TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI THÀNH PHỐ HỒ CHÍ MINH

VIỆN CÔNG NGHỆ THÔNG TIN VÀ ĐIỆN, ĐIỆN TỬ

BÁO CÁO
BÀI TẬP THỰC HÀNH CUỐI KHÓA

Giảng viên hướng dẫn: PhD. Nguyễn Thị Khánh Tiên

Nhóm 9 thực hiện:

2251120184	Đỗ Nguyễn Thiên
09630501002	Lê Triệu Duy
2251120137	Nguyễn Ngọc Minh Châu
2251120164	Dương Đặng Xuân Khanh
2251120173	Nguyễn Thị Bích Nhân
2251120174	Nguyễn Thị Nhi
2251120175	Đặng Tiến Pháp

Tp. Hồ Chí Minh, Tháng 11 năm 2025

MỤC LỤC

CHƯƠNG 1. TÓM TẮT (ABSTRACT)

Báo cáo tập trung mục tiêu xây dựng hệ thống định giá bất động sản tự động dựa trên bộ dữ liệu **Bengaluru House Price từ Kaggle**, nhằm khắc phục tính chủ quan của phương pháp truyền thống và minh bạch hóa thị trường. Trọng tâm là phát triển mô hình hồi quy để dự đoán chính xác giá nhà dựa trên các đặc tính thực tế, đồng thời tối ưu hóa sai số dự báo nhằm hỗ trợ tốt nhất cho người mua và người bán. Sau quá trình làm sạch dữ liệu và so sánh hiệu quả giữa các thuật toán, nhóm đã quyết định lựa chọn **Linear Regression** là mô hình chính thức nhờ ưu thế về tính ổn định và khả năng giải thích tốt hơn so với Random Forest. Kết quả nghiên cứu được hiện thực hóa thành công qua một ứng dụng web Full-stack (sử dụng **Flask và Streamlit**), đặc biệt có sự tích hợp **API Google Gemini** để không chỉ đưa ra mức giá dự đoán tin cậy mà còn đóng vai trò là chatbot định giá bất động sản nhanh chóng và thông minh cho người dùng.

CHƯƠNG 2. BUSINESS UNDERSTANDING

Giai đoạn hiểu nghiệp vụ này đóng vai trò nền tảng trong việc định hình phạm vi nghiên cứu và xác định các tiêu chí thành công. Giai đoạn này thực hiện hai nhiệm vụ cốt lõi: xác định các mục tiêu chiến lược của doanh nghiệp và chuyển đổi chúng thành các yêu cầu kỹ thuật cụ thể cho bài toán khai phá dữ liệu.

2.1. Vấn đề và mục tiêu nghiệp vụ:

Vấn đề cốt lõi của thị trường bất động sản hiện tại là sự phụ thuộc vào các phương pháp định giá truyền thống mang tính định tính. Quy trình này thường dựa trên kinh nghiệm cá nhân của chuyên gia môi giới, dẫn đến sự chủ quan, sai lệch lớn và thiếu tính nhất quán do tác động của hàng loạt yếu tố biến thiên phức tạp. Từ thực trạng trên, mục tiêu nghiệp vụ được xác lập nhằm giải quyết ba vấn đề chính:

- Giảm thiểu tính chủ quan thông qua việc thay thế sự phán đoán cảm tính bằng một phương pháp định lượng dựa trên dữ liệu lịch sử.
- Minh bạch hóa thị trường bằng cách cung cấp cơ sở tham chiếu khách quan, giúp thu hẹp khoảng cách thông tin giữa người mua và người bán.
- Tối ưu hóa quyết định giao dịch, hỗ trợ người bán thiết lập mức giá cạnh tranh và bảo vệ người mua khỏi rủi ro định giá cao hơn giá trị thực (overpricing).

2.2. Yêu cầu khai thác dữ liệu

Để hiện thực hóa các mục tiêu nghiệp vụ nêu trên, nhóm đã chuyển đổi bài toán kinh doanh thành bài toán kỹ thuật cụ thể trong lĩnh vực Học máy (Machine Learning).

2.2.1. Định nghĩa bài toán:

Đây là bài toán hồi quy (Regression) thuộc nhóm học có giám sát (Supervised Learning). Hệ thống cần xây dựng một hàm mục tiêu f(x) nhận đầu vào là các vector đặc trưng của căn nhà (diện tích, vị trí, số phòng) và trả về biến mục tiêu là giá nhà – là một biến định lượng liên tục.

2.2.2. Thiết lập tiêu chí đánh giá kỹ thuật

Các chỉ số đo lường hiệu quả (KPIs) được thiết lập nhằm đảm bảo mô hình không chỉ hoạt động được mà còn phải đạt độ chính xác có ý nghĩa thực tiễn:

- Độ phù hợp của mô hình: Sử dụng chỉ số R-squared (R^2) với yêu cầu ngưỡng lớn hơn 0.8. Yêu cầu này đặt ra tiêu chuẩn rằng mô hình phải giải thích được tối thiểu 80% sự biến thiên của dữ liệu giá nhà, đảm bảo khả năng tổng quát hóa các quy luật thị trường.
- Tối thiểu hóa sai số dự báo: Sử dụng chỉ số RMSE (Root Mean Squared Error) làm hàm mất mát (loss function) cần tối ưu. Việc giảm thiểu RMSE đồng nghĩa với việc giảm biên độ sai lệch giữa giá dự đoán và giá thực tế, trực tiếp phục vụ mục tiêu nghiệp vụ là giúp người dùng định giá "sát" nhất với thị trường.

CHƯƠNG 3. DATA UNDERSTANDING

Giai đoạn hiểu dữ liệu đóng vai trò cầu nối thiết yếu giữa các mục tiêu nghiệp vụ và quá trình tiền xử lý kỹ thuật. Nhiệm vụ trọng tâm của giai đoạn này là thu thập dữ liệu, đánh giá chất lượng ban đầu và thực hiện phân tích khám phá (Exploratory Data Analysis - EDA) nhằm nhận diện các đặc tính thống kê, các vấn đề tiềm ẩn và các giả định cho mô hình hóa.

3.1. Thu thập và mô tả dữ liệu ban đầu

Sử dụng bộ dữ liệu thứ cấp "Bengaluru House Price Data" được thu thập từ Kaggle. Tập dữ liệu bao gồm 13,320 quan sát (observations) đại diện cho các giao dịch bất động sản, với không gian đặc trưng gồm 9 biến số (variables).

Cấu trúc dữ liệu bao gồm biến mục tiêu (Y) là price (đơn vị Lakh Rupee) và các biến dự báo (X) bao gồm cả biến định lượng (numerical) như total_sqft, bath, balcony và biến định danh (categorical) như area_type, location, availability, society, size.

3.2. Đánh giá chất lượng dữ liệu

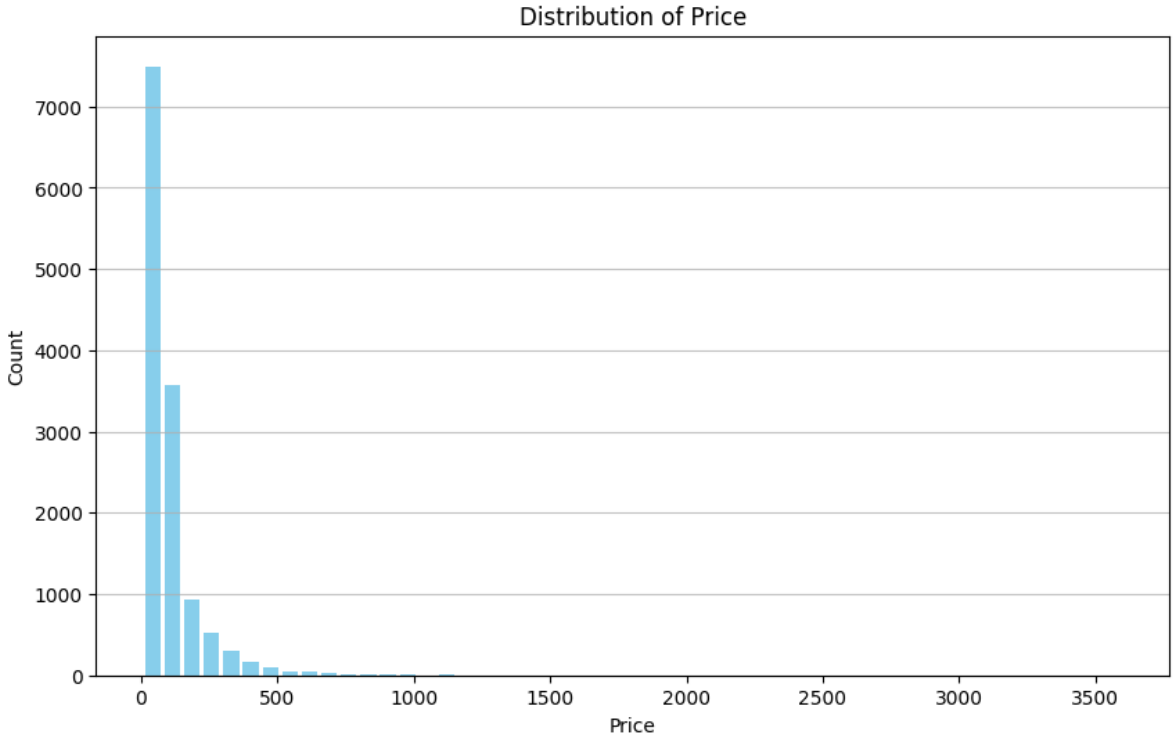
Thông qua thống kê mô tả sơ bộ, báo cáo đã nhận diện các vấn đề nghiêm trọng liên quan đến tính toàn vẹn và nhất quán của dữ liệu:

- Vấn đề dữ liệu khuyết thiếu (Missing Values): Biến society ghi nhận tỷ lệ thiếu hụt lên tới ~41% (5,502 mẫu), đặt ra yêu cầu loại bỏ để tránh nhiễu. Các biến balcony và bath cũng tồn tại giá trị khuyết nhưng ở mức độ thấp hơn.
- Vấn đề định dạng (Format Inconsistency): Dữ liệu chưa ở dạng chuẩn hóa để tính toán. Cụ thể, biến total_sqft chứa các giá trị khoảng (range) thay vì số thực duy nhất; biến size chứa dữ liệu văn bản không đồng nhất ("BHK", "Bedroom").

3.3. Kết quả Phân tích Khám phá Dữ liệu (EDA Report)

Quá trình EDA đã cung cấp những hiểu biết sâu sắc (insights) về cấu trúc nội tại của dữ liệu, định hướng trực tiếp cho chiến lược tiền xử lý ở giai đoạn sau.

3.3.1. Phân phối của biến mục tiêu (Target Distribution)



Hình 3.3.1. Biểu đồ phân phối của biến mục tiêu

Biểu đồ phân phối của biến price thể hiện tính chất lệch phải (Right-skewed) rõ rệt. Đa số các điểm dữ liệu tập trung ở phân khúc giá thấp và trung bình, trong khi phần đuôi bên phải kéo dài với sự xuất hiện của các giá trị ngoại lai (outliers) đại diện cho bất động sản cao cấp.

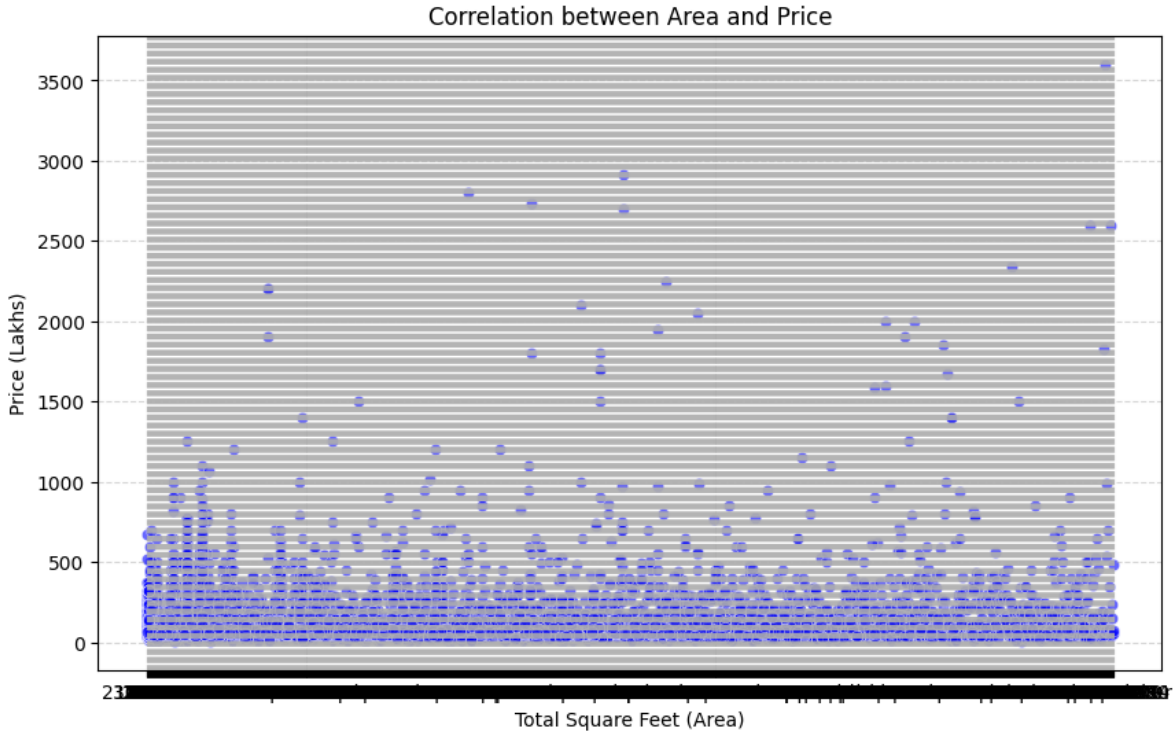
Việc sử dụng mô hình hồi quy tuyến tính trên phân phối này mà không xử lý ngoại lai sẽ dẫn đến sai số MSE lớn do mô hình bị chi phối bởi các giá trị cực đoan.

3.3.2. Đặc điểm không gian đặc trưng

Vấn đề đa dạng cao (High Cardinality) tại biến location: Thuộc tính location chứa hơn 1,300 giá trị duy nhất, trong đó đa phần xuất hiện với tần suất rất thấp (ít hơn 10 lần).

Việc áp dụng kỹ thuật One – Hot Encoding trực tiếp sẽ dẫn đến hiện tượng "lời nguyền số chiều" (Curse of Dimensionality), làm thưa hóa không gian dữ liệu (sparsity) và gia tăng chi phí tính toán cũng như nguy cơ quá khớp (overfitting).

3.3.3. Phân tích tương quan



Hình 3.3.2. Biểu đồ tán xạ thể hiện phân bố điểm dữ liệu giữa biến total_sqft *và biến mục tiêu* price

Biến total_sqft (diện tích) thể hiện mối tương quan tuyến tính thuận chiều mạnh mẽ với price. Đây được xác định là biến dự báo quan trọng nhất (predictor).

Hai biến độc lập size (số phòng ngủ) và bath (số phòng tắm) có sự tương quan mạnh với nhau. Điều này cảnh báo về hiện tượng đa cộng tuyến (Multicollinearity) – việc dư thừa thông tin đầu vào, có thể làm giảm độ ổn định của các hệ số hồi quy trong mô hình tuyến tính.

CHƯƠNG 4. DATA PREPARATION

Giai đoạn chuẩn bị dữ liệu đóng vai trò quyết định đến hiệu năng của mô hình hồi quy. Từ những hiểu biết trong giai đoạn phân tích khám phá (EDA) tiếp tục thiết lập một quy trình tiền xử lý (Preprocessing Pipeline) gồm 5 bước chính nhằm chuyển đổi dữ liệu thô thành dạng vector số học chuẩn hóa, sạch nhiễu và giàu thông tin.

4.1. Làm sạch và chuẩn hóa dữ liệu (Data Cleaning and Standardization)

Quy trình bắt đầu bằng việc xử lý các khiếm khuyết trong cấu trúc dữ liệu thô:

- Xử lý dữ liệu khuyết thiếu (Missing Values): Áp dụng chiến lược loại bỏ (dropping) dựa trên ngưỡng tỷ lệ thiếu hụt và mức độ quan trọng của biến.
    - Các biến có tỷ lệ khuyết thiếu cao (society thiếu khoảng 41%) hoặc ít đóng góp vào khả năng dự báo (availability, area_type, balcony) được loại bỏ hoàn toàn khỏi không gian đặc trưng.
    - Đối với các biến cốt lõi (location, size), phương pháp loại bỏ dòng (row-wise deletion) được áp dụng do tỷ lệ thiếu hụt không đáng kể (thấp hơn 0.2%), đảm bảo tính toàn vẹn của tập mẫu.
- Chuẩn hóa định dạng (Data Parsing): Thực hiện trích xuất thông tin số học từ dữ liệu dạng văn bản hỗn hợp.
    - Biến size: Sử dụng hàm tách chuỗi để đồng nhất các định dạng "BHK" và "Bedroom" về một biến số nguyên (bhk). Giải pháp và sử dụng hàm lambda tách chuỗi ký tự, chỉ giữ lại phần số nguyên đầu tiên để tạo ra cột mới bhk (kiểu int). Ví dụ: "2 BHK" => 2.
    - Biến total_sqft: Xử lý tính không nhất quán của dữ liệu diện tích (dạng khoảng giá trị). Các trường hợp này được chuẩn hóa bằng phương pháp tính trung bình cộng (mean imputation) của cận trên và cận dưới.

4.2. Kỹ thuật đặc trưng (Feature Engineering) và giảm chiều dữ liệu (Dimensionality Reduction)

Để giải quyết vấn đề "Bùng nổ số chiều" đã được nhận diện trong giai đoạn EDA:

- Tạo biến phái sinh**:** Thiết lập biến mới price_per_sqft (giá trên mỗi feet vuông). Đây là biến trung gian quan trọng phục vụ cho việc phát hiện các điểm dị biệt trong bước tiếp theo.

$$Price\_per\_sqlt = \frac{Price \times 100.000}{Total\_sqft}$$

- Giảm chiều dữ liệu cho biến hạng mục (Cardinality Reduction): Đối với biến location (hơn 1,300 giá trị), nhóm áp dụng kỹ thuật gộp nhóm dựa trên tần suất (frequency-based grouping). Các địa điểm xuất hiện dưới 10 lần được gán nhãn chung là "other". Kỹ thuật này giúp giảm đáng kể số lượng biến giả (dummy variables) khi mã hóa, ngăn chặn hiện tượng quá khớp (overfitting).

4.3. Xử lý giá trị ngoại lai (Outlier Removal)

Đây là bước quan trọng nhất để đảm bảo tính ổn định của mô hình Linear Regression. Áp dụng phương pháp lọc nhiễu đa tầng kết hợp giữa kiến thức nghiệp vụ và thống kê:

- Tầng 1 – logic nghiệp vụ (Business Logic): Loại bỏ các quan sát phi thực tế về mặt kiến trúc, cụ thể là các căn nhà có diện tích trung bình mỗi phòng ngủ nhỏ hơn 300 sqft.
- Tầng 2 – thống kê (Statistical Method): Tại mỗi địa điểm (location), giả định giá đất tuân theo phân phối chuẩn. Các điểm dữ liệu nằm ngoài khoảng ($\mu - \sigma$, $\mu + \sigma$) của price_per_sqft bị coi là ngoại lai và bị loại bỏ.
- Tầng 3 – tương quan nội tại: Loại bỏ các trường hợp bất thường về cấu trúc tiện ích, nơi số lượng phòng tắm lớn hơn số phòng ngủ cộng thêm 2 (bath > bhk + 2).

4.4. Mã hóa dữ liệu (Data Encoding)

Để chuyển đổi dữ liệu hạng mục sang dạng số học mà máy tính có thể xử lý, kỹ thuật One-Hot Encoding được áp dụng cho biến location.

- Mỗi địa điểm (sau khi đã giảm chiều) được chuyển thành một vector nhị phân riêng biệt.
- Để tránh hiện tượng "bẫy biến giả" (Dummy Variable Trap) gây ra đa cộng tuyến hoàn hảo, một cột hạng mục được loại bỏ (drop_first=True hoặc tương đương trong cấu hình).

4.5. Kết quả: Tập dữ liệu huấn luyện (Final Training Dataset)

Sau quá trình xử lý nghiêm ngặt, kích thước tập dữ liệu giảm từ 13,320 xuống còn khoảng 7,251 quan sát. Mặc dù số lượng mẫu giảm đi, chất lượng dữ liệu được nâng cao đáng kể:

- Loại bỏ hoàn toàn các giá trị rỗng và định dạng sai.
- Triệt tiêu các điểm dữ liệu nhiễu (outliers) có thể gây sai lệch đường hồi quy.
- Không gian đặc trưng được tối ưu hóa, sẵn sàng cho việc đưa vào huấn luyện mô hình học máy.

CHƯƠNG 5. MODELING

Sau quá trình tiền xử lý dữ liệu là giai đoạn trọng tâm: xây dựng và huấn luyện mô hình. Mục tiêu cốt lõi là xác định hàm mục tiêu f(x) có khả năng xấp xỉ tốt nhất mối quan hệ giữa các đặc trưng đầu vào và giá nhà, đồng thời tối thiểu hóa sai số dự báo trên tập dữ liệu chưa từng thấy (unseen data).

5.1. Thiết lập thực nghiệm (Experimental Setup)

Để đảm bảo tính khách quan, nhất quán và khả năng tái lập (reproducibility) của kết quả nghiên cứu, cấu hình thực nghiệm được thiết lập chặt chẽ như sau:

- Phân hoạch dữ liệu (Data Partitioning): Tập dữ liệu được phân chia theo phương pháp Hold-out với tỷ lệ 80/20.
    - Tập huấn luyện (Training Set – 80%): Được sử dụng để mô hình học các trọng số (weights) và tham số nội tại.
    - Tập kiểm thử (Testing Set – 20%): Đóng vai trò là tập dữ liệu độc lập, dùng để đánh giá năng lực dự báo thực tế của mô hình sau huấn luyện.
    - Tham số random_state = 10 được cố định nhằm đảm bảo sự đồng nhất trong việc xáo trộn và chia dữ liệu giữa các lần thực nghiệm.
- Pipeline xử lý (Processing Pipeline): triển khai cấu trúc pipeline của Scikit-Learn để đóng gói quy trình xử lý. Cơ chế này đảm bảo tính tuần tự: dữ liệu đi qua bộ chuyển đổi (Transformer) trước khi đến bộ ước lượng (Estimator). Điều này có ý nghĩa quan trọng trong việc ngăn chặn rò rỉ dữ liệu (Data Leakage) – hiện tượng thông tin từ tập kiểm thử rò rỉ vào quá trình huấn luyện, gây ra ảo tưởng về độ chính xác.
    - Bước 1: ColumnTransformer áp dụng OneHotEncoder cho các cột vị trí (location) và giữ nguyên các cột số khác.
    - Bước 2: Tích hợp thuật toán hồi quy (Linear Regression hoặc Random Forest) để dự đoán.

5.2. Các mô hình đề xuất

5.2.1. Mô hình cơ sở hồi quy tuyến tính (Linear Regression)

Được chọn làm mô hình cơ sở (Baseline Model) nhờ ưu điểm về tính đơn giản, chi phí tính toán thấp và khả năng giải thích (interpretability) cao. Thuật

toán giả định mối quan hệ tuyến tính giữa các biến độc lập và giá nhà ($Y = \beta X + \varepsilon$). Sử dụng thiết lập mặc định của thư viện Scikit-Learn (với normalize = False).

5.2.2. Mô hình phi tuyến tính (Random Forest Regressor)

Là đại diện của phương pháp Học kết hợp (Ensemble Learning), cụ thể là kỹ thuật Bagging. Random Forest xây dựng nhiều cây quyết định (Decision Trees) và tổng hợp kết quả để đưa ra dự đoán cuối cùng. Thuật toán này có khả năng nắm bắt các mối quan hệ phi tuyến phức tạp và giảm thiểu phương sai (variance), từ đó hạn chế hiện tượng quá khớp (overfitting) thường gặp ở cây quyết định đơn lẻ. Thiết lập số lượng cây quyết định là 100 (n_estimators = 100).

5.3. Chiến lược đánh giá chéo (Cross-Validation Strategy)

Để đánh giá độ ổn định (Stability) của mô hình và khắc phục hạn chế của việc chia dữ liệu ngẫu nhiên một lần (vốn có thể dẫn đến kết quả thiên lệch), nhóm áp dụng kỹ thuật K-Fold Cross Validation.

- Phương pháp: Shuffle Split Cross Validation.
- Cấu hình: Số lượng nếp gấp K = 5 (5 – Fold).
- Quy trình: Tập dữ liệu được chia ngẫu nhiên thành 5 phần bằng nhau. Quá trình huấn luyện và kiểm thử được lặp lại 5 lần, mỗi lần sử dụng một phần khác nhau làm tập kiểm định (validation set).
- Kết quả đầu ra: Giá trị trung bình của chỉ số R^2 qua 5 lần chạy. Chỉ số này phản ánh mức độ tin cậy và ổn định của mô hình khi tiếp xúc với các tập con dữ liệu khác nhau.

