



## Grid Search in sklearn

Grid Search in sklearn is very simple. We'll illustrate it with an example. Let's say we'd like to train a support vector machine, and we'd like to decide between the following parameters:

- kernel: `poly` or `rbf`.
- C: 0.1, 1, or 10.

*(Note: These parameters can be used as a black box now, but we'll see them in detail in the **Supervised Learning Section** of the nanodegree.)*

The steps are the following:

### 1. Import GridSearchCV

```
from sklearn.model_selection import GridSearchCV
```

### 2. Select the parameters:

Here we pick what are the parameters we want to choose from, and form a dictionary. In this dictionary, the keys will be the names of the parameters, and the values will be the lists of possible values for each parameter.

```
parameters = {'kernel':['poly', 'rbf'], 'C':[0.1, 1, 10]}
```

### 3. Create a scorer.

We need to decide what metric we'll use to score each of the candidate models. In here, we'll use F1 Score.

```
from sklearn.metrics import make_scorer
from sklearn.metrics import f1_score
scorer = make_scorer(f1_score)
```



```
# Create the object.  
grid_obj = GridSearchCV(clf, parameters, scoring=scorer)  
# Fit the data  
grid_fit = grid_obj.fit(X, y)
```

## 5. Get the best estimator.

```
best_clf = grid_fit.best_estimator_
```

Now you can use this estimator `best_clf` to make the predictions.

In the next page, you'll find a lab where you can use GridSearchCV to optimize a decision tree model.

[NEXT](#)