	<pre>import import import import import from p</pre>	ting relevant pandas as numpy as numpy as natplotling seaborn and pickle important imp	pd np b.pyplot a s sns rt dump	as plt				
	from s from s from s from s from s	klearn.pre klearn.mod klearn.mod klearn.met klearn.com	processing el_select: el_select: rics impor pose impor	g import S ion import ion import rt accurac rt Column	train_te GridSear cy_score Transforme	chCV, train_test_s		
[2]:	<pre>from s #getti spirit</pre>	<pre>klearn.ense klearn.pipe ng the df = pd.read .head()</pre>	eline imp o	ort Pipeli	Lne		nalysis\\Datasets fo	or ML\\Classifier\\evilsp
t[2]:	id b 0 1 1 2 2 3	0.58 0.47 0.78	0.43 0.35 0.51		0.44	color type green Jinnat black Preta black Jinnat		
	3 4 4 5	0.57 0.41	0.88 0.25	0.42 0.44	2 0.64	green Bhoot green Jinnat		
[3]:	spirit <class RangeIn</class 	'pandas.co ndex: 900 columns (totolumn	ore.frame. entries, (tal 7 colu	.DataFrame				
	2 ro 3 ha 4 ha	one_length otting_fles air_length as_soul	900 no 900 no sh 900 no 900 no	on-null on-null on-null on-null	int64 float64 float64 float64			
[4]:	6 tydtypes memory #seein	olor ype : float64(usage: 49 g shape of spirit.shape	900 no 4), int64 .3+ KB		object object et(2)			
[5]:	(900, "#seein spirit	7) g the cate .type.value	gory wise					
t[5]: [6]:	#visua	300 300 300 type, dtype lizaing th irplot(spi	e dataset	- Itura I)				
t[6]:	_	rn.axisgri			0643b4e50>			-
	600 - .º 400 - 200 -							
	bone - 0.0 -				-			
	0.2 - 0.0 - 1.0 - 0.8 -		-					
	otting_flesh		-					
	- 0.0		-		-			
	0.2 - 0.0 - 1.0 -		-					
	0.8 - In 0.6 - sel 0.4 - 0.2 -		- - -					
[7]:		0 500 id		0.0 0.5 bone_le		0.0 0.5 1.0 rotting_flesh	0.0 0.5 1.0 hair_length	0.0 0.5 1.0 has_soul
t[7]:	spirit	=spirit.dro .head() e_length rotti 0.58			as_soul colo			
	1 2 3 4	0.47 0.78 0.57	0.35 0.51 0.88 0.25	0.81 0.64 0.42 0.44	0.79 blac	ck Preta ck Jinnat en Bhoot		
[8]: t[8]:	#seein spirit	g unique c	olors que()			en Jinnat blue', 'blood'], d	type=object)	
[9]:	x = sp y = sp	<pre>irit.drop(irit['type</pre>	'type',ax:	is=1)	from the i	ndependent variabi	es	
	numeri # Prep catego	rocessing rical_tran	ormer = P: for category sformer =	ipeline(st orical dan Pipeline	ta (steps = [('		coder(handle_unknown=	
[11]:	prepro	<pre>cessor = Co ansformers: ('num', : ('cat',)</pre>	olumnTrans =[numerical_	sformer(_transform_		e_length', 'rottir	sformer to each columns	
[12]: [13]:	# Spli X_trai	t the data	y_train, classifie	y_test =	train_tes	t_split(X, y,trair	n_size=0.80, test_siz	ze=0.2, random_state=42)
[14]:	# Defi	ne the pip	eline for	the class	sifier wit	th the preprocessor	and classifier	
[15]:	# Fit	ne = Pipel. the pipeli. ne.fit(X_t.	ne to the	('class	ocessor', sifier', c	preprocessor),		
[15]:	>	reprocesso num tandardSca	•					
		► Random	ForestCla	assifier				
	X_trai	the pipeli n_transfor t(X_train_	med = pipe transforme	eline['pre ed[:, :5],	eprocessor	'].fit_transform(X	(_train)	
[17]:	# make X_test	prediction transform c clf.pred	ns on the	line['prep	rocessor'].transform(X_test	:)	
[18]:	# Eval	uate the p	ipeline on	n the test	ting data	5], y_test)		
[19]:	#makin	cy: 0.41111 g the pred _pred = cl _pred	iction		ransformed	l[:, :5])		
[19]:		['Bhoot', 'Jinnat', 'Preta',	'Preta', 'Jinnat',	'Bhoot', 'Bhoot',	'Jinnat', 'Bhoot',	Preta', 'Bhoot', ' 'Jinnat', 'Bhoot' 'Jinnat', 'Preta', Preta', 'Jinnat',	, 'Jinnat', 'Bhoot',	
[20]:	# Defi	'Jinnat', 'Jinnat', 'Jinnat', 'Jinnat', 'Jinnat', 'Jinnat', 'Jinnat', 'Jinnat', 'Bhoot',	'Bhoot', 'Jinnat', 'Bhoot', 'Bhoot', 'Bhoot', 'Bhoot', 'Bhoot', 'Bhoot', 'Jinnat', 'Jinnat',	'Bhoot', 'Bhoot', 'Bhoot', 'Bhoot', 'Jinnat', 'Preta', 'Preta', 'Preta', 'Preta', 'Preta', 'Bhoot',	'Preta', 'Bhoot', 'Bhoot', 'Jinnat', 'Bhoot', 'Jinnat', Jinnat', Jinnat', Jinnat', 'Preta', 'Jinnat',	'Preta', 'Preta', 'Bhoot', 'Preta', 'Preta', 'Bhoot', 'Preta', 'Preta', 'Bhoot', 'Preta', 'Bhoot', 'Jinnat' 'Preta', 'Bhoot', 'Bhoot', 'Preta', 'Jinnat', 'Preta', 'Jinnat', 'Preta', 'Preta', 'Bhoot', 'Preta', 'Preta', 'Jinnat', 'Preta', 'Jinnat'], dtype=	'Bhoot', 'Bhoot', 'Preta', 'Preta', 'Jinnat', 'Jinnat', 'Jinnat', 'Bhoot', 'Preta', 'Jinnat',	
	param_'n	<pre>grid = { estimator ax_depth': in_samples in_samples</pre>	s': [100, [5, 10,15 _split':	200, 500, 5, 20,25], [2, 5, 10,	1000],			
[21]: [21]:	grid_s	GridSearch	idSearchC\ X_train_tı	J(estimatoransformeo	or=clf, pa	ram grid=param gri	.d, cv=5, n_jobs=-1)	
	1	earch.fit(idSearchC		ier			
		earch.fit() Gr imator: Ra • RandomF	andomFore		ch.best_p			
[22]:	# Prin print(y_pred print(Best pa	Gr imator: Ra RandomFr t the best 'Best para = grid_se. 'Accuracy:	hyperparameters:', arch.pred:', accurace {'max_dep	<pre>grid_sear ict(X_test cy_score(y oth': 5, '</pre>	_ /_test, y_	<pre>params_) rmed[:, :5]) pred))</pre>	samples_split': 2, '	
[22]: [23]: [24]:	# Prin print(y_pred print(Best pa Accurace # crea nclf =	Grimator: Ra ▶ RandomFr t the best Best paral = grid_se 'Accuracy: arameters: cy: 0.4944 ting an up	hyperparameters:', arch.pred: ', accurac {'max_dep} 444444444444444444444444444444444444	grid_sean ict(X_test cy_score(y oth': 5, ' 1446	/_test, y_	<pre>params_) med[:, :5]) pred)) es_leaf': 1, 'min_</pre>	samples_split': 2, '	
[23]:	# Prin print(y_pred print(Best pa Accurace # crea nclf = #fitti npipel # Fit	Grimator: Ra RandomFr t the best Best para = grid_se 'Accuracy: arameters: cy: 0.4944 ting an up RandomFore ng the new	hyperparameters:', arch.pred: ', accurac {'max_dep 4444444444 graded c. estClassin model line(steps	grid_sean ict(X_test cy_score(y oth': 5, ' 1446 lassifier fier(max_c) s=[('prepr ('class data	<pre>r_test, y_ min_sampl depth= 10,</pre>	<pre>params_) med[:, :5]) pred)) es_leaf': 1, 'min_ min_samples_leaf= preprocessor),</pre>	samples_split': 2, '	n_estimators': 200}
