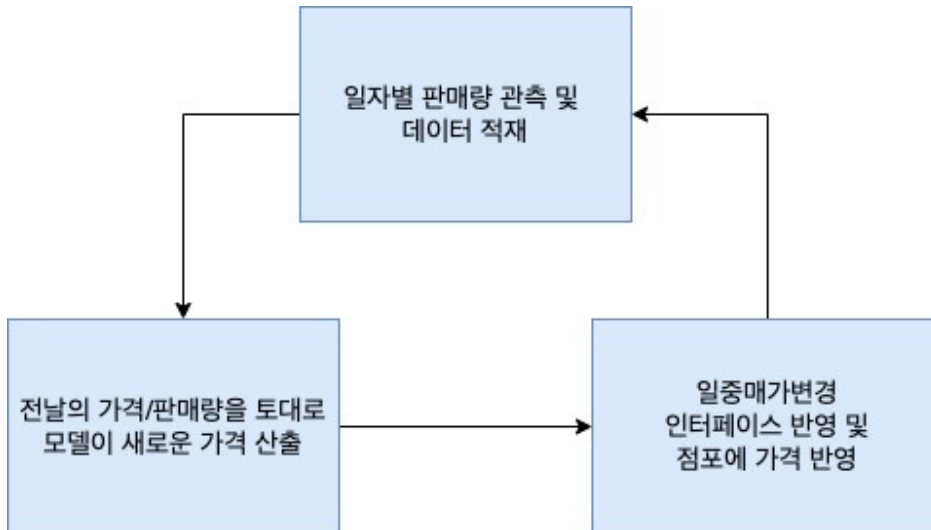


DP 파이프라인 개요

Dynamic Pricing 산출을 위해 DS챗터에서 수집한 데이터 수집 및 전처리 / 모델링 / DP 인터페이스 적재 과정에 대해 설명합니다.



1. 학습 데이터 정보

1) 모델 학습에 사용하는 데이터

i) DP 적용 가능 대상 품목 테이블 (smart-ruler-304409.team_datascience.DP_target_prdt, [링크](#))

- 전점 & 상품 마스터(가공 전체) 테이블 기준으로 DP 적용 대상에 제외되는 CASE들 컬럼별로 구분하여 적재한 테이블 (138개 점포 대상 약 2만개 SKU로 총 400만 행정도)
- 데이터의 소스는 GCP(상품 및 점포, 행사 정보), ADG(아이앤씨 요청으로 필터링하는 정보), 엑셀 파일(바이어들이 수기로 작성한 리스트)
- dp_target_yn 컬럼이 'Y'일 경우 DP 적용 가능한 상품**
- sku_rank 컬럼의 경우 매출금액-이익금액 순으로 우선순위를 표시하였으며 '2022-07-20' 시점에서 최근 2주동안 팔린 상품의 경우만 ranking이 매겨짐(갱신필요)
- 현재 엑셀 업로드를 통해 직접 제거하는 로직이 포함되어 있으므로 해당 테이블은 **2주마다 갱신되는 것**을 목표로하고 있으나, 추후 daily로 적재될 가능성 있으며 **daily로 적재될 경우 일반행사 테이블, POS행사 테이블 등 행사 관련 테이블들의 적재시간 체크가 필요함**
- uprice_daily, ucost_daily, upper_bound, lower_bound 등은 dp_verify 테이블의 정보를 가져오며 daily로 갱신되는 것은 아님
- 현재 테이블 기준으로는 Daily로 끌어와야하는 정보는 없음**

DP_target_prdt

🔍 쿼리

👤 공유

📄 복사

🔖 스텑샷

🗑 삭제

📄 내보내기

스키마

세부정보

미리보기

필터

속성 이름 또는 값 입력

필드 이름	유형	모드	대조	정적 태그	설명
index_day	STRING	NULLABLE			데이터 갱신일자(8월 1주차 부터 2주마다 갱신)
prdt_gcode_cd	STRING	NULLABLE			대분류 코드
prdt_mcode_cd	STRING	NULLABLE			중분류 코드
prdt_dcode_cd	STRING	NULLABLE			소분류 코드
str_cd	STRING	NULLABLE			정포코드
prdt_cd	STRING	NULLABLE			상품코드
prdt_nm	STRING	NULLABLE			상품명
new_prdt_yn	STRING	NULLABLE			신상품 여부(Y: 신상품, N: 신상품 아님)
oper_state_yn	STRING	NULLABLE			운영상태 정상여부(Y: 대상(10, 20, 30), N: 비대상(40이상))
base_target_yn	STRING	NULLABLE			기본적인 DP 적용 타겟 여부(우유, 국산분유, 피코크분유/두유, 노브랜드(중분류), 바이어 조사 미운영 대상 상품 제외 Y)
pos_event_yn	STRING	NULLABLE			POS행사 대상 여부(Y: 대상, N: 비대상)
normal_event_yn	STRING	NULLABLE			일반행사 여부(Y: 대상, N: 비대상)
non_pog_yn	STRING	NULLABLE			미진물상품 여부(Y: 진물알함, N: 진물함)
direct_purchase_yn	STRING	NULLABLE			직매입여부(Y: 직매입상품, N: 기타상품)
season_yn	STRING	NULLABLE			시즌상품여부(Y: 시즌상품, N: 비시즌상품)
pos_week_yn	STRING	NULLABLE			POS 주자 행사 여부(Y: 행사상품 N: 비행사상품)
ic_no_regist	STRING	NULLABLE			신세계 I&C 요청 등록 불가 상품 여부(Y: 등록가능 N: 등록불가)

실행

저장

공유

알림

대보기

실행 시 이 쿼리가 685.03MB를 처리합니다.

