

**TRƯỜNG ĐẠI HỌC SÀI GÒN**  
**KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN KẾT THÚC HỌC PHẦN**  
**Môn: PHÂN TÍCH DỮ LIỆU**

Mã môn học: 841432

Khoa: Công nghệ thông tin

Chuyên ngành: Hệ thống thông tin

Nhóm nghiên cứu:

Trần Nguyên Lộc – 3120410297

Võ Đăng Quang – 3120410429

**Giảng viên phụ trách:**  
**PHAN THÀNH HUẤN**

**Thành phố Hồ Chí Minh, tháng 12 năm 2023**

## **Lời cảm ơn**

Trước hết em xin gửi đến lời cảm ơn chân thành và sâu sắc nhất đến thầy TS. Phan Thành Huân người trực tiếp hướng dẫn và tận tình chỉ bảo cho nhóm chúng em cho tới khi em hoàn thành đồ án của mình.

Tiếp đến em xin giành lời cảm ơn đến quý thầy cô Trường Đại học Sài Gòn – khoa Công nghệ thông tin đã truyền đạt cho em những kiến thức vô cùng quý báu và bổ ích trong suốt quá trình nghiên cứu và học tập tại trường.

Xin chân thành cảm ơn tới những người bạn đã luôn sát cánh cùng em, những lời động viên, những lần hỗ trợ những lúc cần thiết đã phần nào giúp em hoàn thành đồ án này.

Cuối cùng, em xin cảm ơn đến ba mẹ và người thân trong gia đình đã hỗ trợ và tạo điều kiện thuận lợi cho em trong suốt thời gian học tập và nghiên cứu tại Đại học Sài Gòn.

## **BẢNG ĐÁNH GIÁ MỨC ĐỘ THAM GIA**

Danh sách thành viên	MSSV	Mức độ tham gia
Trần Nguyên Lộc	3120410297	100%
Võ Đăng Quang	3120410429	100%

Đồ án này trình bày quá trình tìm kiếm, khám phá và phân tích thông tin từ tập dữ liệu tìm được. Trong đó có sử dụng các các thuật toán và phương pháp khai thác dữ liệu. Cuối cùng, quá trình khai thác dữ liệu dẫn đến việc trích xuất tri thức từ dữ liệu và đưa ra quyết định thông minh có lợi cho việc marketing.

## Mục lục

Lời cảm ơn .....	i
BẢNG ĐÁNH GIÁ MỨC ĐỘ THAM GIA.....	ii
Mục lục.....	iii
Danh mục bảng biểu.....	v
Danh mục hình ảnh .....	v
Chương 1: KHÁI QUÁT ĐỒ ÁN .....	1
1.1. Lí do chọn đề tài .....	1
1.2. Mô tả dữ liệu và cấu trúc dữ liệu.....	2
Chương 2: KHẢO SÁT VÀ TIỀN XỬ LÝ BỘ DỮ LIỆU.....	4
2.1. Khảo sát bộ dữ liệu.....	4
2.2. Tiền xử lí dữ liệu .....	5
2.2.1. Kiểm tra bộ dữ liệu .....	5
2.2.2. Biến đổi và thêm mới dữ liệu .....	8
2.2.3. Bộ dữ liệu tổng thể.....	10
2.3. Thăm dò dữ liệu sau tiền xử lý .....	11
2.3.1. Khái niệm về EDA.....	12
2.3.2. Mục đích sử dụng EDA .....	12
2.3.3. Thực nghiệm EDA đối với dữ liệu phân tích .....	13
2.3.3.1. Tỷ suất lợi nhuận của mặt hàng sản phẩm .....	13
2.3.3.2. Tổng tiền thu được trên mỗi danh mục mặt hàng .....	14
2.3.3.3. Số lượng đơn hàng được đặt qua từng tháng từ năm 2017 – 2021	17
2.4. Kết luận sau quá trình EDA.....	18
Chương 3: BÀI TOÁN VÀ HƯỚNG GIẢI QUYẾT .....	19

3.1. Đặt ra bài toán và hướng xử lý.....	19
3.2. Các thuật toán hồi quy.....	19
3.2.1. Giới thiệu .....	19
3.2.2. Linear Regression .....	20
3.2.3. Support Vector Regression .....	20
3.3. Bài toán 1: Làm thế nào để lấy được giá bán lẻ tốt nhất.....	22
3.3.1. Huấn luyện mô hình.....	22
3.3.1.1. Linear Regression 2 biến.....	22
3.3.1.2. Linear Regression 1 biến.....	25
3.3.1.3. Support Vector Regression .....	27
3.3.2. Thực hiện dự đoán giá bán lẻ tốt nhất của một mặt hàng có giá bán sỉ là $x$ và tổng số lượng bán dự kiến là $y$ trong vòng 5 năm .....	29
3.3.2.1. Mô tả quá trình dự đoán .....	29
3.3.2.2. Kết quả dự đoán của giá bán sỉ 200 USD và tổng lượng bán 500 ....	31
3.3.2.3. Kết quả dự đoán của giá bán sỉ 1000 và tổng lượng bán 500 .....	31
3.3.3. So sánh giữa các mô hình huấn luyện .....	32
3.4. Bài toán 2: Có lợi thế về ngày giao hàng khi trở thành khách hàng có thứ hạng Bạch Kim hay không .....	33
KẾT LUẬN .....	35

## Danh mục bảng biểu

Bảng 1.1. Bảng mô tả cấu trúc của bộ dữ liệu orders.csv .....	2
Bảng 1.2. Bảng mô tả cấu trúc của bộ dữ liệu product-supplier.csv .....	3

## Danh mục hình ảnh

Hình 2.1. Kết quả trả về từ câu lệnh info() đối với bộ dữ liệu orders.csv .....	6
Hình 2.2. Kết quả trả về từ câu lệnh isna().sum() đối với bộ dữ liệu orders.csv .....	6
Hình 2.3. Kết quả trả về từ câu lệnh isna().sum() đối với bộ dữ liệu orders.csv .....	6
Hình 2.4. Kết quả trả về từ câu lệnh info() đối với bộ dữ liệu product-supplier.csv .....	7
Hình 2.5. Kết quả trả về từ câu lệnh isna().sum() đối với bộ dữ liệu product-supplier.csv ..	7
Hình 2.6. Kết quả trả về từ câu lệnh isnull().sum() đối với bộ dữ liệu product-supplier.csv	7
Hình 2.7. Kiểu dữ liệu ngày và tháng ban đầu .....	8
Hình 2.8. Kiểu dữ liệu ngày tháng sau khi được biến đổi để phù hợp cho việc phân tích ....	8
Hình 2.9. Lỗi không thống nhất dữ liệu trên cột Customer Status .....	9
Hình 2.10. Thông nhất kiểu dữ liệu trên cột Customer Status.....	9
Hình 2.11. Thêm một cột dữ liệu mới tên là Item Retail Value .....	9
Hình 2.12. Tạo dataframe mới nhằm thống kê số lượng sản phẩm được bán cũng như giá trị vốn - lãi trung bình của sản phẩm.....	10
Hình 2.13. Thông tin của bảng dữ liệu được merge lại với nhau .....	10
Hình 2.14. Dữ liệu của bảng dữ liệu được merge lại với nhau.....	11
Hình 2.15. Bộ dữ liệu cuối cùng trong quá trình tiền xử lý .....	11
Hình 2.16. Phân bố sản phẩm dựa trên giá bán lẻ và giá bán sỉ .....	13
Hình 2.17. Mặt hàng có tỷ suất lợi nhuận cao nhất .....	14
Hình 2.18. Biểu đồ trực quan hóa tổng tiền thu được trên các danh mục sản phẩm .....	14
Hình 2.19. Biểu đồ trực quan hóa số tiền thu được dựa trên khoảng mức giá bán sỉ.....	15

Hình 2.20. Mô tả thống kê cơ bản trong cột Wholesale Price .....	16
Hình 2.21. Biểu đồ trực quan hóa số tiền thu được dựa trên khoảng mức giá bán lẻ.....	16
Hình 2.22. Mô tả thống kê cơ bản trong cột Retail Price .....	17
Hình 2.23. Mối tương quan giữa số lượng đặt hàng .....	17
Hình 3.1. Minh hoạ phương pháp SVM .....	21
Hình 3.2. Chọn biến độc lập và biến phụ thuộc LNR-2 .....	22
Hình 3.3. Sử dụng hàm train_test_split để sinh các x_train, y_train, x_test và y_test trong LNR-2 .....	22
Hình 3.4. Bắt đầu quá trình huấn luyện mô hình LNR-2.....	23
Hình 3.5. Dự đoán kết quả LNR-2.....	23
Hình 3.6. Trực quan hóa kết quả dự đoán LNR-2 .....	23
Hình 3.7. Đánh giá chỉ tiêu và điểm số hồi quy LNR-2 .....	24
Hình 3.8. Trực quan hóa phân bố dư lượng LNR-2 .....	24
Hình 3.9. Kết quả hệ số của LNR-2.....	25
Hình 3.10. Chọn biến độc lập và biến phụ thuộc LNR-1 .....	25
Hình 3.11. Sử dụng hàm train_test_split để sinh các x_train, y_train, x_test và y_test trong LNR-1 .....	26
Hình 3.12. Trực quan hóa kết quả dự đoán LNR-1 .....	26
Hình 3.13. Trực quan hóa phân bố dư lượng LNR-1.....	26
Hình 3.14. Đánh giá chỉ tiêu và điểm số hồi quy LNR-1 .....	27
Hình 3.15. Kết quả hệ số LNR-1 .....	27
Hình 3.16. Biến độc lập, biến phụ thuộc và tham số train_test_split của SVR .....	27
Hình 3.17. Xác định các tham số của từng kernel trong SVR.....	28
Hình 3.18. Trực quan hóa kết quả dự đoán của SVR .....	28
Hình 3.19. Điểm số hồi quy của SVR.....	29
Hình 3.20. Dự đoán giá bán lẻ tốt nhất dựa trên giá bán sỉ và số lượng bán.....	30
Hình 3.21. Nạp dữ liệu dự đoán vào các mô hình đã được huấn luyện.....	30

Hình 3.22. Kết quả giá bán lẻ tốt nhất dự đoán được trả về với trường hợp giá bán sỉ là 200 USD và tổng lượng bán là 500.....	31
Hình 3.23. Kết quả giá bán lẻ tốt nhất dự đoán được trả về với trường hợp giá bán sỉ là 1000 USD và tổng lượng bán là 500.....	31
Hình 3.24. Điểm số hồi quy giữa các mô hình huấn luyện.....	32
Hình 3.25. Dataframe mới phù hợp cho việc giải quyết bài toán số 2 .....	33
Hình 3.26. Cột Delay for Delivery trước khi chuyển thành số .....	33
Hình 3.27. Cột Delay for Delivery sau khi chuyển thành số .....	33
Hình 3.28. Thời gian trung bình chờ đợi của khách hàng theo từng loại khách hàng.....	34
Hình 3.29. Mô hình hoá dữ liệu của dataframe mới .....	34



# Chương 1: KHÁI QUÁT ĐỒ ÁN

## 1.1. Lí do chọn đề tài

Ngày nay, khi cuộc sống chúng ta ngày một phát triển thì việc mua bán, trao đổi hàng hoá ngày càng được chú trọng và quan tâm. Khách hàng ngày càng có nhiều sự lựa chọn, nhiều cơ hội để tìm cho mình một loại sản phẩm phù hợp nhất và giá cả lại phải chăng. Sự hiểu biết về hành vi tiêu dùng là chìa khóa cho một chiến lược marketing thành công cả trong nước và quốc tế. Những hoạt động marketing chủ yếu này sẽ hiệu quả hơn khi được đặt trên cơ sở một sự hiểu biết về hành vi tiêu dùng.

Xuất phát từ nhận thức trên, nhận thấy được tầm quan trọng của sự thay đổi hành vi của người tiêu dùng của trong thời điểm hiện tại, nên nhóm chúng em quyết định lựa chọn đề tài: "Phân tích các mẫu mua hàng trực tuyến để tìm hiểu các luật kết hợp giữa các mặt hàng" để nhận diện ra mặt hàng nào là tiềm năng, phát hiện những yếu tố quan trọng tác động đến hành vi mua sắm từ đó có những chiến lược Marketing phù hợp.

Mục tiêu của đồ án này sẽ:

Sử dụng dữ liệu về lịch sử mua hàng để phân tích và xác định:

- Mặt hàng nào có tỉ suất lợi nhuận cao nhất?
- Mặt hàng nào bán chạy nhất, mặt hàng nào không bán chạy.
- Mối tương quan giữa giá bán lẻ?

Sử dụng dữ liệu về thời gian để phân tích và xác định:

- Tần suất mua hàng của các thời điểm trong năm.
- Ngày giao hàng có thay đổi không khi trở thành khách hàng hạng Bạch kim?

Sử dụng dữ liệu về lịch sử thanh toán để phân tích và xác định giá bán sỉ – lẻ để rồi từ đó ta có thể dự đoán giá bán lẻ khi đã biết giá bán sỉ.

