

✓ ###전체 데이터

```
# prompt: /content/AdventureWorks Sales.xlsx 보여줘

import pandas as pd

# Load the Excel file into a pandas DataFrame.
df = pd.read_excel('/content/drive/MyDrive/AdventureWorks Sales.xlsx')

# Display the DataFrame.
df
```

✓ ###명확한 분석을 위한 명확한 시트 값 출력

```
import pandas as pd

# 데이터 불러오기
file_path = '/content/drive/MyDrive/AdventureWorks Sales.xlsx'
data = pd.read_excel(file_path, sheet_name=None)

# 모든 시트 이름 출력
sheet_names = data.keys()
sheet_names
```

✓ ###명확한 분석을 위한 명확한 칼럼 값 출력

```
# prompt: 각 시트별 칼럼값을 출력하라
# 시트명=dict_keys(['Sales Order_data', 'Sales Territory_data', 'Sales_data', 'Reseller_data', 'Date_data', 'Product_data', 'Customer_data'])

# 각 시트별 칼럼값 출력
for sheet_name in sheet_names:
    print(f"--- {sheet_name} ---")
    print(data[sheet_name].columns.tolist())
    print("\n\n")
```

✓ ###1. 상품별 매출 기여도 및 인기 상품 분석

```
# Load the Product_data sheet
product_data = pd.read_excel(file_path, sheet_name='Product_data')

# 확인: product_data 컬럼 출력
print(product_data.columns)
```

```
Index(['ProductKey', 'SKU', 'Product', 'Standard Cost', 'Color', 'List Price',
      'Model', 'Subcategory', 'Category'],
      dtype='object')
```


- 색상 및 크기의 영향:

- Black 색상이 Silver 색상보다 높은 매출을 기록했으며, 크기별 매출 기여도는 **38 > 42 > 46** 순으로 나타남.
- 인기 상품의 특정 색상과 크기에 대한 고객 선호도가 반영된 것으로 보임.

2. 일반 상품군 특성

- 다양한 옵션 제공의 효과:

- 동일한 상품군 내에서도 색상 및 크기 옵션이 다르게 제공되어 고객 선택 폭을 넓힘.

- 매출 기여도의 집중화:

- 상위 5개 상품이 전체 매출에서 큰 비중을 차지하며, 나머지 상품은 상대적으로 낮은 기여도를 기록할 가능성이 있음.

3. 상품별 매출 기여도 분포

- Mountain-200 Black, 38:

- 전체 매출의 가장 큰 비중을 차지하며, 고객의 주요 관심 상품.

- Mountain-200 시리즈:

- 상위 5개 상품이 모두 Mountain-200 시리즈로 구성되어 있어 해당 라인이 고객들에게 매우 높은 인기를 끌고 있음.

- 매출의 집중성:

- 특정 상품군(Mountain-200)의 매출 집중 현상은 마케팅 및 재고 관리 전략에 중요한 요소로 작용.

주요 개선점

1. 인기 상품의 집중 관리

- Mountain-200 시리즈:

- 생산 및 재고를 집중적으로 관리하여 인기 상품의 품질을 방지.

- 추가적인 마케팅 투자:

- Mountain-200 시리즈에 대한 광고, 리뷰 강화 등을 통해 매출을 더욱 증대.

2. 고객 선호도 분석 확대

- 색상 및 크기 선호도 분석:

- Black 색상의 높은 선호도를 활용하여 다른 라인업에도 적용 가능성을 검토.

- 다양한 크기 옵션 개발:

- 38, 42 외에도 고객 선호도를 반영한 크기 옵션 추가 가능성 탐색.

3. 매출 다변화 전략

- 상품군 확장:

- 특정 상품군(Mountain-200)에 의존하지 않도록 다른 라인업의 상품 매출 증대를 위한 마케팅 전략 수립.

- 타겟 고객 세분화:

- Mountain-200 시리즈의 주요 구매 고객군을 세분화하여 맞춤형 캠페인 실행.

