L.POINT

롯데 온라인 쇼핑몰 소비패턴 분석을 통한 개인화 서비스 제안

올림푸스 : 김수빈 김아름 박대한 안영은
유찬희 이동일 이용빈

2020.04.29

목차

Ⅰ. 요약

통계청은[1] 온라인쇼핑 동향 조사를 통해 온라인쇼핑 거래액이 꾸준히 증가하는 추세라고 밝혔으며, 정보통신정책연구원(KISDI) 또한 온라인쇼핑 시장의 가파른 성장세를 전망했다. 이처럼, 온라인몰의 대약진은 지속될 것으로 보이고, 온라인몰의 차별화 또한 매우 중요한 부분으로 두각을 나타낼 것이다.

본 분석팀은 '2019 L.point 빅데이터 컴피티션'을 위해 제공된 롯데 쇼핑몰의 온라인 행동 데이터를 다양한 분석 기법을 활용하여 분석했으며, 해당 데이터는 2018년 4~9월간의 롯데그룹 온라인 계열사의 구매내역을 담고 있다. 데이터를 정제, 탐색 및 분석하는 과정에서 롯데 쇼핑몰에 대해 다음과 같은 문제점들을 발견했다.

1. 필요 이상으로 많은 상품 분류로 인한 웹사이트 탐색의 어려움
2. 업무제휴 협약 효과 미미
3. 지속적인 매출 하락 추세 및 전망
4. 상품에 대한 관심이 구매로 이어지지 않음

상품 분류와 업무제휴 협약 효과는 각자 전체 상품군의 재분류와 마케팅 전략 수정을 통해 해결될 수 있을 것이라 보이나, 매출 하락과 상품에 대한 관심이 구매로 이어지지 않는다는 문제점을 해결하기 위해선 보다 심층적인 데이터에 대한 이해가 요구된다. 본 분석팀은 이에 대한 해결책으로 보다 데이터에 입각해 머신러닝 기법을 고객 분류 및 상품 추천에 활용한 온라인몰 UI/UX 개선 방안을 제안한다.

또한, 통계청은 모바일 온라인 주문량의 가파른 상승세를 언급했으며, 대형 글로벌 온라인몰의 UI는 점점 간소화되고 있다. 이렇듯, 온라인몰 상품의 '전시공간'은 점점 협소해질 것이며, 고객이 원하는 물품을 적재적소에 노출하는 것은 매우 중요하다. 따라서 본 분석팀은 구매 경로에 따라 개인화된 추천 상품 노출을 제안한다. 분산분석과 클러스터링을 통해 고객을 세분화하고, 연관 분석 및 협업필터링을 적용할 것이다. 이를 바탕으로 고객별 선호도가 높은 상품 위주로 추천함으로써 개인화된 홈페이지를 제공하고, 고객의 이목을 끌어 실구매로 이어질 수 있도록 다음과 같이 홈페이지를 구성한다.

1. 로그인 후의 메인페이지

분산분석/클러스터링으로 세분화된 고객의 구매 데이터를 기반으로 선호 상품군을 추천한다. 또한, 간소화된 상품 분류 목차와 실시간 쇼핑 검색어를 노출시킬 것이다. 실시간 검색어는 선거

1) 2020년 2월 온라인 쇼핑 동향, 통계청(2020..04.20)

기간 동안 그 이용이 제약될 정도로 파급 효과가 크기 때문에, 이를 활용하여 홈페이지를 새로 구성하여 고객의 관심을 유도할 것이다.

2. 제품 상세 페이지

대분류 카테고리별 소분류 연관 분석을 통해 어떤 상품들이 함께 팔리는지 경향성을 파악하고, 연관성이 높은 상품을 제품 상세 페이지에 같이 추천함으로써 구매율을 높인다.

3. 장바구니 페이지

협업 필터링을 통해 구매 연관도가 매우 높은 타 카테고리 제품을 추천한다.

또한, 저조한 롯데 온라인몰의 매출을 개선하기 위해 본 분석팀은 홈페이지 구성 외에도 고객 니즈에 상응하는 브랜드와의 업무제휴 협약 및 이탈 고객을 위한 Remarketing을 제안한다. 롯데 온라인 쇼핑몰은 '2018년 브랜드 고객충성도 조사'에서 온라인종합쇼핑몰 부문 1위에 선정됐다. 이에 따라, 이탈률 분석을 통해 이탈 주기에 맞춰 푸시 마케팅을 하여 고객 충성도를 유지하는 것이 매우 중요하다고 판단된다. 온라인 쇼핑몰 시장의 성장세가 지속될 것으로 전망됨에 따라, 치열한 온라인 쇼핑몰 시장 경쟁에서 우위를 점하기 위해 차별화된 전략을 수행해야 할 것이다.

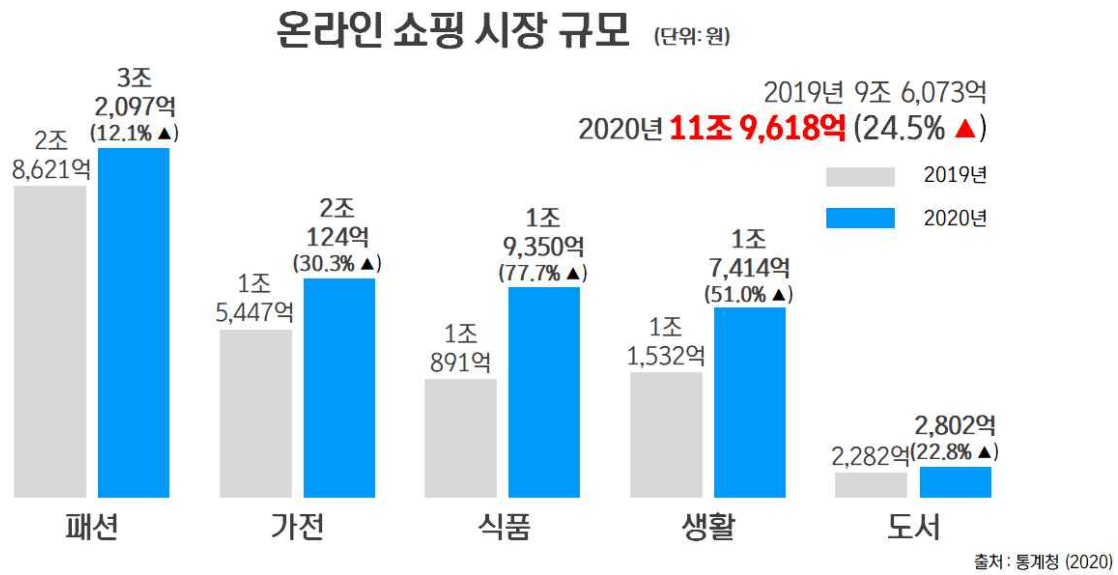
Ⅱ. 분석 배경

1. 온라인쇼핑 동향

온라인 쇼핑 시장은 급속도로 성장하고 있다. 매년 가파른 상승세를 기록하고 있던 온라인 쇼핑 시장은 최근 코로나19 사태까지 맞물려 전문가들의 예상보다 훨씬 빠르고 가파른 성장을 기록했다.

통계청이 최근 발표한 자료에 따르면, 2020년 2월의 국내 상품 소비의 약 30%가 온라인에서 이루어진 것으로 나타났다. 지난달, 전체 소매판매액 34조5045억 원 중 온라인을 통해서만 9조 5465억원을 차지했다. 이는 전체 소매판매액에서 차지하는 비중이 27.7%로, 집계된 이래에 가장 높은 수치이기도 하다.

2020년 2월 온라인 쇼핑 거래액은 총 11조 9천 억 원에 달하며, 지난해 동월 대비 24.5% 증가한 수치다. 특히, 이 중 모바일을 통해 거래된 금액은 8조1436억원으로 31.1% 늘었다. 전체 거래액에서 차지하는 비중은 역대 가장 높은 68.1%를 기록했다.



통계청의 해석에 따르면, 코로나19와 1인 가구 등 소비형태의 변화로 신선식품, 간편조리식, 배달음식 등의 거래가 증가했으며, 코로나19로 인한 가정 내 생활 증가로 세정제, 휴지, 세제 등 생활용품의 거래도 지난해 대비 증가한 것으로 판단된다. 또한, 기존부터 온라인 시장에서 강세를 보이고 있던 패션, 음.식품, 가전.전자, 화장품 등의 상품군도 전년 대비 증가한 것으로 나타났다.

반면, 대형마트 혹은 슈퍼마켓 등 기존 소매 업계는 성장률 하락이 지속되고 있다. 대형마트는 소상공인 지원을 위한 규제 강화와 온라인 경쟁 심화로 인해 점점 성장률이 하락하는 추세로, 기

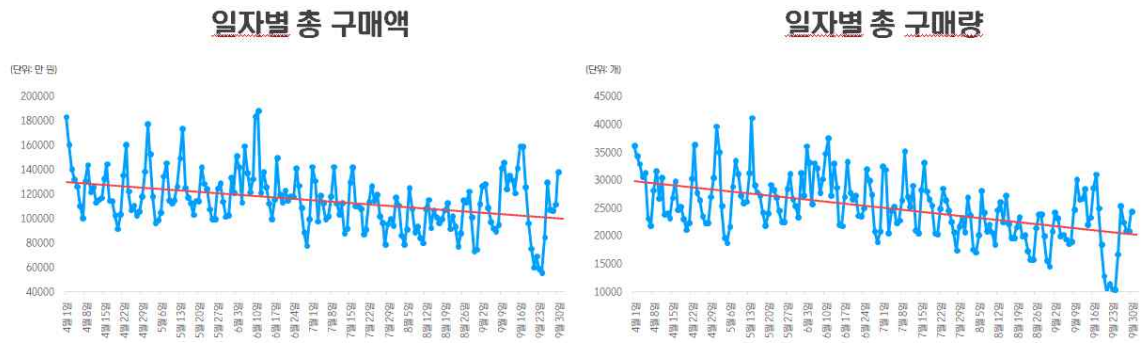
존 유통 기업들은 부진한 점포를 정리하고 자동화를 도입하는 등 구조 개편 작업을 진행하고 있다. 소형 슈퍼마켓 역시 상당 부분 폐점하거나 편의점으로 전환하는 등 슈퍼마켓 시장도 새롭게 개편되는 모습을 보였다.

특히, 롯데백화점과 롯데마트, 롯데슈퍼, 롭스 등 매장을 운영하며 오프라인 유통 부문 국내 1위를 기록하고 있던 롯데그룹은 3~5년 내 오프라인 매장 700여개 중 실적이 부진한 점포 200여 곳의 문을 닫는다고 밝혔다. 이는 전체 점포의 30% 규모다. 이러한 배경에는 2019년 영업이익과 매출액이 각각 4279억원, 17조 6328억원으로 지난 분기보다 영업이익이 크게 감소했다는 점이 크게 작용했다.

2. 롯데 온라인 쇼핑몰 구매 추이 및 전망 예측

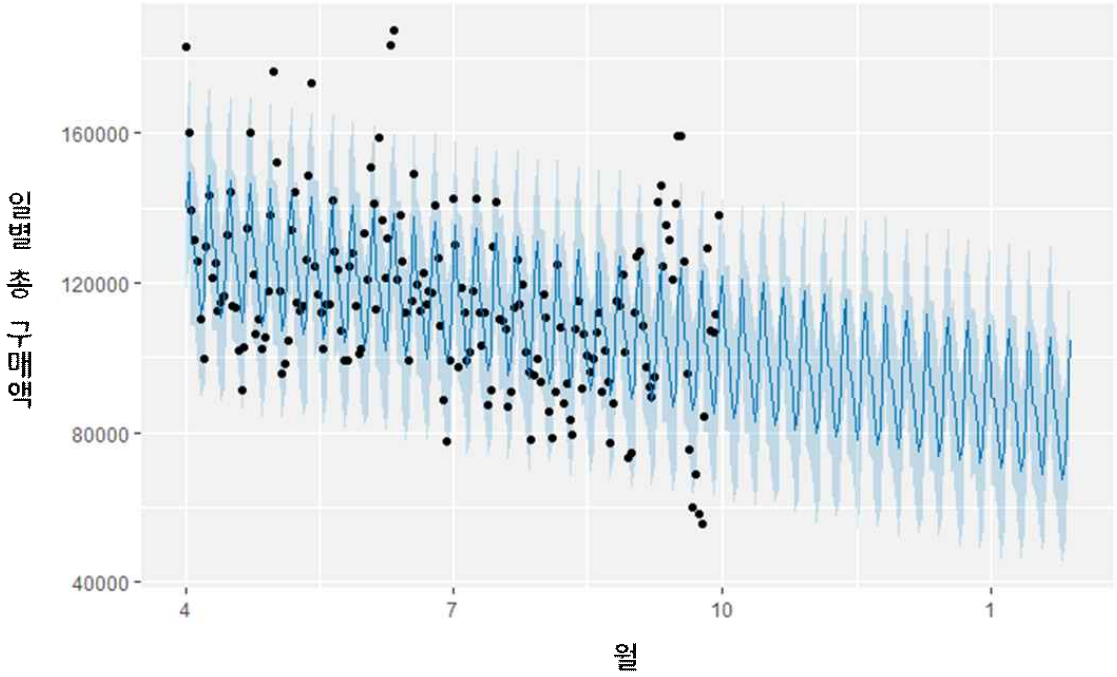
롯데는 온라인 사업에서도 이커머스 업체들의 공세에서 두각을 나타내지 못하고 있다. 온라인 시장의 강자인 쿠팡과 마켓컬리, 29cm 등의 이커머스 업체들과 다르게 뒤늦게 온라인 시장에 뛰어들었으며, 치열한 경쟁 속에서 뚜렷한 성과를 내지 못하고 있다.

롯데 온라인 쇼핑몰의 내부 자료를 보면 롯데 온라인 쇼핑시장의 상황을 더욱 잘 파악할 수 있다. 본 분석팀이 살펴본 롯데 온라인 쇼핑몰의 일자별 총 구매액과 일자별 총 구매량에 따르면, 2018년 4월 1일부터 9월 30일까지 분석한 매출과 판매량 수치 모두 지속적으로 떨어지는 추세에 있음을 확인할 수 있었다.



<그림: 일자별 총 구매액/ 구매량>

온라인 유통업계의 후발주자로 뛰어든 롯데는 온라인 시장에서 뚜렷한 두각을 나타내지 못하고 있으며, 오히려 매출이 하락하는 추세를 보이고 있다. 이에 따라, 향후 전망을 보고자 본 분석팀은 Python을 활용하여 Facebook에서 제공한 Prophet 패키지를 통한 롯데 온라인몰의 매출 추세를 예측하였다. 4월부터 9월까지의 데이터를 기반으로 향후 1월까지의 추세를 그래프로 나타내었으며, 분석 결과 상향추세 없이 점진적으로 조금씩 하향추세를 띄고 있었다.

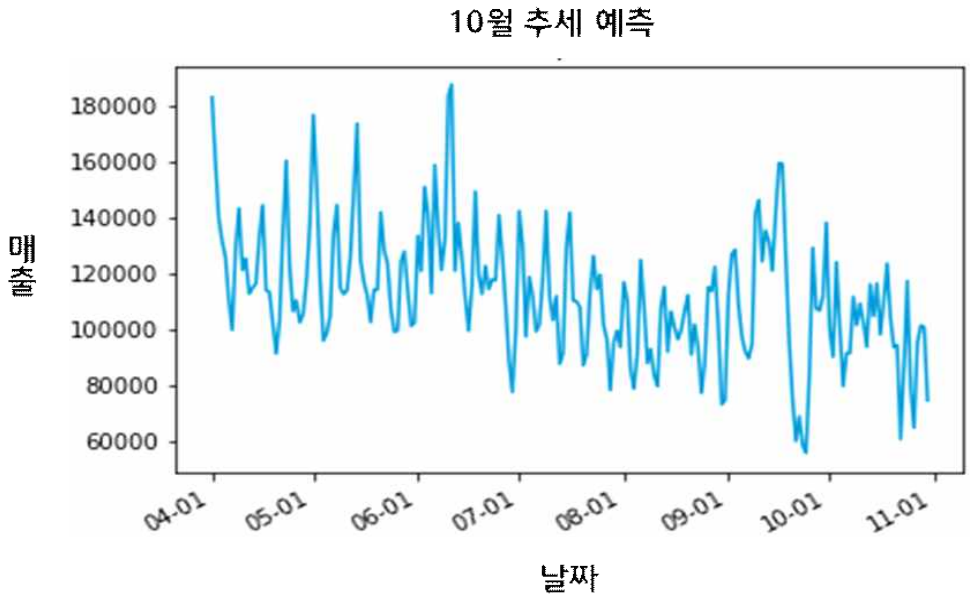


<그래프: 시계열 예측 결과>

<Prophet 시계열 예측>

```
# 데이터 로딩

df =   T_series_lotte5
head(df)


# Date 컬럼 속성 변경
df$Date = as.Date(df$Date, format="%d/%m/%y")

#Prophet에 맞게 컬럼 이름 변경하기
colnames(df) <- c('ds', 'y')


# 예측할 데이터프레임 기간 설정
future <- make_future_dataframe(m, periods = 120)

# 예측
forecast <- predict(m, future)
```

Prophet에서 전체적인 추세선을 보았다면 딥러닝(LSTM)을 활용하여 좀 더 구체화된 수치로 예측하였으며, 기존 4월부터 9월까지 그래프에 예측한 10월 매출을 추가하여 추세를 분석하였다.



<그림 : 4~9월 추세 & 10월 추세 예측>

시각화한 결과 6월말까지 추세 변동 폭이 크지만, 내부 추세선을 보았을 때 큰 변화 없이 유지하는 양상을 띤다. 또한, 7월부터 다시 하락추세선으로 변하여 9월초까지 하락하는 모습을 보였고, 9월초부터 9월말까지 기술적 반등으로 상승추세선으로 바뀌지만 다시 최저점을 찍는 모습을 볼 수 있었다. 예측을 통해 추세선은 큰 이변이 없는 한 상향으로 바뀌지 않을 것으로 전망되며, 본 시점에서 매출 하락을 개선하기 위해서는 문제점을 파악하고 마케팅 전략을 재수립하는 것은 필수적이라 판단된다.

3. 분석목표

위와 같은 배경에서 본 분석팀은 '2019 L.point 빅데이터 컴피티션'을 위해 제공된 롯데 쇼핑몰의 온라인 데이터를 다양한 분석 기법을 활용해 분석했으며, 해당 데이터는 2018년 4~9월 동안의 롯데그룹 온라인 계열사의 구매내역이다. 데이터를 기반으로 본 분석팀은 먼저 롯데 온라인 쇼핑몰의 문제를 파악하고, 매출 향상을 위한 다각화된 전략을 제안하고자 한다.

