**A picture containing application

Description automatically generated**

**TRƯỜNG ĐẠI HỌC SÀI GÒN**

**KHOA CÔNG NGHỆ THÔNG TIN**

**HỌC PHẦN: Nghiên cứu khoa học trong CNTT**

**BÀI TẬP 7: CÁC ĐIỂU ĐÃ HỌC TRONG SÁCH MACHINE LEARNING**

**Nhóm sinh viên thực hiện (nhóm 17)**

Nguyễn Đức Duy Lâm 3122410208

Mai Phúc Lâm 3122410207

Nguyễn Hữu Lộc 3122410213

Văn Tuấn Kiệt 3122410202

**Giảng viên hướng dẫn**

Đỗ Như Tài

**TP. HCM THÁNG 05/2025**



**Bài tập 7: NCKH: CÁC ĐIỀU ĐÃ HỌC TRONG SÁCH MACHINE LEARNING**

**Mục lục**

[❖ Chương 2: Python Ecosystem for Machine Learning 2](#_avz0n8wg0gbg)

[❖ Chương 3: Crash Course in Python and SciPy 2](#_loxdkzfs72gk)

[❖ Chương 4: How To Load Machine Learning Data 2](#_jfq41w3tkvwp)

[❖ Chương 5: Understand Your Data With Descriptive Statistics 3](#_qmkbrycebgza)

[❖ Chương 6: Understand Your Data With Visualization 4](#_frac6yp06qxn)

[❖ Chương 7: Prepare Your Data For Machine Learning 4](#_n3j3xug259ab)

[❖ Chương 8: Feature Selection For Machine Learning 5](#_gv2wmdwwj0tn)

[❖ Chương 9: Evaluate the Performance of Machine Learning Algorithms with Resampling 5](#_qxdmy0bo95js)

[❖ Chương 10: Machine Learning Algorithm Performance Metrics 6](#_c0c5qr9m0pfi)

[❖ Chương 11: Spot-Check Classification Algorithms 7](#_a58ermi54iv3)

[❖ Chương 12: Spot-Check Regression Algorithms 7](#_6yd48ynmnntb)

[❖ Chương 13: Compare Machine Learning Algorithms 8](#_81frjyitd394)

[❖ Chương 14: Automate Machine Learning Workflows with Pipelines 8](#_i2grj3o72jz)

[❖ Chương 15: Improve Performance with Ensembles 9](#_iao22ejxphme)

[❖ Chương 16 :Improve Performance with Algorithm Tuning 10](#_17ygsge2ujgf)

[❖ Chương 17:Save and Load Machine Learning Models 10](#_74vly0p2sdzi)

[❖ Chương 18 :Predictive Modeling Project Template 11](#_gorxgafceplv)

[❖ Chương 19: Your First Machine Learning Project In Python Step-By-Step 11](#_x3j0er4e12yh)

[❖ Chương 20: Regression Machine Learning Case Study Project 11](#_ry9akageyd07)

[❖ Chương 21: Binary Classification Machine Learning Case Study Project 12](#_6bp9knw1s7sy)

## **Chương 2: Python Ecosystem for Machine Learning**

* Hiểu được vì sao Python lại được dùng phổ biến trong Machine Learning
* Biết đến các thư viện quan trọng trong hệ sinh thái của Python như
* NumPy: Giúp xử lý dữ liệu nhanh chóng dưới dạng mảng.
* Matplotlib: Dùng để vẽ biểu đồ, giúp mình hiểu rõ dữ liệu hơn.
* Pandas: Hỗ trợ đọc file, xử lý và phân tích dữ liệu.
* Scikit-learn: Thư viện chính để áp dụng các thuật toán Machine Learning như phân loại, hồi quy,… Nó còn có các công cụ đánh giá và xử lý dữ liệu rất tiện.
* Cuối chương là phần hướng dẫn cài đặt môi trường.
* Chương này giúp em có cái nhìn tổng quan và chuẩn bị sẵn môi trường để bắt đầu học Machine Learning với Python.