###2. 우수 고객 분석 (재구매 유도)

✓ 2-1 상위매출액 고객 명단

Sales_data와 Customer_data를 결합하여 고객별 총 매출액을 계산. 매출 상위 10% 또는 특정 기준(예: 상위 10명)을 기준으로 명단 작성.

```
import pandas as pd

# 데이터 불러오기
file_path = '/content/drive/MyDrive/AdventureWorks Sales.xlsx'
data = pd.read_excel(file_path, sheet_name=None)

# 각 시트 데이터를 변수로 초기화
sales_data = data['Sales_data'] # Sales_data 시트
customer_data = data['Customer_data'] # Customer_data 시트

# Merge Sales_data and Customer_data
customer_sales = pd.merge(sales_data, customer_data, on='CustomerKey')

# Calculate total sales per customer
customer_performance = customer_sales.groupby('Customer')['Sales Amount'].sum().reset_index()

# Sort customers by sales amount
customer_performance = customer_performance.sort_values(by='Sales Amount', ascending=False)

# Extract top customers (e.g., top 10%)
top_10_percent_customers = customer_performance.head(int(len(customer_performance) * 0.1))

print("Top Customers by Sales:")
print(top_10_percent_customers)

# Optionally, save the results to a file
top_10_percent_customers.to_csv("Top_Customers.csv", index=False)
```

```
📄 Top Customers by Sales:
      Customer  Sales Amount
18400  [Not Applicable]  8.045060e+07
9632    Jordan Turner    1.599910e+04
18135    Willie Xu      1.349006e+04
14058    Nichole Nara    1.329538e+04
10156  Kaitlyn Henderson  1.329427e+04
...         ...
7652    Isaac Brooks     4.829390e+03
6776  Gabrielle Peterson  4.829390e+03
9172    Jill Martinez    4.829390e+03
2575    Brianna Martin    4.829390e+03
493    Alexander Walker   4.828310e+03

[1840 rows x 2 columns]
Top Customers by Sales:
      Customer  Sales Amount
18400  [Not Applicable]  8.045060e+07
9632    Jordan Turner    1.599910e+04
18135    Willie Xu      1.349006e+04
14058    Nichole Nara    1.329538e+04
10156  Kaitlyn Henderson  1.329427e+04
...         ...
7652    Isaac Brooks     4.829390e+03
6776  Gabrielle Peterson  4.829390e+03
9172    Jill Martinez    4.829390e+03
2575    Brianna Martin    4.829390e+03
493    Alexander Walker   4.828310e+03

[1840 rows x 2 columns]
```

✓ 고객별 매출 분석 결과,

1. [Not Applicable] 항목이 약 8,045만 달러로 가장 높은 매출

->

고객 정보가 누락된 거래들의 합계로 보이며, 데이터 품질 개선이 필요한 부분.

2. 개별 고객 중에서는 Jordan Turner가 약 16,000달러로 최고 매출

3. Willie Xu(약 13,490달러), Nichole Nara(약 13,295달러), Kaitlyn Henderson(약 13,294달러)가 상위권

주요 개선점:

1. 미분류([Not Applicable]) 거래의 고객 정보 보완 필요 2. 상위 고객 대상 VIP 프로그램 검토 *3. 하위 고객 대상 매출 증대 전략 수립 필요 *

###3. 우수 지역 매장 분석

✓ 3-1) 우수 지역 식별

Sales_data와 Sales Territory_data를 결합하여 지역별 매출을 계산. 매출 상위 지역을 도출하고, 지역별 매출 기여도를 분석.

```
# Load necessary sheets
sales_data = pd.read_excel(file_path, sheet_name='Sales_data')
sales_territory_data = pd.read_excel(file_path, sheet_name='Sales Territory_data')

# Merge Sales_data with SalesTerritory_data
merged_data = pd.merge(sales_data, sales_territory_data, on='SalesTerritoryKey')

# Calculate sales by region and country
region_sales = merged_data.groupby(['Region', 'Country'])['Sales Amount'].sum().reset_index()

# Sort to find top-performing regions
top_regions = region_sales.sort_values(by='Sales Amount', ascending=False)
print(top_regions)

import matplotlib.pyplot as plt

# Bar chart of top regions by sales
plt.figure(figsize=(10, 6))
plt.bar(top_regions['Region'][:10], top_regions['Sales Amount'][:10])
plt.title('Top Regions by Sales')
plt.xlabel('Region')
plt.ylabel('Sales Amount')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

✓ 지역별 매출 분석 결과:

1. 전 세계 지역 중 미국 Southwest 지역이 약 2,418만 달러로 가장 높은 매출을 기록
2. 캐나다가 약 1,636만 달러로 2위
3. 미국 Northwest 지역이 약 1,608만 달러로 3위
4. 호주는 약 1,066만 달러로 4위,
미국의 Central과 Southeast 지역은 각각 약 791만 달러와 788만 달러로 5, 6위
영국(약 767만 달러), 프랑스(약 725만 달러), 미국 Northeast 지역(약 694만 달러)
5. 독일이 약 488만 달러로 가장 낮은 매출.

특징:

1. 미국 내에서는 Southwest 지역이 가장 높은 성과, Northeast 지역이 가장 낮은 성과
2. 북미 지역(미국, 캐나다)이 전반적으로 높은 매출
3. 유럽 국가들(영국, 프랑스, 독일)은 상대적으로 낮은 매출

주요 개선점:

**** 이러한 결과를 토대로 Southwest 지역의 성공 요인을 분석하고, 상대적으로 실적이 저조한 지역의 개선 방안을 검토할 필요가 있습니다. ****

✓ ###4. 우수 지역의 고객 및 매장 분석

우수 지역에서 매출에 기여한 고객 또는 리셀러를 분석하여 추가적인 인사이트를 도출할 수 있습니다.

