10	[3]:				ication.csv	,					
1		expense	expense	coverage		AC	tor_Rating				tic_r
1											
Part 1											
1											
1											
March 15 calc 15 cal											
State											
	505	20.9482		0.427	33496.650	154.3	8.640	8.880	8.680	8.790	
	506 rc	ows × 19 colui	mns								
Transfer		df.iloc[:,-	1]								
	[4]: 1	0									
The content of the	3	1									
The content of the		0									
State Stat	503 504	O O									
Section Sect	Name	e: Start_Tec	_								
March Marc				ing impo	ort LabelEnc	coder					
10	df['	3D_availabl	.e'] = le.				e'])				
Process Communication Co	19]: df.i	lsna().sum()									
	Prod	luction expe	nse	0							
Face Action Company	Budg Movi	get .e_length		0 0							
Control Principle (See Principle (The Account of the Account of	Lead	_ l_Actress_ra	ting	0							
	Crit	ic_rating		0							
Description Comparison Co	3D_a Time	vailable e_taken		0							
Collection State of the control of the collection of the collecti	Genr Avg_	e _age_actors		0							
depth	Coll	ection	r	0							
2	dtyp	pe: int64									
Transition of Newton Control (and September 1) Transition of Newton Control (ıes							
		n sklearn.mo	odel_seled	ction im	mport train_	_test_split					
		rain,x_test,	y_train,	/_test =	train_test	t_split(x,y,	test_size=0.2	0,random_state=	0)		
Sec 1.50 conce	wcss	s = []		port KMe	eans						
	for	<pre>i in range (kmeans = KN</pre>	Means(n_cl			- 'k-means++	', random_sta	te = 42)			
			_	_							
132 15 16 16 16 16 16 16 16	32]: impc	ort matplotl	.ib.pyplot	as plt	:						
15	33]: plt.	plot(wcss))>]					
10 10 10 10 10 10 10 10	33]: plt. 33]: [<ma< td=""><td>plot(wcss)</td><td></td><td></td><td></td><td>)>]</td><td></td><td></td><td></td><td></td><td></td></ma<>	plot(wcss))>]					
	33]: plt. 33]: [<ma< td=""><td>plot(wcss)</td><td></td><td></td><td></td><td>)>]</td><td></td><td></td><td></td><td></td><td></td></ma<>	plot(wcss))>]					
11:	33]: plt. 33]: [<ma< td=""><td>plot(wcss)</td><td></td><td></td><td></td><td>0>]</td><td></td><td></td><td></td><td></td><td></td></ma<>	plot(wcss)				0>]					
1 1 1 1 1 1 1 1 1 1	33]: plt. 33]: [<ma< td=""><td>plot(wcss)</td><td></td><td></td><td></td><td>0>]</td><td></td><td></td><td></td><td></td><td></td></ma<>	plot(wcss)				0>]					
	33]: plt. 33]: [<ma< td=""><td>plot(wcss)</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></ma<>	plot(wcss)									
### Span	33]: plt. 33]: [<ma -="" -<="" 0.5="" 1.0="" 1.5="" 2.0="" td=""><td>plot(wcss)</td><td></td><td></td><td>:2cea7dac7f0</td><td></td><td></td><td></td><td></td><td></td><td></td></ma>	plot(wcss)			:2cea7dac7f0						
Moness	33]: plt. 33]: [<ma -="" -<="" 0.0="" 1.5="" 2.0="" td=""><td>plot(wcss)</td><td></td><td></td><td>:2cea7dac7f0</td><td></td><td></td><td></td><td></td><td></td><td></td></ma>	plot(wcss)			:2cea7dac7f0						
Sel: XWeans(n_clusters-2, random_state-42) 577): y_pred = Xweans.predict(x_best) 588): y_pred = Xweans.predict(x_best) 589: y_pred = Xweans.predict(x_best) 589: x_raxy([1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,	33]: plt. 33]: [<ma -="" 0.0="" 1.5="" 2.0="" []:<="" td=""><td>plot(wcss)</td><td>nes.Line2</td><td>2D at 0x</td><td>2cea7dac7f0</td><td></td><td></td><td></td><td></td><td></td><td></td></ma>	plot(wcss)	nes.Line2	2D at 0x	2cea7dac7f0						
	33]: plt. 33]: [<ma -="" 1.0="" 1.0<="" 1.5="" 2.0="" td=""><td>plot(wcss) atplotlib.li le12 a sklearn.cl ans = KMeans</td><td>nes.Line2 4 uster imp</td><td>Port KMe</td><td>6 8</td><td></td><td>andom_state =</td><td>42)</td><td></td><td></td><td></td></ma>	plot(wcss) atplotlib.li le12 a sklearn.cl ans = KMeans	nes.Line2 4 uster imp	Port KMe	6 8		andom_state =	42)			
Sel: y_red	33]: plt. 33]: [<ma -="" 0.0="" 0.5="" 1.0="" 1.5="" 2.0="" []:="" from="" kmea="" kmea<="" td=""><td>plot(wcss) atplotlib.li le12 a sklearn.cl ans = KMeans ans.fit(x_tr</td><td>nes.Line2 4 uster imp</td><td>Port KMe ers = 2, ain)</td><td>init = 'k-</td><td></td><td>andom_state =</td><td>42)</td><td></td><td></td><td></td></ma>	plot(wcss) atplotlib.li le12 a sklearn.cl ans = KMeans ans.fit(x_tr	nes.Line2 4 uster imp	Port KMe ers = 2, ain)	init = 'k-		andom_state =	42)			
