The content is a proper prop					ature ex	CLUCCIOII	.text i	mport T	fidfVec	torize	r												
Mathematical   Math	_	df =	pd.	read_cs																			
1	2]: _		Unr				Va., Car	s Cas all Lills			Daniel Cu	£:	Cl	.:!!									
Mathematical					Watch The	Exact Mome			-														
Mathematical   Math							_	_															
Mathematical   Math								·															
See 1. 1					State	e Departmer	nt says it ca	an't find en		The	∙ State De	partme	nt tol	d the F	Republi	ican N	Natio.	 RE					
Mathematical																							
## Company of the Control Cont						-			_			•					_						
The second of th						h Is Suddenl	y Attackin	ng Trump. H	ere's W	Jeb	Bush Is S	Suddenl	ly Atta	acking	Trump	. Here	e's W.	RI	EAL				
The second of the content of the con	6	5335 r	OWS	× 4 colun	nns																		
The content of the	4 .						t'])																
The content of the	1	df2.					e_names	_out()]															
1	]:	diz	00	000	0000 00	00000031	00000031	1 000035	00006	0001	0001pt	0002		، حلب	عربي	عن	لم	ما	عاولات	مح	من	، هذا	مرضہ
1																							
Material																							
Martin   Column   C																							
## 12																							
March   Marc																							
The control of the co		6332	0.0	0.007522	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	C	0.0	0.0	0.0	
### 12   18   18   18   18   18   18   18																							
The content of the			df.i	loc[:,3	]																		
1																							
See a second sec		3		REAL FAKE																			
Section   Property Property (Prop.   Special Agency   Property Prop.   Property Prop.   Property Prop.   Pro		6330		REAL																			
NAME THE THE THE THE THE THE THE THE THE TH		6332 6333		FAKE REAL																			
The instruction of the content of	]	6334 Name:	: la	REAL bel, Ler																			
### STATE OF THE TABLE OF THE T						ing <b>impo</b>	<b>rt</b> Labe	elEncode	r														
### ##################################		y = 2																					
### Second State   Se						1, 1])																	
1.0				= df2.il	oc[:,0]																		
A. C.		x_ax: 0																					
The acceptance of the product of the control of the		3		0.0																			
Size   0.00		6330	•	0.0																			
Example of Column and trace import incontinuous months and functional places and incomplete and		6332 6333		0.0																			
See complete with case, yes, catalogs, september, soldiers   case complete with case, yes, catalogs, soldiers, soldiers  case complete with case, yes, catalogs, soldiers, soldiers  case complete with case, yes, catalogs, soldiers, soldiers  case complete with case, yes, catalogs, soldiers, soldiers, case,																							
monorage_nerows_main_yepmatewordfogistic=Trueci=None)  -dama_temport.risocl="("2",")"  -dama_temport.risocl="("2",")"  -form_main_main_temport_pem_temport_riso_pem_ppill  -form_main_main_main_temport_pem_temport_riso_pem_ppill  -form_main_main_temport_pem_temport_		Name:			ngth: 63	35, dtyp	e: floa	t64															
### Predict Under Type of Control		X_ax:	: (0 is =	00,), Ler			e: floa	.t64															
### A Comparison of the Compar		X_ax:	: (0 is =	00,), Ler			e: floa	t64															
### ##################################		<pre>X_ax: len(y 6335 sns.:</pre>	: (0 is = y)	00,), Ler = X_axis plot(x=X_	.to_nump	y() y=y, data:			=True,	ci=Non	<b>e</b> )												
x = ddivation from international selection import train test split x trains, testsy trainsy that = train test split x trains, testsy trainsy that = train test split x trains, testsy trainsy that = train test split from international selection import train test split (x,y,test size=0.20, random_state=0)  from international selection (x,tain)  prod = classifier = Total-internation (x,tain)  from internation matrix y testsy pred labeler (1,0) day = tondescondentials (x,tain)  import selection as ans  diam.plot()  from international selection matrix (x,tanderdomatrix (x,tain))  import selection as ans  diam.plot()  from international selection matrix (x,tanderdomatrix (x,ta		1en (y 6335 sns.:	: (0 is = y)	olot(x=X	.to_nump _axis, y abel="('	y() y=y, data:			=True,	ci=Non	<b>e</b> )												
