**Chương 1 - 4: Tổng quan Python + Load dữ liệu**

**Chương 1**

* Machine Learning trong Python nên học bằng cách **làm project thực tế**, không sa đà lý thuyết hay chỉ học ngôn ngữ Python.
* Các bước làm một dự án Machine Learning bao gồm:
  + Xác định bài toán (Define Problem).
  + Phân tích dữ liệu (Analyze Data).
  + Chuẩn bị dữ liệu (Prepare Data).
  + Đánh giá mô hình (Evaluate Algorithms).
  + Cải thiện kết quả (Improve Results).
  + Trình bày mô hình (Present Results).

**Chương 2**

* **Python** là một ngôn ngữ lập trình cực kỳ phù hợp để làm Machine Learning.
* **SciPy ecosystem** gồm: numpy, matplotlib, pandas giúp xử lý số liệu và đồ họa.
* **Scikit-learn** cung cấp gần như đầy đủ các thuật toán Machine Learning.
* Có thể cài đặt dễ dàng qua bộ **Anaconda**.

**Chương 3**

* Ôn tập nhanh cú pháp Python cơ bản: string, number, boolean, tuple, list, dictionary, if-else, for, while, function.
* **NumPy** hỗ trợ làm việc với mảng số liệu nhanh chóng.
* **Matplotlib** giúp vẽ biểu đồ: line plot, scatter plot.
* **Pandas** cung cấp Series và DataFrame để tổ chức, phân tích dữ liệu linh hoạt hơn.

**Chương 4**

* Dữ liệu Machine Learning thường lưu ở định dạng CSV.
* Có 3 cách phổ biến để load dữ liệu CSV:
  + **Dùng thư viện csv** chuẩn trong Python (đọc file dòng dòng).
  + **Dùng NumPy** (loadtxt) để load nhanh thành mảng số liệu.
  + **Dùng Pandas** (read\_csv) cực kỳ tiện để load thành DataFrame và thao tác dữ liệu ngay.

**Chương 5 - 6: EDA**

**Chương 5**

* Cần phải hiểu dữ liệu ban đầu để làm tốt Machine Learning.
* Các bước phân tích dữ liệu cơ bản:
  + **Xem nhanh** vài dòng đầu tiên (peek at data) để hiểu bố cục, giá trị.
  + **Kiểm tra kích thước** dữ liệu (số dòng, số cột) để xem độ lớn dataset.
  + **Kiểm tra kiểu dữ liệu** (int, float...) của từng cột.
  + **Thống kê mô tả**: trung bình, min, max, độ lệch chuẩn của từng thuộc tính.
  + **Phân phối nhãn** (class distribution): xem tỷ lệ các lớp trong bài toán classification.
  + **Tính hệ số tương quan** giữa các thuộc tính để tìm các mối quan hệ tuyến tính.
  + **Kiểm tra độ lệch phân phối** (skew) để biết có cần chuẩn hóa hay không.

**Chương 6**

* Sau thống kê mô tả, cần trực quan hóa để nhìn dữ liệu dễ dàng hơn.
* Các loại biểu đồ phổ biến:
  + **Histogram:** vẽ phân phối giá trị của từng thuộc tính.
  + **Density plot:** vẽ phân phối xác suất mượt hơn histogram.
  + **Boxplot:** vẽ giá trị trung vị, khoảng min-max và phát hiện outlier.
* Đồ thị đa biến:
  + **Heatmap tương quan**: trực quan hóa hệ số tương quan giữa các biến.
  + **Scatter plot matrix**: vẽ scatter plot giữa tất cả các cặp biến để tìm quan hệ tiềm ẩn.

**Chương 7 - 8: Tiền xử lý dữ liệu**

**Chương 7**

* Tiền xử lý giúp mô hình học nhanh hơn, chính xác hơn.
* Các bước tiền xử lý:
  + **Rescale**: Đưa giá trị về khoảng [0,1] bằng MinMaxScaler.
  + **Standardize**: Đưa dữ liệu về phân phối chuẩn N(0,1) bằng StandardScaler.
  + **Normalize**: Đưa từng hàng dữ liệu thành vector đơn vị bằng Normalizer.
  + **Binarize**: Chuyển dữ liệu thành dạng nhị phân (0 hoặc 1) bằng Binarizer.