array([1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,	33]: plt. 33]: [<ma -="" 0.0="" 0.5="" 10="" 15="" 2.0="" 56]:="" []:="" from="" km<="" kmea="" td="" =""><td>oplot(wcss) atplotlib.li le12 o 2 n sklearn.cl ans = KMeans ans.fit(x_tr</td><td>nes.Line2 4 uster imp (n_clusterain, y_trans=2, rans=2)</td><td>Port KMe ers = 2, ain) adom_sta</td><td>2cea7dac7f0 6 8 eans init = 'k- te=42)</td><td></td><td>andom_state =</td><td>42)</td><td></td><td></td><td></td></ma>	oplot(wcss) atplotlib.li le12 o 2 n sklearn.cl ans = KMeans ans.fit(x_tr	nes.Line2 4 uster imp (n_clusterain, y_trans=2, rans=2)	Port KMe ers = 2, ain) adom_sta	2cea7dac7f0 6 8 eans init = 'k- te=42)		andom_state =	42)			
1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,	33]: plt. 33]: [<ma -="" 0.0="" 0.5="" 10="" 15="" 2.0="" 56]:="" []:="" from="" kmea="" td="" y_pr<="" =""><td>plot(wcss) atplotlib.li le12 o 2 n sklearn.cl ans = KMeans ans.fit(x_tr ans (n_cluste red = kmeans</td><td>nes.Line2 4 uster imp (n_clusterain, y_trans=2, rans=2)</td><td>Port KMe ers = 2, ain) adom_sta</td><td>2cea7dac7f0 6 8 eans init = 'k- te=42)</td><td></td><td>andom_state =</td><td>42)</td><td></td><td></td><td></td></ma>	plot(wcss) atplotlib.li le12 o 2 n sklearn.cl ans = KMeans ans.fit(x_tr ans (n_cluste red = kmeans	nes.Line2 4 uster imp (n_clusterain, y_trans=2, rans=2)	Port KMe ers = 2, ain) adom_sta	2cea7dac7f0 6 8 eans init = 'k- te=42)		andom_state =	42)			
from sklearn.metrics import confusion matrix, ConfusionMatrixDisplay con = confusion_matrix(y_test,y_pred. Tabels=[1,0]) disp = ConfusionMatrixDisplay(confusion_matrix=cm , display_labels=['True', 'Spam']) disp.plot() sklearn.metricsplot.confusion_matrix.confusionMatrixDisplay at 0x2cea7b72170> Tue	33]: plt. 33]: [<ma -="" 0.0="" 0.5="" 10="" 15="" 20="" 56]:="" 58]:="" []:="" from="" kmea="" td="" y_pr="" y_pr<="" =""><td>plot(wcss) atplotlib.li le12 a sklearn.cl ans = KMeans ans.fit(x_tr ans(n_cluster ans(n_cluster ans(n_1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1</td><td>nes.Line2 4 uster imp s(n_clusterain,y_trans=2, rans) predict 0, 1, 1, 1, 1, 0,</td><td>Port KMe ers = 2, ain) adom_sta (x_test) 1, 1, 1, 0,</td><td>2cea7dac7f0 6 8 eans init = 'k- te=42) 1, 1, 0, 0, 0, 1, 1, 1,</td><td>O, O, 1, 1, 1, 1, 1, 0, 1,</td><td>, 1, 1, 1, 1, , 0, 0, 1, 1,</td><td>1, 1, 1, 0,</td><td></td><td></td><td></td></ma>	plot(wcss) atplotlib.li le12 a sklearn.cl ans = KMeans ans.fit(x_tr ans(n_cluster ans(n_cluster ans(n_1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1	nes.Line2 4 uster imp s(n_clusterain,y_trans=2, rans) predict 0, 1, 1, 1, 1, 0,	Port KMe ers = 2, ain) adom_sta (x_test) 1, 1, 1, 0,	2cea7dac7f0 6 8 eans init = 'k- te=42) 1, 1, 0, 0, 0, 1, 1, 1,	O, O, 1, 1, 1, 1, 1, 0, 1,	, 1, 1, 1, 1, , 0, 0, 1, 1,	1, 1, 1, 0,			
disp = ConfusionMatrixDisplay(confusion_matrix=cm , display_labels=['True','Spam']) import seaborn as sns disp.plot() caklearn.matricaplot.confusion_matrix.ConfusionMatrixDisplay st 0x2cea7b72170> Tue	33]: plt. 33]: [<ma -="" 0.0="" 0.5="" 10="" 15="" 20="" 56]:="" 58]:="" []:="" from="" kmea="" td="" y_pr="" y_pr<="" =""><td>plot(wcss) atplotlib.li le12 an sklearn.cl ans = KMeans ans.fit(x_tr ans(n_cluste red = kmeans red ay([1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,</td><td>nes.Line2 4 uster imp (n_cluster ain, y_trans=2, rans) predict 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,</td><td>Port KMe ers = 2, ain) 1, 1, 1, 0, 1, 0, 1, 1,</td><td>2cea7dac7f0 8 eans init = 'k- te=42) 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1,</td><td>O, O, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,</td><td>, 1, 1, 1, 1, , 0, 0, 1, 1, , 1, 1, 1, 1,</td><td>1, 1, 1, 0, 1, 1,</td><td></td><td></td><td></td></ma>	plot(wcss) atplotlib.li le12 an sklearn.cl ans = KMeans ans.fit(x_tr ans(n_cluste red = kmeans red ay([1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,	nes.Line2 4 uster imp (n_cluster ain, y_trans=2, rans) predict 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,	Port KMe ers = 2, ain) 1, 1, 1, 0, 1, 0, 1, 1,	2cea7dac7f0 8 eans init = 'k- te=42) 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1,	O, O, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,	, 1, 1, 1, 1, , 0, 0, 1, 1, , 1, 1, 1, 1,	1, 1, 1, 0, 1, 1,			
