

**ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN**



**BÁO CÁO CUỐI KỲ
MÔN HỌC : MỘT SỐ CHỦ ĐỀ
TRONG MÔ HÌNH HÓA VÀ PHÂN TÍCH DỮ LIỆU**

**PHÂN TÍCH DOANH SỐ BÁN HÀNG
TẠI SIÊU THỊ BIG MART**

Giảng viên: TS. Phạm Đình Tùng
GV. Nguyễn Trung Đức

Sinh viên: Nguyễn Phan Anh 20002030
Vương Thùy Dương 20002042
Vũ Mạnh Đức 20002046

Lớp: K65A5 Khoa học dữ liệu

Hà Nội, 05-2024

Lời Cảm Ơn

Nhóm tác giả xin chân thành cảm ơn thầy Phạm Đình Tùng và thầy Nguyễn Trung Đức đã tận tâm hướng dẫn môn học 'Một số chủ đề trong mô hình hóa và phân tích dữ liệu' và hỗ trợ nhóm trong quá trình thực hiện đề tài của môn học này. Các thầy không chỉ giúp nhóm hiểu rõ hơn về lý thuyết mà còn khám phá ra những ứng dụng thực tế quan trọng của việc phân tích dữ liệu và xây dựng mô hình trong thế giới ngày càng phát triển.

Nhờ có sự hướng dẫn của các thầy, chúng tôi đã có cơ hội áp dụng kiến thức vào thực tế, từ đó phát triển kỹ năng và sự hiểu biết vững về lĩnh vực này.

Một lần nữa, chúng tôi chân thành cảm ơn thầy Phạm Đình Tùng và thầy Nguyễn Trung Đức vì tất cả những kiến thức thú vị mà nhóm đã nhận được.

Trân trọng,
Nguyễn Phan Anh
Vương Thùy Dương
Vũ Mạnh Đức

Mục lục

Chương 1	Giới thiệu đề tài	4
Chương 2	Phương pháp nghiên cứu	5
2.1.	Bộ dữ liệu: BigMart Sales	5
2.2.	Tổng quan hệ thống công việc	6
2.3.	Mô hình hồi quy tuyến tính	7
2.4.	Random Forest	8
Chương 3	Khai phá dữ liệu	10
3.1.	Phân tích đơn biến	10
3.2.	Phân tích hai biến	13
Chương 4	Xử lý dữ liệu.	16
4.1.	Xử lý dữ liệu không phù hợp	16
4.2.	Xử lý dữ liệu bị thiếu	17
4.3.	Thêm các biến mới vào dữ liệu	19
4.4.	Trực quan dữ liệu	21
Chương 5	Xây dựng mô hình	23
5.1.	Mô hình hồi quy tuyến tính	23
5.1.1.	Quá trình xây dựng và cải tiến mô hình	23

5.1.2. Đánh giá mô hình	25
5.1.3. Phân tích ANCOVA	28
5.2. Mô hình học máy Random Forest	32
Chương 6 Thảo luận vấn đề.	34
6.1. Kết luận về mô hình hóa	34
6.2. So sánh với nghiên cứu khác	34
6.3. Định hướng	35

Chương 1

Giới thiệu đề tài

Trong bối cảnh kinh tế hiện đại, sự cạnh tranh gay gắt trong ngành bán lẻ đòi hỏi các doanh nghiệp phải không ngừng cải thiện quy trình quản lý và dự báo doanh số bán hàng. BigMart, với mạng lưới cửa hàng rộng khắp thế giới, đang đứng trước thách thức này và không ngừng tìm kiếm các phương pháp tiên tiến để tối ưu hóa hoạt động kinh doanh của mình. Báo cáo này tập trung vào việc phân tích và dự báo doanh số bán hàng của BigMart thông qua việc sử dụng các kỹ thuật khoa học dữ liệu và máy học.

Việc dự báo chính xác nhu cầu khách hàng không chỉ giúp BigMart nắm bắt được xu hướng thị trường mà còn cải thiện hiệu quả quản lý hàng tồn kho, định giá sản phẩm hợp lý, và tối đa hóa lợi nhuận. Trong thời đại công nghệ thông tin phát triển mạnh mẽ, việc thu thập và phân tích dữ liệu trở nên ngày càng quan trọng, mang lại lợi thế cạnh tranh đáng kể cho các doanh nghiệp.

Nghiên cứu này sử dụng dữ liệu từ BigMart để xây dựng mô hình dự báo doanh số bán hàng, đồng thời khám phá những yếu tố ảnh hưởng đến doanh số của các sản phẩm. Thông qua việc áp dụng các phương pháp phân tích dữ liệu tiên tiến, báo cáo sẽ trình bày những kết quả đạt được với độ chính xác cao, từ đó đưa ra các khuyến nghị giúp BigMart và các doanh nghiệp bán lẻ khác cải thiện chiến lược kinh doanh.

Chương 2

Phương pháp nghiên cứu

2.1. Bộ dữ liệu: BigMart Sales

Bộ dữ liệu được sử dụng trong nghiên cứu này là kết quả doanh số bán hàng của BigMart năm 2013, bao gồm tổng cộng 12 thuộc tính. Trong đó, thuộc tính Item_Outlet_Sales là biến mục tiêu, các thuộc tính còn lại là các biến độc lập. Bộ dữ liệu này được thu thập từ nền tảng Kaggle và chứa thông tin chi tiết về BigMart. Bộ dữ liệu bao gồm doanh số bán hàng của 1559 sản phẩm tại 10 cửa hàng khác nhau trong năm 2013 và cung cấp thông tin chi tiết về sản phẩm, cửa hàng bán sản phẩm và giá trị doanh số bán hàng.

	Item_Identifier	Item_Weight	Item_Fat_Content	Item_Visibility	Item_Type	Item_MRP	Outlet_Identifier	Outlet_Established
0	FDA15	9.300	Low Fat	0.016047	Dairy	249.8092	OUT049	1999
1	DRC01	5.920	Regular	0.019278	Soft Drinks	48.2692	OUT018	2009
2	FDN15	17.500	Low Fat	0.016760	Meat	141.6180	OUT049	1999
3	FDX07	19.200	Regular	0.000000	Fruits and Vegetables	182.0950	OUT010	1998
4	NCD19	8.930	Low Fat	0.000000	Household	53.8614	OUT013	1987
...
8518	FDF22	6.865	Low Fat	0.056783	Snack Foods	214.5218	OUT013	1987
8519	FDS36	8.380	Regular	0.046982	Baking Goods	108.1570	OUT045	2002
8520	NCJ29	10.600	Low Fat	0.035186	Health and Hygiene	85.1224	OUT035	2004
8521	FDN46	7.210	Regular	0.145221	Snack Foods	103.1332	OUT018	2009
8522	DRG01	14.800	Low Fat	0.044878	Soft Drinks	75.4670	OUT046	1997

Hình 2.1: Bigmart dataset

Cấu trúc bộ dữ liệu

- **Item_Identifier:** ID sản phẩm, duy nhất cho mỗi sản phẩm. Dữ liệu dạng ký tự và số, dài 5 ký tự (VD: FDN15).
- **Item_Weight:** Khối lượng của sản phẩm (Dữ liệu dạng số thập phân).
- **Item_Fat_Content:** Lượng chất béo trong sản phẩm với hai giá trị ['Low Fat',

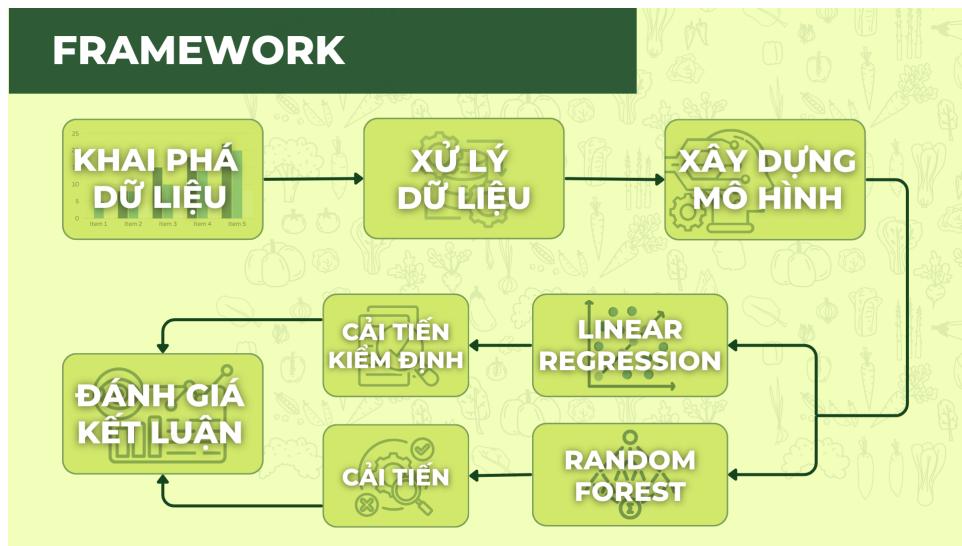
'Regular']. Tuy nhiên, trong dữ liệu, 'Low Fat' cũng có thể viết là 'LF', và 'Regular' cũng có thể viết là 'reg' (Dữ liệu dạng ký tự).

- **Item_Visibility:** Tỷ lệ phần trăm diện tích hiển thị của tất cả các sản phẩm cùng loại trong một cửa hàng được phân bổ cho từng sản phẩm cụ thể (Dữ liệu dạng số thập phân).
- **Item_Type:** Loại danh mục sản phẩm, bao gồm 16 loại sản phẩm khác nhau như Dairy, Soft Drinks, Meat, Fruits and Vegetables, v.v. (Dữ liệu dạng ký tự).
- **Item_MRP:** Giá bán lẻ cao nhất của sản phẩm (Dữ liệu dạng số thập phân).
- **Outlet_Identifier:** ID của cửa hàng bán sản phẩm, là một chuỗi gồm 6 ký tự và số (VD: OUT049). Có tất cả 10 giá trị khác nhau tương ứng với 10 cửa hàng (Dữ liệu dạng ký tự và số).
- **Outlet_Establishment_Year:** Năm thành lập cửa hàng bán sản phẩm (Dữ liệu dạng số nguyên).
- **Outlet_Size:** Kích thước của cửa hàng theo diện tích mặt đất, gồm 3 loại: High, Medium, Small (Dữ liệu dạng chữ).
- **Outlet_Location_Type:** Kích thước của thành phố nơi cửa hàng bán sản phẩm được đặt, với 3 giá trị: Tier 1, Tier 2, Tier 3 (Dữ liệu dạng chữ).
- **Outlet_Type:** Loại cửa hàng bán sản phẩm là cửa hàng tạp hóa hay siêu thị, chia thành 4 loại: Supermarket Type1, Supermarket Type2, Supermarket Type3, Grocery Store (Dữ liệu dạng chữ).
- **Item_Outlet_Sales:** Doanh số bán sản phẩm tại một cửa hàng cụ thể, là biến đầu ra của mô hình dự báo (Dữ liệu dạng số).

2.2. Tổng quan hệ thống công việc

Quá trình nghiên cứu và phân tích dữ liệu được thực hiện theo các bước chi tiết như sau: Đầu tiên, từ bộ dữ liệu thô, nhóm tác giả tiến hành khám phá dữ liệu để hiểu rõ hơn về cấu trúc, tính chất và phân bố của dữ liệu.

Sau khi khám phá dữ liệu, sẽ tiến hành xử lý dữ liệu để chuẩn bị cho việc xây dựng mô hình, bao gồm xử lý các giá trị không phù hợp với dữ liệu, những giá trị còn thiếu và tạo thêm các biến mới cho bộ dữ liệu.



