	====== WINE Q ======	==== UAL ====	:=== ITY :===	=== PRE	=== DIC ===	==== 101T: ====	==== V ====	====	====	===	===	====	===	===	====
In [45]:	import ma	itplotl by impo eaborn	Lib.py ort st as sr	/plot :ats	as pl	lt									
In [46]:	import part import number of the skilled	arn in	np nport aset	metri	cs										
	<pre>data = po    "/hom  print(dat   data.head (1599, 12)</pre>	ne/dhar :a.shap		KUBUNT	U/DAT	ΓASET/	CSV_F	ILES/	winequ	ality	-red	.csv")			
Out[46]:	fixed acidity  0 7.4  1 7.8  2 7.8  3 11.2  4 7.4	0.70 0.88 0.76 0.28	0.0 0.0 0.0 0.0 0.0	id 00 00 00 04 56	1.9 2.6 2.3 1.9	0.076 0.098 0.092 0.075 0.076	fre	25.0 15.0 17.0 11.0		34.0 67.0 54.0 60.0 34.0	0.9978 0.9968 0.9970 0.9980 0.9978	3.51 3.20 3.26 3.16	0.56 0.68 0.65 0.58 0.56	9.4 9.8 9.8 9.8 9.8	5 5 5 6 5
In [47]: Out[47]:	data.isnu  fixed acidity volatile acid citric acid	i <b>ll().</b> , , lity	Fal Fal Fal	se se se											
our dataset h	residual suga chlorides free sulfur d total sulfur density pH sulphates alcohol quality dtype: bool	lioxide dioxide	Fal Fal Fal Fal Fal Fal												
In [48]: Out[48]:	data.desc	cribe()	volatile acidity	citric ac	cid	residual sugar	chlori	aes	ree sulfur dioxide 99.000000	total s dic 1599.00	xide	<b>density</b>	1599.000	pH	<b>sulphates</b> 599.000000
	mean     8.3196       std     1.7410       min     4.6000       25%     7.1000       50%     7.9000       75%     9.2000	096 0. 000 0. 000 0. 000 0.	527821 179060 120000 390000 520000 640000	0.2709 0.1948 0.0000 0.0900 0.2600 0.4200	301 300 300 300 300	2.538806 1.409928 0.900000 1.900000 2.200000 2.600000	0.087 0.047 0.012 0.070 0.079	065 2 000 000 000 2	15.874922 10.460157 1.000000 7.000000 14.000000 21.000000	22.00 38.00 62.00	5324 0000 0000 0000 0000	0.996747 0.001887 0.990070 0.995600 0.996750 0.997835	3.31: 0.154 2.740 3.210 3.310 3.400	1386 0000 0000 0000	0.658149 0.169507 0.330000 0.550000 0.620000 0.730000
In [49]:	checking to plt.subpl	the co	lgsize	e=(15,	etwe			nt att	ribute:		0000	1.003690	4.010		2.000000
Out[49]:	fixed acidity volatile acidity	- 1	-0.26 1 -0.55	0.67 -0.55 1	0.11 0.0019 0.14	0.094 0.061 0.2	-0.15 -0.011 -0.061	-0.11 0.076 0.036	0.67 0.022 0.36	-0.68 0.23 -0.54	0.18 -0.26 0.31	-0.2	0.12 -0.39 0.23		-0.8 -0.6
	residual sugar chlorides free sulfur dioxide total sulfur dioxide	- 0.094	0.0019 0.061 -0.011 0.076	0.14 0.2 -0.061 0.036	0.056 0.19	0.056 1 0.0056 0.047	0.19 0.0056 1 0.67	0.2 0.047 0.67	0.36 0.2 -0.022 0.071	-0.086 -0.27 0.07	0.005 0.37 0.052	-0.22	-0.13 -0.051 -0.19		- 0.4 - 0.2
	density :  pH :  sulphates :  alcohol :		0.022 0.23 -0.26	0.36 -0.54 0.31	0.36 -0.086 0.0055	0.2 -0.27 0.37	-0.022 0.07 0.052 -0.069	0.071 -0.066 0.043	0.34 0.15	-0.34 1 -0.2	0.15 -0.2 1	0.21	-0.17 -0.058 0.25		- 0.0 0.2 0.4
	quality ·	fixed acidity - 0.10	volatile acidity – 65.0-	citric acid -	residual sugar -	-0.13	free sulfur dioxide -	total sulfur dioxide – 61.0	-0.17 - Alguery	-0.058 上	o.25 - salbhates	0.48 - Iohool	quality - 1		0.6