## 1.2. Mô tả dữ liệu và cấu trúc dữ liệu

Bộ dữ liệu được tổng hợp từ các nguồn sau Wholesale & Retail Orders Dataset (kaggle.com):

<https://www.kaggle.com/datasets/gabrielsantello/wholesale-and-retail-orders-dataset?select=orders.csv>

<https://www.kaggle.com/datasets/gabrielsantello/wholesale-and-retail-orders-dataset?select=product-supplier.csv>

Đây là những nguồn được đánh giá là có những bộ dữ liệu đáng tin cậy để phục vụ cho việc nghiên cứu hành vi tiêu dùng.

*Bảng 1.1. Bảng mô tả cấu trúc của bộ dữ liệu orders.csv*

Thuộc tính	Ý nghĩa
Customer ID	Mã khách hàng
Customer Status	Trạng thái khách hàng thân thiết (Ví dụ: Gold, Slive, Platinum)
Date Order was placed	Ngày đặt hàng
Delivery Date	Ngày giao hàng
Order ID	Mã đặt hàng
Product ID	Mã sản phẩm
Quantity Ordered	Số lượng sản phẩm đặt
Total Retail Price for This Order	Tổng giá bán lẻ cho đơn hàng này
Cost Price Per Unit	Giá vốn của mỗi món hàng

*Bảng 1.2. Bảng mô tả cấu trúc của bộ dữ liệu product-supplier.csv*

Thuộc tính	Ý nghĩa
Product ID	Mã sản phẩm
Product Line	Dòng sản phẩm
Product Category	Danh mục sản phẩm
Product Group	Nhóm sản phẩm
Product Name	Tên sản phẩm
Supplier Country	Nước phân phối
Supplier Name	Tên nhà phân phối
Supplier ID	Mã nhà phân phối

## Chương 2: KHẢO SÁT VÀ TIỀN XỬ LÝ BỘ DỮ LIỆU

### 2.1. Khảo sát bộ dữ liệu

Trong đề án môn học này, chúng em sử dụng bộ dữ liệu trong hoạt động mua bán sỉ lẻ từ năm 2017 đến năm 2021, bộ dữ liệu phân tích gồm 2 file csv tương ứng:

➤ Bộ dữ liệu orders.csv (gồm 185004 bản ghi và 9 trường thuộc tính):

Đây là bộ dữ liệu chứa thông tin về lịch sử giao dịch về mua bán sản phẩm đối với khách hàng và người bán. Trong đó, mỗi bản ghi (record) trong bộ dữ liệu tượng trưng cho việc thông tin sản phẩm trong quá trình mua bán như mã sản phẩm, mã đặt hàng, mã khách hàng, mức giá, ngày đặt, ngày giao, số lượng giao dịch, tổng vốn thu được, giá trị mỗi sản phẩm.

Các thuộc tính trong bộ dữ liệu được mô tả bao như sau:

- **Customer ID:** Mã khách hàng.
- **Customer Status:** Trạng thái khách hàng thân thiết.
- **Date Order was placed:** Ngày đặt hàng.
- **Delivery Date:** Ngày giao hàng.
- **Order ID:** Mã đặt hàng.
- **Product ID:** Mã sản phẩm.
- **Quantity Ordered:** Số lượng sản phẩm được đặt hàng.
- **Total Retail Price for This Order:** Tổng giá bán lẻ cho cả đơn hàng.
- **Cost Price Per Unit:** Giá vốn thu được của mỗi món hàng.

⇒ Qua bộ dữ liệu trên, nhóm em có một vài nhận định như sau: Đối với mỗi bản ghi trong bộ dữ liệu chỉ đánh giá cụ thể đối với một sản phẩm thông qua mã sản phẩm và bộ dữ liệu này cũng là một bộ dữ liệu đầy đủ không thiếu hay NULL dòng/trường nào.

➤ Bộ dữ liệu *product-supplier.csv* (gồm 5505 bản ghi và 8 trường thuộc tính):

Đây là bộ dữ liệu mô tả thông tin chi tiết về các sản phẩm được mua bán. Trong đó, mỗi bản ghi (record) thể hiện thông tin chi tiết về từng thuộc tính của sản phẩm như mã sản phẩm, tên sản phẩm, nhóm sản phẩm, phân loại, dòng sản phẩm, nước phân phối, tên nhà phân phối, mã nhà phân phối.

Các thuộc tính trong bộ dữ liệu được mô tả như sau:

- **Product ID:** Mã sản phẩm
- **Product Line:** Dòng sản phẩm
- **Product Category:** Danh mục sản phẩm
- **Product Group:** Nhóm sản phẩm
- **Product Name:** Tên sản phẩm
- **Supplier Country:** Nước phân phối
- **Supplier Name:** Tên nhà phân phối
- **Supplier ID:** Mã nhà phân phối

⇒ Qua bộ dữ liệu trên, nhóm em có một vài nhận định như sau: Bộ dữ liệu trên mô tả rất đầy đủ thông tin chi tiết của từng đối tượng sản phẩm.

Đánh giá tổng thể cho thấy đây là hai bộ dữ liệu khá đầy đủ cho việc phân tích dữ liệu. Ngoài ra, hướng đi chính cho quá trình phân tích dữ liệu trong đồ án môn học này sẽ là tìm ra các bộ luật nhằm xác định các mối liên hệ giữa sản phẩm và người dùng.

## 2.2. Tiền xử lý dữ liệu

### 2.2.1. Kiểm tra bộ dữ liệu

Trước khi tiền xử lý dữ liệu, nhóm chúng em tiến hành kiểm tra bộ dữ liệu một cách tổng quan nhất. Bắt đầu với bộ dữ liệu *orders.csv* trước tiên ta đọc bộ dữ liệu và gán vào biến dataset bằng lệnh *info()*. Kết quả trả về thu được sẽ là:

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 185013 entries, 0 to 185012
Data columns (total 9 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Customer ID                          185013 non-null int64
1   Customer Status                      185013 non-null object
2   Date Order was placed                185013 non-null object
3   Delivery Date                       185013 non-null object
4   Order ID                            185013 non-null int64
5   Product ID                          185013 non-null int64
6   Quantity Ordered                    185013 non-null int64
7   Total Retail Price for This Order    185013 non-null float64
8   Cost Price Per Unit                  185013 non-null float64
dtypes: float64(2), int64(4), object(3)

```

Hình 2.1. Kết quả trả về từ câu lệnh `info()` đối với bộ dữ liệu `orders.csv`

Điều đó cho thấy kiểu dữ liệu tổng thể của toàn bộ bộ dữ liệu trên. Tiếp sau đó là kiểm tra xem bộ dữ liệu trên có chứa tập Null và NaN hay không bằng cách đếm số lượng của chúng chứa trong bộ dữ liệu bằng lệnh `isna().sum()` và `isnull().sum()`, sau đó kết quả trả về sẽ là:

```

Customer ID          0
Customer Status      0
Date Order was placed 0
Delivery Date        0
Order ID             0
Product ID           0
Quantity Ordered     0
Total Retail Price for This Order 0
Cost Price Per Unit  0
dtype: int64

```

Hình 2.2. Kết quả trả về từ câu lệnh `isna().sum()` đối với bộ dữ liệu `orders.csv`

```

Customer ID          0
Customer Status      0
Date Order was placed 0
Delivery Date        0
Order ID             0
Product ID           0
Quantity Ordered     0
Total Retail Price for This Order 0
Cost Price Per Unit  0
dtype: int64

```

Hình 2.3. Kết quả trả về từ câu lệnh `isna().sum()` đối với bộ dữ liệu `orders.csv`

Tiếp sau đó ta vẫn tiến hành kiểm tra bộ dữ liệu như trên đối với bộ dữ liệu *product-supplier.csv* và kết quả thu được như sau:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5504 entries, 0 to 5503
Data columns (total 8 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Product ID            5504 non-null   int64
1   Product Line          5504 non-null   object
2   Product Category      5504 non-null   object
3   Product Group         5504 non-null   object
4   Product Name          5504 non-null   object
5   Supplier Country      5504 non-null   object
6   Supplier Name         5504 non-null   object
7   Supplier ID           5504 non-null   int64
dtypes: int64(2), object(6)
```

Hình 2.4. Kết quả trả về từ câu lệnh `info()` đối với bộ dữ liệu *product-supplier.csv*

```
Product ID      0
Product Line    0
Product Category 0
Product Group   0
Product Name    0
Supplier Country 0
Supplier Name    0
Supplier ID     0
dtype: int64
```

Hình 2.5. Kết quả trả về từ câu lệnh `isna().sum()` đối với bộ dữ liệu *product-supplier.csv*

```
Product ID      0
Product Line    0
Product Category 0
Product Group   0
Product Name    0
Supplier Country 0
Supplier Name    0
Supplier ID     0
dtype: int64
```

Hình 2.6. Kết quả trả về từ câu lệnh `isnull().sum()` đối với bộ dữ liệu *product-supplier.csv*

Kết quả trả về từ việc kiểm tra 2 bộ dữ liệu trên cho thấy không hề có một bản ghi nào bị trống dữ liệu.

### 2.2.2. Biến đổi và thêm mới dữ liệu

Để thuận tiện hơn trong quá trình phân tích dữ liệu, chúng em tiến hành biến đổi dữ liệu bằng cách thay đổi một số trường dữ liệu, thay đổi kiểu dữ liệu, tạo dữ liệu mới và kết hợp dữ liệu với nhau tạo ra một dataframe phù hợp cho quá trình phân tích.

Đầu tiên, đối với bộ dữ liệu *orders.csv*, ta chuyển kiểu dữ liệu của cột *Date Order was placed* và *Delivery Date* về lại dạng ngày tháng bằng số hoàn chỉnh, thay vì kiểu dữ liệu viết tắt tháng như ban đầu.

	Customer ID	Customer Status	Date Order was placed	Delivery Date
0	579	Silver	01-Jan-17	07-Jan-17
1	7574	SILVER	01-Jan-17	05-Jan-17
2	28861	Gold	01-Jan-17	04-Jan-17
3	43796	Gold	01-Jan-17	06-Jan-17
4	54673	Gold	01-Jan-17	04-Jan-17

Hình 2.7. Kiểu dữ liệu ngày và tháng ban đầu

	Customer ID	Customer Status	Date Order was placed	Delivery Date
0	579	Silver	2017-01-01	2017-01-07
1	7574	Silver	2017-01-01	2017-01-05
2	28861	Gold	2017-01-01	2017-01-04
3	43796	Gold	2017-01-01	2017-01-06
4	54673	Gold	2017-01-01	2017-01-04

Hình 2.8. Kiểu dữ liệu ngày tháng sau khi được biến đổi để phù hợp cho việc phân tích

Trong quá trình kiểm tra lại bộ dữ liệu thì nhóm chúng em có nhận thấy một thiếu sót trong quá trình tổng hợp dữ liệu của bên cung cấp bộ dữ liệu, bằng cách sử dụng câu lệnh *unique()* đối với mỗi cột trong bộ dữ liệu, chúng em nhận thấy đối với cột *Customer Status* bị lỗi lặp dữ liệu trùng trong một số bản ghi. Ví dụ: Gold và GOLD, Silver và SILVER, một số lỗi viết hoa và viết thường ko thông nhất dẫn đến lỗi không khớp dữ liệu với nhau:



```
['Silver' 'SILVER' 'Gold' 'GOLD' 'PLATINUM' 'Platinum']
<StringArray>
```

Hình 2.9. Lỗi không thống nhất dữ liệu trên cột Customer Status

Để xử lý vấn đề này, chúng em sử dụng câu lệnh `str.lower()` và `str.capitalize()` để thống nhất lại kiểu dữ liệu chung trong cột này. Ngoài ra chúng em còn chuyển kiểu dữ liệu của cột từ *object* sang *string* để thuận tiện cho việc đọc – ghi dữ liệu:

```
['Silver', 'Gold', 'Platinum']
Length: 3, dtype: string
```

Hình 2.10. Thông nhất kiểu dữ liệu trên cột Customer Status

Ngoài ra, để thuận tiện hơn cho việc phân tích nhóm chúng em quyết định thêm mới dữ liệu bằng cách tạo một cột mới tên là *Item Retail Value* (giá bán lẻ trên từng sản phẩm) dựa trên dữ liệu đến từ cột *Total Retail Price for This Order* (Tổng giá bán lẻ trên order này) và *Quantity Order* (Số lượng sản phẩm) theo công thức:

$$\text{Item Retail Value} = \frac{\text{Total Retail Price for This Order}}{\text{Quantity Order}}$$

	Customer ID	Customer Status	Date Order was placed	Delivery Date	Order ID	Product ID	Quantity Ordered	Total Retail Price for This Order	Cost Price Per Unit	Item Retail Value
0	579	Silver	2017-01-01	2017-01-07	123002578	220101400106	2	92.6	20.70	46.3
1	7574	Silver	2017-01-01	2017-01-05	123004074	210201000009	1	21.7	9.95	21.7
2	28861	Gold	2017-01-01	2017-01-04	123000871	230100500068	1	1.7	0.80	1.7
3	43796	Gold	2017-01-01	2017-01-06	123002851	220100100633	1	47.9	24.05	47.9
4	54673	Gold	2017-01-01	2017-01-04	123003607	220200200043	1	36.9	18.30	36.9

Hình 2.11. Thêm một cột dữ liệu mới tên là Item Retail Value

Sau cùng, bộ dữ liệu này nhóm chúng em đặt tên cho nó là **OrderDf**, tương trưng cho lịch sử bán của từng sản phẩm. Bộ dữ liệu này sẽ là tiền đề để nhóm chúng em thực hiện các quá trình khảo sát và phân tích dữ liệu sau

Bên cạnh đó, để có thể biết được số lượng được bán ra của từng sản phẩm cũng như là giá trị vốn lẫn lãi trung bình thu được của sản phẩm đó, nhóm chúng em tiến hành tạo mới một dataframe dựa trên bộ dữ liệu *orders.csv* đã được tiền xử lý trước như sau: Đầu tiên nhóm (group) các mã sản phẩm với nhau (nhằm tạo ra một bản ghi về sản phẩm là duy nhất trong dataframe), song đếm số lượng sản phẩm được group lại, bên cạnh đó tính toán lại giá trị trung bình trên từng sản phẩm của các cột như *Cost Price Per Unit* (Giá vốn trên từng sản phẩm) và *Item Retail Value* (giá bán

lẻ trên từng sản phẩm) sao cho phù hợp với lại từng sản phẩm đã được nhóm (group) lại với nhau (Hình 2.12).