Ⅲ. 분석 데이터 탐색

본 장에선 프로젝트에 대한 전반적인 분석 파이프라인, 개발 환경 및 데이터를 소개할 것이다. 이어, 분석 목적에 적합한 데이터 전처리 과정과 기초 데이터에 대한 탐색적 데이터 분석 및 시각화 결과를 보여줄 것이다.

1. 데이터 분석 프로세스 및 개발환경

본 분석팀은 Windows10과 Ubuntu 18.04.LTS, 두 가지 운영체제에서 분석을 진행하였고, 데이터베이스로는 Oracle 사의 11g XE, 12c, 클라우드 환경의 ATP를 사용하였다. 분석 언어로는 R과 Python을 사용하였다.

먼저, 분석에 필요한 데이터를 저장 및 팀원 간 공유하기 위해 데이터베이스 및 사용자를 생성하였다. 이후, SQL Developer 프로그램에서 데이터베이스에 연결시켜, 원본 CSV 파일로 수집된 데이터를 임포트하였다. 이후, 데이터 모델링을 통해 임포트된 데이터를 개편하였다. 이 과정에서 생성된 테이블을 SQL을 사용해, 조작 및 조회하여 기초적인 데이터 탐색을 하였다. 또한, 결측치가 포함된 데이터에 대한 정제를 진행하였다.

한편, 데이터 시각화를 위해, Python의 matplotlib, seaborn 패키지를 활용하여 탐색적 데이터 분석을 진행하였고, 이를 R의 ggplot2, ggpubr 등의 시각화 패키지를 이용하여 데이터 탐색을 진행하였다. 이를 통해 얻어낸 인사이트를 활용해, 필요한 컬럼을 구성하고 파생변수를 추가하여 분석에 필요한 데이터를 생성하였다.

탐색을 마친 후, R의 다양한 패키지를 사용 하여 고급 분석을 진행하였다. 분산분석을 통해 성별/연령별 분류를 축소하였고 클러스터링을 통해, 세분화된 고객 집단을 생성하였다. 또한, 고객별 카테고리별 소분류 항목 연관분석과 협업필터링을 통한 개인화된 상품 추천 페이지 구성으로 전략을 도출하였다.

2. 분석 데이터 소개

(1) csv 파일

- Custom.csv

  회원정보에 대한 csv 파일로, 구매 회원들 중 고객의 성별과 연령대 정보를 담고 있다.

- Master.csv

  상품 정보에 대한 데이터를 갖는 csv 파일로, 상품들을 유일하게 구분해주는 상품 코드와 상품명 및 해당 상품이 속하는 대·중·소분류 정보를 담고 있다.

- Session.csv

  세션 행동 데이터 csv 파일로, 구매 세션의 아이디와 세션의 일자, 세션의 일련번호, 해당 세션의 페이지 조회 건수 및 세션 시간 등 행동 데이터를 담고 있다. 또한, 기기유형과 지역 ID로 구분한 대·중 분류 정보를 담고 있다.

- Product.csv

  구매 정보에 대한 csv 파일로, 주문 시 클라이언트 ID와 세션ID를 갖고, 방문자의 행위 번호를 나타내는 히트 일련번호, 구매한 상품코드와 브랜드, 금액 및 구매건수에 대한 데이터를 갖는다.

- Search1.csv

  고객의 세션별 검색 정보에 대한 csv 파일로, 고객이 해당 세션에서 특정 키워드의 검색어를 얼마나 검색했는지에 대한 정보를 담고 있다.

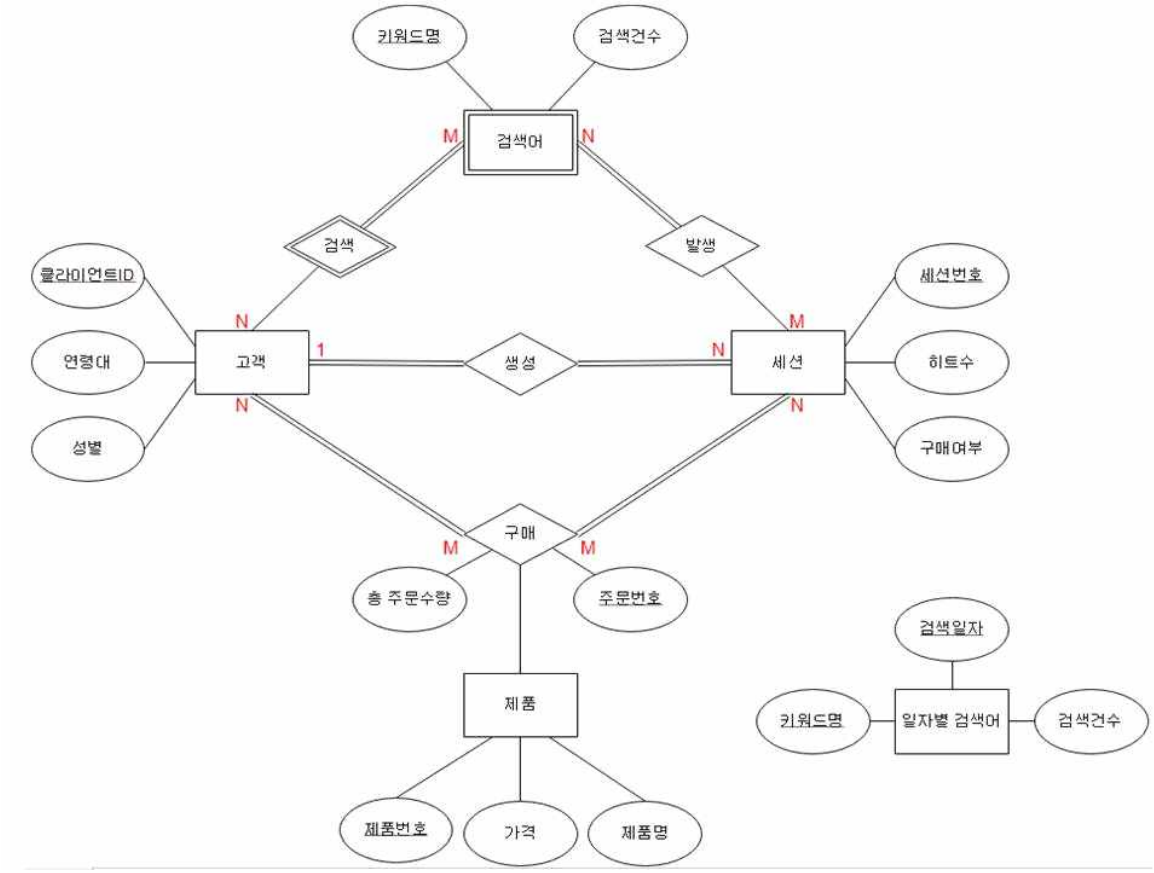
- Search2.csv

  일자별 검색 정보에 대한 csv 파일로, 2018/04/01~2018/09/30 기간의 각 일자별 각 검색어의 총 검색량에 대한 정보를 담고 있다.

(2) 모델링

① 개념적 모델링

원본 데이터를 기준으로 개념적 모델링을 진행한 결과 아래와 같은 Peter-Chen ERD가 생성되었다.



<그림: 개념적 모델링>

세션은 고객에 의해 생성되므로, 생성의 관계를 가지며 구매는 각 고객의 특정 세션에서 특정 제품에 대해 이루어지므로 고객과 세션, 제품은 구매라는 관계를 갖는다. 검색어는 각 세션별로 검색횟수를 나타내므로 고객과 세션과 각각 관계를 가진다.

② 논리적 모델링

개념적 모델링을 바탕으로 논리적 모델링을 진행하여, 베이스테이블과 코드테이블을 구분하였다,

< 베이스테이블 >

1. 고객(클라이언트ID(PK), 성별, 연령대)
2. 상품(상품코드(PK), 상품명, 상품분류코드(FK))
3. 세션(클라이언트ID(FK), 세션ID, 세션일련번호, 세션일자, 총페이지조회건수, 총세션시간값, 기기유형, 지역코드(FK))

4. 검색어(클라이언트ID(FK), 세션ID(FK), 검색키워드명, 검색건수)

5. 검색(클라이언트ID(FK), 세션ID(FK), 검색키워드명(FK))

6. 발생(클라이언트ID(FK), 세션ID(FK), 검색키워드명(FK))

7. 구매(구매번호(PK), 클라이언트ID(FK), 세션ID(FK), 히트일련번호, 상품코드, 상품옵션정보, 상품브랜드, 단일상품금액, 구매건수)

8. 일별검색어(검색일자(PK), 검색키워드명(PK), 검색건수)

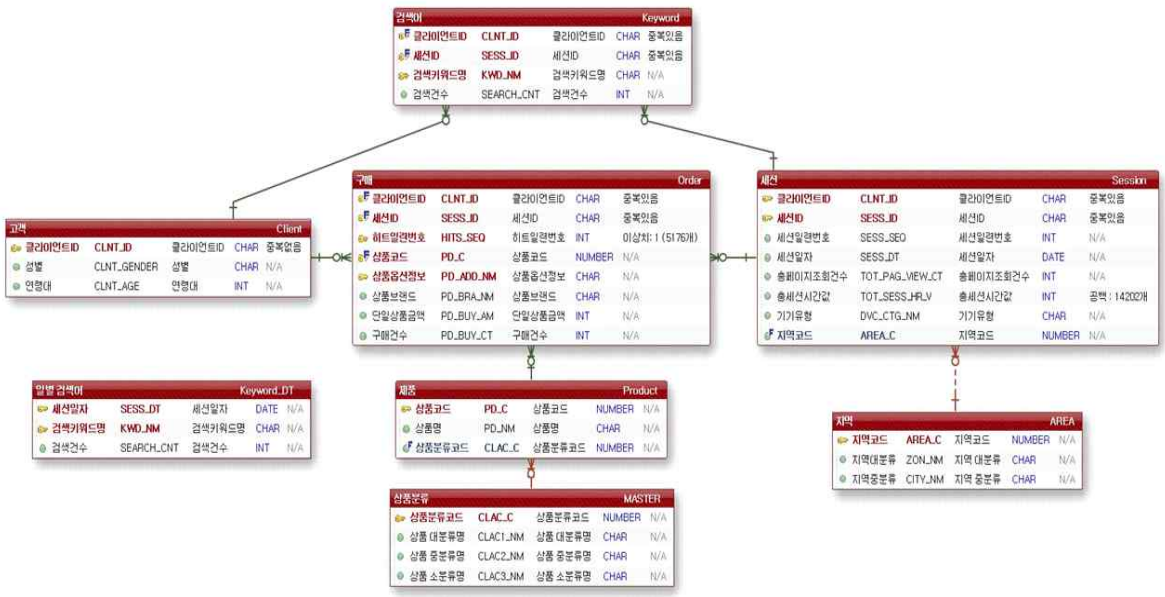
< 코드테이블 >

1. 상품분류(상품분류코드(PK), 상품 대분류명, 상품 중분류명, 상품 소분류명)

2. 지역(지역코드(PK), 지역대분류, 지역중분류)

③ 물리적 모델링

논리적 모델링 후, 아래 그림과 같이 IE 표기법으로 물리적 모델링을 진행 한 후 데이터베이스에 Oracle의 Database에 테이블로 생성하였다.



<그림: 물리적 모델링>

④ Database 구현

Oracle의 데이터베이스에 SQL을 사용하여 테이블을 구성하였으며, 모델링 결과로 생성한 테이블을 관리할 계정을 생성하였다.

```
SET TERMOUT OFF
SET ECHO OFF

-- 사용자의 권한과 테이블 스페이스를 지정한다.
GRANT CONNECT,RESOURCE,UNLIMITED TABLESPACE TO LOTTE
IDENTIFIED BY LOTTE;
ALTER USER LOTTE DEFAULT TABLESPACE USERS;
ALTER USER LOTTE TEMPORARY TABLESPACE TEMP;
CONNECT LOTTE/LOTTE

COMMIT;

SET TERMOUT ON
SET ECHO ON
```

CSV 파일을 임포트해, 원본 데이터를 테이블에 담는다. 코드 테이블을 분리해내는, Master.csv, Session.csv 파일과 인조키를 생성하는 Product.csv 파일을 제외한 csv파일은 데이터를 임포트하여 제약조건을 설정해준 후 사용한다.

주문정보 테이블 생성 및 데이터 임포트

Product.csv 데이터는 주문을 뜻하는 order 테이블에 담는다. 이때, 기본키를 지정하는데, 기존 csv파일의 변수만으로 복합키를 지정할 시, 너무 많은 변수가 복합되어 데이터를 조회 속도를 저하시키므로, 이를 방지하기 위해 인조키인 order_id를 생성한다. order_id에는, 숫자값을 주어 테이블의 데이터를 유일하게 구분하는데, 이때 이 값은 시퀀스를 사용하여 시작값인 1부터 1씩 증가하여 값을 지정한다.

```
-- CSV 파일을 담는 lorder 테이블 생성
CREATE TABLE lorder(
    clnt_id NUMBER,
    sess_id NUMBER,
    hit_seq NUMBER(3),
    pd_c  NUMBER,
    pd_add_nm VARCHAR2(500),
    pd_bra_nm VARCHAR2(500),
    pd_buy_am NUMBER,
    pd_buy_ct NUMBER
);

-- 기존 변수만으로 기본키를 설정하려면, 5개의 컬럼을 갖는 복합키가 되므로
-- 인조키를 생성하여 lorder 테이블의 행을 구분해준다.
```

```
ALTER TABLE lorder
ADD order_id NUMBER;

-- 인조키를 위한 시퀀스 생성
CREATE SEQUENCE order_id_seq
START WITH 1
INCREMENT BY 1;

-- 인조키 order_id에 시퀀스를 이용하여, 인조키를 생성한다.
UPDATE lorder
SET order_id = order_id_seq.NEXTVAL;

-- lorder 테이블에 기본키를 추가한다.
ALTER TABLE lorder
ADD CONSTRAINT lorder_id_pk PRIMARY KEY(order_id);

-- 주문 당 총금액을 컬럼으로 추가한다.
ALTER TABLE lorder
ADD pd_buy_tot NUMBER;

-- 주문 당 총액값을 업데이트한다.
UPDATE lorder
SET pd_buy_tot = pd_buy_am * pd_buy_ct;
```

상품정보 테이블과 상품분류 테이블 분리

Master.csv 데이터는 원자료 그대로의 상품 데이터를 뜻하는 product_raw 테이블에 담는다. 모델링 결과 상품분류에 대한 정보를 코드 테이블로 분리하기로 결정하였으므로, 분류를 뜻하는 Master 테이블을 생성한다. Master 테이블에는 product_raw의 대·중·소분류 값을 담은 후, 각 분류를 유일하게 구분해주는 기본키로 인조키 clac_c를 생성하며, 시퀀스로 1부터 시작하는 숫자값을 부여한다.

```
-- 원본 CSV의 데이터를 담는 테이블 생성
CREATE TABLE product_raw(
    PD_C NUMBER(10,0),
    PD_NM VARCHAR2(400),
    CLAC1_NM VARCHAR2(50),
    CLAC2_NM VARCHAR2(100),
    CLAC3_NM VARCHAR2(100)
);

-- 원본 CSV파일의 상품데이터에서, 분류를 담는 master 코드 테이블을 분리
-- 분류 테이블의 소분류에 따른, 인조키 생성
-- 인조키를 위한 시퀀스 생성
CREATE SEQUENCE master_clac_seq
START WITH 1
INCREMENT BY 1;
```

```
-- 분류를 담고 있는 테이블 생성
CREATE TABLE MASTER(
    clac_c NUMBER(3),
    clac1_nm VARCHAR2(50),
    clac2_nm VARCHAR2(100),
    clac3_nm VARCHAR2(100)
);

-- 대/중/소분류를 담고 있는, master2 테이블에서 master테이블에 분류를 담는다
INSERT INTO MASTER(clac1_nm, clac2_nm, clac3_nm)
SELECT clac1_nm, clac2_nm, clac3_nm
FROM  master2
ORDER BY clac1_nm ASC, clac2_nm ASC, clac3_nm ASC;

-- 소분류로 구분가능한 분류번호를 담는 clac_c를 시퀀스를 이용해 업데이트 해준다.
UPDATE MASTER
SET clac_c = master_clac_seq.NEXTVAL;

-- master 테이블의 기본키를 clac_c로 지정한다.
ALTER TABLE MASTER
ADD CONSTRAINT master_clac_pk PRIMARY KEY(clac_c);
```

분류 정보를 제외한 상품 정보를 담는 Product 테이블에는, product_raw의 상품 정보와, Master 테이블의 기본키인 clac_c를 외래키로 갖는다.

```
-- 원본 CSV에서 필요한 분류정보를 제외한 정보를 갖는 product 테이블 생성
CREATE TABLE product(
    PD_C NUMBER(10,0),
    PD_NM VARCHAR2(400),
    CLAC_C NUMBER(3)
);


-- product 테이블에 master테이블에만 존재하는 clac_c값을 넣어준다.
INSERT INTO product(pd_c, pd_nm,clac_c)
SELECT a.pd_c, a.pd_nm, b.clac_c
FROM product_raw a, master b
WHERE a.clac1_nm = b.clac1_nm
    AND a.clac2_nm = b.clac2_nm
    AND a.clac3_nm = b.clac3_nm;


-- product 테이블에 기본키를 추가해준다.
ALTER TABLE product
ADD CONSTRAINT product_pd_c_pk PRIMARY KEY(pd_c);

-- master의 clac_c를 외래키로 가지므로, 제약조건으로 추가한다.
ALTER TABLE product
ADD CONSTRAINT product_clac_c_fk FOREIGN KEY(clac_c)
REFERENCES master(clac_c);
```

세션정보 테이블과 지역분류 테이블 분리

Session.csv 데이터는, 세션 원본임을 뜻하는 session_raw 테이블에 담는다. 모델링 결과 지역 분류에 대한 정보를 코드 테이블로 분리하기로 결정하였으므로, 지역분류를 뜻하는 Area 테이블에 이를 담는다.

Area 테이블에는 session_raw의 지역 대·중분류 값(zon_nm, city_nm)을 담은 후, 각 분류를 유일하게 구분해주는 기본키로 인조키 area_c를 생성하며, 시퀀스로 1부터 시작하는 숫자값을 부여한다.

분류 정보를 제외한 세션 행동 정보를 담는 Session 테이블에는 session_raw의 상품정보와, Area 테이블의 기본키인 area_c를 외래키로 갖는다. 이는 위의 상품정보 데이터를 분리하는 과정과 동일하므로 코드는 생략한다. 모델링을 바탕으로, 위와 같이 테이블을 정제한 결과는 아래와 같다.