x = df2.voluen  Erom sklearn.mode_ge.ection import train_teat_sp.it  %_nordin_x_tean.y_readin_y_readin_teat_sp.it %_nordin_x_tean.y_readin_y_readin_teat_sp.it %_nordin_x_tean.y_readin_y_readin_teat_sp.it %_nordin_x_tean.y_readin_y_readin_teat_sp.it %_nordin_x_tean.y_readin_y_readin_teat_sp.idepersolue %_nordin_x_tean.y_readin_y_readin_teat_sp.idepersolue %_nordin_x_tean.y_readin_y_readin_teat_sp.idepersolue %_nordin_x_tean.y_readin_y_readin_teat_sp.idepersolue %_nordin_x_tean.y_readin_y_teat_sp.idepersolue %_nordin_x_tean.y_teat_sp.idepersolue %_nordin_x_tean.y_teat_sp.idepersolue %_nordin_x_tean.y_teat_sp.idepersolue %_nordin_x_teat_sp.idepersolue %_nordin_x_teat_sp.ideper		X_ax: len(y 6335 sns.: <axes< td=""><td>: (0 is = y)</td><td>olot(x=X</td><td>.to_nump _axis, y abel="('</td><td>y() y=y, data:</td><td></td><td></td><td>=True,</td><td>ci=Non</td><td><b>e</b>)</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></axes<>	: (0 is = y)	olot(x=X	.to_nump _axis, y abel="('	y() y=y, data:			=True,	ci=Non	<b>e</b> )												
A = dT2_value  # con visions.mode _ nalesticon import troin_test_opiit  v_troin_v_test_v_troin_v_cest = troin_test_opiit(v, v, test_ot_ot_ot_ot_ot_ot_ot_ot_ot_ot_ot_ot_ot		X_ax: len(y 6335 sns.: <axes 0.8<="" 10="" td=""><td>: (0 is = y)</td><td>olot(x=X</td><td>.to_nump _axis, y abel="('</td><td>y() y=y, data:</td><td></td><td></td><td>=True,</td><td>ci=Non</td><td><b>e</b>)</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></axes>	: (0 is = y)	olot(x=X	.to_nump _axis, y abel="('	y() y=y, data:			=True,	ci=Non	<b>e</b> )												
### ### ##############################		X_ax: len(y 6335  sns.: <axes 0.8<="" td=""><td>: (0 is = y)</td><td>olot(x=X</td><td>.to_nump _axis, y abel="('</td><td>y() y=y, data:</td><td></td><td></td><td>=True,</td><td>ci=Non</td><td><b>e</b>)</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></axes>	: (0 is = y)	olot(x=X	.to_nump _axis, y abel="('	y() y=y, data:			=True,	ci=Non	<b>e</b> )												
### ACCIVATION   PROPERTY   PROPERTY   PROPERTY   PROPERTY    ### ACCIVATION   PROPERTY   PROPERTY   PROPERTY    ### ACCIVATION   PROPERTY   PROPERTY    ### ACCIVATION   PROPERTY   PROPERTY    ### ACCIVATION    ### ACCIVATION		X_ax: len(y 6335  sns.: <axes -="" -<="" 0.4="" 0.6="" 0.8="" 1.0="" td=""><td>: (0 is = y)</td><td>olot(x=X</td><td>.to_nump _axis, y abel="('</td><td>y() y=y, data:</td><td></td><td></td><td>=True,</td><td>ci=Non</td><td><b>e</b>)</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></axes>	: (0 is = y)	olot(x=X	.to_nump _axis, y abel="('	y() y=y, data:			=True,	ci=Non	<b>e</b> )												
### Askern.model pelection import train test split  *_Ltadin.x_Leal.y_Ltadin.y_Leal =tadin_Leal_split(x,y,Leal_size=0.20,tandom_state=0)  from akkern.linear model import DodisticReqression classifier = LogisticRegression() classifier = LogisticRegression()  /_pred = disasifier.predict(*_test) /_pred = disasifier.p		X_ax: len () 6335  sns.: <axes -="" -<="" 0.0="" 0.2="" 0.4="" 0.6="" 0.8="" 1.0="" td=""><td>: (0 is = y) regp sSub</td><td>olot(x=X</td><td>.to_nump _axis, y abel="('</td><td>/=y, data: 00',)"&gt;</td><td>=df2, 1</td><td>Logistic</td><td></td><td>ci=Non</td><td><b>e</b>)</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></axes>	: (0 is = y) regp sSub	olot(x=X	.to_nump _axis, y abel="('	/=y, data: 00',)">	=df2, 1	Logistic		ci=Non	<b>e</b> )												
from sklearn.linear_model import logisticRegression classifier = LogisticRegression()  2		X_ax: len (y 6335  sns.: <axes -="" -<="" 0.2="" 0.4="" 0.6="" 0.8="" 1.0="" td=""><td>: (0 is = y) regp sSub</td><td>olot(x=X)</td><td>.to_nump _axis, y abel="('</td><td>/=y, data: 00',)"&gt;</td><td>=df2, 1</td><td>Logistic</td><td></td><td>ci=Non</td><td><b>e</b>)</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></axes>	: (0 is = y) regp sSub	olot(x=X)	.to_nump _axis, y abel="('	/=y, data: 00',)">	=df2, 1	Logistic		ci=Non	<b>e</b> )												