Hình 2.2: Hệ thống công việc

Bước tiếp theo đi xây dựng mô hình hồi quy tuyến tính để dự đoán doanh số bán hàng. Quá trình này bao gồm xây dựng mô hình hồi quy tuyến tính cơ bản, áp dụng các phương pháp cải tiến như Step-wise Selection để chọn ra các biến độc lập quan trọng nhất, và biến đổi hàm log cho biến mục tiêu hoặc các biến độc lập để cải thiện tính tuyến tính của mô hình. Cùng với đó thực hiện các kiểm định giả thuyết với mô hình hồi quy tuyến tính để đánh giá tính phù hợp và độ tin cậy của mô hình.

Ngoài ra, nhóm áp dụng mô hình học máy Random Forest để so sánh với mô hình hóa tuyến tính ở trên. Random Forest là một phương pháp mạnh mẽ và linh hoạt, có khả năng xử lý tốt các dữ liệu phi tuyến tính và tương tác giữa các biến.

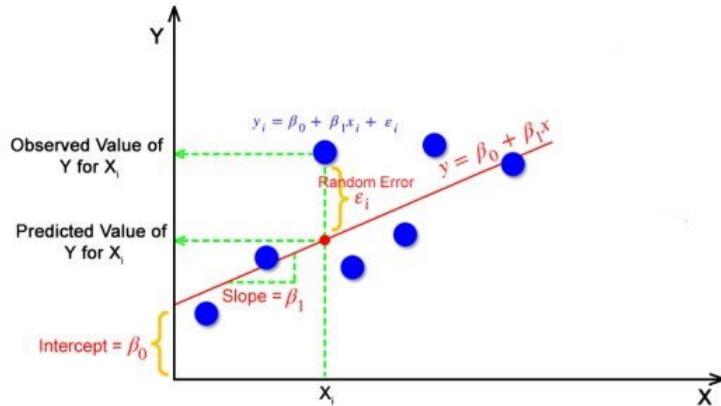
Cuối cùng, tổng kết và đánh giá hiệu quả của hai mô hình (hồi quy tuyến tính và Random Forest). Quá trình này bao gồm so sánh độ chính xác và hiệu quả của hai mô hình, phân tích ưu nhược điểm của từng mô hình, và đưa ra kết luận cùng khuyến nghị dựa trên kết quả phân tích.

2.3. Mô hình hồi quy tuyến tính

Hồi quy tuyến tính (linear regression) là việc giải thích sự thay đổi trong giá trị của một biến phụ thuộc (dependent variable) thông qua sự thay đổi trong giá trị của một biến độc lập (independent variable). Vì vậy, hồi quy tuyến tính là một công cụ thường xuyên được sử dụng trong nghiên cứu kinh tế nói chung và trong phân tích tài chính nói riêng, khi muốn kiểm tra mối liên hệ giữa hai biến số nhất định.

Khi mối quan hệ tuyến tính giữa hai biến là có ý nghĩa, hồi quy tuyến tính cung cấp một mô hình đơn giản để dự báo giá trị của một biến gọi là biến phụ thuộc (depen-

dent variable), dựa trên giá trị của biến thứ hai, được gọi là biến độc lập (independent variable).



Hình 2.3: Mô hình hồi quy tuyến tính

2.4. Random Forest

Random Forest là một thuật toán học máy dựa trên tập hợp (ensemble) của cây quyết định (decision trees). Nó kết hợp các cây quyết định độc lập để tạo ra một mô hình dự đoán mạnh mẽ và ổn định.

Random Forest sử dụng kỹ thuật Bagging (Bootstrap Aggregating) là một phương pháp kết hợp (ensemble) được sử dụng trong Machine Learning để cải thiện hiệu suất của một mô hình dự đoán. Ý tưởng chính của Bagging là tạo ra nhiều mô hình dự đoán độc lập nhau trên các mẫu dữ liệu được lấy ngẫu nhiên từ tập huấn luyện ban đầu.

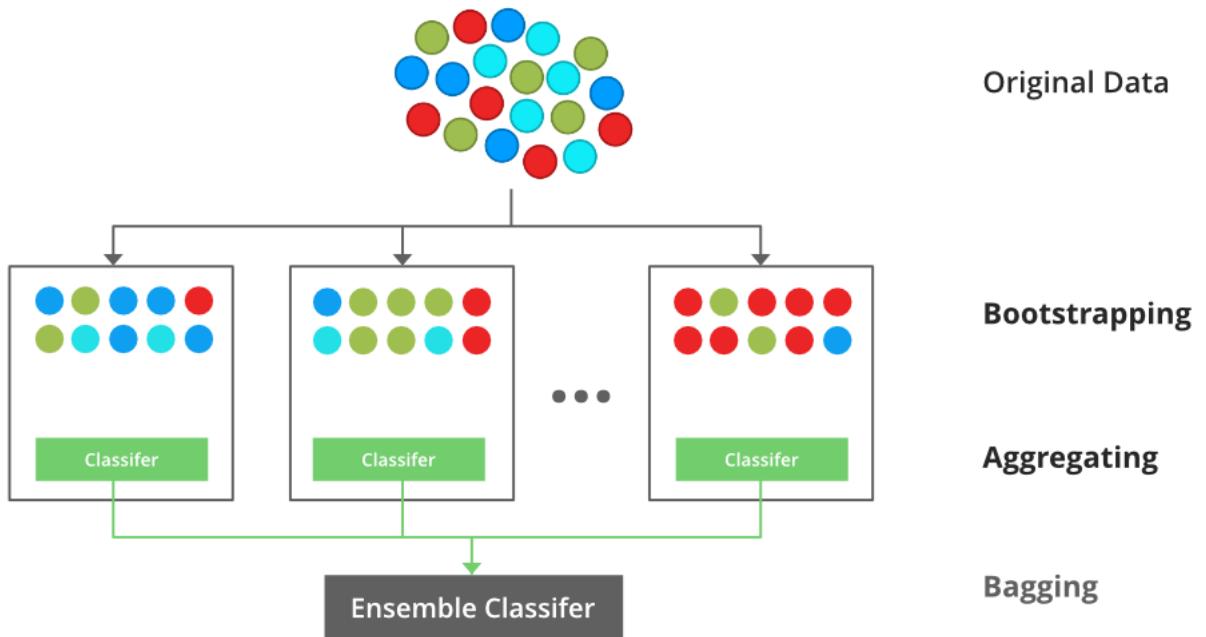
Thuật toán Random Forest

Bước 1: Tạo ngẫu nhiên các tập con (bootstrap samples)

Random Forest sử dụng phương pháp Bootstrap để tạo ra các tập con từ tập dữ liệu huấn luyện ban đầu. Mỗi tập con có số lượng mẫu bằng với số lượng mẫu trong tập dữ liệu gốc, nhưng một số mẫu có thể xuất hiện nhiều lần trong tập con và một số mẫu có thể không xuất hiện.

Bước 2: Xây dựng cây quyết định

Với mỗi tập con, một cây quyết định được xây dựng. Cây quyết định được xây dựng bằng cách chia dữ liệu dựa trên các thuộc tính và giá trị của chúng để tạo ra các quy tắc quyết định.



Hình 2.4: Sơ đồ kỹ thuật bagging

Bước 3: Kết hợp các cây quyết định

Sau khi xây dựng một số cây quyết định, Random Forest kết hợp chúng bằng cách sử dụng một quy tắc đa số hoặc trung bình dự đoán của các cây thành viên. Kết quả là một dự đoán tổng quát và ổn định hơn.

Bước 4: Dự đoán

Sau khi huấn luyện, Random Forest có thể được sử dụng để dự đoán nhãn hoặc giá trị đầu ra cho dữ liệu mới. Dự đoán được tính bằng cách áp dụng quy tắc của các cây thành viên lên dữ liệu mới và sử dụng quy tắc kết hợp để đưa ra kết quả cuối cùng.

Một số đặc điểm và ưu điểm của thuật toán Random Forest bao gồm:

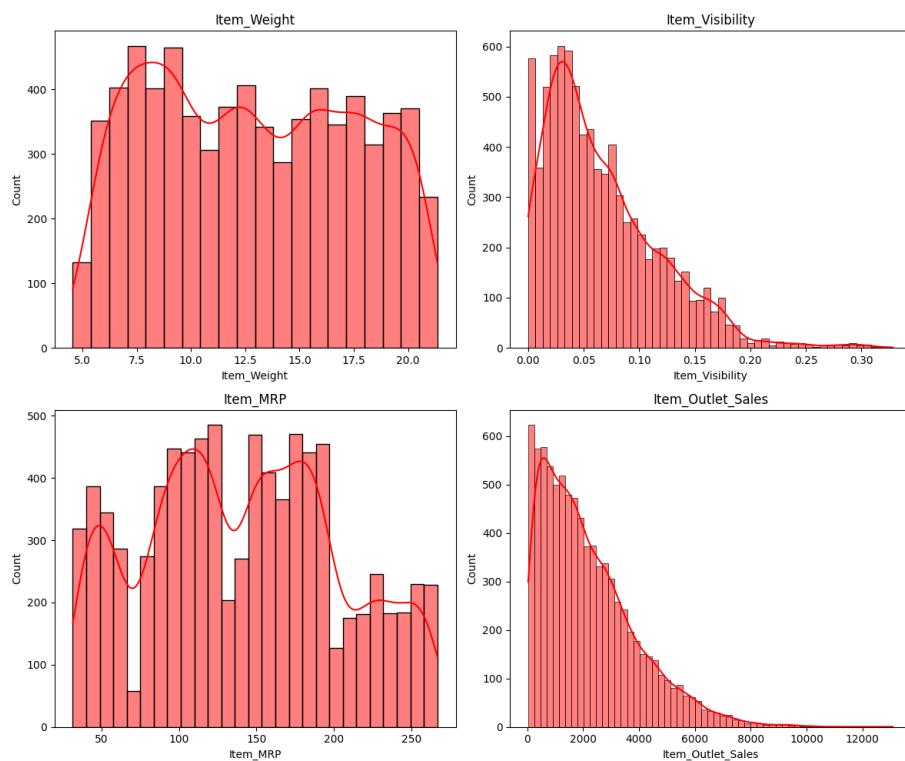
- Khả năng làm việc với dữ liệu lớn và nhiều thuộc tính.
- Khả năng xử lý dữ liệu bị thiếu (missing values) và dữ liệu có nhiễu (noisy data).
- Khả năng xếp hạng quan trọng (feature importance) của các thuộc tính.
- Tính ổn định và tránh overfitting (quá khớp) so với một cây quyết định đơn lẻ.
- Có thể được sử dụng cho cả bài toán phân loại và hồi quy.

Chương 3

Khai phá dữ liệu

3.1. Phân tích đơn biến

Phân tích các biến định lượng



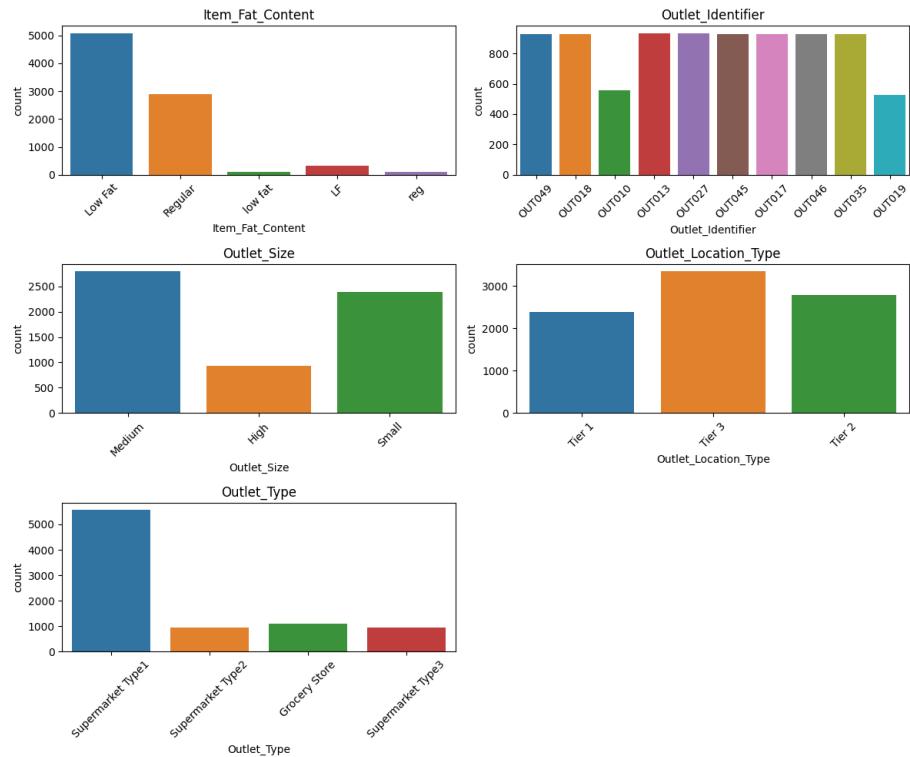
Hình 3.1: Biểu đồ phân phối các biến định lượng

Từ biểu đồ phân phối các biến định lượng, có thể rút ra một số nhận xét sau:

- **Item_Weight:** Biến này có phân phối khá đồng đều và không có dấu hiệu của sự lệch (skewness). Điều này cho thấy trọng lượng của các sản phẩm không có sự biến động lớn và phân phối khá đều, không có xu hướng tập trung ở một phần của phạm vi giá trị.

- **Item_Visibility:** Biến này có phân phối nghiêng về bên trái, có nghĩa là có một số lượng lớn các mẫu có giá trị thấp, thậm chí là bằng 0.
- **Item_MRP:** Biến này có phân phối đa cực (multimodal), với bốn điểm cực đỉnh (modes) khác nhau. Điều này cho thấy có sự phân biệt rõ rệt trong việc định giá sản phẩm, và có thể có một số nhóm sản phẩm có giá cố định hoặc mức giá thị trường nhất định.

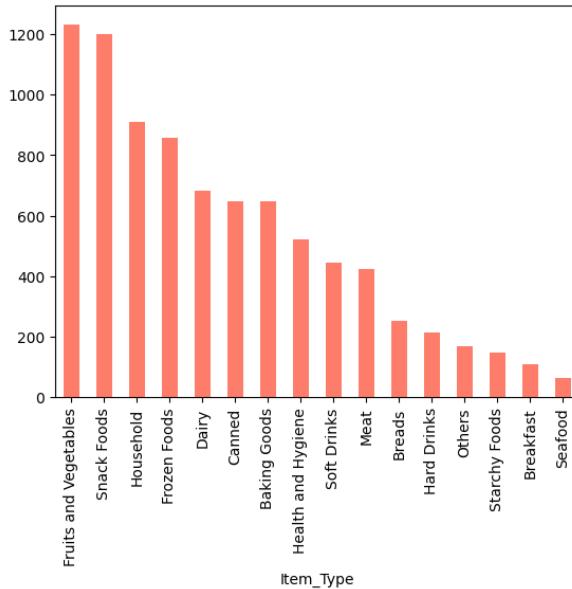
Phân tích các biến định tính



Hình 3.2: Biểu đồ thông kê các biến định tính

- **Item_Fat_Content** Thông qua biểu đồ ghi nhận sản phẩm có 2 loại chất béo là Low-fat và Regular, đồng nghĩa với việc có thể tiến hành gộp các giá trị "LF", "low fat" và "Low Fat" thành một danh mục, cũng như gộp "reg" và "Regular" lại thành một danh mục duy nhất.
- **Outlet Size** Đa số các cửa hàng trong dữ liệu có kích thước trung bình (medium), cho thấy mô hình kinh doanh này phổ biến trong hệ thống cửa hàng của BigMart.
- **Outlet_Location_Type** và **Outlet_Type** Đa số các cửa hàng được thiết lập ở các thành phố Tier 3 và các cửa hàng siêu thị loại 1 (Supermarket Type 1), cho thấy BigMart chủ yếu tập trung vào các khu vực thành phố và loại cửa hàng này.

- **Outlet_Identifier** Ngoại trừ Cửa hàng OUT010 và OUT019, có thể thấy các cửa hàng còn lại có nhiều sản phẩm đáng kể được bán ra.

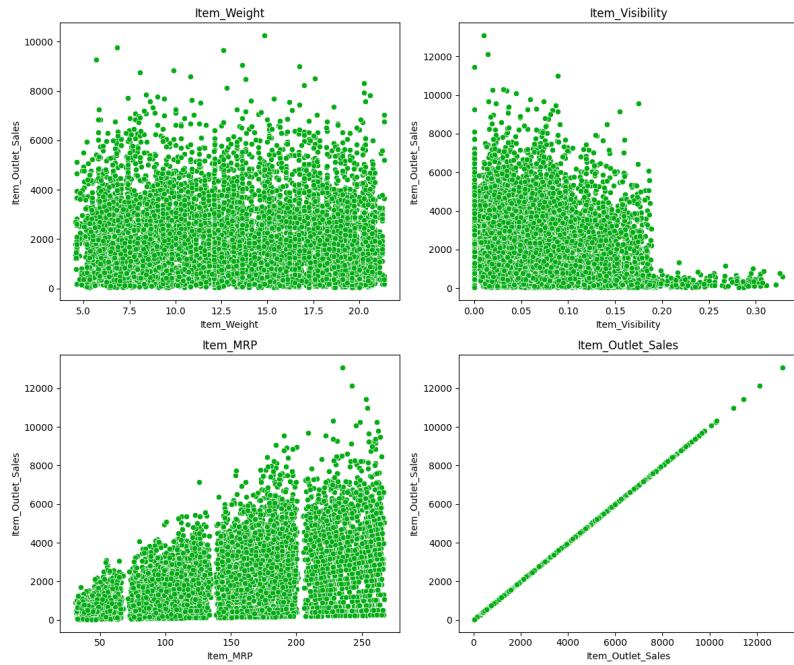


Hình 3.3: Thống kê các loại mặt hàng

- **Item_Type** Thông kê cho thấy top 5 loại sản phẩm bán chạy nhất là Fruits and Veggies, Dairy, Snacks, Frozen và Household. Thông tin này cung cấp cho BigMart một chiến lược để xử lý hàng tồn kho bằng cách tạo ra các gói combo và khuyến mãi dựa trên các sản phẩm bán chạy nhất.

3.2. Phân tích hai biến

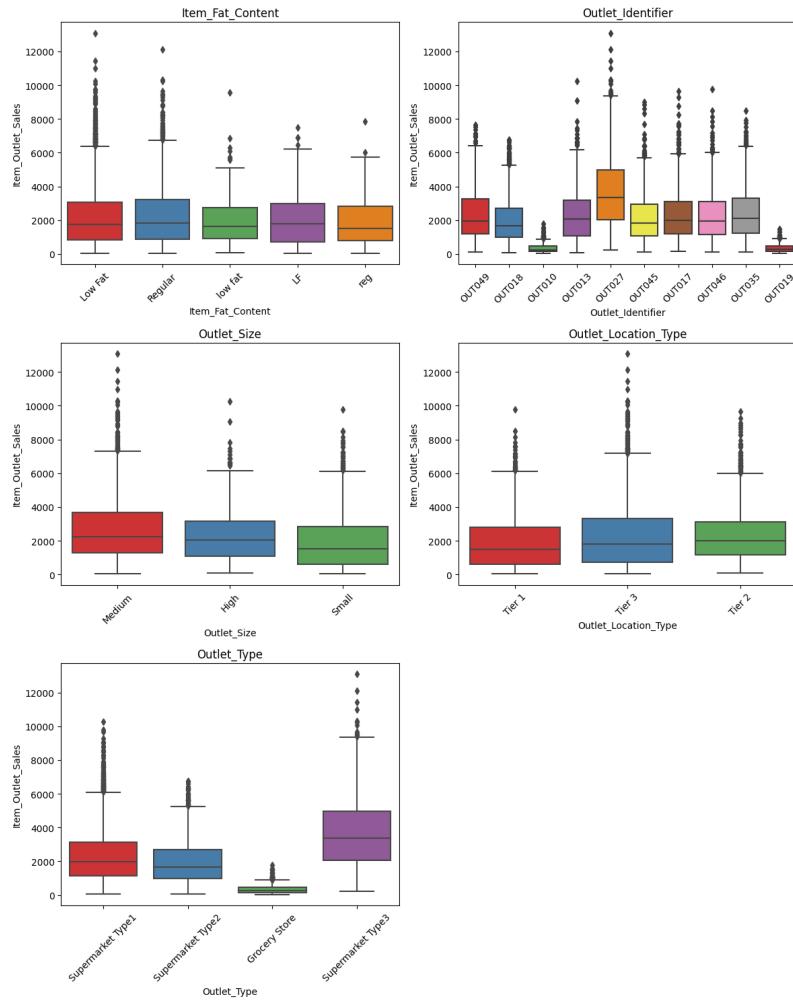
Phân tích các biến định lượng so với Item_Outlet_Sales



Hình 3.4: Biểu đồ phân phối các biến định lượng so với Item_Outlet_Sales

- **Item_Outlet_Sales và Item_Weight:** Item_Outlet_Sales được phân bố đều khắp phạm vi Item_Weight mà không có bất kỳ mẫu nào rõ ràng.
- **Item_Outlet_Sales và Item_Visibility:** Có một chuỗi điểm tại giá trị Item_Visibility bằng 0.0, điều này có vẻ vô lý vì độ hiển thị của sản phẩm không thể hoàn toàn bằng 0. Điều này cần được xử lý ở giai đoạn tiếp theo của phân tích.
- **Item_Outlet_Sales và Item_MRP:** Item_MRP so với Item_Outlet_Sales cho thấy có 4 phân cụm có thể chuyển đổi thành biến định tính để tạo một biến mới.

Phân tích các biến định tính so với Item_Outlet_Sales



Hình 3.5: Biểu đồ thông kê các biến định tính so với Item_Outlet_Sales

- **Item_Outlet_Sales và Item_Fat_Content:** Các sản phẩm có chất béo thấp (Low Fat) có doanh số bán hàng khá cao, đây có thể là do nhu cầu của thị trường đối với các sản phẩm lành mạnh hoặc do chiến lược tiếp thị và quảng cáo của BigMart.
- **Item_Outlet_Sales so với Outlet_Location_Type và Outlet_Type:** Các thành phố thuộc Tier 3, theo sau bởi loại cửa hàng Supermarket Type3, có doanh số bán hàng cao nhất. Điều này có thể phản ánh nhu cầu mạnh mẽ từ phần đông dân số ở các thành phố lớn hoặc mức độ cạnh tranh ít hơn trong các khu vực này.
- **Các cửa hàng tạo doanh thu lớn nhất:** Các cửa hàng OUT027 và OUT013 là những cửa hàng tạo ra doanh thu cao nhất, có thể do vị trí thuận lợi, kích thước cửa hàng hoặc chiến lược kinh doanh hiệu quả.
- **Các cửa hàng hoạt động kém nhất:** Các cửa hàng OUT013 và OUT010, theo sau bởi Grocery Store, là những cửa hàng hoạt động kém nhất. Điều này có thể do

nhiều lý do như vị trí không thuận lợi, kích thước cửa hàng nhỏ, hoặc chiến lược kinh doanh không hiệu quả.

Từ các phân tích trên, ta có thể đánh giá mô hình kinh doanh của các cửa hàng như OUT027 khá tốt, có thể được sử dụng làm mẫu cho các cửa hàng mới, giúp tiết kiệm chi phí và tối ưu hóa vị trí và kích thước cửa hàng một cách hiệu quả.

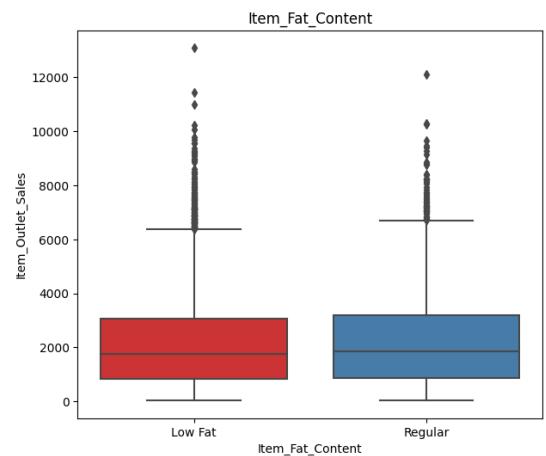
Chương 4

Xử lý dữ liệu

4.1. Xử lý dữ liệu không phù hợp

Item_Fat_Content

```
bigmart_train['Item_Fat_Content'].value_counts()  
  
Item_Fat_Content  
Low Fat      5089  
Regular     2889  
LF          316  
reg          117  
low fat      112  
Name: count, dtype: int64
```



Hình 4.1: Item_Fat_Content ban đầu

Hình 4.2: Item_Fat_Content được xử lý

Item_Fat_Content thông kê thấy có 5 giá trị, tuy nhiên trong đó: 'Low Fat', 'LF' và 'low fat' ta sẽ đưa về thành 'Low Fat' (Sản phẩm có chất béo thấp). Còn 'Regular' và 'reg' sẽ thành 'Regular' (Sản phẩm có lượng chất béo thường). Như vậy ta sẽ chỉ có 2 giá trị cho biến Item_Fat_Content

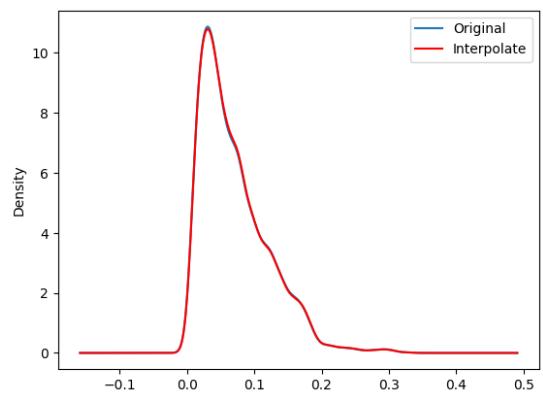
Item_Visibility

Item_Visibility có tận 526 giá trị bằng 0.0 mà độ hiện diện sản phẩm trên kệ hàng không thể bằng 0.0 được. Cho nên xử lý bằng cách

- Biến đổi các giá trị 0.0 thành Nan
- Nhóm lại các sản phẩm cùng loại và thực hiện phương pháp nội suy để suy luận ra diện tích mà mặt hàng được bày tại siêu thị.

```
bigmart_train['Item_Visibility'].value_counts()
```

```
Item_Visibility
0.000000    526
0.076975      3
0.162462      2
0.076841      2
0.073562      2
...
0.013957      1
0.110460      1
0.124646      1
0.054142      1
0.044878      1
Name: count, Length: 7880, dtype: int64
```



Hình 4.3: Item_Visibility ban đầu

Hình 4.4: Item_Visibility gốc và sau khi nội suy

Sau khi sử dụng phương pháp nội suy có thể đánh giá thông qua biểu đồ KDE: Item_Visibility gốc và Item_Visibility sau khi được nội suy có phân phối rất sát nhau, cho thấy bộ dữ liệu không bị ảnh hưởng.

4.2. Xử lý dữ liệu bị thiếu

```
bigmart_train.isnull().sum()
```

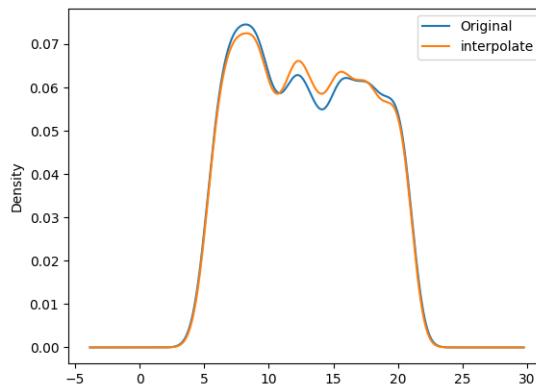
```
Item_Identifier          0
Item_Weight              1463
Item_Fat_Content         0
Item_Visibility          0
Item_Type                0
Item_MRP                 0
Outlet_Identifier        0
Outlet_Establishment_Year 0
Outlet_Size               2410
Outlet_Location_Type     0
Outlet_Type                0
Item_Outlet_Sales         0
dtype: int64
```

Hình 4.5: Thống kê các giá trị bị thiếu

Thống kê cho thấy 2 biến bị thiếu dữ liệu là Item_Weight có 1463 giá trị và Outlet_Size có 2410 giá trị. Nhóm sẽ đi xử lý từng biến sau.

Item_Weight

Với Item_Weight chỉ trọng lượng của mỗi sản phẩm, cho nên cũng sẽ nhóm lại các



Hình 4.6: Item_Weight gốc và sau khi nội suy

sản phẩm cùng loại (Item_Identifier) và thực hiện phương pháp nội suy để tìm ra trọng lượng của các sản phẩm còn thiếu.

Sau khi sử dụng phương pháp nội suy có thể đánh giá thông qua biểu đồ KDE: Item_Weight gốc và Item_Weight sau khi được nội suy có phân phối khá sát nhau, cho thấy bộ dữ liệu không bị ảnh hưởng.

Outlet_Size

Outlet_Type	Grocery Store	Supermarket Type1	Supermarket Type2	Supermarket Type3
Outlet_Size	Small	Small	Medium	Medium

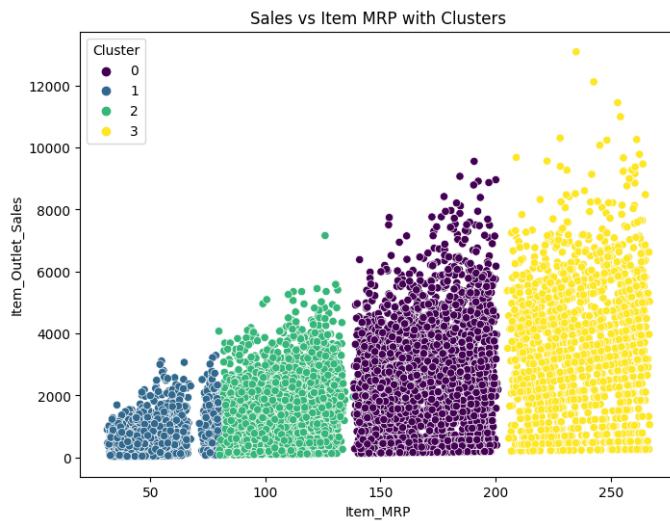
Hình 4.7: Mode của Outlet_Size theo từng Outlet_Type

Nhận thấy Outlet_Size (Kích thước của siêu thị Big Mart) và Outlet_Type (Loại cửa hàng siêu thị) sẽ có mối quan hệ đến nhau. Nhóm sẽ xử lý bằng cách tìm mode của Outlet_size theo từng Outlet_Type.

Kết quả hiển thị rằng Outlet_Size thuộc 'Grocery Store' và 'Supermarket Type1' có xu hướng nhỏ 'Small'. Còn Outlet_Size thuộc 'Grocery Store' và 'Supermarket Type1' có xu hướng nhỏ 'Medium'. Từ đây dễ dàng suy luận giá trị dữ liệu còn thiếu trong Outlet_Size dựa theo mode của Outlet_Type.

4.3. Thêm các biến mới vào dữ liệu

Cluster



Hình 4.8: Biến Item_MRP và Cluster

Phân cụm theo K-means Item_MRP và tạo biến mới tên Cluster. Giá của sản phẩm sẽ được phân theo 4 loại khác nhau

Outlet_Age

	Outlet_Establishment_Year	Outlet_age
0	1999	14
1	2009	4
2	1999	14
3	1998	15
4	1987	26
...
8518	1987	26
8519	2002	11
8520	2004	9
8521	2009	4
8522	1997	16

Hình 4.9: Biến Outlet_Establish_Year và Outlet_Age

Outlet_Establish_Year là năm thành lập của cửa hàng hàng, nhóm sẽ tạo ra biến mới Outlet_Age là số năm hoạt động của cửa hàng. Sẽ lấy mốc vào năm 2013 là thời điểm dữ liệu được thu thập trừ đi Outlet_Establish_Year.

Item_Id

	Item_Identifier	Item_Id
0	FDA15	FD
1	DRC01	DR
2	FDN15	FD
3	FDX07	FD
4	NCD19	NC
...
8518	FDF22	FD
8519	FDS36	FD
8520	NCJ29	NC
8521	FDN46	FD
8522	DRG01	DR

Hình 4.10: Biến Item_Identifier và Item_Id

Để tạo ra biến mới Item_Id, sẽ lấy 2 kí tự đầu trong Item_Identifier và Item_Id bao gồm 3 giá trị tương ứng loại mặt hàng: FD (Food: đồ ăn), DR (Drink: Thức uống) và NC (Non-consumable: sản phẩm không tiêu thụ được)

Item_Fat_Content

	Item_Id	Item_Fat_Content
0	FD	Low Fat
1	DR	Regular
2	FD	Low Fat
3	FD	Regular
4	NC	Non-Edible
...
8518	FD	Low Fat
8519	FD	Regular
8520	NC	Non-Edible
8521	FD	Regular
8522	DR	Low Fat

Hình 4.11: Biến Item_Id và Item_Fat_Content

Nhận thấy dễ dàng những loại mặt hàng không tiêu thụ được (non-consumable) sẽ không thể chứa chất béo. Nên sẽ biến đổi những sản phẩm có chất béo không tiêu thụ được thành giá trị 'Non-Edible'.