구분		내용
온라인 행동 데이터	구매(Order)	구매된 상품명, 상품금액/건수 등 방문자의 상품구매 정보
	검색어(Search)	상품을 구매한 방문자(Visitors)가 검색창에 입력한 검색어에 대한 검색량
	일자별 검색어 (Search Date)	분석데이터 기간 내 일별/검색어별 검색량
	고객(Client)	방문자(Visitors)의 성별/연령 정보
	세션(Session)	상품구매한 방문자(Visitors)의 세션 접속일자, 총 세션시간, 세션기준 발생 지역, 기기유형 등 세션 정보
분류	상품(Master)	상품별 상품명, 대/중/소 분류명 정보
	지역(Area)	지역별 대/중 분류명 정보

< 표: 모델링 결과 생성한 테이블>

3. 데이터 전처리

데이터 전처리는 데이터 정제 및 파생변수 생성, 두 파트로 나누어 진행하였다. 기본적인 데이터 탐색 과정을 거친 후 발견 이상치를 대체하였다. NULL 값으로 분석의 집계함수 결과에 영향을 주는 이상치와 의미상의 이상치를 각각 대체 혹은 제거하여 분석 결과에 왜곡을 주지 않도록 처리했다. 또한, 분석에 필요한 브랜드명이 같은 이름으로 통일되어 있지 않고, 패션의 경우 키즈와 브랜드명이 따로 네이밍되어 있거나, 콜라보레이션의 경우에도 다른 이름을 사용하여 이를 통합하였다. 이는 원본데이터와 함께 제공된 코드북을 참고하였으며 SQL, R 및 Python 언어를 사용하였다.

(1) NULL값 대체

NULL값을 갖는 데이터의 변수에 대하여 집계함수를 적용하면 결과가 NULL이 되어 원하는 결과를 얻을 수 없으므로, 이를 대체한다. SQL의 'SELECT * FROM t WHICH IS NULL', 혹은 R의 summary(), table()함수를 사용하여 결측치를 조회하였다. 조회 결과 세션 정보를 담는 Session 테이블의 총 페이지 조회 건수 값을 갖는 tot_pag_view_ct 컬럼에서 274개의 결측치를 발견하였다. NULL값은 SQL문을 사용하여 우선 숫자값인 0으로 대체하였다.

(2) 이상치 제거

데이터 탐색을 통하여, 의미상의 이상치를 갖는 컬럼을 조회하였다. 이 과정에서 세 가지 컬럼을 발견할 수 있었는데, 각각 주문 시 행동 순서를 의미하는 hit_seq와 총 세션 시간인 tot_sess_hr_v, 그리고 결측치를 0으로 대체한 바 있던 tot_pag_view_ct 이었다. 각 변수의 의미상의 이상치 제거는 R을 사용하여 진행하였다.

① hit_seq 이상치 처리

히트수는 주문까지의 클릭, 이벤트 참여, 검색 등 방문자의 행위에 대해 배열된 일련번호로, 첫 번째 행위는 1로 지정하고 행동이 증가할수록 시퀀스도 증가한다. 주문에 이르기까지 최소 과정인 장바구니에 담겨 있는 물건을 주문한다고 가정할 때의 최소 히트수는, '장바구니 클릭 - 주문 버튼 클릭'의 2이다. 따라서, hit_seq의 값이 1인 관측치는 이상치로 간주하며 이를 대체할 수 없으므로 제거하였다.

```
# hit_seq가 1인 값들은, NA로 처리
df_order$HIT_SEQ <-ifelse(df_order$HIT_SEQ==1, NA, df_order$HIT_SEQ)


table(is.na(df_order$HIT_SEQ))
##   FALSE      TRUE
```

```
## 3983837      4750

# NA를 제거해주는, na.omit()함수를 사용하여 이상치 제거.
df_order <- na.omit(df_order)

summary(df_order$HIT_SEQ)
##      Min. 1st Qu.   Median      Mean 3rd Qu.      Max.
##       2.0      28.0      56.0      88.2     114.0     500.0
```

② tot_sess_hr_v 이상치 처리

세션 내에서 총 시간값을 초 단위로 갖는 컬럼인 tot_sess_hr_v는 summary() 함수 결과 최소값으로 0을 갖는다. 세션에 머문 시간이 0초인 경우는 논리적으로 불가하고, 이를 대체할 수 없으므로 제거하였다.

```
# summary() 함수로 집계값 확인
summary(df_session$TOT_SESS_HR_V)
##      Min. 1st Qu.   Median      Mean 3rd Qu.      Max.
##         0       396       913      1393      1912     39329

# tot_sess_hr_v가 0인 값들은, NA로 처리
df_session$TOT_SESS_HR_V <- ifelse(df_session$TOT_SESS_HR_V == 0, NA,
df_session$TOT_SESS_HR_V)

table(is.na(df_session$TOT_SESS_HR_V))
##
##    FALSE      TRUE
## 2221125     12160

# NA를 제거해주는, na.omit()함수를 사용하여 이상치 제거.
df_session <- na.omit(df_session)

summary(df_session$TOT_SESS_HR_V)
##      Min. 1st Qu.   Median      Mean 3rd Qu.      Max.
##         1       402       920      1400      1919     39329
```

③ tot_pag_view_ct 이상치 처리

세션 내에서 총 페이지 조회 횟수를 건 단위로 갖는 컬럼인 tot_pag_view_ct는 앞의 NULL값 대체 시 0으로 대체하여, summary() 함수 결과 최소값으로 0을 갖는다. 주문 시 총 화면의 조회 건수가 0 값을 갖는 것은 논리적으로 불가하므로 이를 제거하였다.

```
# summary() 함수로 집계값 확인
summary(df_session$TOT_PAG_VIEW_CT)
##      Min. 1st Qu.   Median      Mean 3rd Qu.      Max.
##      0.00    33.00    62.00    93.08   118.00   499.00

# tot_pag_view_ct가 0인 값들은, NA로 처리
df_session$TOT_PAG_VIEW_CT <- ifelse(df_session$TOT_PAG_VIEW_CT == 0, NA,
df_session$TOT_PAG_VIEW_CT)
table(is.na(df_order$HIT_SEQ))
##
##    FALSE
## 3983837

# NA를 제거해주는, na.omit()함수를 사용하여 이상치 제거.
df_session <- na.omit(df_session)

summary(df_session$TOT_PAG_VIEW_CT)
##      Min. 1st Qu.   Median      Mean 3rd Qu.      Max.
##      1.00    33.00    62.00    93.08   118.00   499.00
```

(3) 브랜드명 통합

상품 구매 테이블의 해당 상품의 브랜드 이름을 의미하는 pd_bra_nm 컬럼은 브랜드 파워를 파악하기 위한 좋은 지표이다. 하지만, 원본 데이터에서는 구매율이 높은 '아디다스'와 같은 브랜드의 경우에도, 각 콜라보레이션에 다른 네이밍을 하고, '오리지널'과 '오리지날'과 같은 단어를 혼용하여 브랜드명을 분석에 용이하지 않으므로, 이를 통합하였다. 다수의 브랜드명에 괄호를 포함한 경우가 많아 우선 괄호를 제거하였다.

```
# 괄호를 제거 후 브랜드명 통합
i <- grep("₩₩(", df_order$PD_BRA_NM)
j <- regexpr("₩₩(",df_order$PD_BRA_NM[i])

df_order$PD_BRA_NM[i] <- substr(df_order$PD_BRA_NM[i],1,j-1)
```

각 브랜드의 이름을 포함하는 문자열을 조회하여 이를 하나의 브랜드명으로 통합하였다.

```
# '아디다스'를 포함하는 브랜드명을 c에 담음.
c <- grep("아디다스", df_order$PD_BRA_NM)

table(df_order$PD_BRA_NM[c])
##
##                  아디다스 아디다스 스텔라 스포츠        아디다스 언더웨어
##                     62868                      15                      57
##        아디다스 오리지날        아디다스 오리지널           아디다스 키즈
##                      8734                    5595                    2819
##             아디다스골프 아디다스바이알렉산더왕            아디다스키즈
##                      3652                       1                    7125

# gsub() 함수를 이용하여, 각 이름을 '아디다스'로 통합한다.
df_order$PD_BRA_NM <- gsub('아디다스 골프', '아디다스', df_order$PD_BRA_NM)
df_order$PD_BRA_NM <- gsub('아디다스 스텔라 스포츠', '아디다스',
df_order$PD_BRA_NM)
df_order$PD_BRA_NM <- gsub('아디다스 오리지날', '아디다스', df_order$PD_BRA_NM)
df_order$PD_BRA_NM <- gsub('아디다스 오리지널', '아디다스', df_order$PD_BRA_NM)
df_order$PD_BRA_NM <- gsub('아디다스 키즈', '아디다스', df_order$PD_BRA_NM)
df_order$PD_BRA_NM <- gsub('아디다스키즈', '아디다스', df_order$PD_BRA_NM)
df_order$PD_BRA_NM <- gsub('아디다스 언더웨어', '아디다스', df_order$PD_BRA_NM)
df_order$PD_BRA_NM <- gsub('아디다스바이알렉산더왕', '아디다스',
df_order$PD_BRA_NM)
```

이 외에도, 여러 브랜드명을 통합하여 분석에 사용할 수 있도록 했으며 분석의 왜곡을 방지하는 기초적인 데이터 정제를 수행하였다.

본 분석팀은 고객들의 소비 패턴을 분석하고 개인화 서비스를 제공하기 위해 각 고객의 특징을 파악할 수 있는 다양한 파생변수가 필요하다는 판단 하에, 다양한 파생 변수를 생성했다. 파생변수 생성은 크게 두 가지 방향으로 진행되었는데, 새로운 대분류 생성 후 구매횟수와 구매 비율 생성, 각 고객의 특징을 나타낼 수 있는 파생변수 추가 순으로 진행했다.

(4) 상품 대분류 통합 및 선호 상품군 파생변수 추가

기존에 37개로 지나치게 세분화되어 분류되어 있던 대분류를 비슷한 성향끼리 묶어 10개의 대대분류로 만들었다. 고객들이 어떤 카테고리의 제품을 많이 사는지 파악하기 위한 사전 작업으로, 상품 테이블인 product 테이블의 CLAC1_NM을 다음의 기준으로 묶었다. 그 후, 고객 테이블인 CLIENT, 상품 테이블인 PRODUCT, 주문 테이블인 ORDER를 조인했다. 이를 통해, 고객들의 카테고리별 상품 구매 횟수 및 타 카테고리 대비 구매비율을 새로운 변수로 추가해 고객들의 선호 상품군을 파악했다. 이렇게 구한 구매 횟수와 구매 비율은 추후 고객들의 구매 성향을 나누는 클러스터링, 레이더차트 등에서 고객군의 선호 상품군, 개인화된 선호 상품군 등을 구하는 데 활용했다.

새로운 대분류	기존 대분류	구매 횟수	구매 비율
전자제품 (C1)	계절가전, 냉장/세탁가전, 모바일, 생활/주방가전, 영상/음향가전, 컴퓨터	C1_CT	전자제품
리빙 (C2)	세제/위생, 식기/조리기구, 청소/세탁/욕실용품, 가구, 인테리어/조명, 침구/수예, 주방잡화	C2_CT	리빙
식품 (C3)	건강식품, 과일, 냉동식품, 냉장식품, 음료, 축산물	C3_CT	식품
스포츠 (C4)	구기/필드스포츠, 시즌스포츠, 아웃도어/레저, 헬스/피트니스	C4_CT	스포츠
패션 (C5)	남성의류, 속옷/양말/홈웨어, 스포츠패션, 여성의류, 패션잡화	C5_CT	패션
뷰티 (C6)	퍼스널케어, 화장품/뷰티케어	C6_CT	뷰티
유아용품 (C7)	완구, 유아동의류, 출산/육아용품	C7_CT	유아용품
취미용품 (C8)	자동차용품, 원예/애완	C8_CT	취미용품
사무용품 (C9)	문구/사무용품	C9_CT	사무용품
상품권 (C10)	상품권	C10_CT	상품권

<표: 상품 대분류 통합 및 구매 파생변수 생성>

대분류 통합 및 구매 변수를 추가하기 위한 작업은 다음과 같이 진행했다.

```
# 새로운 대대분류 추가
df_master[df_master$CLAC1_NM %in% c('계절가전', '냉장/세탁가전', '모바일',
'생활/주방가전', '영상/음향가전', '컴퓨터'), 'CN_our'] <- 'C1'
df_master[df_master$CLAC1_NM %in% c('세제/위생', '식기/조리기구', '청소/세탁/욕실용품',
'가구', '인테리어/조명', '침구/수예', '주방잡화'), 'CN_our'] <- 'C2'
df_master[df_master$CLAC1_NM %in% c('건강식품', '과일', '냉동식품', '냉장식품', '음료',
'축산물'), 'CN_our'] <- 'C3'
df_master[df_master$CLAC1_NM %in% c('구기/필드스포츠', '시즌스포츠', '아웃도어/레저',
'헬스/피트니스'), 'CN_our'] <- 'C4'
df_master[df_master$CLAC1_NM %in% c('남성의류', '속옷/양말/홈웨어', '스포츠패션',
'여성의류', '패션잡화'), 'CN_our'] <- 'C5'
df_master[df_master$CLAC1_NM %in% c('퍼스널케어', '화장품/뷰티케어'), 'CN_our'] <- 'C6'
```

```
df_master[df_master$CLAC1_NM %in% c('완구', '유아동의류', '출산/육아용품'), 'CN_our'] <- 'C7'
df_master[df_master$CLAC1_NM %in% c('자동차용품', '원예/애완'), 'CN_our'] <- 'C8'
df_master[df_master$CLAC1_NM %in% c('문구/사무용품'), 'CN_our'] <- 'C9'
df_master[df_master$CLAC1_NM %in% c('상품권'), 'CN_our'] <- 'C10'

# 코드에 해당하는 이름변수 추가
df_master[df_master$CN_our == 'C1', 'CN_Name'] <- '전자제품'
df_master[df_master$CN_our == 'C2', 'CN_Name'] <- '리빙'
df_master[df_master$CN_our == 'C3', 'CN_Name'] <- '식품'
df_master[df_master$CN_our == 'C4', 'CN_Name'] <- '스포츠'
df_master[df_master$CN_our == 'C5', 'CN_Name'] <- '패션'
df_master[df_master$CN_our == 'C6', 'CN_Name'] <- '뷰티'
df_master[df_master$CN_our == 'C7', 'CN_Name'] <- '유아용품'
df_master[df_master$CN_our == 'C8', 'CN_Name'] <- '취미용품'
df_master[df_master$CN_our == 'C9', 'CN_Name'] <- '사무용품'
df_master[df_master$CN_our == 'C10', 'CN_Name'] <- '상품권'

# 고객별 총 구매수량 (CLNT_BUY_CT) 변수 추가
df_buy_ct <- train_order %>%
  group_by(CLNT_ID) %>%
  summarise(CLNT_BUY_CT = sum(PD_BUY_CT))

client <- inner_join(client, df_buy_ct, by='CLNT_ID')

# 고객별 총 구매액 (CLNT_BUY_TOT) 변수 추가
df_buy_tot <- train_order %>%
  group_by(CLNT_ID) %>%
  summarise(CLNT_BUY_TOT = sum(PD_BUY_TOT))

client <- inner_join(client, df_buy_tot, by='CLNT_ID')
```

(5) 파생변수 생성

앞서 밝힌 바와 같이 본 분석팀은 고객들의 특성을 나타낼 수 있는 다양한 파생변수를 고안해냈다. 각 데이터 프레임당 생성한 파생변수는 다음 표와 같다.

Data Frame	변수명	한글명	내용
고객 (Client)	HIT_AVG	세션 평균 히트수	CLNT_ID(클라이언트ID)당 히트 수 평균
	PAG_VIEW_AVG	세션 평균 페이지뷰	CLNT_ID(클라이언트ID)당 페이지 뷰 평균
	SESS_CT	세션 수	CLNT_ID(클라이언트ID)당 생성 세션 수
	CLNT_BUY_CT	총 구매 상품 수	CLNT_ID(클라이언트ID)당 구매 수량
	CLNT_BUY_TOT	총 구매액	CLNT_ID(클라이언트ID)당 구매한 금액
	CLNT_CL	고객 분류 번호	고객 분류 (1: 10, 20대 남 / 2: 10대 여 / 3: 20대 여 / 4: 30대 남 / 5: 30대 여 / 6: 40, 50대 남/ 7: 40, 50대 여/ 8: 60, 70, 80대 남/ 8: 60, 70, 80대 여)
	Cluster_num	클러스터링 결과	각 CLNT_CL 별 2개의 군집
상품 (Master)	CN_our	대분류	대분류 코드(C1~C10)
	CN_Name	대분류명	대분류명
세션 (Session)	DAY_OF_WEEK	구매 요일	구매 일자에 해당하는 요일
주문 (Order)	ORDER_ID	주문 ID	하나의 주문에 대한 고유한 ID
	PD_BUY_TOT	총 금액	상품 단위 당 금액 x 구매 개수 (원)

<표: 데이터 프레임별 생성한 파생변수>

먼저, 고객테이블에는 고객들의 행동을 알 수 있는 SESSION 테이블과 고객들이 주문한 ORDER 테이블, 그리고 고객들의 인구통계학적 데이터를 가진 CLIENT 테이블을 조인해 각 고객들의 행동 변수를 추가했다. 세션 평균 히트수는 각 고객이 한 세션 동안 기록한 평균 히트수, 세션 평균 페이지뷰는 각 고객이 한 세션당 기록한 평균 페이지뷰를, 히트수는 고객이 접속한 횟수인 세션 생성 수를 나타낸다. 고객들의 주문 테이블과 조인하여 생성한 파생변수로는 각 고객들이 구매한 구매 수량, 각 고객들이 구매한 금액의 총액을 말한다. 고객 분류 변호와 클러스터링 결과는 V-2. 고객분류(ANOVA, 클러스터링)에서 자세하게 다루겠지만 명목형 변수인 성별, 연령을 분산분석을 통해 고객들의 구매 상품 성향과 구매액 차이에 따라 고객을 분류하여 1~10까지 고객 분류 번호를 매겼다. 또한 선호 상품군에 따라 클러스터링을 수행해 패션형, 뷰티형, 생활형으로 구분해 클러스터링 번호를 부여했다.

```
# 고객별 총 구매수량 (CLNT_BUY_CT)
df_buy_ct <- train_order %>%
  group_by(CLNT_ID) %>%
  summarise(CLNT_BUY_CT = sum(PD_BUY_CT))

client <- inner_join(client, df_buy_ct, by='CLNT_ID')

# 고객별 총 구매액 (CLNT_BUY_TOT)
df_buy_tot <- train_order %>%
```

```
  group_by(CLNT_ID) %>%
  summarise(CLNT_BUY_TOT = sum(PD_BUY_TOT))

client <- inner_join(client, df_buy_tot, by='CLNT_ID')

# 고객별 총 세션 생성 수 (SESS_CT)
df_sess_ct <- train_session %>%
  group_by(CLNT_ID) %>%
  summarise(SESS_CT = n())

client <- inner_join(client, df_sess_ct, by='CLNT_ID')

# 고객별 세션 평균 히트수 (HIT_AVG)
df_avg_hit <- train_order %>%
  group_by(CLNT_ID) %>%
  summarise(HIT_AVG = round(mean(HIT_SEQ)))
client <- inner_join(client, df_avg_hit, by='CLNT_ID')

# 고객별 세션 평균 페이지뷰수 (PAG_VIEW_AVG)
df_avg_page <- train_session %>%
  group_by(CLNT_ID) %>%
  summarise(PAG_VIEW_AVG = round(mean(TOT_PAG_VIEW_CT)))

client <- inner_join(client, df_avg_page, by='CLNT_ID')
```