```
# Filter sales data for top regions (e.g., top 3 regions)
top_regions = region_sales['Region'][:3]
top_region_data = region_sales_data[region_sales_data['Region'].isin(top_regions)]

# Group by reseller and customer within top regions
top_region_customers = top_region_data.groupby('CustomerKey')['Sales Amount'].sum().reset_index()
top_region_resellers = top_region_data.groupby('ResellerKey')['Sales Amount'].sum().reset_index()

print("Top Customers in Top Regions:")
print(top_region_customers.head())

print("Top Resellers in Top Regions:")
print(top_region_resellers.head())
```

✓ 상위 지역 고객/리셀러 매출 분석 결과

1. 상위 지역 고객 매출 분석
 - 미분류 고객(-1)이 약 4,528만 달러로 압도적 1위
 - 일반 고객(11012-11015)은 81달러~2,501달러로 매우 낮은 수준
2. 상위 지역 리셀러 매출 분석
 - 미분류 리셀러(-1)가 약 1,135만 달러로 최고 매출
 - 리셀러 4번이 약 587천 달러, 3번이 약 362천 달러로 상위권
 - 나머지 리셀러들은 8.5만 달러 이하의 매출 기록

주요 개선점:

1. 고객/리셀러 미분류(-1) 데이터의 정확한 식별 필요
 2. 일반 고객 대상 매출 증대 전략 시급
 3. 상위 리셀러(3번, 4번) 성공 요인 분석 및 타 리셀러 적용 검토
-
-

###5. 데이터 *이탈릭체* 텍스트 특성도출 (Feature Engineering) 단계

RFM(Recency, Frequency, Monetary)은 마케팅에서 고객의 가치를 분석하고 세분화하는 데 널리 사용되는 분석 기법

각 지표는 고객의 행동을 평가하기 위한 주요 요소를 다루고 있으며, 이를 통해 효과적인 타겟 마케팅을 수행

✓ 5-1. Recency (R) 계산(고객이 마지막으로 구매한 시점으로부터 현재까지의 시간)

고객의 가장 최근 구매 날짜와 기준일(분석 시점)을 비교하여 계산합니다..

```
# Load Date_data sheet
date_data = pd.read_excel(file_path, sheet_name='Date_data')

# 확인: date_data 컬럼 출력
print(date_data.columns)

Index(['DateKey', 'Date', 'Fiscal Year', 'Fiscal Quarter', 'Month',
      'Full Date', 'MonthKey'],
      dtype='object')

from datetime import datetime

# Check the 'Full Date' column format
print(sales_with_dates['Full Date'].head())

# Convert 'Full Date' to datetime format
sales_with_dates['Full Date'] = pd.to_datetime(sales_with_dates['Full Date'], errors='coerce', format='%Y %b, %d')

# 기준일 설정: 가장 최신 날짜
reference_date = sales_with_dates['Full Date'].max()

# Recency 계산
recency = sales_with_dates.groupby('CustomerKey')['Full Date'].max().reset_index()
recency['Recency'] = (reference_date - recency['Full Date']).dt.days

