ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH

Báo cáo Bài tập lớn

Khai phá dữ liệu

Dự đoán doanh số bán hàng

Dựa trên dữ liệu lịch sử

GVHD: Thầy Bùi Tiến Đức
Sinh viên: Lê Võ Đăng Khoa 2211606

TP. Hồ Chí Minh, 10/2025

Contents

1 Danh sách hình ảnh và bảng biểu

List of Figures

List of Tables

2 Mô tả bài toán

2.1 Bối cảnh và vấn đề

Trong ngành bán lẻ, việc dự báo doanh số bán hàng là một trong những bài toán quan trọng và thách thức nhất. Đối với một tập đoàn quy mô toàn cầu như **Walmart**, việc dự báo chính xác có ảnh hưởng trực tiếp đến hiệu quả hoạt động kinh doanh.

Bối cảnh: Hoạt động của một chuỗi siêu thị bán lẻ phụ thuộc vào nhiều yếu tố biến động liên tục như: các ngày lễ trong năm, các chương trình khuyến mãi (*MarkDown*), và các yếu tố kinh tế – xã hội bên ngoài (như giá nhiên liệu, tỷ lệ thất nghiệp).

Vấn đề:

- **Tối ưu tồn kho:** Nếu dự báo quá cao, Walmart sẽ bị tồn đọng hàng hóa, tăng chi phí lưu kho (đặc biệt với hàng hóa dễ hỏng). Ngược lại, nếu dự báo quá thấp, cửa hàng sẽ bị *thiếu hàng*, dẫn đến mất doanh thu và giảm sự hài lòng của khách hàng.
- **Quản lý nhân sự:** Dự báo doanh số giúp ban quản lý cửa hàng sắp xếp lịch làm việc và số lượng nhân viên phù hợp với lượng khách hàng dự kiến, tránh lãng phí chi phí nhân công hoặc thiếu người phục vụ.

Vì vậy, vấn đề đặt ra là cần xây dựng một mô hình dựa trên dữ liệu lịch sử để dự đoán doanh số bán hàng hàng tuần một cách chính xác, giúp Walmart đưa ra các quyết định kinh doanh hiệu quả hơn.

2.2 Mô tả dữ liệu

Nguồn dữ liệu: Bộ dữ liệu được sử dụng có tên *"Walmart Sales Forecast"*, được thu thập và công bố công khai trên nền tảng Kaggle (tại địa chỉ: https://www.kaggle.com/datasets/aslanahmedov/walmart-sales-forecast).

Mô tả dữ liệu: Bộ dữ liệu bao gồm thông tin bán hàng lịch sử của **45 cửa hàng Walmart** trong giai đoạn từ năm 2010 đến 2012. Dữ liệu được chia thành ba tệp chính như sau:

Table 2.2.1: Mô tả `stores.csv`

Tên thuộc tính	Kiểu dữ liệu	Ý nghĩa
Store	Integer	Mã định danh của cửa hàng (từ 1 đến 45)
Type	String	Loại cửa hàng (A, B hoặc C)
Size	Integer	Diện tích của cửa hàng

Table 2.2.2: Mô tả tệp `features.csv`

Tên thuộc tính	**Kiểu dữ liệu**	**Ý nghĩa**
Store	Integer	Mã cửa hàng (khóa ngoại liên kết với `stores.csv`)
Date	Date	Ngày tháng năm
Temperature	Double	Nhiệt độ trung bình tại khu vực cửa hàng
Fuel_Price	Double	Giá nhiên liệu trung bình tại khu vực
MarkDown1–5	Double	Dữ liệu ẩn danh về 5 loại chương trình giảm giá (*MarkDown*). Giá trị `NaN` nghĩa là không có giảm giá.
CPI	Double	Chỉ số giá tiêu dùng
Unemployment	Double	Tỷ lệ thất nghiệp tại khu vực
IsHoliday	Boolean	Đánh dấu tuần đó có phải tuần lễ đặc biệt hay không

Table 2.2.3: Mô tả tệp `train.csv`

Tên thuộc tính	**Kiểu dữ liệu**	**Ý nghĩa**
Store	Integer	Mã cửa hàng
Dept	Integer	Mã định danh phòng ban (ví dụ: quần áo, điện tử, ...)
Date	Date	Ngày tháng năm
Weekly_Sales	Double	**Biến mục tiêu** (*Target Variable*) –Doanh số bán hàng trong tuần
IsHoliday	Boolean	Đánh dấu tuần lễ đặc biệt

2.3 Mục tiêu khai phá dữ liệu

Dựa trên bối cảnh và bộ dữ liệu được cung cấp, các mục tiêu khai phá dữ liệu của nhóm được xác định như sau:

Mục tiêu chính: Xây dựng một mô hình có khả năng dự đoán tương đối chính xác giá trị `Weekly_Sales` (doanh số hàng tuần) cho từng phòng ban tại từng cửa hàng.

Mục tiêu phụ:

- Thực hiện phân tích khám phá dữ liệu để hiểu rõ đặc điểm của dữ liệu và mối quan hệ giữa các biến (ví dụ: doanh số tăng hay giảm vào ngày lễ? Nhiệt độ ảnh hưởng đến doanh số ra sao?).
- Phân tích và xác định các yếu tố (*features*) có ảnh hưởng quan trọng nhất đến doanh số bán hàng.
- Sử dụng và đánh giá một vài thuật toán Data Mining khác nhau để tìm ra mô hình có kết quả dự đoán tốt nhất.

3 Tiền xử lý dữ liệu

3.1 Tích hợp dữ liệu

Dữ liệu ban đầu được phân tách thành ba tệp: `train.csv` (chứa dữ liệu doanh số theo phòng ban), `features.csv` (chứa dữ liệu đặc trưng theo tuần của cửa hàng), và `stores.csv` (chứa thông tin mô tả cửa hàng).

Để chuẩn bị cho việc phân tích, nhóm đã tiến hành gộp ba tệp này thành một `DataFrame` duy nhất (`df`) bằng cách sử dụng phương thức `merge` của thư viện `Pandas`, dựa trên các khóa chung như sau:

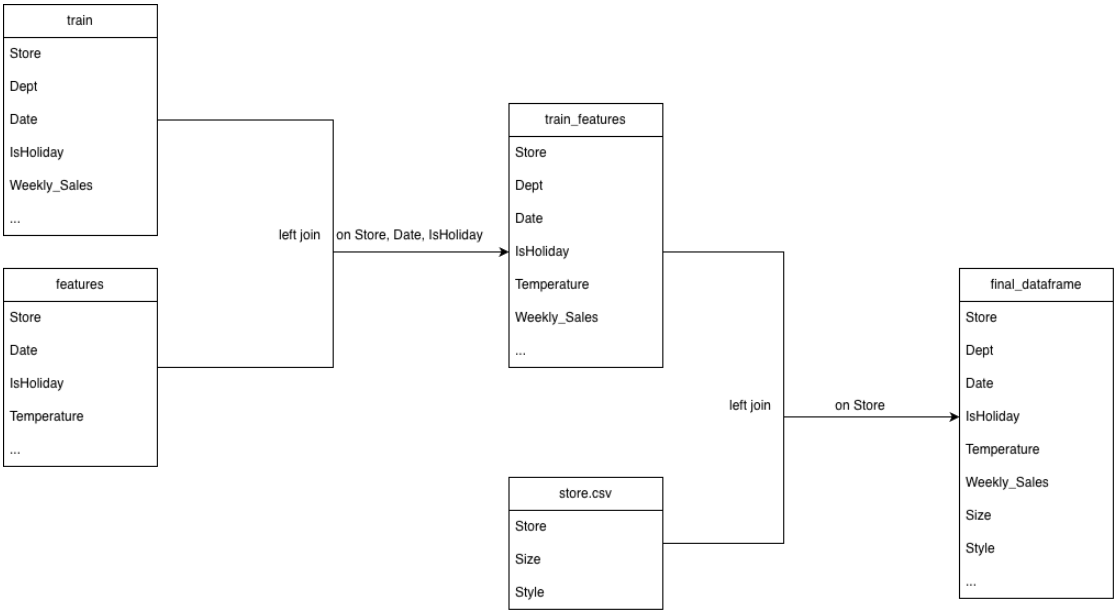

Figure 3.1.1: Tích hợp dữ liệu

- **Gộp `train.csv` với `features.csv`:**
  - **Loại gộp:** Sử dụng *Left Join* (`how="left"`), với `train.csv` là bảng bên trái.
  - **Khóa (Keys):** Gộp trên bộ khóa gồm 3 cột: `["Store", "Date", "IsHoliday"]`. Việc này đảm bảo mỗi bản ghi doanh số của từng phòng ban được ánh xạ chính xác với các đặc điểm của cửa hàng vào đúng ngày tương ứng.
- **Gộp kết quả với `stores.csv`:**
  - **Loại gộp:** Tiếp tục sử dụng *Left Join* (`how="left"`).
  - **Khóa (Key):** Gộp trên cột `["Store"]`.
  - **Mục đích:** Thêm thông tin về loại cửa hàng (`Type`) và kích thước (`Size`) vào từng bản ghi.

Kết quả: Quá trình này tạo ra một `DataFrame` duy nhất, thống nhất chứa **421.570 bản ghi**. DataFrame này bảo toàn tất cả các mẫu trong tệp `train.csv` gốc và được làm giàu thêm các đặc trưng từ hai tệp `features.csv` và `stores.csv`, sẵn sàng cho bước **Khám phá Dữ liệu (EDA)**.

3.2 Khám phá dữ liệu (Exploratory Data Analysis - EDA)

Giai đoạn EDA được thực hiện với mục tiêu tìm hiểu cấu trúc, phân phối và các mối quan hệ bên trong bộ dữ liệu. Nhóm đã sử dụng thư viện **Pandas** để tải và tính toán thống kê, cùng với thư viện **Seaborn** và **Matplotlib** để trực quan hóa.