상품 테이블에는 앞서 '상품 대분류 통합 및 선호 상품군 파생변수 추가'에서 밝힌 바와 같이 대분류 코드와 대대분류명을 넣어주었다. 세션 테이블에는 각 세션일자에 해당하는 요일을 파생변수로 생성하였으며, 주문 테이블에는 PK인 주문ID, 상품 단위당 금액과 구매 개수를 곱한 총 금액 변수를 추가하였다.

```
# session에 요일변수 추가
# SESS_DT: POSIX to Date
df_session$SESS_DT <- as.Date(df_session$SESS_DT, tz = '', tryFormats = c('%Y%m%d',
'%Y-%m-%d'))

# session의 날짜 정보 담는 벡터 생성
sess_date_list <- df_session$SESS_DT
```

```
# 요일 추가, Session 테이블 일자는 '-'이 들어간 세션일자
df_session <-
  df_session %>%
  mutate(DAY_OF_WEEK=weekdays(strptime(sess_date_list, format="%Y-%m-%d")) )

table(is.na(df_session$DAY_OF_WEEK))

# 각 대대분류별, 구매 수는 Cn_CT, 구매비율은 한글로 파생변수 생성
cn1 <- train_order %>%
  mutate(cn1 = ifelse(train_order$CN_our == 'C1', PD_BUY_CT, 0)) %>%
  group_by(CLNT_ID) %>%
  summarise(CLNT_BUY_CT = sum(PD_BUY_CT),
            C1_CT = sum(cn1),
            전자제품 = round(C1_CT/CLNT_BUY_CT, 2)) %>%
  select(CLNT_ID, 전자제품, C1_CT)

cn2 <- train_order %>%
  mutate(cn2 = ifelse(train_order$CN_our == 'C2', PD_BUY_CT, 0)) %>%
  group_by(CLNT_ID) %>%
  summarise(CLNT_BUY_CT = sum(PD_BUY_CT),
            C2_CT = sum(cn2),
            리빙 = round(C2_CT/CLNT_BUY_CT, 2)) %>%
  select(CLNT_ID, 리빙, C2_CT)

cn3 <- train_order %>%
  mutate(cn3 = ifelse(train_order$CN_our == 'C3', PD_BUY_CT, 0)) %>%
  group_by(CLNT_ID) %>%
  summarise(CLNT_BUY_CT = sum(PD_BUY_CT),
            C3_CT = sum(cn3),
            식품 = round(C3_CT/CLNT_BUY_CT, 2)) %>%
  select(CLNT_ID, 식품, C3_CT)

cn4 <- train_order %>%
  mutate(cn4 = ifelse(train_order$CN_our == 'C4', PD_BUY_CT, 0)) %>%
  group_by(CLNT_ID) %>%
  summarise(CLNT_BUY_CT = sum(PD_BUY_CT),
            C4_CT = sum(cn4),
            스포츠 = round(C4_CT/CLNT_BUY_CT, 2)) %>%
```

```
  select(CLNT_ID, 스포츠, C4_CT)

cn5 <- train_order %>%
  mutate(cn5 = ifelse(train_order$CN_our == 'C5', PD_BUY_CT, 0)) %>%
  group_by(CLNT_ID) %>%
  summarise(CLNT_BUY_CT = sum(PD_BUY_CT),
            C5_CT = sum(cn5),
            패션 = round(C5_CT/CLNT_BUY_CT, 2)) %>%
  select(CLNT_ID, 패션, C5_CT)

cn6 <- train_order %>%
  mutate(cn6 = ifelse(train_order$CN_our == 'C6', PD_BUY_CT, 0)) %>%
  group_by(CLNT_ID) %>%
  summarise(CLNT_BUY_CT = sum(PD_BUY_CT),
            C6_CT = sum(cn6),
            뷰티 = round(C6_CT/CLNT_BUY_CT, 2)) %>%
  select(CLNT_ID, 뷰티, C6_CT)


cn7 <- train_order %>%
  mutate(cn7 = ifelse(train_order$CN_our == 'C7', PD_BUY_CT, 0)) %>%
  group_by(CLNT_ID) %>%
  summarise(CLNT_BUY_CT = sum(PD_BUY_CT),
            C7_CT = sum(cn7),
            유아용품 = round(C7_CT/CLNT_BUY_CT, 2)) %>%
  select(CLNT_ID, 유아용품, C7_CT)


cn8 <- train_order %>%
  mutate(cn8 = ifelse(train_order$CN_our == 'C8', PD_BUY_CT, 0)) %>%
  group_by(CLNT_ID) %>%
  summarise(CLNT_BUY_CT = sum(PD_BUY_CT),
            C8_CT = sum(cn8),
            취미용품 = round(C8_CT/CLNT_BUY_CT, 2)) %>%
  select(CLNT_ID, 취미용품, C8_CT)


cn9 <- train_order %>%
  mutate(cn9 = ifelse(train_order$CN_our == 'C9', PD_BUY_CT, 0)) %>%
  group_by(CLNT_ID) %>%
  summarise(CLNT_BUY_CT = sum(PD_BUY_CT),
```

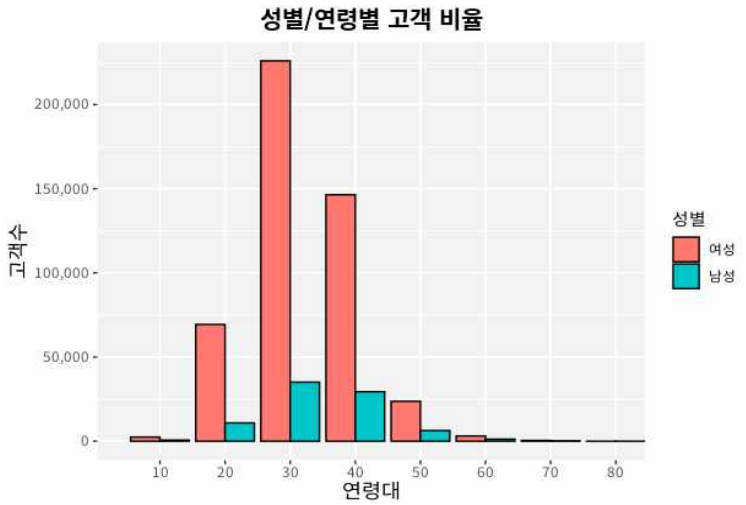
```
            C9_CT = sum(cn9),
            사무용품 = round(C9_CT/CLNT_BUY_CT, 2)) %>%
  select(CLNT_ID, 사무용품, C9_CT)


cn10 <- train_order %>%
  mutate(cn10 = ifelse(train_order$CN_our == 'C10', PD_BUY_CT, 0)) %>%
  group_by(CLNT_ID) %>%
  summarise(CLNT_BUY_CT = sum(PD_BUY_CT),
            C10_CT = sum(cn10),
            상품권 = round(C10_CT/CLNT_BUY_CT, 2)) %>%
  select(CLNT_ID, 상품권, C10_CT)
```

4. 탐색적 데이터 분석 및 시각화

전처리가 완료된 데이터로 탐색적 데이터 분석(EDA)을 수행했다. 생성한 테이블을 살펴본 결과 고객, 세션, 주문 테이블을 주 테이블로 선정하여 이에 대해 기초 분석을 시행하였다.

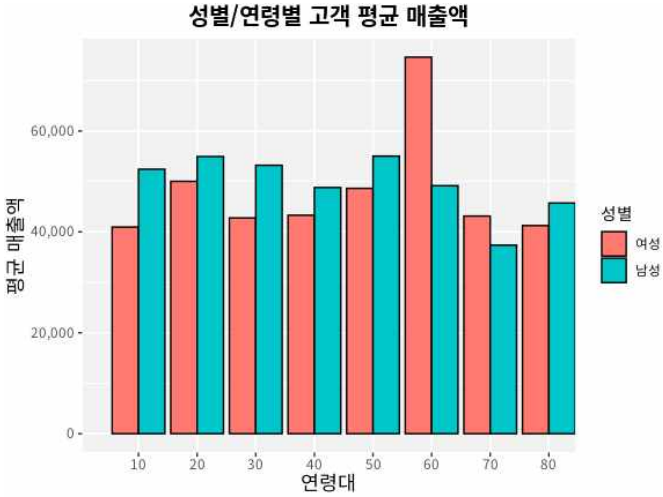
(1) 고객 데이터 탐색

고객 데이터프레임이 갖는 변수로는 성별과 연령대 정보가 있다. 기본적으로 이들의 분포를 살펴보면 아래의 그래프와 같다.



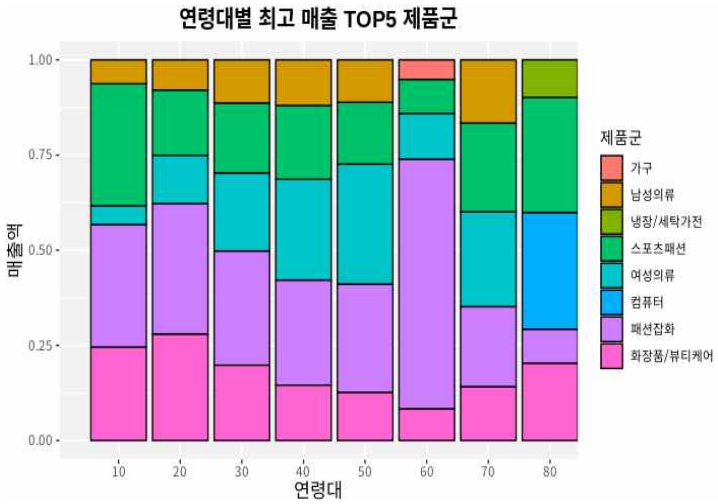
<그래프: 성별/연령별 고객 비율>

그래프 상의 분포로 보아, 여성이 롯데 쇼핑몰의 주요 고객층임을 확인할 수 있고, 그 중 특히 20~40대 여성이 많은 비중을 차지함을 확인할 수 있다. 고객 비율과 각 성별/연령별 고객 총 매출액은 동일한 패턴을 보이나, 평균 매출액을 비교해본다면, 평균 매출액은 60·70대를 제외하고 남성의 매출액이 높음을 확인하였다.



<그래프: 성별/연령별 고객 평균 매출액>

위의 그래프로 보아, 60대 여성이 눈에 띄게 평균 매출액이 많음을 확인할 수 있다. 60대 여성 또한 롯데에서 가장 많이 판매되는 패션을 많이 구매했다. 각 연령대별 매출이 높은 대분류 제품군을 살펴보면 아래 그림과 같다.

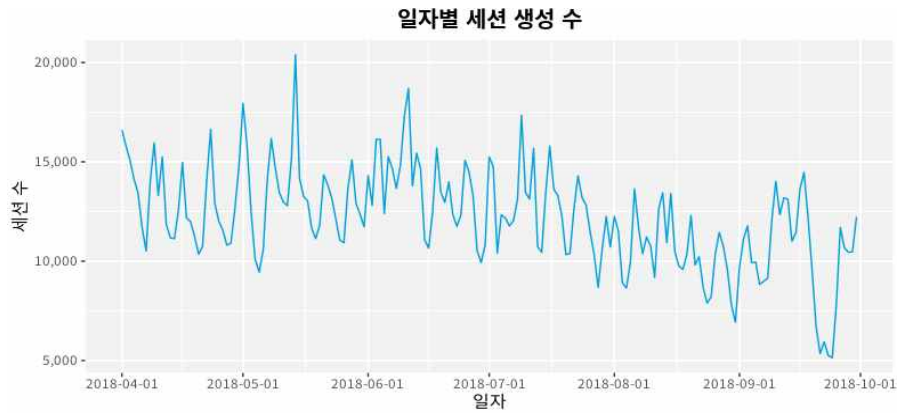


<그래프: 연령대별 최고 매출 상위 5개 제품군>

그래프를 살펴보면 가격대가 높은 냉장/세탁가전과 컴퓨터가 상위 매출 제품군에 포함되며, 그 외에는 대부분 패션관련 제품군임을 확인할 수 있다. 높은 매출을 기록하는 브랜드를 뽑아본다면, 패션 제품들을 제외하고 LG전자가 가장 높은 매출을 기록하였고, 제품군에서는 볼 수 없었던 홍삼 분류에 속하는 정관장이 추출되었다.

(2) 세션 데이터 탐색

세션 데이터는 고객의 행동 정보를 파악할 수 있는 데이터이다. 세션일자를 갖는 시계열 데이터로, 시간에 따른 고객의 행동 혹은 주문정보를 확인할 수 있다. 기초적인 탐색을 위해 시간의 흐름에 따른 세션의 생성 수 변화를 그래프로 나타낸 결과는 아래와 같다.



<그래프: 일자별 세션 생성 수>

위의 그래프로 보아 5월에 최고점을, 9월 중순쯤 최저점을 기록하였다. 5월에는 가정의 달로 어린이날, 어버이날 등 여러 이벤트 준비를 위해 사이트의 방문이 증가하여 최고점을 기록했다. 2018년 9월 중순 중 9월 20일부터 세션의 생성 수가 급감하였고, 이는 9월 22일인 토요일부터 26일인 수요일까지 추석 연휴로 인한 20일쯤부터의 배송 접수 마감으로 인한 온라인 방문 및 주문이 감소한 것으로 추정된다.

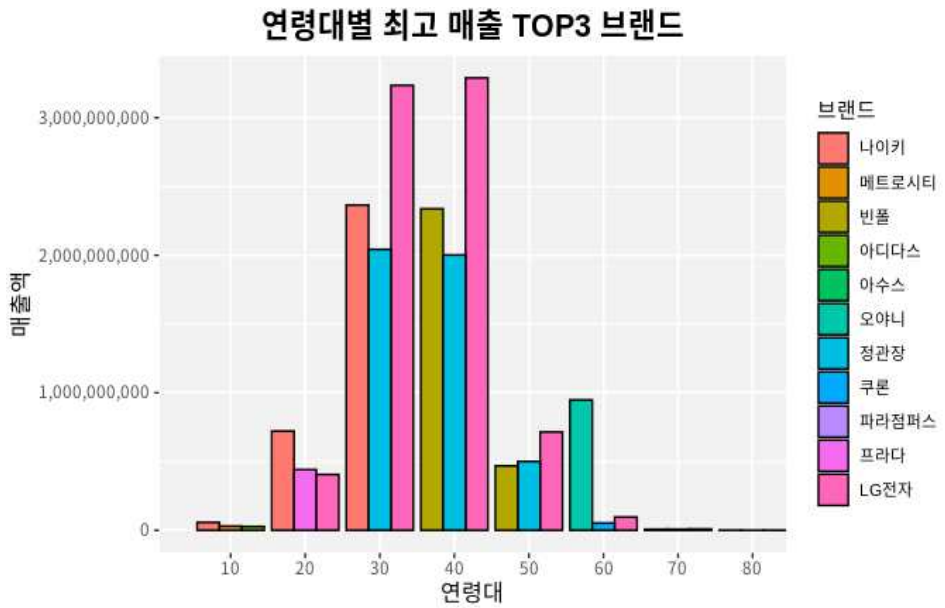
월별 매출이 높은 제품군 3개씩을 살펴보았을 때, 4~9월 간 매출은 1위를 유지하였다. 이외의 제품군들로는, 스포츠패션과 패션잡화의 패션 카테고리에 속하는 분류들과 화장품/뷰티케어의 뷰티 카테고리가 나타나 롯데의 주 고객인 여성의 수요를 확인할 수 있었다. 결과적으로, 총 37개의 대분류 중 높은 매출을 기록하는 상품군은 한정되어 있다는 한계점이 도출된다. 월별 상위 매출 제품군을 시각화한 결과는 아래와 같다.

<그래프: 월별 최고 매출 상위 3개 제품군>

이전에 살펴본 바와 같이, 월별 상위 매출액 제품군에서 패션에 집중된 결과를 볼 수 있다. 해당 그래프로 월별 매출에서 1위를 기록한 패션잡화는 여성 가방으로 인해 매출이 높았음을 확인할 수 있다. 이는 롯데닷컴의 백화점과 같은 고급 브랜드 이미지와 명품가방의 높은 가격대를 그 이유로 분석했다. 특징적인 점은 계절가전제품인 냉방가전이 본격적인 더위가 시작되기 전인 7월에 매출이 높음을 볼 수 있다. 9월에는 홍삼/인삼가공식품이 나와 추석 선물로 인기 있음을 확인할 수 있다. 앞의 매출이 높은 브랜드를 살펴보았을 때, 정관장이 추출된 것과 일치하였다.

