CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

Độc lập – Tự do – Hạnh phúc

Hà Nội, ngày 22 tháng 12 năm 2021

BÁO CÁO KẾT QUẢ VÀ ĐỀ XUẤT DOANH NGHIỆP

Kính gửi: Ban giám đốc Công ty A

Tôi là: Phí Quang Anh - Nhân viên Phân tích dữ liệu phòng Marketing của Công ty A

Trong năm vừa qua, tôi nhận thấy doanh nghiệp đã thực hiện tốt về mặt bán hàng, cơ bản đã đạt được doanh thu đề ra. Tuy vậy, các hoạt động Marketing lại chưa thực sự hiệu quả, đặc biệt là trong quá trình tiếp cận và xúc tiến khách hàng. Công ty chưa tối ưu được các gói, combo sản phẩm, thời điểm quảng cáo để thúc đẩy khách hàng mua, dẫn đến nhiều loại sản phẩm còn tồn kho. Qua bản báo cáo này, tôi sẽ cố gắng làm rõ một số điểm.

Trong quá trình phân tích, tôi sẽ đưa ra được kết quả về doanh thu của các tháng trong năm, doanh thu theo khu vực, doanh thu về các sản phẩm của công ty. Cùng với đó là phân tích khả năng những sản phẩm được khách hàng mua cùng nhau, qua đó giúp cho việc Marketing của công ty được tốt hơn.

Với mong muốn tăng doanh thu và lợi nhuận của doanh nghiệp, đồng thời tối ưu và thực hiện hiệu quả các hoạt động Marketing. Tôi có một số đề xuất dưới bản báo cáo sau đây, mong ban lãnh đạo xem xét và áp dụng vào kế hoạch kinh doanh trong năm 2022 sắp tới.

Tôi cam kết kết quả phân tích trên dựa trên dữ liệu của công ty, có độ chính xác cao và có thể tin cậy.

Xin chân thành cảm ơn.

Ký tên

Phí Quang Anh

MỤC LỤC

I. Quy trình báo cáo dữ liệu

Trong bài báo cáo dưới đây, kết quả dữ liệu và đề xuất sẽ được trình bày theo quy trình như sau:

1. Đặt ra các câu hỏi trước khi xử lý

2. Phân tích doanh thu và sản lượng

- Phân tích doanh thu của 12 tháng trong năm 2021

- Phân tích doanh thu của các thành phố trong năm 2021

- Phân tích doanh thu và sản lượng theo từng khung giờ trong năm 2021

- Phân tích lượng sản phẩm bán ra theo từng loại sản phẩm trong năm 2021

3. Khai phá luật kết hợp với thuật toán Apriori (Association rule)

- Phân tích mối quan hệ giữa các sản phẩm được mua cùng nhau

4. Kết luận

5. Đề xuất, kiến nghị

6. Hạn chế

7. Phụ lục

II. Báo cáo dữ liệu của doanh nghiệp

1. Đặt câu hỏi về bộ dữ liệu trước khi xử lý

- Doanh thu của các tháng trong năm là bao nhiêu?

- Doanh thu tại các thành phố trong năm là bao nhiêu?

- Doanh thu và sản lượng tại thời điểm nào trong ngày là cao nhất (thấp nhất) ? Tại sao?

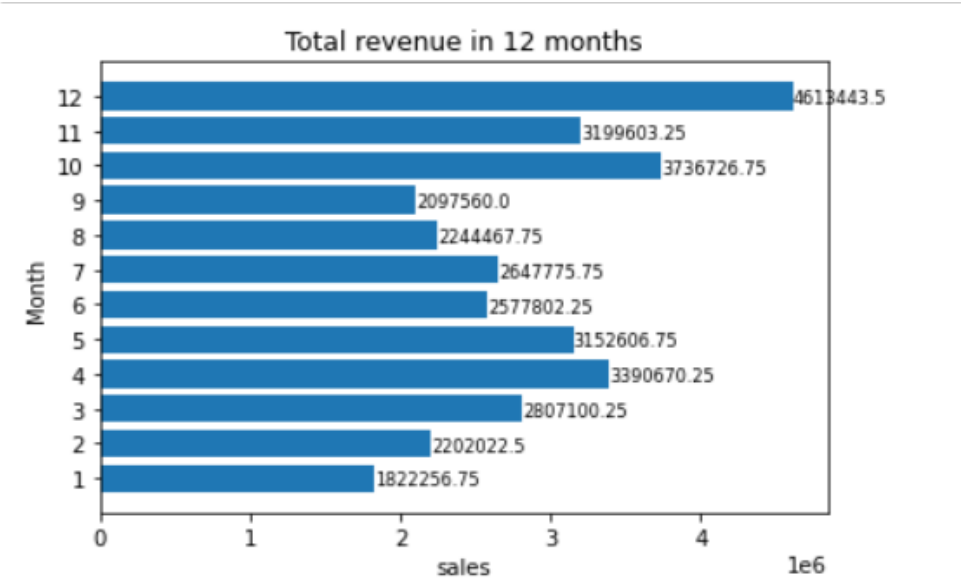
- Các loại sản phẩm nào đang bán chạy nhất (bán chậm nhất)? Doanh thu, sản lượng bán của các sản phẩm đó ra sao?

- Có phân loại nhóm khách hàng được không?

- Các sản phẩm nào được khách hàng thường xuyên mua cùng nhau? Cần phải làm gì để bán được nhiều sản phẩm hơn?

