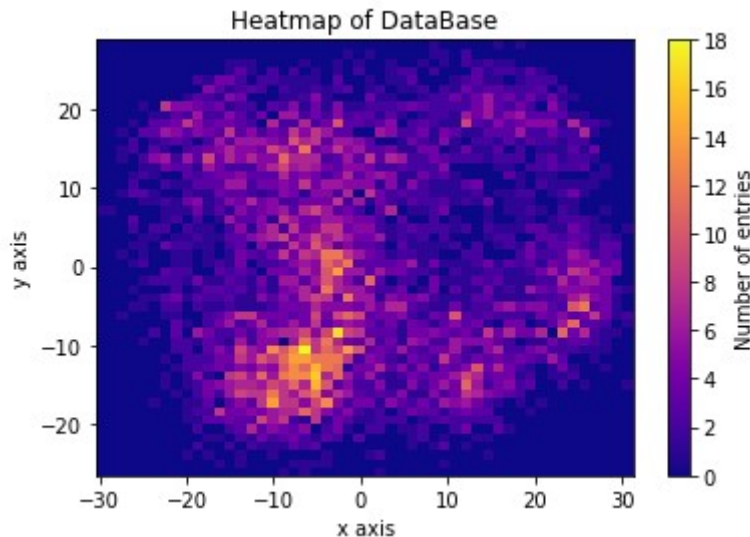


Python 3.7.7 (default, May 6 2020, 11:45:54) [MSC v.1916 64 bit (AMD64)]  
Type "copyright", "credits" or "license" for more information.

IPython 7.13.0 -- An enhanced Interactive Python.

```
In [1]: 'C:/Users/alvar/OneDrive/Escritorio/2do cuatrimestre/AA/práctica 3/  
BDoptdigits.py' = 'C:/Users/alvar/OneDrive/Escritorio/2do cuatrimestre/AA/práctica  
3'
```



--- Pulsar tecla para continuar ---

CON PREPROCESADO:

Fitting 5 folds for each of 14 candidates, totalling 70 fits  
[Parallel(n\_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.  
[Parallel(n\_jobs=-1)]: Done 34 tasks | elapsed: 52.8s  
[Parallel(n\_jobs=-1)]: Done 70 out of 70 | elapsed: 1.7min finished

He estudiado estos modelos a los que asigno un índice:

	params	param_estimator_C
0	{'estimator': LogisticRegression(C=1.0, class_...	0.0001
1	{'estimator': LogisticRegression(C=1.0, class_...	0.0001
2	{'estimator': LogisticRegression(C=1.0, class_...	0.0001
3	{'estimator': LogisticRegression(C=1.0, class_...	0.0001
4	{'estimator': LogisticRegression(C=1.0, class_...	1
5	{'estimator': LogisticRegression(C=1.0, class_...	1
6	{'estimator': LogisticRegression(C=1.0, class_...	1
7	{'estimator': LogisticRegression(C=1.0, class_...	1
8	{'estimator': LogisticRegression(C=1.0, class_...	10000
9	{'estimator': LogisticRegression(C=1.0, class_...	10000
10	{'estimator': LogisticRegression(C=1.0, class_...	10000
11	{'estimator': LogisticRegression(C=1.0, class_...	10000
12	{'estimator': Perceptron(alpha=0.0001, class_w...	NaN
13	{'estimator': Perceptron(alpha=0.0001, class_w...	NaN

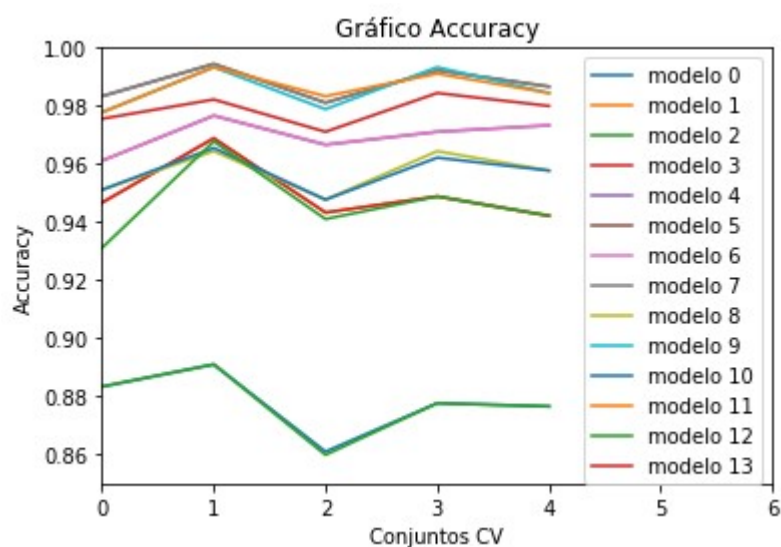
--- Pulsar tecla para continuar ---

Resultados:

	rank_test_score	mean_fit_time	mean_test_score
0	13	0.169153	0.877890

1	10	6.119047	0.949956
2	14	0.335107	0.877668
3	10	18.103411	0.949956
4	6	1.332633	0.969753
5	1	41.407908	0.987545
6	6	1.544068	0.969753
7	1	51.066086	0.987545
8	8	4.051566	0.957074
9	4	6.358805	0.985544
10	9	3.659009	0.956852
11	3	15.864185	0.985989
12	12	0.595410	0.946178
13	5	5.758998	0.978648

