데이터분석 세션 발표

ESC 3조

김진영 정재은 조민주 오다건 강동인

2020.8.22

Outline

1. Target Variable
2. Input Variables
   - 내부변수
   - 외부변수
3. Categorical Analysis
4. Rating Data
5. Modeling
   - 변수 삭제 및 생성
   - Base model
6. 추후 계획

References

1. Target Variable

1. 2019년 실적데이터 에서 취급액 50,000원으로 판매단가가 취급액보다 더 큰 상품의 데이터 값의 의미?
- 2019년 실적데이터에서 취급액이 50,000원인 데이터는 데이터 정제과정에서 발생된 오류값으로.
  취급액이 50,000원에 해당되는 상품에 대한 취급액은 0원으로 변경(주문량 0인 값)
※ 해당내용을 반영한 데이터는 추후 수정하여 업데이트 예정

취급액이 50,000원인 데이터는 주문량 0 [1] → 제거하자.

표 1. 취급액이 50,000인 데이터.

	broadDateTime	broadTime	motherCode	prodCode	prodName	prodGroup	unitPrice	revenue
144	2019-01-02 22:00:00	NaN	100148	200432	무이자 LG 울트라HD TV 55UK6800HNC	가전	1440000	50000.0
147	2019-01-02 22:00:00	NaN	100148	200518	일시불 LG 울트라HD TV 70UK7400KNA	가전	2700000	50000.0
148	2019-01-02 22:00:00	NaN	100148	200451	무이자 LG 울트라HD TV 70UK7400KNA	가전	2990000	50000.0
153	2019-01-02 22:20:00	NaN	100148	200518	일시불 LG 울트라HD TV 70UK7400KNA	가전	2700000	50000.0
154	2019-01-02 22:20:00	NaN	100148	200451	무이자 LG 울트라HD TV 70UK7400KNA	가전	2990000	50000.0
...	...	...	...	...	...	...	...	...
37709	2019-12-25 10:20:00	NaN	100036	200070	구찌 인터로킹 GG 탑핸들 체인 숄더 스몰	잡화	2590000	50000.0
37967	2019-12-28 10:20:00	NaN	100036	200070	구찌 인터로킹 GG 탑핸들 체인 숄더 스몰	잡화	2590000	50000.0
37969	2019-12-28 10:20:00	NaN	100039	200073	버버리 홀스페리 페이톤 크로스백	잡화	880000	50000.0
38025	2019-12-28 21:20:00	NaN	100372	201169	(싱글+싱글)일월 품안애 온수매트	생활용품	198000	50000.0
38123	2019-12-29 23:20:00	NaN	100182	200612	무이자 선일금고 이볼브 시리즈 EV-020	생활용품	440000	50000.0

1993 rows × 8 columns

```
len(data[data["revenue"] == 50000])
```

```
1993
```

```
data = data[data["revenue"] != 50000]
data.shape
```

```
(36316, 8)
```

```
data = data[~data["revenue"].isna()]
data.shape
```

```
(35379, 8)
```

제거 완료

1. Target Variable

표 2. Examples of training data.

	broadDateTime	broadTime	motherCode	prodCode	prodName	prodGroup	unitPrice	revenue
0	2019-01-01 06:00:00	20.0	100346	201072	테이트 남성 셀린니트3종	의류	39900	2099000.0
1	2019-01-01 06:00:00	NaN	100346	201079	테이트 여성 셀린니트3종	의류	39900	4371000.0
2	2019-01-01 06:20:00	20.0	100346	201072	테이트 남성 셀린니트3종	의류	39900	3262000.0
3	2019-01-01 06:20:00	NaN	100346	201079	테이트 여성 셀린니트3종	의류	39900	6955000.0
4	2019-01-01 06:40:00	20.0	100346	201072	테이트 남성 셀린니트3종	의류	39900	6672000.0

```
data['revenue']/data['unitPrice']
```

```
0          52.606516
1         109.548872
2          81.754386
3         174.310777
4         167.218045
              ...
38299      68.628378
38300     286.117978
38301     621.380952
38302      87.120253
38303     314.918919
Length: 35379, dtype: float64
```

정수가 나오지 않는다.

2. 취급액 = 판매단가 X 주문량 에서 주문량이 소수점으로 나오는 이유는?
- 일반적으로 취급액의 수식을 적용하면 일반적으로 판매단가X주문량이 맞습니다.
실제 현업에서 실적을 집계할때 고객이 실제 주문한 금액을 합산해서 보고있기 때문에 수식을 적용한것과는 차이가 있을 수 있습니다.
고객이 상품을 구매할때 판매가를 그대로 지불하지 않고 할인쿠폰 적용, ARS할인, 일시불할인, 카드사할인 등 여러가지 경로로 할인된 금액을 지불하고있습니다.
(단, 결제시 적립금, 상품권 등을 사용하여 실결제금액이 바뀌는 경우는 해당사항 없음)
이에 주문량이 소수점으로 발생할 수 있음을 참고하여 주시기 바랍니다.

할인으로 인한 것이니,
올림하여 정수로 사용하자. [1]

- [x] 2
- [x] 3
- [x] 5
- [x] 7
- [x] 9
- [x] 10
- [x] 11
- [x] 12
- [x] 13
- [x] 14
- [x] 15
- [x] 16
- [x] 17
- [x] 18
- [x] 19
- [x] 20
- [x] 22
- [x] 23
- [x] 25
- [x] 26
- [x] 27
- [x] 30
- [x] 40
- [x] 60

판매량, 그대로 사용해도 괜찮을까?
→ 노출시간으로 나눠서 분당 주문량을 사용하고, test data에서 역시 분당 판매량을 예측한 후 판매량을 구하자.

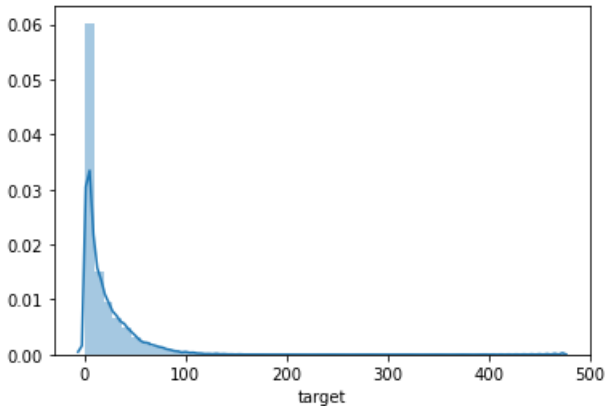

그림 1. 분당 주문량.

Log transformation

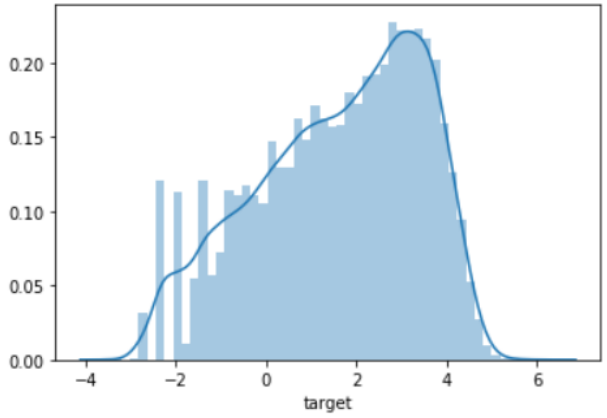

그림 2. Transformed target.

