K-Digital Training iM DiGital Banker Academy 5기

배송 불만과 캠페인 반응 분석을 통한 인사이트

박민정 이우태 전은진 정혜은 한민정 | 1조 으랏차차 로켓배송

목차

기획배경

문제상황 및 목표

요구사항 1 | 구매 패턴 파악

베이스 전처리 | RFM 분석 | 마케팅 전략 제안

요구사항 2 | 고객 이탈 방지와 전략

모델링 전략 | 모델 성능 평가 | 예측 결과 분석 | 관리 방안 제안

요구사항 4 | 맘큐의 현황 진단

요구사항 2과 연계

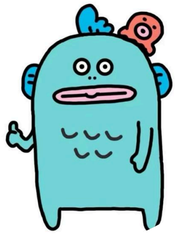
요구사항 3 | 제품의 적기 배송

배송일 예측

팀 구성

박민정

데이터 EDA 및 모델링



이우태

데이터 EDA 및 모델링

전은진

데이터 EDA 및 모델링

정혜은

데이터 EDA 및 모델링

한민정

데이터 EDA 및 모델링

일정

06/16
- 브레인스토밍

06/17
- 주제 선정
- 기획서 작성

06/18
- 시나리오 분석
- 데이터 전처리

06/19
- 데이터 전처리 & EDA
- 외부 데이터 수집

06/20~21
- 외부 데이터 수집
- 데이터 모델링

06/22~24
- 데이터 모델링
- 하이퍼파라미터 튜닝 진행
- 슬라이드 제작

06/25
- 인사이트 도출
- 모델 성능 지표 도출

06/26
- 발표

1. 기획배경

문제 상황 및 목표

K사는 맘큐이다

- **맘큐 | 유한킴벌리 직영 쇼핑몰** [PDF] 2020 유한킴벌리 지속가능성보고서

  아동/유아 제품을 판매하고 있는 자사 K사는
  매월 30만명 이상 방문하는 생활용품 전문 쇼핑몰을 운영하고 있고,
  최근에 가입한 **누적 회원이 50만명 달성**하며,
  유아 관련 제품 판매 업계에 판매량 1위를 유지하고 있었다

- **제공받은 데이터 | 2022-01-01 ~ 2023-08-08**

주어진 데이터는 샘플링된 데이터일 것이다

- 유한킴벌리는 2022년 한 해 동안 약 1조 5091억 원의 매출을 기록
- 하지만, 주어진 데이터에서는 결제금액 기준, 2022년 한 해 매출 31억 원 수준
- 이는 주어진 데이터가 샘플링된 데이터로 추정됨

손 익 계 산 서

제 53 기 2022년 1월 1일부터 2022년 12월 31일까지

제 52 기 2021년 1월 1일부터 2021년 12월 31일까지

유한킴벌리주식회사 (단위 : 원)

과 목	제 53(당) 기		제 52(전) 기	
Ⅰ.매출액(주석2,13)		1,509,163,069,683		1,467,106,564,380

출처 : 다트에 공시된 유한킴벌리 감사보고서

맘큐 온라인몰의 경쟁력

AI 챗봇을 통해 아기 피부 특성, 월령 등 개별 정보를 기반으로 최적의 기저귀를 제안함으로써
➔ 단순 제품 판매를 넘어 개인의 육아 맞춤 편의성 제공 중

어떤 서비스를 제공해줄 수 있어?

육아 관련 서비스로는 다양한 지원이 있습니다. 예를 들어, 육아종합지원센터에서는 영·유아 발달에 적합한 놀잇감 및 도서 대여 서비스, 놀이 체험실과 영·유아 놀이 프로그램, 부모 교육 및 전문 상담 등을 제공합니다. 또한, 시간제 보육 서비스도 제공하여 6~36개월 미만 영아를 대상으로 보육료를 지불하고 이용할 수 있습니다. 이 외에도 부모 교육, 온라인 및 대면 상담 서비스 등이 제공됩니다 [1].

또한, 임신부를 위한 건강보조식품 지원, 출산 교실 및 관련 용품 지원, 출산 가정 방문 산후 관리 서비스, 영·유아 건강검진 및 국가 예방접종 지원 등 다양한 서비스가 제공됩니다 [2].

더 알아보기

[1] 알토란 같은 육아 지원기관

[2] 알토란 같은 육아 지원기관

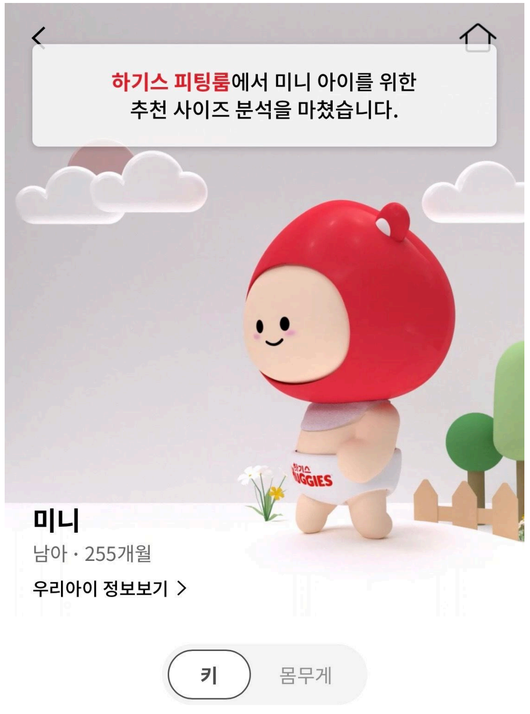
지금 내 아기 정보에 맞춰 기저귀 추천 가능?

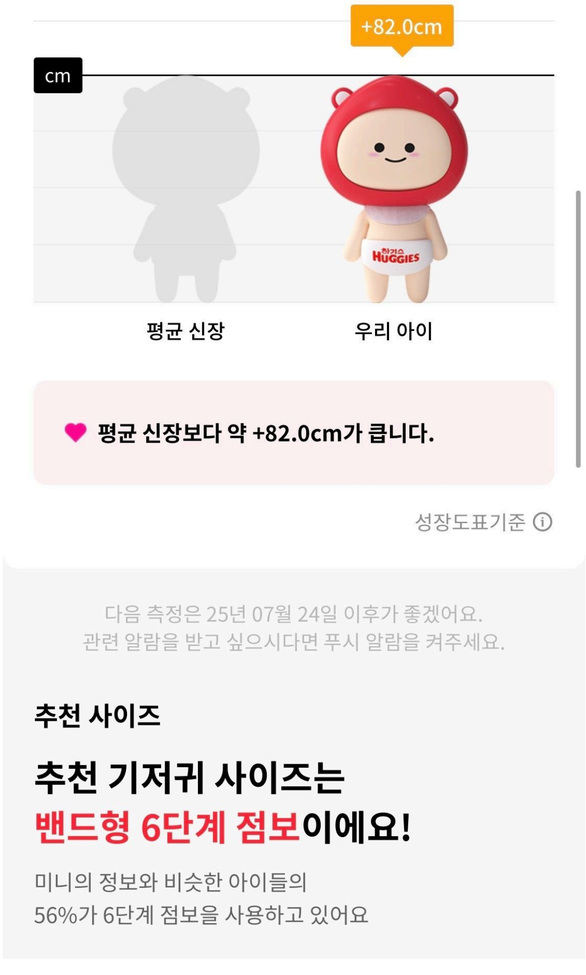
아기의 피부가 민감하다면, 하기스 네이처메이드 오가닉 코튼 커버 기저귀를 추천드립니다. 이 기저귀는 아기 피부를 보호할 수 있는 성분으로 이루어져 있으며, 통기성을 높이는 구조로 만들어져 있어 민감한 아기에게 적합합니다 [1]. 추가로, 기저귀는 위생용품으로 유통기한이 별도로 정해져 있지 않지만, 유한킴벌리는 제조일자로부터 약 3년 이내에 사용하시기를 권장합니다 [2].

출처 : 맘큐 모바일 어플 챗봇과의 대화

맘큐 온라인몰의 경쟁력

아기 정보 입력에 맞추어 또래와 발달 정도를 비교해주고 기저귀 사이즈 추천까지 해주는 추천시스템이 잘 구축되어 있음
→ 현재 맘큐가 가진 문제와 앞으로의 인사이트를 제시하는 데 집중





출처 : 맘큐 모바일 어플

2. 구매 패턴 파악

목적 | RFM 분석 기반으로 고객 행동 이해 및 세분화

데이터 전처리

Member Data

1. 고객 번호는 있고 나머지 열에 대한 값이 없는 행들을 결측으로 판단해 삭제

고객번호	0
성별	266
결혼유무	266
거주지역	266
연령	266
결제등록카드	266
유입경로	266
자녀여부	266

2. 결제 등록 카드

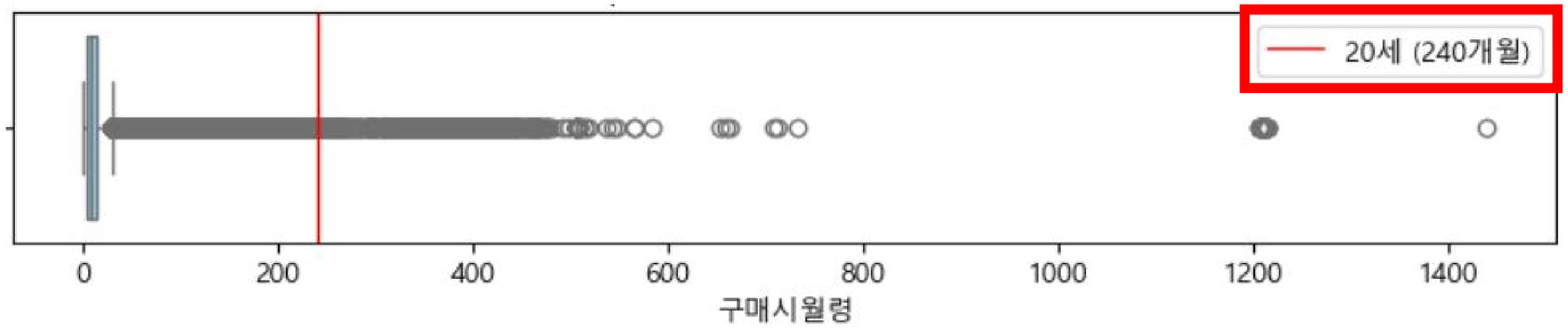
- 국민카드 → <u>국민은행</u>, 신한카드 → <u>신한은행</u>, 외환은행 → <u>하나은행</u>으로 대체

```
df1['결제등록카드'] = df1['결제등록카드'].replace('국민카드', '국민은행').replace('신한카드', '신한은행').replace('외환은행', '하나은행')
```

```
array(['롯데카드', '광주은행', '신협중앙회', '신한은행', '하나은행', '국민은행', '농협중앙회', '우리은행',
       '농협회원조합', '새마을금고', 'BC카드', '현대카드', '기업은행', '대구은행', 'SC제일은행',
       '삼성카드', '부산은행', '우체국', '미래에셋증권', '경남은행', '씨티은행'], dtype=object)
```

Sales Data

1. 구매시월령 결측치 및 이상치 처리



- 자녀월령이 음수인 경우 ⇒ '구매시월령_bin' 파생컬럼 내 '태아' 로 저장

- 자녀월령이 240 이상이거나 결측치인 경우
  ⇒ 기저귀 구매 이력이 있는 고객을 다음 기준으로 자녀월령을 대체하였음

Sales Data

- **자녀월령 대체 기준**
  ⇒ 기저귀 단계에 따라 평균값으로 채워넣음

	1단계 신생아용	2단계 소형	3단계 중형	4단계 대형	5단계 특대형	6단계 점보	7단계 특점보
	태어난지 한달 무렵까지	한달 무렵부터	뒤집고 기는 아기	걷기 시작하는 아기	아장아장 걷는 아기	잘 뛰고 활동적인 아기	성장이 빠르고 활동적인 아기
밴드형	3~4.5kg	4~8kg	7~10kg	9~13kg	11~16kg		
팬티형	뒤집기 시작하면 입히세요		7~10kg	9~13kg	11~16kg	14~19kg	17~24kg

출처 : 맘큐 하기스 기저귀 설명 중 상세페이지

기저귀 단계	대체 월령(개월)
태어난지 한달 무렵까지	5
한달 무렵부터	5
뒤집고 기는 아기	8
걷기 시작하는 아기	13
아장아장 걷는 아기	20
잘 뛰고 활동적인 아기	24
성장이 빠르고 활동적인 아기	29

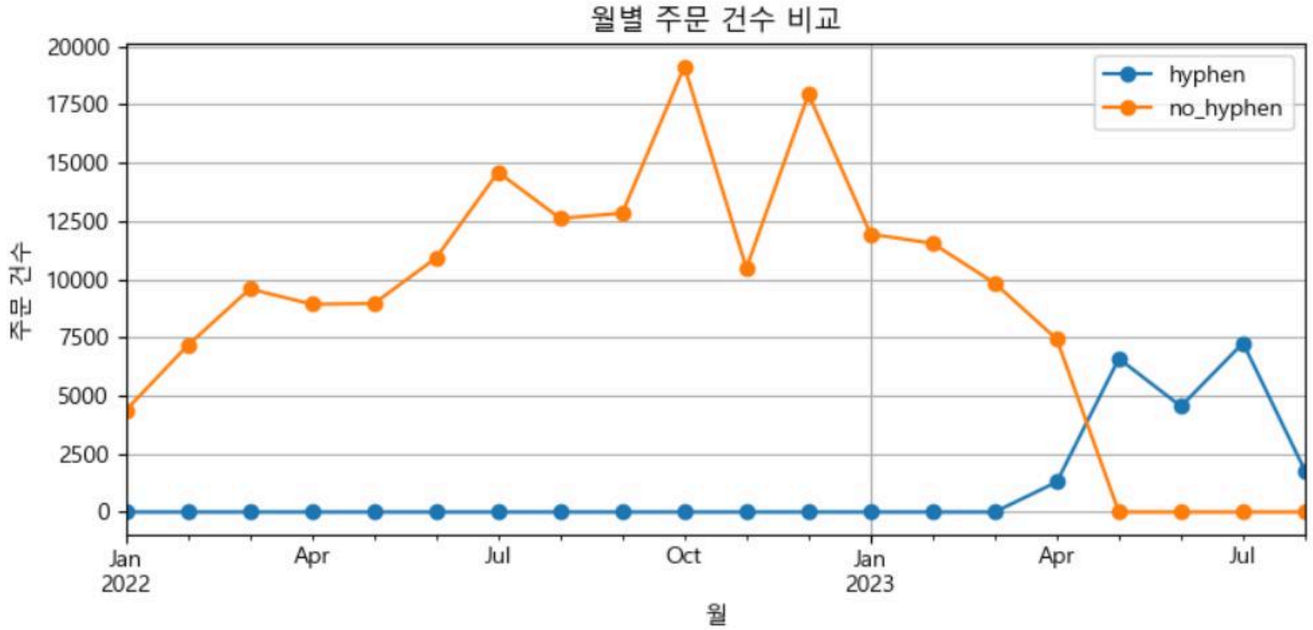
Sales Data

2. 고객 번호 미일치 이슈

Sales Data에는 고객 번호가 있지만 Member Data에는 존재하지 않는 데이터 416건 제거

3. 하이픈이 포함된 주문 번호

주문 번호에 -이 포함된 인스턴스가 존재했음
⇒ 확인 결과, 시간이 지남에 따라 주문 번호에 하이픈이 포함된 형태로 변형됨



데이터 병합

1. Sales + Product 병합 (상품 정보 붙이기)

- 상품명 기준으로 Left Join 수행

2. Sales_Prod + Member 병합 (고객 정보 붙이기)

- 고객번호 기준으로 Left Join 수행

```python
# 1) Sales2 + Product 병합 (상품 정보 붙이기)
sales_prod = Sales2.merge(
    Product,
    on='상품명',
    how='left',                # 모든 주문 행 유지
)

# 2) sales_prod + Member 병합 (고객 정보 붙이기)
full_df = sales_prod.merge(
    Member,
    on='고객번호',
    how='left',                # 모든 주문 행 유지
)
```

병합 후 결측치

■ 결측치 개수

주문번호	0
배송완료일	0
배송시작일	0
구매일	0
구매시월령	32812
고객번호	0
구매금액	0
결제금액	0
물품대분류_x	0
상품명	0
물품대분류_y	0
물품판매량	0
제품단가	0
성별	0
결혼유무	0
거주지역	0
연령	0
결제등록카드	0
유입경로	0
자녀여부	0

물품 대분류 정리 작업

- **수작업으로 11개의 대분류로 나눔**
  → 팬티, 체험팩, 물티슈, 더블하트, 기저귀, SKC, FMC, FC, 기타, SrC, 택배파손

- **기존 대분류 대비 3.57% 의 대분류 변동이 생김** 전체 209,593개 중 7,477개가 다릅니다 (3.57%)

물품대분류_x	
기저귀	43461
더블하트	37228
체험팩	35390
팬티	29031
FMC	25950
FC	12433
SKC	9365
물티슈	8545
기타	8175
SrC	15
total	209593

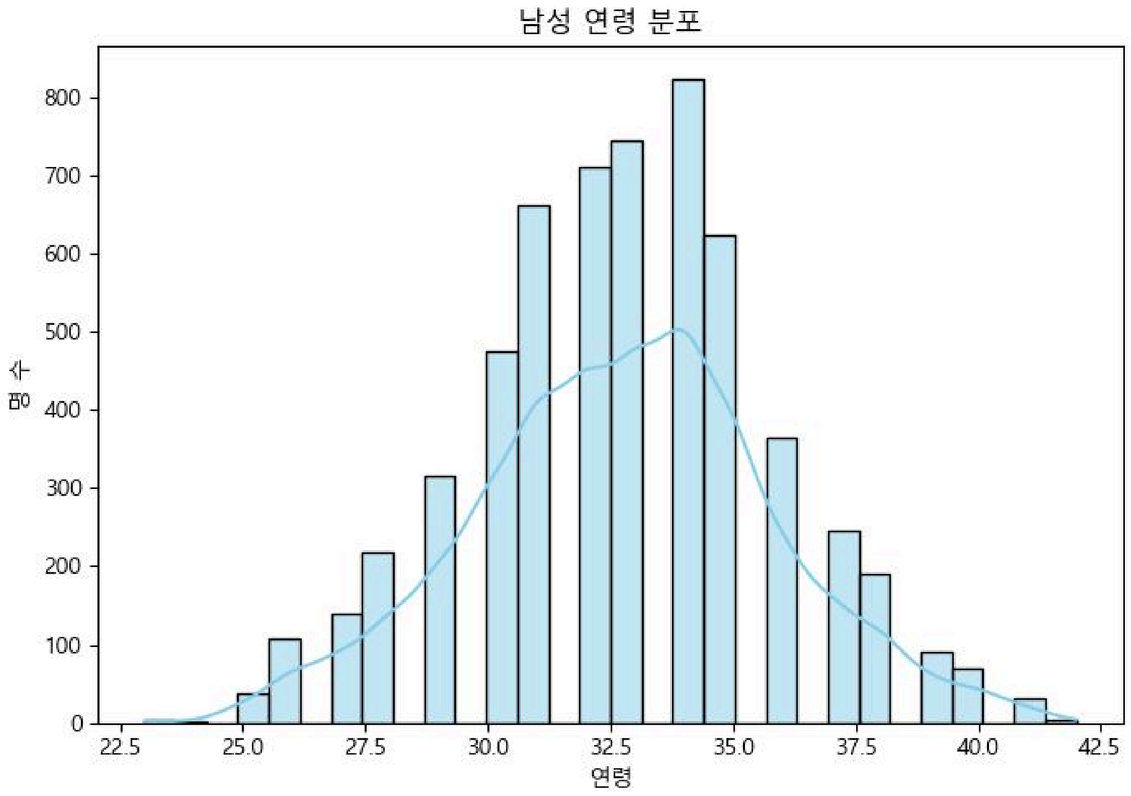
→

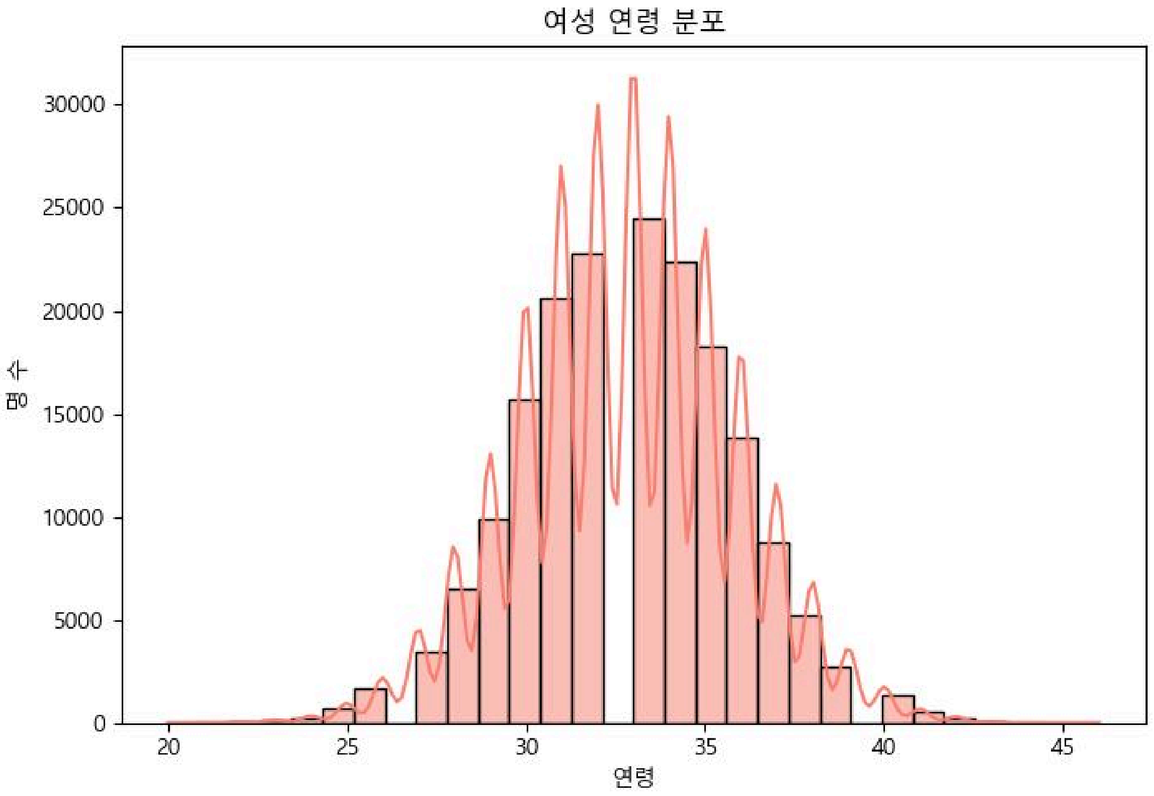
물품대분류	
기저귀	46163
더블하트	37224
체험팩	36755
팬티	25373
FMC	24953
FC	12390
SKC	10468
물티슈	9205
기타	6990
SrC	70
택배파손	2
total	209593

대분류	설명
FC	여성용품
FMC	생활용품
SKC	화장품&스킨케어
기저귀	기저귀
더블하트	브랜드
체험팩	상품 체험팩
팬티	기저귀의 팬티형

이후 데이터 정리

- 구매시월령이 결측이 아닌 행 중, 자녀여부가 No 인 경우를 ‘Yes’로 업데이트 (34,120건)
- 거주지역 : 경상도(경남, 경북), 전라도(전남, 전북), 충청도(충남, 충북) 합침
- 결제금액이 음수인 행은 제거

