Analysis of Amazon's product sales

Dataset

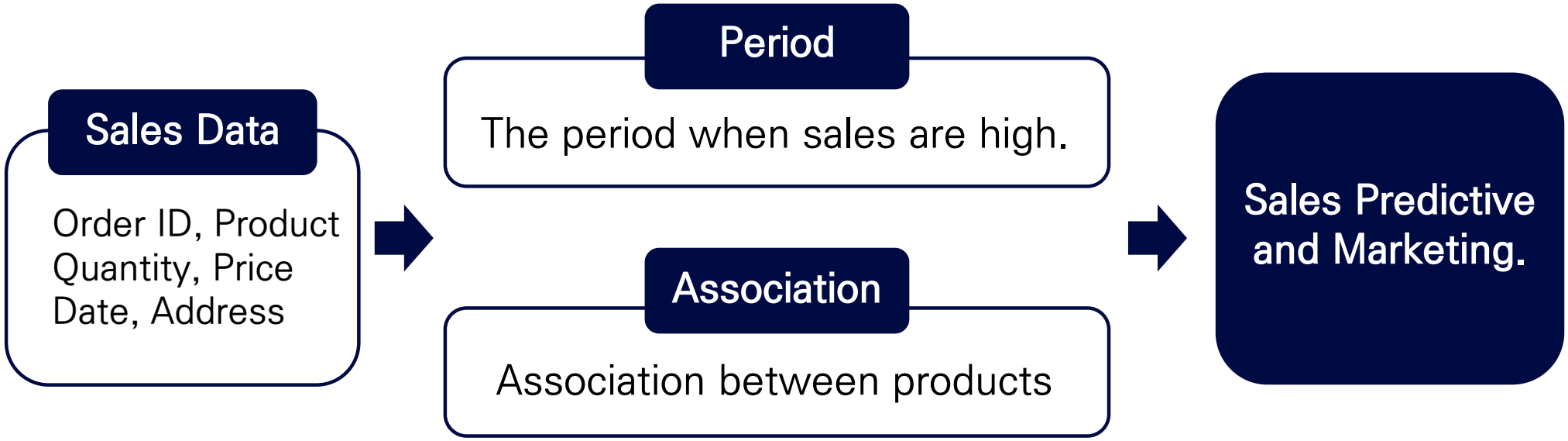
Data Description

- **Open Data Portal** : Kaggle (https://www.kaggle.com/)
- **Title** : Sales Product Data
- **Link** : https://www.kaggle.com/knightbearr/sales-product-data?select=Sales_September_2019.csv
- **Content** : Amazon's product sales data for 2019
- **File Type** : .CSV
- **Column**: Order ID, Product, Quantity Ordered, Price Each, Order Date, Purchase Address