# 결과 확인
print(recency.head())
```

```
0    2017 Jul, 02
1    2017 Jul, 02
2    2017 Jul, 02
3    2017 Jul, 02
4    2017 Jul, 02
Name: Full Date, dtype: object
CustomerKey  Full Date  Recency
0           -1  2020-06-15      0
1          11000  2019-10-04    255
2          11001  2020-05-12     34
3          11002  2019-07-27   324
4          11003  2019-10-11   248
```

✓ Recency(최근성) 분석 결과

1. 데이터 기간 분석

- 기준 날짜: 2017년 7월 2일
- 거래 기간: 2019년 7월 ~ 2020년 6월

2. 고객별 최근 구매 현황

- 가장 최근 구매 고객: CustomerKey(-1), 2020년 6월 15일(Recency: 0일)
- 가장 오래된 구매 고객: CustomerKey(11002), 2019년 7월 27일(Recency: 324일)
- 중간 범위 구매 고객들: 200~300일 사이에 분포

3. Recency 분포 특징

- 0일(매우 최근) ~ 324일(약 11개월)까지 넓게 분포
- CustomerKey 11001은 34일로 비교적 최근 구매

주요 개선점:

1. CustomerKey(-1) 데이터의 정확한 고객 식별 필요
 2. Recency 300일 이상 고객 대상 재구매 유도 전략 수립
 3. 최근 구매 고객(Recency 낮은 고객) 대상 충성도 프로그램 검토
-

✓ 5-2. Frequency (F) 계산(특정 기간 동안 고객이 얼마나 자주 구매했는지)

고객의 가장 최근 구매 날짜와 기준일(분석 시점)을 비교하여 계산합니다.

```
# Frequency: 특정 기간 동안 고객별 구매 횟수 계산
frequency = sales_data.groupby('CustomerKey')['SalesOrderLineKey'].count().reset_index()
frequency.columns = ['CustomerKey', 'Frequency']
```

```
# 모든 고객의 데이터 출력
print(frequency)
```

```
↕
   CustomerKey  Frequency
0           -1      60855
1          11000           8
2          11001          11
3          11002           4
4          11003           9
...          ...         ...
18480        29479           1
18481        29480           5
18482        29481           1
18483        29482           1
18484        29483           1
```

[18485 rows x 2 columns]

✓ Frequency (고객 구매 빈도) 분석 결과

1. 전체 고객 규모

- 총 18,485명의 고객 데이터 분석
- CustomerKey(-1)의 구매 횟수가 60,855회로 압도적 1위
- > 시스템상의 테스트 데이터나 특별 계정으로 추정

2. 일반 고객 구매 패턴

- 대다수 고객이 1~11회 구매에 집중
- 전반적으로 낮은 구매 빈도 경향
- 약 10%만이 다회 구매 고객으로 확인

3. 고객 세그먼트 특성

- 단회성 구매 고객이 대다수
- 소수의 충성 고객이 다회 구매 기록
- 재구매율이 전반적으로 낮은 상황

주요 개선점:

1. CustomerKey(-1) 데이터의 정확한 식별 및 검증 필요
2. 단회 구매 고객 대상 재구매 유도 전략 수립
3. 다회 구매 고객 대상 VIP 마케팅 프로그램 개발
4. 전반적인 고객 충성도 제고 방안 검토

✓ 5-3. Monetary (M) 계산(고객의 총 구매 금액을 계산합니다.)

CustomerKey와 Sales Amount를 활용하여\ 계산합니다.


```
# Calculate Monetary: Total sales amount per customer
monetary = sales_with_dates.groupby('CustomerKey')['Sales Amount'].sum().reset_index()
monetary.columns = ['CustomerKey', 'Monetary']