1 SELECT * FROM `smart-ruler-384489.team_datascience.DP_target_prdt`

Alt+F1을 눌러 접근성 옵션을 확인합니다

결과 저장

데이터 탐색

쿼리 결과

작업 정보

결과

JSON

실행 세부정보

행	index_day	prdt_gcode_cd	prdt_mcode_cd	prdt_dcode_cd	str_cd	prdt_cd	prdt_nm
1	2022-07-28	350	0364	2288	1145	0727344000229	솔드라스루틴355ml병
2	2022-07-28	350	0364	2206	1060	0009029772004	벌스너우즈얼 기록(500ml*4캔+현용잔)
3	2022-07-28	350	0364	2290	1064	4011900695117	스트링제로 세로 비알코250ml캔
4	2022-07-28	350	0364	2292	1102	4053400268545	올라우스달러 년알콜릭 레몬330ml병
5	2022-07-28	350	0364	2286	1062	8801021228761	스틸라750ml병+전용잔2ea
6	2022-07-28	350	0364	2291	1142	8712000025649	하이너엔SL캐그
7	2022-07-28	350	0364	2286	1061	8712000025540	하이너엔330ml병
8	2022-07-28	350	0364	2289	1055	0856571008316	캐스케이드 허니 진저 라임355ml캔
9	2022-07-28	350	0364	2286	1133	8801021229034	스틸라500ml*6캔+각인잔 기록(A)
10	2022-07-28	350	0364	2286	1074	8801021227764	(S)스틸라 기록(500ml*4+B/3처리타지 전용
11	2022-07-28	350	0364	2290	1118	0011711276370	에먼 일퍼티 얼 로제375ml병
12	2022-07-28	350	0364	2290	1109	0850005189664	시가씨티셀쳐 메이어레몬에이드355ml캔
13	2022-07-28	350	0364	2290	1037	3830054842281	슈아미라나 레몬라임 330ml캔
14	2022-07-28	350	0364	2287	1037	4003669016104	슈나이다 텀5 호환박이세500ml병
15	2022-07-28	350	0364	2290	1062	0649245839022	그랜저리크 후버레드 그레이프루트355ml캔
16	2022-07-28	350	0364	2286	1039	8809311392147	하이너엔 기록(500ml*8캔+미니캡필터)
17	2022-07-28	350	0364	2290	1161	9421902883668	쓰샴330ml*6캔+파티팅블러
18	2022-07-28	350	0364	2289	1029	5412858000104	로슈프로10L330ml
19	2022-07-28	350	0364	2292	1064	0860000004442	테라스 믹스 마가리타 946ml
20	2022-07-28	350	0364	2289	1098	0636251899027	올리 펄킨 시트라스 커맨더 750ml
21	2022-07-28	350	0364	2287	1132	5410228234258	로가든(벨기어)500ml
22	2022-07-28	350	0364	2287	1046	4014086096303	5.0박이스500ml캔
23	2022-07-28	350	0364	2289	1158	8801021226651	구스 312 495ml캔(K)
24	2022-07-28	350	0364	2290	1036	0850005189664	시가씨티셀쳐 메이어레몬에이드355ml캔
25	2022-07-28	350	0364	2289	1082	7090038061534	민드하일 러스트375ml병
26	2022-07-28	350	0364	2290	1061	9352568000475	타이달 셀쳐 믹스베리330ml캔
27	2022-07-28	350	0364	2292	1024	6418654005540	꾸꼬논알콜(라거).330ml 캔

ii) 판매량 정보 테이블 (smart-datafabric.temp_1m.dp_temp)

- GCP: 영수증 정보테이블 (smart-ruler-304409.cds_core.TB_DW_RCIPT_DETAIL)
- ADG: 실적 마감 정보테이블 (AOWNER.SLPSS_STR_DAY_SALE)
- 판매량 정보 테이블은 기간에 따라 다른 테이블을 사용
 - 이들 이전의 실적은 GCP에서 마감된 영수증 테이블을 사용
 - 어제의 마감 실적은 GCP(오전 6시 적재) 대신 ADG(00시30분 적재)를 활용하여 사본 생성(01시 내외) 후 DP 산출 (RCP_DUMP_TMP_CODE)
 - 전날 실적을 이용한 모델의 추론은 01시에 시작할 수 있으며, 이후 인터페이스 적재를 위해 07시 이전에 완료되어야 함 (단 해당 ADG 매출 테이블은 summary 테이블이므로 실제 영수증 단위 정보 활용을 위해서는 별도의 테이블 탐색 혹은 GCP 적재 대기 필요)