CHƯƠNG 6. EVALUATION

Giai đoạn đánh giá được thực hiện nhằm kiểm định khả năng tổng quát hóa (generalization capability) của các mô hình đã huấn luyện trên tập dữ liệu kiểm thử độc lập. Mục tiêu không chỉ dừng lại ở việc so sánh các chỉ số thống kê, mà còn đi sâu vào phân tích đặc điểm sai số để lựa chọn giải pháp tối ưu nhất cho bài toán định giá thực tế.

6.1. Khung tiêu chí đánh giá (Evaluation Metrics Framework)

Để đảm bảo tính toàn diện trong việc đo lường hiệu quả mô hình, nghiên cứu sử dụng bộ 4 chỉ số định lượng bao quát các khía cạnh khác nhau của sai số:

- Độ phù hợp (Goodness of Fit): Sử dụng R^2 Score. Đây là chỉ số tiên quyết, phản ánh tỷ lệ biến thiên của giá nhà được giải thích bởi mô hình. Giá trị càng gần 1 càng tốt.
- Biên độ sai số (Error Magnitude):
    - Sử dụng RMSE (Root Mean Squared Error) – căn bậc hai của sai số bình phương trung bình, chỉ số này phản ánh mức độ sai lệch trung bình giữa giá dự đoán và giá thực tế (cùng đơn vị với giá nhà là Lakhs), giá trị càng nhỏ càng tốt.
    - Đồng thời cũng sử dụng MAE (Mean Absolute Error) – sai số tuyệt đối trung bình, giúp đánh giá độ lớn sai số mà không quan tâm chiều hướng (âm/dương).

  Trong đó, RMSE được ưu tiên xem xét hơn vì đặc tính phạt nặng các sai số lớn (large errors), giúp phát hiện các dự đoán lệch chuẩn nghiêm trọng.
- Sai số tương đối (Relative Error): Sử dụng MAPE (Mean Absolute Percentage Error) – sai số phần trăm trung bình, cho biết dự đoán sai lệch bao nhiêu % so với thực tế. Cung cấp cái nhìn trực quan về độ lệch theo phần trăm, giúp các bên liên quan dễ dàng hình dung mức độ tin cậy trong bối cảnh thương mại.

6.2. Kết quả thực nghiệm và so sánh

Dưới đây là bảng tổng hợp kết quả so sánh giữa hai mô hình trên tập kiểm thử (Test Set) và kiểm chứng chéo (Cross-Validation):

Tiêu chí (Metric)	Linear Regression	Random Forest Regressor
MAE	17.99	18.89
RMSE	34.93	41.88
R^2 Score	0.875	0.820
MAPE (%)	20.50%	20.17%
CV R^2 (k=5)	0.821	0.758

Bảng 6.1. Bảng tổng hợp kết quả so sánh

Kết quả thực nghiệm trên tập kiểm thử (Test Set) và kiểm chứng chéo (Cross-Validation) cho thấy sự vượt trội đáng ngạc nhiên của mô hình tuyến tính đơn giản so với mô hình phi tuyến phức tạp:

- Về độ chính xác (R^2 và RMSE): Linear Regression đạt $R^2 = 0.875$, vượt trội hơn so với Random Forest ($R^2 = 0.820$). Tương tự, chỉ số RMSE của Linear Regression (34.93) thấp hơn đáng kể so với Random Forest (41.88), chứng tỏ mô hình tuyến tính có khả năng bám sát dữ liệu thực tế tốt hơn.
- Về độ ổn định (Stability): Kết quả kiểm chứng chéo 5 lần (5-Fold CV) khẳng định tính bền vững của Linear Regression với R^2 trung bình đạt 0.821, trong khi Random Forest chỉ đạt 0.758 và có dấu hiệu biến động mạnh hơn giữa các lần chạy.

6.3. Biện luận kết quả và phân tích lỗi

Phân tích nguyên nhân dẫn đến hiệu năng vượt trội của Linear Regression so với Random Forest tập trung vào đặc thù của dữ liệu sau xử lý:

Tác động của dữ liệu thưa (Sparsity Impact): Việc áp dụng kỹ thuật One-Hot Encoding cho biến location đã tạo ra một không gian đặc trưng nhiều chiều và thưa (high-dimensional sparse matrix). Theo lý thuyết học máy, các mô hình tuyến tính (Linear Models) thường hoạt động rất hiệu quả và hội tụ tốt trên dạng dữ liệu này. Ngược lại, các mô hình dựa trên cây (Tree-based models) như Random Forest thường gặp khó khăn trong việc phân chia không gian dữ liệu thưa nếu không được tinh chỉnh tham số (Hyperparameter Tuning) cực kỳ sâu.

Ngưỡng sai số chấp nhận được: Cả hai mô hình đều ghi nhận mức sai số MAPE xoay quanh ngưỡng 20%. Trong lĩnh vực định giá bất động sản, đây là mức sai số chấp nhận được (acceptable margin), do giá trị căn nhà còn chịu chi phối bởi các biến số

định tính không thể đo lường (intangibles) như phong thủy, nội thất hay tâm lý người bán, những yếu tố mà dữ liệu hiện tại chưa bao quát hết.

6.4. Kết luận: Lựa chọn mô hình tối ưu (Final Model Selection)

Dựa trên nguyên tắc "Occam's Razor" (Ưu tiên giải pháp đơn giản nhất khi hiệu quả tương đương hoặc tốt hơn), nhóm nghiên cứu quyết định lựa chọn Linear Regression làm mô hình cuối cùng để triển khai (Deployment). Quyết định này dựa trên 4 trụ cột chính:

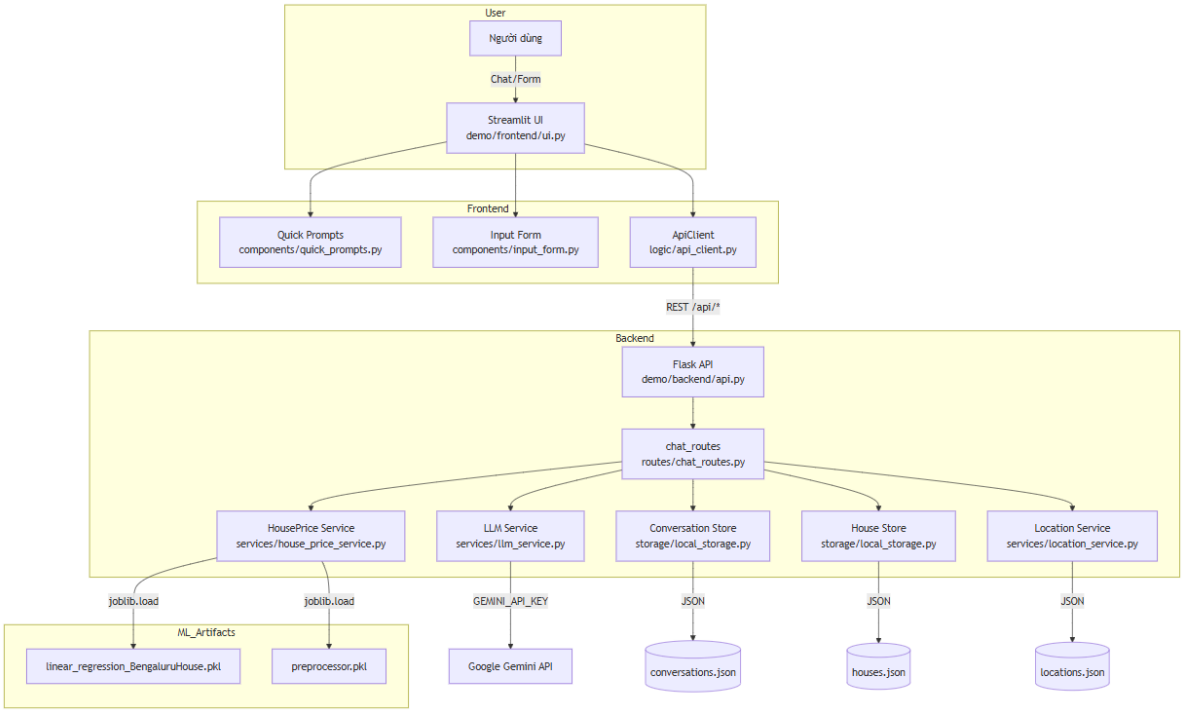
- Hiệu năng vượt trội: Đạt độ chính xác (R^2) cao nhất và sai số (RMSE) thấp nhất.
- Tính ổn định: Duy trì kết quả nhất quán qua kiểm định chéo (Cross-Validation).
- Chi phí tính toán: Thời gian huấn luyện và suy diễn (inference) cực nhanh, phù hợp cho triển khai ứng dụng thời gian thực.
- Khả năng giải thích (Explainability): Các hệ số hồi quy minh bạch giúp dễ dàng lượng hóa tác động của từng đặc trưng (ví dụ: diện tích, vị trí) lên giá nhà, điều mà các mô hình "hộp đen" phức tạp khó đạt được.

CHƯƠNG 7. DEPLOYMENT

Giai đoạn triển khai nhằm mục đích hiện thực hóa mô hình dự báo giá nhà thành một sản phẩm công nghệ hoàn chỉnh, cho phép người dùng cuối tương tác trực quan bằng cách xây dựng một hệ thống ứng dụng Full-stack tích hợp trí tuệ nhân tạo tạo sinh (Generative AI).

7.1. Kiến trúc hệ thống

Hệ thống được thiết kế theo mô hình Client-Server kết hợp với Microservices, tách biệt hoàn toàn giữa giao diện người dùng và logic xử lý nghiệp vụ để đảm bảo khả năng mở rộng.



Hình 7.1.1. Sơ đồ kiến trúc hệ thống

Các thành phần chính:

- Frontend (Giao diện): Sử dụng Streamlit, chạy trên cổng 10001. Đây là nơi người dùng tương tác, nhập liệu và trò chuyện với Chatbot.
- Backend (API Server): Sử dụng Flask, chạy trên cổng 10000. Đây là bộ não trung tâm xử lý các yêu cầu, điều phối dữ liệu và gọi các mô hình AI.
- AI Models Layer:
    - Mô hình dự đoán (Predictive Model): Sử dụng mô hình Linear Regression đã huấn luyện (.pkl) để tính toán giá nhà.

    - Mô hình ngôn ngữ lớn (LLM): Tích hợp Google Gemini Pro để tạo ra trải nghiệm Chatbot thông minh, hỗ trợ tư vấn và giải đáp thắc mắc cho người dùng.
- Storage (Lưu trữ): Sử dụng hệ thống file JSON cục bộ để lưu trữ lịch sử đoạn chat và các bản ghi dự đoán.

7.2. Công nghệ sử dụng

Hệ thống được phát triển hoàn toàn trên hệ sinh thái Python 3.8+, tận dụng sức mạnh của các thư viện mã nguồn mở hàng đầu:

- Web Frameworks: Sự kết hợp giữa Flask (cho Backend hiệu năng cao, nhẹ) và Streamlit (cho Frontend phát triển nhanh, chuyên dụng cho Data App).
- Core AI Libraries**:**
    - Scikit-learn: Thư viện lõi vận hành mô hình hồi quy tuyến tính.
    - Google Generative AI SDK: Cầu nối giao tiếp với Gemini API.
- Environment Management**:** Sử dụng python-dotenv để quản lý biến môi trường, đảm bảo nguyên tắc bảo mật (Security Best Practices) đối với các khóa API nhạy cảm.

7.3. Cơ chế vận hành và luồng dữ liệu

Quy trình hoạt động của hệ thống thực tế được chuẩn hóa qua 4 bước khép kín:

- Bước 1: Khởi tạo (Initialization): script điều phối app.py kích hoạt song song hai tiến trình server: Flask (Backend) và Streamlit (Frontend), thiết lập kênh giao tiếp nội bộ.
- Bước 2: Thu thập và truyền tải (Data Acquisition): người dùng nhập thông số bất động sản (Khu vực, diện tích, số phòng) trên giao diện Web. Frontend đóng gói dữ liệu và gửi yêu cầu POST đến endpoint /api/house/predict.
- Bước 3: Suy diễn (Inference and Processing):
    - Backend nhận yêu cầu, thực hiện tiền xử lý chuẩn hóa dữ liệu đầu vào.
    - Dữ liệu sạch được đưa vào mô hình Linear Regression để tính toán ra con số giá trị thực (ví dụ: 85.5 Lakhs).
- Bước 4: Tổng hợp và phản hồi (Response Synthesis):
    - Kết quả định lượng được chuyển tiếp sang module GenAI.

- Chatbot (Gemini) nhận kết quả giá, kết hợp với ngữ cảnh đầu vào để sinh ra lời tư vấn tự nhiên.
- Hệ thống hiển thị đồng thời mức giá dự đoán và lời khuyên chi tiết cho người dùng trên giao diện.

CHƯƠNG 8. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

8.1. Thành quả đạt được

- Về dữ liệu: Đã xây dựng quy trình chuẩn hóa dữ liệu khép kín, bao gồm làm sạch, giảm chiều dữ liệu (gom nhóm địa điểm) và loại bỏ ngoại lai dựa trên logic nghiệp vụ (diện tích/phòng, giá trần/sàn).
- Về mô hình: Kết quả thực nghiệm chứng minh Linear Regression là mô hình tối ưu nhất cho tập dữ liệu này với R^2 ~87.5% và sai số thấp nhất, vượt trội hơn Random Forest về độ ổn định.
- Về sản phẩm: Đã phát triển thành công hệ thống Full-stack AI tích hợp Chatbot Gemini, chuyển đổi mô hình thuật toán khô khan thành ứng dụng tư vấn bất động sản thông minh, thân thiện với người dùng.

8.2. Hạn chế

- Phạm vi: Dữ liệu chỉ giới hạn tại Bengaluru và chưa cập nhật theo thời gian thực (Real-time), chưa phản ánh được biến động thị trường mới nhất.
- Đặc trưng: Thiếu các yếu tố định tính quan trọng ảnh hưởng đến giá như: hướng nhà, nội thất, tiện ích ngoại khu (trường học, bệnh viện).
- Hạ tầng: Hệ thống lưu trữ lịch sử chat còn dùng file cục bộ (Local JSON), chưa tối ưu cho lượng người dùng lớn.

8.3. Hướng phát triển

- Cải thiện dữ liệu và thuật toán: Xây dựng hệ thống tự động thu thập giá nhà (Crawling) để cập nhật mô hình định kỳ. Thử nghiệm thêm các thuật toán Boosting (XGBoost, CatBoost) để xử lý các ca khó.
- Mở rộng tính năng: Tích hợp bản đồ số (Google Maps), tính năng so sánh giá giữa các khu vực và gợi ý đầu tư.
- Triển khai Cloud: Đưa ứng dụng lên nền tảng đám mây (AWS/GCP), sử dụng Database chuyên dụng và Docker để đảm bảo khả năng vận hành ổn định trên quy mô lớn.

PHỤ LỤC

Practice 1 – EDA trên Titanic Và Iris

Khám phá dữ liệu Titanic

Chuẩn bị dữ liệu

Giới thiệu đề bài

Sử dụng dữ liệu Titanic nhằm giúp sinh viên thực hành đầy đủ quy trình phân tích và xử lý dữ liệu thực tế, bao gồm: khám phá dữ liệu (EDA), trực quan hóa, tiền xử lý dữ liệu, chuẩn hóa thuộc tính, mã hóa biến phân loại và xây dựng các mô hình phân loại cơ bản. Mục tiêu cuối cùng của bài là dự đoán khả năng sống sót của hành khách (biến Survived), đồng thời đánh giá hiệu suất mô hình bằng các chỉ số Accuracy, Precision, Recall, F1 và Cross-Validation.

Công cụ và thư viện sử dụng

Sử dụng các thư viện phổ biến:

- Pandas: đọc, làm sạch, xử lý dữ liệu, tạo DataFrame sau chuẩn hóa/mã hóa.
- NumPy: thao tác mảng số học.
- Matplotlib, Seaborn: trực quan hóa.
- Scikit-learn: tiền xử lý và xây dựng mô hình học máy.

Mô tả các thuộc tính trong bộ dữ liệu

Bộ dữ liệu Titanic gồm 12 thuộc tính mô tả thông tin nhân khẩu học, vé tàu, vị trí và trạng thái sống sót của hành khách. Mỗi thuộc tính được giải thích như sau:

Tên thuộc tính (Variable)	Định nghĩa (Definition)	Giải thích chi tiết
PassengerId	ID hành khách	Số thứ tự định danh duy nhất cho mỗi hành khách
Survived	Sống sót	Biến mục tiêu (Target Variable). 0 = Không qua khỏi (No) 1 = Sống sót (Yes)

Pclass	Hạng vé	Hạng vé của hành khách, đại diện cho địa vị kinh tế xã hội (SES). 1 = Hạng nhất (Upper) 2 = Hạng hai (Middle) 3 = Hạng ba (Lower)
Name	Tên	Tên đầy đủ của hành khách.
Sex	Giới tính	male (Nam) hoặc female (Nữ).
Age	Tuổi	Tuổi của hành khách
SibSp	Số lượng anh chị em/vợ chồng	Số lượng anh chị em ruột (Siblings) hoặc vợ/chồng (Spouses) đi cùng trên tàu.
Parch	Số lượng bố mẹ/con cái	Số lượng bố mẹ (Parents) hoặc con cái (Children) đi cùng trên tàu.
Ticket	Số vé	Mã số vé .
Fare	Giá vé	Số tiền hành khách đã trả cho vé
Cabin	Số hiệu buồng	Mã số phòng ngủ của hành khách
Embarked	Cảng xuất phát	Cảng mà hành khách lên tàu. C = Cherbourg (Pháp) Q = Queenstown (Ireland) S = Southampton (Anh)

Bảng 1. Thông tin bộ dữ liệu Titanic

Khám phá dữ liệu

Nạp dữ liệu và hiển thị 5 dòng đầu tiên

Nạp và chuyển dữ liệu sang dạng DataFrame. Đây là bước kiểm tra ban đầu (data sanity check). Giúp hiểu nhanh cấu trúc dữ liệu: có bao nhiêu cột, tên cột là gì, kiểu dữ liệu sơ bộ, ví dụ vài giá trị mẫu => nắm được nội dung dataset trước khi làm các bước xử lý, thống kê, vẽ biểu đồ,...

Kiểm tra kích thước và thông tin dataset

- Có 891 bản ghi tương ứng với 891 hành khách và 12 thuộc tính.
- Ở cột Age và Cabin có dữ liệu trống.

Thống kê và kiểm tra giá trị thiếu

Thực hiện thống kê mô tả cho các cột dạng số với các thông số cụ thể như sau:

Thông số	**Ý nghĩa**
count	Số lượng giá trị không bị thiếu (non-null) trong mỗi cột
mean	Giá trị trung bình
std	Độ lệch chuẩn (mức độ phân tán dữ liệu)
min, max	Giá trị nhỏ nhất và lớn nhất
25%, 50%, 75%	Các phân vị (quartiles) – chia dữ liệu thành 4 phần bằng nhau (median là 50%)

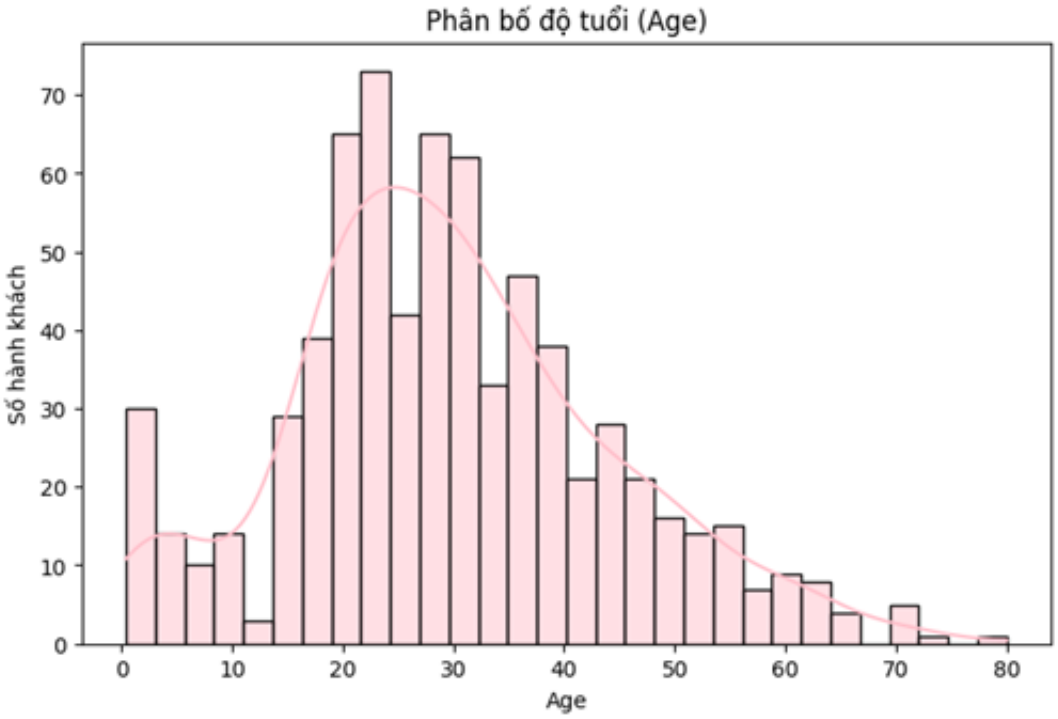
Bảng 1.1.2. Thống kê các cột dạng số của bộ dữ liệu Titanic

- Kết quả cho thấy:
  - Age: Có 714 giá trị. Tuổi trung bình khoảng 29, nhưng min 0.42 và max là 80 -> có giá trị ngoại lai.
  - Fare: Trung bình 32.2 bảng Anh, nhưng có giá trị tối đa tới 512, cho thấy phân bố lệch phải (một số vé rất đắt).
  - SibSp và Parch: Phần lớn hành khách đi một mình (đa số bằng 0).
  - Pclass: Có 3 hạng vé (1, 2, 3) → giá trị này mang tính phân loại (categorical) nhưng đang được lưu dạng số.
  - Survived: Trung bình 0.38 → nghĩa là khoảng 38% hành khách sống sót.
- Kiểm tra giá trị thiếu:
  - Age thiếu 117 người (cần điền giá trị trung vị cho cột này).

- Cabin thiếu 687 (thiếu gần 80%). Do tỷ lệ thiếu quá lớn, thuộc tính này không mang nhiều ý nghĩa thống kê và sẽ được loại bỏ.
- Thuộc tính Embarked: Thiếu 2 mẫu (tỷ lệ rất nhỏ).

Trực quan hóa dữ liệu

Histogram cho thuộc tính Age

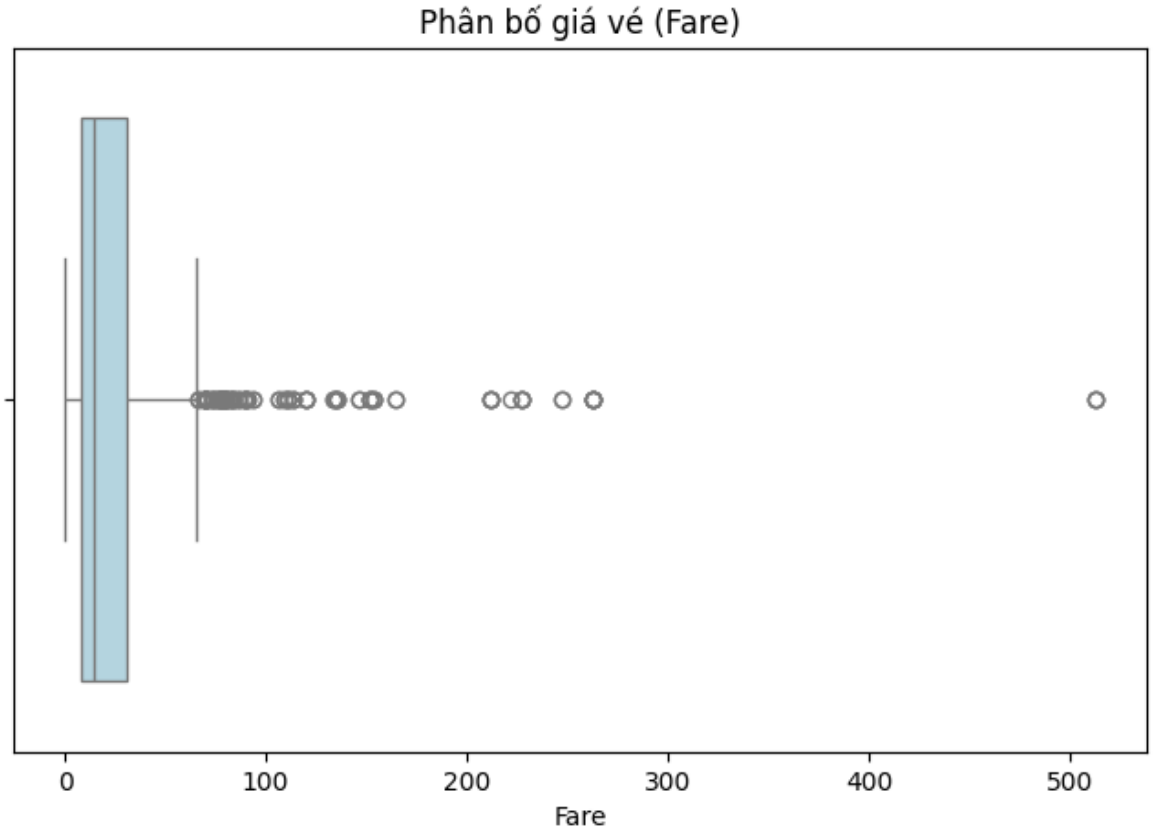


Hình 1.1.1. Biểu đồ Histogram cho thuộc tính Age

Quan sát biểu đồ trên, chúng ta rút ra các nhận xét thống kê quan trọng sau:

- Biểu đồ có dạng lệch phải (Right-skewed). Điều này có nghĩa là đa số dữ liệu tập trung ở phía bên trái (độ tuổi trẻ) và có một cái "đuôi" dài kéo về phía bên phải (người già).
- Tập trung mật độ cao nhất ở khoảng 20 đến 30 tuổi -> Đa số hành khách đi tàu là thanh niên, người trong độ tuổi lao động di cư sang Mỹ hoặc đi du lịch.
- Có một đỉnh phụ nhỏ (small peak) ở khoảng 0-5 tuổi. - > Điều này cho thấy có một lượng đáng kể trẻ sơ sinh và trẻ nhỏ trên tàu.
- Phần đuôi bên phải kéo dài đến khoảng 80 tuổi. Có rất ít hành khách trên 65 tuổi. Những điểm dữ liệu này là ngoại lai (outliers) so với đám đông, nhưng là dữ liệu thật (không phải nhiễu).

