매출 패턴 기반 동일 상품 인식 및 공통 카테고리 체계 생성

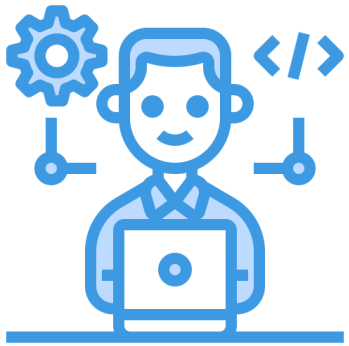
TEAM 3조 Retail Matcher

서창민, 신장운, 한원배

3조 Retail Matcher

서창민

조장



신장운

조원

한원배

조원

목차

1-1. 프로젝트 주제 소개

매출 패턴 기반 동일 상품 인식 및 동일 상품 카테고리 체계 생성

각 마트는 각각의 상품 카테고리를 가지고 상품도 각기 다른 코드와 상품명을 사용하기 때문에 여러 마트의 동일 상품이나 상품군을 비교하는 작업은 쉽지 않습니다.

공산품의 경우에는 바코드를 기반으로 어느 정도 매칭이 가능하나 그마저도 동일한 품목의 다른 브랜드 상품의 경우에 유사한 상품임에도 불구하고 비교하기가 쉽지 않습니다.

이러한 문제를 해결하기 위해, 상품들이 공통으로 가지고 있는 소비 패턴이라는 정보를 활용하여 동일한 상품, 품목, 카테고리로 연결 해줄 수 있는 로직을 구현하는 것이 목표입니다.

1-2. 데이터 설명

gds_0

	CO_CD	STR_CD	PLU_CD	GDS_NM	GDS_CLSS_CD	GDS_TP_NM	VEN_CD
0	1001294	12018	1	배송비	809999.0	일반	0
1	1001294	12018	2	사은행사쿠폰	999999.0	배송료	0
2	1001294	12018	91770	코지유리물병	93331.0	일반	0
3	1001294	12018	91770	크지물병	91701.0	일반	0
4	1001294	12018	50390427	POM석류	12579.0	일반	0
...	...	...	...	...	...	...	...
457561	1002117	363004	9556437014116	망고푸딩	22799.0	일반	106
457562	1002117	363004	9556995200068	매직쿠키/블루베리	22799.0	일반	55
457563	1002117	363004	9556995200075	매직쿠키/딸기잼	22799.0	일반	55
457564	1002117	363004	9654955092	돌)황도슬라이스	20599.0	일반	14004
457565	1002117	363004	9788087049099	탑쿨토시(일반형)	110499.0	일반	44

457566 rows × 7 columns

gdsclss_0

	CO_CD	STR_CD	GDS_CLSS_CD	GDS_CLSS_NM	GDS_CLSS_LVL_TP	UP_GDS_CLSS_CD	GDS_CLSS_GRP_TP
0	1001294	12018	1	식품	1	0	1
1	1001294	12018	101	가공농산물	2	1	1
2	1001294	12018	10151	가공농산물	3	101	1
3	1001294	12018	10152	김치류	3	101	1
4	1001294	12018	10153	소스	3	101	1
...	...	...	...	...	...	...	...
3587	1002201	0	210101	국산담배	3	2101	1
3588	1002201	0	210102	수입담배	3	2101	1
3589	1002201	0	22	애견용품	1	0	1
3590	1002201	0	2201	애견용품	2	22	1
3591	1002201	0	220101	애견용품	3	2201	1

3592 rows × 7 columns

RETAIL& INSIGHT

1-2. 데이터 설명


```
tr_list_0
```

	SAL_DT	STR_CD	POS_NO	TRAN_NO	SALE_KIND_TP_NM	SALE_RTN_SGN	REAL_SAL_DT	REAL_SAL_TTM	PAY_AMT	MSHP_ID	ORN_SA
0	20201220	12018	1	25	매출	1	20201220	105406.0	21880.0	NaN	
1	20201220	12018	1	26	매출	-1	20201220	105504.0	21880.0	NaN	20201
2	20201220	12018	1	27	매출	1	20201220	105703.0	21880.0	012018_0	
3	20201220	12018	1	28	매출	-1	20201220	105800.0	21880.0	012018_0	20201
4	20201220	12018	1	34	매출	1	20201220	110647.0	20300.0	NaN	
...	...	...	...	...	...	...	...	...	...	...	
5898794	20220929	363004	2	531	영업종료	1	20220929	184536.0	0.0	NaN	
5898795	20220929	363004	3	21	영업종료	1	20220929	184749.0	0.0	NaN	
5898796	20220930	363004	1	311	영업종료	1	20220930	200933.0	0.0	NaN	
5898797	20220930	363004	2	685	영업종료	1	20220930	203635.0	0.0	NaN	
5898798	20220930	363004	3	26	영업종료	1	20220930	201415.0	0.0	NaN	

5898799 rows × 13 columns

```
tr_list_0
```

O	TRAN_NO	SALE_KIND_TP_NM	SALE_RTN_SGN	REAL_SAL_DT	REAL_SAL_TTM	PAY_AMT	MSHP_ID	ORN_SAL_DT	ORN_POS_NO	ORN_TRAN_NO
1	25	매출	1	20201220	105406.0	21880.0	NaN	NaN	NaN	NaN
1	26	매출	-1	20201220	105504.0	21880.0	NaN	20201220.0	1	25.0
1	27	매출	1	20201220	105703.0	21880.0	012018_0	NaN	NaN	NaN
1	28	매출	-1	20201220	105800.0	21880.0	012018_0	20201220.0	1	27.0
1	34	매출	1	20201220	110647.0	20300.0	NaN	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...
2	531	영업종료	1	20220929	184536.0	0.0	NaN	NaN	NaN	NaN
3	21	영업종료	1	20220929	184749.0	0.0	NaN	NaN	NaN	NaN
1	311	영업종료	1	20220930	200933.0	0.0	NaN	NaN	NaN	NaN
2	685	영업종료	1	20220930	203635.0	0.0	NaN	NaN	NaN	NaN
3	26	영업종료	1	20220930	201415.0	0.0	NaN	NaN	NaN	NaN

1-2. 데이터 설명

tr_dtl_list_0

	SAL_DT	STR_CD	POS_NO	TRAN_NO	SEQ_NO	DESI_CNCL_TP_NM	SALE_RTN_SGN	PLU_CD	GDS_CLSS_CD	SALE_QTY	SALE_P
0	20201220	12018	1	24	1	전체취소	1	2908990004313	30101	1.0	15(
1	20201220	12018	1	25	1	정상판매	1	2908990003422	30499	1.0	189(
2	20201220	12018	1	25	2	정상판매	1	2908990008472	30702	1.0	29(
3	20201220	12018	1	26	1	정상판매	-1	2908990003422	30499	1.0	189(
4	20201220	12018	1	26	2	정상판매	-1	2908990008472	30702	1.0	29(
...	...	...	...	...	...	...	...	...	...	...	
24084721	20220930	363004	R1	1	2	정상판매	1	8801007980706	21699	1.0	69(
24084722	20220930	363004	R1	1	3	정상판매	1	8801043014830	20299	1.0	44(
24084723	20220930	363004	R1	1	4	정상판매	1	8801043015684	20299	1.0	62(
24084724	20220930	363004	R1	1	5	정상판매	1	8801073210776	20299	1.0	12(
24084725	20220930	363004	R1	1	10001	정상판매	1	499800000001	NaN	1.0	43(

24084726 rows × 20 columns

tr_dtl_list_0

SALE_PRC	SALE_AMT	SGDS_DC_TP_NM	SGDS_DC_PRC	SUBTOT_DC_AMT	EVT_DC_AMT	CPON_DC_AMT	SCALES_GDS_TP_NM	GDS_CPON_DC_AM
1500.0	1500.0	정상	0.0	0.0	0.0	0.0	일반	(
18900.0	18900.0	정상	0.0	0.0	0.0	0.0	일반	(
2980.0	2980.0	정상	0.0	0.0	0.0	0.0	일반	(
18900.0	18900.0	정상	0.0	0.0	0.0	0.0	일반	(
2980.0	2980.0	정상	0.0	0.0	0.0	0.0	일반	(
...	...	...	...	...	...	...	...	
6980.0	6500.0	할인특매	480.0	0.0	0.0	0.0	일반	(
4480.0	4480.0	할인특매	0.0	0.0	0.0	0.0	일반	(
6200.0	5800.0	할인특매	400.0	0.0	0.0	0.0	일반	(
1200.0	1200.0	할인특매	0.0	0.0	0.0	0.0	일반	(
4380.0	4380.0	정상	0.0	0.0	0.0	0.0	일반	(