dp_temp_202207192022-07-19쿼리공유복사스냅샷삭제내보내기

스키마세부정보미리보기

필터속성 이름 또는 값 입력

필드 이름	유형	모드	대조	정책 태그	설명
BIZ_DATE	STRING	NULLABLE			
STR_CODE	STRING	NULLABLE			
SKU_CODE	STRING	NULLABLE			
ORGL_BIZ_DATE	STRING	NULLABLE			
VEN_CODE	STRING	NULLABLE			
PM_FORM_CODE	STRING	NULLABLE			
TOT_SALE_QTY	FLOAT	NULLABLE			
TOT_SALE_CAMT	FLOAT	NULLABLE			
TOT_SALE_PAMT	FLOAT	NULLABLE			
SALE_QTY	FLOAT	NULLABLE			
SALE_CAMT	FLOAT	NULLABLE			
SALE_PAMT	FLOAT	NULLABLE			
ENURL_AMT	FLOAT	NULLABLE			
CCNT	INTEGER	NULLABLE			
PARTLUPRICE_PDN_QTY	FLOAT	NULLABLE			
PARTLUPRICE_PDN_AMT	FLOAT	NULLABLE			
ADRC_SALE_QTY	FLOAT	NULLABLE			
ADRC_CAMT	FLOAT	NULLABLE			
ADRC_PAMT	FLOAT	NULLABLE			
ADRC_ENURL_AMT	FLOAT	NULLABLE			
CRTN_DT	TIMESTAMP	NULLABLE			
CRTN_ID	STRING	NULLABLE			
CHG_DT	TIMESTAMP	NULLABLE			
CHG_ID	STRING	NULLABLE			

스키마 수정행 액세스 정책 보기

선택저장공유설정더보기

실행 시 이 쿼리가 257.63MB를 처리합니다.

1SELECT * FROM `emart-datafabric.temp_tm_dp_temp_20220719` LIMIT 1000

Alt+F1을 눌러 접근성 옵션을 확인합니다.

쿼리 결과결과저장데이터 탐색

작업 정보	결과	JSON	실행 세부정보						
행	BIZ_DATE	STR_CODE	SKU_CODE	ORGL_BIZ_DATE	VEN_CODE	PM_FORM_CODE	TOT_SALE_QTY	TOT_SALE_CAMT	T
1	20220719	1400	11130500000000	20220719	059003	1	52.0	638196.0	
2	20220719	1400	11150900000000	20220719	059003	1	40.0	90916.0	
3	20220719	1400	11420700000000	20220719	059003	1	132.5	226380.0	
4	20220719	1400	11421900000000	20220719	059003	1	77.9	97709.0	
5	20220719	1400	11425200000000	20220719	059003	1	59.76	91217.0	
6	20220719	1400	11473600000000	20220719	473001	1	155.2	167600.0	
7	20220719	1400	11547300000000	20220719	059003	1	23.0	18816.0	
8	20220719	1400	11569200000000	20220719	473001	1	207.2	430940.0	
9	20220719	1400	11657200000000	20220719	059003	1	306.62	269800.0	
10	20220719	1400	11657400000000	20220719	059003	1	194.5	112780.0	
11	20220719	1400	11668800000000	20220719	800767	1	40.0	199200.0	
12	20220719	1400	11761500000000	20220719	059003	1	33.0	479391.0	
13	20220719	1400	11762300000000	20220719	059003	1	187.0	351460.0	
14	20220719	1400	11817800000000	20220719	059003	1	14.0	178178.0	

페이지당 결과 수: 501 - 50 (전체 1000행) <>>

iii) 상품의 가격 제한 상하한선 및 매가Tracking 관련 테이블(링크)

- DP 가격 제한의 상한/하한 등의 기준을 정하기 위해 가공 담당 전 상품(SKU)의 일중 매가, 일중 원가, 상한가, 하한가를 Daily로 적재
- 일중 매가와 일중 원가의 경우 ADG 정보를 토대로 산출하고 있으며, 해당 테이블은 오전 3시 40분경 갱신되며 원매가 정보가 달라질경우 Overwrite 되는 방식으로 적재
- 상한가와 하한가의 경우 매가는 3가지 원가는 2가지 로직을 반영하는데, GCP의 원매가 변경 히스토리 테이블을 활용하며, 이 테이블은 오전 8시쯤 적재가 이루어짐
- 원매가 변경 히스토리 테이블의 적재 시간이 늦는 편이지만, 해당 정보는 DP 가격 추론에 사용되는 정보는 아니므로 직접적인 영향은 없음

작업 정보 결과 JSON 실행 세부정보

행	index_day	prdt_cd	prdt_nm	uprice_daily	ucost_daily	upper_bound	lower_bound
1	2022-07-19	0080432403105	더글랜리벳18Y700ml	262400	172165	262400	172165
2	2022-07-19	8801043005838	농심포테토칩어니언60g	1280	840	1280	840
3	2022-07-19	8801115140696	[구]쉐이킹 라이트 딸기바나나 300ml	1280	757	1280	757
4	2022-07-19	8801043036337	농심 오징어집 83g	1280	925	1280	925
5	2022-07-19	8801062009343	롯데 미니초코칩쿠키69g	1280	763	1280	763
6	2022-07-19	8801062875719	롯데 고깔콘찰옥수수맛 67G	1280	763	1280	763
7	2022-07-19	8801062384921	롯데 쌀로별 오리지널78G	1280	763	1280	763
8	2022-07-19	8801043005814	칩포테토오리지널(60g)	1280	840	1280	840
9	2022-07-19	8801068402223	삼립 푸린의혹신떡신딸기크림빵 100g	1280	756	1280	756
10	2022-07-19	8801043063357	농심 옥수수강 팝콘70g	1280	925	1280	925
11	2022-07-19	8801043036535	농심 닭다리 후라이드치킨맛 66g	1280	925	1280	925
12	2022-07-19	8801097234215	오란씨 파인 500ml	1280	764	1280	764
13	2022-07-19	8801045571577	오뚜기 컵누들매콤한맛 37.8g	1280	813	1280	813
14	2022-07-19	8809750471137	노브랜드 간편하갈비양념180g	1280	725	1280	725

필터 속성 이름 또는 값 입력



필드 이름	유형	모드	대조	정책 태그 ?	설명
index_day	DATE	NULLABLE			일자(일단위로 적재되기때문에 meta 정보로 활용)
prdt_cd	STRING	NULLABLE			상품 코드 - 가공담당 상품 한정
prdt_nm	STRING	NULLABLE			상품명
uprice_daily	INTEGER	NULLABLE			상품의 가장 최근 매가
ucost_daily	INTEGER	NULLABLE			상품의 가장 최근 원가
upper_bound	INTEGER	NULLABLE			상품 가격의 상한선(매가 산출 3가지 로직에 따라 산출된 매가 중 min값으로 반영)
lower_bound	INTEGER	NULLABLE			상품 가격의 하한선(원가 산출 2가지 로직에 따라 산출된 원가 중 max값으로 반영)

→ 앞선 3개 테이블 중 ii)판매량 정보 테이블만 Daily로 정보가 필요하며, 해당 데이터셋 구성은 오전 1시 경 완료됨

2. 모델 개요

1)Thompson Sampling for Dynamic Pricing

https://github.com/emartdt/dp-product/blob/main/models/Thompson_sampling_for_dynamic_pricing/dp_thompson.ipynb

2)Offline Pricing with Offline Data: Phase Transition and Inverse Square Law

https://github.com/emartdt/dp-product/blob/main/models/Online_Pricing_with_Offline_Data/220615%20Online%20Pricing%20with%20Offline%20Data%20%EB%AA%A8%EB%8D%B8%20EA%B5%AC%ED%98%84.ipynb

과거의 (가격, 판매량) 데이터를 기반으로 초기 모델을 학습 후, 새로운 (가격, 판매량) 데이터가 주어질 때마다 알고리즘을 업데이트하는 모델입니다.

Step1. 이 모델은 가격 p 와 판매량 D 사이에 선형 관계가 있다고 가정합니다. $D = \alpha + \beta p + \epsilon$ (ϵ 은 랜덤한 노이즈) 라고 가정하면, 매출(판매량 \times 가격)은 가격에 대한 2차 방정식으로 근사할 수 있습니다. 따라서, α 와 β 를 알 수 있다면 매출을 최대화하는 $p = -\alpha / 2\beta$ 를 구할 수 있습니다.

Step2. 그러나 α 와 β 에 대해서는 아는 바가 없으므로, 먼저 매가의 범위 내 상한가 혹은 하한가로 가격을 설정 후 판매량을 관측합니다.

Step3. 최초의 (가격, 판매량) 패턴을 통해 α 와 β 가 존재하는 범위를 2차 부등식으로 제한하고, 해당 부등식 범위 내에서 판매량을 최대화하는 가격의 근사값을 설정 후 다음 판매량을 업데이트합니다.

Step4. 이후 지속적으로 판매량을 관측 후 가격을 수정해나갑니다.

- 현재 이 모델은 한 상품(12개 점포 기준)의 가격을 업데이트하는데 약 40초가 소요됩니다. 따라서 200개의 상품 가격을 예측한다고 가정 시 약 2시간 30분, 1000개 상품의 경우 10시간 이상이 걸리고 있습니다.
- 다만 모델의 속도 측면에서 개선할 여지가 있어 1000개의 상품은 6시간 이내 예측이 완료되도록 구현할 예정입니다.

3) Bayesian Optimal Pricing

해당 모델은 가격별 판매량을 통해 확률적 가정을 기반으로 구매력을 산출하고, 이를 통해 판매 가격을 결정할 수 있는 Dynamic 모델입니다.

Step1. 판매량이 수요곡선을 따른다는 믿음에서 출발합니다.

- $= \wedge$
- $[] = +$

a, c값을 잘 구할 경우, 상품 총이익 $=()$ 을 최대화하는 가격 $(= / 1 +)$ 을 구할 수 있는 모델입니다.

