

portfolio

김송미

INDEX

- ✓ Intro
- ✓ 쇼핑몰 이벤트 데이터 분석
- ✓ 콘텐츠 이용률 데이터 분석
- ✓ 메이플스토리M 직업별 데이터 분석
- ✓ Text Mining

1. Intro

김송미

생년월일 : 1993.09.06

2012.03 ~ 2013.02

한국IT직업전문학교 게임학부 게임프로그래밍학과

2013.03 ~ 2018.08

세종대학교 소프트웨어융합대학 컴퓨터공학과

세종대학교 소프트웨어융합대학 게임개발동아리 판도라큐브 정회원 및 임원(총무)

세종대학교 소프트웨어융합대학 컴퓨터그래픽스 모바일가상현실연구센터 소속 ITRC 정부과제 참여 연구원

2018.09 ~ 2019.05

(주)링크제니시스 차량 부품 SW 자동화 테스트 프로그램 개발 및 검증(C/C++, Python)

2019.05 ~ 2019.10

(주)레드모바일 모바일 MMORPG 게임 개발 '프로젝트 M' DBA 담당 업무(C/C++, C#, Python, MySQL, MSSQL, Redis)

2021.12 ~ 2022.10

(주)미디움 블록체인 애플리케이션 라이브 운영 데이터 분석 담당 업무(Python, MySQL, Google Data Studio)

2022.10 ~ 2024.10

(주)메가존 TAM2팀 데이터 분석 및 데이터 엔지니어링 대시보드 개발 담당 업무(Python, Tableau, PostgreSQL, MySQL)

2. 쇼핑몰 이벤트 데이터 분석

- 프로젝트명 : 쇼핑몰 이벤트 데이터 분석 (2022)
- 분석 기간 : 3일
- 분석 툴 : Jupyter Notebook
- 분석 언어 : Python, MySQL
- 분석 소개 : 2022년 3일간의 쇼핑몰 이벤트에 따른 매출 변화 분석
- 목적 : 이벤트 기간 내 참여 인원 비율과 해당 이벤트 품목에 따른 매출 확인
이벤트에 참여한 인원의 고액/비고액 유저 분류
- 프로젝트에서의 역할 : 데이터 분석

1. Intro

2. 쇼핑물
이벤트 분석

3. 콘텐츠
이용률 분석

4. 메이플스토리M
직업별 분석

5. Text Mining

6. ETC

```
[ ] import numpy as np
import pandas as pd
```

PyMySQL 설치

```
▶ pip install PyMySQL
```

```
↔ Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting PyMySQL
  Downloading PyMySQL-1.0.2-py3-none-any.whl (43 kB)
  ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 43.8/43.8 KB 1.5 MB/s eta 0:00:00
Installing collected packages: PyMySQL
Successfully installed PyMySQL-1.0.2
```

```
[ ] import pymysql
```

MySQL 계정 접속하기

```
[ ] #쇼핑물 mysql 계정 접속하기
db = pymysql.connect(
    host= ,
    port=3306,
    user= ,
    passwd= ,
    db= ,
    charset='utf8'
)
```

1. Intro

2. 쇼핑물
이벤트 분석3. 콘텐츠
이용률 분석4. 메이플스토리M
직업별 분석

5. Text Mining

6. ETC

```
[ ] cursor = db.cursor(pymysql.cursors.DictCursor)
```

CSV 파일 가져오기

```
[ ] shopping = pd.read_csv("/content/drive/MyDrive/shopping/order_list_20220727_0729.csv")
```

CSV 파일 보기

shopping



주문일시

구매자명

회원아이디

주문상품

구매자

휴대폰번호

주문수량

실결제금액(상품x수량)

0	2022-07-27 14:00:53	현	10	[이벤트] 코베아 2in1 모기 포충기 KECO9DT-16	010-2	1	184470
1	2022-07-27 14:01:08	김	10	[이벤트] 코베아 아웃도어 캠핑웨건 KECO9DT-08	010-6	1	227370
2	2022-07-27 14:01:23	박	10	[이벤트] 코베아 아웃도어 캠핑웨건 KECO9DT-08	010-4	1	227370
3	2022-07-27 14:01:23	현	10	[이벤트] 비비드블랑 썬패치	010-2	1	54000
4	2022-07-27 14:01:37	김	10	[이벤트] 비비드블랑 썬패치	010-9	3	162000
...
1134	2022-07-29 13:54:00	D.Zagarz		충북인삼농협 키즈튼튼홍삼롱스틱	010-7	3	135000
1135	2022-07-29 13:54:00	D.Zagarz		에버그린 데일리플랜 식물성 미세조류 오메가3	010-7	1	54000
1136	2022-07-29 13:54:45	김	10	[이벤트] 코베아 2in1 모기 포충기 KECO9DT-16	010-2	1	184470


```
[ ] #현재 사용 중인 컬럼 변경 코드
shopping.columns = ['order_date', 'order_name', 'order_id', 'order_goods', 'order_phone_no', 'order_many', 'order_discount_count']
```


```
[ ] #현재 가공된 쇼핑 데이터 보기
shopping
```

	order_date	order_name	order_id	order_goods	order_phone_no	order_many	order_discount_count
0	2022-07-27 14:00:53	현	10	[이벤트] 코베아 2in1 모기 포충기 KECO9DT-16	010-2	1	184470
1	2022-07-27 14:01:08	김	10	[이벤트] 코베아 아웃도어 캠핑웨건 KECO9DT-08	010-6	1	227370
2	2022-07-27 14:01:23	박	10	[이벤트] 코베아 아웃도어 캠핑웨건 KECO9DT-08	010-4	1	227370
3	2022-07-27 14:01:23	현	10	[이벤트] 비비드블랑 썬패치	010-2	1	54000
4	2022-07-27 14:01:37	김	10	[이벤트] 비비드블랑 썬패치	010-9	3	162000
...
1134	2022-07-29 13:54:00	D.Zagarzu		충북인삼농협 키즈튼튼홍삼롱스틱	010-7	3	135000
1135	2022-07-29 13:54:00	D.Zagarzu		에버그린 데일리플랜 식물성 미세조류 오메가3	010-7	1	54000
1136	2022-07-29 13:54:45	김	10	[이벤트] 코베아 2in1 모기 포충기 KECO9DT-16	010-2	1	184470
1137	2022-07-29 13:54:45	김	10	[이벤트] 비비드블랑 썬패치	010-2	1	54000
1138	2022-07-29 13:56:24	전	10	딱! 한번 염색삼푸(200ml x 2)+탈모케어삼푸(500ml)+비닐장갑 12장	010-8	1	66000

- 컬럼 이름이 한국어인 경우 데이터가 깨지는 경우가 발생하여 오류 사전 방지 목적으로 컬럼 이름 변경 작업

```
[ ] #아이디 기준으로 매출 그룹화
shopping_by_user2 = shopping.groupby('order_id').agg(sum).reset_index()[['order_id', 'order_discount_count']]
```

```
[ ] shopping_by_user2
```



	order_id	order_discount_count
0		444400
1	10	55000
2	10	54000
3	10	50000
4	10	125820
...		...
675	10	231300
676	10	50000
677	58	384000
678	64	130000
679	80	82500