[23]: [24]:	# Prin print(y_pred print() Best pa Accurac # crea nclf = #fitti npipel # Fit npipel	Grimator: Ra imator: Ra → RandomF t the best 'Best para = grid_se. 'Accuracy: arameters: cy: 0.4944 ting an up RandomFore ng the new ine = Pipe the pipeli ine.fit(X_ reprocessor	hyperparameters:', arch.pred: ', accurace {'max_dep} 444444444 graded c. estClassin model line(steps ne to the train, y_t	grid_sean ict(X_test cy_score(y oth': 5, ' 1446 lassifier fier(max_c c' class data train) nTransfor cat	min_sampl depth= 10, cocessor', sifier', n	<pre>params_) med[:, :5]) pred)) es_leaf': 1, 'min_ min_samples_leaf= preprocessor),</pre>	samples_split': 2, '	n_estimators': 200}
[23]: [24]:	# Prin print(y_pred print() Best pa Accurac # crea nclf = #fitti npipel # Fit npipel	Grimator: Ra imator: Ra ration RandomF t the best Best para grid_se Accuracy: arameters: cy: 0.4944 ting an up RandomFore ng the new ine = Pipe the pipeli ine.fit(X_ reprocesso num tandardSca	hyperparameters:', arch.pred: ', accurace {'max_dep} 444444444 graded c. estClassin model line(steps ne to the train, y_t	grid_sean ict(X_test cy_score(y oth': 5, ') 1446 lassifier fier(max_c s=[('prepr ('class data train) cat neHotEnce	min_sampl depth= 10, cocessor', sifier', n	<pre>params_) med[:, :5]) pred)) es_leaf': 1, 'min_ min_samples_leaf= preprocessor),</pre>	samples_split': 2, '	n_estimators': 200}
[23]: [24]: [25]:	# Prin print(y_pred print() Best pa Accurace # crea nclf = #fitti npipel # Fit npipel # St # fit X_trai nclf.f	Gr imator: Ra imator: Ra ▶ RandomF t the best 'Best para = grid_se. 'Accuracy: arameters: cy: 0.4944 ting an up RandomFore ng the new ine = Pipe the pipeli ine.fit(X	hyperparameters:', arch.pred: ', accurace ', accurace ', accurace 'max_dep 444444444 graded c. estClassin model line(steps ne to the train, y_t Pipeline or: Column ForestClas ne on the med = npig	grid_sean ict(X_test cy_score(y oth': 5, ' 1446 lassifier fier(max_c s=[('prepr ('class data train) cat neHotEnco assifier training peline['pr med[:, :5]	min_sampl depth= 10, cocessor', sifier', n data ceprocesso	pred(:, :5]) pred()) es_leaf': 1, 'min_ min_samples_leaf= preprocessor), (clf)])	samples_split': 2, ' 8, min_samples_spli	n_estimators': 200}
[23]: [24]: [25]: [26]:	# Prin print(y_pred print() Best pa Accurace # crea nclf = #fitti npipel # Fit npipel * St * St RandomF # make	Grimator: Ra imator: Ra RandomF t the best 'Best para = grid_se 'Accuracy: arameters: cy: 0.4944 ting an up RandomFore ng the new ine = Pipe the pipeli ine.fit(X_ reprocessor num tandardSca Random * Random	hyperparameters:', arch.pred: ', accurace {'max_deparameters:', arch.pred: ', accurace {'max_deparameters:', arch.pred: ', accurace for the train, y_train Pipeline or: Column Pipeline ForestCla ine on the med = npip transform sifier(max n_e ns on the	grid_sean ict(X_test cy_score(y oth': 5, ' 1446 lassifier fier(max_c data train) cat neHotEnce assifier training peline['pn med[:, :5] RandomF _depth=10 stimators:	data reprocessor, y_train orestClas , min_samp =200)	pred(:, :5]) pred()) es_leaf': 1, 'min_ min_samples_leaf= preprocessor), clf()]) pr'].fit_transform() sifier ples_leaf=8, min_samples_leaf=8, min_samples_1	<pre>samples_split': 2, ' = 8, min_samples_spli [X_train] mples_split=5,</pre>	n_estimators': 200}
[23]: [24]: [25]: [26]:	# Prin print(y_pred print() Best pa Accurace # crea nclf = #fitti npipel # Fit npipel * St * St RandomF # make X_test y_pred # Eval accura	Gr. imator: Ra imator: Ra RandomF t the best 'Best paradale grid_sellow 'Accuracy: arameters: cy: 0.4944 ting an up RandomFore ng the new ine = Pipe the pipeli ine.fit(X_ reprocessor num tandardSca LandomFore *Random *Random *Random *Random *Random *Random *Random *TandardSca *Random *Random *TandardSca *Random *TandardSca *Random *Random *TandardSca *Random	hyperparameters:', arch.pred: ', accurace {'max_dep} 4444444444 graded c.estClassif model line(steps ne to the train, y_t Pipeline or: Column ForestCla ine on the med = npipe atransform sifier(max_ n_e ns on the ed = npipe edict(X_te ipeline or score(X_te	grid_sean ict(X_test cy_score(y oth': 5, ' 1446 lassifier fier(max_c s=[('prepr ('class data train) nTransfor cat neHotEnco assifier training peline['preprined[:, :5] RandomF depth=10 stimators: testing of est_transfor the test est_transf	data reprocessor, y_train orestClas , min_samp =200) data reprocessor corestClas , min_samp =200; data reprocessor corestClas , min_samp =200;	pred(:, :5]) pred()) es_leaf': 1, 'min_ min_samples_leaf= preprocessor), clf()[) sifier ples_leaf=8, min_samples_leaf=8, min_samples_1	<pre>samples_split': 2, ' = 8, min_samples_spli [X_train] mples_split=5,</pre>	n_estimators': 200}