1-2. 데이터 설명

- 데이터 수집 전: 홈플러스나 이마트같은 큰 마트들이 아닌 지역 내의 작은 마트들이 모여있는 데이터이기에 해당 마트들이 묶이면서 카테고리 통합이 제대로 잘 이뤄져 있지 않기 때문에 이를 통합해주는 과정이 필요 할 것이라 예상
- 실제로 데이터를 받아 본 뒤로는 카테고리 뿐만이 아니라 상품과(상품명) 매출데이터 자체의 비교가 한 파일 안에서 불가능했기 때문에 프로젝트 초기에 이 기준을 세워주는 것 부터 시작
- 데이터프레임 간 병합을 진행해 주면서 묶이는 많은 컬럼들간의 관계를 보면서 많은 인사이트를 얻었고 이를 바탕으로 많은 시도를 진행

2. 분류 결과


```
df['GDS_NM']= df['GDS_NM'].str.replace(pat=r'[^ㅱw]',repl=r' ',regex=True)
df['GDS_NM']
```

```
0                    배송비
1                 감자2kg 박스
2                   델몬트 망고
3              삼립 56시간저온숙성숙
4                 크라운산도스윗밀크
                  ...
22758180        필라델피아크림200
22758181         위스카스 오션피쉬
22758182     시저 연어와쇠고기 1Kg
22758183     시저 연어와쇠고기 1Kg
22758184     시저 연어와쇠고기 1Kg
Name: GDS_NM, Length: 22758185, dtype: object
```

```
df['GDS_NM'] = df['GDS_NM'].str.replace(" ", "")
```

```
df = df.astype({'SALE_QTY' : 'int'})
```

```
df = df.astype({'SALE_PRC' : 'int'})
```

```
df = df.astype({'SALE_AMT' : 'int'})
```

```
feature_df.columns
```

```
Index(['hourly_sale_ratio_1', 'hourly_sale_ratio_10',
       'hourly_sale_ratio_10_group', 'hourly_sale_ratio_11',
       'hourly_sale_ratio_11_group', 'hourly_sale_ratio_12',
       'hourly_sale_ratio_12_group', 'hourly_sale_ratio_13',
       'hourly_sale_ratio_13_group', 'hourly_sale_ratio_14',
       'hourly_sale_ratio_14_group', 'hourly_sale_ratio_15',
       'hourly_sale_ratio_15_group', 'hourly_sale_ratio_16',
       'hourly_sale_ratio_16_group', 'hourly_sale_ratio_17',
       'hourly_sale_ratio_17_group', 'hourly_sale_ratio_18',
       'hourly_sale_ratio_18_group', 'hourly_sale_ratio_19',
       'hourly_sale_ratio_19_group', 'hourly_sale_ratio_1_group',
       'hourly_sale_ratio_2', 'hourly_sale_ratio_20',
       'hourly_sale_ratio_20_group', 'hourly_sale_ratio_21',
       'hourly_sale_ratio_21_group', 'hourly_sale_ratio_22',
       'hourly_sale_ratio_22_group', 'hourly_sale_ratio_23',
       'hourly_sale_ratio_23_group', 'hourly_sale_ratio_2_group',
       'hourly_sale_ratio_3', 'hourly_sale_ratio_3_group',
       'hourly_sale_ratio_4', 'hourly_sale_ratio_4_group',
       'hourly_sale_ratio_5', 'hourly_sale_ratio_5_group',
       'hourly_sale_ratio_6', 'hourly_sale_ratio_6_group',
       'hourly_sale_ratio_7', 'hourly_sale_ratio_7_group',
       'hourly_sale_ratio_8', 'hourly_sale_ratio_8_group',
       'hourly_sale_ratio_9', 'hourly_sale_ratio_9_group', 'sale_prc_mean',
       'sale_prc_mean_group', 'sale_prc_median', 'sale_prc_median_group',
       'is_imported_group', 'is_factory_group', 'is_fresh_group',
       'is_imported', 'is_factory', 'is_fresh', 'name_sim', 'label'],
      dtype='object')
```

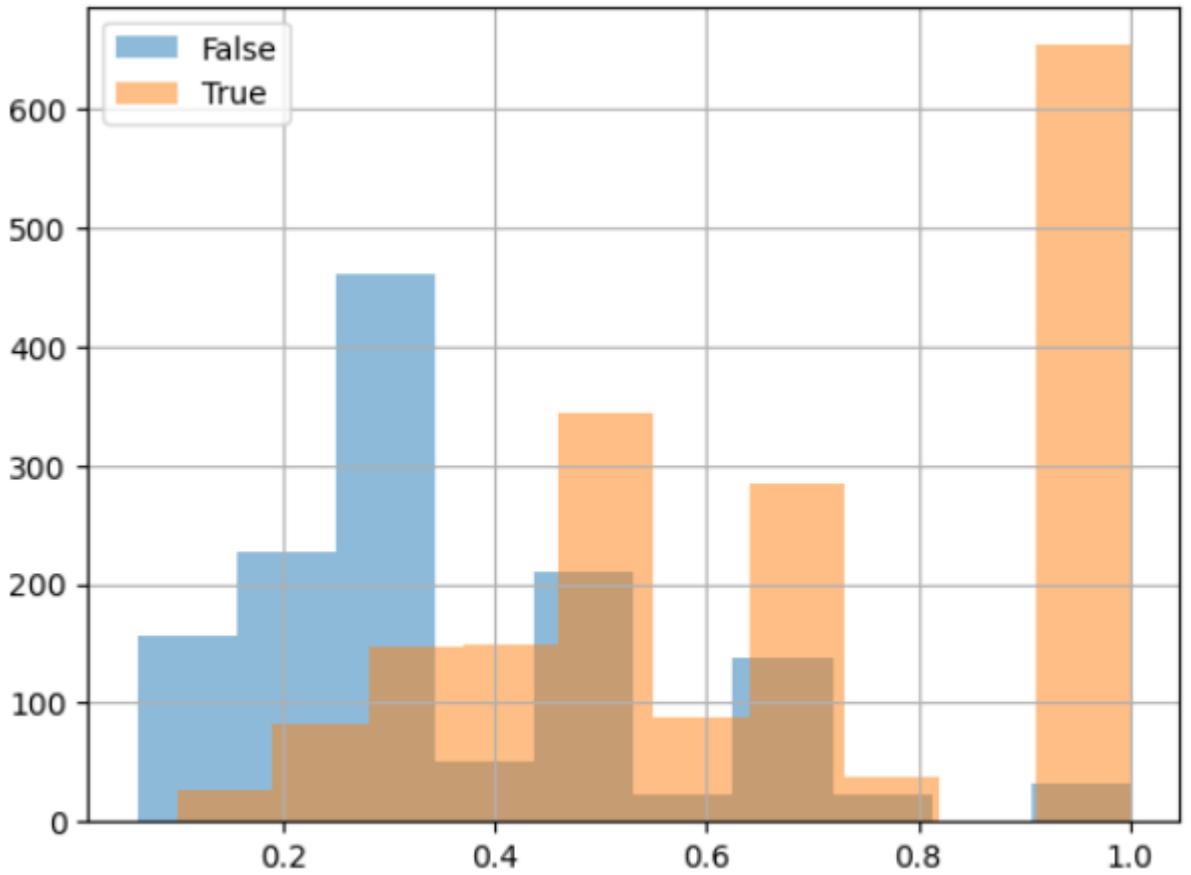
2. 분류 결과

카테고리 커버리지: 11.52%
매출액 커버리지: 10.82%
판매량 커버리지: 10.36%

카테고리 커버리지: 27.24%
매출액 커버리지: 24.02%
판매량 커버리지: 24.25%

카테고리 커버리지: 37.77%
매출액 커버리지: 37.55%
판매량 커버리지: 33.22%

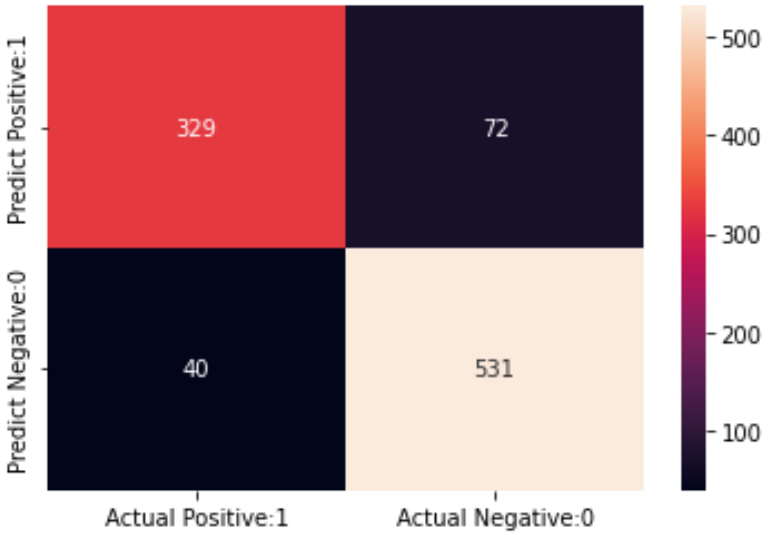
카테고리 커버리지: 53.79%
매출액 커버리지: 55.79%
판매량 커버리지: 51.18%