	Product ID	N Rows	Cost Price Per Unit	Item Retail Value
0	210200100001	56	5.650	12.80
1	210200100003	60	15.585	34.87
2	210200100004	95	16.150	36.30
3	210200100005	227	5.750	13.00
4	210200100006	16	5.150	11.60

Hình 2.12. Tạo dataframe mới nhằm thống kê số lượng sản phẩm được bán cũng như giá trị vốn - lãi trung bình của sản phẩm

### 2.2.3. Bộ dữ liệu tổng thể

Sau quá trình tiền xử lý ở phần trên, dữ liệu chúng ta thu được ở bước này là bộ dữ liệu dataframe về số lượng, giá vốn – lãi trung bình trên từng mặt hàng, do đó cái mà nhóm chúng em thiếu để có thể phân tích đặc tính của mặt hàng đó chính là thông tin tổng thể của từng mặt hàng. Thông tin tổng thể của từng mặt hàng được chứa trong bộ dữ liệu *product-supplier.csv*, cho nên nhóm chúng em quyết định trộn (merge) dữ liệu từ 2 dataframe đó lại với nhau nhằm mục đích tạo ra một bộ dữ liệu hoàn chỉnh hơn.

Kết hợp dữ liệu từ 2 bộ dữ liệu với nhau khác giống với việc join 2 table lại với nhau trong SQL, bằng cách sử dụng câu lệnh *merge()* với tham số điều kiện kết hợp bảng là *Product ID* thì ta có thể trộn dữ liệu từ 2 bảng lại với nhau. Dữ liệu được kết hợp lại sẽ có mô tả như sau:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3124 entries, 0 to 3123
Data columns (total 11 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Product ID            3124 non-null   int64
1   N Rows                3124 non-null   int64
2   Cost Price Per Unit   3124 non-null   float64
3   Item Retail Value     3124 non-null   float64
4   Product Line          3124 non-null   object
5   Product Category      3124 non-null   object
6   Product Group         3124 non-null   object
7   Product Name          3124 non-null   object
8   Supplier Country      3124 non-null   object
9   Supplier Name         3124 non-null   object
10  Supplier ID           3124 non-null   int64
dtypes: float64(2), int64(3), object(6)
```

Hình 2.13. Thông tin của bảng dữ liệu được merge lại với nhau

Và bộ dữ liệu sẽ có dạng như sau:

	Product ID	N Rows	Cost Price Per Unit	Item Retail Value	Product Line	Product Category	Product Group	Product Name	Supplier Country	Supplier Name	Supplier ID
0	210200100001	56	5.650	12.80	Children	Children Sports	A-Team, Kids	A-team Children's Shorts w/Pockets	US	A Team Sports	3298
1	210200100003	60	15.585	34.87	Children	Children Sports	A-Team, Kids	A-team Children's Sweat R.Neck Big Logo	US	A Team Sports	3298
2	210200100004	95	16.150	36.30	Children	Children Sports	A-Team, Kids	A-team Children's Sweat w/Hood,Big Logo	US	A Team Sports	3298
3	210200100005	227	5.750	13.00	Children	Children Sports	A-Team, Kids	A-team Children's T-Shirt	US	A Team Sports	3298
4	210200100006	16	5.150	11.60	Children	Children Sports	A-Team, Kids	A-team Children's T-Shirt w/Big Logo	US	A Team Sports	3298

Hình 2.14. Dữ liệu của bảng dữ liệu được merge lại với nhau

Tuy vậy, dữ liệu bên trong khá là nhiều và có thể gây tốn kém tài nguyên trong quá trình phân tích. Nhóm chúng em quyết định lược bỏ đi những cột không cần thiết như *Product Line*; *Product Group*; *Supplier Country*; *Supplier Name* và *Supplier ID*. Bên cạnh đó, đổi tên các cột sao cho dễ hiểu và phù hợp hơn như *N Rows* đổi thành *Total Sold* (Tổng sản phẩm bán được); *Cost Price Per Unit* đổi thành *Wholesale Price* (Giá sỉ); *Item Retail Value* đổi thành *Retail Price* (Giá lẻ).

	Product ID	Total Sold	Wholesale Price	Retail Price	Product Category	Product Name
0	210200100001	56	5.650	12.80	Children Sports	A-team Children's Shorts w/Pockets
1	210200100003	60	15.585	34.87	Children Sports	A-team Children's Sweat R.Neck Big Logo
2	210200100004	95	16.150	36.30	Children Sports	A-team Children's Sweat w/Hood,Big Logo
3	210200100005	227	5.750	13.00	Children Sports	A-team Children's T-Shirt
4	210200100006	16	5.150	11.60	Children Sports	A-team Children's T-Shirt w/Big Logo

Hình 2.15. Bộ dữ liệu cuối cùng trong quá trình tiền xử lý

Tóm lại, qua một chuỗi tiền xử lý trên, từ 2 bộ dữ liệu về lịch sử mua bán và thông tin mặt hàng, nhóm chúng em đã tóm gọn và xử lý thành một bộ dữ liệu hoàn chỉnh và đầy đủ về thông tin được bán ra như số lượng, giá sỉ-lẻ của từng mặt hàng. Ta gọi bộ dữ liệu này là **ProductDf**, tượng trưng cho thông tin bán ra và giá sỉ - lẻ của từng mặt hàng sản phẩm cụ thể và các thông tin khác của mặt hàng.

Do đó, output cuối cùng sau quá trình tiền xử lý ta sẽ thu được gồm 2 bộ dữ liệu là **OrderDf** và **ProductDf**. Bộ liệu thu được qua quá trình tiền xử lý trên sẽ chính là tiền đề để nhóm chúng em phân tích sâu hơn trong những phần sau.

### 2.3. Thăm dò dữ liệu sau tiền xử lý

Đối với một người làm về phân tích dữ liệu, không dễ để nhìn vào một cột số cụ thể hay toàn bộ bảng dữ liệu để xác định các đặc điểm quan trọng của dữ liệu, nếu thực hiện bằng cách thức thủ công, sẽ mất rất nhiều thời gian và mức độ hiệu quả không được đảm bảo. Vì vậy, *EDA – Phân tích khám phá dữ liệu (Exploratory Data*

*Analyst*) sẽ là một giải pháp phù hợp dành cho các nhà phân tích dữ liệu. Vậy EDA là gì? Mục đích của việc sử dụng EDA như thế nào?

### **2.3.1. Khái niệm về EDA**

Phân tích dữ liệu thăm dò là quá trình mô tả dữ liệu bằng các kỹ thuật thống kê và trực quan hoá nhằm tập trung vào các khía cạnh quan trọng của dữ liệu để tiếp tục phân tích. Điều này bao gồm cả việc kiểm tra tập dữ liệu từ nhiều góc độ, mô tả và tóm tắt nó mà không đưa ra bất kỳ giả định nào khác về nội dung của nó. Trong đề tài của nhóm chúng em sẽ tập trung hiển thị các đặc trưng của dữ liệu trong bộ dữ liệu đã tiền xử lý ở phần trước.

### **2.3.2. Mục đích sử dụng EDA**

Một số mục đích của việc sử dụng EDA vào các dự án phân tích dữ liệu như:

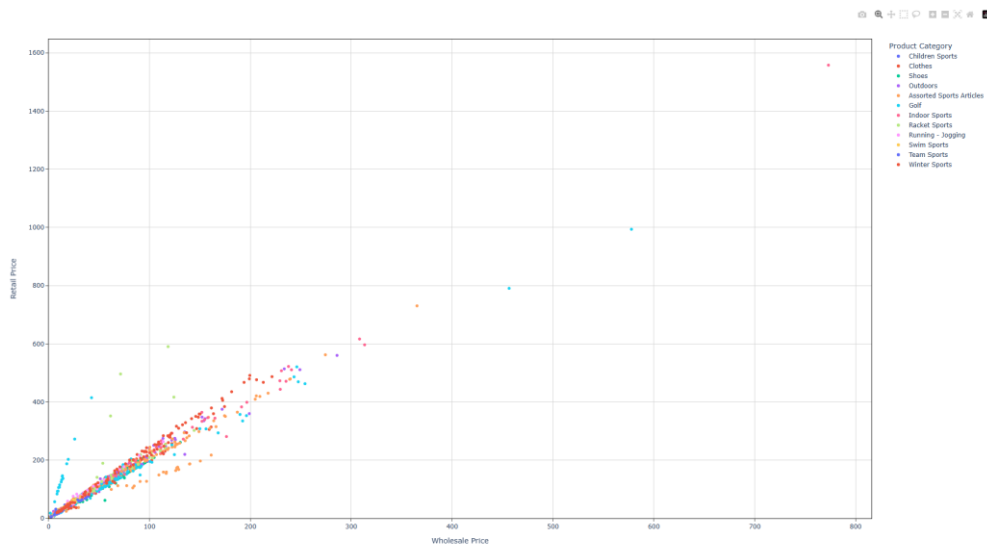
- Tìm hiểu về cấu trúc dữ liệu: EDA là phương pháp giúp xác định cấu trúc dữ liệu bao gồm số lượng, kiểu dữ liệu, trường dữ liệu, sự liên kết giữa các trường dữ liệu,... Khi xác định được cấu trúc dữ liệu, các nhà phân tích dữ liệu có thể hiểu được mối quan hệ giữa các dữ liệu trong tập.
- Điều chỉnh và thay đổi: EDA giúp giải quyết các trường hợp thiếu giá trị, dữ liệu lỗi, các ngoại lệ trong dữ liệu. Điều này giúp các nhà phân tích dữ liệu điều chỉnh các phương án khắc phục kịp thời, tránh những ảnh hưởng nghiêm trọng đến dự án.
- Xác định mối tương quan giữa các biến: Các biến đều chứa các giá trị riêng, EDA có khả năng phát hiện các liên hệ tiềm ẩn và sự ảnh hưởng giữa các biến với nhau, tạo sự liên kết giữa các thông tin dữ liệu nhằm xây dựng một quy trình phân tích tổng thể, rõ ràng.
- Xây dựng cơ sở dữ liệu quan hệ: Các đối tượng dữ liệu quan trọng được phát triển mối quan hệ nhằm cấu trúc hóa dữ liệu theo sơ đồ, tiết kiệm thời gian xử lý những thông tin thừa, hạn chế sự sai sót của kết quả phân tích.

- Chuẩn bị cho bước phân tích tiếp theo: Áp dụng EDA giúp loại bỏ các dữ liệu không cần thiết, dữ liệu thiếu giá trị và chuẩn hóa dữ liệu. Đây là yếu tố nền tảng để chuẩn bị cho các bước phân tích bằng thuật toán học máy.

### 2.3.3. Thực nghiệm EDA đối với dữ liệu phân tích

Để có một cái nhìn bao quát hơn về dữ liệu mà nhóm chúng em đang thực nghiệm phân tích, nhóm chúng em tiến hành phân tích và tổng hợp dữ liệu một lần dưới dạng mô hình nữa nhằm đưa ra cái nhìn trực quan hơn về dữ liệu mà chúng em đang làm việc.

#### 2.3.3.1. Tỷ suất lợi nhuận của mặt hàng sản phẩm



Hình 2.16. Phân bố sản phẩm dựa trên giá bán lẻ và giá bán sỉ

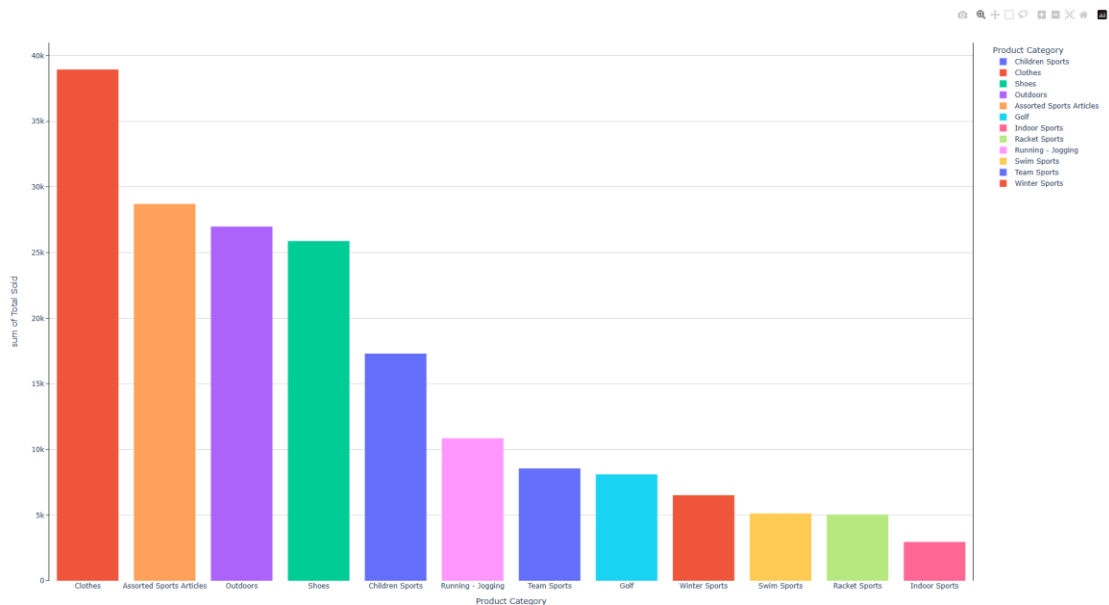
Như hình trên (hình 2.16), ta có trực quan hóa toàn bộ sản phẩm dựa trên giá bán lẻ và giá bán sỉ. Do đó, ta có thể thấy các mặt hàng được phân bố và được đánh màu dựa trên danh mục sản phẩm của chúng. Theo như khảo sát, phần lớn số lượng sản phẩm đều có giá bán lẻ dao động từ 600 USD trở xuống và giá bán sỉ dao động từ 250 USD trở xuống. Ngoài ra, trong quá trình khảo sát của mình, nhóm chúng em đã phát hiện ra rằng danh mục *golf* là danh mục có tỷ suất lợi nhuận cao nhất với 16 sản phẩm.

Ngoài ra dựa trên mô hình trực quan hóa trên ta có thể biết được tỷ suất lợi nhuận của từng sản phẩm dựa trên đường chéo từ trái dưới lên phải trên cùng. Trong đó, sản phẩm có tỷ suất lợi nhuận cao nhất có tên là “*Top-form 325 Treadmill*” với mức giá bán lẻ (Retail Price) là 773 USD và giá bán lẻ (Wholesale Price) là 1557 USD (Hình 2.18).



Hình 2.17. Mặt hàng có tỷ suất lợi nhuận cao nhất

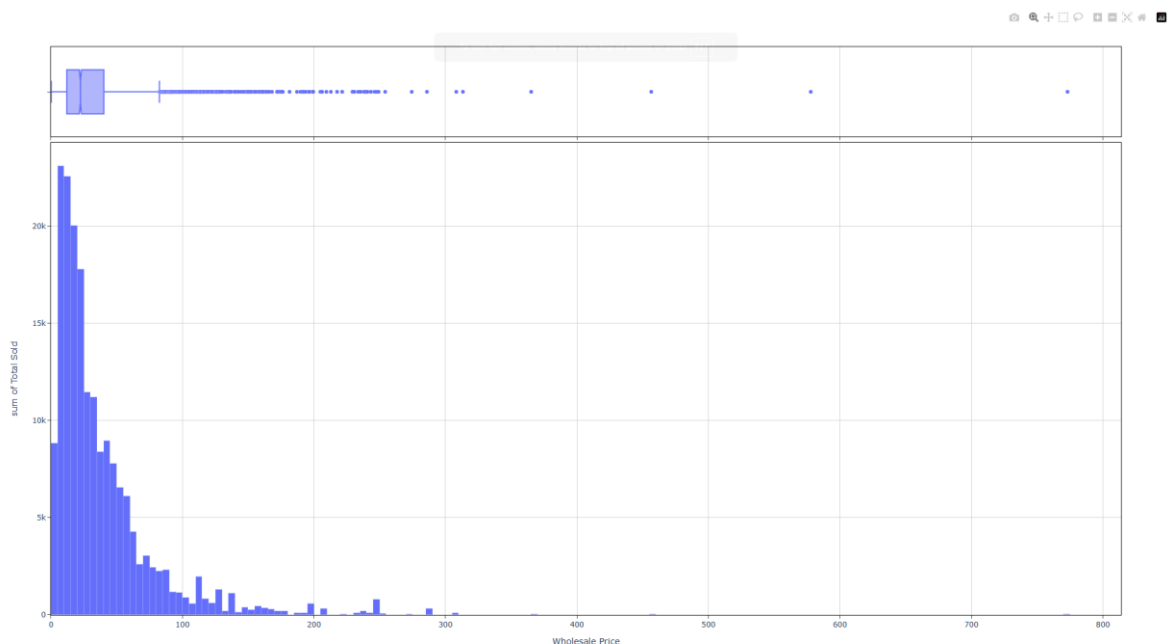
### 2.3.3.2. Tổng tiền thu được trên mỗi danh mục mặt hàng



Hình 2.18. Biểu đồ trực quan hóa tổng tiền thu được trên các danh mục sản phẩm

Biểu đồ trực quan hóa trên cho biết tổng tiền thu được từ các danh mục mặt hàng. Trong đó cho biết danh mục quần áo (Clothes) là danh mục chiếm tỷ trọng tổng thu nhập là lớn nhất và danh mục quần áo thể thao trong nhà (Indoor Sports) chiếm tỷ trọng bé nhất trong tất cả danh mục sản phẩm.

- Tổng tiền thu được từ danh mục quần áo (Clothes) là 38.953 USD
- Tổng tiền thu được từ danh mục quần áo thể thao trong nhà là 2995 USD
- ❖ *Số lượng sản phẩm bán được dao động trong các khoảng mức giá khác nhau*
  - *Bán sỉ*



Hình 2.19. Biểu đồ trực quan hóa số tiền thu được dựa trên khoảng mức giá bán sỉ

Biểu đồ trực quan hóa trên cho biết tổng số tiền thu được từ các sản phẩm có các khoảng mức giá bán sỉ khác nhau. Ở biểu đồ trên thì các sản phẩm có mức giá từ 5 – 10 USD có tổng số lượng bán được 23 ngàn sản phẩm, chiếm tỷ trọng lớn nhất trong toàn bộ sản bộ sản phẩm. Do đó, ta có thể ngầm kết luận rằng ở các khoảng mức giá bán sỉ từ 5 – 10 USD chiếm phần lớn người mua nhất và khoảng mức giá từ 770 – 775 USD thì chiếm số lượng người mua là ít nhất.

Ngoài ra, bảng nhỏ phía trên cũng mô tả thông tin các sản phẩm nằm trong khoảng mức giá ấy. Bên cạnh đó, ta cũng thu được một số thống kê cơ bản của trong cột Wholesale Price như sau:

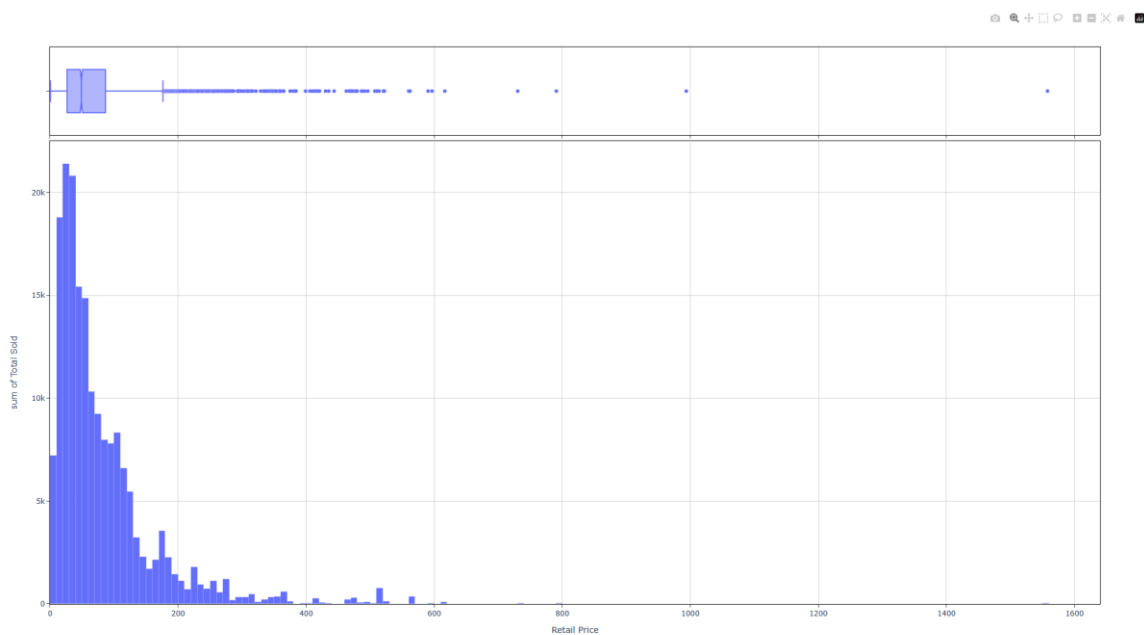
```