2. Phân tích doanh thu và sản lượng trong năm 2021

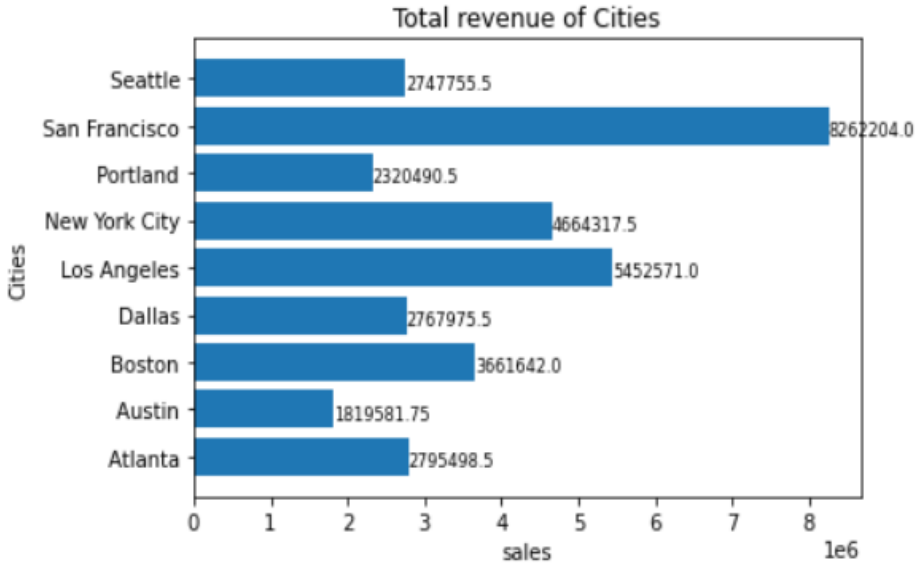
2.1. Phân tích doanh thu của 12 tháng trong năm 2021



Bảng 2.1. Doanh thu của 12 tháng trong năm 2021

Tổng quan, trong 1 năm qua, ta thấy rằng doanh thu của các tháng không đều nhau. Trong khi tháng 12 là tháng có doanh thu cao nhất thì tháng 1 là tháng có doanh thu thấp nhất năm. Trong vòng 4 tháng đầu năm, ta thấy được doanh thu đang tăng dần đều, từ 1,822,256.75$ vào tháng 1, và đạt 3,390,670.25$ vào tháng 4. Nhưng sau đó từ tháng 5 cho tới tháng 9, doanh thu hầu như giảm dần đều, duy tại tháng 7 là tăng nhẹ lên mức 2,647,775.75$. Cuối cùng, trong 3 tháng cuối năm cho thấy sự tăng mạnh về doanh thu, trong đó có tháng 10 và 12 là hai tháng có doanh thu cao nhất năm, lần lượt là 3,736,726.75$ và 4,613,443.5$. Tuy vậy có sự giảm về doanh thu từ tháng 10 đến tháng 11, đạt hơn 3 triệu $ vào tháng 11.

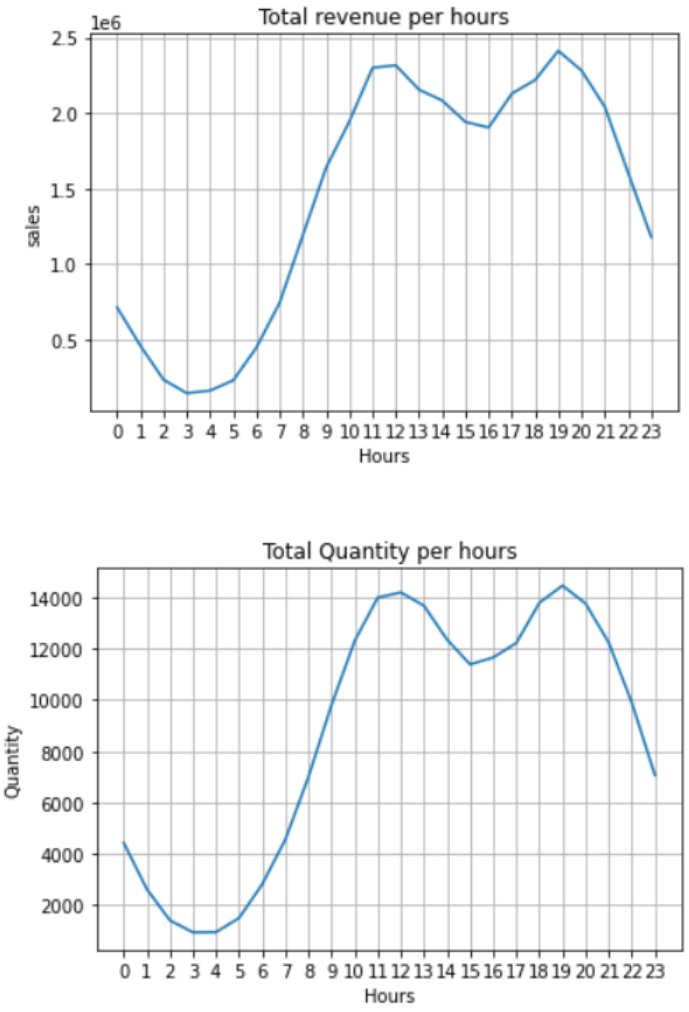
2.2. Phân tích doanh thu của các thành phố trong năm 2021



Bảng 2.2. Doanh thu của các thành phố trong năm 2021

Nhìn chung, ta thấy được San Francisco là thành phố có doanh thu cao nhất, đạt 8,262,204$, sau đó là Los Angeles với 5,452,571$, và New York City đạt 4,664,317.5$. Ngoài ra, Boston là một nơi có doanh số trung bình, vào khoảng gần 3,7 triệu $. Trong khi đó, có 4 thành phố bao gồm Seattle, Portland, Dallas, và Atlanta đều có doanh thu không quá cao, trong khoảng từ 2 đến 3 triệu $. Và cuối cùng là Austin có doanh thu thấp nhất, dưới 2 triệu $.

2.3. Phân tích doanh thu và sản lượng theo từng khung giờ trong năm 2021



Bảng 2.3. Tổng doanh thu và tổng sản lượng tại các thời điểm trong ngày

Về kết quả doanh thu và sản lượng bán ra theo giờ cho ta thấy được hai biểu đồ trên không có nhiều sự khác biệt. Doanh thu và sản lượng cao nhất tại 2 thời điểm, vào lúc 12h và 19h. Trong khoảng thời gian từ 0h cho đến 3h sáng, doanh thu giảm từ 500 nghìn $ xuống gần 200 nghìn $, còn sản lượng giảm từ hơn 4000 xuống còn 1000. Tuy nhiên, trong khoảng 3h đến 12h, doanh thu và sản lượng đồng loạt tăng mạnh lên mức gần 2,3 triệu $ và 14000 sản phẩm. Sau đó giảm tới thời điểm lúc 15h, và tăng lại tới điểm cao nhất vào lúc 19h, khoảng gần 2,5 triệu $ và hơn 14000 sản phẩm. Cuối cùng là giảm dần đều cả về doanh thu lẫn sản lượng vào thời điểm sau 19h.