	A	B	C	D	E	F
1	Order ID	Product	Quantity Ord	Price Each	Order Date	Purchase Address
2	176558	USB-C Charging Cable	2	11.95	04/19/19 08:46	917 1st St, Dallas, TX 75001
3	176559	Bose SoundSport Headphc	1	99.99	2004-07-19 22:30	682 Chestnut St, Boston, MA 02215
4	176560	Google Phone	1	600	2004-12-19 14:38	669 Spruce St, Los Angeles, CA 90001
5	176560	Wired Headphones	1	11.99	2004-12-19 14:38	669 Spruce St, Los Angeles, CA 90001
6	176561	Wired Headphones	1	11.99	04/30/19 09:27	333 8th St, Los Angeles, CA 90001
7	176562	USB-C Charging Cable	1	11.95	04/29/19 13:03	381 Wilson St, San Francisco, CA 94016
8	176563	Bose SoundSport Headphc	1	99.99	2004-02-19 7:46	668 Center St, Seattle, WA 98101
9	176564	USB-C Charging Cable	1	11.95	2004-12-19 10:58	790 Ridge St, Atlanta, GA 30301
10	176565	Macbook Pro Laptop	1	1700	04/24/19 10:38	915 Willow St, San Francisco, CA 94016
11	176566	Wired Headphones	1	11.99	2004-08-19 14:05	83 7th St, Boston, MA 02215
12	176567	Google Phone	1	600	04/18/19 17:18	444 7th St, Los Angeles, CA 90001
13	176568	Lightning Charging Cable	1	14.95	04/15/19 12:18	438 Elm St, Seattle, WA 98101
14	176569	27in 4K Gaming Monitor	1	389.99	04/16/19 19:23	657 Hill St, Dallas, TX 75001
15	176570	AA Batteries (4-pack)	1	3.84	04/22/19 15:09	186 12th St, Dallas, TX 75001
16	176571	Lightning Charging Cable	1	14.95	04/19/19 14:29	253 Johnson St, Atlanta, GA 30301
17	176572	Apple Airpods Headphone	1	150	2004-04-19 20:30	149 Dogwood St, New York City, NY 10001
18	176573	USB-C Charging Cable	1	11.95	04/27/19 18:41	214 Chestnut St, San Francisco, CA 94016
19	176574	Google Phone	1	600	2004-03-19 19:42	20 Hill St, Los Angeles, CA 90001
20	176574	USB-C Charging Cable	1	11.95	2004-03-19 19:42	20 Hill St, Los Angeles, CA 90001

Motivation

I want to help with product marketing and decision-making by analyzing when sales are high or what products are sold together.



Research Questions

Q1. When is the month with the highest sales ?

Q2. When is the time zone with the highest sales?

Q3. What's the product that sells the most together?

Expectations on Finding

Q1. When is the month with the highest sales ?

➡ **A1. November**

Because there is Black Friday, sales will be the highest.

Q2. When is the time zone with the highest sales?

➡ **A2. Between 7 p.m. and 9 p.m**

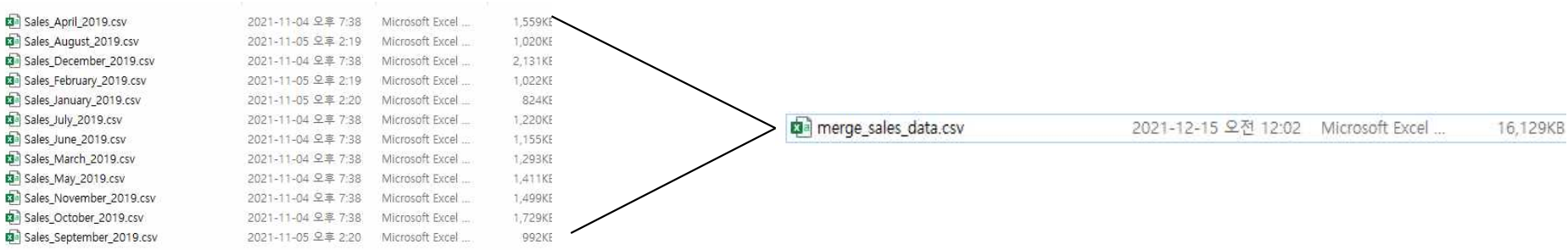
There would be a lot of customers shopping after work and dinner.

Q3. What's the product that sells the most together?

➡ **A3. Electronics and peripherals**

Finding - Preprocessing

'January to December' sales data.csv → merge_sales_data.csv



Import Dataset with COLAB

```
[2] import pandas as pd
    import numpy as np
    import os
    import matplotlib.pyplot as plt
    from google.colab import drive
    drive.mount('/content/drive')

    Mounted at /content/drive
```

Preprocessing

```
[3] data =pd.read_csv('/content/drive/MyDrive/data/merge_sales_data.csv')

[5] data.shape

    (185950, 6)

    data.head()
```

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
0	236670	Wired Headphones	2	11.99	08/31/19 22:21	359 Spruce St, Seattle, WA 98101
1	236671	Bose SoundSport Headphones	1	99.99	08/15/19 15:11	492 Ridge St, Dallas, TX 75001
2	236672	iPhone	1	700.00	2008-06-19 14:40	149 7th St, Portland, OR 97035
3	236673	AA Batteries (4-pack)	2	3.84	08/29/19 20:59	631 2nd St, Los Angeles, CA 90001
4	236674	AA Batteries (4-pack)	2	3.84	08/15/19 19:53	736 14th St, New York City, NY 10001

Finding - Preprocessing

```
[ ] data.isnull().sum() # 데이터 결측치 확인

Order ID            0
Product             0
Quantity Ordered    0
Price Each          0
Order Date          0
Purchase Address    0
dtype: int64
```

```
[ ] data.info()  # order date가 object형식인 것을 확인 가능

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 185950 entries, 0 to 185949
Data columns (total 6 columns):
 #   Column            Non-Null Count   Dtype
---  ------            --------------   -----
 0   Order ID          185950 non-null  int64
 1   Product           185950 non-null  object
 2   Quantity Ordered  185950 non-null  int64
 3   Price Each        185950 non-null  float64
 4   Order Date        185950 non-null  object
 5   Purchase Address  185950 non-null  object
dtypes: float64(1), int64(2), object(3)
memory usage: 8.5+ MB
```

```
[ ] data['Order Date_format']=pd.to_datetime(data['Order Date']) # datetime형식으로 변환 후 'Order Date_format' 컬럼으로 저장
```

```
[ ] data.head()  # Order Date_format 컬럼이 추가 된 것을 확인 가능
```

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Order Date_format
0	236670	Wired Headphones	2	11.99	08/31/19 22:21	359 Spruce St, Seattle, WA 98101	2019-08-31 22:21:00
1	236671	Bose SoundSport Headphones	1	99.99	08/15/19 15:11	492 Ridge St, Dallas, TX 75001	2019-08-15 15:11:00
2	236672	iPhone	1	700.00	2008-06-19 14:40	149 7th St, Portland, OR 97035	2008-06-19 14:40:00
3	236673	AA Batteries (4-pack)	2	3.84	08/29/19 20:59	631 2nd St, Los Angeles, CA 90001	2019-08-29 20:59:00

Step1. 결측치 존재 유무 확인

결측치가 없으므로 그대로 진행

Step2. 데이터 타입 변환

Order Date를 년/월/일/시 로 나누기 위해 데이터 타입을 Object → datetime으로 변환한 데이터를 'Order Date_format '이라는 새로운 컬럼으로 생성

Finding - Preprocessing

Step3. Order Date_format을 년/월/일/시 로 나눈 후 컬럼을 추가하여 각 데이터 저장

```
#분석을 위해 년/월/일/시/분 으로 나누어 컬럼생성
data['Order_year'] = data['Order Date_format'].dt.year
data['Order_month'] = data['Order Date_format'].dt.month
data['Order_day'] = data['Order Date_format'].dt.day
data['Order_hour'] = data['Order Date_format'].dt.hour
data['Order_minute'] = data['Order Date_format'].dt.minute
```

+ 코드 + 텍스트

```
[ ] data.head()
```

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Order Date_format	Order_year	Order_month	Order_day	Order_hour	Order_minute
0	236670	Wired Headphones	2	11.99	08/31/19 22:21	359 Spruce St, Seattle, WA 98101	2019-08-31 22:21:00	2019	8	31	22	21
1	236671	Bose SoundSport Headphones	1	99.99	08/15/19 15:11	492 Ridge St, Dallas, TX 75001	2019-08-15 15:11:00	2019	8	15	15	11
2	236672	iPhone	1	700.00	2008-06-19 14:40	149 7th St, Portland, OR 97035	2008-06-19 14:40:00	2008	6	19	14	40
3	236673	AA Batteries (4-pack)	2	3.84	08/29/19 20:59	631 2nd St, Los Angeles, CA 90001	2019-08-29 20:59:00	2019	8	29	20	59
4	236674	AA Batteries (4-pack)	2	3.84	08/15/19 19:53	736 14th St, New York City, NY 10001	2019-08-15 19:53:00	2019	8	15	19	53

