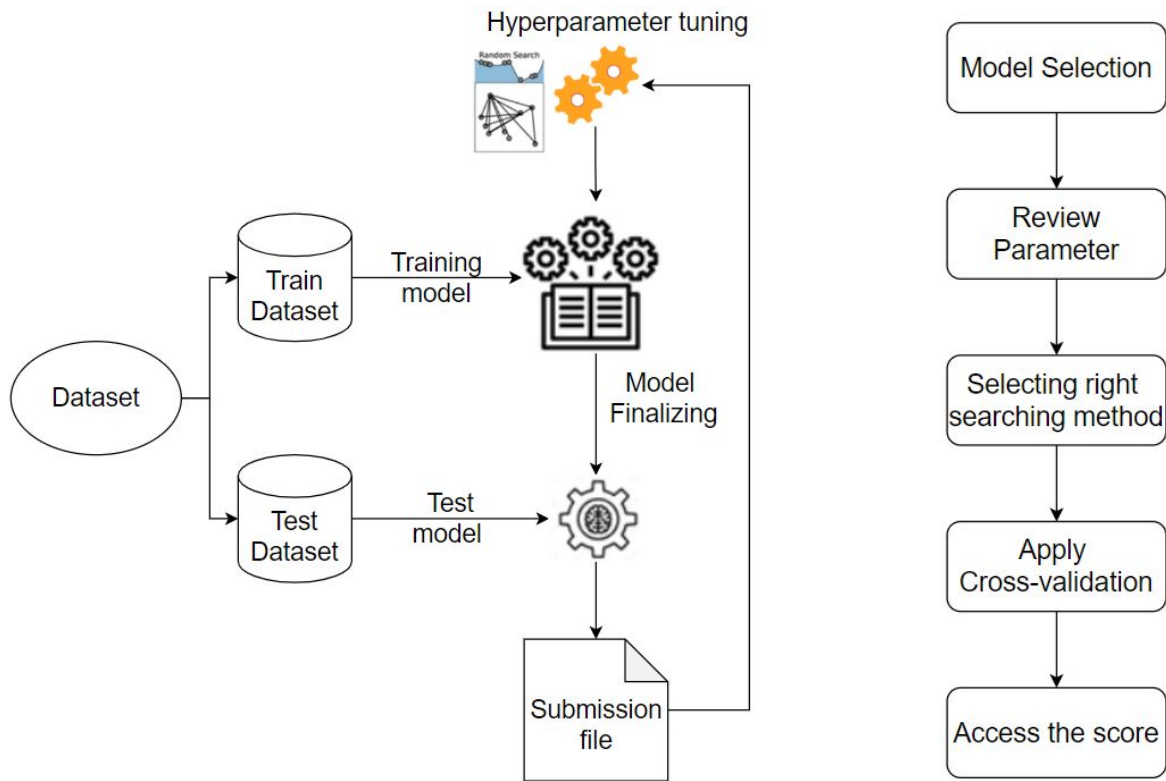


# Hyperparameter Optimization



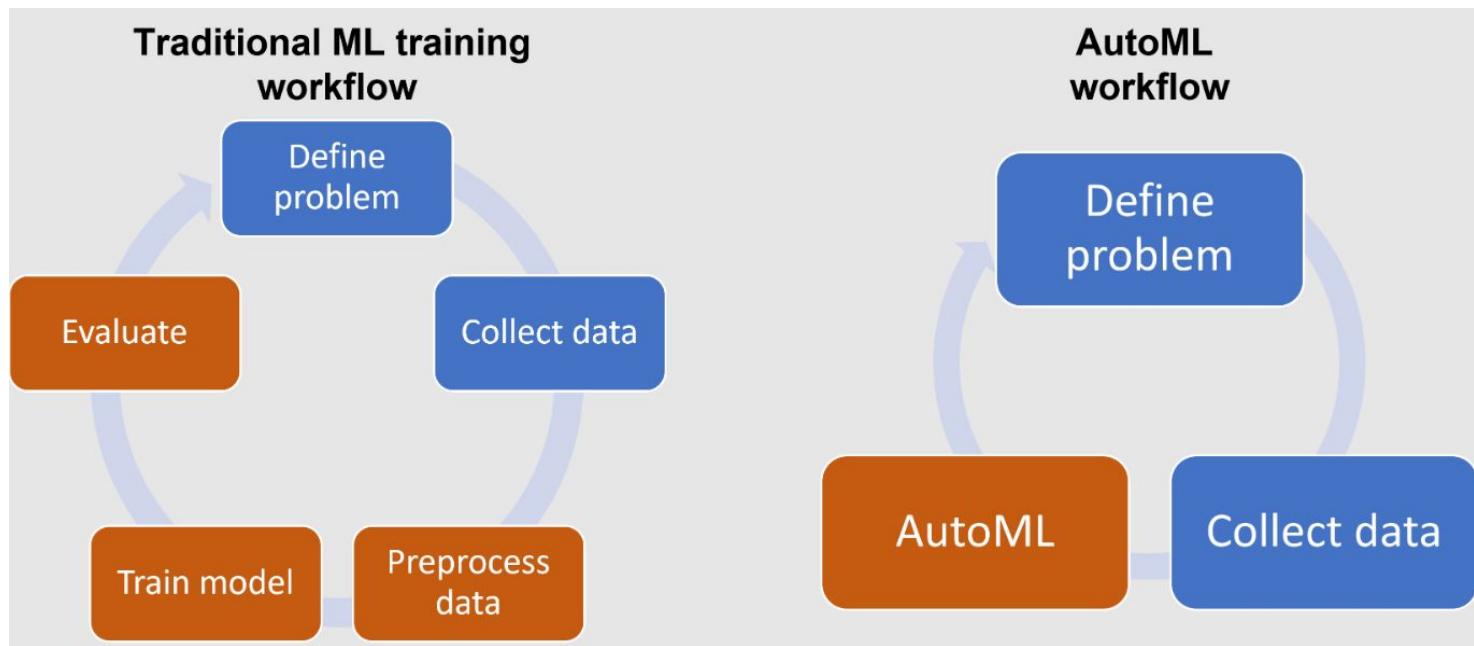
# Nội dung

1. Basic optimization techniques
2. Key parameters and how to use them
3. Bayesian optimization
4. Search Hyperparameters for Deep Learning models



# 1 - Basic optimization techniques

Kiến thức Hyperparameter Optimization liên quan đến việc xây dựng hệ thống AutoML



Nguồn: [microsoft-ML-blog: What is Automated Machine Learning?](https://microsoft-ml-blog.com/2018/05/01/what-is-automated-machine-learning/)

# 1 - Basic optimization techniques

Một số thành phần cần thiết để phân tích và sử dụng:

- Một model với các siêu tham số (hyperparameters) cần phải được tối ưu hóa
- Một không gian tìm kiếm (search space) chứa các vùng biên giá trị để tìm kiếm giữa mỗi hyperparameter.
- Một cơ chế cross-validation.
- Một evaluation metric và objective function

=> Những thành phần này liên kết với nhau để quyết định giải pháp tối ưu.

# 1 - Basic optimization techniques

## Sự khác biệt giữa parameter và hyperparameter

Parameter	Hyperparameter
Là những tham số bên trong một model	Là những tham số chỉ định và kiểm soát quá trình training model
Quá trình predict yêu cầu các tham số trong model	Quá trình tối ưu model cần sử dụng các hyperparameter
Được cập nhật trong quá trình training model	Được thiết lập trước khi training model
Parameter tự động thay đổi bởi model	Hyperparameter được thiết lập bởi người lập trình
=> Internal to the model	=> External to the model

# 1 - Basic optimization techniques

## Sự khác biệt giữa parameter và hyperparameter

Parameter	Hyperparameter
Là những tham số bên trong một model	Là những tham số chỉ định và kiểm soát quá trình training model
Quá trình predict yêu cầu các tham số trong model	Quá trình tối ưu model cần sử dụng các hyperparameter
Được cập nhật trong quá trình training model	Được thiết lập trước khi training model
Parameter tự động thay đổi bởi model	Hyperparameter được thiết lập bởi người lập trình
=> Internal to the model	=> External to the model

