🔹 **CHƯƠNG 1: Welcome**

**Giới thiệu phương pháp học Machine Learning hiệu quả:**

* KHÔNG nên bắt đầu bằng việc học cú pháp Python hay lý thuyết thuật toán một cách quá chi tiết, vì điều này có thể khiến bạn mất định hướng và mất nhiều thời gian.
* Thay vào đó, nên bắt đầu từ **dự án thực tế** – tức là học qua việc thực hành quy trình giải bài toán thực tế bằng ML.

**Quy trình một dự án Machine Learning thực tế gồm 6 bước:**

1. **Xác định vấn đề:** Hiểu rõ bài toán cần giải quyết là gì.
2. **Phân tích dữ liệu:** Khám phá dữ liệu bằng thống kê và trực quan hóa.
3. **Chuẩn bị dữ liệu:** Làm sạch, chuyển đổi, chuẩn hóa dữ liệu.
4. **Đánh giá thuật toán:** Dùng nhiều thuật toán khác nhau để đánh giá hiệu quả.
5. **Cải thiện kết quả:** Tối ưu thuật toán, dùng ensemble hoặc tuning tham số.
6. **Trình bày mô hình cuối cùng:** Lưu mô hình và triển khai.

**Cấu trúc sách:**

* **Lessons:** Hướng dẫn từng phần trong quy trình trên.
* **Projects:** Làm dự án end-to-end để tổng hợp kiến thức.
* **Recipes:** Mẫu mã nguồn Python có thể dùng lại nhanh chóng.

🔹 **CHƯƠNG 2: Python Ecosystem for Machine Learning**

**Lý do nên dùng Python cho ML:**

* Dễ học, phổ biến, hỗ trợ nhiều thư viện.
* Có thể dùng cho cả nghiên cứu lẫn triển khai hệ thống thực tế.

**Các thành phần chính trong hệ sinh thái Python ML:**

* **NumPy:** Xử lý dữ liệu dạng mảng hiệu quả.
* **Pandas:** Làm việc với dữ liệu dạng bảng (DataFrame).
* **Matplotlib:** Vẽ biểu đồ, trực quan hóa dữ liệu.
* **scikit-learn:** Thư viện chứa các thuật toán học máy và công cụ tiền xử lý, đánh giá mô hình.

**Gợi ý:** Dùng **Anaconda** để cài nhanh toàn bộ công cụ cần thiết.

🔹 **CHƯƠNG 3: Crash Course in Python and SciPy**

**Giới thiệu nhanh về ngôn ngữ Python và các thư viện liên quan:**

* **Python:** Biến, gán giá trị, điều kiện, vòng lặp, hàm, tuple, list, dictionary.
* **NumPy:** Tạo mảng, indexing, các phép toán vector.
* **Matplotlib:** Tạo biểu đồ đơn giản (line plot, scatter plot).
* **Pandas:** Tạo Series và DataFrame, truy cập dữ liệu bằng nhãn hoặc chỉ số.

**Mục tiêu:** Cung cấp đủ nền tảng để đọc và viết code ML cơ bản bằng Python.

🔹 **CHƯƠNG 4: How To Load Machine Learning Data**

**Các cách để đọc dữ liệu từ file CSV vào Python:**

1. **Dùng csv.reader:** Thuần túy Python chuẩn.
2. **Dùng numpy.loadtxt():** Nhanh, hiệu quả, nhưng ít linh hoạt hơn.
3. **Dùng pandas.read\_csv():** Linh hoạt, dễ dùng, hỗ trợ đặt tên cột – nên ưu tiên.

**Những điểm cần lưu ý khi load dữ liệu:**

* Có dòng tiêu đề (header) hay không?
* Dấu phân tách (delimiter) là gì?
* Có dòng comment không?
* Dữ liệu có được bao bởi dấu nháy (quotes) không?

🔹 **CHƯƠNG 5: Understand Your Data With Descriptive Statistics**

**Các bước cần thiết để hiểu rõ dữ liệu:**

* Dùng head() để xem vài dòng đầu tiên của dataset.
* Kiểm tra kích thước dữ liệu với .shape.
* Dùng .dtypes để biết loại dữ liệu của mỗi cột.
* Thống kê mô tả với describe() (trung bình, min, max, std...).
* Đếm số lượng của mỗi class (nếu là bài toán phân loại).
* Tính ma trận tương quan giữa các thuộc tính.
* Kiểm tra độ lệch (skewness) của từng cột.

🔹 **CHƯƠNG 6: Understand Your Data With Visualization**

**Trực quan hóa dữ liệu giúp phát hiện xu hướng, ngoại lệ và mối quan hệ:**

* **Biến đơn (Univariate):**
  + Histogram (tần suất xuất hiện).
  + Density plot (đường mật độ).
  + Box plot (phân bố và ngoại lệ).
* **Biến đôi hoặc nhiều (Multivariate):**
  + Scatter plot (phân tán).
  + Correlation matrix.
  + Scatter matrix.

**Công cụ dùng:** matplotlib, pandas.plot().

🔹 **CHƯƠNG 7: Prepare Your Data For Machine Learning**

**Tại sao cần chuẩn bị dữ liệu?**

* Dữ liệu thực tế thường không ở dạng "hoàn hảo" để thuật toán học tốt.
* Tiền xử lý giúp cải thiện độ chính xác và tốc độ huấn luyện.

