NS홈쇼핑

[데이터 분석 분야 - 챔피언스리그]

NS SHOP+ 판매실적 예측을 통한 방송 편성 최적화 방안(모형) 도출

[Team KUAI]

고려대학교 인공지능학과 성민석

고려대학교 인공지능학과 류회성

고려대학교 인공지능학과 조대현

고려대학교 인공지능학과 조희승

2020 빅콘테스트
2020 BIG CONTEST

NS홈쇼핑

팀원 소개

팀장 성민석
고려대학교 인공지능학과 석박통합과정
minsuksung@korea.ac.kr

팀원 류회성
고려대학교 인공지능학과 박사과정
hoesungryu@korea.ac.kr

팀원 조대현
고려대학교 인공지능학과 석박통합과정
1phantasmas@korea.ac.kr

팀원 조희승
고려대학교 인공지능학과 석박통합과정
hscho9384@korea.ac.kr

서론

이투데이

[이슈&인물] 홈쇼핑 [redacted] "채널번호보다 '커머스AI'가 ...

이곳은 현재 이미지 분석, 자연어 처리, 데이터 마이닝, 기계 학습, AI 백엔드 시스템 개발 분야의 전문가 10여 명이 운영 중이다. 원본보기. △'홈쇼핑 ...

2020. 4. 9.

도시경제신문

AI 기업 [redacted], [redacted] 합병 통해 APAC 시장 정조준

[도시경제] AI 패션 스타일링 서비스 기업 [redacted]이 타겟팅 ... [redacted]는 응용 머신러닝(Applied Machine Learning) 기반의 컨텍스츄얼 ... AD는 앞서 TV홈쇼핑업체의 인터넷 쇼핑몰과 오픈마켓 업체 1곳을 통

2020. 2. 6.

매일경제

[redacted], 인공지능(AI) 기반 '스마트 AI 편성 시스템' 첫 도입 - 매일경제

머신 러닝'(빅데이터를 분석하고 가공해서 새로운 정보를

MK

Byline Network

심재석 이유지 남혜현 이종철 박리세윤 엄지용 홍하나 Contributor

대학생에게 홈쇼핑 데이터를 주면 매출을 예측할 수 있을까? SAS 분석 챔피언십 ② 연세대 팀

이종철 | 2019년 11월 8일

대학생에게 특정 홈쇼핑의 데이터를 주면 그다음 해 매출을 예측할 수 있을까? SAS코리아에서 진행한 제17회 SAS 분석 챔피언십 이야기다. 17회를 맞은 SAS 분석 챔피언십은, 현업이 아닌 대학생에게 현업의 데이터를 주고 분석하도록 하는 대회다. 17회의 후원 기업은 [redacted] 홈쇼핑으로, [redacted]홈쇼핑의 실제 매출 데이터로 2018년 매출을 예측하도록 했다. 주어진 데이터는 2013년부터 2017년까지의 판매 상품 정보, 프로그램 실적, 프로그램 편성 정보, 편성 시간표 등이다. 편성정보나 실적에는 쇼 호스트나 PD, 매출액 등이 표기돼 있다.

많이 본 기사

테슬라, 배터리 데이 후 주가 떨어진 이유
by 이종철

우본 차세대 사업, 왜 SK C&C를 선택했나
by 홍하나

카카오페이, 마이데이터 서비스 처음으로 선보였다
by 홍하나

"구글이 30% 수수료를 안 떼가면 어떤 일이 생길까?" 어느 교수의 실험
by 남혜현

https://www.mk.co.kr/news/business/view/2018/09/576315/
https://www.etoday.co.kr/news/view/1882077
http://www.citydaily.kr/news/articleView.html?idxno=831

NS홈쇼핑

Contents

NS홈쇼핑

데이터 설명
Data Description

2020 빅콘테스트
2020 BIG CONTEST

제공 데이터

● 실적 데이터

- 2019년 1월 ~ 12월 프로그램별 실적 데이터
  - 상품별 과거(연속편성횟수/편성분), 판매가, 카테고리 정보 제공함
  - 예측 상품 중 판매가 0인 프로그램 실적은 예측에서 제외
  - 예측 상품 중 과거 실적이 없는 경우는 유사 카테고리 혹은 동일 머더코드로 예측

방송일시	노출(분)	마더코드	상품코드	상품명	상품군	판매단가	취급액
2019-01-01 06:00:00	20.0000	100346	201072	테이트 남성 셀린니트3종	의류	39900	2099000.0000
2019-01-01 06:00:00	nan	100346	201079	테이트 여성 셀린니트3종	의류	39900	4371000.0000
2019-01-01 06:20:00	20.0000	100346	201072	테이트 남성 셀린니트3종	의류	39900	3262000.0000
2019-01-01 06:20:00	nan	100346	201079	테이트 여성 셀린니트3종	의류	39900	6955000.0000
2019-01-01 06:40:00	20.0000	100346	201072	테이트 남성 셀린니트3종	의류	39900	6672000.0000
...	...	...	...	...	...	...	...
2019-12-31 23:20:00	nan	100448	201391	일시불쿠첸압력밥솥 6인용	주방	148000	1664000.0000
2019-12-31 23:40:00	20.0000	100448	201383	무이자쿠첸압력밥솥 10인용	주방	178000	9149000.0000
2019-12-31 23:40:00	nan	100448	201390	일시불쿠첸압력밥솥 10인용	주방	168000	15282000.0000
2019-12-31 23:40:00	nan	100448	201384	무이자쿠첸압력밥솥 6인용	주방	158000	2328000.0000
2019-12-31 23:40:00	nan	100448	201391	일시불쿠첸압력밥솥 6인용	주방	148000	10157000.0000

38300 rows × 7 columns

제공 데이터

● 시청률 데이터

- 2019년 1월 1일 ~12월 31일까지 시청률 데이터
  - 요일별/시간대별 분 단위 시청률 데이터 (단위 %)
  - 오전 6시부터 익일 오전 2시까지의 일별 시청률

시간대	2019-01-01	2019-01-02	2019-01-03	2019-01-04	2019-01-05	2019-01-06	2019-01-07	2019-01-08	2019-01-09	2019-01-10	...	2019-12-23	2019-12-24	2019-12-25	2019-12-26	2019-12-27	2019-12-28	2019-12-29	2019-12-30
02:00	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	...	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
02:01	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0050	...	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
02:02	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0050	...	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
02:03	0.0000	0.0000	0.0140	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0050	...	0.0000	0.0000	0.0170	0.0000	0.0000	0.0000	0.0000	0.0000
02:04	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0050	...	0.0000	0.0000	0.0170	0.0000	0.0000	0.0000	0.0000	0.0000
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
01:56	0.0000	0.0000	0.0000	0.0000	0.0270	0.0000	0.0000	0.0000	0.0000	0.0000	...	0.0130	0.0000	0.0000	0.0000	0.0150	0.0000	0.0000	0.0000
01:57	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0040	0.0000	...	0.0130	0.0000	0.0000	0.0000	0.0150	0.0000	0.0000	0.0000
01:58	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0040	0.0000	...	0.0130	0.0170	0.0000	0.0000	0.0150	0.0000	0.0000	0.0190
01:59	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0040	0.0000	...	0.0130	0.0000	0.0000	0.0000	0.0150	0.0000	0.0000	0.0000
월화수목금토일 02:00-01:59	0.0040	0.0060	0.0020	0.0030	0.0020	0.0030	0.0030	0.0020	0.0030	0.0020	...	0.0100	0.0060	0.0060	0.0070	0.0040	0.0060	0.0040	0.0050

1441 rows × 366 columns

외부 데이터

네이버 트랜드 데이터 Naver Trend Data

- 해당 검색어가 검색된 횟수를 일별/주별/월별 각각 합산
- 조회기간 내 최다 검색량을 100으로 설정하여 상대적인 변화를 나타냄

NAVER 데이터랩

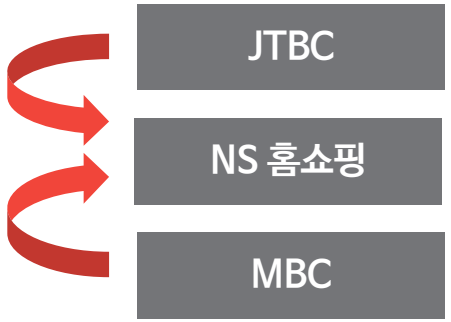
날짜	의류	농수산물	속옷	주방	미용	가전	생활용품	건강기능	잡화	가구
2019-01-01	0.3439	0.1982	5.0683	0.6237	0.3381	3.1349	1.8065	46.1994	5.6853	3.0817
2019-01-02	0.6131	0.1814	5.8685	0.9666	0.7007	5.1652	1.5285	49.4381	8.1430	5.9030
2019-01-03	0.7308	0.2343	7.5706	1.3187	0.7110	4.6937	1.5050	48.6204	8.4690	4.5661
2019-01-04	0.6596	0.1879	6.2758	1.1092	0.5196	4.6175	1.3156	43.9306	7.7646	3.6372
2019-01-05	0.7009	0.3037	10.0166	1.6295	0.9520	2.8369	1.2293	40.5435	6.5957	3.8534
...	...	...	...	...	...	...	...	...	...	...
2020-09-15	0.8308	0.6273	8.1722	1.6277	0.8986	4.9475	3.7359	48.6674	9.6931	5.3514
2020-09-16	0.9648	0.5274	8.7811	1.5862	0.7604	5.9928	3.2757	54.5443	9.8525	4.7231
2020-09-17	0.7874	0.3891	5.3985	1.0999	0.4810	6.7892	3.3550	46.5472	9.4838	4.1475
2020-09-18	0.7851	0.5657	5.5878	1.3431	0.6273	5.2291	3.7680	37.2188	8.9458	4.1781
2020-09-19	0.3912	0.3221	4.8608	0.8168	0.4372	3.6183	2.9797	44.9623	7.5558	4.0439

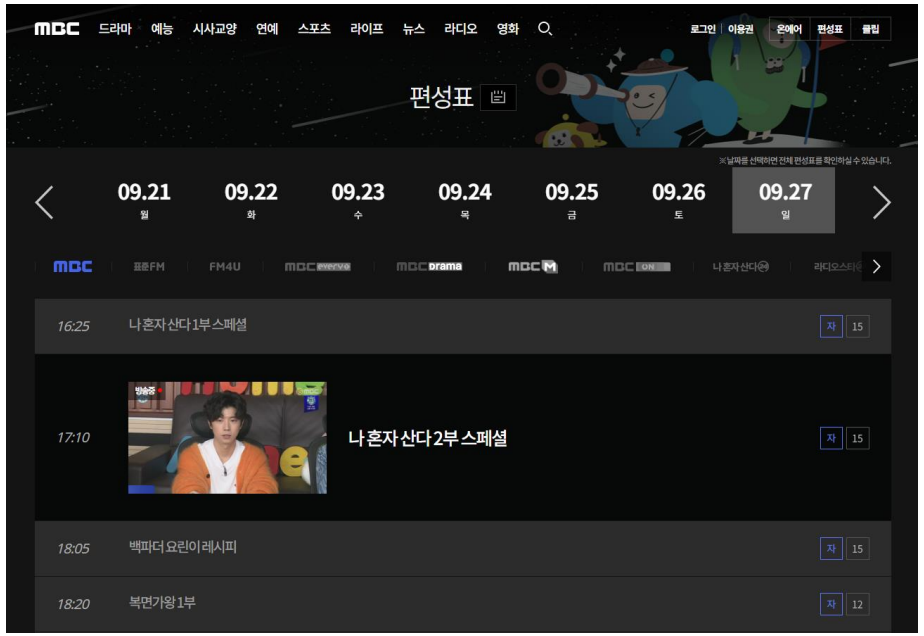
628 rows × 10 columns

외부 데이터

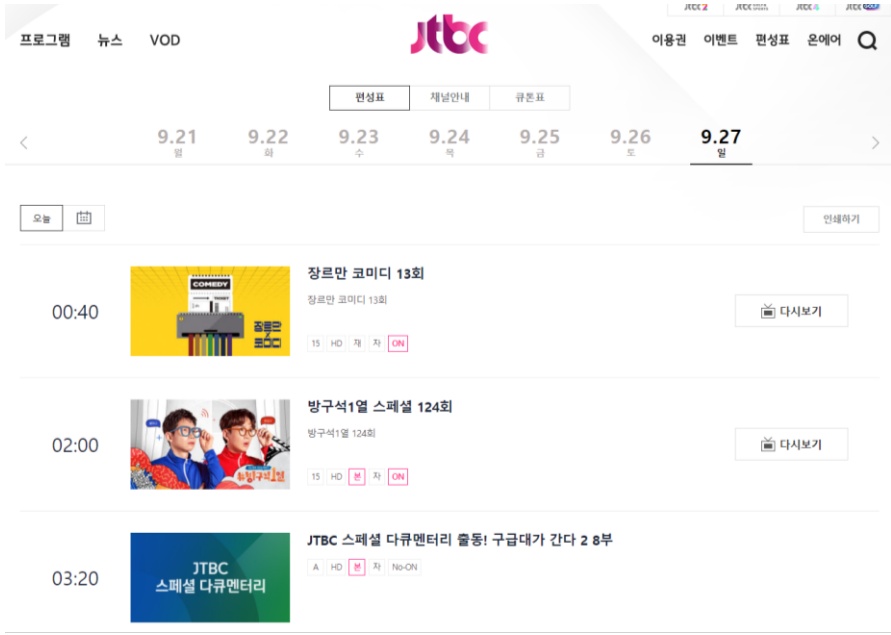
● 이웃 채널 방송편성표

- MBC와 JTBC의 방송 편성표를 활용
  - B tv : MBC(11) – NS홈쇼핑(14) – JTBC(15)
  - Olleh TV : MBC(11) – NS홈쇼핑(12) – JTBC(15)
  - U+ TV : MBC(11) – NS홈쇼핑(13) – JTBC(15)





MBC 시간표



JTBC 시간표

NS홈쇼핑

외부 데이터

● 소비자 물가 지수 Custom Price Index

- 포스트 코로나 상황을 반영하기 위한 지표로 활용
- 2019년 1월부터 2020년 6월까지 지출목적별 지수와 한국은행 기준 금리

학습 데이터 기간 (약 1년) | 평가 데이터 기간

지출목적별	2019. 01	2019. 02	2019. 03	2019. 04	2019. 05	2019. 06	2019. 07	2019. 08	2019. 09	2019. 10	2019. 11	2019. 12	2020. 06
0 총지수	104.2400	104.6900	104.4900	104.8700	105.0500	104.8800	104.5600	104.8100	105.2000	105.4600	104.8700	105.1200	104.8700
01 식료품 · 비주류 음료	108.8000	109.5400	108.4800	109.4000	108.8300	107.6400	106.6800	107.7000	110.5200	110.7400	107.8900	109.0100	111.1800
01.1 식료품	109.2200	110.0400	108.9500	109.9900	109.3100	108.0200	106.9800	108.0600	111.1400	111.2900	108.3600	109.5600	111.7700
빵 및 곡물	112.3000	112.5000	113.2600	113.3900	113.6400	113.5000	113.3900	113.2000	113.0900	113.4800	113.3800	113.0200	113.1500
쌀	118.2500	118.0700	118.0200	117.1000	117.4100	116.8200	116.4700	115.8200	116.0200	116.1600	116.3900	115.4800	114.5800
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
시험응시료	105.2300	105.2300	105.2300	105.2300	105.2300	105.2300	105.2300	105.2300	105.2300	110.0200	110.0200	110.0200	110.0200
장례비	105.4700	105.9900	106.2900	106.3200	106.3500	106.3200	106.3200	106.3200	106.3200	106.3400	106.3800	106.4000	106.8900
한국은행기준금리	1.7500	1.7500	1.7500	1.7500	1.7500	1.7500	1.5000	1.5000	1.2500	1.2500	1.2500	1.2500	0.5000

NS홈쇼핑

탐색적 데이터 분석
Exploratory Data Analysis

2020 빅콘테스트
2020 BIG CONTEST

데이터의 형태, 컬럼 그리고 타입

데이터 컬럼

방송일시	노출(분)	마더코드	상품코드	상품명	상품군	판매단가	취급액
2019-01-01 06:00:00	20.0000	100346	201072	테이트 남성 셀린니트3종	의류	39900	2099000.0000
2019-01-01 06:00:00	nan	100346	201079	테이트 여성 셀린니트3종	의류	39900	4371000.0000
2019-01-01 06:20:00	20.0000	100346	201072	테이트 남성 셀린니트3종	의류	39900	3262000.0000
2019-01-01 06:20:00	nan	100346	201079	테이트 여성 셀린니트3종	의류	39900	6955000.0000
2019-01-01 06:40:00	20.0000	100346	201072	테이트 남성 셀린니트3종	의류	39900	6672000.0000
...	...	...	...	...	...	...	...
2019-12-31 23:20:00	nan	100448	201391	일시불쿠첸압력밥솥 6인용	주방	148000	1664000.0000
2019-12-31 23:40:00	20.0000	100448	201383	무이자쿠첸압력밥솥 10인용	주방	178000	9149000.0000
2019-12-31 23:40:00	nan	100448	201390	일시불쿠첸압력밥솥 10인용	주방	168000	15282000.0000
2019-12-31 23:40:00	nan	100448	201384	무이자쿠첸압력밥솥 6인용	주방	158000	2328000.0000
2019-12-31 23:40:00	nan	100448	201391	일시불쿠첸압력밥솥 6인용	주방	148000	10157000.0000