[23]: [24]: [25]: [26]: [27]:	# Prin print(y_pred print() Best pa Accurace # crea nclf = #fitti npipel # Fit npipel * Fit npipel * St * Accurace # Fit npipel * Pit npipel * Accurace # Make X_trai nclf.f * RandomF # Eval accurace # makin y_test y_test y_test	Grimator: Ra imator: Ra RandomF t the best 'Best paralese arameters: cy: 0.4944 ting an up RandomFore mg the new ine = Pipe the pipelitine.fit(X_ reprocessor num tandardSca Random tandardSca Prediction transform it(X_train) ForestClass prediction transform the pipelitine.fit(X_Train)	hyperparameters:', arch.pred: ', accurace {'max_dep 444444444 graded c. estClassif model line(steps ne to the train, y_t Pipeline or: Column ForestCla and on the med = npipe transform sifier(max_ n_e ns on the ed = npipe edict(X_te ipeline or score(X_te ipeline o	grid_sean ict(X_test cy_score(y oth': 5, ' 1446 lassifier fier(max_c s=[('prepr ('class data train) cat neHotEnce ssifier training peline['prepr depth=10 stimators: testing cest_transfer trains the test est_transfer transfer transfer transfer transfer training peline['prepr est_transfer transfer transfer transfer training peline['prepr est_transfer transfer	min_sampl depth= 10, cocessor', sifier', n data ceprocesso , y_train orestClas , min_samp =200) data eprocessor cransformed[:,	pred(:,:5]) pred()) es_leaf': 1, 'min_ min_samples_leaf= preprocessor), clf()]) pred() preprocessor), clf()]) sifier ples_leaf=8, min_samples_leaf=8, min_sample	samples_split': 2, ' = 8, min_samples_split = [X_train] mples_split=5, = (t)	n_estimators': 200}
[23]: [24]: [25]: [26]: [27]:	# Prin print(y_pred print() Best pa Accurace # crea nclf = #fitti npipel # Fit npipel * Fit npipel * St * Accurace # Fit npipel * Pit npipel * Accurace # Make X_trai nclf.f * RandomF # Eval accurace # makin y_test y_test y_test	### Gr imator: Ra imator: Ra RandomF t the best Best paral grid_se. Accuracy: arameters: cy: 0.4944 ting an up RandomFore ng the new ine = Pipe the pipelitine.fit(X_ reprocessor num tandardSca Random the pipelitine.fit(X_ reprocessor num tandardSca Prediction transform the pipelitine.fit(X_ reprocessor num tandardSca Prediction transform the pipelitine.fit(X_ reprocessor num tandardSca Prediction transform the pipelitine.fit(X_ reprocessor num the pipelitine.fit(X_ reprocessor reprocessor num the pipelitine.fit(X_ reprocessor reprocessor num the pipelitine.fit(X_ reprocessor num the pipelitine.fit(X_ reprocessor reproces	hyperparameters:', arch.predi', accurace ', accurace '	grid_sean ict(X_test cy_score(y) th': 5, ' 446 lassifier fier(max_c) s=[('prepr ('class data train) cat neHotEnco cat neHotEnco ssifier RandomF depth=10 stimators: testing co est_transf cot y'') 778 t(X_test_t 'Bhoot', 'Bhoot', 'Bhoot', 'Jinnat', 'Jinnat', 'Jinnat', 'Jinnat', 'Jinnat',	min_sampl depth= 10, cocessor', sifier', n data ceprocessor , y_train orestClas , min_samp =200) data cormed[:, ting data cormed[:, ting data formed[:, the commodity 'Jinnat', 'Bhoot', 'Preta', 'Preta',	pred(:,:5]) pred()) es_leaf': 1, 'min_ min_samples_leaf= preprocessor), clf()]) sifier ples_leaf=8, min_sa c'].transform(X_test) sifier les_leaf=8, min_sa c'].transform(X_test) cd[:,:5]) Preta', 'Bhoot', ' 'Jinnat', 'Bhoot', ' 'Jinnat', 'Preta' 'Jinnat', 'Preta'	<pre>samples_split': 2, ' st) Bhoot', ,'Innat', ,'Bhoot', 'Preta', ,'Bhoot', ,'Preta',</pre>	n_estimators': 200}
[23]: [24]: [25]: [26]: [27]:	# Prin print(y_pred print() Best pa Accurace # crea nclf = #fitti npipel # Fit npipel * Fit npipel * St * Accurace # Fit npipel * Pit npipel * Accurace # Make X_trai nclf.f * RandomF # Eval accurace # makin y_test y_test y_test	### Gr imator: Ra imator: Ra RandomF ### RandomF ### Ling an up RandomFore ### Ling an up ### RandomFore ### RandomFore ### RandomFore ### RandomFore #### Ling an up ### RandomFore ### RandomFore #### RandomFore #### Ling an up #### RandomFore #### Ling an up ##### Ling an up ####################################	hyperparameters:', arch.preda', accurace ', accurace '	grid_sean ict(X_test cy_score(y) th': 5, ' 446 lassifier fier(max_c) s=[('prepr ('class data train) cat neHotEnco assifier training peline['prepr med[:, :5] RandomF depth=10 stimators testing of est_transf cat 'Preta', ' 'Bhoot', 'Jinnat',	min_sampl depth= 10, cocessor', sifier', n data reprocessor , y_train orestClas , min_samp =200) data reprocessor formed[:, ting data reprocessor , y_train orestClas , min_samp =200)	pred); med[:,:5]) pred)) es_leaf': 1, 'min_ min_samples_leaf= preprocessor), clf)]) sifier ples_leaf=8, min_sa "].transform(X_test) sifier ples_leaf=8, min_sa "].transform(X_test) sifier preta', 'Bhoot', ' 'Jinnat', 'Bhoot', 'Jinnat', 'Bhoot', 'Preta', 'Jinnat', 'Phoot', 'Preta', 'Bhoot', 'Preta', 'Bhoot', 'Preta', 'Bhoot', 'Bhoot', 'Bhoot', 'Bhoot',	samples_split': 2, ' 8, min_samples_spli X_train) mples_split=5, st) mples_split=5, st)	n_estimators': 200}
[23]: [24]: [25]: [26]: [27]:	# Prin print(y_pred print() Best pa Accurace # crea nclf = #fitti npipel # Fit npipel * Fit npipel * St * Accurace # Fit npipel * Pit npipel * Accurace # Make X_trai nclf.f * RandomF # Eval accurace # makin y_test y_test y_test	### Gr. ### Imator: Ra ### RandomF. ### the best ### Best paral ### grid_se. ### Accuracy: ### arandomFore ### ting an up ### RandomFore #### ting an up ###	hyperparameters:', arch.pred: ', accurace	grid_sean ict(X_test cy_score(y) th': 5, ' 446 lassifier fier(max_c) s=[('prepr ('class data train) nTransfor cat neHotEnco assifier training peline['prepr med[:, :5] RandomF depth=10 stimators testing co est_transfor 'Preta', ' 'Bhoot', ' 'Jinnat', ' '	min_sampl depth= 10, cocessor', sifier', n data ceprocesso , y_train orestClas , min_samp =200) data cormed[:, ting data cormed[:	rearams_) red[:, :5]) red() es_leaf': 1, 'min_ min_samples_leaf= preprocessor), clf()]) resifier res	samples_split': 2, ' 8, min_samples_spli 8, min_samples_spli "X_train) mples_split=5, "t) mples_split=5, "t) "t) "preta', 'Preta', 'Preta', 'Preta', 'Jinnat', 'Preta', 'Jinnat', 'Preta', 'Jinnat', 'Preta', 'Jinnat', 'Preta', 'Jinnat', 'Preta', 'Jinnat', 'Preta', 'Bhoot', 'Bhoot', 'Bhoot', 'Bhoot', 'Bhoot', 'Preta', 'Preta',	n_estimators': 200}
