ĐỒ ÁN CUỐI KỲ

Môn: Python cho Khoa học Dữ liệu - K23

1. Mục tiêu

Sinh viên vận dụng các kỹ năng Python để xây dựng một pipeline khoa học dữ liệu hoàn chỉnh gồm: tiền xử lý dữ liệu, huấn luyện mô hình học máy (có tối ưu tham số), và trực quan hóa kết quả. Dự án cần thể hiện khả năng viết mã có cấu trúc hướng đối tượng, tái sử dụng được và có tính tự động hóa.

2. Yêu cầu tổng quát

- Sinh viên làm việc theo nhóm (1–3 người).
- Dữ liệu có thể **tự thu thập** hoặc **do giảng viên cung cấp**.
- Dự án phải thể hiện được **tư duy khoa học dữ liệu hoàn chỉnh**, từ phân tích dữ liệu, tiền xử lý, mô hình hóa, đến diễn giải kết quả.
- Toàn bộ mã nguồn được viết bằng **Python**, có chú thích rõ ràng, đặt trong cấu trúc thư mục hợp lý, có file `README.md` hướng dẫn chạy.

3. Cấu trúc Pipeline

Pipeline của đồ án gồm 3 phần chính:

Phần 1: Tiền xử lý dữ liệu

- Viết một **lớp (class)** chuyên cho việc tiền xử lý dữ liệu của nhóm.
- Chức năng bao gồm: đọc dữ liệu, xử lý giá trị thiếu, mã hóa biến phân loại, chuẩn hóa dữ liệu, và xuất ra tập dữ liệu đã sẵn sàng cho huấn luyện.
- Lớp này phải có khả năng tái sử dụng cho các tập dữ liệu khác có cấu trúc tương tự.

Phần 2: Mô hình học máy

- Viết một **lớp (class)** chịu trách nhiệm toàn bộ quy trình từ:
  1. Nạp dữ liệu đã chuẩn hóa.
  2. Chia dữ liệu thành tập huấn luyện và kiểm định.
  3. Huấn luyện mô hình học máy (Regression hoặc Classification).
  4. Tối ưu siêu tham số (dùng thư viện như `GridSearchCV`, `Optuna`, hoặc tương đương).

5. Lưu kết quả huấn luyện, báo cáo độ chính xác, và lưu mô hình ra file (`pickle` hoặc `joblib`).

- Nên thử nghiệm ít nhất **2–3 mô hình** (ví dụ: Random Forest, XGBoost, CatBoost, SVM, Logistic Regression,...).
- Có phần so sánh giữa các mô hình dựa trên các chỉ số đánh giá phù hợp (RMSE, MAE, Accuracy, F1-score,...).

Phần 3: Trực quan hóa và phân tích

- Thực hiện trực quan dữ liệu đầu vào và kết quả mô hình bằng `matplotlib`, `seaborn`, hoặc `plotly`.
- Phân tích, nhận xét, và trình bày ý nghĩa kết quả theo hướng **giải thích được mô hình**.
- Có ít nhất một phần thể hiện việc **phân tích đặc trưng quan trọng** (Feature Importance, SHAP, hoặc Partial Dependence Plot).

4. Chi tiết kỹ thuật Python bắt buộc thể hiện

1. Tiền xử lý dữ liệu (20%)

Phần này sinh viên phải thể hiện khả năng sử dụng Python và các thư viện dữ liệu phổ biến để làm sạch, chuẩn hóa và trích chọn thông tin. Cần xây dựng **một lớp Python** (ví dụ: `DataPreprocessor`) có các chức năng sau:

- Đọc dữ liệu từ các định dạng khác nhau (`csv`, `xlsx`, `json`) bằng `pandas`.
- Kiểm tra và xử lý dữ liệu bị thiếu, dữ liệu ngoại lai bằng các kỹ thuật:
  - Điền giá trị thiếu (`fillna`, trung bình, trung vị, mode, forward-fill).
  - Phát hiện ngoại lai bằng IQR, z-score hoặc Isolation Forest.
- Chuẩn hóa dữ liệu bằng `StandardScaler`, `MinMaxScaler` hoặc tự cài đặt.
- Mã hóa biến phân loại bằng:
  - `OneHotEncoder` hoặc `LabelEncoder`.
  - Hàm tự định nghĩa chuyển đổi đặc trưng dạng text sang dạng số.
- tạo đặc trưng mới. Ví dụ Xử lý ngày giờ, tạo đặc trưng mới từ thời gian (datetime feature engineering).
- Tự động phát hiện kiểu dữ liệu và áp dụng phương pháp xử lý phù hợp.
- Ghi dữ liệu sau khi xử lý ra file mới hoặc trả về DataFrame đã sẵn sàng cho mô hình.
- Sử dụng kỹ thuật:

  - `@staticmethod` hoặc `@classmethod` cho các hàm tiện ích.
  - `__init__`, `__repr__` để định nghĩa đối tượng rõ ràng.
  - Xử lý lỗi (exception handling) để tránh lỗi khi đọc hoặc xử lý dữ liệu.