classifier = logisticRegrossion() LogisticRegrossion()  y_pred = classifier.predict(x_Lest)  from sklearn.metrics import confusion_matrix(confusionXarrixDisplay com = confusionXarrixDisplay c		X_ax: len(y 6335  sns.: <axes (from<="" -="" 0.0="" 0.2="" 0.4="" 0.6="" 1.0="" td=""><td>: (0 is = y) regp sSub df2. skl</td><td>olot(x=X) olot:xla</td><td>_axis, yabel="('</td><td>/=y, data: 00',)"&gt;  0.3  ('00',)</td><td>=df2, 1 0.4 port tr</td><td>ogistic=</td><td>5 t_split</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></axes>	: (0 is = y) regp sSub df2. skl	olot(x=X) olot:xla	_axis, yabel="('	/=y, data: 00',)">  0.3  ('00',)	=df2, 1 0.4 port tr	ogistic=	5 t_split														
y_pred = classifier.predict(x_test) y_pred		X_ax: len(y 6335  sns.: <axes (="" -="" 0.0="" 0.2="" 0.4="" 0.6="" 1.0="" from="" td="" x_tra<=""><td>c: (0 is = y) regp sSub df2. skl ain,</td><td>olot(x=X_oplot:xla</td><td>_axis, yabel="('</td><td>00',)"&gt; 0.3 ('00',) ection imp</td><td>=df2, 1 0.4 port tr</td><td>4 0. cain_test_sp.</td><td>t_split</td><td></td><td></td><td>.20,r.</td><td>andc</td><td>om_sta</td><td>ate=C</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></axes>	c: (0 is = y) regp sSub df2. skl ain,	olot(x=X_oplot:xla	_axis, yabel="('	00',)"> 0.3 ('00',) ection imp	=df2, 1 0.4 port tr	4 0. cain_test_sp.	t_split			.20,r.	andc	om_sta	ate=C								
y_pred  array([1, 0, 0,, 0, 1, 0])  from sklearn.metrics import confusion_matrix,ConfusionMatrixDisplay com = confusion_matrix[y_test.y_pred, labels=[1,0]) disp = ConfusionMatrixDisplay(confusion_matrix=com, display_labels=['True', 'Spam'])  import seaborn as ans  disp.plot() <pre> csklearn.metrics_plot.confusion_matrix.ConfusionMatrixDisplay at Oxle4e4940160&gt;  from sklearn.metrics import classification_report cr = classification_report(y_test, y_pred) print(cr)</pre>		X_ax: len(y 6335  sns.: <axes (from="" -="" 0.0="" 0.2="" 0.4="" 0.6="" 1.0="" class="" class<="" td="" x_tra=""><td>: (0 is = y) regp sSub df2. skl ain, skl sifi sifi</td><td>olot(x=X) olot(x=X) oplot:xla oplot:xla oplot:xla colot(x=X) oplot:xla oplot:xla</td><td>_axis, yabel="(' del_seley_train, near_modgisticRex_train,</td><td>oy()  7=y, data= 00',)"&gt;  0.3  ('00',)  ection imported the important the imported the imported</td><td>=df2, 1  0.4  port tr train_ t Logis</td><td>4 0. cain_test_sp.</td><td>t_split</td><td></td><td></td><td>.20, r.</td><td>andc</td><td>om_sta</td><td>ate=C</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></axes>	: (0 is = y) regp sSub df2. skl ain, skl sifi sifi	olot(x=X) olot(x=X) oplot:xla oplot:xla oplot:xla colot(x=X) oplot:xla	_axis, yabel="(' del_seley_train, near_modgisticRex_train,	oy()  7=y, data= 00',)">  0.3  ('00',)  ection imported the important the imported	=df2, 1  0.4  port tr train_ t Logis	4 0. cain_test_sp.	t_split			.20, r.	andc	om_sta	ate=C								
### According to the confusion matrix of the confusion		X_ax: len(y) 6335  sns.: <axes 0.0="" 0.2="" 0.4="" 0.6="" 0.8="" 1.0="" class="" from="" logis<="" td="" x="0" x_tra=""><td>c: (0 is = y) regp sSub df2. skl ain, skl sifi stic</td><td>olot(x=X_oplot:xlase)  olot(x=X_oplot:xlase)  olot(x=X_oplot:xlase)</td><td>_axis, yabel="('  o.2  del_sele y_train, near_mod gisticRe x_train, ion()</td><td>oy()  oy()  oy()  oy()  ov()  ov()</td><td>=df2, 1  0.4  port tr train_ t Logis ()</td><td>4 0. cain_test_sp.</td><td>t_split</td><td></td><td></td><td>.20,ra</td><td>andc</td><td>om_sta</td><td>ate=C</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></axes>	c: (0 is = y) regp sSub df2. skl ain, skl sifi stic	olot(x=X_oplot:xlase)	_axis, yabel="('  o.2  del_sele y_train, near_mod gisticRe x_train, ion()	oy()  oy()  oy()  oy()  ov()	=df2, 1  0.4  port tr train_ t Logis ()	4 0. cain_test_sp.	t_split			.20,ra	andc	om_sta	ate=C								
cm = confusion_matrix(y_test,y_pred, labels=[1,0])  from_sklearn.metrics import classification_report  cr = classification_report(y_test, y_pred)  print(cr)  precision_recall f1-score_support  0 0.89 0.92 0.90 615 1 0.92 0.90 0.91 652 accuracy_score(y_test, y_pred)  from_sklearn.metrics import accuracy_score score = accuracy_score(y_test, y_pred)  from_sklearn.metrics import classification_report  precision_report(y_test, y_pred)  precision_report(y_test, y_pred)  from_sklearn.metrics_report(y_test, y_pred)  precision_report(y_test, y_pred)  precision_report(y_test, y_pred)  precision_report(y_test, y_pred)  precision_report(y_test, y_pred)  precision_report(y_test, y_pred)  precision_report(y_test, y_pred)		X_ax: len(y) 6335  sns.: <axes -="" 0.0="" 0.2="" 0.4="" 0.6="" 1.0="" class="" from="" logis<="" td="" x_tra=""><td>c: (0 is = y) regp sSub df2. skl ain, skl sifi sifi stic ed =</td><td>olot(x=X_oplot:xlase)  olot(x=X_oplot:xlase)  olot(x=X_oplot:xlase)</td><td>_axis, yabel="('  o.2  del_sele y_train, near_mod gisticRe x_train, ion()</td><td>oy()  oy()  oy()  oy()  ov()  ov()</td><td>=df2, 1  0.4  port tr train_ t Logis ()</td><td>4 0. cain_test_sp.