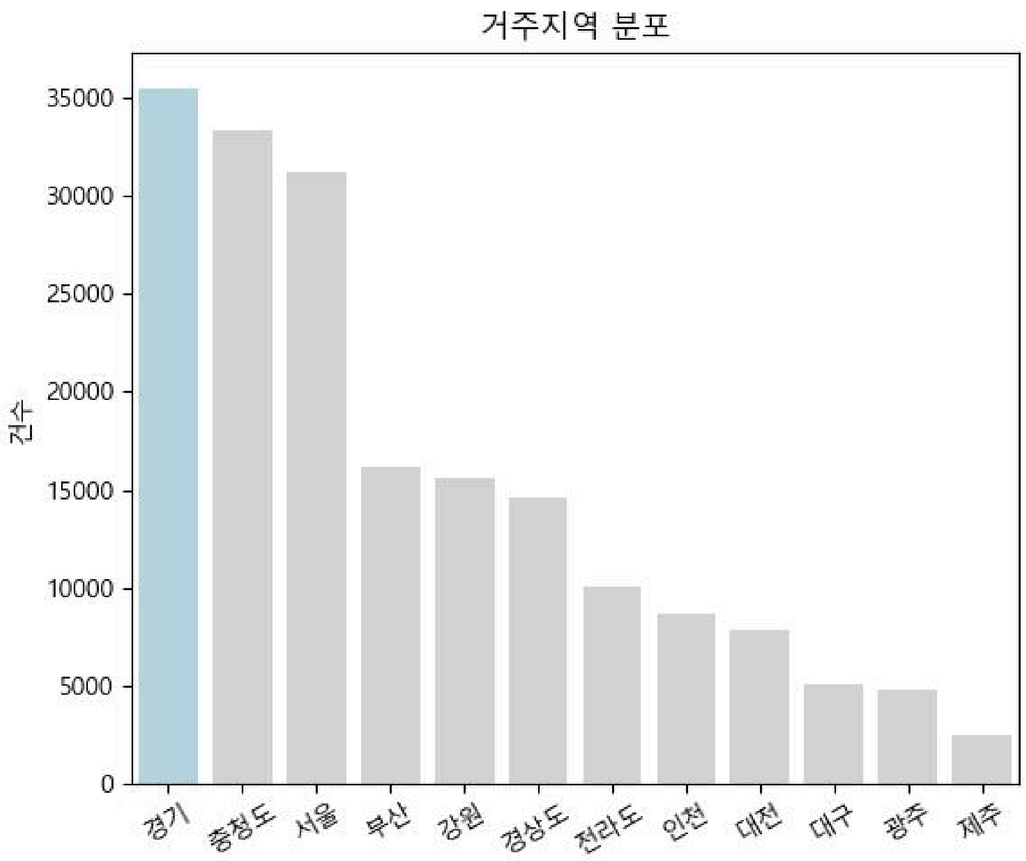




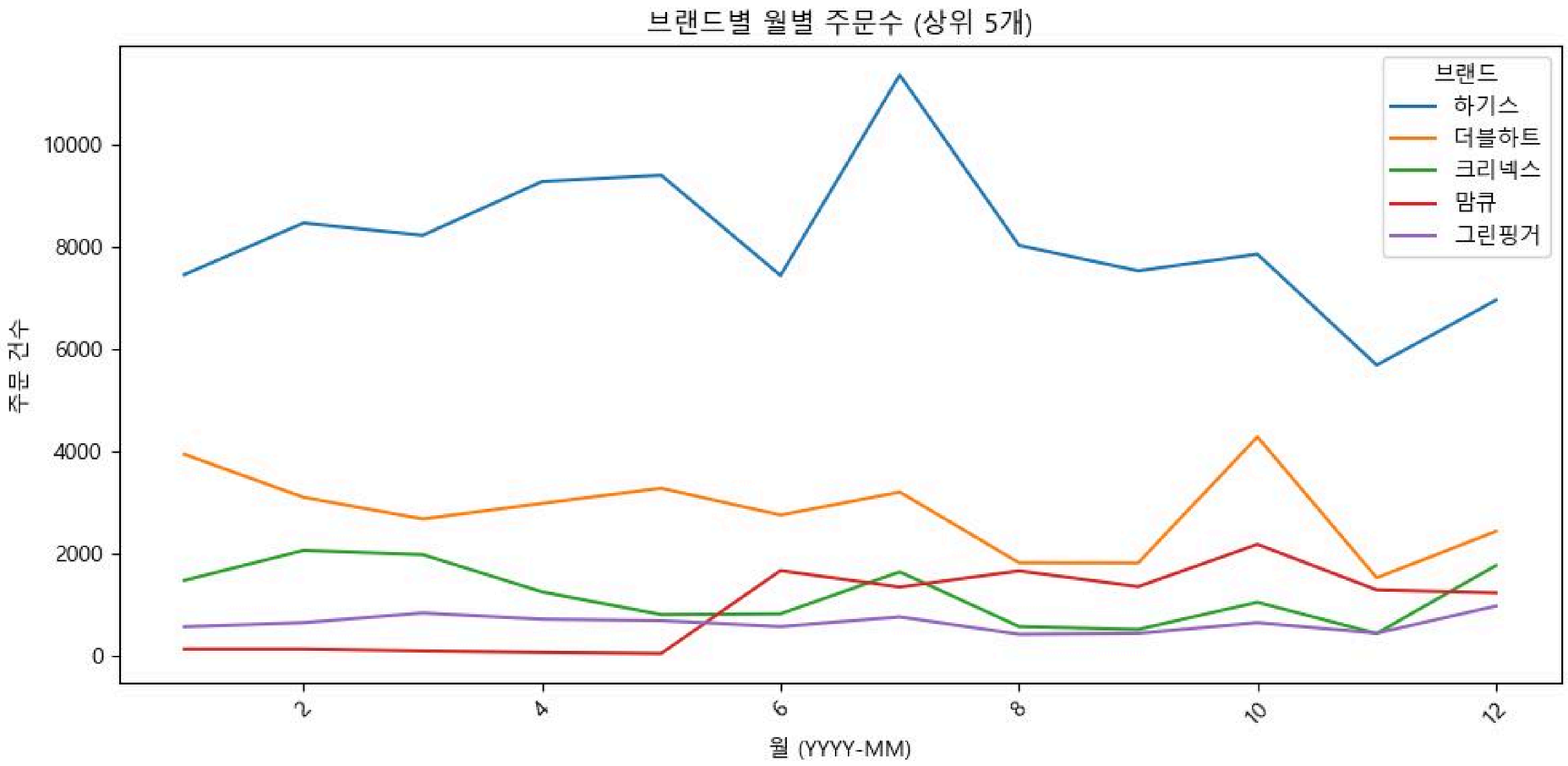

```
성별
여    0.968417
남    0.031583
```

- **전체 고객 중 여성(96%)이 절대적으로 높음**
- **여성은 30~36세 사이에 집중된 반면, 남성은 좀 더 넓은 연령 분포**

⇒ 마케팅 타겟 설정 시 여성 30대 초중반 고객군에 최우선 집중



고객의 거주지역은 경기>충청>서울>부산 순으로 의외로 서울이 가장 주문건수가 많지는 않았음

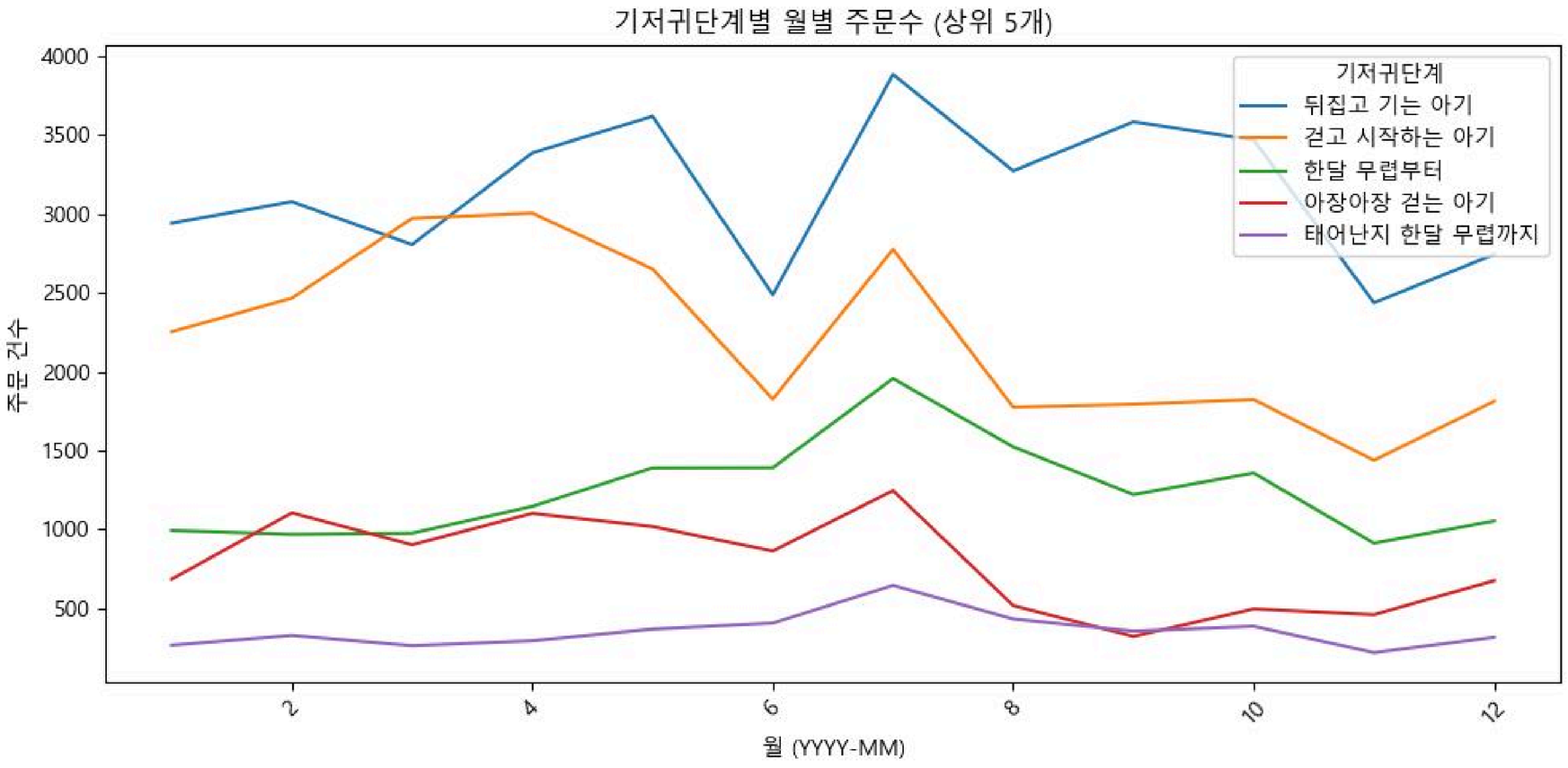


주문수가 가장 많은 상위 브랜드 5개의 월별 주문건수를 시각화한 결과,

하기스가 압도적으로 많음

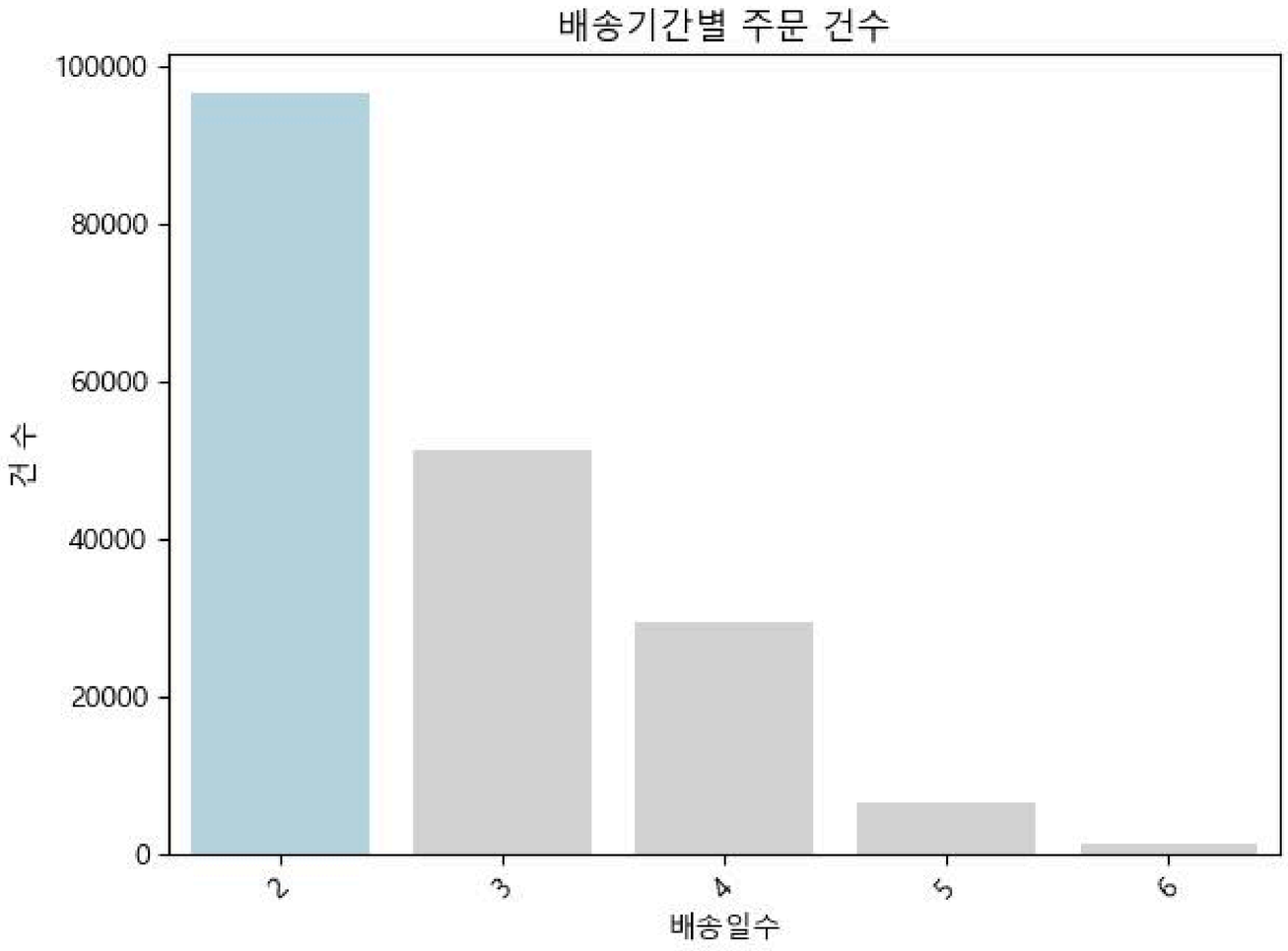
⇒ 유한킴벌리 자사몰인 맘큐의 주력상품은 하기스임을 알 수 있음

⇒ 기저귀 주문수는 7월에 피크를 찍었는데 여름철이라 기저귀를 자주 갈아주는 것으로 보임



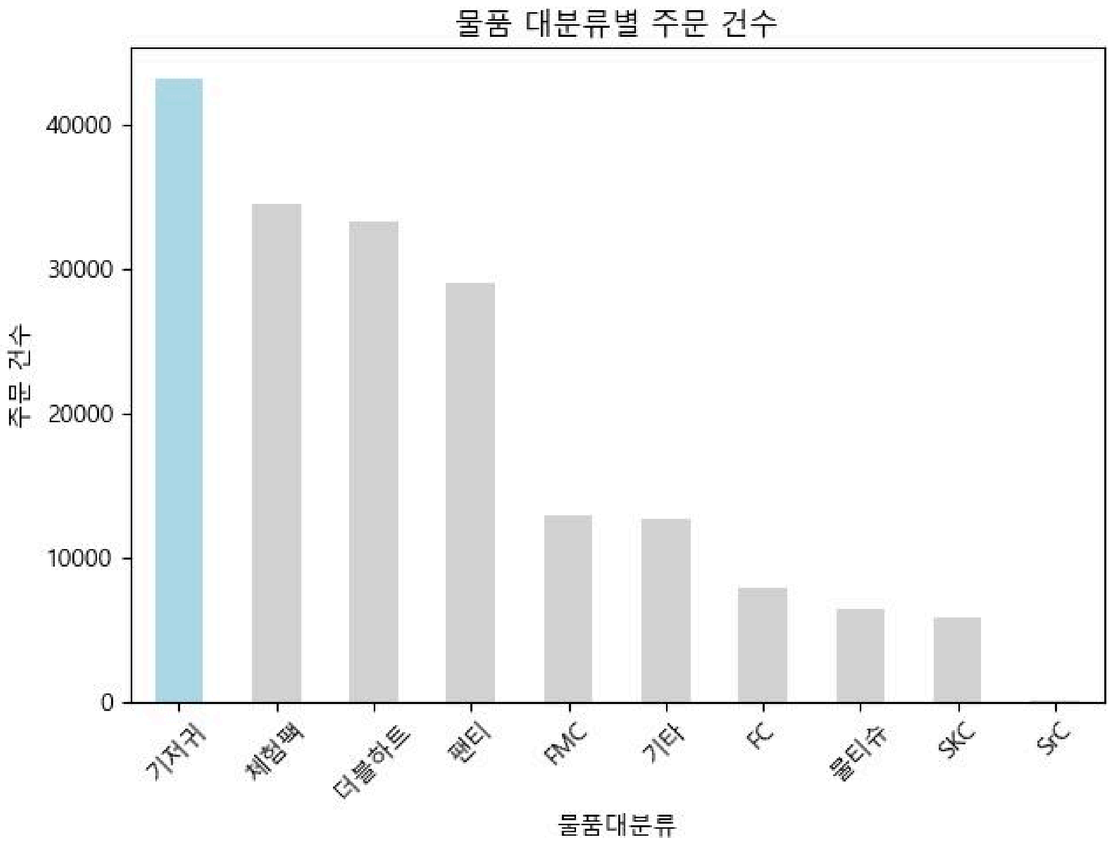
기저귀는 '뒤집고 기는 아기' 기저귀 주문이 가장 많음

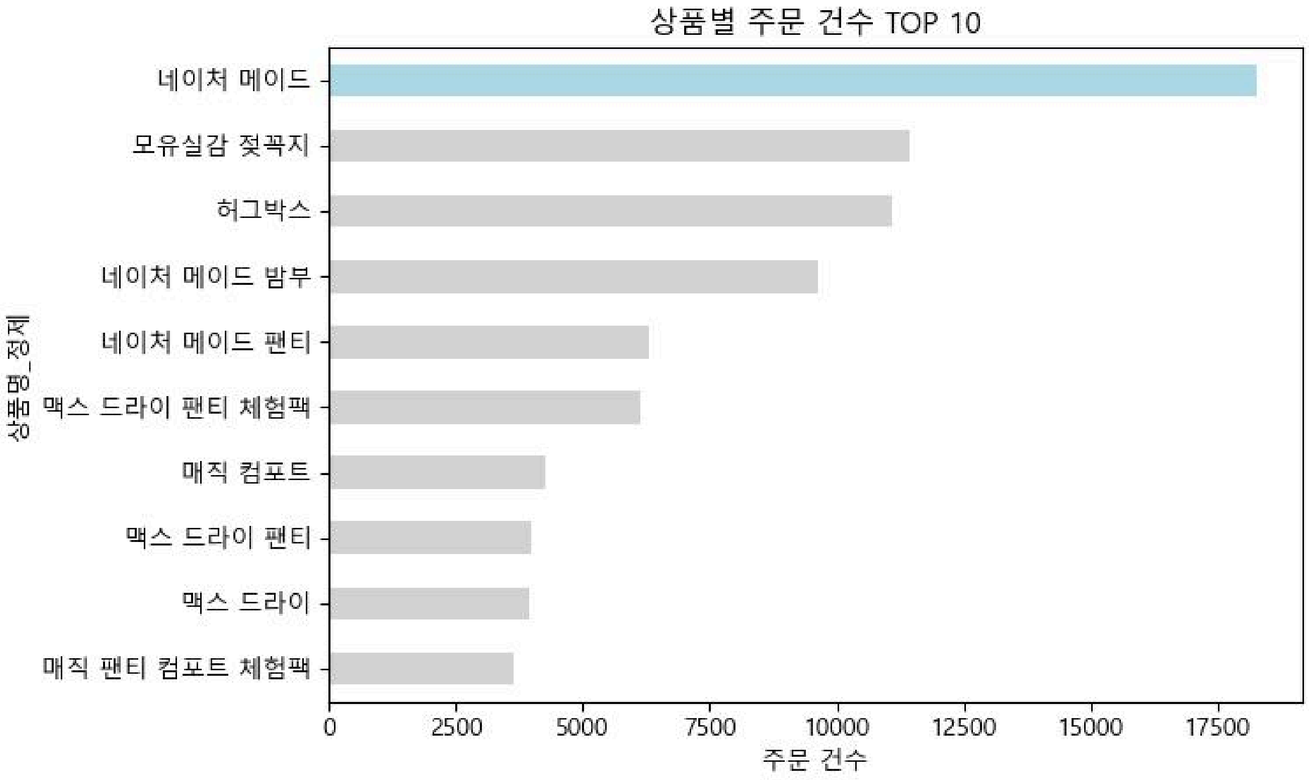
뒤집고 기는 아기 > 걷기 시작하는 아기 > 한달 무렵부터 > 아장아장 걷는 아기 > 태어난지 한달 무렵까지 순으로 주문이 많음



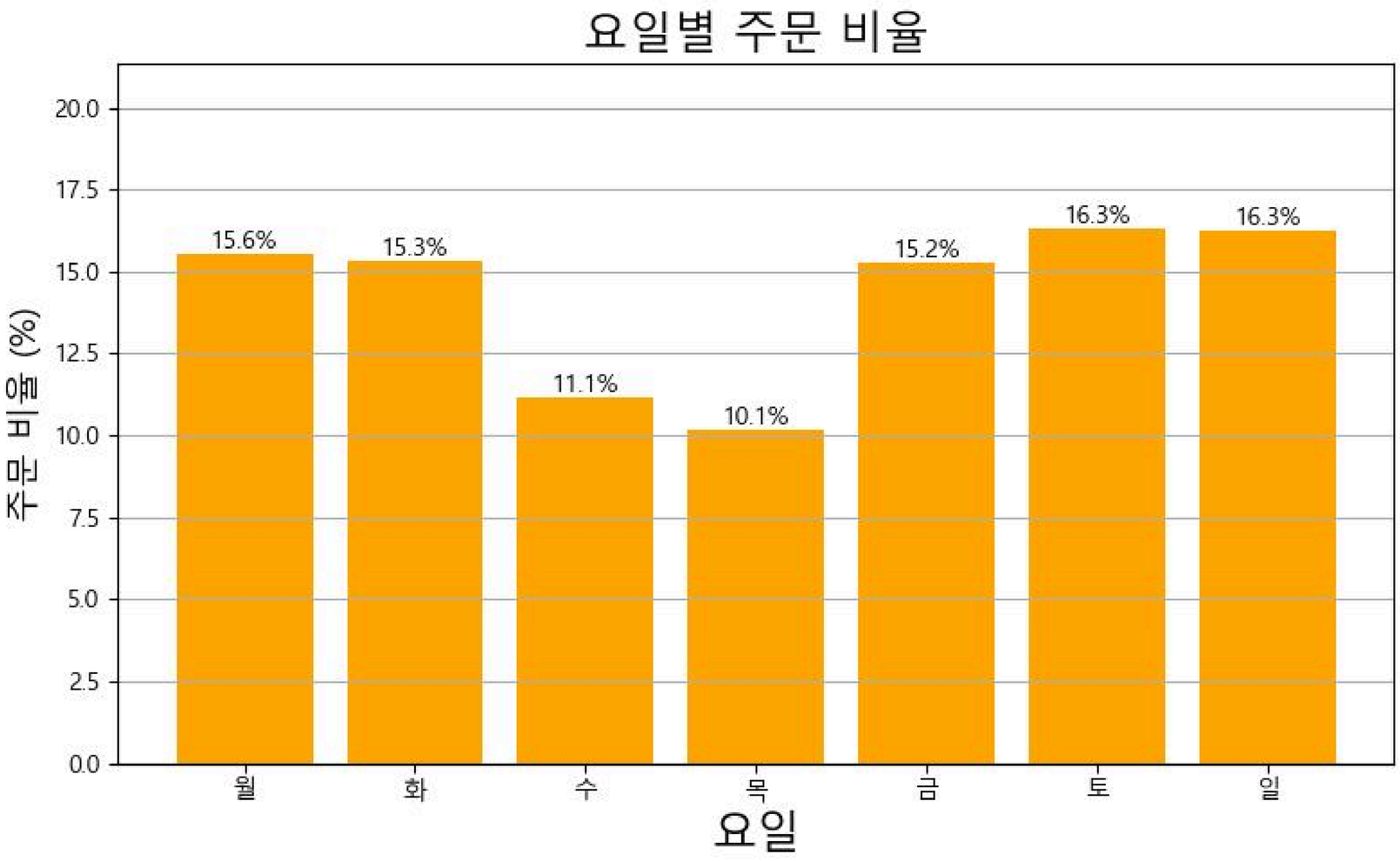
보통 2일 이내로 배송이 완료되는 주문이 가장 많았으나, 3일 4일 걸린 배송도 다수 집계됨

당일, 하루 배송 시장이 커지는 현시점에 이는 잠재적 위협으로 다가옴





대분류 중에서는 기저귀가 가장 많은 주문이 있었으며, 그 중에서 인기있는 기저귀 라인은 네이처 메이드임



월, 화, 금, 토, 일이 근소한 차이로 사실상 수&목을 제외하고는 주문건수가 비슷하였음

맘큐 뉴스 기사에 따른 매출 추이 분석

수집 데이터 : 대한민국 공식 뉴스빅데이터 분석 서비스 빅카인즈에서 맘큐를 키워드로 검색된 기사
수집기간 : 2022년 1월 1일 ~ 2023년 8월 8일

	기준일	전일대비후일_매출diff	전주대비후주_매출	전월대비후월_매출	기사제목
0	2022-03-17	-457200.0	2056267.0	37628528.0	친환경 오가닉 기저귀 출시
1	2022-04-07	610692.0	16287704.0	32826898.0	비닐포장재 재활용 캠페인
2	2022-09-08	1356638.0	-18103202.0	65749259.0	이른둥이용 초소형 기저귀 누적 기부 400판매
3	2022-09-21	-2918893.0	-3263986.0	9956277.0	우주패스
4	2022-10-13	1623739.0	-11256809.0	-38791326.0	기저귀의 누적 기부량
5	2023-04-07	9256563.0	21951521.0	21061077.0	썸머 기저귀·물티슈