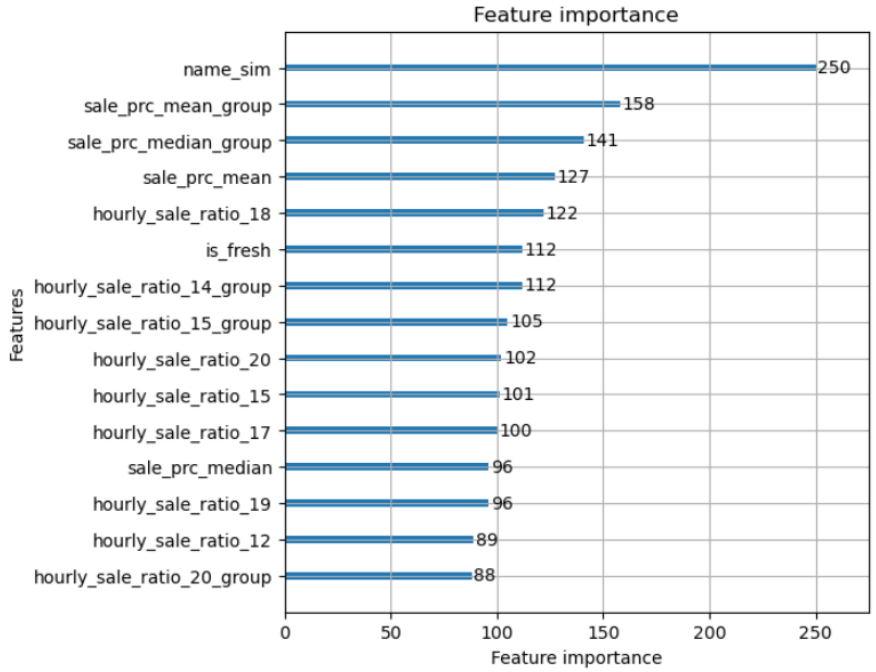


2. 분류 결과

- BayesianOptimization + lightGBM을 이용해 최적의 parameter 값 찾아 학습시켜 모델 생성 후 test set으로 예측

```
Training set score : 0.9065
Test set score : 0.8848
```

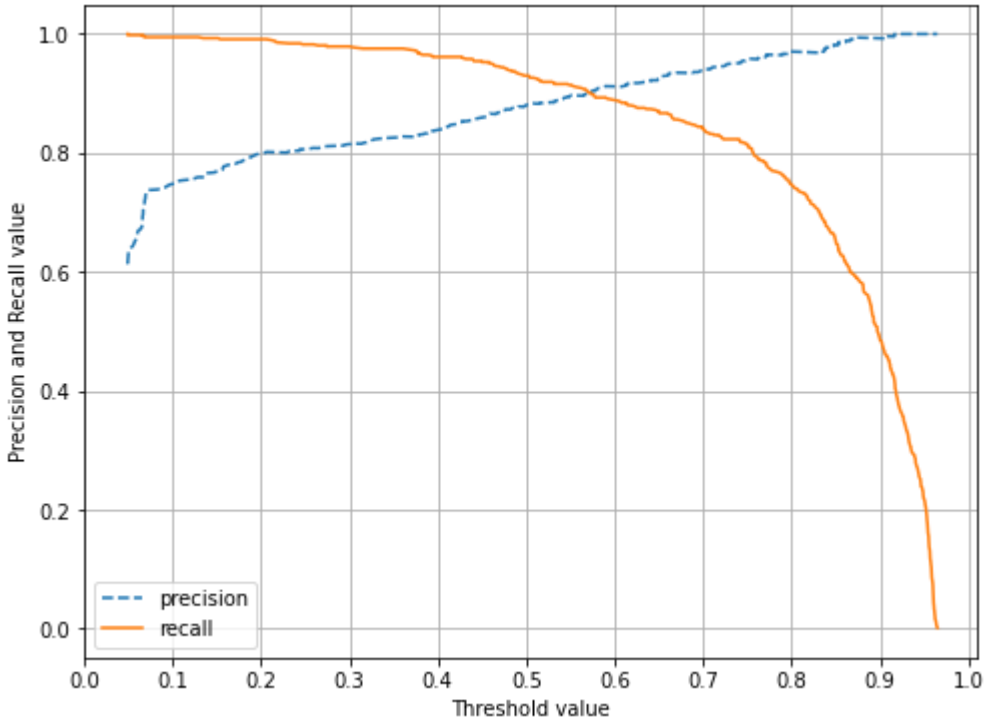





```
accuracy_score: 0.8847736625514403
precision_score: 0.8805970149253731
recall_score: 0.9299474605954466
```

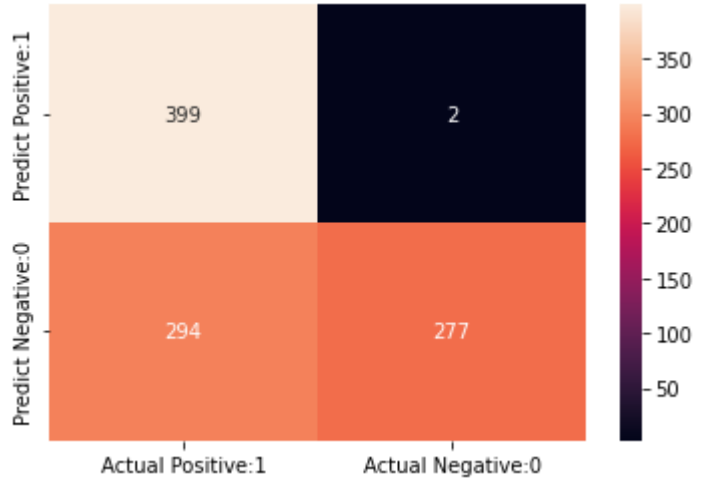
- y_test와 y_pred(model에 넣고 학습시킨 x_test) 을 Confusion Matrix를 통해 성능 평가

2. 분류 결과



- sklearn의 precision_recall_curve()를 사용해 임계값 변화에 따른 정밀도와 재현율의 관계를 파악
  → 0.8~0.9 사이에서 recall이 급격하게 감소하는 것을 확인할 수 있으며, precision도 1에 가까워지는 것을 확인

- threshold: 0.87
  → precision_score: 99%



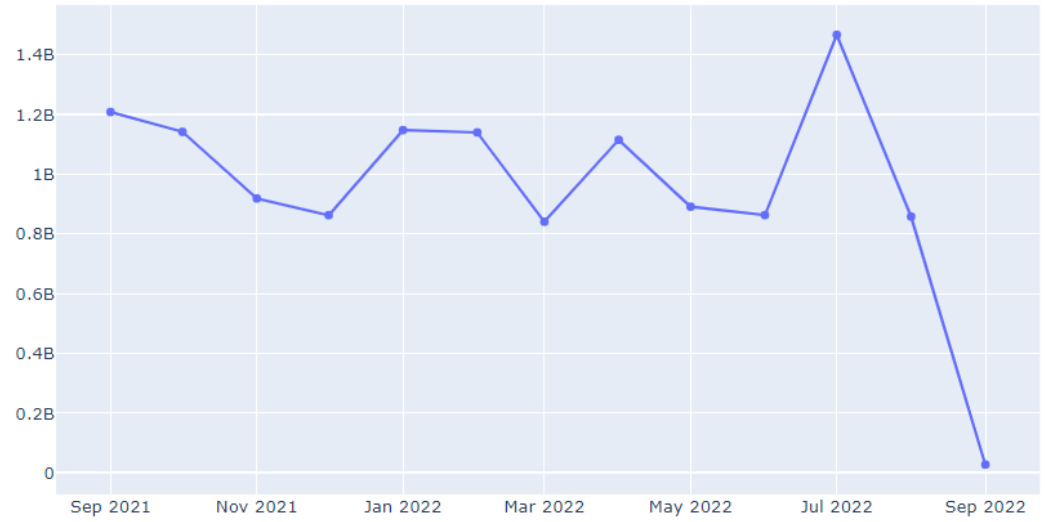

```
[[399   2]
 [294 277]]
accuracy_score: 0.6954732510288066
precision_score: 0.992831541218638
recall_score: 0.4851138353765324
```