[23]: [24]: [25]: [26]: [27]:	# Prin print(y_pred print() Best pa Accurace # crea nclf = #fitti npipel # Fit npipel * Fit npipel * St * Accurace # Fit npipel * Pit npipel * Accurace # Make X_trai nclf.f * RandomF # Eval accurace # makin y_test y_test y_test	earch.fit() Gr imator: Ra Nation: Ra RandomF t the best Best para Gri t the best Best para Gri Accuracy: arameters: Cy: 0.4944 ting an up RandomFore Ing the new ine = Pipe the pipeli ine.fit(X_ reprocessor num tandardSca L Prediction transform Ing the pred prediction transform The pipeli n_transform Ing the pred prediction transform Ing the pred Ing transform Ing transform Ing transform Ing transform Ing the pred Ing transform Ing trans	hyperparameters:', arch.pred: ', accurace	grid_sean ict(X_test cy_score(y) th': 5, ' 446 lassifier fier(max_c) s=[('preproduction of the content of the	min_sampl depth= 10, cocessor', sifier', n data ceprocesso , y_train orestClas , min_samp =200) data cormed[:, ting data cormed[:	rearams_) red[:, :5]) red()) es_leaf': 1, 'min_ min_samples_leaf= preprocessor), clf()]) resifier reles_leaf=8, min_samples_leaf=8, min_samp	samples_split': 2, ' 8, min_samples_split 8, min_samples_split 8, min_samples_split 8, min_samples_split 8, min_samples_split 9, innat', 1, inn	n_estimators': 200}
[23]: [24]: [25]: [26]: [27]:	# Prin print(y_pred print() Best pa Accurace # crea nclf = #fitti npipel # Fit npipel # Fit npipel # Fit x_trai nclf.f RandomF # wake x_test y_pred # Eval accura print() Accurace #makin y_test y_test array()	earch.fit() Gr imator: Ra NandomF RandomF L the best Best para egrid_se Accuracy: arameters: cy: 0.4944 ting an up RandomFore ng the new ine = Pipe the pipeli ine.fit(X_ reprocesso num tandardSca Nandom the pipeli n_transform enclf.pr wate the p cy = nclf.pr l'Accuracy cy: 0.4777 g the pred pred = nc. pred ['Bhoot', 'Jinnat', 'Bhoot', 'Preta',	ndomFore orestClas orestClas meters:', arch.predi ', accurac ', ac	grid_sean ict(X_test cy_score(y) th': 5, ' 446 lassifier fier(max_G s=[('prepr ('class data train) nTransfor cat neHotEnco ssifier ssifier raining peline['predi	min_sampl depth= 10, cocessor', sifier', n data ceprocessor , y_train orestClas , min_samp =200) data cormed[:, ting data cormed[:, ting data cormed[:, the cormed or co	rarams_) rmed[:, :5]) pred)) es_leaf': 1, 'min_ min_samples_leaf= preprocessor), rclf)]) relfit_transform(r'].fit_transform(r'].fit_transform(r'].transform(r'].	Samples_split': 2, ' 8, min_samples_spli 8, min_samples_split X_train) X_train) mples_split=5, st) mples_split=5, st) st) Bhoot', 'Preta', 'Preta'	n_estimators': 200}
[23]: [24]: [25]: [26]: [27]: [29]:	# Prin print(y_pred print() Best pa Accurat # crea nclf = #fitti npipel # Fit npipel # Fit x_trai nclf.f RandomF # make x_test y_pred # Eval accura print(Accurat y_test y_test array()	earch.fit() Gr imator: Ra Nation: Ra RandomF LandomF LandomFore LandomFo	ndomFore orestClas orestClas hyperpara meters:', arch.pred: ', accurac ', acc	grid_sean ict(X_test CY_score() th': 5, ' 446 cat' lassifier fier(max_c data train) cat neHotEnco assifier training peline['prest_transf cat' lest_transf cat' training peline['prest_transf cat' 'Bhoot', 'Bhoot', 'Bhoot', 'Jinnat', 'Preta', 'Preta', 'Jinnat', 'Preta', 'Bhoot', 'Jinnat', 'Preta', 'Pr	data ceprocessor , sifier', n data ceprocessor , y_train orestClas , min_samp =200) data cormed[:, 'Jinnat', 'Bhoot', 'Bhoot', 'Preta', 'Preta', 'Bhoot', 'Preta', 'Bhoot', 'Preta', 'Bhoot', 'Preta', 'Bhoot', 'Preta', 'Bhoot', 'Preta', 'Bhoot', 'Jinnat', 'Preta', 'Bhoot', 'Preta', 'Bhoot', 'Jinnat', 'Preta', 'Jinnat',	rarams_) rmed[:, :5]) pred)) es_leaf': 1, 'min_ min_samples_leaf= preprocessor), rclf)]) relfit_transform(r'].fit_transform(r'].fit_transform(r'].transform(r'].	Samples_split': 2, ' 8, min_samples_spli 8, min_samples_split X_train) X_train) mples_split=5, st) mples_split=5, st) st) Bhoot', 'Preta', 'Preta'	n_estimators': 200} t= 5, n_estimators= 200)
[23]: [24]: [25]: [27]: [29]: [30]:	# Prin print(y_pred print() Best pa Accurat # crea nclf = #fitti npipel # Fit npipel # Fit x_trai nclf.f RandomF # make X_test y_pred # Eval accura print(Accurat y_test y_test array() # Load with of pi # Load with of pi	earch.fit() Gr imator: Ra NandomF RandomF Lathe best 'Best para = grid_se 'Accuracy: arameters: cy: 0.4944 Ling an up RandomFore ng the new ine = Pipe the pipeli ine.fit(X_ reprocesso num tandardSca Nandom the pipeli n_transform it(X_train) tandardSca Prediction transform it(X_train) tandardSca Prediction transform it(X_train) The pipeli n_transform it(X_train) The pred pred pred pred pred pred pred pred	ndomFore orestClas orestClas meters:', arch.pred: ', accurac ', ac	grid_sean ict(X_test CY_score(S) cth': 5, ' 446 cth': 5, ' 446 lassifier fier(max_c) s=[('preprint ('class data train) cat neHotEnco assifier training peline['preprint ('class data train) training peline['preprint ('class data train) stimators: testing of eline['preprint ('preprint ('prep	data ceprocessor , sifier', n data ceprocessor , y_train orestClas , min_samp =200) data cormed[:, 'Jinnat', 'Bhoot', 'Preta', 'Preta', 'Preta', 'Preta', 'Bhoot', 'Preta', 'Bhoot', 'Preta', 'Bhoot', 'Preta', 'Bhoot', 'Preta', 'Bhoot', 'Preta', 'Bhoot', 'Jinnat', 'Preta', 'Bhoot', 'Preta', 'Bhoot', 'Jinnat',	rarams_) rmed[:, :5]) pred)) es_leaf': 1, 'min_ min_samples_leaf= preprocessor), rclf)]) relfit_transform(r'].fit_transform(r'].fit_transform(r'].transform(r'].	Samples_split': 2, ' 8, min_samples_spli 8, min_samples_split X_train) X_train) mples_split=5, st) mples_split=5, st) st) Bhoot', 'Preta', 'Preta'	n_estimators': 200} t= 5, n_estimators= 200)