- order_id 기준으로 매출 그룹화 작업

```
[ ] shopping['new_Date'] = pd.to_datetime(shopping['order_date'])
```

```
[ ] shopping
```

	order_date	order_name	order_id	order_goods	order_phone_no	order_many	order_discount_count	new_Date
0	2022-07-27 14:00:53	현	10	[이벤트] 코베아 2in1 모기 포충기 KECO9DT-16	010-2	1	184470	2022-07-27 14:00:53
1	2022-07-27 14:01:08	김	10	[이벤트] 코베아 아웃도어 캠핑웨건 KECO9DT-08	010-6	1	227370	2022-07-27 14:01:08
2	2022-07-27 14:01:23	박	10	[이벤트] 코베아 아웃도어 캠핑웨건 KECO9DT-08	010-4	1	227370	2022-07-27 14:01:23
3	2022-07-27 14:01:23	현	10	[이벤트] 비비드블랑 썬패치	010-2	1	54000	2022-07-27 14:01:23
4	2022-07-27 14:01:37	김	10	[이벤트] 비비드블랑 썬패치	010-9	3	162000	2022-07-27 14:01:37
...
1134	2022-07-29 13:54:00	D.Zagarzu	10	충북인삼농협 키즈튼튼홍삼롱스틱	010-7	3	135000	2022-07-29 13:54:00
1135	2022-07-29 13:54:00	D.Zagarzu	10	에버그린 데일리플랜 식물성 미세조류 오메가3	010-7	1	54000	2022-07-29 13:54:00
1136	2022-07-29 13:54:45	김	10	[이벤트] 코베아 2in1 모기 포충기 KECO9DT-16	010-2	1	184470	2022-07-29 13:54:45
1137	2022-07-29 13:54:45	김	10	[이벤트] 비비드블랑 썬패치	010-2	1	54000	2022-07-29 13:54:45
1138	2022-07-29 13:56:24	전	10	딱! 한번 염색샴푸(200ml x 2)+탈모케어샴푸(500ml)+비닐장갑 12장	010-8	1	66000	2022-07-29 13:56:24

- order_date를 복사하여 new_date 칼럼 생성
목적 : datetime형의 칼럼을 통한 날짜별 매출 그룹화

```
[ ] #new_Date는 order_date에서 년월일만 추출(일단위 매출 보기 위한 그룹화)
shopping['new_Date'] = pd.to_datetime(shopping['order_date']).dt.date
```

```
[ ] shopping
```



	order_date	order_name	order_id	order_goods	order_phone_no	order_many	order_discount_count	new_Date
0	2022-07-27 14:00:53	현	10	[이벤트] 코베아 2in1 모기 포충기 KECO9DT-16	010-2	1	184470	2022-07-27
1	2022-07-27 14:01:08	김	10	[이벤트] 코베아 아웃도어 캠핑웨건 KECO9DT-08	010-6	1	227370	2022-07-27
2	2022-07-27 14:01:23	박	10	[이벤트] 코베아 아웃도어 캠핑웨건 KECO9DT-08	010-4	1	227370	2022-07-27
3	2022-07-27 14:01:23	현	10	[이벤트] 비비드블랑 썬패치	010-2	1	54000	2022-07-27
4	2022-07-27 14:01:37	김	10	[이벤트] 비비드블랑 썬패치	010-9	3	162000	2022-07-27
...
1134	2022-07-29 13:54:00	D.Zagarzu		충북인삼농협 키즈튼튼홍삼롱스틱	010-7	3	135000	2022-07-29
1135	2022-07-29 13:54:00	D.Zagarzu		에버그린 데일리플랜 식물성 미세조류 오메가3	010-7	1	54000	2022-07-29
1136	2022-07-29 13:54:45	김	10	[이벤트] 코베아 2in1 모기 포충기 KECO9DT-16	010-2	1	184470	2022-07-29
1137	2022-07-29 13:54:45	김	10	[이벤트] 비비드블랑 썬패치	010-2	1	54000	2022-07-29
1138	2022-07-29 13:56:24	전	10	딱! 한번 염색샴푸(200ml x 2)+탈모케어샴푸(500ml)+비닐장갑 12장	010-8	1	66000	2022-07-29

- new_date에서 yyyy-mm-dd형식으로 데이터 정리

[이벤트] 포함하는 상품명 목록

```
[ ] shopping_event_goods = shopping[shopping['order_goods'].str.contains('이벤트')] |
```

```
[ ] shopping_event_goods
```

	order_date	order_name	order_id	order_goods	order_phone_no	order_many	order_discount_count	new_Date
0	2022-07-27 14:00:53	현	0102	[이벤트] 코베아 2in1 모기 포충기 KECO9DT-16	010-2	1	184470	2022-07-27
1	2022-07-27 14:01:08	김	0106	[이벤트] 코베아 아웃도어 캠핑웨건 KECO9DT-08	010-6	1	227370	2022-07-27
2	2022-07-27 14:01:23	박	0104	[이벤트] 코베아 아웃도어 캠핑웨건 KECO9DT-08	010-4	1	227370	2022-07-27
3	2022-07-27 14:01:23	현	0102	[이벤트] 비비드블랑 썬패치	010-2	1	54000	2022-07-27
4	2022-07-27 14:01:37	김	0109	[이벤트] 비비드블랑 썬패치	010-9	3	162000	2022-07-27
...
1120	2022-07-29 13:14:34	김	0104	[이벤트] 코베아 2in1 모기 포충기 KECO9DT-16	010-4	1	184470	2022-07-29
1122	2022-07-29 13:23:10	이	0106	[이벤트] 비비드블랑 썬스틱	010-6	1	50000	2022-07-29
1125	2022-07-29 13:34:27	김	0105	[이벤트] 코베아 2in1 모기 포충기 KECO9DT-16	010-5	1	184470	2022-07-29
1136	2022-07-29 13:54:45	김	0102	[이벤트] 코베아 2in1 모기 포충기 KECO9DT-16	010-2	1	184470	2022-07-29
1137	2022-07-29 13:54:45	김	0102	[이벤트] 비비드블랑 썬패치	010-2	1	54000	2022-07-29


- 이벤트 대상 상품 데이터 추출
- order_goods 기준으로 데이터를 추출할 경우 중복된 이벤트 품목도 중복 제거 없이 출력됨
- 이벤트 품목별로 합산할 경우 중복 제거된 이벤트 대상 상품 데이터 확인 가능

이벤트 품목별로 합산

```
[ ] shopping_event_goods.groupby('order_goods').agg('sum')
```


	order_many	order_discount_count
order_goods		
[이벤트] 비비드블랑 썬스틱	379	18950000
[이벤트] 비비드블랑 썬패치	141	7570796
[이벤트] 코베아 2in1 모기 포충기 KECO9DT-16	139	25567542
[이벤트] 코베아 아웃도어 캠핑웨건 KECO9DT-08	98	22100364

```
[ ] #이벤트 기간 내 일별 매출 계산  
shopping.groupby('new_Date').agg('sum')
```



	order_many	order_discount_count
new_Date		
2022-07-27	637	64018822
2022-07-28	685	71763840
2022-07-29	280	30001970

```
[ ] #쇼핑몰 이벤트 품목만 일별 매출 계산  
shopping_event_goods.groupby('new_Date').agg('sum')
```



	order_many	order_discount_count
new_Date		
2022-07-27	432	43107482
2022-07-28	247	22490980
2022-07-29	78	8590240

- 총 품목/이벤트 품목 일별 매출 비교

2백만원이 기준, 2백만원 이상 구매면 고액 유저, 고액 유저 조회

```
[ ] shopping2_by_user[shopping2_by_user['order_discount_count']>=2000000]
```

	order_id	order_discount_count
127	0103	2174000
244	0104	11618080
313	0105	2492000
462	0107	2474250
524	0108	2224300
539	0108	2190500
551	0108	2554300

- 200만원을 기준으로 고액 유저/비고액 유저의 매출 데이터 추출

2백만원이 기준, 2백만원 미만 구매면 비고액 유저, 비고액 유저 조회

```
[ ] shopping2_by_user[shopping2_by_user['order_discount_count']<2000000]
```

	order_id	order_discount_count
0	0102	55000
1	0102	54000
2	0102	50000
3	0102	125820
4	0102	59000
...
675	0109	50000
676	587	384000
677	647	130000
678	806	82500
679	9	444400

```
sql = '''
SELECT o.member_id AS 'order_id',
       DATE(LEFT(o.ono,8)) AS 'order_date',
       m.name AS 'order_name',
       o.title AS 'order_goods',
       o.member_id AS 'order_phone_no',
       FLOOR((CAST(REPLACE(CURRENT_DATE(),'-','') AS UNSIGNED) - CASE WHEN LEFT(m.birth, 2) < 21
THEN CONCAT('20', CAST(m.birth AS CHAR))
ELSE CONCAT('19', CAST(m.birth AS CHAR)) END) / 10000) AS 'order_age',
       m.Sex AS 'order_sex',
       m.StrCC AS 'order_new_country_code',
       m.NumCC AS 'order_country_code',
       m.Level AS 'order_level'
FROM ecommerce.wm_order o
JOIN ecommerce.wm_member m
ON o.member_id = m.member_id;'''

[ ] cursor.execute(sql)

↔ 30540
```