3. 얻어낸 인사이트 및 기대효과

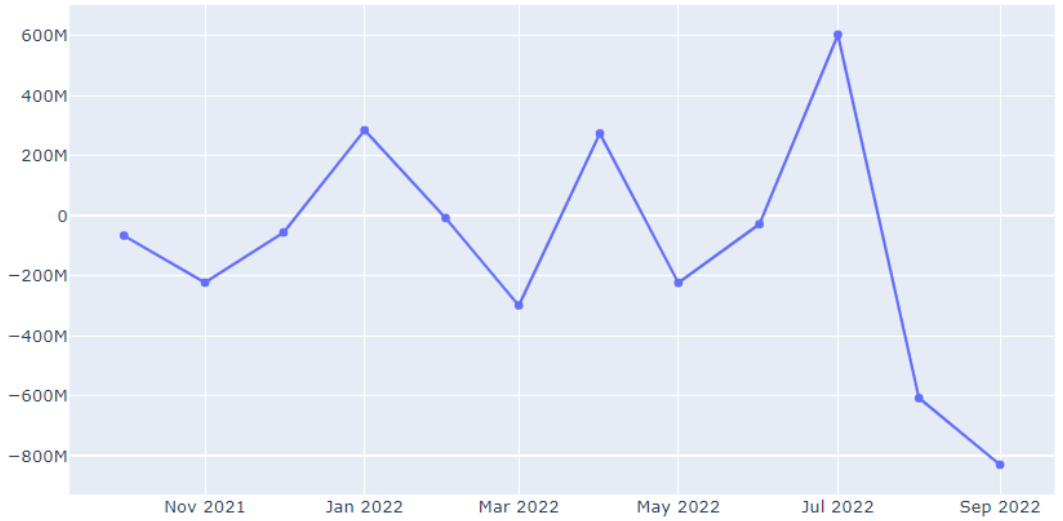
서창민	신장운	한원배
고객군 분석 매출패턴 분석 매출 예측	Apriori 알고리즘 Text categorization K-Modes clustering	카이제곱 동질성 검정 DTW(Dynamic Time Warping) 알고리즘