Boxplot cho thuộc tính Fare



Hình 1.1.2. Biểu đồ Boxplot cho thuộc tính Fare

Quan sát biểu đồ Boxplot, ta thấy các đặc điểm thống kê nổi bật sau:

- Dữ liệu bị lệch phải (Right-skewed) rất mạnh.
- Phần "hộp" (chứa 50% dữ liệu ở giữa - từ phân vị 25% đến 75%) bị nén chặt về phía bên trái (khoảng giá trị thấp, dưới 50). Điều này cho thấy đại đa số hành khách trên tàu mua vé giá rẻ (Hạng 3 và Hạng 2).
- Đường gạch đứng bên trong hộp (đường trung vị) nằm ở mức rất thấp (khoảng 14-30). Điều này phản ánh mức giá phổ biến mà phần lớn mọi người chi trả. - > cần chuẩn hóa hoặc biến đổi khi dùng cho mô hình.

- Râu hộp whispers là hai đoạn thẳng mảnh kéo ra 2 bên mở rộng đến gái trị nằm trong khoảng hợp lí => Râu bên phải kéo dài hơn râu bên trái → chứng tỏ có một số người mua vé cao bất thường.
- Các chấm nằm ngoài râu hộp là ngoại lai outliers và các giá trị cao bất thường so với phần lớn dữ liệu. => Nhiều chấm rời rạc bên phải (ở 100, 200, 300, 500) → các ngoại lai dương (high outliers) — chính là vé hạng 1 hoặc suite đặc biệt, Hành khách hạng 1 (giàu, có khả năng sống sót cao hơn) => Cần kiểm tra xem là hợp lý hay lỗi nhập liệu.

Xử lý giá trị thiếu

Qua quá trình kiểm tra dữ liệu, nhóm nhận thấy thuộc tính Age có 177 giá trị thiếu (chiếm 19.87%). Dựa trên kết quả phân tích biểu đồ phân phối (Histogram) ở mục 2.1.2.4, dữ liệu tuổi có đặc điểm phân phối lệch phải (Right-skewed) và tồn tại các giá trị ngoại lai ở vùng tuổi cao.

Việc sử dụng giá trị trung bình (Mean) trong trường hợp này sẽ không chính xác do bị chi phối bởi các giá trị ngoại lai. Do đó, nhóm quyết định sử dụng phương pháp Gán giá trị (Imputation) bằng Trung vị (Median).

Kết quả sau khi xử lý: Số lượng giá trị thiếu của cột Age đã về 0, đảm bảo tính toàn vẹn dữ liệu cho quá trình huấn luyện mô hình.

Tiền Xử lý dữ liệu

Chuẩn hóa Age và Fare

Áp dụng kỹ thuật Chuẩn hóa (Feature Scaling) để đưa các biến này về cùng một phạm vi phân phối, giúp mô hình hội tụ nhanh hơn và công bằng hơn giữa các biến.

Phương pháp lựa chọn là StandardScaler:

- Nguyên lý: Biến đổi dữ liệu sao cho phân phối có giá trị trung bình bằng 0 và độ lệch chuẩn bằng 1.
- Công thức:

$$X_i' = \frac{X_i - \mu}{\sigma} = \frac{X_i - X_{\text{mean}}}{X_{std}}$$

Hình 1.1.3. Công thức StandardScaler

StandardScaler ít bị ảnh hưởng bởi giá trị ngoại lai hơn so với MinMaxScaler (nơi mà một giá trị cực đại sẽ ép toàn bộ dữ liệu còn lại về một khoảng rất nhỏ). Ngoài ra, StandardScaler phù hợp với giả định của thuật toán Logistic Regression (dữ liệu tuân theo phân phối chuẩn).

Kết quả trung bình gần bằng 0 và độ lệch chuẩn gần bằng 1 -> dữ liệu đã được đưa về cùng thang đo.

Mã hóa biến phân loại

Các mô hình học máy (Machine Learning) chỉ hoạt động được trên dữ liệu dạng số. Do đó, các thuộc tính dạng chữ (string/text) như Sex (male, female) hay Embarked (S, C, Q) cần được chuyển đổi sang dạng số.

Để chuyển đổi dữ liệu dạng chuỗi sang dạng số phục vụ cho quá trình huấn luyện, nhóm thực hiện áp dụng linh hoạt hai kỹ thuật mã hóa khác nhau tùy thuộc vào bản chất của từng biến:

- Đối với Sex đây là biến nhị phân (Binary) chỉ có hai giá trị: Male và Female. Tiến hành mã hóa nhị phân (LabelEncoder).
- Kết quả: Biến đổi thành một cột duy nhất với giá trị 0 đại diện cho Male và 1 đại diện cho Female.
- Đối với Embraked đây là biến danh định (Nominal) gồm 3 giá trị: S, C, Q. Các giá trị này không có thứ tự hơn kém (Cảng S không "lớn hơn" Cảng C). Việc sử dụng Label Encoding (0, 1, 2) có thể khiến các mô hình tuyến tính (như Logistic Regression) hiểu sai rằng có mối quan hệ thứ tự giữa các cảng. -> sử dụng mã hóa one-hot Encoding với kỹ thuật drop_first=True.
- Kết quả: Embarked_Q, Embarked_S: 1 nếu hành khách lên ở cảng đó, ngược lại 0 (Cảng còn lại — C — bị bỏ làm “chuẩn” để tránh trùng lặp thông tin).

Tách train/test

Trước khi huấn luyện model thì cần xác định rõ biến mục tiêu (y) và các biến đặc trưng (X). Trong đó PassengerId, Name, Ticket, và Cabin không hữu ích cho việc dự đoán nên loại bỏ khỏi X.

- **Survived là biến mục tiêu y.**
- **Các** đặc trưng (X): Pclass, Sex, Age, SibSp, Parch, Fare, Embarked_Q, Embarked_S.

Để đánh giá khách quan hiệu năng của mô hình, dữ liệu cần được chia thành 2 phần độc lập:

- Tập **huấn luyện (Train Set) để "dạy" mô hình (80% dữ liệu).**
- **Tập ki**ểm tra (Test Set) để đánh giá hiệu suất của mô hình trên dữ liệu mới hoàn toàn (20% dữ liệu).

Sử dụng hàm train_test_split từ thư viện sklearn.model_selection với các thiết lập sau:

- X**, y: Đầu vào là ma trận đặc trưng (X) và vector nhãn mục tiêu (y).**
- **te**st_size=0.2:
  - Dữ liệu được chia theo tỷ lệ 80/20.
  - Đây là tỷ lệ tiêu chuẩn giúp cân bằng giữa việc có đủ dữ liệu để học và đủ dữ liệu để đánh giá.
- **random**_state=42:
  - Đây là hạt giống ngẫu nhiên (Random Seed).
  - Việc cố định số 42 (hoặc bất kỳ số nguyên nào) đảm bảo rằng mỗi lần chạy lại mã nguồn, dữ liệu sẽ được xáo trộn và chia cắt theo cùng một cách hệt như nhau.
  - Ý nghĩa: Giúp kết quả thí nghiệm có tính tái lập (reproducible), đảm bảo tính nhất quán khi so sánh giữa các mô hình khác nhau hoặc khi báo cáo kết quả.

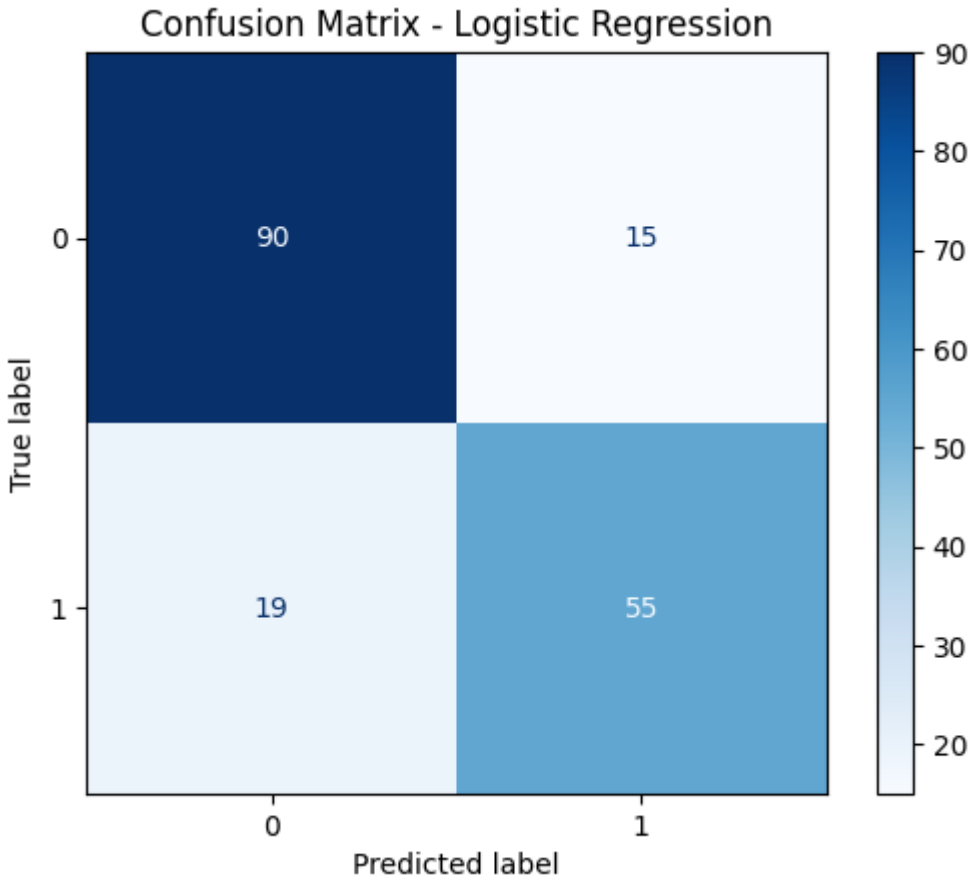
Kết quả Sau khi tách, chúng ta thu được:

- X_train: Chứa các đặc trưng dùng để học.
- y_train: Chứa đáp án đúng tương ứng với X_train.
- X_test: Chứa các đặc trưng dùng để thi.
- y_test: Chứa đáp án đúng của bài thi để so sánh với kết quả dự đoán của máy.

Huấn luyện mô hình

Sử dụng tập dữ liệu đã làm sạch (X_train, y_train) để "dạy" cho máy tính học các quy luật. Theo dàn ý ban đầu, chúng ta sẽ triển khai hai thuật toán phổ biến là Logistic Regression và Decision Tree.

Logistic Regression



Hình 1.1.4. Ma trận nhầm lẫn cho mô hình Logistic Regression

Kết quả đánh giá: Mô hình được đánh giá trên tập test bằng 4 chỉ số: Accuracy (tỷ lệ dự đoán đúng), Precision (độ chính xác khi dự đoán "sống sót"), Recall (khả năng nhận diện hành khách sống sót), F1-score (trung bình điều hòa giữa Precision và Recall).

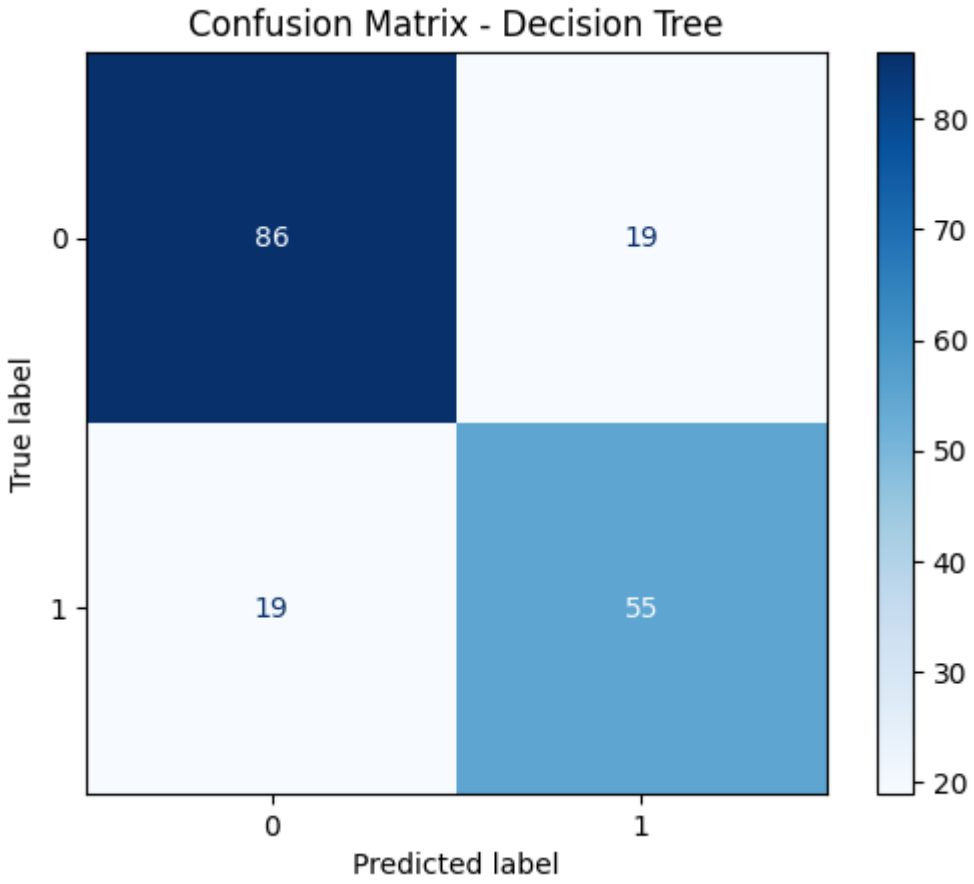
- Mô hình dự đoán đúng 81%. Đây là mức hiệu năng tốt cho bài toán Titanic với một mô hình cơ bản.
- Khi mô hình dự đoán một người là "Sống", độ tin cậy của dự đoán đó là gần 79%.
- Mô hình chỉ tìm ra được khoảng 74% tổng số người thực sự sống sót. Vẫn còn bỏ sót khoảng 26%.
- Chỉ số tổng hợp cho thấy sự cân bằng tương đối tốt giữa độ chính xác và độ phủ.

Nhận xét:

- Logistic Regression cho kết quả ổn định và ít bị quá khớp.
- Precision và Recall cân bằng → mô hình tin cậy

- Phù hợp với dữ liệu nhiều biến phân loại đã encode.

Decision Tree



Hình 1.1.5. Ma trận nhầm lẫn cho mô hình Decision Tree

Kết quả đánh giá:

- Accuracy là 78.77%: Mô hình dự đoán đúng khoảng 78.77% trường hợp.
- Precision (74.32%): Trong số những người mô hình dự đoán là sống sót, khoảng 74.32% dự đoán là đúng.
- Recall (74.32%): Trong số những người thực sự sống sót, mô hình tìm được (dự đoán đúng) 74.32% trong số họ.
- F1-Score (74.32%): Mô hình có một mức hiệu suất dự đoán cân bằng ở mức 74.32%.

Nhận xét:

- Accuracy cao nhưng biến động → dễ overfitting
- Precision và Recall không ổn định
- Decision Tree nhạy cảm với biến nhiễu

So sánh kết quả đánh giá của 2 mô hình

Chỉ số	Logistic Regression (LR)	Decision Tree (DT)	Nhận xét
Accuracy	0.8101	0.7877	LR dự đoán đúng tổng thể nhiều hơn (~2.24%).
Precision	0.7857	0.7432	LR ít nhầm lẫn khi dự đoán sống sót (False Positive thấp hơn).
Recall	0.7432	0.7432	Cả hai đều tìm đúng ~74% người sống sót.
F1-Score	Cao hơn DT	Thấp hơn LR	LR cân bằng tốt hơn giữa Precision và Recall.

Hình 1.1.6. So sánh kết quả đánh giá của hai mô hình LR và DT

Kết luận: Logistic Regression vượt trội hơn Decision Tree trên tập dữ liệu Titanic nhờ khả năng:

- Dự đoán chính xác hơn tổng thể (Accuracy cao hơn),
- Giảm nhầm lẫn khi dự đoán sống sót (Precision cao hơn),
- Duy trì khả năng phát hiện người sống sót tốt (Recall tương đương),
- Cân bằng tổng thể giữa Precision và Recall (F1-Score cao hơn một chút).

Nguyên nhân chính là dữ liệu Titanic có các đặc trưng quyết định sống sót chủ yếu là tuyến tính và dễ tách (Sex, Pclass), do đó mô hình tuyến tính như Logistic Regression hoạt động hiệu quả hơn so với Decision Tree, vốn có thể "overfit" vào các mối quan hệ phức tạp không thực sự quan trọng.

Đánh giá Random Forest bằng Cross-Validation (k = 5)

Cấu hình mô hình

Random Forest là một thuật toán học kết hợp (Ensemble Learning), hoạt động bằng cách xây dựng nhiều cây quyết định trong quá trình huấn luyện và đưa ra kết quả cuối cùng dựa trên cơ chế lấy số đông (Voting) hoặc trung bình.

Tham số cài đặt:

- n_estimators=100 (Mặc định): Sử dụng 100 cây quyết định trong rừng.
- criterion='gini' (Mặc định): Sử dụng chỉ số Gini để phân chia nút.

- random_state=42: Thiết lập hạt giống ngẫu nhiên cố định để đảm bảo kết quả có thể tái lập chính xác trong các lần chạy sau.

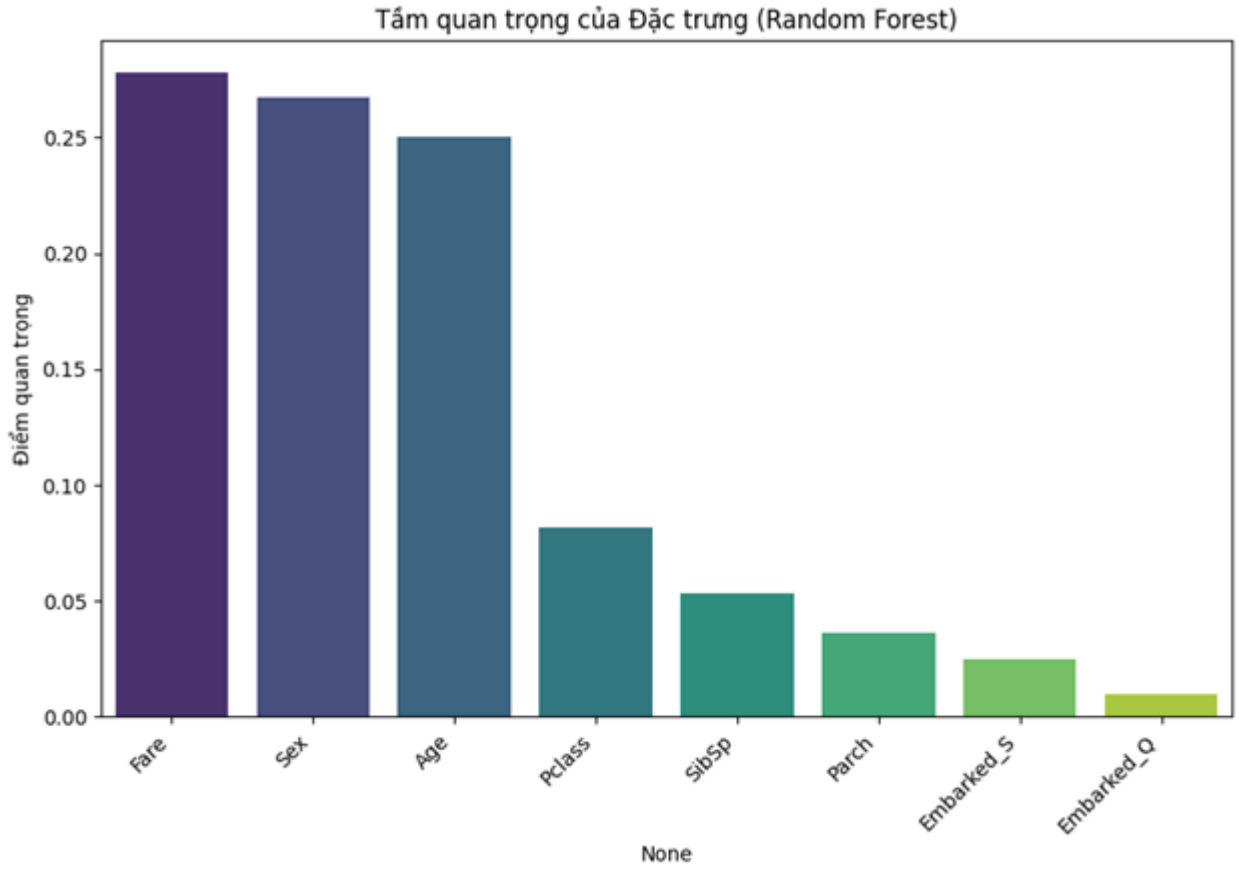
Phương pháp đánh giá: Chia toàn bộ tập dữ liệu X, y thành k=5 phần (folds) bằng nhau. Mô hình sẽ được huấn luyện 5 lần, mỗi lần dùng 4 phần để học và 1 phần để kiểm tra.

Kết quả cross_val_score (k=5)

Thực hiện kiểm định chéo với $k=5$ trên tập huấn luyện, kết quả độ chính xác (Accuracy) thu được qua 5 lần chạy như sau:

- Các điểm số chi tiết: [0.7821, 0.8034, 0.8483, 0.7809, 0.8090]
- Độ chính xác trung bình (Mean Accuracy): 80.47%
- Độ lệch chuẩn (Standard Deviation): 0.0245 (khoảng 2.45%)

Tầm quan trọng của đặc trưng



Hình 1.1.7. Biểu đồ so sánh tầm quan trọng của các đặc trưng (Random Forest)

Nhận xét:

- Fare, Sex, và Age là ba yếu tố chi phối quyết định của mô hình, chiếm tổng cộng khoảng gần 80% tổng tầm quan trọng (0.277946 + 0.267155 + 0.250022 ≃ 0.795123).
- Fare (Giá vé) lại là yếu tố quan trọng nhất, hơn cả Sex. Điều này cho thấy rằng việc người ta đã trả bao nhiêu tiền cho vé (thường liên quan đến cabin và vị trí) là một yếu tố dự đoán rất mạnh mẽ về sự sống sót.
- Pclass (Hạng vé) chỉ đứng thứ tư, cho thấy giá trị thực của vé (Fare) cung cấp thông tin chi tiết hơn so với phân loại chung (Pclass).

Nhận xét chung

Mức độ biến động giữa các lần chạy thấp (độ lệch chuẩn khoảng 2.45%) và điểm trung bình đạt trên 80% cho thấy mô hình Random Forest hoạt động ổn định và có độ tin cậy cao hơn so với Decision Tree đơn lẻ.

Lưu file dữ liệu

Sau khi hoàn tất quá trình làm sạch, xử lý giá trị thiếu, mã hóa và chuẩn hóa, chúng ta cần lưu lại bộ dữ liệu "sạch" này giúp tái sử dụng và dễ dàng chia sẻ.

Kết luận

- Về Dữ liệu:
  - Quy trình tiền xử lý (điền Age bằng trung vị, loại bỏ Cabin) đã giải quyết hiệu quả vấn đề dữ liệu thiếu.
  - Ba yếu tố quyết định khả năng sống sót là: Giới tính (Nữ), Hạng vé (Hạng 1) và Tuổi (Trẻ em).
- Về Mô hình:
  - Logistic Regression đạt độ chính xác cao nhất trên tập test (81%), trong khi Decision Tree có dấu hiệu overfitting (~78.8%).
  - Random Forest được đánh giá là mô hình tối ưu nhất nhờ tính ổn định cao qua kiểm định chéo (Accuracy trung bình ~80.5%, độ lệch chuẩn thấp 2.45%).
- Hướng phát triển:
  - Có thể nâng cao độ chính xác (>85%) bằng kỹ thuật Feature Engineering (trích xuất Danh xưng, tạo biến Kích thước gia đình).