--- Pulsar tecla para continuar ---



--- Pulsar tecla para continuar ---

Mejor modelo:  
 {'estimator': LogisticRegression(C=1.0, class\_weight=None, dual=False, fit\_intercept=True, intercept\_scaling=1, l1\_ratio=None, max\_iter=500, multi\_class='auto', n\_jobs=None, penalty='l2', random\_state=None, solver='lbfgs', tol=0.0001, verbose=0, warm\_start=False), 'estimator\_\_C': 1.0, 'estimator\_\_solver': 'lbfgs', 'poly\_degree': 2}  
 Precisión en training: 100.0  
 Precisión en test: 98.84341637010677

--- Pulsar tecla para continuar ---

CON PREPROCESADO Y REGULARIZACION:

He estudiado estos modelos a los que asigno un índice:

	params	param_estimator__C
0	{'estimator': LogisticRegression(C=1.0, class_...	0.0001
1	{'estimator': LogisticRegression(C=1.0, class_...	0.0001
2	{'estimator': LogisticRegression(C=1.0, class_...	0.0001

```

3 {'estimator': LogisticRegression(C=1.0, class_...      0.0001
4 {'estimator': LogisticRegression(C=1.0, class_...      1
5 {'estimator': LogisticRegression(C=1.0, class_...      1
6 {'estimator': LogisticRegression(C=1.0, class_...      1
7 {'estimator': LogisticRegression(C=1.0, class_...      1
8 {'estimator': LogisticRegression(C=1.0, class_...    10000
9 {'estimator': LogisticRegression(C=1.0, class_...    10000
10 {'estimator': LogisticRegression(C=1.0, class_...    10000
11 {'estimator': LogisticRegression(C=1.0, class_...    10000
12 {'estimator': Perceptron(alpha=0.0001, class_w...      NaN
13 {'estimator': Perceptron(alpha=0.0001, class_w...      NaN

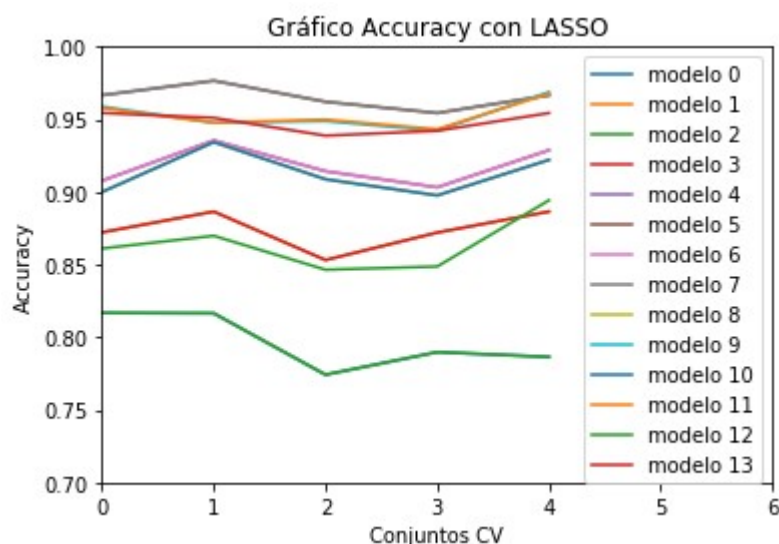
```

--- Pulsar tecla para continuar ---

Resultados:

	rank_test_score	mean_fit_time	mean_test_score
0	13	0.161772	0.796704
1	10	0.625724	0.874111
2	13	0.289624	0.796704
3	10	1.882967	0.874111
4	6	0.767945	0.917929
5	1	8.326545	0.965302
6	6	0.871471	0.917929
7	1	8.695554	0.965302
8	8	2.212480	0.912592
9	4	7.555405	0.953291
10	8	2.214275	0.912592
11	3	7.615837	0.953291
12	12	0.207845	0.864102
13	5	1.867201	0.948175

--- Pulsar tecla para continuar ---



--- Pulsar tecla para continuar ---

Mejor modelo:

```

{'estimator': LogisticRegression(C=1.0, class_weight=None, dual=False,
fit_intercept=True,
                                intercept_scaling=1, l1_ratio=None, max_iter=500,

```

```
multi_class='auto', n_jobs=None, penalty='l2',  
random_state=None, solver='lbfgs', tol=0.0001, verbose=0,  
warm_start=False), 'estimator__C': 1.0, 'estimator__solver': 'lbfgs',  
'poly__degree': 2}  
Precisión en training: 99.86654804270462  
Precisión en test: 96.70818505338077
```

--- Pulsar tecla para continuar ---

In [2]: