

**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI
PHÂN HIỆU TẠI TP. HỒ CHÍ MINH
BỘ MÔN CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO BÀI TẬP LỚN
Học phần: KHAI PHÁ DỮ LIỆU**

**Đề tài: NGHIÊN CỨU VÀ XÂY DỰNG
CÔNG CỤ DỰ ĐOÁN MỨC ĐỘ PHÙ HỢP CỦA XE
DỰA TRÊN ĐẶC ĐIỂM KỸ THUẬT**

Giảng viên hướng dẫn: ThS. NGUYỄN THIỆN DƯƠNG

Sinh viên thực hiện: PHẠM THỊ NGỌC OANH

TRẦN PHƯƠNG ANH

LÊ ĐÌNH KHÔI

ĐỖ VĂN THÀNH ĐƯỢC

ÔN GIA BẢO

Lớp : CQ.63.CNTT

Khoá : K63

TP. Hồ Chí Minh, tháng 12 năm 2025

TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI

PHÂN HIỆU TẠI TP. HỒ CHÍ MINH

BỘ MÔN CÔNG NGHỆ THÔNG TIN



BÁO CÁO BÀI TẬP LỚN

Học phần: KHAI PHÁ DỮ LIỆU

Đề tài: NGHIÊN CỨU VÀ XÂY DỰNG

CÔNG CỤ DỰ ĐOÁN MỨC ĐỘ PHÙ HỢP CỦA XE

DỰA TRÊN ĐẶC ĐIỂM KỸ THUẬT

Giảng viên hướng dẫn: ThS. NGUYỄN THIỆN DƯƠNG

Sinh viên thực hiện: PHẠM THỊ NGỌC OANH

TRẦN PHƯƠNG ANH

LÊ ĐÌNH KHÔI

ĐỖ VĂN THÀNH ĐƯỢC

ÔN GIA BẢO

Lớp : CQ.63.CNTT

Khoa : K63

TP. Hồ Chí Minh, tháng 12 năm 2025

NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

Tp. Hồ Chí Minh, ngày tháng năm

Giảng viên hướng dẫn

Nguyễn Thiện Dương

LỜI CẢM ƠN

Để hoàn thành bài tập lớn này trước hết chúng em xin gửi đến quý thầy, cô **Bộ môn Công nghệ thông tin – Phân hiệu Trường Đại học Giao thông Vận tải tại Thành phố Hồ Chí Minh** lời cảm ơn chân thành vì đã truyền đạt cho chúng em những kiến thức không chỉ từ sách vở, mà còn những kinh nghiệm quý giá từ cuộc sống trong khoảng thời gian học tập tại trường. Đặc biệt chúng em xin gửi đến **thầy Nguyễn Thị Hiền Dương** lời cảm ơn sâu sắc nhất vì thầy đã tận tình hướng dẫn, chỉ bảo chúng em trong suốt quá trình thực hiện đề tài. Tuy đề tài không được lớn nhưng nếu không được sự hướng dẫn chỉ bảo tận tình của thầy thì đề tài bài tập lớn này khó có thể hoàn thành được.

Vì thời gian làm bài tập lớn có hạn cũng như hiểu biết của nhóm còn hạn chế, chúng em cũng đã nỗ lực hết sức để hoàn thành bài tập lớn một cách tốt nhất, nhưng chắc chắn vẫn sẽ có những thiếu sót không thể tránh khỏi. Chúng em kính mong nhận được sự thông cảm và những ý kiến đóng góp chân thành từ quý thầy cô.

Sau cùng, em xin kính chúc quý thầy cô trong **Bộ môn Công nghệ thông tin** luôn mạnh khoẻ, hạnh phúc và thành công hơn nữa trong công việc cũng như trong cuộc sống. Chúng em xin chân thành cảm ơn!

Tp. Hồ Chí Minh, ngày 13 tháng 12 năm 2025

Phạm Thị Ngọc Oanh

Trần Phương Anh

Lê Đình Khôi

Đỗ Văn Thành Được

Ôn Gia Bảo

NHIỆM VỤ THIẾT KẾ BÀI TẬP LỚN

BỘ MÔN: CÔNG NGHỆ THÔNG TIN

-----***-----

1. Tên đề tài

Nghiên cứu và xây dựng công cụ dự đoán mức độ phù hợp của xe dựa trên đặc điểm kỹ thuật.

2. Mục đích thực hiện

Mục đích của đề tài là nghiên cứu và xây dựng một mô hình có khả năng dự đoán mức độ đánh giá của xe dựa trên các thuộc tính kỹ thuật như chi phí mua, chi phí bảo trì, số cửa, số chỗ ngồi, kích thước khoang chứa đồ và mức độ an toàn. Về mặt lý thuyết, đề tài nhằm tìm hiểu các thuật toán học máy phù hợp cho bài toán phân loại, khảo sát các phương pháp tiền xử lý, mã hóa dữ liệu và đánh giá mô hình. Về mặt thực nghiệm, đề tài hướng tới việc áp dụng các kiến thức đã nghiên cứu để xây dựng một hệ thống thực tế, có thể tiếp nhận thuộc tính của xe và tự động đưa ra mức đánh giá, từ đó hỗ trợ người dùng hoặc doanh nghiệp trong việc lựa chọn và phân tích sản phẩm.

3. Mục tiêu thực hiện

Để đạt được mục đích nghiên cứu, đề tài đặt ra các mục tiêu cụ thể. Trước hết, tiến hành thu thập và xử lý bộ dữ liệu đánh giá xe, kiểm tra giá trị thiếu, loại bỏ bản ghi trùng và mã hóa toàn bộ dữ liệu dạng phân loại bằng phương pháp Ordinal Encoding. Sau đó, đề tài triển khai tiền xử lý dữ liệu, chuẩn hóa dữ liệu và chia thành tập huấn luyện – kiểm thử để đảm bảo mô hình được học một cách ổn định.

Tiếp theo, đề tài tiến hành xây dựng và huấn luyện bốn thuật toán gồm Decision Tree, Random Forest, Gradient Boosting và Support Vector Machine. Mỗi thuật toán được đánh giá bằng các thước đo Accuracy, F1-Score, Precision và Recall, đồng thời trực quan hóa bằng ma trận nhầm lẫn, biểu đồ mức độ quan trọng đặc trưng và biểu đồ so sánh tổng quan. Sau khi so sánh, đề tài xác định được mô hình có hiệu suất cao nhất. Cuối cùng, mô hình tốt nhất được lưu lại và tích hợp vào một demo nhỏ, cho phép nhập

vào các thuộc tính của xe và trả về kết quả dự đoán như unacc, acc, good hoặc vgood, giúp minh chứng khả năng ứng dụng thực tế của hệ thống.

4. Nội dung và phạm vi đề tài

Đề tài tập trung giải quyết bài toán dự đoán mức độ đánh giá của xe dựa trên các thuộc tính kỹ thuật do người dùng cung cấp. Đây là một bài toán trong lĩnh vực học máy – phân loại (classification), hướng tới xây dựng hệ thống có khả năng hỗ trợ đánh giá chất lượng và mức độ phù hợp của xe một cách tự động.

Phạm vi đề tài bao gồm việc sử dụng bộ dữ liệu Car Evaluation chứa các thuộc tính như chi phí mua xe (buying), chi phí bảo trì (maint), số cửa (doors), số chỗ (persons), kích thước cốp xe (lug_boot) và mức độ an toàn (safety). Đầu ra của bài toán là nhãn phân loại đánh giá mức độ chấp nhận của xe:

- unacc (không chấp nhận),
- acc (chấp nhận),
- good (tốt),
- vgood (rất tốt).

Input của bài toán: Các thuộc tính kỹ thuật của xe (6 thuộc tính dạng phân loại).

Output của bài toán: Mức đánh giá xe dự đoán bởi mô hình máy học (unacc/acc/good/vgood).

Nội dung chính của đề tài gồm:

1. Khảo sát và phân tích bộ dữ liệu đánh giá xe.
2. Tiền xử lý dữ liệu: làm sạch, loại bỏ trùng lặp, mã hóa các thuộc tính dạng phân loại bằng Ordinal Encoding, chuẩn hóa dữ liệu.
3. Chia dữ liệu thành tập huấn luyện và kiểm thử.
4. Huấn luyện và đánh giá bốn thuật toán: Decision Tree, Random Forest, Gradient Boosting, Support Vector Machine (SVM).
5. So sánh mô hình bằng các thước đo: Accuracy, F1-Score, Precision, Recall và ma trận nhầm lẫn.

6. Lưu mô hình có kết quả tốt và xây dựng demo nhỏ nhập thuộc tính xe và trả về mức đánh giá dự đoán.

7. Tổng hợp kết luận và đề xuất hướng phát triển.

5. Phương pháp thực hiện

Đề tài áp dụng các công nghệ, nền tảng và mô hình như sau:

- Ngôn ngữ lập trình: Python
- Nền tảng: Google Colab
- Các thư viện sử dụng:
 - + Xử lý dữ liệu: Pandas, Numpy
 - + Tiền xử lý & Encoding: OrdinalEncoder, StandardScaler
 - + Mô hình học máy: scikit-learn (DecisionTreeClassifier, RandomForestClassifier, GradientBoostingClassifier, SVC)
 - + Trực quan hóa: Matplotlib, Seaborn
 - + Giải thích mô hình: SHAP
 - + Lưu mô hình: joblib

Phương pháp giải quyết:.

- Áp dụng quy trình học máy tiêu chuẩn: tiền xử lý → huấn luyện → đánh giá → tối ưu → triển khai
- Sử dụng nhiều thuật toán để so sánh hiệu năng và lựa chọn mô hình tối ưu.
- Đánh giá mô hình bằng cả thước đo tổng quan và chi tiết để đảm bảo tính ổn định.
- Xây dựng demo bằng cách tái sử dụng mô hình đã lưu và pipeline encoding-scaling tương tự như giai đoạn huấn luyện.

6. Các kết quả chính dự kiến sẽ đạt được và ứng dụng

Thông qua quá trình thực hiện, đề tài dự kiến đạt được một mô hình học máy có khả năng dự đoán mức độ đánh giá xe với độ chính xác cao. Người thực hiện kỳ vọng đánh giá được hiệu năng của bốn thuật toán khác nhau và xác định được mô hình phù hợp nhất cho bài toán. Bên cạnh đó, đề tài sẽ xây dựng được quy trình xử lý dữ liệu hoàn

chỉnh, từ làm sạch đến mã hóa và chuẩn hóa, cũng như tạo ra một demo minh họa tính ứng dụng thực tế của hệ thống dự đoán.

Về ứng dụng, mô hình có thể hỗ trợ người dùng trong việc tham khảo và lựa chọn mẫu xe phù hợp, giúp các showroom hoặc cửa hàng xe có thêm công cụ đánh giá nhanh. Hệ thống này cũng có thể được phát triển thêm để trở thành một phần của hệ thống tư vấn và gợi ý sản phẩm trong tương lai.

7. Kế Hoạch Thực Hiện:

Bảng 0.1 Bảng kế hoạch thực hiện

Thời gian	Công việc thực hiện	Người phụ trách
Tuần 1	Tìm hiểu đề tài, đặt vấn đề, thu thập tài liệu.	Phạm Thị Ngọc Oanh
Tuần 2, 3	Phân tích bộ và tiền xử lý dữ liệu	Trần Phương Anh
Tuần 4,5	Huấn luyện 2 mô hình đầu (Decision Tree, Random Forest)	Đỗ Văn Thành Được
Tuần 4,5	Huấn luyện Gradient Boosting và SVM, điều chỉnh tham số	Lê Đình Khôi
Tuần 6	Đánh giá mô hình, trực quan hóa kết quả, so sánh kết quả	Đỗ Văn Thành Được Lê Đình Khôi
Tuần 7	Xây dựng giao diện và tích hợp api dự đoán mức độ phù hợp của xe.	Ôn Gia Bảo Phạm Thị Ngọc Oanh
Tuần 8	Viết báo cáo, xây dựng kết quả đánh giá, hoàn thiện nội dung	Tất cả thành viên

8. Giảng viên hướng dẫn

Họ và tên: ThS. Nguyễn Thiện Dương

Đơn vị công tác: Phân hiệu Trường Đại học Giao thông Vận tải tại TPHCM

Điện thoại:

Email:

LỜI MỞ ĐẦU

Trong những năm gần đây, sự phát triển nhanh chóng của ngành công nghiệp ô tô đã kéo theo nhu cầu ngày càng cao về các công cụ hỗ trợ đánh giá và lựa chọn xe dựa trên các đặc tính kỹ thuật. Người tiêu dùng ngày càng quan tâm đến việc lựa chọn một mẫu xe phù hợp không chỉ về chi phí mà còn về độ an toàn, độ tiện dụng và chất lượng tổng thể. Tuy nhiên, việc đánh giá một mẫu xe thường phụ thuộc vào kinh nghiệm hoặc cảm tính cá nhân, dẫn đến thiếu tính khách quan và có thể gây khó khăn cho những người không am hiểu về kỹ thuật ô tô.

Trước bối cảnh đó, các phương pháp học máy (Machine Learning) đã trở thành công cụ hiệu quả hỗ trợ phân tích dữ liệu và đưa ra dự đoán dựa trên các mô hình thống kê. Học máy cho phép xử lý lượng lớn dữ liệu, phát hiện quy luật tiềm ẩn và đưa ra kết luận mang tính khoa học, khách quan. Các thuật toán phân loại cổ điển như Decision Tree, Random Forest, Boosting và Support Vector Machine đã được trình bày và phân tích kỹ lưỡng trong nhiều tài liệu nền tảng về học máy, trong đó có The Elements of Statistical Learning của Hastie, Tibshirani và Friedman (2009), cho thấy chúng có khả năng mô hình hóa hiệu quả các bài toán phân loại có cấu trúc thuộc tính rõ ràng.

Trong đề tài này, nhóm tập trung nghiên cứu và xây dựng một công cụ dự đoán mức độ đánh giá xe dựa trên bộ dữ liệu Car Evaluation từ UCI Machine Learning Repository (Bohanec & Rajkovič, 1990). Bộ dữ liệu này mô tả các thuộc tính đặc trưng của xe như chi phí mua, chi phí bảo trì, số lượng cửa, số chỗ ngồi, kích thước khoang chứa đồ và mức độ an toàn. Dựa trên các thông tin này, mô hình sẽ dự đoán mức đánh giá tổng quan của xe gồm bốn mức: unacc, acc, good và vgood.

Việc triển khai mô hình không chỉ giúp chúng em hiểu rõ hơn quy trình xây dựng một hệ thống học máy hoàn chỉnh — từ thu thập và tiền xử lý dữ liệu đến huấn luyện, đánh giá mô hình và triển khai thử nghiệm — mà còn thể hiện khả năng ứng dụng thực tế của học máy trong các lĩnh vực liên quan đến phân tích sản phẩm. Ngoài ra, công cụ dự đoán được xây dựng có thể trở thành nguồn tham khảo hữu ích cho người dùng có nhu cầu đánh giá sơ bộ chất lượng xe dựa trên các yếu tố kỹ thuật.

Với mục tiêu kết hợp kiến thức lý thuyết và thực nghiệm, đề tài kỳ vọng mang lại một mô hình dự đoán có độ chính xác cao, quy trình thực hiện rõ ràng và một ứng dụng minh họa giúp người dùng nhìn thấy tiềm năng ứng dụng của học máy trong đời sống thực tiễn.

MỤC LỤC

NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN	i
LỜI CẢM ƠN	ii
NHIỆM VỤ THIẾT KẾ BÀI TẬP LỚN.....	iii
LỜI MỞ ĐẦU	vii
MỤC LỤC	viii
DANH MỤC HÌNH ẢNH	x
DANH MỤC BẢNG BIỂU	xi
DANH MỤC VIẾT TẮT	xii
CHƯƠNG 1: LÝ DO CHỌN DATASET VÀ GIỚI THIỆU TỔNG QUAN DATASET	1
1.1 Giới thiệu tổng quan dataset	1
1.1.1 Nguồn dữ liệu sử dụng	1
1.1.2 Mô tả chi tiết dữ liệu	2
1.1.3 Mô tả mục đích bài toán	2
1.1.4 Tiền xử lý dữ liệu.....	2
1.1.5 Mô tả chi tiết các thuộc tính trong dataset.....	8
1.1.6 Giới thiệu các công cụ được sử dụng trong đồ án	10
CHƯƠNG 2: THUẬT TOÁN KHAI THÁC DỮ LIỆU SỬ DỤNG.....	13
2.1. Sử dụng thuật toán phân lớp dựa trên cây quyết định	13
2.1.1. Tổng quan về thuật toán phân lớp dựa trên cây quyết định	13
2.1.2. Lý do lựa chọn thuật toán	13
2.1.3. Quá trình thực hiện.....	15
2.1.4. Kết quả thu được	17
2.2 Thuật toán Random Forest	17
2.2.1. Tổng quan về thuật toán Random Forest.....	17
2.2.2. Lý do chọn thuật toán	17
2.2.3. Quá trình thực hiện.....	18
2.2.4. Kết quả thu được	22
2.3. Sử dụng thuật toán Gradient Boosting	22
2.3.1. Tổng quan về thuật toán	22
2.3.2. Lý do chọn thuật toán	23

2.3.3. Quá trình thực hiện.....	24
2.3.4. Kết quả thu được	24
2.4. Sử dụng thuật toán Support Vector Machine (SVM)	25
2.4.1. Tổng quan về thuật toán	25
2.4.2. Lý do chọn thuật toán	26
2.4.3. Quá trình thực hiện.....	27
2.4.4. Kết quả thu được	27
2.5. So sánh đánh giá.....	29
CHƯƠNG 3: CÔNG CỤ DỰ ĐOÁN MỨC ĐỘ PHÙ HỢP CỦA XE DỰA TRÊN ĐẶC ĐIỂM KỸ THUẬT	31
3.1. Mục tiêu và vai trò của ứng dụng demo	31
3.2. Quy trình xử lý đầu vào của demo	31
3.3. Kiến trúc triển khai	31
3.4. Mô tả hình minh họa giao diện.....	32
3.5. Tính năng Nâng cao: Phân tích Đề xuất (Prescriptive Analysis)	35
3.6. Những hạn chế.....	37
CHƯƠNG 4: KẾT LUẬN.....	38
4.1. Kết quả đạt được.....	38
4.2. Những hạn chế.....	38
4.3. Hướng phát triển trong tương lai	40
4.4. Bảng phân công nhiệm vụ trong nhóm	40
TÀI LIỆU THAM KHẢO.....	43

DANH MỤC HÌNH ẢNH

Hình 2.1 Ma trận nhầm lẫn của thuật toán Decision Tree.....	15
Hình 2.2 Biểu đồ Feature Importance của thuật toán Decision Tree	16
Hình 2.3 Biểu đồ Leaning Curves (Random Forest Classifier)	19
Hình 2.4 Biểu đồ Scalability (Random Forest Classifier).....	19
Hình 2.5 Biểu đồ Performance (Random Forest Classifier).....	20
Hình 2.6 Confusion Matrix của Random Forest trên Test-set.....	21
Hình 2.7 Biểu đồ Feature Importance của Random Forest	22
Hình 2.8 Confusion Matrix của thuật toán Gradient Boosting.....	24
Hình 2.9 Biểu đồ Feature Importance của thuật toán Gradient Boosting.....	25
Hình 2.10 Confusion Matrix của thuật toán SVM.....	28
Hình 2.11 Báo cáo chi tiết classification report.....	28
Hình 3.1 Màn hình Landing Page của ứng dụng Car Evaluation Predictor.....	32
Hình 3.2 Giao diện nhập thông tin và gợi ý câu hỏi của hệ thống	33
Hình 3.3 Giao diện ứng dụng demo dự đoán mức độ phù hợp của xe.....	34
Hình 3.4 Giao diện đánh giá và cải tiến để xuất cho nhà sản xuất	36

DANH MỤC BẢNG BIỂU

Bảng 0.1	Bảng kê hoạch thực hiện.....	vi
Bảng 2.1	Mô tả từng thuộc tính trong dataset.....	14
Bảng 2.1	Bảng đánh giá độ chính xác của thuật toán Decision Tree	16
Bảng 2.1	Bảng đánh giá độ chính xác của thuật toán Random Forest.....	21
Bảng 4.1	Bảng phân công nhiệm vụ.....	42

DANH MỤC VIẾT TẮT

Viết tắt	Giải thích
AI	Artificial Intelligence (Trí tuệ nhân tạo)
API	Application Programming Interface (Giao diện lập trình ứng dụng)
GPU	Graphics Processing Unit (Đơn vị xử lý đồ họa)
SVM	Support Vector Machine
UCI	University of California
DEX	Decision Expert
PCA	Principal Component Analysis
LDA	Linear Discriminant Analysis

CHƯƠNG 1: LÝ DO CHỌN DATASET VÀ GIỚI THIỆU TỔNG QUAN DATASET

1.1 Giới thiệu tổng quan dataset

1.1.1 Nguồn dữ liệu sử dụng

Bộ dữ liệu Đánh giá Xe hơi (Car Evaluation Dataset) là một tập dữ liệu phân loại đa biến kinh điển, có nguồn gốc từ Kho lưu trữ Học máy UCI (UCI Machine Learning Repository).¹ Bộ dữ liệu này cũng được phổ biến rộng rãi trên các nền tảng chia sẻ như Kaggle.³

Bộ dữ liệu này không được tạo ra từ việc thu thập dữ liệu thị trường trực tiếp, mà được dẫn xuất từ một mô hình quyết định phân cấp đơn giản, được gọi là DEX, do M. Bohanec và V. Rajkovic phát triển vào năm 1990.⁵ Mô hình DEX này được thiết kế nhằm mục đích đánh giá mức độ chấp nhận của xe hơi (CAR acceptability) dựa trên các tiêu chí cụ thể về kỹ thuật, chi phí và tiện nghi. Dữ liệu bao gồm 1728 phiên bản (instances) và được xác nhận là không có bất kỳ giá trị thiếu (missing values) nào.⁶

Về mặt kỹ thuật, mặc dù dữ liệu đã được tinh giản bằng cách loại bỏ thông tin cấu trúc trung gian, nó vẫn trực tiếp liên kết mức độ chấp nhận (CAR) với sáu thuộc tính đầu vào.⁵ Vì cấu trúc khái niệm phân cấp cơ bản đã được biết trước, bộ dữ liệu này đặc biệt hữu ích cho các chuyên gia học máy muốn kiểm tra các phương pháp quy nạp mang tính xây dựng (constructive induction) và khám phá cấu trúc (structure discovery) trong mô hình quyết định.¹

Hướng dẫn tải dataset thực hiện trong đồ án và các dataset khác của nhà cung cấp

Để tải bộ dữ liệu phục vụ cho đồ án, người dùng có thể truy cập trang UCI Machine Learning Repository và tìm kiếm khóa “Car Evaluation”. Trang cung cấp các tệp dữ liệu định dạng .data hoặc .csv, có thể tải trực tiếp và sử dụng trong môi trường Python hoặc các phần mềm phân tích dữ liệu.

Ngoài bộ dữ liệu này, UCI còn cung cấp nhiều dataset phổ biến cho các bài toán học máy như Iris, Adult Income, Wine Quality, Bank Marketing hay Breast Cancer, cho phép người học và nhà nghiên cứu mở rộng thực nghiệm sang nhiều dạng bài toán khác nhau cần.

1.1.2 Mô tả chi tiết dữ liệu

Bộ dữ liệu Car Evaluation gồm 1.728 mẫu, trong đó mỗi mẫu đại diện cho một cấu hình xe với 6 thuộc tính đầu vào và 1 thuộc tính đầu ra. Các thuộc tính đầu vào bao gồm: giá mua xe (buying), chi phí bảo trì (maint), số cửa (doors), số chỗ ngồi (persons), kích thước khoang chứa đồ (lug_boot) và mức độ an toàn (safety). Tất cả các thuộc tính đều ở dạng phân loại (categorical). Thuộc tính đầu ra là lớp đánh giá tổng quan của xe với bốn mức: unacc (không chấp nhận), acc (chấp nhận), good (tốt) và vgood (rất tốt). Dữ liệu không chứa giá trị bị khuyết, tuy nhiên số lượng mẫu giữa các lớp không đồng đều, trong đó lớp unacc chiếm tỷ trọng lớn nhất. Điều này đặt ra yêu cầu tiền xử lý và lựa chọn mô hình phù hợp để đảm bảo hiệu quả phân loại.

1.1.3 Mô tả mục đích bài toán

Mục đích của bài toán là xây dựng một mô hình học máy có khả năng dự đoán chính xác mức độ đánh giá của xe dựa trên các thuộc tính kỹ thuật đầu vào. Đây là bài toán phân loại đa lớp, yêu cầu mô hình xác định xem mỗi cấu hình xe thuộc nhóm unacc, acc, good hay vgood. Thông qua bài toán này, đề tài hướng đến việc phân tích tầm quan trọng của các đặc trưng, áp dụng nhiều thuật toán học máy khác nhau và đánh giá mức độ phù hợp của từng thuật toán. Ngoài việc thể hiện quy trình xây dựng mô hình từ dữ liệu thực tế, bài toán còn mang ý nghĩa ứng dụng trong các hệ thống tư vấn lựa chọn xe hoặc các nền tảng đánh giá sản phẩm tự động.

1.1.4 Tiền xử lý dữ liệu

Tiền xử lý dữ liệu (Data Preprocessing) là giai đoạn nền tảng của toàn bộ quy trình xây dựng mô hình học máy. Trong hầu hết các bài toán thực tế, dữ liệu thô thường chứa nhiều vấn đề như giá trị thiếu, sai lệch, dư thừa, định dạng không thống nhất hoặc phân bố mất cân bằng. Nếu không xử lý đúng cách ngay từ đầu, hiệu suất mô hình sẽ bị ảnh hưởng nghiêm trọng, dẫn đến sai số lớn hoặc khả năng khái quát kém.

Trong đề tài này, dữ liệu Car Evaluation là dữ liệu chuẩn hóa nhưng vẫn cần trải qua một chuỗi các bước xử lý hệ thống để đảm bảo tính sẵn sàng cho mô hình. Giai đoạn tiền xử lý được thực hiện theo một quy trình tuần tự, gồm các bước:

1. Tiếp nhận và đọc dữ liệu
2. Làm sạch dữ liệu

3. Tích hợp dữ liệu
4. Thu giảm và rút gọn dữ liệu
5. Chuẩn hóa và mã hóa dữ liệu
6. Chia dữ liệu thành tập huấn luyện và kiểm thử
7. Từng bước sẽ được mô tả chi tiết dưới đây.

1.1.4.1 *Làm sạch dữ liệu*

Làm sạch dữ liệu (Data Cleaning) là bước đầu tiên và giữ vai trò quyết định trong việc đảm bảo dữ liệu không chứa những yếu tố gây nhiễu hoặc sai lệch. Bước này được tiến hành theo ba nhóm công việc chính: kiểm tra giá trị thiếu, kiểm tra bản ghi trùng lặp và kiểm tra sự hợp lệ của từng giá trị trong các thuộc tính.

Đầu tiên, dữ liệu được kiểm tra để xác định xem có tồn tại giá trị thiếu hay không. Việc có các giá trị trống (null) sẽ làm gián đoạn quá trình mã hóa, khiến mô hình học sai và dẫn đến kết quả không ổn định. Để kiểm tra, hàm thống kê tổng số lượng giá trị thiếu theo từng cột được sử dụng. Sau khi kiểm tra toàn bộ 1.728 bản ghi, kết quả cho thấy không có bất kỳ giá trị thiếu nào trong toàn bộ file dữ liệu. Do đó, giai đoạn xử lý như điền giá trị trung bình, giá trị mode hay nội suy không cần thiết.

Tiếp theo, bộ dữ liệu được đánh giá khả năng tồn tại các bản ghi trùng lặp. Bản ghi bị trùng lặp có thể làm thay đổi phân bố dữ liệu, gây thiên lệch trong quá trình huấn luyện. Kết quả cho thấy không xuất hiện bản ghi trùng lặp, chứng tỏ dữ liệu ban đầu đã được xử lý tốt trước khi công bố.

Tiếp đến, kiểm tra tính hợp lệ của các giá trị. Mỗi thuộc tính của bộ dữ liệu chỉ được phép nhận giá trị trong một tập nhất định (ví dụ: buying ∈ {low, med, high, vhigh}). Vì vậy, từng cột được kiểm tra xem có giá trị bất thường, sai chính tả, hoặc sai định dạng hay không.

Dữ liệu thu được không xuất hiện bất kỳ giá trị không mong muốn nào; tất cả các giá trị đều nằm trong tập giá trị đã được định nghĩa ban đầu và được trình bày với định dạng thống nhất giữa các bản ghi. Do đó, có thể kết luận rằng dữ liệu hoàn toàn sạch và không cần thực hiện thêm các bước xử lý ở giai đoạn này.

1.1.4.2 Tích hợp dữ liệu

Tích hợp dữ liệu (Data Integration) là quá trình kết hợp nhiều nguồn dữ liệu khác nhau để tạo thành một bộ dữ liệu hoàn chỉnh. Tuy nhiên, trong bộ dữ liệu Car Evaluation, toàn bộ thông tin đã được chứa trong một tệp duy nhất, có cấu trúc chuẩn với 7 cột và 1.728 bản ghi. Mặc dù không phải thực hiện thao tác ghép nối dữ liệu, bước tích hợp vẫn được tiến hành dưới dạng:

Đầu tiên là kiểm tra nguồn dữ liệu. Xác định rằng dữ liệu được cung cấp từ nguồn chuẩn UCI Machine Learning Repository – một nguồn đáng tin cậy và thường xuyên được sử dụng trong nghiên cứu.

Tiếp theo là xác nhận độ thống nhất cấu trúc. Đảm bảo rằng toàn bộ dòng dữ liệu có chung cấu trúc, cùng số lượng cột, cùng định dạng thuộc tính và không có dòng lỗi.

Cuối cùng, việc kiểm tra tính nhất quán giữa các thuộc tính cho thấy rằng các giá trị đều được thể hiện đúng theo quy định. Chẳng hạn, thuộc tính doors luôn xuất hiện dưới dạng số hoặc số kèm từ, không bị pha trộn với các định dạng chữ không hợp lệ; tương tự, thuộc tính persons luôn nằm trong các giá trị được cho phép gồm 2, 4 hoặc more. Từ đó có thể kết luận rằng không cần thực hiện bước hợp nhất dữ liệu, và quá trình kiểm tra này góp phần đảm bảo tính toàn vẹn của dữ liệu đầu vào.

1.1.4.3 Thu giảm, rút gọn dữ liệu

Thu giảm dữ liệu (Data Reduction) là tập hợp các kỹ thuật nhằm giảm số lượng mẫu hoặc số lượng thuộc tính để mô hình hoạt động hiệu quả hơn mà vẫn giữ được thông tin cốt lõi. Trong đề tài này, bước thu giảm dữ liệu được thực hiện dưới dạng đánh giá thay vì áp dụng các phương pháp giảm chiều. Điều này xuất phát từ nhiều lý do: thứ nhất, dữ liệu chỉ có 6 thuộc tính đầu vào nên không cần sử dụng PCA, LDA hay bất kỳ kỹ thuật giảm chiều nào khác; thứ hai, mỗi thuộc tính đều mang ý nghĩa thực tế quan trọng, không có thuộc tính dư thừa hay ít ảnh hưởng, bởi toàn bộ các thuộc tính đều được thiết kế để đánh giá những khía cạnh khác nhau của chất lượng xe; thứ ba, số lượng mẫu chỉ gồm 1.728 bản ghi, vốn không phải là lớn nên không cần giảm để cải thiện tốc độ xử lý; và cuối cùng, dữ liệu dạng phân loại không xuất hiện hiện tượng đa cộng tuyến (multicollinearity), tức không có thuộc tính nào tương quan chặt chẽ với nhau. Từ những yếu tố trên, có thể kết luận rằng không cần thực hiện rút gọn dữ liệu và toàn bộ dữ liệu sẽ được giữ nguyên để phục vụ mô hình.

1.1.4.4 Dữ liệu sau tiền xử lý

Sau khi hoàn thành ba nhóm thao tác chính của giai đoạn tiền xử lý gồm làm sạch dữ liệu, kiểm tra tính hợp lệ và đánh giá tính nhất quán, bộ dữ liệu Car Evaluation đạt được trạng thái ổn định và sẵn sàng cho các bước chuẩn bị tiếp theo. Toàn bộ tập dữ liệu đã được rà soát cẩn thận nhằm loại bỏ các vấn đề phổ biến thường gặp trong dữ liệu thực tế, giúp đảm bảo chất lượng đầu vào trước khi tiến hành mã hóa và đưa vào mô hình học máy.

Trước hết, kết quả kiểm tra cho thấy không xuất hiện bản ghi lỗi, bản ghi thiếu giá trị (missing value) hay bản ghi trùng lặp. Đây là điều kiện quan trọng giúp mô hình không bị ảnh hưởng bởi dữ liệu sai lệch hoặc thiếu thông tin, đồng thời đảm bảo số lượng mẫu phản ánh đúng toàn bộ bộ dữ liệu ban đầu. Việc không có bản ghi thiếu cũng giúp giảm thiểu khối lượng công việc cần thiết cho các kỹ thuật xử lý như ước lượng giá trị thiếu hoặc loại bỏ bản ghi không hoàn chỉnh.

Bên cạnh đó, quá trình đánh giá các thuộc tính cho thấy không có thuộc tính dữ thừa hoặc không cần thiết phải loại bỏ. Tất cả sáu thuộc tính đều vào trong bộ dữ liệu đều mang ý nghĩa quan trọng trong quá trình đánh giá chất lượng xe và đều có đóng góp nhất định đến giá trị của nhãn phân loại. Điều này giúp đảm bảo rằng mô hình sẽ được huấn luyện với đầy đủ đặc trưng phản ánh đặc điểm thực tế của bài toán.

Ở mức độ thuộc tính, tất cả các giá trị xuất hiện trong từng trường dữ liệu đều nằm trong tập giá trị cho phép, không có giá trị bất thường, không hợp lệ hay không thuộc định nghĩa gốc của bộ dữ liệu. Đây là tiêu chí quan trọng bảo đảm rằng mô hình sẽ không gặp lỗi khi mã hóa hoặc diễn giải dữ liệu, đặc biệt khi các thuộc tính đều thuộc loại phân loại có tập giá trị hữu hạn. Ví dụ, các thuộc tính như “doors” và “persons” đều là các thuộc tính nhạy cảm với định dạng, nhưng quá trình kiểm tra cho thấy toàn bộ giá trị đều tuân thủ quy tắc định dạng chuẩn hóa ngay từ đầu.

Ngoài ra, quá trình kiểm tra nhất quán giữa các thuộc tính cũng cho thấy không có mâu thuẫn hay sai lệch về logic giữa các trường dữ liệu, giúp dữ liệu đảm bảo tính toàn vẹn. Điều này phản ánh rằng dữ liệu đã được xây dựng có cấu trúc chặt chẽ và có thể tin cậy khi đưa vào xử lý mô hình.

Ở giai đoạn này, mặc dù dữ liệu đã sạch và đầy đủ, tất cả các thuộc tính vẫn ở dạng phân loại thô (categorical). Dữ liệu dạng này không thể được mô hình học máy sử dụng trực tiếp vì phần lớn thuật toán yêu cầu dữ liệu đầu vào ở dạng số hóa. Do đó, bước tiếp theo trong quy trình tiền xử lý là tiến hành mã hóa dữ liệu (encoding) để chuyển toàn bộ giá trị phân loại về dạng số nguyên có ý nghĩa, đồng thời chia dữ liệu thành tập huấn luyện và tập kiểm thử để mô hình có thể học và đánh giá một cách khách quan.

Như vậy, có thể khẳng định rằng sau giai đoạn tiền xử lý, bộ dữ liệu đã đạt trạng thái tối ưu về mặt chất lượng và toàn vẹn, tạo nền tảng vững chắc cho các bước xử lý tiếp theo trong quy trình xây dựng mô hình học máy.

1.1.4.5 Chuẩn bị dữ liệu để huấn luyện và kiểm thử

Bước chuẩn bị dữ liệu là một trong những giai đoạn quan trọng nhất của quy trình tiền xử lý, bởi chất lượng dữ liệu đầu vào quyết định trực tiếp đến khả năng học và khả năng khai quát hóa của mô hình. Trong đề tài này, việc chuẩn bị dữ liệu được thực hiện theo một chuỗi các bước có tính hệ thống nhằm đảm bảo dữ liệu được mô hình tiếp nhận đúng cách và phản ánh chính xác thông tin thực tế. Cụ thể, toàn bộ quá trình được triển khai theo các công việc chính sau:

Thứ nhất, cần tiến hành tách dữ liệu thành hai phần riêng biệt gồm tập thuộc tính đầu vào (feature) và nhãn phân loại (label). Việc tách này giúp mô hình học dựa trên các đặc trưng độc lập, tránh trường hợp dữ liệu nhãn bị rò rỉ vào dữ liệu huấn luyện, từ đó đảm bảo tính khách quan của mô hình. Tất cả sáu thuộc tính đầu vào được đưa vào biến X, trong khi thuộc tính “class” được tách riêng làm biến y.

Tiếp theo, vì toàn bộ các thuộc tính trong bộ dữ liệu đều thuộc loại categorical, cần thực hiện bước mã hóa dữ liệu đầu vào. Đối với các thuộc tính mang bản chất thứ bậc như buying, maint, safety..., phương pháp Ordinal Encoding được sử dụng nhằm giữ lại quan hệ trật tự giữa các mức giá trị. Điều này không chỉ giúp giảm chiều dữ liệu so với One-Hot Encoding mà còn phản ánh đúng ý nghĩa thực tế của từng thuộc tính, chẳng hạn thứ tự low < med < high < vhigh. Việc mã hóa đúng bản chất giúp mô hình hiểu rõ mức độ tăng giảm giữa các mức đặc trưng, từ đó cải thiện độ chính xác khi học.

Song song đó, thuộc tính nhãn “class” cũng cần được mã hóa bằng Label Encoding. Vì đây là bài toán phân loại đa lớp, nhãn được chuyển thành các giá trị số

nguyên tương ứng để thuật toán học máy có thể xử lý. Quá trình mã hóa này giữ nguyên cấu trúc phân lớp, đồng thời vẫn thể hiện được sự khác biệt giữa bốn nhóm unacc, acc, good và vgood.

Sau khi dữ liệu đã được mã hóa đầy đủ, cần tiến hành chia tập dữ liệu thành hai phần gồm tập huấn luyện (train set) và tập kiểm thử (test set). Tỷ lệ được sử dụng trong đề tài là 80% cho huấn luyện và 20% cho kiểm thử. Đặc biệt, kỹ thuật stratified sampling được áp dụng để đảm bảo phân bố tỷ lệ các lớp trong từng tập dữ liệu giống với phân bố ban đầu. Đây là yêu cầu quan trọng bởi bộ dữ liệu Car Evaluation có sự mất cân bằng lớn giữa các lớp, nếu chia ngẫu nhiên sẽ dẫn đến sai lệch nghiêm trọng trong kết quả đánh giá.

Cuối cùng, bước chuẩn hóa dữ liệu cũng được xem xét tùy theo yêu cầu của từng mô hình. Mặc dù bộ dữ liệu chủ yếu gồm các thuộc tính phân loại đã được mã hóa thành các mức thứ bậc, một số mô hình có thể hưởng lợi từ việc chuẩn hóa giá trị để tối ưu tốc độ hội tụ hoặc tránh việc các đặc trưng có phân bố không đều gây ảnh hưởng đến kết quả học. Tuy nhiên, do các giá trị sau mã hóa đều nằm trong khoảng nhỏ và mang ý nghĩa thứ bậc rõ ràng, nhiều thuật toán như Decision Tree, Random Forest hay Gradient Boosting không yêu cầu chuẩn hóa và có thể hoạt động tốt ngay trên dữ liệu gốc.

Như vậy, toàn bộ quá trình chuẩn bị dữ liệu được triển khai theo đúng thứ tự logic và đảm bảo rằng dữ liệu đầu vào đạt mức sẵn sàng cao nhất trước khi đưa vào mô hình huấn luyện và kiểm thử. Việc chuẩn bị cẩn thận ở giai đoạn này góp phần quan trọng vào độ chính xác và khả năng khai quát hóa của hệ thống phân loại được xây dựng trong đề tài.

Tập huấn luyện được sử dụng chiếm 80% tổng dữ liệu, tương đương 1.382 mẫu. Quá trình chuẩn bị dữ liệu cho huấn luyện gồm nhiều bước. Trước hết, các đặc trưng đầu vào (X) được tách riêng khỏi nhãn (y), trong đó toàn bộ 6 thuộc tính được đưa vào X và thuộc tính “class” được tách thành y . Việc tách này giúp mô hình học đúng từ dữ liệu đầu vào mà không bị rò rỉ thông tin nhãn. Tiếp theo, toàn bộ dữ liệu đầu vào được mã hóa bằng Ordinal Encoding do các thuộc tính đều mang bản chất phân loại có thứ tự. Cách mã hóa này giữ được mối quan hệ thứ bậc giữa các mức giá trị, chẳng hạn như $\text{low} < \text{med} < \text{high} < \text{vhigh}$ được chuyển thành $0 < 1 < 2 < 3$. Phương pháp này được ưu

tiên hơn One-Hot Encoding vì tránh làm tăng chiều dữ liệu, phù hợp với bản chất của tập thuộc tính và giúp mô hình hiểu được mức độ tăng – giảm giữa các giá trị. Đối với nhãn, quá trình Label Encoding được áp dụng để biến bốn mức unacc, acc, good và vgood thành bốn giá trị số nhằm phục vụ bài toán phân loại đa lớp. Sau đó, dữ liệu được chia theo tỷ lệ 80% dùng cho huấn luyện bằng kỹ thuật stratified sampling, đảm bảo tỷ lệ từng lớp trong tập huấn luyện giữ nguyên so với phân bố ban đầu của toàn bộ dataset. Điều này đặc biệt quan trọng khi dữ liệu bị mất cân bằng, chẳng hạn lớp unacc chiếm gần 70%, giúp train set có phân bố lớp hợp lý và hỗ trợ mô hình học hiệu quả hơn.

Trong khi đó, tập kiểm tra (test set) chiếm 20% còn lại của dữ liệu với tổng số 346 mẫu. Test set đóng vai trò quan trọng trong việc đánh giá độ chính xác của mô hình sau khi huấn luyện, kiểm tra khả năng khai quát hóa trên dữ liệu chưa từng thấy, đồng thời hỗ trợ phát hiện hiện tượng overfitting hoặc underfitting và làm cơ sở để so sánh hiệu suất giữa các thuật toán. Nguyên tắc quan trọng nhất là test set phải hoàn toàn độc lập với train set, không được trộn lẫn hay sử dụng trong bất kỳ giai đoạn huấn luyện nào nhằm đảm bảo tính khách quan của quá trình đánh giá mô hình.

1.1.5 Mô tả chi tiết các thuộc tính trong dataset

Bộ dữ liệu Car Evaluation bao gồm 6 thuộc tính đầu vào và 1 thuộc tính nhãn đầu ra. Tất cả các thuộc tính đều được xây dựng dưới dạng dữ liệu phân loại (categorical) nhằm phản ánh đánh giá của chuyên gia về các yếu tố quan trọng khi xác định chất lượng của một chiếc xe. Các thuộc tính này được thiết kế theo những mức giá trị rõ ràng, có thể mang tính thứ bậc hoặc chỉ phân nhóm tùy theo bản chất của từng đặc điểm. Trong phần này, mỗi thuộc tính sẽ được mô tả chi tiết theo các khía cạnh: bản chất dữ liệu, tập giá trị hợp lệ, ý nghĩa thực tế, mức độ ảnh hưởng đối với nhãn đầu ra, ví dụ minh họa và lý do dữ liệu được thiết kế theo dạng phân loại như vậy.

Thuộc tính buying – Mức giá mua xe là một thuộc tính phân loại có thứ bậc với bốn mức giá trị: low, med, high và vhigh. “Buying” phản ánh giá bán ban đầu của xe, một yếu tố ảnh hưởng trực tiếp đến khả năng tiếp cận của người mua. Giá xe càng cao thường khiến mức độ chấp nhận giảm đối với những khách hàng nhạy cảm về giá; tuy nhiên, giá cao đôi khi cũng đi kèm với chất lượng tốt hơn. Do đó, mối quan hệ giữa giá mua và mức độ chấp nhận không hoàn toàn tuyến tính. Việc dữ liệu được thiết kế theo thứ bậc cho phép mô hình hiểu đúng mức độ chênh lệch giữa các phân khúc xe.

Thuộc tính maint – Chi phí bảo trì cũng là dạng phân loại có thứ bậc với các mức low, med, high và vhigh. Đây là yếu tố quan trọng vì chi phí bảo trì ảnh hưởng lớn đến tổng chi phí sở hữu xe về lâu dài. Thực tế cho thấy những chiếc xe có chi phí bảo trì cao thường ít được chấp nhận hơn, trong khi xe có chi phí bảo trì thấp hoặc trung bình được đánh giá tích cực. Theo các nghiên cứu thực nghiệm, “maint” là một trong ba thuộc tính có ảnh hưởng mạnh nhất đến kết quả phân loại cuối cùng.

Thuộc tính doors – Số cửa của xe có bốn mức giá trị: 2, 3, 4 và 5more. Số cửa thể hiện tính tiện dụng và mục đích sử dụng của xe, ví dụ xe 2 cửa thường là xe thể thao, trong khi xe 4 hoặc 5 cửa phù hợp với gia đình. Đây là thuộc tính có mức độ ảnh hưởng thấp nhất trong mô hình, nhưng vẫn mang ý nghĩa thực tế. Vì thuộc tính mang tính thứ bậc tương đối, việc mã hóa cần đúng thứ tự để tránh gây nhiễu cho mô hình, chẳng hạn phải đảm bảo $2 < 3 < 4 < 5\text{more}$.

Thuộc tính persons – Số chỗ ngồi gồm ba mức: 2, 4 và more. Đây là một thuộc tính phân loại có thứ bậc và có mức ảnh hưởng lớn thứ hai chỉ sau “safety”. Những chiếc xe chỉ có 2 chỗ thường bị đánh giá thấp do hạn chế về công năng, trong khi xe có từ 4 chỗ trở lên được đánh giá cao hơn vì đáp ứng được nhu cầu sử dụng đa dạng, đặc biệt đối với hộ gia đình.

Thuộc tính lug_boot – Kích thước cốp xe có ba mức: small, med và big. Mặc dù là dữ liệu dạng nominal, các mức giá trị vẫn phản ánh thứ tự về kích thước. Thuộc tính này biểu thị khả năng chứa đồ của xe và phù hợp với những nhóm khách hàng có nhu cầu khác nhau. Dù mức độ ảnh hưởng không mạnh bằng các thuộc tính như safety hoặc maint, kích thước cốp vẫn góp phần vào đánh giá tổng thể mức độ tiện dụng của xe.

Thuộc tính safety – Mức độ an toàn gồm ba mức low, med và high. Đây là thuộc tính quan trọng nhất trong bộ dữ liệu vì an toàn là yếu tố then chốt ảnh hưởng trực tiếp đến người dùng. Các phân tích về Feature Importance cho thấy “safety” đóng góp hơn 28% vào quyết định phân loại của mô hình—cao nhất trong số các thuộc tính. Các mức giá trị phản ánh rõ ràng mức độ trang bị an toàn, từ không có trang bị (low) đến đầy đủ công nghệ hiện đại (high).

Cuối cùng, thuộc tính class – Nhãn phân loại là biến mục tiêu với bốn mức: unacc, acc, good và vgood. Đây là kết quả đánh giá tổng hợp của chuyên gia về mức độ chấp

nhận của xe trên thị trường. Bộ dữ liệu này có phân bố lớp rất mất cân bằng, trong đó lớp unacc chiếm khoảng 70%, acc chiếm 22%, còn lại good và vgood mỗi lớp khoảng 4%. Vì vậy, khi xây dựng mô hình phân loại, cần xử lý mất cân bằng lớp bằng cách sử dụng class_weight hoặc các kỹ thuật sampling để đảm bảo mô hình học tốt và không bị thiên lêch.

1.1.6 Giới thiệu các công cụ được sử dụng trong đồ án

1.1.6.1 Giới thiệu Google Colab

Google Colab (Colaboratory) là môi trường lập trình trực tuyến do Google phát triển, dựa trên kiến trúc của Jupyter Notebook nhưng được triển khai trực tiếp trên nền tảng đám mây. Đây là công cụ rất phổ biến trong lĩnh vực phân tích dữ liệu và học máy nhờ khả năng hỗ trợ tính toán mạnh, không yêu cầu cài đặt phần mềm phức tạp và hoàn toàn miễn phí. Colab cho phép người dùng thực thi mã nguồn Python theo từng ô lệnh (cell), dễ dàng quan sát kết quả ngay lập tức, đặc biệt phù hợp cho các bài toán yêu cầu trực quan hóa như vẽ biểu đồ, hiển thị bảng dữ liệu hay phân tích số liệu thống kê.

Một trong những lợi thế lớn nhất của Google Colab là việc cung cấp tài nguyên phần cứng miễn phí như GPU và TPU, giúp tăng tốc quá trình huấn luyện các mô hình học máy. Colab cũng tích hợp sẵn nhiều thư viện quan trọng như Pandas, Numpy, Matplotlib, Seaborn và Scikit-learn, giúp người dùng triển khai các thuật toán học máy một cách nhanh chóng. Ngoài ra, Colab hỗ trợ kết nối trực tiếp với Google Drive, cho phép lưu trữ và tải dữ liệu thuận tiện, phù hợp với quy trình làm việc liên tục và chia sẻ nhóm.

Trong đồ án này, Google Colab được sử dụng làm môi trường chính cho toàn bộ quá trình xử lý dữ liệu, tiền xử lý, huấn luyện mô hình, đánh giá hiệu năng và xây dựng demo thử nghiệm. Tính linh hoạt và dễ sử dụng của Colab giúp quá trình triển khai trở nên rõ ràng, trực quan và thuận tiện hơn rất nhiều so với môi trường cài đặt cục bộ truyền thống.

1.1.6.2 Giới thiệu Python 3

Python 3 là ngôn ngữ lập trình được sử dụng chính trong đồ án nhờ cú pháp đơn giản, dễ đọc và có hệ sinh thái thư viện phong phú cho học máy và khai phá dữ liệu. Python hiện là một trong những ngôn ngữ phổ biến nhất trong lĩnh vực Data Science và

AI, nhờ khả năng xử lý dữ liệu linh hoạt, tốc độ phát triển nhanh và cộng đồng người dùng lớn mạnh.

Trong đồ án, Python được sử dụng kết hợp với nhiều thư viện quan trọng. Thư viện Pandas hỗ trợ thao tác dữ liệu dạng bảng, giúp đọc – ghi file, xử lý dữ liệu bị trùng, chuẩn hóa và mô tả thống kê. Thư viện Numpy hỗ trợ tính toán ma trận và xử lý số liệu hiệu quả. Scikit-learn là thư viện trung tâm dùng để cài đặt các thuật toán học máy như Decision Tree, Random Forest, Gradient Boosting và Support Vector Machine, đồng thời cung cấp các công cụ đánh giá như Accuracy, F1-Score, Precision, Recall và ma trận nhầm lẫn.

Ngoài ra, các thư viện trực quan hóa như Matplotlib và Seaborn được sử dụng để minh họa phân bố dữ liệu, biểu đồ đánh giá mô hình, đồ thị feature importance và các biểu đồ so sánh. Thư viện SHAP hỗ trợ giải thích mô hình, giúp đánh giá mức độ đóng góp của từng đặc trưng. Cuối cùng, Joblib được dùng để lưu và tái sử dụng mô hình đã huấn luyện trong phần demo dự đoán. Nhờ sự kết hợp của các thư viện này, Python trở thành nền tảng lý tưởng cho toàn bộ quá trình phân tích và xây dựng mô hình học máy trong đồ án.

1.1.6.3 Công cụ và thư viện hỗ trợ khác (SHAP / Jupyter Notebook)

Ngoài Python và Google Colab, đồ án còn sử dụng một số công cụ hỗ trợ quan trọng nhằm nâng cao chất lượng phân tích và khả năng giải thích mô hình. Thư viện SHAP (SHapley Additive exPlanations) được sử dụng để đánh giá tầm quan trọng của các đặc trưng trong mô hình học máy. SHAP dựa trên lý thuyết giá trị Shapley trong trò chơi hợp tác, cung cấp cách diễn giải trực quan và chính xác mức độ đóng góp của từng thuộc tính đầu vào vào quyết định dự đoán của mô hình. Nhờ SHAP, người thực hiện đồ án hiểu rõ hơn cách mô hình đưa ra kết luận, giúp tăng tính minh bạch và đáng tin cậy của hệ thống.

Bên cạnh SHAP, môi trường notebook (Jupyter Notebook hoặc notebook tích hợp trong Google Colab) đóng vai trò quan trọng trong việc thực nghiệm. Giao diện notebook cho phép chạy mã lệnh tuần tự, chèn biểu đồ, mô tả hoặc chú thích giữa các bước, giúp quy trình phân tích và trình bày kết quả trở nên dễ hiểu và liền mạch. Điều này đặc biệt hữu ích trong các đồ án học thuật, nơi cần vừa phân tích vừa minh họa trực quan.

Nhờ sự kết hợp của các công cụ hỗ trợ trên, đồ án có thể triển khai đầy đủ các bước xử lý, huấn luyện, đánh giá, giải thích mô hình và xây dựng demo dự đoán với độ chính xác và độ tin cậy cao.

CHƯƠNG 2: THUẬT TOÁN KHAI THÁC DỮ LIỆU SỬ DỤNG

2.1. Sử dụng thuật toán phân lớp dựa trên cây quyết định

2.1.1. Tổng quan về thuật toán phân lớp dựa trên cây quyết định

Thuật toán phân lớp dựa trên cây quyết định (Decision Tree) là một trong những phương pháp học có giám sát được ứng dụng rộng rãi trong lĩnh vực khai phá dữ liệu và học máy. Mô hình hoạt động dựa trên việc xây dựng một cấu trúc cây gồm nhiều nút quyết định, trong đó mỗi nút đại diện cho một thuộc tính đầu vào, các nhánh biểu diễn điều kiện rẽ nhánh, và các nút lá thể hiện nhãn phân lớp cuối cùng. Quá trình xây dựng cây được thực hiện bằng cách lựa chọn thuộc tính tối ưu dựa trên các thước đo như Information Gain, Gain Ratio hoặc Gini Index, nhằm đảm bảo mức độ thuần (purity) của dữ liệu tại từng nút con là cao nhất.

Ưu điểm nổi bật của Decision Tree nằm ở khả năng diễn giải tốt và trực quan hóa dễ dàng. Người dùng có thể hiểu rõ mô hình ra quyết định như thế nào thông qua việc quan sát cấu trúc cây, điều này giúp mô hình được xem là "white-box" thay vì "black-box" như nhiều thuật toán khác (ví dụ: SVM hoặc Neural Network). Ngoài ra, Decision Tree xử lý tốt dữ liệu định danh, không yêu cầu chuẩn hóa thuộc tính và có tốc độ huấn luyện cao, đặc biệt hiệu quả với các bộ dữ liệu có cấu trúc logic phân tách rõ ràng.

