

ĐẠI HỌC QUỐC GIA TP.HCM
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC VÀ KĨ THUẬT MÁY TÍNH



BÁO CÁO BÀI TẬP LỚN
KHO DỮ LIỆU VÀ HỆ HỒ TRỢ QUYẾT ĐỊNH (CO4031)

Xây dựng Kho dữ liệu Thương mại điện tử tập trung vào
hành vi mua sắm của khách hàng

Nhóm lớp L01

Giảng viên hướng dẫn: ThS. Bùi Tiến Đức
Sinh viên: Tôn Việt Tri - 2213603

Thành phố Hồ Chí Minh, 12/2025



Mục lục

1 GIỚI THIỆU ĐỀ TÀI VÀ TỔNG QUAN DỮ LIỆU	2
1.1 Giới thiệu Đề tài và Bối cảnh Nghiệp vụ	2
1.1.1 Mục tiêu Chiến lược và Phân tích	2
1.2 Tổng quan Nguồn Dữ liệu và Phân tích Chất lượng	2
1.2.1 Cấu trúc và Đặc điểm Dữ liệu Thô	2
1.2.2 Phân tích Chất lượng Dữ liệu (DQA) và Chiến lược Xử lý	2
2 QUY TRÌNH ETL VÀ THIẾT KẾ KHO DỮ LIỆU	4
2.1 Quá trình Tiền xử lý Dữ liệu	4
2.1.1 Các bước Làm sạch Dữ liệu	4
2.1.2 Code Minh họa Tiền xử lý	4
2.2 Thiết kế Mô hình Kho Dữ liệu (Star Schema)	5
2.2.1 Cấu trúc Bảng Fact: Fact_Sales	5
2.2.2 Cấu trúc các Bảng Dimension	6
2.3 Quy trình ETL Load và Ánh xa Khóa (SK Mapping)	8
2.3.1 Giai đoạn 1: Trích xuất Dữ liệu Dimension	8
2.3.2 Giai đoạn 2: Tải Dimension	8
2.3.3 Giai đoạn 3: Ánh xa Khóa Thay thế (SK Mapping) và Tải Fact	8
3 PHÂN TÍCH DỮ LIỆU VÀ ỨNG DỤNG THUẬT TOÁN	9
3.1 Phân tích Hiệu suất Tổng hợp (OLAP)	9
3.1.1 Phân tích 1: Chỉ số KPI Tổng hợp	9
3.1.2 Phân tích 2: Xu hướng Doanh thu theo Thời gian	10
3.1.3 Phân tích 3: Phân tích Dia lý và Thị trường Trong tâm	11
3.1.4 Phân tích 4: Hiệu suất Sản phẩm	12
3.2 Khai phá Dữ liệu (Data mining) với Thuật toán Apriori (Market Basket Analysis)	14
3.2.1 Mục tiêu và Phương pháp	14
3.2.2 Bước 1: Chuẩn bị Dữ liệu	14
3.2.3 Bước 2: Thực thi Thuật toán Apriori	14
3.2.4 Kết quả Luật Kết hợp (Top Association Rules)	15
3.2.5 Phân tích Chi tiết: Tại sao Lift lại cao đột biến (> 30)	15
3.2.6 Kiểm chứng Giả thuyết từ Kho Dữ liệu (Validation)	16
3.2.7 Dánh giá Tác động Kinh doanh: Ma trận Lift và Support	16
3.2.8 Phân loại Quy tắc và Đề xuất Chiến lược (Actionable Insights)	17
3.2.9 Hạn chế và Hướng phát triển	17
4 TRỰC QUAN HÓA DỮ LIỆU VÀ HỆ HỖ TRỢ QUYẾT ĐỊNH	18
4.1 Công cụ thực hiện	18
4.2 Dashboard 1: Hiệu suất Kinh doanh Tổng thể	18
4.2.1 Các thành phần trực quan hóa	18
4.2.2 Insight và Hỗ trợ Quyết định	19
4.3 Dashboard 2: Phân tích Giả hàng và Chiến lược Bán chéo	19
4.3.1 Các thành phần trực quan hóa	19
4.3.2 Kịch bản Ứng dụng Chiến lược	20
5 HẠN CHẾ VÀ HƯỚNG PHÁT TRIỂN	21
5.1 Hạn chế của đề tài	21
5.2 Hướng phát triển	21
6 KẾT LUẬN	21



1 GIỚI THIỆU ĐỀ TÀI VÀ TỔNG QUAN DỮ LIỆU

1.1 Giới thiệu Đề tài và Bối cảnh Nghiệp vụ

Mục tiêu cốt lõi của dự án là xây dựng một hệ thống Kho dữ liệu (Data Warehouse - DW) và Hệ thống Hỗ trợ Quyết định (Decision Support System - DSS) nhằm tối ưu hóa các chiến lược kinh doanh cho một nhà bán lẻ trực tuyến. Dự án chuyển đổi dữ liệu giao dịch đơn thuần thành các Insight đa chiều có tính hành động cao.

1.1.1 Mục tiêu Chiến lược và Phân tích

Dự án được định hướng để giải quyết ba nhóm mục tiêu chiến lược quan trọng:

1. Tối ưu Hóa đơn và Bán chéo:

- **Ứng dụng Kỹ thuật Khai phá Dữ liệu:** Sử dụng thuật toán Market Basket Analysis (MBA) - Apriori để phát hiện các quy tắc kết hợp sản phẩm (Association Rules). Mục tiêu không chỉ là tìm ra quy tắc, mà là tìm ra quy tắc có Lift cao (Lift > 1.2) để đảm bảo tính phi ngẫu nhiên và hiệu quả thực tế của việc bán chéo.
- **Tăng AOV:** Đề xuất các gói Combo sản phẩm có giá trị cao, dựa trên Confidence và Lift của luật kết hợp, nhằm tăng Giá trị Đơn hàng Trung bình (AOV - Average Order Value).

2. Phân tích Hiệu suất Đa chiều (OLAP Analysis):

- **Nhận diện Tính Mùa vụ:** Phân tích TotalRevenue theo các chiều Month và Year để xác định chính xác các giai đoạn bùng nổ (ví dụ: mùa lễ hội) và chuẩn bị tồn kho.
- **Phân khúc Thị trường:** Dánh giá hiệu suất kinh doanh dựa trên chiều Country (xác định thị trường trọng điểm).

3. Quản trị và Chuẩn hóa Dữ liệu:

Dảm bảo dữ liệu giao dịch được chuyển đổi thành cấu trúc Star Schema dễ truy vấn, ổn định và có tính toàn vẹn cao.

1.2 Tổng quan Nguồn Dữ liệu và Phân tích Chất lượng

Bộ dữ liệu được sử dụng là **Online Retail Dataset** (nguồn Kaggle), ghi nhận tất cả các giao dịch trong lĩnh vực bán lẻ quà tặng từ 01/12/2010 đến 09/12/2011.

1.2.1 Cấu trúc và Đặc điểm Dữ liệu Thô

- **Quy mô:** Dữ liệu thô chứa 541,909 dòng và 8 cột.
- **Các trường dữ liệu chính:** Bao gồm InvoiceNo, StockCode, Description, Quantity, InvoiceDate, UnitPrice, CustomerID, và Country.
- **Phân loại Khách hàng:** Công ty chủ yếu bán các mặt hàng quà tặng.

1.2.2 Phân tích Chất lượng Dữ liệu (DQA) và Chiến lược Xử lý

Quá trình DQA xác định các vấn đề nghiêm trọng cần được khắc phục trong pha Transform của ETL để đảm bảo chất lượng cho DW.

label=P1: Dữ liệu Thiếu CustomerID:

- **Vấn đề:** Khoảng 24.93% ($\approx 135,000$ dòng) giao dịch thiếu thông tin khách hàng (NULL).



- **Hậu quả nếu bỏ qua:** Bỏ qua một lượng doanh thu và giao dịch lớn, dẫn đến kết quả phân tích bị sai lệch.
- **Chiến lược xử lý:** Gán mã 99999.

label=P2: **Dữ liệu Giao dịch Không hợp lệ:**

- **Vấn đề:** Các giá trị Quantity ≤ 0 (hoàn trả, hủy) và UnitPrice ≤ 0 (giá không hợp lệ).
- **Chiến lược xử lý:** Loại bỏ tất cả các dòng giao dịch không hợp lệ này. Điều này đảm bảo rằng bảng Fact chỉ chứa các giao dịch bán hàng thành công, là nền tảng cho việc tính toán doanh thu thuận.

label=P3: **Lỗi Sai số Số thực:**

- **Vấn đề:** Trong các hệ thống tính toán (như Python), phép nhân giữa hai số thực có thể gây ra sai số nhỏ (ví dụ: 75.60000000000001).
- **Chiến lược xử lý:** Áp dụng làm tròn .round(2) cho UnitPrice và tính toán lại TotalRevenue (Quantity \times UnitPrice). Đây là bước cần thiết để đảm bảo **Độ Chính Xác Tài Chính** của dữ liệu trong Kho.



2 QUY TRÌNH ETL VÀ THIẾT KẾ KHO DỮ LIỆU

2.1 Quá trình Tiền xử lý Dữ liệu

Giai đoạn tiền xử lý đóng vai trò quan trọng trong việc đảm bảo Chất lượng Dữ liệu (Data Quality) trước khi đưa vào Kho dữ liệu. Quá trình này được thực hiện với các bước cụ thể sau:

2.1.1 Các bước Làm sạch Dữ liệu

- Lọc bỏ dữ liệu rác và giao dịch hủy:** Dữ liệu thô chứa các giao dịch có Quantity âm (thường là đơn hàng bị hủy hoặc trả lại) và UnitPrice âm. Chúng ta loại bỏ các dòng này để đảm bảo chỉ phân tích các giao dịch bán hàng thành công thực tế.
- Xử lý Dữ liệu thiếu (CustomerID):** Khoảng 25% giao dịch thiếu mã khách hàng. Thay vì loại bỏ (gây mất mát dữ liệu doanh thu lớn), chúng ta áp dụng kỹ thuật *Imputation* bằng cách gán mã định danh đặc biệt **99999**.
- Khắc phục Lỗi Sai số Số thực:** Trong môi trường tính toán máy tính, phép nhân số thực có thể gây ra sai số nhỏ (ví dụ: $19.9 \times 3 = 59.699999999$).
 - Giải pháp:** Làm tròn (.round(2)) cột UnitPrice và tính toán lại cột TotalRevenue dựa trên giá trị đã làm tròn. Điều này đảm bảo tính chính xác tuyệt đối cho các báo cáo tài chính.
- Chuẩn hóa Chuỗi ký tự:** Loại bỏ các khoảng trắng thừa ở các cột định danh (StockCode, Description, Country) để đảm bảo tính nhất quán khi thực hiện phép JOIN trong SQL sau này.

2.1.2 Code Minh họa Tiền xử lý

```
1 def preprocess_data(file_path):  
2     # ... (Đọc dữ liệu từ CSV)  
3  
4     # Filter negative values (Quantity > 0 & UnitPrice > 0)  
5     # Loại bỏ đơn hàng hủy và lỗi giá  
6     df_clean = df[(df['Quantity'] > 0) & (df['UnitPrice'] > 0)].copy()  
7  
8     # Handle missing CustomerID (Assign 99999)  
9     df_clean['CustomerID'] = df_clean['CustomerID'].fillna(99999)  
10    df_clean['CustomerID'] = df_clean['CustomerID'].astype(int)  
11  
12    # Floating Point Precision  
13    # Khắc phục lỗi sai số số thực, đảm bảo độ chính xác tài chính  
14    df_clean['UnitPrice'] = df_clean['UnitPrice'].round(2)  
15    df_clean['TotalRevenue'] = (df_clean['Quantity'] * df_clean['UnitPrice']).round(2)  
16  
17    # Clean whitespace for key Dimension columns  
18    # Chuẩn hóa chuỗi để tối ưu việc JOIN sau này  
19    df_clean['StockCode'] = df_clean['StockCode'].astype(str).str.strip()  
20    df_clean['Description'] = df_clean['Description'].astype(str).str.strip()  
21    df_clean['Country'] = df_clean['Country'].astype(str).str.strip()  
22  
23    return df_clean
```

```

Removed: 11805 invalid rows (returns/cancellations, negative prices).
Assigned walk-in code (99999) to 132220 rows missing CustomerID.
Normalized InvoiceDate.
CRITICAL FIX: Applied .round(2) to UnitPrice and TotalRevenue to fix floating point precision errors.
Cleaned whitespace from key Dimension columns (StockCode, Description, Country).

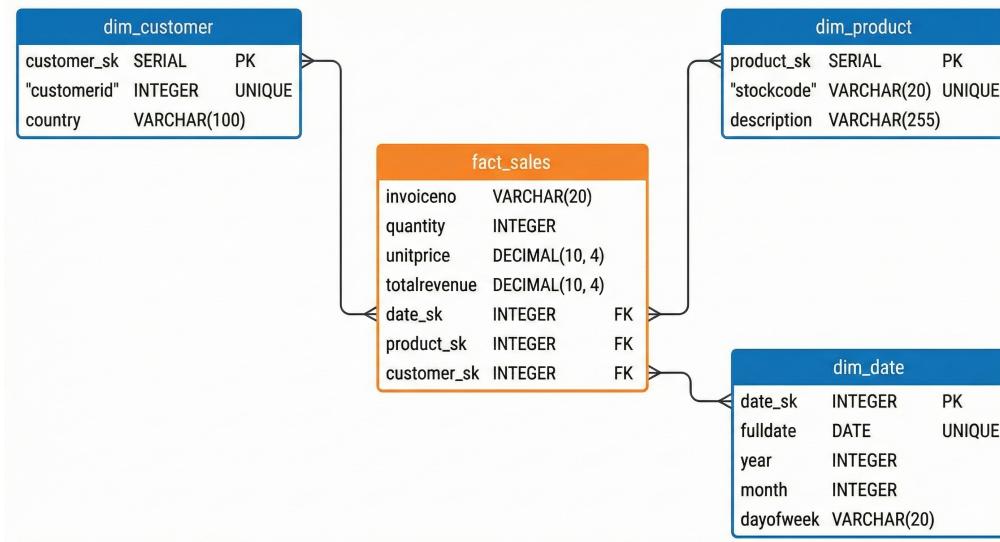
Dataframe info after preprocessing:
<class 'pandas.core.frame.DataFrame'>
Index: 530104 entries, 0 to 541908
Data columns (total 9 columns):
 #   Column      Non-Null Count  Dtype  
--- 
 0   InvoiceNo    530104 non-null   object 
 1   StockCode     530104 non-null   object 
 2   Description   530104 non-null   object 
 3   Quantity      530104 non-null   int64  
 4   InvoiceDate   530104 non-null   datetime64[ns]
 5   UnitPrice     530104 non-null   float64
 6   CustomerID   530104 non-null   int64  
 7   Country       530104 non-null   object 
 8   TotalRevenue  530104 non-null   float64
dtypes: datetime64[ns](1), float64(2), int64(2), object(4)
memory usage: 40.4+ MB
None

```

Hình 1: Kết quả tiền xử lý

2.2 Thiết kế Mô hình Kho Dữ liệu (Star Schema)

Mô hình Kho dữ liệu (Data Warehouse - DW) được xây dựng theo kiến trúc **Star Schema** (Lược đồ Ngôi sao). Mô hình này bao gồm một bảng sự kiện trung tâm (Fact Table) được bao quanh bởi các bảng vệ tinh (Dimension Tables), giúp tối ưu hóa hiệu suất truy vấn đa chiều (OLAP) và phân tích Data Mining.



Hình 2: Mô hình Lược đồ sao (Star Schema) của Kho dữ liệu

2.2.1 Cấu trúc Bảng Fact: Fact_Sales

Mô tả Bảng Fact_Sales: Bảng Fact_Sales là bảng trung tâm lưu trữ các sự kiện giao dịch chi tiết. Mỗi dòng trong bảng đại diện cho một dòng chi tiết trong hóa đơn bán hàng. Bảng này chứa các Khóa ngoại (FK) liên kết đến các bảng Dimension và các Số đo (Measures) định lượng.



