

# Nhóm 3

Chủ đề: Xây  
dựng hệ thống  
gợi ý sản phẩm



# Thành viên

1. Nguyễn Bình An
2. Đinh Ngọc Thảo
3. Đỗ Quang Phước
4. Vũ Phạm Thành Phương
5. Nguyễn Mạnh Cường





# Mục tiêu

Cá nhân hóa trải nghiệm người dùng

Tăng doanh thu và giá trị đơn hàng

Tiết kiệm thời gian tìm kiếm sản phẩm của khách hàng

Khám phá các sản phẩm mới

Phân tích hành vi và phân khúc khách hàng

Tự động hóa và tối ưu hóa việc bán hàng



# Nội dung

## Khám phá dữ liệu

- Top sản phẩm bán chạy
- Tần suất và giá trị mua hàng trung bình

## Xây dựng hệ thống gợi ý

- User-Based Filtering : Dựa trên khách hàng
- Item-Based Filtering : Dựa trên sản phẩm

## Trực quan hóa

- Gợi ý top 5 sản phẩm cho khách hàng
- Vẽ biểu đồ mạng các sản phẩm đi kèm



# Khám phá dữ liệu

## Khai báo thư viện

- Import **pandas** as **pd**
- Import **numpy** as **np**
- Import **matplotlib.pyplot** as **plt**
- Import **seaborn** as **sns**

## Đọc dữ liệu

- `df=pd.read_csv("Online Retail.csv")`
- `df.head(10)`
- `df.info()`
- `df.describe()`



# Bảng thống kê mô tả

	Quantity	UnitPrice	CustomerID
count	541909.000000	541909.000000	406829.000000
mean	9.552250	4.611114	15287.690570
std	218.081158	96.759853	1713.600303
min	-80995.000000	-11062.060000	12346.000000
25%	1.000000	1.250000	13953.000000
50%	3.000000	2.080000	15152.000000
75%	10.000000	4.130000	16791.000000
max	80995.000000	38970.000000	18287.000000

Thống kê mô tả cho ba cột dữ liệu: "Quantity" (Số lượng), "UnitPrice" (Đơn giá), và "CustomerID" (Mã khách hàng).

# Xử lý dữ liệu

- Nhận thấy có những đơn hàng mà số sản phẩm là âm (đây là những đơn hàng đã bị cancel, InvoiceNo có chữ "c")

```
df[df['Quantity']<0].head()
```

- Loại bỏ những đơn hàng đã bị hủy

```
df = df[df['Quantity']>0]
```


```
print(df[df['UnitPrice']<=0].head())
```

```
df = df[df['UnitPrice']>0]
```




## Chuyển cột "InvoiceDate" sang dạng date-time

```
df["InvoiceDate"] =  
pd.to_datetime(df["InvoiceDate"],dayfirst=  
True)  
print(df["InvoiceDate"].min(),  
df["InvoiceDate"].max())
```



## Tạo cột tháng - Lọc data từ tháng 3 đến tháng 7

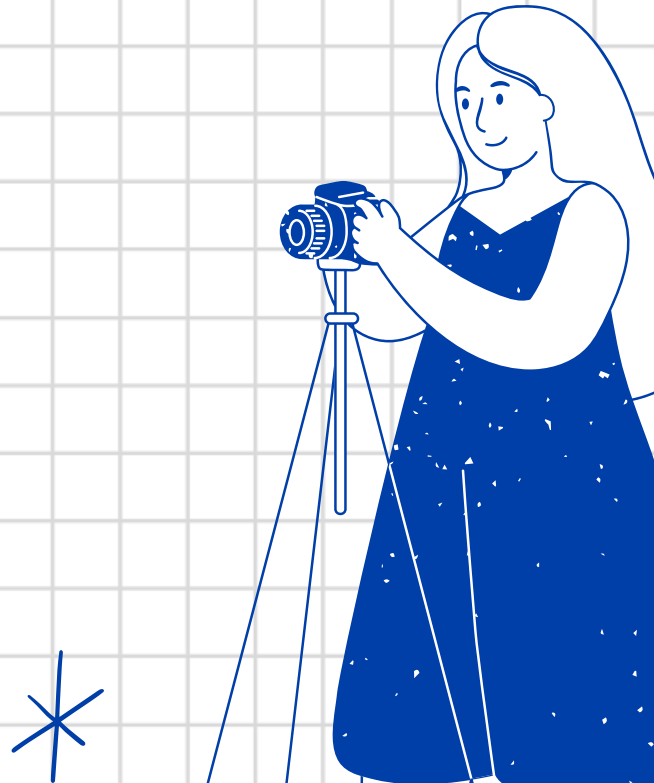
```
df["Month"] =  
df["InvoiceDate"].dt.month  
  
filtered_df = df[(df["Month"] >= 3) &  
(df["Month"] <= 7)]  
print(filtered_df)
```





# Kết quả dữ liệu sau khi xử lí

...	InvoiceNo	StockCode	Description	Quantity	\
105335	545220	21955	DOORMAT UNION JACK GUNS AND ROSES	2	
105336	545220	48194	DOORMAT HEARTS	2	
105337	545220	22556	PLASTERS IN TIN CIRCUS PARADE	12	
105338	545220	22139	RETROSPOT TEA SET CERAMIC 11 PC	3	
105339	545220	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	4	
...	...	...	...	...	
285416	561903	21900	KEY FOB , SHED	24	
285417	561903	48187	DOORMAT NEW ENGLAND	2	
285418	561903	85152	HAND OVER THE CHOCOLATE SIGN	12	
285419	561903	82600	NO SINGING METAL SIGN	12	
285420	561903	21175	GIN + TONIC DIET METAL SIGN	12	
	InvoiceDate	UnitPrice	CustomerID	Country	Month
105335	2011-03-01 08:30:00	7.95	14620.0	United Kingdom	3
105336	2011-03-01 08:30:00	7.95	14620.0	United Kingdom	3
105337	2011-03-01 08:30:00	1.65	14620.0	United Kingdom	3
105338	2011-03-01 08:30:00	4.95	14620.0	United Kingdom	3
105339	2011-03-01 08:30:00	3.75	14620.0	United Kingdom	3
...	...	...	...	...	...
285416	2011-07-31 16:04:00	0.65	17162.0	United Kingdom	7
285417	2011-07-31 16:04:00	7.95	17162.0	United Kingdom	7
285418	2011-07-31 16:04:00	2.10	17162.0	United Kingdom	7
285419	2011-07-31 16:04:00	2.10	17162.0	United Kingdom	7
285420	2011-07-31 16:04:00	2.55	17162.0	United Kingdom	7
[175685 rows x 9 columns]					



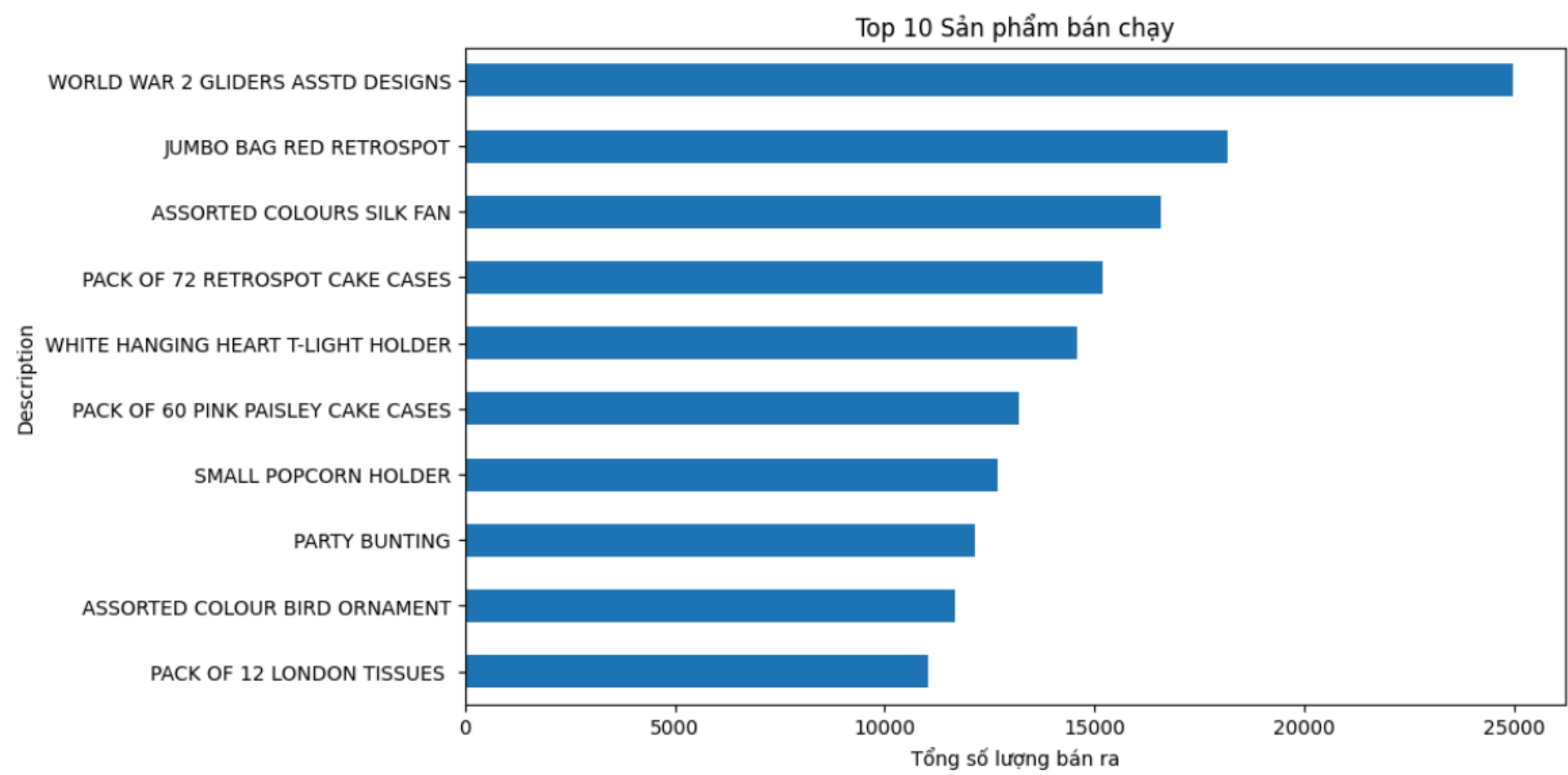
# Top 10 sản phẩm bán chạy nhất

```
top_products =  
filtered_df.groupby('Description')  
['Quantity'].sum().sort_values(ascending =  
False).head(10)  
print(top_products)
```

```
Description  
WORLD WAR 2 GLIDERS ASSTD DESIGNS      24960  
JUMBO BAG RED RETROSPOT                 18167  
ASSORTED COLOURS SILK FAN               16586  
PACK OF 72 RETROSPOT CAKE CASES         15204  
WHITE HANGING HEART T-LIGHT HOLDER      14584  
PACK OF 60 PINK PAISLEY CAKE CASES      13207  
SMALL POPCORN HOLDER                    12684  
PARTY BUNTING                          12143  
ASSORTED COLOUR BIRD ORNAMENT           11691  
PACK OF 12 LONDON TISSUES               11049  
Name: Quantity, dtype: int64
```