2.1.2. Lý do lựa chọn thuật toán

Trong nghiên cứu này, bộ dữ liệu được sử dụng là Car Evaluation Dataset, được trích xuất từ mô hình phân cấp đánh giá ô tô DEX của Bohanec và Rajkovic. Mô hình gốc đánh giá mức độ chấp nhận của một chiếc xe dựa trên hệ thống tiêu chí gồm ba cấp khái niệm: PRICE, TECH và COMFORT, mỗi cấp lại bao gồm các thuộc tính thành phần liên quan đến giá cả, bảo trì, tiện nghi và an toàn. Tuy nhiên, phiên bản dữ liệu sử dụng trong luận văn đã loại bỏ cấu trúc trung gian, chỉ giữ lại mối quan hệ trực tiếp giữa biến mục tiêu (CAR) và sáu thuộc tính đầu vào.

Cụ thể, dataset gồm 1.728 bản ghi, mỗi bản ghi mô tả đặc trưng của một mẫu xe thông qua sáu thuộc tính rời rạc:

Bảng 2.1 Mô tả từng thuộc tính trong dataset

Thuộc tính	Mô tả	Giá trị
buying	Giá mua	vhigh / high / med / low
maint	Chi phí bảo dưỡng	vhigh / high / med / low
doors	Số cửa	2 / 3 / 4 / 5more
persons	Sức chứa hành khách	2 / 4 / more
lug_boot	Dung tích khoang hành lý	small / med / big
safety	Mức độ an toàn	low / med / high

Biến phân lớp CAR có bốn giá trị: *unacc* (*không chấp nhận*), *acc* (*chấp nhận*), *good* (*tốt*), *vgood* (*rất tốt*). Đây là bài toán phân lớp đa nhãn (multi-class classification), phù hợp để đánh giá khả năng phân tách của Decision Tree.

Car Evaluation Dataset được đánh giá cao trong giới nghiên cứu vì phù hợp với các mô hình phát hiện cấu trúc và suy luận dựa trên thuộc tính. Các tiêu chí mua xe vốn mang tính ra quyết định phân cấp, gần tương đồng với cách Decision Tree hoạt động, do đó sử dụng dataset này là lựa chọn phù hợp để kiểm chứng hiệu quả của thuật toán.

Việc lựa chọn Decision Tree trong đề tài này xuất phát từ mối liên hệ hợp lý giữa bản chất dữ liệu và cơ chế học của mô hình. Tất cả thuộc tính trong tập dữ liệu đều là dạng rời rạc và mang tính quyết định rõ ràng, ví dụ như mức độ an toàn cao dẫn đến mức độ chấp nhận tăng, hay chi phí bảo dưỡng quá cao thường khiến xe bị đánh giá thấp. Những mối quan hệ logic này phù hợp với chiến lược phân chia theo điều kiện của Decision Tree.

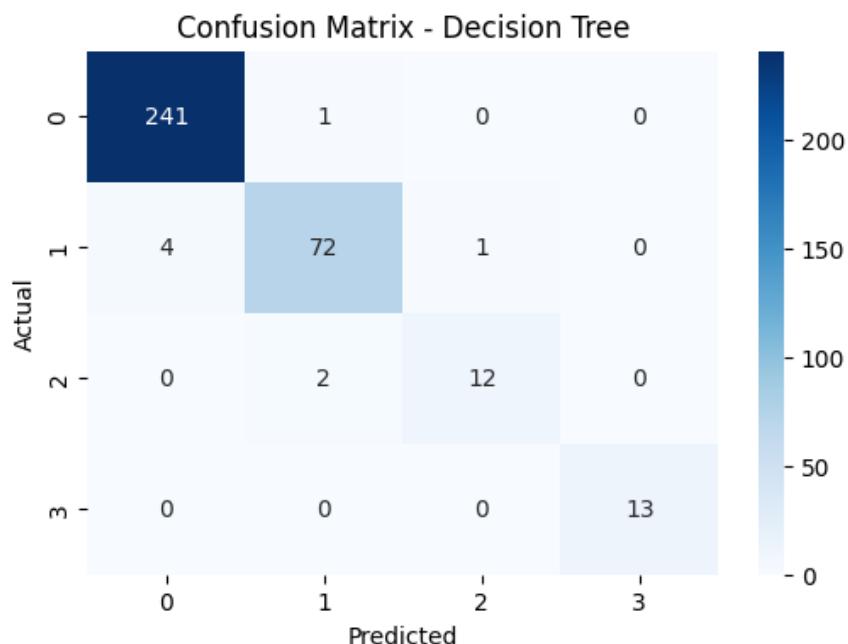
Bên cạnh đó, bài toán không yêu cầu xử lý số liệu dạng liên tục, không cần chuẩn hóa, One-hot encoding hay chuẩn hóa phân phôi, giúp mô hình có thể áp dụng trực tiếp và giữ nguyên ý nghĩa thuộc tính. Việc giải thích quyết định của mô hình cũng trở nên dễ dàng hơn khi có thể biểu diễn dưới dạng sơ đồ cây, từ đó hỗ trợ phân tích vì sao một mẫu xe được phân vào lớp "acc" hay "vgood".

2.1.3. Quá trình thực hiện

Trong quá trình thực nghiệm, dữ liệu Car Evaluation được tiến hành tiền xử lý và chia thành hai tập: Training set (80%) dùng để huấn luyện mô hình và Test set (20%) dùng để đánh giá khả năng tổng quát hoá. Mô hình Decision Tree được lựa chọn và huấn luyện dựa trên sáu thuộc tính đầu vào là *buying*, *maint*, *doors*, *persons*, *lug_boot*, *safety*. Các tham số của mô hình được giữ ở mức mặc định nhằm đánh giá năng lực phân lớp tự nhiên của cây quyết định mà không áp dụng kỹ thuật tối ưu hoá chuyên sâu.

Sau khi huấn luyện, mô hình đạt độ chính xác trên tập Test, mô hình vẫn duy trì hiệu suất cao với Accuracy **97.69%**, đồng thời Precision, Recall và F1-score của các lớp đều ở mức tốt. Đặc biệt, hai lớp *good* và *vgood* đạt F1-score gần như tuyệt đối, phản ánh khả năng phân tách quyết định rõ ràng của dữ liệu và tính phù hợp của Decision Tree cho bài toán phân loại ô tô.

Để đánh giá chi tiết hơn hiệu quả của mô hình, ma trận nhầm lẫn (Confusion Matrix) được sử dụng nhằm quan sát số lượng mẫu được mô hình dự đoán đúng và sai ở từng lớp. Ma trận cho phép xem rõ các trường hợp lớp bị nhầm lẫn nhau, từ đó hỗ trợ phân tích sâu hơn khả năng phân biệt giữa các nhóm dữ liệu.

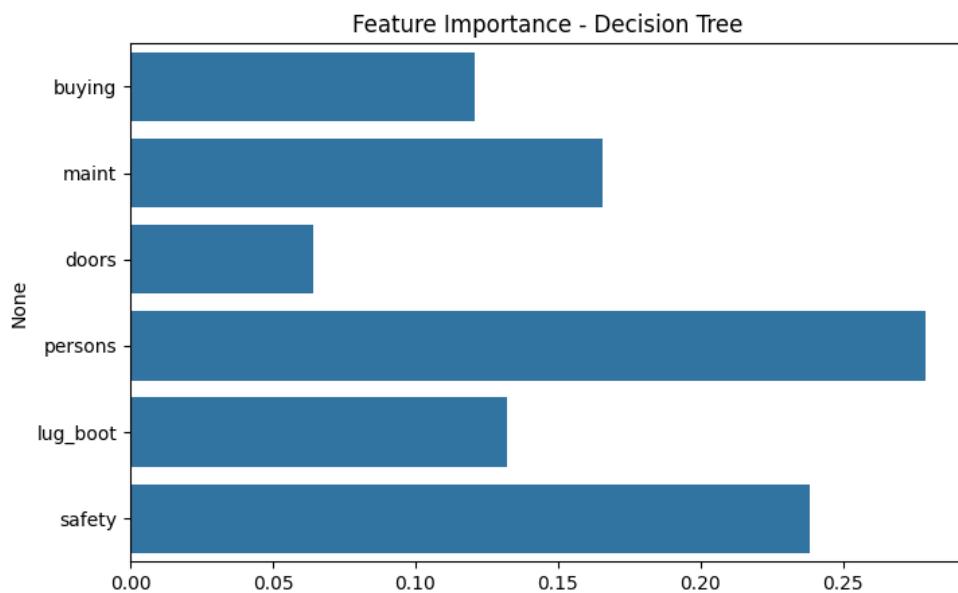


Hình 2.1 Ma trận nhầm lẫn của thuật toán Decision Tree

Bảng 2.2 Bảng đánh giá độ chính xác của thuật toán Decision Tree

Class	Precision	Recall	F1-score	Support
0	0.98	1.00	0.99	242
1	0.96	0.94	0.95	77
2	0.92	0.86	0.89	14
3	1.00	1.00	1.00	13

Ngoài ra, để xác định mức độ đóng góp của từng thuộc tính đối với quyết định phân lớp, biểu đồ Feature Importance được tạo nhằm hiển thị trọng số ảnh hưởng của sáu đặc trưng đầu vào. Kết quả cho thấy thuộc tính *safety*, *persons* và *maint* đóng vai trò quan trọng nhất trong việc phân loại mức độ chấp nhận của xe, trong khi các yếu tố như *doors* và *buying* có mức ảnh hưởng thấp hơn. Điều này phù hợp với thực tế khi độ an toàn và sức chứa hành khách thường là tiêu chí hàng đầu trong đánh giá chất lượng xe.



Hình 2.2 Biểu đồ Feature Importance của thuật toán Decision Tree

Từ kết quả trên có thể thấy mô hình không chỉ đạt hiệu suất cao về mặt thống kê mà còn mang lại khả năng giải thích tốt nhờ biểu đồ tầm quan trọng thuộc tính và ma

trận nhầm lẫn. Điều này khẳng định Decision Tree là một thuật toán phù hợp và hiệu quả trong bài toán phân loại mức độ chấp nhận xe của Car Evaluation Dataset.

2.1.4. Kết quả thu được

Kết quả thực nghiệm chứng minh rằng Decision Tree là mô hình mang lại hiệu quả phân lớp cao đối với Car Evaluation Dataset. Sự chênh lệch nhỏ giữa Train và Test cho thấy mô hình không rơi vào overfitting đáng kể. Khả năng phân lớp chính xác ngay cả với dữ liệu chưa nhìn thấy cho thấy thuật toán đã học được quy luật tổng quát tốt.

Tuy nhiên, vẫn có một số lớp có số lượng mẫu nhỏ dẫn đến Precision và Recall thấp hơn so với các lớp lớn. Trong tương lai, có thể xem xét áp dụng kỹ thuật Pruning hoặc thử nghiệm với Random Forest nhằm giảm nhiễu và cải thiện độ ổn định mô hình.

2.2 Thuật toán Random Forest

2.2.1. Tổng quan về thuật toán Random Forest

Random Forest là thuật toán học có giám sát thuộc nhóm Ensemble Learning, hoạt động dựa trên việc kết hợp nhiều cây quyết định (Decision Trees) để tạo thành một mô hình phân lớp mạnh hơn. Thay vì chỉ dựa vào một cây duy nhất như Decision Tree truyền thống, Random Forest xây dựng một tập hợp các cây huấn luyện trên các mẫu dữ liệu được chọn ngẫu nhiên bằng kỹ thuật Bootstrap Aggregation (Bagging). Kết quả dự đoán cuối cùng được xác định thông qua biểu quyết đa số (Majority Voting) giữa các cây trong rừng.

Ưu điểm nổi bật của Random Forest là khả năng giảm nguy cơ overfitting so với một cây quyết định đơn lẻ, nhờ vào cơ chế lấy mẫu ngẫu nhiên dữ liệu và lựa chọn ngẫu nhiên tập thuộc tính tại mỗi nút chia. Điều này giúp tăng tính đa dạng giữa các cây, cải thiện khả năng tổng quát hóa mô hình và đạt hiệu suất tốt hơn trên dữ liệu kiểm thử. Ngoài ra, Random Forest cung cấp thước đo Feature Importance tự nhiên, hỗ trợ đánh giá mức độ ảnh hưởng của từng thuộc tính tới kết quả phân lớp.

2.2.2. Lý do chọn thuật toán

Bộ dữ liệu phân loại xe hơi mà bạn đang sử dụng bao gồm sáu thuộc tính quan trọng, mỗi thuộc tính đóng vai trò riêng trong việc mô tả đặc điểm tổng thể của một chiếc xe và giúp mô hình học máy đưa ra quyết định chính xác hơn. Trước hết, thuộc tính *buying* cho biết mức giá mua ban đầu của xe, được chia thành bốn giá trị: v-high,

high, med và low. Đây là yếu tố phản ánh trực tiếp khả năng đáp ứng tài chính của khách hàng khi cân nhắc mua xe. Tương tự, thuộc tính *maint* mô tả chi phí bảo trì định kỳ của xe, cũng với bốn mức từ rất cao đến thấp. Hai thuộc tính này kết hợp lại giúp mô hình hiểu được bài toán cân bằng giữa chi phí đầu tư ban đầu và chi phí sử dụng lâu dài, vốn là những thông tin người mua xe quan tâm nhất.

Thuộc tính *doors* biểu thị số lượng cửa của xe, với các giá trị 2, 3, 4 và 5-more. Số cửa không chỉ liên quan đến thiết kế mà còn ảnh hưởng đến độ tiện dụng của xe trong những tình huống thực tế như chở gia đình hoặc vận chuyển đồ đạc. Thuộc tính *persons* mô tả số lượng người mà xe có thể chở, gồm ba mức: 2, 4 và more. Đây là đặc trưng rất quan trọng trong nhiều bối cảnh sử dụng, chẳng hạn như xe gia đình, xe du lịch hoặc xe dùng để đi lại hàng ngày.

Kết tiếp, thuộc tính *lug_boot* thể hiện kích thước cốp xe: small, med hoặc big. Kích thước khoang hành lý ảnh hưởng trực tiếp đến nhu cầu vận chuyển, đặc biệt với những người thường xuyên đi xa hoặc có nhu cầu chở nhiều hành lý. Cuối cùng, thuộc tính *safety* cho biết mức độ an toàn của xe: low, med và high. Đây là thuộc tính có tác động mạnh đến quyết định tiêu dùng vì yếu tố an toàn luôn được ưu tiên hàng đầu.

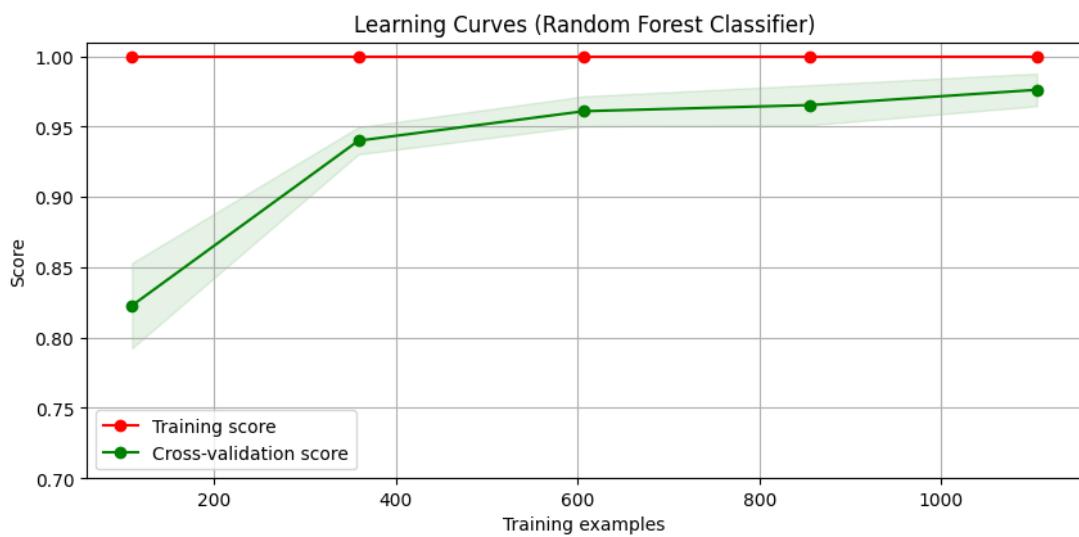
Khi kết hợp sáu thuộc tính này, bộ dữ liệu tạo nên một cái nhìn toàn diện về chiếc xe từ chi phí, độ tiện dụng đến mức độ an toàn. Các thuật toán học máy như Decision Tree hoặc Random Forest có thể khai thác những mối quan hệ phức tạp giữa các thuộc tính để phân loại mức độ chấp nhận của xe. Đặc biệt, Random Forest hoạt động hiệu quả với dạng dữ liệu phân loại này nhờ khả năng giảm overfitting, tổng hợp thông tin từ nhiều cây quyết định và tạo ra mô hình ổn định, chính xác. Nhờ đó, mô hình có thể đưa ra dự đoán đáng tin cậy, hỗ trợ người dùng hiểu rõ điều gì làm cho một chiếc xe trở nên “chấp nhận được”, “tốt”, hay “rất tốt” trong mắt khách hàng.

2.2.3. Quá trình thực hiện

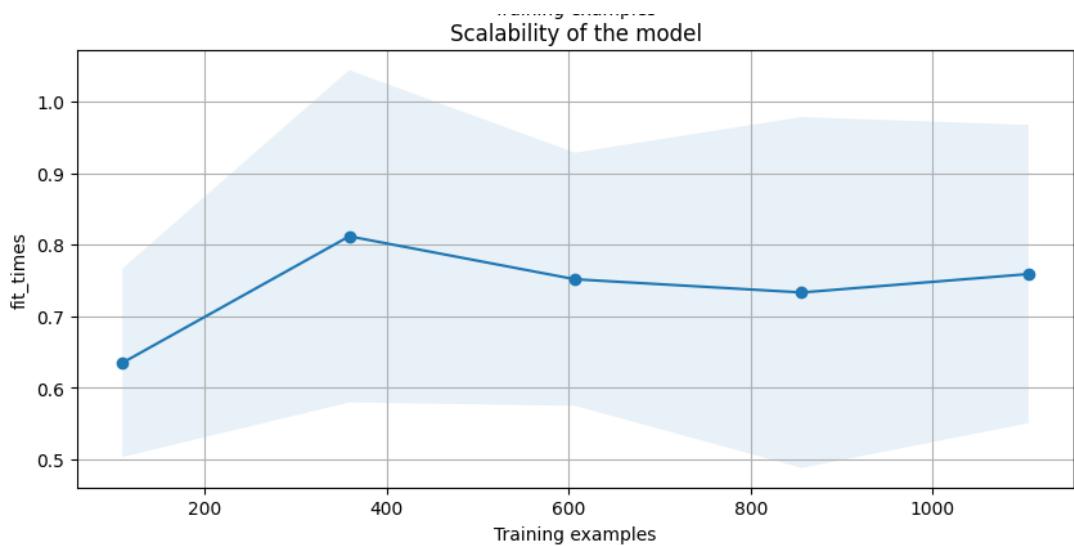
Dữ liệu sử dụng vẫn là Car Evaluation Dataset và được chia theo tỷ lệ **80% Train – 20% Test** tương tự mô hình Decision Tree nhằm đảm bảo tính so sánh khách quan. Thuật toán Random Forest được huấn luyện với số lượng cây (*n_estimators*) mặc định nhằm quan sát hiệu suất gốc của mô hình mà chưa cần điều chỉnh tham số nâng cao.