2. Input Variables - 내부변수

표 3. broadTime이 NaN인 데이터.

	broadDateTime	broadTime	motherCode	prodCode	prodName	prodGroup	unitPrice	revenue
1	2019-01-01 06:00:00	NaN	100346	201079	테이트 여성 셀린니트3종	의류	39900	4371000.0
3	2019-01-01 06:20:00	NaN	100346	201079	테이트 여성 셀린니트3종	의류	39900	6955000.0
5	2019-01-01 06:40:00	NaN	100346	201079	테이트 여성 셀린니트3종	의류	39900	9337000.0
26	2019-01-01 14:00:00	NaN	100377	201226	그렉노먼 여성 구스다운 롱 벤치코트	의류	119000	20841000.0
28	2019-01-01 14:30:00	NaN	100377	201226	그렉노먼 여성 구스다운 롱 벤치코트	의류	119000	47294000.0
...	...	...	...	...	...	...	...	...
38298	2019-12-31 23:40:00	NaN	100448	201384	무이자쿠첸압력밥솥 6인용	주방	158000	2328000.0
38299	2019-12-31 23:40:00	NaN	100448	201391	일시불쿠첸압력밥솥 6인용	주방	148000	10157000.0
38301	2020-01-01 00:00:00	NaN	100448	201390	일시불쿠첸압력밥솥 10인용	주방	168000	104392000.0
38302	2020-01-01 00:00:00	NaN	100448	201384	무이자쿠첸압력밥솥 6인용	주방	158000	13765000.0
38303	2020-01-01 00:00:00	NaN	100448	201391	일시불쿠첸압력밥솥 6인용	주방	148000	46608000.0

14976 rows × 8 columns

```
data.isna().sum()
```

```
broadDateTime      0
broadTime      14976
motherCode         0
prodCode           0
prodName           0
prodGroup          0
unitPrice          0
revenue            0
dtype: int64
```

동일한 제품이 여성용, 남성용/ 무이자, 일시불 등으로 구분되어서 NaN이 생김.
→ 같은 시간에 방영한 제품에 대해서는 같은 값으로 채워주자!


```
data[data['broadTime'].isna()]
```

broadDateTime	broadTime	motherCode	prodCode	prodName	prodGroup	unitPrice	revenue

```
data.isna().sum()
```

```
broadDateTime      0
broadTime          0
motherCode         0
prodCode           0
prodName           0
prodGroup          0
unitPrice          0
revenue            0
dtype: int64
```

2. Input Variables - 내부변수

요일별 추세 파악

broadDateTime
2019-01-01 06:00:00
2019-01-01 06:00:00
2019-01-01 06:20:00
2019-01-01 06:20:00
2019-01-01 06:40:00

→

broadYear	broadMonth	broadDay	broadHour	broadMin	broadDayOfWeek	isWeekend
2019	1	1	6	0	1	0
2019	1	1	6	0	1	0
2019	1	1	6	20	1	0
2019	1	1	6	20	1	0
2019	1	1	6	40	1	0

→

Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
1	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0

그림 3. Training data 시간 변수 파싱.

표 4. 성별 변수.

prodName	isFemale	isMale
테이트 남성 셀린니트3종	0	1
테이트 여성 셀린니트3종	1	0
테이트 남성 셀린니트3종	0	1
테이트 여성 셀린니트3종	1	0
테이트 남성 셀린니트3종	0	1
테이트 여성 셀린니트3종	1	0
오모떼 레이스 파운데이션 브라	0	0
오모떼 레이스 파운데이션 브라	0	0
오모떼 레이스 파운데이션 브라	0	0

표 5. 할부 변수.

prodName	paymentPlan
무이자 LG 통돌이 세탁기	1
무이자 LG 통돌이 세탁기	1
무이자 LG 통돌이 세탁기	1
무이자 쿠첸 풀스텐 압력밥솥 10인용(A1)	1
무이자 쿠첸 풀스텐 압력밥솥 6인용(A1)	1

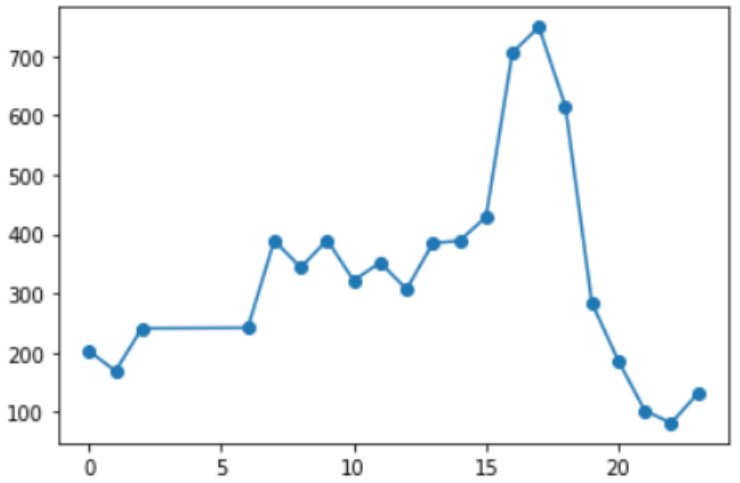

그림 4. 시간대별 판매량 평균.

→ 황금 시간대 반영하는 변수 추가

2. Input Variables - 외부변수

1. 생활물가지수 [2]

- 일상생활에서 소비자들이 자주 구입하는 물품과 기본 생필품을 대상으로 작성된 소비자물가지수의 보조지표
- 기준 (100): 2015년 01월

표 6. Examples of price index data.

	시도별	시점	총지수	생활물가지수	식품	식품 이외	전월세	생활물가 이외	전·월세포함 생활물가지수
0	전국	2019. 01	104.24	104.03	108.79	101.49	104.19	104.53	104.05
1	전국	2019. 02	104.69	104.61	109.33	102.09	104.18	104.91	104.55
2	전국	2019. 03	104.49	104.45	108.76	102.16	104.14	104.58	104.40
3	전국	2019. 04	104.87	104.81	109.29	102.43	104.10	105.10	104.70
4	전국	2019. 05	105.05	105.29	109.18	103.22	104.07	104.93	105.10

- Training data의 상품군: 가구, 가전, 건강기능 등 11가지
- 식품군과 비식품군 두가지로 분류 후 생활물가지수 사용
- 시도별 구분없이 전국 생활물가지수를 사용

표 7. Examples of processed data.

broadYear	broadMonth	식품	식품 이외
2019	1	108.79	101.49
2019	2	109.33	102.09
2019	3	108.76	102.16
2019	4	109.29	102.43
2019	5	109.18	103.22

그림 5. 소비자 물가지수 사용 예시. [3]

표 8. Examples of processed data.

broadYear	broadMonth	식품	식품 이외	foodIndex	nonfoodIndex
2019	1	108.79	101.49	0.34	-0.63
2019	2	109.33	102.09	0.54	0.60
2019	3	108.76	102.16	-0.57	0.07
2019	4	109.29	102.43	0.53	0.27
2019	5	109.18	103.22	-0.11	0.79

2. Input Variables - 외부변수

2. 전국기온 [4]

- 시도별 구분없이 전국 평균 기온을 사용
- 월별 Min-Max normalization

표 9. Examples of temp. data.

	날짜	지점	평균기온(℃)	최저기온(℃)	최고기온(℃)
0	2019-01-01	전국	-2.1	-5.8	2.1
1	2019-01-02	전국	-2.5	-7.0	3.2
2	2019-01-03	전국	-2.1	-7.7	5.0
3	2019-01-04	전국	-0.7	-7.3	5.3
4	2019-01-05	전국	0.2	-4.6	5.3
...	...	...	...	...	...
542	2020-06-26	전국	22.9	19.4	27.6
543	2020-06-27	전국	23.6	19.4	29.1
544	2020-06-28	전국	23.8	19.3	29.2
545	2020-06-29	전국	21.6	18.7	25.5
546	2020-06-30	전국	20.8	18.5	23.8

→

	tempNorm
0	0.164179
1	0.164179
2	0.164179
3	0.164179
4	0.164179
...	...
35374	0.000000
35375	0.343284
35376	0.343284
35377	0.343284
35378	0.343284

3. 강수량 [5]

- 전국 평균 강수량과 수도권 평균 강수량 두가지 변수 추가
- Min-Max normalization

표 10. Examples of rain data.