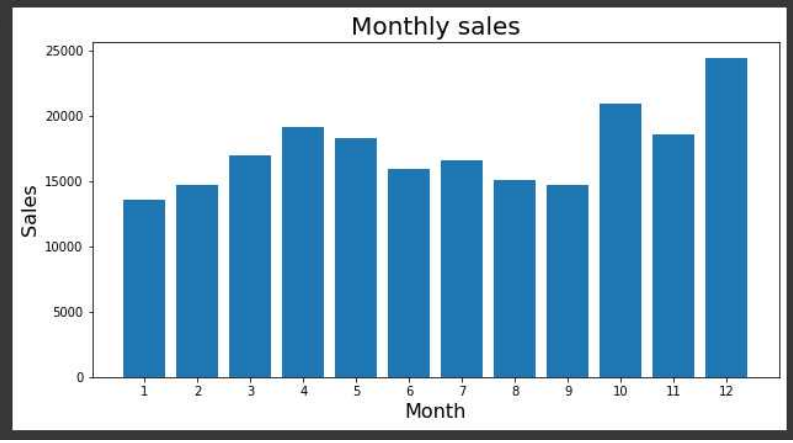
Research Question 1

```
[ ]  fre_month = data.groupby('Order_month')['Quantity Ordered'].sum()
     fre_month.sort_values(ascending = False)

Order_month
12    24441
10    20956
4     19150
11    18606
5     18338
3     16930
7     16613
6     15911
8     15101
2     14729
9     14709
1     13595
Name: Quantity Ordered, dtype: int64
```

〈Code〉

```
labels = fre_month.index
plt.subplots(figsize=(10, 5))
plt.bar(labels, fre_month)
plt.xticks(labels)
plt.title('Monthly sales', fontsize = 20)
plt.xlabel('Month', fontsize = 16)
plt.ylabel('Sales', fontsize = 16)
plt.show()
```



〈Graph〉

판매량이 가장 높은 달을 찾기 위해 Order_month를 기준으로 그룹화해서
'Quantity Ordered' 를 합한 수치를 나타내고, 값 기준으로 내림차순 정렬하여 가장 판매량이 많은 달을 찾아냈음.

* (12월 〉 10월 〉 4월 〉 … 〉 2월 〉 9월 〉 1월) 순으로 판매량이 높았음

Q1. When is the month with the highest sales ?

A1. December

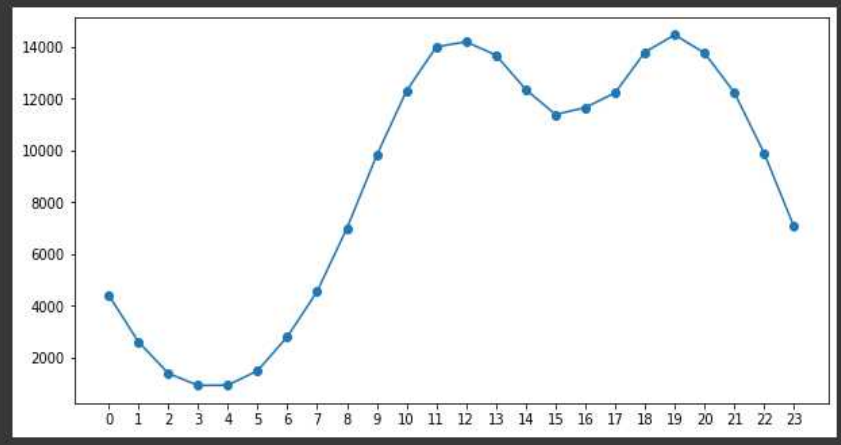
Research Question 2

```
fre_time = data.groupby('Order_hour')['Quantity Ordered'].sum()
fre_time.sort_values(ascending = False)
```

```
Order_hour
19    14470
12    14202
11    14005
18    13802
20    13768
13    13685
14    12362
10    12308
21    12244
17    12229
16    11662
15    11391
22     9899
9      9816
23     7065
8      7002
7      4556
0      4428
6      2810
1      2619
5      1493
2      1398
4       937
3       928
Name: Quantity Ordered, dtype: int64
```