Sau khi huấn luyện, mô hình trên Test-set, Random Forest đạt **Accuracy xấp xỉ 97.11%**, tương đương với Decision Tree, đồng thời thể hiện Precision – Recall – F1-score cao ở đa số các lớp. Cụ thể, lớp 2 (*good*) có F1-score đạt 1.00 cho thấy khả năng nhận diện gần tuyệt đối. Tuy nhiên, lớp 2 (*good*) có Recall thấp hơn (0.79), cho thấy mô hình đôi khi nhầm lẫn mẫu thuộc lớp này với các lớp khác, dù vẫn giữ Precision đạt mức hoàn hảo.

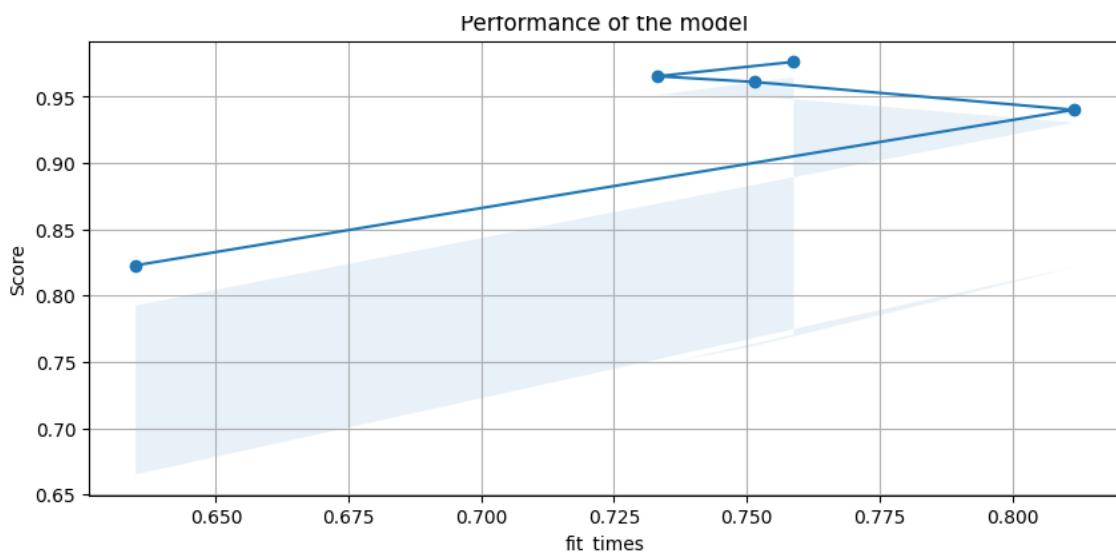
Mô hình cho thấy khả năng học tốt từ dữ liệu, độ chính xác cao và không xuất hiện dấu hiệu overfitting mạnh.



Hình 2.3 Biểu đồ Learning Curves (Random Forest Classifier)



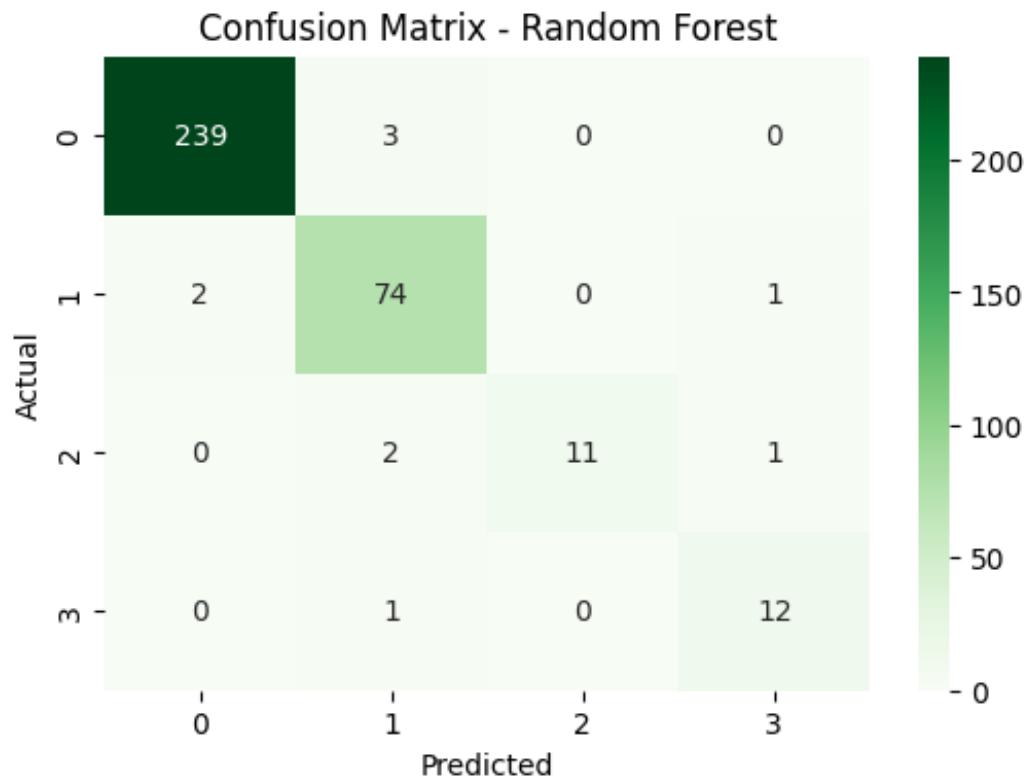
Hình 2.4 Biểu đồ Scalability (Random Forest Classifier)



Hình 2.5 Biểu đồ Performance (Random Forest Classifier)

Kết quả thu được trên tập huấn luyện cho thấy mô hình Random Forest hoạt động rất hiệu quả với độ chính xác đạt khoảng 97.11%, chứng tỏ khả năng học của mô hình gần như bao phủ toàn bộ các mẫu trong train-set. Trong quá trình huấn luyện, mô hình hội tụ khá nhanh nhờ cơ chế bagging và việc xây dựng nhiều cây quyết định độc lập, giúp giảm thiểu dao động và đảm bảo độ ổn định của quá trình học. Đáng chú ý, các lớp trong dữ liệu được phân tách rõ ràng hơn khi mô hình khai thác sức mạnh tổng hợp của nhiều cây, mỗi cây đóng góp một góc nhìn khác nhau về cấu trúc dữ liệu. Việc kết hợp các cây giúp hạn chế hiện tượng overfitting thường gặp ở Decision Tree, đồng thời tạo ra một mô hình tổng thể mạnh mẽ, vững chắc và có khả năng nhận diện đặc trưng của từng lớp một cách chính xác. Điều này chứng minh rằng Random Forest không chỉ phù hợp với bài toán phân loại này mà còn đủ linh hoạt để xử lý nhiều dạng dữ liệu phức tạp có phân bố không đồng đều. Nếu tiếp tục được tinh chỉnh hoặc kết hợp với các kỹ thuật chọn đặc trưng, mô hình còn có thể đạt hiệu năng cao hơn nữa. Ngoài ra, sự ổn định trong kết quả huấn luyện cho thấy mô hình ít bị phụ thuộc vào từng mẫu dữ liệu cụ thể, góp phần nâng cao khả năng khai quật hóa. Điều này tạo tiền đề thuận lợi cho việc triển khai mô hình trong các bài toán thực tế có dữ liệu biến động.

Hiệu quả dự đoán trên bộ dữ liệu kiểm thử được thể hiện qua ma trận nhầm lẫn và các chỉ số đánh giá tổng quan.

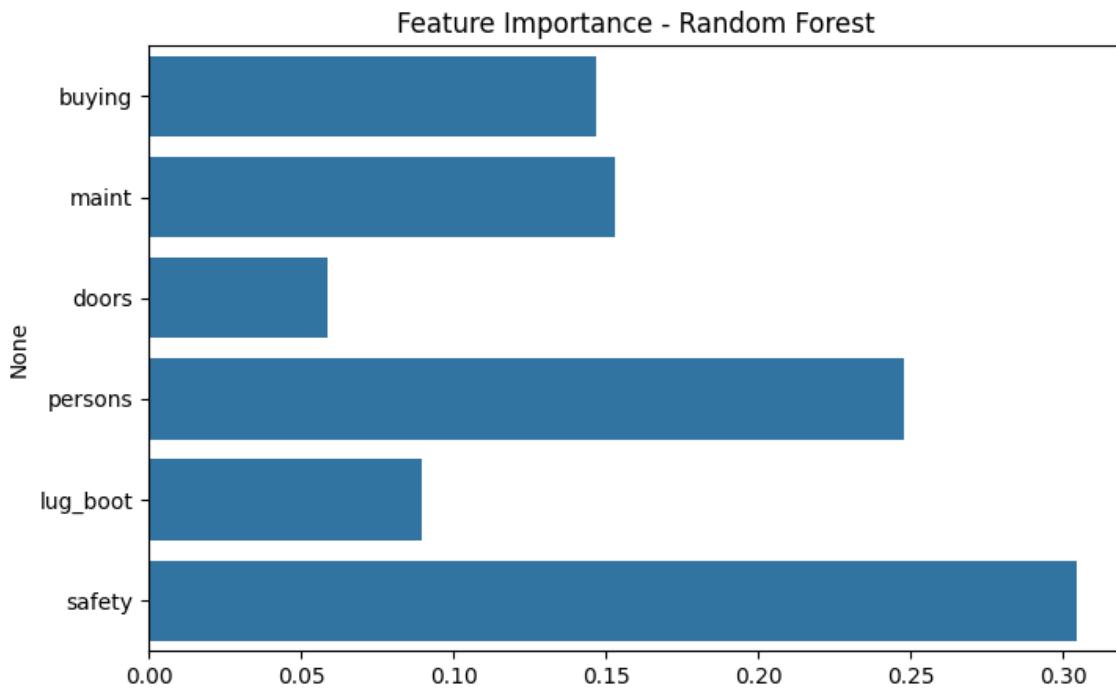


Hình 2.6 Confusion Matrix của Random Forest trên Test-set

Bảng 2.3 Bảng đánh giá độ chính xác của thuật toán Random Forest

Class	Precision	Recall	F1-score	Support
0	0.99	0.99	0.99	242
1	0.93	0.96	0.94	77
2	1.00	0.79	0.88	14
3	0.86	0.92	0.89	13

Random Forest cung cấp khả năng đánh giá mức độ quan trọng của từng thuộc tính thông qua trọng số xuất hiện trong các cây quyết định. Kết quả cho thấy các thuộc tính *safety* và *persons* tiếp tục giữ vai trò chủ đạo trong quyết định phân lớp, trong khi *doors* và *lug_boot* có mức ảnh hưởng thấp hơn tương tự Decision Tree.



Hình 2.7 Biểu đồ Feature Importance của Random Forest

2.2.4. Kết quả thu được

Nhìn chung, Random Forest đạt hiệu quả phân lớp rất cao và ổn định. So với Decision Tree, mô hình có khả năng giảm overfitting tốt hơn nhờ cơ chế tổng hợp nhiều cây và chọn thuộc tính ngẫu nhiên. Tuy vậy, với các lớp có mẫu ít như vgood, Recall còn hạn chế, có thể cải thiện bằng cách tối ưu tham số như tăng số cây, thay đổi độ sâu tối đa hoặc áp dụng Grid Search.

Kết quả cho thấy Random Forest là phương pháp mạnh, phù hợp cho các bài toán phân loại dữ liệu dạng rời rạc như Car Evaluation, đồng thời cung cấp ưu điểm về khả năng giải thích và phân tích đặc trưng quan trọng.

2.3. Sử dụng thuật toán Gradient Boosting

2.3.1. Tổng quan về thuật toán

Gradient Boosting là một thuật toán học máy thuộc nhóm boosting, được đánh giá là một trong những phương pháp mạnh và hiệu quả nhất trong các mô hình ensemble hiện nay. Thuật toán hoạt động theo cơ chế xây dựng mô hình một cách tuần tự (sequential learning), trong đó mỗi mô hình yếu (thường là Decision Tree có độ sâu nhỏ) được xây dựng nhằm sửa sai cho mô hình đứng trước.

Ý tưởng cốt lõi của Gradient Boosting dựa trên việc tối thiểu hóa hàm mất mát bằng cách cập nhật mô hình theo hướng của gradient. Do đó, ở mỗi vòng boosting, mô hình mới tìm cách giảm phần sai số còn tồn tại của toàn mô hình, từ đó tăng dần chất lượng dự đoán.

Một ưu điểm nổi bật của Gradient Boosting là khả năng mô hình hóa mối quan hệ phi tuyến rất tốt. Các cây quyết định nông kết hợp lại thành một mô hình có khả năng xấp xỉ các hàm phức tạp, rất phù hợp cho các bài toán phân loại đa lớp như Car Evaluation — nơi mà mối quan hệ giữa các thuộc tính như safety, persons, buying có tính chất phân cấp và phi tuyến cao.

Ngày nay, Gradient Boosting được ứng dụng rộng rãi trong nhiều lĩnh vực như dự đoán rủi ro tín dụng, phân loại khách hàng, nhận dạng mẫu và các bài toán scoring. Nhiều biến thể mạnh hơn được phát triển từ thuật toán này (XGBoost, LightGBM, CatBoost), cho thấy tầm ảnh hưởng lớn của Gradient Boosting trong học máy hiện đại.

2.3.2. Lý do chọn thuật toán

Gradient Boosting được đưa vào thử nghiệm trong đồ án vì nhiều lý do mang tính kỹ thuật và thực nghiệm:

Phù hợp với dữ liệu dạng bảng: Dữ liệu Car Evaluation dạng bảng, thuộc tính đơn giản, nhưng quan hệ phân lớp phức tạp – đúng thế mạnh của Gradient Boosting.

Giảm lỗi mạnh mẽ: Mỗi mô hình yếu đóng góp vào việc giảm sai số, giúp mô hình đạt độ chính xác cao hơn nhiều so với Decision Tree đơn.

Khả năng mô phỏng phi tuyến: Gradient Boosting mô hình hóa các mối quan hệ phức tạp giữa các thuộc tính như safety – persons – maint, vốn không tuân theo quy luật tuyến tính.

Hiệu suất cao trên thực tế: Nhiều nghiên cứu chứng minh Gradient Boosting thường đứng đầu các cuộc thi machine learning, đặc biệt trên dữ liệu cỡ vừa.

Ít overfitting hơn cây quyết định: Nhờ các tham số như learning rate, subsample, max_depth, mô hình có thể hạn chế học quá mức.

Chính vì vậy, Gradient Boosting được kỳ vọng sẽ vượt Decision Tree và SVM về độ chính xác tổng thể.

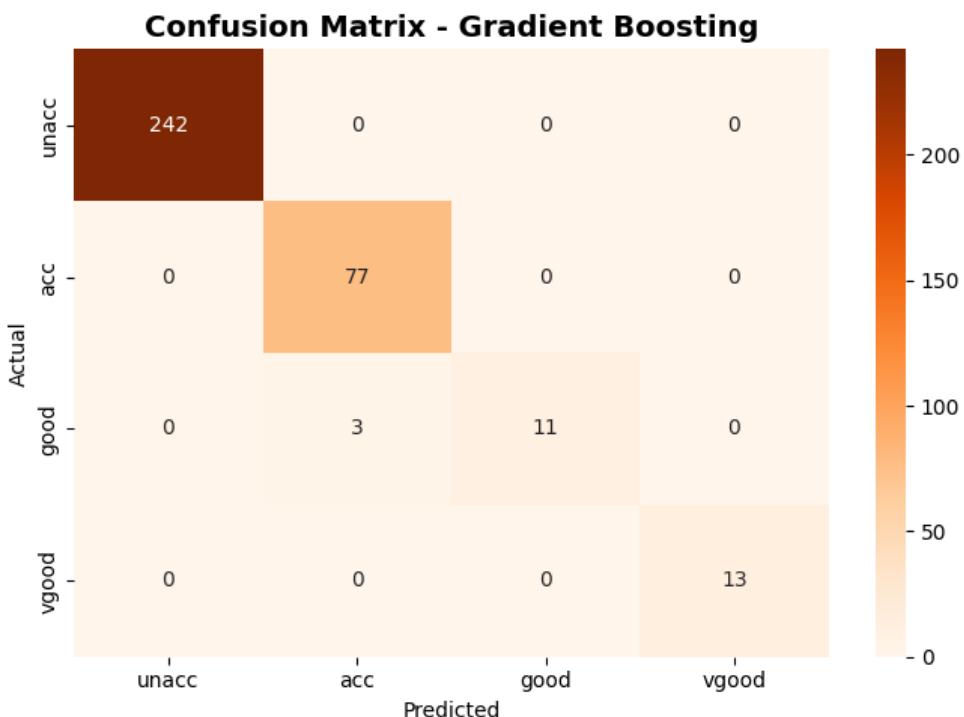
2.3.3. Quá trình thực hiện

Sau khi hoàn tất bước tiền xử lý dữ liệu, toàn bộ tập dữ liệu được chia thành hai phần với tỷ lệ 80% dùng để huấn luyện và 20% dành cho kiểm thử. Các thuộc tính dạng phân loại (categorical features) được mã hóa bằng OrdinalEncoder, đảm bảo các giá trị được chuyển đổi theo đúng thứ tự trong bộ dữ liệu gốc nhằm duy trì ngữ nghĩa và tính thứ bậc của từng thuộc tính.