Cột	Kiểu Dữ liệu	Vai trò	Mục đích
invoiceno	VARCHAR (20)	Khóa nghiệp vụ	Mã hóa đơn gốc. Dùng để tính COUNT DISTINCT cho Tổng số Hóa đơn (Total Transactions).
quantity	INTEGER	Measure	Số lượng bán ra (đã được làm sạch).
unitprice	DECIMAL (10, 4)	Measure	Đơn giá tại thời điểm bán (đã được làm tròn).
totalrevenue	DECIMAL (10, 4)	Measure	Chỉ số tài chính cốt lõi: Quantity × UnitPrice.
date_sk	INTEGER	Foreign Key (FK)	Liên kết đến Dim_Date.
product_sk	INTEGER	Foreign Key (FK)	Liên kết đến Dim_Product.
customer_sk	INTEGER	Foreign Key (FK)	Liên kết đến Dim_Customer.

Bảng 1: Cấu trúc Chi tiết Bảng Fact_Sales

Mục đích Phân tích: Bảng Fact cho phép thực hiện các phép toán tổng hợp (Aggregation) để tính toán Doanh thu, AOV, và tần suất mua hàng.

2.2.2 Cấu trúc các Bảng Dimension

2.2.2.1 Dim_Customer - Bảng Khách hàng

Mô tả Dim_Customer: Bảng này cung cấp ngữ cảnh về người mua.

Cột	Kiểu Dữ liệu	Vai trò và Chức năng
customer_sk	SERIAL (PK)	Khóa thay thế (Surrogate Key).
customerid	INTEGER	Khóa nghiệp vụ. Cột chính dùng để liên kết với dữ liệu nguồn.
country	VARCHAR	Thuộc tính phân tích địa lý.

Bảng 2: Cấu trúc Dim_Customer

Mục đích Phân tích: Phân tích Top Quốc gia tiêu thụ.

2.2.2.2 Dim_Product - Bảng Sản phẩm

Mô tả Dim_Product: Bảng này cung cấp ngữ cảnh về các mặt hàng. Nó đảm bảo mỗi sản phẩm (được định danh bởi StockCode) chỉ xuất hiện một lần duy nhất.



Cột	Kiểu Dữ liệu	Vai trò và Chức năng
product_sk	SERIAL (PK)	Khóa thay thế.
stockcode	VARCHAR	Khóa nghiệp vụ.
description	VARCHAR	Tên mô tả sản phẩm. Đây là trường dữ liệu quan trọng nhất cho thuật toán Apriori.

Bảng 3: Cấu trúc Dim_Product

Mục đích Phân tích: Phân tích doanh thu theo sản phẩm và tìm kiếm quy tắc kết hợp (Association Rules).

2.2.2.3 Dim_Date - Bảng Thời gian

Mô tả Dim_Date: Bảng này được tách ra từ trường thời gian gốc, cung cấp các thuộc tính lịch chi tiết để hỗ trợ việc phân tích sâu và tổng hợp.

Cột	Kiểu Dữ liệu	Vai trò và Chức năng
date_sk	INTEGER (PK)	Khóa thay thế định dạng số (YYYYMMDD).
fulldate	DATE	Ngày giao dịch đầy đủ.
year, month	INTEGER	Thuộc tính cho phân tích xu hướng dài hạn và tính mùa vụ.
dayofweek	VARCHAR	Thuộc tính cho phân tích chu kỳ tuần (ví dụ: Thứ Hai vs Chủ Nhật).

Bảng 4: Cấu trúc Dim_Date

Mục đích Phân tích: Nhận diện xu hướng tăng trưởng theo tháng/năm và các đợt cao điểm mua sắm.



2.3 Quy trình ETL Load và Ánh xạ Khóa (SK Mapping)

Quy trình ETL Load (thực hiện trong `etl_load.py`) là bước chuyển dữ liệu sạch vào Kho dữ liệu. Quy trình này bao gồm ba giai đoạn chính: Trích xuất Dimension, Tải Dimension vào DB, và Ánh xạ Khóa thay thế (SK Mapping) cho bảng Fact.

2.3.1 Giai đoạn 1: Trích xuất Dữ liệu Dimension

Từ tập dữ liệu giao dịch đã làm sạch, chúng ta trích xuất danh sách duy nhất cho các đối tượng Customer, Product và Date.

- **Làm sạch nâng cao:** Tại bước này, bảng `dim_product` được xử lý thêm để loại bỏ các mô tả sản phẩm quá ngắn hoặc không hợp lệ.
- **Tạo Date Dimension:** Từ cột `InvoiceDate`, chúng ta tính toán các thuộc tính phái sinh như `date_sk`, `year`, `month`, `dayofweek`.

```
1 def extract_dim_data(df):
2     # 1. Dim_Customer: Lấy danh sách khách hàng duy nhất
3     dim_customer_data = df[['customerid', 'country']].drop_duplicates(subset=['customerid']).copy()
4
5     # 2. Dim_Product: Lấy danh sách sản phẩm duy nhất và làm sạch Description
6     dim_product_data = df[['stockcode', 'description']].drop_duplicates().copy()
7     dim_product_data = dim_product_data[dim_product_data['description'].str.len() > 1].copy()
8     dim_product_data = dim_product_data.drop_duplicates(subset=['stockcode']).copy()
9
10    # 3. Dim_Date: Tính toán thuộc tính thời gian
11    df['invoicedateonly'] = df['invoicedate'].dt.normalize()
12    dim_date_data = df['invoicedateonly'].drop_duplicates().to_frame()
13    dim_date_data.columns = ['fulldate']
14    dim_date_data['date_sk'] = dim_date_data['fulldate'].dt.strftime('%Y%m%d').astype(int)
15    dim_date_data['year'] = dim_date_data['fulldate'].dt.year
16    # ... (tính month, dayofweek)
17
18    return dim_customer_data, dim_product_data, dim_date_data
```

2.3.2 Giai đoạn 2: Tải Dimension

Đây là bước nạp dữ liệu Dimension vào Database.

```
1     # Load dim_date
2     dim_date_data[['date_sk', 'fulldate', 'year', 'month', 'dayofweek']].to_sql(
3         'dim_date', db_engine, if_exists='append', index=False, chunksize=1000
4     )
5
6     # Load dim_product
7     dim_product_data[['stockcode', 'description']].to_sql(
8         'dim_product', db_engine, if_exists='append', index=False, chunksize=1000
9     )
10
11    # Load dim_customer
12    dim_customer_data[['customerid', 'country']].to_sql(
13        'dim_customer', db_engine, if_exists='append', index=False, chunksize=1000
14    )
```

2.3.3 Giai đoạn 3: Ánh xạ Khóa Thay thế (SK Mapping) và Tải Fact

Để nạp bảng Fact (`fact_sales`), chúng ta cần thay thế các Khóa Nghiệp vụ trong dữ liệu giao dịch bằng Khóa Thay thế (Surrogate Key - SK) tương ứng từ Database.

- **Bước 1: Đọc SK từ Database.** Truy vấn ngược lại DB để lấy cặp giá trị (Business Key, Surrogate Key) cho cả 3 chiều Customer, Product, Date.
- **Bước 2: Merge (JOIN).** Sử dụng phép LEFT JOIN trong Pandas để gắn các SK vào bảng giao dịch chính.