# Xây dựng biểu đồ về top 10 sản phẩm bán chạy nhất

```
top_products.head(10).plot(kind='barh', figsize=(10,6), title='Top 10 Sản phẩm bán chạy')  
plt.xlabel('Tổng số lượng bán ra')  
plt.gca().invert_yaxis()  
plt.show()
```



# Tần suất và giá trị mua trung bình

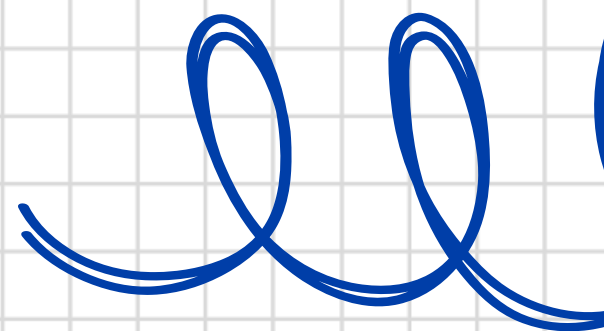
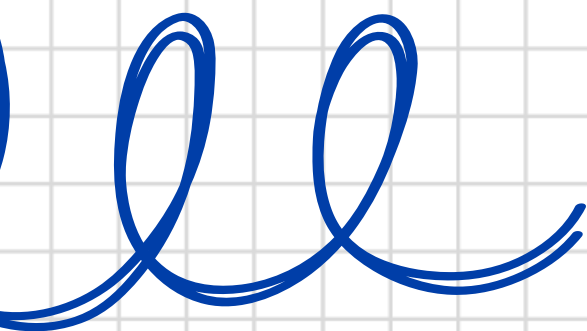


```
customer_stats = filtered_df.groupby('CustomerID').agg({
    'InvoiceNo': 'count',
    'Quantity': 'sum',
    'UnitPrice': 'mean'
}).rename(columns={
    'InvoiceNo': 'Frequency',
    'Quantity': 'TotalQuantity',
    'UnitPrice': 'AveragePrice'
})
# Hiển thị kết quả
print(customer_stats.head())
```

```
...
CustomerID  Frequency  TotalQuantity  AveragePrice
12347.0      42         679         2.759762
12348.0       5         269         8.920000
12352.0      23         156        42.379565
12353.0       4          20         6.075000
12354.0      58         530         4.503793
```