3. 월별 매출 통계 및 매출 차이 비교

Montly Sales

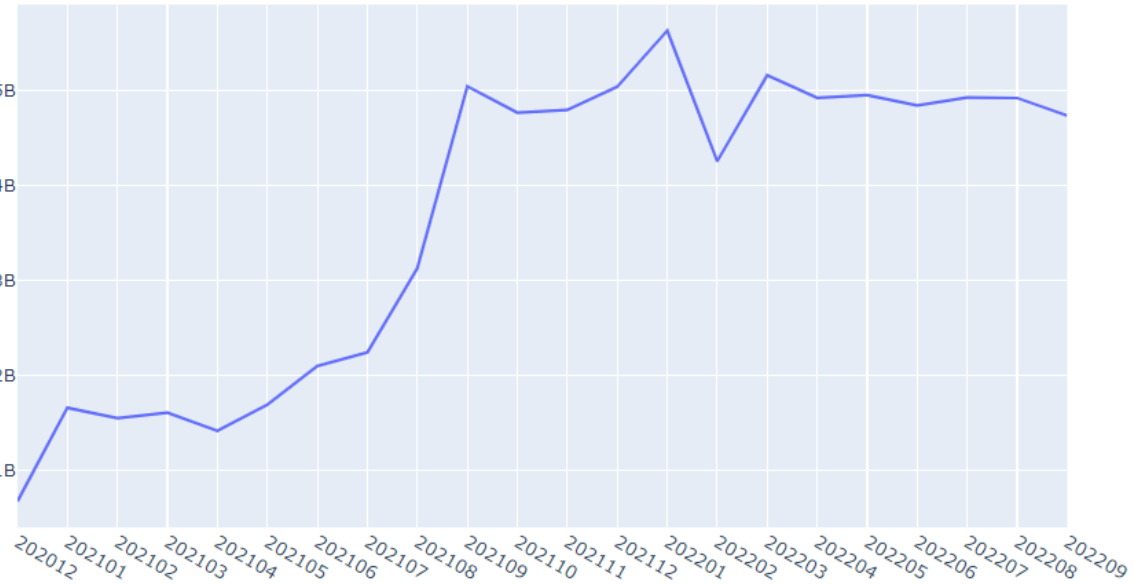


Montly Sales Diff

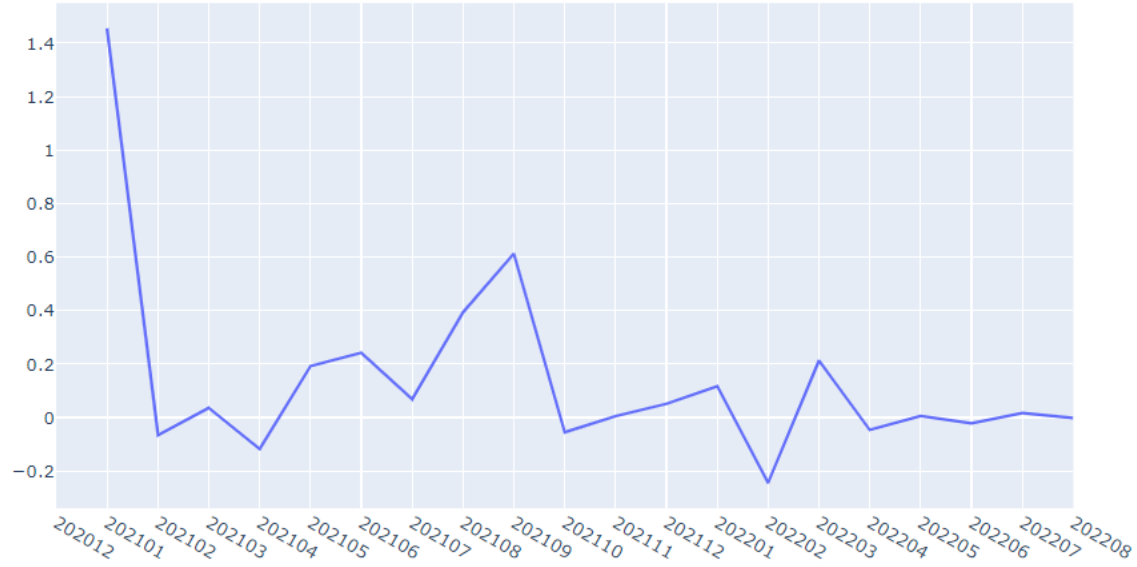


3. 월별 수익 및 성장률 그래프

Montly Revenue

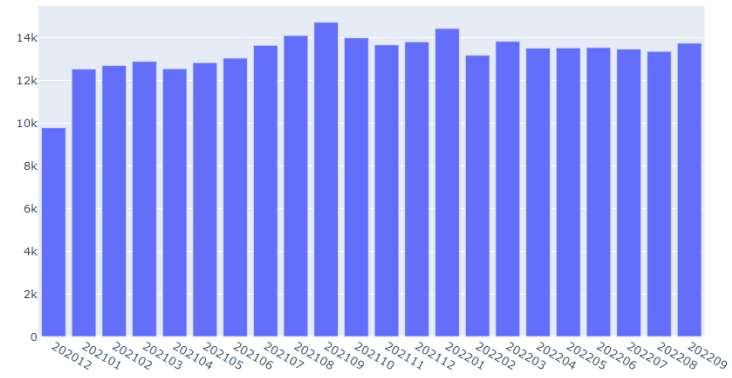


Montly Growth Rate

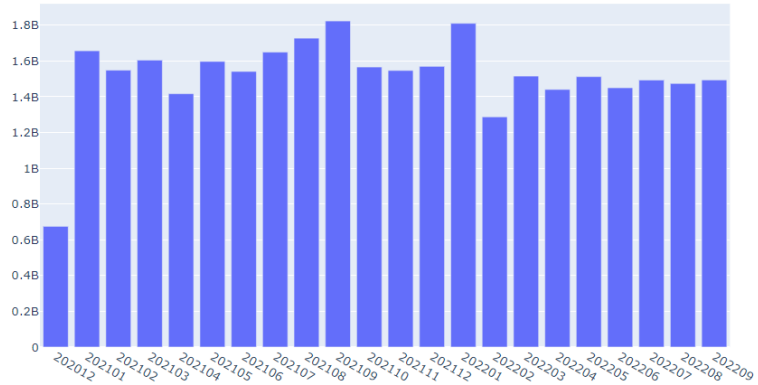


3. 고객군 분석

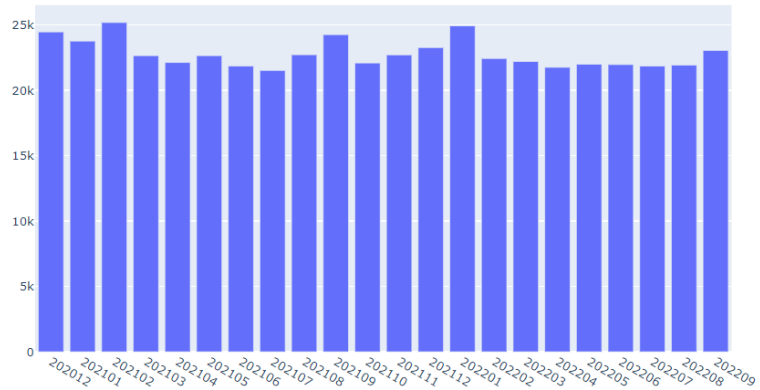
Monthly Active Customers



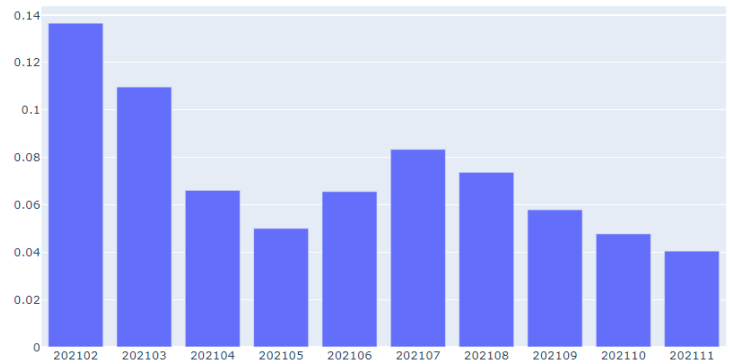
Monthly Total # of Order



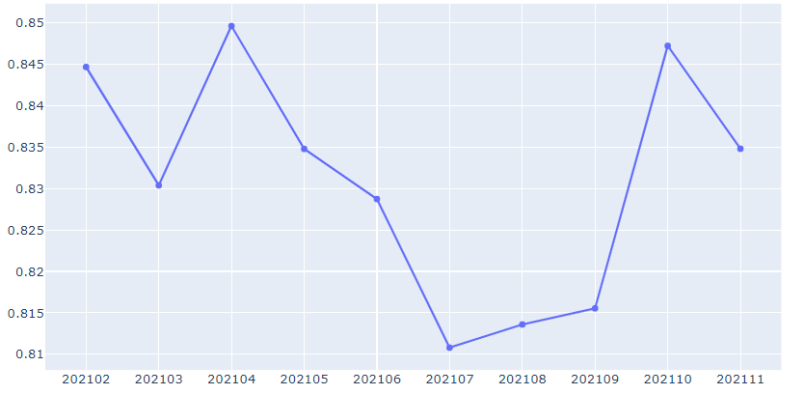
Monthly Order Average



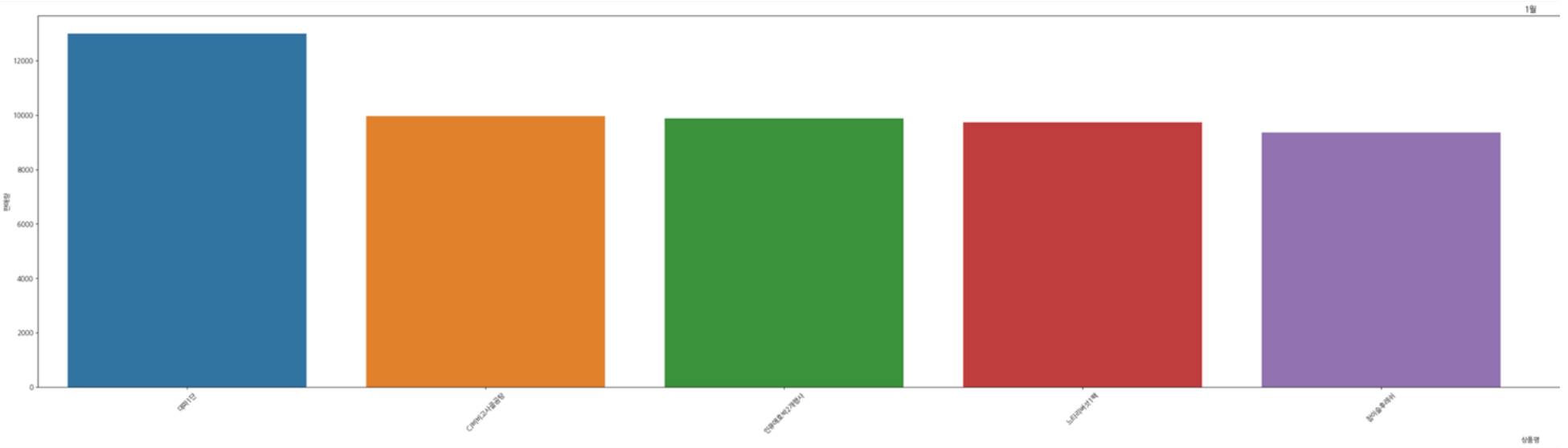
New Customer Ratio

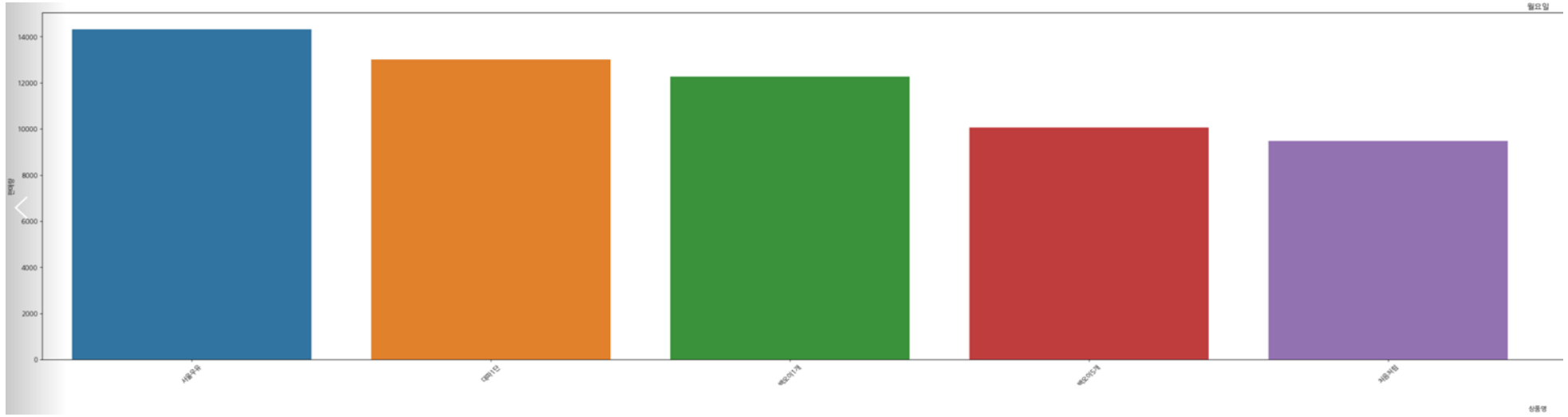


Monthly Retention Rate

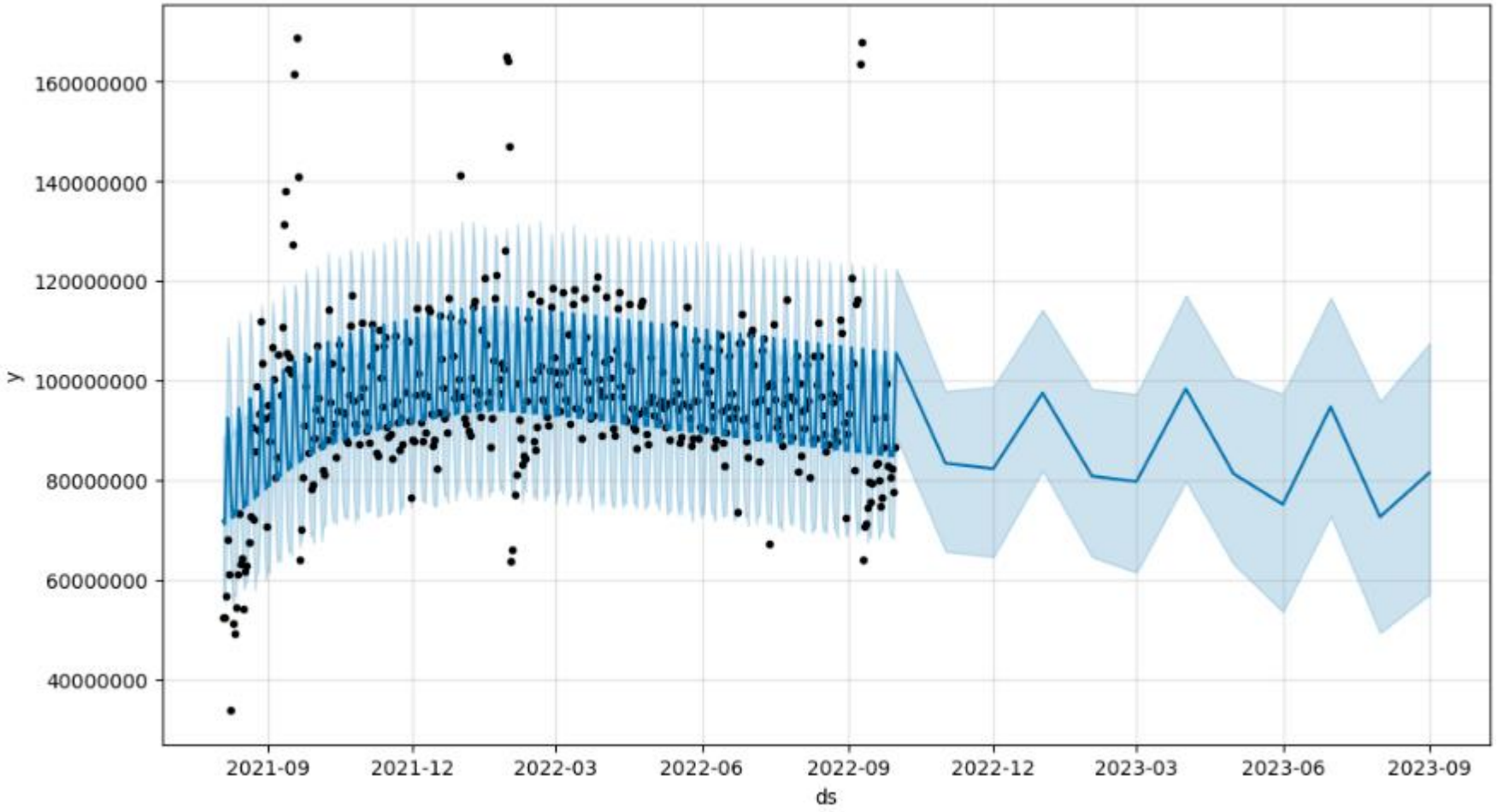


3. 상위 매출 항목 비교

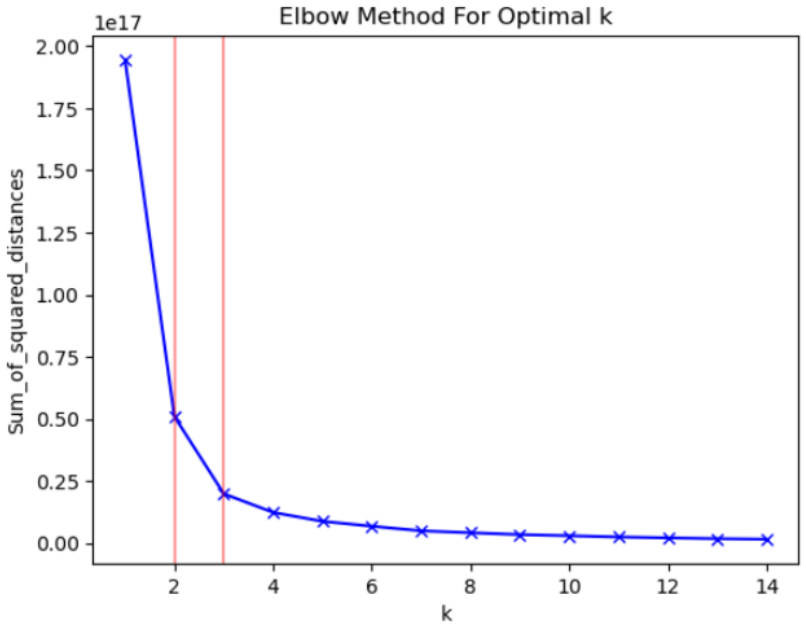

1월
12000
10000
8000
6000
4000
2000
0
상품명

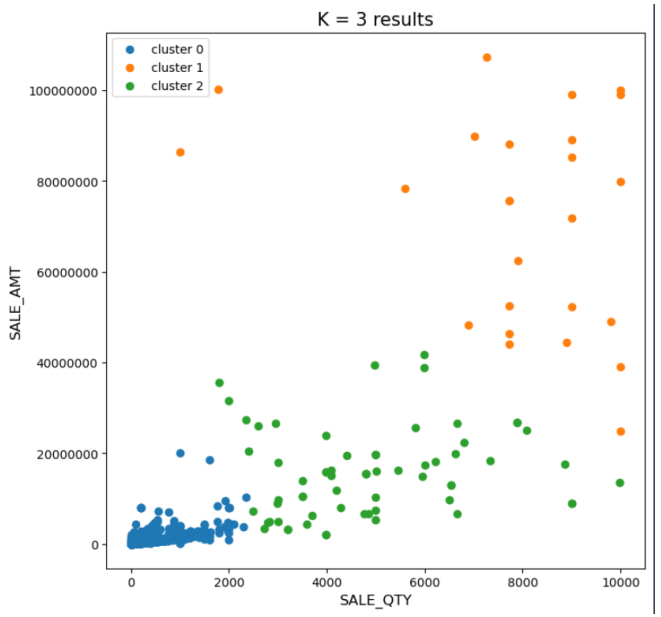

월요일
14000
12000
10000
8000
6000
4000
2000
0
상품명

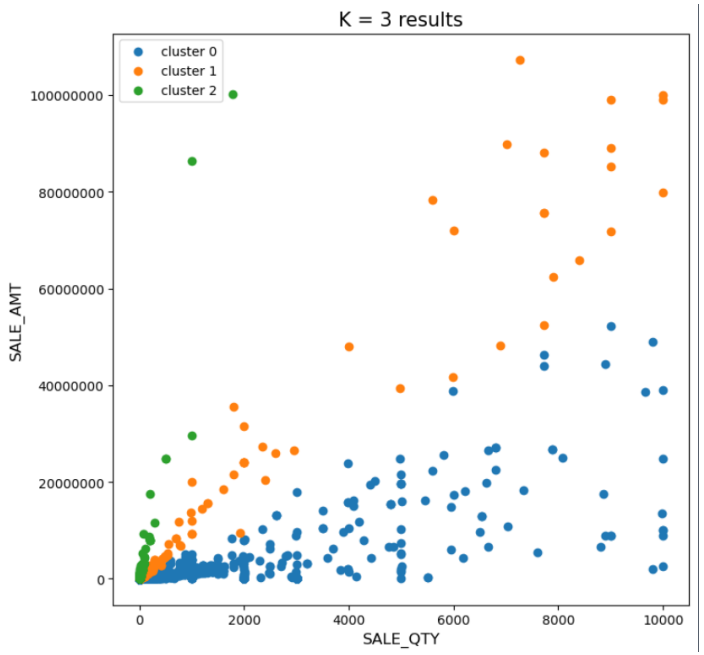
3.Prophet Algorithm



3.Clustering



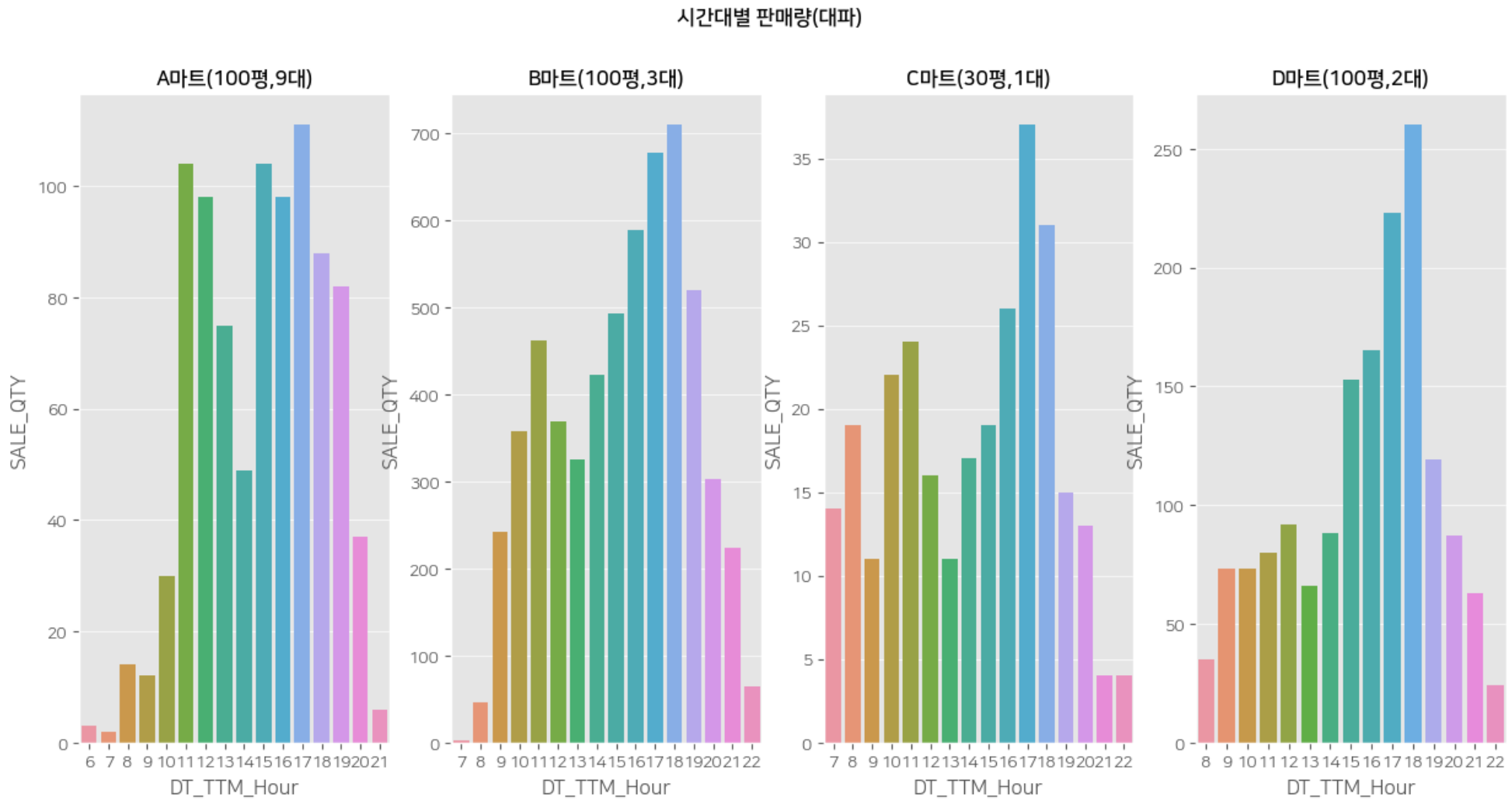




EDA 진행: 시각화