disp.plot() <pre></pre>	33]: plt. 33]: [<ma -="" 0.0="" 0.5="" 10="" 15="" 20="" 56]:="" 58]:="" []:="" from="" from<="" kmea="" td="" y_pr="" =""><td>plot(wcss) atplotlib.li le12 ans = KMeans ans.fit(x_tr ans(n_cluster ans(n_cluster ans(n_cluster ans) ans = kmeans ans.fit(x_tr ans(n_cluster ans) ans.fit(x_tr ans) ans.fit(x_tr) ans.fit(x_t</td><td>nes.Line2 4 .uster imp (n_clusterain,y_trans=2, rar .predict 0, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, etrics imp</td><td>Port KMe ers = 2, ain) 1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0,</td><td>2cea7dac7f0 6 8 eans init = 'k- te=42) 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 0, 1, afusion_matr</td><td>0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1) rix, Confusion</td><td>, 1, 1, 1, 1, , 0, 0, 1, 1, , 1, 1, 1, 1, , 1, 1, 1, 1,</td><td>1, 1, 1, 0, 1, 1, 1, 0,</td><td></td><td></td><td></td></ma>	plot(wcss) atplotlib.li le12 ans = KMeans ans.fit(x_tr ans(n_cluster ans(n_cluster ans(n_cluster ans) ans = kmeans ans.fit(x_tr ans(n_cluster ans) ans.fit(x_tr ans) ans.fit(x_tr) ans.fit(x_t	nes.Line2 4 .uster imp (n_clusterain,y_trans=2, rar .predict 0, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, etrics imp	Port KMe ers = 2, ain) 1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0,	2cea7dac7f0 6 8 eans init = 'k- te=42) 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 0, 1, afusion_matr	0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1) rix, Confusion	, 1, 1, 1, 1, , 0, 0, 1, 1, , 1, 1, 1, 1, , 1, 1, 1, 1,	1, 1, 1, 0, 1, 1, 1, 0,			
Saklearn.metricsplot.confusion_matrix.ConfusionMatrixDisplay at 0x2cea7b72170> Tue	33]: plt. 33]: [<ma -="" 0.0="" 0.5="" 10="" 15="" 20="" 56]:="" 59]:="" []:="" cm=" </td" from="" kmea="" self="" =""><td>n sklearn.cl ans = KMeans ans.fit(x_tr ans(n_cluster ans(n_cluster ans(n_0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0</td><td>nes.Line2 4 4 .uster imp s(n_clusterain,y_trans=2, rans=2, rans=2,</td><td>Port KMe ers = 2, ain) 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, port contest, y</td><td>2cea7dac7f0 6 8 eans init = 'k- te=42) 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0</td><td>0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1) rix, Confusion Ls=[1,0])</td><td>, 1, 1, 1, 1, , 0, 0, 1, 1, , 1, 1, 1, 1, , 1, 1, 1, 1,</td><td>1, 1, 1, 0, 1, 1, 1, 0,</td><td></td><td></td><td></td></ma>	n sklearn.cl ans = KMeans ans.fit(x_tr ans(n_cluster ans(n_cluster ans(n_0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0	nes.Line2 4 4 .uster imp s(n_clusterain,y_trans=2, rans=2,	Port KMe ers = 2, ain) 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, port contest, y	2cea7dac7f0 6 8 eans init = 'k- te=42) 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0	0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1) rix, Confusion Ls=[1,0])	, 1, 1, 1, 1, , 0, 0, 1, 1, , 1, 1, 1, 1, , 1, 1, 1, 1,	1, 1, 1, 0, 1, 1, 1, 0,			
True 37 21 35 30 -25 -20 5pm Fredicted label 5pm Fredicted label 5pm Fredicted label 52]: from sklearn.metrics import classification_report. cr = classification_report(y_test, y_pred) print(cr) precision recall f1-score support 0 0.28 0.18 0.22 44 1 0.51 0.64 0.56 58 accuracy 0.44 102 macro avg 0.39 0.41 0.39 102 weighted avg 0.41 0.44 0.42 102	33]: plt. 33]: [<ma 20<="" td=""><td>plot(wcss) atplotlib.li le12 an sklearn.cl ans = KMeans ans.fit(x_tr ans(n_cluster ans(n_cl</td><td>nes.Line2 4 uster imp (n_cluster ain, y_trans) ars=2, rans predict 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,</td><td>Port KMe ers = 2, ain) 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, port contest, y</td><td>2cea7dac7f0 6 8 eans init = 'k- te=42) 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0</td><td>0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1) rix, Confusion Ls=[1,0])</td><td>, 1, 1, 1, 1, , 0, 0, 1, 1, , 1, 1, 1, 1, , 1, 1, 1, 