- **멤버십 결합 상품인 <u>우주패스는 매출 변화가 적음</u>**
- **친환경 기저귀&썸머 기저귀, 물티슈 출시는 전주대비후주, 월대비후월 매출이 모두 성장**
  → 시장에서 긍정적 평가

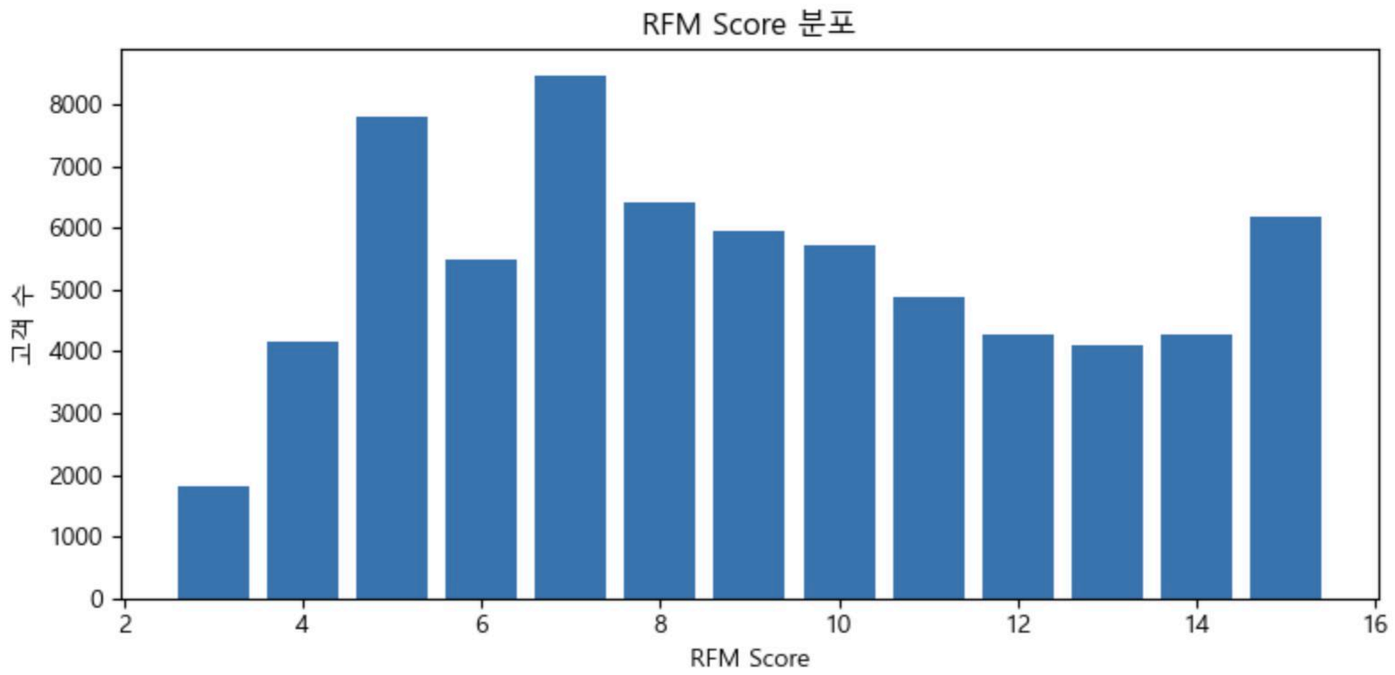
RFM 고객 세분화 분석

RFM

지표	의미	기준값	스코어링
Recency	얼마나 최근에 구매했는가	마지막 구매로부터의 일수	값이 작을수록 5점
Frequency	얼마나 자주 구매했는가	구매 횟수	클수록 5점
Monetary	얼마나 많은 금액을 지출했는가	누적 결제금액	클수록 5점

'가장 최근 배송 다음날'을 기준으로 최근성, 빈도, 지출액을 바탕으로
1~5점 스코어링 후 세그먼트화하여, VIP·신규·이탈고객을 분류하고 타겟 마케팅 전략 수립에 활용하기 위함

RFM 스코어 분포



- RFM 스코어는 3~15점 사이이며
- 가장 많은 고객이 몰려있는 구간은 7점(약 8,400명)과 5점(약 7,800명)

- 스코어가 <u>낮을수록</u> 재구매 가능성이 낮거나 마지막 구매가 오래된 고객,
- 스코어가 <u>높을수록</u> VIP고객(최근성, 빈도, 지출 모두 상위)임을 의미

⇒ 대부분은 중간 등급 : 아직 VIP는 아니지만, 재구매·지출 여력이 있는 “잠재 성장 고객”
⇒ 하위 스코어 : 이미 이탈 위험에 가깝거나, 구매 빈도가 매우 낮은 그룹
⇒ 상위 스코어 : 잦은 구매·높은 지출을 보이는 핵심 VIP

상위 10개 RFM 세그먼트

- **VIP (555)**

얼리 액세스
멤버십 업그레이드 쿠폰
포인트 2배 적립

- **휴면 위험 (221, 112, 113, 111)**

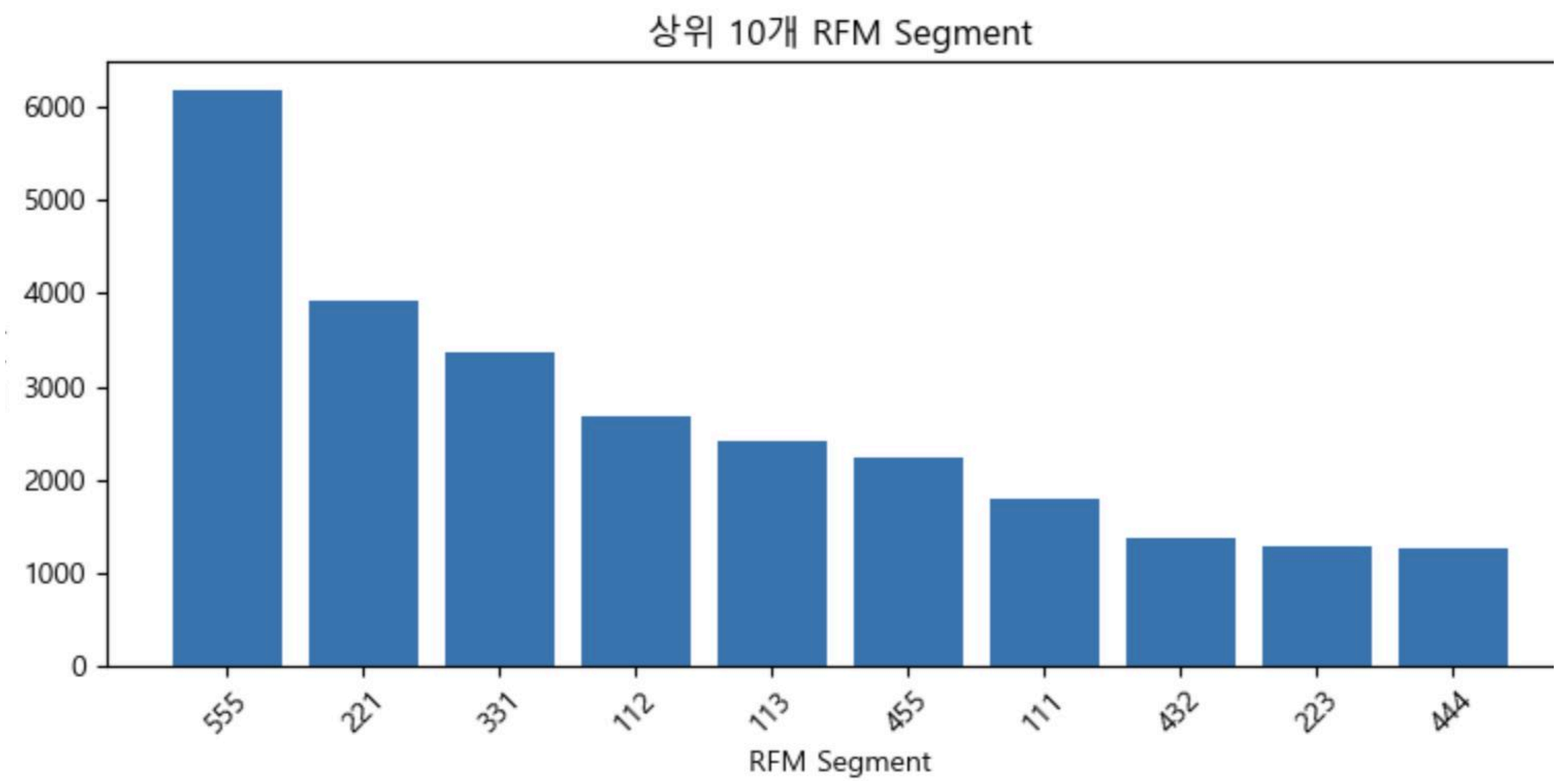
무료배송·샘플·한정 재관여 쿠폰
SMS/푸시 긴급 리인게이지

- **안정 고객**

시즌 기획전
추천인 프로그램
크로스셀링 캠페인

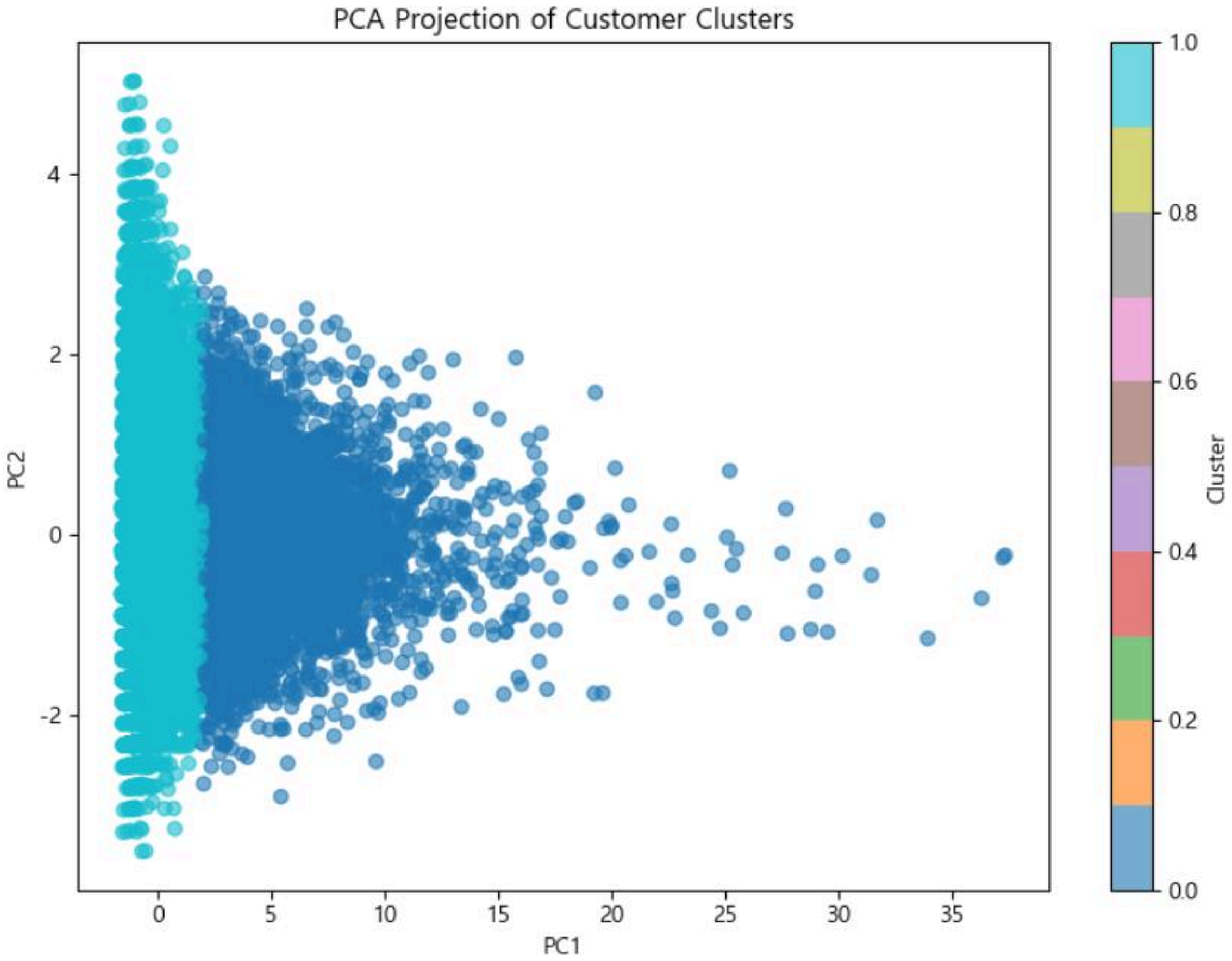
- **신규/고빈도 중저가**

웰컴 2차 구매 할인
번들 세트 제안
AOV 제고 프로모션



세그먼트	의미
VIP (555)	최근 구매·빈도·지출 모두 상위 → 핵심 로열티 고객
휴면 위험 (221, 112, 113, 111)	마지막 구매 오래됨·낮은 재구매 빈도 → 복귀 유도 필요
안정 고객	Recency·Frequency·Monetary 중간 → 꾸준한 재구매 그룹
신규/고빈도 중저가	최근 가입·적극적 구매(중저가) → 2차 구매 전환 타깃

KMeans 클러스터링



Silhouette Score: 0.518

	Recency	Frequency	Monetary	OrderCount	TotalSpend	AvgShipTime	연령
0	91.6	8.8	297038.0	8.8	297038.0	89.7	32.8
1	303.2	1.8	36897.5	1.8	36897.5	89.4	32.8

마케팅 제안

- **고가치 VIP & 잠재 성장군**
  - **VIP 유지·강화**
    - 신상품 론칭 24h 얼리 액세스
    - 구매당 포인트 2배 적립
    - 전용 수량 한정 에디션 안내
  - **잠재 성장 프로모션**
    - 2주간 2회 이상 구매 시 VIP 전용 쿠폰 제공
    - 이메일·앱 푸시 “지금 달성하면 VIP 혜택” 리마인더

- **저가치·휴면 위험군**
  - **긴급 리인게이지(Re-engage)**
    - SMS·앱 푸시 “놓치신 인기템 20% 할인”
    - X월 Y일까지 한정 재방문 시 적립금 제공
  - **2차 구매 전환**
    - 첫 방문 고객 대상 “두 번째 구매 시 15% 할인(7만 원 이상 구매 시 최대 1만 원)” 인센티브
    - 무료배송·샘플 쿠폰 병행 발송

3. 고객 이탈 방지와 전략

목적 | 데이터 기반 고객 이탈 방지 및 재활성화 방안

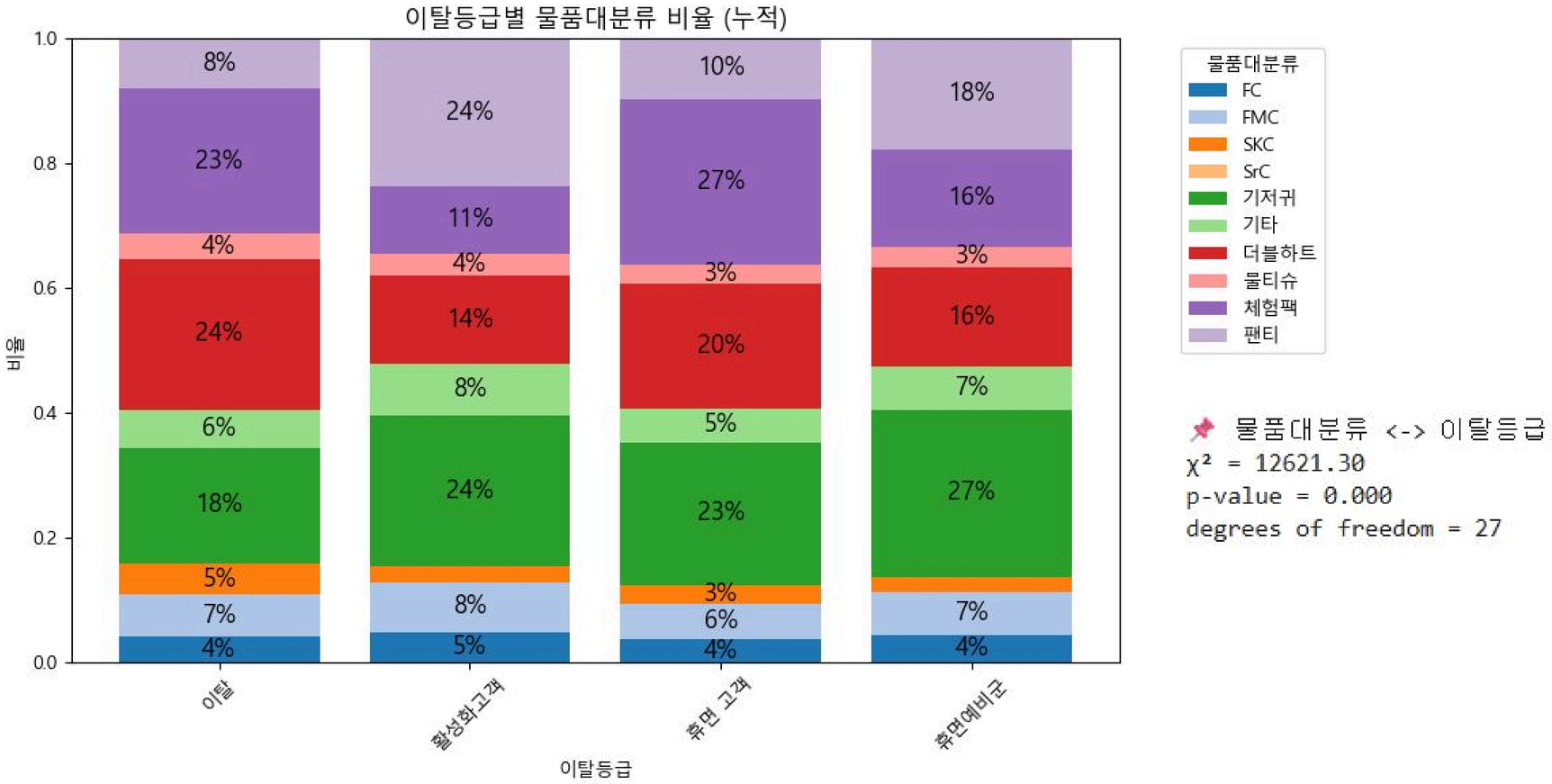
이탈 고객 정의

- 생필품 쇼핑몰인 특성상, 3개월 미구매시 휴면 전환으로 판단됨
- 다음 표에 따라 고객 분류를 정의내리기로 함

마지막 구매일로부터	고객 분류
~3개월	활성화고객
3개월~6개월	휴면예비군
6개월~1년	휴면고객
1년이상	이탈고객

⇒ 이탈 고객 : 최종 구매일로부터 1년 이상 지난 고객

가설1: 이탈고객은 주문하는 물품대분류가 일상적이지 않을 것이다. → 유의



- 카이제곱검정 결과 해당 가설은 유의하였음
- 이탈고객과 휴면 고객은 체험팩(23 ~ 27%)로 비일상적이고 일시적 소비 성격이 강함
- 반면, 활성화고객과 휴면예비군은 기저귀, 팬티형 기저귀(45 ~ 48%)를 구매하는 등의 명확한 육아생필품 소비 목적

가설2: 아이의 성장에 따라 이탈에 차이가 있을 것이다. → 유의

```
📌 구매시월령 <-> 이탈등급
이탈 Shapiro-Wilk: stat=0.664, p=0.000
휴면 고객 Shapiro-Wilk: stat=0.558, p=0.000
휴면예비군 Shapiro-Wilk: stat=0.526, p=0.000
활성화고객 Shapiro-Wilk: stat=0.559, p=0.000
Kruskal-Wallis H-statistic = 5786.771, p-value = 0.000e+00
```

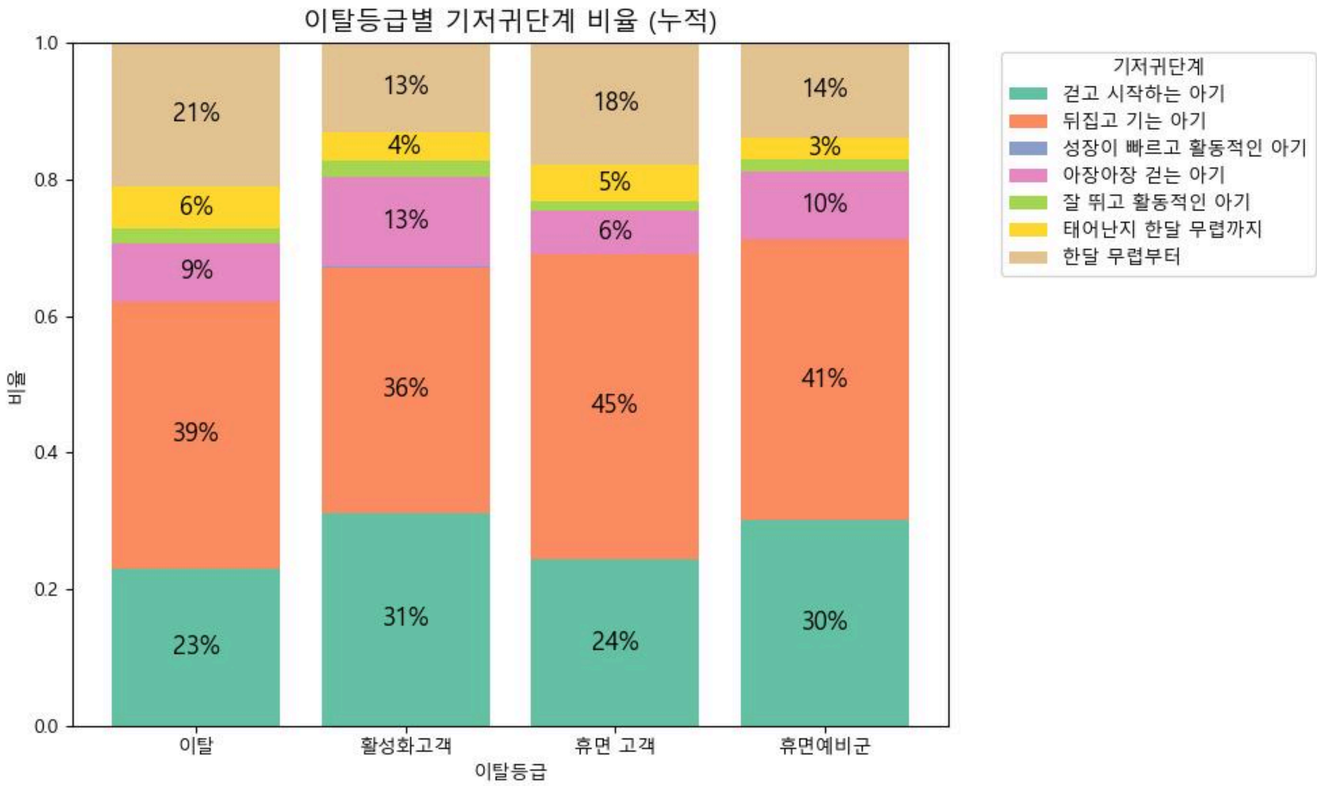
이탈 등급에 따른 자녀월령의 모든 그룹이 정규분포를 따르지 않음

⇒ 크루스칼 왈리스 검정 실시

⇒ 4개의 집단 간 자녀월령 중앙값 차이가 통계적으로 유의

- **중앙값 순위: 활성화고객 > 휴면예비군 > 휴면 고객 > 이탈**
- **활성화고객이 가장 높은, 이탈 고객이 가장 낮은 구매시월령을 보임**

가설2: 아이의 성장에 따라 이탈에 차이가 있을 것이다.