- ‘동일한 카테고리 안에 물품 간에는 유사한 매출패턴이 있다’라는 가설 검증하기 위해서는 ① 계절성이 없고(제철 식품으로 소비 X), ② 고객들이 주기적으로 소비하는 신선식품이라는 강한 가정을 충족하고 있는 물품, ③ 신선식품 중에서 제일 거래건수가 많은 물품으로 EDA를 시도함
- ①, ②, ③에 충족하는 상품으로 대파를 선택함

EDA 진행: 시각화



- A마트, B마트, D마트(대형마트), C마트(소형마트)의 시간대별 판매량 패턴에서 **시계열적 유사함**을 발견

EDA 진행: 카이제곱 동질성 검정

$$X^2 = \sum_{i=1}^{k} \frac{(O_i - E_i)^2}{E_i}$$

O_i = observed frequency counts in each category
E_i = expected frequency counts in each category
k = number of categories

- 관찰되는 빈도가 기대되는 빈도와 유의하게 다른지를 검증
- 자료가 빈도로 기록될 수 없거나 빈도로 전환될 수 없다면 카이제곱검정은 사용할 수 없음 → 범주형 자료에 사용

EDA 진행: 카이제곱 동질성 검정

시간대	A마트	C마트
7	11720.0	19000.0
8	36870.0	50000.0
9	103830.0	57000.0
10	170100.0	61000.0
11	225820.0	49000.0
12	228270.0	31000.0
13	273080.0	35000.0
14	288160.0	36000.0
15	365760.0	61000.0
16	377870.0	80000.0
17	290180.0	91500.0
18	224010.0	64000.0
19	198300.0	22000.0
20	65960.0	28000.0
21	35320.0	15000.0

chi2: 223329.51913499006, p-value: 0.0, df: 14
p-value가 유의수준 0.05보다 낮으므로 귀무가설을 기각합니다 → 집단 간의 분포가 동일하지 않음

- 마트 간의 Scale 차이에 고려가 안 됨

EDA 진행: 카이제곱 동질성 검정

시간대	A마트	C마트
7	9.369052	9.852194
8	10.515153	10.819778
9	11.550510	10.950807
10	12.044142	11.018629
11	12.327494	10.799576
