DỰ ÁN KHAI PHÁ DỮ LIỆU: PHÂN TÍCH & DỰ ĐOÁN KẾT QUẢ THI BẰNG LÁI XE

Đây là đồ án cá nhân cho học phần Khai phá Dữ liệu (KPDL) tại Trường Đại học Kinh tế, Đại học Huế.

Dự án này thực hiện quy trình khai phá dữ liệu toàn diện trên bộ dữ liệu "Driver's License Test Scores Data" (500 quan sát, 17 thuộc tính) với hai mục tiêu chính:

1. Phân tích các yếu tố ảnh hưởng đến khả năng vượt qua kỳ thi (Qualified).
2. Xây dựng và đánh giá các mô hình học máy để dự đoán khả năng đỗ/trượt của học viên.

Xem báo cáo đầy đủ (190 trang) tại đây: DoAn_KhaiPhaDuLieu.pdf

1. Quy trình Tiền xử lý Dữ liệu (Data Preprocessing)

Một quy trình tiền xử lý chi tiết đã được thực hiện để đảm bảo chất lượng dữ liệu đầu vào cho mô hình.

1.1. Làm sạch Dữ liệu (Data Cleaning)

- **Xử lý Giá trị thiếu (Missing Values):** Phát hiện 150 giá trị thiếu (chiếm 30%) trong cột Training.
    - **Giải pháp:** Gán các giá trị thiếu này bằng một nhãn mới là "Unknown" để giữ nguyên thông tin và cho phép mô hình học được từ các trường hợp "không có dữ liệu đào tạo".
- **Xử lý Dữ liệu nhiễu (Outliers):**
    - Sử dụng phương pháp **IQR (Interquartile Range)** để phát hiện các giá trị ngoại lai trong các biến số.
    - **Quyết định phân tích then chốt:** Dự án đã tiến hành phân tích song song trên **cả hai bộ dữ liệu** (Dữ liệu gốc và Dữ liệu đã loại bỏ outliers) để so sánh và đánh giá tác động của outliers đến hiệu suất mô hình .

1.2. Chuyển đổi Dữ liệu (Data Transformation)

- **Mã hóa Biến phân loại (Encoding):**

- Label Encoding (Thư viện sklearn): Cho các biến nhị phân (Gender, Qualified).
- Ordinal Encoding: Cho các biến có thứ bậc (Age Group, Training, Reactions).
- One-Hot Encoding: Cho biến không có thứ bậc (Race).

- **Chuẩn hóa Dữ liệu (Normalization):**
  - RobustScaler: Được sử dụng cho bộ dữ liệu *còn chứa outliers* vì phương pháp này sử dụng trung vị (median) và IQR, giúp giảm ảnh hưởng của các giá trị ngoại lai.
  - MinMaxScaler: Được sử dụng cho bộ dữ liệu *đã làm sạch outliers* để đưa tất cả các biến về khoảng [0, 1].

2. Học máy có giám sát (Supervised Learning) - Dự đoán "Qualified"

2.1. Thử nghiệm Nhanh với LazyPredict

- Sử dụng thư viện LazyPredict để nhanh chóng huấn luyện và so sánh hiệu suất của hơn 20 thuật toán phân loại trên cả hai bộ dữ liệu.
- **Phát hiện quan trọng:** Việc loại bỏ outliers đã cải thiện đáng kể độ chính xác. Ví dụ, mô hình KNeighborsClassifier tăng Accuracy từ **77% (trước) lên 84% (sau)**.

2.2. Phân tích Chuyên sâu 3 Mô hình Tốt nhất

Dựa trên kết quả từ LazyPredict, 3 mô hình hàng đầu đã được chọn để phân tích sâu[15151515]:

1. **RandomForestClassifier**
2. **ExtraTreesClassifier**
3. **AdaBoostClassifier**

Với mỗi mô hình, dự án đã:

- Huấn luyện và đánh giá trên cả 2 bộ dữ liệu (trước và sau khi xử lý outliers).
- So sánh chi tiết các chỉ số: **Accuracy, Precision, Recall, F1-Score, và ROC AUC**.
- Trực quan hóa kết quả bằng **Confusion Matrix** (Ma trận nhầm lẫn) và **ROC Curve** (Đường cong ROC).

Kết luận (Ví dụ): Đối với ExtraTreesClassifier, mô hình sau khi loại bỏ outliers được đánh giá là tối ưu hơn, vì chỉ số **Recall** (khả năng phát hiện các trường hợp "Đỗ") tăng vọt từ **0.9250** lên **0.9750**.

3. Học máy không giám sát (Unsupervised Learning) - Phân cụm Học viên

Dự án cũng khám phá cấu trúc tiềm ẩn của học viên bằng cách sử dụng 3 thuật toán phân cụm:

1. **K-Means** (Sử dụng phương pháp Elbow để tìm k=3)
2. **DBSCAN**
3. **Gaussian Mixture Model (GMM)**

Các mô hình được đánh giá bằng các chỉ số **Silhouette, Davies-Bouldin, và Calinski-Harabasz**.

Kết luận: Thuật toán **GMM cho thấy hiệu suất phân cụm vượt trội** trên bộ dữ liệu này, đặc biệt là sau khi đã loại bỏ outliers.

4. Công cụ & Thư viện

- **Ngôn ngữ:** Python
- **Xử lý dữ liệu:** Pandas, NumPy
- **Học máy:** Scikit-learn
- **Kiểm thử nhanh:** LazyPredict
- **Trực quan hóa:** Matplotlib, Seaborn

5. Tác giả

- **Trần Hoàng Phương Dung**
- **Lớp:** K57 - Kinh Tế Số
- **GVHD:** TS. Hoàng Hữu Trung