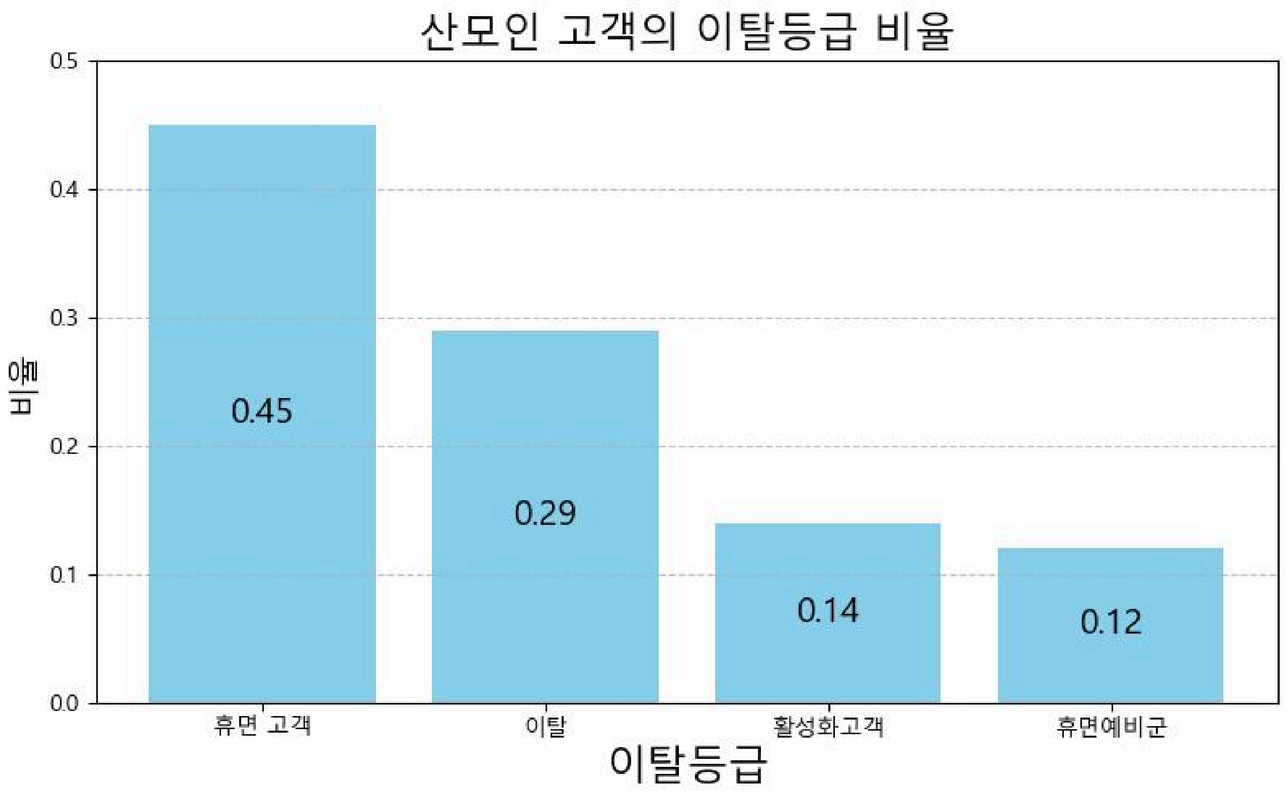
- 모든 고객군에서 '뒤집고 기는 아기' 단계 기저귀가 가장 많이 판매되며 아기의 성장 초중기에 기저귀 수요가 집중
- 활성화/휴면예비군

  → '걷기 시작하는 아기' 단계(30~31%)도 수요가 높음

  → 기저귀 단계를 자연스럽게 전환하며 구매를 지속함
- 이탈/휴면고객

  → 뒤집기 단계 다음인 걷기 시작한 아기 기저귀 구매 수요가 떨어진 것을 보아 이탈 고객은 뒤집기 단계에서 이탈하는 경향

가설3: 육아용품몰이기 때문에 산모는 이탈이 많을 것이다.



산모인 고객 중

휴면 고객 45% > 이탈 29% > 활성화고객 14% > 휴면예비군 12%

사실상 산모의 74%가 맘큐 쇼핑몰을 이용하지 않는다고 볼 수 있음

⇒ 산모는 아이를 양육하며 주요 고객층으로 성장할 수 있는 집단으로 대책 마련이 필요

맘큐는 2022년 5월 24일부터 당일출고 서비스 시작

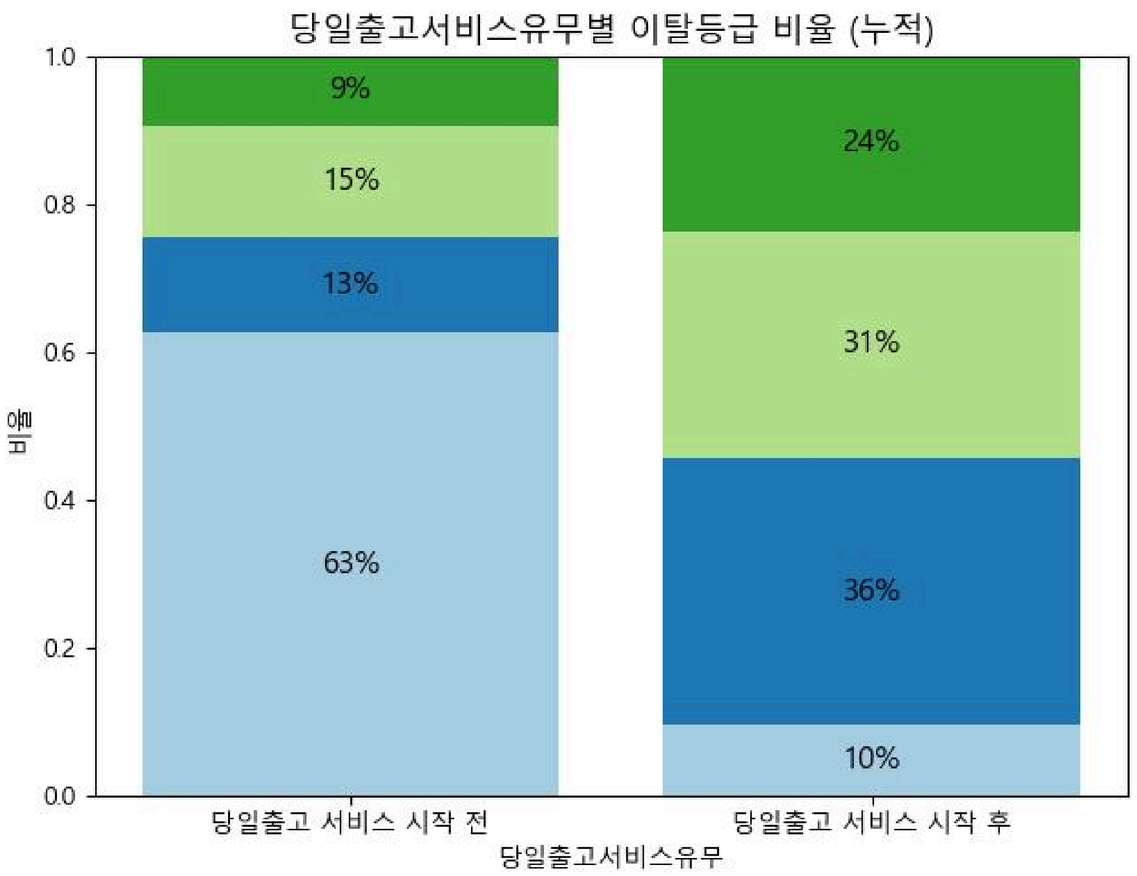
[공지] 당일출고 서비스 시작 안내

22.05.24 10:31

출처 : 맘큐 고객센터

⇒ 가설4: 당일출고 서비스 전후로 이탈 고객이 달라졌을 것이다.

가설4: 당일출고 서비스 전후로 이탈 고객이 달라졌을 것이다.



χ^2 = 47989.21
p-value = 0.000
degrees of freedom = 3

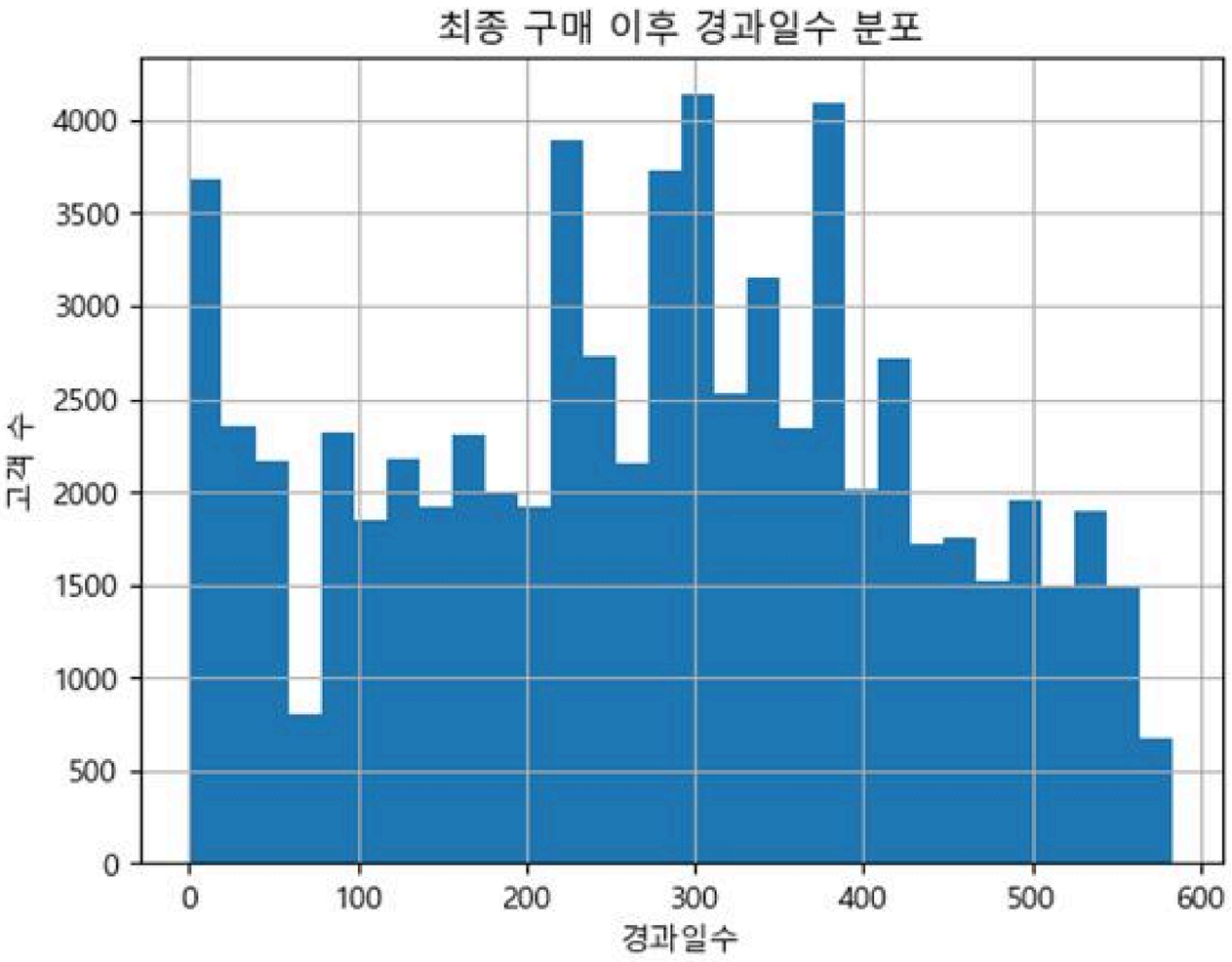
실제로는 서비스 시행 이후에 이탈 고객 비중이 줄고, 활성화 고객이 늘어남

→ 당일출고 서비스가 긍정적인 효과

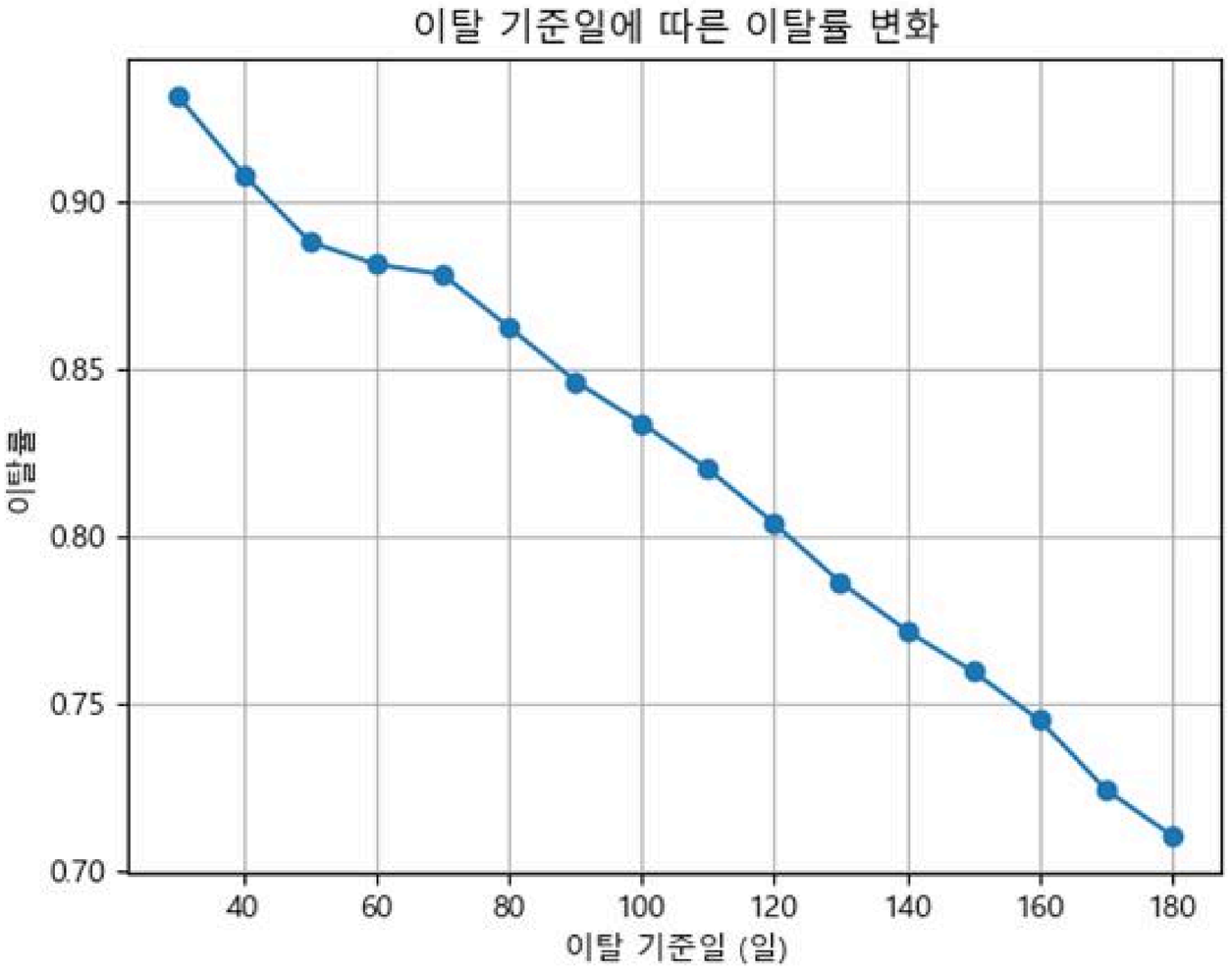
이탈 고객 파악에서 사용한 추가적인 가설(코드에서 확인 가능)

- 이탈고객은 미혼이 많을 것이다.
- 자녀가 없는 사람의 이탈이 클 것이다.
- 이탈고객의 배송에 걸린 시간이 비이탈고객보다 오래 걸렸을 것이다.
- 택배 이동 기간이 긴데 배송이 시작했으니, 취소도 못하며 불만이 쌓여서 이탈했을 것이다.
- 당일출고서비스를 시작한 전후로 이탈 등급에 차이가 있을 것이다.
- 고객등급에 따라 이탈 여부에 차이가 있을 것이다. → 유의하였으나, 당연한 사실이므로 피피티에서는 제외하였음

최종 구매 이후 경과일수 분포



이탈 기준일에 따른 이탈률 변화



이탈 고객 예측 모델

목적 | 이탈 고객을 선제적으로 파악하고 마케팅을 진행

AutoML 라이브러리 적용

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
lightgbm	Light Gradient Boosting Machine	0.9071	0.9325	0.9606	0.9326	0.9464	0.6001	0.6040	0.6790
xgboost	Extreme Gradient Boosting	0.9076	0.9323	0.9602	0.9335	0.9466	0.6036	0.6071	0.4170
rf	Random Forest Classifier	0.8837	0.9124	0.9165	0.9455	0.9308	0.5674	0.5701	3.2460
et	Extra Trees Classifier	0.8763	0.8976	0.9084	0.9445	0.9261	0.5475	0.5515	2.2480
gbc	Gradient Boosting Classifier	0.8741	0.8912	0.9137	0.9371	0.9253	0.5256	0.5274	5.4460
ada	Ada Boost Classifier	0.8512	0.8646	0.8829	0.9392	0.9101	0.4806	0.4885	1.2580
knn	K Neighbors Classifier	0.8219	0.8500	0.8371	0.9481	0.8891	0.4453	0.4686	0.8670
lda	Linear Discriminant Analysis	0.8235	0.8318	0.8367	0.9506	0.8900	0.4530	0.4777	0.1560
ridge	Ridge Classifier	0.8235	0.8317	0.8367	0.9506	0.8900	0.4531	0.4778	0.0990
lr	Logistic Regression	0.8163	0.8291	0.8273	0.9511	0.8849	0.4414	0.4691	1.4120
nb	Naive Bayes	0.8305	0.8101	0.8747	0.9227	0.8980	0.3986	0.4037	0.1020
svm	SVM - Linear Kernel	0.7436	0.7744	0.7510	0.9396	0.8251	0.3232	0.3694	0.5540
dt	Decision Tree Classifier	0.8553	0.7410	0.9017	0.9268	0.9141	0.4577	0.4595	0.2690
dummy	Dummy Classifier	0.1467	0.5000	0.0000	0.0000	0.0000	0.0000	0.0000	0.1280
qda	Quadratic Discriminant Analysis	0.7119	0.1121	0.8000	0.6826	0.7367	0.0000	0.0000	0.1390

고객 이탈 예측 모델 최종 성능

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	Light Gradient Boosting Machine	0.9156	0.9429	0.9624	0.9401	0.9511	0.6423	0.6447

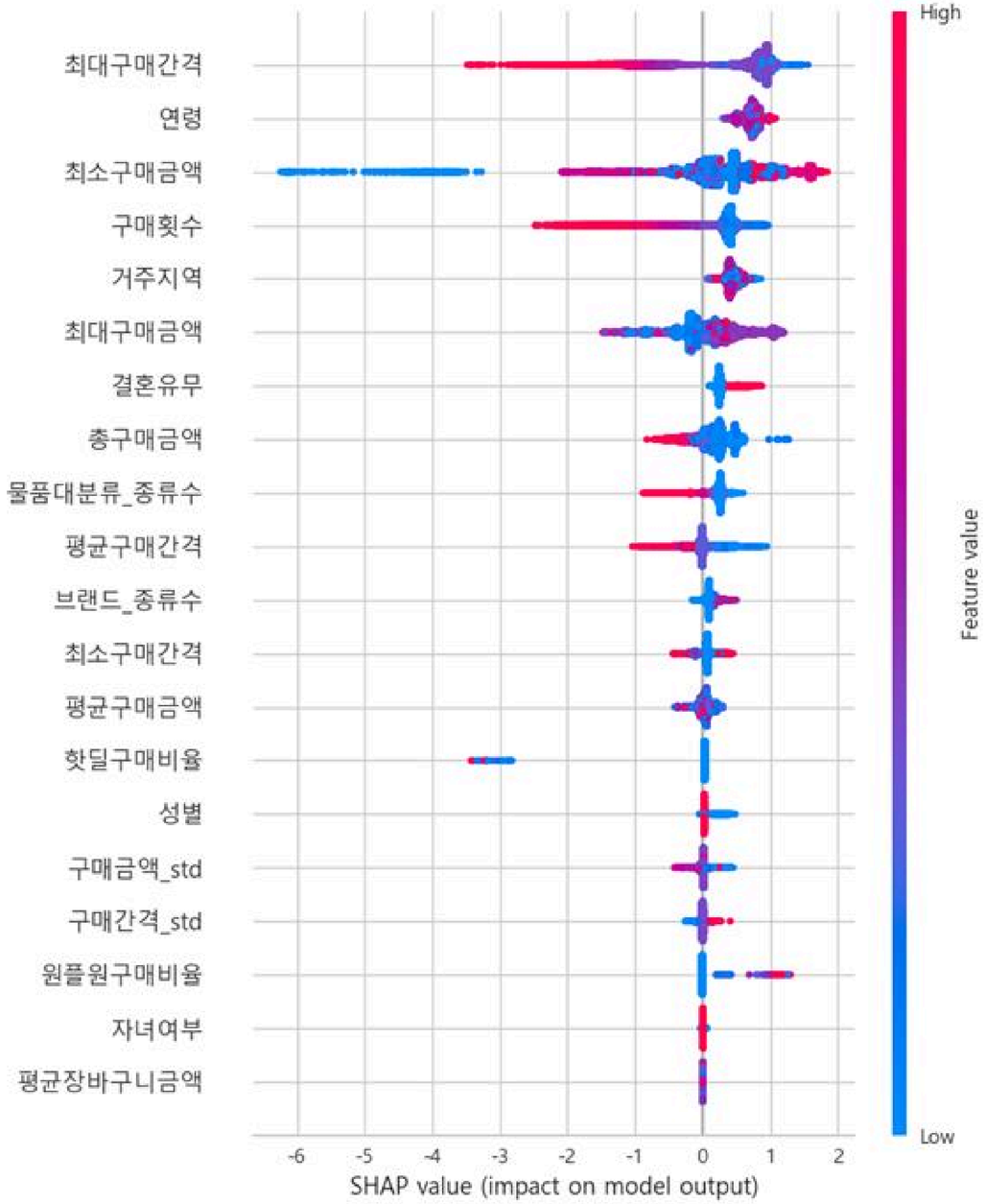
피처 중요도 분석 (Mean |SHAP|) 요약

- 상위 중요 피처
  a. 최대구매간격 (≈0.9)
  b. 연령 (≈0.7)
  c. 최소구매금액 (≈0.6)
  d. 구매횟수 (≈0.45)
  e. 거주지역 (≈0.43)

- 중간 기여 피처
  - 최대구매금액, 결혼유무, 총구매금액 등

- 낮은 기여 피처
  - 브랜드_종류수, 평균구매금액, 핫딜구매비율, 성별, 자녀여부 등

⇒ RFM 지표(구매간격·금액·횟수)가 이탈 예측의 핵심



주요 피처 해석

1. 최대구매간격
   - (간격↑) 빨간 점이 왼쪽에 몰려 있음.

   ⇒ 간격이 크면 이탈 확률을 내림.

2. 최소구매금액
   - 높은 최소구매금액(빨강)은 오른쪽에 몰려 있음.

   ⇒ 금액이 크면 이탈 확률을 올림.

3. 구매횟수
   - (구매횟수↑)붉은 점은 주로 왼쪽에 몰려 있음.

   ⇒ 횟수가 크면 이탈 확률을 내림.