- **Bước 3: Tải Fact.** Nạp dữ liệu đã có đú SK vào bảng fact_sales.

```
1 # 1. Đọc lại SK và Khóa nghiệp vụ từ DB (Look-up Tables)
2 with db_engine.connect() as conn:
3     dim_cust_map = pd.read_sql_query('SELECT customer_sk, customerid FROM dim_customer', conn)
4     dim_prod_map = pd.read_sql_query('SELECT product_sk, stockcode FROM dim_product', conn)
5     dim_date_map = pd.read_sql_query('SELECT date_sk, fulldate FROM dim_date', conn)
6
7     # 2. Thực hiện MERGE (tương đương LEFT JOIN trong SQL) để gắn SK
8     # Gắn Customer SK
9     df_fact = pd.merge(df_transactions, dim_cust_map, on='customerid', how='left')
10    # Gắn Product SK
11    df_fact = pd.merge(df_fact, dim_prod_map, on='stockcode', how='left')
12    # Gắn Date SK
13    df_fact = pd.merge(df_fact, dim_date_map, on='fulldate', how='left')
14
15    # 3. Kiểm tra tính toàn vẹn và Tải vào Fact Table
16    df_fact = df_fact.dropna(subset=['product_sk', 'customer_sk', 'date_sk'])
17
18    fact_cols = ['invoiceno', 'quantity', 'unitprice', 'totalrevenue',
19                  'date_sk', 'product_sk', 'customer_sk']
20
21    df_fact[fact_cols].to_sql(
22        'fact_sales', db_engine, if_exists='append', index=False, method='multi'
23    )
```

3 PHÂN TÍCH DỮ LIỆU VÀ ỨNG DỤNG THUẬT TOÁN

Sau khi dữ liệu đã được nạp thành công vào Kho dữ liệu, chúng tôi tiến hành khai thác thông tin thông qua hai phương pháp: Phân tích xử lý trực tuyến (OLAP) để đánh giá hiệu suất quá khứ và Khai phá dữ liệu (Data Mining) để tìm kiếm quy luật ẩn.

3.1 Phân tích Hiệu suất Tổng hợp (OLAP)

Các truy vấn OLAP dưới đây được thực hiện trực tiếp trên bảng Fact và Dimension để trả lời các câu hỏi nghiệp vụ cốt lõi.

3.1.1 Phân tích 1: Chỉ số KPI Tổng hợp

Câu hỏi nghiệp vụ: Tình hình kinh doanh tổng thể của doanh nghiệp hiện tại như thế nào dựa trên doanh thu, số lượng đơn hàng và giá trị trung bình mỗi đơn hàng?

Ý nghĩa chiến lược:

- **Sức khỏe tài chính:** Tổng doanh thu phản ánh quy mô dòng tiền vào.
- **Hiệu quả kinh doanh:** AOV (Average Order Value) là chỉ số quan trọng để đánh giá khả năng chi tiêu của khách hàng. AOV cao cho thấy chiến lược bán hàng hiệu quả hoặc khách hàng có xu hướng mua sỉ.

```
1 SELECT
2     SUM(fs.totalrevenue) AS "Total Revenue (USD)",
3     COUNT(DISTINCT fs.invoiceno) AS "Total Transactions",
4     (SUM(fs.totalrevenue) / COUNT(DISTINCT fs.invoiceno)) AS "AOV (USD)"
5 FROM fact_sales fs;
```

	Total Revenue (USD) numeric	Total Transactions bigint	AOV (USD) numeric
1	10666684.5400	19960	534.4030330661322645

Hình 3: Kết quả truy vấn KPI Tổng hợp



Đánh giá kết quả: Từ kết quả truy vấn (Hình 3), ta có các chỉ số cốt lõi sau:

- **Tổng Doanh thu (Total Revenue):** Đạt mức **10,666,684.54 USD**. Đây là con số ấn tượng cho thấy quy mô kinh doanh lớn của doanh nghiệp trong giai đoạn khảo sát.
- **Tổng số Giao dịch:** Có **19,960** hóa đơn (Invoice) được ghi nhận.
- **Giá trị Đơn hàng Trung bình (AOV):** Đạt **534.40 USD/dơn**.
 - *Nhân định:* Mức AOV này khá cao so với bán lẻ B2C thông thường (thường dao động 50-100 USD). Điều này cũng có giả thuyết rằng tập khách hàng của công ty bao gồm một tỷ lệ lớn là **nha bán buon (Wholesalers)** hoặc khách hàng mua số lượng lớn, đúng như mô tả của bộ dữ liệu.

3.1.2 Phân tích 2: Xu hướng Doanh thu theo Thời gian

Câu hỏi nghiệp vụ: Doanh thu biến động như thế nào theo từng tháng và từng năm? Có tồn tại tính mùa vụ trong hành vi mua sắm của khách hàng không?

Ý nghĩa chiến lược:

- **Quản lý tồn kho:** Nhận diện các tháng cao điểm để lên kế hoạch nhập hàng sớm.
- **Phân bổ ngân sách Marketing:** Tập trung ngân sách quảng cáo vào các tháng có xu hướng mua sắm mạnh.

```
1   SELECT
2       dd.year,
3       dd.month,
4       SUM(fs.totalrevenue) AS "Monthly Revenue"
5   FROM fact_sales fs
6   JOIN dim_date dd ON fs.date_sk = dd.date_sk
7   GROUP BY dd.year, dd.month
8   ORDER BY dd.year, dd.month;
```

	year integer	month integer	Monthly Revenue numeric
1	2010	12	823746.1400
2	2011	1	691364.5600
3	2011	2	523631.8900
4	2011	3	717639.3600
5	2011	4	537808.6200
6	2011	5	770536.0200
7	2011	6	761739.9000
8	2011	7	719221.1900
9	2011	8	759138.3800
10	2011	9	1058590.1700
11	2011	10	1154979.3000
12	2011	11	1509496.3300
13	2011	12	638792.6800

Hình 4: Biến động Doanh thu theo Tháng (12/2010 - 12/2011)



Dánh giá kết quả: Dữ liệu (Hình 4) cho thấy xu hướng rõ rệt của tính mùa vụ:

- **Tăng trưởng Quý 4 (Peak Season):** Doanh thu bắt đầu tăng mạnh từ Tháng 9 (1.05M USD) và đạt đỉnh vào **Tháng 11/2011** với mức kỷ lục **1,509,496 USD**. Đây là giai đoạn chuẩn bị cho các dịp lễ lớn (Black Friday, Giáng sinh).
- **Sự sụt giảm vào Tháng 12:** Doanh thu Tháng 12/2011 chỉ đạt 638,792 USD. *Lý do:* Dữ liệu giao dịch chỉ được ghi nhận đến ngày 09/12/2011 (theo mô tả bộ dữ liệu), do đó đây không phải là sự suy thoái mà là do dữ liệu chưa trọn vẹn tháng.
- **Giai đoạn thấp điểm:** Các tháng đầu năm (Tháng 2, Tháng 4) có doanh thu thấp nhất, dao động quanh mức 500,000 USD. Đây là thời điểm doanh nghiệp nén tung ra các chương trình khuyến mãi kích cầu.

3.1.3 Phân tích 3: Phân tích Địa lý và Thị trường Trọng tâm

Câu hỏi nghiệp vụ: Doanh thu của công ty đến từ những quốc gia nào là chủ yếu? Mức độ tập trung của thị trường vào các quốc gia đứng đầu là bao nhiêu?

Ý nghĩa chiến lược:

- **Tối ưu hóa Logistics:** Nếu doanh thu tập trung quá lớn vào một quốc gia, cần tối ưu hóa hệ thống kho bãi tại đó.
- **Dánh giá rủi ro:** Mức độ phụ thuộc thị trường.

```
1 WITH TotalRevenue AS (
2     SELECT SUM(totalrevenue) AS TotalRev FROM fact_sales
3 )
4 SELECT
5     dc.country AS "Country",
6     SUM(fs.totalrevenue) AS "Revenue by Country",
7     (SUM(fs.totalrevenue) * 100.0 / (SELECT TotalRev FROM TotalRevenue))
8     AS "Revenue Share (%)"
9 FROM fact_sales fs
10 JOIN dim_customer dc ON fs.customer_sk = dc.customer_sk
11 GROUP BY dc.country
12 ORDER BY "Revenue by Country" DESC
13 LIMIT 5;
```

	Country character varying (100)	Revenue by Country numeric	Revenue Share (%) numeric
1	United Kingdom	9063668.1900	84.9717469004665999
2	Netherlands	285446.3400	2.6760549534354186
3	EIRE	265545.9000	2.4894886410505958
4	Germany	228867.1400	2.1456258422356981
5	France	209024.0500	1.9595971851999665

Hình 5: Top 5 Quốc gia có Doanh thu cao nhất

Dánh giá kết quả: Kết quả (Hình 5) cho thấy sự chênh lệch rất lớn về địa lý:

- **Sự thống trị của thị trường Anh (UK):** United Kingdom đóng góp tới **9,063,668 USD**, chiếm xấp xỉ **85%** tổng doanh thu toàn cầu. Điều này khẳng định đây là thị trường nội địa cốt lõi.
- **Thị trường Quốc tế:** Netherlands, EIRE (Ireland), Germany và France là các thị trường tiếp theo nhưng khoảng cách rất xa (chỉ chiếm khoảng 2% - 2.6% mỗi nước).