지역명	일시	평균일강수량(mm)	최다일강수량(mm)	최다강수량지점	1시간최다강수량(mm)
전국	2020-08-11	32.0	106.6	인천	34.5
전국	2020-08-10	31.0	114.9	순창군	56.2
전국	2020-08-09	23.8	149.5	철원	39.2
전국	2020-08-08	60.5	361.3	순창군	82
전국	2020-08-07	43.8	259.5	광주	62.5
...	...	...	...	...	...
전국	2019-01-02	0.0	0.0	울릉도	목포

→

	rainAvgWholeNorm
0	0.000941
1	0.000941
2	0.000941
3	0.000941
4	0.000941
...	...
35374	0.000941
35375	0.000000
35376	0.000000
35377	0.000000
35378	0.000000

지역명	일시	평균일강수량(mm)	최다일강수량(mm)	최다강수량지점	1시간최다강수량(mm)
서울경기	2020-08-11	62.0	106.6	인천	34.5
서울경기	2020-08-10	25.0	74.3	백령도	28.6
서울경기	2020-08-09	105.5	143.0	파주	39.2
서울경기	2020-08-08	16.1	28.5	이천	12.5
서울경기	2020-08-07	0.1	2.2	이천	1.6
...	...	...	...	...	...
서울경기	2019-01-02	0.0	NaN	NaN	NaN

→

	rainAvgCapNorm
0	0.0
1	0.0
2	0.0
3	0.0
4	0.0
...	...
35374	0.0
35375	0.0
35376	0.0
35377	0.0
35378	0.0

3. Categorical Analysis

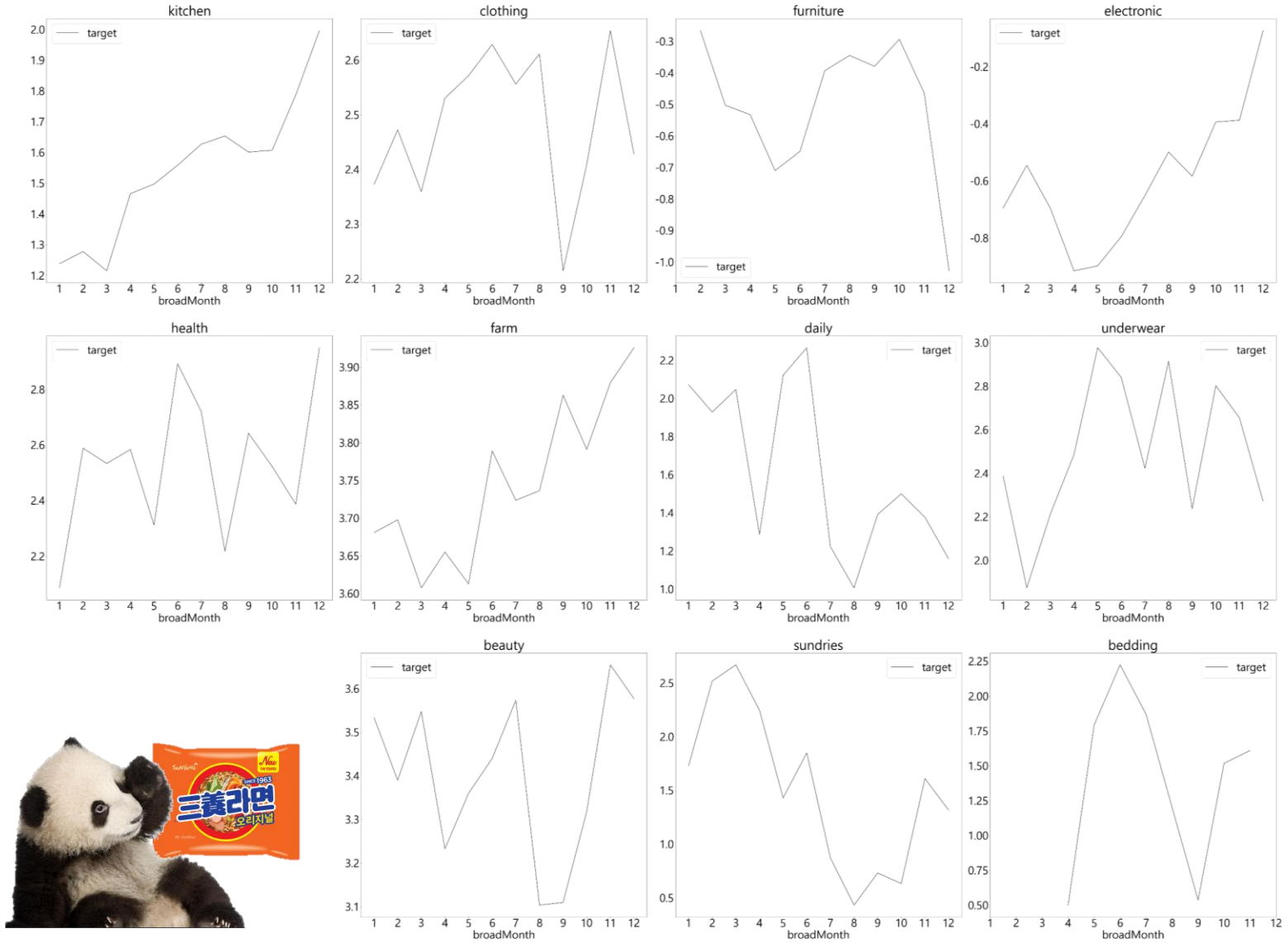

그림 6. 월별 상품군별 target 평균.

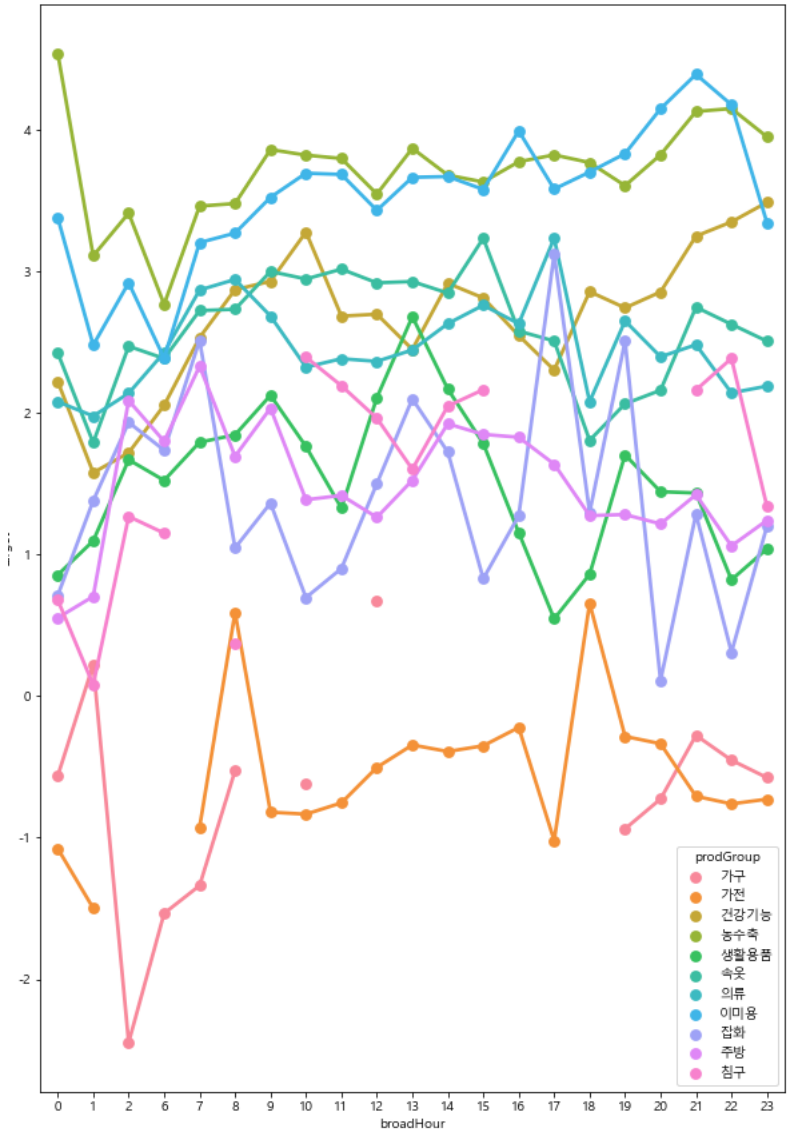

그림 7. 시간별 상품군별 target 평균.