Step2. a와 c를 합리적으로 구하기 위해 Bayesian_Optimal 방법을 사용합니다.

2-1. a와 c를 임의의 랜덤값으로 넣어도 되지만, hyperparameter의 빠른 수렴을 위해 a, c를 계산해줍니다.

- hyperparameter c는 가격탄력성을 나타내며, 이는 Log-Log Model을 통해 계산됩니다.
- $\log() = + \times \log()$, $=$

해당 모델을 기반으로 hyperparameter a, c를 구하고 요일별, 점포별 가중치 등을 넣어 계산해줍니다.

2-2. PyMC를 통해 모델링을 구현해줍니다.

- Model 객체 생성

1) hyperparameter a, c가 Cauchy 사전 분포를 따른다고 가정합니다.

2) 실제 관측치를 통해 dataset에서 결과의 샘플링 분포를 정의해줍니다. (likelihood)

3) Model Fitting을 통해 알려지지 않은 변수에 대한 사후 추정값을 얻는 process를 진행합니다. 해당 과정에서 MCMC 샘플링을 통해 사후 분포에서 추출한 샘플을 기반으로 loss를 계산할 수 있습니다.

→ 현재는 상품별 1000번의 샘플링 진행, chain개수 4개, core는 -1로 설정하여 진행하고 있습니다.

Step3. 최적 가격을 산출해 줍니다.

- $=()$
- $= +()$
- $= / 1 +$

앞서 MCMC로 구한 a, c를 사용하여 최적의 가격을 도출해줄 수 있습니다.

Step4. 최근 판매량을 관측하며 가격을 업데이트합니다.

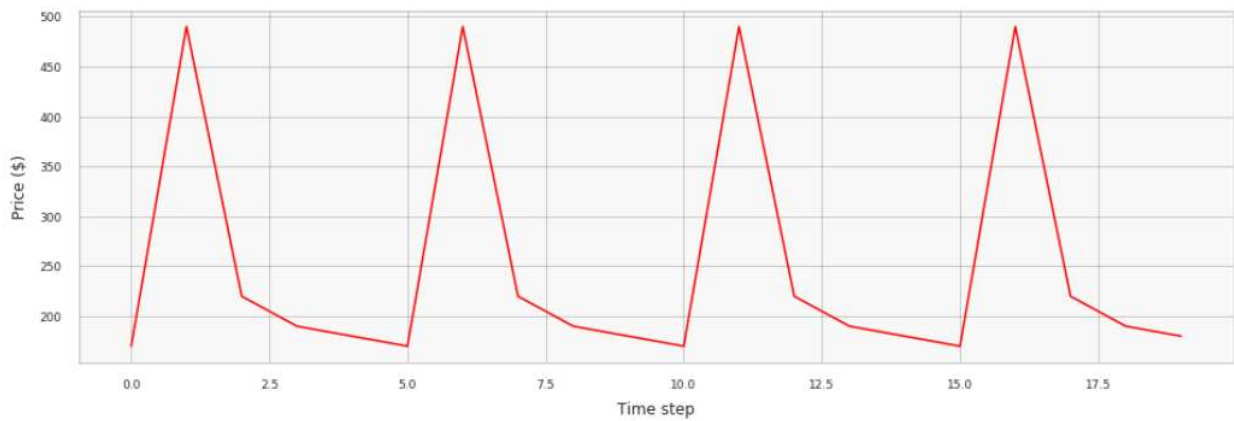
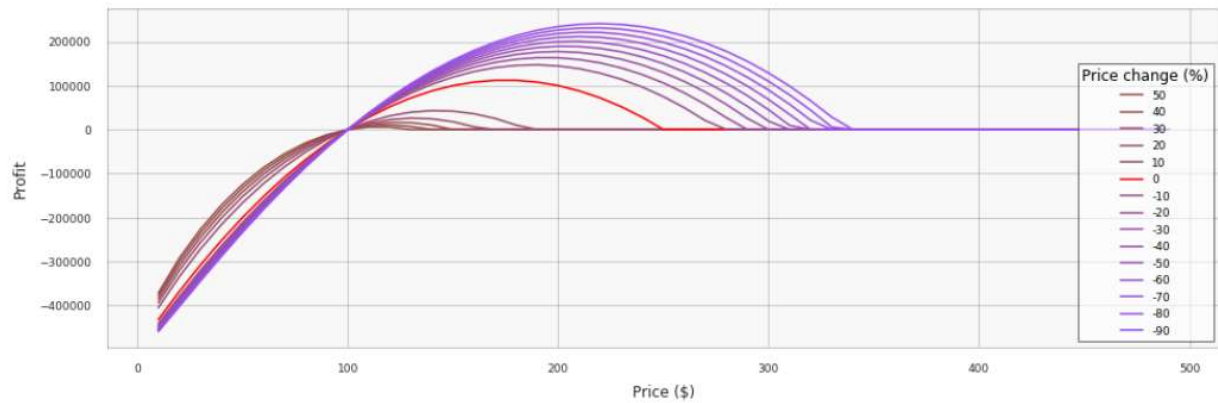
- 현재 해당 모델은 하나의 상품 당 약 40~60초의 시간이 소요됩니다. 1차 목표로 하는 200개의 상품 가격을 예측하게 되면 최대 3시간 20분의 시간이 소요됩니다.
- 시간이 오래 걸리는 부분에서 개선의 여지가 있으며, 대상 상품 확장 시 모델 구동 속도를 좀 더 빠르게 구현할 수 있도록 개선할 예정입니다.