- 데이터 비교 목적으로 작성한 MySQL에 존재하는 이벤트 참여자 정보 추출 쿼리문

인사이트

- 특정 고액 유저들에게 매출에 의존하지 않은 매출 성과 확인
- 이벤트 품목이 4종류로 품목 수와 단가가 낮은 만큼 목표 금액 또한 7000만원으로 비교적 낮게 잡아 총 74,188,702원으로 목표 매출 달성
- 이벤트 품목 뿐만 아니라 비이벤트 품목의 매출도 일정함에 따라 적립금 누적 목적이 아닌 실사용 목적의 구매율 또한 증가

3. 콘텐츠 이용률 데이터 분석

- 프로젝트명 : 콘텐츠 이용률 데이터 분석 (2022)
- 분석 기간 : 14일
- 분석 툴 : Jupyter Notebook, Google Workspace, Tableau
- 분석 언어 : Python, MySQL
- 분석 소개 : 2022년 콘텐츠 이용률 및 이용권 구매 유저 데이터 분석
- 목적 : 웹툰, 비디오, 영화, 구독권 콘텐츠 품목에 따른 매출 확인
 각각의 콘텐츠에 대한 유저 연령대, 콘텐츠 장르별 순위 분석
- 프로젝트에서의 역할 : 데이터 분석

- 전체 기간 및 특정 월에만
이용권 구매한 유저 데이터
추출

```
#전체 기간 이용권 구매 유저 수 구하기
sql_1 = '''SELECT  u.user_id AS '유저 아이디',
    u.name AS '성함',
    u.phone_no AS '아이디(폰번호)',
    v.product_id AS '구매한 이용권 종류',
    v.reg_date AS '구매일자',
    FLOOR((CAST(REPLACE(CURRENT_DATE(), '-', '')) AS UNSIGNED) - CASE WHEN LEFT(u.birthday, 2) < 21 THEN CONCAT('20', CAST(u.birthday AS CHAR))
    ELSE CONCAT('19', CAST(u.birthday AS CHAR)) END) / 10000) AS '연령대',
    u.sex AS '성별',
    u.new_country_code AS '국가',
    u.contry_code AS '국가코드',
    u.level AS '등급',
    COUNT(v.user_id) AS '구매횟수'
FROM platform_db.users u
JOIN platform_db.voucher v
ON u.user_id = v.user_id
WHERE DATE(v.reg_date) >= STR_TO_DATE('2021-12-01', '%Y-%m-%d')
    AND DATE(v.reg_date) <= STR_TO_DATE('2022-05-31', '%Y-%m-%d')
GROUP BY v.user_id
ORDER BY v.reg_date;'''
```

```
[ ] #1월 이용권 구매 유저 수 구하기
sql_3 = '''SELECT  u.name AS '성함',
    u.phone_no AS '아이디(폰번호)',
    v.product_id AS '구매한 이용권 종류',
    FLOOR((CAST(REPLACE(CURRENT_DATE(), '-', '')) AS UNSIGNED) - CASE WHEN LEFT(u.birthday, 2) < 21 THEN CONCAT('20', CAST(u.birthday AS CHAR))
    ELSE CONCAT('19', CAST(u.birthday AS CHAR)) END) / 10000) AS '연령대',
    u.sex AS '성별',
    u.new_country_code AS '국가',
    u.level AS '등급',
    COUNT(v.user_id) AS '구매횟수', v.reg_date AS '구매일자'
FROM platform_db.users u
JOIN platform_db.voucher v
ON u.user_id = v.user_id
WHERE DATE(v.reg_date) >= STR_TO_DATE('2022-01-01', '%Y-%m-%d')
    AND DATE(v.reg_date) <= STR_TO_DATE('2022-01-31', '%Y-%m-%d')
GROUP BY v.user_id
ORDER BY v.reg_date;'''
```

```
[ ] df_INNER_JOIN_1 = pd.merge(user_1, user_2, how='right',
                                on=['user_id', 'name', 'phone_no'])
```

```
[ ] df_INNER_JOIN_1
```



	user_id	name	phone_no	product_id_x	count_x	reg_date_x	product_id_y	count_y	reg_date_y
0	311c543a4ac125292bfe5b177d46ffca4c2bdec5		0107	3	1	2022-01-01 00:34:56	3	1	2022-01-01 00:34:56
1	c504b87bed1f9991f1ccddf90051acd2681baede		0102	3	2	2022-01-01 00:40:48	3	2	2022-01-01 00:40:48
2	580792e7d2abb355eb9c815e7185d54cb4acfcc6		0105	3	1	2022-01-01 00:48:55	3	1	2022-01-01 00:48:55
3	85ef58622df93a020edd0496bdb6eae98985c7d0		0108	3	1	2022-01-01 00:52:56	3	1	2022-01-01 00:52:56
4	629fd3f0291d88c091cce5f9930fb1cccd79ed1f	kyung	813	3	1	2022-01-01 00:53:51	3	1	2022-01-01 00:53:51
...
15364	9e33172fc7e197269ce1511f8c883a55f032a19b		0108	3	2	2021-12-31 18:51:39	3	1	2022-01-31 23:50:46
15365	e88b4bb5839006217fb5af059818736ecdb25462		0104	3	2	2021-12-02 15:59:02	3	1	2022-01-31 23:51:47
15366	db9d9c618b3fb14a7be3355a2746abe3bdb3723f		0109	3	1	2022-01-31 23:52:28	3	1	2022-01-31 23:52:28
15367	28aa10d8a1e82d70783ae8e4e3c4e8667073a212		0104	3	1	2022-01-31 23:53:10	3	1	2022-01-31 23:53:10
15368	c1ed722c83699abdfec2d82476e5c2f6085ecc4c		0109	3	1	2022-01-31 23:54:20	3	1	2022-01-31 23:54:20

- user_1에는 sql_1 쿼리문, user_2에는 sql_3 쿼리문을 저장하여 python에서 **Right Join** 작업
사유 : 전체 기간에 이용권 구매 유저 데이터에는 해당 유저가 반드시 존재해야 함
전체 데이터 존재 여부 확인