2.4. Phân tích lượng sản phẩm bán ra và doanh thu theo từng loại sản phẩm trong năm 2021

Bảng 2.4. Tổng doanh thu và sản lượng của các loại sản phẩm

Tổng quan biểu đồ cho ta thấy được mối quan hệ giữa 3 biến: Tổng sản lượng, doanh thu và sản phẩm. Trong tất cả các sản phẩm, ta thấy được 2 loại pin AA và AAA là 2 sản phẩm bán chạy nhất của công ty, tuy vậy lại có doanh thu thấp nhất. Trong khi đó, Macbook Pro và Thinkpad tuy có mức sản lượng chỉ khoảng 5000 chiếc mỗi loại, nhưng lại đóng góp doanh thu cao nhất, lần lượt là 8 triệu $ và 5 triệu $. Bên cạnh đó, sản phẩm LG Dryer, LG Washing Machine, và Vareebadd Phone vừa có doanh thu thấp, và vừa có sản lượng thấp. Ngoài ra, Google Phone và iPhone tuy sản lượng không quá cao (hơn 5000 sản phẩm) cũng đóng góp lớn vào doanh thu, khi mà doanh thu bán được chỉ sau Macbook và Thinkpad. Còn lại các sản phẩm khác tuy có sản lượng khác nhau, nhưng doanh thu thì không quá khác biệt, chỉ trên dưới 2 triệu $.

3. Khai phá luật kết hợp với thuật toán Apriori (Association rule)

3.1. Phân tích các sản phẩm hay được mua cùng nhau

	index	All Products
0	iPhone, Lightning Charging Cable	882
1	Google Phone, USB-C Charging Cable	856
2	iPhone, Wired Headphones	361
3	Vareebadd Phone, USB-C Charging Cable	312
4	Google Phone, Wired Headphones	303
5	iPhone, Apple Airpods Headphones	286
6	Google Phone, Bose SoundSport Headphones	161
7	Vareebadd Phone, Wired Headphones	104
8	Google Phone, USB-C Charging Cable, Wired Head...	77
9	Vareebadd Phone, Bose SoundSport Headphones	60

Bảng 3.1. Các cặp sản phẩm thường được mua cùng nhau

Bảng 3.1 phía trên mang đến thông tin về các cặp, nhóm sản phẩm hay được mua cùng và tần suất của chúng. Đầu tiên, cặp sản phẩm iPhone, sạc Lightning và cặp Google phone, sạc type C hay được mua kèm với nhau nhất, với tổng lần xuất hiện lần lượt 882 và 856 lần. Xuất hiện ít nhất trong top 10 là sản phẩm điện thoại Vareebadd mua kèm cùng tai nghe Bose với 60 lần. Bên cạnh đó, các cặp sản phẩm top 3 đến top 6, được mua cùng nhau với tần suất tên dưới 300 lần. Cuối cùng là cặp sản phẩm từ top 7 đến top 9 được khách hàng lựa chọn trong cùng 1 lần mua lần lượt là 161,104, và 77 lần. Tuy vậy, điều này chưa khẳng định được các cặp sản phẩm này là được mua ngẫu nhiên, hay là các cặp mà khách hàng chọn mua thường xuyên. Do vậy cùng đi đến kiểm định Apriori để xác định điều này.

3.2. Kiểm định mối quan hệ giữa các sản phẩm được mua cùng nhau bằng thuật toán Apriori

	20in Monitor	27in 4K Gaming Monitor	27in FHD Monitor	34in Ultrawide Monitor	AA Batteries (4-pack)	AAA Batteries (4-pack)	Apple Airpods Headphones	Bose SoundSport Headphones	Flatscreen TV	Google Phone	LG Dryer	LG Washing Machine	Lightning Charging Cable	Macbook Pro Laptop	ThinkPad Laptop
0	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	True	False	True	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	True	False	False	False	False	False
4	False	False	False	False	False	False	False	True	False	True	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
7131	False	False	False	False	False	False	True	False	False	False	False	False	False	False	False
7132	False	False	False	False	False	False	False	False	False	False	False	False	True	False	False
7133	False	False	False	True	True	False	False	False	False	False	False	False	False	False	False
7134	False	False	False	False	False	True	False	False	False	False	False	False	False	False	False
7135	False	False	False	False	False	False	False	False	False	True	False	False	False	False	False

7136 rows × 19 columns

Bảng 3.2.a. Mã hóa các sản phẩm xuất hiện trong lần mua cùng nhau của khách hàng

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
15	(Vareebadd Phone)	(USB-C Charging Cable)	0.084221	0.289098	0.051570	0.612313	2.118015	0.027221	1.833701
8	(Google Phone)	(USB-C Charging Cable)	0.229260	0.289098	0.139714	0.609413	2.107985	0.073436	1.820088
12	(Lightning Charging Cable)	(iPhone)	0.248459	0.261351	0.141676	0.570220	2.181818	0.076741	1.718668
13	(iPhone)	(Lightning Charging Cable)	0.261351	0.248459	0.141676	0.542091	2.181818	0.076741	1.641247
9	(USB-C Charging Cable)	(Google Phone)	0.289098	0.229260	0.139714	0.483277	2.107985	0.073436	1.491591
20	(USB-C Charging Cable, Wired Headphones)	(Google Phone)	0.028447	0.229260	0.012192	0.428571	1.869368	0.005670	1.348795
2	(Apple Airpods Headphones)	(iPhone)	0.133128	0.261351	0.052270	0.392632	1.502316	0.017477	1.216147
4	(Bose SoundSport Headphones)	(Google Phone)	0.111127	0.229260	0.031951	0.287516	1.254103	0.006474	1.081764
19	(Wired Headphones)	(iPhone)	0.229680	0.261351	0.064742	0.281879	1.078547	0.004715	1.028586
10	(Google Phone)	(Wired Headphones)	0.229260	0.229680	0.059137	0.257946	1.123065	0.006480	1.038091