4) Deep reinforcement learning for supply chain and price optimization

- 본 모델은 강화학습을 사용하여 상품별 최적 가격을 제안하는 모델입니다.
- 일반적인 지도/비지도 학습과는 꽤나 다른 학습유형으로 학습 과정 방법은 Reward(상품이익)을 최대화 하는 방향으로 직접 모델이 부딪치며 학습합니다.

i) 모델 알고리즘 설명

- DP 모델은 최적의 가격을 제안하기 위해 다양한 가격-판매량의 pair가 필요하지만, 현재 이마트에서는 바이어가 상품의 가격을 컨트롤 하기 때문에 한정된 종류의 가격에서 판매한 정보만 있다는 문제가 있습니다.
Q. 이를 해결하기 위해서는? 다양한 가격과 그에 해당하는 판매량의 데이터를 생성해야한다.
→ 이를 위해서 본모델에서는 profit을 고려한 HI-LO Pricing 전략과 유사한 방법을 사용합니다. (*일부 제품은 가격인하용으로 판매하고(로스 리더/미끼 상품) 일부는 정상가격으로 판매하여 높은 이익을 달성하고자 하는 전략)



-본 수식에서는 Price의 증감을 고려하여 상품의 판매량을 예측하고 일정 가격으로 올라가면 상품의 Profit이 0이 되는 구조를 가지며, 이마트와 유사하게 시간(T)에 따라 행사가격의 편차를 두어 마치 행사와 비행사때 원&매가가 달라지는 점등의 요소를 반영하여 Fitting을 하게 됩니다.

1.k, a, b 파라미터 구하기

$$d(p_t, p_{t-1}) = d_0 - k \cdot p_t - a \cdot s((p_t - p_{t-1})^+) + b \cdot s((p_t - p_{t-1})^-)$$

where

$$x^+ = x \text{ if } x > 0, \text{ and } 0 \text{ otherwise}$$

$$x^- = x \text{ if } x < 0, \text{ and } 0 \text{ otherwise}$$

-상품의 최근 2년간 판매량 기반으로 위에 정의한 수식에 따라 hyper parameter 인 k(기울기), a(가격 증가시 coefficient), b(가격 하락시 coefficient) Fitting 하여 구해줍니다. (d0 초기판매량, s: root(비선형 반영), pt:상품의 현재가격, pt-1: 상품의 이전가격, d(pt,pt-1): 현시점의 판매량

(상품의 현재가격, 이전가격, 판매량, 초기 판매량 등을 알고 있으므로 구할 수 있습니다.)

→ 다양한 상품 가격에 따른 판매량 & 이익금액 정보 구하기 완료!

2. 강화학습 시작

- 다양한 가격에 대한 판매량(예측이지만..)이 준비 되었으므로 이를 상품당 시점별로 강화학습을 진행합니다.
- 여기서 사용하는 강화학습 모델은 DQN을 사용하며 Model Based 방법중 기본 방법인 MLP를 사용합니다, input(State), output(action), reward는 아래와 같이 구성됩니다.