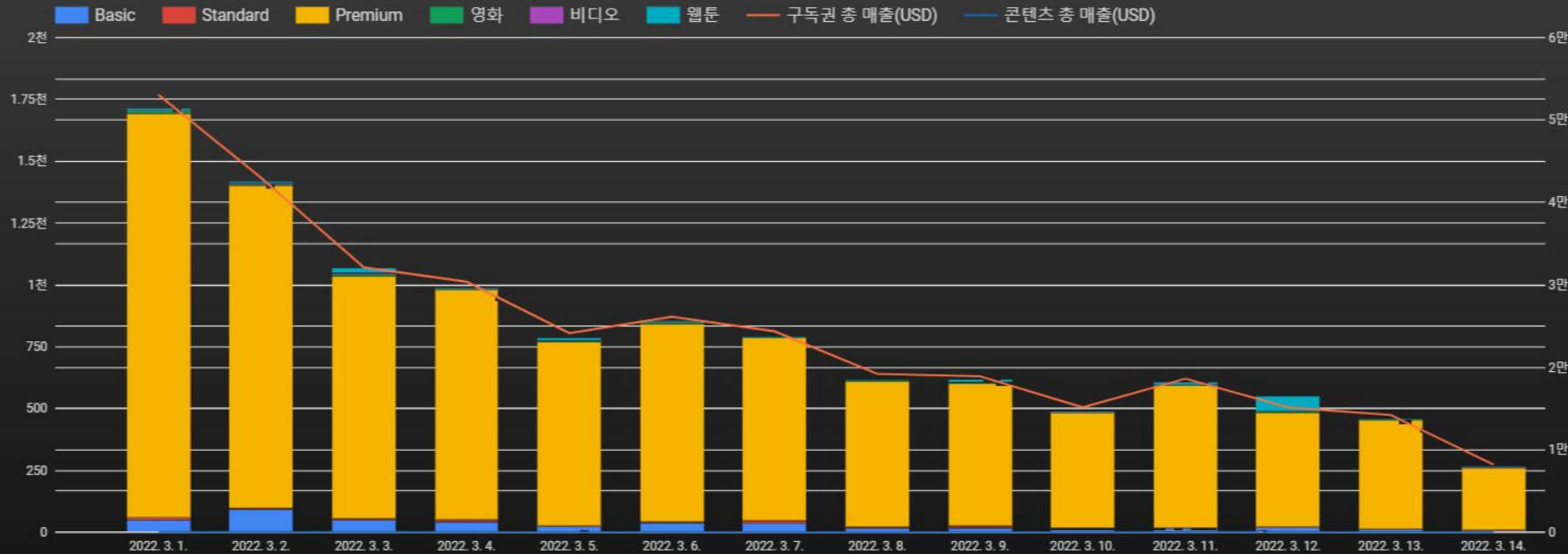
1. Intro

2. 쇼핑물
이벤트 분석3. 콘텐츠
이용률 분석4. 메이플스토리M
직업별 분석

5. Text Mining

6. ETC

구독권 및 콘텐츠 일별 매출



카테고리	판매 수량(개)	총매출(USD)	총매출
베이직	449	5,388	767.79
스탠다드	50	1,200	171.5
프리미엄	10,505	336,160	48,007.85

카테고리 ▲	판매 수량(개)	총매출(USD)	총매출
MV	57	143.64	20.52
TV	15	9.31	1.33
WT	151	45.01	6.43

- Google Workspace에 Python에서 입력했던 쿼리문으로 대시보드 제작
- 구독권의 종류와 콘텐츠 기준으로 판매 수량, 매출 표시

1. Intro

2. 쇼핑몰
이벤트 분석

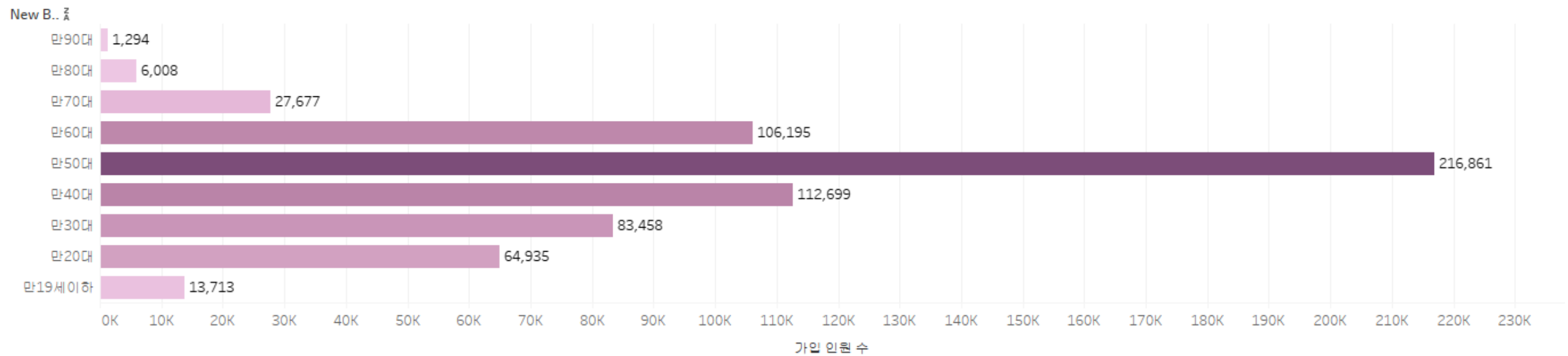
3. 콘텐츠
이용률 분석

4. 메이플스토리M
직업별 분석

5. Text Mining

6. ETC

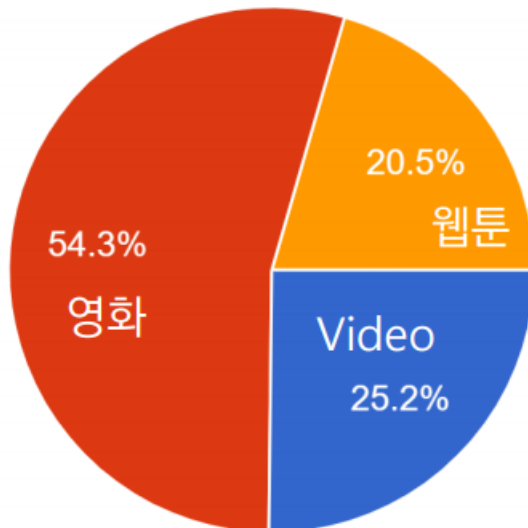
연령대별 가입 인원 수



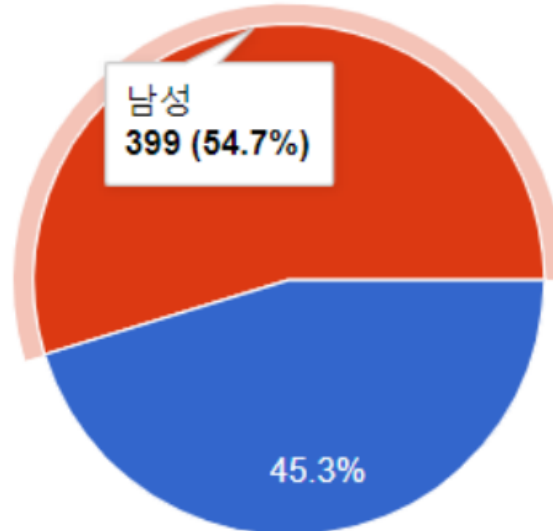
콘텐츠 이벤트 – 선호도 조사 결과

- 내용 : 기간 내 이벤트 대상 웹툰 및 영화를 1개 이상 감상하고 선호도 조사 참여
- 기간 : 14일
- 목표 : 유저의 피드백 확인 및 서비스 개선
- 결과 : 총 참여자 729명 (종료 후에도 18명 추가 참여)