3.2.1 Thống kê mô tả (Descriptive Statistics)

Sau khi tải và gộp dữ liệu, nhóm thực hiện hàm `.describe()` trên các cột dữ liệu số để có cái nhìn tổng quan đầu tiên.

Table 3.2.1: Thống kê mô tả các thuộc tính số

Thuộc tính	Count	Mean	Std	Min	25%	50% (Median)	75%	Max
Store	421570	22.20	12.79	1.00	11.00	22.00	33.00	45.00
Dept	421570	44.26	30.49	1.00	18.00	37.00	74.00	99.00
Weekly_Sales	421570	15981.26	22711.18	-4988.94	2079.65	7612.03	20205.85	693099.36
Temperature	421570	60.09	18.45	-2.06	46.68	62.09	74.28	100.14
Fuel_Price	421570	3.36	0.46	2.47	2.93	3.45	3.74	4.47
MarkDown1	150681	7246.42	8291.22	0.27	2240.27	5347.45	9210.90	88646.76
MarkDown2	111248	3334.63	9475.36	-265.76	41.60	192.00	1926.94	104519.54
MarkDown3	137091	1439.42	9623.08	-29.10	5.08	24.60	103.99	141630.61
MarkDown4	134967	3383.17	6292.38	0.22	504.22	1481.31	3595.04	67474.85
MarkDown5	151432	4628.98	5962.89	135.16	1878.44	3359.45	5563.80	108519.28
CPI	421570	171.20	39.16	126.06	132.02	182.32	212.42	227.23
Unemployment	421570	7.96	1.86	3.88	6.89	7.87	8.57	14.31
Size	421570	136727.92	60980.58	34875	93638	140167	202505	219622

Nhận xét Weekly_Sales từ Bảng 3.1.1:

- **Giá trị *Min* là số âm (∼4960.94):** có thể đại diện cho các giao dịch bị trả hàng (*refunds*) hoặc lỗi nhập liệu. Đây là một vấn đề cần xử lý trong bước làm sạch dữ liệu.
- **Trung bình (mean) là 15,981 và trung vị (med): 7,612:** Giá trị trung bình lớn hơn đáng kể so với giá trị trung vị. Điều này cho thấy dữ liệu bị lệch phải (right-skewed).

3.2.2 Phân tích biến mục tiêu (Target Variable Analysis)

Biến mục tiêu của bài toán là **Weekly_Sales**. Việc hiểu rõ phân phối của nó là điều bắt buộc. Nhóm đã sử dụng biểu đồ phân phối (*Histogram*) để trực quan hóa.

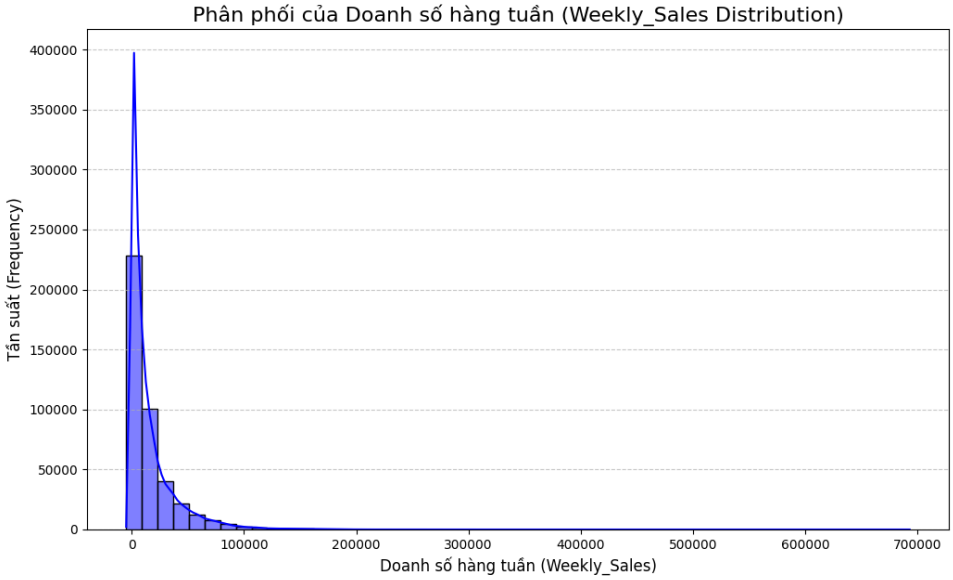

Figure 3.2.1: Phân phối của Doanh số hàng tuần (Weekly_Sales Distribution)

Phân tích Hình 3.1.1:

- Biểu đồ này xác nhận một cách trực quan những gì chúng ta thấy trong `.describe()`
- **Phân phối lệch:** Biểu đồ cho thấy rõ ràng rằng `Weekly_Sales` không tuân theo phân phối chuẩn mà bị **lệch phải (right-skewed)** rất mạnh. Điều này có nghĩa là phần lớn doanh số hàng tuần ở mức thấp đến trung bình, và chỉ một số ít tuần có doanh số cực cao (tạo thành "đuôi phải").
- **Giá trị âm:** Quan sát ở phía bên trái trục 0 có một nhóm nhỏ dữ liệu âm, xác nhận các giá trị âm trong Bảng 3.1.1.
- **Ý nghĩa:** Phần lớn các quan sát (doanh số hàng tuần của một phòng ban) có giá trị tương đối thấp (khoảng 0 - 50,000), nhưng có một số ít trường hợp có doanh số rất cao (các giá trị ngoại lệ - outliers) kéo giá trị trung bình lên. Những ngoại lệ này có thể là các tuần lễ hội lớn (Black Friday, Giáng sinh).

3.2.3 Phân tích tương quan (Correlation Analysis)

Để hiểu mối quan hệ tuyến tính giữa các biến số, nhóm đã tính toán **ma trận tương quan (correlation matrix)** và trực quan hóa bằng **biểu đồ nhiệt (heatmap)**.

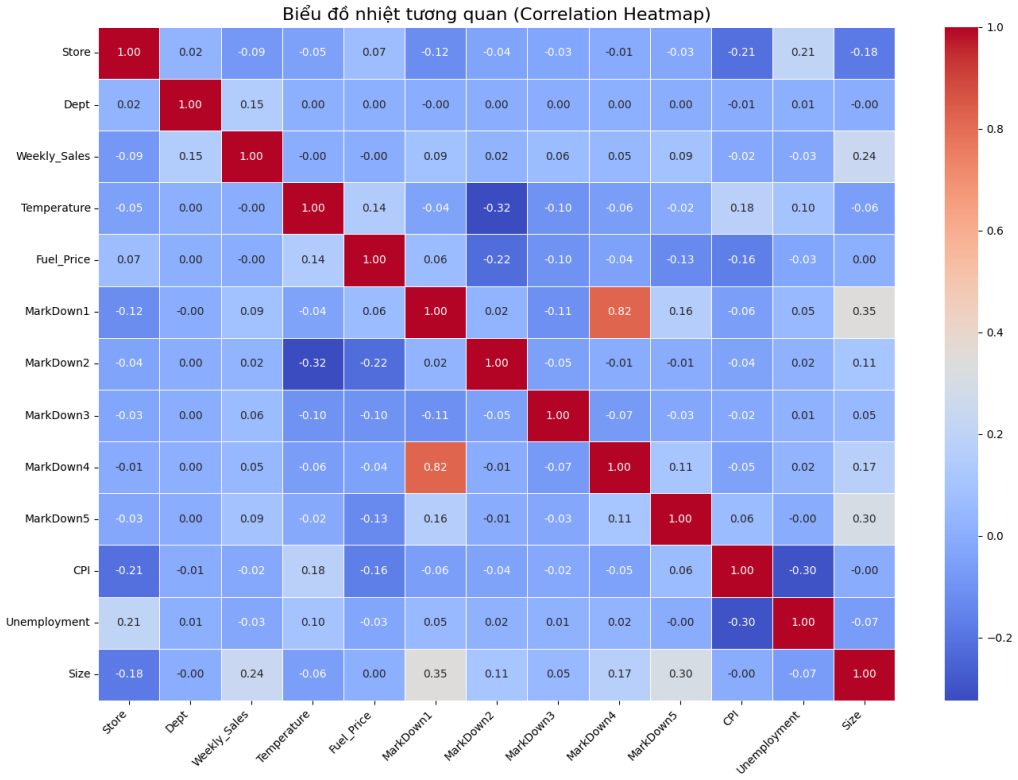

Figure 3.2.2: Biểu đồ nhiệt tương quan giữa các biến số

Phân tích Hình 3.1.2:

- Biểu đồ nhiệt hiển thị hệ số tương quan Pearson:
  - Giá trị gần +1.0 (màu đỏ đậm): tương quan đồng biến mạnh (X tăng thì Y tăng).
  - Giá trị gần −1.0 (màu xanh đậm): tương quan nghịch biến mạnh (X tăng thì Y giảm).
  - Giá trị gần 0: ít hoặc không có tương quan tuyến tính.
- **Phát hiện quan trọng về `Weekly_Sales`:**
  - **Hàng `Weekly_Sales`:** đa số các ô (so với các biến số khác như `Temperature`, `Fuel_Price`, `CPI`, `Unemployment`, và các `MarkDown`) đều có màu rất nhạt, và các con số đều rất gần 0 (ví dụ: -0.00, 0.00, 0.09, -0.02, -0.03).
  - **Kích thước Cửa hàng (Size):** là yếu tố có ảnh hưởng tuyến tính MẠNH NHẤT, nhưng vẫn chỉ ở mức yếu (+0.24).
  - **Tuy nhiên:** Điều này KHÔNG có nghĩa là chúng không quan trọng. Nó chỉ có nghĩa là một mô hình tuyến tính đơn giản (ví dụ:**Sales** $= A \times$ **Temperature** $+ B$) sẽ không hoạt động tốt. Mối quan hệ có thể phức tạp hơn (phi tuyến tính).
- **Phát hiện thú vị khác:** `MarkDown1` và `MarkDown4` có tương quan dương rất mạnh ($r \approx +0.85$), cho thấy hai chương trình khuyến mãi này thường được tung ra cùng lúc. Hiện tượng này cần được lưu ý vì có thể ảnh hưởng đến độ ổn định của các mô hình hồi quy.