3. Categorical Analysis

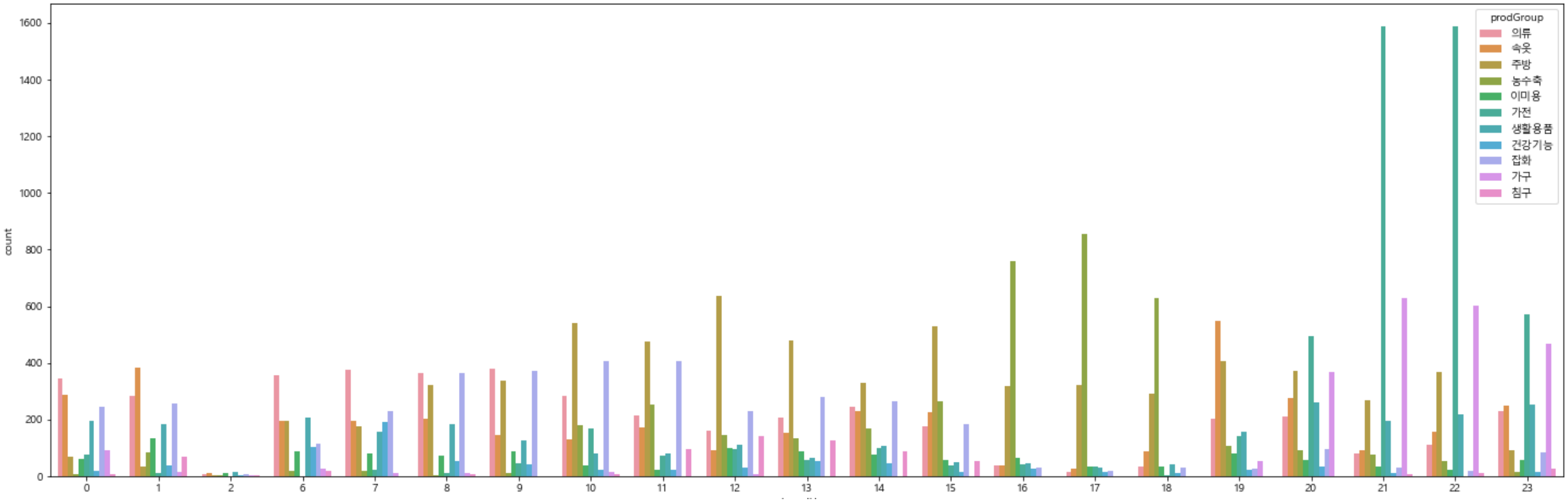

그림 8. 시간별 상품군별 방송횟수.

4. Rating Data

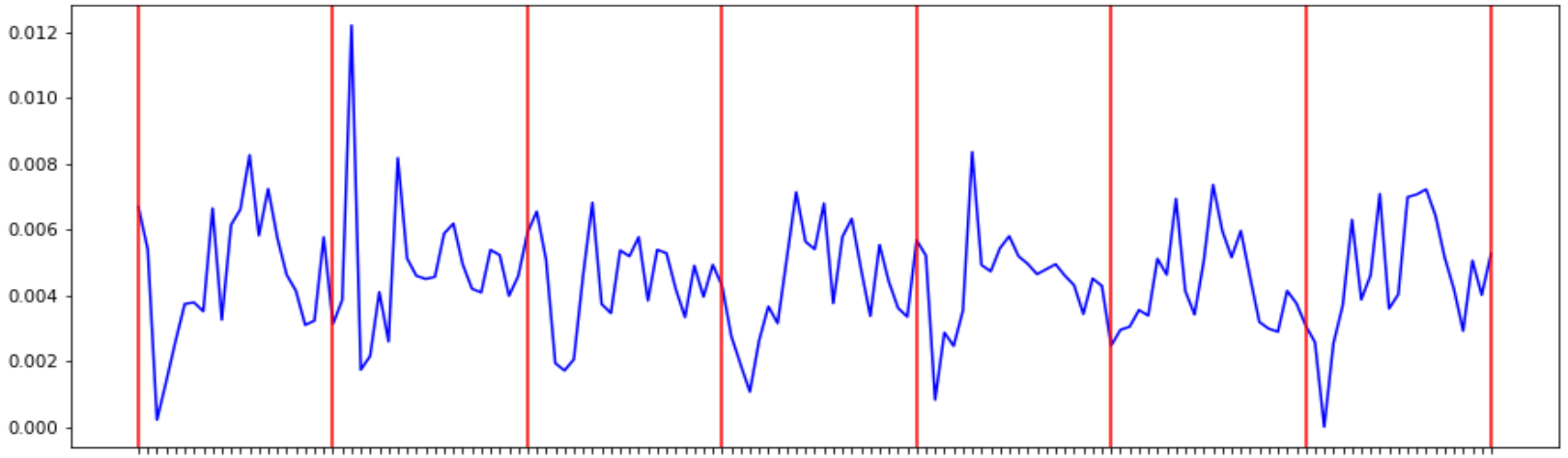

그림 9. 요일, 시간별 평균 시청률.

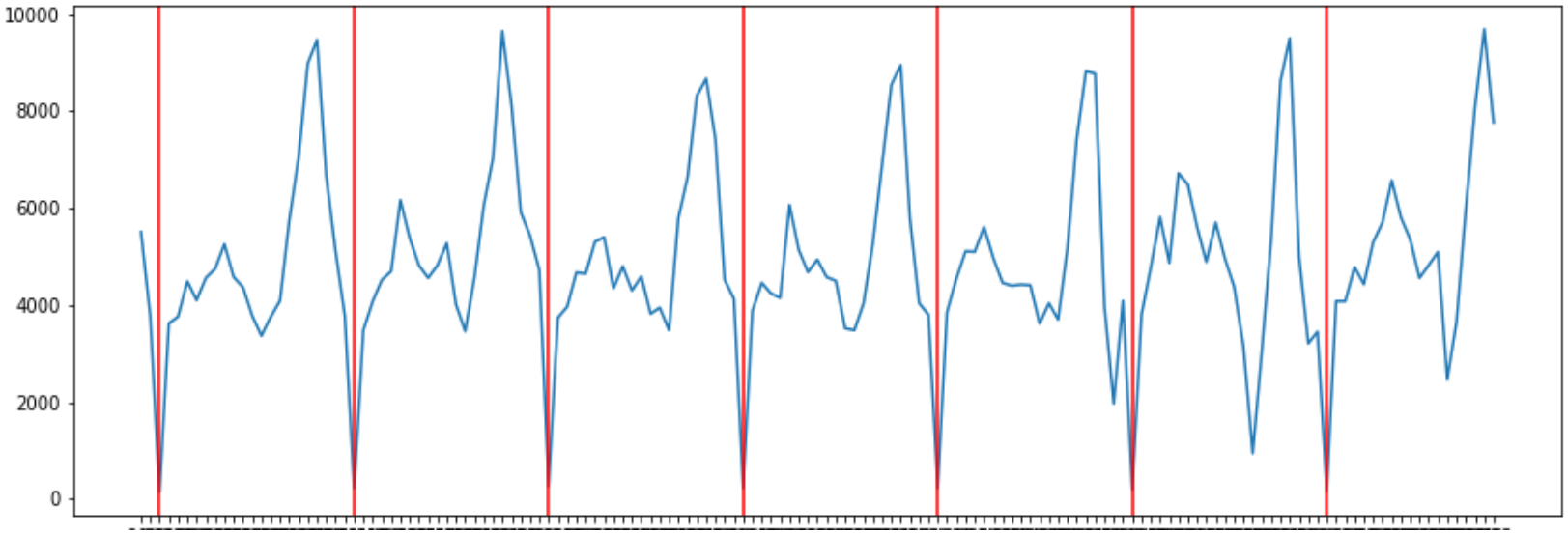

그림 10. 요일, 시간별 노출시간 합.