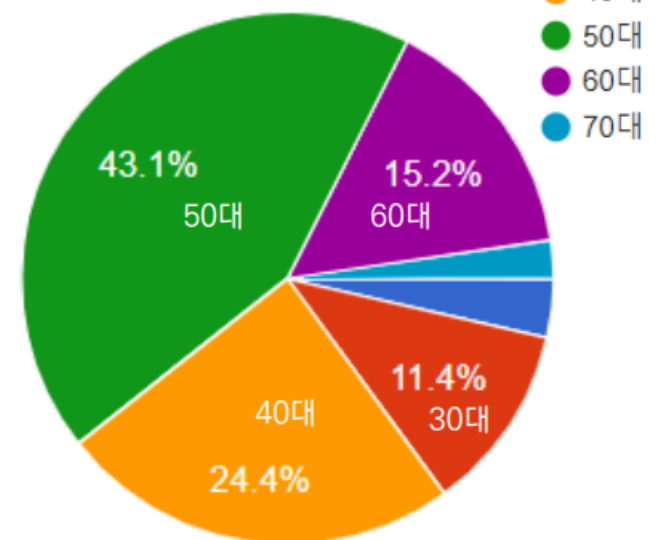
선호하는 콘텐츠 종류



응답자 성별



응답자 연령대



1. Intro

2. 쇼핑몰
이벤트 분석

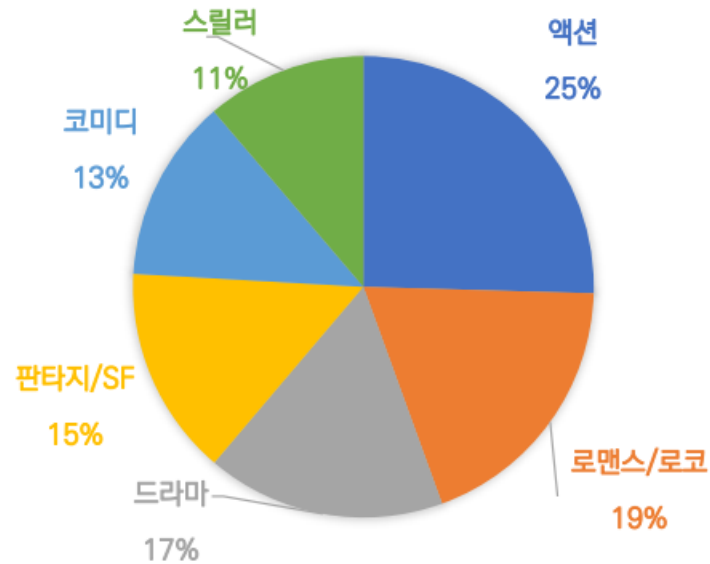
3. 콘텐츠
이용률 분석

4. 메이플스토리M
직업별 분석

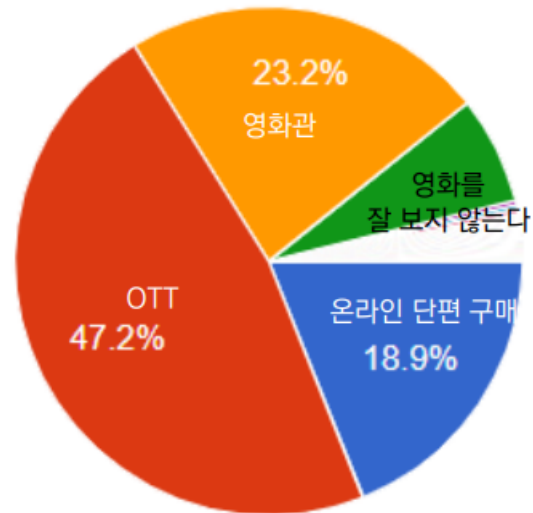
5. Text Mining

6. ETC

선호하는 영화 장르



주로 영화를 감상하는 루트



- 온라인 구매(단편 재생권 구매)
- OTT 어플 구독 (넷플릭스, 왓챠, 티빙 등)
- 영화관
- 영화를 잘 보지 않는다
- Tv
- TV
- 유튜브
- TV영화

1. Intro

2. 쇼핑물
이벤트 분석

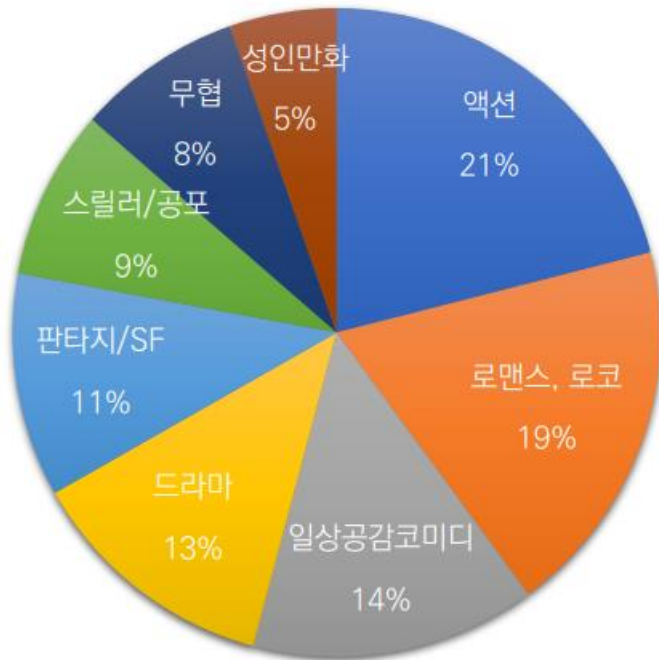
3. 콘텐츠
이용률 분석

4. 메이플스토리M
직업별 분석

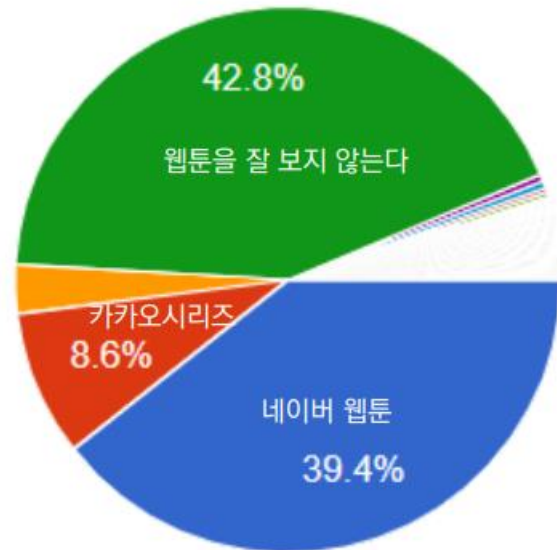
5. Text Mining

6. ETC

선호하는 웹툰 장르



주로 웹툰을 감상하는 루트



- 네이버웹툰
- 카카오시리즈
- 레진코믹스
- 웹툰을 잘 보지 않는다
- 유튜브
- 유튜브
- 만화가게

인사이트

- 유저 연령대 절반 이상이 50대에 분포되어 있음
- 젊은 세대 신규 유입을 위한 웹소설, 오디오, 창작 등 신규 콘텐츠 런칭 필요
- 유저 유치용 무료 이용 콘텐츠 증가, 할인 이벤트 필요
- 유저 연령대를 고려하여 콘텐츠 이용 방법에 대한 안내, 조회 수, 인기도 등 콘텐츠를 세부적으로 파악할 수 있는 요소 필요

4. 메이플스토리M 직업별 데이터 분석

- 프로젝트명 : 메이플스토리M 직업별 데이터 분석 (2024)
- 분석 기간 : 6개월
- 분석 툴 : Tableau
- 분석 언어 : SQL
- 분석 소개 : 메이플스토리M 직업별 레벨 능력치 분석
- 목적 : 직업별로 레벨 50까지 달성했을 경우의 능력치 변화도 확인
어떤 직업이 능력치가 가장 좋고 육성도 시각화
레벨 50 이후 직업별 능력치 예측
- 프로젝트에서의 역할 : 데이터 분석

- 가설 : 직업별로 레벨이 높아질수록 전체적인 공격력이 비례하며 증가할 것이다.
마법사의 경우만 마법 공격력이 높아지고 그 이외의 직업은 물리 공격력과 마법 공격력이 비례한다.