## **Chương 3: Crash Course in Python and SciPy**

* Các cú pháp cơ bản trong Python ( gán không cần khai báo kiểu dữ liệu, các kiểu dữ liệu)
* Cấu trúc điều khiển
* Cấu trúc dữ liệu (tuple, list, dictionary)
* Thư viện NumPy (thao tác với mảng)
* Thư viện Matplotlib (tạo biểu đồ)

## **Chương 4: How To Load Machine Learning Data**

* Mục tiêu: Tìm hiểu 3 cách phổ biến để tải dữ liệu CSV trong Python trước khi bắt đầu bất kỳ dự án Machine Learning nào.
* Nội dung chính:
* CSV là định dạng phổ biến cho dữ liệu Machine Learning.
* Cần lưu ý: có header không, ký tự phân cách (delimiter), có comment không, có dùng dấu ngoặc kép không.
* Bộ dữ liệu sử dụng: Pima Indians Diabetes (768 dòng, 9 cột, bài toán phân loại nhị phân).
* Tải CSV bằng thư viện chuẩn Python:
* Dùng csv.reader + numpy để đọc và chuyển sang mảng.
* Kết quả: (768, 9)
* Tải CSV bằng NumPy:
* Dùng numpy.loadtxt, phù hợp nếu không có header và cùng định dạng dữ liệu.
* Kết quả: (768, 9)
* Tải CSV bằng Pandas:
* Dùng pandas.read\_csv là cách linh hoạt và được khuyên dùng nhất.
* Kết quả: (768, 9)

## **Chương 5: Understand Your Data With Descriptive Statistics**

* Mục tiêu: Áp dụng các thống kê mô tả để hiểu rõ dữ liệu trước khi đưa vào mô hình Machine Learning.
* Nội dung chính:
* 1. Xem trước dữ liệu bằng .head(20).
* 2. Kiểm tra kích thước dữ liệu bằng .shape.
* 3. Kiểm tra kiểu dữ liệu từng cột bằng .dtypes.
* 4. Tóm tắt thống kê bằng .describe().
* 5. Kiểm tra phân bố nhãn (class) bằng groupby().size().
* 6. Ma trận tương quan giữa các thuộc tính bằng .corr().
* 7. Kiểm tra độ nghiêng (skewness) của phân phối dữ liệu bằng .skew().
* Ví dụ sử dụng Pima Indians Dataset:
* Số dòng và cột: (768, 9)
* Class phân bố: 500 người không mắc (0), 268 người mắc (1)
* Một số thuộc tính có skew cao: test (2.27), pedi (1.91), age (1.12)

## **Chương 6: Understand Your Data With Visualization**

* Mục tiêu: Sử dụng biểu đồ để hiểu rõ dữ liệu, giúp cải thiện lựa chọn và hiệu quả của các thuật toán Machine Learning.
* Trực quan hóa đơn biến (Univariate Plots)
* Histogram – Cho thấy phân phối và các giá trị ngoại lai (outlier).
* Density Plot – Đường cong mật độ mượt mà hơn, dễ hình dung phân phối.
* Boxplot (Box and Whisker Plot) – Thể hiện trung vị, tứ phân vị và các giá trị bất thường.
* Trực quan hóa đa biến (Multivariate Plots)
* Correlation Matrix – Cho biết mức độ tương quan giữa các cặp biến.
* Scatter Plot Matrix – Biểu đồ phân tán giữa các cặp thuộc tính.

## **Chương 7: Prepare Your Data For Machine Learning**

* Mục tiêu: Nhiều thuật toán Machine Learning yêu cầu dữ liệu cần được xử lý trước. Chương này hướng dẫn bạn cách chuẩn bị dữ liệu bằng scikit-learn.
* Các kỹ thuật xử lý dữ liệu:
* Rescale (chuẩn hóa về khoảng 0-1): dùng MinMaxScaler để đưa các thuộc tính về cùng thang đo.
* Standardize (đưa về phân phối chuẩn N(0,1)): dùng StandardScaler.
* Normalize (chuẩn hóa độ dài vector = 1): dùng Normalizer cho dữ liệu thưa (nhiều số 0).
* Binarize (nhị phân hóa): chuyển giá trị > threshold thành 1, còn lại là 0.