1,</td><td>1, 1, 1, 0, 1, 1, 1, 0,</td><td></td><td></td><td></td></ma>	plot(wcss) atplotlib.li le12 an sklearn.cl ans = KMeans ans.fit(x_tr ans(n_cluster ans(n_cl	nes.Line2 4 uster imp (n_cluster ain, y_trans) ars=2, rans predict 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,	Port KMe ers = 2, ain) 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, port contest, y	2cea7dac7f0 6 8 eans init = 'k- te=42) 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0	0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1) rix, Confusion Ls=[1,0])	, 1, 1, 1, 1, , 0, 0, 1, 1, , 1, 1, 1, 1, , 1, 1, 1, 1,	1, 1, 1, 0, 1, 1, 1, 0,			
From sklearn.metrics import classification_report or = classification_report(y_test, y_pred) print(cr) precision recall f1-score support 0 0.28 0.18 0.22 44 1 0.51 0.64 0.56 58 accuracy macro avg 0.39 0.41 0.39 102 weighted avg 0.41 0.44 0.42 102	33]: plt. 33]: [<ma 20<="" td=""><td>n sklearn.cl ans = KMeans ans.fit(x_tr ans(n_cluster ans(n_clu</td><td>nes.Line2 d uster imp (n_cluster ain, y_trans) (n_cluster ain, y_trans) predict 0, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,</td><td>Port KMe ers = 2, ain) 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, port con test, y isplay(c</td><td>2cea7dac7f0 6 8 eans init = 'k- te=42) 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, afusion_matr pred, label confusion_ma</td><td>0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,</td><td>1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,</td><td>1, 1, 1, 0, 1, 1, 1, 0, Y =['True','Spam'</td><td></td><td></td><td></td></ma>	n sklearn.cl ans = KMeans ans.fit(x_tr ans(n_cluster ans(n_clu	nes.Line2 d uster imp (n_cluster ain, y_trans) (n_cluster ain, y_trans) predict 0, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,	Port KMe ers = 2, ain) 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, port con test, y isplay(c	2cea7dac7f0 6 8 eans init = 'k- te=42) 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, afusion_matr pred, label confusion_ma	0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,	1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,	1, 1, 1, 0, 1, 1, 1, 0, Y =['True','Spam'			
Spam 36 8 -15 -10	33]: plt. 33]: [<ma 20<="" td=""><td>n sklearn.cl ans = KMeans ans.fit(x_tr ans(n_cluster ans(n_clu</td><td>nes.Line2 d uster imp (n_cluster ain, y_trans) (n_cluster ain, y_trans) predict 0, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,</td><td>Port KMe ers = 2, ain) 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, port con test, y isplay(c</td><td>2cea7dac7f0 6 8 eans init = 'k- te=42) 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, pred, label confusion_matr pred, label confusion_matrix.co</td><td>0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,</td><td>1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,</td><td>1, 1, 1, 0, 1, 1, 1, 0, Y =['True','Spam'</td><td></td><td></td><td></td></ma>	n sklearn.cl ans = KMeans ans.fit(x_tr ans(n_cluster ans(n_clu	nes.Line2 d uster imp (n_cluster ain, y_trans) (n_cluster ain, y_trans) predict 0, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,	Port KMe ers = 2, ain) 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, port con test, y isplay(c	2cea7dac7f0 6 8 eans init = 'k- te=42) 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, pred, label confusion_matr pred, label confusion_matrix.co	0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,	1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,	1, 1, 1, 0, 1, 1, 1, 0, Y =['True','Spam'			
Spam 36 8 -15 -10	33]: plt. 33]: [<ma -="" 0.0="" 0.5="" 10="" 15="" 20="" 56]:="" 58]:="" 60]:="" 61]:="" <skl<="" cm=" " disp="" from="" kmea="" td="" v_pr="" y_pr="" =""><td>n sklearn.cl ans = KMeans ans.fit(x_tr ans (n_cluster ans (</td><td>nes.Line2 d uster imp (n_cluster ain, y_trans) (n_cluster ain, y_trans) predict 0, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,</td><td>Port KMe ers = 2, ain) ndom_sta (x_test) 1, 1, 1, 0, 1, 1, 1, 1, port con _test,y_ isplay(confusion</td><td>2cea7dac7f0 8 eans init = 'k- te=42) 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, infusion_matr pred, label confusion_ma n_matrix.Co</td><td>0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,</td><td>1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,</td><td>1, 1, 1, 0, 1, 1, 1, 0, Y =['True','Spam'</td><td></td><td></td><td></td></ma>	n sklearn.cl ans = KMeans ans.fit(x_tr ans (n_cluster ans (nes.Line2 d uster imp (n_cluster ain, y_trans) (n_cluster ain, y_trans) predict 0, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,	Port KMe ers = 2, ain) ndom_sta (x_test) 1, 1, 1, 0, 1, 1, 1, 1, port con _test,y_ isplay(confusion	2cea7dac7f0 8 eans init = 'k- te=42) 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, infusion_matr pred, label confusion_ma n_matrix.Co	0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,	1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,	1, 1, 1, 0, 1, 1, 1, 0, Y =['True','Spam'			
True Spam Predicted label	33]: plt. 33]: [<ma 20<="" td=""><td>n sklearn.cl ans = KMeans ans.fit(x_tr ans (n_cluster ans (</td><td>nes.Line2 d uster imp (n_cluster ain, y_trans) (n_cluster ain, y_trans) predict 0, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,</td><td>Port KMe ers = 2, ain) ndom_sta (x_test) 1, 1, 1, 0, 1, 1, 1, 1, port con _test,y_ isplay(confusion</td><td>2cea7dac7f0 6 8 eans init = 'k- te=42) 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, infusion_matr pred, label confusion_ma n_matrix.Co</td><td>0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,</td><td>1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,</td><td>1, 1, 1, 0, 1, 1, 1, 0, Y =['True','Spam'</td><td></td><td></td><td></td></ma>	n sklearn.cl ans = KMeans ans.fit(x_tr ans (n_cluster ans (nes.Line2 d uster imp (n_cluster ain, y_trans) (n_cluster ain, y_trans) predict 0, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,	Port KMe ers = 2, ain) ndom_sta (x_test) 1, 1, 1, 0, 1, 1, 1, 1, port con _test,y_ isplay(confusion	2cea7dac7f0 6 8 eans init = 'k- te=42) 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, infusion_matr pred, label confusion_ma n_matrix.Co	0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,	1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,	1, 1, 1, 0, 1, 1, 1, 0, Y =['True','Spam'			
Tue	33]: plt. 33]: [<ma 20<="" td=""><td>n sklearn.cl ans = KMeans ans.fit(x_tr ans (n_cluster ans (</td><td>nes.Line2 d uster imp (n_cluster ain, y_trans) (n_cluster ain, y_trans) predict 0, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,</td><td>Port KMe ers = 2, ain) ndom_sta (x_test) 1, 1, 1, 0, 1, 1, 1, 1, port con _test,y_ isplay(confusion</td><td>22cea7dac7f0 6 8 eans init = 'k- te=42) 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0 atrix.Co -35 -30 -25</td><td>0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,</td><td>1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,</td><td>1, 1, 1, 0, 1, 1, 1, 0, Y =['True','Spam'</td><td></td><td></td><td></td></ma>	n sklearn.cl ans = KMeans ans.fit(x_tr ans (n_cluster ans (nes.Line2 d uster imp (n_cluster ain, y_trans) (n_cluster ain, y_trans) predict 0, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,	Port KMe ers = 2, ain) ndom_sta (x_test) 1, 1, 1, 0, 1, 1, 1, 1, port con _test,y_ isplay(confusion	22cea7dac7f0 6 8 eans init = 'k- te=42) 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0 atrix.Co -35 -30 -25	0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,	1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,	1, 1, 1, 0, 1, 1, 1, 0, Y =['True','Spam'			
### Predicted label from sklearn.metrics import classification_report	33]: plt. 33]: [<ma 20<="" td=""><td>n sklearn.cl ns = KMeans ns.fit(x_tr ns(n_cluster red = kmeans red ny([1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1</td><td>nes.Line2 d uster imp (n_cluster ain, y_trans) (n_cluster ain, y_trans) predict 0, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,</td><td>Port KMe ers = 2, ain) ndom_sta (x_test) 1, 1, 1, 0, 1, 1, 1, 1, port con _test,y_ isplay(confusion)</td><td>22cea7dac7f0 6 8 eans init = 'k- te=42) 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0</td><td>0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,</td><td>1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,</td><td>1, 1, 1, 0, 1, 1, 1, 0, Y =['True','Spam'</td><td></td><td></td><td></td></ma>	n sklearn.cl ns = KMeans ns.fit(x_tr ns(n_cluster red = kmeans red ny([1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1	nes.Line2 d uster imp (n_cluster ain, y_trans) (n_cluster ain, y_trans) predict 0, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,	Port KMe ers = 2, ain) ndom_sta (x_test) 1, 1, 1, 0, 1, 1, 1, 1, port con _test,y_ isplay(confusion)	22cea7dac7f0 6 8 eans init = 'k- te=42) 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0	0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,	1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,	1, 1, 1, 0, 1, 1, 1, 0, Y =['True','Spam'			
<pre>cr = classification_report(y_test, y_pred) print(cr)</pre>	33]: plt. 33]: [<ma 20<="" td=""><td>n sklearn.cl n sklearn.me ced ny([1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,</td><td>nes.Line2 duster imp s(n_cluster ain, y_tra rs=2, ran s.predict 0, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, etrics imp matrix(y_ onMatrixDi as sns</td><td>Port KMe ers = 2, ain) 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, port con test, y isplay (confusion confusion</td><td>22cea7dac7f0 8 2ans init = 'k- te=42) 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, infusion_matr pred, label confusion_ma n_matrix.Co -35 -30 -25 -20 -15</td><td>0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,</td><td>1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,</td><td>1, 1, 1, 0, 1, 1, 1, 0, Y =['True','Spam'</td><td></td><td></td><td></td></ma>	n sklearn.cl n sklearn.me ced ny([1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,	nes.Line2 duster imp s(n_cluster ain, y_tra rs=2, ran s.predict 0, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, etrics imp matrix(y_ onMatrixDi as sns	Port KMe ers = 2, ain) 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, port con test, y isplay (confusion confusion	22cea7dac7f0 8 2ans init = 'k- te=42) 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, infusion_matr pred, label confusion_ma n_matrix.Co -35 -30 -25 -20 -15	0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,	1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,	1, 1, 1, 0, 1, 1, 1, 0, Y =['True','Spam'			
print(cr) precision recall f1-score support 0 0.28 0.18 0.22 44 1 0.51 0.64 0.56 58 accuracy 0.44 102 macro avg 0.39 0.41 0.39 102 weighted avg 0.41 0.44 0.42 102	33]: plt. 33]: [<ma 20<="" td=""><td>n sklearn.cl n sklearn.me ced ny([1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,</td><td>nes.Line2 duster imp s(n_cluster ain, y_trans=2, rans) s.predict 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,</td><td>Port KMe ers = 2, ain) adom_sta (x_test) 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, port con test, y_ isplay (confusion confusion Spam</td><td>22cea7dac7f0 8 2ans init = 'k- te=42) 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, infusion_matr pred, label confusion_ma n_matrix.Co -35 -30 -25 -20 -15</td><td>0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,</td><td>1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,</td><td>1, 1, 1, 0, 1, 1, 1, 0, Y =['True','Spam'</td><td></td><td></td><td></td></ma>	n sklearn.cl n sklearn.me ced ny([1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,	nes.Line2 duster imp s(n_cluster ain, y_trans=2, rans) s.predict 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,	Port KMe ers = 2, ain) adom_sta (x_test) 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, port con test, y_ isplay (confusion confusion Spam	22cea7dac7f0 8 2ans init = 'k- te=42) 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, infusion_matr pred, label confusion_ma n_matrix.Co -35 -30 -25 -20 -15	0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,	1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,	1, 1, 1, 0, 1, 1, 1, 0, Y =['True','Spam'			