2. Thiết kế lớp mô hình học máy có tối ưu tham số, lưu mô hình (25%)

Sinh viên cần xây dựng một **lớp Python** (ví dụ: `ModelTrainer`) thể hiện khả năng sử dụng các thư viện học máy (`scikit-learn`, `xgboost`, `catboost`, `lightgbm`) kết hợp kỹ thuật tối ưu và quản lý mô hình. Cụ thể:

- Thiết kế class có cấu trúc rõ ràng gồm các phương thức:
  - `load_data()` — nạp dữ liệu đã chuẩn hóa.
  - `split_data()` — chia train/test bằng `train_test_split`.
  - `train_model()` — huấn luyện mô hình.
  - `optimize_params()` — tối ưu siêu tham số.
  - `evaluate()` — đánh giá mô hình theo thước đo phù hợp.
  - `save_model()` và `load_model()` — lưu và nạp mô hình bằng `joblib` hoặc `pickle`.
- Áp dụng các kỹ thuật Python:
  - Dùng decorator `@staticmethod`, `@property` khi cần.
  - Dùng `argparse` hoặc `configparser` để truyền tham số linh hoạt khi chạy script.
  - Ghi log quá trình huấn luyện (`logging`) và lưu lại kết quả thí nghiệm.
- Tối ưu tham số mô hình bằng ít nhất một trong các phương pháp:
  - `GridSearchCV`, `RandomizedSearchCV` (scikit-learn).
  - `Optuna` hoặc `Hyperopt` (nâng cao).
- Ghi lại các kết quả thực nghiệm (accuracy, F1-score, RMSE, v.v.) vào file CSV hoặc JSON.
- So sánh kết quả giữa các mô hình và lưu biểu đồ đánh giá (barplot, confusion matrix, ROC curve).

Điểm cộng:

- Có cơ chế **tự động thử nhiều mô hình** và chọn mô hình tốt nhất.
- Có tính năng **reproducibility** bằng cách cố định random seed.
- Có docstring chuẩn và hướng dẫn sử dụng trong mỗi class.

5. Rubric tóm tắt

Tiêu chí	**Trọng số**
Lựa chọn và mô tả dữ liệu hợp lý, có ý nghĩa thực tế	10%
Xây dựng lớp tiền xử lý (tự động hoá, tái sử dụng được, thể hiện kỹ năng Python)	20%
Thiết kế lớp mô hình học máy (tối ưu tham số, lưu mô hình, có log và kết quả)	25%
Trực quan, phân tích và đánh giá kết quả	20%
Tổ chức mã nguồn, tài liệu, báo cáo	15%
Sáng tạo, cải tiến kỹ thuật hoặc phân tích sâu	10%
Tổng cộng	**100%**

6. Sản phẩm nộp

- Thư mục mã nguồn Python (theo cấu trúc module).
- File `README.md` hướng dẫn cài đặt và chạy.
- Báo cáo định dạng PDF (5–10 trang) trình bày:
  - Giới thiệu bài toán và dữ liệu.
  - Giới thiệu về các phương thức và công dụng của các phương thức trong các Class.
  - Giới thiệu tổng quan về thư viện chứa mô hình máy học và thư viện tối ưu tham số.
  - Phân tích kết quả chạy thí nghiệm và có kèm theo hình ảnh minh hoạ.

7. Gợi ý đề tài

- Dự đoán giá nhà, giá thuê căn hộ.
- Phân loại bệnh dựa trên dữ liệu y tế.
- Dự đoán khả năng vỡ nợ ngân hàng.
- Phân loại cảm xúc bình luận mạng xã hội.
- Dự đoán lượng tiêu thụ điện/năng lượng.

Lưu ý: Sinh viên cần trình bày rõ vai trò của từng thành viên trong nhóm trong báo cáo cuối cùng.