맘큐에 제안하는 고객의 이탈을 방지하고 관리 방안 제안

1. 구매 주기 관리
   - 장기 미구매(최대구매간격↑) + 이탈 모델로 이탈 분류된 고객에게
     - SMS·앱 푸시로 재구매 알림
     - 주기별 자동 할인 코드 발송

2. 금액 기반 인센티브
   - 최소구매금액↑ 전략
     - 기준 금액(평균 최소구매금액) 이상 구매 시 추가 포인트·쿠폰 지급

3. 비일상적이고 일시적 소비 성격이 강한 이탈 고객을 재유입시키기 위해 반복구매 유도
   → 첫 구매 후, 일상적 상품 정기구독 시, 적용 가능한 쿠폰 제공

4. 아기가 '뒤집기를 하는 단계'는 이탈 리스크가 높은 구간
   → 이 시점 고객에게는 선제적으로 다음 단계 기저귀 추천을 해줘야 함
   → '걷기 시작하는 아기' 제품 구매까지 전환된다면 기저귀 구독제 제공으로 정기적 구매 유도

맘큐에 제안하는 인사이트 기반의 전략

5. 산모의 이탈 문제 전략

[문제]

- 산모 고객의 약 74%가 이탈
- 이탈한 산모 고객은 일시적 이탈일 가능성이 높고 아기가 성장함에 따라 재구매 및 재방문 여지가 충분함
- 이들을 방치한다면, 타 브랜드의 충성 고객이 될 수 있는 손실로 이어짐
- 이탈한 산모의 시기별 니즈 변화에 맞춰 맞춤형 접근이 필요함

[전략]

→ 개발한 이탈 예측 모델을 기반으로 이탈 징후를 보이는 고객에 대해 쿠폰/리필 알림/성장 단계별 제품 추천을 자동화

→ 고객 여정 기반 맞춤 캠페인 운영

- 출산할 것으로 예상되는 시점&아기 성장 단계에 맞는 허그박스 캠페인 진행
- 해당 캠페인을 고객이 가장 많이 유입된 인스타그램 > 블로그 순으로 홍보 진행

4

맘큐의 현황 진단

목적 | 맘큐의 경쟁력을 위한 인사이트를 도출

맘큐에서 진행 중인 캠페인 : 허그박스

출처 : 맘큐 허그박스 캠페인 페이지

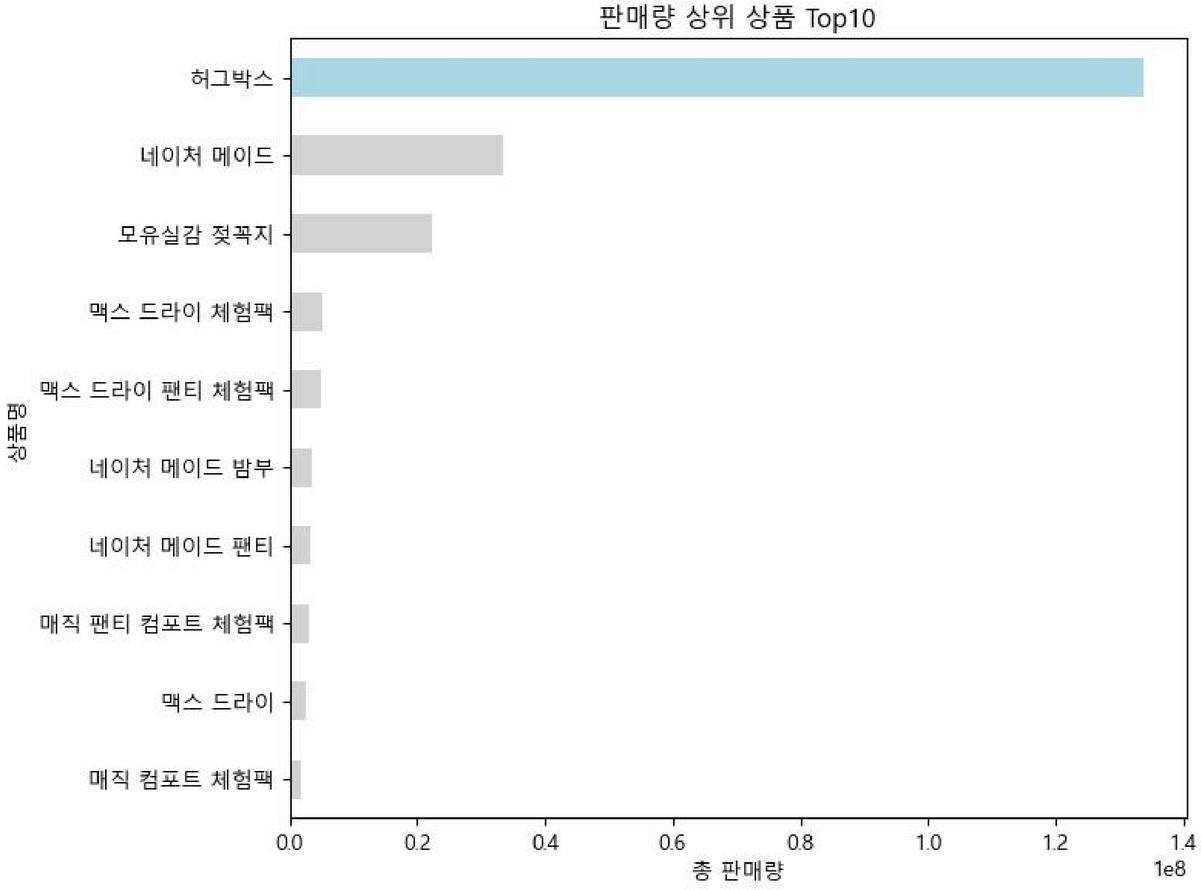
출처 : 실제 소비자가 받은 맘큐 허그박스

허그박스 수령 후, 후기 작성 시 배송비를 돌려주고 추첨을 통해 쇼핑지원금 3만 원 제공

블로그/카페/SNS 등에 후기를 남기면 당첨확률 상승

- **연간 출생아 24만명 중 16만명에 달하는 약 60%가 허그박스 이용** 출처 :한국경제

이 가운데 올해로 10주년을 맞은 '맘큐'는 누적 회원 수 200만명 이상을 보유하며 국내 대표 육아 플랫폼으로 성장했다. '하기스 허그박스'의 경우 연간 약 14만명의 회원이 맘큐 앱에서 신청해 무상으로 육아용 제품을 받고 있다. 지난해 출생아 수(24만명)의 60%에 해당하는 숫자다.



실제로 허그박스 판매량은 전제품 중 압도적으로 많은 판매량을 보임

- **제공받은 데이터의 약 5%가 허그박스 주문 건임**

```
tab = df.copy()
tab['연도'] = df['구매일'].str.split('-').str[0]
tab = tab[tab['연도'] == '2022']

hug = df[df['상품명'].str.contains('허그박스', na=False)].shape[0]

print(f'{bold}2022년 매출 :{end} {tab['결제금액'].sum()} 원')
print(f'전체 데이터 중 허그박스 주문 건수 비율 : {bold}{hug / df.shape[0]* 100:.2f}%{end}로 {bold}{hug}{end}건 입니다.')
```

```
2022년 매출 : 3128951607.0 원
전체 데이터 중 허그박스 주문 건수 비율 : 5.73%로 12016건 입니다.
```

허그박스는 약 50%가 출산 전 3개월 전부터 출산할 시점에 주문을 하였음

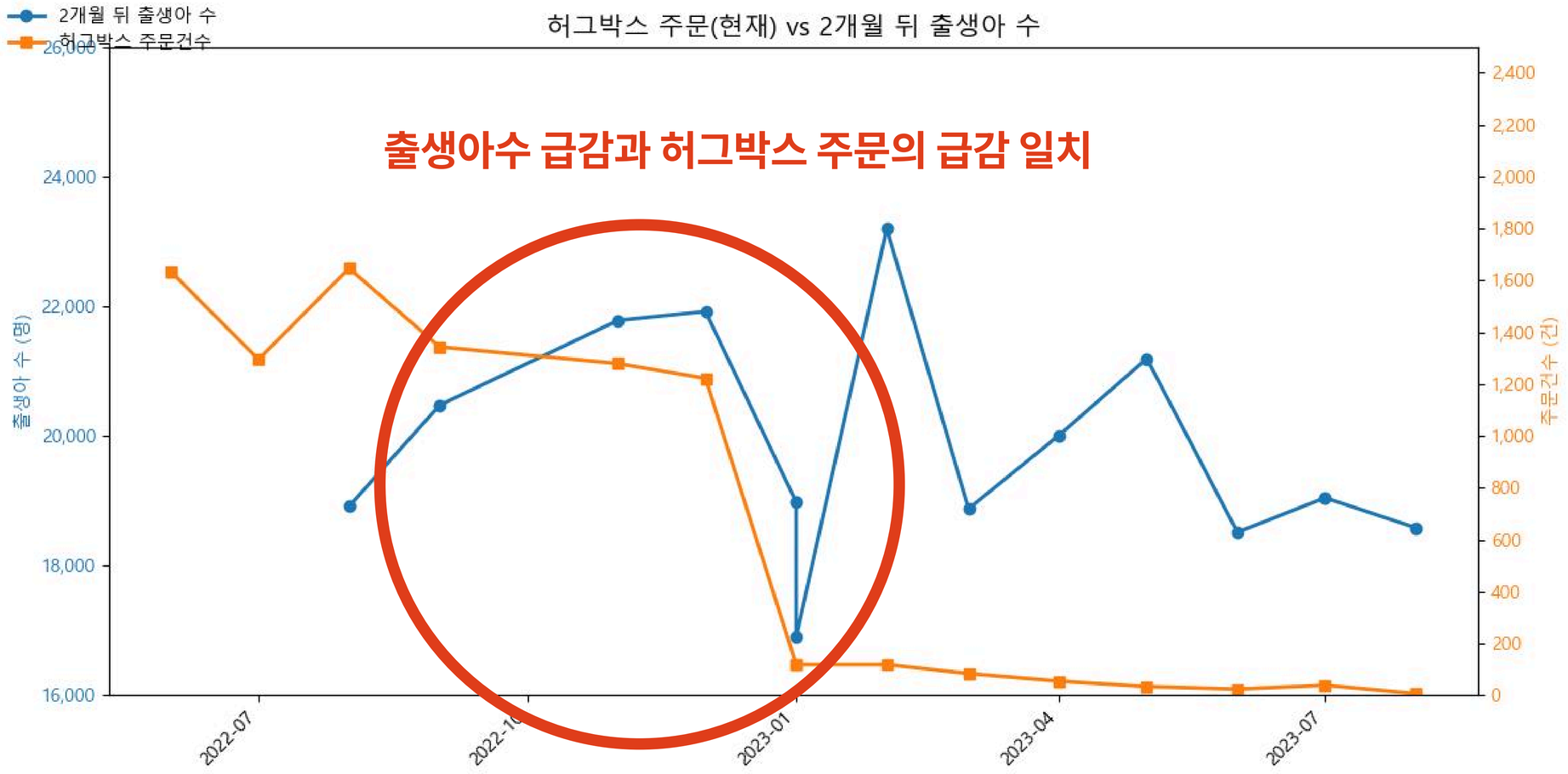
```
{1: 0.2877961253889306, 2: 0.298317539613317, 3: 0.12094734129753854}
▶ 가장 겹치는 시프트: +2개월
```

⇒ 주문건수와 시프트된 월별 출생아 수의 피어슨 상관계수가 2개월 전에 가장 강했음

출생아수와 허그박스 주문의 연관성

월별 출생아수 데이터를 2개월 시프트해 허그박스 주문건과 일치하는지 시각화하였음

→ 제공받은 맘큐 데이터가 샘플링된 데이터로 허그박스 주문건수가 실제보다 축소된 것으로 추정



허그박스 캠페인 반응과 성과 진단

- 사용한 외부 데이터
- 2022년 1월 1일 ~ 2023년 8월 8일 | 출생아수 (출처 : 통계청 kosis)
- 2018년 1월 1일 ~ 2025년 6월 23일 | 맘큐 허그박스 리뷰 데이터 (출처 : 맘큐 허그박스 캠페인 페이지)

KKO**********

2025-06-24 20:01

구성품도 좋고 아기가 태어나면 써보고 정착할거같아요
감사히 잘쓰겠습니다!

2만 5천 건의 리뷰와 날짜를 스크래핑

	content	date
0	이제곧 출산을 앞두고 우리아기 기저귀는 하기스 네이처 메이드로 해야지 하고 결정후,...	2018-01-01
1	허: 허름한 옷을 내가 입더라도그: 그냥 먹고 싶지 않은 음식을 먹더라도박: 박수받...	2018-01-01
2	하: 하루동안 내 아기를 쳐다보다기: 기저귀 고급진 하기스 기저귀를 하루종일 갈다가...	2018-01-01

허그박스 리뷰 텍스트 마이닝

- **목적 : 시점별로 리뷰 내 잠재 주제를 추출하여 변화를 파악하고 대표 허그박스에 대한 반응을 파악하고자 함**

1. 전처리 및 형태소 분석

- **띄어쓰기 보정 : pykospacing 라이브러리의 Spacing 객체를 사용**

[‘구성이알차고기저귀가너무좋네요’ → ‘구성이 알차고 기저귀가 너무 좋네요’ **]**

- **형태소 분석기 비교 - Okt vs Kiwi**

VS

→ **워드클라우드 : Okt 선택** 하나하나 불용어 처리를 해야 해서 Okt가 소비자 직관적인 반응 표현을 더욱 잘 담아냈음

→ **LDA 토픽 모델링 : Kiwi 선택** 품사에 따라 필터링할 수 있어서 주제 분류에 더 적합

허그박스 리뷰 텍스트 마이닝

1. 전처리 및 형태소 분석

- 불용어 처리 : 직접 데이터를 보며 조사, 의미없는 불용어 리스트를 수기 입력하여 제거
- 토큰 필터링 기준 : 의미 있는 단어만 추출하기 위해 2자 미만 단어 제외

```
불용어 = {
    '의', '이', '가', '은', '는', '을', '를', '에', '에서', '도', '으로', '로', '와', '과',
    '하게', '한', '함', '요', '니다', '좀', '잘', '것', '수', '해', '더', '같아요', '같아',
    '되다', '있다', '없다', '하다', '정말', '해서', '그리고', '그런데', '이런', '저런', '그런',
    '합니다', '입니다', '없는', '있는', '인데', '해도', '또한', '하지만', '해주시고', '주시고',
    'ㅎㅎ', 'ㅋㅋ', '게요', '되었어요', '되었답니다', '되었는데', '했네요', '했어요', '했음',
    '했지만', '했답니다', '하겠습니다', '하구요', '하시', '하는데요', '되네요', '있네요',
    '있었어요', '있었답니다', '없었어요', '없었답니다', '네', '근데', '그래도', '이지만',
    '어떤', '왜', '또', '다', '등', '및', '까지', '많이', '조금', '아주', '되게', '정말로',
    '완전', '되었네요', '이예요', '좋네요', '좋았어요', '좋았네요', '좋구요', '좋겠어요',
    '좋을', '좋은데', '좋습니다', '좋아요', '감사', '감사해요', '감사합니다', '감사합니당',
    '감사할', '감사히', '하겠습니', '해봐요', '해봤어요', '해보려구요', '해봤는데',
    '해보면서', '되니까', '좋겠네요', '좋았', '쓸게요', '쓸께요', '쓸께', '쓸게', '써요',
    '썼어요', '썼어용', '써야', '써보려', '써보니', '써보려고', '써보고', '써봤는데',
    '써보면', '쓰고', '쓰고있어요', '쓰는중', '쓰고있습니다',
    '박스', '허그', '하기스', '기저귀', '사용'
}
```

허그박스 리뷰 텍스트 마이닝

2. 문서 벡터화

- 각 문서에서 단어의 출현 빈도를 기반으로 문서-단어 행렬을 생성하는 **BoW** 방식을 구현한 **Sklearn**의 **CountVectorizer**로 **단어 벡터화**

3. 토픽 모델링(LDA)

- **목적 : 약 2만 5천 건의 리뷰 속에서 소비자들이 허그박스에 대해 이야기하는 공통 주제를 알고자 함**
- 생성된 단어 행렬을 기반으로 **Latent Dirichlet Allocation (LDA)** 알고리즘을 적용
- LDA라는 분석 기법을 사용해 주제를 자동으로 3가지로 분류

허그박스 리뷰 워드클라우드

2022년 1월 ~ 2023년 8월

출산
구성
베이
좋은
단계
체험
보고
알찬
다양한
맞는
종류
별로
엄마
아이
주문
이렇게
생각
용품
후기
성품
아가
패드
도움
우리
필요한
임신
가지
받아
여러
한테
다양하게
이라
선택
부터
하나
맘큐
좋았습니다
알차고
받았습니다
들어
조리
할지
블로그
첫째
예비
앞두고
아니라
받아서
태어나면
고민
있고
산모
빠르고
안심
바로알차게
모르는
처음
포장
같이
네이처
있어요
있게
있어
있는
있을
있습니다
있었습니다
있었는데
있었어요
있었어
있었
있었
있었

허그박스 리뷰의 상위 주제 3가지

2022년 1월 1일 ~ 23년 8월 8일

(제공받은 데이터와 동일한 기간의 주제)

◆ 2022~2023.8 LDA Topics

	Topic 1	Topic 2	Topic 3
0	구성	기저귀	하기스
1	사용	종류	박스
2	알차	박스	기저귀
3	제품	감사	아기
4	감사	샘플	출산
5	배송	다양	엄마
6	다양	아기	위하
7	샘플	신청	감사
8	빠르	출산	선물

2023년 8월 9일 ~ 25년 6월 23일

(제공 데이터 이후 기간의 리뷰 스크래핑)

◆ 2023.8 이후 LDA Topics

	Topic 1	Topic 2	Topic 3
0	기저귀	기저귀	후기
1	구성	박스	박스
2	사용	하기스	블로그
3	감사	단계	이벤트
4	하기스	체험	감사
5	박스	사용	신청
6	다양	신청	구성
7	샘플	아기	작성
8	제품	다양	하기스

블로그를 통한 수익이 인기를 끌었음

SNS 홍보글을 작성하는 리뷰가 늘어난 것이 주제로도 나타남

```python
blog = df[df.content.str.contains('blog', na=False)].shape[0] / df.shape[0]
insta = df[df.content.str.contains('insta', na=False)].shape[0] / df.shape[0]
http = df[df.content.str.contains('http', na=False)].shape[0] / df.shape[0]

display(HTML(f'블로그 작성비율 : <b>{blog*100:.1f}%</b>'))
display(HTML(f'인스타 작성비율 : <b>{insta*100:.1f}%</b>'))
display(HTML(f'모든 SNS 작성비율 : <b>{http*100:.1f}%</b>'))
```

블로그 작성비율 : **16.6%**

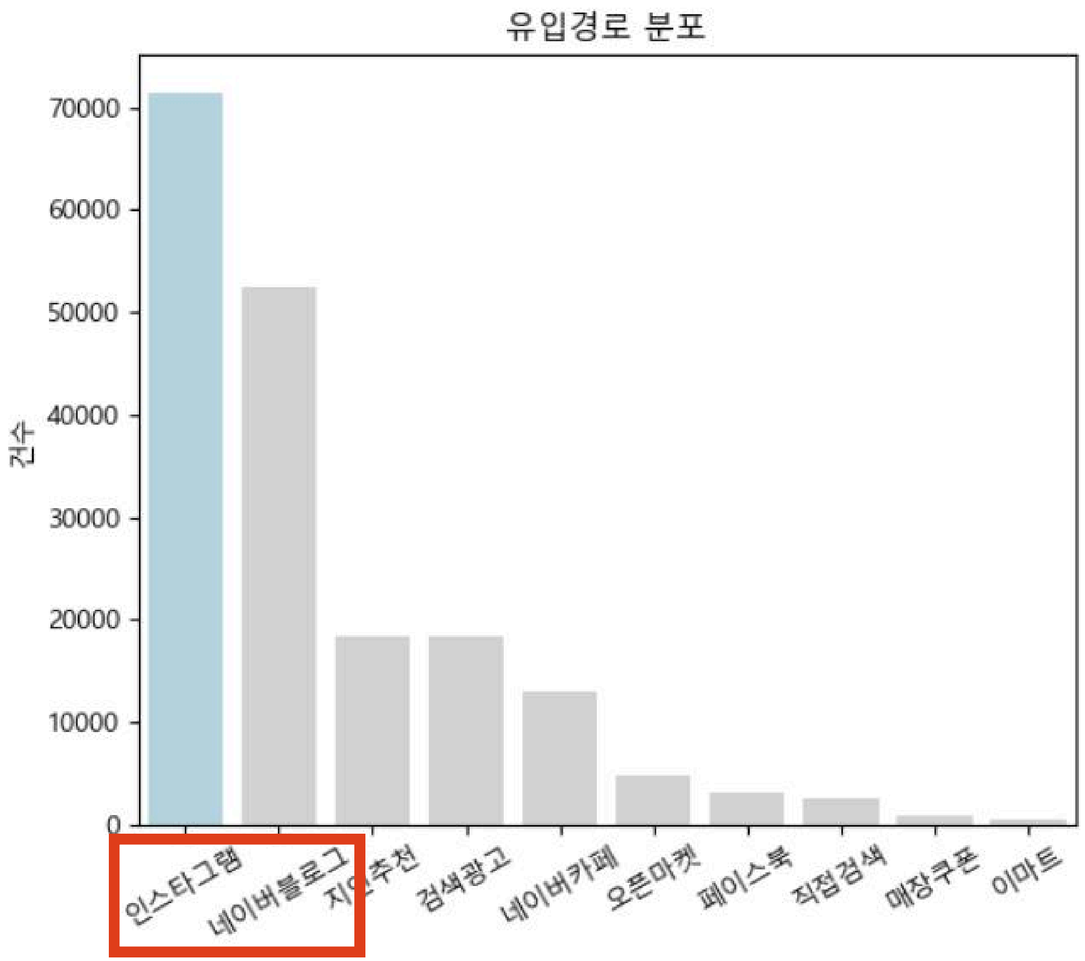
인스타 작성비율 : **7.7%**

모든 SNS 작성비율 : **24.1%**

실제로 허그박스 리뷰 데이터 중 24%가 SNS 홍보글을 작성

⇒ 허그박스 캠페인이 소비자의 자발적인 맘큐 홍보로 이어짐

허그박스 효과

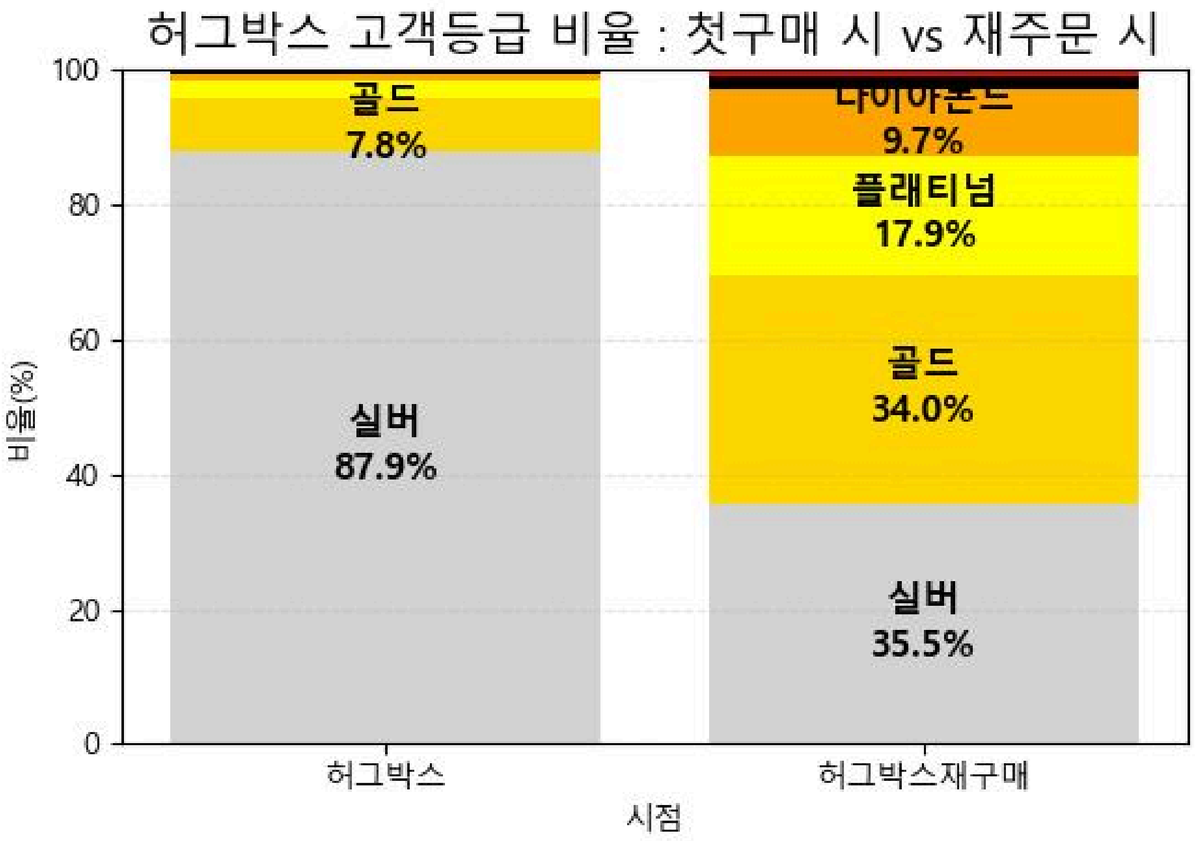


실제로 고객들의 유입경로를 살폈을 때
Top2가 인스타그램, 네이버 블로그가 월등히 높았음

허그박스 효과

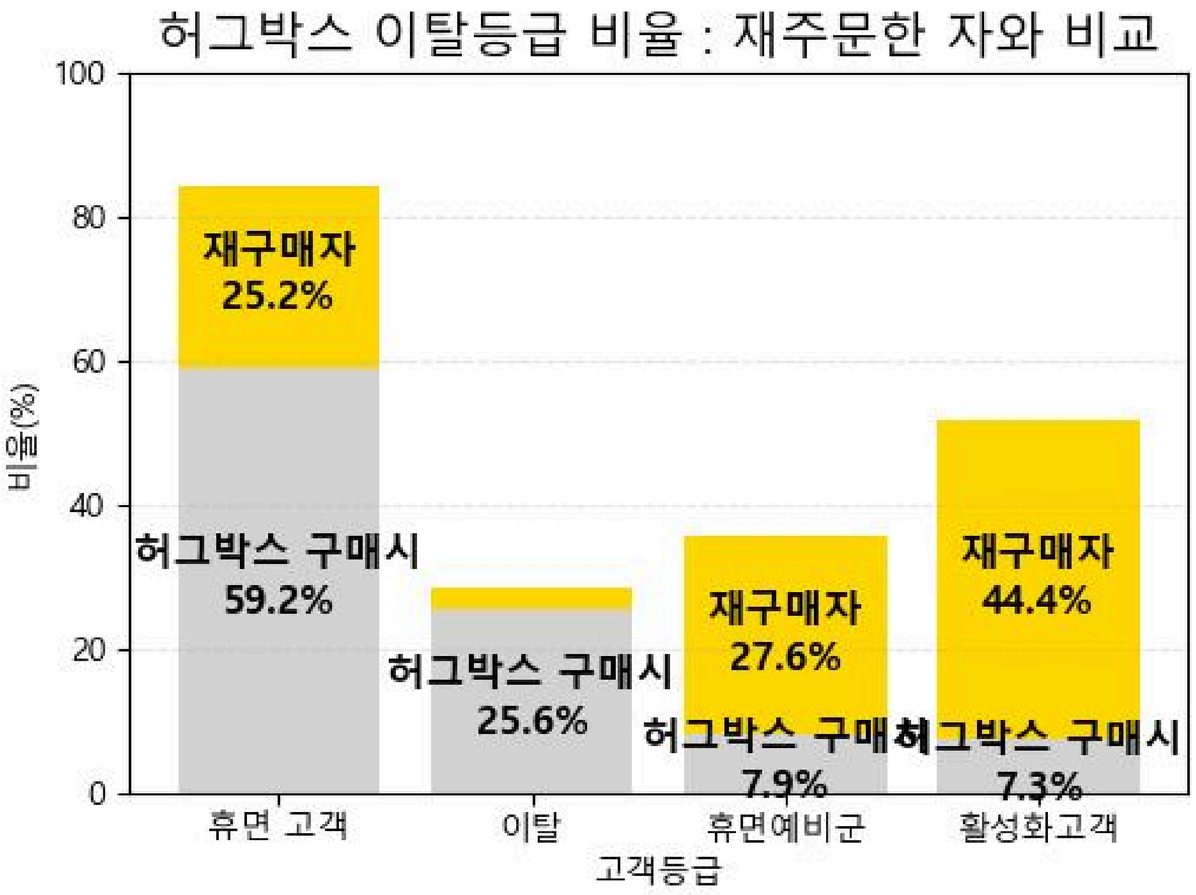
```
허그박스 재구매 전환율
전체 허그박스 구매자 수: 11,064명
재구매한 고객 수(2회 이상): 2,794명
재구매 전환율: 25.25%
```

허그박스 구매를 했던 고객 중
25%가 재주문했으며, 고객 등급의 상향이 이루어짐



- **고객등급 순서 : 브론즈 > 실버 > 골드 > 플래티넘 > 다이아몬드 > 스타**
  (출처 : 맘큐 고객센터)

허그박스 효과



허그박스를 통해 재구매한 사람은
3개월 이내 주문한 활성화 고객이 44%
3~6개월 사이에 주문한 휴면예비군 27.6%

⇒ 허그박스를 통해 유입된 고객은 활성화 고객으로
기업에 이익이 되는 고객군임

허그박스 캠페인 평가

허그박스 캠페인을 통해 유입된 고객 중 약 25%가 재구매로 전환

활성화 고객 비중도 높아 충성고객 확보에 도움이 되고 있음

또한, SNS 리뷰 이벤트를 병행함으로써 소비자가 자발적인 홍보 주체가 되었으며

기존 고객이 새로운 고객을 유입시키는 선순환 구조가 형성

↓

허그박스는 단순 체험 제공을 넘어, 브랜드 인지도 제고와 충성고객 확보까지

유의미한 효과를 거둔 캠페인

↓

따라서 맘큐는 허그박스 캠페인을 장기적으로 유지할 필요가 있고, 유사한 캠페인을 진행하기를 권함

5

제품의 적기 배송

제품의 적기 배송을 통해 고객의 만족을 높이고자 한다

유한킴벌리 자사브랜드 물류센터

2024년 다트에 공시된 감사보고서에 따르면
물류센터와 공장은
대전, 김천, 충주, 안양 총 4곳

5. 토지의 공시지가

보고기간 종료일 현재 당사가 보유중인 토지의 공시지가 내역은 다음과 같습니다.