3.3 Các bước làm sạch và chuẩn bị dữ liệu

3.3.1 Xử lý giá trị thiếu (Missing Values)

Phát hiện ban đầu: Từ kết quả `df.info()`, tập dữ liệu có 421,570 bản ghi. Tuy nhiên, các cột `MarkDown1 - MarkDown5` có số lượng giá trị `non-null` thấp hơn đáng kể:

- `MarkDown1`: 150,681 (thiếu ∼64.2%)
- `MarkDown2`: 111,248 (thiếu ∼73.6%)
- `MarkDown3`: 137,091 (thiếu ∼67.5%)
- `MarkDown4`: 134,967 (thiếu ∼68.0%)
- `MarkDown5`: 151,432 (thiếu ∼64.1%)

Hành động: Điền giá trị `0` vào các cột `MarkDown` bị thiếu.

Lý do: Trong bối cảnh bán lẻ, giá trị thiếu ở các cột giảm giá (`MarkDown`) không nhất thiết là lỗi, mà thường đại diện cho trường hợp "không có chương trình giảm giá". Do đó, việc thay thế các giá trị thiếu bằng `0` (tương đương không giảm giá) là hợp lý và nhất quán về mặt nghiệp vụ.

3.3.2 Xử lý nhiễu và giá trị ngoại lệ (Noise/Outliers)

Phát hiện ban đầu:

- Kết quả `df.describe()` cho thấy giá trị tối thiểu của `Weekly_Sales` là `-4988.94`, một bất thường nghiêm trọng.
- Phát hiện 1,285 bản ghi có `Weekly_Sales < 0`.

Hành động: Đã xử lý bằng cách gán các giá trị âm thành `0`.

Lý do:

- **Nguyên nhân khả dĩ:**
  - Lỗi nhập liệu (data entry error).
  - Doanh thu ròng (Net Sales) có thể bị âm nếu số hàng trả lại vượt quá hàng bán ra.
- **Tại sao không xoá dữ liệu:** Xoá 1,285 hàng (0.3%) sẽ làm mất thông tin quý giá khác.
- **Tại sao không lấy giá trị tuyệt đối:** Dễ gây sai lệch vì biến âm thành dương.
- **Giải pháp chọn:** Chuyển giá trị âm thành `0` là hợp lý nhất, xem như "không có doanh thu".

Kỹ thuật áp dụng: Đây là một kỹ thuật *Outlier Handling* dạng **Flooring/Capping** –chặn các giá trị vượt ngoài ngưỡng hợp lý (ở đây là nhỏ hơn 0) về giá trị biên hợp lệ.

3.3.3 Chọn lọc thuộc tính (Feature Engineering & Selection)

Feature Engineering (Tạo đặc trưng):

Phát hiện ban đầu: Cột `Type` có kiểu `object`, cột `Date` có kiểu `object` và `IsHoliday` có kiểu `bool`. Các mô hình học máy không thể xử lý trực tiếp các kiểu dữ liệu này.

Hành động: Tạo các đặc trưng mới:

- `Year`, `Month`, `WeekOfYear`, `Day`
- Chuyển `IsHoliday` từ `True/False` sang `1/0`.
- Tạo 3 cột `Type_A, Type_B, Type_C` và cho chúng kiểu `1/0` dựa theo `Type`.

Lý do:

- **Xu hướng (Trend):** Cột `Year` giúp mô hình nhận biết sự thay đổi doanh số qua các năm.
- **Tính thời vụ (Seasonality):** `Month` và `WeekOfYear` hỗ trợ mô hình học chu kỳ bán hàng.
- **Mã hóa biến Phân loại bằng One-Hot Encoding:** cho phép mô hình (đặc biệt là Linear Regression) gán một **trọng số (hệ số) riêng biệt** cho từng loại mà không tạo ra bất kỳ một "thứ tự" giả mạo nào

Feature Selection (Lựa chọn đặc trưng):

Hành động: Sử dụng `SelectKBest` với hàm `f_regression` để đánh giá tầm quan trọng của từng đặc trưng trong việc dự đoán `Weekly_Sales`.

Kết quả:

Đặc trưng	Điểm (Score)
Size	26647.905144
Type_A	15009.499841
Dept	9445.215074
Type_B	7385.855222
Type_C	3871.207762
Store	3082.286020
MarkDown5	1076.346995
MarkDown1	940.157460
MarkDown3	627.806676
MarkDown4	592.619310
Month	340.566300
WeekOfYear	323.135235
Unemployment	282.117400
CPI	184.627819
MarkDown2	181.034378
IsHoliday	68.811742
Year	43.114262
Day	16.140026
Temperature	2.254012
Fuel_Price	0.006127

Nhận xét:

- **Quan trọng nhất:** Các đặc trưng `Size` (kích thước), `Type` (loại cửa hàng), và `Dept` (phòng ban) có điểm số cao vượt trội so với phần còn lại. Điều này khẳng định yếu tố quan trọng nhất để dự đoán doanh số là "đó là cửa hàng nào" và "phòng ban nào".
- **Không quan trọng:** `Fuel_Price` (giá xăng) và `Temperature` (nhiệt độ) có điểm số gần như bằng 0. Điều này xác nhận chúng gần như không có ảnh hưởng (một cách tuyến tính), hoàn toàn phù hợp với kết quả từ biểu đồ tương quan.

3.3.4 Chuẩn hóa dữ liệu (Normalization)

Hành động: Áp dụng `StandardScaler` để chuẩn hóa các cột số: `Temperature`, `Fuel_Price`, `CPI`, `Size` `Unemployment`, và các cột `MarkDown`.

Kết quả: Sau khi chuẩn hóa, tất cả các đặc trưng có `mean` $\sim$`0,00` và `std` $\sim$`1,00`.

Lý do: Các đặc trưng ban đầu có thang đo rất khác nhau (ví dụ: `MarkDown1` tới 88,000 trong khi `Fuel_Price` chỉ quanh 3–4). Chuẩn hóa theo **Z-score** giúp đưa chúng về cùng thang đo, đảm bảo mô hình học dựa trên sức mạnh dự đoán thực, không bị ảnh hưởng bởi độ lớn của giá trị.

3.4 Kết quả sau tiền xử lý

3.4.1 Thay đổi về cấu trúc và tính đầy đủ (Từ `df.info()`)

- **Tính toàn vẹn dữ liệu:** Tổng số bản ghi vẫn là 421,570, cho thấy không có hàng nào bị xóa trong quá trình làm sạch.
- **Xử lý giá trị thiếu:** Tất cả 21 cột hiện tại đều có 421,570 giá trị `non-null`. Điều này xác nhận rằng 5 cột `MarkDown1–5` (trước đây bị thiếu dữ liệu nghiêm trọng) đã được điền đầy đủ bằng giá trị 0.
- **Feature Engineering:**
  - Tổng số cột tăng từ 16 lên 21 cột.
  - Cột `Date` và `Type` (kiểu `object`) đã bị loại bỏ.
  - Các đặc trưng thời gian mới được tạo ra: `Year`, `Month`, `Day`, `WeekOfYear`, `Type_A`, `Type_B`, `Type_C`.
  - Cột `IsHoliday` (trước đây là `bool`) đã được chuyển đổi thành kiểu `int64` (0 hoặc 1).
- **Kiểu dữ liệu:** Tập dữ liệu cuối cùng gồm 11 cột `float`, 10 cột `int` —tất cả đều là định dạng số mà mô hình học máy có thể đọc được.

3.4.2 Chuẩn hóa và biến đổi dữ liệu (Từ `df.describe()`)

- **Xác nhận chuẩn hóa:** Tất cả 10 cột được đưa vào `StandardScaler` (`Temperature`, `Fuel_Price`, `MarkDown1–5`, `CPI`, `Unemployment`, `Size`) đều có giá trị trung bình xấp xỉ 0.00 và độ lệch chuẩn là 1.00.
- **Ý nghĩa:** Việc chuẩn hóa loại bỏ ảnh hưởng của thang đo khác biệt, giúp mô hình đánh giá đúng tầm quan trọng thực sự của các đặc trưng.

- **Quan sát từ `df.head()`:** Các cột như `Temperature` và `Fuel_Price` không còn giá trị gốc (ví dụ: 42.31, 2.572) mà là các giá trị Z-score đã được chuẩn hóa (ví dụ: -0.963, -1.720).

3.4.3 Xử lý nhiễu (Từ log xử lý dữ liệu)

- Log xác nhận rằng 1,285 bản ghi `Weekly_Sales` âm đã được phát hiện và xử lý.
- Không có hàng nào bị xóa, xác nhận rằng chiến lược flooring các giá trị âm về 0 đã được thực hiện thành công.
- Điều này giúp bảo toàn dữ liệu mà vẫn đảm bảo tính logic cho biến mục tiêu.