count    3124.000000
mean      34.103967
std       40.766120
min        0.200000
25%       11.996711
50%       22.368443
75%       40.180476
max       773.089744
Name: Wholesale Price, dtype: float64

```

Hình 2.20. Mô tả thống kê cơ bản trong cột Wholesale Price

### ○ Bán lẻ



Hình 2.21. Biểu đồ trực quan hóa số tiền thu được dựa trên khoảng mức giá bán lẻ

Biểu đồ trên cho biết tổng tiền thu được từ các sản phẩm có các khoảng mức giá bán lẻ khác nhau. Ở biểu đồ trên thì các sản phẩm có mức giá từ 20 – 29.99 USD có tổng số lượng bán được 21.387 sản phẩm, chiếm tỷ trọng lớn nhất trong toàn bộ sản phẩm. Do đó, ta có thể ngầm kết luận rằng ở các khoảng mức giá bán lẻ từ 20 – 29.99 USD chiếm phần lớn người mua nhất và khoảng mức giá từ 1555 – 1559.99 USD thì chiếm số lượng người mua là ít nhất.

Ngoài ra, bảng nhỏ phía trên cũng mô tả thông tin các sản phẩm nằm trong khoảng mức giá ấy. Bên cạnh đó, ta cũng thu được một số thống kê cơ bản của trong cột Retail Price như sau:



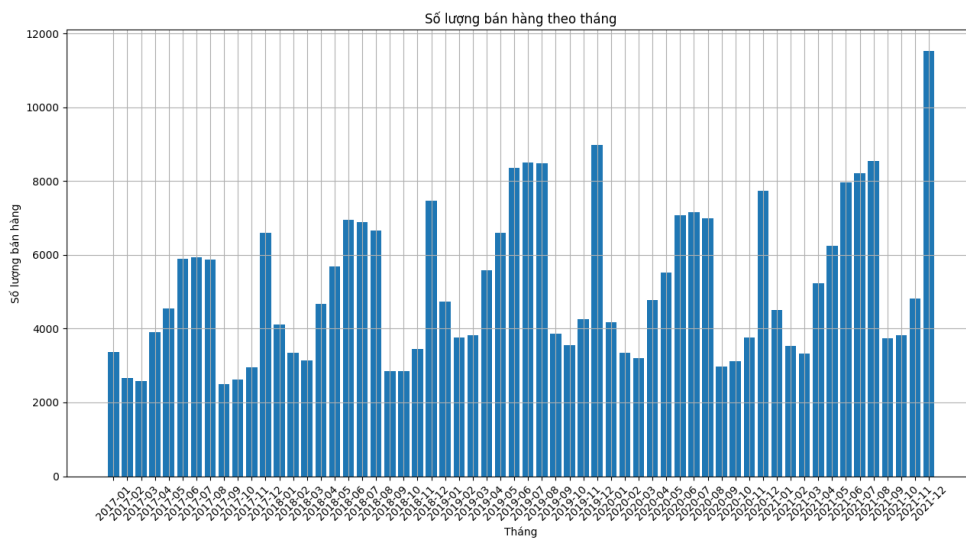
```