# Hệ thống gợi ý khách hàng



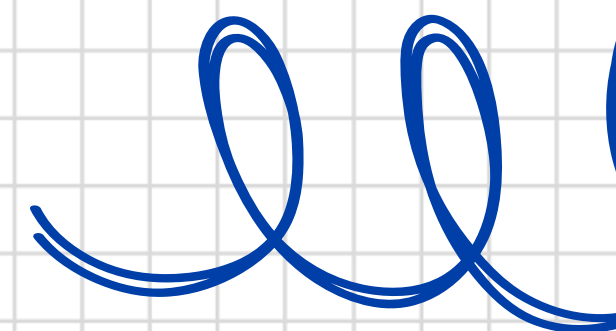
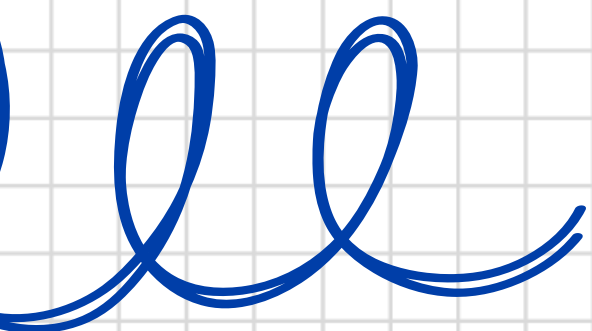


# Tạo ma trận tương quan giữa người dùng và sản phẩm

```
#Xây dựng hệ thống gợi ý sản phẩm
from sklearn.metrics.pairwise import cosine_similarity
from sklearn.preprocessing import MinMaxScaler
import pandas as pd
#Tạo ma trận tương quan giữa người dùng và sản phẩm
user_product_matrix = filtered_df.groupby(['CustomerID', 'Description'])['Quantity'].sum().unstack().fillna(0)
user_product_matrix = user_product_matrix.astype(int)
user_product_matrix.to_csv("user_product_matrix", index=True)
```



# Cách 1: Hệ thống gợi ý dựa trên khách hàng tương tự



# Tạo hệ thống gợi ý

```
# Ma trận tương đồng giữa các khách hàng
user_similarity = cosine_similarity(user_product_matrix)
user_similarity_df = pd.DataFrame(user_similarity, index=user_product_matrix.index, columns=user_product_matrix.index)
#Hệ thống gợi ý dựa trên khách hàng tương tự
def recommend_products(customer_id, user_similarity_df, user_product_matrix, top_n=5):
    # Lấy danh sách người dùng tương tự
    similar_users = user_similarity_df[customer_id].sort_values(ascending=False).index[1:] # Bỏ chính người dùng đó
    similar_users_weights = user_similarity_df[customer_id].sort_values(ascending=False).iloc[1:]

    # Tính điểm gợi ý dựa trên người dùng tương tự
    recommendations = user_product_matrix.loc[similar_users].T.dot(similar_users_weights).sort_values(ascending=False)

    # Loại bỏ các sản phẩm mà người dùng đã mua
    already_purchased = user_product_matrix.loc[customer_id]
    recommendations = recommendations[already_purchased == 0]

    return recommendations.head(top_n)
# 4. Gợi ý sản phẩm cho một khách hàng cụ thể (ví dụ: CustomerID = 12347.0)
customer_id = 12347.0
recommended_products = recommend_products(customer_id, user_similarity_df, user_product_matrix)
print(f"Gợi ý sản phẩm cho khách hàng {customer_id}:")
print(recommended_products)
```

# Kết quả gợi ý cho khách hàng cụ thể:

Gợi ý sản phẩm cho khách hàng 12347.0:

Description

WORLD WAR 2 GLIDERS ASSTD DESIGNS	533.719852
-----------------------------------	------------

DOUGHNUT LIP GLOSS	417.449766
--------------------	------------

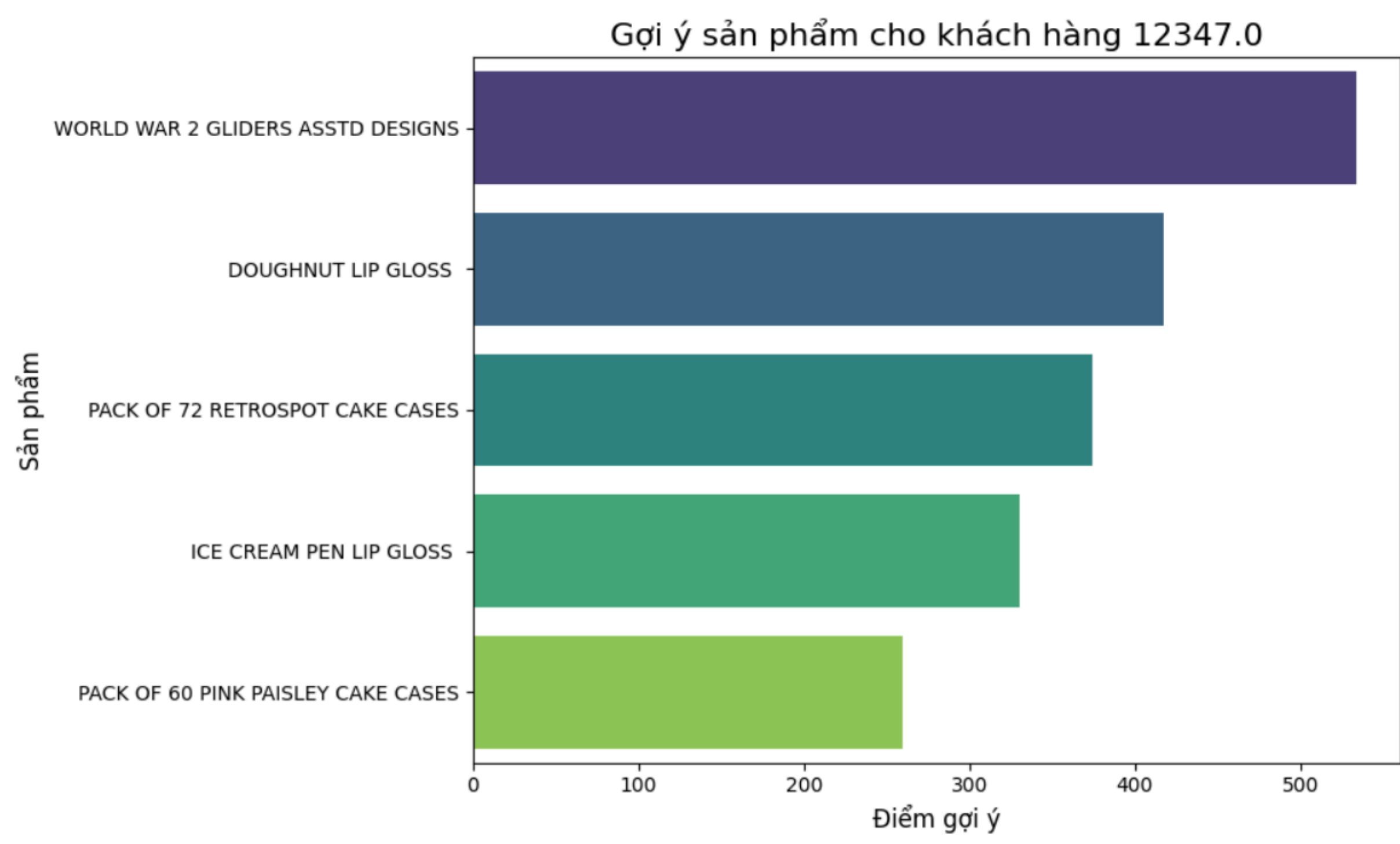
PACK OF 72 RETROSPOT CAKE CASES	374.714204
---------------------------------	------------

ICE CREAM PEN LIP GLOSS	329.965421
-------------------------	------------

PACK OF 60 PINK PAISLEY CAKE CASES	259.943669
------------------------------------	------------

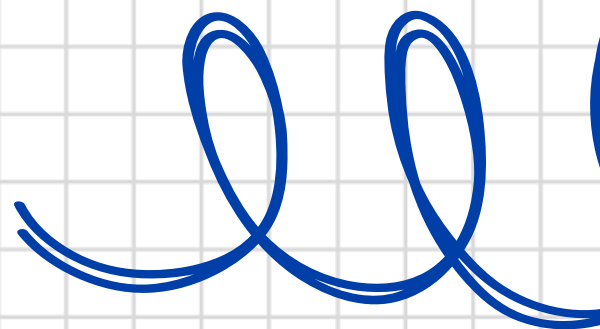
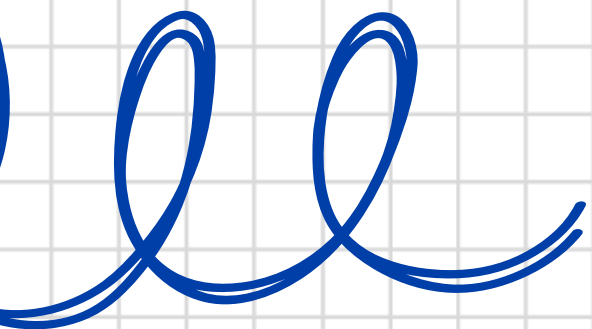
dtype: float64

# Trực quan hóa gợi ý 5 sản phẩm cho khách hàng cụ thể





# Cách 2: Hệ thống gợi ý dựa trên sản phẩm tương tự





# Tạo hệ thống gợi ý

```
#Hệ thống gợi ý dựa trên sản phẩm tương tự
#Tính độ tương đồng giữa các sản phẩm
item_similarity = cosine_similarity(user_product_matrix.T)
item_similarity_df = pd.DataFrame(item_similarity, index=user_product_matrix.columns, columns=user_product_matrix.columns)
#Gợi ý sản phẩm cho một sản phẩm cụ thể
def recommend_similar_items(product_name, item_similarity_df, top_n=5):
    # Lấy danh sách sản phẩm tương tự
    similar_items = item_similarity_df[product_name].sort_values(ascending=False).iloc[1:top_n+1]
    return similar_items

# 4. Gợi ý sản phẩm tương tự cho một sản phẩm cụ thể (ví dụ: "WHITE HANGING HEART T-LIGHT HOLDER")
product_name = "WHITE HANGING HEART T-LIGHT HOLDER"
recommended_items = recommend_similar_items(product_name, item_similarity_df)
print(f"Gợi ý sản phẩm tương tự cho '{product_name}':")
print(recommended_items)
```