3.4.4 Tóm tắt trước và sau tiền xử lý

Table 3.4.1: So sánh trạng thái dữ liệu trước và sau tiền xử lý

Đặc điểm	**Trước tiền xử lý**	**Sau tiền xử lý**
Tổng số cột	16	21
Giá trị Null	Có (ở 5 cột MarkDown)	Không
Cột `Date`	`object (chuỗi)`	Bị loại bỏ, thay bằng `Year`, `Month`, `WeekOfYear`, `Day`
Cột `Type`	`object (chuỗi)`	Bị loại bỏ, thay bằng `Type_A`, `Type_B`, `Type_C`
Cột `IsHoliday`	`bool (True/False)`	`int (1/0)`
Thang đo (Scale)	Khác biệt lớn (ví dụ: 100.14 vs 88,646.76)	Các cột số liên tục đã được chuẩn hóa (`Mean=0, Std=1`)
Giá trị âm `Weekly_Sales`	Có (1,285 bản ghi)	Không (min = 0)

3.4.5 Kết luận

Tập dữ liệu `df` hiện đã hoàn toàn sạch, đầy đủ, có cấu trúc và được chuẩn hóa. Nó đã sẵn sàng cho bước tiếp theo là **phân chia dữ liệu (train/test split)** và đưa vào các mô hình học máy để huấn luyện.

4 Áp dụng thuật toán

Sau khi dữ liệu đã được làm sạch, tiền xử lý, bước tiếp theo là lựa chọn và áp dụng các thuật toán học máy phù hợp để xây dựng mô hình dự đoán.

4.1 Lựa chọn Nhóm Thuật toán

4.1.1 Phân tích yêu cầu bài toán

Mục tiêu cốt lõi của dự án này là **dự đoán giá trị của biến `Weekly_Sales` (Doanh số hàng tuần)**.
Từ quá trình Khám phá Dữ liệu (EDA) và Tiền xử lý, ta xác định:

- **Biến mục tiêu (Target Variable):** `Weekly_Sales`
- **Bản chất của biến:** Đây là một *giá trị số liên tục* (continuous), không phải là nhãn hoặc danh mục.
- Giá trị có thể là bất kỳ số thực dương nào: ví dụ 15981.25, 2079.65, v.v.

Do đó, bài toán đặt ra câu hỏi:

> "*Doanh số tuần tới sẽ là bao nhiêu?*"

4.1.2 Lý do lựa chọn Hồi quy (Regression)

Dựa trên bản chất của biến mục tiêu, nhóm thuật toán phù hợp để giải quyết bài toán này là **Hồi quy (Regression)**.
Hồi quy là một lớp các thuật toán học máy có giám sát (*Supervised Learning*) được thiết kế để dự đoán một giá trị đầu ra **liên tục**. Mục tiêu của mô hình hồi quy là học một hàm toán học f từ các đặc trưng đầu vào X (ví dụ: `Store`, `Dept`, `Temperature`, `IsHoliday`) để ánh xạ đến giá trị đầu ra y (tức là `Weekly_Sales`) sao cho:

$$y \approx f(X)$$

Nói cách khác, mô hình sẽ cố gắng tìm ra một *đường thẳng* biểu diễn mối quan hệ giữa các yếu tố đầu vào và doanh số.

4.1.3 Phân tích và loại trừ các nhóm thuật toán khác

Để làm rõ hơn lý do lựa chọn Hồi quy, bảng dưới đây trình bày sự khác biệt giữa các nhóm thuật toán phổ biến và lý do tại sao chúng không phù hợp với bài toán này:

Table 4.1.1: Phân tích các nhóm thuật toán và lý do loại trừ

Nhóm Thuật toán	Mục đích chính	Vì sao không phù hợp với bài toán này?
Classification (Phân loại)	Dự đoán một nhãn hoặc danh mục rời rạc (ví dụ: "Email là Spam/Không Spam", "Doanh thu là Cao/Thấp").	Mục tiêu của chúng ta là dự đoán một *giá trị số chính xác* (ví dụ: $15,981), không phải một nhãn. Việc ép `Weekly_Sales` thành nhãn sẽ làm mất nhiều thông tin liên tục.
Clustering (Phân cụm)	Gom nhóm các điểm dữ liệu tương đồng mà không cần nhãn (học không giám sát).	Bài toán này có biến mục tiêu rõ ràng là `Weekly_Sales`. Đây là một bài toán học có giám sát, nên không thể dùng phân cụm.
Association Rules (Luật kết hợp)	Tìm các quy tắc đồng xuất hiện (ví dụ: "Nếu mua Sữa thì thường mua Bánh mì").	Kỹ thuật này dùng để tìm mối quan hệ giữa các đặc trưng, không dùng để dự đoán một giá trị số.

4.1.4 Kết luận

Từ phân tích trên, có thể kết luận rằng:

- Bản chất của bài toán là dự đoán một **biến liên tục**.
- Do đó, nhóm thuật toán **Hồi quy (Regression)** là lựa chọn phù hợp nhất và duy nhất.
- Các bước tiếp theo sẽ tập trung vào việc lựa chọn và đánh giá các thuật toán cụ thể trong nhóm Hồi quy, ví dụ:
  - `Linear Regression`
  - `Ridge / Lasso Regression`
  - `Decision Tree Regression`
  - `Gradient Boosting / XGBoost / LightGBM`
  - `Random Forest Regressor`
  - `Neural Network (MLP Regressor)`

4.2 Hồi quy tuyến tính (Linear Regression)

4.2.1 Giới thiệu

Hồi quy tuyến tính (Linear Regression) là một thuật toán cơ bản, giả định rằng có một mối quan hệ tuyến tính giữa các đặc trưng đầu vào (ví dụ: `Temperature`, `CPI`) và biến mục tiêu (`Weekly_Sales`). Mô hình này cố gắng tìm ra phương trình đường thẳng phù hợp nhất để dự đoán y từ X:

$$y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p + \epsilon \tag{1}$$

4.2.2 Lý do áp dụng

Linear Regression được áp dụng chủ yếu như một mô hình cơ sở (*baseline model*). Đây là mô hình đơn giản, dễ huấn luyện và cung cấp điểm chuẩn cho việc đánh giá các mô hình phức tạp hơn. Mọi mô hình sau này đều cần chứng minh rằng hiệu quả vượt trội hơn mô hình tuyến tính cơ bản này.

4.2.3 Kết quả (Tập kiểm tra)

Sau khi huấn luyện trên **337,256 mẫu** và kiểm tra trên **84,314 mẫu**, kết quả thu được:

- R-squared (R^2): **0.0926**
- MAE (Lỗi tuyệt đối trung bình): **$14,561.79**
- RMSE (Lỗi toàn phương trung bình): **$21,752.04**

4.2.4 Phân tích

- **$R^2 = 0.0926$:** Mô hình chỉ giải thích được 9.26% sự biến động của `Weekly_Sales`, tức là hầu như không có quan hệ tuyến tính giữa đặc trưng và mục tiêu.
- **Sai số lớn:** MAE ở mức $14,500 trong khi trung vị doanh số chỉ khoảng $7,612. Điều này cho thấy độ sai lệch rất cao.

Kết luận: Hồi quy tuyến tính không phù hợp với bộ dữ liệu này.

4.3 Hồi quy Cây quyết định (Decision Tree Regression)

4.3.1 Giới thiệu

Hồi quy Cây quyết định (Decision Tree Regression) là một mô hình phi tham số, hoạt động bằng cách chia dữ liệu thành các tập con nhỏ dựa trên các quy tắc *if–then–else*. Mục tiêu là chia dữ liệu sao cho các mẫu trong cùng một "lá"(*leaf node*) có giá trị `Weekly_Sales` gần nhau nhất.

4.3.2 Lý do áp dụng

Decision Tree được chọn vì những lý do sau:

- **Nắm bắt quan hệ phi tuyến tính:** Không giống Linear Regression, mô hình này có thể học được mối quan hệ phức tạp giữa các đặc trưng.
- **Tự động học quy tắc kết hợp:** Ví dụ, "*Nếu Store = 1 và IsHoliday = 1 thì Weekly_Sales tăng mạnh.*"
- **Nhược điểm:** Dễ bị *overfitting* (học vẹt) nếu cây quá sâu.

4.3.3 Kết quả (Tập kiểm tra)

- R-squared (R^2): **0.9617**
- MAE (Lỗi tuyệt đối trung bình): **$1,766.13**
- RMSE (Lỗi toàn phương trung bình): **$5,568.92**

4.3.4 Phân tích

- **Hiệu suất vượt trội:** R^2 đạt 96,17%, mô hình đã học được phần lớn quy luật ẩn trong dữ liệu.
- **Sai số giảm mạnh:**
  - MAE giảm từ \$14,561 xuống \$1,766 (giảm 88%)
  - RMSE giảm từ \$21,752. xuống \$4,466 (giảm 79%)
- **Kết luận:** Decision Tree rất phù hợp với bài toán dự đoán doanh số, xác nhận rằng mối quan hệ giữa các đặc trưng là phi tuyến tính.

4.4 Hồi quy Rừng ngẫu nhiên (Random Forest Regression)

4.4.1 Giới thiệu

Random Forest là một thuật toán học máy Ensemble (Kết hợp). Thay vì chỉ xây dựng một Cây quyết định (Decision Tree) duy nhất, nó xây dựng một "khu rừng" gồm hàng trăm cây (trong trường hợp của chúng ta là 100 cây).
Nguyên lý hoạt động:

- **Bagging (Bootstrap Aggregating)**: Mỗi cây trong rừng được huấn luyện trên một mẫu dữ liệu ngẫu nhiên (có lặp lại) từ tập huấn luyện.
- **Random Feature**: Tại mỗi nút của cây, thay vì xem xét tất cả các đặc trưng, cây chỉ được phép chọn ngẫu nhiên từ một tập con các đặc trưng.
- **Bỏ phiếu (Averaging)**: Đối với bài toán Hồi quy, dự đoán cuối cùng của Random Forest là giá trị **trung bình** của dự đoán từ tất cả 100 cây.