(3) 주문 및 검색 데이터 탐색

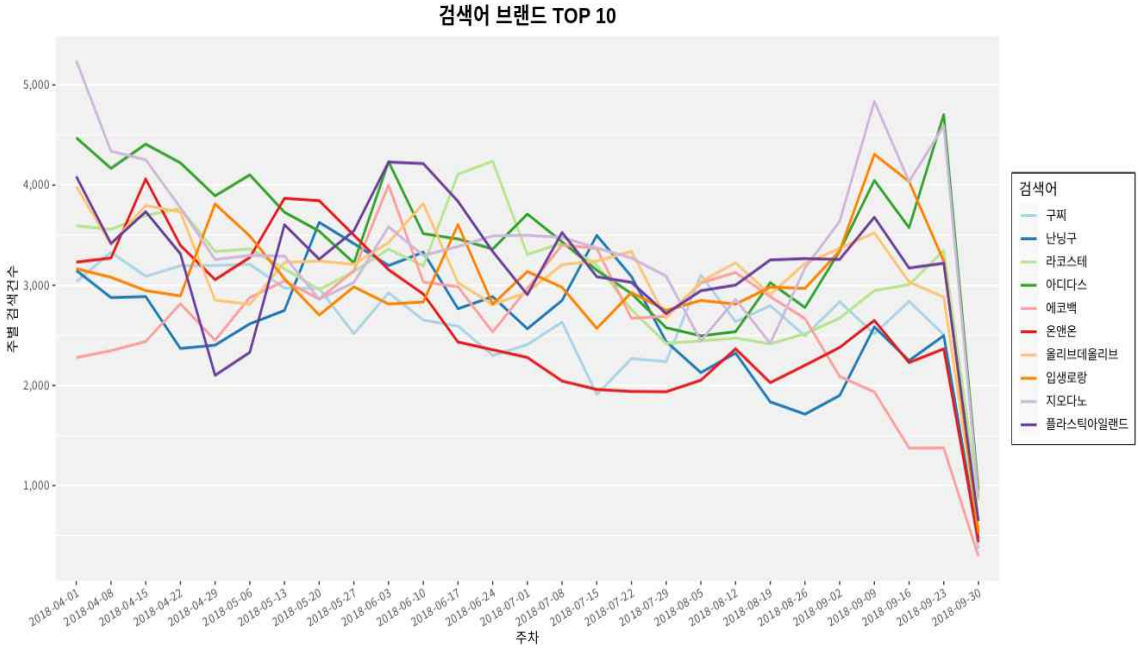
주문 데이터는 각 주문의 상품별 구매 수량과 같은 정보를 담고 있다. 이 중 브랜드에 대해 살펴보려고 한다. 연령대별 최고 매출을 기록한 상위 3개의 브랜드를 시각화한 결과는 아래와 같다.



<그래프: 연령대별 최고 매출 상위 3개 브랜드>

LG전자는 10대와 80대를 제외한 모든 연령대에서 상위 매출 브랜드에 속해, 롯데의 전자제품 중 주력 제품임을 확인할 수 있다. 10대는 스포츠 브랜드를 주로 구매하며, 이 중 나이키는 10~30대의 연령대에서 인기가 높은 스포츠 브랜드이다. 패션잡화는 10대는 메트로시티, 20대는 프라다, 60대는 오야니, 쿠론이 인기가 높으며, 빈폴은 중년의 인기가 높음을 확인할 수 있었다.

검색 데이터에 대한 탐색을 진행한 결과, 검색량이 많은 데이터를 출력한 후 이들의 주차별 변화를 살펴보았다.



<그래프: 검색량 상위 10위 브랜드의 검색건수 추이>

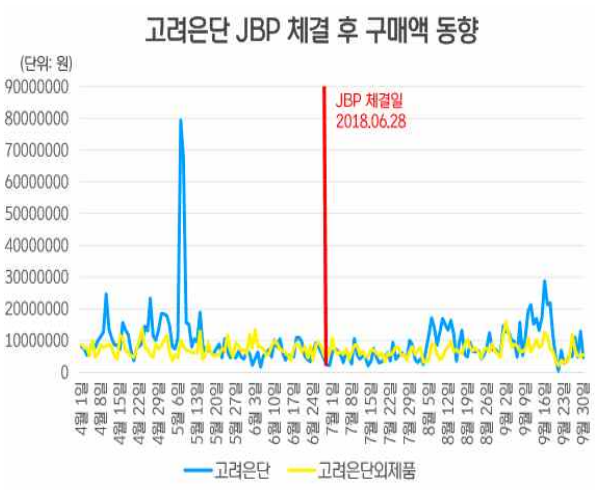
상위 검색어 10개 브랜드는 아디다스, 지오다노, 플라스틱아일랜드, 올리브 데 올리브, 라코스테, 입생로랑, 구찌, 온앤온, 에코백, 난닝구로 검색건수 추이가 비슷함을 알 수 있다.

Ⅳ. 문제 원인 파악

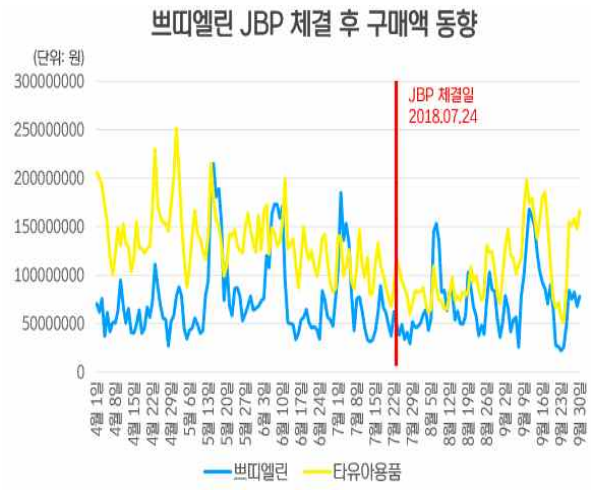
1. 미미한 업무 제휴 협약(JBP) 효과

롯데는 2018년 6월 28일, 7월 24일에 각각 프리미엄 유아용품제조 기업 쁘띠엘린 및 고려은단과 업무 제휴 협약을 체결했다. 업무 제휴 협약(Joint Business Plan)이란 상품개발 단계부터 유통회사와 제조회사가 함께 소비자를 분석하고 판매데이터를 공유해 고객에게 최적의 상품과 쇼핑환경을 제공하는 기업 간 파트너십을 뜻한다. 또한, 롯데와 협약을 맺은 제휴사는 서로의 강점을 기반으로 단독 상품개발을 위해 상호 협력하는 것을 볼 수 있었다. 공동 마케팅으로 고객에게 차별화된 서비스를 제공함과 동시에 신상품을 대상으로 대형행사를 정기적으로 진행하며, 자사 고객에 최적화된 타겟 마케팅도 진행한다.

본 분석팀은 업무 제휴 협약(JBP)의 효과를 보기 위해 쁘띠엘린/고려은단 및 해당 브랜드가 속한 카테고리 내 제품들의 판매액 동향을 시각화 해 보았다. 단순 추세 비교를 위해 쁘띠엘린/고려은단의 총 판매액에 특정 숫자를 곱했고, 해당 브랜드 외 제품은 실제 판매액이다.



<그래프: 고려은단 구매액 동향>



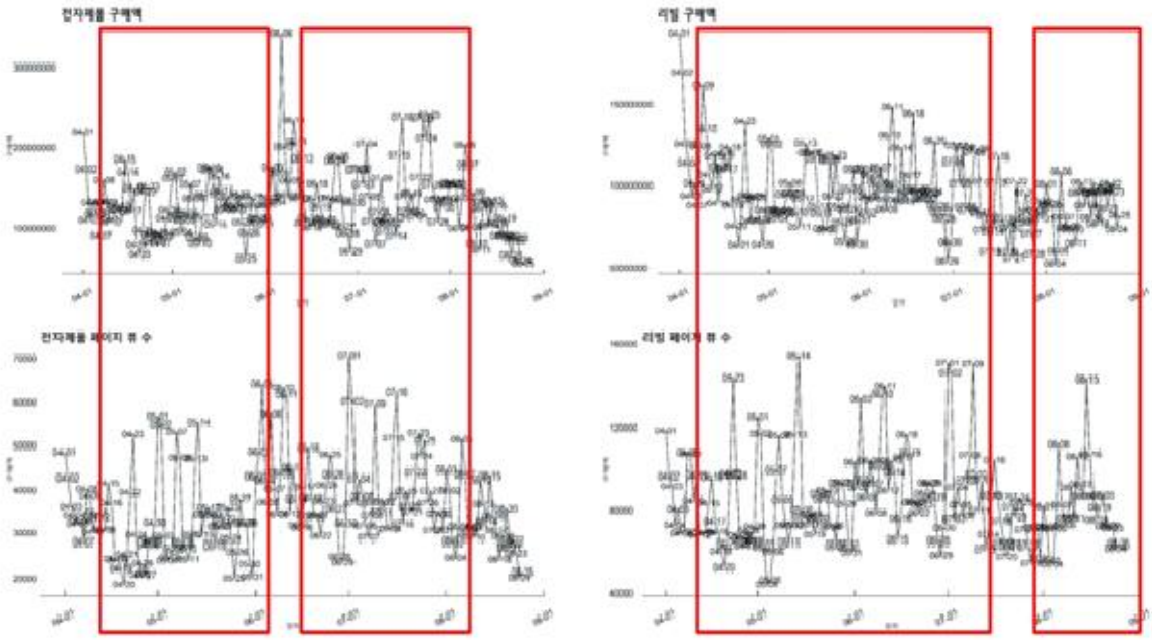
<그래프: 쁘띠엘린 구매액 동향>

그래프를 통해 도출한 인사이트는 다음과 같다.

1. 업무 제휴 협약이 지속적인 판매액 상승 곡선을 야기하지 않았다.
2. 업무 제휴 협약 브랜드의 판매액과 해당 브랜드가 속한 카테고리 내 제품의 판매액을 비교했을 때, 추세의 차이가 미미했다.
3. 업무 제휴를 통한 매출액 향상을 위해선 다른 마케팅 전략이 필요할 것으로 보인다.

2. 미흡한 홈페이지 구성

대대분류 카테고리에 속한 상품의 총 판매액과 관련된 세션 데이터를 분석하면서, 상품의 총 구매액과 고객의 행동 데이터 간의 모순점을 발견할 수 있었다. 페이지 뷰 수와 구매액의 추세가 일치하지 않다는 점이다. 우리는 이에 대한 이유로 고객이 홈페이지에서 원하는 제품을 구매할 때 많은 시간을 할애했거나 중도 포기했을 것이라고 추측했다. 이에 따라, 본 분석팀은 홈페이지에 대한 문제점을 파악하기 위해 원 데이터가 수집된 기간인 2018년 4~9월 당시
롯데 온라인 쇼핑몰 홈페이지의 UI를 살펴보았으며 다음과 같은 문제점을 발견했다:



<그래프: 제품군별 시계열 분석>

1. 복잡한 상품 카테고리 분류 방식
2. 메인페이지 및 세부 페이지 상품 추천 개인화 서비스 부재
3. 진부한 페이지 구성으로 인한 고객 흥미 유발 실패

먼저, 복잡한 상품 카테고리 분류로 고객이 원하는 상품을 찾기 위해 많은 시간을 할애해야 하며, 웹사이트 탐색에 어려움을 주고 있다. 또한, 고객 맞춤 페이지 구성이 아닌 무분별한 상품군 나열로 고객의 흥미 유발 및 구매를 유도하지 못하고 있다. 온라인 쇼핑몰 시장 경쟁이 치열해짐에 따라, 경쟁사와는 다른 차별화된 페이지 구성으로 고객의 마음을 끄는 것이 중요하다 생각된다. 이와 같은 문제점을 바탕으로, 다음 장에서는 온라인몰의 UI/UX 개선 및 매출 향상을 위한 추천 전략을 제안하고자 한다.

Ⅴ. 고급 분석 및 추천 전략

1. 고객 세분화

고객별 상품 추천 개인화 서비스를 위해, 비슷한 특성을 가진 고객들은 집단으로 묶어 분류하였으며, 고객 세분화 단계는 다음과 같다.

Normal vs Vip	성별/연령별 분류	Normal / Vip 구매 금액별 ANOVA	선호 상품별 고객 분류
고객별 구매 금액의 사분위수로 판정 -> 6개월 동안 641,450원 이상 구매한 고객을 Vip로 정의	Normal/Vip 각각 16그룹으로 분류 1. 10대 남성 2. 10대 여성 3. 20대 남성 4. 20대 여성 5. 30대 남성 6. 30대 여성 7. 40대 남성 8. 40대 여성 9. 50대 남성 10. 50대 여성 11. 60대 남성 12. 60대 여성 13. 70대 남성 14. 70대 여성 15. 80대 남성 16. 80대 여성	앞에서 나눈 집단을 ANOVA로 차이가 없는 집단은 통합 1. 10~20대 남성 2. 10대 여성 3. 20대 여성 4. 30대 남성 5. 30대 여성 6. 40~50대 남성 7. 40~50대 여성 8. 60~80대 남성 9. 60~80대 여성 10. Vip	상품군 구매 개수를 변수로 한 클러스터링으로 고객 분류 1, 패션형 2. 뷰티형 3. 생활형

(1) 구매 금액을 기준으로, 이상치를 확인하여 Normal과 Vip 고객으로 분류한다.

(2) 상품 추천 시, 연령대와 성별은 기본적으로 고려되어야 한다고 판단되어, Vip와 Normal 고객을 각각 10~80대 & 남/여로 세분화하여 총 32개의 집단으로 분류한다.

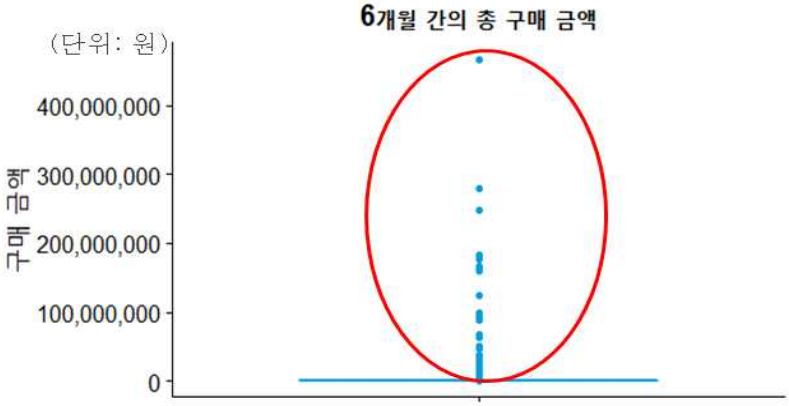
(3) 세분화한 Normal과 Vip고객을 구매금액별 ANOVA 분석을 통해, 차이가 없는 집단은 통합한다.

(4) 카테고리별 구매 개수를 변수로 한 클러스터링을 통해 선호 상품별로 고객을 분류한다.

위와 같은 고객 세분화를 통해 총 30개의 세분화된 고객 집단으로 분류하였으며, 추후 집단별 선호 상품군을 토대로 상품을 추천하여 홈페이지를 구성할 것이다.

Step 1. 고객별 구매 금액의 사분위수 확인

```
quantile(df_client$CLNT_BUY_TOT)
0%      25%      50%      75%     100%
100    63200   129445   294500 393030100
```



<그래프: 6개월 간 총 구매 금액 Box Plot>

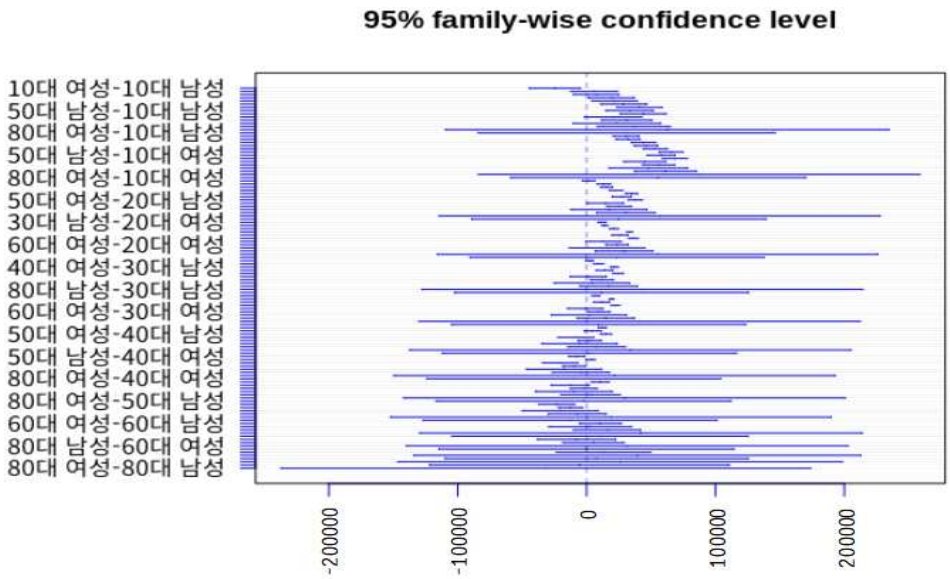
- [Q3 + 1.5 * IQR]이상의 경우 이상치로 규정하여 Vip로 분할
- 294500 + 1.5 * (294500 - 63200) = 641450
- 641,450원 이상 구매한 고객을 Vip로 정의한다.