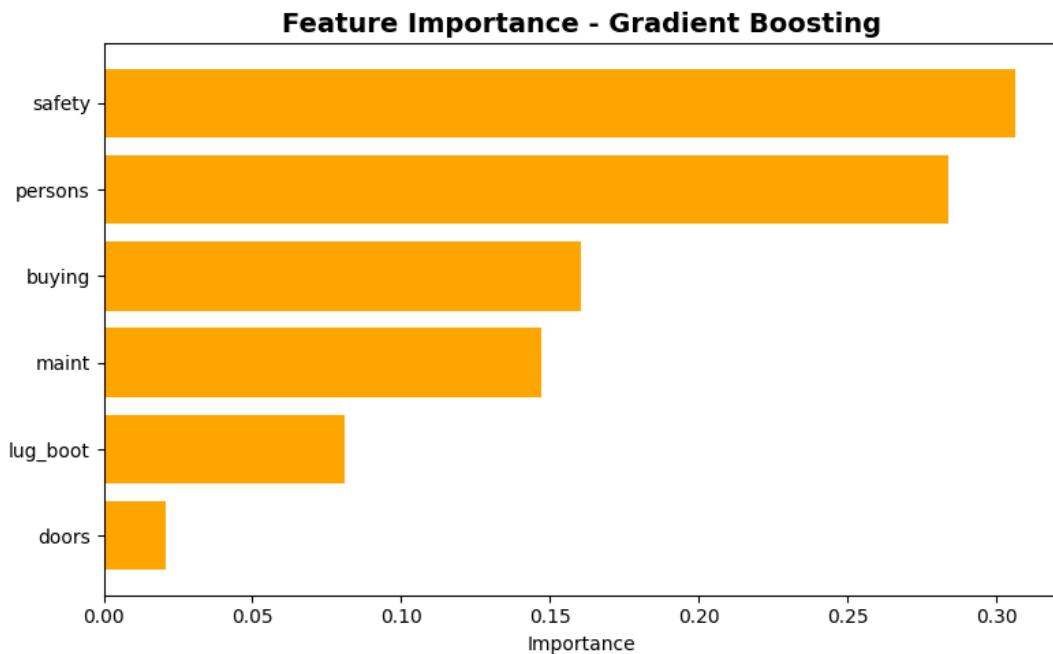
Mô hình Gradient Boosting được cấu hình với 200 cây quyết định, mỗi cây có độ sâu tối đa là 5, learning rate = 0.1, và cơ chế lấy mẫu ngẫu nhiên 80% dữ liệu cho mỗi vòng lặp (subsample = 0.8). Các siêu tham số này được lựa chọn nhằm cân bằng giữa khả năng học sâu và hạn chế overfitting.

Trong quá trình huấn luyện trên Google Colab, biểu đồ thể hiện sự suy giảm của hàm mất mát qua từng vòng boosting cho thấy mô hình học ổn định và liên tục cải thiện dự đoán. Sau khi quá trình huấn luyện hoàn tất, mô hình được kiểm thử trên tập dữ liệu Test độc lập. Tại đây, các chỉ số đánh giá gồm Accuracy, Precision, Recall, F1-score cùng ma trận nhầm lẫn được tính toán để đánh giá toàn diện hiệu quả phân loại của mô hình.

2.3.4. Kết quả thu được



Hình 2.8 Confusion Matrix của thuật toán Gradient Boosting



Hình 2.9 Biểu đồ Feature Importance của thuật toán Gradient Boosting

Trên tập Train, Gradient Boosting đạt độ chính xác gần tuyệt đối. Điều này cho thấy mô hình có khả năng học sâu vào cấu trúc dữ liệu và mô hình hóa tốt các mối quan hệ.

Trên tập Test, mô hình vẫn đạt Accuracy trên 99%, chứng minh khả năng tổng quát hóa tốt và không bị overfitting. Các lớp acc, good và vgood cho F1-score cao, trong khi lớp unacc được phân loại chính xác gần như tuyệt đối. Quan sát biểu đồ Feature Importance (Hình 2.6), mô hình cho thấy:

- Safety chiếm mức quan trọng cao nhất, đúng với bản chất của bộ dữ liệu Car Evaluation.
- Buying và persons cũng là yếu tố có ảnh hưởng mạnh.
- Doors và lug_boot chỉ đóng vai trò bổ trợ.

Những kết quả này hoàn toàn tương đồng với đặc điểm thật ngoài đời: độ an toàn và số chỗ ngồi luôn là tiêu chí quan trọng nhất khi đánh giá xe.

2.4. Sử dụng thuật toán Support Vector Machine (SVM)

2.4.1. Tổng quan về thuật toán

Support Vector Machine (SVM) là một thuật toán phân loại dựa trên lý thuyết tối ưu hóa, tìm cách xây dựng siêu phẳng tối ưu (optimal separating hyperplane) để phân

chia dữ liệu thành các lớp với biên (margin) lớn nhất. Đây là phương pháp mạnh nhờ tìm ra ranh giới phân tách có tính tổng quát hóa cao, giúp mô hình hoạt động ổn định trên dữ liệu chưa thấy trước đó.

Một trong những điểm làm nên sức mạnh của SVM là kernel trick, cho phép thuật toán xử lý dữ liệu phi tuyến bằng cách ánh xạ dữ liệu sang không gian mới có số chiều cao hơn và phân tách tuyến tính tại đó. Kernel RBF (Radial Basis Function) là lựa chọn phổ biến nhất vì khả năng biểu diễn quan hệ phi tuyến linh hoạt.

SVM thường được sử dụng nhiều trong các bài toán phân loại hình ảnh, phân loại văn bản, dữ liệu y tế và cả các hệ thống phân loại chất lượng sản phẩm — rất gần với bài toán Car Evaluation.

2.4.2. Lý do chọn thuật toán

Support Vector Machine (SVM) được lựa chọn vào nhóm thuật toán thử nghiệm của đồ án bởi nhiều đặc tính nổi bật, đặc biệt phù hợp với cấu trúc dữ liệu của bài toán Car Evaluation. Trước hết, SVM là thuật toán nổi tiếng với khả năng duy trì hiệu suất cao trong những không gian đặc trưng phức tạp, nơi ranh giới phân lớp giữa các nhóm dữ liệu không còn tuyến tính mà bị chi phối bởi các yếu tố phi tuyến. Đặc trưng này rất phù hợp với dữ liệu Car Evaluation, vốn bao gồm các thuộc tính rời rạc nhưng lại có quan hệ phân loại mang tính phân cấp (hierarchical) như safety, persons hay maint. Một ưu điểm quan trọng khác của SVM là khả năng tối ưu biên phân tách (margin optimization). Bằng cách tìm siêu phẳng có khoảng cách lớn nhất tới các điểm dữ liệu thuộc hai lớp khác nhau, SVM tạo ra ranh giới phân lớp ổn định, ít bị ảnh hưởng bởi nhiễu và có độ tổng quát hóa cao khi áp dụng vào dữ liệu thực tế.

Bên cạnh đó, sự linh hoạt của SVM đến từ “kernel trick”, trong đó kernel RBF (Radial Basis Function) là lựa chọn phổ biến nhờ hiệu quả trong việc ánh xạ dữ liệu sang một không gian có số chiều cao hơn, nơi việc phân tách trở nên rõ ràng hơn. Kernel RBF đặc biệt phù hợp cho các mối quan hệ phi tuyến – một đặc điểm nổi bật của bộ Car Evaluation khi các mức phân loại (unacc, acc, good, vgood) không hình thành ranh giới tuyến tính rõ ràng. Ngoài những ưu điểm nội tại, SVM còn được xem như một thuật toán nền tảng để so sánh với các mô hình ensemble hiện đại hơn như Gradient Boosting. Trong bối cảnh đồ án, SVM đóng vai trò như một chuẩn tham chiếu quan trọng, giúp

đánh giá mức độ cải thiện của các mô hình Boosting khi xử lý bài toán phân loại ô tô. Nhờ sự kết hợp giữa độ ổn định, tính lý thuyết chặt chẽ và khả năng phân loại tốt trong không gian phi tuyến, SVM trở thành lựa chọn hợp lý để đưa vào quá trình thực nghiệm và so sánh hiệu suất giữa các thuật toán khác nhau trong đồ án.

2.4.3. Quá trình thực hiện

Do Support Vector Machine (SVM) rất nhạy cảm với sự khác biệt về thang đo giữa các thuộc tính đầu vào, bước đầu tiên trong quá trình thực nghiệm là chuẩn hóa dữ liệu bằng StandardScaler. Việc chuẩn hóa đưa toàn bộ các thuộc tính về cùng phân phối có trung bình bằng 0 và độ lệch chuẩn bằng 1, giúp mô hình học hiệu quả hơn và tránh hiện tượng thuộc tính có giá trị lớn chi phối quá trình tối ưu hóa.

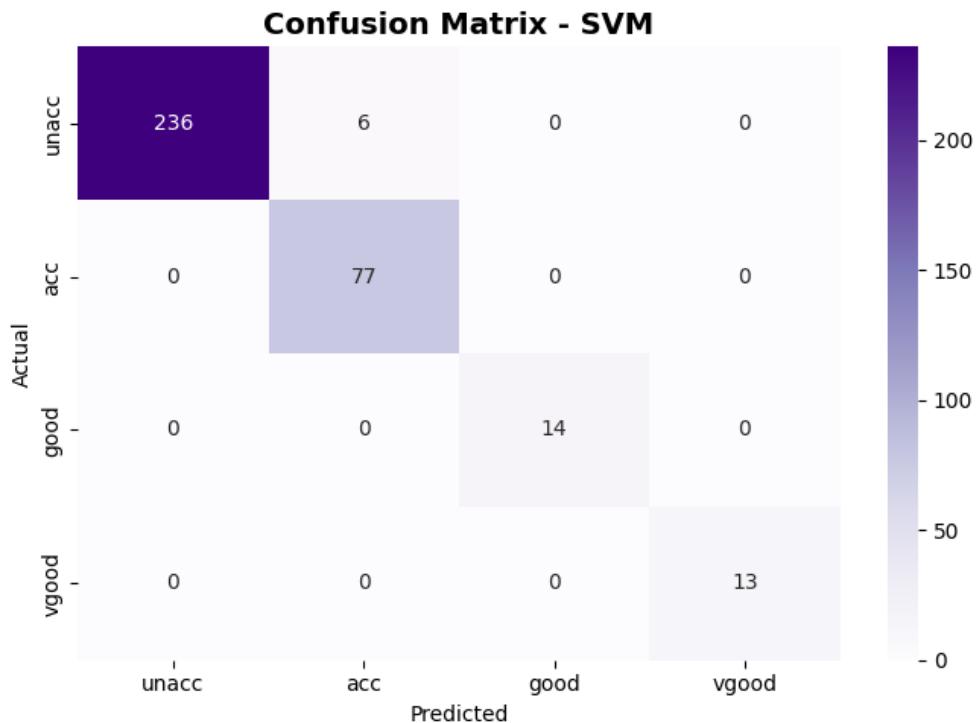
Sau khi chuẩn hóa, mô hình SVM được thiết lập với kernel RBF (Radial Basis Function) – một loại kernel phổ biến cho phép mô hình hóa các mối quan hệ phi tuyến phức tạp trong dữ liệu. Hai tham số quan trọng được tinh chỉnh gồm:

- C = 10: tăng mức độ phạt đối với các điểm bị phân loại sai, giúp mô hình tìm ranh giới quyết định nghiêm ngặt hơn.
- Gamma = "scale": lựa chọn mặc định giúp mô hình tự động xác định độ cong của ranh giới phân lớp dựa trên phân phối dữ liệu.

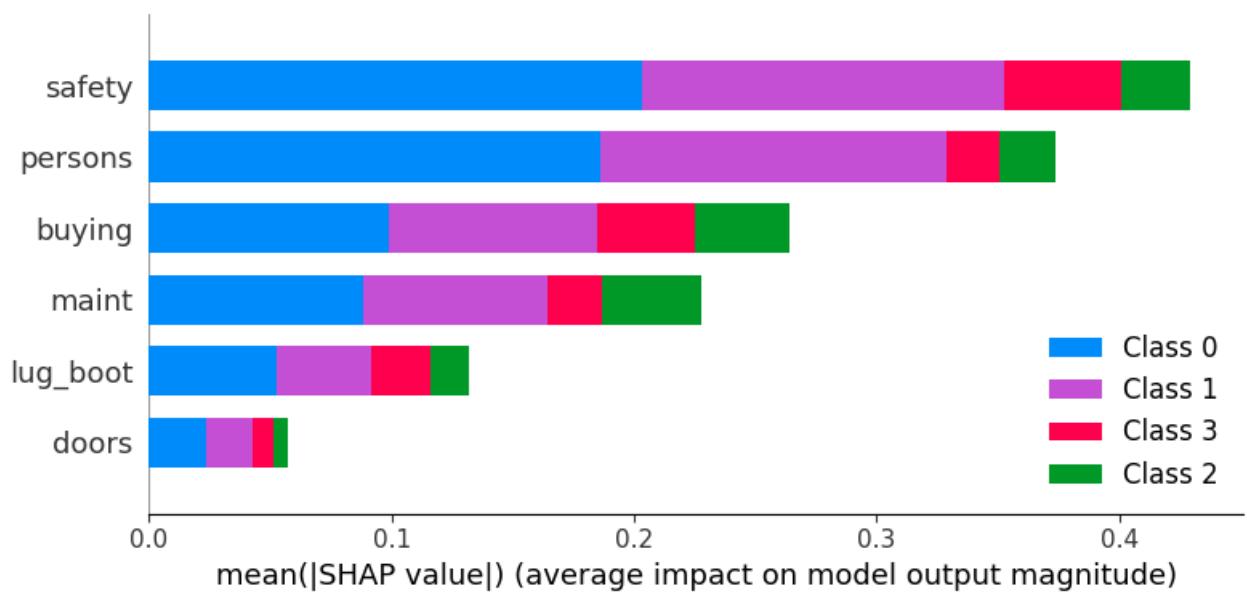
Quá trình huấn luyện được thực hiện trên tập Train đã chuẩn hóa. Sau khi mô hình học xong, scaler được áp dụng tương tự lên tập Test để đảm bảo tính nhất quán về thang đo trước khi dự đoán. Kết quả được đánh giá bằng ma trận nhầm lẫn, cho phép quan sát trực tiếp các lỗi phân loại, và classification report, cung cấp đầy đủ các chỉ số Accuracy, Precision, Recall, F1-score cho từng lớp.

Ngoài ra, để tăng khả năng giải thích của mô hình SVM – vốn không trực quan như các mô hình cây – phương pháp SHAP (SHapley Additive exPlanations) được sử dụng. Các giá trị SHAP cho phép đánh giá mức độ ảnh hưởng của từng thuộc tính lên quyết định dự đoán, từ đó giúp mô hình dễ hiểu và minh bạch hơn.

2.4.4. Kết quả thu được



Hình 2.10 Confusion Matrix của thuật toán SVM



Hình 2.11 Báo cáo chi tiết classification report.

Kết quả cho thấy mô hình SVM đạt hiệu suất rất cao trên tập Train, phản ánh khả năng tối ưu hóa mạnh trong không gian đặc trưng sau khi chuẩn hóa. Điều này cho thấy SVM đã tìm được ranh giới phân lớp hiệu quả khi sử dụng kernel RBF.

Trên tập Test, mô hình đạt Accuracy 98%, một con số rất cao đối với bài toán phân loại đa lớp như Car Evaluation. Các lớp unacc, good và vgood đạt F1-score cao,

cho thấy mô hình phân biệt rõ được sự khác biệt giữa các nhóm này. Một số nhầm lẫn nhỏ xuất hiện giữa hai lớp acc và good – điều hoàn toàn hợp lý do sự tương đồng về mặt ngữ nghĩa và đặc trưng giữa hai lớp.

Biểu đồ SHAP hình 2.11 cung cấp thêm thông tin diễn giải mô hình. Kết quả SHAP khẳng định:

- Safety là đặc trưng đóng vai trò quyết định quan trọng nhất.
- Buying và persons cũng có mức ảnh hưởng lớn đến dự đoán.
- Doors có mức đóng góp thấp hơn và chủ yếu mang tính hỗ trợ.

Nhờ việc sử dụng SHAP, mô hình SVM – vốn thường được xem là khó giải thích – trở nên minh bạch hơn, giúp người dùng hiểu rõ lý do đăng sau các dự đoán. Điều này đặc biệt hữu ích trong các bài toán đánh giá như Car Evaluation, nơi việc giải thích quyết định mang ý nghĩa quan trọng trong ứng dụng thực tế.

2.5. So sánh đánh giá

Khi so sánh bốn thuật toán gồm Decision Tree, Random Forest, Gradient Boosting và Support Vector Machine (SVM), có thể nhận thấy mỗi mô hình đại diện cho một hướng tiếp cận khác nhau, đồng thời thể hiện ưu – nhược điểm riêng biệt trong bài toán Car Evaluation.

Decision Tree là mô hình đơn giản và dễ giải thích nhất, nhưng cũng dễ rơi vào tình trạng overfitting do quá bám sát dữ liệu huấn luyện. Điều này khiến hiệu suất của nó thấp nhất trong bốn mô hình, đặc biệt khi dữ liệu có quan hệ phức tạp và phi tuyến.

Random Forest, nhờ cơ chế bagging và việc sử dụng nhiều cây quyết định, đã giải quyết tốt nhược điểm của Decision Tree. Mô hình cho kết quả ổn định hơn, giảm được nhiều và cải thiện đáng kể độ chính xác so với một cây đơn lẻ.

Gradient Boosting tiếp tục nâng cao hiệu suất bằng cách học tuần tự từ sai số của các mô hình trước đó. Khả năng mô hình hóa quan hệ phi tuyến rất mạnh giúp Gradient Boosting thường đạt kết quả tốt nhất trong nhóm tree-based, đồng thời vẫn cho phép phân tích mức độ quan trọng của từng đặc trưng thông qua Feature Importance.

SVM khác biệt hoàn toàn so với ba mô hình trên vì không dựa trên cấu trúc cây. Thay vào đó, SVM tìm ranh giới phân lớp tối ưu bằng cách tối đa hóa khoảng cách giữa các nhóm dữ liệu. Với kernel RBF, SVM có thể mô hình hóa quan hệ phi tuyến tốt và đạt hiệu suất cao trên dữ liệu đã được chuẩn hóa. Tuy nhiên, hạn chế của SVM là chi phí tính toán cao, đặc biệt với tập dữ liệu lớn, và khả năng giải thích gần như không có. Việc giải thích quyết định của mô hình đòi hỏi các phương pháp hỗ trợ như SHAP, gây phức tạp trong phân tích.

Tổng thể, cả bốn mô hình đều đem lại góc nhìn bổ sung cho bài toán: Decision Tree đơn giản nhưng dễ dàng khai thác; Random Forest ổn định và mạnh mẽ hơn; Gradient Boosting vượt trội về hiệu suất và khả năng mô hình hóa phi tuyến; trong khi SVM đóng vai trò đối trọng quan trọng với nền tảng lý thuyết mạnh mẽ và hiệu quả cao, dù hạn chế về tốc độ và khả năng diễn giải.

