In [1]:
```
## How to tune Hyper-parameters using Grid Search in Python
def Snippet_142():
    print()
    print(format('How to tune Hyper-parameters using Grid Search in Python','*^82'))
    import warnings
    warnings.filterwarnings("ignore")

    # load libraries
    import numpy as np
    from sklearn import linear_model, datasets
    from sklearn.model_selection import GridSearchCV

    # Load data
    iris = datasets.load_iris()
    X = iris.data
    y = iris.target

    # Create logistic regression
    logistic = linear_model.LogisticRegression()

    # Create Hyperparameter Search Space
    # Create regularization penalty space
    penalty = ['l1', 'l2']

    # Create regularization hyperparameter space
    C = np.logspace(0, 4, 10)

    # Create hyperparameter options
    hyperparameters = dict(C=C, penalty=penalty)

    # Create grid search using 5-fold cross validation
    clf = GridSearchCV(logistic, hyperparameters, cv=5, verbose=0)

    # Fit grid search
    best_model = clf.fit(X, y)

    # View best hyperparameters
    print('Best Penalty:', best_model.best_estimator_.get_params()['penalty'])
    print('Best C:', best_model.best_estimator_.get_params()['C'])
Snippet_142()
```

```
*************How to tune Hyper-parameters using Grid Search in Python*************
Best Penalty: l1
Best C: 7.742636826811269
```

In [ ]:
```
```