Bảng 3.2.b. Kết quả của thuật toán Apriori

Qua việc phân tích, kết quả tại bảng 2.3 cho ta thấy được mối liên hệ giữa các sản phẩm được mua cùng nhau. Đầu tiên, các cặp sản phẩm như (Điện thoại Varrebadd; sạc type-C), (Điện thoại Google; sạc type-C), (Sạc Lightning; iPhone) đều có hệ số tin cậy(confidence) khá cao, lần lượt là 61%,60% và 54%, kết hợp hệ số lift đều lớn hơn 2, điều này càng chứng tỏ rằng khi khách hàng mua các cặp sản phẩm này khả năng rất cao là không hề ngẫu nhiên. Hơn nữa tỷ lệ xuất hiện trong các cuộc giao dịch thực tế cũng cao (antecedent support), khi các cặp này xuất hiện tới hơn 20% trong các cuộc giao dịch của 5 cặp đầu tiên, ngoại trừ cặp đầu tiên với 8%. Ví dụ: Khi 1 khách hàng mua điện thoại Varrebad, có tới 61% là họ sẽ mua kèm sạc type-C, tương tự với

các cặp sản phẩm bên dưới. Ngoài ra còn 1 số cặp sản phẩm được mua kèm phổ biến khác như (tai nghe Airpod; iPhone), (Điện thoại Google; sạc type-C; tai nghe không dây)...

4. Kết luận

Từ kết quả phân tích phía trên, có thể thấy những tháng cuối năm là khách hàng mua nhiều nhất, điều này có thể giải thích bởi đây là thời điểm diễn ra nhiều sự kiện quan trọng với người dân tại Mỹ. Có thể kể đến như Black Friday, Cyber Monday, đặc biệt là Giáng sinh, khách hàng họ sẽ thường mua quà vào dịp này để tặng người thân, bạn bè hoặc cho ngay bản thân mình.. Tại thời điểm này, việc áp dụng quảng cáo và các chiến dịch xúc tiến là hoàn toàn phù hợp.

Ngoài ra, với việc biết được khách hàng sẽ thường mua vào thời điểm nào trong ngày, việc sử dụng quảng cáo, các chương trình, sự kiện sẽ được tổ chức, cải thiện đáng kể so với trước, giúp tiết kiệm và tối ưu được các chi phí đó.

Kết quả cũng chỉ ra được những sản phẩm nào đang mang tới doanh thu chính, đồng thời cho thấy sản phẩm bán ra của từng loại. Từ đó cho biết được sản phẩm nào cần được thúc đẩy bán hơn nữa, ví dụ như áp dụng giảm giá, bán chéo, hoặc ưu đãi tặng kèm khi mua các sản phẩm khác…

Quan trọng nhất, việc nhận định được các sản phẩm có khả năng cao được mua cùng nhau, giúp ta có thể triển khai, thiết kế các combo sản phẩm, thêm nữa còn có thể thúc đẩy bán chéo với các sản phẩm còn đang tồn kho, giúp tăng doanh thu và lợi nhuận của doanh nghiệp.

5. Đề xuất, kiến nghị

Tôi muốn đề xuất rằng doanh nghiệp cần thiết kế các gói combo sản phẩm như sau:

1. Điện thoại Vareebadd + Cáp sạc type C

2. Điện thoại Google + Cáp sạc type C

3. Điện thoại iPhone + Cáp sạc Lightning

4. Điện thoại iPhone + Airpod (hoặc Wired Headphones)

5. Điện thoại Google + Wired Headphone

Với việc bán các sản phẩm theo gói combo như trên, doanh nghiệp sẽ thúc đẩy việc bán các sản phẩm nhanh chóng hơn, thêm với việc có ưu đãi giá hợp lý khi mua combo, điều này giúp khách hàng dễ dàng thực hiện quyết định mua hàng của mình hơn, họ sẽ thấy có lợi hơn là khi mua lẻ từng sản phẩm. Hơn nữa, chúng ta cũng cần đưa ra thêm các dịch vụ tặng kèm như sửa chữa tại nhà, bảo hành 2-3 năm khi khách hàng mua Máy giặt LG và Máy rửa bát LG, nhằm xúc tiến bán hai mặt hàng này.

Ngoài ra, tôi đề xuất thêm trong năm tới, doanh nghiệp nên đánh giá, và giao cho phòng ban Marketing lên kế hoạch tổ chức hai chiến dịch Marketing ngắn hạn, chiến dịch thứ nhất nên diễn ra đầu năm, trong thời điểm từ tháng 1 tới tháng 3. Chiến dịch thứ hai diễn ra trong khoảng từ tháng 6 tới tháng 9. Bởi trong 2 giai đoạn này có thể thấy doanh thu tháng thấp hơn so với các tháng khác trong năm. Đồng thời, các hoạt động sự kiện nên được diễn ra vào thời điểm trước từ 1-2 tiếng với khung giờ 12h và 19h, do khách hàng thông thường sẽ chọn 2 khung giờ chính để mua hàng, bao gồm từ 10h-14h và từ 17h-21h. Trong đó khung giờ 12h và 19h là thời điểm nhiều khách hàng nhất - phản ảnh qua sản lượng thời điểm đó.

Thêm nữa, chúng ta cũng cần đẩy mạnh các kế hoạch xúc tiến như đã nêu trên tại các thành phố Seattle, Portland, Dallas, Atlanta và Austin, giúp mang lại thêm doanh thu từ các địa điểm này.

6. Hạn chế

Tuy rằng cũng đã chỉ ra được một số điểm cần chú ý và có ý nghĩa trong việc quyết định trong bài báo cáo này. Nhưng tôi nhận thấy vẫn còn một số hạn chế, khiến cho việc xử lý và đưa ra các kết quả dữ liệu chưa được tối ưu.