38300 rows × 7 columns

데이터 형태

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 38300 entries, 2019-01-01 06
Data columns (total 14 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   노출(분)   38300 non-null  float64
 1   마더코드    38300 non-null  int64
 2   상품코드    38300 non-null  int64
 3   상품명     38300 non-null  object
 4   상품군     38300 non-null  object
 5   판매단가    38300 non-null  int64
 6   취급액     38300 non-null  float64
 7   연       38300 non-null  int64
 8   월       38300 non-null  int64
 9   주       38300 non-null  int64
 10  일       38300 non-null  int64
 11  시       38300 non-null  int64
 12  분       38300 non-null  int64
 13  요일      38300 non-null  object
dtypes: float64(2), int64(9), object(3)
memory usage: 4.4+ MB
```

데이터 타입

NS홈쇼핑

데이터의 기초 통계량

학습 데이터와 평가 데이터의
분포가 유사한 것으로 확인할 수 있음

	노출(분)	마더코드	상품코드	판매단가	취급액
count	38300.0000	38300.0000	38300.0000	38300.0000	38300.0000
mean	20.4451	100390.9859	201219.9726	456644.0078	23116300.6789
std	3.4164	249.9397	735.7036	726115.8633	20747022.7854
min	2.4667	100000.0000	200000.0000	0.0000	103000.0000
25%	20.0000	100155.0000	200550.0000	59000.0000	7712000.0000
50%	20.0000	100346.0000	201167.0000	109000.0000	16961000.0000
75%	20.0000	100596.0000	201863.0000	499000.0000	32718250.0000
max	60.0000	100849.0000	202513.0000	7930000.0000	322009000.0000

학습 데이터

	노출(분)	마더코드	상품코드	판매단가	취급액
count	2891.0000	2891.0000	2891.0000	2891.0000	.0000
mean	20.1797	100388.8212	201200.4618	401441.7848	nan
std	4.4062	256.7364	775.0423	604648.0012	nan
min	5.9500	100003.0000	200003.0000	0.0000	nan
25%	20.0000	100148.0000	200417.0000	40900.0000	nan
50%	20.0000	100388.0000	201277.0000	79900.0000	nan
75%	20.0000	100593.0000	201818.0000	548000.0000	nan
max	60.0000	100849.0000	202511.0000	4320000.0000	nan

평가 데이터

마더코드 / 상품코드

- **마더코드(100000~), 상품코드(200000~)로 상품의 특징 판별**
  - 같은 상품명이라도 상품 가격이 다르면 상품코드의 차이가 있음을 발견
  - 이를 통해 상품코드: 마더코드 + 판매단가를 결합한 Key값으로 인식가능

1271	2019-01-14 23:40:00	20.0	100049	202044	마리노블 밍크 롱코트	clothes	499000
2355	2019-01-26 00:00:00	20.0	100049	202043	마리노블 밍크 롱코트	clothes	399000

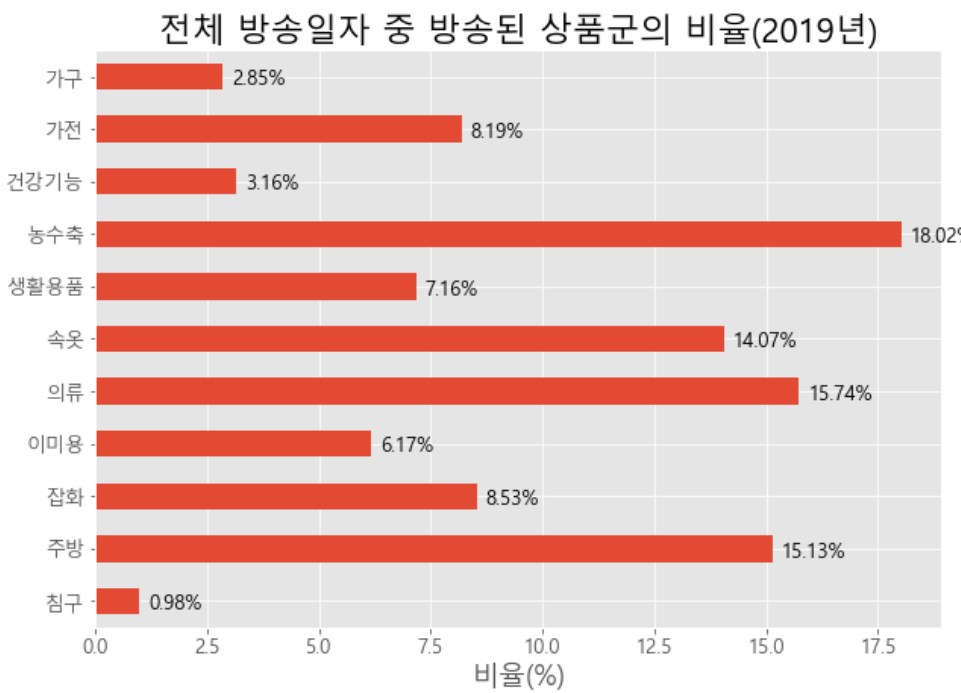
상품군 / 상품명

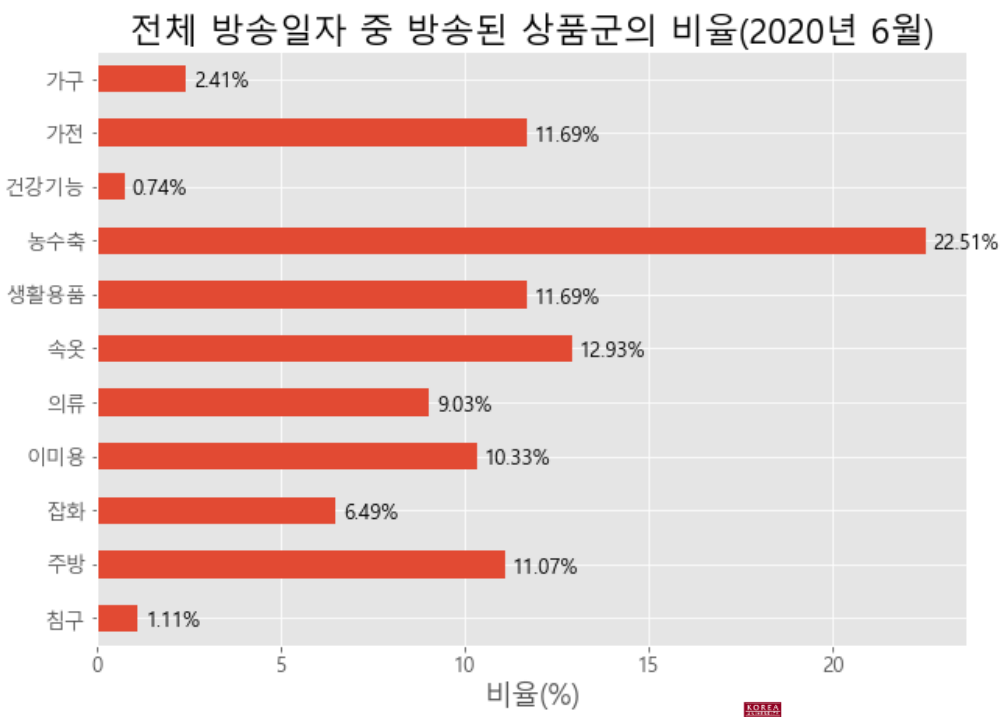
상품군

- 상품군 별로 통계값들을 살펴보아 전체적인 상품들의 특징들을 잡을 것

상품명

- 대략 1800개의 상품이 존재함

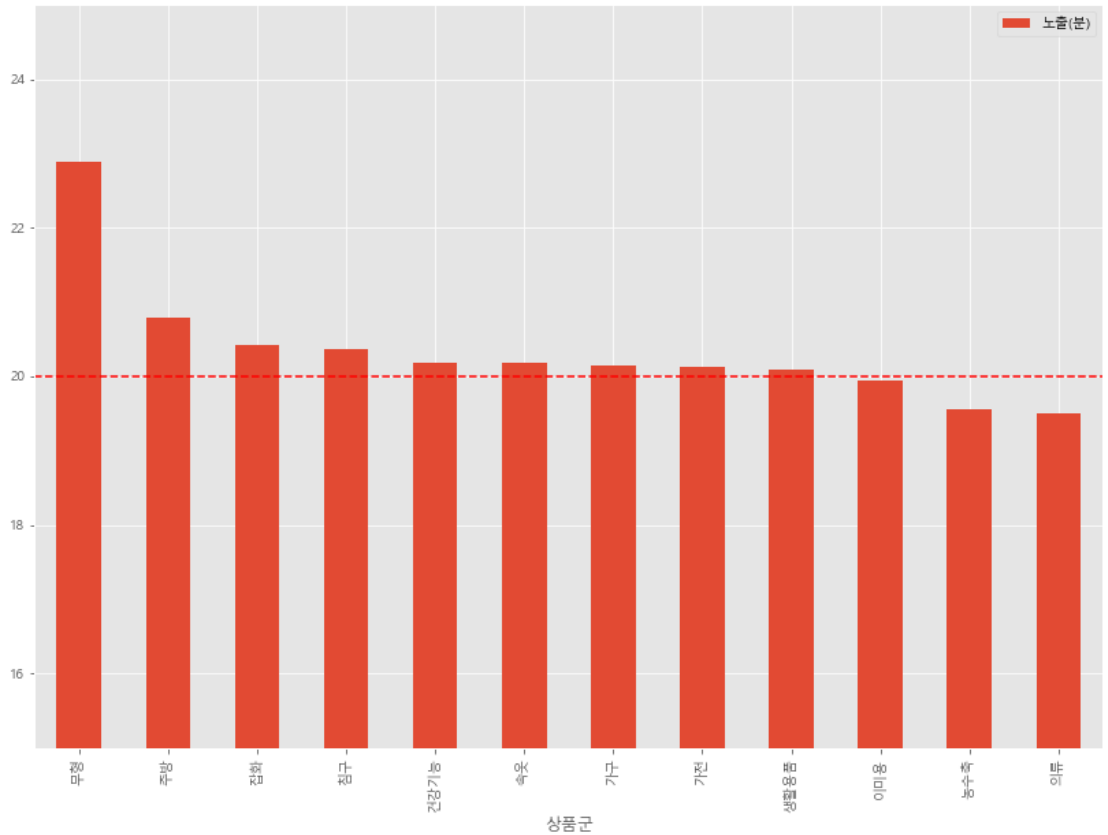




노출(분)

● 전체 분포 확인

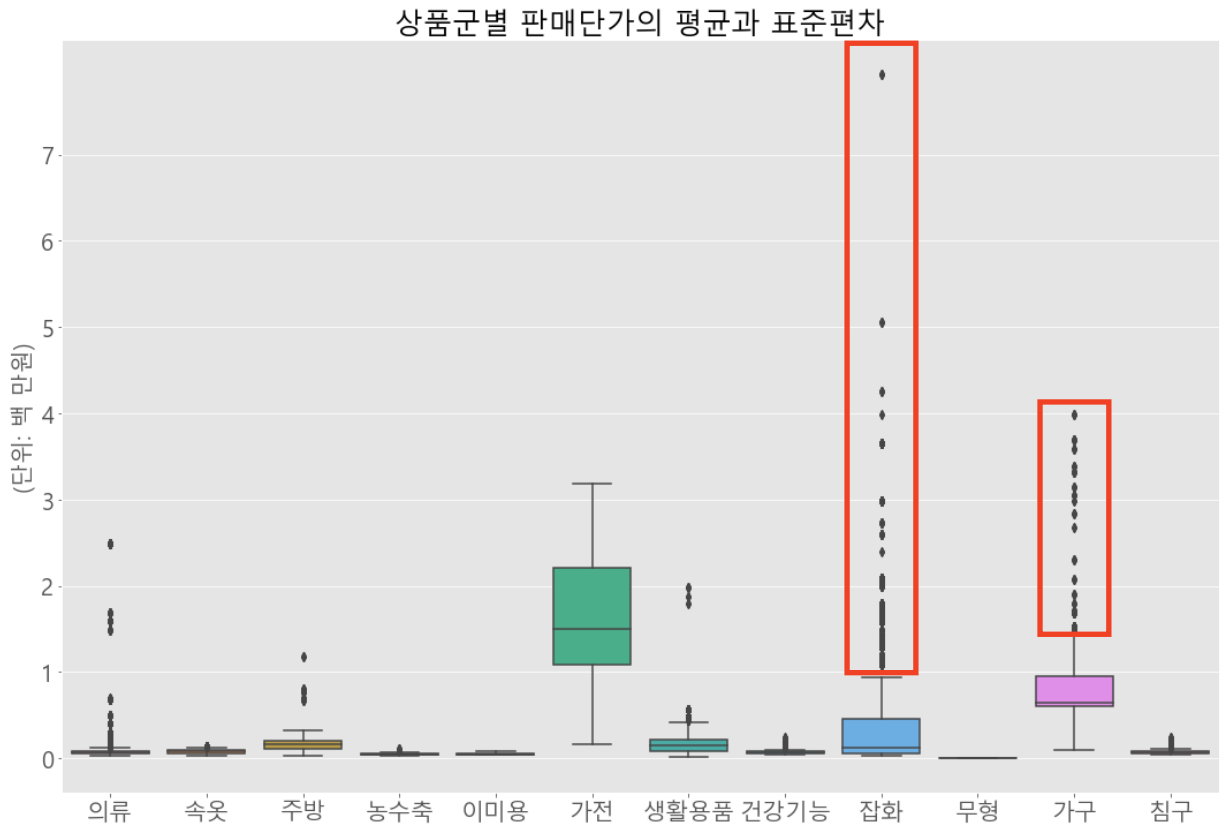
- 무형을 제외한 상품군들의 노출(분)은 20분 전후로 고르게 분포



NS홈쇼핑

판매단가

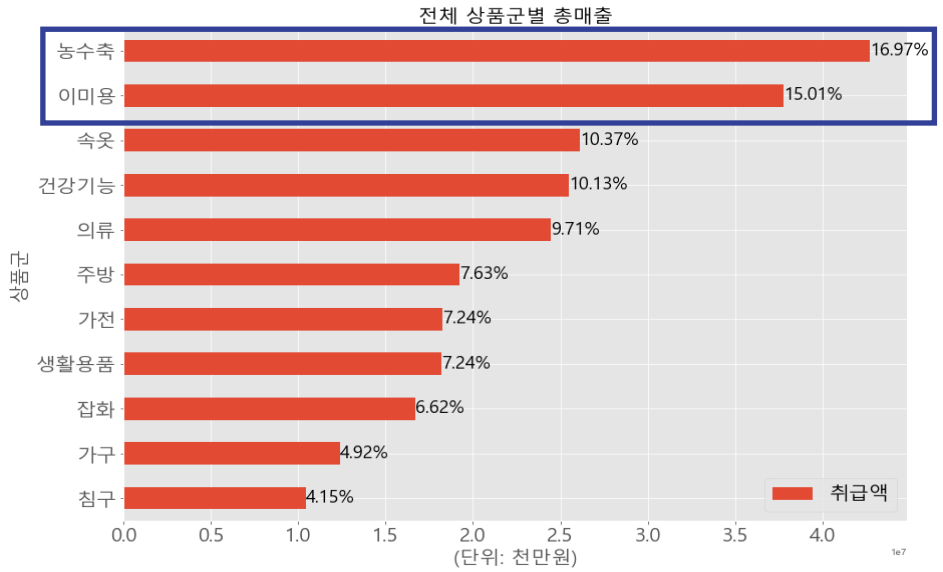
- 가전의 단가가 평균적으로 높음
- 잡화와 가구에 Outlier가 많음

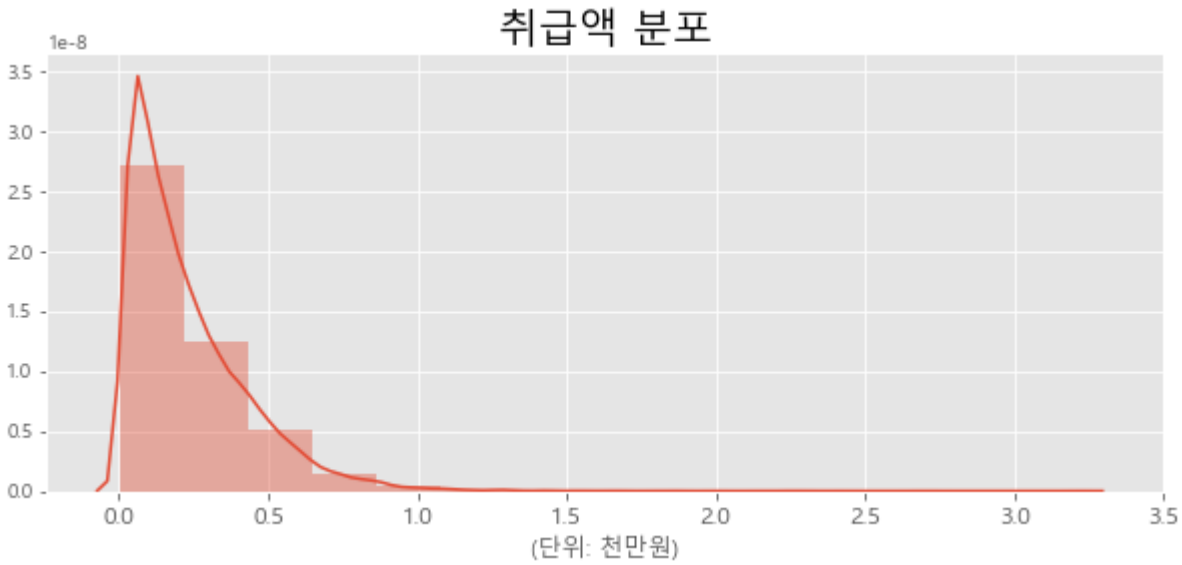


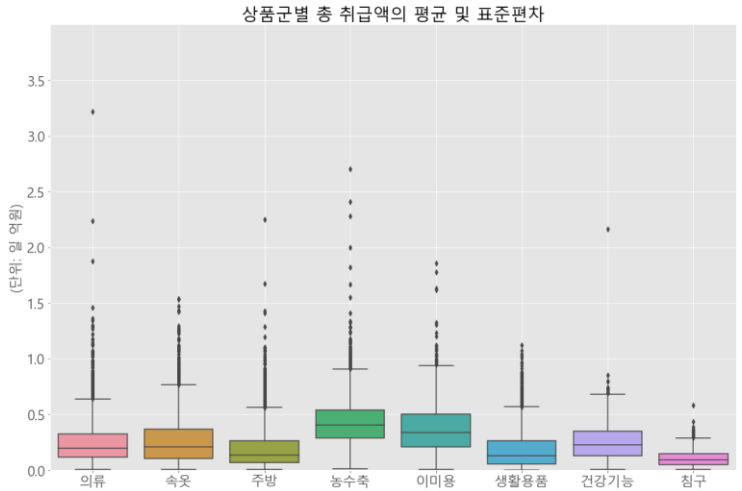
NS홈쇼핑

취급액

- 전체 분포 확인
  - 5천만원 이하에 집중
- 전체 상품군 별 총 매출
  - 농수축과 이미용의 취급액이 가장 높음



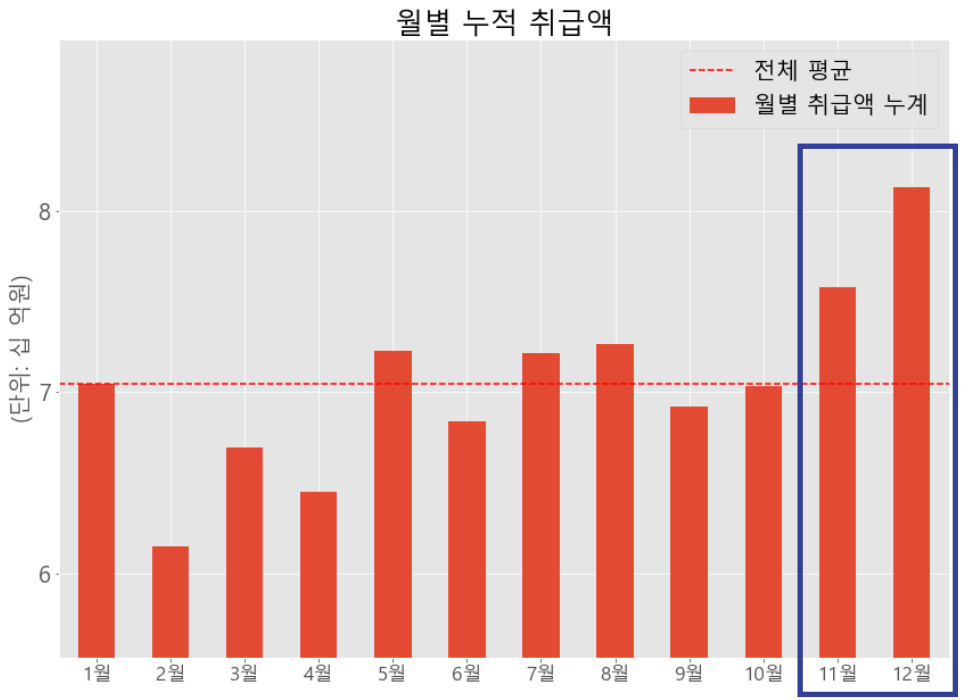




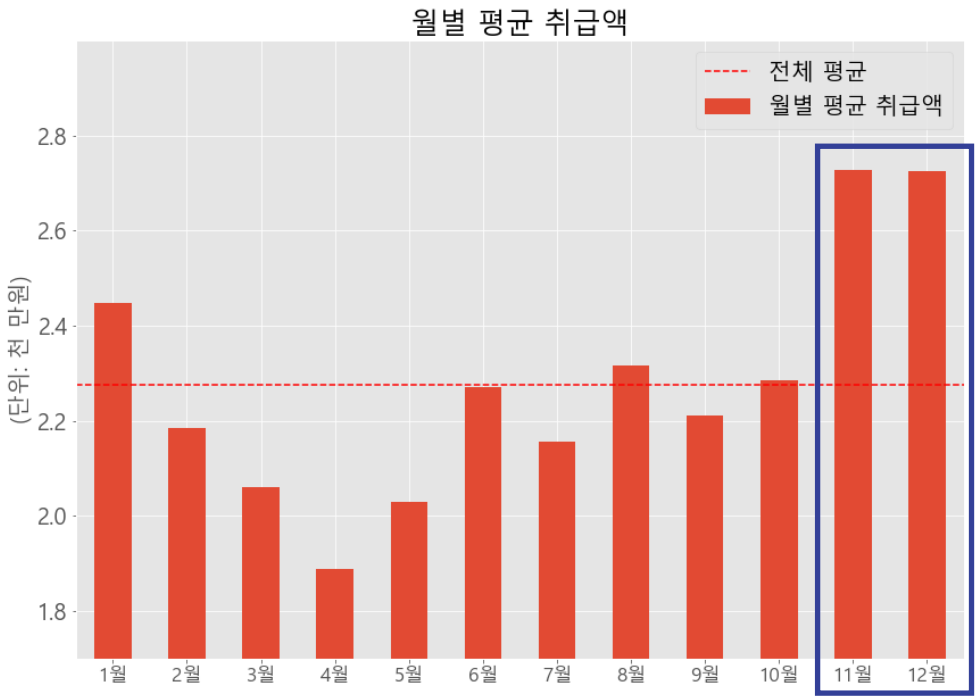
2020 빅콘테스트
2020 BIG CONTEST

월별 상품 총 판매량

- 연말(11,12월)에 상품 판매량이 높음



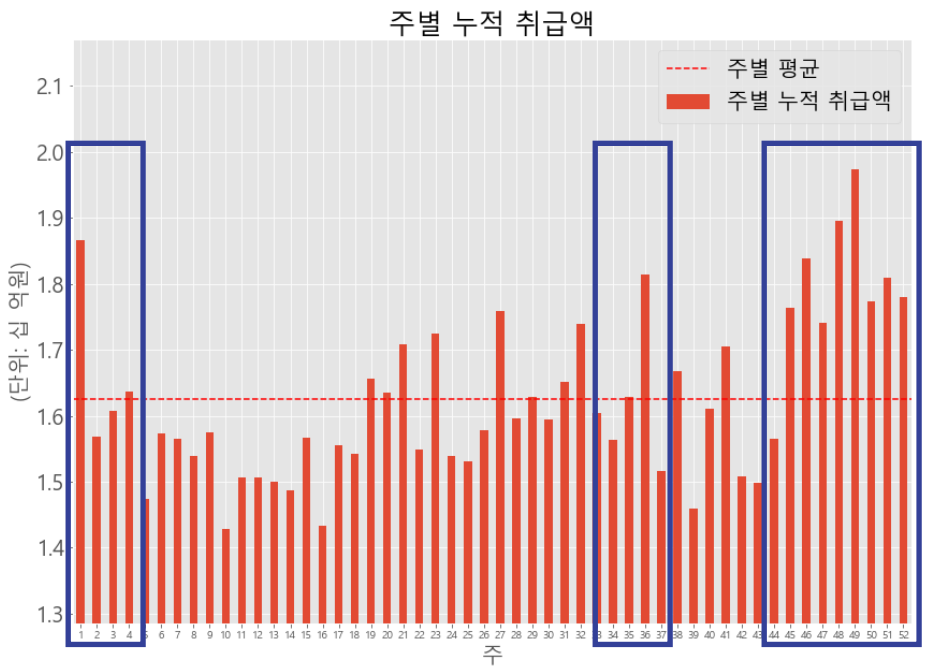
총 판매량 합계



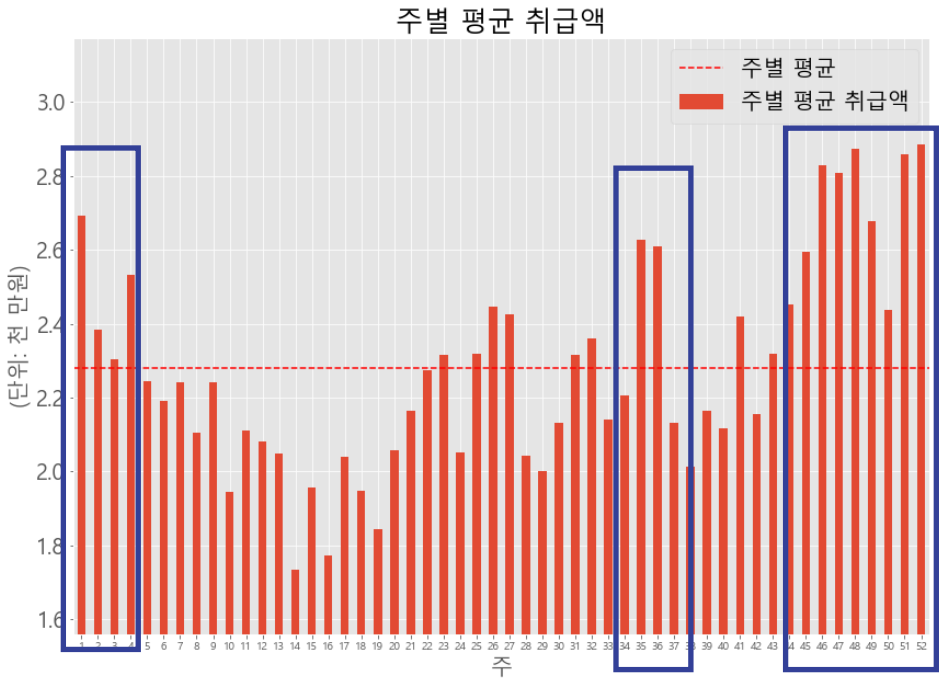
평균 판매량 합계

주별 상품 총 판매량

- 연말 이외에도 특정 주간(추석, 설날, 여름휴가철)에 판매량이 높음



총 판매량 합계

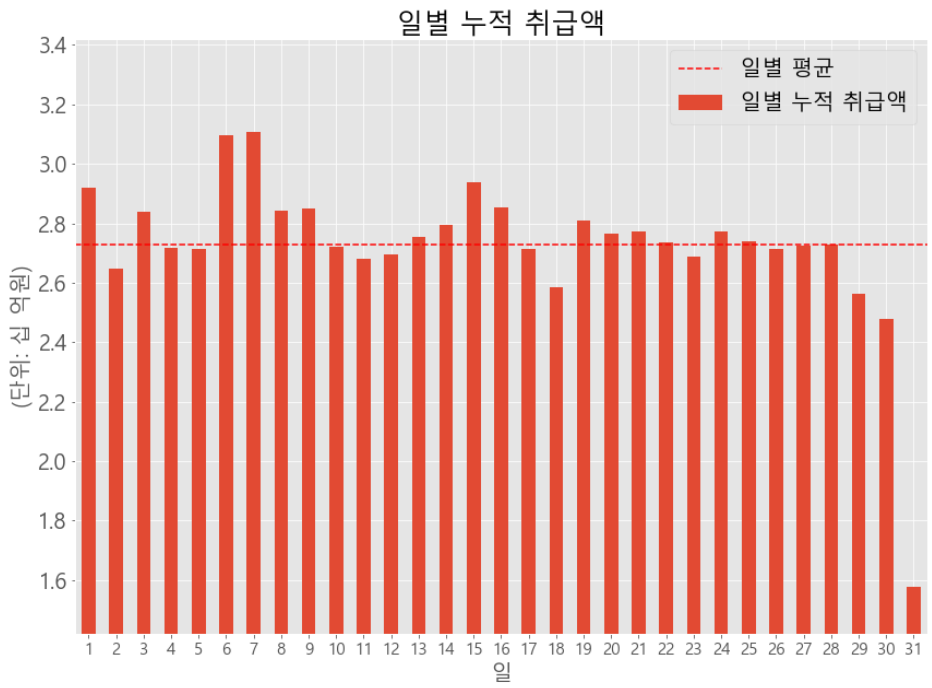


