A red and black logo

Description automatically generated

TP. Hồ Chí Minh – năm 2025

TRƯỜNG ĐẠI HỌC VĂN LANG

**KHOA CÔNG NGHỆ THÔNG TIN**

Nhập Môn Phân Tích Dữ Liệu Và Học Sâu

KHOA: CÔNG NGHỆ THÔNG TIN

***Đề tài:***

**Báo cáo về Mô hình Dự đoán Thuốc bằng Mạng Nơ-ron**

**SVTH: Nguyễn Thành Đạt-207ct68621**

**GVHD: Huỳnh Thái Học**

**LHP:**

**Mô hình Dự đoán Thuốc bằng Mạng Nơ-ron**

**Giới thiệu**

Mô hình này được xây dựng để dự đoán loại thuốc phù hợp cho bệnh nhân dựa trên các đặc điểm như tuổi, giới tính, huyết áp, cholesterol và tỷ lệ Na/K trong máu. Mô hình sử dụng kiến trúc mạng nơ-ron nhiều lớp (MLP) với framework PyTorch, được huấn luyện trên tập dữ liệu gồm 200 mẫu và kiểm thêm trên tập dữ liệu mới.

**Dữ liệu**: <https://www.kaggle.com/datasets/jeevanrh/drug200csv?resource=download>

**Mô tả dữ liệu**:

Các biến (cột) trong tập dữ liệu:

* Age (Tuổi): Tuổi của bệnh nhân (từ 15 đến 74 tuổi)
* Sex (Giới tính): Giới tính của bệnh nhân (M - Nam, F - Nữ)
* BP (Huyết áp): Mức huyết áp (LOW - Thấp, NORMAL - Bình thường, HIGH - Cao)
* Cholesterol (Cholesterol): Mức cholesterol (NORMAL - Bình thường, HIGH - Cao)
* Na\_to\_K (Tỉ lệ Natri/Kali): Tỉ lệ Natri đến Kali trong máu (giá trị liên tục từ 6.269 đến 38.247)
* Drug (Thuốc): Loại thuốc được kê đơn (drugA, drugB, drugC, drugX, drugY)

**Xử lý dữ liệu**

* Loại bỏ giá trị thiếu (nếu có).
* Mã hóa các biến phân loại (Sex, BP, Cholesterol) bằng one-hot encoding.
* Xử lý outliers bằng phương pháp IQR cho các cột số (Age, Na\_to\_K).
* Chuẩn hóa dữ liệu bằng StandardScaler.

**Kiến trúc Mô hình**

Mô hình mạng nơ-ron gồm:

* Lớp đầu vào: Kích thước bằng số đặc trưng.
* 3 lớp ẩn: 128, 64 và 32 nơ-ron với hàm kích hoạt ReLU.
* Dropout (0.5) để tránh overfitting.
* Lớp đầu ra: Số nơ-ron bằng số lớp thuốc cần phân loại.
* Hàm loss: CrossEntropyLoss.
* Tối ưu hóa: Adam với learning rate 0.001 và L2 regularization.

**Huấn luyện Mô hình**

* **Optimizer**: Adam với learning rate 0.001 và L2 regularization
* **Hàm mất mát**: CrossEntropyLoss
* **Số epoch**: 50
* **Kích thước batch**: 32
* **Phân chia dữ liệu**: 80% huấn luyện, 20% kiểm tra

**Lưu và Tải Mô hình**

Sau khi huấn luyện, mô hình được lưu vào tệp drug\_model.pth bằng lệnh:

* **torch.save(model.state\_dict(), 'drug\_model.pth')**

Khi chạy lại, mô hình được tải bằng:

* **model.load\_state\_dict(torch.load('drug\_model.pth', weights\_only=True))**

Việc này giúp tái sử dụng mô hình mà không cần huấn luyện lại từ đầu.

**Kết quả Đánh giá**

Mô hình đạt được:

* Độ chính xác huấn luyện: ~80-90%
* Loss giảm ổn định qua các epoch
* Báo cáo classification trên tập kiểm tra cho thấy hiệu suất tốt trên hầu hết các lớp thuốc

**Ưu điểm**

1. Xử lý đa dạng dữ liệu: Số liệu (Age, Na\_to\_K) và phân loại (Sex, BP, Cholesterol).
2. Chống overfitting: Sử dụng Dropout và L2 regularization.
3. Tương thích GPU: Tốc độ tính toán nhanh hơn khi có CUDA.
4. Dễ triển khai: Lưu và tải mô hình đơn giản với PyTorch.

**Hạn chế**

1. Kích thước tập dữ liệu nhỏ (200 mẫu)
2. Chưa tối ưu siêu tham số
3. Chưa xử lý cân bằng dữ liệu nếu các lớp không đồng đều

**Ứng dụng Tiềm năng**

Mô hình có thể được tích hợp vào hệ thống hỗ trợ quyết định trong y tế, giúp bác sĩ đề xuất loại thuốc phù hợp dựa trên đặc điểm bệnh nhân.

**Hướng Phát triển**

1. Thu thập thêm dữ liệu: Cải thiện độ chính xác.
2. Thử nghiệm mô hình khác: CNN hoặc Transformer nếu có dữ liệu hình ảnh/bệnh án.
3. Triển khai web/mobile: Xây dựng giao diện người dùng đơn giản.
4. Giải thích mô hình (XAI): Sử dụng SHAP/LIME để hiểu các yếu tố ảnh hưởng đến dự đoán.

**Kết luận**

Mô hình đã chứng minh hiệu quả trong việc phân loại thuốc dựa trên đặc điểm bệnh nhân. Với những cải tiến tiếp theo, đây có thể trở thành công cụ hữu ích trong lĩnh vực y tế.