

## Optimizando parámetros con grid\_search

Los hiperparámetros son parámetros que no se aprenden directamente dentro de los estimadores. En scikit-learn se pasan como argumentos al constructor de la clase del estimador. Entre estos parámetros podemos destacar **C**, **kernel** y **gamma** para **Support Vector Classifier(SVC)**

Cualquier parámetro proporcionado al construir un estimador puede optimizarse de esta manera. Específicamente, para encontrar los nombres y los valores actuales de todos los parámetros para un estimador determinado se puede utilizar:

`estimator.get_params()`

Una búsqueda tiene los siguientes elementos:

- Un estimador (regresor o clasificador como `sklearn.svm.SVC()`)
- Un espacio de parámetros
- Un método para buscar o muestrear candidatos
- Un esquema de validación cruzada
- Una función de puntuación.

**GridSearchCV** lo que hace internamente es ir probando con los parámetros especificados con el parámetro `param_grid`. Por ejemplo, la siguiente `param_grid`:

```
param_grid = [  
    {'C': [1, 10, 100, 1000], 'kernel': ['linear']},  
    {'C': [1, 10, 100, 1000], 'gamma': [0.001, 0.0001], 'kernel': ['rbf']},  
]
```

Especifica que se deben explorar dos funciones: uno con un kernel lineal y los valores C en [1, 10, 100, 1000], y la segunda con un kernel RBF, y la combinación de valores C que van en [1, 10, 100, 1000] y valores gamma en [0,001, 0,0001].

```
Cs = [0.001, 0.01, 0.1, 1, 10, 100]  
gammas = [0.001, 0.01, 0.1, 1, 10, 100]  
  
from sklearn.grid_search import GridSearchCV  
  
param_grid = {'C': Cs, 'gamma': gammas}  
grid_search = GridSearchCV(SVC(), param_grid, cv=5, verbose=2)
```

Para obtener las puntuaciones y los mejores valores para éstos parámetros podemos hacerlo mediante los métodos `score` y la propiedad `_best_params_`