12	12.338284	10.341742
13	12.517520	10.463103
14	12.571271	10.491274
15	12.809733	11.018629
16	12.842306	11.289782
17	12.578257	11.424094
18	12.319446	11.066638
19	12.197536	9.998798
20	11.096804	10.239960
21	10.472205	9.615805

chi2: 0.39437935988495926, p-value: 0.9999999980638956, df: 14
p-value가 유의수준 0.05보다 높으므로 귀무가설을 기각하지 못합니다 → 집단 간의 분포가 동일함

- 마트 간의 Scale 차이를 조정하기 위해서 로그 변환을 함
  - 시계열 분포(시간대)의 동질함을 확인함

EDA 진행: 카이제곱 동질성 검정

월	A마트	B마트	C마트
5	18503600.0	3235300.0	426000.0
6	26333040.0	6272200.0	223000.0
7	61938240.0	6314500.0	631000.0
8	11503540.0	1660900.0	218000.0
9	3481420.0	39000.0	37000.0

chi2: 2782522.0004169317, p-value: 0.0, df: 8
p-value가 유의수준 0.05보다 낮으므로 귀무가설을 기각합니다 → 집단 간의 분포가 동일하지 않음

월	A마트	B마트	C마트
5	16.733476	14.989632	12.962195
6	17.086335	15.651638	12.314927
7	17.941648	15.658359	13.355061
8	16.258165	14.322870	12.292250
9	15.062951	10.571317	10.518673

chi2: 0.3500421057045774, p-value: 0.9999659967407711, df: 8
p-value가 유의수준 0.05보다 높으므로 귀무가설을 기각하지 못합니다 → 집단 간의 분포가 동일함

- **계절성이 존재**하는 수박 상품에서도 동질성을 확인

DTW (Dynamic Time Warping)