〈Code〉

```
labels = fre_time.index
plt.subplots(figsize=(10, 5))
plt.plot(labels, fre_time,marker = 'o')
plt.xticks(labels)
plt.show()
```



〈Graph〉

판매량이 가장 높은 시간대를 찾기 위해 Order_hour를 기준으로 그룹화해서
'Quantity Ordered' 를 합한 수치를 나타내고, 값 기준으로 내림차순 정렬하여 가장 판매량이 많은 시간대를 찾아냈음.

* (19시 〉 12시 〉 11시 〉 … 〉 2시 〉 4시 〉 3시) 순으로 판매량이 높았음

Q2. When is the time zone with the highest sales?

A2. 19:00 ~ 19:59

Research Question 3

Q3. What's the product that sells the most together?

연관성분석 (Apriori)를 하기 위해 데이터셋을 다시 정제하는 과정을 진행함.

```
[ ] # duplicated 함수를 통해 같은 ID로 구매한 데이터만 추출하여 data1으로 저장
    data1 = data[data['Order ID'].duplicated(keep =False)]

[ ]  #data1에서 Order ID, Product 컬럼만 추출해서 data2에 저장
    data2=data1.iloc[:,:2]

[ ] data2.head()
```

	Order ID	Product
46	236716	AA Batteries (4-pack)
47	236716	USB-C Charging Cable
60	236729	iPhone
61	236729	Apple Airpods Headphones
62	236730	Google Phone

```
[ ] # Product컬럼의 범주형 데이터를 pd.get_dummies를 활용하여 0,1의 형태로 변환 → 연관성 분석을 하기 위함
    data2 = pd.get_dummies(data2)

[ ] # 결과 확인
    data2.head()
```

	Order ID	Product_20in Monitor	Product_27in 4K Gaming Monitor	Product_27in FHD Monitor	Product_34in Ultrawide Monitor	Product_AA Batteries (4-pack)	Product_AAA Batteries (4-pack)	Product_Apple Airpods Headphones	Product_Bose SoundSport Headphones	Product_Flat
46	236716	0	0	0	0	1	0	0	0	
47	236716	0	0	0	0	0	0	0	0	
60	236729	0	0	0	0	0	0	0	0	
61	236729	0	0	0	0	0	0	1	0	
62	236730	0	0	0	0	0	0	0	0	

Step1. 중복ID 추출

같은 ID로 구매한 제품들은 같이 구매한 것이기 때문에 중복ID를 추출하기 위해 duplicate함수를 활용해 Order ID가 중복인 데이터를 추출

Step2. 필요한 컬럼 추출

데이터 중 Order ID와 Product만 필요하기에 해당하는 컬럼 2개만 추출하여 data2에 저장

Step3. 범주형 데이터→ 수치형 데이터

Apriori를 하기 위해선 모든 데이터가 True/False 또는 0/1로 이루어져야하기 때문에 pd.get_dummies()를 이용하여 One-Hot Encoding 진행.

* Product의 제품들이 컬럼으로 올라가고 해당하면 1, 아니면 0으로 변환되었음.