[23]: [24]: [25]: [26]: [27]: [29]: [30]:	# Prin print(y_pred print() Best pa Accurate # crea nclf = #fitti npipel # Fit npipel # Fit x_trai nclf.f RandomF # make x_test y_pred # Eval accura print(Accurate # makin y_test y_test array(X_test X_test	earch.fit () Gr imator: Ra imator: Ra RandomF RandomF the best Best para Gr imator: Ra RandomF the para Care and	ndomFore orestClas orestClas meters:', arch.pred ', accurac 'max_der 444444444 graded c. estClassin model line(steps ne to the train, y_ Pipeline or: Column or: Co	grid_sean ict(X_test cy_score(y) th': 5, ' 446 th': 5, ' 446 lassifier fier(max_c) s=[('preproduction ('class data train) training peline['preproduction ('class data train) training peline['preproduction ('preproduction ('preproducti	data ceprocessor , sifier', n data ceprocessor , y_train orestClas , min_samp =200) data ceprocessor , y_train orestClas , min_samp =200) data cormed[:, ting data cormed[:, the cormed in the	pred[;,:5]) pred()) es_leaf': 1, 'min_ min_samples_leaf= preprocessor), pred() preliation (X_testine) preta', 'Bhoot', ' 'Jinnat', 'Bhoot', 'Jinnat', 'Bhoot', 'Jinnat', 'Preta', 'Jinnat', 'Preta', 'Jinnat', 'Preta', 'Bhoot', 'Jinnat', 'Preta', 'Bhoot', 'Jinnat', 'Bhoot', 'Jinnat', 'Bhoot', 'Bhoot', 'Bhoot', 'Jinnat', 'J	Samples_split': 2, ' 8, min_samples_spli 8, min_samples_split X_train) X_train) mples_split=5, st) mples_split=5, st) st) Bhoot', 'Preta', 'Preta'	n_estimators': 200} t= 5, n_estimators= 200)
[23]: [24]: [25]: [26]: [27]: [29]: [30]:	# Prin print(y_pred print() Best pa Accurace # crea nclf = #fitti npipel # Fit npipel # Fit npipel # St RandomF # make X_test y_pred # Eval accura print() Accurace #makin y_test y_test array() X_test bo 70	earch.fit () Gr imator: Ra imator: Ra PandomF the best Best para Para Para Para Para Para Para Para	ndomFore orestClas orestClas meters:', arch.pred: ', accurac 'max_der 444444444 graded c. estClassin model line(steps ne to the train, y_ Pipeline or: Column or:	grid_sean ict(X_test cy_score(y) th': 5, ' 446 th': 5, ' 446 lassifier fier(max_c) se[('preproduction ('class data train) training peline['preproduction ('class data train) training peline['preproduction ('preproduction ('preproduc	min_sampl depth= 10, cocessor', sifier', n data ceprocessor , y_train orestClas , min_samp =200) data ceprocessor formed[:, ting data cormed[:, the data cormed	pred(;,:5]) pred()) es_leaf': 1, 'min_ min_samples_leaf= preprocessor), pred()) pred() preprocessor), pred() pred() preta', 'Bhoot', ' 'Jinnat', 'Bhoot', ' 'Jinnat', 'Bhoot', 'Jinnat', 'Preta', 'Jinnat', 'Preta', 'Jinnat', 'Preta', 'Jinnat', 'Preta', 'Bhoot', 'Jinnat', 'Preta', 'Bhoot', 'Jinnat', 'Bhoot', 'Jinnat',	Samples_split': 2, ' 8, min_samples_spli 8, min_samples_split X_train) X_train) mples_split=5, st) mples_split=5, st) st) Bhoot', 'Preta', 'Preta'	n_estimators': 200} t= 5, n_estimators= 200)
[23]: [24]: [25]: [26]: [27]: [29]: [30]:	# Prin print(y_pred print() Best pa Accurat # crea nclf = #fitti npipel # Fit npipel # Fit x_trai nclf.f RandomF # Eval accura print() Accurat # make x_test y_pred # Eval accura print() Accurat # make x_test y_test y_test array() X_test bo 70 827 231 588 39	earch.fit() Gr imator: Ra imator	### AndomFore orestClas inversely meters: arch.predict inversely arch.predict inversely arch.predict inversely arch.predict graded c. estClassin model line(steps me to the train, y_t pipeline or: Column or: Column or: Column inter ipeline or score(X_te ipeline or cifier(max n_e may may preta', 'Jinnat', 'Jinnat', 'Preta', 'Bhoot', 'Jinnat', 'Jinnat', 'Bhoot', 'Jinnat', 'Bhoot', 'Jinnat', 'Bhoot', 'Jinnat', 'Jinnat', 'Bhoot', 'Jinnat', 'Bhoot', 'Jinnat', 'Jinnat', 'Bhoot', 'Jinnat', 'J	grid_sean ict(X_test CY_score(S) cth': 5, ' 446 lassifier fier(max_c) s=[('preprint ('class data train) cat meHotEnce stimators training peline['preprint ('class data train) stimators training peline['preprint ('class training trainin	data ceprocessor , sifier', n data ceprocessor , y_train orestClas , min_samp =200) data cormed[:, 'Jinnat', 'Bhoot', 'Preta', 'Preta', 'Preta', 'Preta', 'Bhoot', 'Preta', 'Bhoot', 'Preta', 'Bhoot', 'Preta', 'Bhoot', 'Preta', 'Bhoot', 'Jinnat', 'Preta', 'Bhoot', 'Preta', 'Jinnat',	pred(;,:5]) pred()) es_leaf': 1, 'min_ min_samples_leaf= preprocessor), pred() pr	Samples_split': 2, ' 8, min_samples_spli 8, min_samples_split X_train) X_train) mples_split=5, st) mples_split=5, st) st) Bhoot', 'Preta', 'Preta'	n_estimators': 200} t= 5, n_estimators= 200)
[23]: [24]: [25]: [26]: [27]: [29]: [30]: [31]:	# Prin print(y_pred print() Best pa Accurat # crea nclf = #fitti npipel # Fit npipel # Fit npipel # Fit x_trai nclf.f WandomF # make x_test y_pred # Eval accura print() Accurat # make x_test y_pred # Load with of cl X_test bo 70 827 231 588 39 897 578 779 25 84	earch.fit () Gr imator: Ra imator: Ra PandomF t the best Best paral grid_se 'Accuracy: arameters: cy: 0.4944 ting an up RandomFore ng the new ine = Pipe the pipeli ine.fit (X_ reprocesso num tandardSca Prediction transform tandardSca Prediction transform tandardSca Prediction transform the pipeli n_transform the pipeli n_transform cy: 0.4777 g the pred pred = nc prediction transform "Preta', 'Jinnat', 'Bhoot', 'Preta', 'Pr	### And One Process or State of the process o	grid_sean ict(X_test cy_score(y) th': 5, ' 446 lassifier fier(max_c) s=[('preprint ('class data train) cat neHotEnco assifier training peline['preprint ('preprint ('preprin	data ceprocessor data ceprocessor y_train orestClas min_samp cormed[:, data ceprocessor ormed[:, data cormed[:,	pred: (; :5]) pred)) es_leaf': 1, 'min_ min_samples_leaf= preprocessor), clf)]) prediction (X_testine) preprocessor), clf)]) preta', 'Bhoot', 'shoot', 'Jinnat', 'Bhoot', 'Jinnat', 'Preta', 'Jinnat', 'Preta', 'Jinnat', 'Preta', 'Bhoot', 'Jinnat', 'Preta', 'Bhoot', 'Jinnat', 'Preta', 'Bhoot', 'Jinnat',	Samples_split': 2, ' 8, min_samples_spli 8, min_samples_split X_train) X_train) mples_split=5, st) mples_split=5, st) st) Bhoot', 'Preta', 'Preta'	n_estimators': 200} t= 5, n_estimators= 200)