</td><td>t_split</td><td></td><td></td><td>.20, r</td><td>andc</td><td>om_sta</td><td>ate=C</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></axes>	c: (0 is = y) regp sSub df2. skl ain, skl sifi sifi stic ed =	olot(x=X_oplot:xlase)	_axis, yabel="('  o.2  del_sele y_train, near_mod gisticRe x_train, ion()	oy()  oy()  oy()  oy()  ov()	=df2, 1  0.4  port tr train_ t Logis ()	4 0. cain_test_sp.	t_split			.20, r	andc	om_sta	ate=C								
cm = confusion_matrix(y_test,y_pred, labels=[1,0])  from_sklearn.metrics import classification_report  cr = classification_report(y_test, y_pred)  print(cr)  precision_recall f1-score_support  0 0.89 0.92 0.90 615 1 0.92 0.90 0.91 652 accuracy_score(y_test, y_pred)  from_sklearn.metrics import accuracy_score score = accuracy_score(y_test, y_pred)  from_sklearn.metrics import classification_report  precision_report(y_test, y_pred)  precision_report(y_test, y_pred)  from_sklearn.metrics_report(y_test, y_pred)  precision_report(y_test, y_pred)  precision_report(y_test, y_pred)  precision_report(y_test, y_pred)  precision_report(y_test, y_pred)  precision_report(y_test, y_pred)  precision_report(y_test, y_pred)		X_ax: len(y 6335  sns.: <axes (from="" -="" 0.0="" 0.2="" 0.4="" 0.6="" 1.0="" class="" logis="" td="" x_tra="" y_pre="" y_pre<=""><td>c: (0 is = y) regp sSub df2. skl ain, skl sifi sifi stic ed = ed</td><td>olot(x=X_oplot:xlase)  olot(x=X_oplot:xlase)  olot(x=X_oplot:xlase)</td><td>_axis, yabel="('  o.2  del_sele y_train, near_mod gisticRe x_train, ion() fier.pre</td><td>oy()  /=y, data= 00',)"&gt;  0.3  ('00',)  ection import egression y_test = del import egression y_train)</td><td>=df2, 1  0.4  port tr train_ t Logis ()</td><td>4 0. cain_test_sp.</td><td>t_split</td><td></td><td></td><td>.20, ra</td><td>andc</td><td>om_sta</td><td>ate=C</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></axes>	c: (0 is = y) regp sSub df2. skl ain, skl sifi sifi stic ed = ed	olot(x=X_oplot:xlase)	_axis, yabel="('  o.2  del_sele y_train, near_mod gisticRe x_train, ion() fier.pre	oy()  /=y, data= 00',)">  0.3  ('00',)  ection import egression y_test = del import egression y_train)	=df2, 1  0.4  port tr train_ t Logis ()	4 0. cain_test_sp.	t_split			.20, ra	andc	om_sta	ate=C								
disp.plot() <pre></pre>		X_ax: len(y) 6335  sns.: <axes 0.0="" 0.2="" 0.4="" 0.6="" 0.8="" 10="" array="" class="" from="" from<="" logis="" td="" x="0" x_tra="" y_pre=""><td>c: (0 is = y) regp sSub df2. skl ain, sklisifi sifi stic ed = ed y([1 skl</td><td>Olot(x=X]  Olot(x=X]  Olot(x=X]</td><td>_axis, yabel="(' o.2 del_sele y_train, near_mod gisticRe x_train, ion() fier.pre</td><td>oy()  /=y, data: 00',)"&gt;  0.3  ('00',)  ection import egression y_test =  del import egression y_train)</td><td>=df2, 1  0.4  port tr  train_ t Logis ()  fusion_</td><td>4 0.  rain_tes  test_sp:</td><td>t_split lit(x,y,</td><td>,test_</td><td>size=0</td><td></td><td>andc</td><td>om_sta</td><td>ate=C</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></axes>	c: (0 is = y) regp sSub df2. skl ain, sklisifi sifi stic ed = ed y([1 skl	Olot(x=X]	_axis, yabel="(' o.2 del_sele y_train, near_mod gisticRe x_train, ion() fier.pre	oy()  /=y, data: 00',)">  0.3  ('00',)  ection import egression y_test =  del import egression y_train)	=df2, 1  0.4  port tr  train_ t Logis ()  fusion_	4 0.  rain_tes  test_sp:	t_split lit(x,y,	,test_	size=0		andc	om_sta	ate=C								