- **Khuyến nghị:** Doanh nghiệp nên duy trì vị thế tại UK nhưng cần có chiến lược thâm nhập mạnh hơn vào các thị trường Châu Âu lân cận (Đức, Pháp) để giảm bớt rủi ro phụ thuộc duy nhất vào một thị trường.

3.1.4 Phân tích 4: Hiệu suất Sản phẩm

Câu hỏi nghiệp vụ: Những sản phẩm nào đang là "ngôi sao" mang lại doanh thu cao nhất cho công ty?

Ý nghĩa chiến lược:

- **Nguyên lý Pareto (80/20):** Xác định nhóm sản phẩm chủ lực.
- **Chiến lược Bundle:** Các sản phẩm này là ứng cử viên sáng giá để làm sản phẩm mới.

```
1   SELECT
2       dp.stockcode AS "Stock Code",
3       dp.description AS "Product Description",
4       SUM(fs.totalrevenue) AS "Total Revenue"
5   FROM fact_sales fs
6   JOIN dim_product dp ON fs.product_sk = dp.product_sk
7   GROUP BY 1, 2
8   ORDER BY "Total Revenue" DESC
9   LIMIT 10;
```

	Stock Code character varying (20)	Product Description character varying (255)	Total Revenue numeric
1	DOT	DOTCOM POSTAGE	206248.7700
2	22423	REGENCY CAKESTAND 3 TIER	174484.7400
3	23843	PAPER CRAFT , LITTLE BIRDIE	168469.6000
4	85123A	WHITE HANGING HEART T-LIGHT HOLDER	104518.8000
5	47566	PARTY BUNTING	99504.3300
6	85099B	JUMBO BAG RED RETROSPOT	94340.0500
7	23166	MEDIUM CERAMIC TOP STORAGE JAR	81700.9200
8	M	Manual	78110.2700
9	POST	POSTAGE	78101.8800
10	23084	RABBIT NIGHT LIGHT	66964.9900

Hình 6: Top 10 Sản phẩm/Dịch vụ theo Doanh thu

Dánh giá kết quả: Danh sách Top 10 (Hình 6) tiết lộ nhiều thông tin thú vị về cơ cấu sản phẩm:

- **Dịch vụ vận chuyển (DOTCOM POSTAGE):** Đầu danh sách với doanh thu 206,248 USD. Điều này cho thấy chi phí vận chuyển đóng góp một phần đáng kể vào dòng tiền, phản ánh mô hình kinh doanh Online Retail.
- **Sản phẩm bán chạy nhất:** "REGENCY CAKESTAND 3 TIER" là mặt hàng vật lý bán chạy nhất (174,484 USD). Đây là sản phẩm trang trí/nhà bếp cao cấp.
- **Các mặt hàng thủ công và quà tặng:** "PAPER CRAFT, LITTLE BIRDIE" và "WHITE HANGING HEART T-LIGHT HOLDER" cũng nằm trong top, xác nhận thị hiếu khách hàng thiên về đồ trang trí và quà tặng sự kiện.



- **Dữ liệu nhiễu:** Mã sản phẩm "*Manual*" (78,110 USD) xuất hiện trong top. Đây thường là các khoản điều chỉnh thủ công của nhân viên (hoàn tiền, đền bù, phí dịch vụ ngoài). Cần loại trừ mã này khi phân tích tồn kho thực tế.



3.2 Khai phá Dữ liệu (Data mining) với Thuật toán Apriori (Market Basket Analysis)

3.2.1 Mục tiêu và Phương pháp

Mục tiêu của phân tích giỏ hàng (Market Basket Analysis - MBA) là phát hiện các mối quan hệ ẩn giữa các sản phẩm thường xuyên được mua cùng nhau.

- **Thuật toán: Apriori.** Đây là thuật toán phổ biến nhất để tìm kiếm các tập mục phổ biến (Frequent Itemsets) và sinh ra các luật kết hợp (Association Rules).
- **Công cụ:** Thư viện Python mlxtend.
- **Đầu vào:** Dữ liệu giao dịch từ bảng fact_sales và dim_product.

3.2.2 Bước 1: Chuẩn bị Dữ liệu

Thuật toán Apriori yêu cầu dữ liệu đầu vào dưới dạng ma trận "One-Hot Encoded" (hoặc Transaction Basket), nơi mỗi hàng là một hóa đơn và mỗi cột là một sản phẩm.

```
1 def prepare_mba_input(engine):
2     # 1. Truy vấn dữ liệu: Lấy InvoiceNo và Description
3     query = """
4         SELECT fs.invoiceno, dp.description
5         FROM fact_sales fs
6         JOIN dim_product dp ON fs.product_sk = dp.product_sk
7         """
8     df = pd.read_sql(query, engine)
9
10    # 2. Pivot Table: Chuyển đổi sang ma trận One-Hot
11    # Index: InvoiceNo, Columns: Description, Values: 1 (có mua) hoặc 0 (không mua)
12    basket = (df.groupby(['invoiceno', 'description'])['invoiceno']
13               .count().unstack().reset_index().fillna(0)
14               .set_index('invoiceno'))
15
16    # 3. Encoding: Đảm bảo giá trị là nhị phân (0/1)
17    def encode_units(x):
18        return 1 if x >= 1 else 0
19    basket_sets = basket.applymap(encode_units)
20
21    return basket_sets
```

3.2.3 Bước 2: Thực thi Thuật toán Apriori

Chúng tôi thiết lập các siêu tham số (Hyperparameters) để lọc nhiễu và tìm ra các quy tắc có ý nghĩa thực tiễn.

- **Min Support = 0.015 (1.5%):** Sản phẩm phải xuất hiện trong ít nhất 1.5% tổng số hóa đơn. Điều này giúp loại bỏ các sản phẩm quá hiếm gặp.
- **Min Confidence = 0.5 (50%):** Nếu khách mua A, phải có ít nhất 50% khả năng họ mua B.
- **Min Lift = 1.2:** Chỉ chấp nhận các mối quan hệ tích cực (việc mua A làm tăng khả năng mua B lên ít nhất 20% so với ngẫu nhiên).

```
1 from mlxtend.frequent_patterns import apriori, association_rules
2
3 def run_mba(basket_sets):
4     # 1. Tìm Frequent Itemsets
5     frequent_itemsets = apriori(basket_sets, min_support=0.015, use_colnames=True)
6
7     # 2. Sinh Association Rules
8     rules = association_rules(frequent_itemsets, metric="lift", min_threshold=1.2)
9
10    # 3. Lọc theo Confidence
11    rules = rules[rules['confidence'] >= 0.5]
```



13

```
return rules[['antecedents', 'consequents', 'support', 'confidence', 'lift']]
```

3.2.4 Kết quả Luật Kết hợp (Top Association Rules)

Dựa trên ngữ cảnh lọc $Support \geq 0.015$, $Confidence \geq 0.5$ và $Lift \geq 1.2$, thuật toán đã trích xuất được các quy tắc có độ mạnh rất cao. Bảng dưới đây liệt kê 6 quy tắc hàng đầu, được sắp xếp theo độ đo Lift.