직업	세부 직업
모험가	전사(히어로, 팔라딘, 다크나이트), 마법사(아크메이지(불, 독), 아크메이지(썬, 콜), 비숍), 궁수(보우마스터, 신궁, 패스파인더), 도적(나이트로드, 새도어, 듀얼블레이드), 해적(바이퍼, 캡틴, 캐논슈터)
레지스탕스	배틀메이지, 와일드헌터, 메카닉, 제논
영웅	은월, 팬텀, 루미너스, 에반, 메르세데스, 아란
시그너스	소울마스터, 플레임위자드, 윈드브레이커, 나이트워커, 스트라이커
데몬	데몬슬레이어, 데몬어벤저
노바 종족	카이저, 엔젤릭버스터

1. Intro

2. 쇼핑몰
이벤트 분석3. 콘텐츠
이용률 분석4. 메이플스토리M
직업별 분석

5. Text Mining

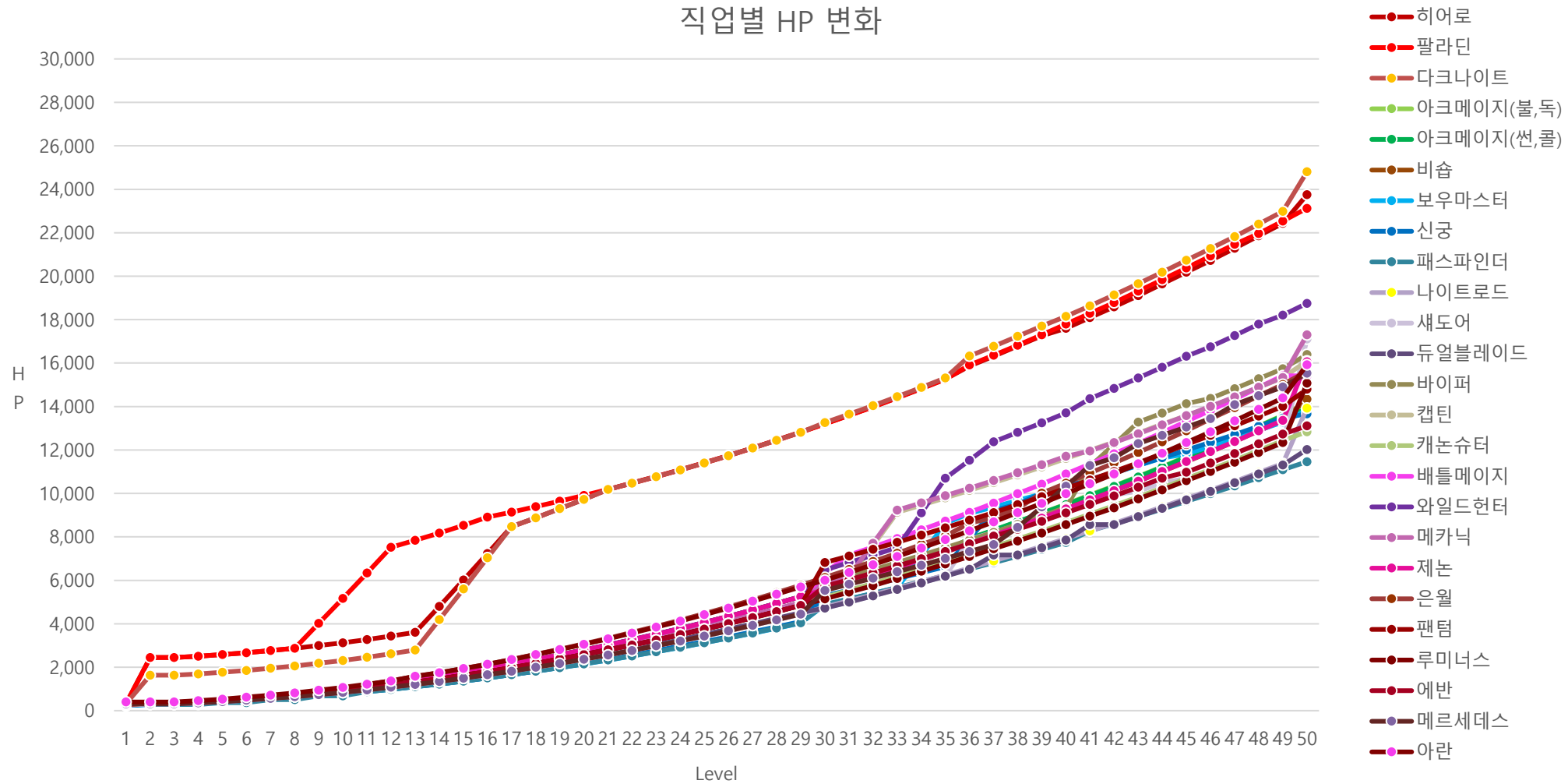
6. ETC

• 직업별 캐릭터 능력치 작성한 데이터 시트 정리

직업	세부직업	서버	닉네임	데이터
모험가 - 전사	히어로	제니스	송이494	히어로 데이터!A1
	팔라딘	제니스	송이464	팔라딘 데이터!A1
	다크나이트	제니스	송이414	다크나이트 데이터!A1
모험가 - 마법사	아크메이지(불,독)	제니스	송이424	아크메이지(불,독) 데이터!A1
	아크메이지(썬,콜)	제니스	송이484	아크메이지(썬,콜) 데이터!A1
	비숍	유니온	송이808	비숍 데이터!A1
모험가 - 궁수	보우마스터	유니온	송이818	보우마스터 데이터!A1
	신궁	제니스	송이434	신궁 데이터!A1
	패스파인더	유니온	송이838	패스파인더 데이터!A1
모험가 - 도적	나이트로드	유니온	송이848	나이트로드 데이터!A1
	새도어	유니온	송이858	새도어 데이터!A1
	듀얼블레이드	아케인	송이878	듀얼블레이드 데이터!A1
모험가 - 해적	바이퍼	제니스	송이454	바이퍼 데이터!A1
	캡틴	유니온	송이828	캡틴 데이터!A1
	캐논슈터	유니온	송이868	캐논슈터 데이터!A1
레지스탕스	배틀메이지	아케인	송이898	배틀메이지 데이터!A1
	와일드헌터	아케인	송이343	와일드헌터 데이터!A1
	메카닉	스카니아	송이353	메카닉 데이터!A1
	제논	스카니아	송이363	제논 데이터!A1
	블래스터	크로아	송이737	블래스터 데이터!A1
영웅	은월	아케인	송이373	은월 데이터!A1
	팬텀	아케인	송이383	팬텀 데이터!A1
	루미너스	아케인	송이393	루미너스 데이터!A1
	에반	아케인	송이303	에반 데이터!A1
	메르세데스	스카니아	송이313	메르세데스 데이터!A1
	아란	스카니아	송이323	아란 데이터!A1
	소울마스터	스카니아	송이202	소울마스터 데이터!A1
시그너스	미하일	크로아	송이242	미하일 데이터!A1
	플레임위자드	스카니아	송이212	플레임위자드 데이터!A1
	윈드브레이커	스카니아	송이232	윈드브레이커 데이터!A1
	나이트워커	크로아	송이252	나이트워커 데이터!A1
	스트라이커	크로아	송이262	스트라이커 데이터!A1
데몬	데몬슬레이어	크로아	송이717	데몬슬레이어 데이터!A1
	데몬어벤저	크로아	송이727	데몬어벤저 데이터!A1