4.4.2 Lý do áp dụng

Chúng ta đã thấy Decision Tree (R^2=0.9617) hoạt động rất tốt, nhưng nó có một điểm yếu chí mạng là overfitting (học vẹt). Một cây quyết định đơn lẻ sẽ cố gắng học thuộc lòng mọi chi tiết trong dữ liệu huấn luyện, bao gồm cả nhiễu, dẫn đến việc dự đoán kém trên dữ liệu mới.
Random Forest là giải pháp trực tiếp cho vấn đề này:

- **Chống Overfitting**: Bằng cách lấy trung bình dự đoán của 100 cây (mỗi cây hơi khác nhau một chút), các lỗi và "sự học vẹt" ngẫu nhiên của từng cây sẽ tự triệt tiêu lẫn nhau.
- **Tính ổn định (Robustness)**: Mô hình cuối cùng trở nên ổn định và có khả năng **tổng quát hóa (generalize)** tốt hơn trên dữ liệu mà nó chưa từng thấy (tập test).
- **Độ chính xác cao**: Đây là một trong những thuật toán "tiêu chuẩn vàng" trong học máy, thường xuyên cho độ chính xác cao mà không cần tinh chỉnh quá nhiều.

4.4.3 Kết quả (Tập kiểm tra)

- R-squared (R^2): **0.9778**
- MAE (Lỗi tuyệt đối trung bình): **\$1,330.80**
- RMSE (Lỗi toàn phương trung bình): **\$3,404.74**

4.4.4 Phân tích

- **So sánh với Decision Tree**: R^2 đã tăng từ 0.9617 lên 0.9778. Quan trọng hơn, chỉ số lỗi MAE/RMSE đã giảm từ 1,766/4,466 xuống còn 1,330/3,404.
- **Ý nghĩa**: Điều này cho thấy Random Forest không chỉ giữ được khả năng nắm bắt các quy luật phi tuyến tính phức tạp của Decision Tree, mà còn cải thiện nó bằng cách giảm nhiễu và tạo ra các dự đoán chính xác hơn.
- **Kết luận**: Random Forest là mô hình vượt trội, cân bằng được giữa độ chính xác cao và khả năng chống overfitting, khiến nó trở thành mô hình tốt nhất trong ba mô hình chúng ta đã thử nghiệm.

4.5 So sánh kết quả

4.5.1 Bảng so sánh hiệu suất

Table 4.5.1: So sánh hiệu suất giữa các thuật toán

Chỉ số (Metric)	Linear Regression (Cơ sở)	Decision Tree Regression (Bị Overfitting)	Random Forest Regression (Mô hình tối ưu)
R-squared (R^2)	0.0926	0.9617	0.9778
MAE	$14,561.79	$1,766.13	$1,330.80

4.5.2 Trực quan hóa so sánh

Biểu đồ so sánh (MAE, R^2) cho thấy một câu chuyện rõ ràng:

- **Linear Regression** thất bại trong việc mô hình hóa dữ liệu (lỗi cao nhất).
- **Decision Tree** cải thiện đáng kể bằng cách nắm bắt các quy tắc phi tuyến tính (lỗi giảm mạnh).
- **Random Forest** tinh chỉnh và tối ưu hóa, giảm lỗi xuống mức thấp nhất bằng cách kết hợp sức mạnh của nhiều cây và loại bỏ nhiễu.

4.5.3 Kết luận chung

Cuộc thử nghiệm này minh họa một quy trình chuẩn trong khoa học dữ liệu:

- Bắt đầu với một **mô hình cơ sở (Linear Regression)** để hiểu giới hạn của dữ liệu (là tuyến tính hay không).
- Thử một **mô hình phức tạp hơn (Decision Tree)** để nắm bắt các quy luật phi tuyến tính, chấp nhận rủi ro overfitting.
- Kết thúc bằng một **mô hình Ensemble (Random Forest)** để giữ lại sức mạnh của mô hình phức tạp nhưng loại bỏ overfitting, tạo ra một mô hình cuối cùng mạnh mẽ, chính xác và đáng tin cậy.

5 Giao diện người dùng

Phần này mô tả các thành phần chính của giao diện ứng dụng, bao gồm quá trình tải dữ liệu, khám phá, tiền xử lý, huấn luyện mô hình và chạy thử dự đoán. Các hình minh họa bên dưới cho thấy luồng thao tác trực quan mà người dùng trải nghiệm trong ứng dụng.

5.1 Giao diện tải tập dữ liệu

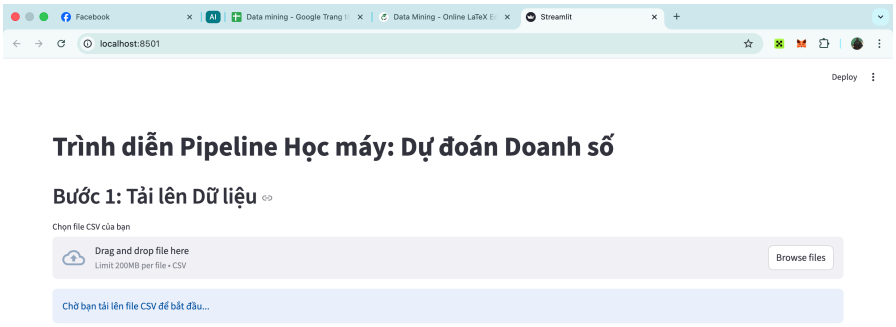

Figure 5.1.1: Giao diện trước khi tải tập dữ liệu

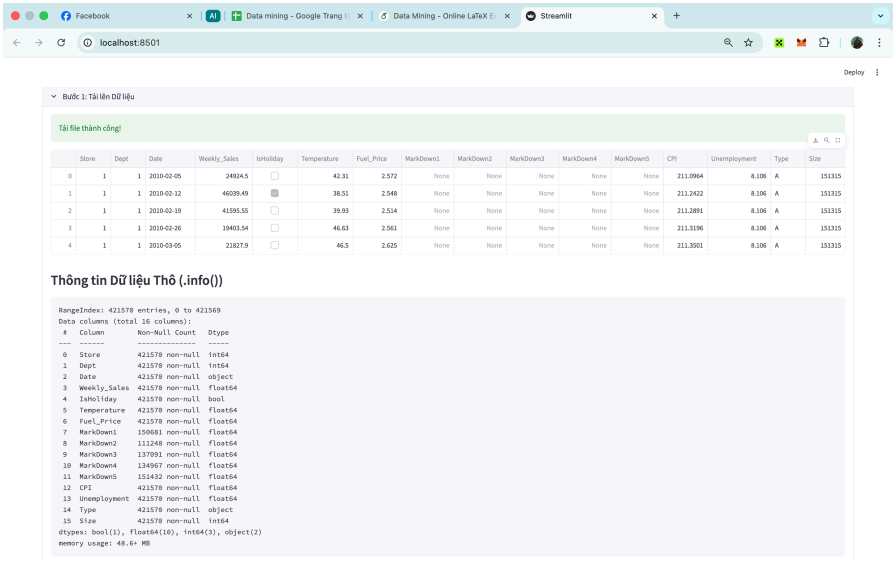

Figure 5.1.2: Giao diện sau khi tải tập dữ liệu

Sau khi người dùng tải tệp dữ liệu lên (Hình 5.1.2), ứng dụng tự động đọc nội dung và hiển thị 5 dòng đầu tiên của tập dữ liệu thô. Đồng thời, kết quả của phương thức `.info()` cũng được trình bày, giúp người dùng xác minh nhanh tính toàn vẹn và định dạng của dữ liệu.

5.2 Giao diện khám phá dữ liệu

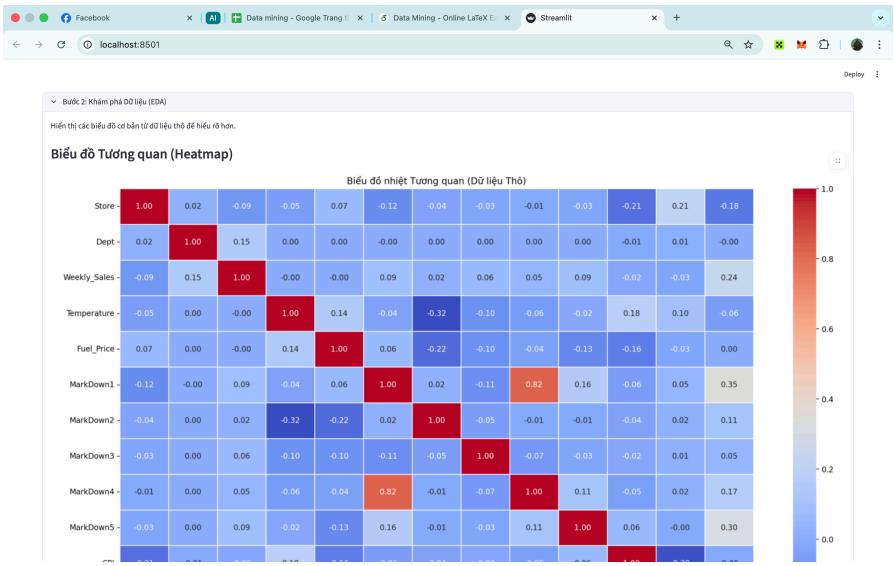

Figure 5.2.1: Phân tích và khám phá dữ liệu

Tại bước này (Hình 5.2.1), hệ thống tự động sinh ra biểu đồ Ma trận tương quan giữa các biến. Nhờ đó, người dùng có thể nhanh chóng nhận biết các đặc điểm của dữ liệu thô.

5.3 Giao diện tiền xử lý dữ liệu

Khi người dùng nhấn "*Bắt đầu Tiền xử lý*", giao diện sẽ thu gọn bước trước và mở lần lượt các mục con trong Bước 3: Xử lý giá trị thiếu, xử lý nhiễu, tạo đặc trưng và chuẩn hóa dữ liệu.