In [50]:	data prepared the defining the x = np.as y = np.as	feature	es and	iloc[	:, :-						es				
In [51]:	<pre>standardizir from skle # it help X = prepr</pre>	earn in	n <mark>port</mark> Standa	prepr ardize	the	diffe					'S				
In [52]:		earn.mo X_test test_	odel_s :, y_t _size=	rain, =0.2,	y_te rando	est = om_sta	train <sub>-</sub> te=0)	_test	_split						
	print("Tr print("Te Training data Testing datas validation and	esting uset: (12 set: (326 and mod	datas 279, 11 0, 11)	set:", ) (1279, (320,) ection											
In [53]:	<pre>from skle from skle # Number k_range = k_scores</pre>	earn.mo of k f range = []	odel_s *rom 1 e(1, 2	select 1 to 2 26)	ion i	import	cross	s_val	_score		4	4 00			
	# It'		ge: ghbors <sup>c</sup> old c	sClass cross val_sc	ifier <i>valic</i> ore(k	r(n_ne dation knn, X	ighboı <i>with</i>	rs=k) ′acc	uracy'	scor	ing				
In [54]:	# plotting  # matplot1  # Plot according  plt.plot(	ng the ib inl	accur Line ⁄ for	every	k nı		betwee	en 1	and 26						
Out[54]:	plt.ylabe plt.ylabe Text(0, 0.5,						y¹)								
	0.50 - 0.50 - 0.49 - 0.49 - 0.48 - 0.48	5	10	15	20	25									
In [55]:	# Train t knn = KNe knn.fit(X	the mod	KNN del ar	nd pre	dict			9)							
In [56]:	<pre>from skle # classif print(met</pre>	earn.me earn.me icatio	etrics etrics on rep classi	impo impo port f	rt ac rt rc or te ion_r	oc_auc est se report	_score t (y_te:	e st, y		digi	.ts=3,	zero_	_divis	ion=	1))
	print('ac	precisi 1.0 1.0	Lon 000 000 625 599	ccurac recall 0.000 0.000 0.704 0.599 0.333 0.000	f1-scc 0.0 0.0	ore su 900 900 662 599 340	pport  2 11 135 142 27 3	y_pr	ed))						
In [57]:	accuracy macro avg weighted avg accuracy 0.59  model g	ives	accı			0.590		ring	and 10	fola	Is				
In [58]:	accuracy print('cr	= cros	ss_val alidat e 0.54	L_scor ion s 72327044 with	e(knr core' 4025158	n, X, ; , acc	y, scouracy	oring mean	='accu ()) ng and	racy'	, cv		)	_	
In [59]:	<pre># Calcula print('ro</pre>	ion scor	c_auc _score	score	with c_auc	52270734 n mult. C_scor	711184 iclass	s par	ameter			)			
In [60]:	2. Logistic F  # import from skle	Regress <i>module</i> earn.li	sion e Inear_	_model	impo	ort Lo	gistio	cRegr	ession						
In [61]:  Out[61]:  In [62]:	<pre># Train a logreg = logreg.fi  LogisticRegre  # Predict</pre>	Logist t(X_tr	cicReç cain,	gressi y_tra	<b>in)</b> Ltinomi	ial', so	lass=	'mult	inomia	l', s	olvei	c='newt	con-cg	')	
In [63]:	<pre>y_pred =  # classif print(met print('ac</pre>	icatio	on rep classi	oort ificat	ion_r y_scc	report ore(y_				digi	ts=3,	zero_	_divis	ion=	1))
	3 4 5 6 7 8 accuracy macro avg weighted avg	0.7	000 689 638 321 000	0.000 0.000 0.770 0.634 0.333 0.000	0.6 0.7 0.6 0.3 0.6	000 727 636 327 000 634 282	2 11 135 142 27 3 320 320 320								
In [64]:	model g  # Calculation	ives of the cv	<i>score</i> ss_val	e with L_scor	'roc e(loc	c_auc_ greg, :	ovr′ s X, y,	scor	ing='r				=10)		
In [65]:	# Calcula print('rc y_tes	ion scor	e with _auc _score	score ', ro	0.812 <i>With</i> <b>c_auc</b>	22395666 n mult. c_scor	925059 iclas:	s par	ameter						