print(monetary)
```

```
↩
CustomerKey    Monetary
0             -1  8.045060e+07
1            11000  8.248990e+03
2            11001  6.383880e+03
3            11002  8.114040e+03
4            11003  8.139290e+03
...          ...    ...
18480         29479  2.049098e+03
18481         29480  2.442030e+03
18482         29481  3.374990e+03
18483         29482  2.049098e+03
18484         29483  2.049098e+03
```

[18485 rows x 2 columns]

✓ Monetary(고객 구매 총액) 분석 결과

1. 전체 고객 규모 및 특이점

- 총 18,485명의 고객 데이터 분석
- CustomerKey(-1)의 구매 총액이 80,450,600원으로 비정상적으로 높음
- > 시스템 테스트 데이터나 통합 계정으로 추정되어 검증 필요

2. 일반 고객 구매 금액 분포

- 대다수 고객: 2,000원~10,000원 범위
- 상위 고객군(11000 ~ 11003): 6,000원 ~ 8,000원 수준
- 하위 고객군(29479~29483): 2,000원 수준

3. 고객 세그먼트 특성

- 소액 구매 고객이 다수
- 상위 10% 고객이 높은 구매력 보유
- 전반적으로 낮은 객단가 경향

주요 개선점:

1. 우수 고객 관리 방안(EX)

- 상위 10% 고객 대상 프리미엄 서비스 도입
- 맞춤형 할인 및 감사 선물 프로그램 검토

2. 마케팅 전략 세분화(EX)

- 구매 금액 기준 고객 등급제 도입
- 등급별 차별화된 마케팅 전략 수립

3. 매출 증대 방안(EX)

- 하위 고객 대상 할인 쿠폰 제공
- 추천 상품 캠페인 실시
- CustomerKey(-1) 데이터 정확성 검증

###6. Recency, Frequency, Monetary 데이터를 결합하여 RFM 테이블 생성

✓ 6-1 RFM 테이블 생성

```
# RFM 테이블 생성
rfm_table = pd.merge(recency[['CustomerKey', 'Recency']], frequency, on='CustomerKey')
rfm_table = pd.merge(rfm_table, monetary[['CustomerKey', 'Monetary']], on='CustomerKey')
```

```
# RFM 데이터 확인
print(rfm_table)
```

```
↩ CustomerKey Recency Frequency Monetary
0          -1         0      60855  8.045060e+07
1         11000        255         8  8.248990e+03
2         11001         34        11  6.383880e+03
3         11002        324         4  8.114040e+03
4         11003        248         9  8.139290e+03
...      ...      ...      ...      ...
18480      29479        496         1  2.049098e+03
18481      29480        180         5  2.442030e+03
18482      29481        884         1  3.374990e+03
18483      29482        482         1  2.049098e+03
18484      29483        491         1  2.049098e+03

[18485 rows x 4 columns]
```

✓ 6-2 각 지표에 대해 점수를 부여하여 고객 세그먼트화

1~5 스케일로 Recency, Frequency, Monetary를 점수화합니다.

```
# Frequency의 최소값과 최대값 확인
print(rfm_table['Frequency'].min(), rfm_table['Frequency'].max())

# Frequency 점수화
bins = [rfm_table['Frequency'].min()-1, 1, 10, 50, 100, rfm_table['Frequency'].max()]
labels = [1, 2, 3, 4, 5]

rfm_table['F_Score'] = pd.cut(rfm_table['Frequency'], bins=bins, labels=labels)

# Recency와 Monetary도 동일하게 처리
rfm_table['R_Score'] = pd.qcut(rfm_table['Recency'], 5, labels=[5, 4, 3, 2, 1])
rfm_table['M_Score'] = pd.qcut(rfm_table['Monetary'], 5, labels=[1, 2, 3, 4, 5])

# 확인
print(rfm_table[['R_Score', 'F_Score', 'M_Score']])
```

```
↩ 1 60855
   R_Score F_Score M_Score
0         5         5         5
1         2         2         5
2         5         3         5
3         1         2         5
4         2         2         5
...      ...      ...      ...
18480      1         1         4
18481      3         2         4
18482      1         1         5
18483      1         1         4
18484      1         1         4

[18485 rows x 3 columns]
```

✓ RFM 고객 세그먼트 분석 결과

1. 최우수 고객군 분석

- 최고점(5,5,5) 달성 고객: 1명

- Recency, Frequency, Monetary 모든 측면에서 우수
- > 핵심 고객으로서 특별 관리 필요

2. 일반 고객군 특성

- 대다수 고객이 낮은 F, M 점수 기록
- 구매 빈도(F)와 구매 금액(M) 모두 저조
- 충성 고객층이 매우 부족한 상황

3. 세그먼트 분포 특징

- 고점수대 고객 비중이 극히 낮음
- 저점수대 고객이 다수를 차지
- 고객 가치 양극화 현상 발생

주요 개선점:

1. 최우수 고객 유지 전략

- VIP 혜택 프로그램 도입
- 맞춤형 서비스 제공

2. 일반 고객 충성도 제고

- 구매 빈도 증대 프로모션
- 객단가 상승 유도 전략

3. 세그먼트별 차별화 전략

- 점수대별 맞춤 마케팅 실시
- 등급 상승 동기부여 프로그램 개발

✓ 6-3 RFM 점수를 바탕으로 고객 세그먼트 생성

고객 세그먼트를 정의하여 각 고객을 분류합니다.