Tue		X_ax: len(y) 6335  sns.: <axes -="" 0.0="" 0.2="" 0.4="" 0.6="" 0.8="" 10="" array="" class="" cm="&lt;/td" from="" logis="" x="0" x_tra="" y_pre=""><td>c: (0 is = y) regp sSub df2. skl ain, skl sifi stic ed = ed y([1 skl con</td><td>olot(x=X]  olot(x=X]  olot(x=X]</td><td>_axis, y abel="('  del_sele y_train, near_mod gisticRe x_train, ion() fier.pre</td><td>oy()  /=y, data:  00',)"&gt;  0.3  ('00',)  ection import egression y_test =  del import egression y_train)  aport con y_test,y_j</td><td>=df2, 1  0.4  port tr  train_ t Logis ()  fusion_pred, 1</td><td>matrix, (labels=[</td><td>t_split lit(x,y,</td><td>,test_</td><td>size=0</td><td>lay</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></axes>	c: (0 is = y) regp sSub df2. skl ain, skl sifi stic ed = ed y([1 skl con	olot(x=X]	_axis, y abel="('  del_sele y_train, near_mod gisticRe x_train, ion() fier.pre	oy()  /=y, data:  00',)">  0.3  ('00',)  ection import egression y_test =  del import egression y_train)  aport con y_test,y_j	=df2, 1  0.4  port tr  train_ t Logis ()  fusion_pred, 1	matrix, (labels=[	t_split lit(x,y,	,test_	size=0	lay											
True   585   67		X_ax: len(y) 6335  sns.: <axes 0.0="" 0.2="" 0.4="" 0.6="" 0.8="" 10="" array="" class="" cm="disp&lt;/td" from="" logis="" x="0" x_tra="" y_pre=""><td>c: (0 is = y) regp sSub df2. skl ain, sifi sifi stic ed = ed y([1 skl con = C</td><td>olot(x=X_oplot:xlase)  olot(x=X_oplot:xlase)  olot(x=X_oplot:xlase)</td><td>_axis, y abel="('  del_sele y_train, near_mod gisticRe x_train, ion() fier.pre , 0, trics im matrix(y nMatrixD</td><td>oy()  /=y, data:  00',)"&gt;  0.3  ('00',)  ection import egression y_test =  del import egression y_train)  aport con y_test,y_j</td><td>=df2, 1  0.4  port tr  train_ t Logis ()  fusion_pred, 1</td><td>matrix, (labels=[</td><td>t_split lit(x,y,</td><td>,test_</td><td>size=0</td><td>lay</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></axes>	c: (0 is = y) regp sSub df2. skl ain, sifi sifi stic ed = ed y([1 skl con = C	olot(x=X_oplot:xlase)	_axis, y abel="('  del_sele y_train, near_mod gisticRe x_train, ion() fier.pre , 0, trics im matrix(y nMatrixD	oy()  /=y, data:  00',)">  0.3  ('00',)  ection import egression y_test =  del import egression y_train)  aport con y_test,y_j	=df2, 1  0.4  port tr  train_ t Logis ()  fusion_pred, 1	matrix, (labels=[	t_split lit(x,y,	,test_	size=0	lay											
Tue 585 67 -400 -300 -300 -52 563 -200 -100  from sklearn.metrics import classification_report  cr = classification_report(y_test, y_pred) print(cr)  precision recall fl-score support  0 0.89 0.92 0.90 615 1 0.92 0.90 0.91 652 accuracy 0.91 1267 macro avg 0.91 0.91 0.91 1267 weighted avg 0.91 0.91 0.91 1267  from sklearn.metrics import accuracy_score score = accuracy_score(y_test, y_pred) score		X_ax: len(y) 6335  sns.: <axes 0.0="" 0.2="" 0.4="" 0.6="" 0.8="" 10="" array="" class="" cm="disp" disp<="" from="" logis="" td="" x="0" x_tra="" y_pre=""><td>c (0 is = y)  regp Sub  df2. skl ain, sifi sifi stic ed = ed y([1 skl con = C</td><td>olot(x=X]  olot(x=X]  olot(x=X]</td><td>_axis, yabel="('  _o.2  del_sele y_train, near_mod gisticRe x_train, ion()  fier.pre , 0,  trics im matrix(ynMatrixD) as sns</td><td>oy()  oy()  oy()</td><td>=df2, 1  0.4  port tr  train_ t Logis ()  est)</td><td>matrix, abels=[: on_matrix</td><td>confusion (1,0]) x=cm , (</td><td>onMatr</td><td>size=0 ixDisp:</td><td>lay ls=['T</td><td>True</td><td>e','Sp</td><td>pam']</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></axes>	c (0 is = y)  regp Sub  df2. skl ain, sifi sifi stic ed = ed y([1 skl con = C	olot(x=X]	_axis, yabel="('  _o.2  del_sele y_train, near_mod gisticRe x_train, ion()  fier.pre , 0,  trics im matrix(ynMatrixD) as sns	oy()	=df2, 1  0.4  port tr  train_ t Logis ()  est)	matrix, abels=[: on_matrix	confusion (1,0]) x=cm , (	onMatr	size=0 ixDisp:	lay ls=['T	True	e','Sp	pam']								
from sklearn.metrics import classification_report  cr = classification_report(y_test, y_pred) print(cr)  precision recall fl-score support  0 0.89 0.92 0.90 615 1 0.92 0.90 0.91 652 accuracy 0.91 1267 macro avg 0.91 0.91 0.91 1267 weighted avg 0.91 0.91 0.91 0.91 1267  from sklearn.metrics import accuracy_score score = accuracy_score(y_test, y_pred) score		X_ax: len(y) 6335  sns.: <axes 0.0="" 0.2="" 0.4="" 0.6="" 0.8="" 10="" array="" class="" cm="disp" disp<="" from="" logis="" td="" x="0" x_tra="" y_pre=""><td>c (0 is = y)  regp Sub  df2. skl ain, sifi sifi stic ed = ed y([1 skl con = C</td><td>olot(x=X]  olot(x=X]  olot(x=X]</td><td>_axis, yabel="('  _o.2  del_sele y_train, near_mod