In [66]:	Adding poly  from skle  from skle  # Add pol	nomial arn.pr	featur eproc	res to t cessin	g imp ort m	oort P	olynor ipeli	nialF ne			le l				
		nomialF n make newton-	e_pipe	eline <mark>(</mark>	Polyr Logis	nomial sticRe	Featu	res(d			'mul1	inomia	al',		
<pre>In [67]: Out[67]:</pre>	# Train a poly = Po poly.fit( Pipeline(step	and fit Olynomi X_trai	the LalRectin, y_	<i>3rd d</i> gressi	egree on(3)	e poly									
In [68]: In [69]:	<pre># Test ou y_pred = # Classif</pre>	Logi It-of-s poly.p	sticRe Sample Oredic	gression e <i>test</i> et(X_t	n(multi solve set	i_class= er='newt			,						
	<pre>print(met print('ac  3 4 5 6 7</pre>		on 000 000 071 677		y_scc	ore(y_ ore su 000 000 684 647				digi	ts=3)	)))			
	accuracy macro avg weighted avg accuracy 0.61		283 313	0.000 0.312 0.616	0.6 0.2 0.6	616 293 612	3 320 320 320 320								
In [70]:	<pre># Calcula accuracy print('cr # Calcula print('rc y_tes</pre>	= cros	ss_val alidat c_auc _score	L_scor ion s score	e(pol core with c_auc	ly, X, with n mult. c_score	y, so roc_au iclass e(	corin uc_ov s par	g='roc r scor ameter		ovr', accı			)	
In [71]:	# Train a	F 0.64044  Free  earn.tr  and fit	ee in	nport Decis	Decis	sionTr	eeClas	ssifi ficat	er	del					
Out[71]: In [72]:	tree = De tree.fit( DecisionTreeC # Evaluat y_pred =	X_trai	in, y_er(rand	_train om_state	) e=0) out-				set						
In [73]:	<pre># Classif print(met print('ac</pre>	rics.c	classi	lficat ccurac	y_scc	ore(y_	test,			round	(), (	digits=	=3))		
	4 5 6 7 8 accuracy macro avg weighted avg accuracy 0.69	0.6	761 710 528 000	0.273 0.756 0.690 0.704 0.000	0.3 0.7 0.6 0.6 0.6	758 700 603 000 694 899	11 135 142 27 3 320 320 320 320								
In [74]:	<pre># Calcula accuracy print('cr # Calcula</pre>	ite cv = cros	score ss_val	e with L_scor	'roc e(tre	c_auc_e ee, X, with	ovr' s y, so roc_a	corin uc_ov	g='roc r scor					)	
In [75]:	print('ro y_tes cross validat roc_auc_score 4. Random	ion score 0.66147	ee.pre	roc_auc 60282	proba c_ovr s	a(X_te	st), r	829851	41795		'))				
Out[75]: In [76]:	<pre># Train a forest = forest.fi  RandomForestC # Test ou</pre>	Random	nFores	stClas y_tra om_state	sifi∈ in) ==0)						_stat	te=0)			
In [77]:	<pre>y_pred =  # Classif print(met print('ac</pre>	icatio rics.c	on rep classi	oort Lficat ccurac	ion_r y_sco	report ore(y_	test,			round	(), (	digits=	=3))		
	3 4 5 6 7 8 accuracy macro avg weighted avg	0.6 0.7 0.7 0.5 0.6	000 000 733 732 533 000	0.000 0.000 0.815 0.711 0.593 0.000	0.6 0.6 0.7 0.7 0.5 0.6	000 000 772 721 561 000	2 11 135 142 27 3 320 320 320								