Price_Per_Unit

	Item_MRP	Item_Weight_interpolate	Price_Per_Unit
0	249.8092	6.135	40.718696
1	48.2692	19.500	2.475344
2	141.6180	6.920	20.465029
3	182.0950	18.250	9.977808
4	53.8614	20.100	2.679672
...
8518	214.5218	8.235	26.050006
8519	108.1570	15.300	7.069085
8520	85.1224	7.670	11.098096
8521	103.1332	15.200	6.785079
8522	75.4670	8.510	8.868038

Hình 4.12: Biến Item_MRP, Item_Weight_interpolate, Price_Per_Unit

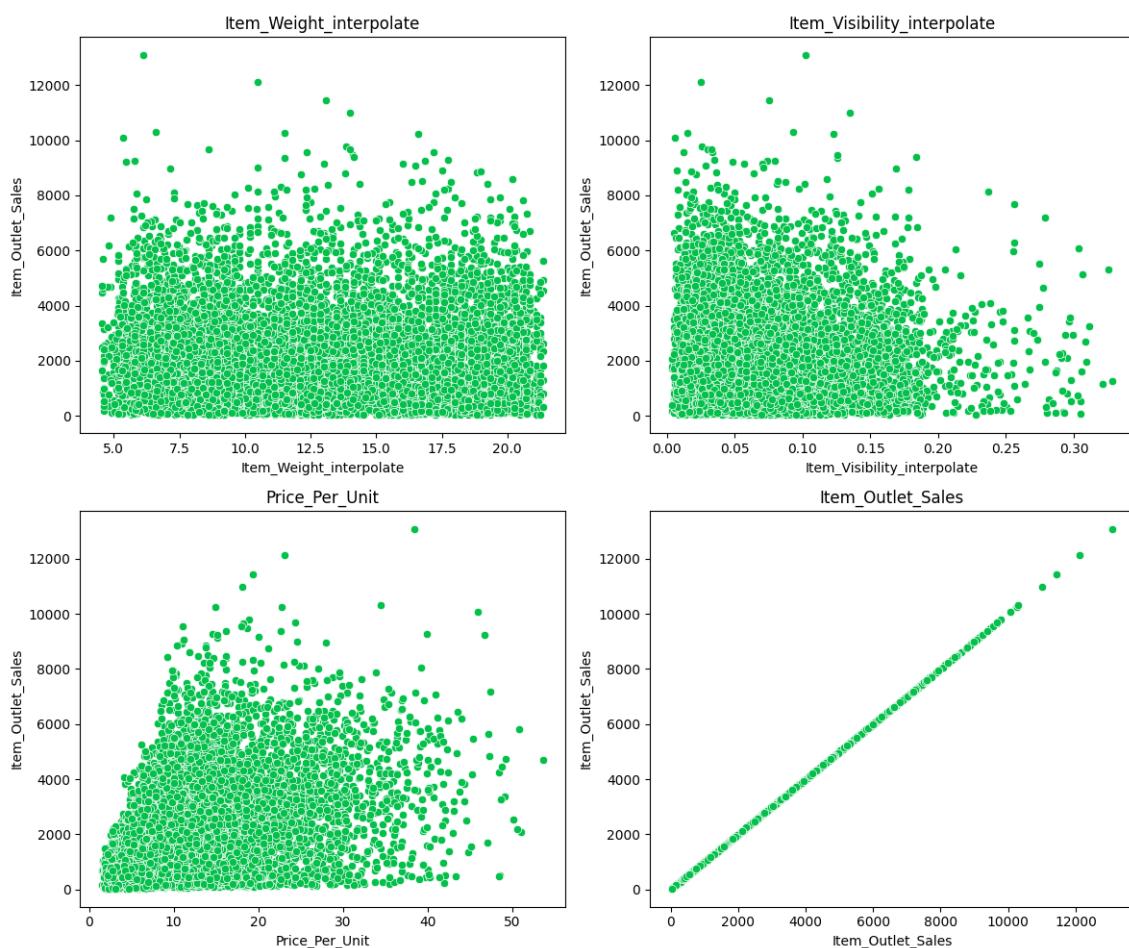
Biến Price_Per_Unit (giá sản phẩm trên 1 đơn vị) được tạo bằng cách lấy Item_MRP

(giá bán lẻ cao nhất sản phẩm) chia cho Item_Weight_interpolate (trọng lượng sản phẩm).

4.4. Trực quan dữ liệu

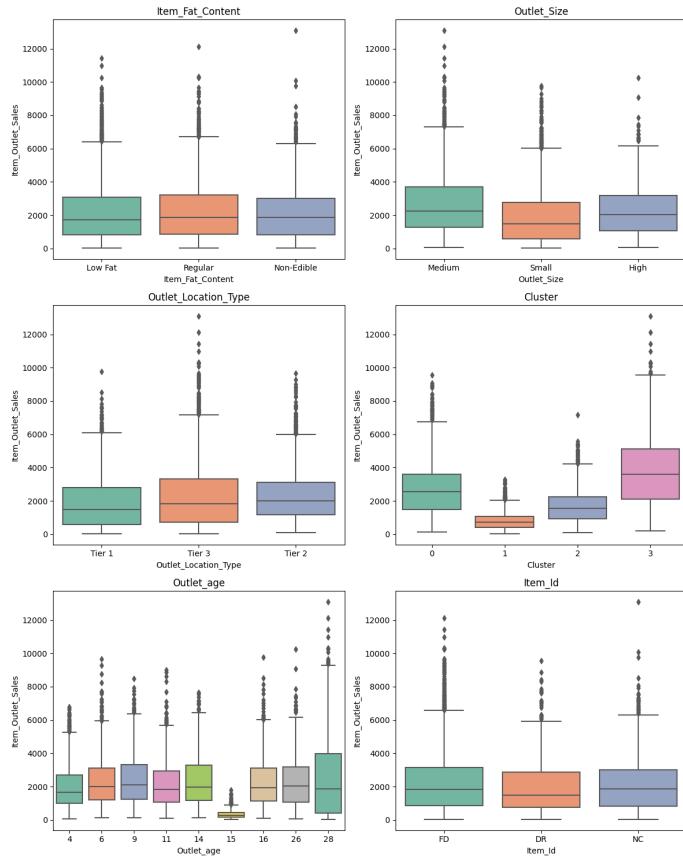
Sau khi thực hiện các bước xử lý dữ liệu sẽ thu được 12 biến độc lập và 1 biến phụ thuộc (Item_Outlet_Sales)

Biến định lượng

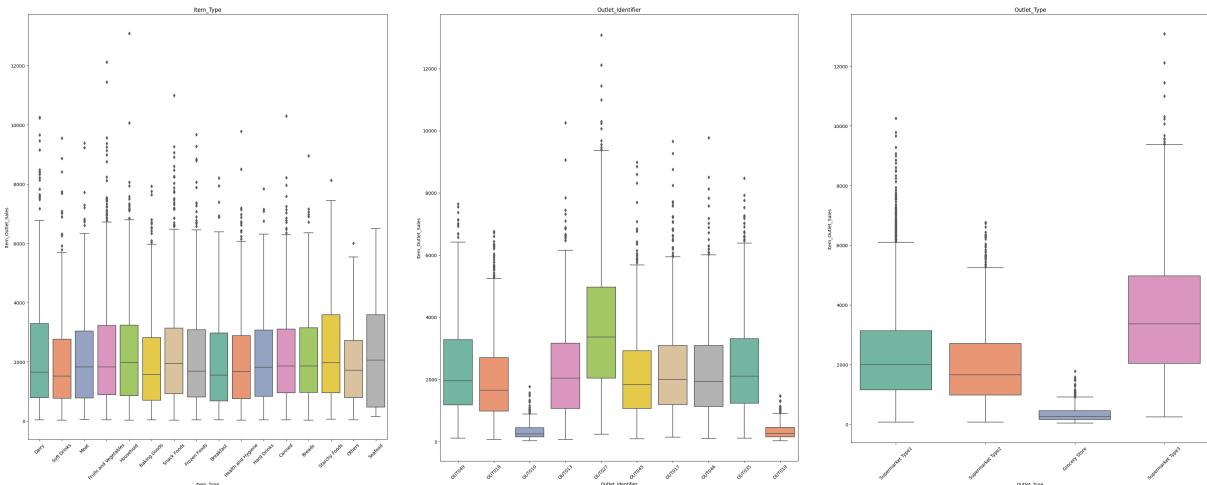


Hình 4.13: Biến định lượng của dữ liệu mới

Biến định tính



Hình 4.14: Biến định tính của dữ liệu mới



Hình 4.15: Item_Type

Hình 4.16: Outlet_Identifier

Hình 4.17: Outlet_Type

Chương 5

Xây dựng mô hình

5.1. Mô hình hồi quy tuyến tính

5.1.1. Quá trình xây dựng và cải tiến mô hình

```
linear_model <- lm(Item_Outlet_Sales ~ ., data = data)
summary(linear_model)

Residuals:
    Min      1Q  Median      3Q     Max 
-4125.2 -686.7 -103.5  578.7 7566.8 

Coefficients: (17 not defined because of singularities)
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) -255.322   172.543 -1.480   0.139    
Item_Weight_interpolate 37.279    4.930   7.561 4.41e-14 ***
Item_Visibility_interpolate -259.362  248.233 -1.045   0.296    
Price_Per_Unit 35.211    3.770   9.339 < 2e-16 ***
Item_Fat_ContentNon-Edible -4.004   147.637 -0.027   0.978    
Item_Fat_ContentRegular 37.880    28.968   1.308   0.191    
Item_TypeBreads 7.541    85.519   0.088   0.930    
Item_TypeBreakfast 41.188   118.585   0.347   0.728    
Item_TypeCanned 23.862   63.879   0.374   0.709    
Item_TypeDairy -13.605   67.195   -0.202  0.840    
Item_TypeFrozen Foods -10.723   59.933   -0.179  0.858    
Item_TypeFruits and Vegetables 60.979   55.850   1.092   0.275    
Item_TypeHard Drinks 58.352   141.942   0.411   0.681    
Item_TypeHealth and Hygiene 18.455   101.786   0.181   0.856    
Item_TypeHousehold 25.104   96.318   0.261   0.794    
Item_TypeMeat 34.401   71.846   0.479   0.632    
Item_TypeOthers NA        NA        NA        NA        
Item_TypeSeafood 217.550   150.723   1.443   0.149    
Item_TypeSnack Foods -1.422    56.185   -0.025  0.980

Call:
lm(formula = Item_Outlet_Sales ~ Cluster + Outlet_Identifier +
Price_Per_Unit + Item_Weight_interpolate, data = data)

Residuals:
    Min      1Q  Median      3Q     Max 
-4048.6 -681.6 -104.4  577.7 7530.4 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) -229.439   127.295 -1.802   0.0715 .  
Cluster1    -1457.461   53.928 -27.026 < 2e-16 *** 
Cluster2     -799.229   37.559 -21.279 < 2e-16 *** 
Cluster3     821.862   43.501 18.893 < 2e-16 *** 
Outlet_IdentifierOUT013 1955.258   61.585 31.749 < 2e-16 *** 
Outlet_IdentifierOUT017 2028.094   61.645 32.899 < 2e-16 *** 
Outlet_IdentifierOUT018 1654.804   61.629 26.851 < 2e-16 *** 
Outlet_IdentifierOUT019 19.568   69.807   0.280  0.7792  
Outlet_IdentifierOUT027 3375.410   61.531 54.857 < 2e-16 *** 
Outlet_IdentifierOUT025 2065.017   61.594 33.526 < 2e-16 *** 
Outlet_IdentifierOUT045 1854.151   61.610 30.095 < 2e-16 *** 
Outlet_IdentifierOUT046 1920.294   61.606 31.170 < 2e-16 *** 
Outlet_IdentifierOUT049 2024.151   61.601 32.859 < 2e-16 *** 
Price_Per_Unit 35.354    3.762   9.399 < 2e-16 *** 
Item_Weight_interpolate 37.470   4.923   7.612 2.99e-14 *** 
---
Signif. codes:  0 '****' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

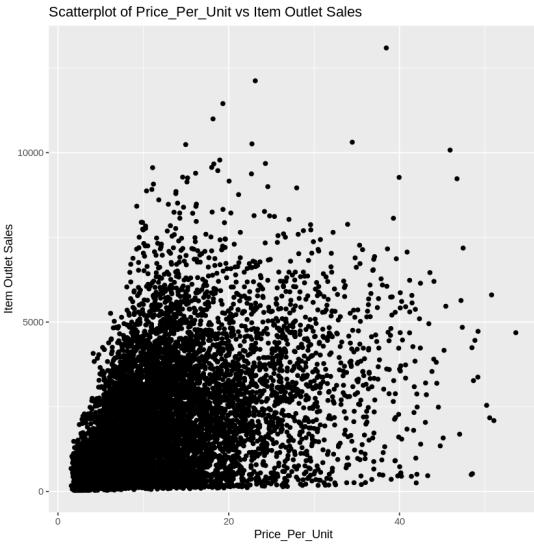
Residual standard error: 1148 on 8508 degrees of freedom
Multiple R-squared:  0.5481, Adjusted R-squared:  0.5473 
F-statistic:  737 on 14 and 8508 DF, p-value: < 2.2e-16
```

Hình 5.1: Mô hình đầy đủ

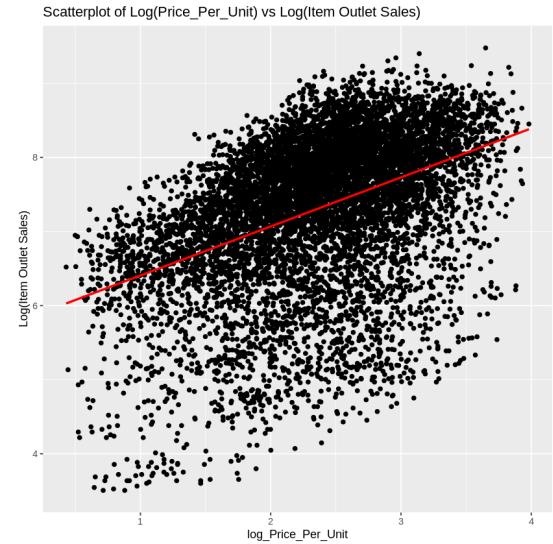
Hình 5.2: Áp dụng STEP-WISE

Nhiều biến phân loại khác (như Item_Type, Outlet_Size, Outlet_Location_Type, Outlet_Type) và một số biến khác không có ý nghĩa thống kê ($p\text{-value} > 0.05$), điều này cho thấy chúng không có ảnh hưởng đáng kể đến doanh thu.