평균 판매량 합계

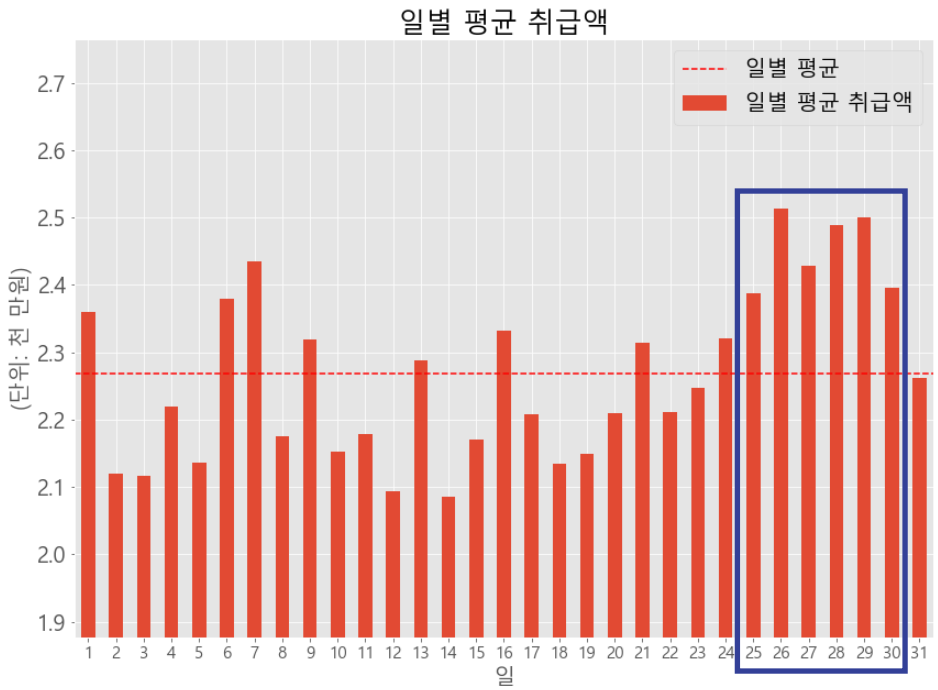
NS홈쇼핑

일별 상품 총 판매량

- 매월 25일 이후로 판매량이 많음



총 판매량 합계

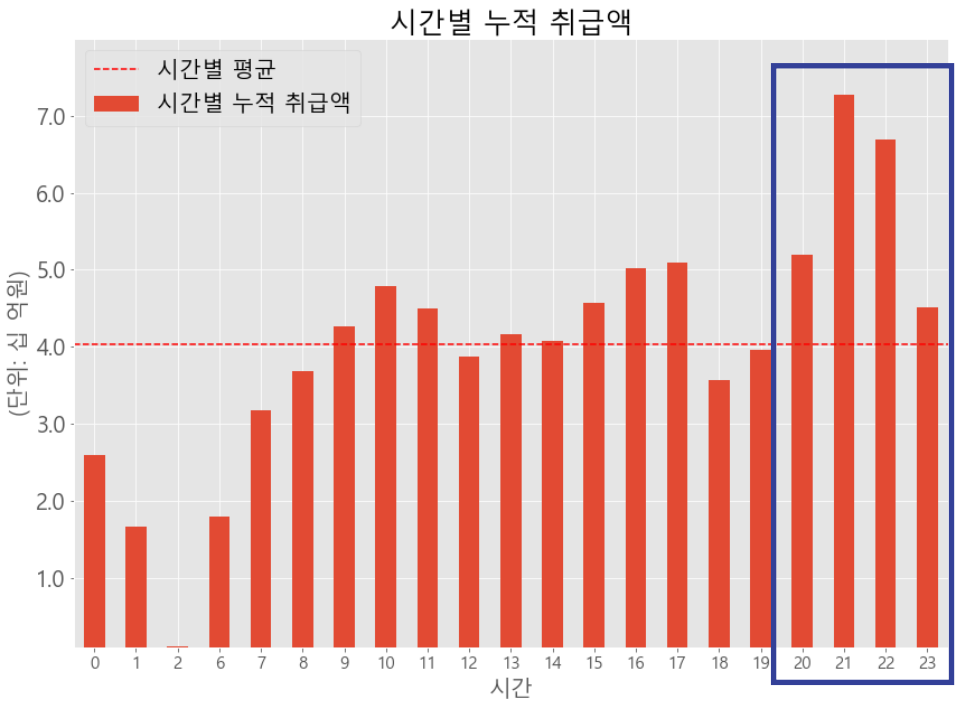


평균 판매량 합계

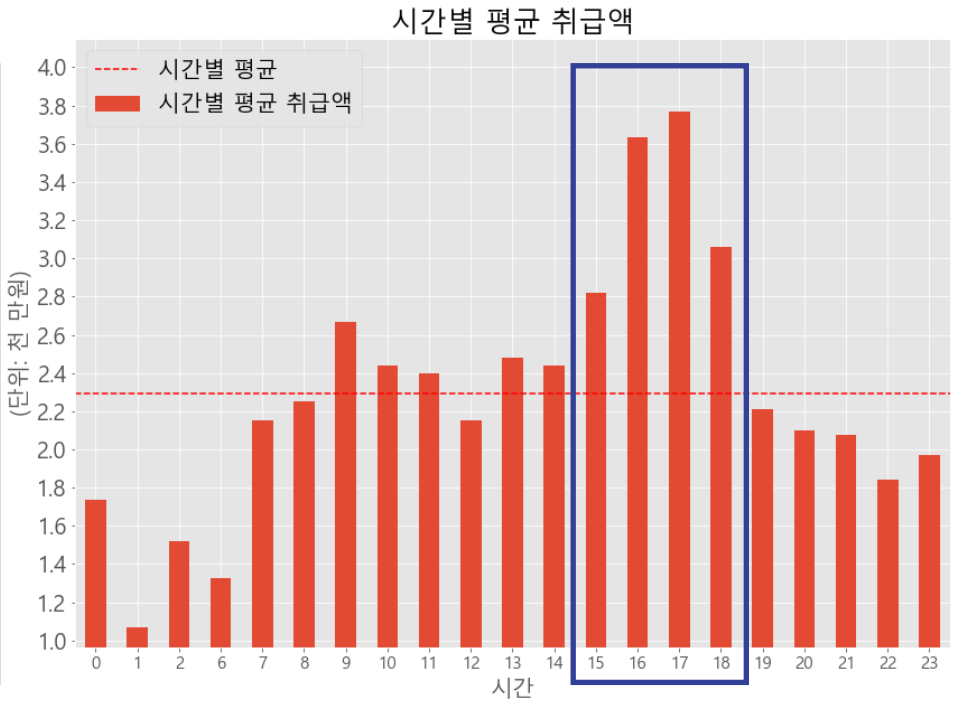
NS홈쇼핑

시간별 상품 총 판매량

- 오후시간(15~18시)과 저녁시간(20~23시)에 판매량이 많음
- 2~5시 사이에는 방영하지 않음



총 판매량 합계

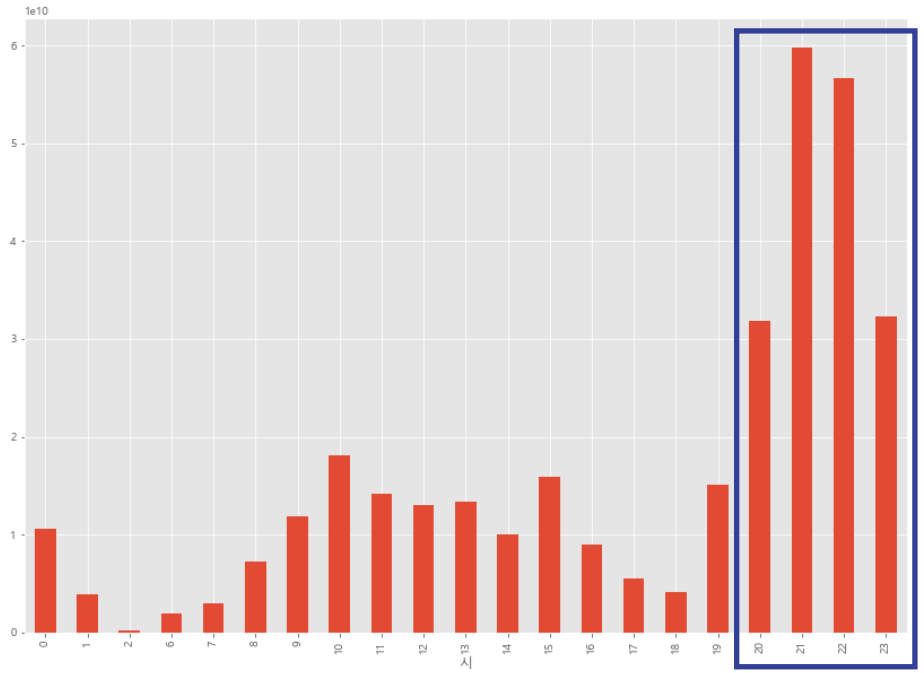


평균 판매량 합계

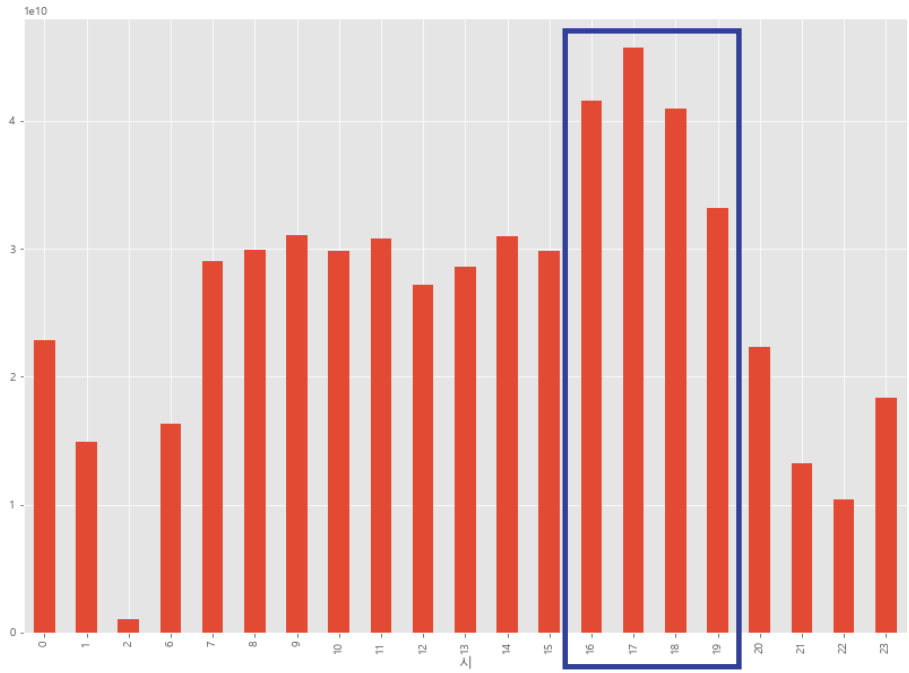
NS홈쇼핑

시간별 상품 총 판매량

- 총 판매량과 평균 판매량의 차이가 발생하는 원인
  - 저녁에는 가격이 높은 상품들은 오후에는 가격이 낮은 상품들 위주로 판매가 많기 때문이라고 추정



단가 10만원 이상

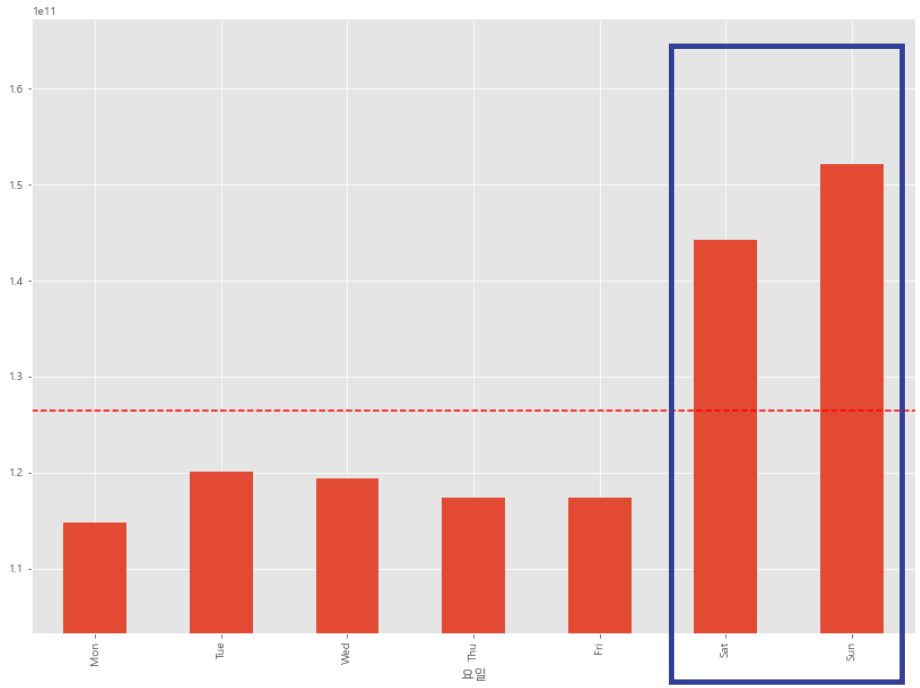


단가 10만원 미만

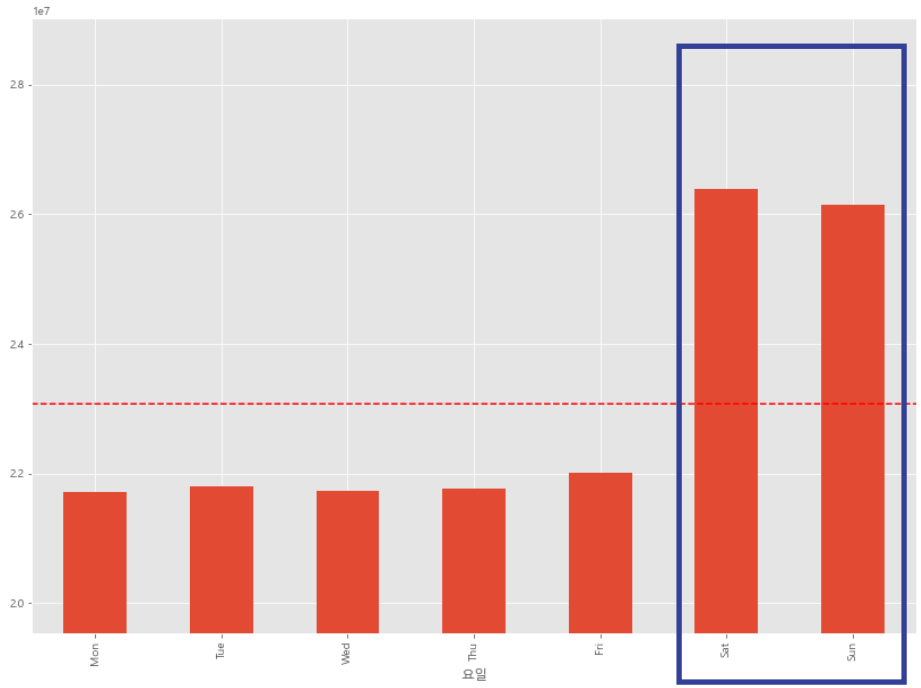
NS홈쇼핑

요일별 상품 총 판매량

- 주말(토,일)에 판매량이 많음



총 판매량 합계

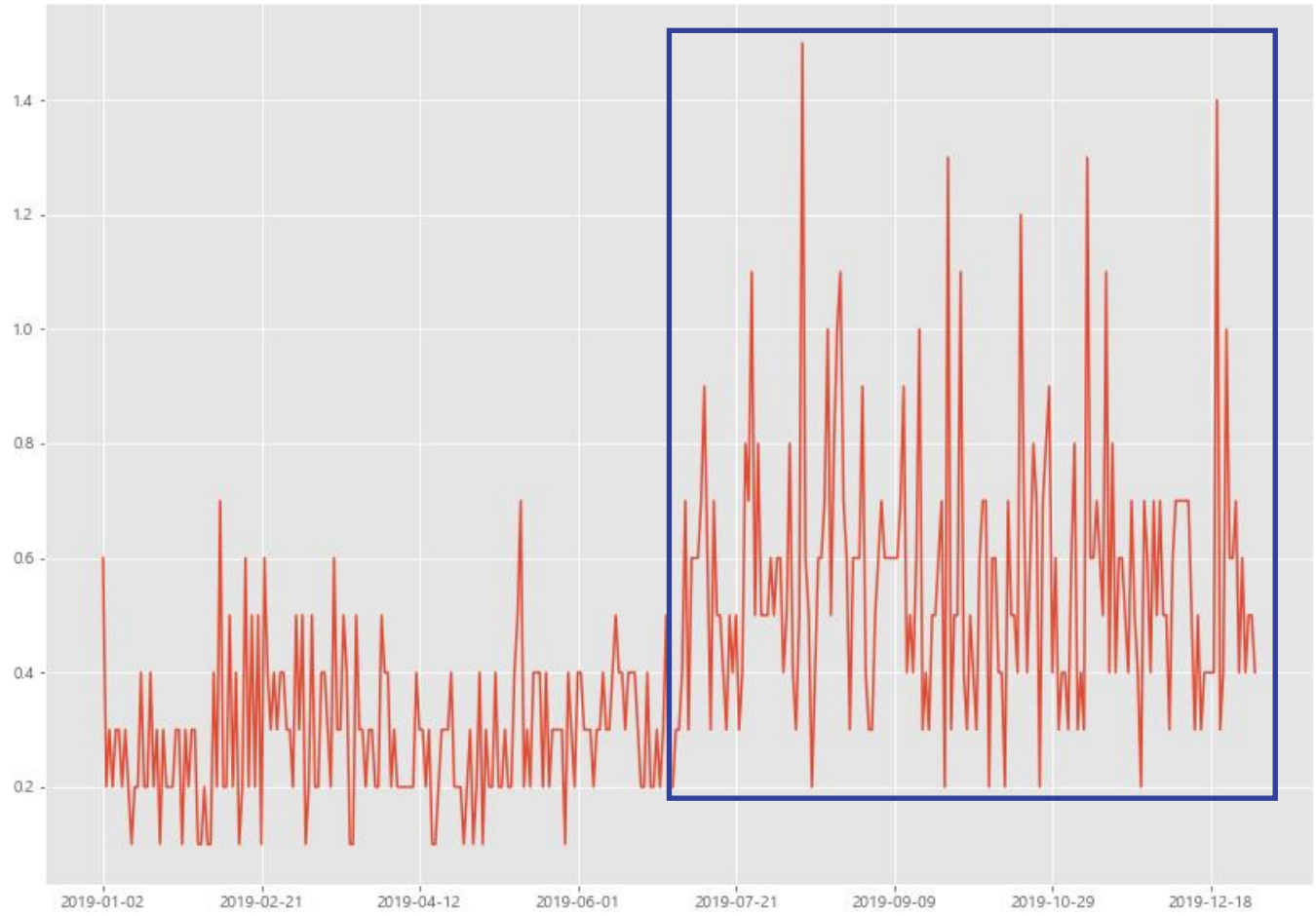


평균 판매량 합계

NS홈쇼핑

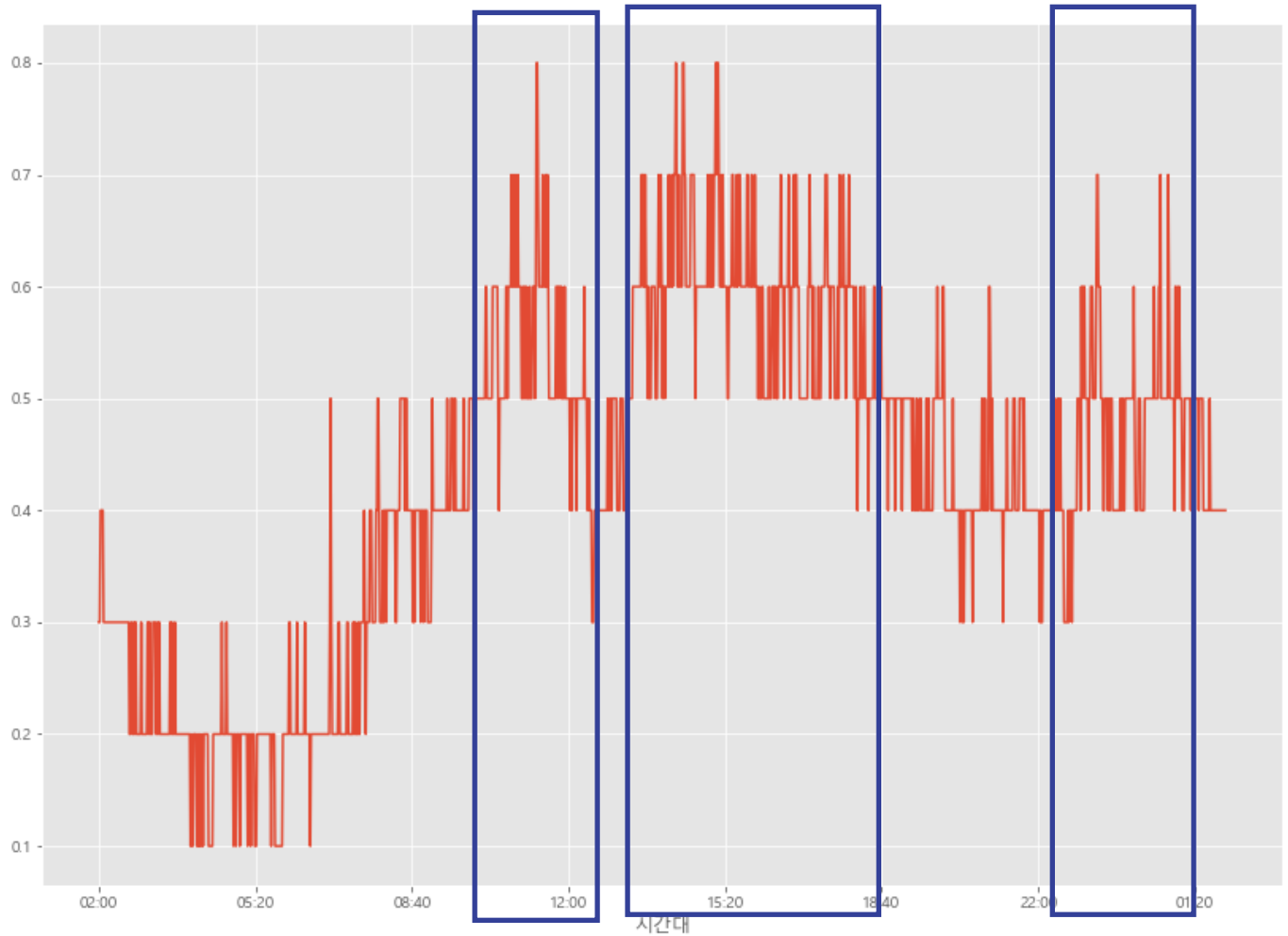
일별 평균 시청률

- 상반기에 비해 하반기(7월 1일 이후)에 시청률이 높음



시간대별 평균 시청률

- 10~12시, 14~18시, 21~01시에 시청률이 높음



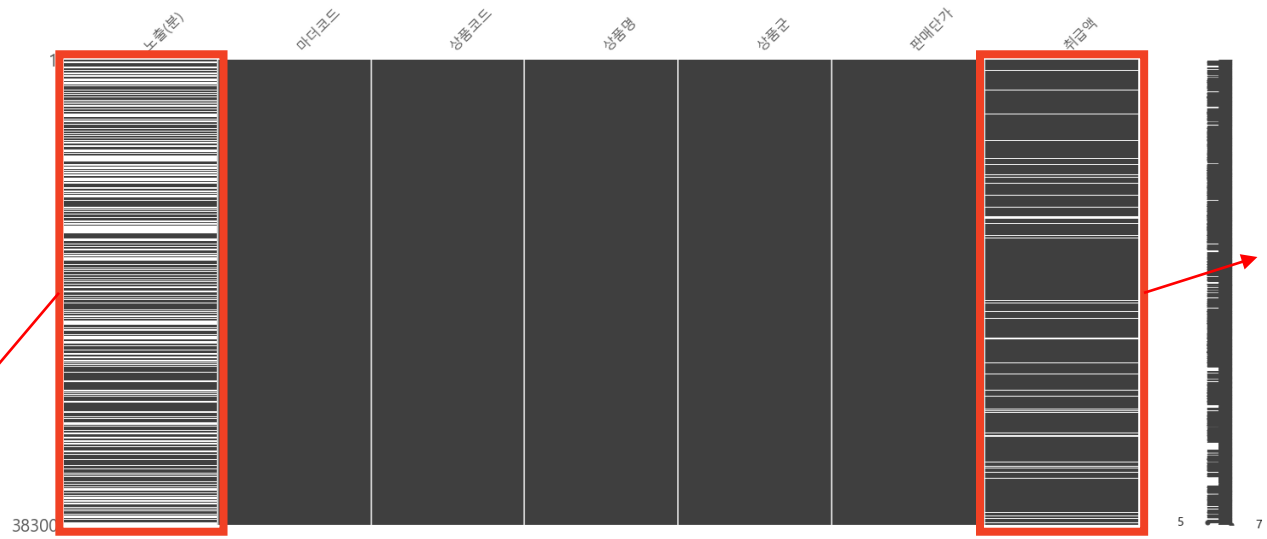
NS홈쇼핑

데이터 전처리
Data Preprocessing

2020 빅콘테스트
2020 BIG CONTEST

NS홈쇼핑

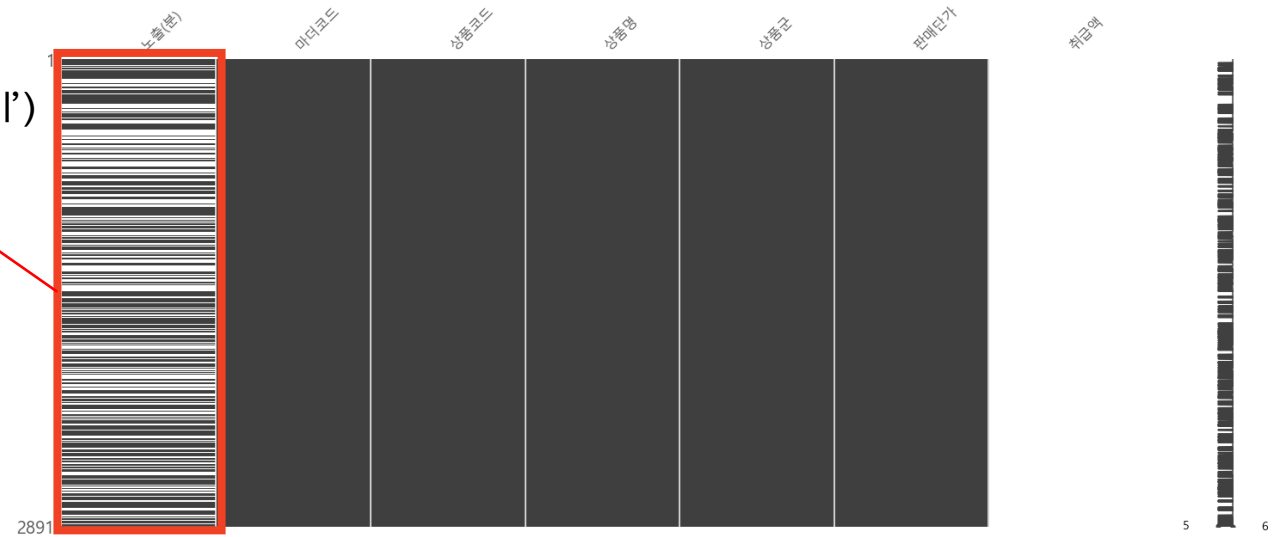
결측치 처리



학습 데이터

무형 상품군으로 인한
취급액 결측치 처리
→ drop 처리

동일/유사 상품군으로
방영으로 인한 노출시간
결측치 처리
→ fillna(method='ffill')

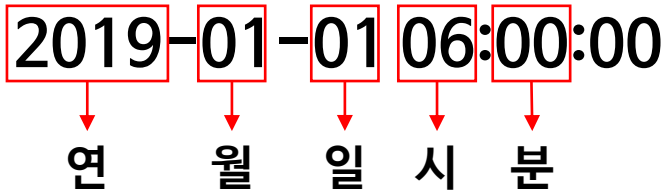


평가 데이터

방송 시간

● 세부 방송 시간 추출

- 연, 월, 일, 시, 분으로 세부적으로 분리



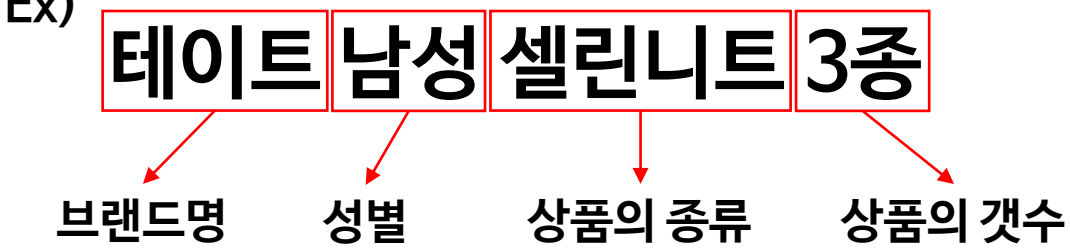
방송일시	노출(분)	마더코드	상품코드	상품명	상품군	판매단가	취급액
2019-01-01 06:00:00	20.0000	100346	201072	테이트 남성 셀린니트3종	의류	39900	2099000.0000
2019-01-01 06:00:00	nan	100346	201079	테이트 여성 셀린니트3종	의류	39900	4371000.0000
2019-01-01 06:20:00	20.0000	100346	201072	테이트 남성 셀린니트3종	의류	39900	3262000.0000
2019-01-01 06:20:00	nan	100346	201079	테이트 여성 셀린니트3종	의류	39900	6955000.0000
2019-01-01 06:40:00	20.0000	100346	201072	테이트 남성 셀린니트3종	의류	39900	6672000.0000
...	...	...	...	...	...	...	...
2019-12-31 23:20:00	nan	100448	201391	일시불쿠첸압력밥솥 6인용	주방	148000	1664000.0000
2019-12-31 23:40:00	20.0000	100448	201383	무이자쿠첸압력밥솥 10인용	주방	178000	9149000.0000
2019-12-31 23:40:00	nan	100448	201390	일시불쿠첸압력밥솥 10인용	주방	168000	15282000.0000
2019-12-31 23:40:00	nan	100448	201384	무이자쿠첸압력밥솥 6인용	주방	158000	2328000.0000
2019-12-31 23:40:00	nan	100448	201391	일시불쿠첸압력밥솥 6인용	주방	148000	10157000.0000

38300 rows × 7 columns

자연어 처리 Natural Language Processing

- 주어진 데이터에서 가장 핵심이 되는 특성은 **상품명**에 있다고 판단
- 상품명을 추출하기 위해서 자연어처리기술을 활용
  - 사람이 분류할 수 없는 분량의 상품명들에 대한 특성 추출 자동화
  - 사람이 확인할 수 없는 자연어 사이의 관계에 대한 대수적 표현과 내재된 표현 추출

Ex)



자연어 처리 Natural Language Processing

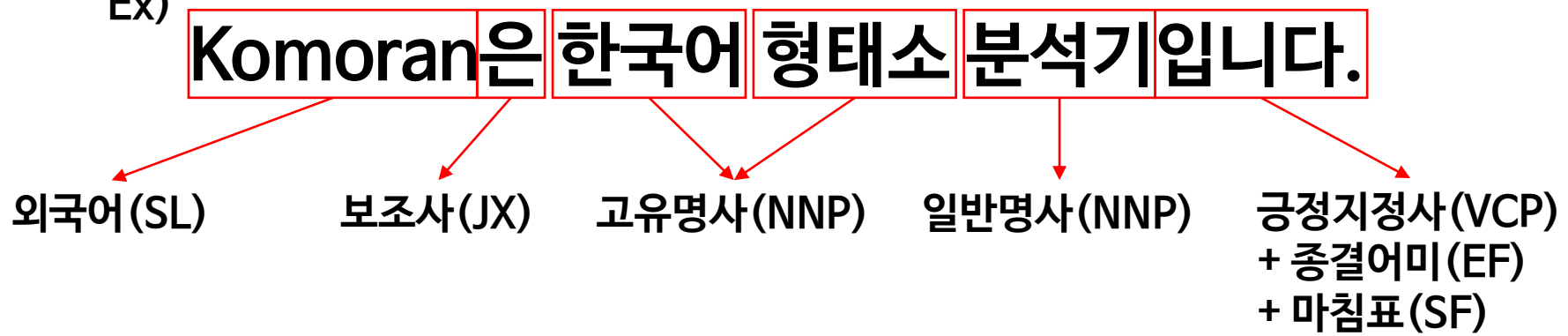
불용어 제거

- 상품명 분석 과정에서 성능 저해를 불러 일으키는 큰 요소 제거
- 더 높은 정확도를 위해 특수문자와 큰 공백들, 낱개의 문자들을 제거
  - 예시) 그(관사), 와(전치사), 그리고(접속사)

형태소 분석

- 한 문장 내에서 각 단어 요소들이 문장 내 어떤 역할을 하고 있는지 분석
- Java 기반의 한국어 형태소 분석기인 Komoran을 통하여 형태소 분석
  - 1차적으로 고유 상품명에 대해 형태소 분석을 진행

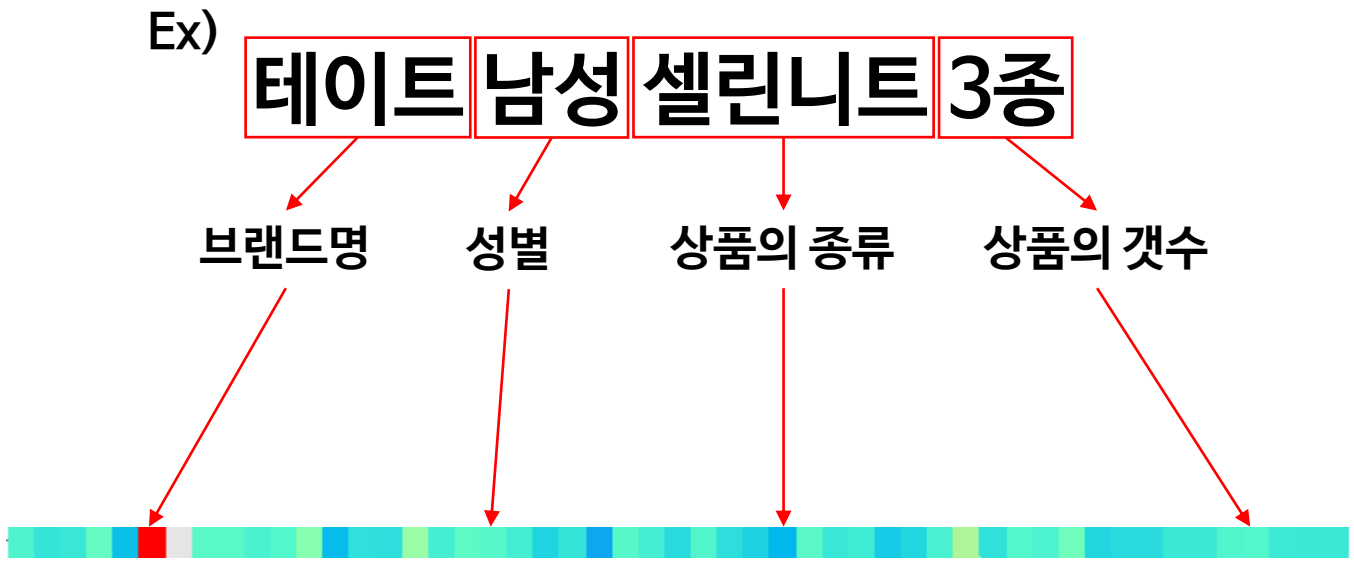
Ex)



자연어 처리 Natural Language Processing

● 임베딩 벡터 생성

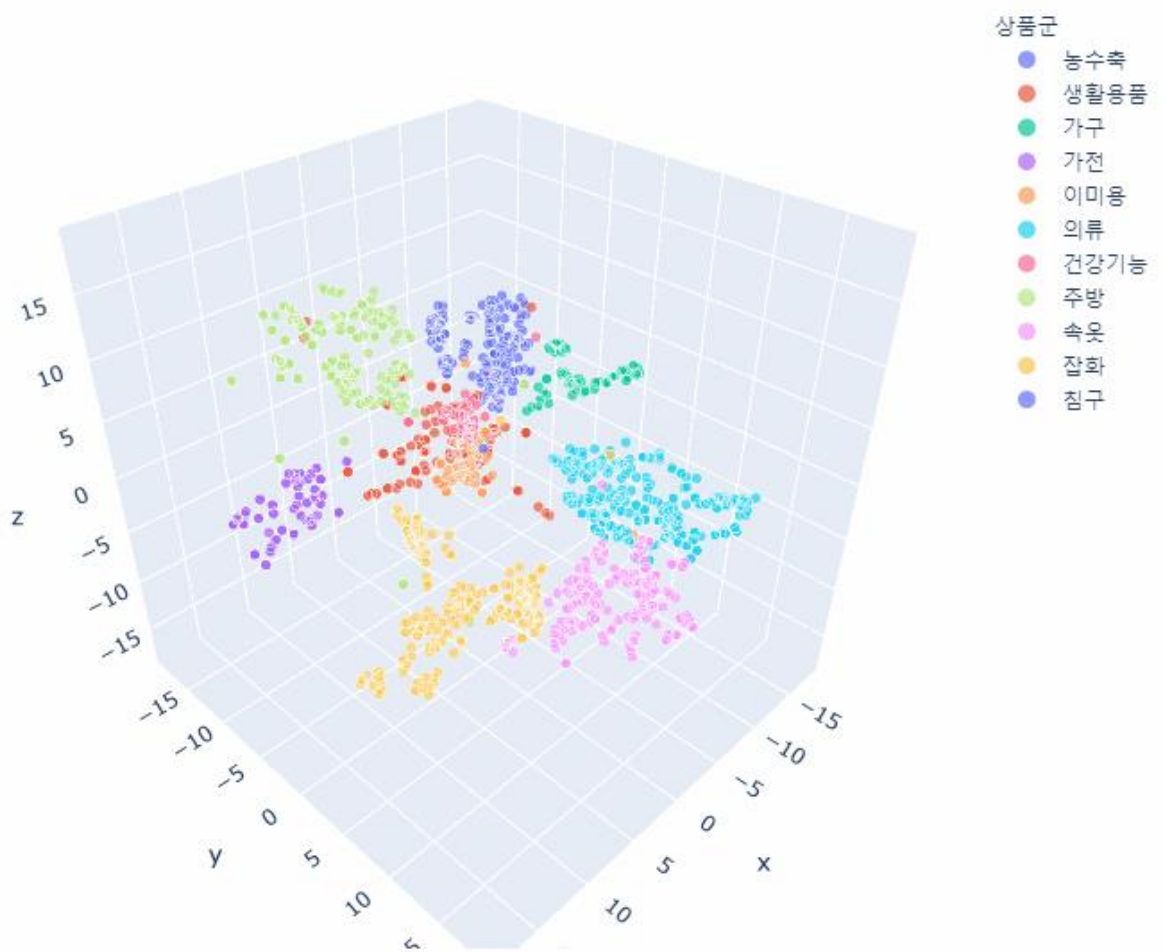
- Word2Vec 모델을 이용하여 상품명으로부터 유의미한 단어들을 파생변수 추출
- 100차원 공간으로 상품명을 매핑



자연어 처리 Natural Language Processing

PCA를 통해서 차원 축소

- 모델이 데이터를 해석하는 능력에 있어서 변수의 수를 조절해줄 필요



NS홈쇼핑

특성 공학
Feature Engineering

특성 생성하기 Feature Generation