gisticRe x_train, ion()  fier.pre , 0,  trics im matrix(ynMatrixD) as sns</td><td>oy()  oy()  oy()</td><td>=df2, 1  0.4  port tr  train_ t Logis ()  est)</td><td>matrix, abels=[: on_matrix</td><td>confusion (1,0]) x=cm , (</td><td>onMatr</td><td>size=0 ixDisp:</td><td>lay ls=['T</td><td>True</td><td>e','Sp</td><td>pam']</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></axes>	c (0 is = y)  regp Sub  df2. skl ain, sifi sifi stic ed = ed y([1 skl con = C	olot(x=X]	_axis, yabel="('  _o.2  del_sele y_train, near_mod gisticRe x_train, ion()  fier.pre , 0,  trics im matrix(ynMatrixD) as sns	oy()	=df2, 1  0.4  port tr  train_ t Logis ()  est)	matrix, abels=[: on_matrix	confusion (1,0]) x=cm , (	onMatr	size=0 ixDisp:	lay ls=['T	True	e','Sp	pam']								
from sklearn.metrics import classification_report  cr = classification_report(y_test, y_pred) print(cr)  precision recall f1-score support  0 0.89 0.92 0.90 615 1 0.92 0.90 0.91 652  accuracy 0.91 1267 macro avg 0.91 0.91 0.91 1267 weighted avg 0.91 0.91 0.91 1267  from sklearn.metrics import accuracy_score  score = accuracy_score(y_test, y_pred) score		X_ax:   len (y   6335	control skl sifi stic ed = cd y([1 skl control skl con	olot(x=X]	_axis, yabel="('  _o.2  del_sele y_train, near_mod gisticRe x_train, ion()  fier.pre , 0,  trics im matrix(ynMatrixD) as sns	oy()  /=y, data= 00',)">  0.3  ('00',)  ection import egression y_test =  del import egression y_train)  port con y_test,y_j  confusion  confusion	est)  port tr train_ t Logis ()  fusion_pred, l onfusio		confusion (1,0]) x=cm , (	onMatr	size=0 ixDisp:	lay ls=['T	True	e','Sp	pam']								
from sklearn.metrics import classification_report  cr = classification_report(y_test, y_pred) print(cr)  precision recall f1-score support  0 0.89 0.92 0.90 615 1 0.92 0.90 0.91 652 accuracy 0.91 1267 macro avg 0.91 0.91 0.91 1267 weighted avg 0.91 0.91 0.91 0.91 1267  from sklearn.metrics import accuracy_score  score = accuracy_score(y_test, y_pred) score		X_ax:   len ()   6335	control skl sifi stic ed = cd y([1 skl control skl con	olot(x=X]	_axis, yabel="('  _o.2  del_sele y_train, near_mod gisticRe x_train, ion()  fier.pre , 0,  trics im matrix(ynMatrixD) as sns	oy()  /=y, data= 00',)">  0.3  ('00',)  ection import egression y_test =  del import egression y_train)  port con y_test,y_j  confusion  confusion	=df2, 1  0.4  port tr  train_ t Logis ()  fusion_pred, 1 onfusio		confusion (1,0]) x=cm , (	onMatr	size=0 ixDisp:	lay ls=['T	True	e','Sp	pam']								
<pre>from sklearn.metrics import classification_report  cr = classification_report(y_test, y_pred) print(cr)      precision    recall f1-score</pre>		X_ax:   len ()   6335	condistance of the state of the	Olot (x=X)  Polot	_axis, yabel="('  _o.2  del_sele y_train, near_mod gisticRe x_train, ion()  fier.pre , 0,  trics im matrix(ynMatrixD) as sns	oy()	=df2, 1  0.4  port tr  train_ t Logis ()  fusion_pred, 1 onfusio		confusion (1,0]) x=cm , (	onMatr	size=0 ixDisp:	lay ls=['T	True	e','Sp	pam']								
<pre>from sklearn.metrics import classification_report  cr = classification_report(y_test, y_pred) print(cr)</pre>		X_ax:   len ()   6335	condistance of the state of the	Olot (x=X)  Polot	_axis, yabel="('  _o.2  del_sele y_train, near_mod gisticRe x_train, ion()  fier.pre , 0,  trics im matrix(ynMatrixD) as sns	oy()	est)  port tr train_ t Logis ()  fusion_:pred, l onfusio	matrix, (abels=[:on_matrix] .x.Confus .x.Confus .x.Confus .x.Confus	confusion (1,0]) x=cm , (	onMatr	size=0 ixDisp:	lay ls=['T	True	e','Sp	pam']								
cr = classification_report(y_test, y_pred) print(cr)  precision recall f1-score support  0 0.89 0.92 0.90 615 1 0.92 0.90 0.91 652  accuracy 0.91 1267 macro avg 0.91 0.91 0.91 1267 weighted avg 0.91 0.91 0.91 1267  from sklearn.metrics import accuracy_score score = accuracy_score(y_test, y_pred) score		X_ax:   len ()   6335	condistance of the state of the	olot(x=X_axis)  olot(x=X_oplot:xla)  olot(x=X_oplot	_axis, y abel="('  o.2  del_sele y_train, near_mod gisticRe x_train, ion()  fier.pre , 0,  trics im matrix(y nMatrixD  as sns  splot.	out of the second of the second out of the secon	est)  port tr train_ t Logis ()  fusion_:pred, l onfusio	matrix, (abels=[:on_matrix] .x.Confus .x.Confus .x.Confus .x.Confus	confusion (1,0]) x=cm , (	onMatr	size=0 ixDisp:	lay ls=['T	True	e','Sp	pam']								