0 0.28 0.18 0.22 44 1 0.51 0.64 0.56 58 accuracy 0.44 102 macro avg 0.39 0.41 0.39 102 weighted avg 0.41 0.44 0.42 102	33]: plt. 33]: [<ma 20<="" td=""><td>n sklearn.cl ans = KMeans ans.fit(x_tr ans(n_cluster ans(n_cluster ans(n_o, o, o</td><td>nes.Line2 4 uster imp (n_cluster ain, y_trans) ars=2, rans predict 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 1, 1, etrics imp matrix(y_onMatrixDi as sns Predicted lal</td><td>Port KMe ers = 2, ain) adom_sta (x_test) 1, 1, 1, 0, 1, 1, 1, 1, port con test, y_ isplay(confusion confusion Spam bel</td><td>22cea7dac7f0 6 8 eans init = 'k- te=42) 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1,</td><td>0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1) rix, Confusion s=[1,0]) atrix=cm , d</td><td>1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,</td><td>1, 1, 1, 0, 1, 1, 1, 0, Y =['True','Spam'</td><td></td><td></td><td></td></ma>	n sklearn.cl ans = KMeans ans.fit(x_tr ans(n_cluster ans(n_cluster ans(n_o, o, o	nes.Line2 4 uster imp (n_cluster ain, y_trans) ars=2, rans predict 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 1, 1, etrics imp matrix(y_onMatrixDi as sns Predicted lal	Port KMe ers = 2, ain) adom_sta (x_test) 1, 1, 1, 0, 1, 1, 1, 1, port con test, y_ isplay(confusion confusion Spam bel	22cea7dac7f0 6 8 eans init = 'k- te=42) 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1,	0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1) rix, Confusion s=[1,0]) atrix=cm , d	1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,	1, 1, 1, 0, 1, 1, 1, 0, Y =['True','Spam'			
1 0.51 0.64 0.56 58 accuracy 0.44 102 macro avg 0.39 0.41 0.39 102 weighted avg 0.41 0.44 0.42 102	33]: plt. 33]: [<ma 20<="" td=""><td>n sklearn.cl ans = KMeans ans.fit(x_tr ans (n_cluster ans (</td><td>nes.Line2 duster imp s(n_cluster imp matrix(y_onMatrixDi as sns Predicted lal etrics imp</td><td>Port KMe ers = 2, ain) ndom_sta (x_test) 1, 1, 1, 0, 1, 1, 1, 1, port con test, y_ isplay (confusion confusion splay (confusion confusion confusion splay (confusion confusion confusion</td><td>22cea7dac7f0 8 2ans init = 'k- te=42) 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1,</td><td>0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,</td><td>1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,</td><td>1, 1, 1, 0, 1, 1, 1, 0, Y =['True','Spam'</td><td></td><td></td><td></td></ma>	n sklearn.cl ans = KMeans ans.fit(x_tr ans (n_cluster ans (nes.Line2 duster imp s(n_cluster imp matrix(y_onMatrixDi as sns Predicted lal etrics imp	Port KMe ers = 2, ain) ndom_sta (x_test) 1, 1, 1, 0, 1, 1, 1, 1, port con test, y_ isplay (confusion confusion splay (confusion confusion confusion splay (confusion confusion	22cea7dac7f0 8 2ans init = 'k- te=42) 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1,	0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,	1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,	1, 1, 1, 0, 1, 1, 1, 0, Y =['True','Spam'			
macro avg 0.39 0.41 0.39 102 weighted avg 0.41 0.44 0.42 102	33]: plt. 33]: [<ma 20<="" td=""><td>plot (wcss) atplotlib.li tel2 a sklearn.cl ans = KMeans ans.fit (x_tr ans (n_cluste ced = kmeans ced ay([1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0,</td><td>nes.Line2 4 4 uster imp (n_cluster ain, y_trans) (n_cluster ain, y_t</td><td>Port KMe ers = 2, ain) adom_sta (x_test) 1, 1, 1, 0, 1, 1, 1, 1, 20ort confusion confusion confusion 21 8 Spam bel port cla crt(y_test)</td><td>22cea7dac7f0 6 8 eans init = 'k- te=42) 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1,</td><td>0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,</td><td>1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,</td><td>1, 1, 1, 0, 1, 1, 1, 0, Y =['True','Spam'</td><td></td><td></td><td></td></ma>	plot (wcss) atplotlib.li tel2 a sklearn.cl ans = KMeans ans.fit (x_tr ans (n_cluste ced = kmeans ced ay([1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0,	nes.Line2 4 4 uster imp (n_cluster ain, y_trans) (n_cluster ain, y_t	Port KMe ers = 2, ain) adom_sta (x_test) 1, 1, 1, 0, 1, 1, 1, 1, 20ort confusion confusion confusion 21 8 Spam bel port cla crt(y_test)	22cea7dac7f0 6 8 eans init = 'k- te=42) 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1,	0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,	1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,	1, 1, 1, 0, 1, 1, 1, 0, Y =['True','Spam'			