● 네이버 트랜드 결합

방송일시	노출(분)	마더코드	상품코드	상품명	상품군	판매단가	취급액
2019-01-01 06:00:00	20.0000	100346	201072	테이트 남성 셀린니트3종	의류	39900	2099000.0000
2019-01-01 06:00:00	nan	100346	201079	테이트 여성 셀린니트3종	의류	39900	4371000.0000
2019-01-01 06:20:00	20.0000	100346	201072	테이트 남성 셀린니트3종	의류	39900	3262000.0000
2019-01-01 06:20:00	nan	100346	201079	테이트 여성 셀린니트3종	의류	39900	6955000.0000
2019-01-01 06:40:00	20.0000	100346	201072	테이트 남성 셀린니트3종	의류	39900	6672000.0000
...	...	...	...	...	...	...	...
2019-12-31 23:20:00	nan	100448	201391	일시불쿠첸압력밥솥 6인용	주방	148000	1664000.0000
2019-12-31 23:40:00	20.0000	100448	201383	무이자쿠첸압력밥솥 10인용	주방	178000	9149000.0000
2019-12-31 23:40:00	nan	100448	201390	일시불쿠첸압력밥솥 10인용	주방	168000	15282000.0000
2019-12-31 23:40:00	nan	100448	201384	무이자쿠첸압력밥솥 6인용	주방	158000	2328000.0000
2019-12-31 23:40:00	nan	100448	201391	일시불쿠첸압력밥솥 6인용	주방	148000	10157000.0000

38300 rows × 7 columns

실적 데이터

날짜	의류	농수산물	속옷	주방	미용	가전	생활용품	건강기능	잡화	가구
2019-01-01	0.3439	0.1982	5.0683	0.6237	0.3381	3.1349	1.8065	46.1994	5.6853	3.0817
2019-01-02	0.6131	0.1814	5.8685	0.9666	0.7007	5.1652	1.5285	49.4381	8.1430	5.9030
2019-01-03	0.7308	0.2343	7.5706	1.3187	0.7110	4.6937	1.5050	48.6204	8.4690	4.5661
2019-01-04	0.6596	0.1879	6.2758	1.1092	0.5196	4.6175	1.3156	43.9306	7.7646	3.6372
2019-01-05	0.7009	0.3037	10.0166	1.6295	0.9520	2.8369	1.2293	40.5435	6.5957	3.8534
...	...	...	...	...	...	...	...	...	...	...
2020-09-15	0.8308	0.6273	8.1722	1.6277	0.8986	4.9475	3.7359	48.6674	9.6931	5.3514
2020-09-16	0.9648	0.5274	8.7811	1.5862	0.7604	5.9928	3.2757	54.5443	9.8525	4.7231
2020-09-17	0.7874	0.3891	5.3985	1.0999	0.4810	6.7892	3.3550	46.5472	9.4838	4.1475
2020-09-18	0.7851	0.5657	5.5878	1.3431	0.6273	5.2291	3.7680	37.2188	8.9458	4.1781
2020-09-19	0.3912	0.3221	4.8608	0.8168	0.4372	3.6183	2.9797	44.9623	7.5558	4.0439

628 rows × 10 columns

네이버 트랜드 데이터

시	분	embed_0	embed_1	embed_2	embed_3	embed_4	embed_5	embed_6	embed_7	embed_8	embed_9	월간누적방영횟수	주간누적방영횟수	prime_time	네이버트랜드
.0000	0.0000	-10.2002	-7.5838	1.9713	-0.2518	-0.7333	1.0323	0.2297	-1.3625	-4.5576	1.2723	1.0000	1.0000	0.0000	0.0264
.0000	0.0000	-10.6281	-7.2654	1.6163	0.2269	-0.6321	0.5803	0.7824	-2.0310	-4.2865	1.5060	1.0000	1.0000	0.0000	0.0264
.0000	20.0000	-10.2002	-7.5838	1.9713	-0.2518	-0.7333	1.0323	0.2297	-1.3625	-4.5576	1.2723	1.0000	1.0000	0.0000	0.0264
.0000	20.0000	-10.6281	-7.2654	1.6163	0.2269	-0.6321	0.5803	0.7824	-2.0310	-4.2865	1.5060	1.0000	1.0000	0.0000	0.0264
.0000	40.0000	-10.2002	-7.5838	1.9713	-0.2518	-0.7333	1.0323	0.2297	-1.3625	-4.5576	1.2723	1.0000	1.0000	0.0000	0.0264
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
.0000	20.0000	11.2997	-0.5756	-2.3893	8.1535	1.5604	-0.7269	1.8258	-0.6493	-2.9599	0.5341	17.0000	1.0000	1.0000	0.1270
.0000	40.0000	11.3231	-0.5708	-2.3749	8.1365	1.5969	-0.6616	1.8768	-0.6743	-2.9661	0.6485	17.0000	1.0000	1.0000	0.1270
.0000	40.0000	11.2997	-0.5756	-2.3893	8.1535	1.5604	-0.7269	1.8258	-0.6493	-2.9599	0.5341	17.0000	1.0000	1.0000	0.1270
