In [3]:											<pre>import pandas as pd import numpy as np import lightgbm</pre>													
In [4]: In [5]:																								
In [6]:	datase			data/credi)																		
In [7]:			_	ction impo : y_test = t:	_	_		("Class",a	axis=1),da	ıtaset['Cla	ass'], test	_size=0.3,ra	ndom_state	e=1)										
In [8]:	model	= lightg	bm。LGBMCl	assifier(r	andom_stat	e=1, n_est	timators=1	1000, max_	_depth=5,	learning_:	rate=0.01,	num_leaves=2	**5, num_	threads=7,	boosting_	_type='gbdt	.')							
In [9]: Out[9]:	[Light(GBM] [War		in) n_threads i rate=0.01,					urrent val	lue: num_t	hreads=7													
In [10]:	from s		num_leave	es=32, num_	threads=7,	random_s	tate=1)																	
<pre>In [11]: Out[11]:</pre>				el.predict	(X_test))																			
In [12]: Out[12]:				model.pred	ict(X_test))																		
In [13]:		ore(y_tes 284552845		predict(X_	test))																			
Out[13]: In [14]:	import		329																					
In [15]:	shap_v	alues =	explainer	plainer(mo	es(X_test)	r shap va	lues outp	ut has cha	anged to a	a list of	ndarray													
In [16]:	shap.s	summary_p	lot(shap_	values[1],	X_test)		High	1																
	V1 Amount V11 V6 V3 V12 V7 V24 V26 V28 V10 V23 V18 V16 V9 V17	Namount V11 V6 V3 V12 V7 V24 V26 V28 V10 V23 V10 V23 V10 V24 V26 V28 V10 V27 V27 V29 V17 V22 Shape_df = pd.DataFrame(shap_values[1]) shape_df = pd.DataFrame(shap_values[1]) shape_df :columns=['SNAP_' + x for x in X_test.columns]																						
In [17]:						st.columns	5]																	
In [18]: Out[18]:	shaps_		SHAP_V2	SHAP_V3	SHAP_V4	SHAP_V5	SHAP_V6	SHAP_V7	SHAP_V8	SHAP_V9	SHAP_V10	SHAP_V20	SHAP_V21	SHAP_V22	SHAP_V23	SHAP_V24	SHAP_V25	SHAP_V:						
	0 1 2	0.267923 0.222537 0.313281	0.026187 -0.031799 0.043180	-0.360517	0.096680 0.035063 0.132564		-0.172885 0.101742 0.016414	-0.009762 -0.047244 -0.140499	0.315159 -0.671800 0.332134			0.011126 0.037300 0.052215		0.010222 -0.018498 -0.021751	0.036375 0.070905 0.077856		0.046767 -0.011108 0.032576	0.1906 -0.1820 -0.1418						
	3 4 		0.034531	0.120766		-0.037736 -0.099159 	0.099604 -0.262002 	-0.127117 -0.010835 	0.269727 0.366207 	-0.001978 0.046293 	0.020701 0.068214 	0.032644 0.060793 	-0.113817 -0.035104 	-0.067512 -0.028320 	-0.088461 0.068364 	0.047333 0.076332 	0.040821 -0.110480 	-0.1487 -0.0108						
	85438 85439 85440	0.158552 -0.840244 0.246434	0.021435	0.020947	1.314426 -0.445624 0.861696	-0.093363 0.071210 0.115187	0.138800 -0.254195 0.147481	-0.037225 -0.028524 -0.161775			0.023987	0.040831 0.038124 0.069358	-0.132675 -0.022372 -0.019974	0.019527 -0.023075 -0.005285	-0.107559 0.022921 0.164845	0.158676 -0.161585 0.602639	-0.109952 0.019072 0.006854	-0.08898 0.05294 0.04248						
	85442	-1.384565 0.274288 ows × 29 c	0.030377			-0.081379 -0.089329	0.115247 0.137106	0.155819	0.329918 0.355571	-0.149688 0.026731	0.000070	0.043882 0.064782	0.017633	0.074342 0.044505	0.026348 0.047854	0.108705 -0.035079	0.035058 0.040009	-0.0988 -0.0948						
In [19]:	positi		_df = sha	ps_df.appl	y(lambda x	: x>0).ast	type(int)																	
Out[19]:	0	SHAP_V1 1	SHAP_V2 1 0	SHAP_V3 S 1	HAP_V4 SH 1	AP_V5 SH 1 0	0 1	0 0	P_V8 SHAF 1 0	P_V9 SHAP_ 1	_ v10 SH . 0	AP_V20 SHAP 0 0	_V21 SHAP _ 0 0	_V22 SHAP _ 1 0	_ V23 SHAP _ 1	_ V24 SHAP _ 0	v25 SHAP 1	_ V26 SH 1						