Sản phẩm A (Antecedents)	Sản phẩm B (Consequents)	Support	Conf	Lift
REGENCY TEA PLATE GREEN	REGENCY TEA PLATE ROSES	0.016	0.84	37.55
REGENCY TEA PLATE ROSES	REGENCY TEA PLATE GREEN	0.016	0.72	37.55
POPPY'S PLAYHOUSE KITCHEN	POPPY'S PLAYHOUSE BEDROOM	0.016	0.72	33.87
POPPY'S PLAYHOUSE BEDROOM	POPPY'S PLAYHOUSE KITCHEN	0.016	0.74	33.87
SET/6 RED SPOTTY PAPER PLATES	SET/6 RED SPOTTY PAPER CUPS	0.018	0.66	30.97
SET/6 RED SPOTTY PAPER CUPS	SET/6 RED SPOTTY PAPER PLATES	0.018	0.82	30.97

Bảng 5: Top Quy tắc Kết hợp Mạnh nhất (Lift > 30)

3.2.5 Phân tích Chi tiết: Tại sao Lift lại cao đột biến (> 30)

Chỉ số **Lift** đo lường mức độ tương quan giữa hai sản phẩm A và B so với sự xuất hiện ngẫu nhiên.

$$Lift(A \rightarrow B) = \frac{P(A \cap B)}{P(A) \times P(B)}$$

Nếu $Lift = 1$, hai sản phẩm độc lập. Nếu $Lift > 1$, chúng có mối quan hệ tích cực. Các kết quả trên cho thấy Lift dao động từ **30.97** đến **37.55**, một con số cực kỳ cao. Điều này không phải ngẫu nhiên mà phản ánh 3 đặc tính mua sắm cốt lõi của ngành hàng quà tặng/lưu niệm:

1. Đồng bộ cho Bộ Sưu tập:

- *Ví dụ: POPPY'S PLAYHOUSE (KITCHEN ↔ BEDROOM).*
- *Giải thích:* Đây là dòng sản phẩm đồ chơi mô hình (dollhouse). Khách hàng không mua một "nhà bếp" riêng lẻ làm gì cả; họ mua để hoàn thiện một ngôi nhà búp bê. Việc mua một phòng (Kitchen) tạo ra nhu cầu gần như bắt buộc phải mua phòng còn lại (Bedroom). Đây là mối quan hệ **Bổ sung Hoàn hảo**.

2. Thẩm mỹ và Phối màu:

- *Ví dụ: REGENCY TEA PLATE (GREEN ↔ ROSES).*
- *Giải thích:* "Regency" là tên một phong cách thiết kế trà chiều cổ điển. Người mua những sản phẩm này thường là người sưu tầm hoặc mua để trang trí. Họ có xu hướng mua trọn bộ các biến thể màu sắc (Xanh và Hồng) để trưng bày cùng nhau. Hành vi này tạo ra sự kết dính chặt chẽ giữa các biến thể sản phẩm (Product Variants), đẩy Lift lên tới **37.55**.

3. Chức năng cho Sự kiện/Tiệc tùng:



- **Ví dụ: RED SPOTTY PAPER (PLATES ↔ CUPS).**
- **Giải thích:** Đây là sản phẩm dùng cho tiệc tùng. Khi tổ chức tiệc, người dùng cần cả đĩa (Plates) để đựng thức ăn và cốc (Cups) để đựng đồ uống. Hơn nữa, họ cần chúng phải **đồng bộ về họa tiết** (Red Spotty - Chấm bi đỏ). Xác suất một người mua đĩa chấm bi đỏ mà lại mua cốc màu khác là rất thấp, dẫn đến Confidence (Độ tin cậy) rất cao (0.82) và Lift đạt **30.97**.

Kết luận: Thuật toán Apriori đã hoạt động hiệu quả trong việc "tái hiện" lại cấu trúc danh mục sản phẩm của doanh nghiệp. Những con số Lift > 30 này là bằng chứng thép cho thấy chiến lược bán hàng theo **Combo (Bundling)** là con đường ngắn nhất để tối ưu hóa doanh thu.

3.2.6 Kiểm chứng Giả thuyết từ Kho Dữ liệu (Validation)

Để xác thực giả thuyết "Aesthetic Matching", chúng tôi thực hiện truy vấn kiểm chứng trực tiếp trên Data Warehouse đối với dòng sản phẩm *REGENCY TEA PLATE*.

```
1   WITH regency_stats AS (
2     SELECT
3       fs.invoiceno,
4       MAX(CASE WHEN dp.description = 'REGENCY TEA PLATE GREEN' THEN 1 ELSE 0 END) as has_green,
5       MAX(CASE WHEN dp.description = 'REGENCY TEA PLATE ROSES' THEN 1 ELSE 0 END) as has_roses
6     FROM fact_sales fs
7     JOIN dim_product dp ON fs.product_sk = dp.product_sk
8     WHERE dp.description IN ('REGENCY TEA PLATE GREEN', 'REGENCY TEA PLATE ROSES')
9     GROUP BY fs.invoiceno
10   )
11   SELECT
12     COUNT(*) as total_orders_involving_regency,
13     SUM(CASE WHEN has_green = 1 AND has_roses = 0 THEN 1 ELSE 0 END) as green_only,
14     SUM(CASE WHEN has_green = 0 AND has_roses = 1 THEN 1 ELSE 0 END) as roses_only,
15     SUM(CASE WHEN has_green = 1 AND has_roses = 1 THEN 1 ELSE 0 END) as bought_both
16   FROM regency_stats;
```

Kết quả thực nghiệm:

	total_orders_involving_regency bigint	green_only bigint	roses_only bigint	bought_both bigint
1	507	62	126	319

- Tổng số hóa đơn có mua ít nhất một loại đĩa Regency: **507**.
- Số hóa đơn chỉ mua màu Xanh (Green Only): **62** (12.2%).
- Số hóa đơn chỉ mua màu Hồng (Roses Only): **126** (24.8%).
- Số hóa đơn mua **CẢ HAI** (Bought Both): **319** (63%).

Kết luận: Tỷ lệ mua trùng bộ lên tới **63%** (cao gấp đôi so với mua lẻ từng màu) đã xác nhận định lượng rằng mối quan hệ Lift cao là do hành vi mua sắm có chủ đích, không phải ngẫu nhiên.

3.2.7 Đánh giá Tác động Kinh doanh: Ma trận Lift và Support

Cần cân bằng giữa Lift (Độ mạnh quan hệ) và Support (Độ phổ biến) để đánh giá đúng tác động doanh thu. Dựa trên dữ liệu thực tế từ 351 luật tìm được:

Sản phẩm (Cặp ví dụ)	Lift	Support	Đánh giá Tác động
----------------------	------	---------	-------------------



Sản phẩm (Cặp ví dụ)	Lift	Support	Dánh giá Tác động
REGENCY TEA PLATE	> 20	< 1.0%	Niche Market: Quan hệ cực mạnh nhưng ít giao dịch. Phù hợp bán Bundle giá trị cao.
JUMBO SHOPPER VINTAGE RED PAISLEY	5.53	3.41%	Mass Market: Support cao (>3%), Lift thấp hơn nhưng mang lại tổng doanh thu lớn do là hàng Best-seller.
JUMBO BAG RED RET-ROSPOT	10-20	≈ 1-2%	Balanced: Cân bằng giữa độ mạnh và độ phổ biến, thích hợp cho Cross-sell.

Bảng 6: So sánh Hiệu quả giữa Lift và Support

3.2.8 Phân loại Quy tắc và Đề xuất Chiến lược (Actionable Insights)

Dựa trên phân tích phân bố của **351 luật kết hợp** được tìm thấy từ dữ liệu, chúng tôi phân loại chúng thành 3 nhóm chiến lược:

- **Tổng số luật tìm được:** 351 luật.
- **Nhóm 1: Bundle Strategy (Lift > 20): 12 luật (3.4%).**
 - *Dặc điểm:* Quan hệ cực mạnh, các sản phẩm này gần như luôn được mua cùng nhau (tỷ lệ ngẫu nhiên rất thấp).
 - *Hành động:* Dóng gói cứng (Hard bundle) hoặc giảm giá khi mua trọn bộ để tối ưu hóa giá trị đơn hàng.
- **Nhóm 2: Cross-sell Strategy (10 ≤ Lift ≤ 20): 149 luật (42.5%).**
 - *Dặc điểm:* Chiếm tỷ trọng lớn và có độ tin cậy cao. Thường là các sản phẩm bổ sung (Complementary) như đồ trang trí cùng phong cách.
 - *Hành động:* Sử dụng làm nòng cốt cho hệ thống gợi ý (Recommendation Engine) trên website (mục "Thường được mua cùng").
- **Nhóm 3: Monitor Strategy (Lift < 10): 190 luật (54.1%).**
 - *Dặc điểm:* Chiếm đa số (>50%). Đây là các sản phẩm phổ biến, khách hàng mua vì nhu cầu cao (như các loại túi Jumbo) hơn là do mối liên kết đặc thù.
 - *Hành động:* Sắp xếp vị trí gần nhau để tăng sự tiện lợi (Traffic flow), không cần ưu tiên giảm giá sâu.