Khám phá dữ liệu Iris

Chuẩn bị dữ liệu

Giới thiệu đề bài

Bài tập yêu cầu sinh viên thực hiện phân tích khám phá dữ liệu (EDA) trên bộ dữ liệu Iris – một bộ dữ liệu kinh điển trong học máy, gồm ba loài hoa: Setosa, Versicolor và Virginica. Mục tiêu chính:

- Hiển thị thông tin đặc trưng và nhãn.
- Tính toán thống kê mô tả (mean, std).
- Trực quan hóa mối quan hệ giữa các đặc trưng bằng pairplot.
- Chuẩn hóa dữ liệu về thang giá trị [0,1] bằng MinMaxScaler.
- Giải thích ý nghĩa của chuẩn hóa trong học máy.

Nội dung bài giúp sinh viên hiểu rõ cấu trúc dataset, mối tương quan giữa thuộc tính, và vai trò của tiền xử lý dữ liệu.

Công cụ và thư viện sử dụng

Dự án sử dụng ngôn ngữ Python cùng các thư viện:

- Scikit-learn (sklearn): Tải dữ liệu mẫu (load_iris) và thực hiện chuẩn hóa (MinMaxScaler).
- Pandas (pd): Tạo và thao tác với DataFrame để hiển thị dữ liệu dưới dạng bảng.
- Seaborn (sns) & Matplotlib (plt): Vẽ biểu đồ trực quan hóa dữ liệu.

Mô tả các thuộc tính trong bộ dữ liệu

Mỗi mẫu hoa được mô tả bởi 4 thuộc tính định lượng:

- sepal length (cm): Chiều dài đài hoa.
- sepal width (cm): Chiều rộng đài hoa.
- petal length (cm): Chiều dài cánh hoa.
- petal width (cm): Chiều rộng cánh hoa.

Species (Nhãn): Loài hoa, được mã hóa thành số nguyên (0: Setosa, 1: Versicolor, 2: Virginica).

Khám phá dữ liệu

Hiển thị thông tin các đặc trưng và nhãn

Load dữ liệu sau đó chuyển đổi dữ liệu thành DataFrame và xuất ra màn hình.

- Xuất ra màn hình 4 đặc trưng:Sepal Length, Sepal Width, Petal Length, Petal Width
- Nhãn (target): Loài hoa (*Setosa*, *Versicolor*, *Virginica*).

Tính trung bình và độ lệch chuẩn từng đặc trưng

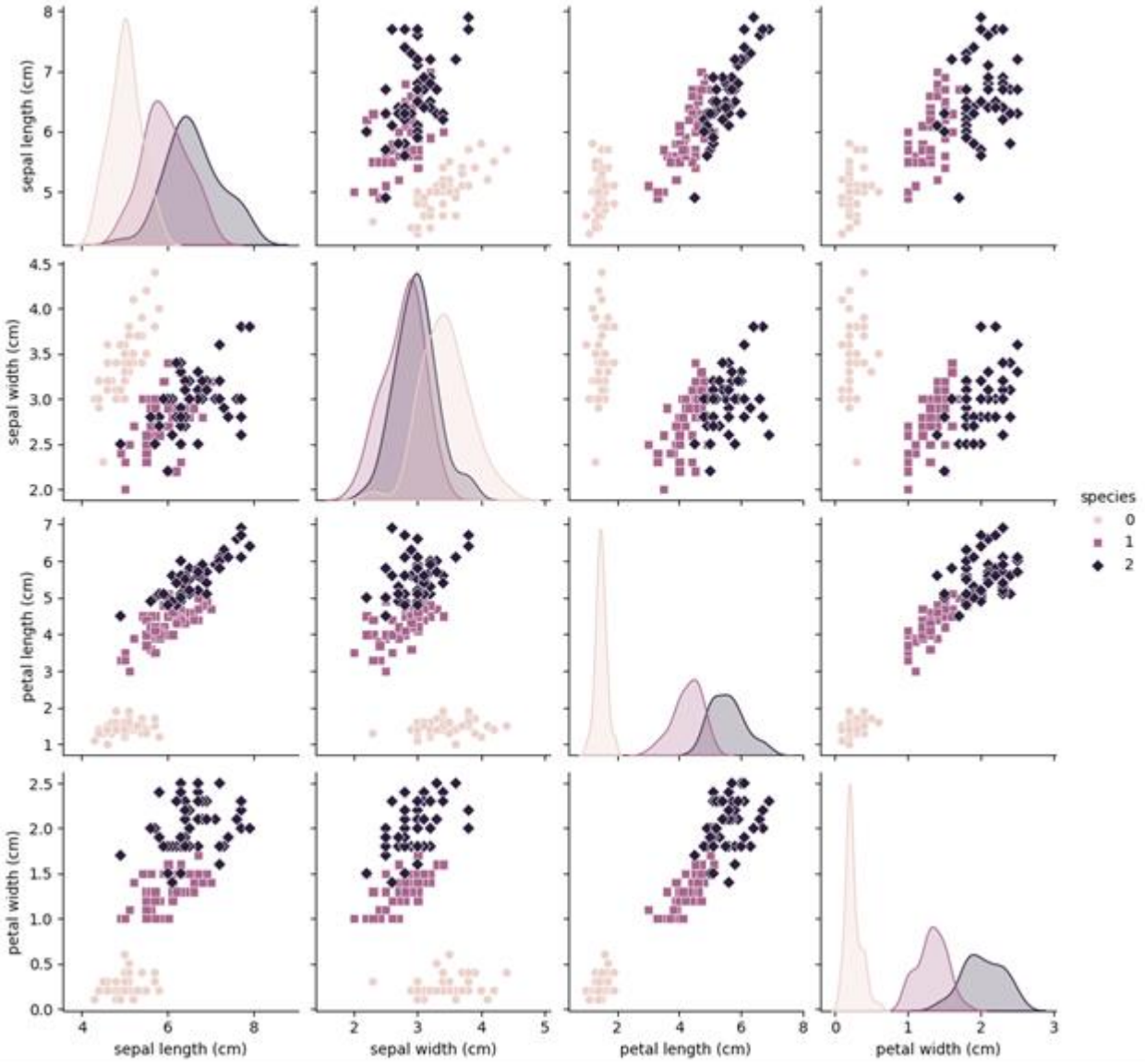
Kết quả thống kê mô tả cho toàn bộ dataset như sau:

Đặc trưng	**Trung bình (Mean)**	**Độ lệch chuẩn (Std)**	**Nhận xét sơ bộ**
sepal length	5.84 cm	0.83	Đài hoa có kích thước dài nhất.
sepal width	3.06 cm	0.44	Độ biến thiên thấp nhất (dữ liệu tập trung).
petal length	3.76 cm	1.77	Độ biến thiên cao nhất, cho thấy sự khác biệt lớn giữa các loài về chiều dài cánh hoa.
petal width	1.20 cm	0.76	Kích thước nhỏ nhất.

Bảng 1.2.1. Thống kê mô tả bộ dữ liệu Iris

Trực quan hóa dữ liệu

Biểu đồ cặp thể hiện mối tương quan giữa các đặc trưng (sns.pairplot). Sử dụng sns.pairplot để vẽ biểu đồ phân tán cho từng cặp đặc trưng, kết hợp tô màu theo loài (hue='species').



Hình 1.2.1. Biểu đồ cặp thể hiện mối tương quan giữa các đặc trưng

Kết quả từ biểu đồ cho thấy:

Kết quả	**Ý nghĩa**
Setosa tách biệt hoàn toàn khỏi hai loài khác	Có thể dễ dàng phân loại chỉ với 1–2 đặc trưng
Versicolor và Virginica chồng lấn nhẹ	Cần mô hình học máy để phân biệt chính xác
Petal length ↔ Petal width có tương quan mạnh	Hai đặc trưng này rất hữu ích cho phân loại
Sepal width ít thay đổi	Ảnh hưởng nhỏ hơn trong dự đoán

Bảng 2.2.2. Kết quả phân tích từ biểu đồ tương quan giữa các đặc trưng

Phân tích mối tương quan giữa các đặc trưng Iris

- Loài *Setosa* tách biệt hoàn toàn khỏi hai loài khác → Có thể dễ dàng phân loại chỉ với một vài đặc trưng.
- Loài *Versicolor* và *Virginica* có sự chồng lấn nhẹ → Cần sử dụng các mô hình học máy (như Decision Tree, SVM) để phân biệt chính xác.
- Có mối tương quan mạnh giữa *Petal length* và *Petal width* → Hai đặc trưng này mang thông tin phân loại cao.
- *Sepal width* có độ biến thiên nhỏ và khả năng phân tách thấp → Ảnh hưởng ít hơn trong việc dự đoán loài.

Kết luận:

- Hai đặc trưng liên quan đến cánh hoa (Petal length, Petal width) thể hiện khả năng phân biệt rõ rệt giữa các loài hoa Iris, đặc biệt là giúp tách loài Setosa ra khỏi các loài còn lại.
- Trong khi đó, hai đặc trưng về đài hoa (Sepal length, Sepal width) có mức độ tương quan thấp hơn và đóng vai trò phụ trong quá trình phân loại.

Chuẩn hóa dữ liệu bằng MinMaxScaler

Mô tả kỹ thuật chuẩn hóa

MinMaxScaler là kỹ thuật biến đổi dữ liệu để đưa tất cả các giá trị về một khoảng cố định, thường là [0, 1]. Công thức toán học cho mỗi điểm dữ liệu x:

$$X_{scaled} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Hình 1.2.2. Công thức MinMaxScaler

Lý do chọn MinMaxScaler

Trong bộ dữ liệu Iris, các đặc trưng có biên độ giá trị khác nhau (ví dụ: sepal length từ 4.3-7.9, trong khi petal width từ 0.1-2.5). Sự chênh lệch này có thể khiến các thuật toán học máy dựa trên khoảng cách (như KNN, K-Means) bị thiên vị, coi trọng

đặc trưng có giá trị lớn hơn. Và MinMaxScaler là biện pháp phù hợp với các thuật toán nhạy cảm với thang đo (KNN, SVM, Neural Network) như này.

MinMaxScaler giúp đưa tất cả về cùng một hệ quy chiếu [0, 1] mà vẫn giữ nguyên phân phối gốc của dữ liệu. Không làm mất quan hệ tương đối giữa các giá trị.

Kết quả sau chuẩn hóa

- Các đặc trưng đều nằm trong khoảng [0, 1].
- Giúp mô hình học máy hoạt động ổn định, tránh bị "ảnh hưởng quá mức" bởi cột có giá trị lớn.

Ý nghĩa của chuẩn hóa trong học máy

- Đảm bảo tính công bằng giữa các đặc trưng.
- Tăng tốc độ hội tụ.
- Cải thiện độ chính xác.

Kết luận

Qua phần phân tích bộ dữ liệu Iris, chúng ta đã thấy được cấu trúc rõ ràng của dữ liệu và sự phân cụm tự nhiên giữa các loài hoa. Việc áp dụng kỹ thuật chuẩn hóa MinMaxScaler là bước tiền xử lý cần thiết để đảm bảo dữ liệu đầu vào đạt chuẩn, tạo tiền đề cho việc xây dựng các mô hình học máy có độ chính xác và hiệu suất cao sau này.

Kết luận chương

Chương 2 đã hoàn thành các mục tiêu đề ra về việc làm quen và xử lý dữ liệu dạng bảng thông qua hai bài toán thực nghiệm Titanic và Iris.

Practice 2 (Customer Churn)

Chuẩn bị dữ liệu

Giới thiệu đề bài

Bài thực hành yêu cầu sinh viên xây dựng mô hình phân loại để dự đoán khách hàng có rời bỏ dịch vụ (churn) hay không, sử dụng bộ dữ liệu Kaggle – Customer Churn Dataset.

Theo đúng yêu cầu được giao, sinh viên cần thực hiện các bước:

- Chuẩn bị dữ liệu theo quy trình tiền xử lý đã học ở slide Chương 4.
- Lựa chọn 5 đặc trưng quan trọng: Tenure, MonthlyCharges, ContractType, InternetService, PaymentMethod.
- Chia dữ liệu thành tập train/test.
- Huấn luyện ba mô hình phân loại gồm:
  - Logistic Regression
  - Random Forest
  - Support Vector Machine (SVM)
- Đánh giá mô hình bằng các chỉ số: Accuracy, Precision, Recall, F1.

Ngoài các bước trên, bài thực hành còn bao gồm hai yêu cầu mở rộng:

- (Yêu cầu 2) Áp dụng Cross-validation (k = 5) để đánh giá đúng mức độ ổn định và khả năng tổng quát hóa của mô hình Random Forest.
- (Yêu cầu 4) Giải thích vì sao một mô hình đạt kết quả kỹ thuật tốt (chỉ số đẹp) nhưng chưa chắc phù hợp về mặt nghiệp vụ, đặc biệt trong ngữ cảnh bài toán churn — nơi FP và FN ảnh hưởng trực tiếp đến chi phí chăm sóc khách hàng và doanh thu.

Công cụ và thư viện sử dụng

Sử dụng các thư viện Python sau được sử dụng nhằm hỗ trợ xử lý dữ liệu, trực quan hóa và xây dựng mô hình phân loại:

- Pandas: dùng để đọc dữ liệu, thao tác bảng và xử lý các bước tiền xử lý như lọc cột, kiểm tra thiếu, mô tả dữ liệu.
- NumPy: hỗ trợ các phép tính nền tảng và xử lý mảng số.
- Matplotlib và Seaborn: dùng để trực quan hóa dữ liệu, vẽ histogram, biểu đồ phân phối, heatmap tương quan và confusion matrix.
- Scikit-learn: thư viện chính phục vụ cho toàn bộ quy trình xây dựng và đánh giá mô hình,

Mô tả các thuộc tính trong bộ dữ liệu

Bộ dữ liệu Telco Customer Churn bao gồm 21 thuộc tính, phản ánh thông tin cá nhân, dịch vụ viễn thông mà khách hàng sử dụng và trạng thái rời bỏ dịch vụ (Churn). Ý nghĩa của từng thuộc tính được mô tả như sau:

- customerID: Mã định danh duy nhất của mỗi khách hàng.
- gender: Giới tính của khách hàng.
- SeniorCitizen: Khách hàng có thuộc nhóm người cao tuổi hay không (1 = Có, 0 = Không).
- Partner: Khách hàng có vợ/chồng hoặc bạn đời hay không.
- Dependents: Khách hàng có người phụ thuộc hay không.
- tenure: Số tháng khách hàng đã sử dụng dịch vụ.
- PhoneService: Khách hàng có dùng dịch vụ điện thoại hay không.
- MultipleLines: Có sử dụng nhiều đường dây điện thoại hay không.
- InternetService: Loại dịch vụ Internet mà khách hàng đăng ký (DSL, Fiber optic, None).
- OnlineSecurity: Có sử dụng dịch vụ bảo mật trực tuyến hay không.
- OnlineBackup: Có sử dụng dịch vụ sao lưu trực tuyến hay không.
- DeviceProtection: Dịch vụ bảo vệ thiết bị.
- TechSupport: Dịch vụ hỗ trợ kỹ thuật.
- StreamingTV: Dịch vụ truyền hình trực tuyến.
- StreamingMovies: Dịch vụ xem phim trực tuyến.
- Contract: Loại hợp đồng đã ký (Month-to-month, One year, Two year).
- PaperlessBilling: Hình thức thanh toán không giấy tờ (e-billing).
- PaymentMethod: Phương thức thanh toán (Credit card, Bank transfer, Electronic check, Mail check).
- MonthlyCharges: Phí dịch vụ khách hàng trả hàng tháng.
- TotalCharges: Tổng chi phí khách hàng đã chi trả.
- Churn: Biến mục tiêu, cho biết khách hàng có rời bỏ dịch vụ hay không.

Khám phá dữ liệu

Dữ liệu được nạp bằng Pandas với **7.043 dòng và 21 cột**. Qua kiểm tra với df.info(), toàn bộ các thuộc tính đều không có giá trị thiếu. Dataset bao gồm ba dạng dữ liệu chính:

- Biến số: tenure, SeniorCitizen, MonthlyCharges
- Biến phân loại: các thuộc tính dịch vụ như Contract, InternetService, PaymentMethod,...
- Biến mục tiêu: Churn

Kiểm tra chất lượng dữ liệu (Missing Values – Outliers – Duplicates)

Dữ liệu được kiểm tra chất lượng ban đầu thông qua ba bước: giá trị thiếu, ngoại lai và trùng lặp. Kết quả cho thấy toàn bộ 21 cột đều không có giá trị thiếu, với df.isnull().sum() trả về 0 cho tất cả các thuộc tính.

Ba biến số tenure, MonthlyCharges và TotalCharges được kiểm tra ngoại lai bằng phương pháp Interquartile Range (IQR). Các khoảng Lower–Upper lần lượt cho từng biến đều bao trọn toàn bộ giá trị quan sát, dẫn đến số lượng ngoại lai bằng 0 ở cả ba thuộc tính. Điều này cho thấy phân phối của dữ liệu ổn định và không cần áp dụng capping hoặc transformation.

Cuối cùng, kiểm tra trùng lặp bằng df.duplicated() cho thấy 7.043/7.043 dòng dữ liệu đều là duy nhất, không tồn tại bản ghi trùng, do đó không cần thực hiện thêm bước loại bỏ dữ liệu trùng lặp.

Thống kê mô tả

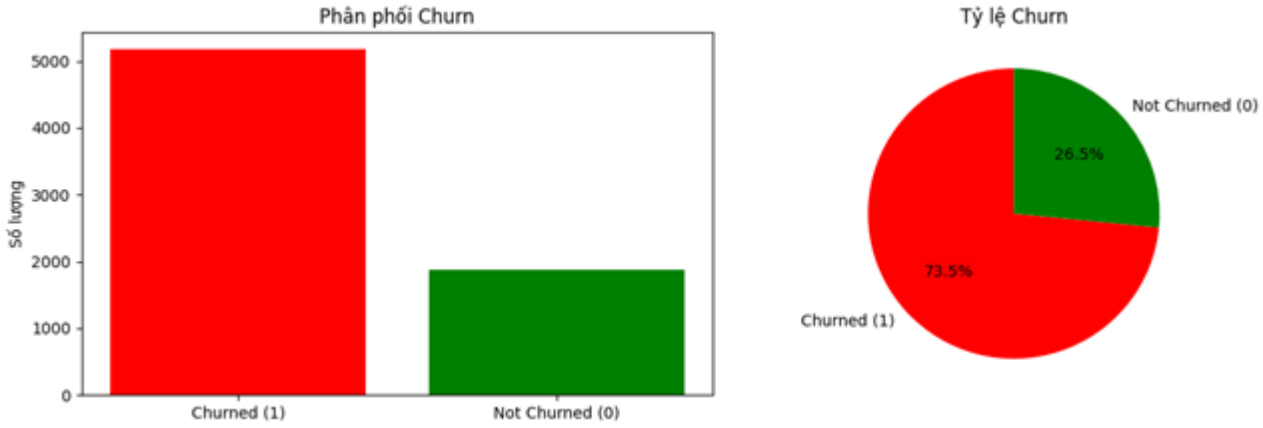
Kết quả thống kê mô tả cho bốn biến định lượng cho thấy: SeniorCitizen có giá trị trung bình 0.162 (~16% khách hàng là người cao tuổi); tenure trung bình 32.37 tháng, dao động 0–72 với độ lệch chuẩn ~24; MonthlyCharges trung bình 64.76 USD, trải từ 18.25–118.75 USD với độ lệch chuẩn 30.09; TotalCharges trung bình khoảng 2,283 USD và biến thiên rộng do là chi phí tích lũy.

Tổng quan, cả bốn biến định lượng đều có mức phân tán rõ rệt và phân phối hợp lý, không xuất hiện dấu hiệu bất thường.

Kiểm tra mất cân bằng nhãn

Phân phối của biến mục tiêu Churn được kiểm tra bằng cách đếm tần suất và tính tỷ lệ phần trăm của từng lớp. Kết quả cho thấy sự chênh lệch rõ rệt:

- No (không rời bỏ): 5.174 khách hàng — 73.46%
- Yes (rời bỏ): 1.869 khách hàng — 26.54%

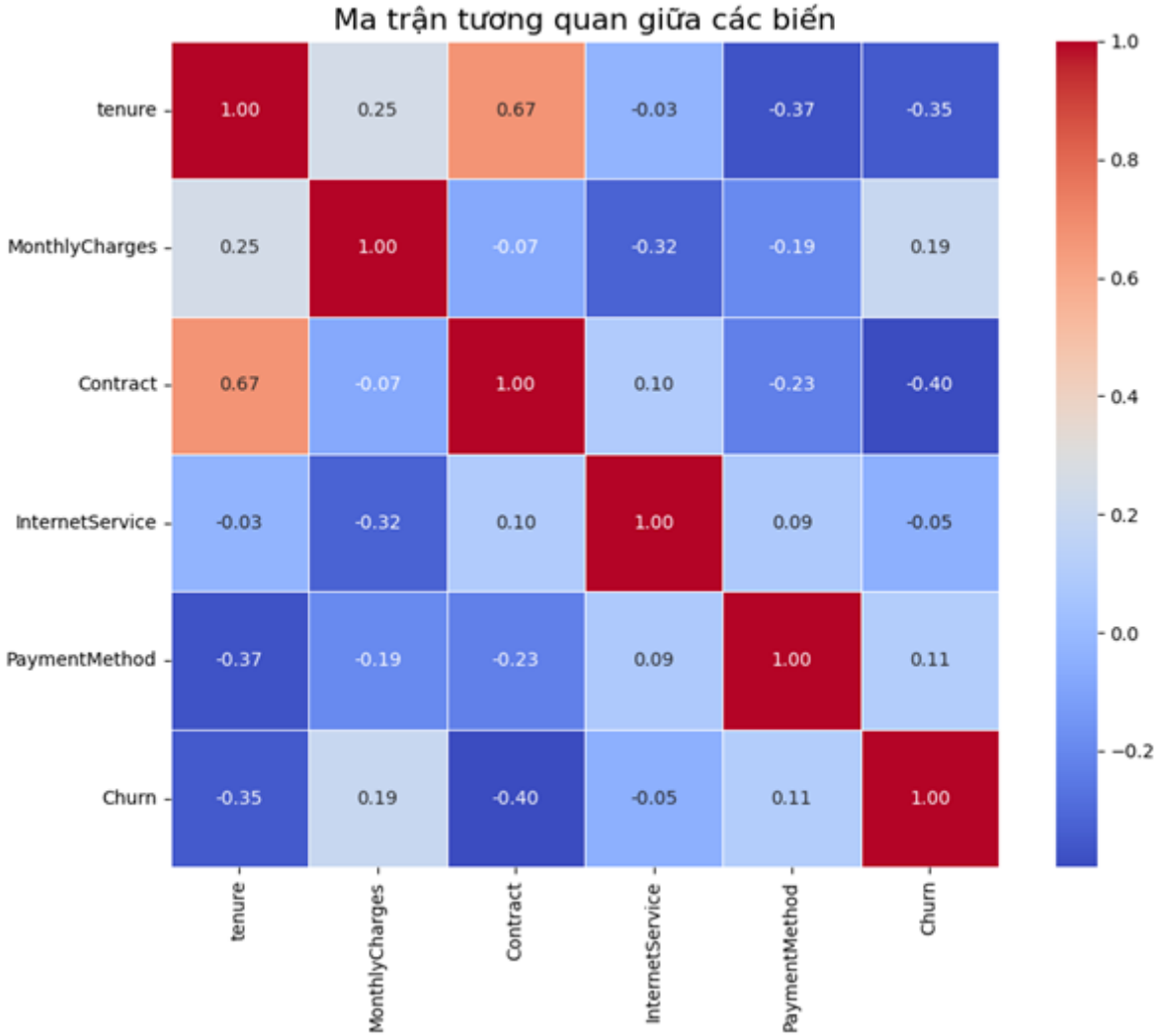


Hình 2.2.1. Trực quan hóa mất cân bằng nhãn Churn

Biến mục tiêu mất cân bằng rõ rệt, với lớp No chiếm gần ba phần tư dữ liệu.

Ma trận tương quan giữa các biến

Ma trận tương quan được sử dụng để quan sát mức độ liên hệ giữa các biến đầu vào và biến mục tiêu Churn. Trong bài thực hành, nhóm chỉ xét các biến theo đúng yêu cầu đề bài: tenure, MonthlyCharges, Contract, InternetService, PaymentMethod cùng với cột mục tiêu Churn.



Hình 2.2.2. Ma trận tương quan giữa các biến đầu vào và biến mục tiêu Churn

- Contract – Churn: –0.40: Tương quan âm khá mạnh. Khách hàng ký hợp đồng dài hạn (1–2 năm) có xu hướng ít churn hơn so với hợp đồng theo tháng → hoàn toàn hợp lý về mặt nghiệp vụ.
- tenure – Churn: –0.35: Khách hàng gắn bó lâu hơn thì tỷ lệ rời bỏ thấp hơn. Đây là biến quan trọng trong nhiều mô hình churn thực tế.
- MonthlyCharges – Churn: 0.19: Tương quan dương nhẹ: phí hàng tháng càng cao thì khả năng churn có xu hướng tăng, nhưng mức độ không mạnh.
- InternetService – Churn: –0.05: Tương quan rất yếu. Loại hình Internet không ảnh hưởng quá lớn đến churn ở dữ liệu này.