precision recall f1-score support  0 0.89 0.92 0.90 615 1 0.92 0.90 0.91 652  accuracy 0.91 1267 macro avg 0.91 0.91 0.91 1267 weighted avg 0.91 0.91 0.91 1267  from sklearn.metrics import accuracy_score  score = accuracy_score(y_test, y_pred) score		X_ax:   len (y   6335   Sns.:   < Axes	condistance of the state of the	olot(x=X_axis)  olot(x=X_axis)	_axis, y abel="('  o.2  del_sele y_train, near_mod gisticRe x_train, ion()  fier.pre , 0,  trics im matrix(y nMatrixD as sns  splot.	ovy()  /=y, data= 00',)">  0.3  ('00',)  ection import egression y_test =  del import egression y_train)  port con y_test,y_j  oisplay(con confusion  for the con y test, y_j  oisplay(con confusion  span abel	est)  port tr train_ t Logis ()  fusion_ pred, l onfusio		confusion (1,0]) x=cm , o	onMatr	size=0 ixDisp:	lay ls=['T	True	e','Sp	pam']								
1 0.92 0.90 0.91 652  accuracy 0.91 1267 macro avg 0.91 0.91 0.91 1267 weighted avg 0.91 0.91 0.91 1267  from sklearn.metrics import accuracy_score  score = accuracy_score(y_test, y_pred) score		X_ax:   len ()   6335	classification skl sification skl skl skl skl sification skl	Olot (x=X)  Polot	_axis, y abel="('  0.2  del_sele y_train, near_mod gisticRe x_train, ion() fier.pre , 0,  trics im matrix(y nMatrixD  as sns  splot.  Predicted la  trics im	ovy()  /=y, data: 00',)">  0.3  ('00',)  ection import egression y_train)  edict(x_train)  confusion /-test,y_n or test,y_n or	=df2, 1  0.4  port tr train_ t Logis ()  fusion_pred, 1 onfusio  n_matri:	matrix, (abels=[:on_matrix])  -500 -400 -300 -100	confusion (1,0]) x=cm , o	onMatr	size=0 ixDisp:	lay ls=['T	True	e','Sp	pam']								
macro avg 0.91 0.91 0.91 1267 weighted avg 0.91 0.91 0.91 1267  from sklearn.metrics import accuracy_score score = accuracy_score(y_test, y_pred) score		X_ax:   len ()   6335	classification skl sification skl skl skl skl sification skl	Olot (x=X)  Polot	_axis, y abel="('  0.2  del_sele y_train, near_mod gisticRe x_train, ion() fier.pre , 0,  trics im matrixD as sns  splot.  Predicted la trics im tion_rep	ovy()	=df2, 1  0.4  port tr  train_ t Logis ()  fusion_pred, 1 onfusio  n_matri:	matrix, (abelse [index])  -500 -400 -300 -200 -100	confusion (1,0]) x=cm , confusion (1,0) (2) x=cm	onMatr displa	size=0 ixDisp:	lay ls=['T	True	e','Sp	pam']								
<pre>weighted avg 0.91 0.91 0.91 1267  from sklearn.metrics import accuracy_score score = accuracy_score(y_test, y_pred) score</pre>		X_ax:   len ()   6335	classification skl sification skl skl skl skl sification skl	Olot (x=X)  Olot (	_axis, y abel="('  0.2  del_sele y_train, near_mod gisticRe x_train, ion() fier.pre , 0,  trics im matrix(y nMatrixD  as sns  splot.  Predicted la trics im tion_rep recision 0.89	y ()  y ()  y ()  y ()  y ()  o ()  o ()  ('00',)  ction import  y ()  etion import  egression  y ('00',)  aport con  y ('test, y ()  confusion  o ('57)  563  Spam  abel  aport (y test)  o ('00',)  confusion  o ('00',)	est)  fusion_pred, lonfusio  n_matri:  st, y_p  ll f1-  92		Confusion  Confusion  SionMath	onMatr displa	size=0 ixDisp:	lay ls=['T	True	e','Sp	pam']								
<pre>score = accuracy_score(y_test, y_pred) score</pre>		X_ax:   len ()   6335	clacu	Olot (x=X)  Polot	_axis, y abel="('  o.2  del_sele y_train, near_mod gisticRe x_train, ion()  fier.pre , 0,  trics im matrix(y nMatrixD  as sns  splot.  Predicted la  trics im tion_rep recision 0.89 0.92	over (y_test)  y()  over (y_test)  confusion  confusion	=df2, 1  0.4  port tr train_ t Logis ()  est)  fusion_ pred, 1 onfusio  n_matri:  2  1  1  1  1  1  1  2  2  3  3  4  4  4  5  6  7  7  8  8  8  8  8  9  9  1  1  1  1  1  1  1  1  1  1  1	matrix, (alpha below ation_reported)  ation_reported)  ation_reported)  ation_reported)	confusion lit(x,y) ession  confusion lit(x,y) ession  sionMatr	onMatr displa rixDispla t	size=0 ixDisp:	lay ls=['T	True	e','Sp	pam']								