Research Question 3

```
[11] # ID를 기준으로 동일한 ID를 가진 행은 더해서 encoder_data에 저장
     encoder_data = data2.groupby('Order ID').sum()

     # 결과 확인
     encoder_data.head()
```

Order ID	Product_20in Monitor	Product_27in 4K Gaming Monitor	Product_27in FHD Monitor	Product_34in Ultrawide Monitor	Product_AA Batteries (4-pack)	Product_AAA Batteries (4-pack)	Product_Apple Airpods Headphones	Product_Bose SoundSport Headphones	Product_Flatscr
141275	0	0	0	0	0	0	0	0	
141290	0	0	0	0	1	0	1	0	
141365	0	0	0	0	0	0	0	0	
141384	0	0	0	0	0	0	0	0	
141450	0	0	0	0	0	0	0	1	

```
[13] #제품명을 쉽게 알아보기 위해 컬럼에 공통적으로 들어가는 "Product_"을 공백으로 바꿔주었음.
     encoder_data.columns = encoder_data.columns.str.replace('Product_','')

[14] # 결과 확인
     encoder_data.head()
```

Order ID	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	Lightn Charg Ca
141275	0	0	0	0	0	0	0	0	0	0	0	0	
141290	0	0	0	0	1	0	1	0	0	0	0	0	
141365	0	0	0	0	0	0	0	0	0	0	0	0	

Step4. 중복제거

같은 ID로 구매한 데이터들을 합치기 위해 Order ID를 기준으로 그룹화해서 각 숫자를 합하고, 이를 encoder_data로 저장

- 같은ID가진 행이 제거되고, 더해짐으로써 기존 행에 1이 하나만 존재했던 것이 구매한 개수에 따라 2~3개 존재하게 됨.

Step5. 컬럼명 변경

get_dummies를 사용한 탓에 컬럼명이 Product_000 로 변경되어서 이를 replace()함수를 사용하여 “Product_”를 공백으로 바꾸어 주었음.

Research Question 3

```
# Apriori 분석을 하기 전, 데이터가 0,1만 있는지 확인하기 위해 max함수를 이용해 각 컬럼별 최대값 검색
# 3개의 컬럼을 제외하고는 2가 있는 것을 확인
encoder_data.max()
```

```
20in Monitor                2
27in 4K Gaming Monitor      2
27in FHD Monitor            2
34in Ultrawide Monitor      2
AA Batteries (4-pack)       2
AAA Batteries (4-pack)      2
Apple Airpods Headphones    2
Bose SoundSport Headphones  2
Flatscreen TV               2
Google Phone                2
LG Dryer                    1
LG Washing Machine          1
Lightning Charging Cable    2
Macbook Pro Laptop          2
ThinkPad Laptop             2
USB-C Charging Cable        2
Vareebadd Phone             1
Wired Headphones            2
iPhone                      2
dtype: uint8
```

Step6. 데이터 중 0,1을 제외한 값 처리

Apriori를 사용하려면 데이터셋에 0,1만 존재해야 하기 때문에 다른 값이 존재하는지 확인하기 위해 max를 이용하였음.

- 3개의 컬럼을 제외한 컬럼에서 2가 최대값임을 확인. (같은 ID를 기준으로 데이터를 합쳤기 때문에 중복데이터가 삭제되지 않고 집계된 것이라 생각)

```
[ ]  # 2는 제거하지않고, 1로 replace 해주었음.
     encoder_data =encoder_data.replace(2,1)
```

```
     # 데이터 내 2가 대체되어 사라진 것을 확인 할 수 있음.
     encoder_data[encoder_data == 2].sum()
```

```
20in Monitor                0.0
27in 4K Gaming Monitor      0.0
27in FHD Monitor            0.0
34in Ultrawide Monitor      0.0
AA Batteries (4-pack)       0.0
AAA Batteries (4-pack)      0.0
Apple Airpods Headphones    0.0
Bose SoundSport Headphones  0.0
Flatscreen TV               0.0
Google Phone                0.0
LG Dryer                    0.0
LG Washing Machine          0.0
Lightning Charging Cable    0.0
Macbook Pro Laptop          0.0
ThinkPad Laptop             0.0
USB-C Charging Cable        0.0
Vareebadd Phone             0.0
Wired Headphones            0.0
iPhone                      0.0
dtype: float64
```