3.2.9 Hạn chế và Hướng phát triển

Hạn chế: Thuật toán Apriori hiện tại chưa xét đến yếu tố thời gian (Seasonality) và số lượng mua (Quantity), có thể bỏ qua các quy tắc chỉ xuất hiện vào mùa Giáng sinh.

Hướng phát triển: Áp dụng **Temporal Association Rules** để phân tích MBA theo từng Quý, nhằm phát hiện các combo quà tặng đặc thù cho dịp lễ cuối năm.

4 TRỰC QUAN HÓA DỮ LIỆU VÀ HỆ HỖ TRỢ QUYẾT ĐỊNH

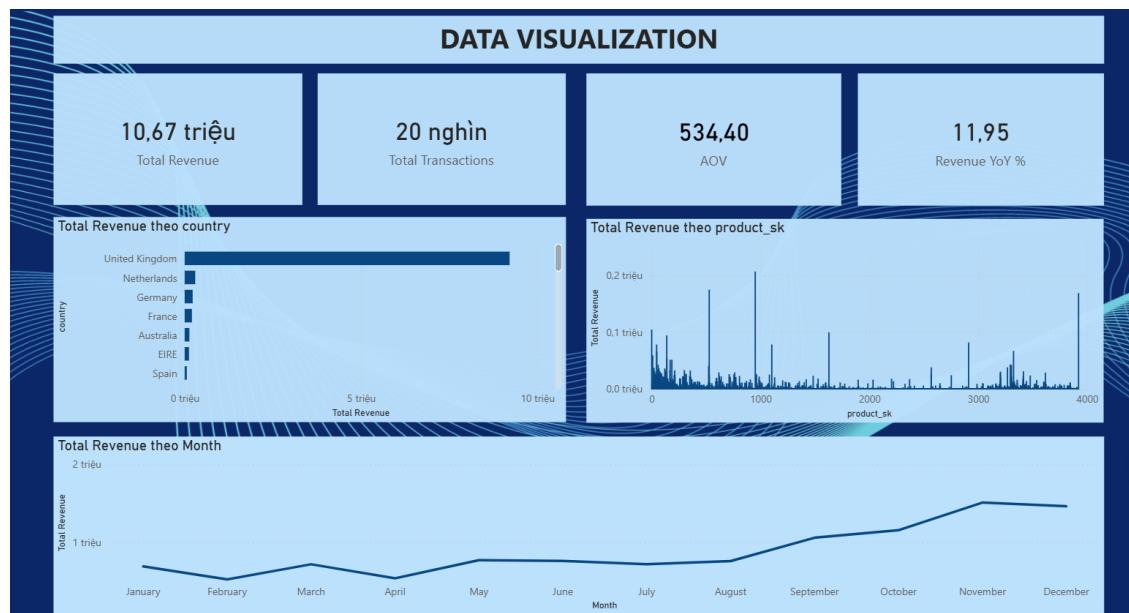
4.1 Công cụ thực hiện

Ở đây để trực quan hóa dữ liệu một cách cụ thể và chi tiết em đã lựa chọn công cụ Power BI của microsoft để xây dựng dashboard.

Power BI được lựa chọn nhờ khả năng kết nối mạnh mẽ với nhiều nguồn dữ liệu, tính tương tác cao (Interactive Drilling), và khả năng trực quan hóa sinh động, giúp ban lãnh đạo dễ dàng nắm bắt bức tranh toàn cảnh về hiệu suất kinh doanh. Hệ thống bao gồm 2 Dashboard chính: **Dashboard Hiệu suất Tổng thể** và **Dashboard Phân tích Giảm hàng**.

4.2 Dashboard 1: Hiệu suất Kinh doanh Tổng thể

Dashboard này cung cấp cái nhìn toàn cảnh về sức khỏe của doanh nghiệp thông qua các chỉ số KPI cốt lõi và các biểu đồ xu hướng theo thời gian thực.



Hình 7: Dashboard phân tích hiệu suất kinh doanh tổng thể trên Power BI

4.2.1 Các thành phần trực quan hóa

Dựa trên Hình 7, các thành phần chính bao gồm:

- **KPI Cards:** Hiển thị các con số quan trọng nhất tại thời điểm báo cáo:
 - **Total Revenue:** 10.67 triệu (Tổng doanh thu).
 - **Total Transactions:** 20 nghìn (Tổng số đơn hàng).
 - **AOV (Average Order Value):** 534.40 (Giá trị trung bình đơn hàng - mức khá cao, cho thấy khách hàng mua số lượng lớn hoặc sản phẩm giá trị cao).
 - **Revenue YoY %:** 11.95 (Tăng trưởng so với cùng kỳ).

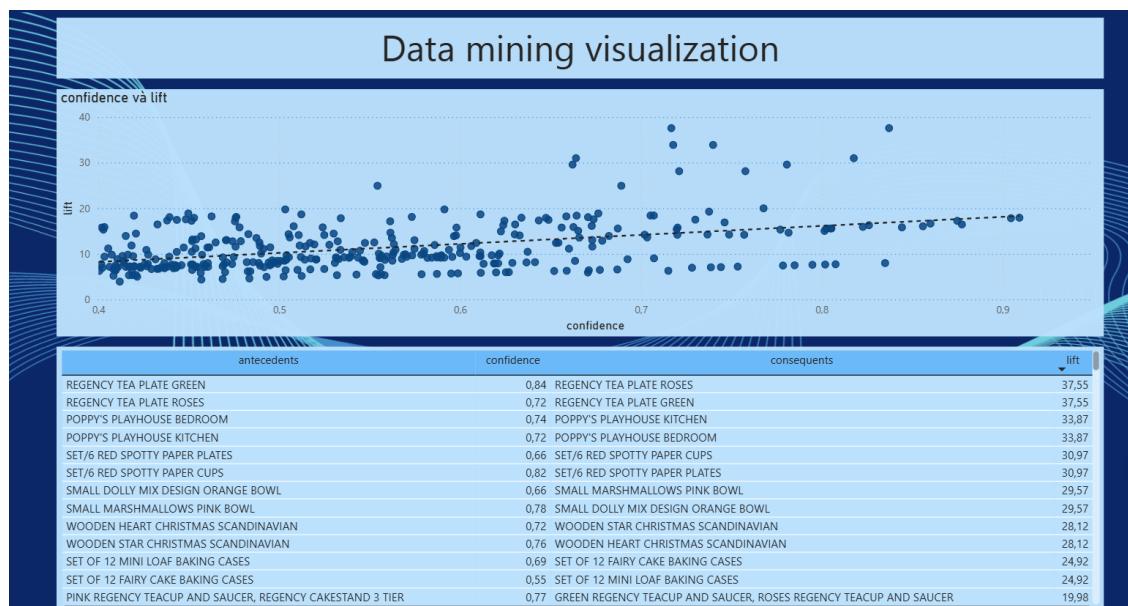
- **Total Revenue theo Country (Biểu đồ Cột ngang):** Thể hiện sự phân bổ doanh thu theo địa lý. United Kingdom chiếm tỷ trọng áp đảo, tiếp theo là Netherlands, Germany và France.
- **Total Revenue theo Month (Biểu đồ Đường):** Thể hiện tính mùa vụ của doanh thu.

4.2.2 Insight và Hỗ trợ Quyết định

- **Nhận định:** Biểu đồ xu hướng cho thấy doanh thu duy trì ổn định ở mức thấp từ tháng 1 đến tháng 8, sau đó bắt đầu tăng trưởng nóng từ **Tháng 9** và đạt đỉnh vào **Tháng 11** (giai đoạn chuẩn bị cho Giáng sinh và Năm mới).
- **Hành động đề xuất:**
 - **Quản lý Tồn kho:** Cần lên kế hoạch nhập hàng và dự trữ tồn kho tối đa vào cuối tháng 8 để đáp ứng nhu cầu tăng vọt trong Quý 4.
 - **Marketing:** Tập trung ngân sách quảng cáo lớn nhất vào thị trường Anh (UK) vào Tháng 10 và đầu Tháng 11. Đối với các thị trường ngách như Đức hay Pháp, cần có chiến dịch riêng biệt để tăng thị phần.