score		X_ax:   len (y   6335   Sns.:   < Axes	consisting the state of the sta	Olot (x=X)  Polot	_axis, y abel="('  o2  del_sele y_train, near_mod gisticRe x_train, ion()  fier.pre , 0,  trics im matrixD  as sns  splot.  Predicted la trics im tion_rep recision 0.89 0.92 0.91	over (y_test)  y()  over (y_test)	=df2, 1  0.4  port tr  train_ t Logis ()  fusion_pred, 1 onfusio  n_matri:  1 f1-4 92 90 91	matrix, (ain_test_sticReground)  ation_reground)  ation_reground)  score 0.90 0.91 0.91	Confusion  Confusion  it_split  lit(x,y)  ession  confusion  1,0])  x=cm , of  615  652  1265  1265	onMatr displa rixDispla t	size=0 ixDisp:	lay ls=['T	True	e','Sp	pam']								
0.9060773480662984		len() 6335  sns.: <axes 0.0="" 0.2="" 0.4="" 0.6="" 0.8="" 10="" array="" cdisp="" class="" cm="disp&lt;/td" from="" logis="" skle="" v_pre="" x="0"><td>conted skl</td><td>olo, ), Lender School (1)   Construction of the confusion of the confusion</td><td>.to_nump  _axis, y abel="('  del_sele y_train, near_mod gisticRe x_train, ion()  fier.pre , 0,  trics im matrix(y nMatrixD  as sns  splot.  Predicted la trics im tion_rep recision      0.89      0.92      0.91      0.91      0.91  trics im</td><td>oy()  "=y, data: 00',)"&gt;  0.3  ('00',)  ection import egression y_test = elel import egression y_train)  aport con y_test,y_ 0isplay(con confusion confusion</td><td>adf2, l  O.4  port tr  train_ t Logis ()  est)  fusion_ pred, l onfusio  n_matri:  1 f1- 92 90 91 91 uracy_s</td><td>matrix, (ain_test_spicates</td><td>Confusion  Confusion  it_split  lit(x,y)  ession  confusion  1,0])  x=cm , of  615  652  1265  1265</td><td>onMatr displa rixDispla t</td><td>size=0 ixDisp:</td><td>lay ls=['T</td><td>True</td><td>e','Sp</td><td>pam']</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></axes>	conted skl	olo, ), Lender School (1)   Construction of the confusion	.to_nump  _axis, y abel="('  del_sele y_train, near_mod gisticRe x_train, ion()  fier.pre , 0,  trics im matrix(y nMatrixD  as sns  splot.  Predicted la trics im tion_rep recision      0.89      0.92      0.91      0.91      0.91  trics im	oy()  "=y, data: 00',)">  0.3  ('00',)  ection import egression y_test = elel import egression y_train)  aport con y_test,y_ 0isplay(con confusion	adf2, l  O.4  port tr  train_ t Logis ()  est)  fusion_ pred, l onfusio  n_matri:  1 f1- 92 90 91 91 uracy_s	matrix, (ain_test_spicates	Confusion  Confusion  it_split  lit(x,y)  ession  confusion  1,0])  x=cm , of  615  652  1265  1265	onMatr displa rixDispla t	size=0 ixDisp:	lay ls=['T	True	e','Sp	pam']								
		len() 6335 sns.: <axes 0.0="" 0.2="" 0.4="" 0.6="" 0.8="" 10="" array="" cdisp="" chisp="" class="" from="" logis="" score="" score<="" skle="" td="" v_pre="" x="0"><td>conted skl</td><td>olo,), Lender (a), Lender (b), Lender (c), Learn. Modern (c), Learn. M</td><td>.to_nump  _axis, y abel="('  del_sele y_train, near_mod gisticRe x_train, ion()  fier.pre , 0,  trics im matrix(y nMatrixD  as sns  splot.  Predicted la  trics im tion_rep recision     0.89     0.92     0.91     0.91     0.91  trics im y_score(</td><td>oy()  "=y, data: 00',)"&gt;  0.3  ('00',)  ection import egression y_test = elel import egression y_train)  aport con y_test,y_ 0 isplay(con confusion confusio</td><td>adf2, l  O.4  port tr  train_ t Logis ()  est)  fusion_ pred, l onfusio  n_matri:  1 f1- 92 90 91 91 uracy_s</td><td>matrix, (ain_test_spicates</td><td>Confusion  Confusion  it_split  lit(x,y)  ession  confusion  1,0])  x=cm , of  615  652  1265  1265</td><td>onMatr displa rixDispla t</td><td>size=0 ixDisp:</td><td>lay ls=['T</td><td>True</td><td>e','Sp</td><td>pam']</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></axes>	conted skl	olo,), Lender (a), Lender (b), Lender (c), Learn. Modern (c), Learn. M	.to_nump  _axis, y abel="('  del_sele y_train, near_mod gisticRe x_train, ion()  fier.pre , 0,  trics im matrix(y nMatrixD  as sns  splot.  Predicted la  trics im tion_rep recision     0.89     0.92     0.91     0.91     0.91  trics im y_score(	oy()  "=y, data: 00',)">  0.3  ('00',)  ection import egression y_test = elel import egression y_train)  aport con y_test,y_ 0 isplay(con confusion confusio	adf2, l  O.4  port tr  train_ t Logis ()  est)  fusion_ pred, l onfusio  n_matri:  1 f1- 92 90 91 91 uracy_s	matrix, (ain_test_spicates	Confusion  Confusion  it_split  lit(x,y)  ession  confusion  1,0])  x=cm , of  615  652  1265  1265	onMatr displa rixDispla t	size=0 ixDisp:	lay ls=['T	True	e','Sp	pam']								