count    3124.000000
mean      73.575585
std       84.644969
min        0.625000
25%       26.658506
50%       48.963529
75%       86.635305
max      1557.833333
Name: Retail Price, dtype: float64

```

Hình 2.22. Mô tả thống kê cơ bản trong cột Retail Price

### 2.3.3.3. Số lượng đơn hàng được đặt qua từng tháng từ năm 2017 – 2021



Hình 2.23. Mối tương quan giữa số lượng đơn hàng và các tháng trong năm

Qua biểu đồ trên cho thấy sức mua sắm được duy trì ở mức ổn định và có xu hướng tăng dần qua các năm. Tuy nhiên từ cuối năm 2019 đến cuối năm 2021 là khoảng thời gian cả thế giới phải hứng chịu đại dịch Covid-19 nên sức mua hàng ở khoảng thời gian này có phần chững lại. Nhưng sau đại dịch, tốc độ tăng trưởng lại phục hồi một cách chóng mặt có thể thấy rõ nhất vào tháng 11 năm 2021, số lượng đơn hàng đạt trên 10000 đơn.

Trung bình ở các tháng 5, 6, 7 và 11 là khoảng thời gian có số lượng đơn hàng nhiều nhất trong năm. Nguyên do là thời gian tháng 5, 6 và 7 là rơi vào lúc học sinh được nghỉ hè nên các gia đình có xu hướng mua sắm cá sản phẩm như *Clothes*, *Assorted Sports Artcles*, *Outdoors*, *Shoes*, *Children Sports* để phục vụ việc vui chơi trong hè (hình 2.18). Tháng 11 hằng năm là mùa mua sắm cuối năm nhằm kích thích

sự tăng trưởng trở lại của nền kinh tế thông qua ngày đặc biệt là “Black Friday” nên sức mua hàng khi ấy luôn ở mức cao nhất trong năm.

## **2.4. Kết luận sau quá trình EDA**

Kết quả thu được sau khi áp dụng phương pháp EDA như sau:

- Danh mục Golf có sản phẩm có tỷ suất lợi nhuận cao nhất với 16 sản phẩm.
- Danh mục quần áo (Clothes) là danh mục bán chạy nhất. (Hình 2.18)
- Danh mục thể thao trong nhà (Indoor Sport) là danh mục bán ít nhất. (Hình 2.18)
- Hầu hết giá các mặt hàng đều dưới 50, dao động trên cả 2 loại mặt hàng bán lẻ và bán sỉ. (Hình 2.19 và 2.21)
- Giá bán lẻ thường cao hơn nhiều so với giá bán sỉ.
- Sức mua hàng luôn ở mức ổn định qua các năm và có xu hướng tăng lên. Đặc biệt tăng nhiều vào tháng 5, 6, 7 và 11.

## **Chương 3: BÀI TOÁN VÀ HƯỚNG GIẢI QUYẾT**

### **3.1. Đặt ra bài toán và hướng xử lý**

Đối với dữ liệu mà ta đã tiền xử lý trên, ta đã thu được một bộ dữ liệu chứa các thông tin bán ra và giá sỉ - lẻ của từng mặt hàng sản phẩm cụ thể và một bộ dữ liệu chứa lịch sử bán hàng của sản phẩm. Với bộ dữ liệu của mình, nhóm chúng em đặt ra 2 bài toán cần được giải quyết như sau:

- Bài toán 1: Làm thế nào để có được giá bán lẻ tốt nhất dựa trên giá bán buôn.
- Bài toán 2: Có lợi thế về ngày giao hàng khi trở thành khách hàng có thứ hạng bạch kim hay không.

Đối với bài toán 1, nhóm chúng em có hướng giải quyết là sử dụng khả năng huấn luyện mô hình và đưa ra dự đoán của 2 thuật toán hồi quy là Linear Regression (LNR) và Suport Vector Regression.

Đối với bài toán 2, ta có thể thực hiện bằng cách gom cụm dữ liệu và tính toán sao cho ra được kết quả mong muốn và từ đó thu ra được kết quả.

### **3.2. Các thuật toán hồi quy**

#### **3.2.1. Giới thiệu**

Phân tích hồi quy là phương pháp thống kê được phát triển từ thế kỉ 19 và trải qua một quá trình lịch sử dài. Lịch sử đã chứng kiến sự phát triển liên tục của công nghệ và phần mềm thống kê. Phân tích hồi quy đã trở thành một công cụ mạnh mẽ trong nghiên cứu và ứng dụng thống kê trong nhiều lĩnh vực khác nhau. Hiện tại phân tích hồi quy vẫn là một kĩ thuật quan trọng được sử dụng để khám phá mối quan hệ giữa một biến phụ thuộc và một hay nhiều biến độc lập. Phân tích hồi quy thường được sử dụng để dự đoán, hiểu sự mạnh yếu và hướng của mối quan hệ, và xác định

các biến quan trọng trong một tập dữ liệu. Trong đề án này sẽ sử dụng hồi quy tuyến tính để phân tích dữ liệu.

### 3.2.2. Linear Regression

Linear Regression (Hồi quy tuyến tính) là một thuật toán học có giám sát (supervised learning) trong Machine Learning<sup>1</sup>. Nó là một phương pháp thống kê dùng để ước lượng mối quan hệ giữa các biến độc lập (input features) và biến phụ thuộc (output target).

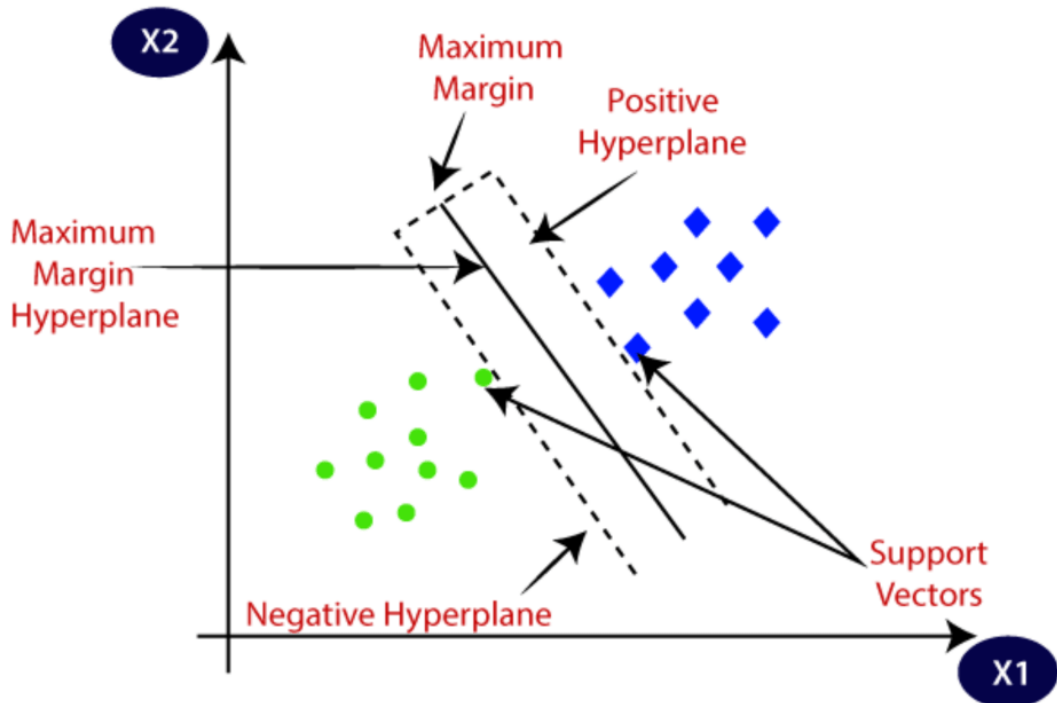
Trong Linear Regression, chúng ta sẽ gặp hai loại bài toán đó là Hồi quy đơn biến và Hồi quy đa biến. Hồi quy tuyến tính đơn biến là mối quan hệ giữa hai biến số liên tục trên trục hoành  $x$  và trên trục tung  $y$ . Phương trình hồi quy tuyến tính đơn biến có dạng như phương trình đường thẳng  $y = ax + b$  với  $x$  là biến độc lập và  $y$  là biến phụ thuộc vào  $x$ . Đối với Hồi quy tuyến tính đa biến, ta có thể hiểu một cách đơn giản là sẽ có nhiều biến độc lập  $x_1, x_2, \dots, x_n$  và nhiều hệ số  $a_1, a_2, \dots, a_n$  thay vì chỉ một biến  $x$  duy nhất.

Thuật toán này thích hợp để dự đoán các giá trị đầu ra là các đại lượng liên tục như doanh số hay giá cả thay vì cố gắng phân loại chúng thành các đại lượng rời rạc như màu sắc và chất liệu của quần áo, hay xác định đối tượng trong một bức ảnh là mèo hay chó

### 3.2.3. Support Vector Regression

Support Vector Machine (SVM) là một thuật toán có giám sát, mô hình nhận dữ liệu đầu vào và xem chúng như những vector trong không gian sau đó phân chia chúng vào các lớp khác nhau bằng cách xây dựng siêu phẳng trong không gian nhiều chiều làm mặt phân cách các lớp dữ liệu. Để có được kết quả phân lớp tối ưu thì phải xác định siêu phẳng (hyperplane) có khoảng cách đến các điểm dữ liệu (margin) của tất cả các lớp xa nhất có thể. SVM có khả năng phân lớp nhanh và tiết kiệm bộ nhớ. Tuy nhiên đối mặt với kho dữ liệu lớn hay số chiều lớn hơn số mẫu dữ liệu huấn

luyện thì trở nên kém hiệu quả, nhạy cảm với nhiễu hoặc thiếu thông tin xác suất phân lớp.



Hình 3.1. Minh họa phương pháp SVM

Trong đề án nhóm sử dụng thuật toán Support Vector Regression (SVR) để tiên đoán dữ liệu sau khi huấn luyện mô hình. Thuật toán này sử dụng cơ chế hồi quy của mô hình SVM. Ba hàm kernel thông dụng sẽ áp dụng vào thuật toán SVR đó là:

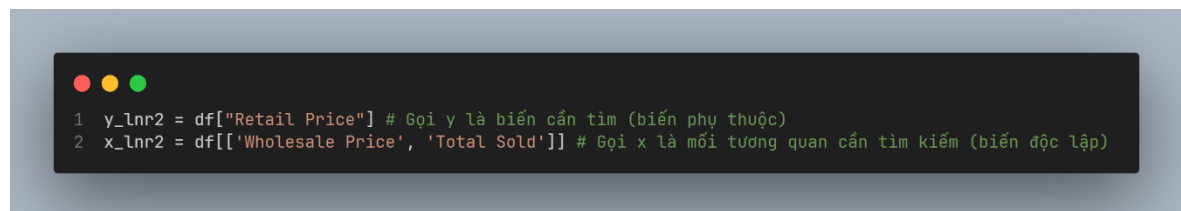
- Mô hình kernel RBF (Radial Basis Function) là một mô hình phi tuyến tính, được sử dụng rộng rãi trong các bài toán phân loại và hồi quy. Mô hình này có thể xử lý các bài toán phân loại phi tuyến tính và hồi quy phi tuyến tính.
- Mô hình kernel linear là một mô hình phi tuyến tính, được sử dụng rộng rãi trong các bài toán phân loại và hồi quy. Mô hình này có thể xử lý các bài toán phân loại phi tuyến tính và hồi quy phi tuyến tính.
- Mô hình kernel Poly (Polynomial) là một mô hình phi tuyến tính, được sử dụng rộng rãi trong các bài toán phân loại và hồi quy. Mô hình này có thể xử lý các bài toán phân loại phi tuyến tính và hồi quy phi tuyến tính.

### 3.3. Bài toán 1: Làm thế nào để lấy được giá bán lẻ tốt nhất

#### 3.3.1. Huấn luyện mô hình

##### 3.3.1.1. Linear Regression 2 biến

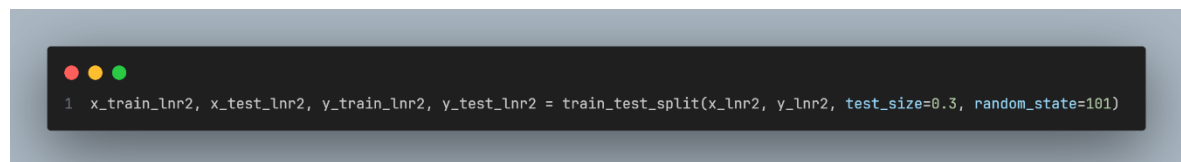
Đầu tiên, chúng ta chọn *Retail Price* làm biến phụ thuộc và *Wholesale Price*, *Total Sold* làm biến độc lập trong tập dữ liệu đã được tiền xử lý trước đó, trong code này thì nó được gọi là *df*.