# Kết quả gợi ý cho sản phẩm cụ thể:

Gợi ý sản phẩm tương tự cho 'WHITE HANGING HEART T-LIGHT HOLDER':

Description

GIN + TONIC DIET METAL SIGN	0.773976
-----------------------------	----------

FAIRY CAKE FLANNEL ASSORTED COLOUR	0.728101
------------------------------------	----------

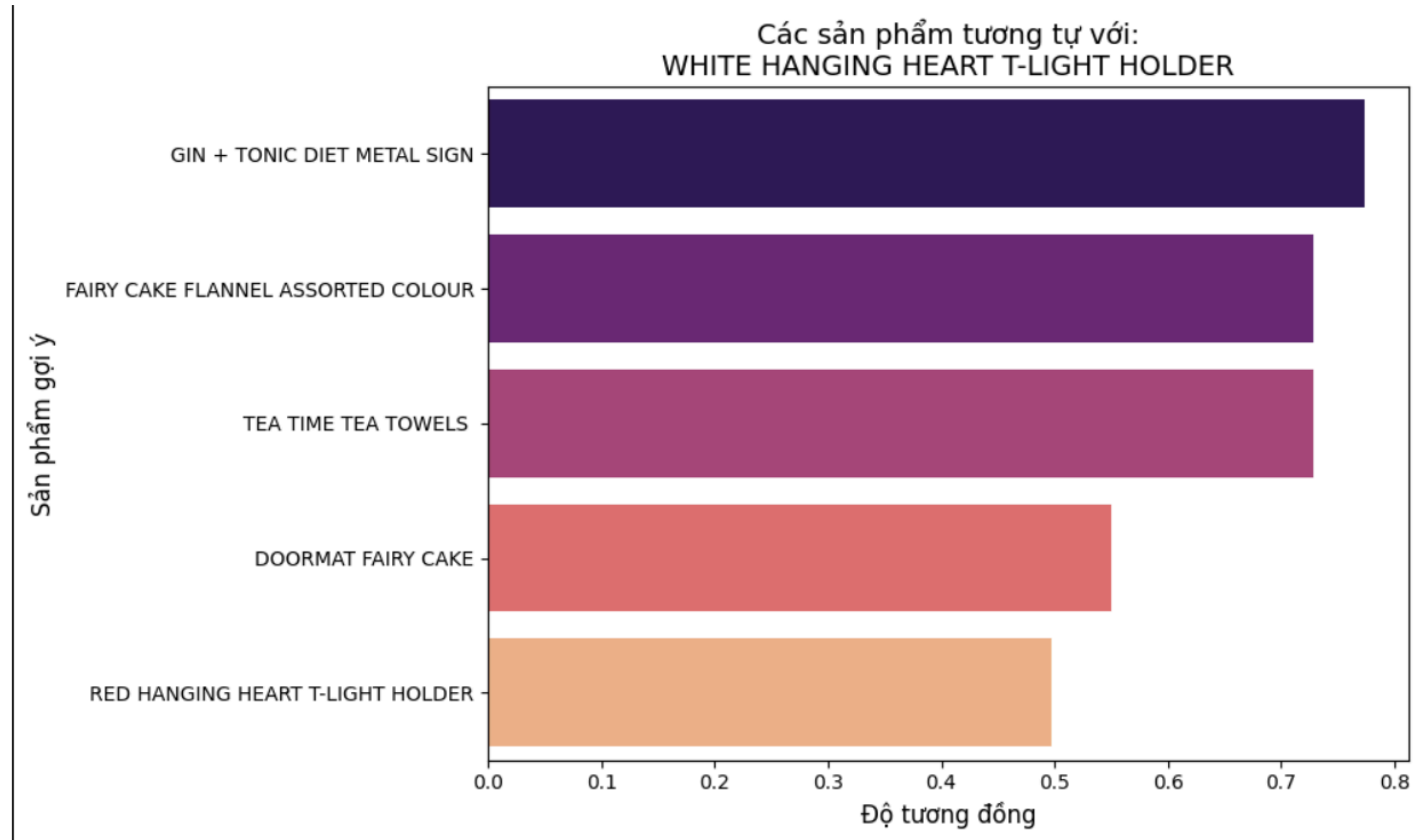
TEA TIME TEA TOWELS	0.728031
---------------------	----------