4.3 Dashboard 2: Phân tích Giỏ hàng và Chiến lược Bán chéo

Dashboard này trực quan hóa kết quả của thuật toán Apriori, giúp bộ phận bán hàng nhận diện các mẫu mua sắm ẩn và thiết kế các gói combo sản phẩm hiệu quả.



Hình 8: Trực quan hóa kết quả khai phá luật kết hợp (Association Rules)

4.3.1 Các thành phần trực quan hóa

Dựa trên Hình 8:

- **Scatter Plot (Biểu đồ Phân tán - Confidence và Lift):**
 - Trục hoành (X): Confidence (Độ tin cậy).
 - Trục tung (Y): Lift (Độ mạnh của luật).



- Đường xu hướng (Trendline) cho thấy mối tương quan dương: Confidence càng cao thì Lift thường cũng càng cao.
- **Data Table:** Liệt kê chi tiết các luật tìm được. Ví dụ:
 - *Regency Tea Plate Green* → *Regency Tea Plate Roses* (Lift: 37.55, Confidence: 0.84).
 - *Poppy's Playhouse Bedroom* → *Poppy's Playhouse Kitchen* (Lift: 33.87).

4.3.2 Kịch bản Ứng dụng Chiến lược

Dựa trên kết quả từ Dashboard trên, em đề xuất 3 chiến lược bán chéo cụ thể:

Chiến lược 1: Combo Đồng bộ

- **Cơ sở dữ liệu:** Các cặp sản phẩm có **Lift rất cao (> 30)** như bộ đĩa *Regency Tea Plate* hoặc bộ đồ chơi *Poppy's Playhouse*. Đây là các sản phẩm mang tính sưu tập hoặc bổ trợ trực tiếp.
- **Hành động:** Đóng gói cung thành một SKU mới (ví dụ: "Full Set Regency Tea") với giá ưu đãi giảm 5-10% so với mua lẻ.
- **Mục tiêu:** Tối đa hóa giá trị đơn hàng (AOV) và thanh lý nhanh các biến thể màu sắc ít phô biến hơn bằng cách bán kèm.

Chiến lược 2: Gợi ý tại Quầy

- **Cơ sở dữ liệu:** Các cặp sản phẩm có **Confidence > 70%** và Lift ở mức trung bình khá (10-20).
- **Hành động:** Khi khách hàng thêm sản phẩm A vào giỏ hàng, hệ thống website sẽ tự động hiển thị popup: "Khách hàng mua A thường mua kèm B".
- **Mục tiêu:** Tăng tỷ lệ chuyển đổi dựa trên tâm lý đám đông.

Chiến lược 3: Sắp xếp Cửa hàng

- **Hành động:** Dựa trên biểu đồ phân tán, nhóm các sản phẩm có mối liên kết về mặt vật lý. Ví dụ: Dặt các loại *Baking Cases* (Khuôn làm bánh) gần nhau trên danh mục website.
- **Mục tiêu:** Tăng khả năng mua sắm ngẫu hứng của khách hàng.



5 HẠN CHẾ VÀ HƯỚNG PHÁT TRIỂN

Mặc dù dự án đã đạt được những kết quả khả quan trong việc xây dựng hệ thống báo cáo và khai phá dữ liệu, em nhận thấy vẫn còn một số hạn chế cần khắc phục trong tương lai.

5.1 Hạn chế của đề tài

- Chất lượng dữ liệu đầu vào:** Bộ dữ liệu *Online Retail* chứa một lượng lớn các bản ghi thiếu thông tin định danh khách hàng ('CustomerID' bị null). Điều này làm giảm độ chính xác khi phân tích hành vi cá nhân hóa và bắt buộc cần phải loại bỏ một phần dữ liệu đáng kể trong quá trình tiền xử lý (ETL).
- Giới hạn của thuật toán Apriori:** Thuật toán hiện tại xử lý tốt trên tập dữ liệu tĩnh nhưng tồn kém tài nguyên tính toán khi dữ liệu mở rộng lên quy mô lớn (Big Data). Ngoài ra, Apriori truyền thống chưa xét đến yếu tố thời gian (thứ tự mua hàng) và số lượng sản phẩm trong giỏ, dẫn đến việc bỏ sót các luật kết hợp mang tính chuỗi (Sequential Patterns).
- Phạm vi phân tích:** Dashboard hiện tại tập trung chủ yếu vào góc độ bán hàng (Sales) và sản phẩm (Product). Các khía cạnh khác như phân khúc khách hàng (Customer Segmentation - RFM) hay dự báo nhu cầu (Forecasting) chưa được khai thác sâu.

5.2 Hướng phát triển

Để nâng cao tính ứng dụng thực tiễn, em đề xuất các hướng phát triển tiếp theo cho giai đoạn 2 của dự án:

- Áp dụng các thuật toán nâng cao:**
 - Sử dụng thuật toán **FP-Growth** để cải thiện tốc độ xử lý so với Apriori.
 - Triển khai **Temporal Association Rules** để phát hiện các quy luật mua sắm theo mùa vụ (ví dụ: các combo quà tặng chỉ xuất hiện vào dịp Giáng sinh).
- Tích hợp Phân tích RFM:** Kết hợp kết quả MBA với mô hình RFM (Recency, Frequency, Monetary) để đưa ra các gợi ý sản phẩm riêng biệt cho từng nhóm khách hàng (khách hàng VIP, khách hàng mới, khách hàng có nguy cơ rời bỏ).
- Xây dựng hệ thống Real-time:** Nâng cấp kiến trúc hệ thống để hỗ trợ xử lý dữ liệu dòng (Streaming Data), cho phép cập nhật Dashboard và đưa ra gợi ý ngay lập tức khi giao dịch vừa phát sinh.

6 KẾT LUẬN

Dự án "Xây dựng Kho dữ liệu và Khai phá luật kết hợp cho hệ thống E-commerce" đã hoàn thành trọn vẹn quy trình từ xử lý dữ liệu thô đến hỗ trợ ra quyết định. Thông qua việc áp dụng quy trình ETL, xây dựng mô hình dữ liệu Star Schema và ứng dụng thuật toán Data Mining, em đã rút ra được những kết luận quan trọng:

- Về mặt kỹ thuật:** Tìm hiểu thêm về quy trình xây dựng Data Warehouse và sử dụng thành thạo các công cụ hiện đại như Python, SQL và Power BI để trực quan hóa dữ liệu phức tạp.
- Về mặt kinh doanh:** Đã chỉ ra được các điểm nóng về doanh thu theo thời gian và địa lý, đồng thời phát hiện ra các quy luật mua sắm (Association Rules) có giá trị cao. Những quy luật này (ví dụ: các cặp sản phẩm có Lift > 20) là cơ sở vững chắc để doanh nghiệp xây dựng chiến lược đóng gói sản phẩm (Bundling) và bán chéo (Cross-selling).



Tài liệu

- [1] Jiawei Han, Micheline Kamber, and Jian Pei. *Data Mining: Concepts and Techniques*. 3rd Edition, Morgan Kaufmann, 2011.
- [2] Ralph Kimball and Margy Ross. *The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling*. 3rd Edition, Wiley, 2013.
- [3] Rakesh Agrawal and Ramakrishnan Srikant. "Fast Algorithms for Mining Association Rules". *Proc. 20th Int. Conf. Very Large Data Bases, VLDB*, pp. 487-499, 1994.
- [4] Ulrik Thyge Pedersen. "Online Retail Dataset". *Kaggle Repository*. <https://www.kaggle.com/datasets/ulrikthygepedersen/online-retail-dataset>
- [5] Microsoft. "Power BI Documentation - Data Visualization". <https://learn.microsoft.com/en-us/power-bi/>