- PaymentMethod – Churn: 0.11: Tương quan yếu: phương thức thanh toán chỉ ảnh hưởng nhẹ.

Cơ sở lựa chọn 5 đặc trưng cho bài toán Churn

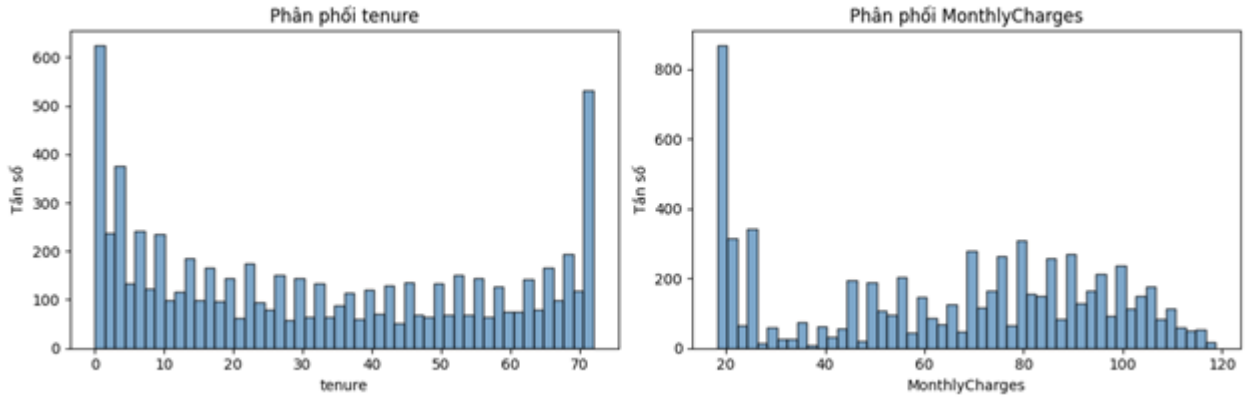
Bài thực hành yêu cầu sử dụng năm đặc trưng: **tenure**, **MonthlyCharges**, **Contract**, **InternetService** và **PaymentMethod**.

- tenure phản ánh thời gian gắn bó, giá trị càng cao thì khả năng churn càng thấp.
- MonthlyCharges thể hiện mức phí; chi phí càng cao thì khách càng dễ cân nhắc rời mạng.
- Contract là yếu tố ràng buộc chính: hợp đồng theo tháng luôn có tỷ lệ churn cao hơn hợp đồng 1–2 năm.
- InternetService liên quan đến trải nghiệm dịch vụ, dễ ảnh hưởng đến sự hài lòng.
- PaymentMethod phản ánh mức thuận tiện trong thanh toán; những phương thức kém ổn định thường xuất hiện nhiều ở nhóm churn.

Kết quả tương quan cho thấy **tenure** và **Contract** có tương quan âm rõ rệt với Churn, mang giá trị dự báo tốt. Các biến còn lại có tương quan thấp hơn nhưng vẫn mang ý nghĩa hành vi rõ ràng và phù hợp mục tiêu bài toán.

Phân phối các đặc trưng được chọn

- Về các biến số:



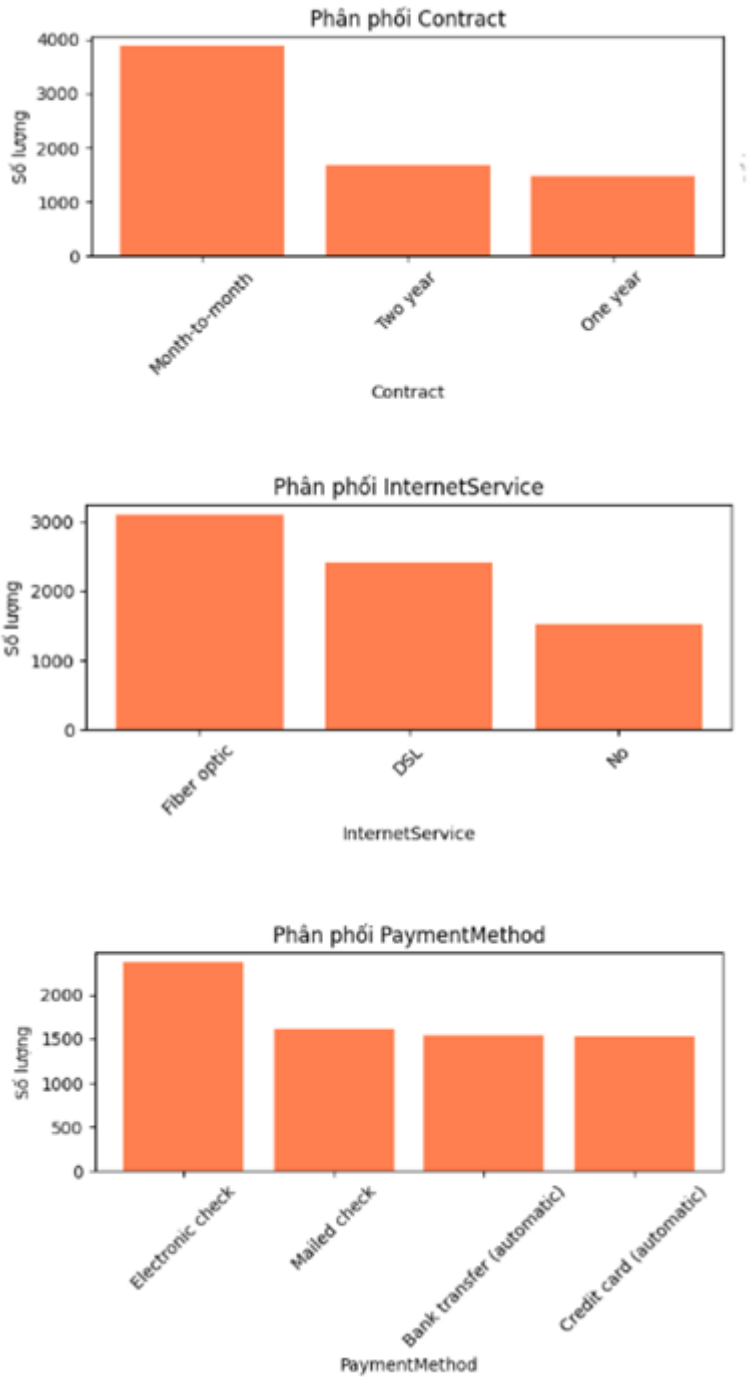
Hình 2.2.3. Phân phối của biến tenure và MonthlyCharges

Phân phối tenure cho thấy hai cụm rõ: nhóm mới dùng (0–5 tháng) và nhóm dùng rất lâu (60–72 tháng) – cũng là hai nhóm churn cao trong thực tế. Dạng phân phối "lõm giữa" phản ánh đúng hành vi ngành viễn thông và cho thấy tenure phân biệt churn khá tốt.

MonthlyCharges tập trung nhiều ở mức 20–30 USD, còn các gói cao 70–110 USD rải rác, phân phối lệch phải. Dù có thể log transform, dữ liệu này không cần vì mức trải dài không quá lớn và các mô hình dùng trong bài không yêu cầu phân phối chuẩn. Nhóm vì thế giữ nguyên biến.

Tóm lại, tenure và MonthlyCharges đều mô tả rõ hành vi sử dụng (mức gắn bó và mức chi tiêu) và vẫn phù hợp đưa vào mô hình churn dù phân phối không chuẩn.

- Về các biến phân loại:



Hình 2.2.4. Phân phối ba biến phân loại: Contract, InternetService, PaymentMethod

- Phân phối Contract cho thấy nhóm *Month-to-month* chiếm nhiều nhất, phù hợp với việc hợp đồng theo tháng có rủi ro churn cao và tương quan âm rõ với Churn.

- InternetService tập trung chủ yếu ở *Fiber optic*, tiếp theo là DSL; nhóm này thường churn cao hơn do chi phí lớn và kỳ vọng chất lượng cao.
- PaymentMethod nổi bật ở *Electronic check*, một phương thức kém tiện lợi nên thường xuất hiện nhiều hơn ở nhóm churn.

Tổng thể, ba biến này phản ánh đúng các yếu tố ảnh hưởng đến churn: mức ràng buộc, loại dịch vụ và sự thuận tiện thanh toán. Các khác biệt phân phối đặc biệt Month-to-month và Electronic check giải thích rõ vì sao churn tập trung vào một số hành vi nhất định và củng cố lý do chọn năm đặc trưng trong bài toán.

Tiền xử lý dữ liệu (Data Preprocessing)

- Bước 1: Mã hóa và chuẩn hóa kiểu dữ liệu: Ba biến phân loại Contract, InternetService và PaymentMethod được mã hóa bằng One-Hot Encoding để tách từng nhóm thành cột nhị phân (0/1). Sau mã hóa, số đặc trưng đầu vào tăng từ 5 lên 9 features (không tính Churn), và các giá trị Boolean được chuẩn hóa về 0/1 để đảm bảo sự thống nhất trong toàn bộ dataset.
- Bước 2: Tách dữ liệu và chia train/test: Dữ liệu được tách thành X (9 đặc trưng) và y (nhãn Churn dạng 0/1), sau đó chia theo tỷ lệ 80/20, tương ứng khoảng 5.600 mẫu train và 1.400 mẫu test. Tham số stratify=y được sử dụng để giữ nguyên tỷ lệ mất cân bằng lớp (73% Non-churn, 27% Churn), giúp mô hình được đánh giá đúng bản chất của dữ liệu gốc.
- Bước 3: Chuẩn hóa hai biến số tenure và MonthlyCharges: Hai biến số được chuẩn hóa bằng StandardScaler để đưa về cùng thang đo, giúp Logistic Regression và SVM hoạt động ổn định. Scaler chỉ được fit trên tập train, sau đó mới transform cho cả train và test nhằm tránh data leakage, tức mô hình "nhìn trước" phân phối dữ liệu test.

Kết thúc ba bước, toàn bộ dữ liệu đều ở dạng số hóa, nhất quán, không bị rò rỉ thông tin và sẵn sàng để tiến hành huấn luyện mô hình ở phần tiếp theo.

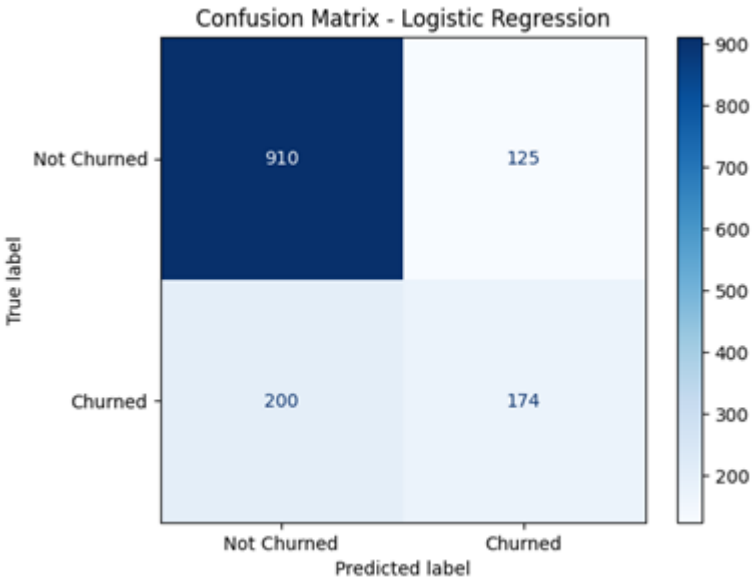
Huấn luyện mô hình

Dữ liệu sau tiền xử lý đã sẵn sàng để đưa vào huấn luyện.

Logistic Regression

Mô hình Logistic Regression được huấn luyện trên 9 đặc trưng đầu vào sau khi hoàn tất các bước mã hóa và chuẩn hóa. Kết quả đánh giá trên tập test cho thấy:

- Accuracy: 0.7693
- Precision: 0.5819
- Recall: 0.4652
- F1-score: 0.5170



Hình 2.4.1. Ma trận nhầm lẫn – Logistic Regression

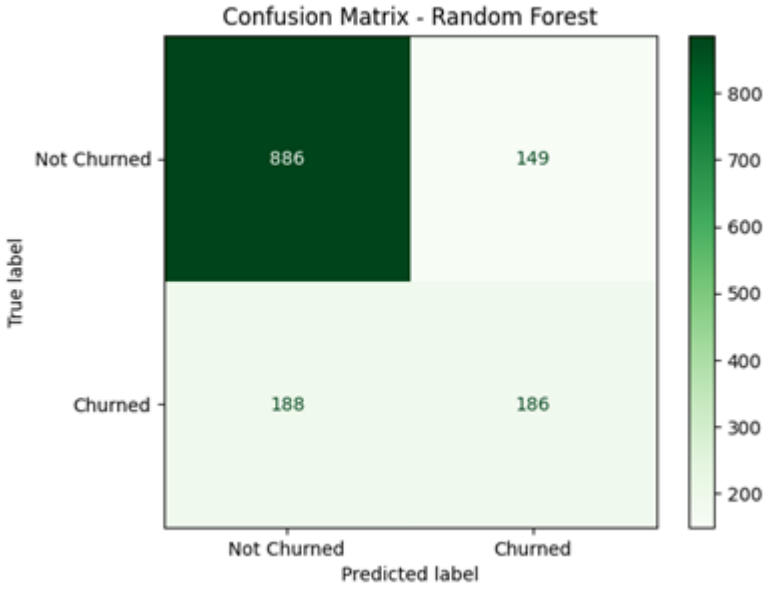
Ma trận nhầm lẫn cho Logistic Regression cho thấy mô hình dự đoán tốt lớp không rời bỏ, với 910 TN và 125 FP, thể hiện khả năng hạn chế "gọi nhầm" khách trung thành. Tuy nhiên, mô hình bỏ sót 200 khách churn (FN), làm Recall của lớp churn giảm mạnh và cho thấy LR không nhạy với lớp thiểu số, lớp quan trọng nhất trong bài toán churn.

Tổng thể, Logistic Regression ổn định và dễ diễn giải nhưng không phải lựa chọn tối ưu khi mục tiêu là tăng khả năng nhận diện churn. Các mô hình tiếp theo sẽ được so sánh để đánh giá khả năng xử lý lớp thiểu số tốt hơn.

Random Forest

Random Forest cho hiệu suất ổn định với accuracy khoảng 0.76, precision 0.55 và recall 0.49, cao hơn Logistic Regression ở khả năng nhận diện lớp churn. Ma trận

nhầm lẫn (Hình 3.x) cho thấy mô hình dự đoán đúng 186 khách churn (TP) và bỏ sót 188 trường hợp (FN). So với Logistic Regression, số TP tăng lên và FN giảm xuống, cho thấy RF bắt được tín hiệu churn tốt hơn.



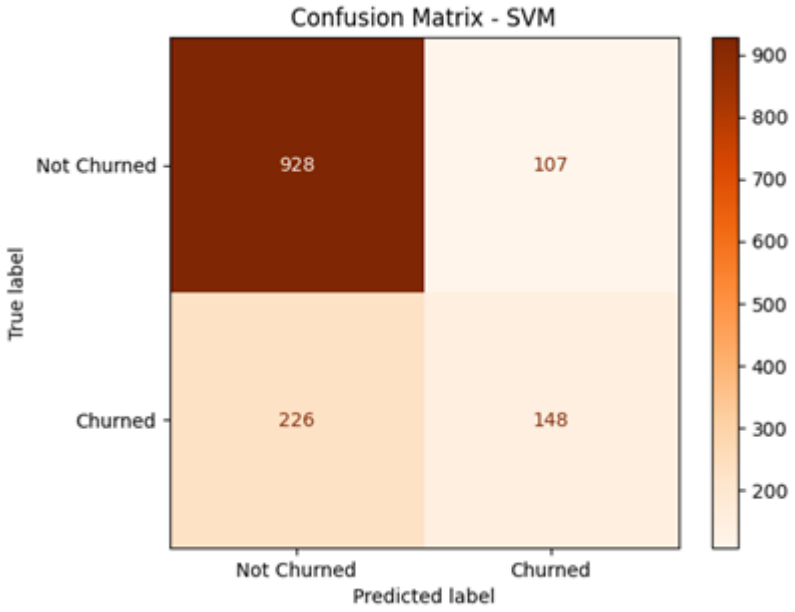
Hình 2.4.2. Ma trận nhầm lẫn của mô hình Random Forest

Tuy nhiên, số FP tăng lên 149, thể hiện sự đánh đổi quen thuộc của các mô hình ensemble: nhận diện churn tốt hơn nhưng dễ "gọi nhầm" một phần khách không rời bỏ. Với bài toán churn, nơi ưu tiên giảm tối đa số FN, mức đánh đổi này được xem là chấp nhận được.

Tổng thể, Random Forest tận dụng tốt quan hệ phi tuyến trong dữ liệu, giữ cân bằng giữa hai lớp và phù hợp hơn Logistic Regression trong mục tiêu phát hiện khách hàng có nguy cơ churn.

Support Vector Machine (SVM)

Mô hình SVM với kernel RBF cho kết quả accuracy khoảng 0.76, precision 0.58, nhưng recall chỉ đạt 0.39, thấp nhất trong ba mô hình. Điều này cho thấy SVM dự đoán ổn ở mức tổng thể nhưng gặp khó khi nhận diện lớp churn — lớp quan trọng trong bài toán.



Hình 2.4.3. Ma trận nhầm lẫn của mô hình SVM

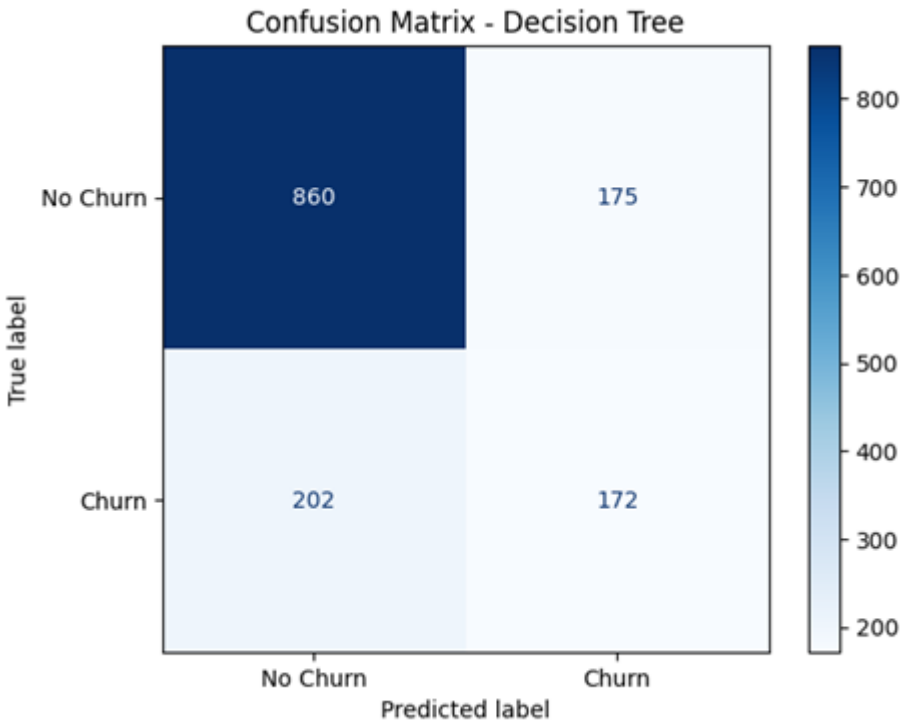
Ma trận nhầm lẫn cho thấy SVM dự đoán đúng 928 khách không churn (TN) và có FP thấp nhất (107) trong ba mô hình. Điều này cho thấy SVM hoạt động rất tốt ở lớp đa số và có xu hướng phân loại "an toàn".

Tuy vậy, mô hình chỉ nhận diện được 148 khách churn (TP) và bỏ sót 226 trường hợp (FN), mức FN cao nhất. Điều này phản ánh rõ sự lệch về lớp không churn khi dữ liệu mất cân bằng, khiến hiệu quả nhận diện churn rất thấp.

Tổng thể, dù SVM có độ chính xác cao ở lớp không churn, FN lớn khiến mô hình không phù hợp cho mục tiêu nghiệp vụ, nơi việc phát hiện đúng khách hàng chuẩn bị rời bỏ dịch vụ quan trọng hơn nhiều.

Decision Tree

Mô hình Decision Tree cho kết quả thấp hơn so với ba mô hình trước, với accuracy khoảng 0.73, precision 0.49, recall 0.46 và F1-score 0.47. Các chỉ số đều ở mức trung bình, phản ánh tính chất dễ overfit và kém ổn định của cây quyết định trên dữ liệu thực tế.



Hình 2.4.4. Ma trận nhầm lẫn của mô hình Decision Tree

Từ ma trận nhầm lẫn, Decision Tree dự đoán tương đối đồng đều giữa hai lớp nhưng không nổi trội ở mặt nào. Số FN vẫn cao, nên khả năng phát hiện churn còn yếu; trong khi FP cũng không thấp, cho thấy mô hình không cân bằng tốt giữa hai nhóm khách hàng. Đây là tình huống thường gặp khi cây quyết định chưa được tối ưu bằng pruning hoặc khi dữ liệu chứa nhiều nhiễu.

Về tổng thể, Decision Tree cho hiệu suất thấp hơn Logistic Regression, Random Forest và SVM, đồng thời độ ổn định kém. Với đặc thù bài toán churn cần nhận diện chính xác lớp thiểu số, Decision Tree đơn lẻ không phải lựa chọn phù hợp nếu không được tăng cường bởi các kỹ thuật ensemble như Random Forest hoặc Gradient Boosting.

So sánh hiệu năng các mô hình

Sau khi huấn luyện bốn mô hình gồm Logistic Regression, Random Forest, SVM và Decision Tree, các chỉ số đánh giá Accuracy – Precision – Recall – F1 được tổng hợp lại trong dưới đây:

Model	Accuracy	Precision	Recall	F1-score

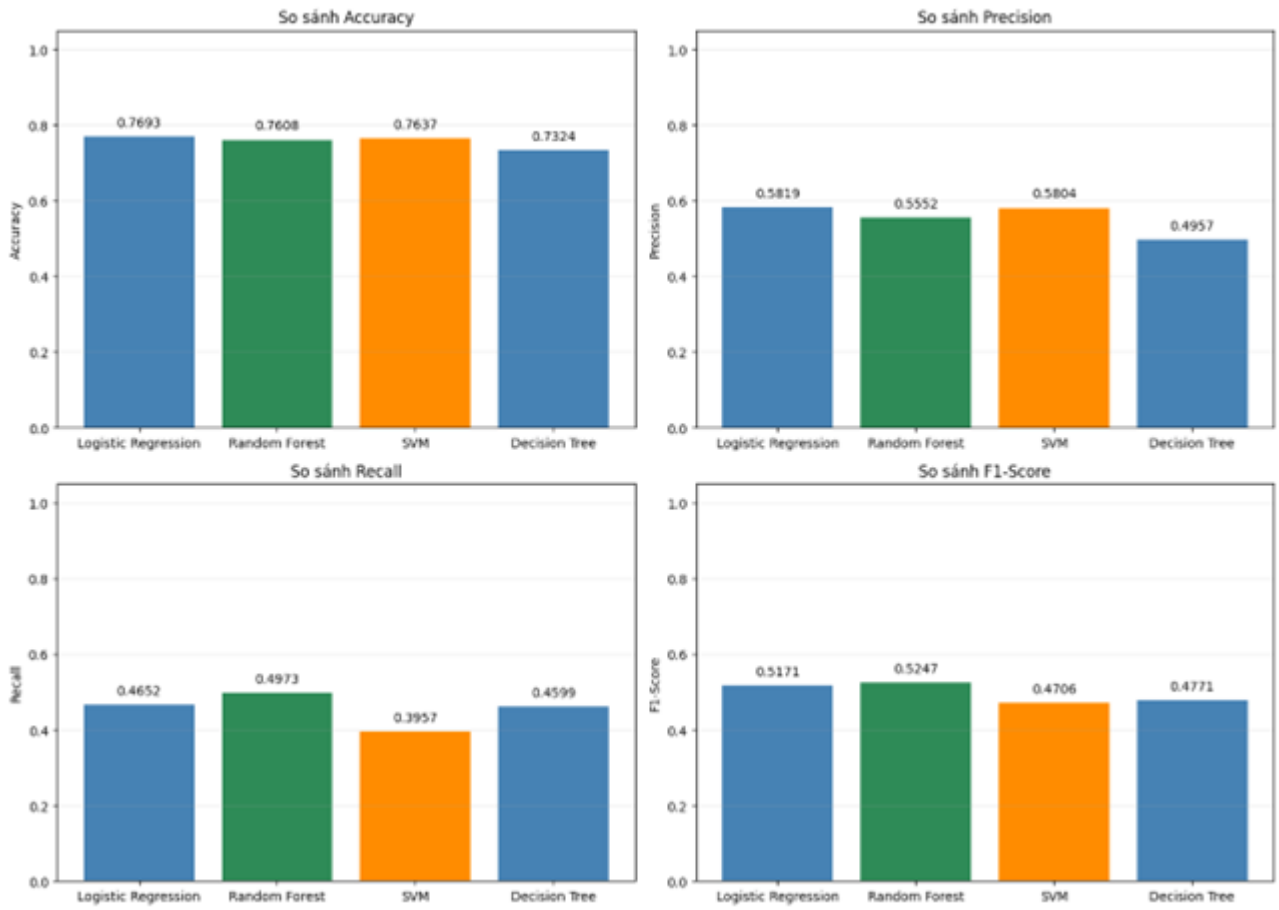
Logistic Regression	0.7693	0.5819	0.4652	0.5171
Random Forest	0.7608	0.5552	0.4973	0.5247
SVM	0.7637	0.5804	0.3957	0.4706
Decision Tree	0.7324	0.4957	0.4599	0.4771

Bảng 2.5.1. So sánh hiệu năng bốn mô hình phân loại churn

- Logistic Regression đạt F1-score cao nhất (0.517), cho thấy mô hình ổn định và cân bằng tốt giữa Precision – Recall. LR là mô hình tuyến tính, dễ diễn giải và phù hợp với dữ liệu không quá phức tạp.
- Random Forest có Recall cao nhất (0.497), nghĩa là nhận diện churn tốt hơn LR. Dù accuracy không cao bằng, RF phù hợp hơn về mặt nghiệp vụ vì giảm đáng kể số khách churn bị bỏ sót.
- SVM có Recall thấp nhất (0.396), thể hiện sự lệch mạnh về lớp không churn và bỏ sót nhiều churn thật. Hiệu suất này không đáp ứng yêu cầu bài toán.
- Decision Tree cho hiệu năng thấp nhất, dễ overfit và thiếu ổn định; chỉ phù hợp tham khảo, không dùng làm mô hình chính.