Thứ nhất, là về vấn đề dữ liệu doanh thu của công ty không có "Customer ID", khiến cho rất khó có thể xác định khách hàng là ai? Họ mua những thời điểm nào trong năm? Bao lâu rồi họ không mua hàng? Tổng số tiền họ đã chi là bao nhiêu, những sản phẩm nào mà khách hàng đó hay mua?. Điều này khiến cho tôi không thể thực hiện được phương pháp RFM (Recency, Frequency and Monetary), nên không thể phân cụm và xác định được đâu là khách hàng VIP, khách hàng trung thành, khách hàng mua 1 lần… từ đó rất hạn chế cho doanh nghiệp trong việc sử dụng email marketing, quảng cáo, đề xuất bán chéo… sao cho phù hợp với từng nhóm khách hàng.

Thứ hai, tuy rằng xác định được doanh thu của các thành phố cao thấp khác nhau, thời điểm khách hàng mua hàng…, nhưng không có dữ liệu thêm để giải thích rằng tại sao thành phố đó lại có doanh thu cao, hoặc thấp? Tại sao thời điểm đó khách hàng lại mua hàng nhiều?... mà chỉ có thể sử dụng phân tích định tính để trả lời.

Cuối cùng, do kinh nghiệm và thực lực còn hạn chế, do vậy chưa thể diễn tả được tất cả ý nghĩa mà dữ liệu mang lại.

7. Phụ lục

7.1. Bảng dữ liệu

7.2. Quy trình xử lý dữ liệu

In [1]:
```
#TASK 1: IMPORT DATA
```

In [18]:
```
import pandas as pd
import numpy as np

import matplotlib.pyplot as plt
import os
pd.options.mode.chained_assignment = None
```

In [3]:
```
path = "E:/Marketing PTIT/Datamining/final/"
data = pd.read_csv(path + "sales2019_6.csv")
data.head()
```

Out[3]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
0	209921	USB-C Charging Cable	1	11.95	06/23/19 19:34	950 Walnut St, Portland, ME 04101
1	209922	Macbook Pro Laptop	1	1700.0	06/30/19 10:05	80 4th St, San Francisco, CA 94016
2	209923	ThinkPad Laptop	1	999.99	06/24/19 20:18	402 Jackson St, Los Angeles, CA 90001
3	209924	27in FHD Monitor	1	149.99	06/05/19 10:21	560 10th St, Seattle, WA 98101
4	209925	Bose SoundSport Headphones	1	99.99	06/25/19 18:58	545 2nd St, San Francisco, CA 94016

In [4]:
```
#TASK 2: Clean data
```

In [5]:
```
frames=[]

for file in os.listdir(path):
    if file.endswith(".csv"):
        filepath = path + file
        df1 = pd.read_csv(filepath)
        frames.append(df1)
        result = pd.concat(frames)

df = result
df
```

Out[5]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
0	141234	iPhone	1	700	01/22/19 21:25	944 Walnut St, Boston, MA 02215
1	141235	Lightning Charging Cable	1	14.95	01/28/19 14:15	185 Maple St, Portland, OR 97035
2	141236	Wired Headphones	2	11.99	01/17/19 13:33	538 Adams St, San Francisco, CA 94016
3	141237	27in FHD Monitor	1	149.99	01/05/19 20:33	738 10th St, Los Angeles, CA 90001
4	141238	Wired Headphones	1	11.99	01/25/19 11:59	387 10th St, Austin, TX 73301
...	...	...	...	...	...	...
11681	259353	AAA Batteries (4-pack)	3	2.99	09/17/19 20:56	840 Highland St, Los Angeles, CA 90001
11682	259354	iPhone	1	700	09/01/19 16:00	216 Dogwood St, San Francisco, CA 94016
11683	259355	iPhone	1	700	09/23/19 07:39	220 12th St, San Francisco, CA 94016
11684	259356	34in Ultrawide Monitor	1	379.99	09/19/19 17:30	511 Forest St, San Francisco, CA 94016
11685	259357	USB-C Charging Cable	1	11.95	09/30/19 00:18	250 Meadow St, San Francisco, CA 94016

186850 rows × 6 columns

In [6]:
```
#ADD column
```

In [7]:
```
df["Month"] = df["Order Date"].str[0:2]
df.head()
```

Out[7]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month
0	141234	iPhone	1	700	01/22/19 21:25	944 Walnut St, Boston, MA 02215	01
1	141235	Lightning Charging Cable	1	14.95	01/28/19 14:15	185 Maple St, Portland, OR 97035	01
2	141236	Wired Headphones	2	11.99	01/17/19 13:33	538 Adams St, San Francisco, CA 94016	01
3	141237	27in FHD Monitor	1	149.99	01/05/19 20:33	738 10th St, Los Angeles, CA 90001	01
4	141238	Wired Headphones	1	11.99	01/25/19 11:59	387 10th St, Austin, TX 73301	01

In [8]:
```
### print(set(df["Month"]))
df = df.dropna(how="all")
df.head()
```

Out[8]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month
0	141234	iPhone	1	700	01/22/19 21:25	944 Walnut St, Boston, MA 02215	01
1	141235	Lightning Charging Cable	1	14.95	01/28/19 14:15	185 Maple St, Portland, OR 97035	01
2	141236	Wired Headphones	2	11.99	01/17/19 13:33	538 Adams St, San Francisco, CA 94016	01
3	141237	27in FHD Monitor	1	149.99	01/05/19 20:33	738 10th St, Los Angeles, CA 90001	01
4	141238	Wired Headphones	1	11.99	01/25/19 11:59	387 10th St, Austin, TX 73301	01

In [9]:
```
#Task 3: Answer the questions
#3.1. San pham nao co doanh thu cao nhat
```

In [10]:
```
df =df[df["Month"] != "Or"]
df.head()
print(set(df["Month"]))
print(df["Quantity Ordered"].dtypes)
print(df["Price Each"].dtypes)
```

```
{'01', '02', '07', '12', '08', '04', '11', '06', '09', '05', '03', '10'}
object
object
```

In [11]:
```
df["Quantity Ordered"] = pd.to_numeric(df["Quantity Ordered"], downcast = "integer")
df["Price Each"] = pd.to_numeric(df["Price Each"], downcast = "float")

print(df["Price Each"].dtypes)
print(df["Quantity Ordered"].dtypes)
```

```
float32
int8
```

In [12]:
```
df["Sales"] = df["Quantity Ordered"] * df["Price Each"]
df.head()
moving_column = df.pop("Sales")
df.insert(4, "Sales", moving_column)
```