Sử dụng phương pháp forward stepwise selection cho thấy hiệu quả tương tự so với mô hình ban đầu nhưng đã loại bỏ được các biến không cần thiết, giúp mô hình trở nên đơn giản hơn và dễ giải thích hơn.



Hình 5.3: Biểu đồ phân tán Price_Per_Unit và Item_Outlet_Sales



Hình 5.4: Sau khi biến đổi hàm log

Biến đổi hàm log

Đồ thị phân tán này cho thấy mối quan hệ giữa Price_Per_Unit (giá mỗi đơn vị sản phẩm) và Item_Outlet_Sales (doanh thu bán hàng tại cửa hàng). Tuy nhiên, không có sự phân bố rõ ràng, cho thấy có thể có nhiều yếu tố khác ảnh hưởng đến doanh thu.

Khi chuyển đổi hàm log, Log(Price_Per_Unit) cho thấy một mối quan hệ tuyến tính rõ ràng hơn với Log(Item_Outlet_Sales), cho thấy rằng có thể sử dụng biến đổi log để cải thiện mô hình hồi quy.

```

Call:
lm(formula = log_Item_Outlet_Sales ~ Cluster + Outlet_Identifier +
log_Price_Per_Unit + Item_Weight_interpolate, data = data)

Residuals:
    Min      1Q   Median      3Q     Max 
-2.18851 -0.27858  0.05397  0.36687  1.39781 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 2.753001  0.134115 20.527 < 2e-16 ***
Cluster1    -0.224489  0.045447 -4.940 7.98e-07 ***
Cluster2    -0.098639  0.021611 -4.564 5.08e-06 ***
Cluster3    0.050969  0.020618  2.472  0.0135 *  
Outlet_IdentifierOUT013 1.939143  0.027997 69.264 < 2e-16 ***
Outlet_IdentifierOUT017 1.996287  0.028026 71.230 < 2e-16 ***
Outlet_IdentifierOUT018 1.791217  0.028021 63.923 < 2e-16 ***
Outlet_IdentifierOUT019 0.027675  0.031736  0.872  0.3832  
Outlet_IdentifierOUT027 2.499375  0.027973 89.349 < 2e-16 ***
Outlet_IdentifierOUT035 2.013113  0.028004 71.887 < 2e-16 ***
Outlet_IdentifierOUT045 1.922092  0.028009 68.623 < 2e-16 ***
Outlet_IdentifierOUT046 1.959081  0.028009 69.946 < 2e-16 ***
outlet_IdentifierOUT049 1.996465  0.028005 71.288 < 2e-16 ***
log_Price_Per_Unit    0.834443  0.035223 23.690 < 2e-16 ***
Item_Weight_interpolate 0.069030  0.003196 21.597 < 2e-16 ***

---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.522 on 8508 degrees of freedom
Multiple R-squared:  0.737,    Adjusted R-squared:  0.7366 
F-statistic: 1703 on 14 and 8508 DF,  p-value: < 2.2e-16

```

Hình 5.5: Kết quả mô hình log

- R-squared (R^2): 0.737, chỉ số này chỉ ra rằng mô hình giải thích được khoảng 73.7% sự biến động của dữ liệu, một mức độ tương đối tốt.
- F-statistic: 1703, với p-value < 0.001, cho thấy mô hình là phù hợp và có hiệu suất tốt.

Mô hình hồi quy với biến đổi log đã cải thiện đáng kể hiệu suất so với mô hình gốc, chỉ số R^2 từ 0.5481 lên 0.737. Các biến độc lập đã được chọn đều có ý nghĩa thống kê và có ảnh hưởng đáng kể đến doanh thu tại các cửa hàng. Những kết quả này có thể hữu ích để tối ưu hóa chiến lược kinh doanh và dự báo doanh thu cho các cửa hàng.

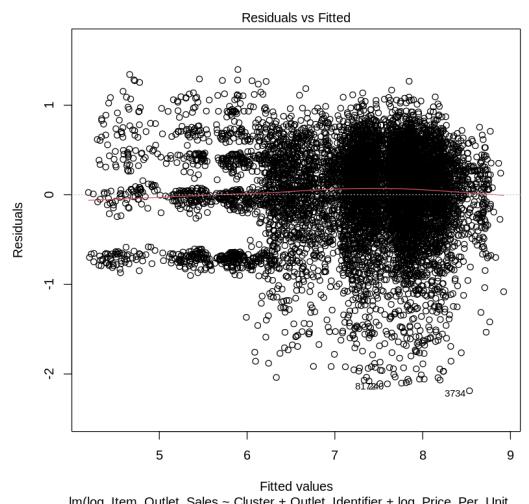
5.1.2. Đánh giá mô hình

Kiểm tra giả thiết tuyến tính của dữ liệu và giả thiết phần dư có trung bình bằng 0

```
[1] t.test(model_step_log$residuals, mu = 0)

One Sample t-test

data: model_step_log$residuals
t = 1.6654e-15, df = 8522, p-value = 1
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
-0.01107486 0.01107486
sample estimates:
mean of x
9.408885e-18
```



Hình 5.6: Kiểm định t.test

Hình 5.7: Biểu đồ Residuals & Fitted

Biểu đồ Residuals vs Fitted (Đã thử nghiệm) cho thấy một mô hình hồi quy tốt với các điểm dữ liệu phân bố đồng đều xung quanh đường thẳng trung bình 0, không có biểu hiện rõ ràng của mối quan hệ phi tuyến tính.

Nhưng khi kiểm định t_test, p-value = 1: Chúng ta không có đủ bằng chứng để bác bỏ giả thuyết không (null hypothesis) rằng giá trị trung bình của phần dư bằng 0. Điều này có nghĩa là giá trị trung bình của phần dư không khác biệt đáng kể so với 0.

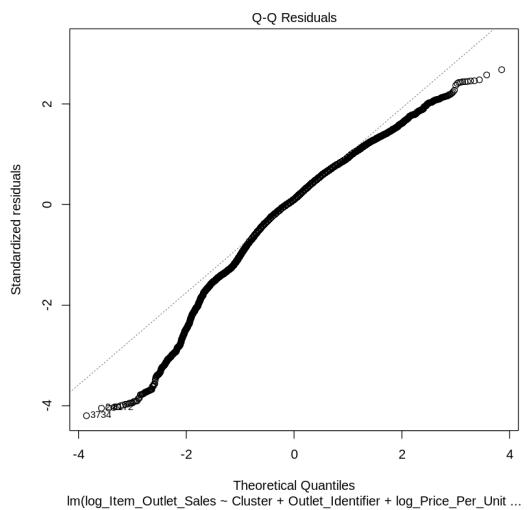
Kiểm tra giả thiết phần dư có phân phối chuẩn

Biểu đồ Q-Q (Quantile-Quantile) Normal cho thấy phân phối của các giá trị phần dư khá gần với đường chéo.

```
[ ] shapiro.test(sample(model_step_log$residuals, 5000))
→
Shapiro-Wilk normality test

data: sample(model_step_log$residuals, 5000)
W = 0.96299, p-value < 2.2e-16
```

Hình 5.8: Kiểm định Shapiro.test



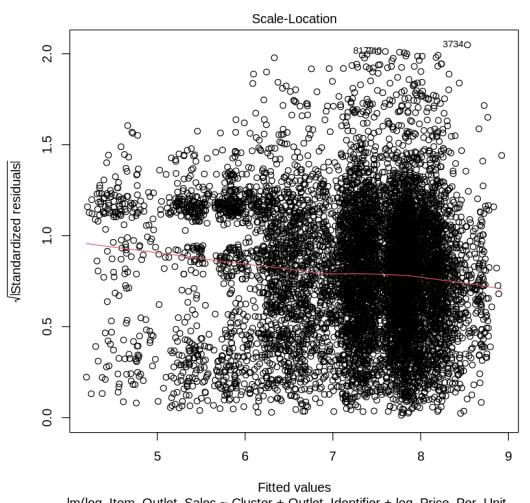
Hình 5.9: Biểu đồ QQ-Residuals

Dù kết quả từ biểu đồ Q-Q cho thấy phần dư khá gần với phân phối chuẩn, kiểm tra Shapiro-Wilk cho thấy rằng phần dư của mô hình không tuân theo phân phối chuẩn ($p\text{-value} < 2.2\text{e-}16$)

Kiểm tra giả thiết phương sai không đồng nhất

```
▶ ncvTest(model_step_log)
→ Non-constant Variance Score Test
Variance formula: ~ fitted.values
Chisquare = 32.86281, Df = 1, p = 9.8897e-09
```

Hình 5.10: Kiểm định ncv.test



Hình 5.11: Biểu đồ Scale-Location

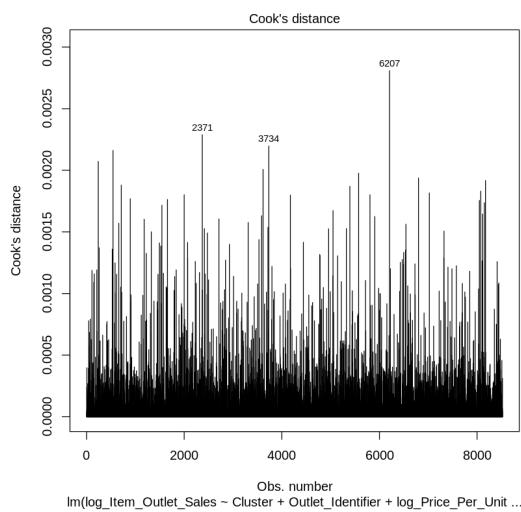
Kết quả kiểm tra phương sai không đồng đều (heteroscedasticity) với ncvTest cho thấy:

- Chisquare = 32.86281: Giá trị của thống kê kiểm tra.

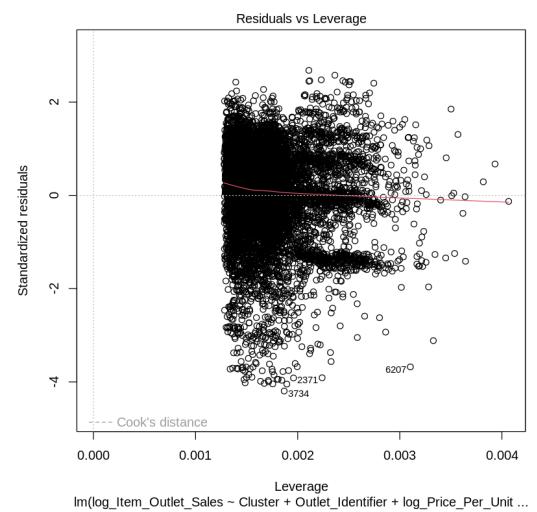
- Df = 1: Độ tự do của kiểm tra.
- p = 9.8897e-09: Giá trị p rất nhỏ, nhỏ hơn rất nhiều so với mức ý nghĩa thông thường (0.05)

Giá trị p rất nhỏ: Chúng ta có đủ bằng chứng để bác bỏ giả thuyết không (null hypothesis) rằng phương sai của phần dư là đồng đều. Điều này có nghĩa là phương sai của phần dư thay đổi theo giá trị dự đoán, hay nói cách khác là mô hình đang gặp phải vấn đề về phương sai không đồng đều (heteroscedasticity).