DOORMAT FAIRY CAKE	0.550677
--------------------	----------

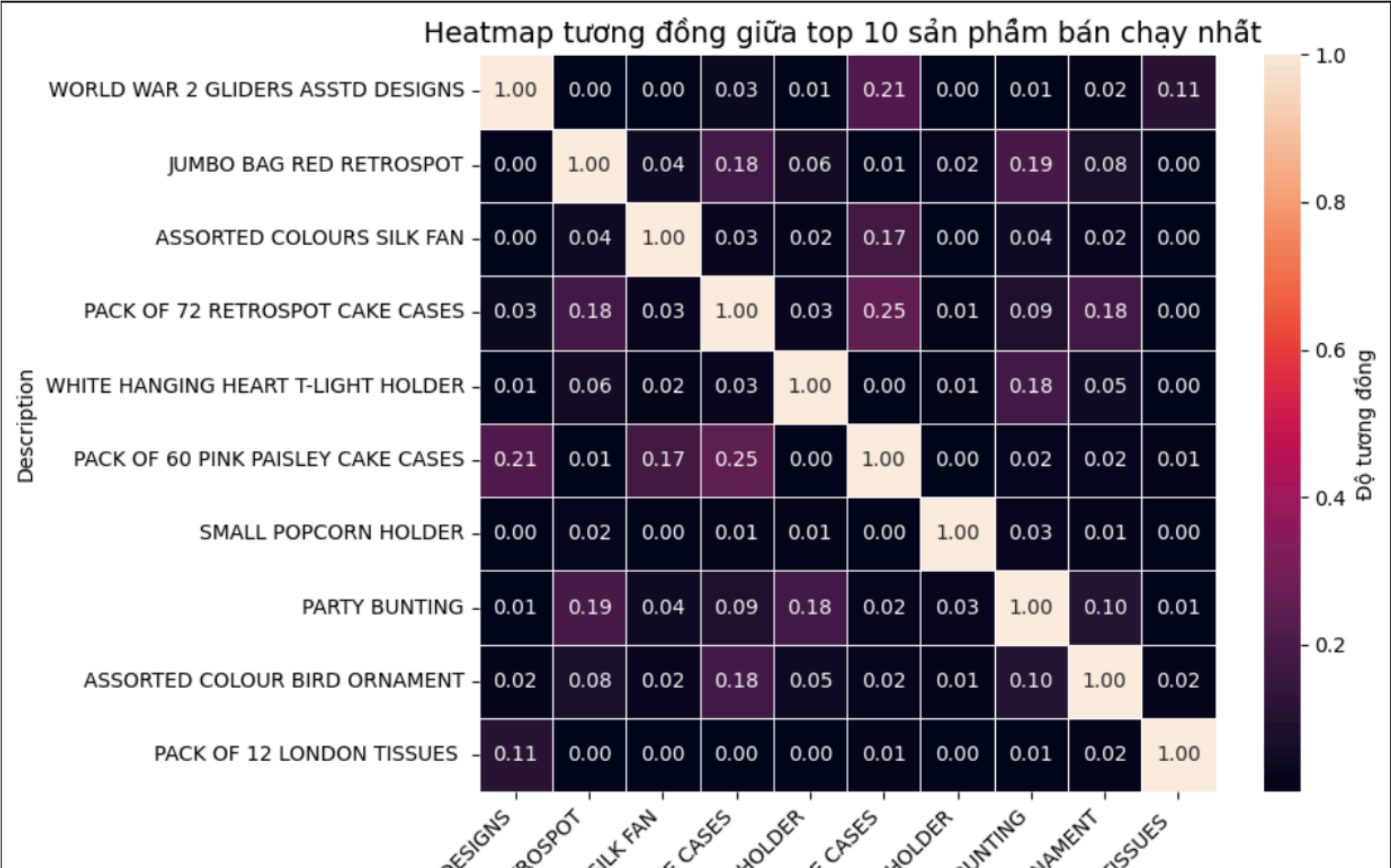
RED HANGING HEART T-LIGHT HOLDER	0.497785
----------------------------------	----------

Name: WHITE HANGING HEART T-LIGHT HOLDER, dtype: float64

# Trực quan hóa gợi ý 5 sản phẩm cho sản phẩm cụ thể

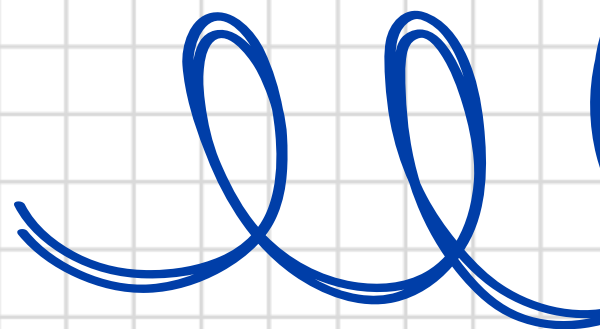
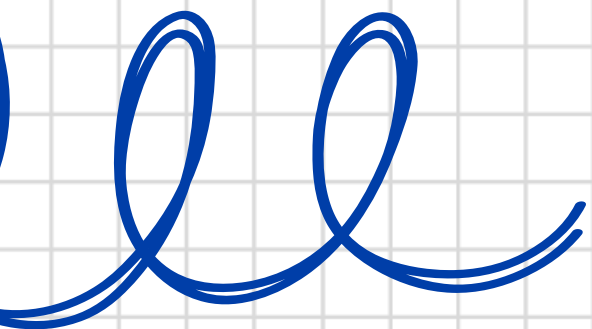