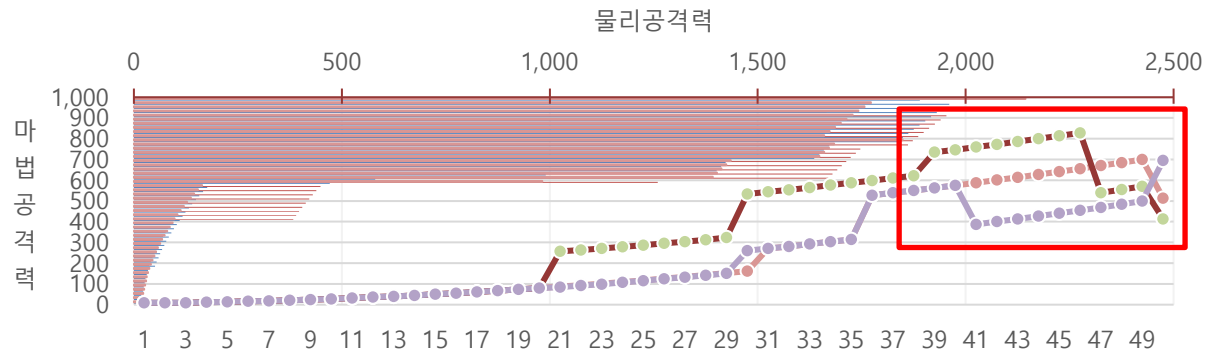
직업	세부직업	레벨	HP	MP	인기도	스타포스 개수	물리공격력	물리공격력 증가 확률	물리대미지 증가 확률	마법공격력	마법공격력 증가 확률	마법대미지 증가 확률	보스공격력 증가 확률	치명타확률	치명타피해확률	최대대미지	최종 대미지 증가 확률	방어를 무시 확률	물리방
모험가-전사	히어로	22	10,478	1,112	0	0	117	10	0	94	0	0	0	0	0	29,999,999	0	0	0
모험가-전사	히어로	23	10,778	1,198	0	0	124	10	0	101	0	0	0	0	0	29,999,999	0	0	0
모험가-전사	히어로	24	11,090	1,287	0	0	132	10	0	109	0	0	0	0	0	29,999,999	0	0	0
모험가-전사	히어로	25	11,413	1,379	0	0	140	10	0	117	0	0	0	0	0	29,999,999	0	0	0
모험가-전사	히어로	26	11,749	1,475	0	0	149	10	0	126	0	0	0	0	0	29,999,999	0	0	0
모험가-전사	히어로	27	12,095	1,574	0	0	157	10	0	134	0	0	0	0	0	29,999,999	0	0	0
모험가-전사	히어로	28	12,453	1,676	0	0	166	10	0	143	0	0	0	0	0	29,999,999	0	0	0
모험가-전사	히어로	29	12,822	1,782	0	0	176	10	0	153	0	0	0	0	0	29,999,999	0	0	0
모험가-전사	히어로	30	13,204	1,891	0	0	471	16	2	162	0	0	0	0	0	29,999,999	0	0	0
모험가-전사	히어로	31	13,592	2,000	0	0	580	20	2	271	0	0	0	0	0	29,999,999	0	0	0
모험가-전사	히어로	32	13,985	2,112	0	0	990	20	6	281	0	0	0	0	0	29,999,999	0	0	0
모험가-전사	히어로	33	14,401	2,231	0	0	1,401	20	10	292	0	0	0	0	0	29,999,999	0	0	0
모험가-전사	히어로	34	14,828	2,353	0	0	1,412	20	10	303	0	0	0	0	0	29,999,999	0	0	0
모험가-전사	히어로	35	15,255	2,475	0	0	1,423	20	10	314	0	0	0	0	0	29,999,999	0	0	0
모험가-전사	히어로	36	15,868	2,646	0	0	1,636	20	10	527	0	1.5	0	12.1	0	29,999,999	0	0	0
모험가-전사	히어로	37	16,319	2,775	0	0	1,648	20	10	539	0	1.5	0	12.1	0	29,999,999	0	0	0
모험가-전사	히어로	38	16,781	2,907	0	0	1,659	20	10	550	0	1.5	0	12.1	0	29,999,999	0	0	0
모험가-전사	히어로	39	17,254	3,042	0	0	1,672	20	10	563	0	1.5	0	12.1	0	29,999,999	0	0	0
모험가-전사	히어로	40	17,597	3,184	0	0	1,836	20	10	575	0	0	0	12.1	0	29,999,999	0	0	0
모험가-전사	히어로	41	18,093	3,326	0	0	1,849	20	10	588	0	0	0	12.1	0	29,999,999	0	0	0
모험가-전사	히어로	42	18,591	3,468	0	0	1,862	20	10	601	0	0	0	12.1	0	29,999,999	0	0	0
모험가-전사	히어로	43	19,110	3,616	0	0	1,875	20	10	614	0	0	0	12.1	0	29,999,999	0	0	0
모험가-전사	히어로	44	19,641	3,768	0	0	1,889	20	10	628	0	0	0	12.1	0	29,999,999	0	0	0
모험가-전사	히어로	45	20,185	3,923	0	0	1,903	20	10	642	0	0	0	12.1	0	29,999,999	0	0	0
모험가-전사	히어로	46	20,727	4,078	0	0	1,917	20	10	656	0	0	0	12.1	0	29,999,999	0	0	0
모험가-전사	히어로	47	21,281	4,236	0	0	1,931	20	10	670	0	0	0	12.1	0	29,999,999	0	0	0
모험가-전사	히어로	48	21,859	4,401	0	0	1,946	20	10	685	0	0	0	12.1	0	29,999,999	0	0	0
모험가-전사	히어로	49	22,436	4,566	0	0	1,961	20	10	700	0	0	0	12.1	0	29,999,999	0	0	0
모험가-전사	히어로	50	23,749	4,817	0	5	1,890	20	10	514	0	3.7	0	10	2.8	29,999,999	0	0	0