[23]: [24]: [25]: [26]: [27]: [29]: [30]: [31]:	# Prin print(y_pred print() Best pa Accurace # crea nclf = #fitti npipel # Fit npipel # Fit npipel # Fit npipel # Fit x_trai nclf.f Random # Make x_test y_pred # Eval accura print() Accurace # makin y_test y_test array() X_test bo 70 827 231 588 39 # Joad with or cl X_test bo 70 827 231 588 39 897 578 779 25 84 180 rows # Load with or cl X_test bo 70 827 231 588 39 897 578 779 25 84 180 rows # Joad with or cl X_test bo 70 827 231 588 799 25 84 180 rows # Joad with or cl X_test bo 70 827 231 588 799 25 84 180 rows # Joad with or cl X_test bo 70 827 231 588 799 25 84 180 rows # Joad with or cl X_test bo 70 827 231 588 799 25 84 180 rows # Joad with or cl X_test bo 70	carch.fit (Companies) Grimator: Ra imator: Ra RandomF the best Best parae grid_se 'Accuracy: arameters: cy: 0.4944 ting an up RandomFor ng the new ine = Pipe the pipeli ine.fit(X_ reprocesso num tandardSca the pipeli n_transform t(X_train) ForestClass prediction transform t(X_train) the pipeli n_transform t(X_train) ForestClass prediction transform tandardSca prediction transform t(X_train) f"Accuracy cy: 0.4777 g the pred pred = nc pred pred = nc pred pred = nc pred the pipeli n_transform t(X_train) the pipeli n_transform t(X_train) the pipeli n_transform t(X_train) the pred pred = nc pred	### AndomFore: ### AndomFore: ### OrestClass ### AndomFore: ### OrestClass ### AndomFore: ### AndomFore:	grid_sean ict(X_test i	cocessor', sifier', n depth= 10, cocessor', sifier', n data ceprocessor cormed[:, cransforme Preta', ' 'Preta', 'Preta', 'Preta', 'Preta', 'Bhoot', 'Preta', 'Bhoot', 'Preta', 'Bhoot', 'Preta', 'Jinnat',	pred: (; :5]) pred)) es_leaf': 1, 'min_ min_samples_leaf= preprocessor), clf)]) prediction (X_testine) preprocessor), clf)]) preta', 'Bhoot', 'shoot', 'Jinnat', 'Bhoot', 'Jinnat', 'Preta', 'Jinnat', 'Preta', 'Jinnat', 'Preta', 'Bhoot', 'Jinnat', 'Preta', 'Bhoot', 'Jinnat', 'Preta', 'Bhoot', 'Jinnat',	samples_split': 2, ' 8, min_samples_spli X_train) mples_split=5, tt) mples_split=5, mples_split=5, tt) mples_split=5, mples_split=5, tt) mples_split=5, mples	n_estimators': 200} t= 5, n_estimators= 200)
[23]: [24]: [25]: [26]: [27]: [28]: [30]: [31]: [32]:	# Prin print(y_pred print() Best pa Accurace # crea nclf = #fitti npipel # Fit npipel # Fit npipel # Fit npipel # Fit x_trai nclf.f Random # Make x_test y_pred # Eval accura print() Accurace # makin y_test y_test array() X_test bo 70 827 231 588 39 # Joad with or cl X_test bo 70 827 231 588 39 897 578 779 25 84 180 rows # Load with or cl X_test bo 70 827 231 588 39 897 578 779 25 84 180 rows # Joad with or cl X_test bo 70 827 231 588 799 25 84 180 rows # Joad with or cl X_test bo 70 827 231 588 799 25 84 180 rows # Joad with or cl X_test bo 70 827 231 588 799 25 84 180 rows # Joad with or cl X_test bo 70 827 231 588 799 25 84 180 rows # Joad with or cl X_test bo 70	earch.fit() Gr imator: Ra P RandomF. L the best Best paral E grid se. 'Accuracy: arameters: cy: 0.4944. ting an up RandomFor. Ing the new ine = Pipe. the pipelitine.fit(X_ reprocessor num tandardSca P Random the pipelitine.fit(X_ reprocessor num tandardSca L Handom the pipelitine.fit(X_train) ForestClass Prediction L Transform	### Andom Fore: ### Andom Fore: ### OrestClas ### OrestClas ### Andom Fore: ### Andom	grid_seasict(X_testicy_score(y) tot(X_testicy_score(y) that is it is i	rmer depth= 10, depth= 20,	premas_) med(:,:5]) pred)) res_leaf': 1, 'min_ min_samples_leaf= preprocessor), clf)]) pred) preta', 'ghoot', 'Jinnat', 'ghoot', 'Jinnat', 'Jinnat', 'Preta', 'Jinnat', 'Preta', 'Jinnat', 'Preta', 'Jinnat', 'Preta', 'Jinnat', 'Preta', 'Bhoot', 'Jinnat', 'Preta', 'Bhoot', 'Jinnat' 'Shoot', 'Jinnat', 'Preta', 'Bhoot', 'Bhoot', 'Jinnat', 'Preta', 'Bhoot', 'Jinnat', 'Preta', 'Jinnat', 'Preta', 'Jinnat', 'Preta', 'Jinnat', 'Preta' 'Jinnat', 'Jinnat', 'Prota', 'Jinnat', 'Prota', 'Jinnat', 'Bhoot', 'Jinnat', 'Bhoot', 'Jinnat',	samples_split': 2, ' X_train) mples_split-5, st) mples_split-5, m	n_estimators': 200} t= 5, n_estimators= 200)
[23]: [24]: [25]: [26]: [27]: [28]: [30]: [31]: [32]:	# Prin print(y_pred print() Best pa Accurace # crea nclf = #fitti npipel # Fit npipel # Fit npipel # Fit npipel # Fit x_trai nclf.f Random # Make x_test y_pred # Eval accura print() Accurace # makin y_test y_test array() X_test bo 70 827 231 588 39 # Joad with or cl X_test bo 70 827 231 588 39 897 578 779 25 84 180 rows # Load with or cl X_test bo 70 827 231 588 39 897 578 779 25 84 180 rows # Joad with or cl X_test bo 70 827 231 588 799 25 84 180 rows # Joad with or cl X_test bo 70 827 231 588 799 25 84 180 rows # Joad with or cl X_test bo 70 827 231 588 799 25 84 180 rows # Joad with or cl X_test bo 70 827 231 588 799 25 84 180 rows # Joad with or cl X_test bo 70	earch.fit () Gr imator: Ra NandomF RandomF Lthe best Best paral grid_se 'Accuracy: arameters: Cy: 0.4944 ting an up RandomForm ng the new ine = Pipe the pipelit ine.fit (X_ reprocesso num tandardSca Lunch Random the pipelit n_transform tandardSca Lunch Random the pipelit n_transform tandardSca Lunch Preta', Preta	andomFore: andomFore: orestClas orestClas hyperpara meters:', arch.pred: ', accurac {'max_der 44444444 graded c. estClassis model line (steps ne to the train, y_ Pipeline or: Column or: Column or: Column forestCla ine on the med = npipe edict(X_te ifier(max_ n_e classis ne to the train, y_ ifier(max, n_e ifier(max, n_e classis ne to the train, y_ ifier(max, n_e	grid_sean ict(X_test icy_score(s) ict(X_test icy_score(s) ict(X_test icy_score(s) ict(X_test icy_score(s) ict(X_test icy_score(s) ict(X_test icy_score(s) ict(X_test) ict(X_te	data reprocessor, sifier', n data reprocessor, y_train orestClas min_sampl data reprocessor, y_train orestClas min_samp adata reprocessor, y_train orestClas min_samp adata reprocessor reprocesso	prediction in the state of the	samples_split': 2, ' samples_split': 5, "Jinnat', 'Bhoot', 'Preta', 'Jinnat', 'Bhoot', 'Bhoot', 'Bhoot', 'Bhoot', 'Preta', 'Preta', 'Preta', 'Preta', 'Preta', 'Jinnat', 'Bhoot',	n_estimators': 200} t= 5, n_estimators= 200)