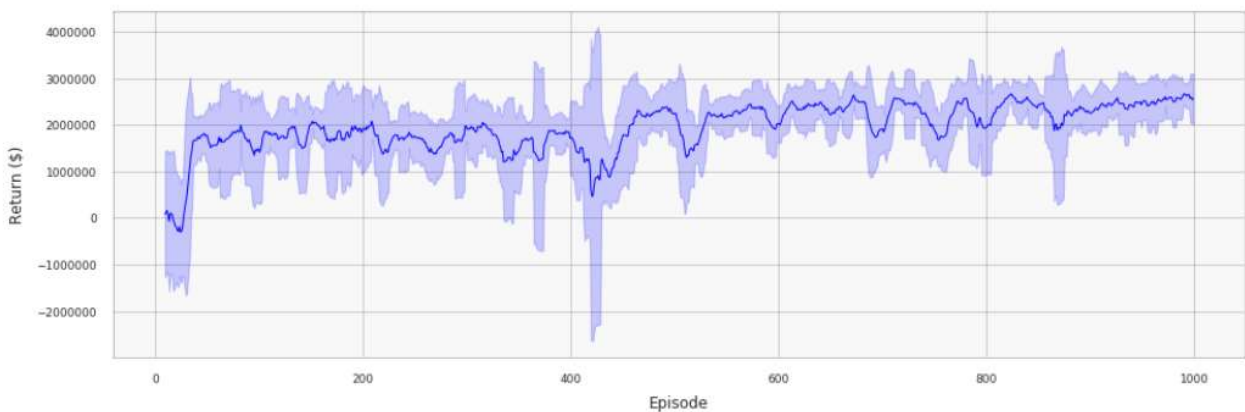
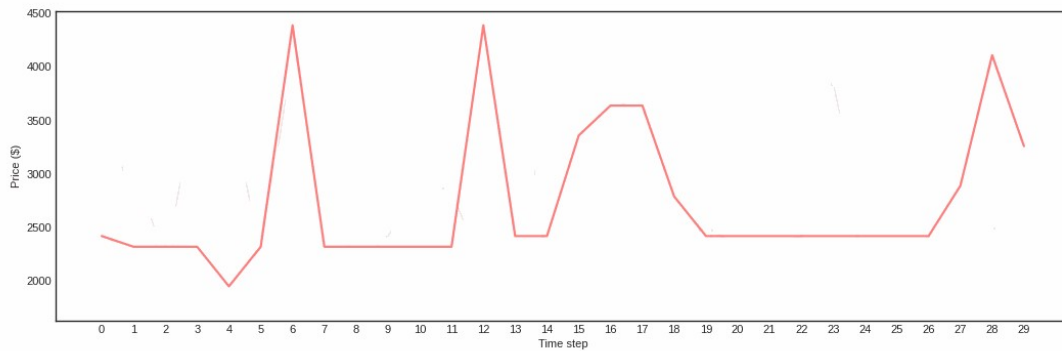
$$s_t = (p_{t-1}, p_{t-2}, \dots, p_0, 0, \dots) \mid (0, \dots, 1, \dots, 0)$$

- state: 상품 가격의 종류와 해당 시점에서 그 상품의 가격에 해당될경우 1 아니면 0 으로 가정
-여기서 상품 가격의 종류는 DP 적용 가능 대상 품목 테이블에 있는 upper_bound와 lower_bound를 사용하여 100원 단위 혹은 200원 단위등 상품 가격의 범위에 맞춰 끝자리가 80원으로 맞춰져 들어 갑니다.

Ex. [1180.0, 1480.0, 1280.0, 1580.0, 1380.0, 1680.0] (상품의 가격 범위에따라 T, action 종류등이 유동적으로 변함)

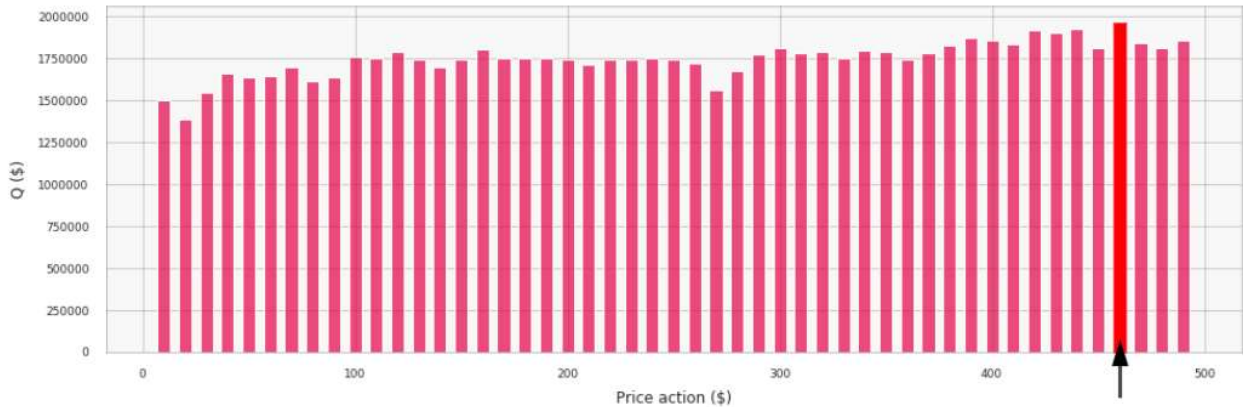
- action: 상품 가격의 종류(해당 시점에서 Q-Values 값이 가장 높은 상품이 action으로 선택됨) (강화학습 알고리즘은 일반적인 DL/ML 방법과 다릅니다)
- reward: 해당 시점의 price 후보와 다음 시점에서 미리 선택된 price를 통해 계산된 상품이익금액(profit: (가격-원가)*판매량)
- 앞서 정의한 것들에 의거하여 상품별로 학습 시작!