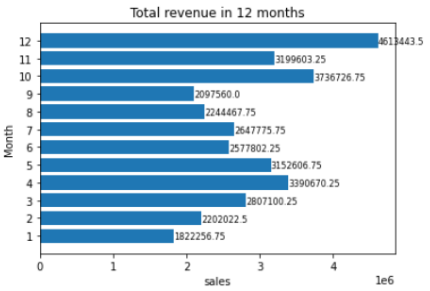
In [13]:
```
df.groupby("Month").sum()["Sales"]
```

Out[13]:
```
Month
01    1822256.75
02    2202022.50
03    2807100.25
04    3390670.25
05    3152606.75
06    2577802.25
07    2647775.75
```

In [14]:
```
sale_value = df.groupby("Month").sum()["Sales"]
sale_value.max()
```

Out[14]:
```
4613443.5
```

In [47]:
```
months = range (1,13)
plt.barh(months, sale_value)
plt.title("Total revenue in 12 months")
width = 0.8
ind = np.arange(len(sale_value))
for index, value in enumerate(sale_value):
    plt.text(value +2 , index + 0.8,
             str(value), size = 8 )
plt.yticks(months)
plt.ylabel ("Month")
plt.xlabel ("sales")
plt.show()
```



In [16]:
```
#Thanh pho nao co doanh thu cao nhat?

df.head()
```

Out[16]:

	Order ID	Product	Quantity Ordered	Price Each	Sales	Order Date	Purchase Address	Month
0	141234	iPhone	1	700.000000	700.000000	01/22/19 21:25	944 Walnut St, Boston, MA 02215	01
1	141235	Lightning Charging Cable	1	14.950000	14.950000	01/28/19 14:15	185 Maple St, Portland, OR 97035	01
2	141236	Wired Headphones	2	11.990000	23.980000	01/17/19 13:33	538 Adams St, San Francisco, CA 94016	01
3	141237	27in FHD Monitor	1	149.990005	149.990005	01/05/19 20:33	738 10th St, Los Angeles, CA 90001	01
4	141238	Wired Headphones	1	11.990000	11.990000	01/25/19 11:59	387 10th St, Austin, TX 73301	01

In [48]:
```
sample_address = "944 Walnut St, Boston, MA 02215"
sample_address.split(",")[1]
address_to_city = lambda address:address.split(",")[1]
df["City"] = df["Purchase Address"].apply(address_to_city)
```

In [49]:
```
sales_value_city = df.groupby("City").sum()["Sales"]
print(sales_value_city)
sales_value_city.max()
```

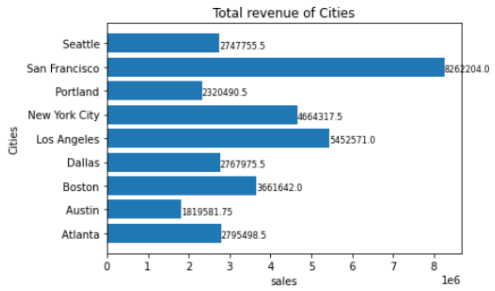
```
City
 Atlanta          2795498.50
 Austin           1819581.75
 Boston           3661642.00
 Dallas           2767975.50
 Los Angeles      5452571.00
 New York City    4664317.50
 Portland         2320490.50
 San Francisco    8262204.00
 Seattle          2747755.50
Name: Sales, dtype: float32
```

Out[49]:
```
8262204.0
```

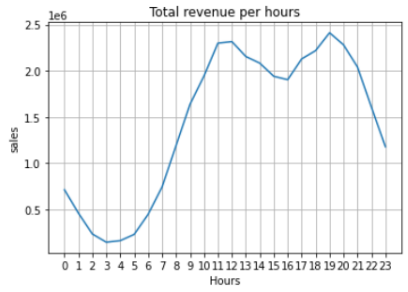
In [50]:
```
cities = []
for city, sales in sales_value_city.items():
    cities.append(city)
print(cities)
```

```
[' Atlanta', ' Austin', ' Boston', ' Dallas', ' Los Angeles', ' New York City', ' Portland', ' San Francisco', ' Seattle']
```

In [63]:
```
plt.barh( cities ,sales_value_city )
ind = np.arange(len(sales_value_city))
for index, value in enumerate(sales_value_city):
    plt.text(value +2 , index - 0.2,
             str(value), size = 8 )
plt.yticks(cities)
plt.title("Total revenue of Cities")
plt.ylabel ("Cities")
plt.xlabel ("sales")
plt.show()
```

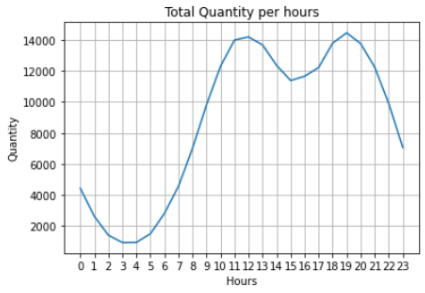


In [46]:
```
hours = [hour for hour, sales in sales_value_hour.items()] # SORT gía trị đúng vị trí
plt.plot(hours , sales_value_hour)
plt.xticks(hours)
plt.title("Total revenue per hours")
plt.grid()
plt.xlabel ("Hours")
plt.ylabel ("sales")
plt.show()
```



In [26]:
```
#2 Khung giờ nào bán được nhiều sản phẩm nhất?
```

In [47]:
```
count_hour = df.groupby("Hours").sum()["Quantity Ordered"]
plt.plot(hours , count_hour)
plt.title("Total Quantity per hours")
plt.xticks(hours)
plt.grid()
plt.xlabel ("Hours")
plt.ylabel ("Quantity")
plt.show()
```



In [53]:
```
### Sản phẩm nào được mua cùng nhau
```

In [68]:
```
df_dup = df[df["Order ID"].duplicated(keep = False)]
df_dup
```

Out[68]:

	Order ID	Product	Quantity Ordered	Price Each	Sales	Order Date	Purchase Address	Month	City
41	141275	USB-C Charging Cable	1	11.95	11.95	01/07/19 16:06	610 Walnut St, Austin, TX 73301	01	Austin
42	141275	Wired Headphones	1	11.99	11.99	01/07/19 16:06	610 Walnut St, Austin, TX 73301	01	Austin
57	141290	Apple Airpods Headphones	1	150.00	150.00	01/02/19 08:25	4 1st St, Los Angeles, CA 90001	01	Los Angeles
58	141290	AA Batteries (4-pack)	3	3.84	11.52	01/02/19 08:25	4 1st St, Los Angeles, CA 90001	01	Los Angeles
133	141365	Vareebadd Phone	1	400.00	400.00	01/10/19 11:19	20 Dogwood St, New York City, NY 10001	01	New York City
...	...	...	...	...	...	...	...	...	...
11628	259303	AA Batteries (4-pack)	1	3.84	3.84	09/20/19 20:18	106 7th St, Atlanta, GA 30301	09	Atlanta
11639	259314	Wired Headphones	1	11.99	11.99	09/16/19 00:25	241 Highland St, Atlanta, GA 30301	09	Atlanta
11640	259314	AAA Batteries (4-pack)	2	2.99	5.98	09/16/19 00:25	241 Highland St, Atlanta, GA 30301	09	Atlanta
11677	259350	Google Phone	1	600.00	600.00	09/30/19 13:49	519 Maple St, San Francisco, CA 94016	09	San Francisco
11678	259350	USB-C Charging Cable	1	11.95	11.95	09/30/19 13:49	519 Maple St, San Francisco, CA 94016	09	San Francisco

14649 rows × 9 columns

In [66]:
```
groupProduct = lambda product: ", ".join(product)
```

In [69]:
```
df_dup["All Products"] = df_dup.groupby('Order ID')["Product"].transform(groupProduct)
df_dup.head()
```

Out[69]:

	Order ID	Product	Quantity Ordered	Price Each	Sales	Order Date	Purchase Address	Month	City	All Products
41	141275	USB-C Charging Cable	1	11.95	11.95	01/07/19 16:06	610 Walnut St, Austin, TX 73301	01	Austin	USB-C Charging Cable, Wired Headphones
42	141275	Wired Headphones	1	11.99	11.99	01/07/19 16:06	610 Walnut St, Austin, TX 73301	01	Austin	USB-C Charging Cable, Wired Headphones
57	141290	Apple Airpods Headphones	1	150.00	150.00	01/02/19 08:25	4 1st St, Los Angeles, CA 90001	01	Los Angeles	Apple Airpods Headphones, AA Batteries (4-pack)
58	141290	AA Batteries (4-pack)	3	3.84	11.52	01/02/19 08:25	4 1st St, Los Angeles, CA 90001	01	Los Angeles	Apple Airpods Headphones, AA Batteries (4-pack)
133	141365	Vareebadd Phone	1	400.00	400.00	01/10/19 11:19	20 Dogwood St, New York City, NY 10001	01	New York City	Vareebadd Phone, Wired Headphones

In [70]:
```
df_dup = df_dup[['Order ID',"All Products"]].drop_duplicates()
df_dup.head()
```

Out[70]:

	Order ID	All Products
41	141275	USB-C Charging Cable, Wired Headphones
57	141290	Apple Airpods Headphones, AA Batteries (4-pack)
133	141365	Vareebadd Phone, Wired Headphones
153	141384	Google Phone, USB-C Charging Cable
220	141450	Google Phone, Bose SoundSport Headphones

In [95]:
```
df_dup["All Products"].value_counts().head(5).reset_index()
```

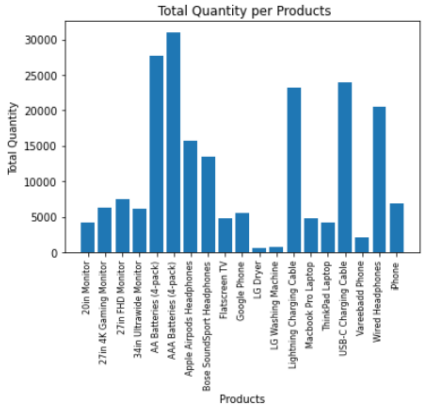
Out[95]:

	index	All Products
0	iPhone, Lightning Charging Cable	882
1	Google Phone, USB-C Charging Cable	856
2	iPhone, Wired Headphones	361
3	Vareebadd Phone, USB-C Charging Cable	312
4	Google Phone, Wired Headphones	303

In [84]:
```
all_products = df.groupby("Product").sum()["Quantity Ordered"]
all_products
```

Out[84]:
```
Product
20in Monitor                  4129.0
27in 4K Gaming Monitor        6244.0
27in FHD Monitor              7550.0
34in Ultrawide Monitor        6199.0
AA Batteries (4-pack)        27635.0
AAA Batteries (4-pack)       31017.0
Apple Airpods Headphones     15661.0
Bose SoundSport Headphones   13457.0
Flatscreen TV                 4819.0
Google Phone                  5532.0
LG Dryer                       646.0
LG Washing Machine             666.0
Lightning Charging Cable     23217.0
Macbook Pro Laptop            4728.0
ThinkPad Laptop               4130.0
USB-C Charging Cable         23975.0
Vareebadd Phone               2068.0
Wired Headphones             20557.0
iPhone                        6849.0
Name: Quantity Ordered, dtype: float64
```

In [85]:
```
products_is = [product for product, quant in all_products.items()]
plt.bar(x= products_is , height = all_products)
plt.title(" Total Quantity per Products")
plt.xticks(products_is, rotation = 90, size = 8)
plt.xlabel ("Products")
plt.ylabel ("Total Quantity")
plt.show()
```