Kiểm tra điểm ảnh hưởng mô hình



Hình 5.12: Biểu đồ Cook



Hình 5.13: Biểu đồ Residuals & Leverage

Tại điểm quan sát 2371, 5049, và 5574:

- Các quan sát này có giá trị Cook's Distance cao, cho thấy rằng chúng có ảnh hưởng lớn đến các hệ số ước lượng của mô hình hồi quy.
- Chúng có leverage cao và phần dư chuẩn hóa lớn, nghĩa là chúng không chỉ có giá trị dự đoán xa so với giá trị thực mà còn có thể thay đổi đáng kể độ dốc của đường hồi quy nếu chúng bị loại bỏ.

Kiểm tra giả thuyết giữa các biến độc lập không có mối quan hệ đa cộng tuyến

A matrix: 4 × 3 of type dbl

	GVIF	Df	GVIF^(1/(2*Df))
Cluster	10.986691	3	1.491001
Outlet_Identifier	1.003361	9	1.000186
log_Price_Per_Unit	16.653264	1	4.080841
Item_Weight_interpolate	6.722447	1	2.592768

Hình 5.14: Kiểm định vif

Chỉ số VIF cho ra kết quả Cluster: 1.491001, Outlet_Identifier: 1.000186, log_Price_Per_Unit: 4.080841 và Item_Weight_interpolate: 2.592768. Tất cả đều < 10 cho nên không có vấn đề nghiêm trọng về đa cộng tuyến.

5.1.3. Phân tích ANCOVA

Kiểm định mô hình ancova

ancova <- aov(log_Item_Outlet_Sales ~ Cluster + Outlet_Identifier + log_Price_Per_Unit + Item_Weight_interpolate, data = data)

summary(ancova)

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Cluster	3	2283	760.9	2792.36	<2e-16 ***
Outlet_Identifier	9	4062	451.3	1656.23	<2e-16 ***
log_Price_Per_Unit	1	26	26.0	95.23	<2e-16 ***
Item_Weight_interpolate	1	127	127.1	466.44	<2e-16 ***
Residuals	8508	2318	0.3		

Signif. codes: 0 |****| 0.001 ** 0.01 * 0.05 . 0.1 ' > 1

Hình 5.15: Mô hình ANCOVA

Kết quả phân tích ANCOVA cho thấy rằng tất cả các biến độc lập (Cluster, Outlet_Identifier, log_Price_Per_Unit và Item_Weight_interpolate) đều có ảnh hưởng đáng kể đến biến phụ thuộc (log_Item_Outlet_Sales), với mức ý nghĩa p-value < 2e-16 cho tất cả các biến.

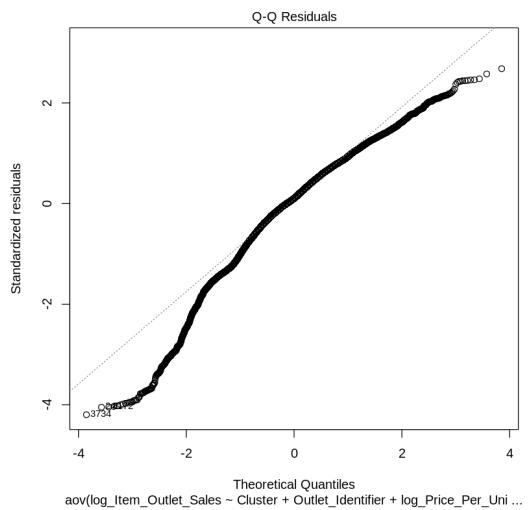
Kiểm tra giả thiết phần dư có phân phối chuẩn

```
#Kiểm tra phần dư có phân phối chuẩn
plot(ancova,which=2)
shapiro.test(sample(ancova$residuals,5000))

Shapiro-Wilk normality test

data: sample(ancova$residuals, 5000)
W = 0.9634, p-value < 2.2e-16
```

Hình 5.16: Kiểm định Shapiro.test



Hình 5.17: Biểu đồ QQ-Residuals

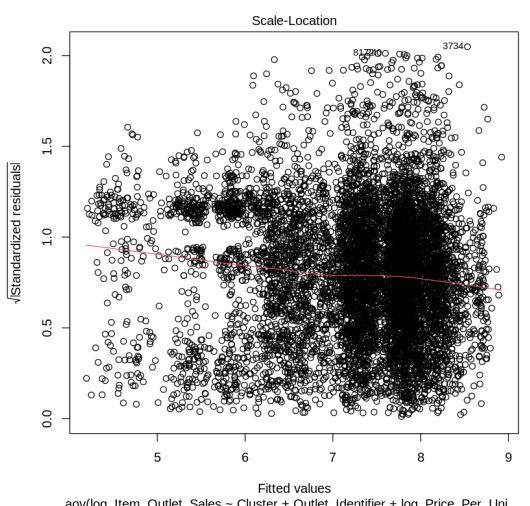
- Các điểm trong Q-Q plot không nằm hoàn toàn trên đường chéo, điều này cho thấy phần dư không tuân theo phân phối chuẩn.
- Kết quả kiểm định Shapiro-Wilk có giá trị p-value < 2.2e-16, nghĩa là bác bỏ giả thuyết không (null hypothesis) rằng phần dư của mô hình ANCOVA không tuân theo phân phối chuẩn.

Kiểm tra giả thiết phương sai không đồng nhất

A tibble: 1 × 4			
df1	df2	statistic	p
9	8513	11.39794	6.165996e-18

A tibble: 1 × 4			
df1	df2	statistic	p
3	8519	0.9075128	0.4364571

Hình 5.18: Kiểm định levene_test

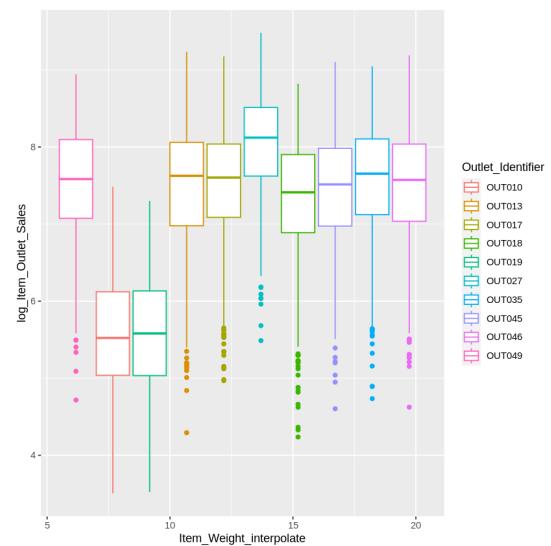
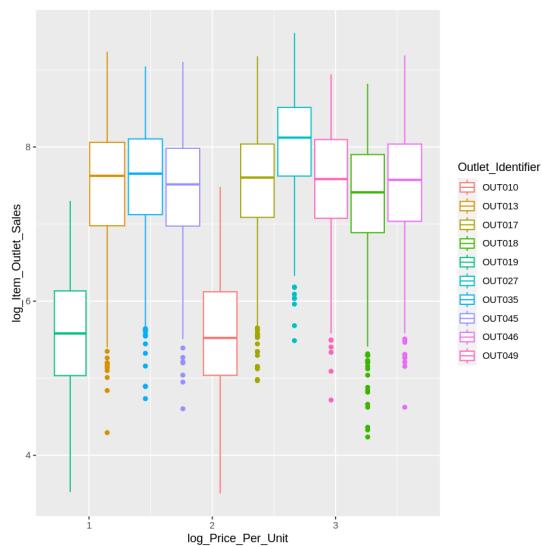


Hình 5.19: Biểu đồ Scale-Location

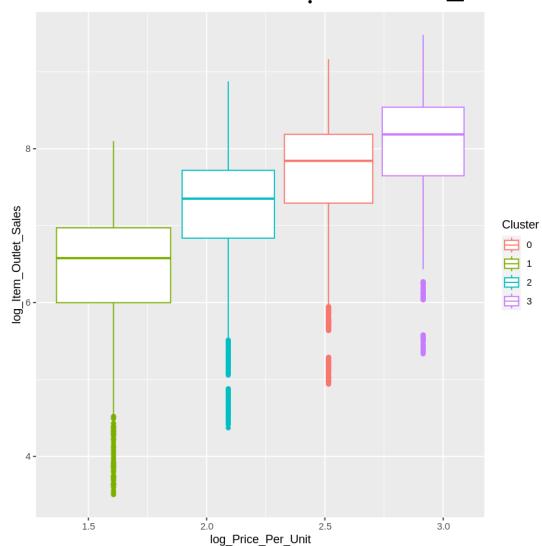
Kết quả Levene's Test:

- Outlet_Identifier: Giá trị $p = 6.166e-18 (< 0.05)$, chúng ta bác bỏ giả thuyết không (null hypothesis) rằng các nhóm có phương sai bằng nhau. Điều này ngụ ý rằng các nhóm trong Outlet_Identifier có phương sai không đồng nhất.
- Cluster: Giá trị $p = 0.4365 (> 0.05)$, không có đủ bằng chứng để bác bỏ giả thuyết không (null hypothesis) rằng các nhóm có phương sai bằng nhau. Điều này ngụ ý rằng các nhóm trong Cluster có phương sai đồng nhất.

Kiểm tra dữ liệu ảnh hưởng đến mô hình

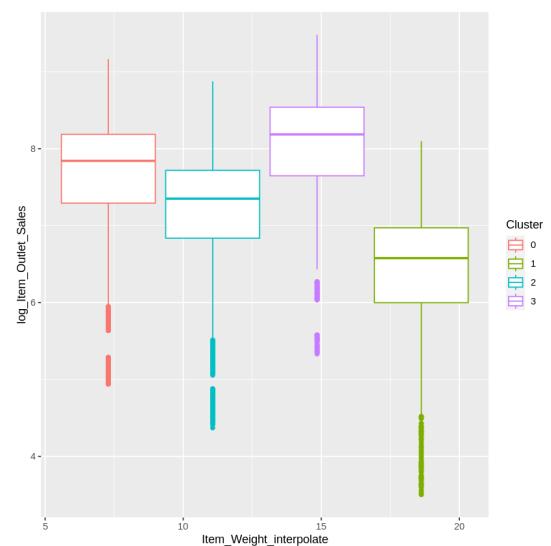


Hình 5.20: Kiểm định levene_test



Hình 5.22: Kiểm định levene_test

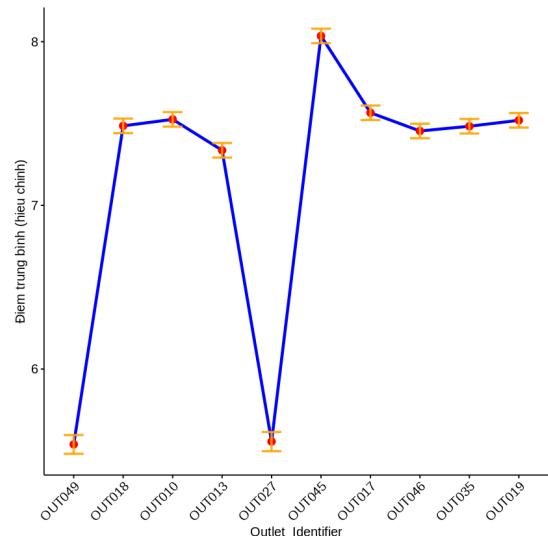
Hình 5.21: Biểu đồ Scale-Location



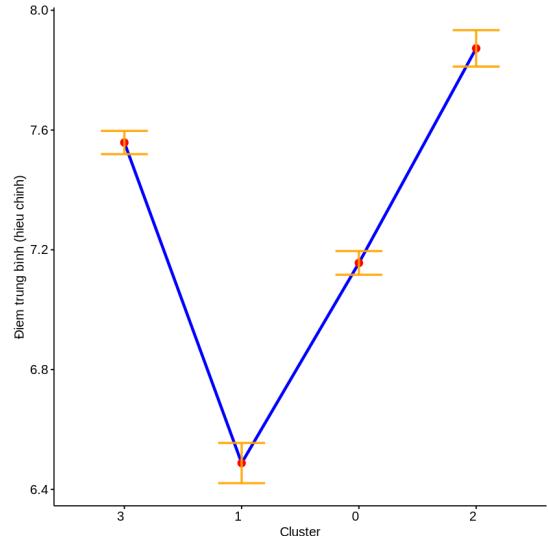
Hình 5.23: Biểu đồ Scale-Location

Biểu đồ hộp cho phép xem xét phân phối của biến phụ thuộc log_Item_Outlet_Sales theo các nhóm của các biến độc lập log_Price_Per_Unit và Item_Weight_interpolate, được phân loại bởi Outlet_Identifier và Cluster. Từ đó dễ dàng có thể quan sát xem có sự khác biệt đáng kể giữa các nhóm

