Pass Task 9.1P: Grid Search with Cross-Validation

In [2]:
```
# Import necessary libraries
from sklearn.model_selection import GridSearchCV
from sklearn.svm import SVC
from sklearn.datasets import load_digits
from sklearn.model_selection import train_test_split

# Load the dataset
digits = load_digits()
X = digits.data
y = digits.target

# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)

# Define the parameter grid
param_grid = {'C': [0.001, 0.01, 0.1, 1, 10, 100],
              'gamma': [0.001, 0.01, 0.1, 1, 10, 100]}

# Initialize the model
svc = SVC()

# Perform grid search with cross-validation
grid_search = GridSearchCV(svc, param_grid, cv=5, return_train_score=True)
grid_search.fit(X_train, y_train)

# Print results
print(f"Test Score: {grid_search.score(X_test, y_test)}")
print(f"Best Parameters: {grid_search.best_params_}")
print(f"Best Cross-Validation Score: {grid_search.best_score_}")
print(f"Best Estimator: {grid_search.best_estimator_}")
```

```
Test Score: 0.9916666666666667
Best Parameters: {'C': 1, 'gamma': 0.001}
Best Cross-Validation Score: 0.9909528648857917
Best Estimator: SVC(C=1, gamma=0.001)
```

In [ ]:

Using grid search and cross-validation on the popular digits dataset from the scikit-learn module, this Python method tunes the hyperparameters of a Support Vector Machine (SVM) classifier. The code starts by importing the dataset and dividing it into training and testing sets. A random state is then set to ensure reproducibility. Using a parameter grid, the best values for two critical hyperparameters—gamma, which defines the influence of a single training example, and C, which regulates the trade-off between a smooth decision boundary and accurately classifying training points—are found in order to maximize the performance of the SVM model.
The various combinations of C and gamma are then assessed using a grid search with 5-fold cross-validation, with the combination that produces the highest cross-validation score being chosen in the end. The code provides a thorough picture of the model's tuning procedure and ultimate performance by printing out the test score, optimal parameters, best cross-validation score, and best estimator. This method guarantees that the SVM model retains its high accuracy while generalizing well to new data.