CHƯƠNG 3: CÔNG CỤ DỰ ĐOÁN MỨC ĐỘ PHÙ HỢP CỦA XE DỰA TRÊN ĐẶC ĐIỂM KỸ THUẬT

3.1. Mục tiêu và vai trò của ứng dụng demo

Ứng dụng demo đóng vai trò cầu nối giữa kết quả thí nghiệm (mô hình học máy) và tính ứng dụng thực tế. Để thực sự thực hiện được các mục tiêu đã liệt kê, cần phân tách các mục tiêu này thành các yêu cầu kỹ thuật, tiêu chí đánh giá và kế hoạch kiểm chứng. Phần dưới đây trình bày từng mục tiêu, vì sao nó quan trọng và cách triển khai/đo lường cụ thể.

3.2. Quy trình xử lý đầu vào của demo

Ứng dụng được xây dựng với luồng xử lý rõ ràng và nhất quán, đảm bảo mô hình nhận đúng dạng dữ liệu cần thiết:

Người dùng nhập thông tin: Người dùng chọn các thuộc tính kỹ thuật của xe như giá mua, chi phí bảo trì, số cửa, sức chứa, kích thước khoang hành lý và mức an toàn thông qua giao diện.

Chuẩn hóa giá trị đầu vào: Trước khi gửi đến mô hình, backend tiến hành kiểm tra sự hợp lệ của dữ liệu, chuyển các giá trị dạng chữ về dạng số theo đúng mã hóa đã sử dụng khi huấn luyện mô hình, đảm bảo trùng khớp hoàn toàn với pipeline tiền xử lý

Điều này đảm bảo không xảy ra sai lệch giữa dữ liệu huấn luyện và dữ liệu người dùng nhập.

Áp dụng mô hình đã huấn luyện: Backend tải mô hình được lưu trước đó và thực hiện dự đoán thông qua hàm dự đoán duy nhất.

Do mô hình đã được tối ưu hóa từ các bước huấn luyện, thời gian phản hồi nhanh và ổn định.

Trả về kết quả dự đoán: Kết quả trả về gồm: mức độ phù hợp của xe: unacc, acc, good hoặc vgood, thông tin tổng hợp về các thuộc tính người dùng đã nhập.

Quy trình xử lý được thiết kế nhẹ và tối giản để đảm bảo tốc độ và độ chính xác.

3.3. Kiến trúc triển khai

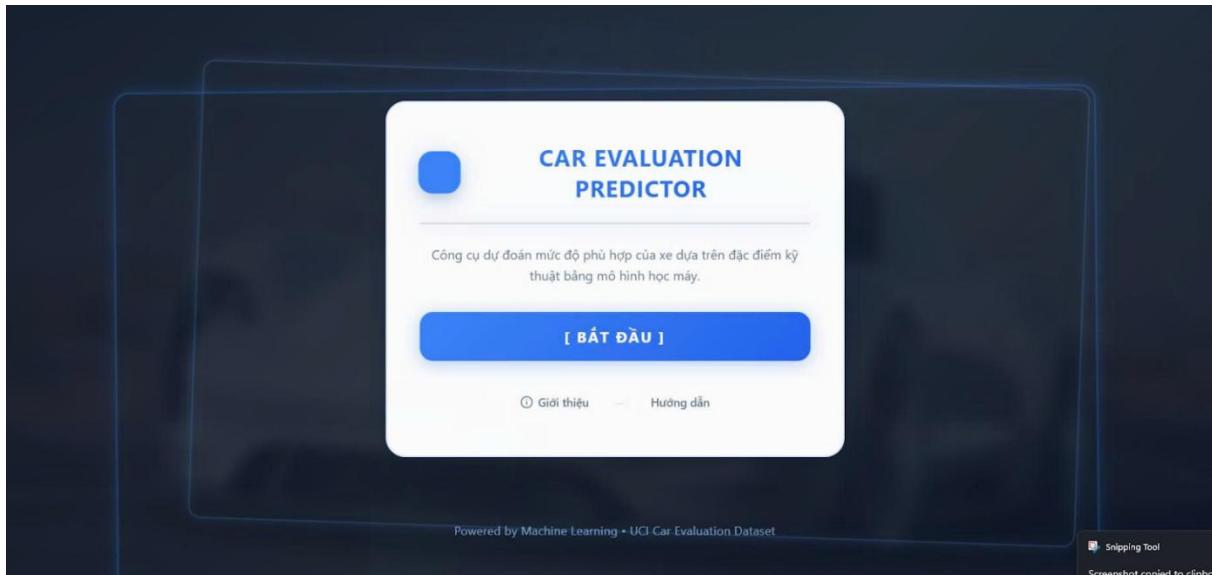
Kiến trúc triển khai của ứng dụng demo được thiết kế theo mô hình client-server tách biệt nhằm tối ưu hóa khả năng mở rộng, linh hoạt và tốc độ phản hồi.

Đây là kiến trúc phổ biến trong các hệ thống áp dụng mô hình học máy vào ứng dụng thực tế, đặc biệt khi cần chia tách rõ ràng giữa phần giao diện người dùng và phần xử lý mô hình. Việc tách biệt này giúp bảo đảm cả hai phần có thể phát triển độc lập nhưng vẫn giao tiếp hiệu quả với nhau thông qua giao thức HTTP.

Phân tích kiến trúc Frontend (TypeScript / Web App): Phần giao diện được phát triển bằng TypeScript kết hợp với kiến trúc SPA (Single-page Application), mang lại các đặc điểm sau: Tương tác trực tiếp và thời gian thực, luồng gửi yêu cầu và nhận kết quả và sự độc lập của giao diện.

Backend được triển khai bằng FastAPI – một framework Python hiện đại, nhanh nhẹ, phù hợp cho các ứng dụng có mô hình học máy. Phần backend đảm nhiệm các nhiệm vụ trọng yếu: Xử lý dữ liệu đầu vào, gọi mô hình và thực hiện dự đoán, trả kết quả qua API JSON.

3.4. Mô tả hình minh họa giao diện



Hình 3.1 Màn hình Landing Page của ứng dụng Car Evaluation Predictor

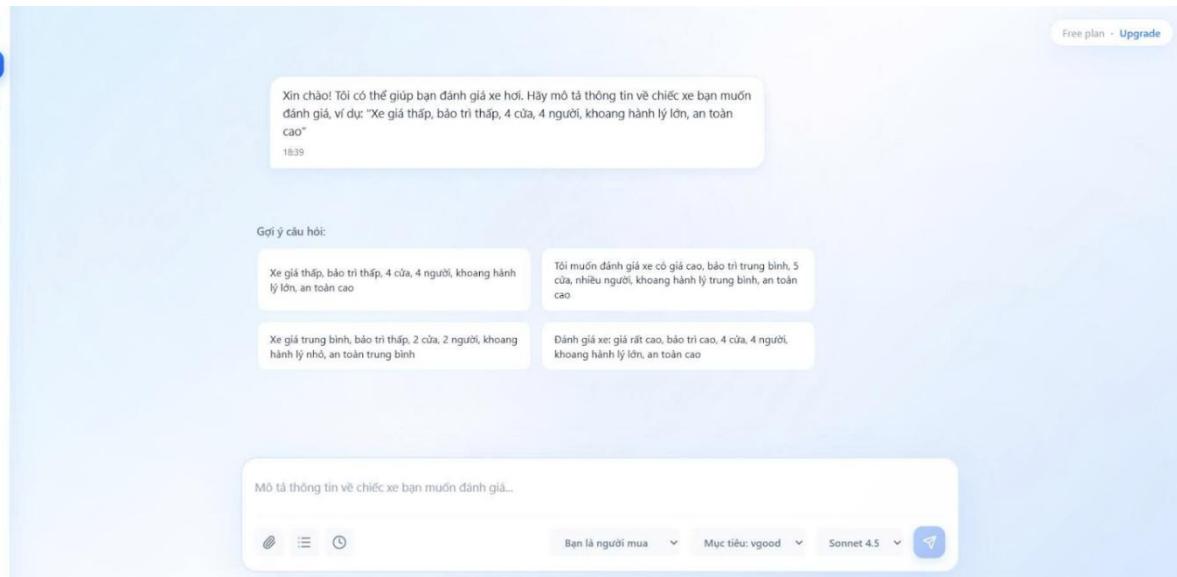
Màn hình chào được thiết kế với phong cách tối giản, hiện đại và nhấn mạnh vào nhận diện của ứng dụng. Vùng giao diện trung tâm đặt trong một khung nổi (floating card), giúp thu hút sự tập trung của người dùng ngay khi truy cập. Nội dung chính của màn hình gồm:

Tiêu đề “CAR EVALUATION PREDICTOR” thể hiện rõ chức năng của ứng dụng.

Mô tả ngắn gọn: giới thiệu công cụ dự đoán mức độ phù hợp của xe dựa trên đặc điểm kỹ thuật bằng mô hình học máy.

Nút “BẮT ĐẦU”: đóng vai trò điều hướng người dùng sang trang đánh giá xe.

Màn hình này giúp tạo ấn tượng chuyên nghiệp, đồng thời đảm bảo người dùng hiểu rõ vai trò của ứng dụng trước khi bắt đầu sử dụng.



Hình 3.2 Giao diện nhập thông tin và gợi ý câu hỏi của hệ thống

Đây là giao diện chính cho phép người dùng tương tác với hệ thống dự đoán. Giao diện được chia thành các khu vực chức năng rõ ràng:

Vùng hội thoại mô phỏng AI: Tại đây, hệ thống đưa ra tin nhắn giới thiệu và hướng dẫn người dùng mô tả chiếc xe muốn đánh giá. Điều này giúp trải nghiệm sử dụng trở nên tự nhiên và trực quan hơn, giống như đang trò chuyện với một trợ lý ảo.

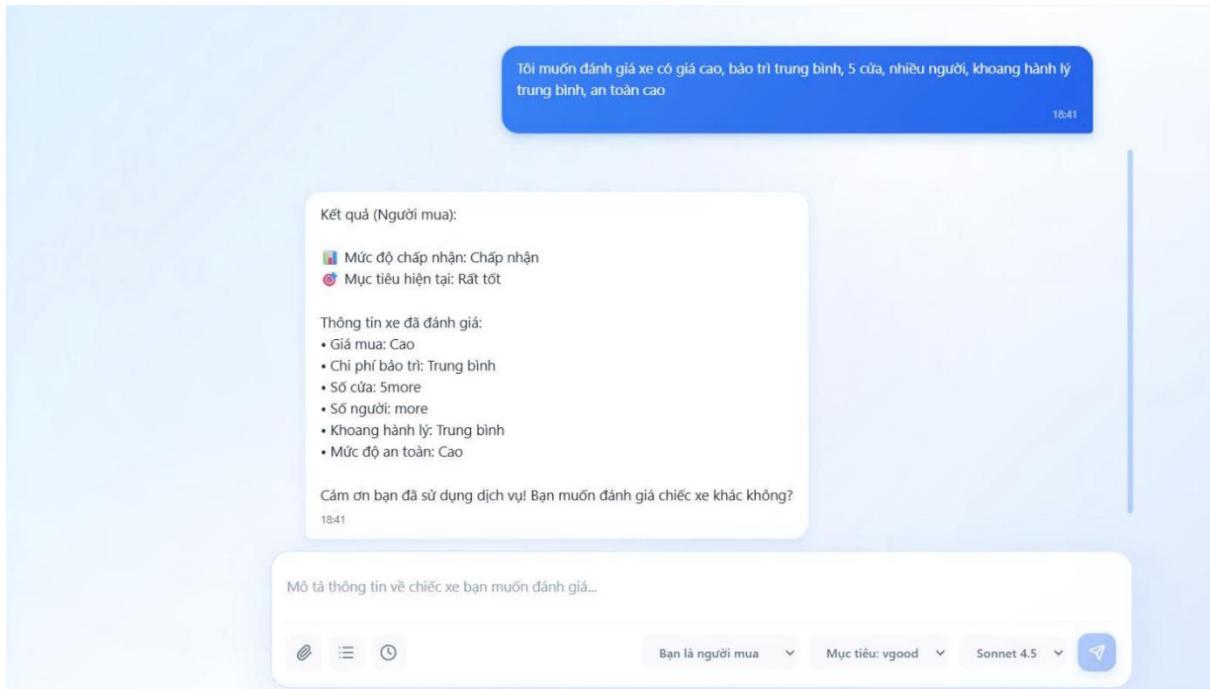
Danh sách gợi ý câu hỏi: Hệ thống hiển thị một loạt các mô tả xe mẫu giúp người dùng hiểu nhanh cách nhập dữ liệu. Những gợi ý này dựa trên cấu trúc thuộc tính có trong dataset, tạo điều kiện để người dùng thao tác nhanh mà không cần suy nghĩ nhiều về cú pháp.

Vùng nhập thông tin đánh giá xe: Bên dưới là trường nhập liệu nơi người dùng có thể mô tả chiếc xe theo ngôn ngữ tự nhiên hoặc lựa chọn thông tin thông qua các dropdown.

Đồng thời, giao diện còn hiển thị ba lựa chọn quan trọng: Loại người dùng (Buyer / Seller / Inspector), mục tiêu đánh giá (acc, good, vgood,...)

Tổng thể, màn hình này thể hiện đầy đủ quy trình nhập liệu → gửi → nhận kết quả → diễn giải, đảm bảo trải nghiệm mạch lạc và trực quan.

Giao diện ứng dụng demo được thiết kế theo hướng tối giản, tập trung vào trải nghiệm người dùng và đảm bảo việc nhập liệu diễn ra nhanh chóng, chính xác. Các thuộc tính kỹ thuật của xe được bố trí dưới dạng danh sách chọn (dropdown), giúp người dùng chỉ có thể chọn các giá trị hợp lệ, đồng nhất với dữ liệu huấn luyện. Cách bố trí này làm giảm đáng kể lỗi nhập liệu — một vấn đề thường xảy ra trong các ứng dụng dự đoán phụ thuộc vào đầu vào rời rạc.



Hình 3.3 Giao diện ứng dụng demo dự đoán mức độ phù hợp của xe

Ngay khi người dùng hoàn tất lựa chọn và nhấn nút "Dự đoán", giao diện gửi dữ liệu đến backend và hiển thị kết quả phân loại ngay trên màn hình. Kết quả được trình bày rõ ràng dưới dạng nhãn (unacc, acc, good, vgood) kèm bảng tóm tắt các thuộc tính đã nhập nhằm giúp người dùng dễ dàng kiểm tra lại thông tin trước khi đưa ra quyết định.

Giao diện đồng thời thể hiện một phần luồng xử lý của hệ thống: dữ liệu đầu vào chuẩn hóa → gửi qua API → mô hình dự đoán → phản hồi kết quả → hiển thị trực quan cho người dùng.

Thiết kế trực quan, dễ hiểu, không yêu cầu người dùng có kiến thức về học máy. Điều này minh chứng rằng mô hình trong đề tài hoàn toàn có thể được triển khai trong thực tế và trở thành một công cụ hỗ trợ ra quyết định cho người dùng cuối.

3.5. Tính năng Nâng cao: Phân tích Đề xuất (Prescriptive Analysis)

Chức năng này được xây dựng dành riêng cho vai trò Nhà sản xuất (Inspector), với mục tiêu hỗ trợ trả lời câu hỏi: “Cần thay đổi những thuộc tính nào của xe, với chi phí tối thiểu, để đạt được mức đánh giá cao hơn?”. Khác với các mô hình phân loại truyền thống chỉ dừng lại ở việc dự đoán nhãn chất lượng, hệ thống được mở rộng theo hướng hỗ trợ ra quyết định, giúp nhà sản xuất chủ động xác định phương án cải tiến sản phẩm hiệu quả.

Về cơ chế hoạt động, hệ thống không chỉ sử dụng mô hình học máy (như SVM hoặc Gradient Boosting) để dự đoán mức đánh giá của xe, mà còn kết hợp với thuật toán Tìm kiếm theo chiều rộng (Breadth-First Search – BFS) nhằm khám phá không gian thuộc tính. Mỗi trạng thái trong quá trình tìm kiếm tương ứng với một cấu hình thuộc tính cụ thể của xe, trong khi mỗi bước chuyển trạng thái thể hiện một thay đổi nhỏ trên một thuộc tính. Mô hình học máy đóng vai trò như một hàm đánh giá, được sử dụng để dự đoán mức phân loại của từng trạng thái mới được sinh ra trong quá trình tìm kiếm.

Quá trình tìm kiếm được thiết kế dựa trên hai nguyên tắc tối ưu chính. Thứ nhất, BFS đảm bảo ưu tiên số bước thay đổi tối thiểu. Nhờ đặc tính duyệt theo từng mức độ sâu, thuật toán luôn tìm ra giải pháp với số lượng thay đổi thuộc tính ít nhất để đạt được mục tiêu (ví dụ từ mức Acceptable lên Good hoặc Very Good). Nguyên tắc này phản ánh trực tiếp yêu cầu tối ưu chi phí trong thực tế sản xuất, khi mỗi thay đổi thuộc tính đều gắn liền với chi phí cải tiến.

Thứ hai, quá trình sinh các trạng thái kế tiếp được ưu tiên theo mức độ quan trọng của thuộc tính. Thứ tự thử thay đổi các thuộc tính không mang tính ngẫu nhiên mà dựa trên kết quả phân tích Feature Importance từ mô hình học máy. Những thuộc tính có ảnh hưởng lớn đến kết quả phân loại, chẳng hạn như safety, sẽ được xem xét trước các thuộc

tính có mức ảnh hưởng thấp hơn như doors. Cách tiếp cận này giúp giảm số lượng trạng thái cần duyệt, đồng thời tăng tính hợp lý và thực tiễn của các đề xuất cải tiến.

Quy trình xử lý bắt đầu khi người dùng lựa chọn vai trò Inspector và cấu hình xe hiện tại được hệ thống đánh giá ở mức thấp (unacc hoặc acc). Khi đó, mô-đun BFS được kích hoạt để phân tích các khả năng cải tiến. Thuật toán lần lượt duyệt qua các cấu hình thay đổi thuộc tính theo thứ tự ưu tiên và sử dụng mô hình học máy để dự đoán kết quả của từng cấu hình mới. Quá trình tìm kiếm dừng lại ngay khi đạt được mức đánh giá mục tiêu (good hoặc vgood), đảm bảo rằng giải pháp thu được là phương án cải tiến tối ưu về số bước thay đổi.

Kết quả đầu ra cho Nhà sản xuất được trình bày dưới dạng một báo cáo đề xuất cải tiến cụ thể. Ví dụ, hệ thống có thể chỉ ra rằng để nâng cấp chiếc xe từ mức Acceptable lên Very Good, cần thực hiện hai bước thay đổi tối thiểu, bao gồm nâng cấp mức độ an toàn từ Med lên High và giảm chi phí bảo trì từ VHigh xuống High. Mỗi bước thay đổi đều mang ý nghĩa thực tiễn rõ ràng, giúp nhà sản xuất dễ dàng đánh giá và triển khai trong thực tế.