```
# 고객 세그먼트 정의 함수
def rfm_segment(row):
    if row['R_Score'] == 5 and row['F_Score'] == 5 and row['M_Score'] == 5:
        return 'Best Customers'
    elif row['R_Score'] <= 2:
        return 'At Risk'
    elif row['F_Score'] >= 4:
        return 'Frequent Buyers'
    elif row['M_Score'] >= 4:
        return 'Big Spenders'
    else:
        return 'Others'

# 고객 세그먼트 생성
rfm_table['Segment'] = rfm_table.apply(rfm_segment, axis=1)

# 세그먼트별 고객 수 확인
print(rfm_table['Segment'].value_counts())
```

```
Segment
At Risk          7392
Others           6767
Big Spenders     4311
Frequent Buyers    14
Best Customers     1
Name: count, dtype: int64
```

✓ RFM 고객 세그먼트 분석 결과

1. 최우수 고객군 분석

- 최고점(5,5,5) 달성 고객: 1명
 - Recency, Frequency, Monetary 모든 측면에서 우수
 - → 핵심 고객으로서 특별 관리 필요
-

2. 일반 고객군 특성

- 대다수 고객이 낮은 F, M 점수 기록
 - 구매 빈도(F)와 구매 금액(M) 모두 저조
 - 충성 고객층이 매우 부족한 상황
-

3. 세그먼트 분포 특징

- 고점수대 고객 비중이 극히 낮음
 - 저점수대 고객이 다수를 차지
 - 고객 가치 양극화 현상 발생
-

주요 개선점

1. 최우수 고객 유지 전략

- VIP 혜택 프로그램 도입
 - 맞춤형 서비스 제공
-

2. 일반 고객 충성도 제고

- 구매 빈도 증대 프로모션
 - 객단가 상승 유도 전략
-

3. 세그먼트별 차별화 전략

- 점수대별 맞춤 마케팅 실시
 - 등급 상승 동기부여 프로그램 개발
-

고객 세그먼트

RFM 점수를 바탕으로 세그먼트를 생성한 결과:

- **At Risk (7392명, 40%):**
 - 이 그룹은 Recency 점수가 낮아 오랜 기간 구매하지 않은 고객군입니다.
 - 재구매를 유도하기 위해 리마케팅 캠페인과 맞춤형 혜택이 필요합니다.
- **Others (6767명, 37%):**
 - 특별히 두드러지지 않는 평균적인 고객군입니다.
 - 이 그룹을 고가 구매 또는 충성 고객으로 전환하기 위한 전략이 필요합니다.
- **Big Spenders (4311명, 23%):**
 - Monetary 점수가 높은 고객군으로, 이들을 대상으로 프리미엄 서비스를 제공하거나 로열티 프로그램을 강화할 수 있습니다.
- **Frequent Buyers (14명, <1%):**
 - Frequency 점수가 높은 고객군으로, 이들은 자주 구매하지만 구매 금액은 낮을 가능성이 있습니다.
 - 고가 상품 추천 및 교차 판매 전략이 효과적일 수 있습니다.
- **Best Customers (1명, <1%):**
 - R, F, M 점수가 모두 높은 최우수 고객입니다.
 - 이 고객을 유지하기 위해 특별한 혜택 제공이 필요합니다.

결론 및 제안

1. At Risk 고객군 공략

- 이탈 가능성이 높은 고객군에 리마케팅 캠페인(예: 할인 혜택, 맞춤형 메시지)을 통해 재구매를 유도.
-

2. Big Spenders와 Frequent Buyers 강화

- 구매력이 높은 고객(Big Spenders)을 대상으로 프리미엄 서비스, 감사 선물 제공.
 - 자주 구매하지만 금액이 낮은 고객(Frequent Buyers)에게 고가 상품 추천 및 할인 제안.
-

3. Best Customers 유지

- 가장 가치 있는 고객(Best Customers)에게 지속적으로 높은 만족을 제공하여 충성도를 유지.
-

4. 데이터 검증

- $CustomerKey = -1$ 과 같은 비정상 데이터는 추가 검토 후 제거하거나 별도로 분석.
-

5. 추가 분석 제안

- 각 세그먼트별로 구매 패턴, 지역, 상품 선호도를 분석하여 세분화된 마케팅 전략을 설계.
-
-