- 시청률과 노출시간은 서로 상이한 양상을 보인다.

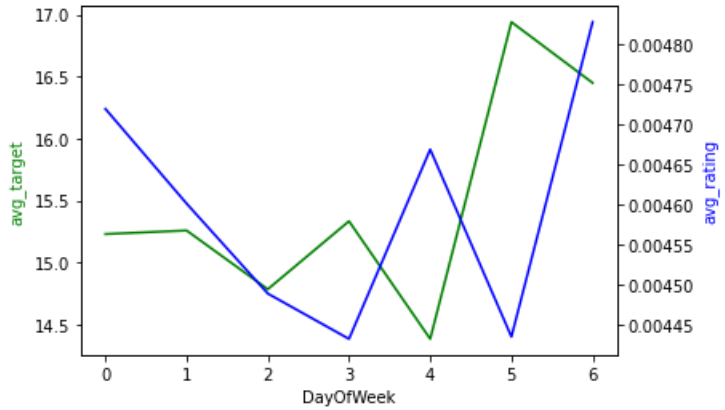

그림 11. 요일별 평균 시청률과 target 평균.

- 과연 시청률이 target에 유의미한 영향을 미칠까? 검증해보자.

4. Rating Data

Transfer Entropy [6]

- Entropy: $H(x) = -\sum p(x)\, log\, p(x)$, 전체 상태에 대한 정보량의 기대치
  - ✓ **The average amount of information** to encode independent draws of the discrete variables i following a probability distribution $p(x)$.
  - ✓ Roughly, we can say entropy describes the degree of uncertainty to explain the system of X when the outcomes occur at $p(x)$. -> 불확실성을 측정할 수 있다!
- Conditional entropy: $H(X|Y) = -\sum_{j=1}^{M} p(y_i) \sum_{i=1}^{N} p(x_i|y_i)\, log\, p(x_i|y_i)$
  - ✓ Y의 값이 관측되었을 때, X가 발생할 확률에 대한 정보량의 기대치
- Transfer Entropy: $TE_{X\to Y} = H\left(Y_{t+1}\,|Y_t^{(l)}\right) - H\left(Y_{t+1}\,|Y_t^{(l)}, X_t^{(k)}\right)$; $X_t^{(k)} = X_t, X_{t-1}, \cdots, X_{t-k+1}$, $Y_t^{(l)} = Y_t, Y_{t-1}, \cdots, Y_{t-l+1}$
  - ✓ Transfer entropy describes the **information flow** by measuring the amount of lowering prediction uncertainty.
  - ✓ Y들의 과거 데이터만 있을 때의 불확실성과 X 값들이 같이 있을 때의 불확실성을 비교해서 얼마나 불확실성이 감소했는지를 계산
  - ✓ X가 Y에 영향을 많이 주면 많이 줄수록 X가 있을 때의 불확실성은 점점 작아질 것이다. -> $H\left(Y_{t+1}\,|Y_t^{(l)}, X_t^{(k)}\right)$ 값이 작아진다. -> $TE_{X\to Y}$ 값은 증가한다.

4. Rating Data

```
------------------------------------------------------------
Direction        TE    Eff. TE  Std.Err.   p-value    sig
------------------------------------------------------------
     X->Y    0.0011     0.0010    0.0001    0.0000    ***
     Y->X    0.0091     0.0089    0.0001    0.0000    ***
------------------------------------------------------------
Bootstrapped TE Quantiles (1000 replications):
------------------------------------------------------------
Direction      0%     25%     50%     75%    100%
------------------------------------------------------------
     X->Y  0.0000  0.0002  0.0002  0.0003  0.0007
     Y->X  0.0000  0.0001  0.0002  0.0002  0.0007
------------------------------------------------------------
Number of Observations: 37368
------------------------------------------------------------
p-values: < 0.001 '***', < 0.01 '**', < 0.05 '*', < 0.1 '.'
```

그림 12. Results of transfer entropy.

- p-value < 0.001
- 시청률과 판매량은 밀접하게 관련되어있다. 그러나,

- 시계열이란? : 시간의 흐름에 따라 **일정한 간격**으로 기록된 데이터.
- Transfer entropy는 두 시계열 데이터간의 인과관계를 파악하는 기법.
  - ✓ 상품들이 일정한 간격을 가지고 출현하지 않는데, 시계열 분석 방법을 쓸 수 있을까
  - ✓ 만약 시간, 일 단위로 묶는다면 한 간격에 여러 개의 관측치가 존재한다. 이는 시계열 데이터가 아닌 패널데이터
  - -> Transfer entropy를 사용하기에 부적합한 데이터일 수도 있다. 결과를 맹신할 수 없다.
- 시청률 데이터를 어떻게 쓸지는 아직 논의 중.
- 우선 base model을 구축하고 추후에 다시 고려해보자

5. Modeling – 변수 삭제 및 생성

1. prodName vectorize – Term Frequency-Inverse Document Frequency (**TF-IDF**) vectorizer
   - TF-IDF는 단어의 빈도와 역 문서 빈도를 사용하여 Document-Term Matrix 내의 각 단어들마다 중요한 정도를 가중치로 주는 방법으로 문서 내에서 특정 단어의 중요도를 구하는 작업에 쓰일 수 있다.[7]
   - Training data와 Test data prodName을 합쳐서 corpus를 형성한 후 TF-IDF 수행.

```
string = string.lower()
string = re.sub('[^ㅱwㅱs]', ' ', string)
string = re.sub('ㅱd+', ' ', string)
string = string.replace('여자', '여성').replace('남자', '남성')
string = string.replace('ㅱ(무ㅱ)', '무이자').replace('무ㅱ)', '무이자')
string = string.replace('ㅱ(일ㅱ)', '일시불').replace('일ㅱ)', '일시불')
string = re.sub('밥솥', ' 밥솥 ', string)
string = re.sub('침대', ' 침대 ', string)
string = re.sub('에어컨', ' 에어컨 ', string)
string = re.sub('노트북', ' 노트북 ', string)
string = re.sub('티셔츠', ' 티셔츠 ', string)
string = re.sub('스타일러', ' 스타일러 ', string)
string = re.sub('손질', ' 손질 ', string)
string = string.replace("s/s","ss").replace("ss", "시즌")
string = string.replace("f/w", "시즌").replace("썸머", "시즌")
string = string.replace("lg", " lg ")
string = string.replace("울트라hd", " uhd ")
string = string.replace("tv", " tv ")
string = string.replace("김치", " 김치 ")
string = string.replace("기초세트", " 기초세트 ")
string = ' '.join([x for x in string.split() if len(x) > 1])
return string
stopwords = {'세트', '인용', '패키지', '시리즈', '매', '봉', '종', ''
             'aab의', 'arc', 'bna', 'by', 'ev', 'fq', 'fxkr', 'hnc', 'in',
              'jk', 'kg', 'kna', 'knb', 'af', 'vbc',
              'nt', 'nu', 'pat', 'qs', 'tq', 'uk', 'um', 'un', 'gabl'
             'crp', 'fg', '일시불', 'aae', 'bna', 'ev', 'dv', 'm','b', 'qv','fq','jk','kwa', 'nt', 'nu',
 'gs','tq', 'a','ia', 'wwj', 'hnc','x','j','ih', 'l','kg','g', 'fs',
 'the','arc', 'bna', 'by', 'ev', 'fq', 'fxkr', 'in', 'jk', 'kg', 'kna', 'knb',
 'nt', 'nu', 'pat', 'qs', 'tq', 'uk', 'um', 'un', 'gabl', 'crp', 'fg'}
```