## **Chương 8: Feature Selection For Machine Learning**

* Mục tiêu: Áp dụng các kỹ thuật chọn đặc trưng (feature selection) để nâng cao hiệu suất mô hình.
* Nội dung chính:
* 1. Univariate Selection – chọn các đặc trưng có liên quan thống kê mạnh nhất.
* 2. Recursive Feature Elimination (RFE) – loại bỏ lặp đặc trưng không quan trọng.
* 3. Principal Component Analysis (PCA) – giảm chiều dữ liệu bằng biến chính.
* 4. Feature Importance – đánh giá mức độ quan trọng qua cây quyết định.
* Ví dụ kết quả:
* Univariate Selection chọn plas, test, mass, age.
* RFE chọn preg, mass, pedi (ranking = 1).
* PCA giữ lại 3 thành phần chính có phương sai cao nhất (88%, 6%, 2.5%).
* ExtraTreesClassifier cho điểm quan trọng nhất: plas (22%), mass (14.5%), age (15.4%).

## **Chương 9: Evaluate the Performance of Machine Learning Algorithms with Resampling**

* Evaluate Machine Learning Algorithms: Bạn cần đánh giá mô hình bằng dữ liệu chưa thấy trước đó. Các kỹ thuật thống kê như resampling giúp ước lượng chính xác hiệu suất của mô hình.
* Split into Train and Test Sets: Chia dữ liệu thành tập huấn luyện và kiểm tra (ví dụ 67/33). Đơn giản và nhanh, nhưng có độ biến thiên cao. Ví dụ sử dụng Logistic Regression trên tập Pima Indians đạt ~75.6%.\
* K-fold Cross Validation: Chia dữ liệu thành k phần (folds), mỗi phần được dùng làm test một lần. Độ chính xác ổn định hơn, thường dùng k=10. Ví dụ Logistic Regression với 10-fold CV đạt ~76.95% ± 4.84%.
* Leave One Out Cross Validation: LOOCV là một biến thể đặc biệt của K-fold với k bằng số mẫu. Chi tiết và chính xác, nhưng tốn thời gian. Ví dụ: Logistic Regression đạt ~76.82% ± 42.19%.
* Repeated Random Test-Train Splits: Chia ngẫu nhiên dữ liệu nhiều lần để huấn luyện và đánh giá. Tốc độ nhanh, kết quả đáng tin cậy. Ví dụ: 10 lần chia ngẫu nhiên đạt ~76.5% ± 1.7%.
* What Techniques to Use When
* Dùng K-fold (k=10) khi cần đánh giá chính xác.
* Train/test split khi dữ liệu lớn hoặc thuật toán chậm.
* LOOCV và repeated split hữu ích khi cân bằng giữa thời gian và độ chính xác.

## **Chương 10: Machine Learning Algorithm Performance Metrics**

* Hiểu về 3 chỉ số là Accuracy (Độ chính xác), Logarithmic Loss (Mất mát Logarit) và Area Under ROC Curve (AUC - Diện tích dưới đường cong ROC)
* Hiểu về hai phương pháp tiện lợi để dự đoán kết quả phân loại: Ma trận Nhầm lẫn (Confusion Matrix) và Báo cáo phân loại (Classification Report).
* Hiểu về ba chỉ số cho các bài toán hồi quy: Mean Absolute Error (Lỗi tuyệt đối trung bình), Mean Squared Error (Lỗi bình phương trung bình) và R2.

## **Chương 11: Spot-Check Classification Algorithms**

* Các thuật toán học máy tuyến tính: Học cách sử dụng hồi quy Logistic và phân tích phân biệt tuyến tính để giải quyết các bài toán phân loại với các mối quan hệ tuyến tính giữa các đặc trưng và nhãn
* Các thuật toán học máy phi tuyến:
* k-Nearest Neighbors (k-NN):
* Naive Bayes
* Cây quyết định (CART)
* Máy vector hỗ trợ (SVM)

