [25]: [26]: [27]: [28]: [29]: [30]:	Accuracy of Accura	curacy of this representation of this representation of this representation of the repre	f this mode model is 75  ics import  y_test,lg_r  todels('Logi  Vector C  import SVC  auto', kerne  _train)  size=200, c  tion_shape= probability rbose=False  edict(X_test)  ore(y_test,  s',acc*100	confusioned)  istic Reclassif  classif  class_weellovr', y=False, ellowr', ellowr', y=False, ellowr',	egressi  Fication  eight=N degree rando	on',acc*  on)  one, coe =3, gamm	f0=0.0, a='auto',				
[39]: acc = accuracy_score(y_test,nb_pred)  [40]: print('Accuracy is ',acc*100)  Accuracy is 73.66412213740458  [41]: confusion_matrix(y_test,nb_pred)  [41]: array([[144, 33],	[34]:	confusion array([[1] mod_list.  Naive-E from skle	_matrix() 69, 8], 77, 8]] append(Mathematical Mathematical Mathematica	y_test,svc_  y_test,svc_  y_test,svc_  Classific  e_bayes imp  train)	_pred) ',acc*10 cation	ıssianN						
	[37]:		B(priors=	=None, var_								