- 같은 단어이지만 다르게 표기되어 있는 것들을 같게 처리
- Stopwords: corpus에서 단어장을 생성할 때 무시할 수 있는 단어 설정[8]

```
['cerini', 'led', 'lg', 'nnf', 'tv', 'uhd', '가스레인지', '가스와이드그릴레
인지', '갑오징어', '고칼슘검은콩두유', '골드', '국내산', '귀한', '기능성',
'기초세트', '김치', '남성', '냉장고', '노비타', '니트', '단하루', '대용량',
'대형', '더블', '데일리', '두유', '드라이셀', '드로즈', '라쉬반', '라이크
라', '락앤락', '락토핏', '란쥬', '런닝', '레이스', '레이프릴', '루이띠에',
'린나이', '릴렉스', '마르엘라로사티', '매직쉐프', '매직스페이스', '멀티',
'멋진밥상', '목걸이', '무료설치', '무선', '무이자', '문어', '박스', '밥솥',
'백팩', '베스트', '벽걸이', '보루네오', '보몽드', '분쇄믹서기', '브라탑',
'블랙', '비데', '비버리힐스폴로클럽', '삼성', '샌들', '생유산균골드', '선글
라스', '선일금고', '세탁기', '소가죽', '소파', '손질', '숄더백', '순면',
'쉐이핑', '슈퍼싱글', '스마트', '스탠드', '스텐', '스텐큐브', '스트레치',
'시그니처', '시즌', '심리스', '싱글', '씽큐', '아가타', '아키', '안동간고등
어', '언더셔츠', '에버라스트', '에어컨', '에어프라이어', '에지리', '엘렌실
라', '여성', '옛날', '올리고', '원피스', '월드컵', '유귀열의', '유로탑',
'이볼브', '자수', '자연산', '전자', '전자레인지', '종근당건강', '종세트',
'쥐포', '지이링클', '참존', '천연소가죽', '초특가', '침대', '칼리베이직',
'캐리어', '컬렉션', '코몽트', '코튼', '쿠미투니카', '쿠쿠전기', '크로스백',
'크리스티나앤코', '탑뉴스', '테스토', '토트백', '통돌이', '통오징어', '트렁
크', '트레킹화', '티셔츠', '파뷔에', '팔찌', '팬츠', '포기', '푸마', '풀세
트', '풍기인견', '프라다', '프라이팬', '프리미엄', '프리미엄형', '플랫타
입', '피시원', '피올레', '한일', '헤드', '호두아몬드', '화이트', '후라이
팬', '흥양농협']
```

그림 13. Result of TF-IDF with 150 max features.

5. Modeling – 변수 삭제 및 생성

2. isFemale, isMale, paymentPlan, isWeekend, isPrimeTime, motherCode, prodCode 변수 삭제 + 데이터 타입 변경

```
Index(['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct',
       'Nov', 'Dec', 'broadDay', 'broadDayOfWeek', 'broadHour', 'broadMin',
       'broadTime', 'prodName', 'prodGroup', 'unitPrice', 'priceIndex',
       'tempNorm', 'rainAvgWholeNorm', 'rainAvgCapNorm', 'target'],
      dtype='object')
```

Jan ~ Dec => **category**

broadDay, broadDayOfWeek, broadHour, broadMin => **category**

prodName => **TF-IDF**

prodGroup => **category**

unitPrice => **scaling**

→

```
broadMonth          category
broadDay            category
broadDayOfWeek      category
broadHour           category
broadMin            category
broadTime            float64
prodName              object
prodGroup           category
unitPrice              int64
priceIndex             int64
tempNorm             float64
rainAvgWholeNorm     float64
rainAvgCapNorm       float64
target               float64
dtype: object
```

3. 상품군별로 unitPrice의 격차가 크기 때문에 prodGroup별 scaling 수행

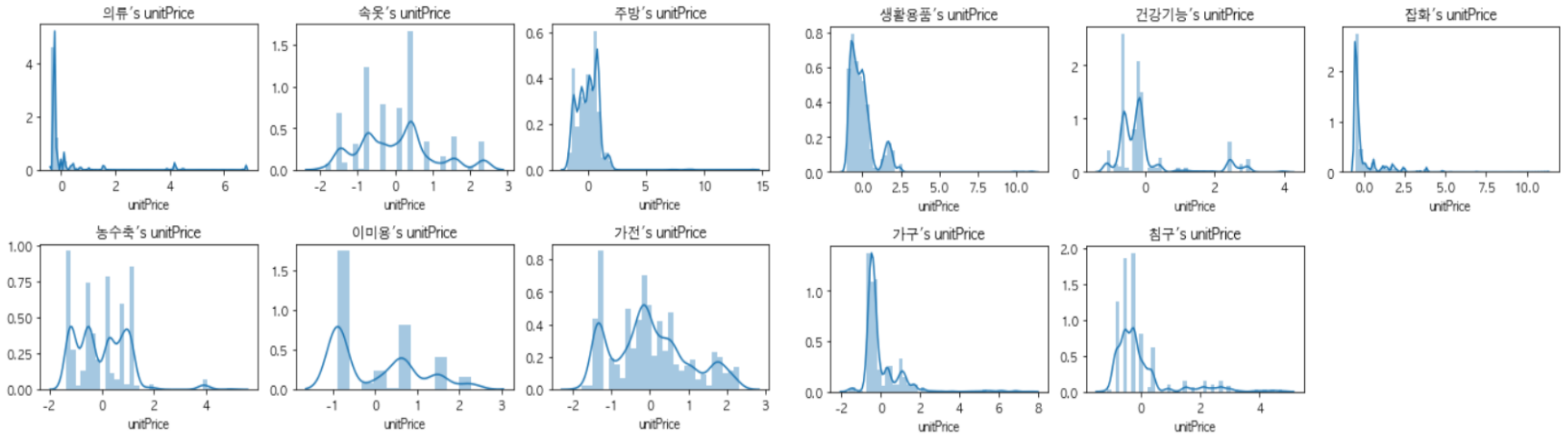

그림 14. prodGroup별 unitPrice scaling.