# Độ tương đồng giữa 10 sản phẩm bán chạy nhất





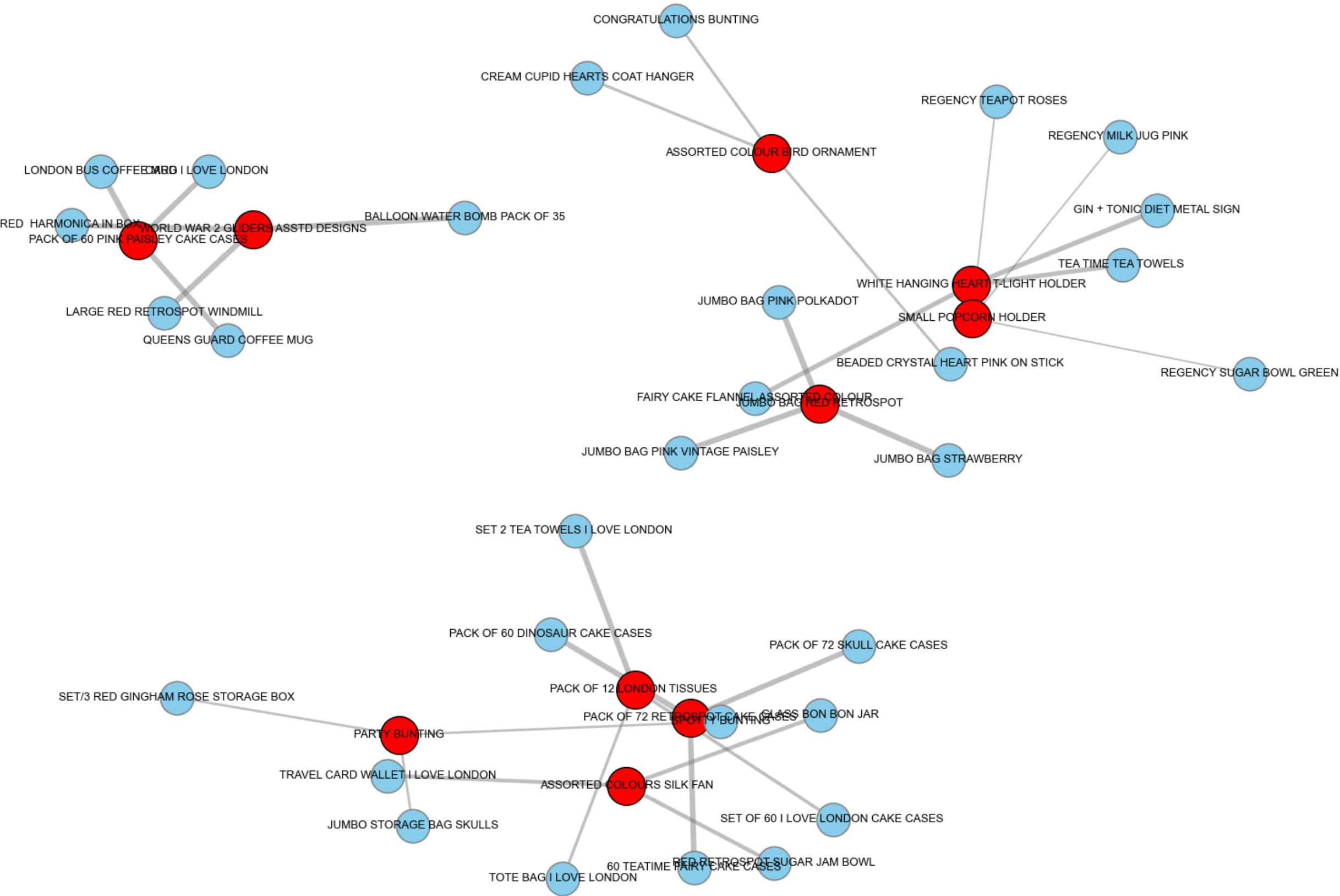
# Biểu đồ mạng các sản phẩm thường đi kèm



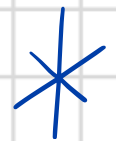
Biểu đồ mạng: Top 10 sản phẩm bán chạy & gợi ý sản phẩm liên quan

Top sản phẩm

Sản phẩm liên quan







**Cảm ơn thầy và các bạn đã lắng  
nghe phần trình bày của nhóm  
chúng em**