.0000	40.0000	11.3231	-0.5708	-2.3749	8.1365	1.5969	-0.6616	1.8768	-0.6743	-2.9661	0.6485	17.0000	1.0000	1.0000	0.1270
.0000	40.0000	11.2997	-0.5756	-2.3893	8.1535	1.5604	-0.7269	1.8258	-0.6493	-2.9599	0.5341	17.0000	1.0000	1.0000	0.1270

네이버 트랜드 데이터와 결합된 실적 데이터

특성 생성하기 Feature Generation

평일 / 휴일

- EDA에서 주말(토,일)에 상품 판매가 높았음
- 휴일이 있는 특정 주간에 판매량이 높았음을 이용하여 피처 생성
- 휴일에 따른 상품 판매차이를 고려

남성 / 여성

- 같은 상품(주로 의류)일지라도, 남성/여성에 따라 상품의 판매량이 다른 것을 확인
- 성별에 따른 상품 판매차이를 고려

2019-01-01 06:00:00	20.0000	100346	201072	테이트	남성	셀린니트3종	의류	39900	2099000.0000
2019-01-01 06:00:00	nan	100346	201079	테이트	여성	셀린니트3종	의류	39900	4371000.0000
2019-01-01 06:20:00	20.0000	100346	201072	테이트	남성	셀린니트3종	의류	39900	3262000.0000
2019-01-01 06:20:00	nan	100346	201079	테이트	여성	셀린니트3종	의류	39900	6955000.0000

특성 생성하기 Feature Generation

● 무이자 / 일시불

- 가전과 같이 판매가격이 높은 상품들은 구입 방법으로 무이자, 일시불 선택가능
- 일시불이 무이자보다 가격이 저렴

● 상품별 방영 횟수

- 주력 상품의 경우, 1회성이 아닌 여러 번 방영을 통해 수익을 극대화 가능
- 판매횟수와 취급액의 관계를 확인 가능

2019-06-01 09:30:00	30.0	100683	202032	무이자 매직쉐프 10리터 듀얼쿡 에어프라이어	kitchen	164000	55647000.0
2019-06-01 09:30:00	NaN	100683	202035	일시불 매직쉐프 10리터 듀얼쿡 에어프라이어	kitchen	154000	71581000.0

무이자 / 일시불 예시

한일 대용량 스텐 분쇄믹서기	401
안동간고등어 20팩	318
무이자 LG전자 매직스페이스 냉장고	308
일시불 LG전자 매직스페이스 냉장고	308
무이자 LG 울트라HD TV 65UK6800HNC	295
일시불 LG 울트라HD TV 65UK6800HNC	295
일시불 LG 울트라HD TV 55UK6800HNC	277
무이자 LG 울트라HD TV 55UK6800HNC	277
무이자 LG 통돌이 세탁기	275
일시불 LG 통돌이 세탁기	275
AAB의 소곱창전골 800g x 8팩	252
무이자 LG 울트라HD TV 70UK7400KNA	247
일시불 LG 울트라HD TV 70UK7400KNA	247
에코라믹 통주물 스톤 냄비세트	231
일시불 쿠쿠전기밥솥 10인용 (QS)	191
무이자 쿠쿠전기밥솥 6인용(QS)	191
무이자 쿠쿠전기밥솥 10인용(CRP-QS107FG/FS)	191
일시불 쿠쿠전기밥솥 6인용(QS)	191
(쿠)무이자 쿠첸 압력밥솥 6인용 (A1)	177
(쿠)일시불 쿠첸 압력밥솥 6인용 (A1)	177

상품별 방영 횟수 예시

특성 생성하기 Feature Generation

방송 재핑타임 Zapping Time

- TV 인기 프로그램이 끝난 뒤 시청자들이 리모컨을 들고 채널을 돌리는 짧은 순간
- MBC, JTBC의 사이에 있으므로, 해당 채널들의 주요 방영표를 참고하여 이들을 재핑타임 확인가능

방송 골든타임 Golden Time

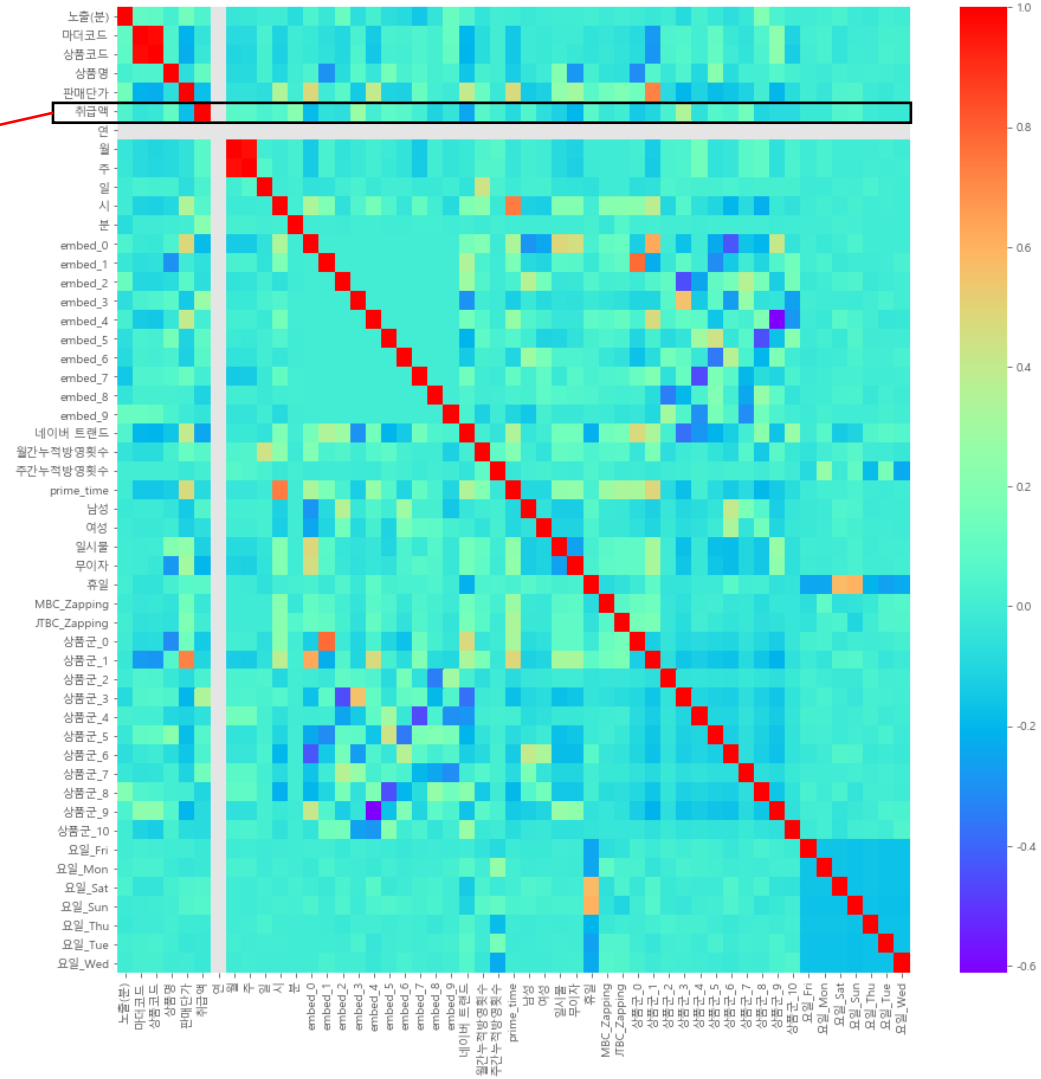
- 시청취율이 높아서 광고비가 가장 비싼 방송 시간대
- 평일은 오후 8시 ~ 밤 12시 사이, 토요일은 오후 7시 ~ 오후 11시 30분 사이, 일요일은 오후 6시 ~ 오후 11시 30분 사이가 황금시간대

http://plus.hankyung.com/apps/newsinside.view?aid=202003096861A
https://ko.wikipedia.org/wiki/%ED%99%A9%EA%B8%88%EC%8B%9C%EA%B0%84%EB%8C%80

특성 선택하기 Feature Selection

● 생성된 피처들 사이의 상관계수

취급액은 앞서
자연어 처리된 피처와
상대적으로 상관계수가
높은 것을 확인



특성 선택하기 Feature Selection

● 최종 선택된 피처들

'노출(분)', '월', '일', '시', '분', '요일_Mon', '요일_Tue', '요일_Wed', '요일_Thu', '요일_Fri', '요일_Sat', '요일_Sun' '월간누적방영횟수', '주간누적방영횟수', 'prime_time', 'MBC_Zapping', 'JTBC_Zapping',	방송시간 관련 피처들
'마더코드', '상품코드', '네이버 트랜드', '상품군_0', '상품군_1', '상품군_2', '상품군_3', '상품군_4', '상품군_5', '상품군_6', '상품군_7', '상품군_8', '상품군_9', '상품군_10',	상품군 관련 피처들
'embed_0', 'embed_1', 'embed_2', 'embed_3', 'embed_4', 'embed_5', 'embed_6', 'embed_7', 'embed_8', 'embed_9', '남성', '여성', '일시불', '무이자', '휴일',	상품명 관련 피처들
'판매단가',	판매단가 관련 피처들

선택된 전체 피처들

NS홈쇼핑

모델링
Modeling

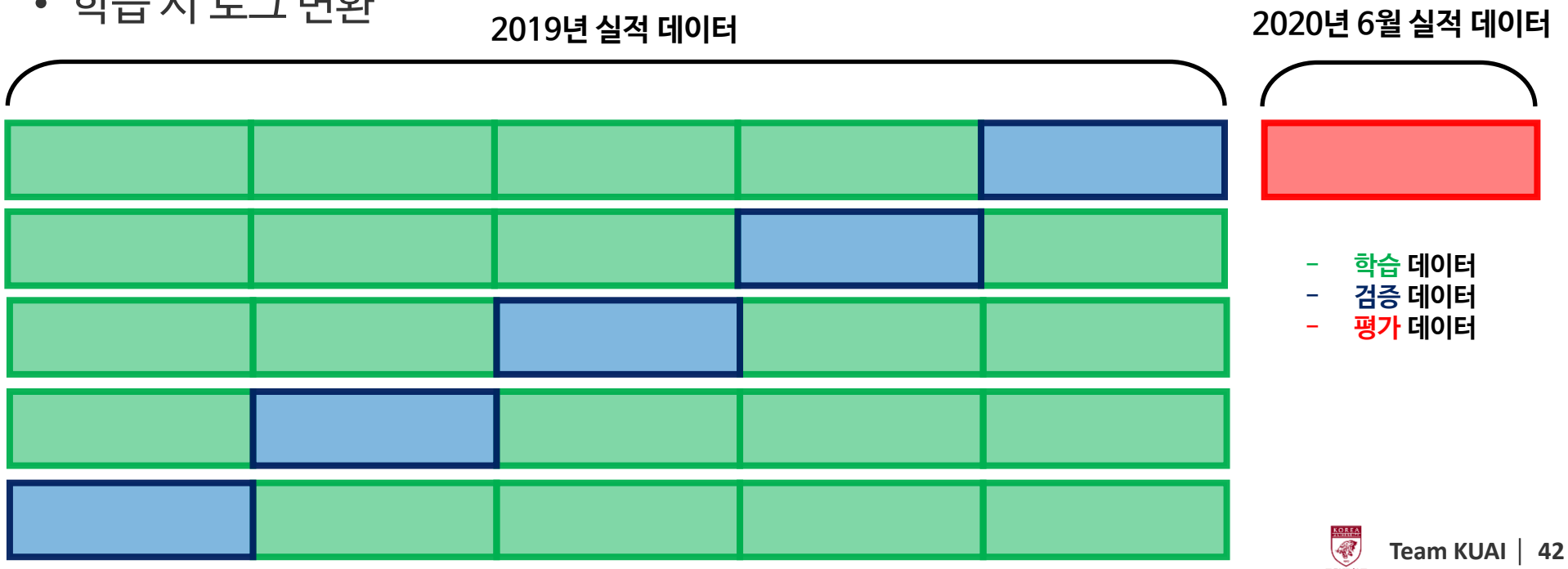
2020 빅콘테스트
2020 BIG CONTEST

학습, 검증 및 평가 데이터 분리

- **교차 검증 Cross Validation**
  - 실적 데이터가 충분하지 않으므로 교차검증 진행
- **취급액 데이터 스케일링 처리**
  - 최소 금액 단위가 100만원이므로 학습 시 모든 취급액을 100만원으로 나눔
  - 학습 시 로그 변환