Step 2. Normal Client 분산분석

```
fit.aov <- aov(CLNT_BUY_TOT ~ CLNT_CL, data = normal_client)
               Df         Sum Sq          Mean Sq F value       Pr(>F)
CLNT_CL     15    69942756299453   4662850419964     240.2 <0.0000000000000002
Residuals 496621 9640037505106198     19411256280

CLNT_CL       ***
Residuals
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

- 대립가설 채택으로 집단간 평균 구매금액은 같지 않으며,
  어떤 집단간 차이가 있는지 보기 위해 사후분석 수행

```
# 사후분석 - 평균의 다중비교
fit.glht <- glht(fit.aov, linfct = mcp(CLNT_CL = 'Tukey'))
fit.mcp <- TukeyHSD(fit.aov)
plot(fit.mcp, col = 'mediumblue')
```



<그래프: 고객집단별 구매액 ANOVA>

ANOVA 결과를 토대로 평균 구매 금액이 차이가 없는 집단은 통합한 후,
분류변수(CLNT_CL) 추가

그룹번호	1	2	3
해당 집단	10~20대 남성	10대 여성	20대 여성
그룹번호	4	5	6
해당 집단	30대 남성	30대 여성	40~50대 남성
그룹번호	7	8	9
해당 집단	40~50대 여성	60~80대 남성	60~80대 여성

<표: 세분화한 고객 그룹 분류 번호>

Step 3. Vip Client 분산분석

```
fit.aov2 <- aov(CLNT_BUY_TOT ~ CLNT_CL, data = vip_client)
```

성별 연령별 총 구매 금액으로 ANOVA 분석을 실시했으나,
평균 구매 금액이 큰 차이를 보이지 않아 VIP 고객은 따로 세분화하지 않음
일반 고객 분류 번호 9번 다음으로, VIP를 10번 그룹으로 지정

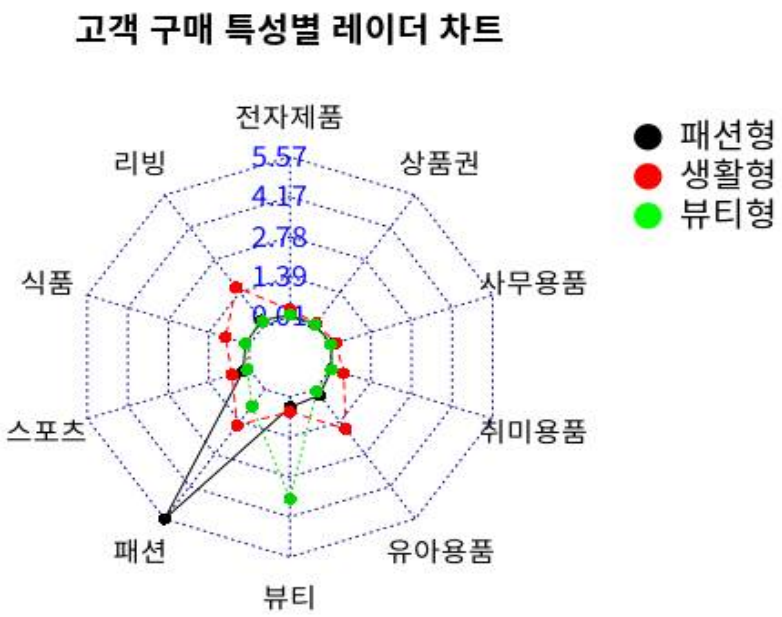
Step 4. 상품군 구매 개수를 변수로 클러스터링 한 고객 분류 (제품 선호도별 분류)

모든 고객을 담은 테이블 생성

```
all <- rbind(normal_client, vip_client)
```

```
# 상품 구매 비율 변수만으로 클러스터링 시행하기 위해 테이블 생성
df_cluster <- all[, c(9:18)]
# 샘플링
set.seed(1234)
n <- nrow(df_cluster)
sample <- sample(n, replace = F, .01*n)
# 클러스터 개수 구하기
df_cluster_test <- df_cluster[sample, ]
# nc <- NbClust(df_cluster_test, method = 'kmeans')
# NbClust 수행 결과 3이 나옴
# 클러스터링 수행
set.seed(1234)
kmc <-kmeans(df_cluster, centers = 3)
# 클러스터링 변수 추가
all$Cluster_CL <- kmc$cluster

# 3개의 집단으로 클러스터링한 결과를 레이더 차트로 선호 상품군 비교하기
# radar 차트를 그리기 위한 data frame 구성
t_radar <- all %>%
  group_by(Cluster_CL) %>%
  summarise(
    전자제품 = mean(C1_CT),
      리빙 = mean(C2_CT),
      식품 = mean(C3_CT),
      스포츠 = mean(C4_CT),
      패션 = mean(C5_CT),
      뷰티 = mean(C6_CT),
      유아용품 = mean(C7_CT),
      취미용품 = mean(C8_CT),
      사무용품 = mean(C9_CT),
      상품권 = mean(C10_CT)
  )
  t_radar <- as.data.frame(t_radar)
  t_radar <- t_radar[-1] # CLNT_CL 지워줌
  a <- range(t_radar)
  t_radar <- rbind(rep(a[2], 10), rep(a[1], 10), t_radar)
```

```
  rownames(t_radar) <- c('Max', 'Min', 'Cluster1', 'Cluster2', 'Cluster3')
# radar chart를 통한 cluster별 고객 특성 파악
radarchart(t_radar,
           axistype = 1,
           caxislabels = c( round(a[1], 2),   # 최소값
                            round((a[2]-a[1])*0.25, 2), # 25%
                            round((a[2]-a[1])*0.50, 2), # 50%
                            round((a[2]-a[1])*0.75, 2), # 75%
                            round(a[2], 2) ),# 100%
           title = '고객 구매 특성별 레이더 차트'
)
colors_in = c('black', 'red', 'green')
a <- '패션형'
b <- '생활형'
c <- '뷰티형'
legend(x='topright', y=1,
       legend=c(a, b, c),
       bty='n',pch=20, col=colors_in, text.col='black', cex=1.2,pt.cex=3 )
```



<그림 : 클러스터링 결과 레이터 차트로 제품 선호도 비교>

총 3개의 집단으로 클러스터링한 결과 크게 패션형, 생활형, 뷰티형 집단으로 분류함

2. 고급 분석을 활용한 홈페이지 구성 전략

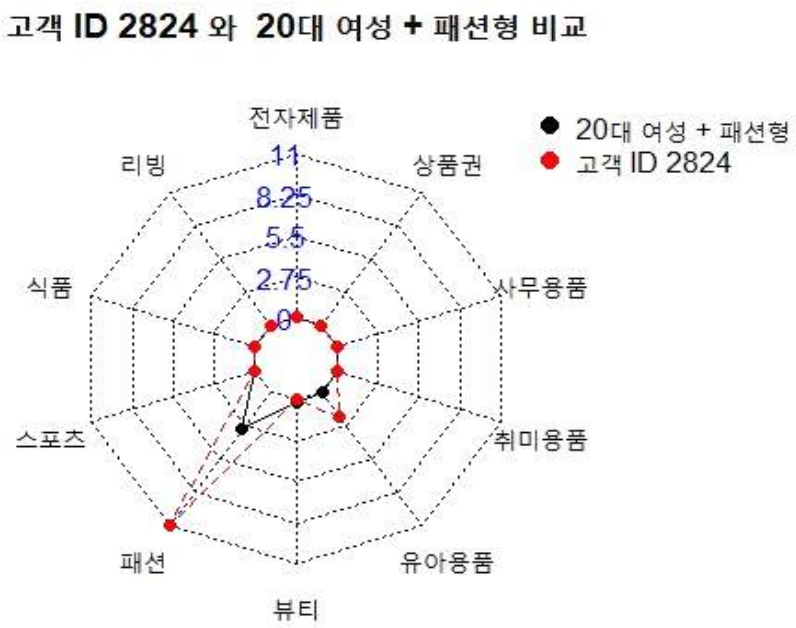
고객이 원하는 물품을 적재적소에 노출하는 것은 매우 중요하다. 앞서 구매 데이터와 행동 데이터의 추세가 일치하지 않는 문제점을 볼 수 있었다. 온라인 롯데몰에서 원하는 제품을 찾기 위해 많은 시간을 할애해야 하며, 홈페이지 구성상에 문제가 있는 것으로 파악된다. 따라서 본 분석팀은 구매 경로에 따라 개인화된 추천 상품 노출을 제안한다. 온라인 쇼핑몰은 오프라인과 다르게 개인에게 맞는 페이지를 구축할 수 있다는 장점이 있다. 이를 활용하여, 분산분석과 클러스터링을 통해 고객을 세분화하고, 연관 분석 및 협업필터링을 적용하여 고객별 선호도가 높은 상품을 추천함으로써 홈페이지 방문 고객의 관심이 실구매로 이어지도록 다음과 같이 유도한다.

(1) 메인페이지 - 아노바, 클러스터링

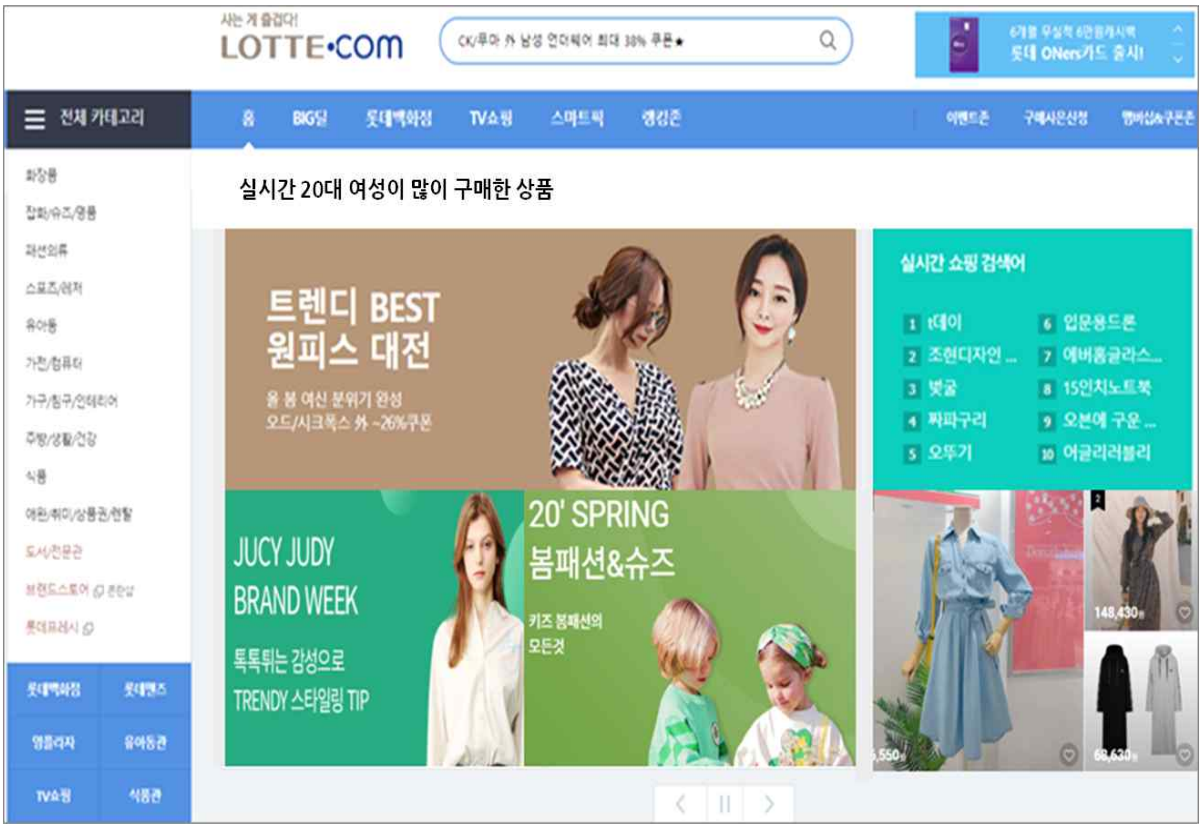
메인페이지는 고객이 사이트에 접속했을 때 처음 마주하는 공간으로, 이는 곧 사이트를 대표하는 페이지이다. 따라서 첫 번째로, 각 고객별 구매 패턴을 분석하여 선호 상품군 기반의 메인페이지를 구성하여 개인화된 서비스를 제공할 것이다. 구매 정보가 부족한 고객에겐 성별/연령별 분류를 통한 선호 상품군을 추천한다. 구매 정보가 충분한 고객은 앞서 분산분석/클러스터링으로 세분화된 고객 집단별 선호 상품군 위주로 노출시킨다.

두 번째로, 간소화된 상품 분류 목차와 실시간 쇼핑 검색어를 노출시킬 것이다. 롯데 온라인 쇼핑몰은 상품군의 대분류가 총 37개로, 필요 이상으로 많아 웹사이트 탐색이 다소 어렵다는 단점이 있다. 고객이 원하는 상품을 보다 신속하게 찾을 수 있도록 한 눈에 들어오는 분류 방식이 필요하다. 실시간 검색어는 포털 사이트에서 선거 기간 동안 그 이용이 제약될 정도로 파급 효과가 크기 때문에, 이를 활용하면 고객의 이목을 끌 수 있는 좋은 마케팅 전략이 될 수 있을 것이다.

세분화된 고객 집단으로 분류하고, 그 집단에 속하는 특정 고객 한 명을 예시로 들어 메인 페이지 구성을 살펴보면 다음과 같다. 고객 ID 2824는 세분화된 집단으로 분류한 결과, 20대 패션형 여성 집단에 속했다. 20대 패션형 여성 고객 집단과 고객 ID 2824를 레이더 차트로 상품 선호도를 비교해보았다. 2824번 고객은 집단에 비해 패션 구매 수가 현저하게 높았으며, 2순위로 유아용품의 구매수도 높았다. 따라서 2824번 고객의 구매 패턴을 분석한 결과를 기반으로 패션 카테고리와 유아용품 위주로 메인페이지를 구성하였다. 또한, 실시간 쇼핑 검색어와 2824번 고객이 속한 고객 집단이 많이 구입한 상품을 배치함으로써, 고객별 개인화된 페이지를 제공한다.



<그림 : 20대 패션형 여성 집단과 고객ID 2824 제품 선호도 비교>



<그림: 고객ID2824의 메인페이지 구성 예시 >

(2) 제품 상세 페이지 - 연관분석

대분류 카테고리별 소분류 연관 분석을 통해 어떤 상품들이 함께 팔리는지 경향성을 파악하고, 연관성이 높은 상품을 제품 상세 페이지에 같이 추천함으로써 구매율을 높인다. 총 10개의 대대분류별 소분류 항목의 연관분석을 수행하였으며, 연관규칙이 없는 대대분류 C9(사무용품)와 C10(상품권)은 분석하지 않았다. 연관분석의 지표로는 지지도(Support), 신뢰도(Confidence), 향상도(Lift)를 사용하였다.

- 지지도(Support) :

- 전체 거래에서 X, Y를 동시에 포함한 거래 비율
- P(X∩Y) = (X,Y를 동시에 포함하는 거래수) / (전체 거래수)

- 신뢰도 (Confidence) :

- X를 구입한 거래 중 Y를 같이 구입한 비율
- 신뢰도가 높을수록 유용한 규칙일 가능성이 높다.
- P(X∩ Y) / P(X) = (X,Y를 동시에 포함하는 거래수) / (품목 X를 포함하는 거래수)

- 향상도 (Lift) :

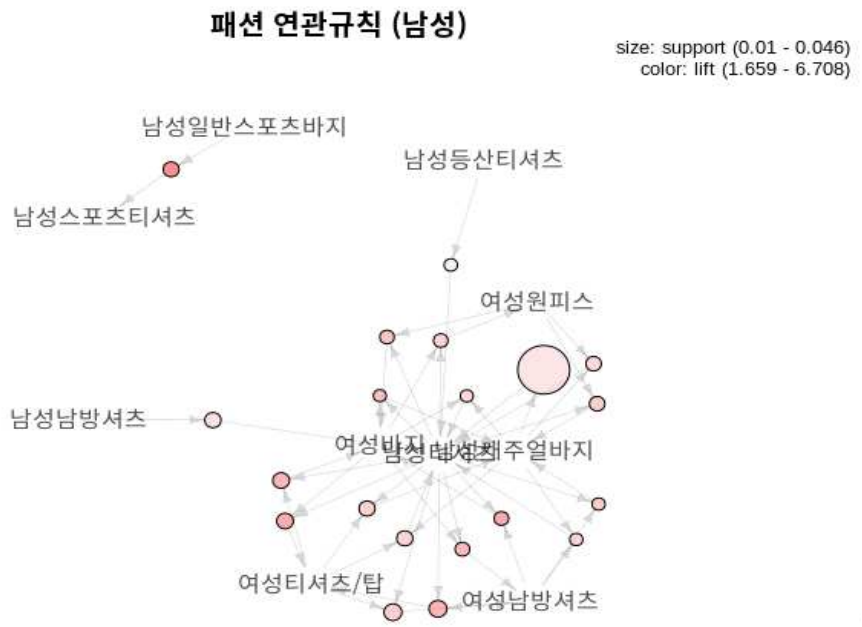
- X를 구매한 사람이 Y를 구매할 확률과 X의 구매와 상관없이 Y를 구매할 확률의 비율
- 연관규칙의 신뢰도/지지도
- Lift(향상도)가 1이면, 두 상품이 서로 독립적인 관계
- Lift가 > 1이면, 양의 상관관계
- Lift가 < 1면, 음의 상관관계

다음은 대분류 카테고리 중 연관규칙이 가장 많이 생성된 패션 소분류 항목의 연관분석 결과이다.

<패션 카테고리 연관분석>

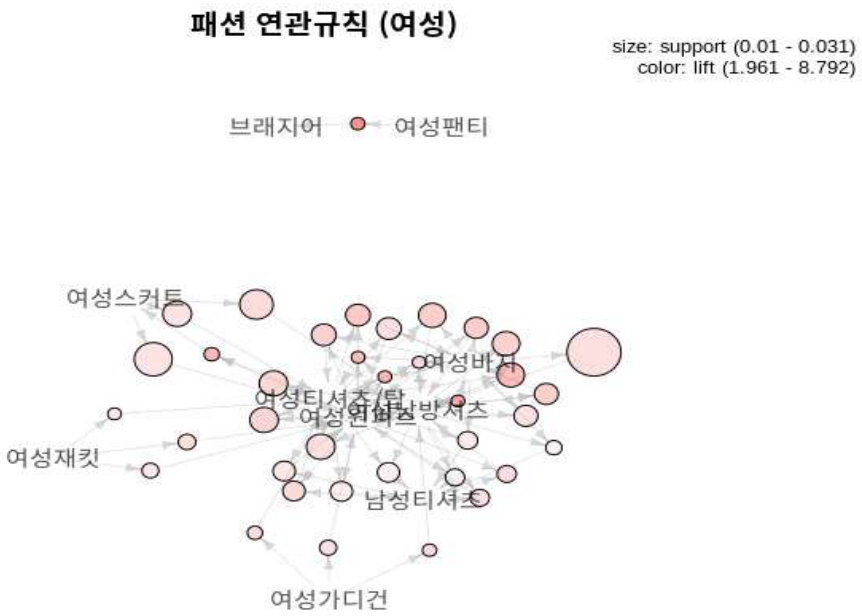
```
# 데이터 전처리
library(dplyr)
a <- train_order
a <- inner_join(a, df_product, by=c('PD_C', 'CN_our'))
a <- inner_join(a, df_master, by=c('CLAC_C', 'CN_our'))

df <- a %>%
    filter(CN_our == 'C5')

clac3 <- df[c('CLAC3_NM', 'CLNT_ID')]
```

```
# 소분류(CLAC_3) 품목 개수 확인
length(unique(clac3$CLAC3_NM))
# 고객 id별로 구매 항목 모으기
clac3_list <- split(clac3$CLAC3_NM, clac3$CLNT_ID)

# 중복 제거하고 transaction으로 변환
library(arules)
clac3_list <- sapply(clac3_list, unique)
clac3_list <- as(clac3_list, 'transactions')

# 연관규칙 생성 - 지지도 0.01 이상, 신뢰도 0.3이상
clac3_rule <- apriori(clac3_list, parameter = list(support = 0.01, confidence =
0.3, minlen = 2))