[23]: [24]: [25]: [26]: [27]: [28]: [30]: [31]: [32]:	# Prin print(y_pred print() Best pa Accurace # crea nclf = #fitti npipel # Fit npipel # Fit npipel # Fit npipel # Fit x_trai nclf.f Random # Make x_test y_pred # Eval accura print() Accurace # makin y_test y_test array() X_test bo 70 827 231 588 39 # Joad with or cl X_test bo 70 827 231 588 39 897 578 779 25 84 180 rows # Load with or cl X_test bo 70 827 231 588 39 897 578 779 25 84 180 rows # Joad with or cl X_test bo 70 827 231 588 799 25 84 180 rows # Joad with or cl X_test bo 70 827 231 588 799 25 84 180 rows # Joad with or cl X_test bo 70 827 231 588 799 25 84 180 rows # Joad with or cl X_test bo 70 827 231 588 799 25 84 180 rows # Joad with or cl X_test bo 70	earch.fit() Gr imator: Ra imator: Ra PandomF RandomF t the best Best paral grid_se. 'Accuracy: arameters: cy: 0.4944 ting an up RandomForm and the pipeli ine.fit(X_ reprocess num tandardSca Landom the pipeli ine.fit(X_ reprocess num the pipeli num the pipeli num the pipeli num	### AndomFore: orestClas ### AndomFore: orestClas ### AndomFore: orestClas ### AndomFore: orestClas ### Andom Andom ### Andom #### Andom #### Andom #### Andom ### Andom #### Andom ##### Andom #### Andom ####	grid_sean ict(X_test icy_score(y) ict(X_test icy_score(y) ict(x_test icy_score(y) ict(x_test icy_score(y) ict(x_test) ict(x_te	data cecessor, sifier', n depth= 10, cocessor', sifier', n data ceptocessor, sifier', n cocessor', sifier', n data ceptocessor, y_train orestClas , min_samp =200) data ceptocessor, cormed[:, 'Jinnat', 'Preta', 'Preta', 'Preta', 'Preta', 'Bhoot', 'Preta', 'Preta', 'Jinnat', 'Preta', 'Preta', 'Preta', 'Bhoot', 'Bhoot', 'Preta', 'Pret	predication in the control of the co	samples_split': 2, ' samples_s	n_estimators': 200} t= 5, n_estimators= 200)
[23]: [24]: [25]: [26]: [27]: [28]: [32]: [33]:	# Prin print(Y_pred print() Best pa Accurat # crea nclf = #fitti npipel # Fit npipel # Fit npipel # Fit x_trai nclf.f Randomf # make X_test y_pred # Eval accurat print() Accurat # makin y_test y_test array() # Save with or cl X_test bo 70 827 231 588 39 897 578 779 25 84 180 rows y_pred array()	earch.fit() frimator: Ra imator: Ra imator: Ra PandomF RandomF Lathe best Best parse Accuracy: arameters: cy: 0.4944 fing an up RandomFor Ing the new ine = Pipe LandardSca LandardSca	### A CONTRICT OF THE PROPERTY	grid_sean ict(X_test icy_score(s) ict(X_test ict(X_test) ict(X_tes	rest, Y_ min_sampl depth= 10, cocessor', sifier', n cocessor', sifier', n data ceprocesso , y_train orestClas processor' cormed[:, min_samp coder complete and a complete complete and a c	predication in the control of the co	Samples_split': 2, ' samples_split': 3, ' samples_split': 5, ' samples_split': 2, ' samples_split': 5, ' samples_split': 5, ' samples_split': 2, ' samples_s	n_estimators': 200} t= 5, n_estimators= 200)
[23]: [24]: [25]: [26]: [27]: [28]: [30]: [31]: [32]:	# Prin print(y_pred print() Best part Accurate # crea nclf = #fitti npipel # Fit npipel # Fit npipel # Fit npipel Accurate # fit x_trai nclf.f Randomf # make x_test y_pred # Eval accurate print() Accurate # makin y_test y_test array() # Save with or pi # Load with or cl X_test bo 70 827 231 588 39 897 578 79 25 84 180 rows y_pred array() # Make y_pred array()	earch.fit () fimator: Ra imator: Ra RandomF. RandomF. L the best Best parase. Accuracy: arametase. Accuracy: aramedase. Accuracy: aramedase. In an up RandomFore In an up RandomFore In an up RandomFore In an up In an	### A CONTRICT OF THE PROPERTY	grid_sean ict(X_test ict(X_te	rmer depth= 10, depth= 10,	premss_) med(:,:5]) pred(); med(:,:5]) pred(); pred(); es_leaf': l, 'min_ min_samples_leaf= preprocessor), clf()]; preta', 'shoot', 'Jinnat', 'shoot', 'Jinnat', 'shoot', 'Preta', 'Jinnat', 'Preta', 'Jinnat', 'Preta', 'Jinnat', 'Preta', 'Jinnat', 'Preta', 'Jinnat', 'Preta', 'Bhoot', 'Jinnat', 'Preta', 'Jinnat', 'Preta', 'Jinnat', 'Preta', 'Jinnat', 'Preta', 'Bhoot', 'Bhoot', 'Jinnat', 'Preta', 'Bhoot', 'Bhoot', 'Jinnat', 'Preta', 'Bhoot', 'Bhoot', 'Preta', 'Bhoot', 'Preta', 'Bhoot', 'Jinnat', 'Preta', 'Bhoot', 'Bhoot', 'Preta', 'Bhoot', '	samples_split': 2, ' samples_split': 3, ' samples_split': 5, ' samples_split': 2, ' samples_split': 5, ' samples_split': 2, ' samples_split': 2, ' samples_split': 2, ' samples_split': 5, ' samples_split': 2, ' samples_s	n_estimators': 200} t= 5, n_estimators= 200)