## 2 - Key parameters and how to use them

### 1- Linear Models

Hai mô hình linear phổ biến trong machine learning thường được sử dụng là Linear Regression và Logistic Regression kết hợp với regularization

- **C**: Giá trị C càng nhỏ thì regularization càng lớn.
- **alpha** (Lasso, Ridge, ElasticNet): Kiểm soát quá trình regularization, alpha là số dương. Với Lasso thì alpha càng cao sẽ mất càng nhiều thời gian xử lý.
- **l1\_ratio**: Thường sử dụng tỷ lệ trong list [.1, .5, .7, .9, .95, .99, 1], và chỉ dùng với ElasticNet.

## 2 - Key parameters and how to use them

### 2- Support-Vector Machines (SVM)

SVM là một thuật toán supervised learning hiệu quả cho bài toán classification và regression - nó có thể tự động fit linear và non-linear models.

- **C**: giá trị phạt (penalty value). Giảm C làm cho biên độ giữa các class lớn vì vậy sẽ bỏ qua nhiều nhiễu (noise) hơn làm cho model được tổng quát hơn. Giá trị tốt nhất cho C thường nằm trong range **`np.logspace(-3, 3, 7)`**.
- **kernel**: Tham số quyết định mức độ phi tuyến tính (non-linearity) trong SVM. Thường được set trong ['linear', 'poly', 'rbf', 'sigmoid']. Kernel thường được dùng nhiều nhất là rbf.
- **degree**: Được dùng với kernel='poly' - là số chiều của polynomial expansion. Giá trị tốt nhất thường nằm trong [2, 5]



## 2 - Key parameters and how to use them

### 2- Support-Vector Machines (SVM)

- **gamma**: hệ số cho kernel 'rbf', 'poly' và 'sigmoid'. Giá trị gamma cao có thể fit data tốt hơn tuy nhiên dễ dẫn đến hiện tượng overfitting. Với gamma thấp, đường SVM sẽ ít bị ảnh hưởng bởi local data và giúp cho kết quả sẽ được hiệu quả hơn. Giá trị hiệu quả thường nằm trong **np.logspace(-3, 3, 7)**.
- **nu** (Regression - nuSVR, Classification - nuSVC): Tham số set một giá trị bỏ qua (tolerance) các điểm dữ liệu nằm gần lề (margin) và không được phân loại chính xác (bỏ qua các điểm phân loại sai gần margin). Giá trị hiệu quả thường nằm trong [0, 1]

## 2 - Key parameters and how to use them

### 3- Support-Vector Machines (SVM)

- **epsilon**: Xác định độ lỗi của SVR bằng cách xác định một phạm vi rộng cho epsilon - phạm vi không phạt cho những dự đoán sai. Giá trị hiệu quả nằm trong **`np.logspace(-4, 2, 7)`**
- **penalty, loss** và **dual**: Thường là tổ hợp của ('l1', 'squared\_hinge', False), ('l2', 'hinge', True), ('l2', 'squared\_hinge', True), and ('l2', 'squared\_hinge', False) trong LinearSVC. Với ('l2', 'hinge', True) chính là SVC(kernel='linear').

## 2 - Key parameters and how to use them

### 3- Random Forests

Tương tự như SVM Random Forests cũng được sử dụng cho 2 task Classification - RandomForestClassifier và Regression - RandomForestRegressor.

- **max\_features**: Số lượng feature được lấy ở mỗi lần phân tách (split) - max\_features có thể quyết định tới performance của RandomForest. Max\_features càng thấp thì tốc độ càng nhanh.
- **min\_samples\_leaf**: Thông số xác định đến độ sâu (depth) của RandomForest. min\_samples\_leaf càng cao làm giảm variance và tăng bias.
- **bootstrap**: mang giá trị True/False cho phép sử dụng bootstrapping.
- **n\_estimators**: số lượng tree của RandomForest. Số lượng tree càng nhiều thì chi phí tính toán càng lớn.