```
SCALED_UNIT = 10000000
```
executed in 14ms, finished 21:37:45 2020-09-27

```
X_train, X_valid, y_train, y_valid = train_test_split(data['train'].drop('취급액', axis=1),
                                                      data['train']['취급액'] / SCALED_UNIT,
                                                      shuffle=True,
                                                      test_size=TEST_SIZE,
                                                      random_state=RANDOM_STATE)
```
executed in 30ms, finished 21:37:45 2020-09-27



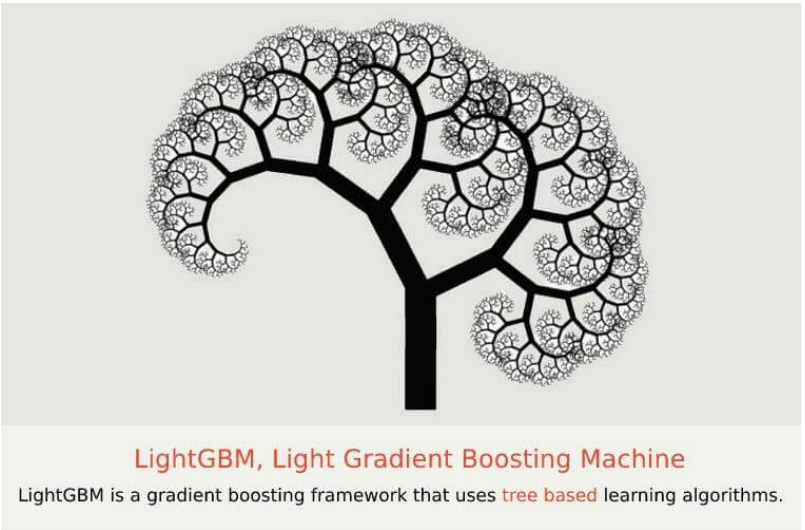
머신러닝 모델 Machine Learning Model

● 머신러닝 모델을 선택해본 이유

- 주어진 데이터가 선형적인 관계를 가지고 있다고 가정
- 결과에 대한 모델의 설명가능성(XAI)을 위해서

● 사이킷런에서 제공하는 머신러닝 11개의 모델

- LinearRegression, DecisionTreeRegressor, RandomForestRegressor, XGBRegressor, LGBMRegressor, KNeighborsRegressor, SVR, AdaBoostRegressor, GradientBoostingRegressor, VotingRegressor, BaggingRegressor



머신러닝 모델 Machine Learning Model

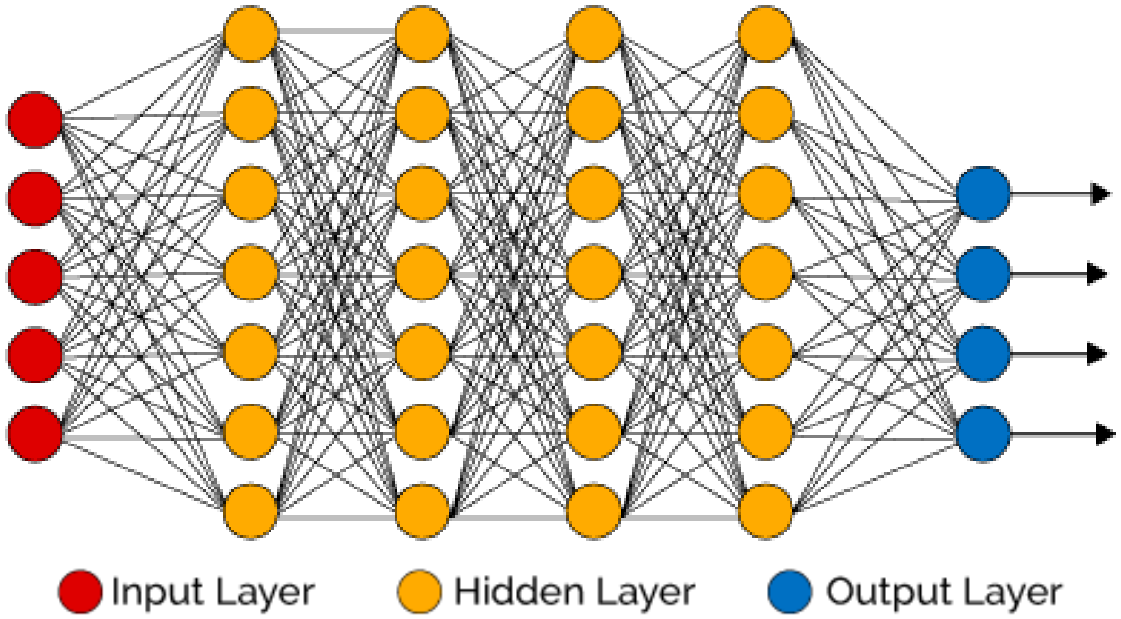
● GridSearhCV를 통해서 모델의 하이퍼파라미터 튜닝 결과

```
ml_models['LinearRegression'] = LinearRegression()

ml_models['DecisionTreeRegressor'] = DecisionTreeRegressor(max_depth=20,
                                                           random_state=RANDOM_STATE)

ml_models['RandomForestRegressor'] = RandomForestRegressor(n_estimators=1000,
                                                           max_depth=20,
                                                           random_state=RANDOM_STATE,
                                                           n_jobs=-1)

ml_models['XGBRegressor'] = XGBRegressor(n_estimators=1000,
                                         max_depth=20,
                                         learning_rate=0.001,
                                         random_state=RANDOM_STATE,
                                         n_jobs=-1)

ml_models['LGBMRegressor'] = LGBMRegressor(n_estimators=1000,
                                           max_depth=20,
                                           learning_rate=0.001,
                                           random_state=RANDOM_STATE,
                                           n_jobs=-1,
                                           boosting_type='gbdt',
                                           num_leaves=31,
                                           subsample_for_bin=200000,
                                           objective=None,
                                           class_weight=None,
                                           min_split_gain=0.0,
                                           min_child_weight=0.001,
                                           min_child_samples=20,
                                           subsample=1.0,
                                           subsample_freq=0,
                                           colsample_bytree=1.0,
                                           reg_alpha=0.0,
                                           reg_lambda=0.0)

ml_models['KNeighborsRegressor'] = KNeighborsRegressor(n_neighbors=5,
                                                       weights='uniform',
                                                       algorithm='auto',
                                                       leaf_size=30,
                                                       p=2,
                                                       metric='minkowski',
                                                       metric_params=None,
                                                       n_jobs=-1,)
```

```
ml_models['SVR'] = SVR(kernel='rbf',
                       degree=3,
                       gamma='scale',
                       coef0=0.0,
                       tol=0.001,
                       C=1.0,
                       epsilon=0.1,
                       shrinking=True,
                       cache_size=200,
                       verbose=False,
                       max_iter=-1,)

ml_models['AdaBoostRegressor'] = AdaBoostRegressor(base_estimator=RandomForestRegressor(random_state=RANDOM_STATE,
                                                                                        n_jobs=-1),
                                                   n_estimators=100,
                                                   learning_rate=0.1,
                                                   loss='linear',
                                                   random_state=RANDOM_STATE)

ml_models['GradientBoostingRegressor'] = GradientBoostingRegressor(loss='ls',
                                                                   learning_rate=0.1,
                                                                   n_estimators=100,
                                                                   subsample=0.1,
                                                                   criterion='friedman_mse',
                                                                   min_samples_split=2,
                                                                   min_samples_leaf=1,
                                                                   min_weight_fraction_leaf=0.0,
                                                                   max_depth=3,
                                                                   min_impurity_decrease=0.0,
                                                                   min_impurity_split=None,
                                                                   init=None,
                                                                   random_state=None,
                                                                   max_features=None,
                                                                   alpha=0.9,
                                                                   verbose=0,
                                                                   max_leaf_nodes=None,
                                                                   warm_start=False,)
```

딥러닝 모델 Deep Learning Model

- **딥러닝을 선택한 이유**
  - 데이터의 비선형성을 보다 효과적으로 파악
- **다층 완전 연결 신경망** Fully-Connected Layer Network
  - 활성화 함수: Leaky ReLU
- **손실함수** Loss Function
  - 평균 제곱 오차 MSE
- **최적화함수** Optimizer Function
  - Adam [*Diederik P et al.*, 2014]
- **학습률 스케쥴러**
  - 2만 에폭마다 0.1배로 감소



NS홈쇼핑

성능 평가
Testing

2020 빅콘테스트
2020 BIG CONTEST

NS홈쇼핑

예측 데이터

- 2020년 6월 1일 오전 6시부터 1달 매출 실적 예측

방송일자	머더코드	상품명	노출(분)	판매가	분당실적
2020-06-01 6:20	12345678	남성 티셔츠 세트	20	59,800	
2020-06-01 6:40	12345678	남성 티셔츠 세트	20	59,800	
2020-06-01 7:00	12345678	남성 티셔츠 세트	20	59,800	
2020-06-01 7:20	23456789	여성 란쥬쉐이퍼&팬티	20	69,900	
2020-06-01 7:40	23456789	여성 란쥬쉐이퍼&팬티	20	69,900	
2020-06-01 8:00	23456789	여성 란쥬쉐이퍼&팬티	20	69,900	
2020-06-01 8:20	34567890	여성 모자 3종	10	29,900	
2020-06-01 8:30	34567890	여성 모자 3종	10	29,900	
2020-06-01 8:40	34567890	여성 모자 3종	10	29,900	
2020-06-01 8:50	34567890	여성 모자 3종	10	29,900	

평가방법 Metric

- 평균 절대 비율 오차 (MAPE, Mean Absolute Percentage Error)
  - MAE나 MSE와 달리 크기 의존적 에러의 단점을 커버하기 위한 평가지표

$$MAPE = \frac{100\%}{n} \sum \left| \frac{A_t - F_t}{A_t} \right|$$

(A_t: 실제값, F_t: 예측값)

In [140]:

```
# MAPE
EPS = 1e-3
def MAPE(y_true, y_pred):
    y_true, y_pred = np.array(y_true), np.array(y_pred)
    return np.mean(np.abs((y_true - y_pred) / (y_true + EPS)))
```

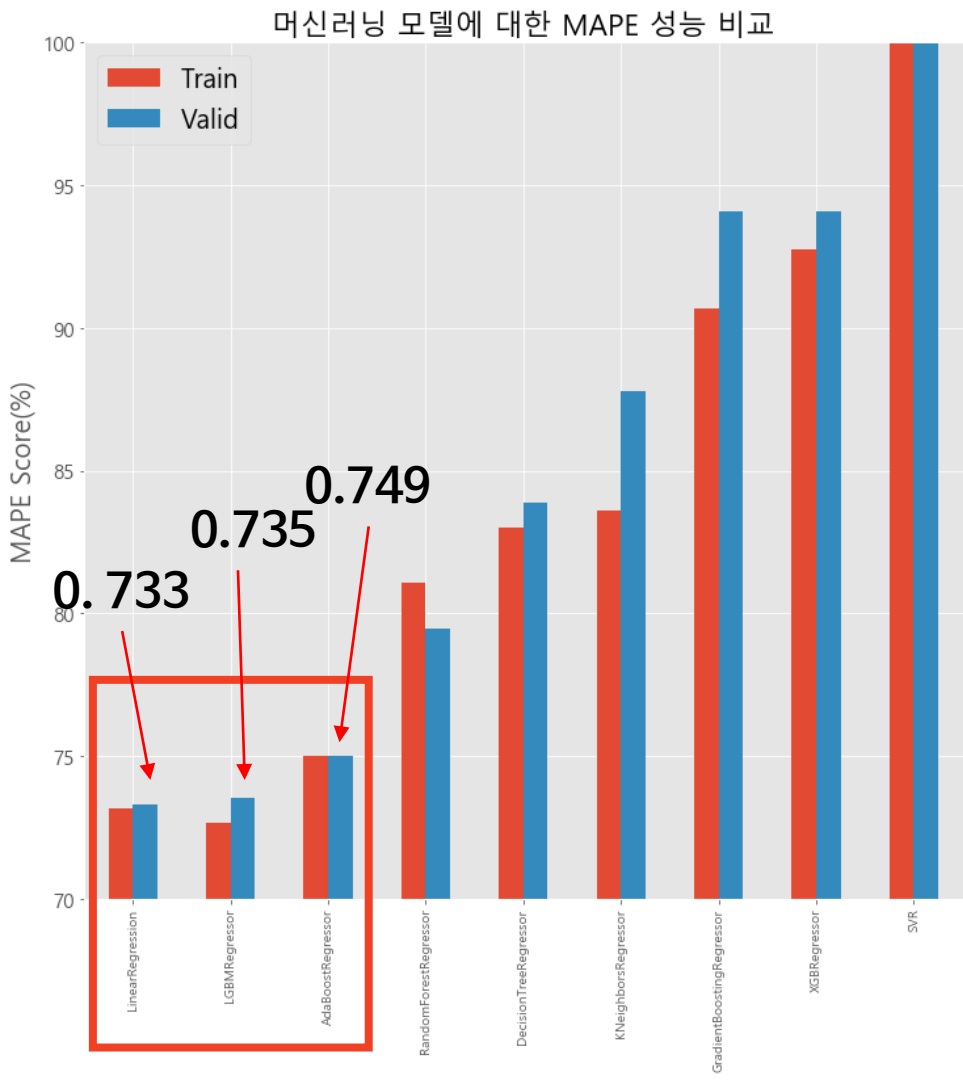
executed in 15ms, finished 15:39:27 2020-09-27

결과 Result

● 머신러닝 모델에 대한 교차검증 결과

- AdaBoostRegressor, Random ForestRegressor 순서로 평균 MAPE값이 낮은 것을 확인
- DecisionTreeRegressor는 상당히 Overfitting 된 것을 알 수 있음

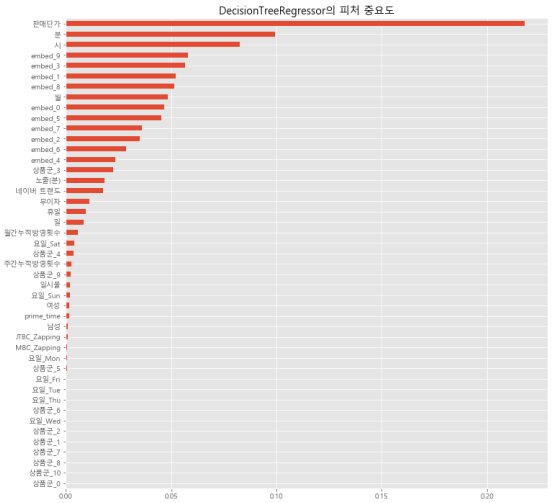
```
[1] LinearRegression          의 평균 Train MAPE 값: 0.7318039926 | [0.73545158 0.73393485 0.72745791 0.73131755 0.73085807]
[1] LinearRegression          의 Valid MAPE 값: 0.7331396711
[1] LinearRegression          모델 --> 저장 완료

[2] DecisionTreeRegressor     의 평균 Train MAPE 값: 0.8301266088 | [0.75949131 0.83254713 1.00532472 0.77844082 0.77482906]
[2] DecisionTreeRegressor     의 Valid MAPE 값: 0.8385661175
[2] DecisionTreeRegressor     모델 --> 저장 완료
------------------------------------------------------------
[3] RandomForestRegressor     의 평균 Train MAPE 값: 0.8105587503 | [0.8142241  0.81715455 0.80423051 0.81200006 0.80518452]
[3] RandomForestRegressor     의 Valid MAPE 값: 0.7948033064
[3] RandomForestRegressor     모델 --> 저장 완료
------------------------------------------------------------
[4] XGBRegressor              의 평균 Train MAPE 값: 0.9275779062 | [0.92648443 0.88516923 0.96054545 0.93801558 0.92767484]
[4] XGBRegressor              의 Valid MAPE 값: 0.9409796319
[4] XGBRegressor              모델 --> 저장 완료

[5] LGBMRegressor             의 평균 Train MAPE 값: 0.7265386846 | [0.72961279 0.73084394 0.72318444 0.72486103 0.72419121]
[5] LGBMRegressor             의 Valid MAPE 값: 0.7351041871
[5] LGBMRegressor             모델 --> 저장 완료

[6] KNeighborsRegressor       의 평균 Train MAPE 값: 0.8361924741 | [0.78715705 0.87533056 0.80119159 0.86129674 0.85598643]
[6] KNeighborsRegressor       의 Valid MAPE 값: 0.8776096271
[6] KNeighborsRegressor       모델 --> 저장 완료
------------------------------------------------------------
[7] SVR                       의 평균 Train MAPE 값: 3.7231471367 | [3.75349046 3.78558863 3.63969259 3.71144483 3.72551917]
[7] SVR                       의 Valid MAPE 값: 3.6706475529
[7] SVR                       모델 --> 저장 완료

[8] AdaBoostRegressor         의 평균 Train MAPE 값: 0.7499168790 | [0.74942683 0.7574487  0.74558176 0.75347935 0.74364775]
[8] AdaBoostRegressor         의 Valid MAPE 값: 0.7499369223
[8] AdaBoostRegressor         모델 --> 저장 완료