```
1 y_lnr2 = df["Retail Price"] # Gọi y là biến cần tìm (biến phụ thuộc)
2 x_lnr2 = df[['Wholesale Price', 'Total Sold']] # Gọi x là mối tương quan cần tìm kiếm (biến độc lập)
```

Hình 3.2. Chọn biến độc lập và biến phụ thuộc LNR-2

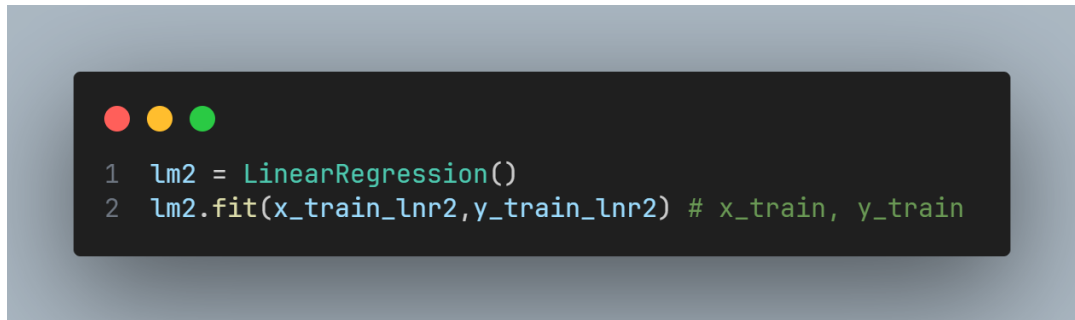
Tiếp sau đó ta sử dụng hàm *train\_test\_split* trong thư viện *sklearn* với tham số *test\_size* (Kích thước của tập kiểm tra) là 0.3 và *random\_state* (số lần tái tạo trong việc kiểm soát việc xáo trộn dữ liệu trước khi áp dụng phân chia) là 101 lần. Hàm này sẽ trả về các tham số là *x\_train*, *y\_train*, *x\_test* và *y\_test*.



```
1 x_train_lnr2, x_test_lnr2, y_train_lnr2, y_test_lnr2 = train_test_split(x_lnr2, y_lnr2, test_size=0.3, random_state=101)
```

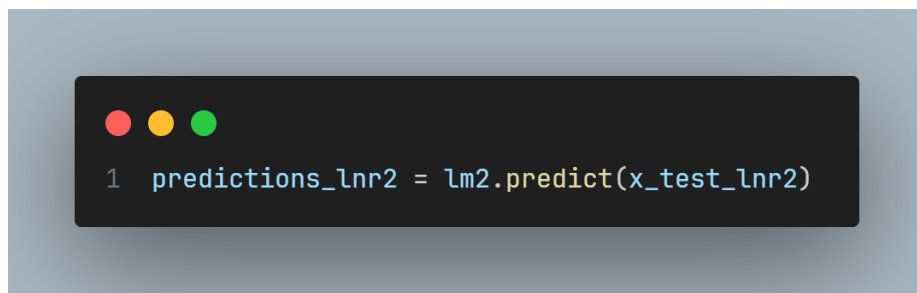
Hình 3.3. Sử dụng hàm *train\_test\_split* để sinh các *x\_train*, *y\_train*, *x\_test* và *y\_test* trong LNR-2

Giải thích cho các tham số trong hàm *train\_test\_split* như sau: *test\_size* là đại diện cho tỷ lệ của tập dữ liệu để chia kiểm tra, *random\_state* là kiểm soát cách dữ liệu được xáo trộn trước khi thực hiện phân chia.



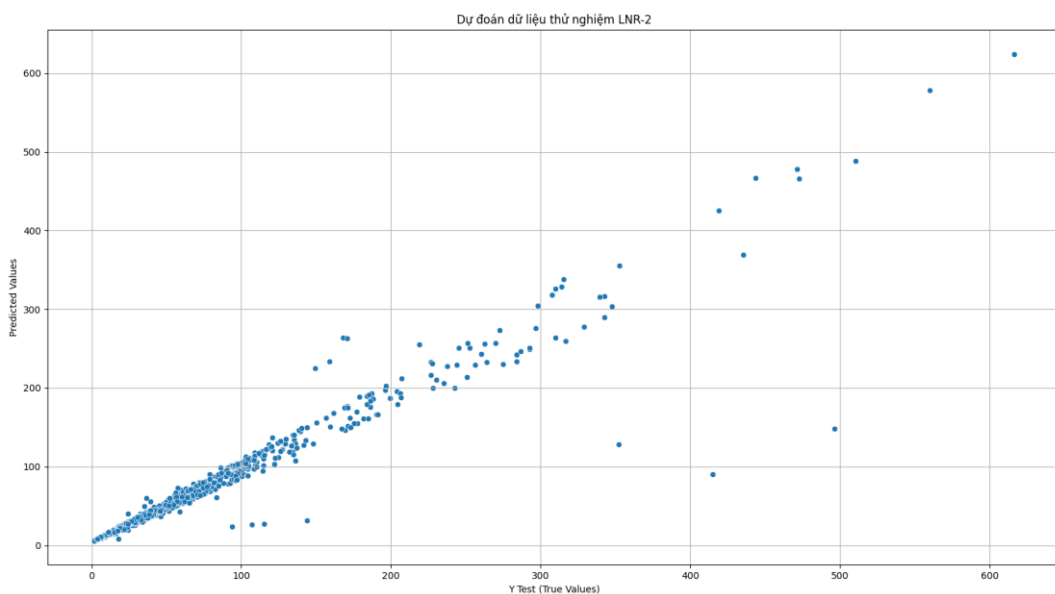
Hình 3.4. Bắt đầu quá trình huấn luyện mô hình LNR-2

Sau khi có được  $x_{train}$  và  $y_{train}$  từ bước trên, ta tiến hành huấn luyện mô hình LNR. Sau khi chạy xong huấn luyện mô hình hồi quy, ta tiếp tục chạy thử nghiệm dự đoán  $predictions\_lnr2$  với  $x_{test}$  và  $y_{test}$ :



Hình 3.5. Dự đoán kết quả LNR-2

Sau khi thực nghiệm dự đoán, ta có mô hình trực quan hóa như sau:



Hình 3.6. Trực quan hóa kết quả dự đoán LNR-2

Sau đó, ta đánh giá mô hình huấn luyện bằng các chỉ tiêu đánh giá MAE, MSE, RMSE và đánh giá hồi quy bằng điểm số hồi quy (điểm số hồi quy đánh giá theo  $y_{test}$  và kết quả đánh giá  $predictions\_lnr2$  mà ta đã chạy trên):

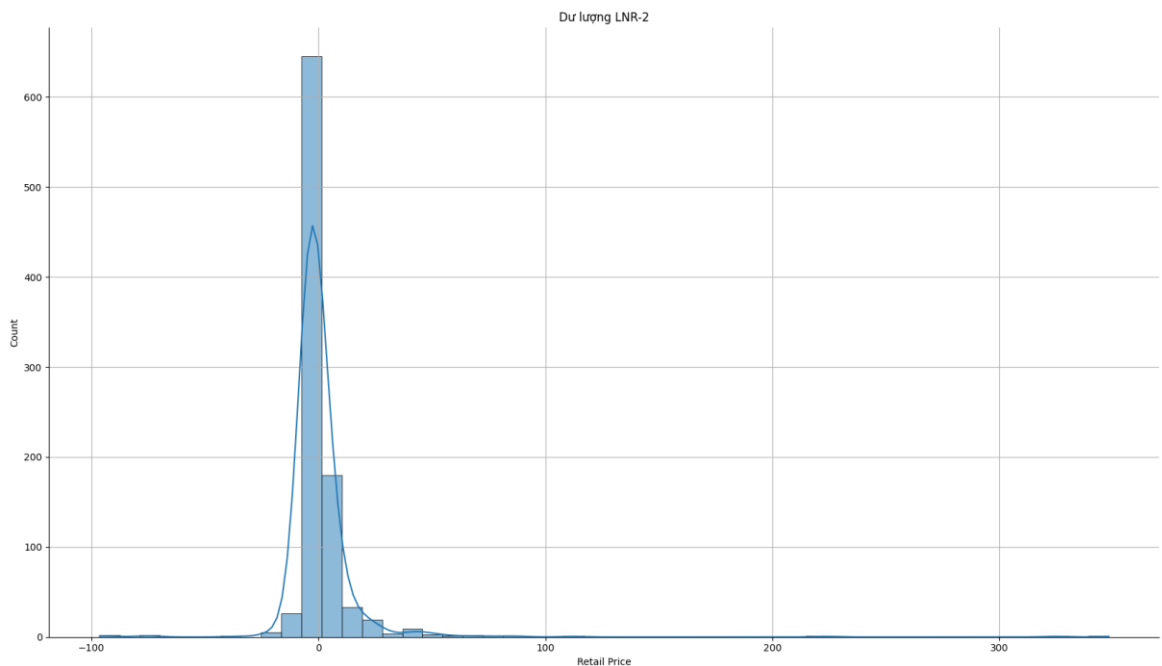
```
MAE: 7.02461112447222
MSE: 436.30270301862566
RMSE: 20.887860182858024
Điểm số hồi quy: 0.923505948694306
```

Hình 3.7. Đánh giá chỉ tiêu và điểm số hồi quy LNR-2

MAE là 7.024611124472225 có nghĩa là trung bình mô hình dự đoán sai lệch khoảng 7.02 đơn vị so với giá trị thực tế, MSE là 436.3027030186259 có nghĩa là bình phương sai số dự đoán của mô hình là 436.30 và RMSE là 20.88786018285803 có nghĩa là mô hình dự đoán sai lệch khoảng 20.89 đơn vị so với giá trị thực tế. Với một dataframe có 3124 bản ghi thì mức độ sai số này là có thể chấp nhận được, kết luận rằng mô hình rất tốt và phù hợp.

Bên cạnh đó điểm số hồi quy cũng sắp xỉ 0.9235, cho thấy rằng phương pháp hồi quy LNR này phù hợp với bộ dữ liệu trong việc huấn luyện mô hình.

Tiếp sau đó là trực quan hóa dữ liệu của một số mô hình:



Hình 3.8. Trực quan hóa phân bố dự lượng LNR-2



Trực quan hóa mức độ phân bổ dư lượng (Residuals) là sự khác biệt giữa giá trị thực tế và giá trị dự đoán bởi mô hình đã được phân tích ở trên. Ở hình trên ta có thể thấy mức độ phân bổ dư lượng là đồng đều và không bị tủa ra quá nhiều. Kết luận rằng phần dư lượng này phân bổ bình thường và nó không phải là vấn đề trong việc phân tích này.

Cuối cùng, ta kiểm tra hệ số (Coefficients) giữa các biến độc lập bằng hàm `coef_` cho ra kết quả như sau:

	Coefficient
Wholesale Price	2.010749
Total Sold	-0.003920

Hình 3.9. Kết quả hệ số của LNR-2

Kết quả từ hình trên cho thấy hệ số của *Wholesale Price* là 2.01 và *Total Sold* là -0.0039, do đó có thể kết luận là không có mối tương quan đối với *Total Sold*, điều này có nghĩa là có một mối quan hệ nghịch đảo giữa biến độc lập và biến phụ thuộc. Do đó ta sẽ thử lại phương pháp LNR một lần nữa nhưng không có biến *Total Sold*, điều này sẽ được trình bày trong phần kế tiếp.

### 3.3.1.2. Linear Regression 1 biến

Cũng giống như LNR 2 biến nhưng lần này ta bắt đầu lại chỉ với 1 biến độc lập là *Wholesale Price* thay vì 2 như trước là *Wholesale Price* và *Retail Price*, ta gọi phương pháp sử dụng lần này là LNR-1:

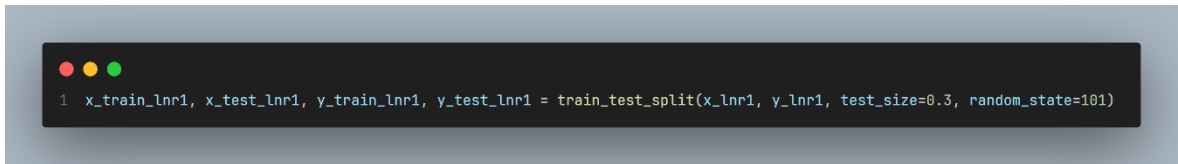
```