(단위 : 천원)

내 역	소 재 지	장 부 가 액		공 시 지 가	
		당기말	전기말	당기말	전기말
대전공장부지	대덕구 문평동 외	15,589,510	15,589,510	48,806,769	45,017,849
김천공장부지	김천시 대광동 외	17,102,356	17,102,356	35,238,183	33,372,626
충주공장부지	충주시 대소원면	7,427,445	7,427,445	16,627,288	15,496,681
안양물류센타	군포시 당정동 외	37,171,710	37,171,710	110,969,301	102,803,183
합 계		77,291,021	77,291,021	211,641,541	196,690,339

출처 : 다트에 공시된 유한킴벌리 감사보고서2022

출처 : 유한킴벌리 공식홈페이지

* 네이버 지도에서 추가적인 유한킴벌리 부지들이 확인되었으나, 대리점이 혼용되어 공식 자료에 있는 명확한 사실만을 사용하였습니다.

유한킴벌리 공장별 생산품 차이

세계 최고의 가치 창출을 목표로 완공된 대전공장

대전공장은 유아아동용품 전문사업장으로 아기 기저귀, 아기 물티슈, 영유아 스킨케어 등의 완제품과 함께 첨단 부직포 소재 등을 생산, 공급하고 있습니다.

아시아 1위 티슈 사업장, 김천공장

김천공장은 미용티슈, 화장실용 화장지, 키친타올 등 가정용품과 B2B 전용 화장지 및 핸드타월, 부직포를 생산하는 아시아 최고의 위생제지 생산공장입니다.

여성 · 웰니스케어용품 글로벌 벤치마킹센터 충주공장

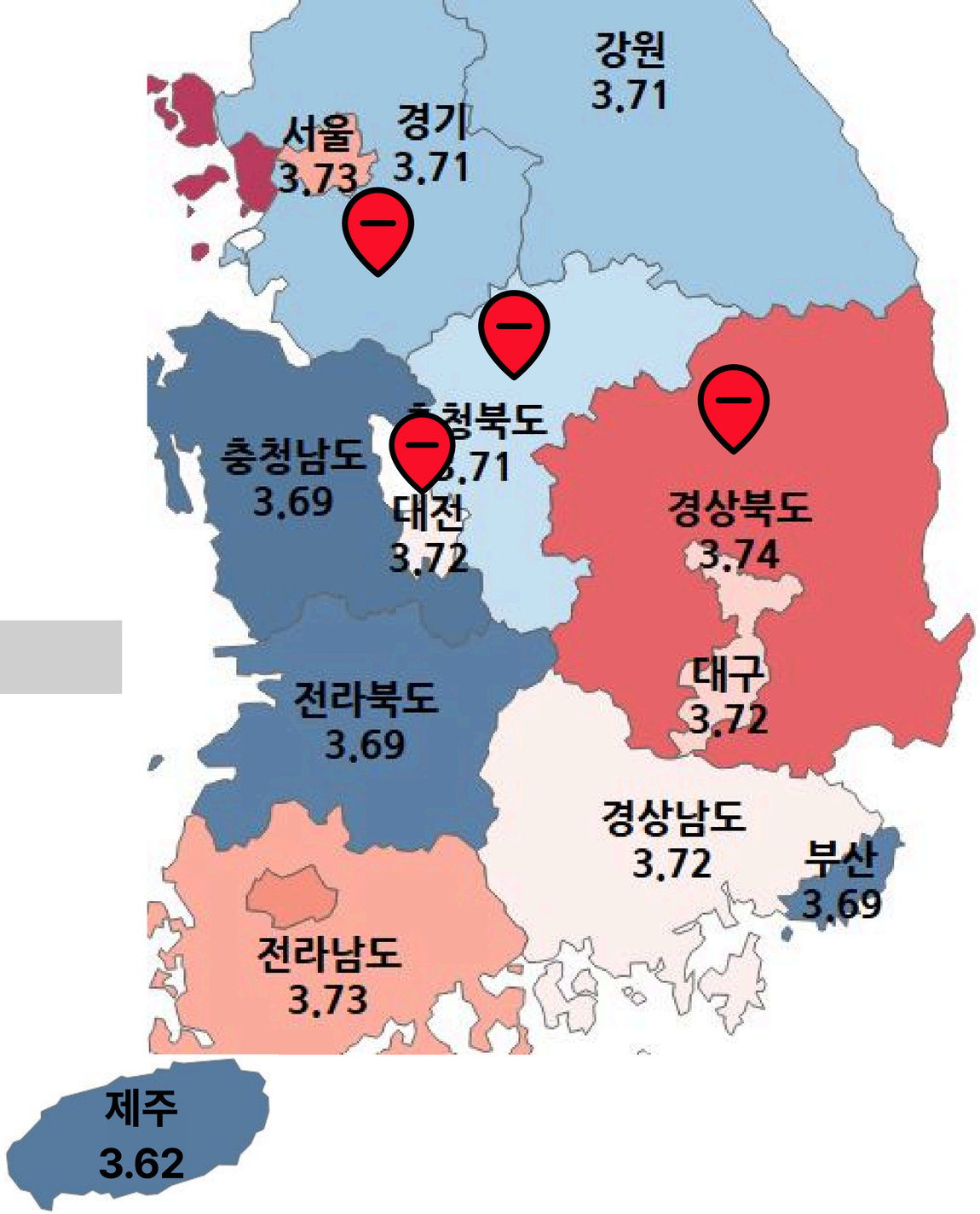
충주공장은 1971년에 설립된 유한킴벌리 제1공장인 군포공장이
2011년도에 충주지역으로 이전되어 신설된 세계 최고 수준의 위생용품 공장입니다.

특히 활성화고객의 주문은 생활용품과 기저귀 등의 생활필수품을 많이 주문하는 특성

배송 프로세스 구축 시, 공장과 물류센터 활용이 필요함

자사브랜드의 기저귀 평균 배송 소요시간
3.62일 ~ 3.74일

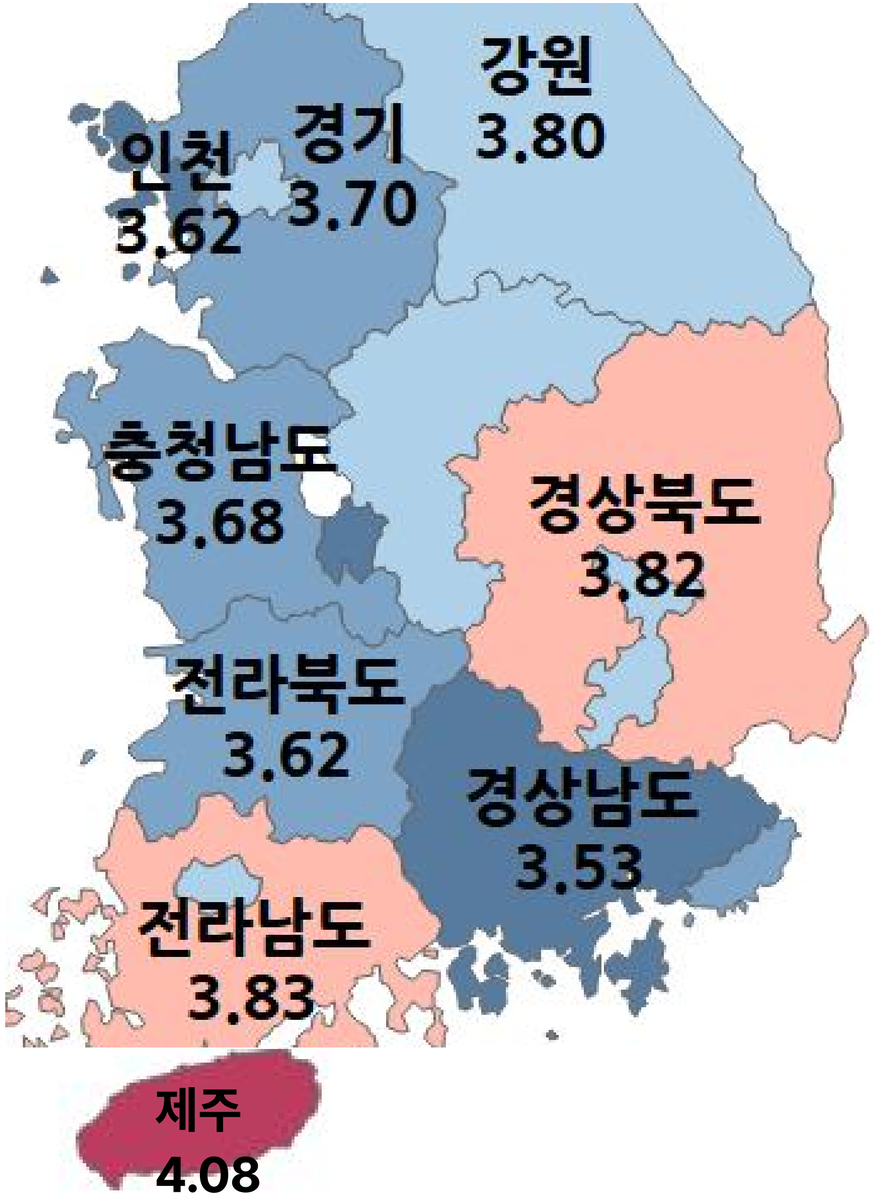
- 대전에서 아기 기저귀&아기 물티슈 등의 영유아 제품 생산
- 대전 인근의 배송일은 상대적으로 빠른 편에 속함
- 기저귀의 경우, 전반적으로 배송의 편차가 적지만,
  기저귀 생산 공장 근처는 특히 배송이 원활함



자사브랜드의 생활용품 평균 배송 소요시간
3.52일 ~4.08일

- 경상북도 김천에서 미용티슈, 화장지 등 생활용품 생산
- 생활용품공장이 위치한 경상북도에서 생활용품을 구매하였는데도 배송이 느린 편에 속함
- 대전 기저귀 공장을 긍정적인 사례로 벤치마킹해볼 수 있음

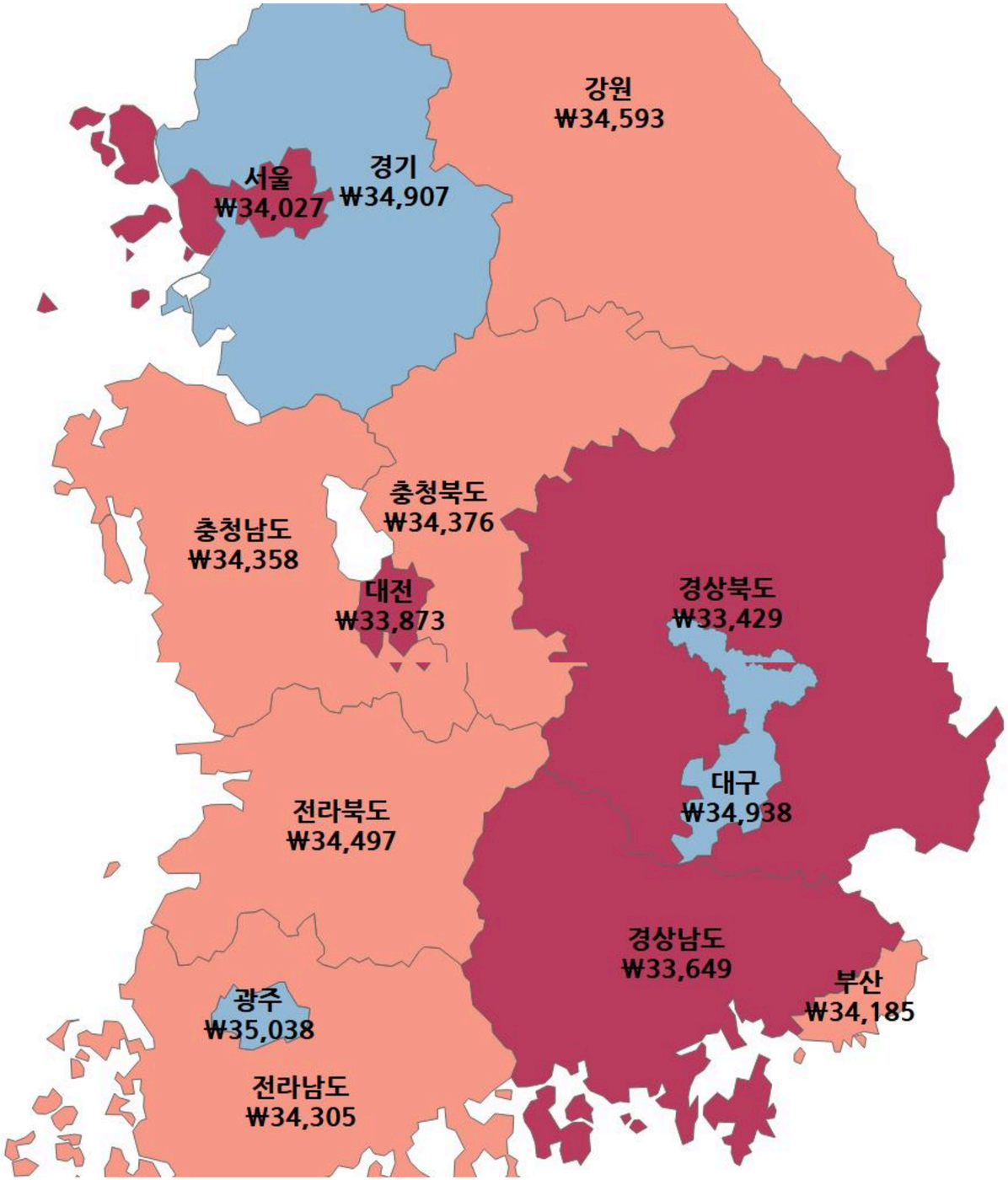
활성화 고객의 경우 생활용품 구매 비중이 높았는데
배송이 느린 경상북도, 전라남도, 제주 배송 프로세스 개선 필요



지역별 평균 주문금액

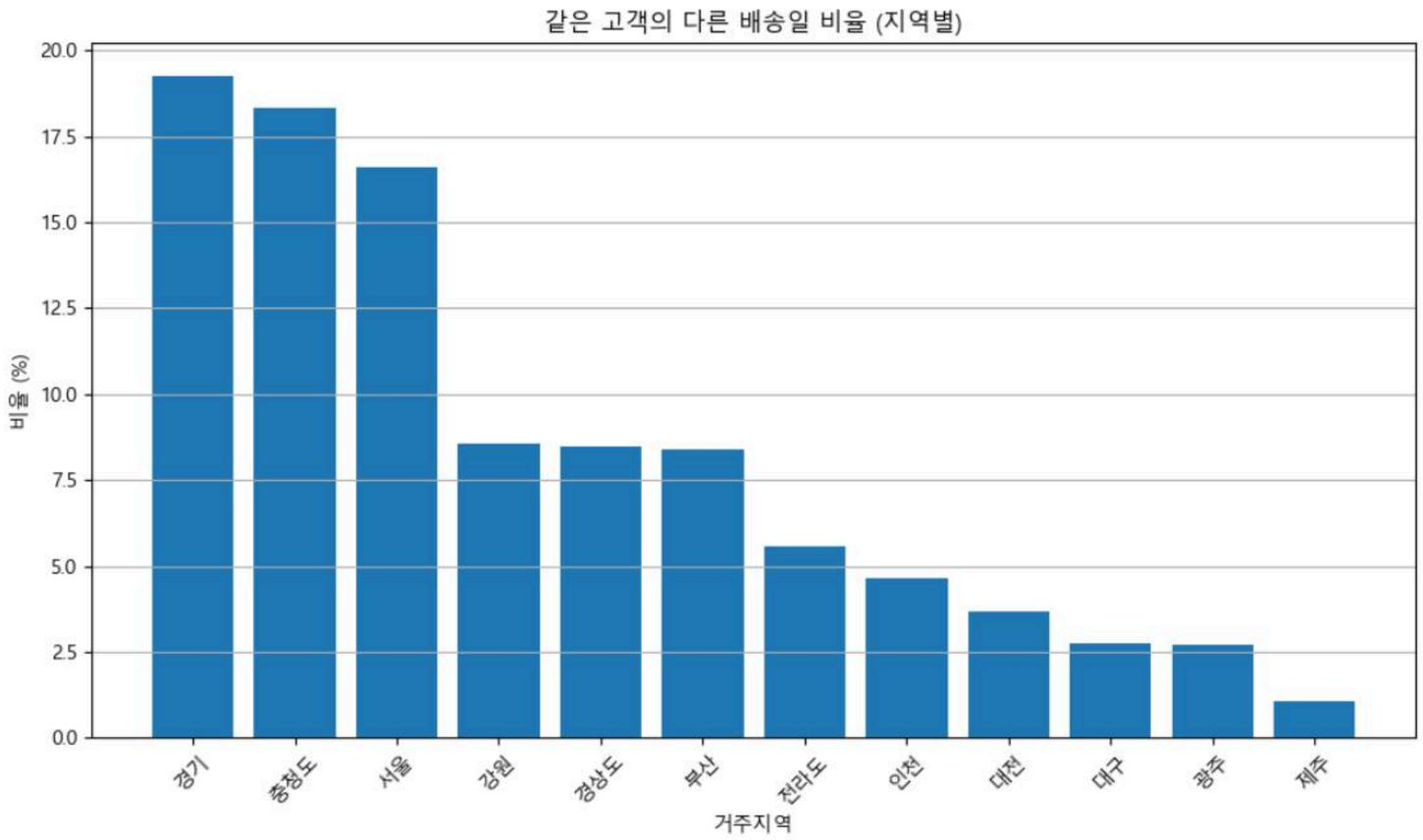
1위 제주

제주는 가장 평균 주문금액이 높은 지역이었는데
멀리 떨어져 있다보니 한 번에 주문을 많이 하는 것으로 추정



제주도 배송의 평가

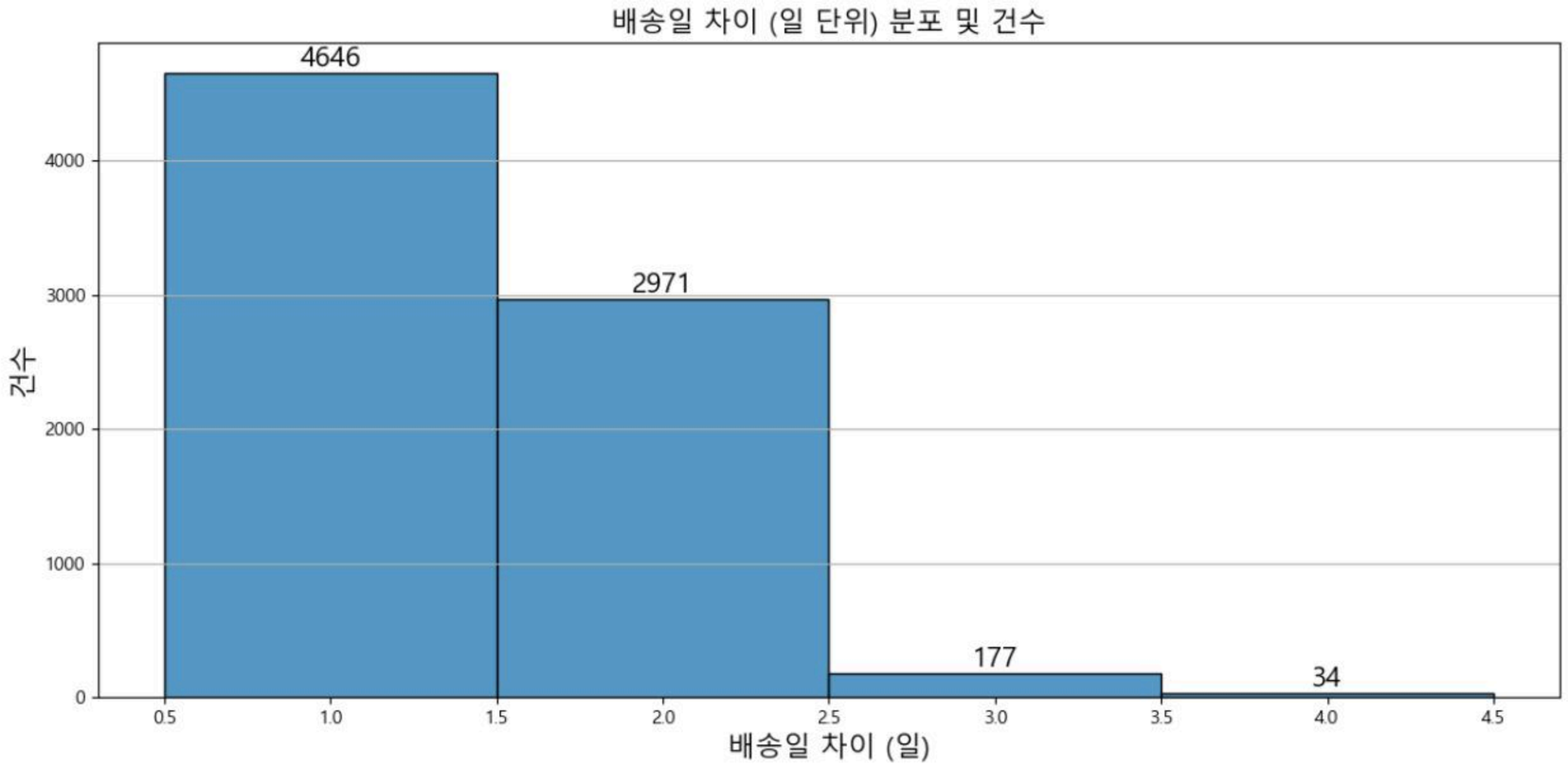
[지역별 쪼개기 배송 발생율]



제주도는 생활용품은 4일로 가장 느렸지만,
기저귀는 3.62일로 가장 빨랐고 평균 주문금액도 가장 높은 지역임
배를 건너야 하는 특성으로 배송이 느린 점을 고려하면 고객 관리와 배송 품질이 양호한 지역으로 평가함