# 향상도 높은순으로 내림차순하기
inspect(sort(clac3_rule, by = 'lift'))

# 시각화를 위해 연관규칙 2그룹으로 분리하기
C5_1 <- clac3_rule[c(1:3,14,34:41,49,50,55,56,58,59,64,65), ]  # 남성의류
C5_2 <- clac3_rule[c(4:13,15,33,42:48,51:54,60:63,66:75), ]    # 여성의류

library(arulesViz)
plot(C5_1, method = "graph",
     control = list(type = 'items'),
     main = '패션 연관규칙 (남성)',
     vertex.label.cex = 0.7,
     edge.arrow.size = 0.3)
```



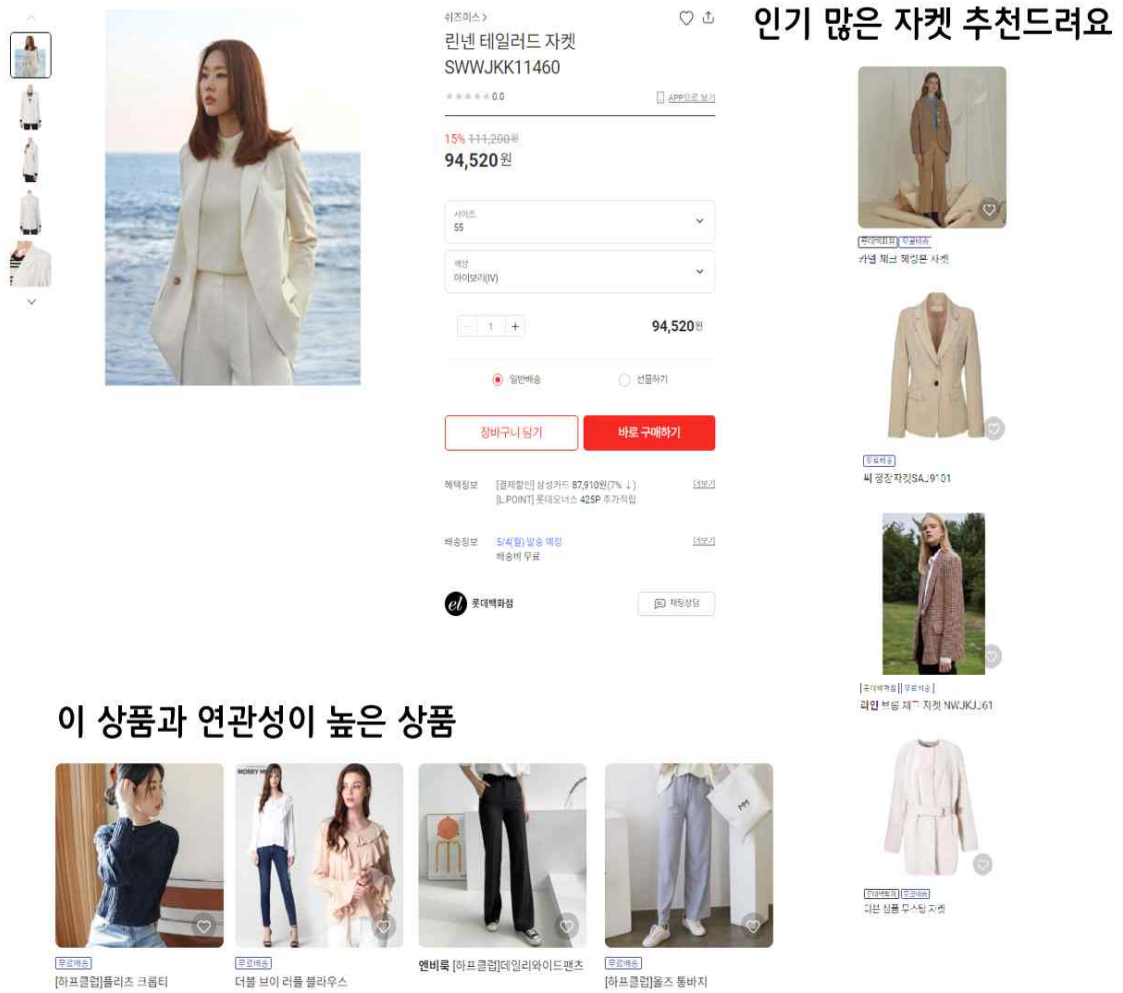
<그래프 : 패션 연관규칙(남성)>

```
plot(C5_2, method = "graph",
     control = list(type = 'items'),
     main = '패션 연관규칙 (여성)',
     vertex.label.cex = 0.7,
     edge.arrow.size = 0.3)
```



<그래프 : 패션 연관규칙(여성)>

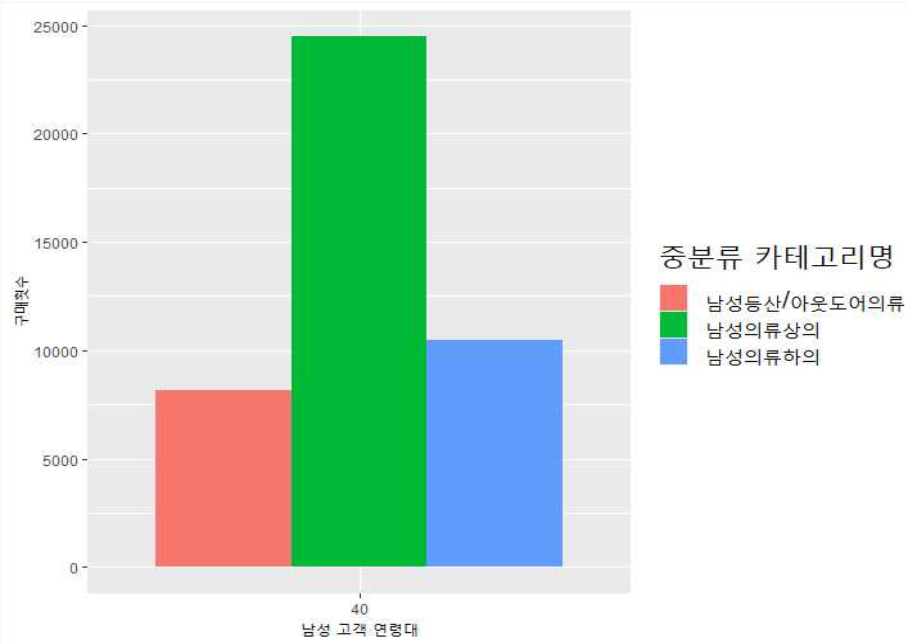
패션 카테고리 소분류 항목의 연관분석을 수행한 결과 첫 번째로, 겉옷을 살 때 티셔츠나 원피스 등 상의를 같이 사는 패턴을 보였다. 두 번째로, 상의와 하의를 동시에 사는 패턴을 보였고, 세 번째로 스포츠 의류를 산 고객들은 또 다른 스포츠 의류를 구매하는 패턴을 볼 수 있었다. 이를 활용하여 제품 상세 페이지에 선택한 제품과 연관성이 높은 상품을 배치하는 전략을 도출했다. 페이지 하단에는 함께 사면 좋은 상품으로 연관성이 높은 상품을 배치하고, 사이드 메뉴에는 선택한 제품과 동일한 소분류 항목 내에서 인기가 많은 상품을 배치하여 고객의 관심을 끌 것이다. 다음의 예시는 여성 자켓을 선택했을 시, 상세페이지 하단에 연관분석을 바탕으로 연관성이 높았던 하의와 상의를 추천하며, 사이드 메뉴에는 같은 성향을 가진 고객들이 많이 구매한 자켓을 추천하여 페이지를 구성하였다.



< 그림: 제품 상세페이지 구성 예시 >

(3) 장바구니 페이지 - 아이템기반 협업필터링

협업필터링(Collaborative Filtering) 혹은 추천시스템(Recommendation System)이란 사용자로부터 얻은 기호정보에 따라 관심사를 자동적으로 예측하는 방법이다. 해당 접근법은 사용자들의 과거 경향이 미래에도 유지된다는 전제를 가지고 있다. 본 분석에서 사용된 아이템기반협업필터링(IBCF, Item Based Collaborative Filtering)은 사용자들 대신에 평가된 아이템이 지표로 사용된다. IBCF는 대부분의 사람들이 과거에 좋아했던 상품과 비슷한 상품이면 좋아하는 경향이 있고, 반대로 선호하지 않았던 상품과 비슷한 상품이면 싫어하는 경향이 있다는 점을 기반으로 하고 있다. 이 필터링 방법은 고객이 선호도를 입력한 기존의 상품들과 예측하고자 하는 상품과의 유사도(Similarity)를 계산하여 고객의 선호도를 예측하는 방법이다. 즉, 예측하고자 하는 상품과 비슷한 상품들에 대하여 고객이 높은 평가를 하였다면 그 상품도 높게 평가를 할 것이라고 예측하고, 낮은 평가를 하였다면 그 상품도 낮게 평가할 것이라고 예측하는 것이다. 아이템 기반 협동적 필터링 방법은 상품들 간의 유사도를 계산하기 위하여 두 상품에 모두 선호도를 입력한 고객들의 선호도를 사용한다. 그러나 고객들 간의 유사도가 전혀 고려되지 않기 때문에 만약 특정 고객과 전혀 선호도가 비슷하지 않은 사용자들의 평가를 기반으로 한다면 상품들 간의 상관관계의 정확도가 떨어지고, 아울러 추천 시스템의 예측 능력과 추천 능력이 저하될 수 있다.



<그래프: 40대 남성 최다 판매 중분류 품목 TOP3>

본 분석팀은 IBCF를 통한 상품 추천을 구매 직전 페이지에 적용하여, 고객이 구매하고자 하는 상품과 유사도가 높은 중분류 내 인기 상품을 추천하는 방식을 제안하고 있다. 이를 통해, 우리는 고객으로 하여금 깜빡하고 구매하지 않은 상품 혹은 구매 예정 상품에 어울리는 상품을 추가구매 하도록 유도하는 것을 목표로 한다. 또한, IBCF는 고객의 구매 정보가 충분할 경우 보다 높은 예측 정확도를 보여주는데, 아래와 같이 40대 남성의 특징적인 구매 패턴을 보이지 않는 고객에게 보다

관심을 끌 수 있는 상품을 추천할 수 있다. 40대 남성의 최다 구매 중분류 카테고리는 남성 의류 상/하의 및 남성등산/아웃도어의류이다. 그러나, 918번 고객은 애견용품, 메이크업 및 여성 의류를 주로 구매한 것을 확인할 수 있다. IBCF에 따라, 918번 고객에는 나이/성별에 맞는 상품이 아닌 구매한 상품에 기반하여 건강보조식품, 유아의류하의 및 유아 위생용품이 추천되었다.

고객식별번호	성별	나이대	총구매액	중분류명	구매횟수
918	M	40	1757470	고양이용품	2
918	M	40	1757470	남성의류아우터	1
918	M	40	1757470	남성일반스포츠의류	1
918	M	40	1757470	메이크업	7
918	M	40	1757470	미용소품	2
918	M	40	1757470	선케어	1
918	M	40	1757470	성인침구	2
918	M	40	1757470	스킨케어	4
918	M	40	1757470	안경/선글라스	1
918	M	40	1757470	애견용품	21
918	M	40	1757470	여성가방	2
918	M	40	1757470	여성속옷	1
918	M	40	1757470	여성스포츠화	1
918	M	40	1757470	여성양말류	1
918	M	40	1757470	여성의류상의	8
918	M	40	1757470	여성의류전신	2
918	M	40	1757470	여성의류하의	3
918	M	40	1757470	여성일반스포츠의류	1
918	M	40	1757470	이미용가정	1
918	M	40	1757470	헤어케어	1

고객식별번호	성별	나이대	총구매액	추천상품1	추천상품2	추천상품3
918	**M**	**40**	**1757470**	**건강보조식품**	**유아의류하의**	**유아위생용품**

<표: 아이템기반협업필터링(IBCF) 추천 결과>

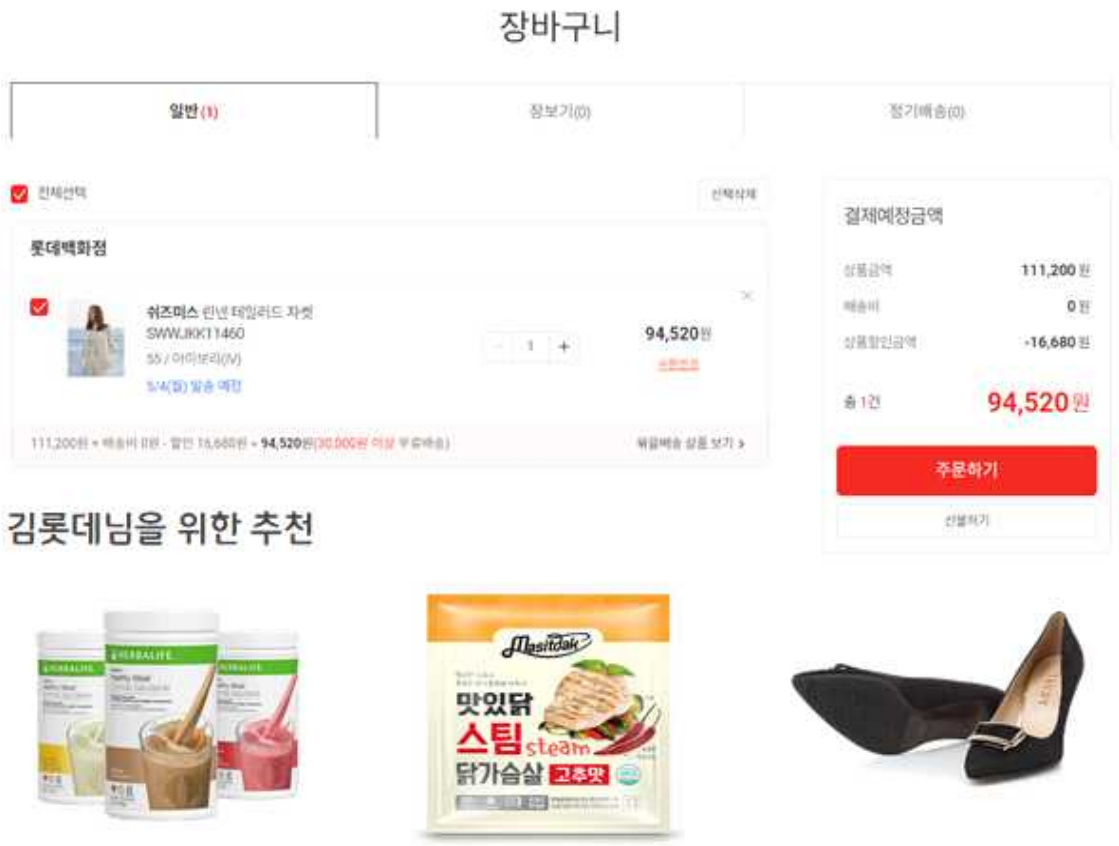
다음은 고객 ID 2824의 구매내역과 아이템기반협업필터링(IBCF)의 결과로 추천된 상품을 나타낸 것이다. 아이템기반협업필터링을 통해 다른 카테고리 내 함께 구매되는 제품 규칙을 발견하였으며, 분석결과 고객 ID 2824은 건강보조식품, 닭고기류, 여성화를 구매할 확률이 높았다. 따라서 장바구니 하단 페이지에 추천함으로써, 결제 직전 한 번 더 상품 구매를 유도한다.

고객식별번호	성별	나이대	중분류명
2824	F	20	여성의류상의
2824	F	20	남성의류상의
2824	F	20	유아의류상의
2824	F	20	여성가방
2824	F	20	유아동속옷

<그림 19 고객 ID 2824의 구매내역>

고객식별번호	성별	나이대	추천상품1	추천상품2	추천상품3
2824	F	20	건강보조식품	닭고기류	여성화

<그림 20 고객 ID 2824의 추천내역>



<그림: 고객 ID 2824의 장바구니 페이지 구성 예시>

3. 기타 마케팅 전략

(1) 선호도가 높은 브랜드와 업무 제휴 협약 (JBP)

앞서 보았듯이, 롯데닷컴은 쁘띠엘린과 고려은단 브랜드와 업무제휴 협약(JBP)을 체결했지만, 효과가 미미했다. 특히 고려은단의 경우, 소비자의 건강기능식품의 관심이 날로 증대하고 있음에도 불구하고 협약을 맺은 6월28일을 기점으로 7월 매출은 전달 대비 1,105,300원 하락하여 6개월 동안 최저치를 기록했다. 이에 따라, 마케팅 홍보가 제대로 전달되지 않았으며 고객의 충분한 관심을 끌지 못한 것으로 판단된다. 특히 업무 제휴 협약(JBP)의 경우 브랜드 파워가 고객에게 큰 영향을 미치기 때문에, 저조한 효과를 개선하기 위해 매출과 판매량이 높은 브랜드와의 협약 마케팅 전략을 제안한다. 매출과 판매량은 곧 고객의 브랜드 선호도를 나타내는 가장 중요한 지표이기 때문에 다음과 같이 6개월 동안의 브랜드 판매량과 매출을 분석했다.

<그래프: 최고 매출/최다 판매 브랜드 TOP 10>

나이키, 아이다스, 빈폴의 경우 최고 매출, 최다 판매 TOP 10에 동시에 속했으며, 특히 나이키의 경우 10~50대 남녀 연령층에서 모두 매출 1위를 기록했다. 이를 참고하여, 브랜드 선호도가 높은 브랜드와 업무제휴 협약을 맺어 다양한 프로모션을 제공한다면 구매율을 높일 수 있을 것이다. 온라인 롯데몰의 매출이 꾸준히 하락하는 본 시점에서 협약을 위한 브랜드 재고는 필수적이라 판단된다.

<그림: 최고 매출/최다 판매 브랜드 TOP 50>

(2) 이탈 고객을 위한 Remarketing

롯데 온라인 쇼핑몰은 '2018년 브랜드 고객충성도 조사'에서 온라인종합쇼핑몰 부문 1위에 선정됐다. 이를 유지하기 위해선 이탈률 분석을 통해 이탈 주기에 맞춰 푸시 마케팅을 하고, 고객 충성도를 유지하는 것이 매우 중요하다. 온라인 쇼핑몰 시장의 성장세가 지속될 것으로 전망됨에 따라, 해당 분야의 경쟁이 더욱 치열해질 것으로 판단된다. 치열한 온라인 시장 경쟁에서 우위를 점하기 위해서는 누적되는 고객의 데이터를 빠르게 분석하고 개인화 서비스를 빠르게 적용해야 할 것이다. 따라서, 본 분석팀은 카테고리의 간소화, 고객 니즈에 상응하는 브랜드와의 업무 제휴 협약 및 데이터에 기반한 홈페이지 개인화 서비스를 제안한다.

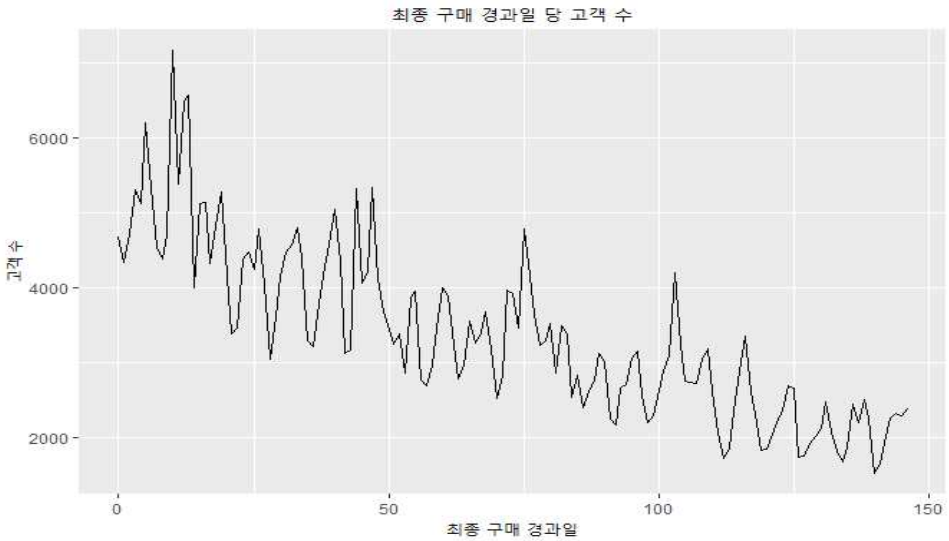
<Normal Client 이탈률 계산>

(1) 이탈률 계산을 위한 dataframe 구성