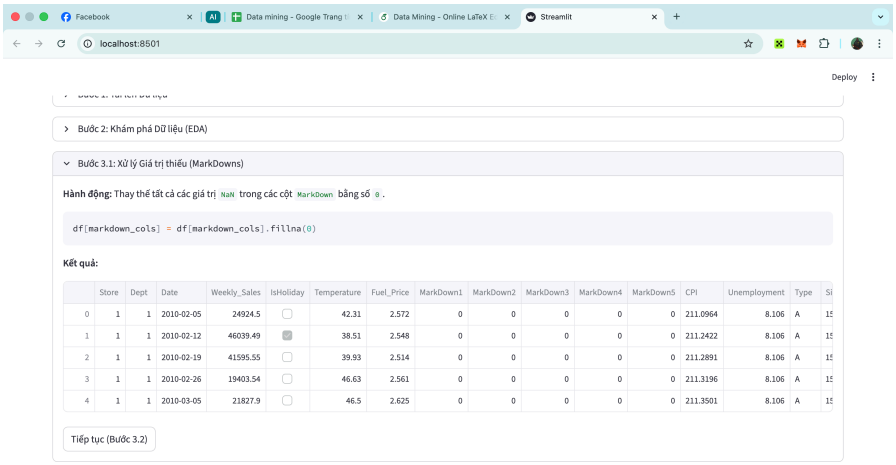

Figure 5.3.1: Xử lý dữ liệu khuyết

Hình 5.3.1 minh họa quá trình xử lý giá trị thiếu, trong đó ứng dụng áp dụng phép `fillna(0)` và hiển thị kết quả ngay sau khi thực thi.

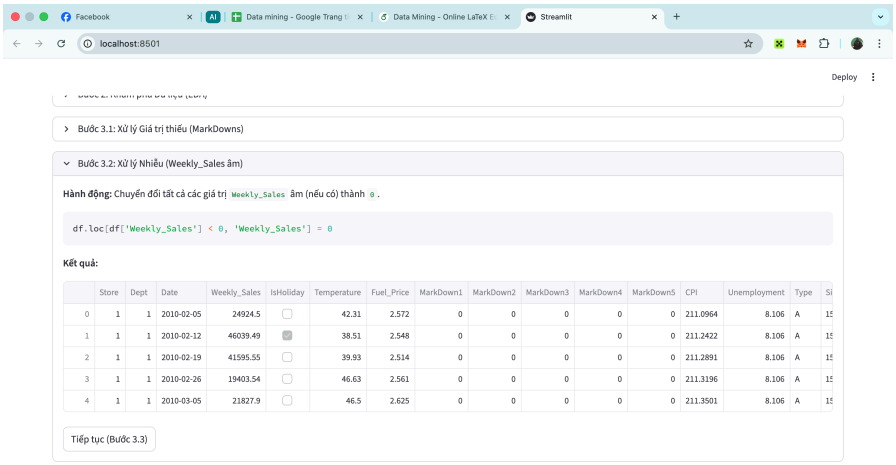

Figure 5.3.2: Xử lý dữ liệu nhiễu

Bước kế tiếp (Hình 5.3.2) minh họa việc loại bỏ hoặc làm trơn các giá trị bất thường trong dữ liệu.

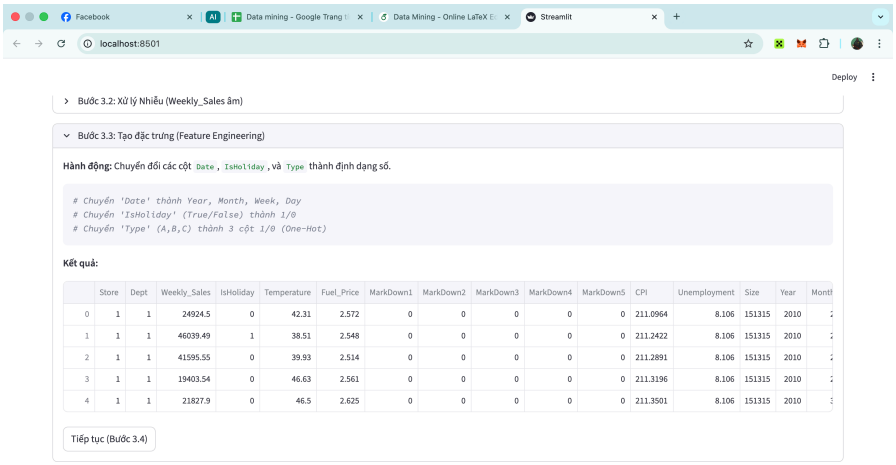

Figure 5.3.3: Tạo đặc trưng mới

Tiếp theo, mô-đun tạo đặc trưng (Hình 5.3.3) sinh thêm các biến phụ như `Month`, `WeelOfYear` hoặc `IsHoliday` để tăng khả năng học của mô hình.

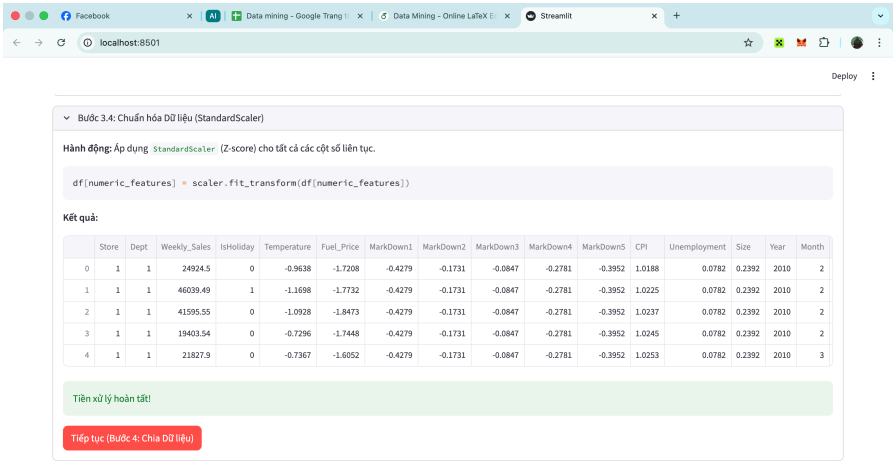

Figure 5.3.4: Chuẩn hóa dữ liệu

Cuối cùng, dữ liệu được chuẩn hóa (Hình 5.3.4) nhằm đảm bảo các đặc trưng có cùng thang đo, giúp mô hình hội tụ ổn định hơn.

5.4 Giao diện huấn luyện mô hình

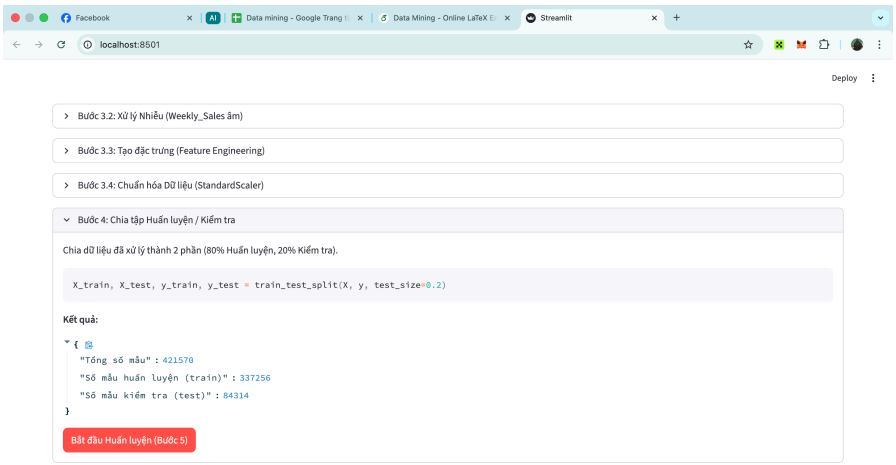

Figure 5.4.1: Chia tập dữ liệu huấn luyện và kiểm tra

Bước chuẩn bị huấn luyện (Hình 5.4.1) hiển thị thông tin về quá trình chia dữ liệu, ví dụ 80% cho huấn luyện và 20% cho kiểm tra.

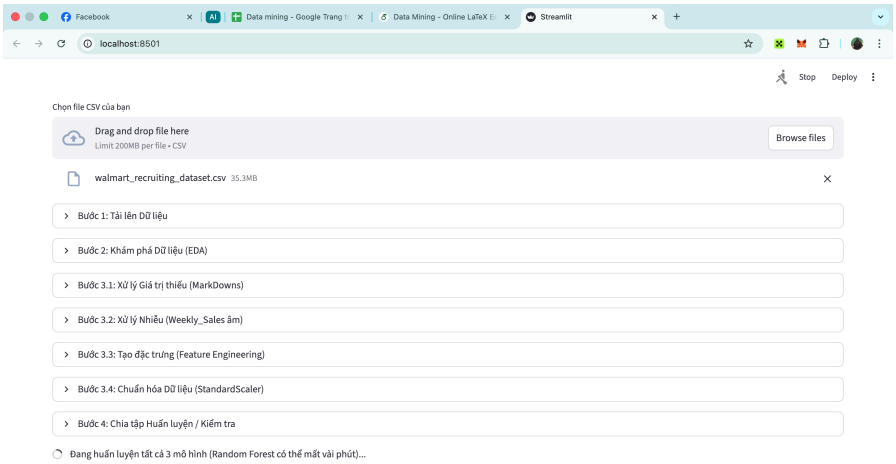

Figure 5.4.2: Giao diện trong quá trình huấn luyện

Khi người dùng nhấn "*Bắt đầu Huấn luyện*", ứng dụng hiển thị tiến trình đang được huấn luyện.