[23]: [24]: [25]: [26]: [27]: [28]: [32]: [33]: [33]:	# Prin print(y_pred print() Best part Accurate # crea nclf = #fitti npipel # Fit npipel # Fit npipel # Fit npipel Accurate # fit x_trai nclf.f Randomf # make x_test y_pred # Eval accurate print() Accurate # makin y_test y_test array() # Save with or pi # Load with or cl X_test bo 70 827 231 588 39 897 578 79 25 84 180 rows y_pred array() # Make y_pred array()	earch.fit() imator: Ra imator: Ra RandomF. RandomF. Lthe best Best parale. 'Accuracy: arametars: cy: 0.4944 ting an up RandomForm frandomForm arametars: cy: 0.4944 ting an up RandomForm arametars: cy: 0.4944 ting an up RandomForm frandomForm arametars: cy: 0.4944 ting an up RandomForm arametars: cy: 0.4777 granding arametars: cy: 0.4777 granding arametars	## A CONTRICT OF THE PROPERTY	grid_sean ict(X_test icy_score() ict(X_test icy_score() ict(X_test icy_score() ict(X_test icy_score() ict(X_test)	rest, Y_ min_sampl depth=10, depth=20, defined: defin	premas_) med(c,:5]) med(c,:5]) med(c,:5]) pred(c); med(c,:5]) red(c); min_samples_leaf= preprocessor), clf)]) preta', 'Bhoot', 'Jinnat', 'Bhoot', 'Jinnat', 'Bhoot', 'Preta', 'Jinnat', 'Preta', 'Jinnat', 'Preta', 'Jinnat', 'Preta', 'Jinnat', 'Preta', 'Jinnat', 'Preta', 'Bhoot', 'Jinnat', 'Preta', 'Bhoot', 'Preta', 'Bhoot', 'Preta', 'Bhoot', 'Preta', 'Bhoot', 'Jinnat', 'Preta', 'Bhoot', 'Jinnat', 'Bhoot', 'Jinnat', 'Preta', 'Bhoot', 'Bhoot', 'Jinnat', 'Preta',	Samples_split': 2, ' samples_split': 3, ' samples_split': 2, ' samples_split': 2, ' samples_split': 2, ' samples_split': 3, ' samples_split': 2, ' samples_s	n_estimators': 200} t= 5, n_estimators= 200)
[23]: [24]: [25]: [26]: [27]: [28]: [32]: [33]: [33]:	# Prin print(y_pred print() Best part Accurate # crea nclf = #fitti npipel # Fit npipel # Fit npipel # Fit npipel Accurate # fit x_trai nclf.f Randomf # make x_test y_pred # Eval accurate print() Accurate # makin y_test y_test array() # Save with or pi # Load with or cl X_test bo 70 827 231 588 39 897 578 79 25 84 180 rows y_pred array() # Make y_pred array()	earch.fit() imator: Ra imator: Ra PandomF RandomF lebest paral grid_se 'Accuracy: arameters: cy: 0.4944 ting an up RandomFor ng the new ine = Pipe the pipeli ine.fit(X, reprocess num tandardSca Pandom the pipeli ine.fit(X, reprocess num tandardSca lebest paral che pipeli ine.fit(X, reprocess num tandardSca lebest paral prediction transform the pipeli ine.fit(X, reprocess num tandardSca lebest paral prediction tandardSca lebest paral prediction tandardSca lebest paral prediction the pipeli ine.fit(X, reprocess num tandardSca lebest paral prediction tandardSca lebest paral prediction tandardSca lebest paral prediction the pipeli ine.fit(X, reprocess lebest paral lebest para	## A CONTROLL OF THE PROPERTY	grid_sease ict (X_test ict (X_test ict) grid_sease ict ict) grid_sease ict ict) grid_sease ict ict ict) grid_sease ict	rest, y min_sampl depth= 10, dept	prediction of the color of the	samples_split': 2, ' samples_split samples_split	n_estimators': 200} t= 5, n_estimators= 200)
[23]: [24]: [25]: [26]: [27]: [28]: [32]: [33]: [33]:	# Prin print(y_pred print() Best part Accurate # crea nclf = #fitti npipel # Fit npipel # Fit npipel # Fit npipel Accurate # fit x_trai nclf.f Randomf # make x_test y_pred # Eval accurate print() Accurate # makin y_test y_test array() # Save with or pi # Load with or cl X_test bo 70 827 231 588 39 897 578 79 25 84 180 rows y_pred array() # Make y_pred array()	earch.fit () imator: Ra imator: Ra PandomF PandomF LandomF L	## A Company of the control of the c	grid_sea ict (X_test ict (X_test ict (X_test ict (X_test) ict (X_test) ict (X_test) ict (Y_class) data train (I) ict (Y_class) data ict (Y_class) ict (Y_cla	rest, Y_ min_sampl depth= 10, cocessor', sifier', n depth= 10, depth= 10, cocessor', sifier', n depth= 10, cocessor', sifier', n depth= 10, depth= 10, cocessor', sifier', n depth= 10, cocessor', sifier', 'Preta',	preprocessor), cif) preprocessor), cif)) preta',	samples_split': 2, ' Bhoot', 'Jinat', 'Jinnat', 'Preta', 'Preta', 'Jinnat', 'Jinnat', 'Preta', 'Bhoot', 'Bhoot', 'Bhoot', 'Bhoot', 'Bhoot', 'Bhoot', 'Preta', 'P	n_estimators': 200} t= 5, n_estimators= 200)
[23]: [24]: [25]: [26]: [27]: [30]: [31]: [32]: [33]:	# Prin print(y_pred print() Best part Accurate # crea nclf = #fitti npipel # Fit npipel # Fit npipel # Fit npipel Accurate # fit x_trai nclf.f Randomf # make x_test y_pred # Eval accurate print() Accurate # makin y_test y_test array() # Save with or pi # Load with or cl X_test bo 70 827 231 588 39 897 578 79 25 84 180 rows y_pred array() # Make y_pred array()	earch.fit () imator: Ra imator: Ra imator: Ra PandomF RandomF Lacameters: Cy: 0.4944 Ling an up RandomFore RandomFore	## A CONTON TO A C	grid_sea ict (X_test ict (X_test ict (X_test ict (X_test) ict (X_test) ict (X_test) ict (X_test) ict (Y_nered) ict (Y	cocessor', inin_sampl depth= 10, cocessor', ifier', n cocessor', ifier', n depth= 10, cocessor', ifier', n cocessor'	warams_) meded; '5]) meded; '5]) meded; '5]) meded; '1, 'min_ min_samples_leaf= preprocessor), clf)) mediamatiles_leaf= preprocessor), clf)) preta', 'Bhoot', 'shoot', 'shoot', 'bhoot', 'Jinnat', 'Bhoot', 'Bhoot', 'Jinnat', 'Preta', 'Bhoot', 'Jinnat', 'Preta', 'Bhoot', 'Jinnat', 'Preta', 'Bhoot', 'Preta', 'Bhoot', 'Preta', 'Bhoot', 'Preta', 'Bhoot', 'Preta', 'Bhoot', 'Preta', 'Bhoot', 'Jinnat', 'Preta', 'Bhoot', 'Preta', 'Bhoot', 'Shoot', 'Preta', 'Bhoot', 'Jinnat', 'Preta', 'Bhoot', 'Shoot', 'Jinnat', 'Bhoot', 'Jinnat', 'Preta', 'Bhoot', 'Bhoot', 'Preta', 'Bhoot', 'Jinnat', 'Preta', 'Bhoot', 'Bhoot', 'Jinnat', 'Preta', 'Bhoot', 'Bhoot', 'Jinnat', 'Preta', 'Bhoot', 'Bhoot', 'Jinnat', 'Preta', 'Bhoot', 'Bhoot', 'Jinnat', 'Phoot', 'Jinnat', 'Preta', 'Bhoot', 'Bhoot', 'Jinnat', 'Preta', 'Bhoot', 'Bhoot', 'Jinnat', 'Preta', 'Bhoot', 'Bhoot', 'Jinnat', 'Preta', 'Bhoot', 'Preta', 'Bhoot	Samples_split': 2, ' **S, min_samples_spli **S, min_samples_split-5, **Inat', 'Innat', 'Inn	n_estimators': 200} t= 5, n_estimators= 200)