```
colnames(normal_client)
normal <- normal_client[c(1,29,30)]
# CLNT_ID, CLNT_CL, Cluster_num

session <- train_session
session$SESS_DT <- as.Date(session$SESS_DT, tz = '', tryFormats = c('%Y%m%d', '%Y-%m-%d'))


# normal client들의 session 정보만 따로 담기
normalsess <- inner_join(normal, session, by='CLNT_ID')


# 동일 일자에 생성한 세션 정보 제거
dfnor <- normalsess %>% distinct(CLNT_ID, SESS_DT) %>% arrange(CLNT_ID, SESS_DT)


# clnt_id 당 총 구매 일자 수 구하기
dfnor <- dfnor %>% group_by(CLNT_ID) %>% mutate(cnt = n())

# clnt_id와 sess_dt로 정렬하기
dfnor <-dfnor[order(dfnor$CLNT_ID, dfnor$SESS_DT),]


# SESS_DTlag 열에 앞 행에 있는 SESS_DT를 저장
# 앞 행에 있는 SESS_DT와의 간격을 계산
dfnor$SESS_DTlag<- shift(dfnor$SESS_DT, n=1, fill=NA, type = "lag")
dfnor$SESS_DTdiff<-dfnor$SESS_DT - dfnor$SESS_DTlag



dfnor$SESS_DTfinal<-
                    ifelse(dfnor$CLNT_ID == shift(dfnor$CLNT_ID, n=1, fill = NA, type = "lag"),
```

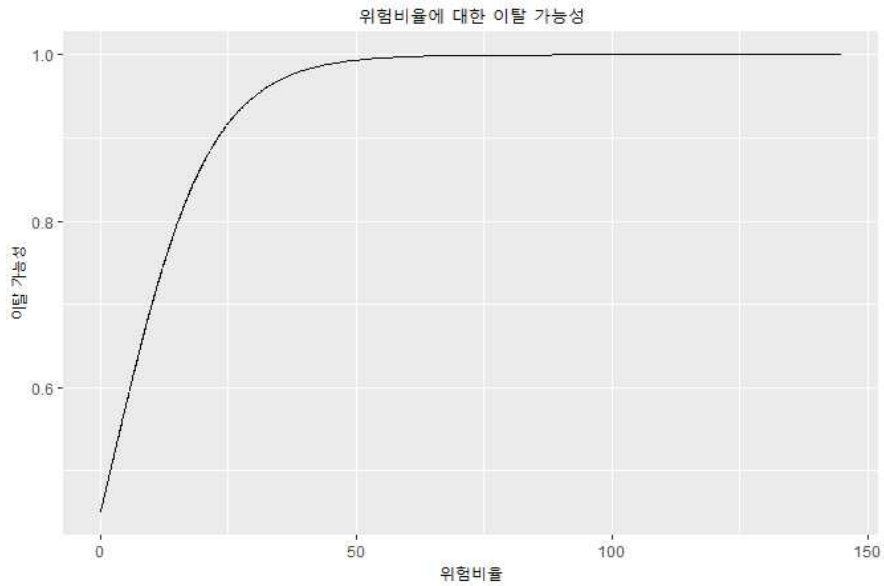
```
                # TRUE: 앞 행의 CLNT_ID와 일치하면(동일 CLNT_ID의 구매일자가 맞으면)
                dfnor$SESS_DT - shift(dfnor$SESS_DT, n=1, fill=NA, type = "lag"),   NA)
                # FALSE: clnt_id가 한 행밖에 없을 때

(2) 결측치 처리
b <-dfnor %>% filter(cnt==1)
b$SESS_DTfinal <- 147
b$mean_term <- 147
b <- b[c(1,3,7)]
# 구매 일자가 하루 뿐인 경우
# 즉, 평균 구매 주기를 구할 수 없는 경우
# 2018/04/01~2018/08/25 총 147일을 구매 주기로 넣음

(3) 평균 구매 주기 구하기
a <- dfnor %>% filter(cnt >= 2)
a <- na.omit(a)
# 구매 일자가 2일 이상
# 즉, 평균 구매 주기를 구할 수 있는 경우
finalnor_df<-a %>% group_by(CLNT_ID) %>% mutate(mean_term
=round(mean(SESS_DTfinal))) %>% select(CLNT_ID,cnt,mean_term) %>%
distinct(CLNT_ID,cnt,mean_term)
finalnor_df <- rbind(finalnor_df,b)
# a, b를 finalnor_df로 합쳐줌

(4) 최종 구매 경과일 구하기
# 마지막 구매 일자를 구함
dfnor<- dfnor %>% group_by(CLNT_ID) %>% mutate(last_day = max(SESS_DT))
lastnor <- dfnor %>% distinct(CLNT_ID, last_day)

# 마지막 구매 일자로 부터 2018년 8월 25일까지의 간격을 구함
finalnor_df <- inner_join(finalnor_df, lastnor, by='CLNT_ID')
finalnor_df$finalterm <- as.numeric(as.Date('2018-08-25') - finalnor_df$last_day)

# riskratio=finalterm/mean_term
finalnor_df$riskratio <- finalnor_df$finalterm/finalnor_df$mean_term

(5) 이탈 기준 정하기
aa <- finalnor_df %>% group_by(finalterm) %>% summarise(n=n())

# finalterm의 분포를 확인
ggplot(data=aa,aes(x=finalterm,y=n)) +
  geom_line() +
```

```
  ggtitle('최종 구매 경과일 당 고객 수') +
  xlab('최종 구매 경과일') +
  ylab('고객 수')+
  theme(plot.title = element_text(hjust=.5))
```



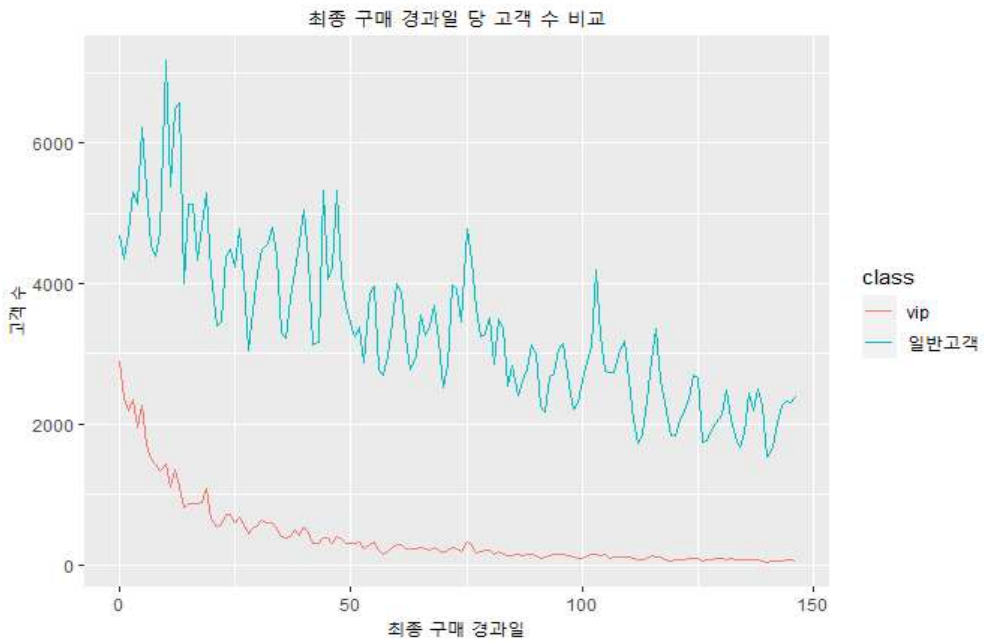
<그래프: 구매 경과일 별 고객 수>

```
finalnor_df$churn <- ifelse(finalnor_df$finalterm>=53, 1, 0)

# 기울기가 완만해지기 시작하는 53을 이탈 기준으로 정함
# 즉, 최종 구매 경과일이 53보다 크거나 같으면 이탈했다고 정의함

(6) Logistic Regression 수행
fitnor <- glm(churn ~ riskratio, data = finalnor_df, family = binomial())
summary(fitnor)

finalnor_df$y <-
exp(-0.2022393+0.1046793*finalnor_df$riskratio)/(1+exp(-0.2022393+0.1046793*finalnor
_df$riskratio))



ggplot(data=finalnor_df, aes(x=riskratio, y=y)) +
  geom_line() +
  ggtitle('위험비율에 대한 이탈 가능성') +
  xlab('위험비율') +
  ylab('이탈 가능성') +
  theme(plot.title = element_text(hjust=.5))
```



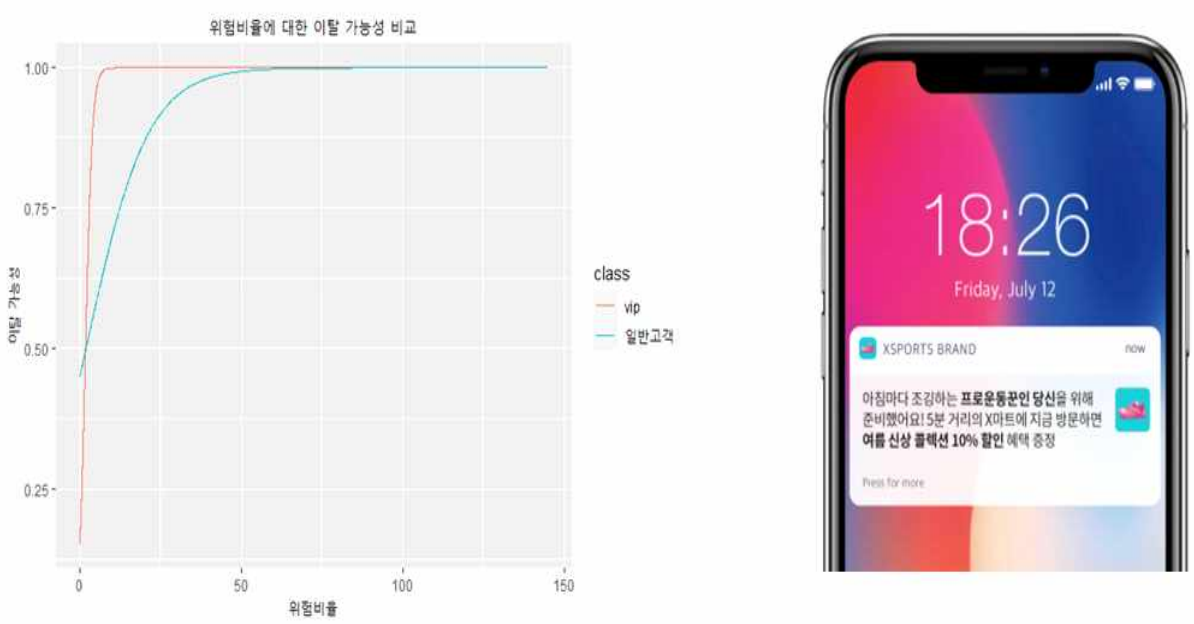
<그래프: 이탈 위험비율에 대한 이탈 가능성>

위와 같은 방법으로 VIP고객도 동일하게 이탈률 분석을 수행했는데, 이탈 기준 일수가 21일로 53일인 일반 고객보다 현저하게 적은 결과를 볼 수 있었다.



<그래프: 최종 구매 경과일 당 고객 수 비교>

이탈률 분석 결과, VIP 고객은 일반고객에 비해 이탈 기준 날짜가 짧기 때문에 약간만 오래 접속을 하지 않아도 이탈할 가능성이 높았다. 일반 고객의 경우 VIP 고객보다 평균 접속 주기보다 더 오랜 기간 접속하지 않더라도 재접속할 확률이 높았다. 이에 따라, 각 고객 이탈 주기에 맞춰 선호 상품 판매 정보를 e-mail, 쿠폰 발송 등을 통해 Push Marketing을 진행하는 전략을 도출했다.



<그림 50 이탈 고객을 위한 Remarketing>

VI. 결론 및 제언

지금까지 본 분석팀은 롯데 온라인 쇼핑몰 소비패턴을 분석해 개인화 서비스를 제공하는 다양한 전략을 세웠다. 본 분석의 배경에서도 보았듯이, 유통 기업의 1위를 달리던 롯데는 온라인 유통 업계의 후발 주자로 출발해 쿠팡, 마켓컬리 등 온라인 공룡 기업과의 경쟁에서 크게 두각을 드러내지 못하고 있다. 오히려 본 분석팀이 살펴본 바에 따르면, 롯데 온라인 쇼핑몰의 매출은 점점 떨어지고 있는 추세이며 시계열 예측을 통한 결과 역시 하강세였다. 이에 본 분석팀은 롯데가 온라인 유통에서도 경쟁력을 가질 수 있도록 디지털 마케팅 시장에서 필수 요소로 꼽히고 있는 개인화 서비스를 통한 상품 추천 전략을 내세웠다. 이는 롯데 쇼핑몰을 이용하는 고객들의 성향을 분석한 것을 기반으로, 홈페이지 이동 경로별로 고객에게 최적의 UI/UX를 제공하고 상품을 추천한다는 전략이다.

먼저, 로그인 전에는 기존의 메인 페이지로 구성하되, 각 시즌에 알맞은 상품군 위주로 트렌디한 느낌을 주면서 주목을 끈다. 로그인 후에는 개인화된 페이지를 보여주는데, 이 페이지는 두 가지 레이어로 구성된다. 고객의 구매 성향이 충분히 누적되지 않은 경우, 성별/연령별로 분류된 고객들이 선호하는 상품을 중심으로 추천한다. 고객이 롯데 쇼핑몰을 이용한 내역이 쌓이고, 클러스터링 번호가 매겨지는 것은 물론 해당 고객이 평균보다 더 많이 구매한 특정 상품군이 쌓이는 경우 해당 상품군을 중심으로 메인 페이지를 구성해 고객들의 이목을 끈다. 이와 더불어 간소화한 상품 목차, 실시간 쇼핑 검색어를 통해 고객들의 관심을 환기한다.

둘째로, 제품 상세 페이지에는 연관 분석을 통해 함께 구매하는 상품을 제시함으로써, '페어링' 형식으로 상품을 추천한다. 특히 속옷과 같은 경우, 브래지어를 사면 꼭 팬티를 구입하듯이 높은 상관성을 가진 상품은 각 제품 상세페이지에 눈에 띄게 표시해줌으로써 함께 구입하도록 유도한다.

셋째로, 장바구니로 페이지에는 선택한 제품과 다른 카테고리의 제품을 보여주면서 상품을 추천한다. 협업 필터링을 통해 구매 연관도가 높은 다른 카테고리 제품을 추천함으로써, 큰 인과관계가 없는 제품이라 쉽게 떠오르지 않던 제품이라 할지라도 자신에게 필요하거나 구매욕을 자극하는 상품을 보여주어 쇼핑 만족감을 극대화한다.

마지막으로, 본 분석팀이 구한 이탈률을 토대로 고객이 빠져나가는 주기를 계산한다. 고객의 이탈이 예상되기 전, 고객이 최근에 봤던 상품을 중심으로 이메일 또는 SMS를 통해 푸시 마케팅을 추진한다.

한계점 및 향후 연구 과제

분 분석팀은 '롯데 온라인 쇼핑몰 소비패턴 분석을 통한 개인화 서비스 제공' 프로젝트를 진행하면서, 몇 가지 한계점을 발견했다.

1. 데이터의 한계

먼저 본 데이터는 2018년 4월~9월까지의 데이터로, 꽤 시간이 지난 데이터이며, 기간도 6개월로 짧아 분석을 진행하기에 한계를 보였다. 시계열 분석의 경우 1~2년 정도의 데이터가 쌓여야 계절적 요인에 대한 분석 등이 가능하기 때문에, 6개월 데이터로 구매 패턴을 분석하고 미래를 예측하는 데에는 아쉬움이 있었다.

특히, 클러스터링이나 상품 추천 알고리즘 등의 머신러닝 결과는 새로운 데이터가 유입되어야 예측정확도가 높은 유의미한 결과를 낼 수 있는데, 6개월의 짧은 기간의 데이터로는 다소 협소한 분석을 진행할 수밖에 없었다.

2. 고객 세분화 한계

현재 아마존은 고객들을 0.1명 단위로 세분화해 상품을 추천하고 홈페이지를 구성하고 있다고 한다. 본 분석팀은 고객들을 30개의 집단으로 나누고, 각 개인의 선호 상품군을 분석해 개인화된 서비스를 제공하려는 전략을 세웠다. 본 분석팀이 아마존 같은 글로벌 기업처럼 세분화하는 것은 힘들겠지만, 상품군뿐만 아니라 다양한 파생 변수들을 추가하여 고객들의 특성을 파악했다면 더욱 정교하고 세밀한 상품 추천이 가능했을 것이다.

3. 알고리즘의 정교성

본 분석팀이 분석한 범위는 각 개인이 어떤 성향을 띄느냐에 따라 메인 페이지 구성 방향을 제안한 것이지, 구체적인 알고리즘을 제시한 것이 아니다. 물론 응용 프로그래머와 함께 협력해야 하는 부분이긴 하지만, 개인화된 페이지에 구성할 상품을 직접 추출하고 배치하는 부분까지 정교하게 진행했으면 하는 아쉬움이 남는다.

위의 세 가지 한계점은, 데이터의 보완, 프로젝트 범위나 기간의 연장 등을 통해 더욱 발전시킬 수 있을 것이다. 본 분석팀의 한계점을 향후 연구 과제로 삼아 온라인에서 뚜렷한 입지를 다질 수 있을 것이다.

특히, 롯데그룹은 '쿠팡'을 따라잡겠다는 목표로 롯데의 온라인 쇼핑몰을 통합하고 다양한 서비스를 통합한 '롯데온'을 이번주에 출시했다. 본 분석팀이 제안한 방향과 맞물리는 부분도 상당 부분 발견할 수 있다. 그러나 결과적으로는 반복되는 서버 오류와 단순한 UX/UI로 실망을 안겨주었다. 오프라인 유통에서 강세를 보이던 롯데그룹은 온라인 매출을 향상하기 위해서는 명예 회복에 힘을 써야 할 것으로 보인다.