dataset	41		de estate d	-1						salita	- 14			41	balanced account			4	A b-l		l		a laka asasah
dataset	tipo	tecnica	densidad	riesgo	pureza	n_train	n_test	n_features	es_grande	cv_spiits	n_iter	modelo	mejor_configuracion {'classifierC':	cv_f1_macro	cv_balanced_accuracy	cv_mcc	cv_conen_kappa	test_f1_macro	test_balanced_accuracy	test_mcc	test_conen_kappa	searcn_time_sec	n_jobs_searcn
													np.float64(9.877700294007917), 'classifier_class_weight: 'balanced', 'classifier_dual': False, 'classifier_ftt_intercept': False, 'classifier_penalty': 'l2', 'classifier_solver':										
plass	aumentado	pcsmote	50	25	entropia	268	43	9	FALSO	5	10	LogisticRegression	'saga', 'classifiertol': np.float64(2.3270677083837777e-05)}	0,73034431	0,77019425	0,6740974	0,664656793	0,596264846	0,746031746	0.54163513	0,515573227	0,493	4
<u> </u>	aumentado	pesmote	30	23	епсторна	200	40		1800	,	10	<u>Cognitive</u> gi ession	[classifier_c': np.float64(9.877700294007917), classifier_class_weight': 'balanced', 'classifier_dual': False, 'classifier_fit_intercept': False, 'classifier_penalty': 12', 'classifier_solver': 'saga', 'classifier_tot':	0,73034431	0,77013425	0,0740374	0,004030733	0,530204040	0,740031740	0,34103313	0,333773227	0,433	-
glass	aumentado	pcsmote	50	75	entropia	268	43	9	FALSO	5	10	LogisticRegression	np.float64(2.3270677083837777e-05)}	0,70374612	0,741736597	0,63441091	0,627837193	0,596264846	0,746031746	0,54163513	0,515573227	0,491	4
glass	base	base				149	65	9	FALSO	5	10	LogisticRegression	f'classifier_C: np.float64(0.4042872735027334), 'classifier_dlass_weight': 'balanced', 'classifier_dual': False, 'classifier_tintercept': True, 'classifier_penalty: '12', 'classifier_solver': 'liblinear', 'classifier_tol': np.float64(0.0008706020878304854)}	0,54994696	0,603114478	0,46841298	0,445878493	0,603764081	0,694271912	0,54605056	0,512956603	0,641	4
dataset	tipo	tecnica	densidad	riesgo	pureza	n_train	n_test	n_features	es_grande	cv_splits	n_iter	modelo	mejor_configuracion	cv_f1_macro	cv_balanced_accuracy	cv_mcc	cv_cohen_kappa	test_f1_macro	test_balanced_accuracy	test_mcc	test_cohen_kappa	search_time_sec	n_jobs_search
													"classifier_C: np.float64(0.4689400963537689), 'classifier_class_weight': 'balanced', 'classifier_flit_intercept': False, 'classifier_flit_intercept': False, 'classifier_penalty': '12', 'classifier_solver': 'Saga', 'classifier_tot':										
wdbc	aumentado	pcsmote	25	25	entropia	513	114	30	FALSO	5	10	LogisticRegression	np.float64(5.4041038546473305e-05)}	0,97206785	0,970114416	0,9458009	0,944222747	0,980955563	0,976190476	0,9626219	0,961923848	4,86	4
wdbc	aumentado	pcsmote	75	25	entropia	513	114	30	FALSO	5	10	LogisticRegression	rclassifier_c': rclassifier_class_weight': balanced', classifier_dual': False, classifier_dual': False, classifier_filer_penalty: '12', 'classifier_solver': saga', 'classifier_toi': np.float64(5.401038846473305e-05)	0,97206785	0,970114416	0,9458009	0,944222747	0,980955563	0,976190476	0,9626219	0,961923848	1,98	4
wdbc	base	base				398	171	30	FALSO	5	10	LogisticRegression	[classifier_C: np.float64(0.404872735027334), 'classifier_class_weight': 'balanced', 'classifier_dual': False, 'classifier_fit_intercept': True, 'classifier_partly': '12', 'classifier_solver': 'liblinear', 'classifier_tol': np.float64(0.008706020878304854)}	0,96791257	0,967655172		0,935868348	0,987514603	0,987514603		0,975029206	5,386	4
dataset	tipo	tecnica	densidad	riesgo	pureza	n train	n test	n features	es grande	cv splits	n iter	modelo	mejor configuracion {'classifierC':	cv f1 macro	cv balanced accuracy	cv mcc	cv cohen kappa	test f1 macro	test balanced accuracy	test mcc	test cohen kappa	search time sec	n jobs search
heart	aumentado	pcsmote	75.	25	proporcion	444	60	13	FALSO	51	10	LogisticRegression	Lidassirier	0,55160728	0,559953114	0,59573829	0,592746852	0,204761905	0,216071429	0,21416135	0,209809264	2,454	4
heart	aumentado		75		proporcion	444	60	13	FALSO	5	10	LogisticRegression	[classifier_C: np.float64(0.35849855803404745), 'classifier_class_weight': None, 'classifier_dual': False, 'classifier_fit_intercept': True, 'classifier_partly': 'tl', 'classifier_solver': 'liblinear', 'classifier_tol': np.float64(2.1930485556643678e-05))	0,5483699	0,556177289		0,592740632	0,254019665	0,262824675		0,203603264		4
heart	base	base				207	90	13	FALSO	5	10	LogisticRegression	[classifier_C: np.float64(0.4042872735027334), 'classifier_class_weight': 'balanced', 'classifier_dual': False, 'classifier_nti_intercept': True, 'classifier_partly':12, 'classifier_solver': 'liblinear', 'classifier_tol': np.float64(0.0008706020878304854))	0,31236192	0,326200452		0,301878153	0,374784041	0,407575758	0,37472798	0,370748299	1,339	4