Tổng hợp lại, Random Forest là lựa chọn phù hợp nhất khi mục tiêu là giảm FN. Logistic Regression đứng thứ hai nhờ F1-score cao, trong khi SVM và Decision Tree không đáp ứng mục tiêu nghiệp vụ.

Biểu đồ dưới đây thể hiện so sánh trực quan Accuracy, Precision, Recall và F1 của bốn mô hình Logistic Regression, Random Forest, SVM và Decision Tree.



Hình 2.5.1. So sánh Accuracy, Precision, Recall và F1-score của bốn mô hình phân loại churn

Xác định mô hình tốt nhất theo F1-score

Nhóm sử dụng F1-score làm tiêu chí chính vì chỉ số này cân bằng giữa Precision và Recall, phù hợp với dữ liệu mất cân bằng và mục tiêu nhận diện đúng khách hàng có nguy cơ churn. Kết quả cho thấy Random Forest đạt F1-score cao nhất (0.5247), vượt Logistic Regression (0.5171), Decision Tree (0.4771) và SVM (0.4706), chứng tỏ mô hình này cân bằng tốt giữa hạn chế dự đoán sai và khả năng phát hiện churn. Vì vậy, xét theo F1-score, Random Forest là mô hình phù hợp nhất để chọn làm mô hình chính.

Đánh giá mô hình Random Forest bằng Cross-Validation (k = 5)

Random Forest được đánh giá lại bằng **Cross-Validation k = 5** để kiểm tra mức độ ổn định và khả năng khái quát hóa. Phương pháp này chia dữ liệu thành 5 phần,

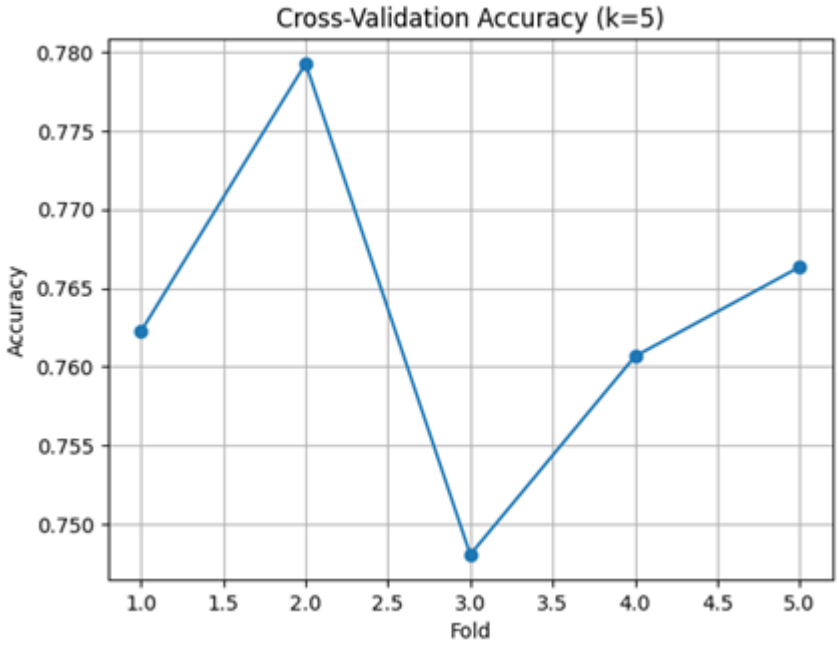
luân phiên train trên 4 phần và test trên 1 phần, giúp giảm ảnh hưởng của việc chia dữ liệu ngẫu nhiên.

- Mô hình sử dụng: RandomForestClassifier(n_estimators=100, random_state=42)
- Chỉ số đánh giá: Accuracy
- Số fold: k = 5

Kết quả Accuracy của mô hình Random Forest ở từng fold được ghi nhận như sau:

- Fold 1: 0.7622
- Fold 2: 0.7793
- Fold 3: 0.7480
- Fold 4: 0.7607
- Fold 5: 0.7663

Giá trị Mean Accuracy đạt 0.7633, trong khi Variance chỉ 0.00010, thể hiện độ biến thiên rất nhỏ giữa các lần đánh giá.



Hình 2.6.1. Kết quả Accuracy của Random Forest qua 5 folds (Cross-Validation k=5)

Cross-Validation cho thấy độ sai lệch giữa các fold rất nhỏ, với Accuracy chỉ dao động từ 0.7480–0.7793, chứng tỏ Random Forest ổn định và không phụ thuộc vào cách

chia dữ liệu. Mean Accuracy ~0.7633 cũng khớp với kết quả trên tập test, cho thấy hiệu năng đáng tin cậy.

Variance ở mức 0.00010, cho thấy mô hình có độ biến thiên thấp và không có dấu hiệu overfitting. Nhìn chung, Random Forest có khả năng khái quát hóa tốt và phù hợp làm mô hình chính cho bài toán churn.

Kết luận kỹ thuật

Kết quả thực nghiệm cho thấy Logistic Regression ổn định với F1-score cao nhất, nhưng Random Forest nhận diện churn tốt hơn nhờ Recall cao, giảm số trường hợp bị bỏ sót. SVM chịu ảnh hưởng mạnh từ mất cân bằng dữ liệu nên Recall thấp và lệch về lớp không churn.

Vì trong bài toán churn, Recall quan trọng hơn Accuracy để tránh thất thoát khách hàng, Random Forest phù hợp hơn về mặt nghiệp vụ, nhất là khi cần phát hiện sớm khách có nguy cơ rời bỏ dịch vụ.

Practice 3 (House Price Prediction)

Bài tập 1 – Dự đoán giá nhà (Bengaluru House Price Prediction)

Chuẩn bị dữ liệu

Giới thiệu đề bài

Trong thị trường bất động sản, việc định giá chính xác một căn nhà là bài toán phức tạp do phụ thuộc vào nhiều yếu tố đa dạng như vị trí địa lý, diện tích, tiện ích xung quanh và quy mô thiết kế. Bài thực hành này tập trung giải quyết vấn đề trên thông qua việc xây dựng mô hình học máy (Machine Learning) để dự đoán giá nhà tại thành phố Bengaluru (Ấn Độ).

Về mặt kỹ thuật, đây là một bài toán **Hồi quy (Regression)** điển hình. Khác với bài toán phân loại (Classification) ở Chương 3 (nơi biến mục tiêu là nhãn rời rạc 0/1), bài toán này yêu cầu mô hình dự đoán một giá trị liên tục.

- **Biến mục tiêu (Target Variable):** price (Giá nhà) - là đại lượng liên tục cần dự báo.
- **Biến đầu vào (Features):** Bao gồm các thông tin thuộc tính của căn nhà như location (vị trí), total_sqft (tổng diện tích), bath (số phòng tắm), và size (số phòng ngủ/quy mô).

Mục tiêu cuối cùng là xây dựng một mô hình có khả năng tổng quát hóa tốt, giảm thiểu sai số dự đoán (RMSE/MAE) trên tập dữ liệu kiểm thử.

Giới thiệu bộ dữ liệu

Dữ liệu được sử dụng trong bài thực hành là bộ **Bengaluru House Data**, được thu thập từ nền tảng Kaggle. Bộ dữ liệu này cung cấp thông tin chi tiết về thị trường bất động sản tại Bengaluru với các đặc điểm sau:

- **Nguồn dữ liệu:** File Bengaluru_House_Data.csv.
- **Kích thước dữ liệu:** Bộ dữ liệu bao gồm 13,320 quan sát (dòng) và 9 thuộc tính (cột).

Mô tả chi tiết các thuộc tính:

Để hiểu rõ ý nghĩa của từng trường dữ liệu trước khi đưa vào mô hình, bảng dưới đây mô tả chi tiết các thuộc tính có trong dataset:

STT	Tên thuộc tính	Ý nghĩa	Kiểu dữ liệu gốc	Vai trò
1	area_type	Phân loại diện tích (VD: Super built-up Area, Plot Area)	Categorical	Feature
2	availability	Tình trạng sẵn có (VD: Ready To Move, 19-Dec)	Categorical	Feature
3	location	Vị trí/Khu vực của ngôi nhà tại Bengaluru	Categorical	Feature Quan trọng
4	size	Quy mô nhà (số phòng ngủ, VD: 2 BHK, 4 Bedroom)	Categorical	Feature Quan trọng
5	society	Tên dự án hoặc khu dân cư	Categorical	Feature
6	total_sqft	Tổng diện tích sàn (đơn vị: feet vuông)	Object/Mixed	Feature Quan trọng
7	bath	Số lượng phòng tắm	Float	Feature Quan trọng
8	balcony	Số lượng ban công	Float	Feature
9	price	Giá nhà (đơn vị: Lakhs Rupee)	Float	Target (Biến mục tiêu)

Hình 3.1.1. Mô tả các thuộc tính trong bộ dữ liệu Bengaluru House Price

Dữ liệu mẫu: Dưới đây là 5 dòng đầu tiên của dữ liệu thô để có cái nhìn tổng quan về định dạng dữ liệu trước khi xử lý:

	area_type	availability	location	size	society	total_sqft	bath	balcony	price
0	Super built-up Area	19-Dec	Electronic City Phase II	2 BHK	Coomee	1056	2.0	1.0	39.07
1	Plot Area	Ready To Move	Chikka Tirupathi	4 Bedroom	Theanmp	2600	5.0	3.0	120.00
2	Built-up Area	Ready To Move	Uttarahalli	3 BHK	NaN	1440	2.0	3.0	62.00
3	Super built-up Area	Ready To Move	Lingadheeranahalli	3 BHK	Soiewre	1521	3.0	1.0	95.00
4	Super built-up Area	Ready To Move	Kothanur	2 BHK	NaN	1200	2.0	1.0	51.00
...	...	...	...	...	...	...	...	...	...
13315	Built-up Area	Ready To Move	Whitefield	5 Bedroom	ArsiaEx	3453	4.0	0.0	231.00
13316	Super built-up Area	Ready To Move	Richards Town	4 BHK	NaN	3600	5.0	NaN	400.00
13317	Built-up Area	Ready To Move	Raja Rajeshwari Nagar	2 BHK	Mahla T	1141	2.0	1.0	60.00
13318	Super built-up Area	18-Jun	Padmanabhanagar	4 BHK	SollyCl	4689	4.0	1.0	488.00
13319	Super built-up Area	Ready To Move	Doddathoguru	1 BHK	NaN	550	1.0	1.0	17.00

13320 rows × 9 columns

Hình 3.1.2. 5 dòng dữ liệu đầu tiên (Head)

Công cụ và Thư viện sử dụng

Sử dụng các thư viện phổ biến:

- Pandas: đọc, làm sạch, xử lý dữ liệu, tạo DataFrame sau chuẩn hóa/mã hóa.
- NumPy: thao tác mảng số học.
- Matplotlib, Seaborn: trực quan hóa.
- Scikit-learn: tiền xử lý và xây dựng mô hình học máy.

Làm sạch và Khám phá dữ liệu (Data Cleaning & Exploration)

Lược bỏ các thuộc tính không cần thiết

Bước đầu tiên trong quy trình làm sạch là loại bỏ các thông tin ít có giá trị dự báo hoặc nằm ngoài phạm vi bài toán. Dựa trên nhận định ban đầu về thị trường bất động sản và cấu trúc dữ liệu, nhóm thực hiện đã quyết định loại bỏ 4 cột:

- area_type (Loại khu vực)
- society (Tên khu dân cư - có quá nhiều giá trị riêng biệt gây nhiễu)

- balcony (Số ban công - ít ảnh hưởng trọng yếu đến giá so với diện tích)
- availability (Tình trạng sẵn có)

Việc loại bỏ này giúp giảm chiều dữ liệu, đơn giản hóa mô hình mà không làm mất đi các thông tin cốt lõi về giá trị căn nhà.

Xử lý dữ liệu thiếu (Missing Values)

Dữ liệu sau khi lọc cột được kiểm tra tính toàn vẹn bằng hàm isnull().sum(). Kết quả thống kê cho thấy mức độ thiếu hụt dữ liệu là rất thấp so với tổng kích thước bộ dữ liệu (13,320 mẫu):

- location: thiếu 1 giá trị.
- size: thiếu 16 giá trị.
- bath: thiếu 73 giá trị.
- total_sqft và price: đầy đủ dữ liệu.

Với tỷ lệ thiếu chưa đến 1%, nhóm quyết định áp dụng phương pháp **loại bỏ dòng (dropping rows)** đối với các mẫu chứa giá trị null. Phương pháp này đảm bảo tính trung thực của dữ liệu huấn luyện mà không làm giảm đáng kể kích thước mẫu.

Kỹ thuật Feature Engineering (Tạo đặc trưng mới)

Để nâng cao hiệu quả của mô hình hồi quy, các thuộc tính dạng chuỗi hoặc hỗn hợp cần được chuẩn hóa và chuyển đổi.

- Xử lý thuộc tính size (Quy mô phòng ngủ): Dữ liệu gốc trong cột size không đồng nhất (ví dụ: "2 BHK", "4 Bedroom"). Một đặc trưng số học mới là BHK (Bedroom Hall Kitchen) được tạo ra bằng cách tách lấy phần số đứng đầu chuỗi ký tự.
  - Ví dụ minh họa: "2 BHK" → 2; "4 Bedroom" → 4.
  - Khám phá: Sau khi chuyển đổi, dữ liệu cho thấy sự phân hóa lớn về số lượng phòng ngủ, cá biệt có căn nhà lên tới 43 phòng ngủ.
- Chuẩn hóa thuộc tính total_sqft (Tổng diện tích): Cột total_sqft chứa dữ liệu hỗn hợp, bao gồm cả số thực và các khoảng giá trị (range).
  - **Vấn đề:** Các giá trị như '1133 - 1384' không thể đưa trực tiếp vào tính toán.

- **Giải pháp:** Xây dựng hàm xử lý convert_sqft_tonum:
    - Nếu giá trị là một khoảng (có dấu '-'), tính trung bình cộng của hai đầu mút. Ví dụ: '1133 - 1384' → 1258.5
    - Nếu giá trị là số đơn thuần, chuyển sang kiểu float.
    - Các trường hợp đơn vị lạ (như 'Sq. Meter') được chuyển thành None và loại bỏ để tránh nhiễu.

- Tạo đặc trưng dẫn xuất price_per_sqft: Để hỗ trợ việc phát hiện các điểm dữ liệu bất thường (outliers) ở bước sau, một đặc trưng mới là "Giá trên mỗi foot vuông" được tính toán. Công thức:

$$\text{price\_per\_sqft} = \frac{\text{price} \times 1,000,000}{\text{total\_sqft}}$$

Hình 3.1.3. Công thức tính giá trên mỗi đơn vị diện tích

Xử lý ngoại lai (Outlier Removal)

Xử lý dựa trên tiêu chuẩn xây dựng (Diện tích tối thiểu mỗi phòng)

Trong thực tế xây dựng, một căn phòng ngủ tiêu chuẩn không thể có diện tích quá nhỏ. Nhóm đã thiết lập một ngưỡng logic dựa trên kiến thức miền (domain knowledge): trung bình mỗi phòng ngủ phải có diện tích tối thiểu là **300 sqft**.

Quy tắc loại bỏ:

$$\frac{\text{total\_sqft}}{\text{BHK}} < 300$$

- **Phân tích dữ liệu:** Khi kiểm tra dữ liệu thực tế, mã nguồn đã phát hiện những điểm dữ liệu vô lý, ví dụ: một căn nhà có tổng diện tích 1020 sqft nhưng lại được ghi nhận có tới 6 phòng ngủ. Điều này tương đương mỗi phòng chỉ khoảng 170 sqft, rất khó xảy ra trong thực tế và có khả năng cao là lỗi dữ liệu.
- **Hành động:** Loại bỏ toàn bộ các dòng dữ liệu vi phạm quy tắc này để đảm bảo tính thực tế của dữ liệu đầu vào.

Xử lý dựa trên phân phối giá (Price per Square Foot)

Tại mỗi địa điểm (location), giá nhà trên mỗi foot vuông (price_per_sqft) thường tuân theo một phân phối chuẩn xung quanh giá trị trung bình. Những mức giá quá thấp hoặc quá cao so với mặt bằng chung khu vực đó thường là ngoại lai.

- Phương pháp thống kê: Nhóm sử dụng kỹ thuật lọc theo Trung bình (Mean - m) và Độ lệch chuẩn (Standard Deviation - st).
  - Với mỗi location, tính m và st của price_per_sqft.
  - Chỉ giữ lại các quan sát nằm trong khoảng tin cậy:

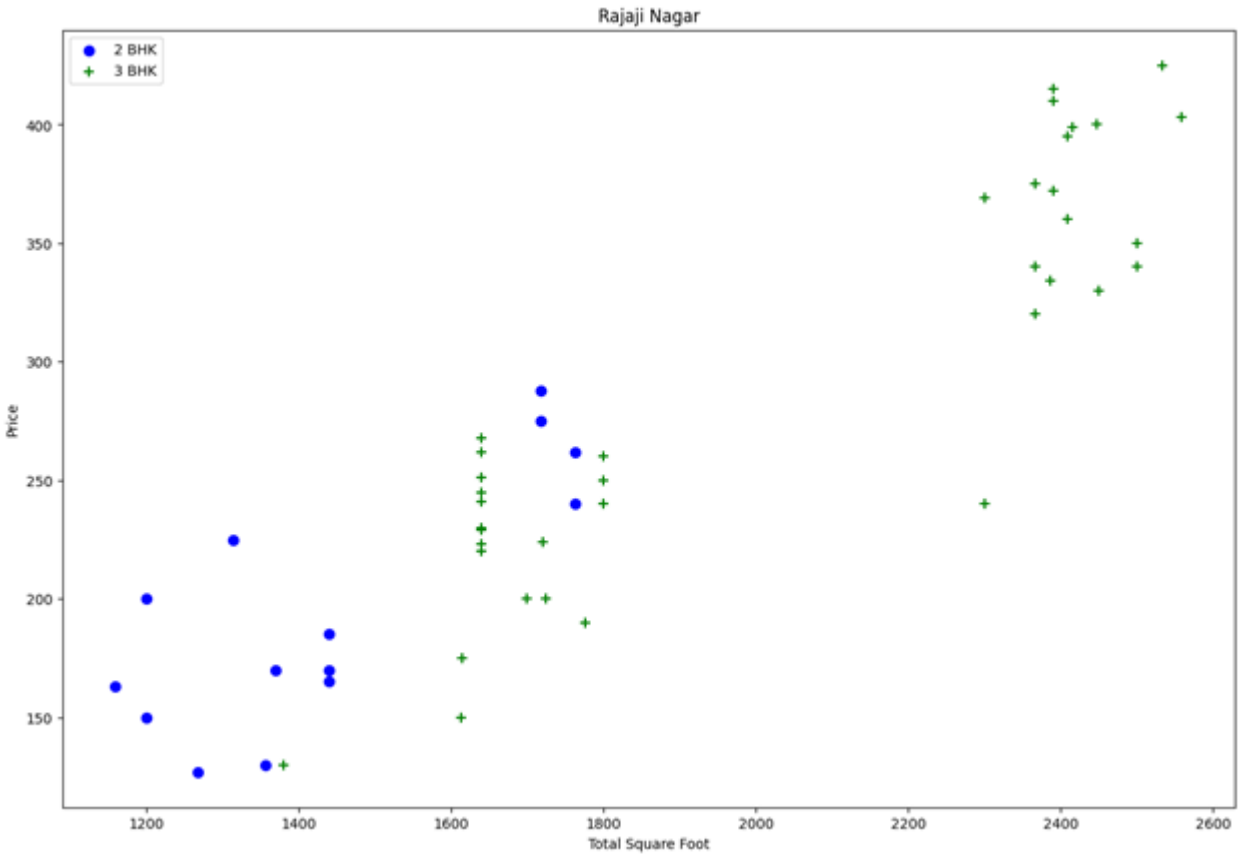
$$\mathbf{(m-st) \leq price\_per\_sqft \leq (m+st)}$$

- **Hiệu quả:** Bước này giúp loại bỏ các điểm dữ liệu cực đoan (extreme cases) mà mô hình tổng quát khó có thể dự đoán đúng. Kích thước dữ liệu sau bước này giảm từ khoảng 12,500 xuống còn **10,241 mẫu**.

Xử lý bất thường về giá giữa các loại căn hộ (BHK Price Logic)

Một hiện tượng phi logic được phát hiện thông qua trực quan hóa dữ liệu là: tại cùng một vị trí và điều kiện tương đương, giá của căn hộ nhỏ (ví dụ: 2 BHK) đôi khi lại cao hơn giá của căn hộ lớn (ví dụ: 3 BHK).

Trực quan hóa sự bất thường (Scatter Plot): Biểu đồ dưới đây minh họa dữ liệu tại khu vực "Rajaji Nagar". Các điểm màu xanh (2 BHK) và màu xanh lá (3 BHK) có sự chồng lấn, thậm chí nhiều điểm 2 BHK nằm cao hơn 3 BHK trên trục giá trị.



Hình 3.1.4. Biểu đồ phân tán giá nhà theo diện tích tại Rajaji Nagar trước khi xử lý ngoại lai. Trục hoành là Tổng diện tích, trục tung là Giá nhà

Giải thuật xử lý (remove_bhk_outliers): Để giải quyết vấn đề này, nhóm xây dựng một từ điển thống kê cho mỗi địa điểm.

- Quy tắc: Với cùng một vị trí, chúng ta loại bỏ những căn nhà có N phòng ngủ nếu price_per_sqft của nó **thấp hơn** mức trung bình (mean) của các căn nhà có N-1 phòng ngủ.
- Ý nghĩa: Đảm bảo tính nhất quán của thị trường – nhà rộng hơn, nhiều phòng hơn thì đơn giá trung bình thường phải cao hơn hoặc ít nhất là tương đương.

Sau bước này, dữ liệu trở nên tách biệt rõ ràng hơn giữa các nhóm phòng ngủ, giảm thiểu sự nhiễu loạn giúp mô hình hồi quy hội tụ tốt hơn. Số lượng mẫu giảm xuống còn **7,329**.