1 y_lnr1 = df["Retail Price"] # Gọi y là biến cần tìm (biến phụ thuộc)
2 x_lnr1 = df[['Wholesale Price']] # Gọi x là mối tương quan cần tìm kiếm (biến độc lập)

```

Hình 3.10. Chọn biến độc lập và biến phụ thuộc LNR-1

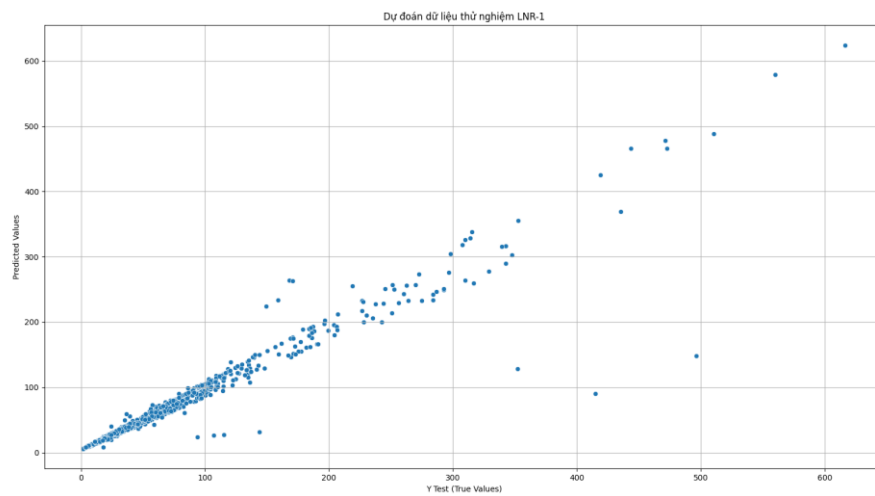
Ta cũng sử dụng lại hàm `train_test_split` giống như lần trước và các tham số `text_size` là 0.3 và `random_state` là 101 lần.



Hình 3.11. Sử dụng hàm `train_test_split` để sinh các  $x_{train}$ ,  $y_{train}$ ,  $x_{test}$  và  $y_{test}$  trong LNR-1

Các bước huấn luyện mô hình hồi quy và dự đoán kết quả ta cũng thực hiện tương tự như LNR-2. Các kết quả thu được được trực quan hóa như sau:

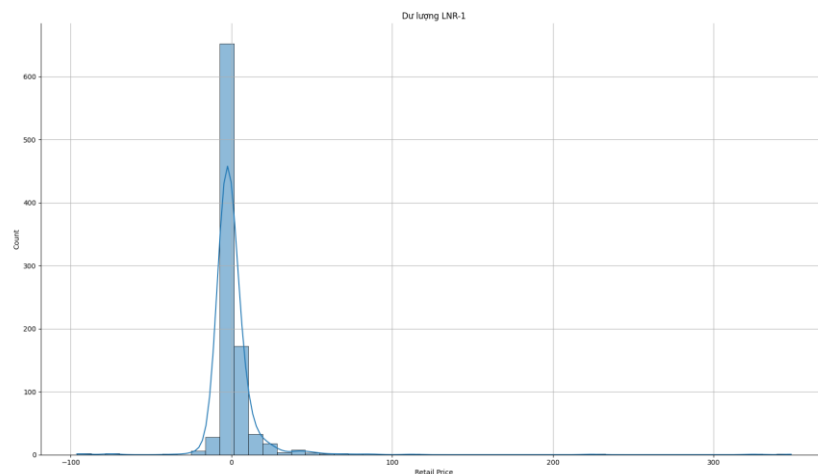
*Mô hình trực quan hóa cho việc dự đoán:*



Hình 3.12. Trực quan hóa kết quả dự đoán LNR-1

Có thể thấy rằng kết quả dự đoán của LNR-1 và LNR-2 là không có sự khác biệt nào cả, giống nhau đến 99%.

*Mô hình trực quan hóa phân bố dư lượng*



Hình 3.13. Trực quan hóa phân bố dư lượng LNR-1

Có thể thấy rằng sự phân bố dư lượng của LNR-1 không có quá nhiều sự khác biệt với LNR-2. Kết luận rằng nó giống với LNR-2, không phải là vấn đề quá lớn trong việc phân tích dữ liệu.

*Đánh giá chỉ tiêu phương sai và điểm số hồi quy LNR-1*

```
MAE: 7.011067143547811
MSE: 436.0616075169243
RMSE: 20.88208819818852
Điểm số hồi quy: 0.9235445518060841
```

Hình 3.14. Đánh giá chỉ tiêu và điểm số hồi quy LNR-1

Điểm số hồi quy và đánh giá chỉ tiêu phương sai MAE, MSE và RMSE không có quá nhiều sự khác biệt so với LNR-2. Có thể đánh giá rằng LNR-1 cũng rất phù hợp cho việc phân tích dữ liệu.

*Kết quả hệ số LNR-1*

```
Coefficient
Wholesale Price 2.010243
```

Hình 3.15. Kết quả hệ số LNR-1

Kết quả hệ số của LNR-1 và LNR-2 là tương đồng nhau, không có quá nhiều sự khác biệt. Kết luận đối với LNR-1 rằng ta có thể giữ phương pháp này cùng với LNR-2 cho những giai đoạn phân tích dữ liệu sau.

### 3.3.1.3. Support Vector Regression

Cũng giống như LNR-2, chúng ta chọn *Retail Price* làm biến phụ thuộc và *Wholesale Price*, *Total Sold* làm biến độc lập. Các tham số của *train\_test\_split* ta cũng giữ nguyên tương tự, gồm *test\_size* là 0.3 và *random\_state* là 101 lần.

```
1 y_svr = df["Retail Price"] # Biến phụ thuộc
2 x_svr = df[["Wholesale Price", "Total Sold"]] # Biến độc lập
3 x_train_svr, x_test_svr, y_train_svr, y_test_svr = train_test_split(x_svr, y_svr, test_size=0.3, random_state=101)
```

Hình 3.16. Biến độc lập, biến phụ thuộc và tham số *train\_test\_split* của SVR

Tiếp sau đó, ta xác định các tham số của từng kernel trong SVR mà ta chọn, trong đây ta chọn 3 kernel để huấn luyện mô hình gồm *RBF*, *Linear* và *Poly*:

```

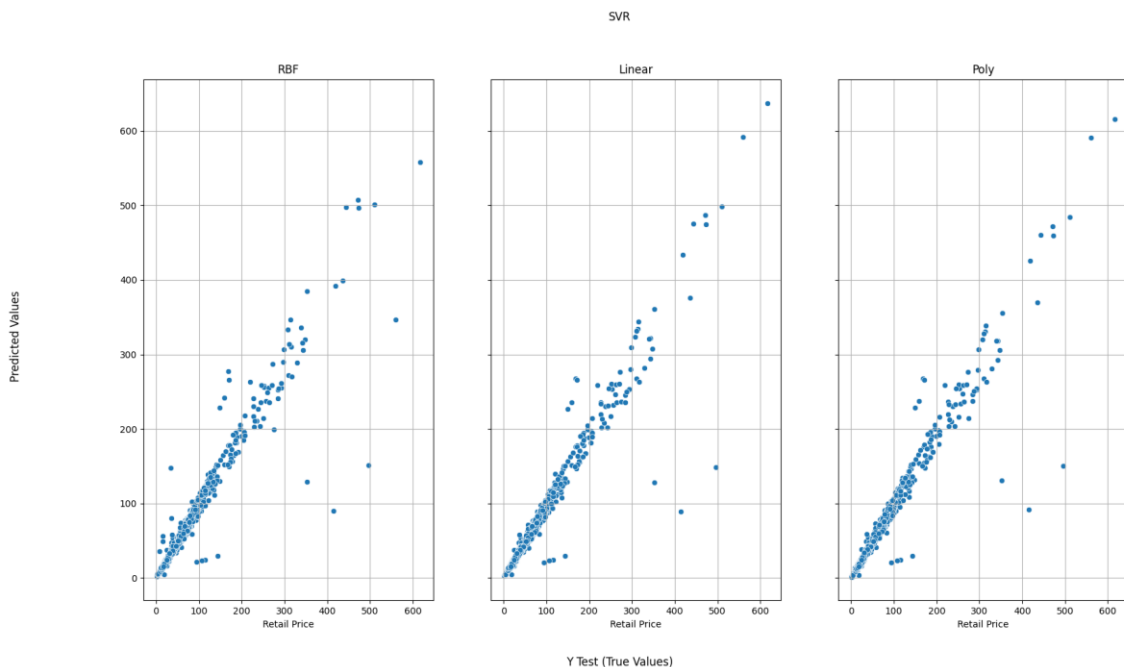
1 svr_rbf = SVR(kernel="rbf", C=100, gamma=0.1, epsilon=0.1)
2 svr_lin = SVR(kernel="linear", C=100, gamma="auto")
3 svr_poly = SVR(kernel="poly", C=100, gamma="auto", degree=3, epsilon=0.1, coef0=1)

```

Hình 3.17. Xác định các tham số của từng kernel trong SVR

Trong đó, đối với từng kernel gồm có các tham số  $C$ ,  $\gamma$ ,  $\epsilon$ ,  $\text{degree}$  và  $\text{coef0}$ . Trong đó  $C$  là tham số chuẩn hóa,  $\gamma$  là Hệ số kernel,  $\epsilon$  là Epsilon trong mô hình epsilon-SVR (Nó chỉ định ống epsilon mà trong đó không có hình phạt nào được liên kết trong hàm mất mát huấn luyện với các điểm được dự đoán trong khoảng cách epsilon từ giá trị thực tế),  $\text{degree}$  là Bậc của hàm kernel đa thức ('poly'),  $\text{coef0}$  là hạng tử độc lập trong hàm.

Sau khi huấn luyện mô hình, ta mô hình trực quan hóa dự đoán như sau:



Hình 3.18. Trực quan hóa kết quả dự đoán của SVR

Mô hình trên là ba đồ thị phân tán khi tiến hành dự đoán giá bán lẻ ứng với mỗi giá trị  $x$  được đưa ra, mỗi đồ thị tương ứng với một mô hình kernel khác nhau. Ta có thể thấy trong khoảng giá trị tiên đoán từ 0 tới 300 của cả ba mô hình đều tương đồng với nhau. Tuy nhiên từ khoảng sau giá trị 300, mô hình kernel RBF có bắt đầu có sự khác biệt so với hai mô hình còn lại. Hai mô hình kernel Linear và Polynomial có kết quả khá giống với mô hình hồi quy tuyến tính hai biến LNR-2 (hình 3.6) và hồi quy tuyến tính một biến LNR-1 (hình 3.12). Điều này chứng tỏ khi dự đoán mức giá thấp hơn 300 thì kết quả của cả ba mô hình đều cho tương tự nhau nhưng khi dự đoán mức giá cao hơn 300 thì mô hình kernel RBF có giá trị không chính xác.

```
Điểm số hồi quy (RBF): 0.9116193226410126
Điểm số hồi quy (Linear): 0.9246798038288828
Điểm số hồi quy (Poly): 0.9248268576359893
```

Hình 3.19. Điểm số hồi quy của SVR

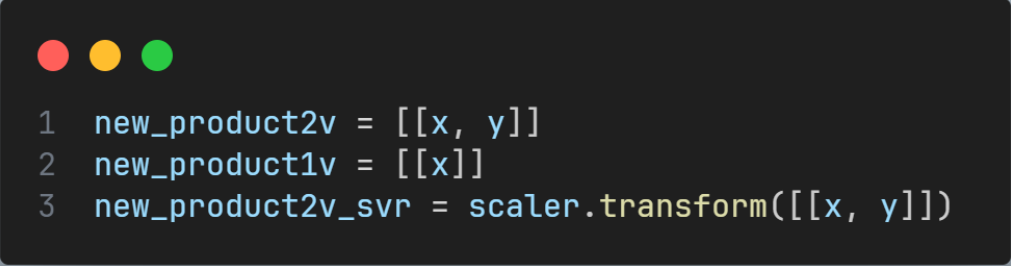
Đánh giá điểm số hồi quy của SVR (hình 3.19): Điểm số hồi quy của SVR trên ba mô hình kernel đều cho mức tốt và chấp nhận được, cho nên cả ba phương pháp này đều sẽ được đưa vào để dự đoán cho những phân phân tích sau.

### 3.3.2. Thực hiện dự đoán giá bán lẻ tốt nhất của một mặt hàng có giá bán sỉ là $x$ và tổng số lượng bán dự kiến là $y$ trong vòng 5 năm

#### 3.3.2.1. Mô tả quá trình dự đoán

Sau quá trình huấn luyện các mô hình LNR-2, LNR-1 và SVR trên, ta tiếp tục tiếp hành công đoạn thử nghiệm các dự đoán trên các mô hình đã được huấn luyện đó.

Cụ thể hơn, ta muốn dự đoán giá bán lẻ tốt nhất (best retail price) dựa trên giá bán sỉ (wholesale price) là  $x$  và thêm một thông tin là tổng lượng bán (total sold) dự kiến là  $y$  trong vòng 5 năm, lưu ý rằng tổng lượng bán dự kiến trong vòng 5 năm này do nhóm giả định rằng trong vòng 5 năm sẽ bán được  $y$  món,  $x$  và  $y$  là 2 giá trị trường hợp cụ thể mà nhóm đề ra. Sau khi có ý tưởng và thông tin như thế, nhóm bắt đầu quá trình phân tích:



```

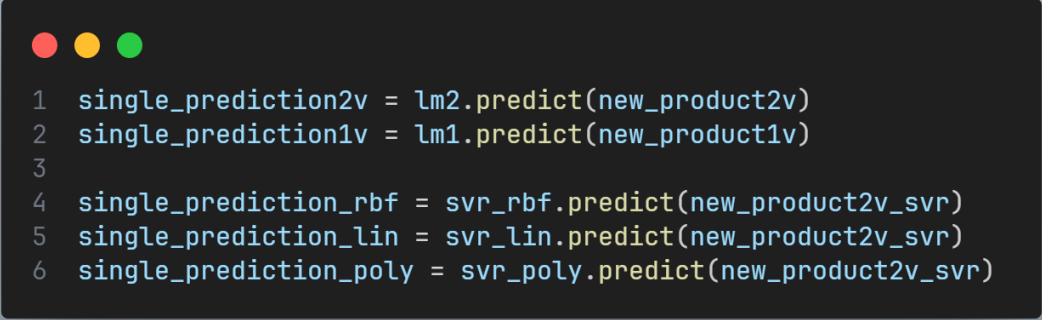
1 new_product2v = [[x, y]]
2 new_product1v = [[x]]
3 new_product2v_svr = scaler.transform([[x, y]])

```

Hình 3.20. Dự đoán giá bán lẻ tốt nhất dựa trên giá bán sỉ và số lượng bán

Ta bắt đầu với việc tạo mới một sản phẩm mà ta muốn dự đoán, sản phẩm này có giá bán lẻ (retail price) là  $x$  và tổng số lượng bán được (total sold) là  $y$ . Đối với các mô hình đã được huấn luyện thì điều này có nghĩa là `new_product2v` sẽ là input đầu vào cho mô hình LNR-2, `new_product1v` sẽ là input đầu vào cho mô hình LNR-1 và `new_product2v_svr` sẽ là input đầu vào cho mô hình SVR. Lưu ý rằng mô hình LNR-1 không có tham số tổng số lượng bán (total sold) do đó nó không thể dự đoán kết quả dựa trên số lượng bán mà chỉ có thể dự đoán trên giá bán sỉ (wholesale price), lý do mà nhóm vẫn quyết định giữ mô hình này lại là bởi vì nhóm cũng muốn dự đoán bán lẻ tốt nhất dựa trên giá bán sỉ mà không có tham số tổng lượng bán.

Sau quá trình huấn luyện các mô hình LNR-2, LNR-1 và SVR trên, ta tiếp tục tiếp hành công đoạn thử nghiệm các dự đoán trên các mô hình đã được huấn luyện đó.