오히려 물류센터가 위치한 경기도, 충청도의 쪼개기 배송이 가장 많이 발생함

제주도 외 지역의 배송 평가



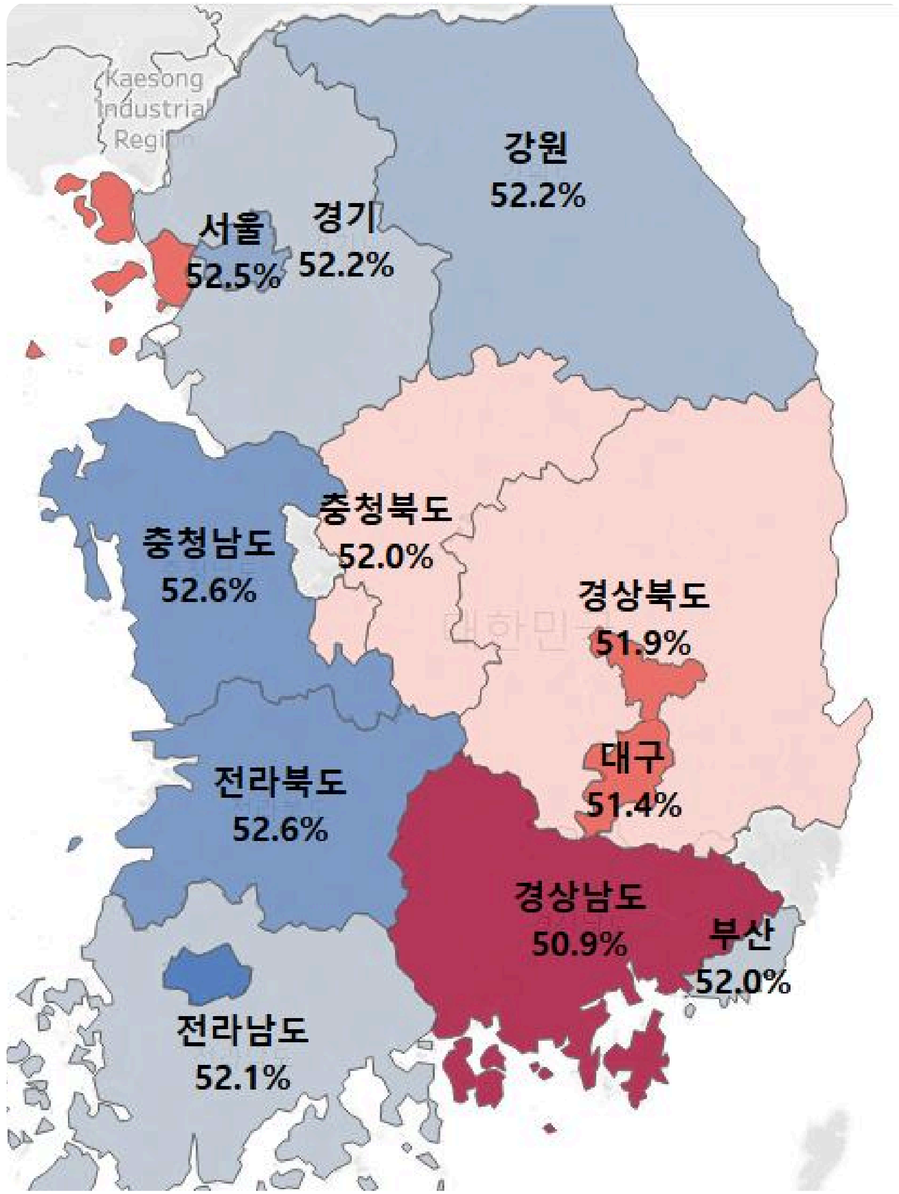
쪼개기 배송은 동일한 일자에 주문했음에도 서로 다른 날짜에 배송된 건으로 최대 4일 차이로 나뉘어 배송된 것으로 나타남

⇒ 2번 이상 택배 박스를 뜯는 작업 또는 예측 불가능한 배송으로 고객들의 불만이 있을 것으로 보이며 개선이 필요한 부분임

제주도 외 지역의 배송 지연

- 3일내 배송 건은 충청북도<경상북도<대구<경상남도 순으로 많으며, 경상도가 가장 적음

⇒ 경상도의 배송 불만이 보다 많을 것으로 판단



유한킴벌리, '대리점 공동물류센터' 설립 2014-07-16

유한킴벌리, '대리점 공동물류센터' 설립

- 제조업계 최초로 대리점 공동물류센터 개소
- 물류 효율 ↑, 물류비 감소 ↓의 혁신적 상생협력 모델

출처 : 유한킴벌리 공식사이트 보도자료

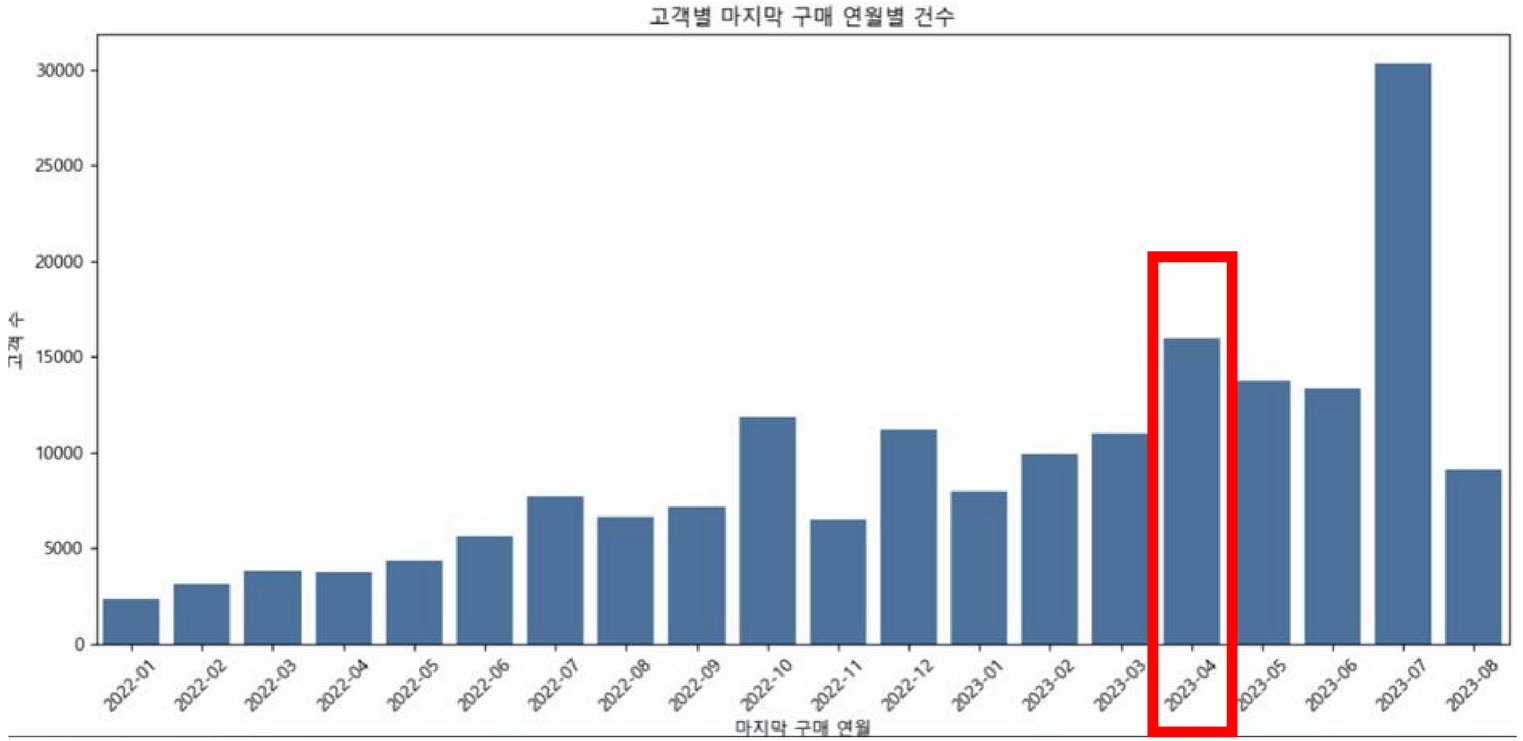
경상남도처럼 배송 프로세스 개선은 필요하지만,
물류센터를 추가적으로 구축하기 어렵다면 대리점과 물류센터를 결합해
물류 효율을 상승시킬 수 있음

적기 배송일 예측 모델

목적 | 쪼개기 배송 및 지역별 배송 편차로 인해 발생하는 배송일 예측의 불확실성 문제를 해소하고,
고객 불편을 줄이기 위한 적기 배송일 예측 모델을 구축

외부 데이터

- 2023 국내 우편 물류 시스템 데이터 사용 (https://www.kdx.kr/data/view/39665)



고객별 구매 월별 건수를 파악한 결과 2023년 7월이 가장 많은 것을 알 수 있음
하지만, 7월 데이터가 존재하지 않아 2번째로 많은 2023년 4월 데이터를 활용하기로 결정함

외부 데이터

2023 국내 우편 물류 시스템 데이터 사용 (https://www.kdx.kr/data/view/39665)

[배송 도착일]

	DELIV_DATE
0	2023-04-01
1	2023-04-01
2	2023-04-01
3	2023-04-01
4	2023-04-01

[배송 출발일]

RCV_DATE
2023-03-31
2023-03-31
2023-03-31
2023-03-31
2023-03-31

따라서, 2023년 4월 1일 ~ 2023년 4월 30일까지 총 30개의 데이터를 가져옴
이때, 데이터를 확인한 결과 2023년 4월에 배송이 출발한 것이 아닌 완료된 데이터임을 확인함

외부 데이터 전처리

```
rcv_raw = df["RCV_DATE"].astype(str).str.strip()
print("\n🔍 RCV_DATE 원시 값 샘플:")
print(rcv_raw.unique()[:20])
```

```
🔍 RCV_DATE 원시 값 샘플:
['20230428.0' '20230427.0' '20230421.0']
```

배송 출발일과 배송 도착일이 둘 다 20230401 형식으로 되어있는 것을 확인
따라서, datetime으로 바꿈
또한, 배송 출발일에서 결측이 존재해 데이터 양과 메모리를 고려하여 drop하기로 결정함

외부 데이터 전처리

T	U	V	W	X	Y
DELIV_CTPV_NM	DELIV_CTGG_NM	DELIV_ADSTRD	EUPD_IVNM		
경기도	안산시 단원구	고잔동	인천광역시 중구 운서동		
경기도	안산시 단원구	고잔동	충청남도 천안시 동남구 청당동		
경기도	안산시 단원구	고잔동	서울특별시 강서구 화곡1동		
경기도	안산시 단원구	고잔동	인천광역시 중구 운서동		
경기도	안산시 단원구	고잔동	경기도 평택시 비전동		
경기도	안산시 단원구	고잔동	경기도 평택시 비전동		
경기도	안산시 단원구	고잔동	경기도 성남시 수정구 고등동		
경기도	안산시 단원구	고잔동	서울특별시 동대문구 신설동		
경기도	안산시 단원구	고잔동	인천광역시 계양구 계산4동		
경기도	안산시 단원구	고잔동	전라북도 군산시 중앙로1가동		
경기도	안산시 단원구	고잔동	경기도 용인시 수지구 풍덕천2동		
경기도	안산시 단원구	고잔동	서울특별시 종로구 종로2가		
서울특별시	성북구	장위2동	인천광역시 남동구 논현동		
서울특별시	성북구	장위3동	경상남도 남해군 서면 서상리		
서울특별시	성북구	장위1동	인천광역시 중구 운서동		
서울특별시	성북구	장위1동	경기도 평택시 비전동		
서울특별시	성북구	장위3동	경기도 평택시 비전동		
제주특별자치도	제주시	노형동	서울특별시 관악구 신림동		

배송 출발지 열은 DELIV_CTPV_NM→시도, DELIV_CTGG_NM→시군구, DELIG_ADSTRD→행정동을 나타냄
하지만, 배송 도착지 열은 EUPD_IVNM으로 한개의 열에 모든 주소 정보지가 포함되어 있는 것을 확인함
따라서, 배송 출발지의 3개 열을 도착지 주소 형태에 맞춰 1개의 형태로 맞춤

외부 데이터 병합

```
📎 병합 완료: 20230403_정제.csv (누적 행 수: 2011019)
📎 병합 완료: 20230404_정제.csv (누적 행 수: 4407104)
📎 병합 완료: 20230405_정제.csv (누적 행 수: 6589844)
📎 병합 완료: 20230406_정제.csv (누적 행 수: 9206893)
📎 병합 완료: 20230407_정제.csv (누적 행 수: 12116319)
C:\Users\hdemn\AppData\Local\Temp\ipykernel_144424\374219
_memory=False.
  df = pd.read_csv(file_path, encoding='utf-8-sig')
📎 병합 완료: 20230408_정제.csv (누적 행 수: 12746719)
📎 병합 완료: 20230409_정제.csv (누적 행 수: 12746733)
📎 병합 완료: 20230410_정제.csv (누적 행 수: 15200991)
📎 병합 완료: 20230411_정제.csv (누적 행 수: 18128583)
📎 병합 완료: 20230412_정제.csv (누적 행 수: 20455782)
📎 병합 완료: 20230413_정제.csv (누적 행 수: 22884735)
📎 병합 완료: 20230414_정제.csv (누적 행 수: 25399589)
C:\Users\hdemn\AppData\Local\Temp\ipykernel_144424\374219
_memory=False.
  df = pd.read_csv(file_path, encoding='utf-8-sig')
📎 병합 완료: 20230415_정제.csv (누적 행 수: 26011236)
📎 병합 완료: 20230416_정제.csv (누적 행 수: 26011322)
📎 병합 완료: 20230417_정제.csv (누적 행 수: 28160764)
📎 병합 완료: 20230418_정제.csv (누적 행 수: 31001128)
📎 병합 완료: 20230419_정제.csv (누적 행 수: 33290732)
```

```
📎 병합 완료: 20230422_정제.csv (누적 행 수: 38210138)
📎 병합 완료: 20230423_정제.csv (누적 행 수: 38210288)
📎 병합 완료: 20230424_정제.csv (누적 행 수: 39702724)
📎 병합 완료: 20230425_정제.csv (누적 행 수: 42115369)
📎 병합 완료: 20230426_정제.csv (누적 행 수: 44151856)
📎 병합 완료: 20230427_정제.csv (누적 행 수: 46268534)
📎 병합 완료: 20230428_정제.csv (누적 행 수: 48322982)
C:\Users\hdemn\AppData\Local\Temp\ipykernel_144424\37421
_memory=False.
  df = pd.read_csv(file_path, encoding='utf-8-sig')
📎 병합 완료: 20230429_정제.csv (누적 행 수: 48946195)
📎 병합 완료: 20230430_정제.csv (누적 행 수: 48946400)

📅 날짜 변환 결과:
RCV_DATE null 수: 0
DELIV_DATE null 수: 0
```

전처리 완료한 2023년 4월 총 30개의 csv를 하나의 파일로 병합하였음

병합 후 데이터 정보

```
✅ 최종 병합 및 날짜 처리 완료!
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 48946400 entries, 0 to 48946399
Data columns (total 23 columns):
 #   Column              Dtype
---  ------              -----
 0   DELIV_DATE          datetime64[ns]
 1   DELIV_PO_REGI_POCD  int64
 2   DISC_NO             int64
 3   DELIV_AREA_NO       int64
 4   MAIL_KIND_SP_CD     int64
 5   MAIL_SP_CD          int64
 6   SPCL_TRT_CD         float64
 7   DOM_EM_YN           object
 8   RCV_PO_REGI_POCD    float64
 9   VST_PRCL_YN         object
 10  ELECT_MAIL_YN       object
 11  CNTR_REGI_YN        object
 12  PRSN_APPO_YN        object
 13  CONG_MAIL_SP_CD     float64
 14  MAIL_WGHT           float64
 15  MAIL_VOLM           float64
 16  RCV_DATE            datetime64[ns]
 17  SEMI_REGI_YN        object
 18  BLD_MNGM_NO         object
 19  DELIV_CTPV_NM       object
 20  DELIV_CTGG_NM       object
 21  DELIV_ADSTRD_NM     object
 22  EUPD_IVNM           object
dtypes: datetime64[ns](2), float64(5), int64(5), object(11)
memory usage: 8.4+ GB
```

	DELIV_DATE	DELIV_PO_REGI_POCD	DISC_NO	DELIV_AREA_NO	MAIL_KIND_SP_CD	MAIL_SP_CD	SPCL_TRT_CD	DOM_EM_YN	RCV_PO_REGI_POCD	VST_P
0	2023-04-01	40086	6064895955535	43501	1	2	1.0	N	10248.0	
1	2023-04-01	40086	6865289638668	43501	1	2	1.0	N	35656.0	
2	2023-04-01	40086	6892037503973	43501	1	2	1.0	N	14559.0	
3	2023-04-01	40086	6064889866430	43501	1	2	1.0	N	10248.0	
4	2023-04-01	40086	6096359238210	43501	1	2	1.0	N	40308.0	
...	...	...	...	...	...	...	...	...	...	
48946395	2023-04-30	41627	6861500034808	43006	1	2	1.0	N	10186.0	
48946396	2023-04-30	41627	7854151289393	43006	1	2	1.0	N	41627.0	
48946397	2023-04-30	41627	7493702010161	43006	1	2	1.0	N	49374.0	

병합 후 데이터 정보

```
merged_df.describe()
```

	DELIV_DATE	DELIV_PO_REGI_POCD	DISC_NO	DELIV_AREA_NO	MAIL_KIND_SP_CD	MAIL_SP_CD
count	48946400	4.894640e+07	4.894640e+07	4.894640e+07	4.894640e+07	4.894640e+07
mean	2023-04-15 03:43:26.351040256	3.937026e+04	3.681630e+12	2.435491e+04	1.597996e+00	1.402004e+00
min	2023-04-01 00:00:00	1.001500e+04	1.000000e+12	1.000000e+01	1.000000e+00	1.000000e+00
25%	2023-04-08 00:00:00	2.014600e+04	1.105804e+12	1.105400e+04	1.000000e+00	1.000000e+00
50%	2023-04-14 00:00:00	4.017500e+04	2.583201e+12	1.160400e+04	2.000000e+00	1.000000e+00
75%	2023-04-21 00:00:00	5.604800e+04	6.096360e+12	4.300800e+04	2.000000e+00	2.000000e+00
max	2023-04-30 00:00:00	8.609100e+04	8.740070e+12	8.299900e+04	2.000000e+00	2.000000e+00

RCV_DATE	배송소요일
48946400	4.894640e+07
2023-04-12 17:55:53.606850560	2.408018e+00
2020-07-16 00:00:00	0.000000e+00
2023-04-06 00:00:00	1.000000e+00
2023-04-12 00:00:00	2.000000e+00
2023-04-19 00:00:00	3.000000e+00
2023-04-28 00:00:00	1.016000e+03

병합한 데이터의 기술 통계를 파악하였음

배송 데이터 소개

```
RangeIndex: 199582 entries, 0 to 199581
Data columns (total 22 columns):
 #   Column   Non-Null Count   Dtype
---  ------   --------------   -----
 0   주문번호     199582 non-null  object
 1   배송완료일    199582 non-null  object
 2   배송시작일    199582 non-null  object
 3   구매일      199582 non-null  object
 4   구매시월령    167839 non-null  float64
 5   고객번호     199582 non-null  float64
 6   구매금액     199582 non-null  int64
 7   결제금액     199582 non-null  int64
 8   물품대분류    199582 non-null  object
 10  성별       199582 non-null  object
 11  결혼유무     199582 non-null  object
 12  거주지역     199582 non-null  object
 13  연령       199582 non-null  int64
 14  결제등록카드   199582 non-null  object
 15  유입경로     199582 non-null  object
 16  자녀여부     199582 non-null  object
 17  출산예정     199582 non-null  int64
 18  배송소요일    199582 non-null  int64
 19  총구매소요일   199582 non-null  int64
 20  지연배송     199582 non-null  int64
 21  휴일       199582 non-null  int64
dtypes: float64(2), int64(8), object(12)
```

	구매시월령	고객번호	구매금액	결제금액	연령	출산예정	배송소요일	총구매소요일	지연배송	휴일
count	167839.000000	1.995820e+05	199582.000000	199582.000000	199582.000000	199582.000000	199582.000000	199582.000000	199582.000000	199582.000000
mean	14.852067	2.019052e+14	30570.077612	21712.598080	32.816677	0.072993	2.730412	3.730412	0.201982	0.192517
std	38.531947	4.780635e+09	30555.336378	21337.720041	2.946891	0.260125	0.902544	0.902544	0.401480	0.394278
min	-10.000000	2.018123e+14	0.000000	0.000000	20.000000	0.000000	2.000000	3.000000	0.000000	0.000000
25%	2.000000	2.019022e+14	7400.000000	6000.000000	31.000000	0.000000	2.000000	3.000000	0.000000	0.000000
50%	6.000000	2.019043e+14	17500.000000	12602.000000	33.000000	0.000000	2.000000	3.000000	0.000000	0.000000
75%	13.000000	2.019081e+14	51200.000000	34799.000000	35.000000	0.000000	3.000000	4.000000	0.000000	0.000000
max	1439.000000	2.019123e+14	598000.000000	539000.000000	46.000000	1.000000	6.000000	7.000000	1.000000	1.000000

배송 데이터는 Sales_data와 Members_data를 결합한 데이터로 기술 통계를 파악함

⭐ 외부데이터와 배송 데이터 결합 사용 전략

외부 데이터와 배송 데이터를 활용하여 AutoML 기반 배송 도착 예정일 예측기를 만들고자 함
이때, 총 3가지 버전을 만들어 예측을 시도하였음

1. 단순 날짜 예측 함수를 구현
2. LightGBM과 결합하여 반 자동 도착 예정일 예측 시스템을 구현
3. AutoML 중 H2O 모델을 사용하여 배송 도착 예정일 예측 시스템을 구현

따라서, 외부 데이터로 각각의 모델을 만든 후 배송 데이터에 대입하여 결과를 파악하고 성능을 평가하고자 함

1. 단순 날짜 예측 함수를 구현

```python
# 1. 배송소요일 계산 완료된 merged_df 기준
merged_df['배송소요일'] = (merged_df['DELIV_DATE'] - merged_df['RCV_DATE']).dt.days

# 2. 출발지-도착지 조합별 평균 소요일 계산
route_avg_days = (
    merged_df
    .groupby(['DELIV_CTPV_NM', 'DELIV_CTGG_NM', 'EUPD_IVNM'])['배송소요일']
    .mean()
    .reset_index(name='평균소요일')
)

# 3. 예측 함수 정의
def 예상_배송일자_예측(deliv_ctpv, deliv_ctgg, eupd_ivnm, 발송일):
    발송일 = pd.to_datetime(발송일)

    # 해당 경로의 평균 소요일 불러오기
    row = route_avg_days[
        (route_avg_days['DELIV_CTPV_NM'] == deliv_ctpv) &
        (route_avg_days['DELIV_CTGG_NM'] == deliv_ctgg) &
        (route_avg_days['EUPD_IVNM'] == eupd_ivnm)
    ]

    if not row.empty:
        days = row['평균소요일'].values[0]
    else:
        days = merged_df['배송소요일'].mean()  # 해당 경로 없음 → 전국 평균 사용

    예측일 = 발송일 + pd.Timedelta(days=round(days))
    return 예측일

# 4. 예시 사용
예상일자 = 예상_배송일자_예측("서울특별시", "강남구", "부산광역시 해운대구", "2025-06-23")
print(f"📦 예상 도착일: {예상일자.date()}")
```

```python
# 4. 예시 사용
예상일자 = 예상_배송일자_예측("서울특별시", "강남구", "부산광역시 해운대구", "2025-06-23")
print(f"📦 예상 도착일: {예상일자.date()}")
```

```
📦 예상 도착일: 2025-06-25
```

외부 데이터를 활용해 날짜 예측 함수를 만들었으며,
임의로 값을 넣은 결과 예상 도착일이 정상적으로 잘 나오는 것을 확인함

1. 단순 날짜 예측 함수를 구현

```python
# 3. 예측 함수 정의
def 예상_배송일자_예측(deliv_ctpv, deliv_ctgg, eupd_ivnm, 발송일):
    발송일 = pd.to_datetime(발송일)
    row = route_avg_days[
        (route_avg_days['DELIV_CTPV_NM'] == deliv_ctpv) &
        (route_avg_days['DELIV_CTGG_NM'] == deliv_ctgg) &
        (route_avg_days['EUPD_IVNM'] == eupd_ivnm)
    ]

    if not row.empty:
        days = row['평균소요일'].values[0]
    else:
        days = merged_df['배송소요일'].mean()

    return 발송일 + pd.Timedelta(days=days)

# 1. 랜덤 샘플 1000개 추출 (시드 고정)
sample_df = merged_df.sample(n=1000, random_state=42).copy()

# 2. 예상 배송완료일 계산
sample_df['예상_배송완료일'] = sample_df.apply(
    lambda row: 예상_배송일자_예측(
        row['DELIV_CTPV_NM'],
        row['DELIV_CTGG_NM'],
        row['EUPD_IVNM'],
        row['RCV_DATE']
    ), axis=1
)

# 3. 예상 배송소요일 계산
sample_df['예상_배송소요일'] = (sample_df['예상_배송완료일'] - sample_df['RCV_DATE']).dt.days
```