[9] GradientBoostingRegressor 의 평균 Train MAPE 값: 0.9066890942 | [0.88868854 0.96839212 0.9080427  0.90510936 0.86321275]
[9] GradientBoostingRegressor 의 Valid MAPE 값: 0.9406838051
[9] GradientBoostingRegressor 모델 --> 저장 완료
------------------------------------------------------------
```

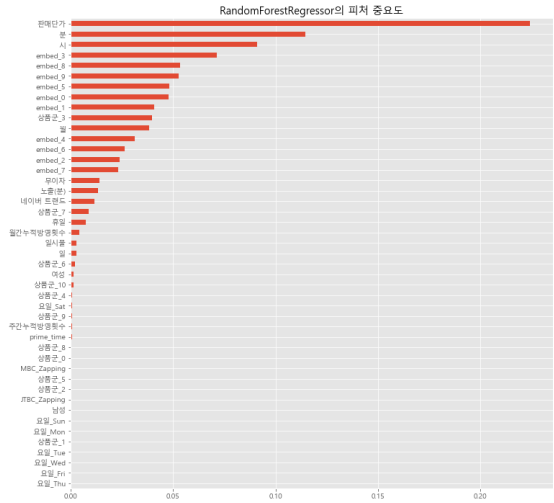


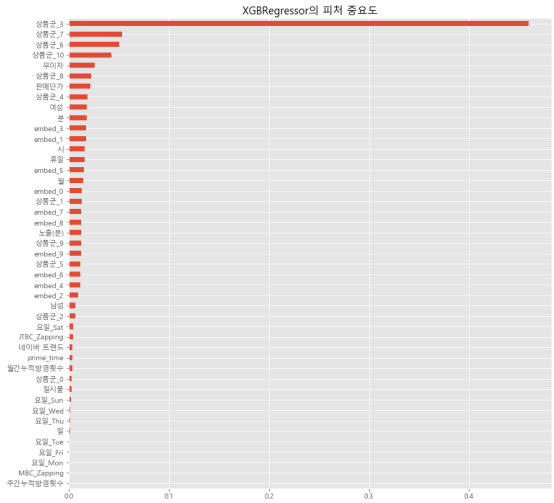
NS홈쇼핑

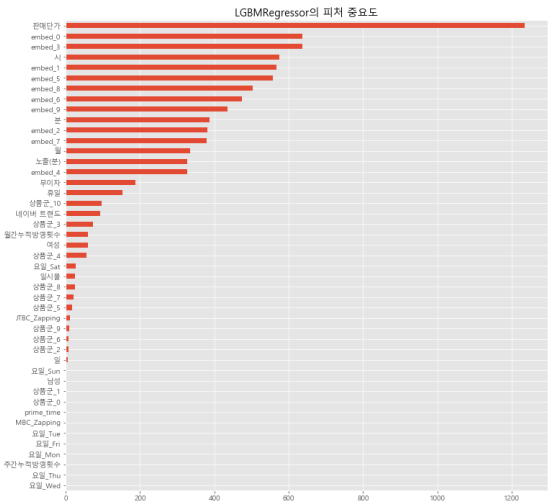
결과 Result

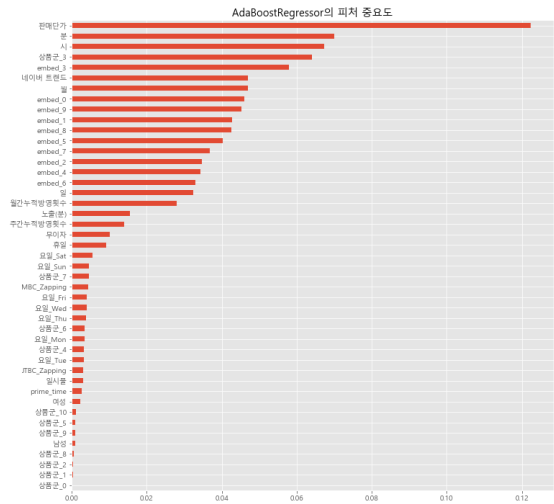
● 트리 계열의 모델들의 피처 중요도

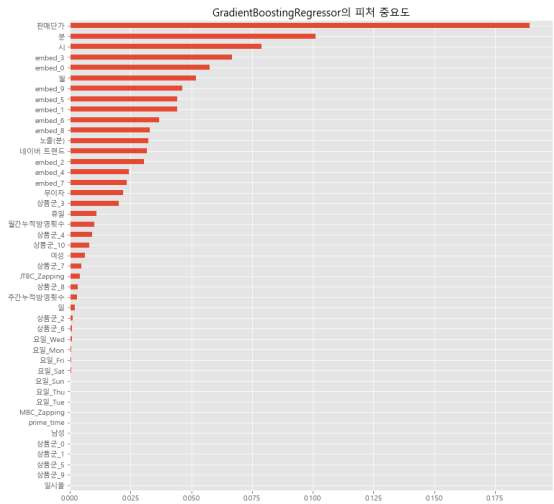










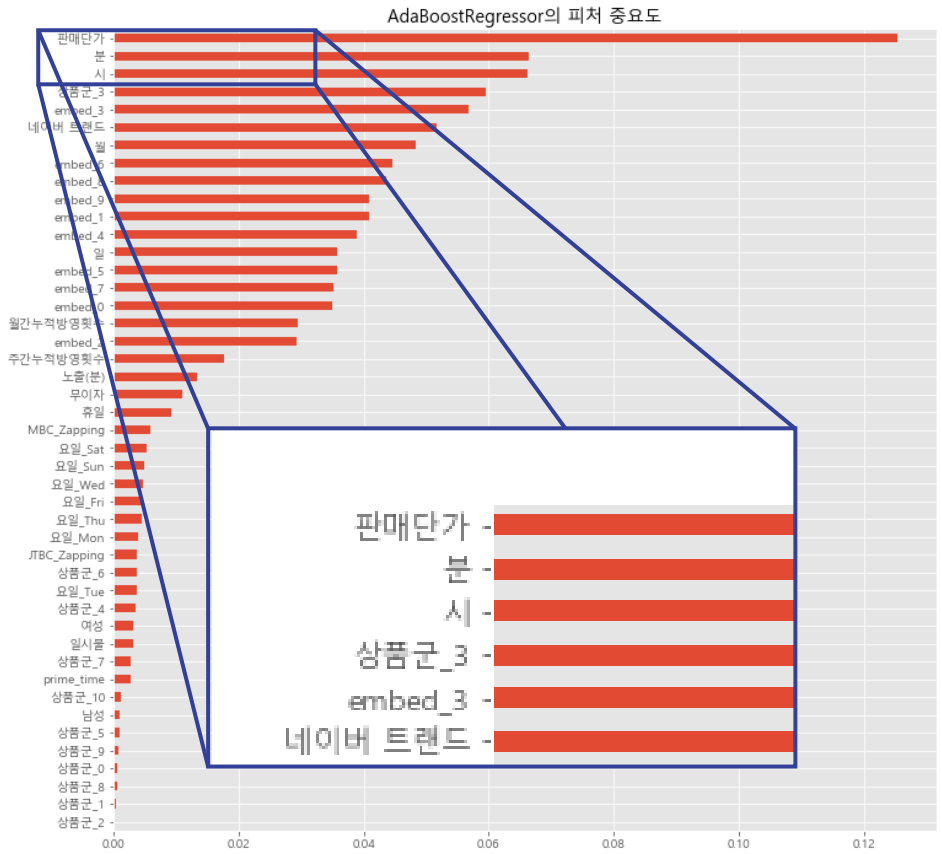


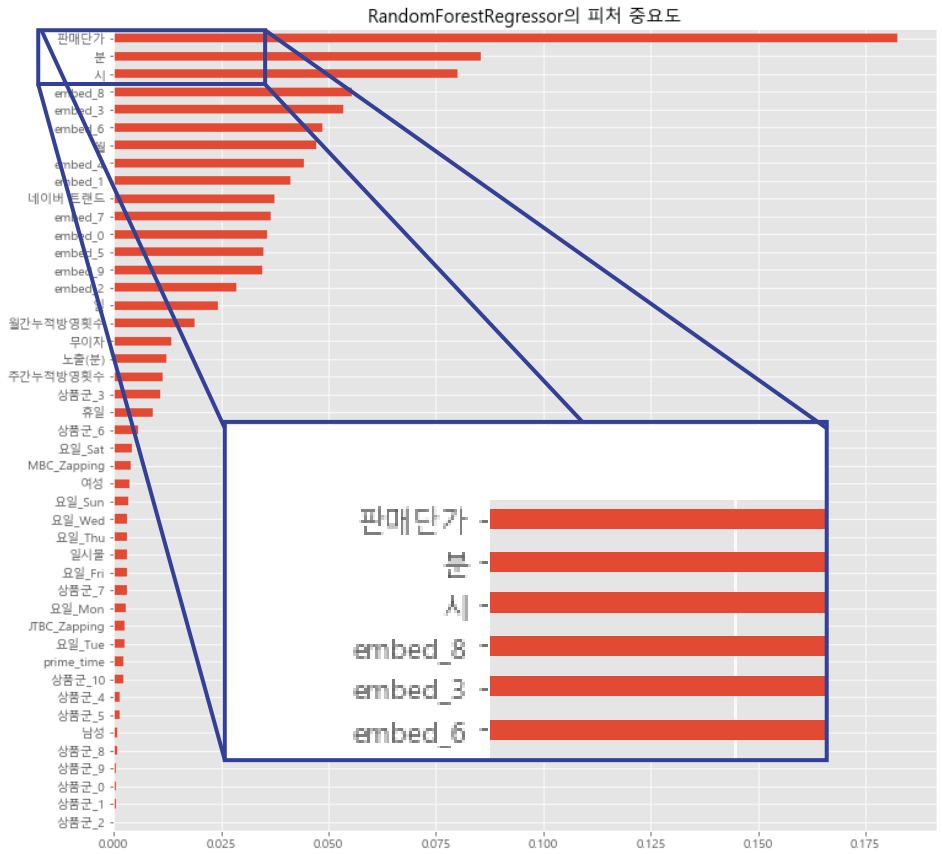
2020 빅콘테스트
2020 BIG CONTEST

결과 Result

● 트리 계열의 모델들의 피처 중요도

- **판매단가**와 **시간**과 상품명에 대한 **임베딩 벡터**를 중요하다고 판단

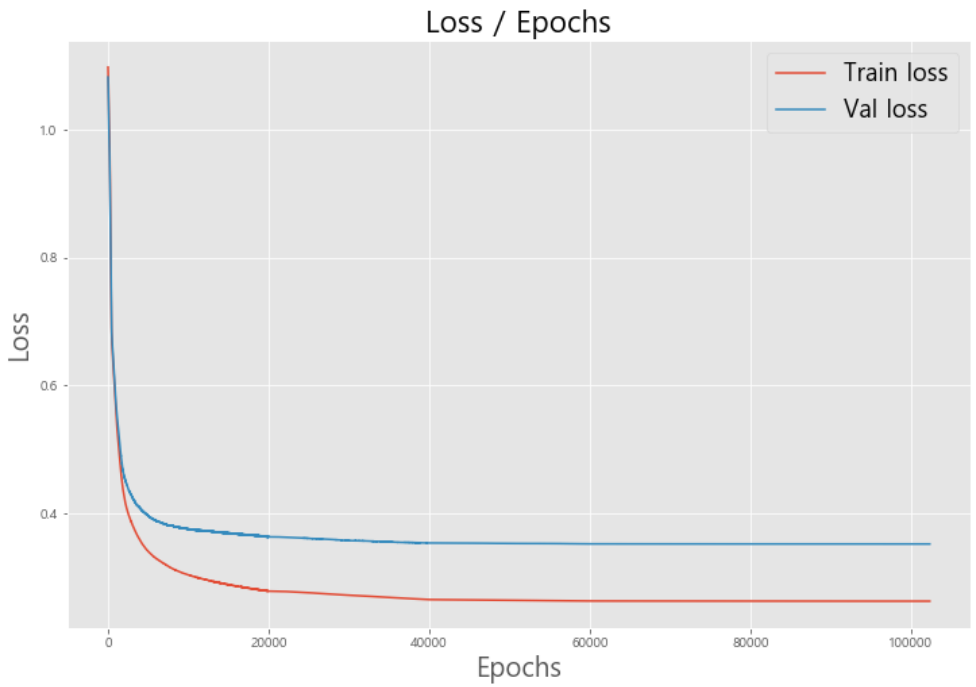


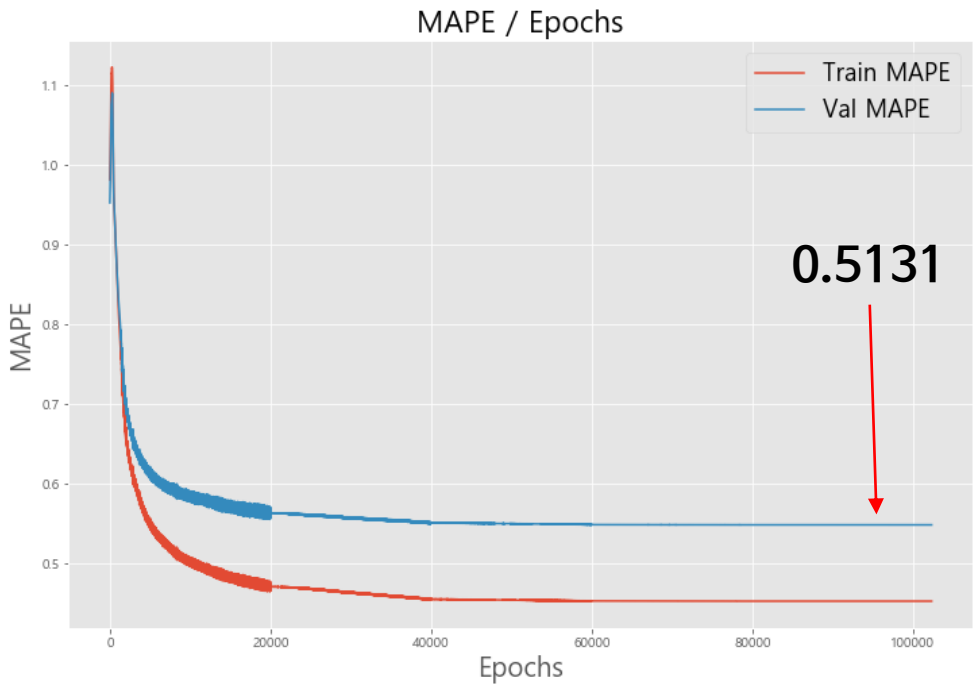


결과 Result

딥러닝 모델 성능 결과

- Epoch: 100,000
- MAPE: 0.51





결과 Result

• 딥러닝 모델 성능 결과

방송일시	노출(분)	마더코드	상품코드	상품명	상품군	판매단가	취급액
2020-06-01 6:20	20	100650	201971	잭필드 남성 반팔셔츠 4종	의류	59800	35764806
2020-06-01 6:40	20	100650	201971	잭필드 남성 반팔셔츠 4종	의류	59800	42647963
2020-06-01 7:00	20	100650	201971	잭필드 남성 반팔셔츠 4종	의류	59800	25745363
2020-06-01 7:20	20	100445	202278	쿠미투니카 쿨 레이시 란쥬쉐이퍼&팬티	속옷	69900	9748303
2020-06-01 7:40	20	100445	202278	쿠미투니카 쿨 레이시 란쥬쉐이퍼&팬티	속옷	69900	12330183
2020-06-01 8:00	20	100445	202278	쿠미투니카 쿨 레이시 란쥬쉐이퍼&팬티	속옷	69900	5540869
2020-06-01 8:20	20	100381	201247	바비리스 퍼펙트 볼륨스타일러	이미용	59000	5527927
2020-06-01 8:40	20	100381	201247	바비리스 퍼펙트 볼륨스타일러	이미용	59000	9026042
2020-06-01 9:00	20	100381	201247	바비리스 퍼펙트 볼륨스타일러	이미용	59000	3186639
2020-06-01 9:20	20	100638	201956	램프쿡 자동회전냄비	주방	109000	31417572
2020-06-01 9:40	20	100638	201956	램프쿡 자동회전냄비	주방	109000	44901996
2020-06-01 10:00	20	100638	201956	램프쿡 자동회전냄비	주방	109000	17039123
2020-06-01 10:20	20	100348	201091	벨레즈온 심리스 원피스 4종 패키지	속옷	59900	26778431
2020-06-01 10:40	20	100348	201091	벨레즈온 심리스 원피스 4종 패키지	속옷	59900	41225848
2020-06-01 11:00	20	100348	201091	벨레즈온 심리스 원피스 4종 패키지	속옷	59900	17432840
2020-06-01 11:20	20	100012	200016	AAC 삼채포기김치 10kg	농수축	40900	33431859
2020-06-01 11:40	20	100012	200016	AAC 삼채포기김치 10kg	농수축	40900	46906843
2020-06-01 12:00	20	100012	200016	AAC 삼채포기김치 10kg	농수축	40900	24293520
2020-06-01 12:20	20	100080	200217	아키 라이크라 릴렉스 보정브라 패키지(뉴아키28차)	속옷	99900	30771537
2020-06-01 12:40	20	100080	200217	아키 라이크라 릴렉스 보정브라 패키지(뉴아키28차)	속옷	99900	39104044
2020-06-01 13:00	20	100080	200217	아키 라이크라 릴렉스 보정브라 패키지(뉴아키28차)	속옷	99900	17847631
2020-06-01 13:20	20	100570	201673	KT휴대폰_삼성갤럭시 노트10	무형	0	0
2020-06-01 13:20	0	100570	201671	(특)KT휴대폰_삼성갤럭시 A31	무형	0	0
2020-06-01 13:40	20	100570	201673	KT휴대폰_삼성갤럭시 노트10	무형	0	0
2020-06-01 13:40	0	100570	201671	(특)KT휴대폰_삼성갤럭시 A31	무형	0	0
2020-06-01 14:00	20	100570	201673	KT휴대폰_삼성갤럭시 노트10	무형	0	0
2020-06-01 14:00	0	100570	201671	(특)KT휴대폰_삼성갤럭시 A31	무형	0	0
2020-06-01 14:20	60	100554	201641	DB손해보험 참좋은운전자보험(1912)	무형	0	0
2020-06-01 15:20	20	100362	201150	에이유플러스 슈퍼선스틱 1004(최저가)	이미용	39900	23456266

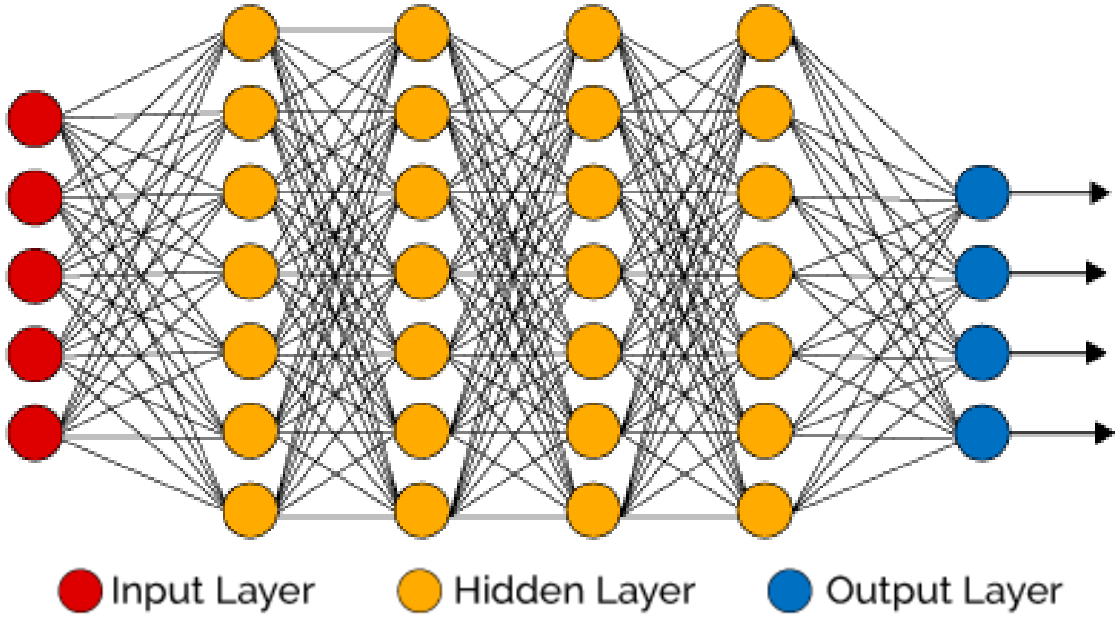
NS홈쇼핑

결론 및 토론

Conclusion & Discussion

결론 Conclusion

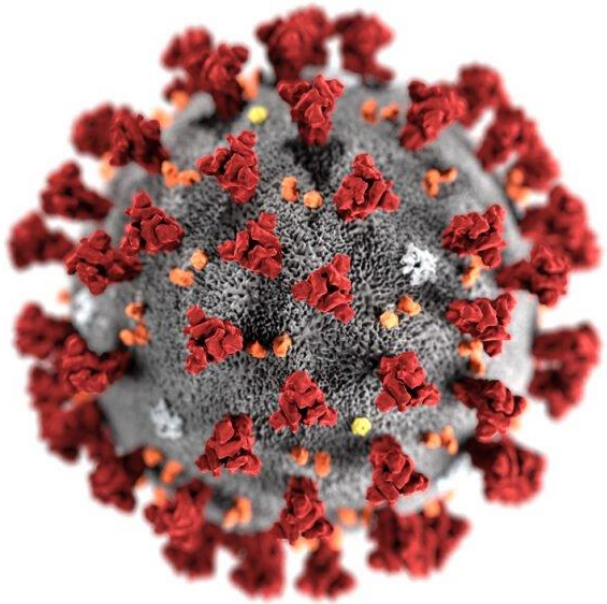
- 모델 성능 Performance