Trung bình hiệu chỉnh



Hình 5.24: Trung bình hiệu chỉnh so với Outlet_Identifier



Hình 5.25: Trung bình hiệu chỉnh so với Cluster

Các biểu đồ trên biểu diễn các điểm trung bình được điều chỉnh cho các biến và phân loại tương ứng;

- Biến Outlet_Identifier: Các cửa hàng có mã OUT045 và OUT049 có điểm trung bình thấp nhất. Cửa hàng OUT027 có điểm trung bình cao nhất.
- Biến Cluster: Cluster 1 có điểm trung bình thấp nhất. Cluster 2 có điểm trung bình cao nhất.

5.2. Mô hình học máy Random Forest

Để so sánh với mô hình hồi quy tuyến tính, áp dụng mô hình học máy Random Forest để đi dự đoán doanh số bán hàng mỗi sản phẩm tại các cửa hàng khác nhau bằng các phương pháp sau:

Chọn lọc đặc trưng quan trọng

feature	XGBRF_importance
8 Outlet_Type	0.360933
9 Cluster	0.279483
10 Outlet_age	0.104685
2 Price_Per_Unit	0.093090
6 Outlet_Size	0.061110
5 Outlet_Identifier	0.060821
7 Outlet_Location_Type	0.022062
0 Item_Weight_interpolate	0.007700
4 Item_Type	0.003847
1 Item_Visibility_interpolate	0.003326
11 Item_Id	0.001707
3 Item_Fat_Content	0.001237

Hình 5.26: Mô hình ANCOVA

Để mô hình hoạt động tốt, sẽ chọn ra 5 biến quan trọng nhất ảnh hưởng tới mô hình bằng XGBRFRegressor, bao gồm: Outlet_Type, Cluster, Outlet_age, Price_Per_Unit, Outlet_Size

Biến đổi hàm log



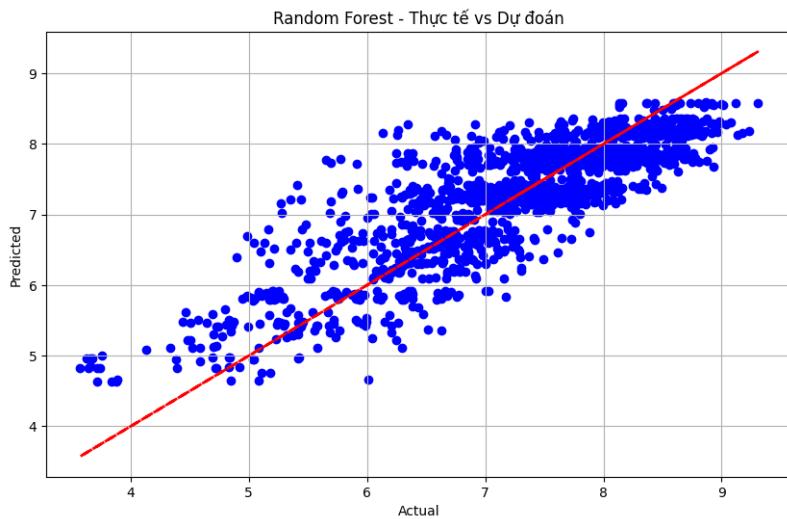
Hình 5.27: Biểu đồ phân tán Price_Per_Unit và Item_Outlet_Sales



Hình 5.28: Sau khi biến đổi hàm log

Tương tự như mô hình hồi quy tuyến tính, ở mô hình random forest cũng đi biến đổi hàm log cho biến mục tiêu có thể cải thiện hiệu suất của mô hình

Kết quả 2 mô hình



Hình 5.29: Kết quả Random Forest

Mô hình Random Forest được huấn luyện và đánh giá trên tập dữ liệu kiểm tra với các tham số được tinh chỉnh. Kết quả đánh giá mô hình trên tập kiểm tra cho thấy:

- Mean Squared Error (MSE): 0.2678, đây là sự sai lệch bình phương trung bình giữa dự đoán và giá trị thực tế. Giá trị MSE càng thấp càng tốt, vì nó cho biết mô hình dự đoán gần giá trị thực tế hơn.
- Mean Absolute Error (MAE): 0.3978, đây là sự sai lệch trung bình giữa dự đoán và giá trị thực tế. MAE càng thấp thì mô hình dự đoán càng chính xác.
- R-squared (R^2): 0.7437, đây là tỷ lệ phương sai giải thích bởi mô hình so với tổng phương sai của dữ liệu. Giá trị R^2 càng gần 1 thì mô hình càng tốt.

Biểu đồ thể hiện sự tương quan giữa giá trị dự đoán và giá trị thực tế. Các điểm dữ liệu gần đường thẳng đường chéo (đường dự đoán) cho thấy mô hình dự đoán khá tốt. Tuy nhiên, vẫn còn một số điểm dữ liệu rải rác không nằm gần đường thẳng này, điều này có thể là do mô hình chưa thể dự đoán chính xác cho những trường hợp này.

Chương 6

Thảo luận vấn đề

6.1. Kết luận về mô hình hóa

1. Hồi quy tuyến tính sau khi biến đổi hàm log:

- Mô hình đã cải thiện sau khi áp dụng biến đổi hàm log cho biến phụ thuộc và một số biến độc lập.
- Tuy nhiên, kết quả từ các kiểm định giả thuyết cho thấy mô hình chưa thực sự đáng tin cậy. Điều này có thể do một số nguyên nhân như: không đáp ứng được các giả định của mô hình, có sự thiếu sót trong quá trình mô hình hóa, hoặc dữ liệu không phản ánh đúng mối quan hệ giữa các biến.

2. Random Forest:

- Kết quả từ mô hình Random Forest khá ấn tượng với chỉ số R^2 lên tới 0.7437, cho thấy mô hình giải thích được một phần lớn sự biến thiên của biến phụ thuộc.
- Tính linh hoạt của Random Forest đã giúp mô hình mô hình hóa mối quan hệ phức tạp giữa các biến một cách hiệu quả hơn so với hồi quy tuyến tính, dẫn đến hiệu suất dự đoán tốt hơn.

6.2. So sánh với nghiên cứu khác

Trong nghiên cứu 'Sales Prediction of Big Mart based on Linear Regression, Random Forest, and Gradient Boosting' của tác giả Ruiyun Kang, đã được công bố trên tạp chí *Advances in Economics, Management and Political Sciences* vào tháng 9 năm 2023, tác giả đã tiếp cận vấn đề dự đoán doanh số bán hàng tại Big Mart bằng cách sử dụng ba mô hình học máy khác nhau, bao gồm Linear Regression, Random Forest và Gradient Boosting.

Trong nghiên cứu của tác giả, các mô hình học máy đã đạt được các kết quả R^2 tương ứng là 0.567, 0.574 và 0.609. Điều này cho thấy mức độ giải thích của các mô hình đối với biến phụ thuộc là từ trung bình đến cao.

Trong báo cáo của nhóm chúng tôi, chúng tôi đã tiếp cận vấn đề này một cách khác bằng cách thêm các biến mới và biến đổi hàm log. Kết quả là cả hai mô hình Linear Regression và Random Forest đã cải thiện đáng kể hiệu suất, vượt qua các mô hình trong nghiên cứu của tác giả Ruiyun Kang. Điều này cho thấy sự hiệu quả của việc thêm các biến mới và biến đổi hàm log trong việc dự đoán doanh số bán hàng tại Big Mart.

Kết luận lại cả 2 báo cáo đều có các mô hình thử nghiệm không hoạt động hiệu quả và cần phải cải thiện thông qua lựa chọn mô hình khác và đánh giá. Tác giả Ruiyun Kang đề xuất hướng phát triển thêm trong tương lai: Nếu dữ liệu cho phép, việc tích hợp phân tích chuỗi thời gian hoặc sử dụng các mô hình phức tạp như LSTM có thể là cách tiếp cận tiềm năng cho việc cải thiện trong các nghiên cứu sau này.

6.3. Định hướng

Để có thể cải thiện mô hình, nhóm đã tham khảo một số đề xuất sau:

- Sử dụng mô hình khác: Mô hình Robust Regression có khả năng xử lý tốt hơn các giá trị ngoại lai và nhiễu trong dữ liệu so với hồi quy tuyến tính thông thường. Hay mô hình Ridge và LASSO là hai phương pháp hồi quy linh hoạt có thể giúp kiểm soát overfitting và cải thiện hiệu suất dự đoán của mô hình.
- Xử lý outlier: Một bước quan trọng để cải thiện mô hình là xử lý các giá trị ngoại lai trong dữ liệu. Sử dụng phương pháp như phạm vi tứ phân vị (IQR) để xác định và loại bỏ các outlier khỏi tập dữ liệu.
- Tạo thêm các biến đặc trưng mới: Một cách tiếp cận khác để cải thiện mô hình là tạo ra các biến đặc trưng mới. Điều này có thể bao gồm việc phát triển các biến tương tác giữa các biến hiện có, tạo biến đếm cho các sự kiện hoặc các biến đặc trưng phức tạp hơn để mô hình hóa mối quan hệ giữa các biến. Bằng cách này, có thể cải thiện khả năng giải thích của mô hình và tăng hiệu suất dự đoán.

Tài liệu tham khảo

- [1] Prasad, E. Durga, et al. (2023). BIG MART SALES PREDICTION USING MACHINE LEARNING AND PYTHON. *International Journal of Creative Research Thoughts (IJCRT)*, 11(5), 773. ISSN: 2320-2882
- [2] Kang, Ruiyun. (2023). Sales Prediction of Big Mart based on Linear Regression, Random Forest, and Gradient Boosting. *Advances in Economics, Management and Political Sciences*, 17, 201-208. DOI: 10.54254/2754-1169/17/20231094
- [3] Malik, Nikita, & Singh, Karan. (2020). SALES PREDICTION MODEL FOR BIG MART. *Advances in Economics, Management and Political Sciences*, 3, 22-32.
- [4] Rosenthal, Sonny. (2017). Regression Analysis, Linear. In: Visual Data: Collection, Analysis and Representation (pp. 1-50). ISBN: 9781118901762. DOI: 10.1002/9781118901731.iecrm0208
- [5] Johnson, Jeffrey. (1998). Visual Data: Collection, Analysis and Representation. (p. 18).