임의로 값을 넣는 것이 아닌 외부 데이터 전체를 가지고 예상 도착일을 도출하는 함수를 구현함
이때, 메모리 초과로 1000개로 외부 데이터를 샘플링하여 시도하였음

1. 단순 날짜 예측 함수를 구현

```
15453888    1
40990724    3
47691680    2
23146285    1
20020085    3
           ..
38297160    3
26738637    2
40453464    2
19864524    1
24230713    3
Name: 예상_배송소요일, Length: 1000, dtype: int64
```

```
📊 예측 성능 (샘플 1000개 기준)
MAE : 1.11일
RMSE: 2.00일
```

그 결과, 예상 배송 소요일이 나오며 모델 예측 성능은 다음과 같음

1. 단순 날짜 예측 함수를 구현

	배송시작일	거주지역	예상_배송완료일
13864	2022-03-12	경기	2022-03-13 09:55:48.125994731
15760	2022-03-17	경상도	2022-03-19 10:02:29.894593469
17995	2022-03-24	부산	2022-03-25 11:15:50.614946314
19489	2022-03-29	경기	2022-04-01 07:35:22.129416070
21884	2022-04-03	충청도	2022-04-05 05:46:44.847959090
...	...	...	...
192713	2023-07-09	서울	2023-07-11 17:20:38.779837528
194514	2023-07-20	충청도	2023-07-23 00:11:50.699003537
194625	2023-07-21	충청도	2023-07-24 02:49:30.854039205
196056	2023-07-27	강원	2023-07-28 22:50:50.153497958
199503	2023-08-09	강원	2023-08-12 05:26:15.170612111

📊 성능 평가 결과 (샘플 100개 기준)
MAE (평균 절대 오차): 1.04일
RMSE (평균 제곱근 오차): 1.37일

외부 데이터 기반으로 만든 예측 함수을 배송 데이터에 대입하여 결과를 확인하고자 함
그 결과, **배송 시작일과 거주지역을 고려한 예상 배송 완료일이 결과**로 나왔으며, 예측 성능 수치는 다음과 같음

2. LightGBM과 결합하여 반 자동 도착 예정일 예측 시스템 구현

단순 예측 함수에서 발전시켜 LightGBM과 결합한 반 자동 도착 예정일 예측 시스템을 구현하고자 함
이 역시 외부 데이터 기반으로 예측 모델을 만든 후 배송 데이터에 넣어 결과를 확인하고자 함

이때, 모델의 정확성을 높이기 위해 기존 변수였던 배송 출발일, 도착일, 거주지역에 더하여
추가로 배송 소요일, 년도, 월, 일, 요일 등 변수를 추가하여 실행하였음

2. LightGBM과 결합하여 반 자동 도착 예정일 예측 시스템 구현

```python
merged_df['배송소요일'] = (merged_df['DELIV_DATE'] - merged_df['RCV_DATE']).dt.days

# 2 날짜 파생 변수 생성
merged_df['deliv_year'] = merged_df['DELIV_DATE'].dt.year
merged_df['deliv_month'] = merged_df['DELIV_DATE'].dt.month
merged_df['deliv_dayofweek'] = merged_df['DELIV_DATE'].dt.dayofweek

# 3 지역 인코딩
le_ctpv = LabelEncoder().fit(merged_df['DELIV_CTPV_NM'])
le_ctgg = LabelEncoder().fit(merged_df['DELIV_CTGG_NM'])
le_dest = LabelEncoder().fit(merged_df['EUPD_IVNM'].astype(str))

merged_df['o_ctpv'] = le_ctpv.transform(merged_df['DELIV_CTPV_NM'])
merged_df['o_ctgg'] = le_ctgg.transform(merged_df['DELIV_CTGG_NM'])
merged_df['dest'] = le_dest.transform(merged_df['EUPD_IVNM'].astype(str))

# 4 학습 데이터 구성
features = ['o_ctpv', 'o_ctgg', 'dest', 'deliv_year', 'deliv_month', 'deliv_dayofweek']
X = merged_df[features]
y = merged_df['배송소요일']

X_train, X_val, y_train, y_val = train_test_split(X, y, test_size=0.2, random_state=42)

train_data = lgb.Dataset(X_train, y_train)
val_data = lgb.Dataset(X_val, y_val, reference=train_data)

params = {
    'objective': 'regression',
    'metric': ['l1', 'l2'],
    'boosting_type': 'gbdt',
    'learning_rate': 0.1,
    'num_leaves': 31,
```

```python
# 8 예시 실행
예상일자 = 배송예측_모델("서울특별시", "강남구", "부산광역시 해운대구", "2025-06-23")
if 예상일자 is not None:
    print(f"📦 모델 기반 예상 도착일: {예상일자.date()}")
else:
    print("⚠️ 예측 실패: 결과가 없습니다.")
```

```
MAE: 1.0382979941077102
RMSE: 2.2191778644082305
📦 모델 기반 예상 도착일: 2025-06-27
```

먼저, 임의로 값을 넣어 바로 결과를 예측하는 방식으로 실행을 시도해 결과를 확인함

2. LightGBM과 결합하여 반 자동 도착 예정일 예측 시스템 구현

```python
feature_cols = ['DELIV_CTPV_NM', 'DELIV_CTGG_NM', 'EUPD_IVNM', 'deliv_year', 'deliv_month', 'deliv_dayofweek']
X = merged_df[feature_cols]
y = merged_df['배송소요일']

X_train, X_val, y_train, y_val = train_test_split(X, y, test_size=0.2, random_state=42)

train_data = lgb.Dataset(X_train, label=y_train)
valid_data = lgb.Dataset(X_val, label=y_val)

params = {
    'objective': 'regression',
    'metric': 'rmse',
    'verbosity': -1,
    'seed': 42
}

#model = lgb.train(params, train_data, valid_sets=[valid_data], early_stopping_rounds=50)

model = lgb.train(params, train_data, valid_sets=[valid_data])

# 모델 저장
joblib.dump(model, 'lightgbm_delivery_model.pkl')
print("모델 저장 완료")
```

```
모델 저장 완료
```

```python
# 원본 데이터 로드 및 전처리 (merged_df 가 있다고 가정)
merged_df['배송소요일'] = (merged_df['DELIV_DATE'] - merged_df['RCV_DATE']).dt.days
merged_df['deliv_year'] = merged_df['RCV_DATE'].dt.year
merged_df['deliv_month'] = merged_df['RCV_DATE'].dt.month
merged_df['deliv_dayofweek'] = merged_df['RCV_DATE'].dt.dayofweek

# 동일한 label encoding 수행 (중요!)
cat_cols = ['DELIV_CTPV_NM', 'DELIV_CTGG_NM', 'EUPD_IVNM']
for col in cat_cols:
    le = LabelEncoder()
    merged_df[col] = le.fit_transform(merged_df[col])
    # 주의: 실제로는 train 때 쓴 인코더 저장 후 재사용해야 함

feature_cols = ['DELIV_CTPV_NM', 'DELIV_CTGG_NM', 'EUPD_IVNM', 'deliv_year', 'deliv_month', 'deliv_dayofweek']

# 모델 불러오기
model = joblib.load('lightgbm_delivery_model.pkl')

# 예측
merged_df['예상_배송소요일'] = model.predict(merged_df[feature_cols], num_iteration=model.best_iteration)
merged_df['예상_배송완료일'] = merged_df['RCV_DATE'] + pd.to_timedelta(merged_df['예상_배송소요일'].round(), unit='D')

# 성능 평가
y_true = merged_df['배송소요일']
y_pred = merged_df['예상_배송소요일']
print("MAE:", mean_absolute_error(y_true, y_pred))
print("RMSE:", np.sqrt(mean_squared_error(y_true, y_pred)))

# 결과 일부 출력
print(merged_df[['RCV_DATE','예상_배송소요일', '예상_배송완료일']].head())
```

앞에서 결과를 본 후 파라미터를 다르게 설정하여 모델을 실행하고 저장하기로 판단함

2. LightGBM과 결합하여 반 자동 도착 예정일 예측 시스템 구현

```
MAE: 1.0777442768882373
RMSE: 1.7833690261765462
```

	RCV_DATE	예상_배송소요일	예상_배송완료일
0	2023-03-31	1.907007	2023-04-02
1	2023-03-31	2.617619	2023-04-03
2	2023-03-31	3.311753	2023-04-03
3	2023-03-31	1.907007	2023-04-02
4	2023-03-31	2.413816	2023-04-02

저장한 모델을 불러와 외부 데이터 기반 날짜 예측 시스템을 구현하고 결과를 확인함

2. LightGBM과 결합하여 반 자동 도착 예정일 예측 시스템 구현

```python
# 9. 결과 확인
print(df[['배송시작일', '거주지역', '예상_배송소요일', '예상_배송완료일']].head())
```

```
       배송시작일 거주지역  예상_배송소요일  예상_배송완료일
0 2022-01-02   서울   4.91377 2022-01-07
1 2022-01-02  충청도   4.91377 2022-01-07
2 2022-01-02   서울   4.91377 2022-01-07
3 2022-01-02   서울   4.91377 2022-01-07
4 2022-01-02   강원   4.91377 2022-01-07
```

```python
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
import numpy as np

# 실제 배송 완료일 컬럼이 있다고 가정
# 실제 배송소요일 계산
df['배송소요일'] = (pd.to_datetime(df['배송완료일']) - df['배송시작일']).dt.days

# MAE 계산
mae = mean_absolute_error(df['배송소요일'], df['예상_배송소요일'])

# RMSE 계산
rmse = np.sqrt(mean_squared_error(df['배송소요일'], df['예상_배송소요일']))

print(f"MAE: {mae:.3f}")
print(f"RMSE: {rmse:.3f}")
```

```
MAE: 40.503
RMSE: 54.175
```

앞서 만든 예측 모델을 활용하여 배송 데이터에 넣었을 때 얼마나 잘 예측하는지를 평가하고자 함

2. LightGBM과 결합하여 반 자동 도착 예정일 예측 시스템 구현

```
df['거주지역'].unique()
```

```
array(['서울', '충청도', '강원', '제주', '인천', '대구', '경기', '부산', '경상도', '광주',
       '전라도', '대전'], dtype=object)
```

실행한 결과 예측 성능이 안 좋은 것을 확인함

또한, 배송 데이터에는 거주 지역이 시도 내용 총 17개만 있을 뿐 시군구, 상세 주소에 대한 정보가 없어 데이터 다양성 부족으로 예측 성능이 떨어지는 것으로 추측됨

따라서, 2번 버전은 더 이상의 성능 개선이 어렵다고 판단하고 3번 버전인 AutoML을 시도하고자 함

3. AutoML 중 H2O 모델을 사용하여 배송 도착 예정일 예측 시스템 구현

해당 버전도 똑같이 외부 데이터로 학습 및 AutoML 기반 예측 모델을 만든 후 배송 데이터를 통해 테스트하고자 함
먼저, 외부 데이터로 학습을 진행함

이때, 이 버전 또한 모델의 정확성을 높이기 위해 기존 변수였던 배송 출발일, 도착일, 거주지역에 더하여
추가로 배송 소요일, 년도, 월, 일, 요일 등 변수를 추가하여 실행하였음
또한, AutoML 메모리 이슈로 H2O 모델을 4GB로 초기화 한 후 실행하였음

3. AutoML 중 H2O 모델을 사용하여 배송 도착 예정일 예측 시스템 구현

```python
# JVM 메모리 8GB로 초기화
h2o.init(max_mem_size="4G")

merged_df['DELIV_DATE'] = pd.to_datetime(merged_df['DELIV_DATE'], errors='coerce')
merged_df['RCV_DATE'] = pd.to_datetime(merged_df['RCV_DATE'], errors='coerce')

# ✅ 확인
print(merged_df[['DELIV_DATE', 'RCV_DATE']].dtypes)

# (샘플 데이터로 줄이기)
merged_sample = merged_df.sample(frac=0.1, random_state=42).copy()

# 배송소요일 계산 및 날짜 특성 추가
merged_sample['배송소요일'] = (merged_sample['DELIV_DATE'] - merged_sample['RCV_DATE']).dt.days
merged_sample['deliv_year'] = merged_sample['DELIV_DATE'].dt.year
merged_sample['deliv_month'] = merged_sample['DELIV_DATE'].dt.month
merged_sample['deliv_dayofweek'] = merged_sample['DELIV_DATE'].dt.dayofweek

# H2OFrame 생성
h2o_df = h2o.H2OFrame(merged_sample[['DELIV_CTPV_NM','DELIV_CTGG_NM','EUPD_IVNM',
                                     'deliv_year','deliv_month','deliv_dayofweek','배송소요일']])

# AutoML 실행 (5분 제한)
aml = H2OAutoML(max_models=20, seed=42, max_runtime_secs=300, sort_metric="MAE")
aml.train(x=['DELIV_CTPV_NM','DELIV_CTGG_NM','EUPD_IVNM','deliv_year','deliv_month','deliv_dayofweek'],
          y='배송소요일', training_frame=h2o_df)

print(aml.leaderboard.head())
aml.leader.save_mojo(path="h2o_best_model.mojo")
```

H2O_cluster_uptime:	02 secs
H2O_cluster_timezone:	Asia/Seoul
H2O_data_parsing_timezone:	UTC
H2O_cluster_version:	3.46.0.7
H2O_cluster_version_age:	2 months and 27 days
H2O_cluster_name:	H2O_from_python_hdemn_378qrq
H2O_cluster_total_nodes:	1
H2O_cluster_free_memory:	3.980 Gb
H2O_cluster_total_cores:	0
H2O_cluster_allowed_cores:	0
H2O_cluster_status:	locked, healthy
H2O_connection_url:	http://127.0.0.1:54321
H2O_connection_proxy:	{"http": null, "https": null}
H2O_internal_security:	False
Python_version:	3.12.3 final

3. AutoML 중 H2O 모델을 사용하여 배송 도착 예정일 예측 시스템 구현

```
|| (done) 100%
```

model_id	mae	rmse	mse	rmsle	mean_residual_deviance
GBM_1_AutoML_1_20250623_231016	0.96149	2.1096	4.4504	0.366462	4.4504
GBM_4_AutoML_1_20250623_231016	0.986353	2.12316	4.5078	nan	4.5078
DeepLearning_1_AutoML_1_20250623_231016	0.988565	2.17383	4.72553	0.374743	4.72553
GBM_3_AutoML_1_20250623_231016	1.01119	2.13905	4.57553	0.376314	4.57553
GBM_2_AutoML_1_20250623_231016	1.02124	2.14296	4.59227	0.378449	4.59227
GBM_5_AutoML_1_20250623_231016	1.03841	2.15136	4.62834	0.383066	4.62834
DRF_1_AutoML_1_20250623_231016	1.08093	2.17859	4.74625	0.395082	4.74625
GBM_grid_1_AutoML_1_20250623_231016_model_1	1.08133	2.18738	4.78462	0.395851	4.78462
XRT_1_AutoML_1_20250623_231016	1.15596	2.29347	5.25999	0.424397	5.25999
GLM_1_AutoML_1_20250623_231016	1.17567	2.23175	4.98071	nan	4.98071

[10 rows x 6 columns]

H2O 모델을 돌린 결과 성능이 가장 좋은 모델인 'GBM_1_AutoML_1_20250623_231016'이 선택되었으며 저장함

3. AutoML 중 H2O 모델을 사용하여 배송 도착 예정일 예측 시스템 구현

```
print("📦 예측:", h2o_예측기("서울특별시","서초구","부산광역시 해운대구","2023-06-23"))
```

```
Parse progress: |[redacted]| (done) 100%
gbm prediction progress: |[redacted]| (done) 100%
📦 예측: 2023-06-24 00:00:00
```

```
📊 MAE: 0.9632
📊 RMSE: 2.1716
```

저장한 AutoML 모델을 불러와 배송 예측 결과와 성능을 확인하였음

3. AutoML 중 H2O 모델을 사용하여 배송 도착 예정일 예측 시스템 구현

따라서, 앞선 AutoML 모델을 활용하여 배송 데이터에 적용한 후 결과를 파악하고자 함

이때, AutoML 특성상 모델에 적용시키기 위해서는 데이터 타입, 변수 등 모든 조건을 같게 설정해줘야 하기에
배송 데이터를 맞게 설정해준 뒤 대입함

3. AutoML 중 H2O 모델을 사용하여 배송 도착 예정일 예측 시스템 구현

```python
# 2. 데이터 처리 (생략된 원본 로드, df는 존재한다고 가정)
df['deliv_year'] = df['배송시작일'].dt.year
df['deliv_month'] = df['배송시작일'].dt.month
df['deliv_dayofweek'] = df['배송시작일'].dt.dayofweek
df['DELIV_CTPV_NM'] = '서울특별시'
df['DELIV_CTGG_NM'] = '서초구'
df['EUPD_IVNM'] = df['거주지역']

# 3. H2OFrame 변환
h2o_df = h2o.H2OFrame(df[['DELIV_CTPV_NM', 'DELIV_CTGG_NM', 'EUPD_IVNM',
                          'deliv_year', 'deliv_month', 'deliv_dayofweek']])

# ✅ 4. MOJO 모델 로드
# ✅ 4. 정확한 zip 경로로 MOJO 모델 불러오기
model = h2o.import_mojo("h2o_best_model.mojo/GBM_1_AutoML_1_20250623_231016.zip")

# ✅ 5. 예측 수행
preds = model.predict(h2o_df).as_data_frame().values.flatten()

# 6. 예측 결과 결합 및 날짜 계산
df['예상_배송소요일'] = preds
df['예상_배송완료일'] = df['배송시작일'] + pd.to_timedelta(df['예상_배송소요일'].round(), unit='D')

# 7. 결과 출력
print(df[['배송시작일', '예상_배송소요일', '예상_배송완료일']].head())
```

가장 성능이 좋았던 ‘GBM_1_AutoML_1_20250623_231016’ 모델을 불러와 배송 데이터에 적용시킴

3. AutoML 중 H2O 모델을 사용하여 배송 도착 예정일 예측 시스템 구현

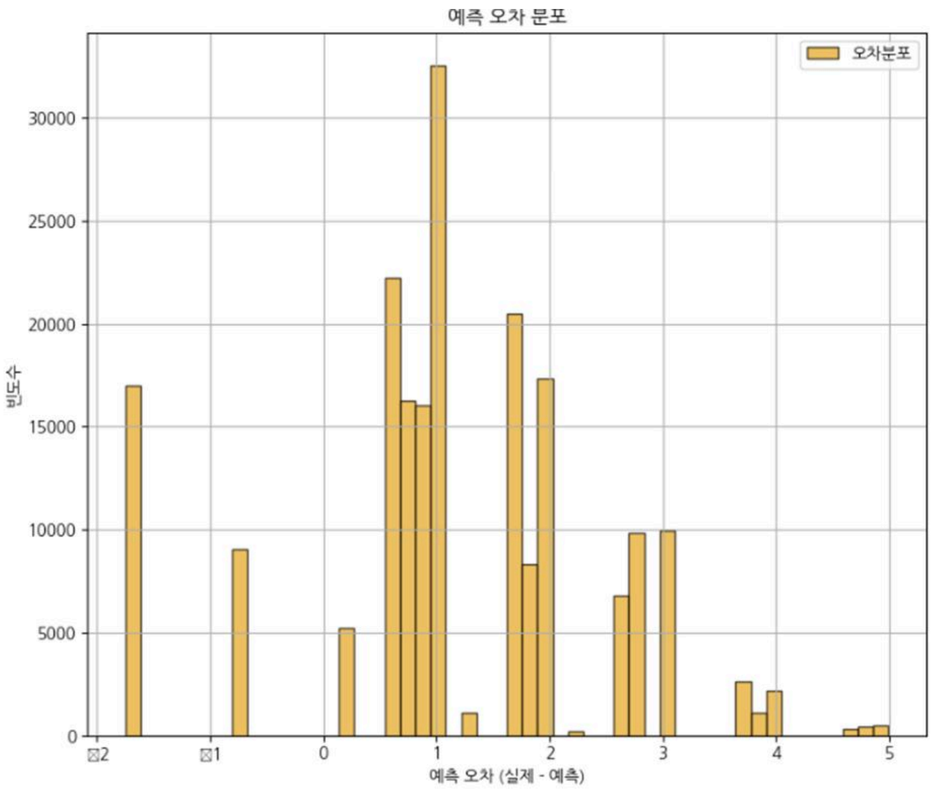
	배송시작일	예상_배송소요일	예상_배송완료일
0	2022-01-01	1.017932	2022-01-02
1	2022-01-02	1.009090	2022-01-03
2	2022-01-03	3.748754	2022-01-07
3	2022-01-04	1.261576	2022-01-05
4	2022-01-05	1.167850	2022-01-06

```
count    199582.000000
mean          1.592018
std           0.960118
min           1.009090
25%           1.017932
50%           1.261576
75%           1.352834
max           3.748754
Name: 예상_배송소요일, dtype: float64
```

```
MAE: 1.5037903829960664
RMSE: 1.7425075778790646
```

예상 배송 소요일과 예상 도착 예정일 결과를 확인하였으며, 해당 결과의 성능 평가는 다음과 같음

3. AutoML 중 H2O 모델을 사용하여 배송 도착 예정일 예측 시스템 구현



또한, 배송 데이터를 AutoML로 돌린 결과 실제 배송 완료일과 예상 배송 예측값의 오차 범위를 그래프로 나타냄
최소 -2일부터 최대 +5일까지의 오차가 존재하지만 대부분 +1일, +2일에 모여있어
실제 배송 완료일과 예상 배송 완료일의 차이가 크지 않음을 발견함

배송 인사이트 전략 제안

1. 물류센터·공장 위치 기반의 배송 지연 문제 파악

- **쪼개기 배송** 문제
  - 유한킴벌리는 대전, 김천, 충주, 안양에 자사 생산 거점을 두고 있음
  - 공장 인접 지역인 경기도, 충청도의 쪼개기 배송이 많음
  - 쪼개기 배송으로 주로 1일에서 최대 4일까지 늦게 도착함

- **쪼개기 배송**으로 배송 도착일의 예측력 부족
  - 고객 입장에서는 n회에 걸친 택배 박스 처리, 출고 일자 분산 등 시간적 비효율 문제를 느낌

2. 배송 품질 개선이 시급한 지역 : 경상북도

- 자사브랜드 배송이 느린 지역으로 꼽혔으며, 쪼개기 배송 비율이 상위 5위
- 김천에 생필품을 생산하는 공장이 위치하였는데도 배송이 느림

  → 특히, 충성고객의 주문이 가장 많은 기저귀, 생활용품의 배송이 느림
- 느린 배송+ 예측 불가능한 배송의 불만 유발

→ 배송이 느린편인데 공장이 위치해 현실적인 물류 연계 최적화 가능

3. 경상북도의 배송 프로세스 최적화 방안

- 기존 대리점과 물류센터 통합 운영하여 배송 루트 최적화 → 효율성 증가

4. 전반적인 배송 지연 문제 해결 : 적기 배송일 예측 모델 활용

- **AutoML 기반 배송 도착 예정일 예측 모델로:**
  - 주문 상품 조합 + 거주 지역 + 요일 + 배송지 위치 + 배송 출발일 기반으로
  - 예상 도착일 자동 예측 → 배송 일정의 불확실성을 감소
  - 고객에게 도착일 안내를 미리 메시지로 안내해 배송 일정 지연 사전 대응 → 신뢰 확보 및 재구매율 재고 기대
  - 또한, 맘큐는 배송 지연이 예상되는 주문을 사전에 식별 → CS비용 절감 및 고객 만족도 향상
  - 지역별, 경로별 배송 소요일을 고려한 상품 재고 관리 최적화 → 전략적 대응 가능
  - 정확한 배송 일정 예측 → 고객 이탈 방지 및 데이터 기반 의사결정에 활용해 장기적인 성과 기대

마무리

자체 평가 의견

<프로젝트 결과물에 대한 완성도 평가>

외부데이터와 맘큐 데이터를 활용하여 예측 모델을
안정적으로 구현함

<잘한 점>
AutoML과 하이퍼파라미터 튜닝을 통해
예측 모델의 안정성과 정확도 향상에 기여함
외부 데이터를 많이 분석에 활용하여 현실을 많이 담고자 함

<아쉬운 점>
시군구 등 상세 위치 정보가 있었다면
예측 모델의 성능을 높일 수 있었음

<추후 개선점이나 보완할 점>

모델 성능 향상을 위한 데이터 다양성 확보 필요함

<느낀 점이나 경험한 성과>

Hyperparameter 튜닝·AutoML 기반
이탈 예측 모델 개발함
외부·배송데이터 활용하여 AutoML기반
배송 도착 예정일 예측기를 만듦

자료 출처

- **통계청, 「인구동향조사」, 2025.03, 2025.06.25, 월.분기.연간 인구동향(출생,사망,혼인,이혼)**
- **맘큐 허그박스 리뷰 (스크래핑) |** https://www.momq.co.kr/shop/neoevent_view.html?ev_no=202004180005&cmt_type=&page=1#none
- **기사 |** https://www.hankyung.com/article/202502245438i
- **빅카인즈 |** https://www.bigkinds.or.kr/v2/intro/index.do

QnA

1조 발표는 여기까지입니다. 감사합니다 👋