  - 머신러닝 모델보다 딥러닝 모델의 성능이 더 우수
  - 머신러닝 모델 중에서도 랜덤 포레스트 모델이 가장 우수
  - 딥러닝 모델에서는 활성화 함수를 Leaky ReLU로 설정하고 레이어를 깊게 쌓지 않았을 때 성능이 더 좋았음



결론 Conclusion

한계점 Limitation

- 주어진 데이터의 피처가 풍부했다면 보다 정확한 취급액 예측이 가능
  - 쇼호스트 정보, 제작 PD명, 해당 상품에 대한 소비자 정보 등
- 소비자 물가 지수 Custom Price Index
  - 포스트 코로나 시대의 시장 상황 반영한 취급액 예측



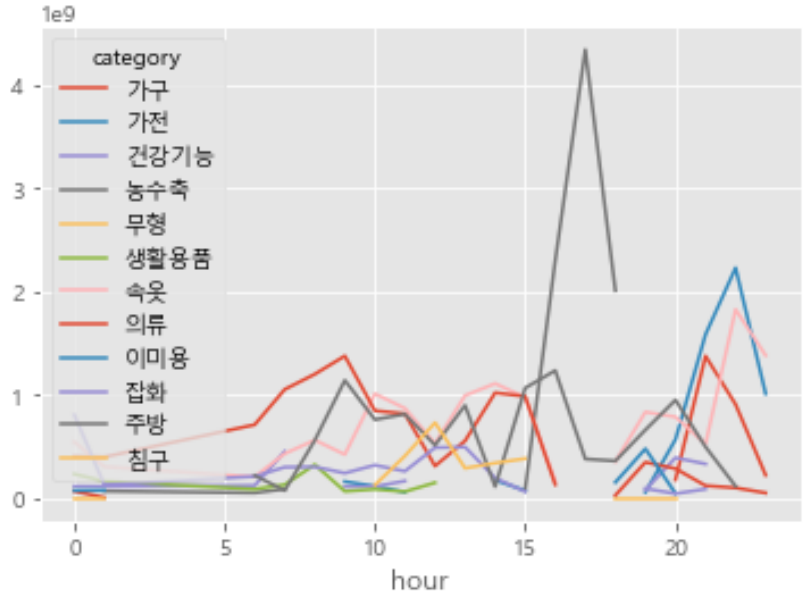
NS홈쇼핑

방송 편성 최적화 방안

Broadcasting Schedule Optimization

방송 편성 최적화 모델

- 상품군별 프라임 시간 활용
  - 상품군 별 최적화 방법
  - 과거 데이터에서 일별, 시간별 매출이 최대인 상품군으로 사전을 생성
  - 최대 3순위까지의 상품군을 선정 후 시간표 생성
  - Ex) prime_dict = {…, '21': [가전, 의류, 주방],
    '22': [가전, 속옷, 의류], …}
- 한계점
  - 상품군 내 상품별 최적화는 힘듦

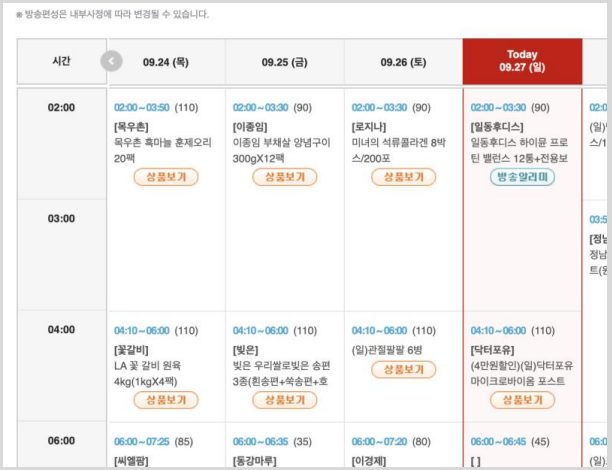


NS홈쇼핑

방송 편성 최적화 모델

- 상품 관련 피처(외부 데이터 포함)만으로 최적 방송 시간표 생성
- 상품 별 최대 수익을 고려한 최대 수익 편성표 생성

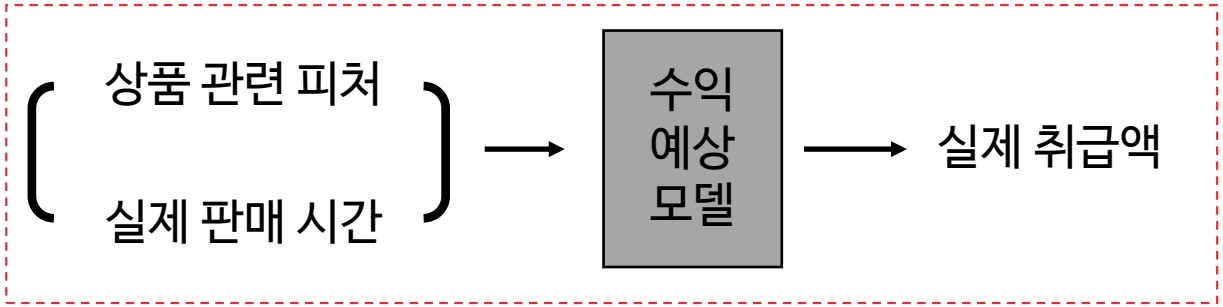
상품 관련 피처 → 시간 생성 모델 →



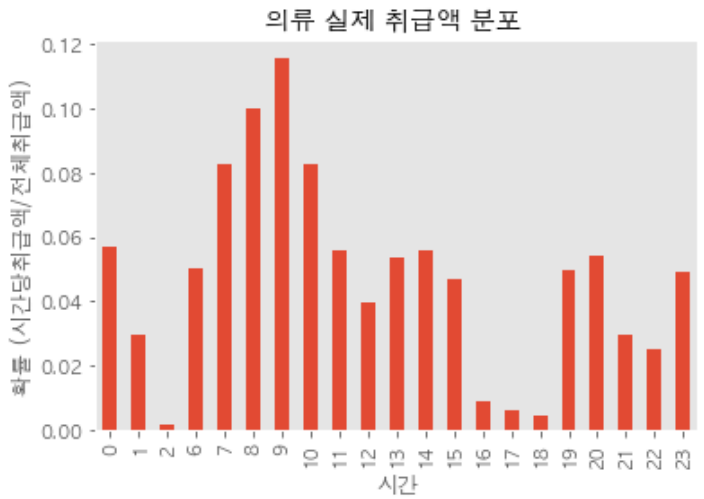
최대 수익 반영 시간표 생성

방송 편성 최적화 모델

- 앞서 만든 *수익 예측 모델* 활용하여 최대 수익 편성표 생성
  - 실제 취급액 ⇔ 실제 취급액 분포



[Process 1]

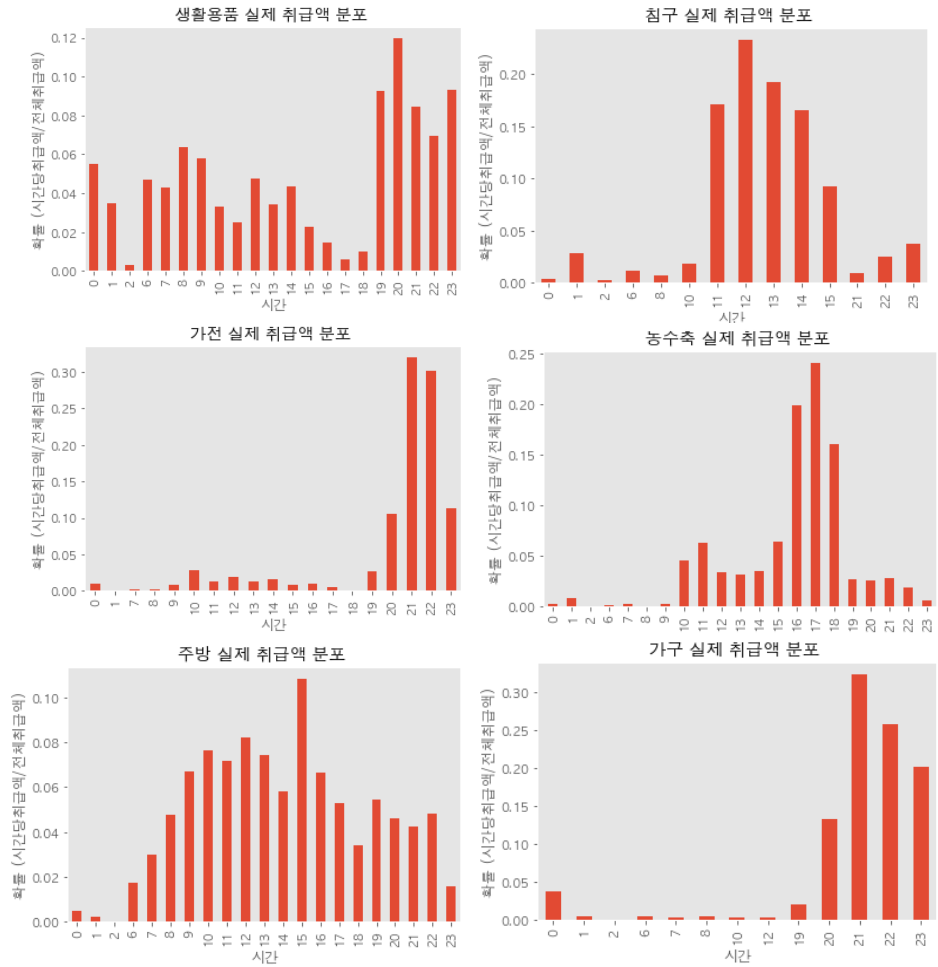


실제 취급액 분포

NS홈쇼핑

방송 편성 최적화 모델

● 실제 취급액 분포



각 상품 별 프라임 타임 분포를 따르는 생성 모델 만들기

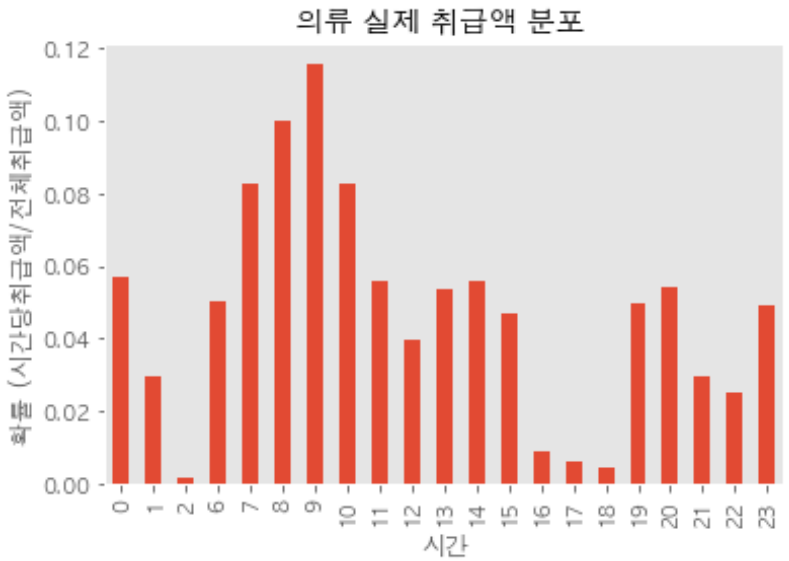
※ 방송편성은 내부사정에 따라 변경될 수 있습니다.

시간	09.24 (목)	09.25 (금)	09.26 (토)	Today 09.27 (일)
02:00	02:00~03:50 (110) [목우촌] 목우촌 흑마늘 훈제오리 20팩 상품보기	02:00~03:30 (90) [이종임] 이종임 부채살 양념구이 300gX12팩 상품보기	02:00~03:30 (90) [로지나] 미녀의 석류콜라겐 8박스/200포 상품보기	02:00~03:30 (90) [일동후디스] 일동후디스 하이뮨 프로틴 밸런스 12통+전용보 방송알리미
03:00				
04:00	04:10~06:00 (110) [꽃갈비] LA 꽃 갈비 원육 4kg(1kgX4팩) 상품보기	04:10~06:00 (110) [빛은] 빛은 우리쌀로빛은 송편 3종(흰송편+쑥송편+호 상품보기	04:10~06:00 (110) (일)관절팔팔 6병 상품보기	04:10~06:00 (110) [닥터포유] (4만원할인)(일)닥터포유 마이크로바이옴 포스트 상품보기
06:00	06:00~07:25 (85) [씨엘팜]	06:00~06:35 (35) [동강마루]	06:00~07:20 (80) [이경제]	06:00~06:45 (45) [ ]

최대 수익 반영 시간표 생성

방송 편성 최적화 모델

- 실제 취급액 분포와 추천 시간에 대한 취급액 분포가 같게 학습
  - 쿨백-라이블러 발산(Kullback-Leibler divergence)
    - $KL(p||g) = -\sum p_i \log(p_i) - (-\sum p_i \log q_i) = -\sum p_i \log(\frac{q_i}{P_i})$



동일한 분포

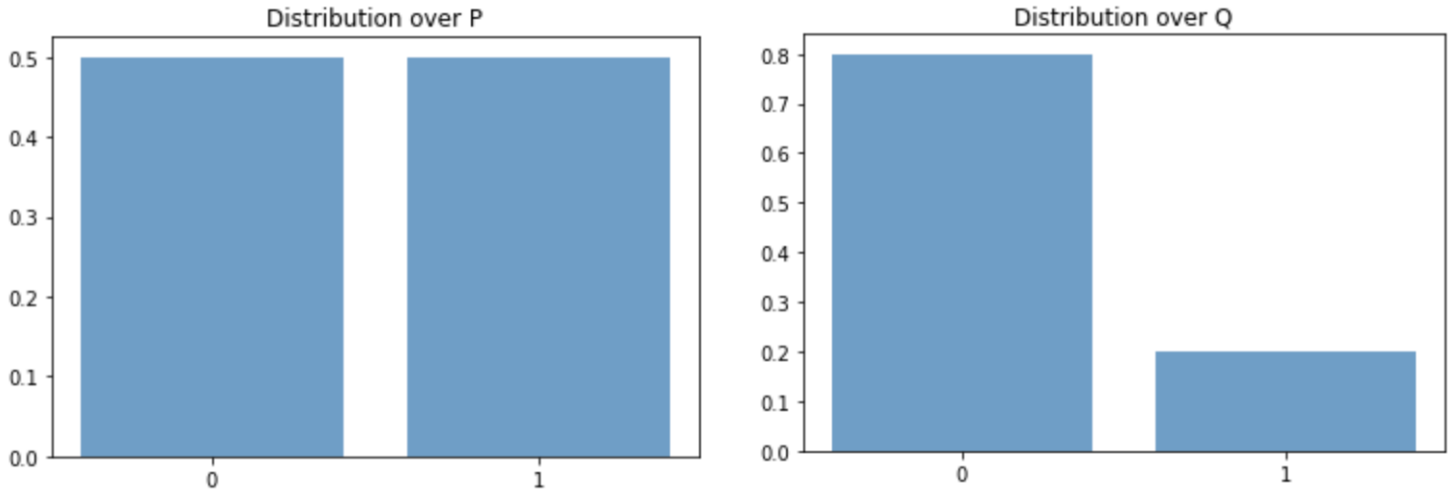
학습

?

추천 시간에 대한
예상 수익 분포

방송 편성 최적화 모델

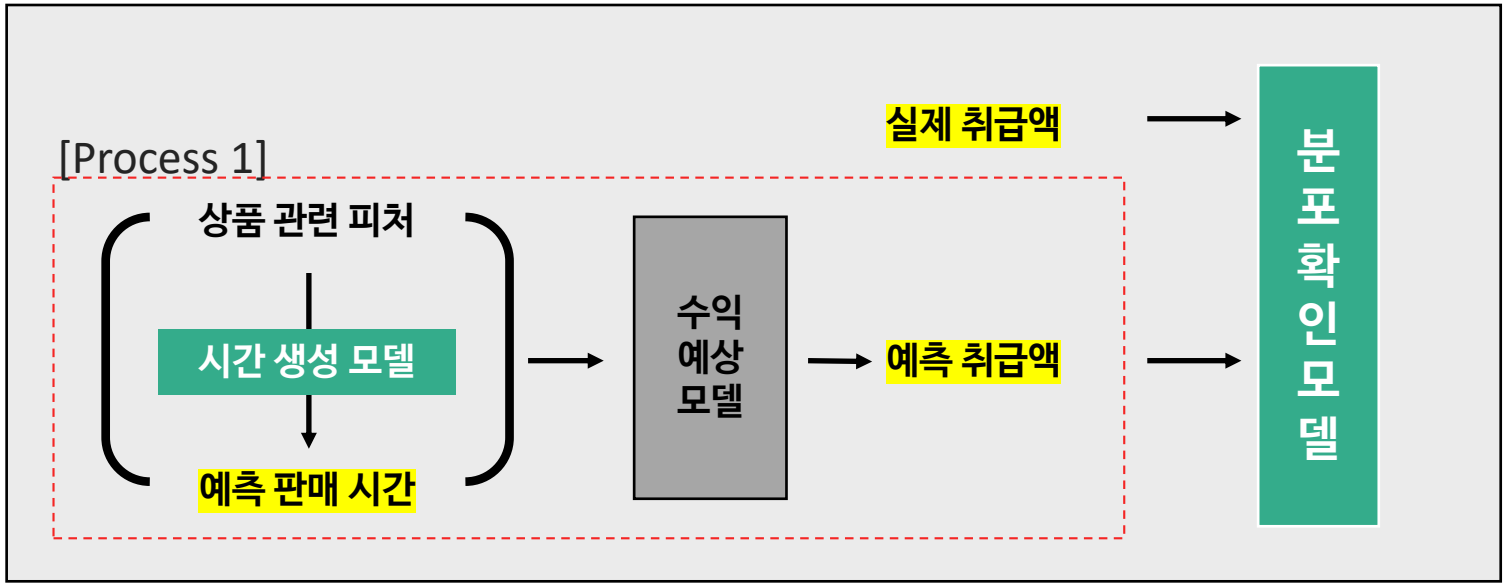
- **쿨백-라이블러 발산**(Kullback - Leibler divergence)
  - 두 확률분포의 차이를 계산하는 데에 사용하는 함수
  - 확률공간 Ω와 이산확률변수 $X = x_1, x_2 \ \ldots x_n$, 그리고 확률 P, Q가 주어졌을, 두 확률 분포의 차이를 계산
  - Ex)



$$KL(p||g) = -(p_1 \log(q_1) + p_2 \log(q_2))$$
$$= -(0.5 \log(0.8) + 0.5\log(0.2) = -(-0.916290)$$

방송 편성 최적화 모델

- ***시간 생성 모델*** 과 ***분포 확인 모델*** 로 구성
  - **시간 생성 모델**
    - 입력값은 상품관련 피처이며 출력은 생성 판매 시간
    - 앞서 만든 수익 예상 모델을 사용하기 위해서는 시간 생성 모델 필수
  - **분포 확인 모델**
    - 실제 취급액과 예측 취급액 사이에 분포를 확인하는 모델



NS홈쇼핑

질의응답

QnA

2020 빅콘테스트
2020 BIG CONTEST

참고 자료 Reference

https://github.com/KUAI-Bigcontest/debug

참고 자료 Reference

- https://www.bigcontest.or.kr/
- https://www.mk.co.kr/news/business/view/2018/09/576315/
- https://www.etoday.co.kr/news/view/1882077
- http://www.citydaily.kr/news/articleView.html?idxno=831
- http://plus.hankyung.com/apps/newsinside.view?aid=202003096861A
- https://ko.wikipedia.org/wiki/%ED%99%A9%EA%B8%88%EC%8B%9C%EA%B0%84%EB%8C%80

NS홈쇼핑

감사합니다

The End

2020 빅콘테스트
2020 BIG CONTEST