위의 이유로 2를 1로 replace 진행

* 모든 컬럼에 2가 사라진 것 확인가능

Research Question 3

```
[ ] #Apriori분석을 위해 import
    from mlxtend.frequent_patterns import apriori

    #절제된 encoder_data를 사용해 apriori로 최소지지도0.005 이상인 itemsets를 찾아 지지도 기준으로 내림차순 하였음.
    frequent_product = apriori(encoder_data, min_support=0.005,  use_colnames=True, )
    frequent_product.sort_values('support',ascending=False,inplace=True)

[ ] # 지지도 상위 10개의 조합을 검색
    # len = 1인 itemsets를 제외하고 가장 높은 지지도를 보여주는
    # (iPhone, Lightning Charging Cable) 조합이 전체 sales 데이터 중 가장 많이 팔린 조합인 것으로 결과가 나옴.
    frequent_product.head(10)
```

	support	itemsets
13	0.289098	(USB-C Charging Cable)
16	0.261351	(iPhone)
10	0.248459	(Lightning Charging Cable)
15	0.229680	(Wired Headphones)
9	0.229260	(Google Phone)
45	0.141676	(iPhone, Lightning Charging Cable)
41	0.139714	(USB-C Charging Cable, Google Phone)
6	0.133128	(Apple Airpods Headphones)
7	0.111127	(Bose SoundSport Headphones)
5	0.107483	(AAA Batteries (4-pack))

Step7. 연관성분석(지지도 활용)

Apriori 를 사용하기 위해 import를 해주고, encoder_data를 활용해 지지도가 0.005이상인 데이터들을 frequent_product에 저장하고, 내림차순 정렬해주었음.

지지도(support) 상위 10개의 조합을 검색한 결과 itemsets가 1개인 것을 제외하고는 [iPhone, Lightning Charging Cable] 조합이 가장 높게 나옴.

* 지지도 = A제품과 B제품을 같이 구입한 횟수 / 전체 구매 횟수

Q3. What's the product that sells the most together?

A3. iPhone – Lightning Charging Cable

Discussions

1. 예측결과와 분석결과

- Q1의 경우 블랙프라이데이의 할인 덕분에 11월이 가장 많이 판매될 줄 알았으나, Amazon은 11월 중순부터 12월 말까지 블랙프라이데이 + 크리스마스 할인을 진행하는데 고객들이 많이 찾는 인기상품들은 12월 초에 물량이 많이 풀리는 것과, 크리스마스 선물을 구매하는 것이 더해져 이러한 분석 결과가 나오지 않았나 생각하게 되었습니다.

- Q2의 경우에는 예측했던 19시~21시가 정확히 맞은 건 아니지만, 19시는 가장 많은 판매량, 20,21시 또한 상위권에 포함되어 있어 어느정도 맞는 예측이었다고 생각합니다. 그리고 11:00~12:59의 판매량이 높은 이유는 점심시간과 관련이 있을 것이라 생각합니다.

- Q3는 분석결과 상위권 조합은 모두 전자기기 + 주변기기로 예상이 맞았습니다.

2. 기대효과

- 분석한 내용으로 보면 10월 11월 12월 / 19시~21시, 11시~1시 등 구매를 많이 하는 기간 전에 광고를 하는 등의 마케팅에 활용이 가능할 것이라 생각합니다. 또한 해당 데이터에 구매자의 정보까지 더해진다면 성별,나이,지역에 따른 구매성향을 분석해 고객추천 시스템과 같은 서비스까지 확장 할 수 있습니다. 더 방대한 데이터를 가지고 제품들의 조합을 분석한다면 온라인 뿐 아니라 오프라인에서도 매장 내 상품 진열 방식에 변화를 주는 등의 활용이 가능할 것이라 생각합니다.