## **Chương 12: Spot-Check Regression Algorithms**

* Học cách kiểm tra sơ bộ (spot-check) các thuật toán học máy cho các bài toán hồi quy sử dụng Python và thư viện scikit-learn.
* Lựa chọn các thuật toán phù hợp cho bài toán hồi quy của mình, đồng thời hiểu rõ hơn về cách áp dụng các thuật toán tuyến tính và phi tuyến vào các bài toán dự đoán giá trị liên tục.
* **Chương 13: Compare Machine Learning Algorithms**
* Mục tiêu: So sánh hiệu quả của nhiều thuật toán Machine Learning một cách nhất quán bằng Python và scikit-learn.
* Nội dung chính:
* So sánh các thuật toán trên cùng 1 bộ dữ liệu với cùng phương pháp đánh giá (cross validation).
* Dùng k-fold cross validation với cùng random seed để đảm bảo tính công bằng.
* Các thuật toán được so sánh gồm: Logistic Regression (LR), Linear Discriminant Analysis (LDA), k-Nearest Neighbors (KNN), Decision Tree (CART), Naive Bayes (NB), Support Vector Machine (SVM).
* Kết quả ví dụ:
* LR: 0.769515 (0.048411)
* LDA: 0.773462 (0.051592)
* KNN: 0.726555 (0.061821)

## **Chương 14: Automate Machine Learning Workflows with Pipelines**

* Mục tiêu: Sử dụng Pipelines trong scikit-learn để tự động hóa quy trình tiền xử lý và huấn luyện mô hình, đồng thời tránh rò rỉ dữ liệu (data leakage).
* Nội dung chính:
* Pipelines cho phép chuỗi các bước xử lý dữ liệu và huấn luyện mô hình trong 1 quy trình.
* Đảm bảo rằng các bước xử lý chỉ áp dụng trên dữ liệu của từng fold khi cross-validation.
* Ví dụ 1: Pipeline chuẩn hóa và huấn luyện mô hình
* Chuẩn hóa dữ liệu bằng StandardScaler.
* Huấn luyện mô hình Linear Discriminant Analysis (LDA).
* Đánh giá với KFold cross-validation.
* Kết quả: Accuracy ~ 0.773
* Ví dụ 2: Pipeline trích xuất đặc trưng và huấn luyện mô hình
* Trích xuất đặc trưng bằng PCA (3 features).
* Chọn đặc trưng bằng SelectKBest (6 features).
* Kết hợp đặc trưng bằng FeatureUnion.
* Huấn luyện mô hình Logistic Regression.
* Kết quả: Accuracy ~ 0.776

## **Chương 15: Improve Performance with Ensembles**

* Mục tiêu: Sử dụng các kỹ thuật ensemble để tăng độ chính xác của mô hình, bao gồm Bagging, Boosting và Voting.
* Nội dung chính:
* Bagging: tạo nhiều mô hình trên các tập con khác nhau của dữ liệu.
* Boosting: tạo chuỗi mô hình, mô hình sau sửa lỗi cho mô hình trước.
* Voting: kết hợp dự đoán của nhiều mô hình khác nhau.
* Bagging Algorithms:
* Bagged Decision Trees – Sử dụng 100 cây CART với BaggingClassifier.
* Kết quả: Accuracy ~ 0.770
* Random Forest – 100 cây, chọn 3 đặc trưng ngẫu nhiên tại mỗi split.
* Kết quả: Accuracy ~ 0.770
* Extra Trees – 100 cây, chọn 7 đặc trưng ngẫu nhiên tại mỗi split.
* Kết quả: Accuracy ~ 0.760
* Boosting Algorithms:
* AdaBoost – Sử dụng 30 cây đơn giản, chú trọng vào điểm sai.
* Kết quả: Accuracy ~ 0.760
* Stochastic Gradient Boosting – Sử dụng 100 cây, nâng cao hiệu suất.
* Kết quả: Accuracy ~ 0.764
* Voting Ensemble:
* Kết hợp Logistic Regression, Decision Tree và SVM bằng VotingClassifier.
* Kết quả: Accuracy ~ 0.729