5. Modeling – base model

1. Linear Regression

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=3,
                                                    stratify=X.prodGroup)
```

MAPE: 84.28987118360318

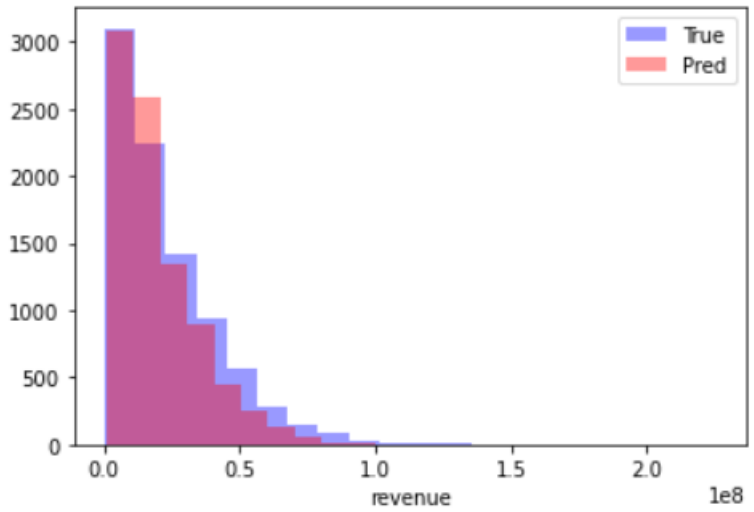

그림 15. Results of Linear Regression.

2. Support Vector Machine - Regression

- SVM은 분류 문제를 해결하기 위해 개발되었지만 최근에는 회귀분석과 관련된 문제를 해결하기 위해 확장되었다. [9]
- Kernel은 Input space의 데이터를 선형분류가 가능한 고차원으로 확장시킨 후 분류를 진행하는 것! [10]

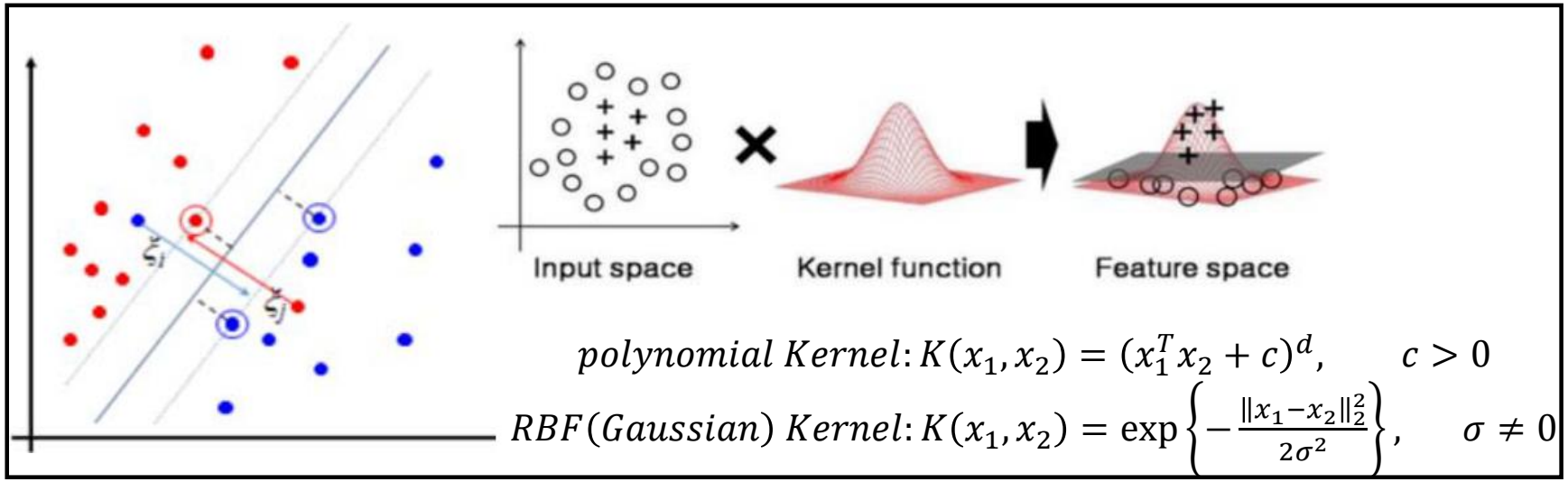

그림 16. Support Vector Machine[11]

코드가 오류도 안 나고 끝나지도 않아요 결과가 안나와요....

5. Modeling – base model

3. Random Forest

1. For b=1 to B (b is the number of decision trees):
   a. Draw bootstrap sample Z^* of size N from the training data.
   b. Grow a random forest tree T_b to the bootstrapped data, by recursively repeating the following steps for each terminal node of the tree, until the minimum node size n_{min} is reached.
      I. Select m variables at random from p variables.
      II. Pick the best variable / split- point among the m.
      III. Split the node into two(or more) daughter nodes.
   c. Estimate OOB error by applying the tree to the OOB data.
2. Output the ensemble of trees $\{T_b\}_{1\cdots B}$ by using majority voting.

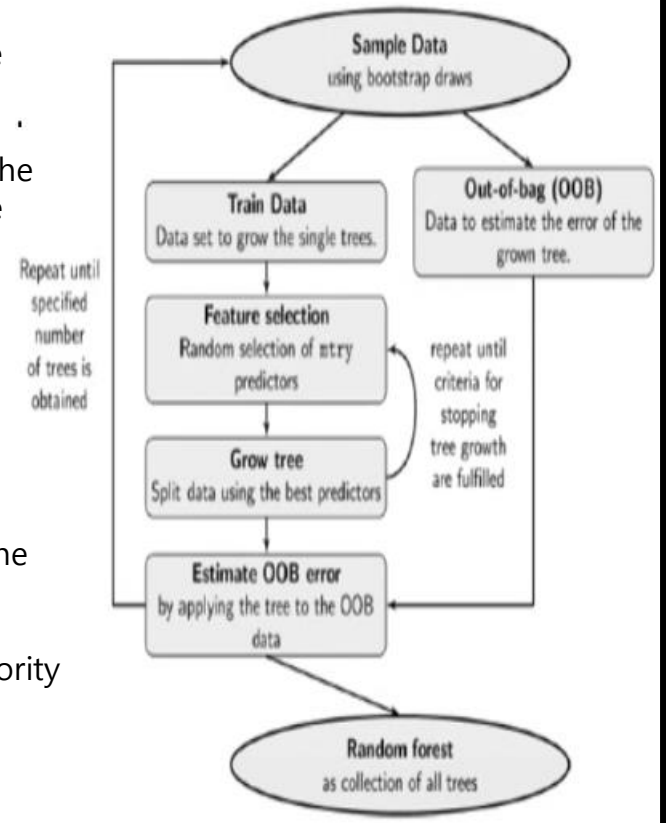

그림 17. Algorithm of random forest. [11]

	features	importances		features	importances
6	prodGroup	0.602996	**72**	삼성	0.000486
7	unitPrice	0.277610	**144**	패키지	0.000387
124	침대	0.036361	**27**	김치	0.000370
4	broadMin	0.020253	**75**	선글라스	0.000362
3	broadHour	0.013370	**52**	매직스페이스	0.000318
58	무이자	0.010980	**32**	단하루	0.000316
47	루이띠에	0.003858	**99**	안동간고등어	0.000298
147	푸마	0.003682	**29**	냉장고	0.000289
5	broadTime	0.003341	**14**	lg	0.000279
112	유로탑	0.003030	**97**	아가타	0.000248
60	박스	0.002375	**103**	에어프라이어	0.000199
86	스마트	0.002182	**51**	매직쉐프	0.000184
17	uhd	0.002077	**116**	전자	0.000175
16	tv	0.001544	**159**	화이트	0.000168
8	priceIndex	0.001376	**10**	rainAvgWholeNorm	0.000166
137	통돌이	0.001329	**107**	옛날	0.000146
15	nnf	0.001309	**1**	broadDay	0.000143
39	드로즈	0.001152	**9**	tempNorm	0.000122
92	시리즈	0.001056	**114**	자수	0.000117
76	선일금고	0.000970	**77**	세탁기	0.000093
113	이볼브	0.000940	**37**	두유	0.000045
57	무선	0.000895	**161**	흥양농협	0.000039
131	쿠쿠전기	0.000785	**158**	호두아몬드	0.000038
2	broadDayOfWeek	0.000721	**85**	슈퍼싱글	0.000037
61	밥솥	0.000643	**21**	고칼슘검은콩두유	0.000034

표 11. 변수별 중요도 상위 50개.