Xử lý ngoại lai về tiện ích (Số phòng tắm)

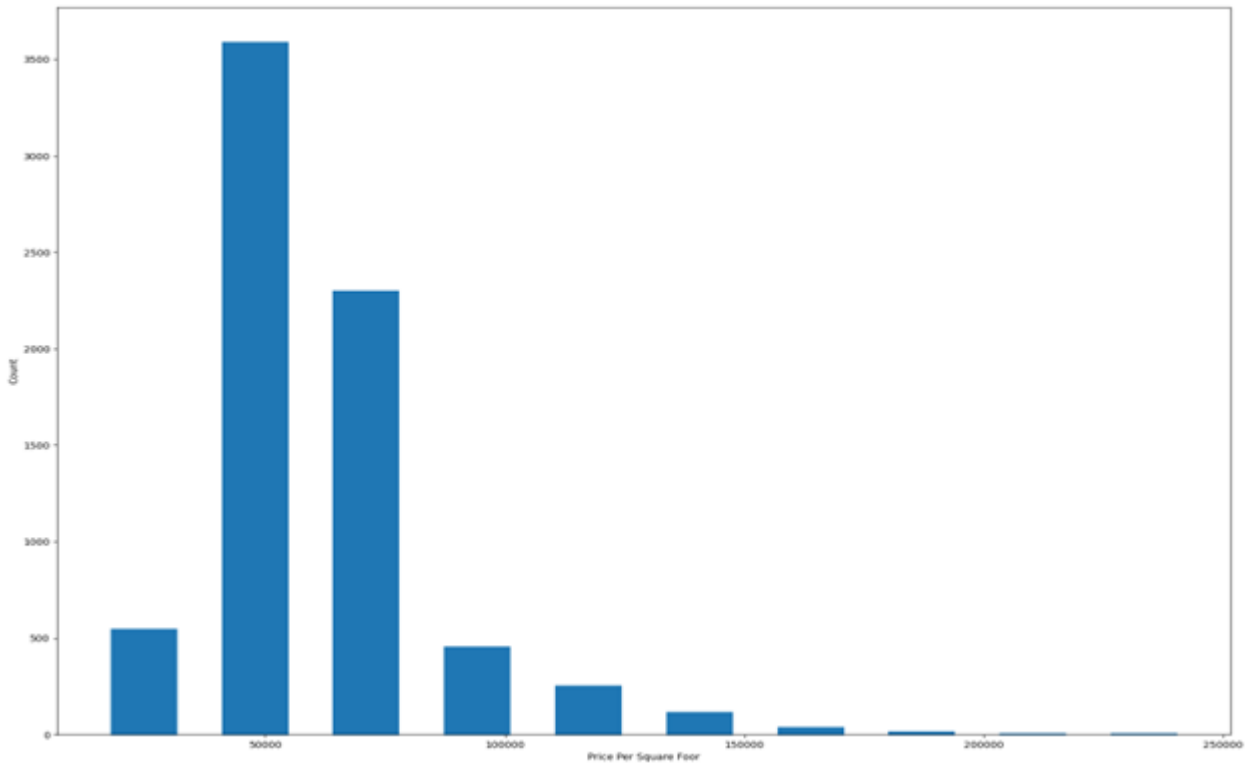
Cuối cùng, số lượng phòng tắm (bath) được kiểm tra dựa trên số lượng phòng ngủ. Thông thường, số phòng tắm không nên vượt quá số phòng ngủ quá nhiều.

- **Quy tắc:** Loại bỏ nếu bath > BHK + 2.
- **Ví dụ:** Dữ liệu cho thấy có trường hợp nhà 4 phòng ngủ nhưng có tới 7 phòng tắm, hoặc 10 phòng ngủ có 12 phòng tắm. Những trường hợp này được coi là thiết kế đặc thù hoặc lỗi nhập liệu và bị loại bỏ để tránh gây nhiễu.

Tổng kết và Đánh giá phân phối dữ liệu

Sau quy trình làm sạch gồm 4 bước trên, bộ dữ liệu cuối cùng còn lại **7,251 quan sát** (giảm khoảng 45% so với ban đầu). Tuy số lượng giảm, nhưng "độ sạch" và tính quy luật của dữ liệu tăng lên đáng kể.

Biểu đồ Histogram dưới đây biểu diễn phân phối của price_per_sqft sau khi xử lý:



Hình 3.1.5. Phân phối chuẩn (Normal Distribution) của Price Per Square Foot sau khi xử lý ngoại lai

Nhận xét: Dữ liệu sau khi lọc ngoại lai đã gọn gàng hơn và loại bỏ được các giá trị cực đoan (nhiễu). Mặc dù biểu đồ vẫn cho thấy xu hướng **lệch phải (Right-Skewed)** – điều thường thấy ở dữ liệu giá cả bất động sản – nhưng dải giá trị đã thu hẹp lại trong khoảng hợp lý (từ 0 đến 250,000). So với dữ liệu gốc hỗn loạn, phân phối này đã ổn định hơn nhiều, tạo tiền đề tốt để các mô hình hồi quy học được quy luật chung của thị trường."

Tiền xử lý dữ liệu (Data Preprocessing)

Sau khi đã làm sạch và loại bỏ các điểm dữ liệu nhiễu, bước tiếp theo là chuyển đổi dữ liệu về định dạng phù hợp nhất để đưa vào các thuật toán học máy. Trọng tâm của giai đoạn này là xử lý dữ liệu phân loại (categorical data) và chia tập dữ liệu để huấn luyện.

Giảm chiều dữ liệu (Dimensionality Reduction)

Một thách thức lớn trong bộ dữ liệu này là thuộc tính location (địa điểm). Ban đầu, có tới **1,304** địa điểm khác nhau. Nếu áp dụng kỹ thuật One-Hot Encoding trực tiếp cho tất cả các giá trị này, số lượng đặc trưng (số cột) của dữ liệu sẽ tăng lên quá lớn (gọi là hiện tượng bùng nổ chiều dữ liệu - curse of dimensionality), gây khó khăn cho việc huấn luyện và dễ dẫn đến overfitting.

- **Giải pháp:** Nhóm áp dụng kỹ thuật gom nhóm dựa trên tần suất xuất hiện.
  - Thống kê số lượng bản ghi cho mỗi địa điểm.
  - Các địa điểm có số lượng bản ghi nhỏ hơn hoặc bằng 10 (<= 10) sẽ được gán nhãn chung là **"other"**.
- **Kết quả:** Số lượng địa điểm duy nhất giảm từ 1,304 xuống còn **242**. Điều này giúp giảm đáng kể số chiều của vector đặc trưng sau này mà vẫn giữ lại thông tin của các khu vực quan trọng nhất.

Mã hóa biến phân loại (One-Hot Encoding)

Các thuật toán học máy không thể làm việc trực tiếp với dữ liệu dạng văn bản. Do đó, biến location cần được chuyển đổi sang dạng số. Nhóm sử dụng kỹ thuật **One-Hot Encoding** thông qua hàm pd.get_dummies. Quy trình:

- Tạo các cột nhị phân (0/1) tương ứng với từng địa điểm trong danh sách 242 địa điểm đã rút gọn.
- Loại bỏ cột other để tránh bẫy đa cộng tuyến (dummy variable trap) – tình trạng các biến độc lập có mối quan hệ tuyến tính hoàn hảo với nhau.
- Ghép các cột mới này vào DataFrame chính và loại bỏ cột location gốc dạng chữ.

Hình ảnh minh họa dữ liệu sau khi mã hóa (Bảng Dummy Variables):

	1st Block Jayanagar	1st Phase JP Nagar	2nd Phase Judicial Layout	2nd Stage Nagarbhavi	5th Block Hbr Layout	5th Phase JP Nagar	6th Phase JP Nagar	7th Phase JP Nagar	8th Phase JP Nagar	9th Phase JP Nagar	...	Vishveshwarya Layout	Vishwapriya Layout	Vittasandra	Whitefield	Yelachenahall
0	True	False	False	False	False	False	False	False	False	False	...	False	False	False	False	Fals
1	True	False	False	False	False	False	False	False	False	False	...	False	False	False	False	Fals
2	True	False	False	False	False	False	False	False	False	False	...	False	False	False	False	Fals
3	True	False	False	False	False	False	False	False	False	False	...	False	False	False	False	Fals
4	True	False	False	False	False	False	False	False	False	False	...	False	False	False	False	Fals
5	True	False	False	False	False	False	False	False	False	False	...	False	False	False	False	Fals
6	True	False	False	False	False	False	False	False	False	False	...	False	False	False	False	Fals
8	False	True	False	False	False	False	False	False	False	False	...	False	False	False	False	Fals
9	False	True	False	False	False	False	False	False	False	False	...	False	False	False	False	Fals
10	False	True	False	False	False	False	False	False	False	False	...	False	False	False	False	Fals

10 rows × 242 columns

Hình 3.1.6. Một phần của dữ liệu sau khi

Kết quả cuối cùng là một bộ dữ liệu (data8 trong code) hoàn toàn ở dạng số, bao gồm các cột đặc trưng: total_sqft, bath, BHK và 241 cột đại diện cho vị trí (đã loại bỏ cột other).

Chia tập dữ liệu (Train/Test Split)

Để đánh giá khách quan hiệu năng của mô hình, dữ liệu được chia thành hai tập độc lập: tập huấn luyện (Training set) và tập kiểm thử (Test set).

- **Biến độc lập (X):** Tất cả các cột trừ cột price.
- **Biến phụ thuộc (y):** Cột price.
- **Tỷ lệ chia:** 80% cho huấn luyện và 20% cho kiểm thử.
- **Kích thước tập dữ liệu:**
  - **Tập huấn luyện:** 5,800 mẫu.
  - **Tập kiểm thử:** 1,451 mẫu.

Việc chia dữ liệu này đảm bảo mô hình được học trên một lượng dữ liệu đủ lớn và được đánh giá trên những dữ liệu chưa từng thấy (unseen data) để kiểm tra khả năng tổng quát hóa.

Huấn luyện và Đánh giá mô hình

Sau khi hoàn tất quá trình tiền xử lý, nhóm đã tiến hành huấn luyện hai mô hình chính là **Linear Regression** (Hồi quy tuyến tính) và **Random Forest Regressor** (Rừng ngẫu nhiên) trên tập huấn luyện (80% dữ liệu). Kết quả dự báo trên tập kiểm thử (20% dữ liệu) được đánh giá thông qua 5 chỉ số kỹ thuật: MAE, RMSE, R^2, MAPE và điểm kiểm chứng chéo (CV R^2).

Bảng tổng hợp kết quả thực nghiệm

Dưới đây là bảng so sánh hiệu năng chi tiết giữa hai mô hình dựa trên dữ liệu chạy thực tế:

Metric (Chỉ số)	**Linear Regression**	**Random Forest**	**Nhận định**
MAE (Sai số tuyệt đối trung bình)	17.99	18.89	Linear Regression có sai số trung bình thấp hơn (~0.9 đơn vị).
RMSE (Căn bậc hai sai số bình phương trung bình)	34.93	41.88	Linear Regression vượt trội hơn hẳn, cho thấy mô hình này ít mắc các lỗi dự báo lớn (large errors) hơn so với Random Forest.
R^2 (Hệ số xác định)	0.8746	0.8198	Linear Regression giải thích được khoảng 87.46% sự biến thiên của giá nhà, cao hơn mức 81.98% của Random Forest.

MAPE (Sai số phần trăm trung bình tuyệt đối)	20.50%	20.17%	Random Forest có sai số tương đối thấp hơn một chút (chênh lệch ~0.33%), nhưng không đáng kể so với sự chênh lệch của RMSE.
CV R^2 (k=5) (Điểm kiểm chứng chéo trung bình)	0.8215	0.7581	Linear Regression cho thấy độ ổn định cao hơn hẳn khi kiểm thử trên nhiều tập dữ liệu con khác nhau.

Bảng 3.1.1. So sánh hiệu năng giữa Linear Regression và Random Forest

Phân tích chi tiết

- Về độ chính xác tổng thể (R^2 Score): Mô hình Linear Regression đạt kết quả R^2 là 0.8746, nghĩa là mô hình này giải thích được khoảng 87.5% biến thiên của giá nhà dựa trên các đặc trưng đầu vào. Trong khi đó, Random Forest chỉ đạt 0.8198. Điều này cho thấy sau khi xử lý ngoại lai kỹ càng, mối quan hệ giữa các biến độc lập (diện tích, vị trí, số phòng) và biến mục tiêu (giá) có tính tuyến tính mạnh mẽ, giúp mô hình hồi quy tuyến tính hoạt động hiệu quả hơn các mô hình phi tuyến phức tạp.
- Về sai số dự báo (RMSE & MAE): Chỉ số RMSE của Linear Regression là 34.93, thấp hơn đáng kể so với 41.88 của Random Forest. Vì RMSE phạt nặng các sai số lớn (outliers trong dự báo), kết quả này chứng tỏ Linear Regression đưa ra các dự đoán ổn định hơn và ít khi đưa ra các mức giá "lệch pha" quá xa so với thực tế.
- Về độ ổn định (Cross-Validation): Khi thực hiện kiểm chứng chéo với k=5 (chia dữ liệu thành 5 phần khác nhau để kiểm tra), Linear Regression vẫn duy trì được phong độ với điểm trung bình 0.821, trong khi Random Forest giảm xuống còn 0.758. Sự sụt giảm này cho thấy Random Forest có dấu hiệu bị Overfitting (học tủ) trên tập huấn

luyện và khả năng tổng quát hóa trên dữ liệu mới kém hơn so với Linear Regression.

Kết luận lựa chọn mô hình

Dựa trên các số liệu thực nghiệm:

- **Linear Regression** chiến thắng ở 4/5 chỉ số quan trọng (MAE, RMSE, R^2, CV R^2).
- Mô hình này không chỉ chính xác hơn mà còn đơn giản, thời gian huấn luyện nhanh và dễ giải thích (interpretable).

→ **Quyết định:** Nhóm lựa chọn **Linear Regression** làm mô hình cuối cùng để triển khai cho bài toán dự đoán giá nhà tại Bengaluru.

Kết luận

Tổng kết quy trình thực hiện

Bài toán dự đoán giá nhà tại Bengaluru đã được giải quyết thông qua một quy trình khoa học dữ liệu toàn diện. Từ bộ dữ liệu thô ban đầu với nhiều nhiễu và định dạng không đồng nhất, nhóm đã thực hiện các bước xử lý quan trọng:

- **Làm sạch dữ liệu:** Loại bỏ các thuộc tính thừa và các dòng dữ liệu thiếu để đảm bảo chất lượng đầu vào.
- **Kỹ thuật đặc trưng (Feature Engineering):** Chuyển đổi thành công các dữ liệu phức tạp (như size, total_sqft) về dạng số học và tạo thêm biến phái sinh price_per_sqft hỗ trợ phân tích.
- **Xử lý ngoại lai (Outlier Removal):** Đây là bước đóng góp lớn nhất vào hiệu suất mô hình. Việc loại bỏ các điểm dữ liệu phi thực tế (như diện tích phòng quá nhỏ, giá quá cao so với mặt bằng chung, hay số phòng tắm bất hợp lý) đã giúp dữ liệu trở nên "sạch" và tuân theo phân phối chuẩn hơn.

Kết luận về mô hình

Dựa trên kết quả thực nghiệm so sánh giữa hai thuật toán Linear Regression và Random Forest, nhóm rút ra các kết luận sau:

- **Mô hình tối ưu: Linear Regression** được xác định là mô hình phù hợp nhất cho bài toán này. Với hệ số xác định **$R^2 \approx 87.46\%$** trên tập

kiểm thử, mô hình giải thích được phần lớn sự biến thiên của giá nhà dựa trên các đặc trưng đã chọn.

- **Độ ổn định:** Kết quả kiểm chứng chéo (Cross-Validation k=5) cho thấy Linear Regression duy trì độ chính xác ổn định ở mức **~82%**, cao hơn đáng kể so với Random Forest (~75%). Điều này chứng tỏ mô hình không bị hiện tượng học vẹt (overfitting) và có khả năng tổng quát hóa tốt trên các tập dữ liệu khác nhau.
- **Sai số dự báo:** Với chỉ số **RMSE thấp hơn** (34.93 so với 41.88), Linear Regression cho thấy mức độ tin cậy cao hơn trong việc đưa ra các mức giá dự báo sát với thực tế, hạn chế các sai số lớn có thể gây rủi ro trong định giá bất động sản.

Nhận định chung

Kết quả vượt trội của mô hình tuyến tính so với mô hình phi tuyến (Random Forest) trong trường hợp này cho thấy: sau khi đã xử lý kỹ lưỡng các điểm ngoại lai và chuẩn hóa dữ liệu, mối quan hệ giữa các yếu tố (như diện tích, vị trí, số phòng) và giá nhà tại Bengaluru mang tính **tuyến tính mạnh mẽ**. Việc áp dụng các mô hình quá phức tạp (như Random Forest) trong trường hợp này không những không tăng độ chính xác mà còn làm giảm hiệu quả tính toán và độ ổn định của dự báo.

Bài tập 2 – Phân cụm khách hàng (Customer Segmentation)

Chuẩn bị dữ liệu

Giới thiệu đề bài

Trong lĩnh vực bán lẻ và tiếp thị, việc hiểu rõ hành vi khách hàng là yếu tố then chốt để xây dựng các chiến lược kinh doanh hiệu quả. Bài tập này giải quyết bài toán Phân khúc khách hàng (Customer Segmentation).

Khác với các bài toán trước (Phân loại hay Hồi quy), đây là bài toán thuộc nhóm Học không giám sát (Unsupervised Learning). Dữ liệu đầu vào không có nhãn (label) trước. Nhiệm vụ của mô hình là tự động phát hiện các nhóm khách hàng có đặc điểm tương đồng nhau dựa trên dữ liệu về thu nhập và thói quen chi tiêu. Thuật toán được sử dụng là K-Means Clustering.

Giới thiệu bộ dữ liệu

Bộ dữ liệu được sử dụng là Mall Customers Dataset, chứa thông tin cơ bản về khách hàng thẻ thành viên của một trung tâm thương mại.

- **Nguồn dữ liệu:** File Mall_Customers.csv.
- **Kích thước:** Bộ dữ liệu bao gồm 200 quan sát và 5 thuộc tính.

Mô tả các thuộc tính dữ liệu bao gồm các thông tin nhân khẩu học và hành vi tiêu dùng:

STT	Tên thuộc tính	Ý nghĩa
1	CustomerID	Mã định danh khách hàng duy nhất.
2	Genre	Giới tính khách hàng (Male/Female).
3	Age	Độ tuổi của khách hàng.
4	Annual Income (k$)	Thu nhập hàng năm (đơn vị: nghìn đô la).
5	Spending Score (1-100)	Điểm chi tiêu (được gán bởi trung tâm thương mại dựa trên hành vi mua sắm).

Bảng 3.2.1. Mô tả các thuộc tính bộ dữ liệu Mall Customers

Dữ liệu mẫu dưới đây là 5 dòng đầu tiên của dữ liệu để hình dung cấu trúc:

	CustomerID	Genre	Age	Annual Income (k$)	Spending Score (1-100)
0	1	Male	19	15	39
1	2	Male	21	15	81
2	3	Female	20	16	6
3	4	Female	23	16	77
4	5	Female	31	17	40

Hình 3.2.1. 5 dòng dữ liệu đầu tiên của Mall Customers Dataset

Công cụ và thư viện sử dụng

Sử dụng các thư viện phổ biến:

- Pandas: đọc, làm sạch, xử lý dữ liệu, tạo DataFrame sau chuẩn hóa/mã hóa.
- NumPy: thao tác mảng số học.
- Matplotlib, Seaborn: trực quan hóa.
- Scikit-learn: tiền xử lý và xây dựng mô hình học máy.

Khám phá và tiền xử lý dữ liệu

Khám phá dữ liệu ban đầu (Data Exploration)

Bộ dữ liệu Mall_Customers.csv được nạp vào và kiểm tra tổng quan. Kết quả thống kê ban đầu cho thấy:

- **Kích thước dữ liệu:** 200 dòng (khách hàng) và 5 cột (thuộc tính).
- **Các thuộc tính:** CustomerID, Genre (Giới tính), Age (Tuổi), Annual Income (k$) (Thu nhập), Spending Score (1-100) (Điểm chi tiêu).
- **Kiểm tra giá trị thiếu:** Sử dụng hàm df.isnull().sum(), kết quả trả về là **0** cho tất cả các cột. Điều này xác nhận bộ dữ liệu hoàn toàn sạch, không cần thực hiện các bước xử lý giá trị null (Imputation).

Lựa chọn đặc trưng (Feature Selection)

Mục tiêu của bài toán là phân khúc khách hàng dựa trên hành vi tiêu dùng và mức thu nhập để phục vụ chiến lược marketing. Do đó, nhóm thực hiện đã quyết định chỉ giữ lại hai biến số quan trọng nhất để đưa vào mô hình:

- **Annual Income (k$):** Đại diện cho năng lực tài chính.
- **Spending Score (1-100):** Đại diện cho hành vi mua sắm thực tế tại trung tâm thương mại.

Các cột thông tin nhân khẩu học khác như Genre (Giới tính), Age (Tuổi) và mã định danh CustomerID được loại bỏ khỏi quá trình huấn luyện để tập trung phân tích mối tương quan giữa Thu nhập và Chi tiêu.

--- Dữ liệu đã chọn cho Phân cụm (X) ---

	Annual Income (k$)	Spending Score (1-100)
0	15	39
1	15	81
2	16	6
3	16	77
4	17	40

Hình 3.2.2. Dữ liệu sau khi rút gọn (5 dòng đầu)

Chuẩn hóa dữ liệu (Data Standardization)

Thuật toán K-Means hoạt động dựa trên việc tính toán khoảng cách (thường là khoảng cách Euclidean) giữa các điểm dữ liệu. Nếu các biến có đơn vị và độ lớn khác nhau (ví dụ: Thu nhập hàng nghìn đô la so với Điểm chi tiêu từ 1-100), biến có giá trị lớn hơn sẽ chi phối kết quả phân cụm, dẫn đến sai lệch.

- **Giải pháp:** Sử dụng kỹ thuật **StandardScaler** từ thư viện Scikit-learn.
- Cơ chế: Biến đổi dữ liệu sao cho mỗi đặc trưng có giá trị trung bình (mean) bằng 0 và độ lệch chuẩn (standard deviation) bằng 1.

$$z = \frac{x - \mu}{\sigma}$$

- **Kết quả:** Tạo ra tập dữ liệu mới X_scaled có cùng một thang đo chuẩn, giúp thuật toán K-Means hội tụ nhanh và chính xác hơn.

--- Dữ liệu đã được Chuẩn hóa (Standard Scaled) ---

	Annual Income (k$)	Spending Score (1-100)
0	-1.738999	-0.434801
1	-1.738999	1.195704
2	-1.700830	-1.715913
3	-1.700830	1.040418
4	-1.662660	-0.395980

Hình 3.2.3. Dữ liệu sau khi chuẩn hóa StandardScaler (5 dòng đầu)

Kết thúc giai đoạn này, dữ liệu đã sẵn sàng để đưa vào huấn luyện mô hình phân cụm.

Thuật toán K-Means Clustering

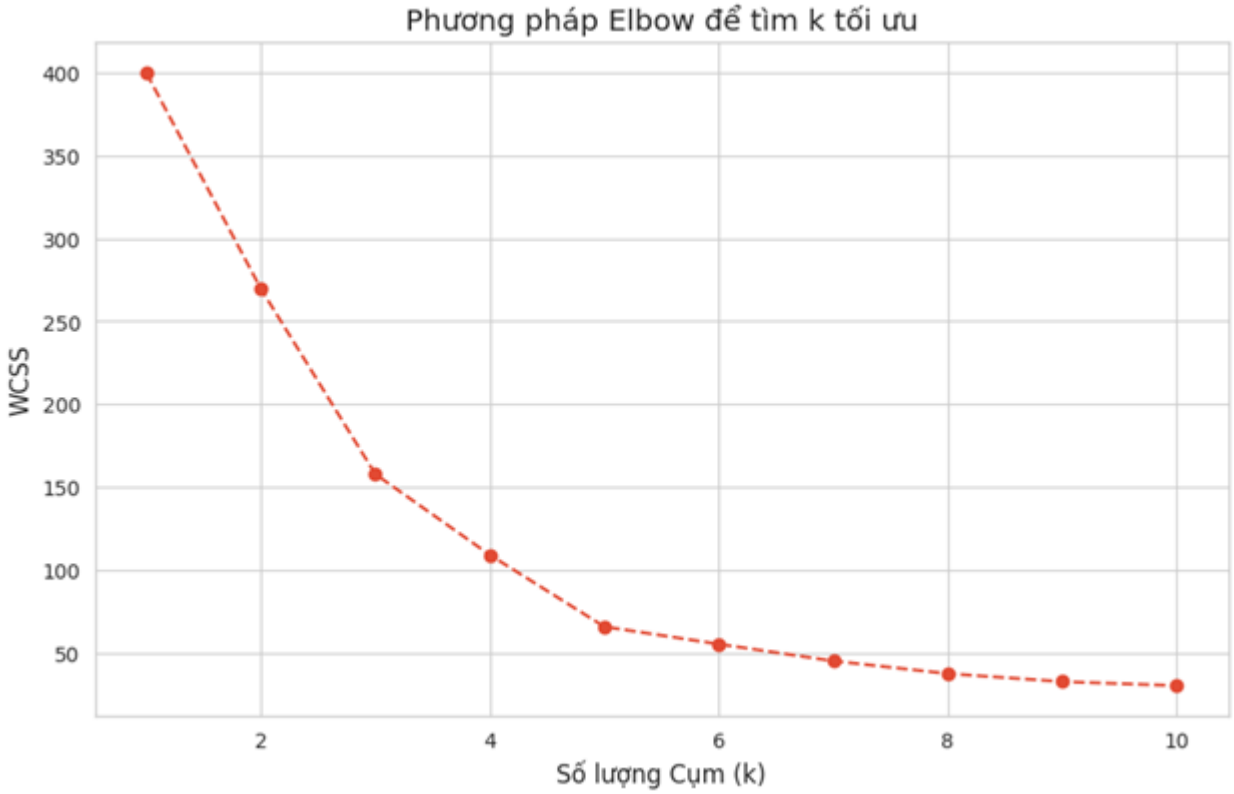
Đây là giai đoạn cốt lõi của bài toán phân khúc khách hàng. Nhóm sử dụng thuật toán K-Means, một phương pháp học không giám sát phổ biến để gom nhóm dữ liệu dựa trên khoảng cách Euclidean. Quy trình bao gồm hai bước chính: xác định số lượng cụm tối ưu (k) và thực hiện phân cụm chi tiết.