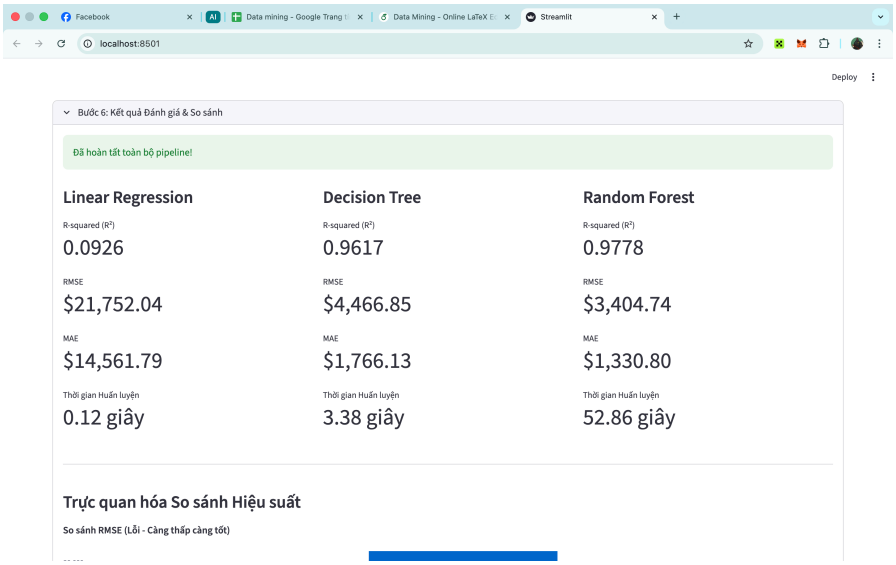

Figure 5.4.3: Hoàn tất huấn luyện mô hình

Sau khi hoàn thành, ứng dụng tự động mở mục kết quả và hiển thị bảng điều khiển (dashboard) tổng hợp các chỉ số đánh giá (Hình 5.4.3).

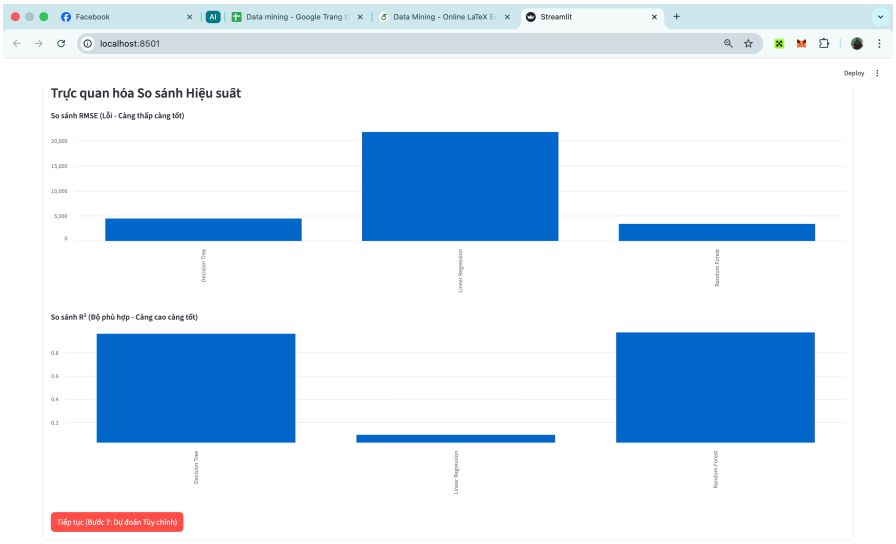

Figure 5.4.4: So sánh hiệu suất giữa các mô hình

Bảng điều khiển (Hình 5.4.4) giúp so sánh hiệu suất giữa các mô hình theo các chỉ số RMSE và R^2.

5.5 Giao diện chạy thử mô hình

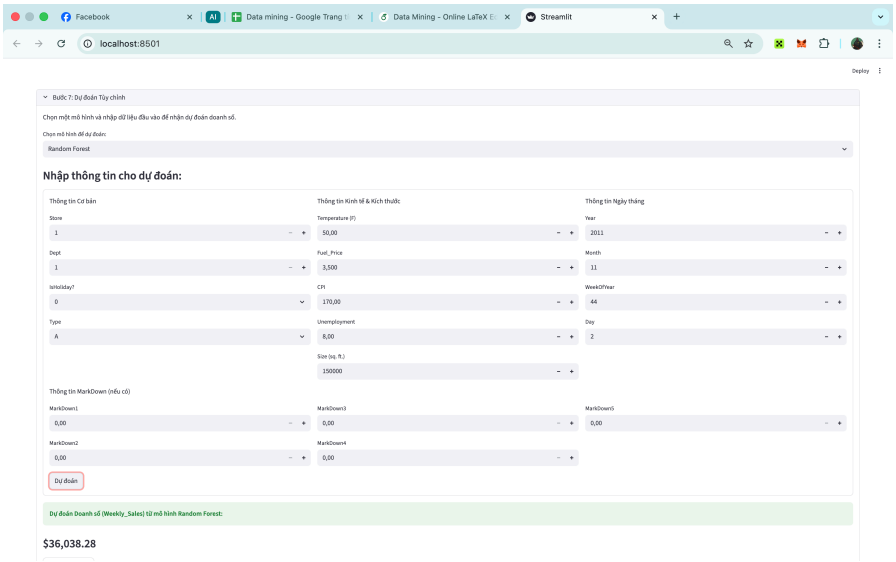

Figure 5.5.1: Giao diện dự đoán doanh thu

Người dùng có thể chọn một trong các mô hình đã huấn luyện (ví dụ: Random Forest) và nhập các tham số đầu vào thông qua biểu mẫu `st.form`. Sau khi nhấn "*Dự đoán*", ứng dụng tự động áp dụng quy trình tiền xử lý tương ứng và hiển thị kết quả doanh thu dự đoán ngay trên giao diện.

6 Phân tích và Thảo luận kết quả

Sau khi huấn luyện và đánh giá ba mô hình hồi quy trên cùng một tập dữ liệu kiểm tra (*test set*), chúng ta thu được một cái nhìn rõ ràng về hiệu suất và mức độ phù hợp của từng phương pháp.

6.1 Phân tích ý nghĩa của kết quả

Các kết quả đánh giá không chỉ giúp xác định mô hình nào hoạt động tốt nhất, mà còn củng cố các giả thuyết đã được đưa ra trong giai đoạn Khám phá Dữ liệu (EDA) và Lựa chọn Đặc trưng (Feature Selection).

Hồi quy tuyến tính (*Linear Regression*)

- **Kết quả:** $R^2 = 0.0926$, $RMSE = \$21{,}752$, $MAE = \$14{,}561$.
- **Phân tích:** Mô hình chỉ giải thích được 9.26% sự biến động của dữ liệu. Sai số trung bình quá lớn khiến mô hình gần như không có giá trị ứng dụng thực tế.
- **Nguyên nhân:** Các đặc trưng như `Fuel_Price` và `Temperature` có tương quan tuyến tính gần như bằng 0 với `Weekly_Sales`. Do đó, mô hình tuyến tính (chỉ học được mối quan hệ dạng $y = ax + b$) đã không thể nắm bắt được bản chất phi tuyến tính của dữ liệu.

Mô hình Decision Tree và Random Forest

- **Kết quả:** Decision Tree đạt $R^2 = 0.9617$, còn Random Forest đạt $R^2 = 0.9778$.
- **Phân tích:** Hai mô hình dựa trên cây đều cho kết quả vượt trội, chứng minh rằng `Weekly_Sales` là một biến có thể dự đoán được với độ chính xác rất cao.
- **Nguyên nhân:** Không giống như Linear Regression, mô hình cây có thể học các mối quan hệ phức tạp, phi tuyến tính. Các đặc trưng quan trọng nhất được xác định là các yếu tố định danh và phân loại như `Dept`, `Store`, `Size` và `Type`, thay vì các yếu tố kinh tế.

Random Forest vượt trội hơn Decision Tree

- **Kết quả:** Random Forest đạt $RMSE = \$3{,}404$, giảm 24% so với Decision Tree ($RMSE = \$4{,}466$).
- **Phân tích:** Mặc dù cả hai đều hiệu quả, Random Forest thể hiện khả năng tổng quát hóa tốt hơn.
- **Nguyên nhân:** Decision Tree đơn lẻ dễ bị *overfitting* (học vẹt). Trong khi đó, Random Forest là mô hình tổ hợp gồm 100 cây, mỗi cây được huấn luyện trên một mẫu con khác nhau của dữ liệu. Kết quả trung bình từ nhiều cây giúp giảm nhiễu và sai lệch, tăng độ ổn định của mô hình.

6.2 Hạn chế của mô hình

Mặc dù Random Forest đạt độ chính xác rất cao ($R^2 = 0.9778$), nó vẫn tồn tại một số hạn chế:

1. **Tính diễn giải (Explainability):** Linear Regression là một mô hình "hộp trắng"(*white-box*) —dễ hiểu và có thể giải thích rõ ràng ảnh hưởng của từng biến. Ngược lại, Random Forest là "hộp đen"(*black-box*), gồm hàng trăm cây phức tạp, rất khó để lý giải một dự đoán cụ thể.
2. **Thời gian huấn luyện:** Random Forest mất **52.86 giây** để huấn luyện, chậm hơn 15 lần so với Decision Tree (3.38 giây) và 440 lần so với Linear Regression (0.12 giây). Khi mở rộng lên các tập dữ liệu lớn (hàng chục triệu dòng), chi phí tính toán là vấn đề đáng kể.
3. **Khả năng dự đoán dữ liệu mới:** Mô hình được huấn luyện trên dữ liệu của 45 cửa hàng và 99 phòng ban, do đó sẽ hoạt động rất chính xác trong phạm vi này. Tuy nhiên, nếu xuất hiện cửa hàng hoặc phòng ban mới, mô hình sẽ không thể dự đoán chính xác.