Hình 3.4 Giao diện đánh giá và cải tiến đề xuất cho nhà sản xuất

Về giá trị khoa học dữ liệu, chức năng này đã chuyển đổi mô hình phân loại thụ động thành một công cụ hỗ trợ chiến lược chủ động. Thay vì chỉ cung cấp kết quả đánh giá, hệ thống còn đưa ra các gợi ý cải tiến có thể giải thích được, kết hợp chặt chẽ giữa

mô hình học máy và thuật toán tìm kiếm. Cách tiếp cận này không chỉ nâng cao tính ứng dụng của mô hình mà còn tăng khả năng giải thích và hỗ trợ ra quyết định trong bối cảnh sản xuất và tối ưu hóa chất lượng sản phẩm.

Mô tả: Khi người dùng chọn vai trò Inspector và xe hiện tại đạt mức thấp (unacc hoặc acc), giao diện sẽ kích hoạt phân tích BFS.

Đầu ra cho Nhà sản xuất: Hệ thống sẽ hiển thị một báo cáo:

"Để nâng cấp chiếc xe này từ 'Acceptable' lên 'Very Good', bạn cần thực hiện 2 bước thay đổi tối thiểu sau:

1. Nâng cấp **Safety** từ 'Med' lên 'High'.
2. Giảm chi phí **Maint** từ 'VHigh' xuống 'High'."

Giá trị KPDL: Chuyển đổi mô hình phân loại thụ động thành công cụ hỗ trợ chiến lược chủ động, giúp nhà sản xuất tối ưu hóa chi phí và đạt được mục tiêu chất lượng sản phẩm.

3.6. Những hạn chế

Mặc dù đề tài đã đạt được các mục tiêu đề ra, vẫn tồn tại một số hạn chế xuất phát từ dữ liệu, mô hình, quy trình đánh giá cũng như phạm vi triển khai ứng dụng demo. Các hạn chế này ảnh hưởng nhất định đến khả năng tổng quát hóa và mức độ ứng dụng của hệ thống.

Giao diện mới dừng ở mức minh họa chức năng: Demo hiện chỉ thể hiện được luồng nhập liệu – gửi yêu cầu – nhận kết quả. Chưa có: lưu lịch sử dự đoán, tính năng giải thích mô hình (explainability), biểu đồ trực quan hóa kết quả, đề xuất xe phù hợp dựa trên bối cảnh người dùng.

Chưa tích hợp khả năng xử lý lỗi nâng cao: Một số trường hợp mô tả không chuẩn hoặc nhập liệu lệch chuẩn có thể gây lỗi hoặc trả về thông báo chưa trực quan.

Chưa tối ưu hiệu suất và triển khai thực tế: Demo hoạt động tốt trong môi trường chạy cục bộ nhưng chưa được triển khai trên server thực hoặc cloud để kiểm chứng khả năng đáp ứng nhiều người dùng.

CHƯƠNG 4: KẾT LUẬN

4.1. Kết quả đạt được

Qua quá trình nghiên cứu và thực nghiệm trên bộ dữ liệu Car Evaluation, đề tài đã đạt được nhiều kết quả quan trọng cả về mặt kỹ thuật lẫn mặt ứng dụng. Trước hết, nhóm đã xây dựng thành công hệ thống tiền xử lý dữ liệu bao gồm mã hóa thuộc tính, chia tập dữ liệu và chuẩn hóa (đối với mô hình SVM). Tiếp theo, nhóm đã triển khai và đánh giá bốn thuật toán phân loại gồm Decision Tree, Random Forest, Gradient Boosting và Support Vector Machine. Trong đó, Gradient Boosting cho kết quả tốt nhất, đạt độ chính xác cao và giữ mức F1-score ổn định ở tất cả các lớp. SVM cũng đạt hiệu suất cao và hoạt động ổn định khi dữ liệu được chuẩn hóa, đóng vai trò quan trọng trong việc so sánh với các mô hình ensemble. Ngoài ra, nhóm đã xây dựng được hệ thống đánh giá mô hình dựa trên các chỉ số Accuracy, Precision, Recall, F1-score, Confusion Matrix và biểu đồ Feature Importance/SHAP. Cuối cùng, nhóm cũng xây dựng được một bản demo dự đoán mức độ phù hợp của xe dựa trên các thuộc tính đầu vào, hỗ trợ người dùng tương tác trực quan với mô hình.

4.2. Những hạn chế

Mặc dù đề tài đã đạt được các mục tiêu đề ra, vẫn tồn tại một số hạn chế xuất phát từ dữ liệu, mô hình, quy trình đánh giá cũng như phạm vi triển khai ứng dụng demo. Các hạn chế này ảnh hưởng nhất định đến khả năng tổng quát hóa và mức độ ứng dụng của hệ thống.

4.2.1 Hạn chế về bộ dữ liệu.

Quy mô dữ liệu nhỏ và mang tính tổng hợp: Bộ dữ liệu Car Evaluation chỉ gồm 1.728 mẫu, được xây dựng theo mô hình DEX mang tính mô phỏng, không phải dữ liệu thực tế từ thị trường xe. Điều này khiến mô hình học được logic do bộ dữ liệu quy định, nhưng chưa chắc phản ánh đầy đủ biến động trong thực tế.

Thiếu đa dạng thuộc tính kỹ thuật: Dataset chỉ gồm sáu thuộc tính cơ bản (giá mua, chi phí bảo trì, số cửa, chỗ ngồi, khoang hành lý, an toàn). Các yếu tố quan trọng khác như mức tiêu hao nhiên liệu, công nghệ an toàn chủ động, loại động cơ, hệ truyền động, thương hiệu, độ tin cậy... chưa được đưa vào. Điều này hạn chế khả năng mô hình hóa các yếu tố phức tạp trong quá trình đánh giá xe thực tế.

Phân bố lớp mêt cân bằng: Một số lớp như unacc xuất hiện nhiều hơn đáng kể so với các lớp như good hoặc vgood. Điều này khiến mô hình có xu hướng nghiêng về lớp phổ biến, gây giảm hiệu suất ở các lớp ít dữ liệu.

4.2.2 Hạn chế về mô hình học máy.

Khả năng diễn giải của mô hình còn hạn chế: Các mô hình mạnh như SVM và Gradient Boosting cho độ chính xác cao nhưng khó diễn giải lý do đưa ra dự đoán. Điều này gây khó khăn khi mô hình được sử dụng trong môi trường cần tính minh bạch (ví dụ tư vấn kỹ thuật, hỗ trợ khách hàng).

Hiệu suất của một số mô hình chưa ổn định: Decision Tree và Random Forest cho thấy dao động đáng kể trong độ chính xác giữa train và test set, phản ánh mức độ overfitting và ảnh hưởng của phân bố lớp. Điều này làm giảm độ tin cậy khi triển khai trong môi trường thực tế.

Chưa tối ưu siêu tham số chuyên sâu: Do phạm vi đê tài và thời gian có hạn, các mô hình chưa được chạy grid search hoặc Bayesian optimization toàn diện. Vì vậy kết quả hiện tại có thể chưa phải là hiệu suất tối ưu nhất.

4.2.3. Hạn chế trong đánh giá và kiểm thử mô hình

Thiếu tập dữ liệu thực tế để kiểm chứng: Kết quả đánh giá chủ yếu dựa trên việc chia train-test trong cùng một dataset. Không có một tập dữ liệu ngoại sinh (external test set) để đo mức độ tổng quát hóa trong môi trường thực tế.

Chưa kiểm thử đầy đủ trên dữ liệu người dùng tự nhập: Mặc dù demo hoạt động ổn định, nhưng chưa có một bộ test case lớn từ người dùng thực. Điều này khiến khó đánh giá mô hình trong các tình huống biên (edge-case) hoặc mô tả không chuẩn.

4.2.4. Hạn chế trong ứng dụng demo

Giao diện mới dùng ở mức minh họa chức năng: Demo hiện chỉ thể hiện được luồng nhập liệu – gửi yêu cầu – nhận kết quả. Chưa có: lưu lịch sử dự đoán, tính năng giải thích mô hình (explainability), biểu đồ trực quan hóa kết quả, đê xuất xe phù hợp dựa trên bối cảnh người dùng.

Chưa tích hợp khả năng xử lý lỗi nâng cao: Một số trường hợp mô tả không chuẩn hoặc nhập liệu lệch chuẩn có thể gây lỗi hoặc trả về thông báo chưa trực quan.

Chưa tối ưu hiệu suất và triển khai thực tế: Demo hoạt động tốt trong môi trường chạy cục bộ nhưng chưa được triển khai trên server thực hoặc cloud để kiểm chứng khả năng đáp ứng nhiều người dùng.

4.2.5. Hạn chế về phạm vi đề tài

Chỉ tập trung vào phân loại mức độ phù hợp: Đề tài chưa mở rộng sang các chức năng thực tiễn hơn như: so sánh hai mẫu xe, đề xuất xe phù hợp theo ngân sách, dự đoán giá trị xe theo thông số kỹ thuật, phân tích hành vi người dùng trong lựa chọn xe.

Chưa có đánh giá từ người dùng hoặc chuyên gia: Đề tài chưa thực hiện khảo sát người dùng thực hoặc chuyên gia tư vấn xe để đánh giá mức độ hợp lý của kết quả dự đoán.

4.3. Hướng phát triển trong tương lai

Dựa trên các hạn chế đã được phân tích, đề tài có thể được mở rộng theo nhiều hướng nhằm tăng tính ứng dụng, cải thiện độ chính xác của mô hình và nâng cấp trải nghiệm người dùng. Các hướng phát triển đề xuất dưới đây phù hợp với lộ trình tiến hóa của một hệ thống dự đoán ứng dụng trong thực tế.

4.3.1 Mở rộng và nâng cao chất lượng dữ liệu

Thu thập dữ liệu thực tế từ thị trường xe: Việc bổ sung dữ liệu thật (từ showroom, hãng xe, website thương mại hoặc lịch sử giao dịch) sẽ giúp mô hình phản ánh đúng nhu cầu thực tế hơn, giảm sự lệ thuộc vào bộ dữ liệu tổng hợp Car Evaluation.

Bổ sung thêm thuộc tính kỹ thuật quan trọng: Một hệ thống đánh giá xe thực tế cần nhiều yếu tố hơn 6 thuộc tính hiện tại, ví dụ: công nghệ an toàn (ABS, ESP, ADAS), mức tiêu hao nhiên liệu, dung tích động cơ, loại hộp số, mô-men xoắn, thương hiệu, năm sản xuất, độ bền linh kiện, chi phí sửa chữa, mức mất giá theo thời gian.

Việc mở rộng thuộc tính sẽ giúp mô hình học sâu hơn về đặc điểm của từng loại.

4.3.2. Cải thiện và tối ưu hóa mô hình học máy

Thử nghiệm các mô hình tiên tiến hơn: Một số thuật toán có thể thay thế hoặc bổ sung cho các mô hình hiện tại: XGBoost, LightGBM, CatBoost (cho bài toán phân loại mạnh, tốc độ nhanh), Mạng nơ-ron (DNN) nếu số lượng dữ liệu được mở rộng, Thuật toán Explainable AI (XAI) kết hợp với tree-based để tăng tính minh bạch.

Tối ưu siêu tham số (Hyperparameter Tuning): Áp dụng: Grid Search, Random Search, Bayesian Optimization (Optuna, Hyperopt)
nhằm tối ưu hóa mô hình cả về độ chính xác lẫn tốc độ.

Xây dựng pipeline huấn luyện – đánh giá tự động: Tạo một quy trình tự động gồm tiền xử lý → huấn luyện → đánh giá → lưu mô hình → triển khai (ML workflow). Điều này giúp giảm sai sót thủ công và dễ tái sử dụng.

4.3.4 Nâng cấp ứng dụng demo thành hệ thống hoàn chỉnh

Tích hợp tính năng giải thích mô hình (Model Explainability) Thêm SHAP, LIME hoặc Feature Importance để người dùng hiểu được lý do mô hình đưa ra kết quả (vì sao xe này chỉ được đánh giá acc?).

Lưu lịch sử đánh giá và xuất báo cáo: Ứng dụng nên cho phép lưu lại các dự đoán trước đó, xuất file PDF/CSV, theo dõi lịch sử đánh giá theo thời gian.

Điều này hữu ích cho showroom, đại lý, hoặc người bán xe chuyên nghiệp.

Thêm chức năng gợi ý xe phù hợp: Dựa trên nhu cầu người dùng (giá, an toàn, số chỗ, quãng đường di chuyển), hệ thống có thể gợi ý danh sách các mẫu xe phù hợp nhất, bảng so sánh hai hoặc nhiều mẫu xe.

Xây dựng phiên bản mobile hoặc API công khai: Cho phép tích hợp vào ứng dụng quản lý xe, ứng dụng showroom hoặc chatbot hỗ trợ khách hàng.

Triển khai ứng dụng trên môi trường thực: Đưa demo lên server/cloud (AWS, Azure, Render...) giúp kiểm thử hiệu năng trong điều kiện tải thực tế và hỗ trợ nhiều người dùng cùng lúc.

4.3.4 Cải thiện khả năng kiểm thử và giám sát mô hình sau triển khai

Xây dựng bộ kiểm thử mở rộng: kiểm thử dữ liệu biên, kiểm thử sai lệch input, kiểm thử hiệu năng (latency test).

Theo dõi drift dữ liệu theo thời gian: Thiết lập hệ thống log để phát hiện khi phân bố dữ liệu người dùng thay đổi so với dữ liệu huấn luyện, từ đó đưa ra cảnh báo và kế hoạch huấn luyện lại (retraining).

Đánh giá theo phản hồi thực tế của người dùng: Tích hợp cơ chế “chấm điểm kết quả dự đoán” để thu thập phản hồi, giúp cải thiện mô hình trong các lần retrain sau này.

4.4. Bảng phân công nhiệm vụ trong nhóm

Dưới đây là bảng phân công nhiệm vụ giữa các thành viên trong nhóm trong suốt quá trình thực hiện đề tài:

Bảng 4.1 Bảng phân công nhiệm vụ

Thành viên	Nhiệm vụ
Tất cả thành viên	<ul style="list-style-type: none"> - Tìm hiểu đề tài và bộ dữ liệu Car Evaluation. - Xây dựng kết quả đánh giá và hoàn thiện nội dung báo cáo.
Phạm Thị Ngọc Oanh	<ul style="list-style-type: none"> - Tìm hiểu đề tài, đặt vấn đề, thu thập tài liệu tham khảo. - Viết api tích hợp model và xử lý nội dung trả lời của chatbot
Trần Phương Anh	<ul style="list-style-type: none"> - Phân tích bộ dữ liệu Car Evaluation. - Thực hiện tiền xử lý dữ liệu
Đỗ Văn Thành Được	<ul style="list-style-type: none"> - Xây dựng và huấn luyện các mô hình Decision Tree, Random Forest. - Thực hiện trực quan hóa Feature Importance và đánh giá mô hình.
Lê Đình Khôi	<ul style="list-style-type: none"> - Triển khai mô hình Gradient Boosting, SVM, thực hiện chuẩn hóa dữ liệu và phân tích kết quả bằng SHAP - So sánh giữa các mô hình và thảo luận kết quả.
Ôn Gia Bảo	<ul style="list-style-type: none"> - Xây dựng giao diện website - Tích hợp các chức năng cơ bản của chatbot

TÀI LIỆU THAM KHẢO

- [1] Datasets - UCI Machine Learning Repository, truy cập vào 10/12/2025,
<http://archive.ics.uci.edu/datasets>
- [2] Car Evaluation, truy cập vào 10/12/2025,
https://www.iitp.ac.in/~arijit/dokuwiki/lib/exe/fetch.php?media=courses:2017:c_s551:13_report.pdf
- [3] ML Ready Car Evaluation Dataset - Kaggle, truy cập vào 10/12/2025,
<https://www.kaggle.com/datasets/arshmankhalid/ml-ready-car-evaluation-dataset>
- [4] Car Evaluation dataset classification - Kaggle, truy cập vào 10/12/2025,
<https://www.kaggle.com/code/johnmantios/car-evaluation-dataset-classification>
- [5] Car Evaluation - UCI Machine Learning Repository, truy cập vào 10/12/2025,
<https://archive-beta.ics.uci.edu/dataset/19/car+evaluation/files>
- [6] sharmaroshan/Car_Evaluation: Evaluating a Car based on some popular attributes which could be beneficial in decision making while purchasing a Car, Who do not have enough knowledge about Cars. - GitHub, truy cập vào 10/12/2025,
https://github.com/sharmaroshan/Car_Evaluation
- [7] Car Evaluation ML Model - Kaggle, truy cập vào 10/12/2025,
<https://www.kaggle.com/code/kanyianalyst/car-evaluation-ml-model>
- [8] Exploratory Analysis of Car Evaluation Dataset with SQL | by Noshin Nawar Neha - Medium, truy cập vào 10/12/2025,
<https://medium.com/noshin-nawar-neha-data-analytics-blogs/exploratory-analysis-of-car-evaluation-dataset-with-sql-d67e6d93262e>
- [9] Using Decision Tree Method for Car Selection Problem | by Hafidz Jazuli - Medium, truy cập vào 10/12/2025,
<https://medium.com/machine-learning-guy/using-decision-tree-method-for-car-selection-problem-5272675451f9>
- [10] A Comparative Study of Categorical Variable Encoding Techniques for Neural Network Classifiers - International Journal of Computer Applications, truy cập vào 10/12/2025,
<https://www.ijcaonline.org/archives/volume175/number4/potdar-2017-ijca-915495.pdf>
- [11] car-evaluation-ML.ipynb - GitHub, truy cập vào 10/12/2025,
<https://github.com/MatthewCarterIO/car-evaluation-ML/blob/master/car-evaluation-ML.ipynb>
- [12] Categorical Data Encoding Techniques in Machine Learning - GeeksforGeeks, truy cập vào 10/12/2025,
<https://www.geeksforgeeks.org/machine-learning/categorical-data-encoding-techniques-in-machine-learning/>

- [13] Encoding Categorical Variables: One-Hot vs. Ordinal - Business Analytics Institute, truy cập vào 10/12/2025,
<https://businessanalyticsinstitute.com/encoding-categorical-variables-one-hot-vs-ordinal/>
- [14] Class distribution using the car evaluation dataset. - ResearchGate, truy cập vào 10/12/2025,
https://www.researchgate.net/figure/Class-distribution-using-the-car-evaluation-dataset_tb1_370225342