```
Best Score:  -0.5743755357669909
Best_Params:  {'n_estimators': 142, 'min_samples_split': 2, 'min_samples_leaf': 2, 'max_features': 'auto', 'max_depth': 8, 'bootstrap': False}
```

5. Modeling – base model

3. Random Forest

MAPE: 81.35033476425907

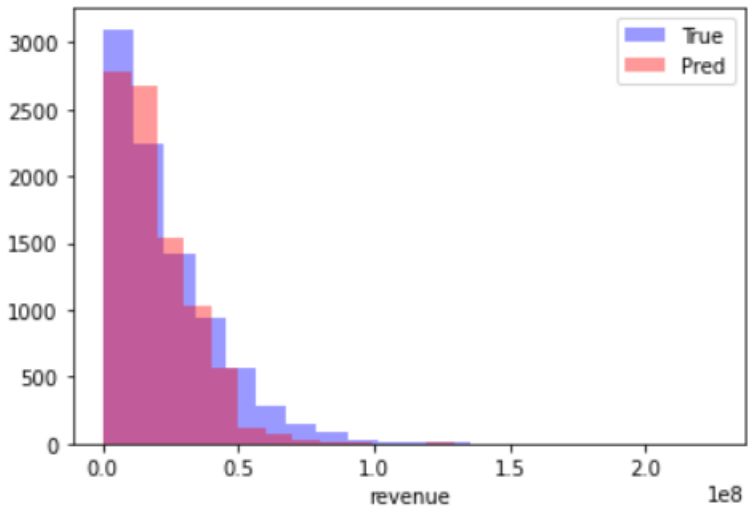

그림 18. Results of Random Forest.

4. Gradient Boosting(XGBoost)

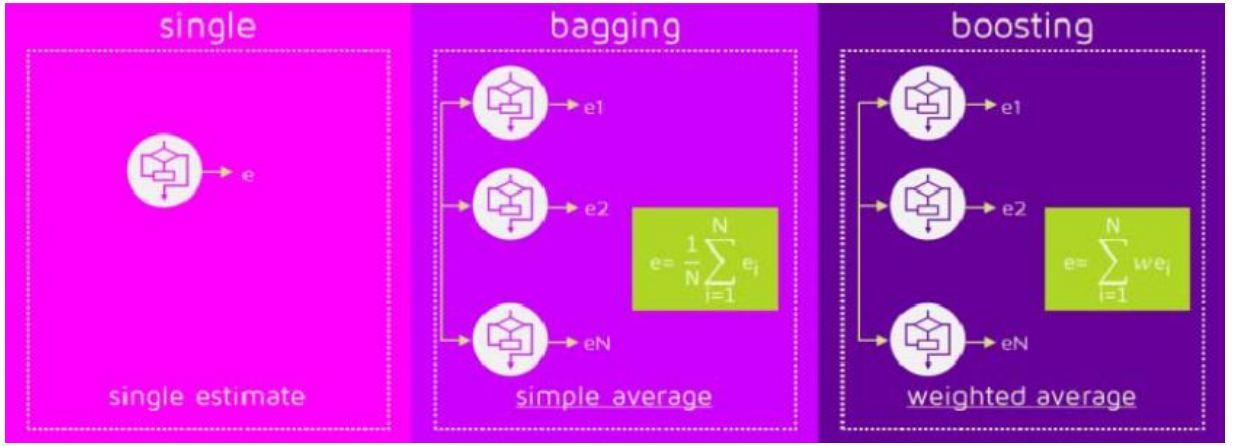

그림 18. 모델별 작동 방법. [11]

Algorithm 1: Gradient_TreeBoost

1 $F_0(\mathbf{x}) = \arg\min_\gamma \sum_{i-1}^{N} \Psi(y_i, \gamma)$.
2 For $m = 1$ to M do:
3 $\quad \tilde{y}_{im} = -\left[\frac{\partial \Psi(y_i, F(\mathbf{x}_i))}{\partial F(\mathbf{x}_i)}\right]_{F(\mathbf{x})=F_{m-1}(\mathbf{x})}, \; i = 1, N$
4 $\quad \{R_{lm}\}_1^L = L$ – terminal node $tree(\{\tilde{y}_{im}, \mathbf{x}_i\}_1^N)$
5 $\quad \gamma_{lm} = \arg\min_\gamma \sum_{\mathbf{x}_i \in R_{lm}} \Psi(y_i, F_{m-1}(\mathbf{x}_i) + \gamma)$
6 $\quad F_m(\mathbf{x}) = F_{m-1}(\mathbf{x}) + \nu \cdot \gamma_{lm} 1(\mathbf{x} \in R_{lm})$
7 endFor.

그림 19. Algorithm of gradient boosting. [11]

MAPE: 47.17919819007912

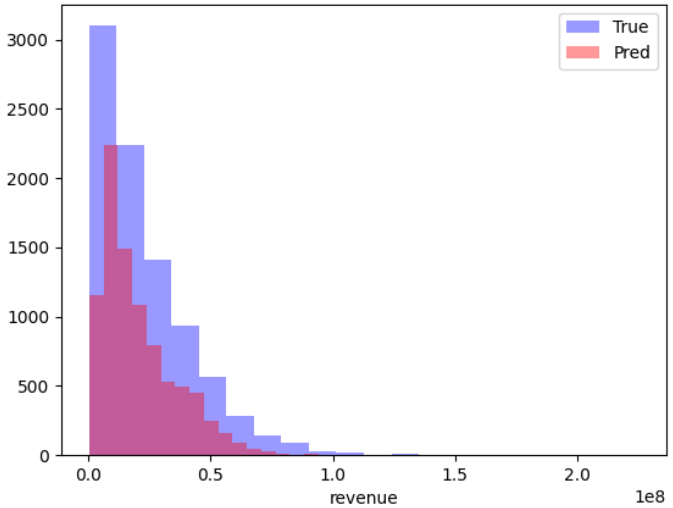

그림 18. Results of XGBoost.

6. 추후 계획

References

1. "데이터분석분야 챔피언 리그 문제 및 데이터 자주 묻는 질문 (ver. 8.6)." *빅콘테스트*, 2020년 8월 18일 접속, https://www.bigcontest.or.kr/community/faq.php?UfGubun=A&fIdx=31.
2. "생활물가지수." *KOSIS*, 2020년 8월 4일 수정, 2020년 8월 18일 접속, http://kosis.kr/statHtml/statHtml.do?orgId=101&tblId=DT_1J17005&vw_cd=MT_ZTITLE&list_id=P2_6&seqNo=&lang_mode=ko&language=kor&obj_var_id=&itm_id=&conn_path=MT_ZTITLE.
3. '2020년 1월 소비자물가동향." *KDI 경제정보센터*, 2020년 2월 4일 수정, 2020년 8월 20일 접속, http://eiec.kdi.re.kr/policy/materialView.do?num=197246&topic=
4. "기온분석." *기상청 기상자료개방포털*, 2020년 8월 18일 접속, https://data.kma.go.kr/stcs/grnd/grndTaList.do?pgmNo=70.
5. "조건별통계." *기상청 기상자료개방포털*, 2020년 8월 18일 접속, https://data.kma.go.kr/climate/RankState/selectRankStatisticsDivisionList.do?pgmNo=179.
6. "[예제] 시계열 Data로부터 Transfer Entropy 구하기." *종혁의 저장소*, 2019년 2월 19일 수정, 2020년 8월 12일 접속, https://mons1220.tistory.com/154
7. "TF-IDF." *딥 러닝을 이용한 자연어 처리 입문*, 2020년 3월 14일 수정, 2020년 8월 19일 접속, https://wikidocs.net/31698.
8. "Scikit-Learn의 문서 전처리 기능." *데이터 사이언스 스쿨*, 2016년 6월 14일 수정, 2020년 8월 21일 접속, https://datascienceschool.net/view-notebook/3e7aadbf88ed4f0d87a76f9ddc925d69/.
9. 박승환, et al. "Support Vector Machine-Regression 을 이용한 주기신호의 이상탐지." 품질경영학회지 38.3 (2010): 355.
10. "Kernel-SVM." *ratsgo's blog for textmining*, 2017년 5월 30일 수정, 2020년 8월 21일 접속, https://ratsgo.github.io/machine%20learning/2017/05/30/SVM3/.
11. 손소영, "데이터마이닝 이론 및 응용", 연세대학교 산업공학과