## **Chương 16 :Improve Performance with Algorithm Tuning**

* Ở chương này ta sẽ được làm quen với 2 thuật toán điều chỉnh thuật toán
* Chiến lược điều chỉnh tham số tìm kiếm lưới (grid search algorithm tuning strategy.)
* Chiến lược điều chỉnh tham số tìm kiếm ngẫu nhiên(random search algorithm tuning strategy.)
* ví dụ 1 : sử dụng Ridge Regression (hồi quy Ridge) để tìm tham số alpha tối ưu từ một tập hợp giá trị được định nghĩa trước, thông qua GridSearchCV từ scikit-learn.
* Kết quả trung bình : 0.2761084412929244
* Kết quả tốt nhất : 1.0
* ví dụ 2 : thực hiện tìm kiếm siêu tham số bằng phương pháp ngẫu nhiên (RandomizedSearchCV) để chọn giá trị tốt nhất cho siêu tham số alpha của mô hình Ridge Regression
* Kết quả trung bình : 0.27610755734028525
* Kết quả tốt nhất :0.9779895119966027

## **Chương 17:Save and Load Machine Learning Models**

* Trong chương này ta sẽ tìm hiểu qua 2 thuật toàn :
* Cách sử dụng Pickle để tuần tự và khử tuần tự mô hình máy học
* Cách sử dụng Joblib để tuần tự và khử tuần tự mô hình máy học
* Ví dụ 1 : Thực hiện huấn luyện mô hình Logistic Regression . Lưu và tải lại với thư viện Pickle .Đánh giá trên tập test
* Kết quả accuracy : 0.7834645669291339
* Ví dụ 2 : Thực hiện huấn luyện mô hình Logistic Regression . Lưu và tải lại với thư viện JobLib .Đánh giá trên tập test
* Kết quả accuracy : 0.7874015748031497

## **Chương 18 :Predictive Modeling Project Template**

* Trong chương học về cách cấu trúc dự án với 6 Nhiệm vụ
* Define Problem.
* Summarize Data.
* Prepare Data.
* Evaluate Algorithms.
* Improve Results.
* Present Results.

## **Chương 19: Your First Machine Learning Project In Python Step-By-Step**

* Giúp người mới bắt đầu hoàn thành dự án ML đầu tiên với Python và scikit-learn.
* Nội dung chính:
* Load dữ liệu: Sử dụng thư viện pandas để load dataset từ CSV.
* Tóm tắt dữ liệu: In ra số dòng, kiểu dữ liệu, thống kê mô tả, biểu đồ phân phối.
* Tách dữ liệu: Chia tập dữ liệu thành training và validation set.
* Định nghĩa mô hình: Thử nhiều thuật toán ML khác nhau như: Logistic Regression KNN Decision Tree SVM Naive Bayes
* Dự đoán: Dùng mô hình tốt nhất để dự đoán trên tập validation.
* Dataset: Iris Dataset (phân loại hoa).

## **Chương 20: Regression Machine Learning Case Study Project**

* Triển khai bài toán hồi quy (regression) để dự đoán giá trị số liên tục.
* Nội dung chính:
* Load dữ liệu: Dataset thường là Boston Housing (dự đoán giá nhà).
* Phân tích dữ liệu: Kiểm tra phân phối giá trị, tương quan, trực quan hóa với scatter plot.
* Tiền xử lý: Chuẩn hóa dữ liệu (StandardScaler) để phù hợp với một số mô hình.
* Chọn mô hình: Các thuật toán hồi quy như: Linear Regression, Ridge, Lasso
* Dataset: Boston Housing (dự đoán giá nhà)

## **Chương 21: Binary Classification Machine Learning Case Study Project**

## Xây dựng mô hình phân loại nhị phân, ví dụ như phát hiện bệnh tiểu đường.

## Nội dung chính:

* Load dữ liệu: Dataset Pima Indians Diabetes (1 = có bệnh, 0 = không).
* Phân tích dữ liệu: Kiểm tra tỷ lệ class, phân tích thống kê, xác định mất cân bằng dữ liệu.
* Tiền xử lý: Chuẩn hóa dữ liệu và xử lý giá trị thiếu.
* Chọn mô hình: Các thuật toán phân loại nhị phân như: Logistic Regression , KNN
* Cross-validation: Dùng k-fold (thường k=10) để đánh giá độ ổn định mô hình.
* Dataset: Pima Indians Diabetes