6.3 Hướng phát triển

Dựa trên kết quả và các hạn chế đã nêu, có thể mở rộng và cải thiện mô hình theo các hướng sau:

- **Tinh chỉnh siêu tham số (Hyperparameter Tuning):** Sử dụng `GridSearchCV` hoặc `RandomizedSearchCV` để tối ưu các tham số như `max_depth` hoặc `min_samples_leaf`, giúp giảm thêm lỗi RMSE và tăng độ chính xác.
- **Thử nghiệm các mô hình *Ensemble* nâng cao:** Áp dụng các thuật toán như *Gradient Boosting*, *XGBoost* hoặc *LightGBM*, vốn có khả năng học mạnh mẽ hơn nhờ cơ chế huấn luyện tuần tự —mỗi cây mới học để sửa lỗi của cây trước đó.
- **Kỹ thuật tạo đặc trưng (Feature Engineering) nâng cao:**
  - *Đặc trưng dựa trên thời gian (Lag Features):* Ví dụ: `LastWeekSales`, `Avg4WeeksSales`.
  - *Đặc trưng sự kiện (Event-based Features):* Thay vì chỉ dùng `IsHoliday`, có thể tạo thêm biến `DaysUntilNextHoliday` —phản ánh hành vi mua sắm tăng trước kỳ lễ.
- **Triển khai mô hình (Deployment):**
  - Lưu mô hình `RandomForestRegressor` và `StandardScaler` bằng `joblib` hoặc `pickle`.
  - Xây dựng API với `Flask` hoặc `FastAPI` để nhận dữ liệu đầu vào (*JSON*), tải mô hình đã lưu, xử lý và trả về kết quả dự đoán theo thời gian thực.

7 Ràng buộc bổ sung

Mặc dù mô hình *Random Forest* (mô hình tốt nhất của chúng ta) đạt được hiệu suất rất cao ($R^2 = 0.9778$), vẫn còn một phần nhỏ (2.22%) phương sai của `Weekly_Sales` mà mô hình không thể giải thích được. Phần "lỗi" (error) còn lại này không chỉ là nhiễu ngẫu nhiên, mà đại diện cho các yếu tố phức tạp trong thế giới thực mà bộ dữ liệu của chúng ta không ghi lại được. Đây được gọi là các **ràng buộc ngoại sinh** (*external constraints*). Ba trong số các ràng buộc quan trọng nhất bao gồm:

7.1 Đối thủ cạnh tranh

Vấn đề: Mô hình của chúng ta chỉ được huấn luyện trên dữ liệu nội bộ của Walmart (ví dụ: `Store`, `Dept`, `Size`, `MarkDown` của Walmart). Nó hoàn toàn "mù" (*blind*) trước các hành động và sự tồn tại của các đối thủ cạnh tranh trong cùng khu vực (như Target, Costco, Kmart).

Cơ chế tác động: Hành vi mua sắm của khách hàng bị ảnh hưởng mạnh mẽ bởi một thị trường cạnh tranh.

Ví dụ (Chiến tranh giá cả): Giả sử mô hình của chúng ta dự đoán doanh số cao cho Tuần 30 tại Cửa hàng 5. Tuy nhiên, cùng tuần đó, một cửa hàng Target ở bên kia đường tung ra chương trình "giảm giá sốc" lớn. Kết quả: Một lượng đáng kể khách hàng chuyển sang mua hàng của Target, khiến doanh số thực tế của Walmart thấp hơn nhiều so với dự đoán.

Tác động đến mô hình: Mô hình không "nhìn thấy" được chương trình giảm giá của Target, nên lỗi này là một *lỗi không thể giải thích được* (unexplained error) do thiếu đặc trưng `Competitor_Activity`.

7.2 Sự kiện bất khả kháng (Thiên nga đen)

Vấn đề: Đây là các thảm họa hiếm gặp, có tác động cực lớn và nằm ngoài mọi dự đoán thông thường (ví dụ: đại dịch, thiên tai quy mô lớn, khủng hoảng tài chính).

Cơ chế tác động: Các sự kiện "Thiên nga đen" phá vỡ hoàn toàn các quy luật lịch sử mà mô hình đã học.

Ví dụ (Đại dịch COVID-19): Mô hình được huấn luyện trên dữ liệu 2010–2012, một giai đoạn "bình thường". Nó không thể dự đoán được các hành vi phi logic do đại dịch gây ra:

- **Hoảng loạn mua sắm (Panic Buying):** Mô hình dự đoán doanh số tháng 3 (tháng thấp điểm) là \$10,000, nhưng doanh số thực tế vọt lên \$150,000 do người dân tích trữ hàng hóa.
- **Lệnh phong tỏa (Lockdowns):** Mô hình dự đoán doanh số Giáng sinh là \$100,000, nhưng doanh số thực tế là 0 vì cửa hàng buộc phải đóng cửa.

Tác động đến mô hình: Các sự kiện này khiến các dự đoán dựa trên lịch sử trở nên vô nghĩa —vì quan hệ giữa đặc trưng và mục tiêu không còn giữ nguyên.

7.3 Thay đổi về luật/chính sách

Vấn đề: Các mô hình học máy giả định rằng các quy luật của quá khứ sẽ tiếp tục đúng trong tương lai. Tuy nhiên, các thay đổi về luật pháp hoặc chính sách của chính phủ có thể thay đổi các "luật chơi" chỉ sau một đêm.

Cơ chế tác động: Các thay đổi này tạo ra các *điểm gãy cấu trúc* (*structural breaks*) trong dữ liệu.

Ví dụ (Quy định giờ mở cửa): Mô hình được huấn luyện trên dữ liệu 2010–2012, khi cửa hàng mở cửa 7 ngày/tuần. Nó học rằng "Chủ nhật là ngày có doanh số tốt". Nhưng năm 2013, một luật mới được ban hành, cấm các cửa hàng lớn mở cửa vào Chủ nhật. Kết quả: Mô hình tiếp tục dự đoán doanh số cao cho Chủ nhật, nhưng thực tế là $0.

Tác động đến mô hình: Sự thay đổi đột ngột về luật (như tăng thuế, thay đổi lương tối thiểu, quy định an toàn sản phẩm) khiến mô hình trở nên "lỗi thời". Nó cần được *huấn luyện lại* (retrain) với dữ liệu mới để học được "quy tắc bình thường mới".

8 Tài liệu tham khảo

1. Diebold, F. X. (2015). *Elements of Forecasting (Advanced Version)*. Truy cập tại: https://www.sas.upenn.edu/~fdiebold/NoHesitations/BookAdvanced.pdf

2. IBM, *Exploratory Data Analysis (EDA)*. Truy cập tại: https://www.ibm.com/think/topics/exploratory-data-analysis

3. Viblo, *Data Visualization với Seaborn*. Truy cập tại: https://viblo.asia/p/data-visualization-voi-seaborn-oOVlYP9vZ8W

4. Matplotlib, *Gallery Examples*. Truy cập tại: https://matplotlib.org/stable/gallery

5. W3Schools, *Python Pandas Tutorial*. Truy cập tại: https://www.w3schools.com/python/pandas/default.asp

6. Scikit-learn, *Machine Learning Library for Python*. Truy cập tại: https://scikit-learn.org/stable/

7. Streamlit, *Official Documentation*. Truy cập tại: https://docs.streamlit.io/

8. Arun Pandey (2021), *Regression Algorithms Explained*. Truy cập tại: https://arunp77.medium.com/regression-algorithms-29f112797724

9. iHOCLAPTRINH, *Machine Learning - Thuật toán StandardScaler*. Truy cập tại: https://ihoclaptrinh.com/thuan-toan-machine-learning-standardscaler

10. Scikit-learn, *Model Evaluation Documentation*. Truy cập tại: https://scikit-learn.org/stable/modules/model_evaluation.html

11. Scikit-learn, *Linear Regression Documentation*. Truy cập tại: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html

12. Scikit-learn, *Decision Tree Classifier Documentation*. Truy cập tại: https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html

13. Scikit-learn, *Random Forest Classifier Documentation*. Truy cập tại: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

14. Lex Jansen (2007), *Data Capping and Flooring Approaches*. Truy cập tại: https://www.lexjansen.com/nesug/nesug07/sa/sa16.pdf (xem mục 2.1)

15. Scikit-learn, *Preprocessing API*. Truy cập tại: https://scikit-learn.org/stable/api/sklearn.preprocessing.html

16. Scikit-learn, *Feature Selection Documentation*. Truy cập tại: https://scikit-learn.org/stable/modules/feature_selection.html

17. Stock, J. H., & Watson, M. W. (2011). *Introduction to Econometrics (3rd Edition)*. Pearson Education. (Tham khảo phần "Linear Regression with Multiple Regressors", Chương 6)

18. Taleb, N. N. (2007). *The Black Swan: The Impact of the Highly Improbable*. Random House.

19. Angrist, J. D., & Pischke, J. S. (2009). *Mostly Harmless Econometrics: An Empiricist's Companion*. Princeton University Press. (Tham khảo phần "Regression Discontinuity Design", Chương 6)

9 Phụ lục

9.1 Mã nguồn

Toàn bộ mã nguồn được lưu ở GitHub:
https://github.com/khoale2k4/data-mining_BTL.git

9.2 Nguồn dữ liệu

Bộ dữ liệu được sử dụng trong dự án được lấy từ Kaggle:
https://www.kaggle.com/datasets/aslanahmedov/walmart-sales-forecast