(DQN 및 알고리즘에 대한 자세한 설명은 추후에 디테일하게 설명드리도록 하겠습니다)



3.결과 산출

- T 시점을 가장 최근으로, 가격 종류 후보를 잘 학습된 모델에 넣어 Q-value 값이 가장 높은 Price를 optimal price로 제시합니다.



ii) 구동시간 & 기타 요건 및 필요사항

- 그렇다면 현실로 돌아와서... 모델 학습에 소요되는 시간은?
- DQN은 기본적으로 DeepLearning을 사용함으로 GPU가 필요하나 현재 Airflow는 GPU가 없으며 CPU학습으로 시간이 더 오래걸리고 점포별로 학습 대상 SKU가 다르므로 **점포-SKU별로 모두 학습**을 진행해야함입니다.
- 현재 진행중인 부분은 DP 적용 가능 대상 품목 테이블에서 Target으로 선정한 rank 상위 200개를 상품별로 새로 학습 후 결과를 저장하는 방식으로 진행됩니다.
- 즉, 200개 상품을 1개씩 학습-inference 하는 과정을 거치고 있으며 1개 상품마다 대략 30초 정도의 시간이 소요됩니다. (200개 기준 1시간 30분 내외) 최대 적용 가능하다는 가정하에 1000개를 학습한다면 학습시간 최대 9시간 소요, **물리적으로 시간 불가 개선 필요**
- 이러한 방식을 Airflow 상에서 새벽 5시에 학습하도록 설정하였으며 점포마다 병렬 처리를 통해 진행합니다.

	prdt_cd	upper_bound	lower_bound	uprice_daily	ucost_daily	str_cd	optimal_price	pred_sale_qty	pred_sale_profit	pred_sale_amt	k	a	b	d_o
0	8801155974046	4080	3130	4780	3130	1022	3780.0	11.0	7150.0	41580.0	0.000000	38.809830	0.467190	108.639774
1	8801007737089	6480	2533	4320	2533	1022	3880.0	92.0	124200.0	356960.0	0.553538	956.031991	13.405379	4650.312276
2	8801094691592	5000	3370	4980	3370	1022	3780.0	2.0	820.0	7560.0	0.021069	4.995070	0.001609	91.416395

→ 즉, 현재 200개 상품에 12개 점포에 한해서는 최대 2시간이면 예측결과값이 나올 수 있으나, 오전에 Airflow가 구동되는 파일이 많으므로 이는 더 늦어질 수 있습니다.

*Airflow 상에서 병렬처리가 가능한 최대 점포수는 20개로, 점포가 확장되거나 상품 개수가 1000개까지 늘어난다면 학습에 소요되는 시간은 비약적으로 늘어날 수 있습니다.

- 모델 학습 시간 자체를 줄이기 위해 상품별로 학습하는 방법에서 중분류 & 소분류 별로 한번에 학습하는 방법등을 고려하고 있으며, 다만 현재 요건 정의등 우선 처리해야할 사항이 많아 모델 소요시간을 줄이는 작업은 현재 후순위로 진행될 예정입니다.

3. DP 일중매가변경 테이블 적재

위에서 설명한 모델들이 출력한 결과를 하나의 데이터셋으로 취합 후, 일중매가변경 인터페이스 명세서에 따라 다음과 같이 적재합니다.

- 인터페이스 정의서 : [_IF_v1.0.xlsx](#)
- GCP 저장 장소 : [smart-ruler-304409.team_datascience.dp_daily_price](#)
- 적재 시간 : 매일 오전 7시 이전까지
- 스키마 (키 컬럼은 굵게 표시)

컬럼ID	컬럼명	타입	입력값	예시
IF_ID	인터페이스ID	CHAR(7)	고유 인터페이스 ID(PRCR002)	'PRCR002'
IF_DATE_KEY	인터페이스일자KEY	CHAR(8)	적용일	'20220714'
IF_BAT_SEQNO	인터페이스배치순번	NUMBER(4)	당일 작업 순서	1
IF_DATA_SEQNO	인터페이스데이터순번	NUMBER(10)	라인 번호 (1 ~ 전체라인수)	123
UPRICE_APPY_DATE	매가적용일자	CHAR(8)	적용일	'20220714'
STR_CODE	점포코드	CHAR(4)	점포코드	'1038'
SKU_CODE	상품코드	CHAR(13)	상품코드	'8801005001533'
UPRICE	일중매가	NUMBER(11,2)	가격	11380
CRTN_DT	생성일시	DATE	생성시간 (파티션 키)	'2022-07-08 07:38:42.944549'

CRTN_ID	생성ID	VARCHAR2(13)	생성자 사번	'216173'
CHG_DT	변경일시	DATE	생성시간(수정 불가하므로 생성시간과 동일)	'2022-07-08 07:38:42.944549'
CHG_ID	변경ID	VARCHAR2(13)	생성자 사번	'216173'

• 적재 결과

영	IF_ID	IF_DATE_KEY	IF_BAT_SEQNO	IF_DATA_SEQNO	UPRICE_APPPY_DATE	STR_CODE	SKUL_CODE	UPRICE	CRTN_DT	CRTN_ID	CHG_DT	CHG_ID
1	PRCR002	20220714	1	123	20220714	1038	8801005001533	11380	2022-07-14 08:55:11.456469 UTC	216173	2022-07-14 08:55:11.456469 UTC	216173

상기 테이블을 활용하여 DevOps 챗터에서 일중매가변경 히스토리 테이블을 생성, ez1 인터페이스에 반영하면 DP 가격 산출 및 반영이 완료됩니다. 이후 매장에 가격이 배포 되고, 해당 가격을 기준으로 관측된 판매량을 통해 또다시 모델을 업데이트 해나가게 됩니다.