	33]: plt. 33]: [<ma 20<="" td=""><td>plot (wcss) atplotlib.li tel2 a sklearn.cl ans = KMeans ans.fit(x_tr ans(n_cluste ced = kmeans ced ay([1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 0</td><td>nes.Line2 4 4 uster imp (n_cluster ain, y_trans) (n_cluster ain,</td><td>Port KMe ers = 2, ain) ndom_sta (x_test) 1, 1, 1, 0, 1, 1, 1, 1, port con test, y_ isplay(confusion confusion confu</td><td>22cea7dac7f0 6 8 eans init = 'k- te=42) 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 1, 1, 0, 1, 0, 0, 0, 1, 1, 1, 0,</td><td>0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,</td><td>1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,</td><td>1, 1, 1, 0, 1, 1, 1, 0, Y =['True','Spam'</td><td></td><td></td><td></td></ma>	plot (wcss) atplotlib.li tel2 a sklearn.cl ans = KMeans ans.fit(x_tr ans(n_cluste ced = kmeans ced ay([1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 0	nes.Line2 4 4 uster imp (n_cluster ain, y_trans) (n_cluster ain,	Port KMe ers = 2, ain) ndom_sta (x_test) 1, 1, 1, 0, 1, 1, 1, 1, port con test, y_ isplay(confusion confusion confu	22cea7dac7f0 6 8 eans init = 'k- te=42) 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 1, 1, 0, 1, 0, 0, 0, 1, 1, 1, 0,	0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,	1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,	1, 1, 1, 0, 1, 1, 1, 0, Y =['True','Spam'			
	33]: plt. 33]: [<ma 20<="" td=""><td>n sklearn.cl ans = KMeans ans.fit(x_tr ans(n_cluster ans(n_cluster ans = ced ay([1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1,</td><td>nes.Line2 4 4 uster imp (n_cluster in, y_trans) (n_</td><td>Port KMe ers = 2, ain) ndom_sta (x_test) 1, 1, 1, 0, 1, 1, 1, 1, port confusion confusion standard (x_test) 21 8 Spam bel confusion confusion</td><td>22cea7dac7f0 8 2ans init = 'k- te=42) 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1,</td><td>0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,</td><td>1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,</td><td>1, 1, 1, 0, 1, 1, 1, 0, Y =['True','Spam'</td><td></td><td></td><td></td></ma>	n sklearn.cl ans = KMeans ans.fit(x_tr ans(n_cluster ans(n_cluster ans = ced ay([1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1,	nes.Line2 4 4 uster imp (n_cluster in, y_trans) (n_	Port KMe ers = 2, ain) ndom_sta (x_test) 1, 1, 1, 0, 1, 1, 1, 1, port confusion confusion standard (x_test) 21 8 Spam bel confusion	22cea7dac7f0 8 2ans init = 'k- te=42) 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1,	0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,	1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,	1, 1, 1, 0, 1, 1, 1, 0, Y =['True','Spam'			
	33]: plt. 33]: [<ma 20<="" td=""><td>n sklearn.cl ans = KMeans ans.fit(x_tr ans(n_cluster ans(n_cluster ans = ced ay([1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1,</td><td>nes.Line2 4 4 uster imp (n_cluster in, y_trans) (n_</td><td>Port KMe ers = 2, ain) ndom_sta (x_test) 1, 1, 1, 0, 1, 1, 1, 1, port confusion confusion standard (x_test) 21 8 Spam bel confusion confusion</td><td>22cea7dac7f0 8 2ans init = 'k- te=42) 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1,</td><td>0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,</td><td>1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,</td><td>1, 1, 1, 0, 1, 1, 1, 0, Y =['True','Spam'</td><td></td><td></td><td></td></ma>	n sklearn.cl ans = KMeans ans.fit(x_tr ans(n_cluster ans(n_cluster ans = ced ay([1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1,	nes.Line2 4 4 uster imp (n_cluster in, y_trans) (n_	Port KMe ers = 2, ain) ndom_sta (x_test) 1, 1, 1, 0, 1, 1, 1, 1, port confusion confusion standard (x_test) 21 8 Spam bel confusion	22cea7dac7f0 8 2ans init = 'k- te=42) 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1,	0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,	1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,	1, 1, 1, 0, 1, 1, 1, 0, Y =['True','Spam'			