```

1 single_prediction2v = lm2.predict(new_product2v)
2 single_prediction1v = lm1.predict(new_product1v)
3
4 single_prediction_rbf = svr_rbf.predict(new_product2v_svr)
5 single_prediction_lin = svr_lin.predict(new_product2v_svr)
6 single_prediction_poly = svr_poly.predict(new_product2v_svr)

```

Hình 3.21. Nạp dữ liệu dự đoán vào các mô hình đã được huấn luyện

Ta nạp các dữ liệu input đầu vào vào các mô hình đã được huấn luyện sẵn là *lm2*, *lm1* và *svr\_rbf*, *svr\_lin*, *svr\_poly*. Giá trị dự đoán trả về sẽ được mô tả trong các phần sau

### 3.3.2.2. Kết quả dự đoán của giá bán sỉ 200 USD và tổng lượng bán 500

	Predictions
LinReg 2v	405.128544
LinReg 1v	406.775804
SVR RBF	366.424174
SVR Linear	414.994275
SVR Poly	403.906097

Hình 3.22. Kết quả giá bán lẻ tốt nhất dự đoán được trả về với trường hợp giá bán sỉ là 200 USD và tổng lượng bán là 500

Đối với giá bán sỉ là 200 USD và tổng lượng bán là 500 thì kết quả dự đoán được trả về như trên (hình 3.22) trên, cho ta thấy các mô hình dự đoán 1 khoảng gần tương tự nhau và độ chênh lệch là không quá nhiều. Trong đó mô hình SVR Linear cho kết quả dự đoán giá bán lẻ cao nhất là 414.99 USD và mô hình SVR RBF thì lại cho kết quả dự đoán giá bán lẻ cao nhất là 366.42 USD thấp nhất trong tất cả mô hình.

### 3.3.2.3. Kết quả dự đoán của giá bán sỉ 1000 và tổng lượng bán 500

	Predictions
LinReg 2v	2013.728033
LinReg 1v	2014.970024
SVR RBF	254.626363
SVR Linear	2064.905605
SVR Poly	3362.845883

Hình 3.23. Kết quả giá bán lẻ tốt nhất dự đoán được trả về với trường hợp giá bán sỉ là 1000 USD và tổng lượng bán là 500

Đối với giá bán sỉ là 1000 USD và tổng lượng bán là 500 thì kết quả dự đoán được trả về như trên (hình 3.23) trên, nó có một sự chênh lệch rất lớn. Cụ thể là mô hình huấn luyện là SVR RBF lại trả về giá trị cực kì thấp nếu so với các mô hình huấn luyện còn lại và SVR Poly thì lại cho kết quả dự đoán giá bán lẻ tốt nhất là cao nhất so với tất cả mô hình khác. Sự chênh lệch này có thể làm ảnh hưởng đến kết quả dự

đoán nếu ta phải chọn một mô hình để dự đoán, do đó ở phần tiếp theo nhóm sẽ tiến hành phân tích để chọn ra mô hình nên được sử dụng nhất trong việc dự đoán kết quả của bài toán.

### 3.3.3. So sánh giữa các mô hình huấn luyện

Sau quá trình chạy thực nghiệm ở phần 3.3.2.2 và 3.3.2.3, nhóm đã kết luận được rằng không phải mô hình huấn luyện nào cũng ra kết quả tốt nhất và việc chọn ra một mô hình huấn luyện có kết quả chuẩn xác nhất là điều cần thiết, do đã thực nghiệm ở cả 2 trường hợp mà không rút ra được mô hình huấn luyện nào là tốt nhất, do đó nhóm đã tiến hành so sánh điểm số hồi quy giữa các phương pháp để tìm ra mô hình huấn luyện tốt nhất:

	R2
LinReg 2v	0.923506
LinReg 1v	0.923545
SVR RBF	0.911619
SVR Linear	0.924680
SVR Poly	0.924827

Hình 3.24. Điểm số hồi quy giữa các mô hình huấn luyện

Điểm số hồi quy trên cho thấy mô hình huấn luyện SVR Poly là mô hình huấn luyện phù hợp nhất do nó có điểm số hồi quy là cao nhất, bên cạnh đó thì SVR RBF lại có điểm số hồi quy là thấp nhất. Minh chứng thực tế nhất đó là việc ta đã chạy dự đoán ở 2 phần 3.3.2.2 và 3.3.2.3 đều cho kết quả dự đoán đối với mô hình SVR RBF là thấp nhất. Kết luận cuối cùng của nhóm đó là mô hình huấn luyện tốt nhất đó là mô hình SVR Poly cho việc giải quyết bài toán tìm giá bán lẻ tốt nhất.



### 3.4. Bài toán 2: Có lợi thế về ngày giao hàng khi trở thành khách hàng có thứ hạng Bạch Kim hay không

Khách hàng được chia thành ba cấp bậc: Bạc, Vàng và Bạch Kim nhưng điều đó có lợi ích cho khách hàng hay không. Trong bài toán này ta sẽ cùng nhau giải quyết liệu ngày giao hàng có nhanh hơn khi ta hạng Bạch Kim không?

Để thuận tiện giải quyết bài toán này, đầu tiên nhóm em tạo ra một cột dữ liệu mới gọi là *Delay for Delivery*. Sau đó, cột *Delay for Delivery* này được thêm vào dataframe bằng cách lấy hiệu của cột *Delivery Date* và cột *Date Order was placed*. Cuối cùng, các cột không cần thiết được loại bỏ khỏi dataframe và ta có được như (hình 3.25)

	Customer	Status	Date Order was placed	Delivery Date	Delay for Delivery
0		Silver	2017-01-01	2017-01-07	6 days
1		Silver	2017-01-01	2017-01-05	4 days
2		Gold	2017-01-01	2017-01-04	3 days
3		Gold	2017-01-01	2017-01-06	5 days
4		Gold	2017-01-01	2017-01-04	3 days

Hình 3.25. Dataframe mới phù hợp cho việc giải quyết bài toán số 2

Ta tiếp tục chuyển đổi cột *Delay for Delivery* thành số (Hình 3.27)

#	Column	Non-Null Count	Dtype
0	Customer Status	185013 non-null	string
1	Date Order was placed	185013 non-null	datetime64[ns]
2	Delivery Date	185013 non-null	datetime64[ns]
3	Delay for Delivery	185013 non-null	timedelta64[ns]

dtypes: datetime64[ns](2), string(1), timedelta64[ns](1)

Hình 3.26. Cột *Delay for Delivery* trước khi chuyển thành số

Data columns (total 4 columns):			
#	Column	Non-Null Count	Dtype
0	Customer Status	185013 non-null	string
1	Date Order was placed	185013 non-null	datetime64[ns]
2	Delivery Date	185013 non-null	datetime64[ns]
3	Delay for Delivery	185013 non-null	int64

dtypes: datetime64[ns](2), int64(1), string(1)

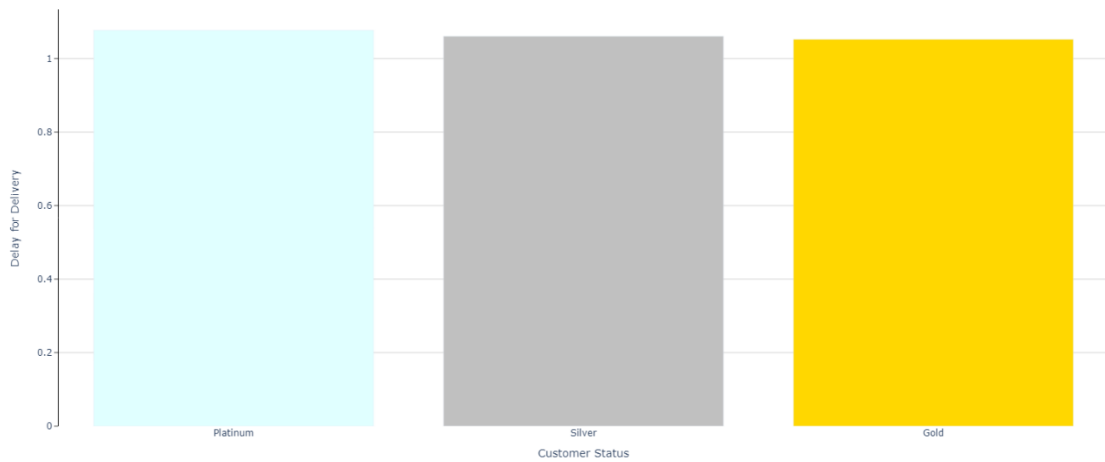
Hình 3.27. Cột *Delay for Delivery* sau khi chuyển thành số

Tiếp theo ta gom cụm theo cột *Customer Status* và tính trung bình cột *Delay for Delivery* và có được dataframe cho thời gian giao hàng trung bình cho các hạng thành viên như (hình 3.28)

	Customer Status	Delay for Delivery
0	Gold	1.052040
1	Platinum	1.077015
2	Silver	1.060438

Hình 3.28. Thời gian trung bình chờ đợi của khách hàng theo từng loại khách hàng

Mô hình trực quan hoá dữ liệu trên:



Hình 3.29. Mô hình hoá dữ liệu của dataframe mới

Từ các phân tích trên ta có thể thấy thời gian giao hàng gần như không có sự chênh lệch nên vì vậy khi trở thành khách hàng hạng Bạch Kim thì thời gian giao hàng cũng không thay đổi so với khách hàng hạng Bạc và Vàng.

## KẾT LUẬN

Trong đồ án này nhóm chúng em đã thực hiện phân tích trên hai bộ dữ liệu *orders.csv* và *product-supplier.csv* để giải quyết các mục tiêu đã đề ra.

➤ Đối với dữ liệu về lịch sử mua hàng nhóm đã phân tích và xác định:

- Mặt hàng nào có tỉ suất lợi nhuận cao nhất?

Danh mục Golf có sản phẩm có tỷ suất lợi nhuận cao nhất với 16 sản phẩm.

- Mặt hàng nào bán chạy nhất, mặt hàng nào không bán chạy.
  - Danh mục Clothes là danh mục bán chạy nhất.
  - Danh mục Indoor Sport là danh mục bán ít nhất.
- Môi trường quan giữa giá bán lẻ?
  - Giá bán lẻ thường cao hơn nhiều so với giá bán sỉ.
  - Hầu hết giá các mặt hàng đều dưới 50, dao động trên cả 2 loại mặt hàng bán lẻ và bán sỉ.

➤ Đối với dữ liệu về thời gian nhóm đã phân tích và xác định:

- Tần suất mua hàng của các thời điểm trong năm.
  - Sức mua hàng luôn ở mức ổn định qua các năm và có xu hướng tăng lên.
  - Đặc biệt tăng nhiều vào tháng 5, 6, 7 và 11.
- Ngày giao hàng có thay đổi không khi trở thành khách hàng hạng Bạch kim?

Trở thành khách hàng hạng Bạch Kim thì thời gian giao hàng cũng không thay đổi so với khách hàng hạng Bạc và Vàng.

➤ Đối với dữ liệu về lịch sử thanh toán nhóm đã phân tích và xác định:

- Khi ta có giá sỉ là 200 USD mong muốn tổng số lượng bán ra là 500 thì khi đó hệ thống sẽ dự đoán giá bán lẻ tốt nhất là 403.9 USD.
- Khi ta có giá sỉ là 1000 USD mong muốn tổng số lượng bán ra là 500 thì khi đó hệ thống sẽ dự đoán giá bán lẻ tốt nhất là 3362.84 USD.