Xác định số lượng cụm tối ưu (Phương pháp Elbow)

Một trong những thách thức lớn nhất của K-Means là thuật toán không tự động biết được có bao nhiêu nhóm trong dữ liệu. Để giải quyết vấn đề này, nhóm sử dụng Phương pháp Khuỷu tay (Elbow Method).

- Nguyên lý thực hiện:
  - Chạy thuật toán K-Means nhiều lần với số lượng cụm k tăng dần từ 1 đến 10.
  - Tại mỗi giá trị k, tính toán chỉ số WCSS (Within-Cluster Sum of Squares) tổng bình phương khoảng cách từ mỗi điểm dữ liệu đến tâm cụm của nó. Giá trị này càng nhỏ nghĩa là các điểm trong cùng một cụm càng nằm gần nhau (cụm càng gọn).

- Phân tích biểu đồ: Biểu đồ dưới đây biểu diễn mối quan hệ giữa k và WCSS:



Hình 3.2.4. Biểu đồ phương pháp Elbow.

Trục hoành là số lượng cụm (k), trục tung là giá trị WCSS.

Nhận xét:

- Khi k tăng từ 1 đến 3, giá trị WCSS giảm rất mạnh, cho thấy việc chia nhỏ dữ liệu mang lại hiệu quả lớn.
- Tại vị trí k=5, đồ thị tạo thành một "khuỷu tay" rõ rệt. Từ điểm này trở đi, việc tăng thêm số cụm (k=6, 7...) chỉ làm giảm WCSS không đáng kể nhưng lại làm mô hình phức tạp hơn.

→ **Kết luận:** k=5 là giá trị tối ưu để cân bằng giữa độ chính xác và tính đơn giản.

Thực nghiệm Phân cụm với các giá trị K khác nhau

Mặc dù biểu đồ Elbow gợi ý k=5, nhóm vẫn tiến hành thử nghiệm với các giá trị k=3 và k=4 để quan sát sự thay đổi trong cấu trúc các nhóm khách hàng và đảm bảo không bỏ sót các góc nhìn khác.

Kịch bản 1: Phân cụm với k = 3

- **Mô tả:** Dữ liệu được chia thành 3 nhóm lớn.
- **Kết quả phân bố:**
  - Cụm 2: 123 khách hàng (Chiếm đa số ~61%).
  - Cụm 1: 39 khách hàng.
  - Cụm 0: 38 khách hàng.
- **Đánh giá:** Với k=3, thuật toán gộp chung những khách hàng có thu nhập trung bình và thấp vào một nhóm lớn (Cụm 2). Cách chia này quá tổng quát, không tách biệt được những khách hàng có hành vi "lạ" (ví dụ: thu nhập thấp nhưng chi tiêu cao) để làm Marketing hiệu quả.

Kịch bản 2: Phân cụm với k = 4

- **Mô tả:** Tăng số cụm lên 4 để kỳ vọng sự tách biệt rõ hơn.
- **Kết quả phân bố:**
  - Cụm 2: 100 khách hàng (Vẫn chiếm 50%).
  - Cụm 3: 39 khách hàng.
  - Cụm 1: 38 khách hàng.
  - Cụm 0: 23 khách hàng.

Đánh giá: Đã bắt đầu xuất hiện sự tách biệt ở các nhóm biên (thu nhập cao/thấp), nhưng nhóm khách hàng "phổ thông" vẫn bị gộp chung. Việc này chưa tối ưu hóa được chiến lược chăm sóc khách hàng.

Kịch bản 3: Phân cụm với k = 5 (Tối ưu)

- **Mô tả:** Đây là giá trị được chọn từ phương pháp Elbow.
- **Kết quả phân bố:**
  - Cụm 0: 81 khách hàng (Nhóm phổ thông).
  - Cụm 1: 39 khách hàng.
  - Cụm 3: 35 khách hàng.
  - Cụm 4: 23 khách hàng.
  - Cụm 2: 22 khách hàng.

Đánh giá:

  - Sự phân bố số lượng khách hàng trở nên đồng đều và hợp lý hơn.

- Nhóm "khổng lồ" ở trung tâm đã được tách gọn lại (Cụm 0 - 81 khách), nhường chỗ cho các nhóm đặc thù ở 4 góc của biểu đồ (ví dụ: Thu nhập cao - Chi cao, Thu nhập thấp - Chi cao).
- Đây là kết quả phân cụm tốt nhất để đưa vào phân tích nghiệp vụ chi tiết ở phần sau.

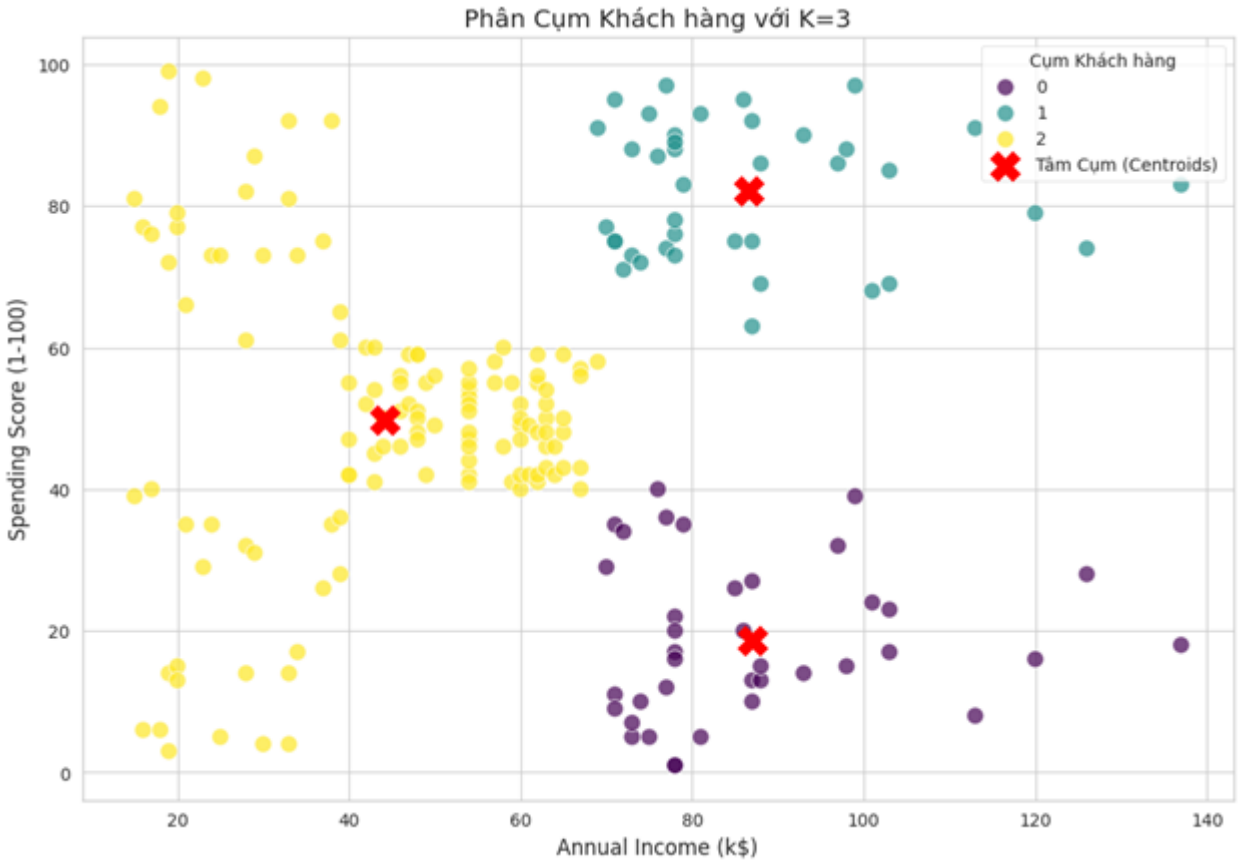
Phân tích và Trực quan hóa kết quả

Sau khi áp dụng thuật toán K-Means, bước tiếp theo là trực quan hóa dữ liệu để đánh giá mức độ phân tách của các cụm trên không gian hai chiều: Thu nhập hàng năm (Trục hoành) và Điểm chi tiêu (Trục tung).

Trực quan hóa và Đánh giá các kịch bản phân cụm

Nhóm đã tiến hành vẽ biểu đồ phân tán (Scatter Plot) cho ba kịch bản k=3, 4, 5 để so sánh cấu trúc phân nhóm.

Kịch bản 1: Phân cụm với k = 3



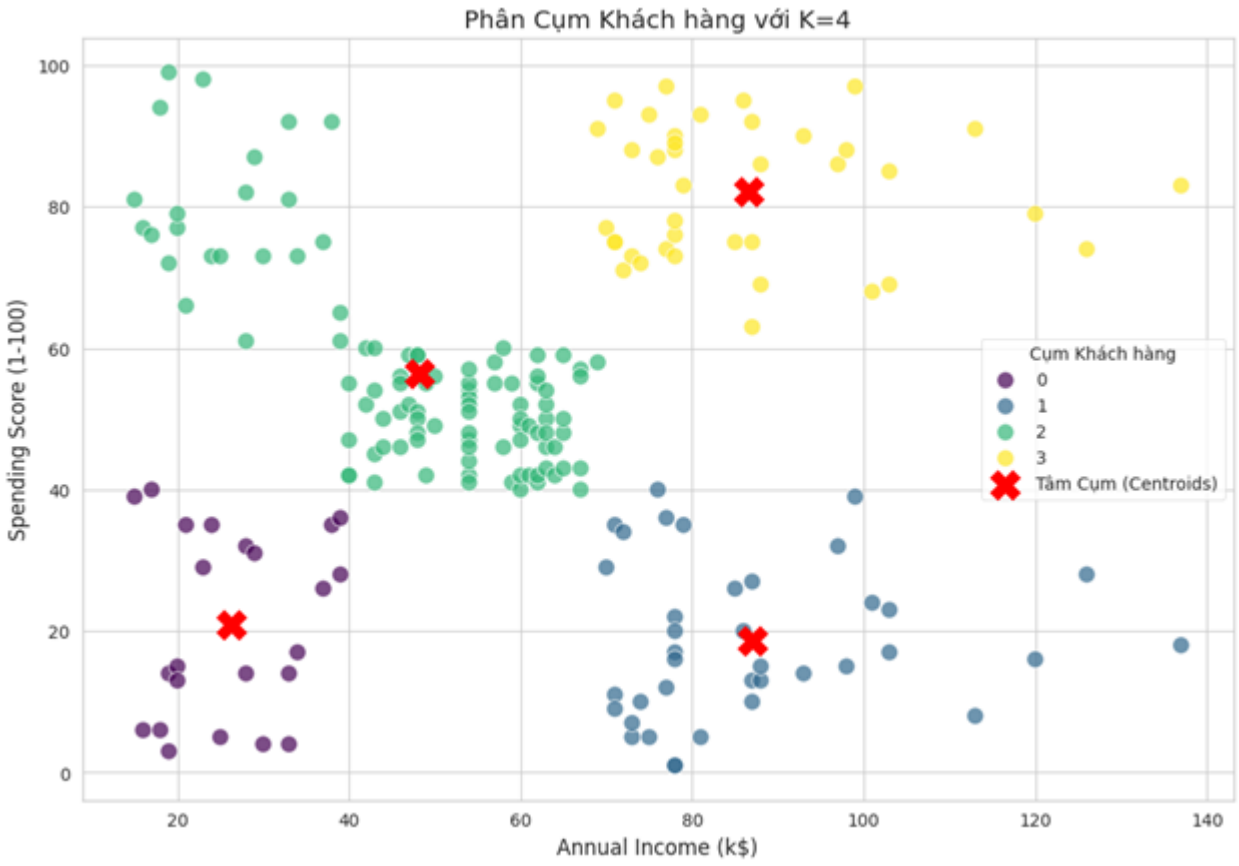
Hình 3.2.5. Biểu đồ phân cụm khách hàng với k=3

Phân tích: Khi chia thành 3 nhóm, biểu đồ cho thấy sự phân tách quá tổng quát.

- Nhóm khách hàng ở giữa (chiếm đa số) bị gộp chung một cách lộn xộn.
- Mô hình không phân biệt được những khách hàng có mức chi tiêu đối lập nhau trong cùng một mức thu nhập trung bình.

⇒ **Kết luận:** k=3 chưa đủ độ chi tiết để tối ưu hóa chiến lược kinh doanh.

Kịch bản 2: Phân cụm với k = 4



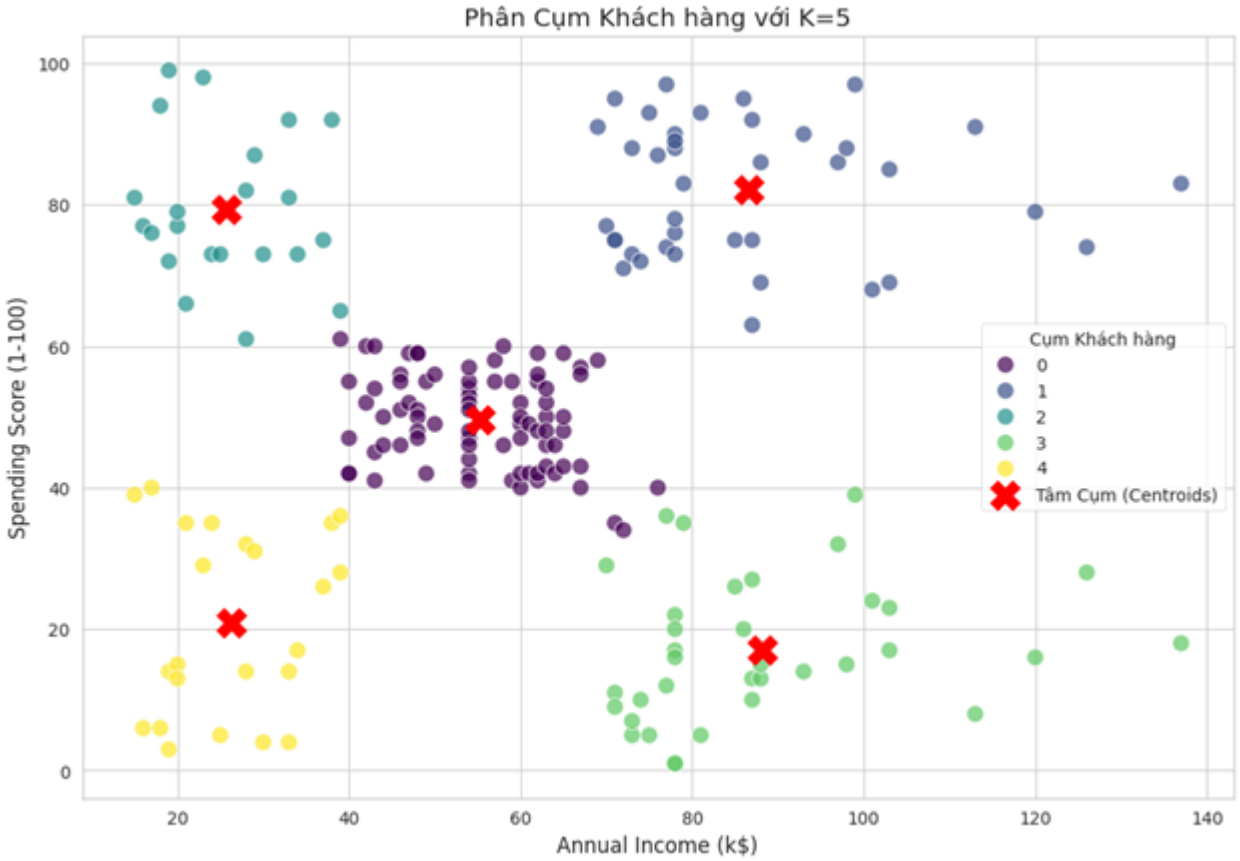
Hình 3.2.6. Biểu đồ phân cụm khách hàng với k=4

Phân tích: Với 4 nhóm, sự phân tách ở hai cực (thu nhập thấp và cao) đã rõ ràng hơn.

- Tuy nhiên, nhóm trung tâm vẫn là một khối lớn chưa được định hình rõ ràng.
- Việc gộp chung khách hàng "trung bình" với các nhóm có xu hướng chi tiêu cao/thấp nhẹ làm giảm hiệu quả của các chương trình khuyến mãi mục tiêu.

⇒ **Kết luận:** Mặc dù k=4 đã cải thiện so với k=3, nhưng việc chưa tách biệt được nhóm khách hàng "Trung bình". Do đó, k=4 vẫn chưa phải là phương án tối ưu nhất để lựa chọn.

Kịch bản 3: Phân cụm với k = 5 (Mô hình tối ưu)



Hình 3.2.7. Biểu đồ phân cụm khách hàng với k=5

Phân tích: Đây là kịch bản cho kết quả phân tách rõ ràng và hợp lý nhất về mặt nghiệp vụ.

- Biểu đồ hiển thị 5 đám mây điểm tách biệt: 4 nhóm ở 4 góc đại diện cho các hành vi tiêu dùng cực đoan và 1 nhóm trung tâm đại diện cho khách hàng đại chúng.
- Kết quả này hoàn toàn phù hợp với phân tích từ biểu đồ Elbow (điểm khuỷu tay) ở phần trước.

⇒ **Quyết định:** Nhóm chọn **k=5** làm kết quả cuối cùng để xây dựng chân dung khách hàng.

Phân tích đặc điểm từng nhóm (Customer Profiling)

Dựa trên kết quả phân cụm với k=5, nhóm tiến hành tính toán giá trị trung bình của các thuộc tính để định danh từng nhóm khách hàng. Bảng thống kê trung bình các cụm (k=5):

--- Đặc điểm các Nhóm Khách hàng (k=5) ---

	Cluster_k5	Annual Income (k$)	Spending Score (1-100)
3	3	88.200000	17.114286
1	1	86.538462	82.128205
0	0	55.296296	49.518519
4	4	26.304348	20.913043
2	2	25.727273	79.363636

Hình 3.2.8. Đặc điểm hành vi của 5 nhóm khách hàng

Mô tả chi tiết các nhóm khách hàng:

- Nhóm Khách hàng Cẩn Thận (Cụm 4):
  - *Đặc điểm:* Thu nhập thấp và chi tiêu rất hạn chế.
  - *Hành vi:* Họ mua sắm có kế hoạch, chi li và thường chỉ mua khi có nhu cầu thiết yếu hoặc giảm giá sâu.
- Nhóm Khách hàng Tiêu Xài (Cụm 2):
  - *Đặc điểm:* Thu nhập thấp nhưng chỉ số chi tiêu lại rất cao.
  - *Hành vi:* Nhóm này có thể là giới trẻ, sinh viên hoặc những người đam mê xu hướng thời trang. Họ nhạy cảm với các sản phẩm "hot" giá rẻ, sẵn sàng chi tiêu vượt mức thu nhập.
- Nhóm Khách hàng Trung Lập (Cụm 0):
  - *Đặc điểm:* Thu nhập và chi tiêu đều ở mức trung bình.
  - *Hành vi:* Đây là nhóm đông đảo nhất (81 khách hàng), đại diện cho dòng khách hàng phổ thông. Họ mua sắm ổn định nhưng không quá đột biến.

- Nhóm Khách hàng Tiết Kiệm (Cụm 3):
  - *Đặc điểm:* Thu nhập rất cao nhưng chi tiêu lại thấp.
  - *Hành vi:* Đây là nhóm khách hàng lý trí hoặc chưa hài lòng với sản phẩm hiện tại của trung tâm. Họ có tiềm lực tài chính mạnh nhưng cần lý do thuyết phục để "mở ví".
- Nhóm Khách hàng Ưu Tú / Mục Tiêu (Cụm 1):
  - *Đặc điểm:* Thu nhập cao và chi tiêu rất mạnh tay.
  - *Hành vi:* Đây là nhóm khách hàng VIP, mang lại lợi nhuận cao nhất. Họ quan tâm đến chất lượng, thương hiệu và trải nghiệm dịch vụ cao cấp hơn là giá cả.

Kết luận

Tổng kết Kỹ thuật và Lựa chọn Mô hình

Quá trình thực nghiệm trên bộ dữ liệu Mall Customers đã chứng minh hiệu quả của thuật toán học không giám sát K-Means trong việc phát hiện các cấu trúc ẩn của dữ liệu khách hàng.

- Về tiền xử lý: Việc chuẩn hóa dữ liệu bằng StandardScaler là bước thiết yếu, giúp loại bỏ sự chênh lệch về đơn vị giữa "Thu nhập" (k) và "Điểm chi tiêu" (1-100), đảm bảo thuật toán tính toán khoảng cách Euclidean chính xác.
- Về xác định số cụm (k): Phương pháp Elbow (Khuỷu tay) đã cung cấp cơ sở định lượng rõ ràng để chọn k. Biểu đồ WCSS cho thấy điểm gãy mạnh nhất tại k=5.
- Về kết quả phân cụm:
  - Các thử nghiệm với k=3 và k=4 cho thấy sự chồng lấn giữa các nhóm khách hàng có hành vi đối lập (ví dụ: gộp chung người tiết kiệm và người chi tiêu nhiều), dẫn đến thông tin thiếu giá trị thực tiễn.
  - Mô hình với k=5 tạo ra sự phân tách rõ ràng nhất trên không gian hai chiều, định hình được 5 nhóm khách hàng riêng biệt không bị trùng lắp. Đây được xác định là mô hình tối ưu cuối cùng.

Chân dung Khách hàng và Khuyến nghị Chiến lược (Dựa trên dữ liệu)

Dựa trên các giá trị trung bình (mean) của từng cụm thu được từ code, chúng ta có thể đúc kết các khuyến nghị hành động cụ thể cho từng nhóm:

- Nhóm Khách hàng Ưu tú (Cluster 1 - Thu nhập Cao, Chi tiêu Cao):
  - *Đặc điểm dữ liệu:* Đây là nhóm có chỉ số lý tưởng nhất.
  - *Khuyến nghị:* Đây là nhóm cần được ưu tiên hàng đầu. Cần áp dụng các chính sách VIP, gửi thông tin sản phẩm cao cấp và duy trì sự hài lòng để tối đa hóa lợi nhuận lâu dài.
- Nhóm Khách hàng Tiềm năng / "Tiêu xài" (Cluster 2 - Thu nhập Thấp, Chi tiêu Cao):
  - *Đặc điểm dữ liệu:* Mặc dù thu nhập thấp nhưng điểm chi tiêu lại rất cao (~79/100).
  - *Khuyến nghị:* Nhóm này có xu hướng mua sắm ngẫu hứng. Chiến lược phù hợp là tiếp thị các sản phẩm thời trang, xu hướng (trending) với mức giá vừa phải hoặc các chương trình khuyến mãi ngắn hạn để kích thích hành vi mua ngay lập tức.
- Nhóm Khách hàng Cần Khai thác / "Tiết kiệm" (Cluster 3 - Thu nhập Cao, Chi tiêu Thấp):
  - *Đặc điểm dữ liệu:* Có năng lực tài chính mạnh (~88k) nhưng điểm chi tiêu lại thấp nhất (~17/100).
  - *Khuyến nghị:* Đây là nhóm khó tiếp cận nhưng tiềm năng lớn. Cần tìm hiểu lý do họ ít chi tiêu (do sản phẩm không phù hợp hay dịch vụ chưa tốt?) để điều chỉnh. Nên tiếp thị các sản phẩm đề cao tính thực dụng, chất lượng bền vững thay vì hào nhoáng.
- Nhóm Khách hàng Trung lập (Cluster 0 - Thu nhập TB, Chi tiêu TB):
  - *Đặc điểm dữ liệu:* Chiếm số lượng đông đảo nhất (81/200 khách).
  - *Khuyến nghị:* Duy trì các chương trình tích điểm đổi quà, ưu đãi định kỳ để giữ chân nhóm này, đảm bảo nguồn doanh thu ổn định cho trung tâm.
- Nhóm Khách hàng Cẩn trọng (Cluster 4 - Thu nhập Thấp, Chi tiêu Thấp):

- *Đặc điểm dữ liệu:* Chỉ số thấp ở cả hai tiêu chí.
- *Khuyến nghị:* Hạn chế chi ngân sách marketing lớn cho nhóm này. Có thể tiếp cận thông qua các đợt xả hàng cuối mùa (Clearance Sale) hoặc các sản phẩm thiết yếu giá rẻ.

Đánh giá chung

Bài thực hành đã hoàn thành mục tiêu đề ra: Từ dữ liệu thô ban đầu, thông qua các kỹ thuật Khai phá dữ liệu (Data Mining), chúng ta đã phân loại thành công tập khách hàng thành các nhóm có ý nghĩa. Kết quả này cung cấp cơ sở định lượng vững chắc để doanh nghiệp chuyển từ cách tiếp cận đại trà sang các chiến lược tiếp thị mục tiêu (Targeted Marketing) hiệu quả hơn.