	3	1 1	1	1 0	1 0	1	1 1 1 0	0 0	1 1	1 0	1 1	0 0 0	1 0 0	0 0 0	1 0	1 1	1 1 0	0						
	85438 85439	1 1	1	1 1	1 1 0	0 0	 1 0	0 0	1 1 0	0	1 1 1	0 0	0 0	 1 0	0	1 1	0 0	0 0						
	85440 85441 85442	1 0	1 1	0 1	1 1 1	1 0 0	1 1 1	0 1 0	1 1	1 0	0 1 1	0 0 0	0 1 0	0 1	1 1	1 1 0	1 1 1	1 0 0						
		ows × 29 c	columns	'	'	O	'	Ü	'	'	1	0	O .	'	'	0	'	Ü						
In [20]: In [21]:				rowth(posi			support=0.	.01, use_c	colnames =T	!rue)														
In [22]:	freque	ent_items				ite	msets																	
Out[22]:		suppoi0.884660.850480.78036	0			(SH	AP_V2) AP_V1) P_V23)																	
<pre>In [23]: In [35]: Out[35]:</pre>	frequent fre	0.01052 0 rows × 2 0 r	columns ets ets['leng ets[frequ rt 8 27 6 39 33 07 (SHAP_)	<pre>/20, SHAP_V16 /20, SHAP_V16 /20, SHAP_V16 /20, SHAP_V16 /23, SHAP_V13 /26, SHAP_V2</pre>	guent_item ts['length (SHAP_V23, (SHAP_V9, (SHAP_V3, (SHAP_V6, (SHAP_V10,	Sets['iter '] > 2].so ite SHAP_V2, SH SHAP_V2, SH SHAP_V2, SH SHAP_V2, SH SHAP_V2, SH	AP_V nsets'].ar ort_values emsets len HAP_V1) HAP_V1) HAP_V1) HAP_V1) HAP_V1.	s("support																
	312639 63008	7 0.01000 3 0.01000)7 (SHAP_)7 (SHAP_	_V10, SHAP_V2 V10, SHAP_V2	3, SHAP_V11, 3, SHAP_V2, S	SHAP_V4, SH SHAP_V14, SH	IAP_V	8																
	343194 6025674	7 0.01000 I rows × 3		'2, SHAP_V3, S	SHAP_AMOUNT	, 5ПАР_V9, 5	пар	9																
In [25]: In [41]:				s['itemset						.',ascendi	ng =False). h	nead(50)												
	226 648 226 238 248 248 248 248 258 266 266 266 267 2759468 266 266 266 266 2759468 266 266 2759468 266 266 2759468 266 2759468 266 2759468 266 2759468 266 2759468	8 0.36958 0 0.36879 6 0.36695 7 0.36690 1 0.36563 3 0.36556 4 0.36446 7 0.36389 2 0.36336 4 0.36240 5 0.35926 7 0.35468 9 0.35294 9 0.35105 3 0.34879 0.34858 9 0.34858 1 0.34559 1 0.34503	68	(SHAP_V2 (SHAP_V2 (SHAP_V2 (SHAP_V2 (SHAP_V2 (SHAP_V2 (SHAP_V2 (SHAP_V2 (SHAP_V1 (SHAP_V1 (SHAP_V3 (SHAP_V4 (SH	SHAP_V4, SHA	HAP_V3) HAP_V1) HAP_V2) HAP_V1) HAP_V2) HAP_V3) HAP_V3) HAP_V1) HAP_V1) HAP_V2) HAP_V1) HAP_V3) HAP_V3) HAP_V3) HAP_V3) HAP_V3) HAP_V3) HAP_V3) HAP_V1) HAP_V1) HAP_V1) HAP_V1) HAP_V1) HAP_V1) HAP_V2) HAP_V1) HAP_V1) HAP_V1) HAP_V1) HAP_V1) HAP_V1) HAP_V1) HAP_V2) HAP_V1) HAP_V2) HAP_V1) HAP_V2) HAP_V1) HAP_V2) HAP_V1) HAP_V2) HAP_V25) HAP_V25) HAP_V25) HAP_V25) HAP_V25) HAP_V25) HAP_V25) HAP_V25)	3 3 3 3 3 3 3 3 3 3 3 3 3 4 3 4 3 3 3 3																	
In [51]:	296311	0.342770.342570.34257	79		SHAP_V3, SF	IAP_V18)	4 3 2 V4']==1)) & /	cive ~'	dfila	V14'1-	•inder												
In [51]: In [52]:		lex = pos		ps_df[(pos	ıtive_shap	s_df[' <mark>SHA</mark> I	.V4']==1)) & (posit	cive_shaps	s_df['SHAP	_V14']==1)]	•index												
Out[52]: In [54]:		300007022 un(y_test		4_index])																				
Out[54]: In [55]:		568079972 525680799		0015800007	0222253431	/ 0.00159	3000070223	225343 <i>#de</i> 1	lta percer	ntuale														
Out[55]: In [49]:	2.32709	921904324	178								0.7	/	,											
In [49]: In [56]:	#SI Pu			.001580000							el gruppo /	/ num gruppo)	/ evento	medio										
Out[56]: In [57]:	<pre>import files_ for f_</pre>	o in fil	= [("mode es_object		l),("shap_	values.pk	',shap_val	lues),("fr	requent_it	emsets.pk	",frequent_	_itemsets)]												
In []:		pickle	.dump(f_o	[1], f)																				