직업별 HP 변화

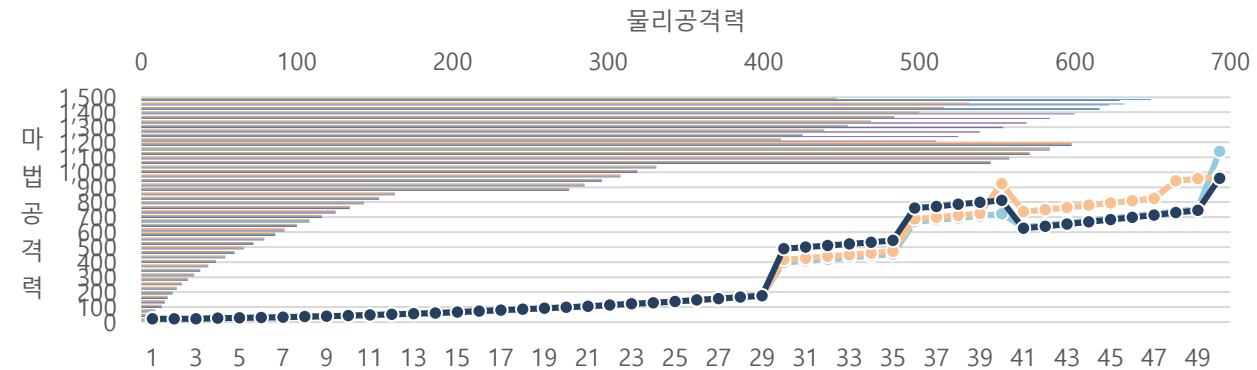


- 레벨이 높아질수록 전체적인 공격력이 비례하며 증가하는 모습

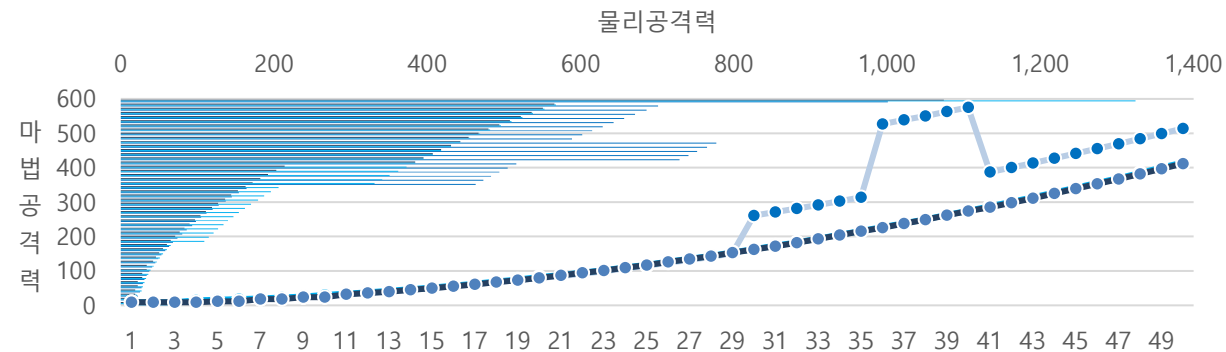
모험가-전사 물리/마법 공격력 변화



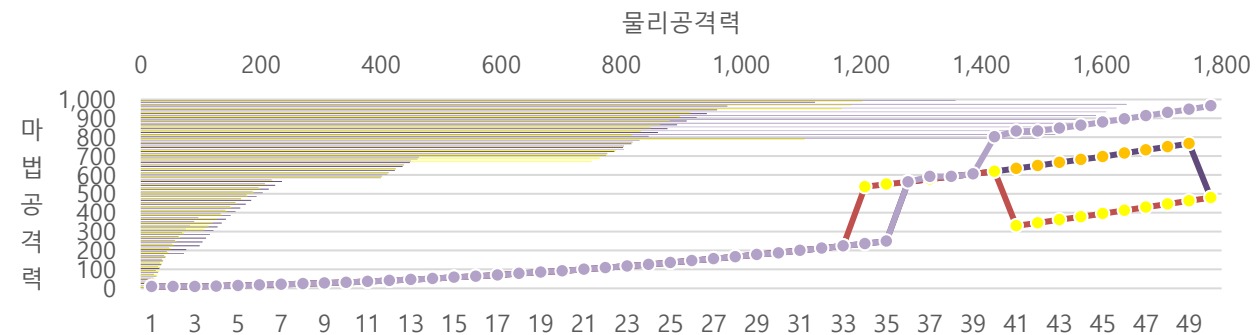
모험가-마법사 물리/마법 공격력 변화



모험가-궁수 물리/마법 공격력 변화



모험가-도적 물리/마법 공격력 변화



- 물리공격력과 마법공격력은 레벨에 비례하여 증가 (물리공격력 : 막대, 마법공격력 : 꺾은선)
- 그러나 100% 증가만 하지 않는다. 중간에 감소되었다가 다시 증가하는 형태를 보인다.

인사이트

- 전사 직업군 물리공격력 최대값 2144, 마법사 직업군 마법공격력 최대값 1138으로 물리공격력 1위는 전사, 마법공격력 1위는 마법사
- HP나 단순 공격력 뿐만 아니라 증가 확률, 대미지, 저항 확률 등 능력치의 다양한 기준으로 다른 시점에서 분석 필요
- 레벨이 높아짐에 따라 능력치가 반드시 정비례하여 증가하지 않음
- 능력치의 감소 및 급증 원인은 아이템 장착 여부, 버프 아이템 사용 여부, 퀘스트 진행 여부 파악 필요
- 환경적 요인을 최소화 한 이후에 100레벨의 예상 능력치 파악 가능

5. Text Mining

- 프로젝트명 : Text Mining (2024)
- 분석 기간 : 3일
- 분석 툴 : Jupyter Notebook
- 분석 언어 : Python
- 분석 소개 : 삼성닷컴(<https://www.samsung.com/>) 제품 리뷰에 대한 긍정/부정적인 단어 분류 및 분포도 시각화
- 목적 : 삼성 제품에 대한 구매 유저들의 리뷰 분석
- 프로젝트에서의 역할 : 데이터 분석

```
[ ] # 패키지 로드
import seaborn as sns # 그래프 패키지
import matplotlib.pyplot as plt # as 뒤에는 라이브러리 별명
import matplotlib.font_manager as fm
from PIL import Image
```

```
import pandas as pd
import numpy as np
import attr
import babel
import time
import glob
```

```
[ ] # sklearn 패키지의 metrics 모듈 로드하기
import sklearn.metrics
```

```
[ ] from wordcloud import WordCloud
```

```
▶ # 텍스트 파일 읽기
its = open('/content/samsungcomment.txt', 'r', encoding='utf-8')
its = its.read()
```

```
[ ] # 문자열 치환(엔터 제거)
result = its.replace('\n', ' ')
print(result)
```

☞ 옷감이 상하지 않고 관리가 가능해서 추천합니다 이전작보다 착용감이 너무 좋아짐 오븐없이 이걸로 다 할 수 있어서 너무 좋아요 깔끔한 디자인에 성능까지 좋아요~

- 삼성 가전제품 상품평 문서 읽어오는 과정

```
# 폰트 위치 확인
# 반복문(범위가 있는 반복문)을 통해 확인
for f in fm.fontManager.ttflist :
    if 'Go' in f.name : # 글꼴명 중에 고딕이 들어가면 출력
        print(f.name)

# 칼라맵 가져오기
palettes = ['spring', 'summer', 'seismic', 'PuBu']

# 한글 처리
mask = np.array(Image.open('pngwing.com (1).png')) # 마스크 이미지 인식
#font_path = 'C:\Windows\Fonts\HMKMMAG.TTF' # 글꼴 경로 지정
font_path = '/content/hy궁서-yoond1004.ttf' # 글꼴 경로 지정
wc = WordCloud(font_path = font_path, background_color = 'white', width=1800, height=1400, mask=mask, colormap = 'seismic')
wc.generate(result)

figure = plt.figure(figsize=(20, 10))
plt.axis('off') # 축 표시 안 함
plt.imshow(wc)
figure.savefig
```

1. Intro

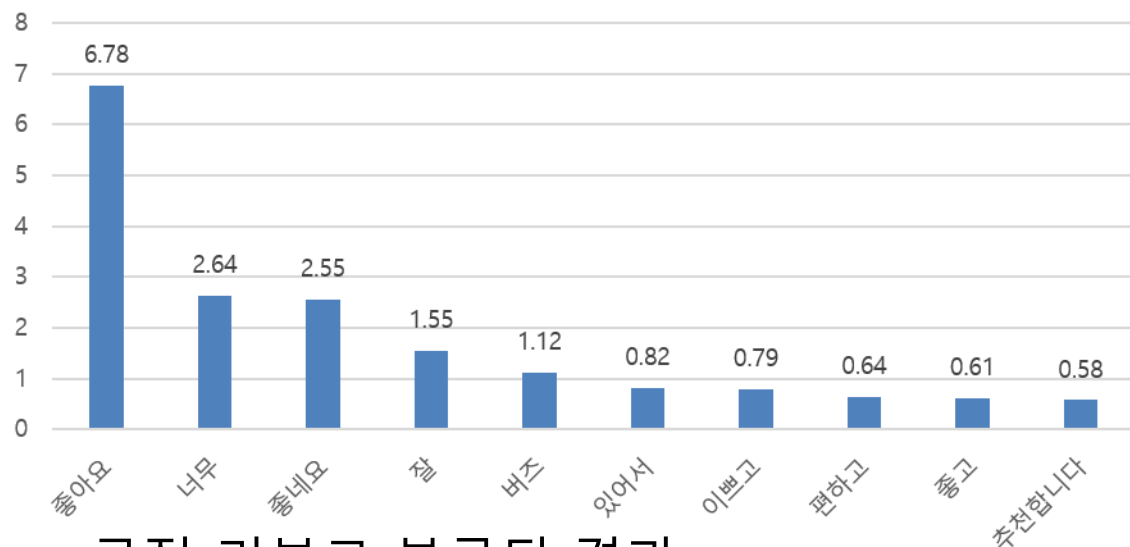
2. 쇼핑몰
이벤트 분석3. 콘텐츠
이용률 분석4. 메이플스토리M
직업별 분석

5. Text Mining

6. ETC



TOP10 긍정 리뷰 단어 비율 (%)



• 긍정 리뷰로 분류된 결과



TOP10 부정 리뷰 단어 비율(%)



• 부정 리뷰로 분류된 결과

인사이트

- 가전제품에 대한 리뷰에서 긍정적인 단어는 가장 많이 분포되어 있는 단어는 '좋아요'로 제품의 사용성, 디자인 모든 측면에 대한 단어로 평가
- 부정 리뷰로 분류된 단어로는 '너무', '좀' 등의 부사가 많은 비중을 차지
- 긍정 리뷰의 경우 신제품 출시에 맞춰 리뷰 이벤트를 진행함에 따라 하나의 리뷰 내용마다 '좋아요'의 단어가 연속으로 반복되는 현상 발생
- 리뷰를 통해 어떤 가전 제품에서 문제가 있다면 주로 충전이 되지 않는 문제, 소음 문제가 주요 원인임을 확인

End Of Document

김송미 / ksm9051@gmail.com

blog.naver.com/setakim96