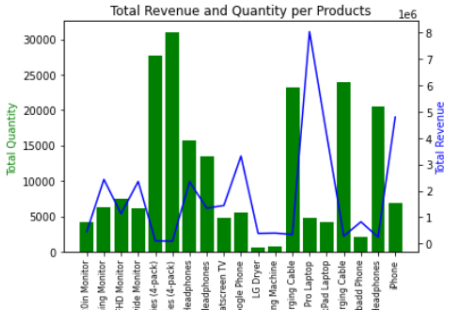
In [94]:
```
x = products_is
y1 = all_products
y2 = total_prices

fig, ax1 = plt.subplots()
plt.title("Total Revenue and Quantity per Products")
ax2 = ax1.twinx()
ax1.bar(x, y1, color = "g")
ax2.plot(x, y2, 'b-')

ax1.set_xticklabels(products_is, rotation = 90, size = 8)
ax1.set_xlabel('Products', color = "r")
ax1.set_ylabel('Total Quantity ', color='g')
ax2.set_ylabel('Total Revenue', color='b')

plt.show()
```

```
C:\Users\Admin\AppData\Local\Temp/ipykernel_6828/1053424210.py:11: UserWarning: FixedFormatter should only be used together wit
h FixedLocator
  ax1.set_xticklabels(products_is, rotation = 90, size = 8)
```



In [7]:
```
transaction_list = []
for i in data["Order ID"].unique():
    tlist = list(set(data[data["Order ID"] ==i]["Product"]))
    if len(tlist) > 0:
        transaction_list.append(tlist)
print (len(transaction_list))
```

```
7136
```

In [8]:
```
transaction_list
```

Out[8]:
```
[['USB-C Charging Cable', 'Wired Headphones'],
 ['Apple Airpods Headphones', 'AA Batteries (4-pack)'],
 ['Wired Headphones', 'Vareebadd Phone'],
 ['Google Phone', 'USB-C Charging Cable'],
 ['Bose SoundSport Headphones', 'Google Phone'],
 ['Apple Airpods Headphones', 'iPhone'],
 ['Apple Airpods Headphones', 'Google Phone'],
 ['Apple Airpods Headphones', 'Wired Headphones', 'iPhone'],
 ['Apple Airpods Headphones', 'USB-C Charging Cable'],
 ['Lightning Charging Cable', 'Wired Headphones'],
 ['Lightning Charging Cable', 'iPhone'],
 ['Bose SoundSport Headphones', '27in FHD Monitor'],
 ['Google Phone', 'USB-C Charging Cable'],
 ['Wired Headphones', 'iPhone'],
 ['Google Phone', 'USB-C Charging Cable'],
 ['AAA Batteries (4-pack)', 'AA Batteries (4-pack)'],
 ['USB-C Charging Cable', 'Wired Headphones', 'Vareebadd Phone'],
 ['Google Phone', 'USB-C Charging Cable'],
 ['Wired Headphones', 'Vareebadd Phone'],
```

In [9]:
```
from mlxtend.preprocessing import TransactionEncoder
```

In [10]:
```
te = TransactionEncoder()
te_ary = te.fit(transaction_list).transform(transaction_list)
te_ary
```

Out[10]:
```
array([[False, False, False, ..., False,  True, False],
       [False, False, False, ..., False, False, False],
       [False, False, False, ...,  True,  True, False],
       ...,
       [False, False, False, ..., False, False, False],
       [False, False, False, ..., False,  True, False],
       [False, False, False, ..., False, False, False]])
```

In [11]:
```
te_ary.astype("int")
```

Out[11]:
```
array([[0, 0, 0, ..., 0, 1, 0],
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 1, 1, 0],
       ...,
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 1, 0],
       [0, 0, 0, ..., 0, 0, 0]])
```

In [12]:
```
df2 = pd.DataFrame(te_ary, columns=te.columns_)
df2
```

Out[12]:

	20in Monitor	27in 4K Gaming Monitor	27in FHD Monitor	34in Ultrawide Monitor	AA Batteries (4-pack)	AAA Batteries (4-pack)	Apple Airpods Headphones	Bose SoundSport Headphones	Flatscreen TV	Google Phone	LG Dryer	LG Washing Machine	Lightning Charging Cable	Macbook Pro Laptop	ThinkPad Laptop
0	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	True	False	True	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	True	False	False	False	False	False
4	False	False	False	False	False	False	False	True	False	True	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
7131	False	False	False	False	False	False	True	False	False	False	False	False	False	False	False
7132	False	False	False	False	False	False	False	False	False	False	False	False	True	False	False
7133	False	False	False	True	True	False	False	False	False	False	False	False	False	False	False
7134	False	False	False	False	False	True	False	False	False	False	False	False	False	False	False
7135	False	False	False	False	False	False	False	False	False	True	False	False	False	False	False

7136 rows × 19 columns

In [13]:
```
df2.shape
```

Out[13]:
```
(7136, 19)
```

In [14]:
```
frq_items = apriori(df2, min_support = 0.01, use_colnames = True)
rules = association_rules(frq_items, metric ="lift", min_threshold = 1)
frq_items["length"] = frq_items["itemsets"].apply( lambda x : len(x))
```

In [18]:
```
rules = rules.sort_values(['confidence'], ascending = False)
rules.head(10)
```

Out[18]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
15	(Vareebadd Phone)	(USB-C Charging Cable)	0.084221	0.289098	0.051570	0.612313	2.118015	0.027221	1.833701
8	(Google Phone)	(USB-C Charging Cable)	0.229260	0.289098	0.139714	0.609413	2.107985	0.073436	1.820088
12	(Lightning Charging Cable)	(iPhone)	0.248459	0.261351	0.141676	0.570220	2.181818	0.076741	1.718668
13	(iPhone)	(Lightning Charging Cable)	0.261351	0.248459	0.141676	0.542091	2.181818	0.076741	1.641247
9	(USB-C Charging Cable)	(Google Phone)	0.289098	0.229260	0.139714	0.483277	2.107985	0.073436	1.491591
20	(USB-C Charging Cable, Wired Headphones)	(Google Phone)	0.028447	0.229260	0.012192	0.428571	1.869368	0.005670	1.348795
2	(Apple Airpods Headphones)	(iPhone)	0.133128	0.261351	0.052270	0.392632	1.502316	0.017477	1.216147
4	(Bose SoundSport Headphones)	(Google Phone)	0.111127	0.229260	0.031951	0.287516	1.254103	0.006474	1.081764
19	(Wired Headphones)	(iPhone)	0.229680	0.261351	0.064742	0.281879	1.078547	0.004715	1.028586
10	(Google Phone)	(Wired Headphones)	0.229260	0.229680	0.059137	0.257946	1.123065	0.006480	1.038091