	accuracy 0.70 /home/dhanola ning: Precisi _division` pa _warn_prf(a /home/dhanola ning: Precisi _division` pa _warn_prf(a /home/dhanola	d/anacond on and F trameter dverage, d/anacond on and F trameter dverage, d/anacond	to con modifi la3/lib score to con modifi la3/lib	are ill trol thi er, msg_ /python3 are ill trol thi er, msg_ /python3	l-defir is beha _start, 3.8/sit l-defir is beha _start, 3.8/sit	ned and avior. len(rece-packaned and avior. len(rece-packa	being s sult)) ges/skl being s sult)) ges/skl	et to ( earn/mo et to ( earn/mo	0.0 in l etrics/_ 0.0 in l etrics/_	abels v classif abels v	vith no ication vith no	on.py:124 on.py:124 o predict	ed samp :5: Unde :ed samp	les. U finedM les. U finedM	se `zero etricWar se `zero etricWar
In [78]:	model g  # Calcula accuracy	iverage,	accl	er, msg.  Uracy  with	is beha start,  Of  'roc	0.7093	sult)) 375 ovr's	scori	ng and	10 f	olds			les. U	se `zero
	print('cr # Calcula print('ro y_tes	ate roc oc_auc_ st, for	c_auc score est.p	score ', ro oredic	with c_auc t_pro	n mult. c_score bba(X_	iclas: e( test),	s par	ameter ti_cla				nean()	)	
In [79]:	# Fitting from skle classifie classifie	<i>Kerne</i> earn.sv er = SV	el SVM vm imp /C(ker	1 to toort S	he Tr VC rbf',	rand			)						
Out[79]: In [80]: In [81]:	<pre># Predict y_pred =  # making from skle</pre>	classi	lfier.	predi natrix	ct(X_										
Out[81]:		earn.me usion_ nap(cm,	etrics _matri	impo ix(y_t	rt co est, e, fn	y_pre	d )	rix							
	0 0 - 2 - 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	8 101 39 1 0	3 34 97 18	0 0 0 0 6 0 8 0 2 0		- 80 - 60 - 40 - 20									
In [83]:	# k-Fold from skle accuracie	cross earn.mo	valio	dation select	for ion i ore(e	import estima /=y_tr	cross tor=c <sup>·</sup> ain, c	s_val lassi	_score fier,		rain				
Out[83]: In [94]:	# we can accuracie /home/dhanola st populated warnings.wa 0.60436146653	es.mear /anaconc class ir urn(("The	da3/lib n y has e least	/python3 only 8 populat	3.8/sit member ted cla	ce-packa rs, whic ass in y	ges/skl h is le has on	ss than	n n_spli	ts=10.		py:666: digits=		rning:	The lea
	print(met print('ac		on o		y_sco	ore (y_ ore su 000 000 709 655 872				-110		S			
	accuracy macro avg weighted avg accuracy 0.64 /home/dhanola ning: Precisi _division` pa _warn_prf(a /home/dhanola	0.3 0.6 /anacond .on and F .rameter .verage, ./anacond	301 306 da3/lib score to con modifi da3/lib	0.288 0.644  /python3 are ill trol thi er, msg_ /python3	0.6 0.2 0.6 3.8/sit 1-defir is beha _start, 3.8/sit	644 289 621 ce-packa ned and avior. len(re	320 320 320 ges/skl being s sult)) ges/skl	et to (	0.0 in l	abels v classi1	vith no	predict on.py:124	ed samp 5: Unde	les. U finedM	se `zero etricWar
	ning: Precisi _division` pa _warn_prf(a /home/dhanola ning: Precisi _division` pa _warn_prf(a	on and Farameter average, and concerning the concerning to the concerning the concerning to the concerning the	to con modificallas/lib f-score to con modificallas	are ill trol thi er, msg_ /python3 are ill trol thi er, msg_	l-defir is beha _start, 3.8/sit l-defir is beha _start,	ned and avior. len(rece-packaned and avior. len(re	being s sult)) ges/skl being s sult))	et to (	0.0 in l	abels v classi1	vith no	predict on.py:124	ed samp 5: Unde	les. U finedM	se `zero etricWar
	====== THE EN	==== D	===	===:	===	====	====	===	===	===		====	===	===	====