**Chương 8**

* Chọn ra những đặc trưng (feature) quan trọng nhất giúp mô hình đơn giản và hiệu quả.
* Các phương pháp chọn đặc trưng:
  + **Univariate Selection**: Dựa trên kiểm định thống kê (chi-squared test).
  + **Recursive Feature Elimination (RFE**): Lọc đặc trưng dựa trên mô hình (ví dụ: Logistic Regression).
  + **Principal Component Analysis (PCA)**: Biến đổi features thành tập mới giữ lại phương sai lớn nhất.
  + **Feature Importance**: Đo độ quan trọng features bằng mô hình cây (tree models).

**Chương 9 - 14: Mô hình ML**

**Chương 9**

* Đánh giá hiệu suất của thuật toán máy học: Sử dụng phương pháp resampling để ước lượng độ chính xác.
* Kỹ thuật đánh giá:
  + **Train/Test Sets**: Chia dữ liệu thành hai phần để huấn luyện và kiểm tra.
  + **K-fold Cross Validation**: Chia dữ liệu thành k phần, huấn luyện trên k-1 phần và kiểm tra trên phần còn lại.
  + **Leave One Out Cross Validation**: Mỗi lần chỉ giữ lại một mẫu để kiểm tra.
  + **Repeated Random Test-Train Splits**: Lặp lại quá trình phân chia dữ liệu ngẫu nhiên nhiều lần.

**Chương 10**

* Các chỉ số đánh giá thuật toán: Quan trọng cho việc so sánh các mô hình.
* Các chỉ số cho bài toán phân loại:
  + **Accuracy**: Tỷ lệ dự đoán đúng so với tổng số dự đoán.
  + **Logarithmic Loss**: Đánh giá xác suất dự đoán.
  + **Area Under ROC Curve (AUC**): Đánh giá khả năng phân biệt giữa các lớp.
  + **Confusion Matrix**: Bảng hiển thị số lượng dự đoán đúng và sai.
  + **Classification Report**: Báo cáo chi tiết về độ chính xác và độ nhạy.

**Chương 11**

* Spot-Check Thuật toán phân loại: Thử nghiệm với nhiều thuật toán khác nhau để tìm ra mô hình tốt nhất.
* Thuật toán được thử nghiệm:
  + **Logistic Regression**
  + **Linear Discriminant Analysis (LDA)**
  + **k-Nearest Neighbors (KNN)**
  + **Classification and Regression Trees (CART)**
  + **Naive Bayes**
  + **Support Vector Machines (SVM)**

**Chương 12**

* Spot-Check Thuật toán hồi quy: Tương tự như chương 11 nhưng cho bài toán hồi quy.
* Thuật toán hồi quy:
  + **Linear Regression**
  + **Ridge Regression**
  + **LASSO Regression**
  + **Elastic Net Regression**
  + **k-Nearest Neighbors (KNN)**
  + **Decision Trees (CART)**
  + **Support Vector Regression (SVR)**

**Chương 13**

* So sánh các thuật toán máy học: Thiết kế thử nghiệm để so sánh nhiều thuật toán.
* Các bước:
  + Đảm bảo mỗi thuật toán được đánh giá trên cùng một dữ liệu.
  + Sử dụng boxplot để trực quan hóa kết quả so sánh.

**Chương 14**

* Tự động hóa quy trình làm việc với Pipelines: Sử dụng Pipelines trong scikit-learn để tự động hóa các quy trình.
* Lợi ích Pipelines:
  + Giảm thiểu rò rỉ dữ liệu.
  + Tổ chức quy trình chuẩn cho việc chuẩn bị dữ liệu và mô hình hóa.

**Chương 15 - 17: Tinh chỉnh mô hình**

**Chương 15**

* Cải thiện hiệu suất với các phương pháp Ensemble: Tạo ra các mô hình mạnh mẽ hơn bằng cách kết hợp nhiều mô hình.
* Các phương pháp:
  + **Bagging**: Tạo nhiều mô hình từ các mẫu khác nhau.
  + **Boosting**: Xây dựng các mô hình tuần tự, mỗi mô hình cải thiện sai sót của mô hình trước.
  + **Voting**: Kết hợp dự đoán từ nhiều mô hình khác nhau.

**Chương 16**

* Tuning thuật toán: Tinh chỉnh tham số của mô hình để cải thiện hiệu suất.
* Các phương pháp tuning:
  + **Grid Search**: Tìm kiếm tất cả các tổ hợp tham số.
  + **Random Search**: Tìm kiếm ngẫu nhiên trong khoảng tham số.

**Chương 17**

*  Lưu và tải mô hình máy học: Sử dụng pickle và Joblib để lưu trữ mô hình cho việc sử dụng sau này.
*  Lưu ý khi lưu mô hình: Đảm bảo phiên bản Python và các thư viện tương ứng khi tải lại mô hình.