**Các kỹ thuật phổ biến để xử lý dữ liệu:**

* **Rescaling (Min-Max Scaling):** Đưa dữ liệu về cùng thang đo.
* **Standardization:** Biến đổi dữ liệu về phân phối chuẩn (mean = 0, std = 1).
* **Normalization:** Đưa vector đặc trưng về độ dài = 1.
* **Binarization:** Chuyển dữ liệu sang dạng nhị phân.
* **One-hot encoding:** Mã hóa biến phân loại (categorical).
* **Imputation:** Thay thế dữ liệu thiếu bằng trung bình, trung vị...

**Thư viện:** sklearn.preprocessing

🔹 **CHƯƠNG 8: Feature Selection For Machine Learning**

**Mục tiêu:** Chọn ra các đặc trưng (feature) quan trọng nhất để giảm nhiễu và tăng hiệu quả.

**Kỹ thuật lựa chọn đặc trưng:**

* **Univariate Selection:** Chọn đặc trưng dựa trên kiểm định thống kê (chi-square, ANOVA...).
* **Recursive Feature Elimination (RFE):** Loại bỏ dần đặc trưng ít quan trọng.
* **Principal Component Analysis (PCA):** Biến đổi đặc trưng sang không gian mới ít chiều hơn.
* **Feature Importance:** Dùng thuật toán như Random Forest để xác định độ quan trọng đặc trưng.

🔹 **CHƯƠNG 9: Evaluate Algorithms with Resampling**

**Tại sao cần đánh giá mô hình đúng cách?**

* Đánh giá sai có thể chọn nhầm mô hình.

**Các kỹ thuật resampling:**

* **Train/Test Split:** Chia 1 phần làm train, phần còn lại test.
* **K-Fold Cross Validation:** Chia tập dữ liệu thành K phần và luân phiên làm test.
* **Leave-One-Out Cross Validation (LOOCV):** Mỗi lần chỉ lấy 1 điểm làm test.
* **Repeated Random Test-Train Splits:** Random nhiều lần và lấy trung bình kết quả.

**Công cụ dùng:** cross\_val\_score()

🔹 **CHƯƠNG 10: Performance Metrics**

**Chọn metric phù hợp với bài toán là rất quan trọng:**

**Đối với Classification:**

* **Accuracy:** Tỷ lệ dự đoán đúng.
* **LogLoss:** Đánh giá xác suất đầu ra.
* **ROC AUC:** Đánh giá khả năng phân biệt class.
* **Confusion Matrix:** Phân tích chi tiết đúng/sai theo class.
* **Precision, Recall, F1 Score:** Đặc biệt quan trọng với dữ liệu mất cân bằng.

**Đối với Regression:**

* **MAE (Mean Absolute Error):** Sai số trung bình tuyệt đối.
* **MSE (Mean Squared Error):** Sai số bình phương trung bình.
* **RMSE:** Căn bậc hai của MSE.
* **R² Score:** Đo độ phù hợp của mô hình.

🔹 **CHƯƠNG 11: Spot-Check Classification Algorithms**

**Ý tưởng:** Thử nhanh nhiều thuật toán phân loại để biết cái nào phù hợp.

**Các thuật toán được kiểm tra:**

* Logistic Regression.
* K-Nearest Neighbors (KNN).
* Decision Tree.
* Naive Bayes.
* Support Vector Machine (SVM).

**Đánh giá bằng cross-validation để khách quan.**

🔹 **CHƯƠNG 12: Spot-Check Regression Algorithms**

**Thử nhanh các thuật toán hồi quy phổ biến:**

* Linear Regression.
* Lasso và ElasticNet.
* KNN Regressor.
* Decision Tree Regressor.
* Support Vector Regressor (SVR).

🔹 **CHƯƠNG 13: Compare Machine Learning Algorithms**

**So sánh hiệu suất nhiều thuật toán:**

* Dùng danh sách mô hình và lặp để huấn luyện + đánh giá.
* Vẽ biểu đồ **boxplot** để thấy sự khác biệt hiệu suất.

**Giúp chọn ra mô hình tốt nhất để tiếp tục cải tiến.**

🔹 **CHƯƠNG 14: Automate Workflows with Pipelines**

**Pipelines giúp tổ chức quy trình ML rõ ràng và hiệu quả:**

* Gộp bước xử lý dữ liệu và huấn luyện vào 1 khối duy nhất.
* Dễ tái sử dụng, giảm lỗi lập trình.

**Ví dụ:** Chuẩn hóa + mô hình trong 1 pipeline: Pipeline([('scaler', StandardScaler()), ('model', SVM())])

🔹 **CHƯƠNG 15: Improve Performance with Ensembles**

**Kết hợp nhiều mô hình để nâng cao độ chính xác:**

* **Bagging:** Mô hình độc lập → Random Forest.
* **Boosting:** Mô hình tuần tự → AdaBoost.
* **Voting:** Kết hợp nhiều mô hình khác loại.

**Thư viện:** BaggingClassifier, AdaBoostClassifier, VotingClassifier từ sklearn.

🔹 **CHƯƠNG 16: Improve Performance with Algorithm Tuning**

**Tối ưu tham số giúp mô hình đạt hiệu quả cao nhất:**

* **Grid Search:** Thử tất cả tổ hợp tham số.
* **Random Search:** Thử ngẫu nhiên vài tổ hợp.

**Thư viện:** GridSearchCV, RandomizedSearchCV

🔹 **CHƯƠNG 17: Save and Load Machine Learning Models**

**Sau khi huấn luyện xong mô hình:**

* Lưu lại để tránh huấn luyện lại.
* Dễ dàng dùng lại trong production hoặc deploy.

**Cách lưu mô hình:**

* **pickle:** pickle.dump(model) và pickle.load().
* **joblib:** Tối ưu cho dữ liệu lớn – joblib.dump(), joblib.load().