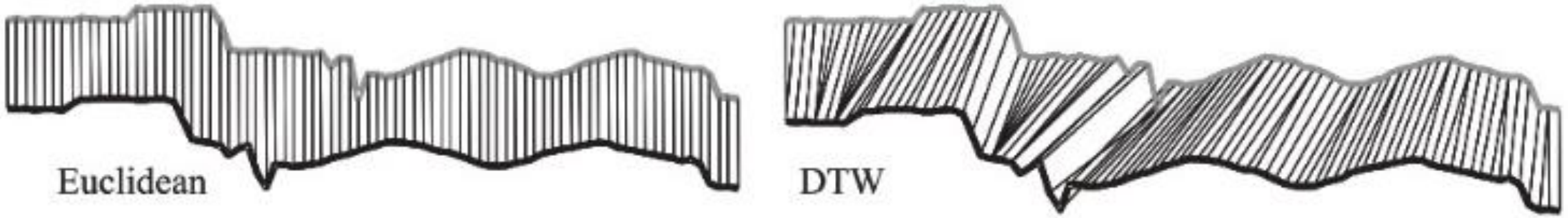


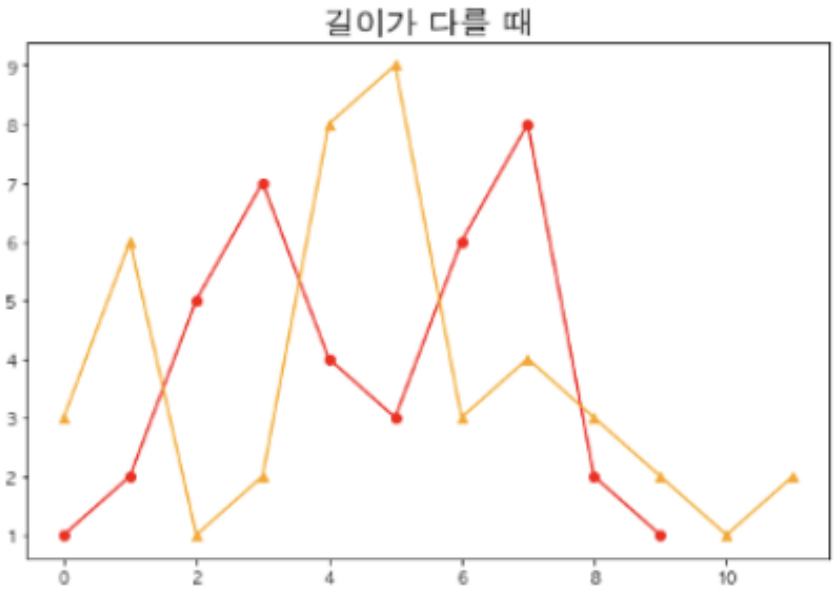
Fig. 1. Note that, while the two time series have an overall similar shape, they are not aligned in the time axis. Euclidean distance, which assumes the i^{th} point in one sequence is aligned with the i^{th} point in the other, will produce a pessimistic dissimilarity measure. The nonlinear dynamic time warped alignment allows a more intuitive distance measure to be calculated

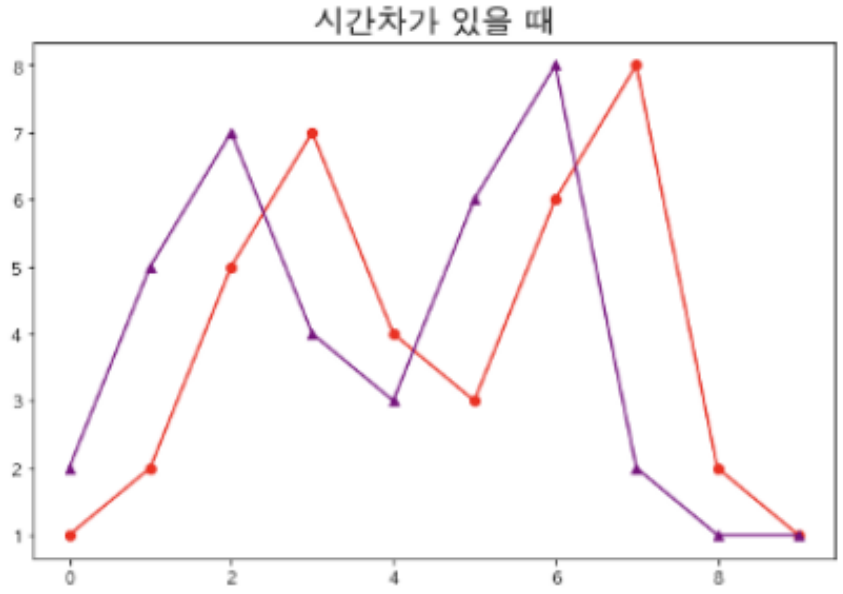
- DTW는 시퀀스를 **시간의 길이를 고려하지 않고 인식**할 수 있는 Metric으로서 시계열의 유사도를 측정함

DTW (Dynamic Time Warping)

DTW를 사용하는 이유?

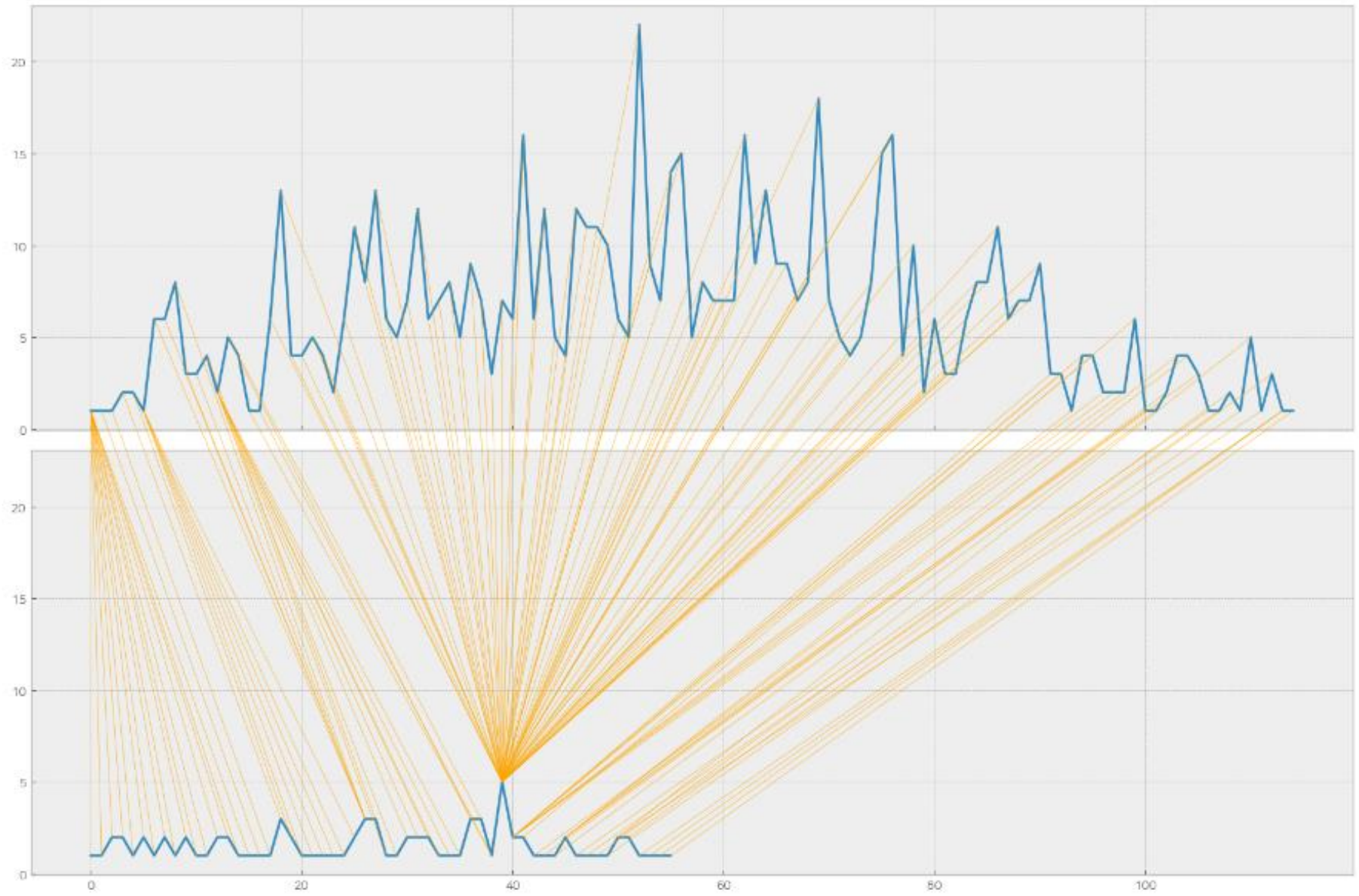
- 두개의 시계열 데이터 길이가 달라도 유사도 비교 가능
- 비슷한 패턴이지만 시간차가 있는 경우(shift 발생) 유사도 비교 가능





DTW (Dynamic Time Warping)

B마트와 C마트의 dtw 값은 44.7101778122163l5 입니다



4-1. 기획발표 이후의 개선

1. 기획안 발표 때 받은 피드백 요약, 피드백을 통해 수정 및 개발한 사항

- 기획안 발표 이후 강사님과도 피드백을 진행하였고 기업 담당자님과 멘토님이 함께 미팅을 진행한 이후로 멘토님이 진행해주시는 방향에 따라 프로젝트를 진행하였습니다.

1. 멘토링을 통해 받은 피드백 정리, 피드백을 통해 수정 및 개발한 사항

- 멘토님과의 많은 피드백을 통해 최종적인 결과물을 만들 수 있었습니다. 단순한 방향성 제시가 아닌 통계 분석의 인사이트와 코드 단 에서부터의 세세한 피드백과 어떻게 하면 더 효율적이고 생산적이며 좋은 코드를 짤 수 있는지에 대한 피드백, 관련된 지식까지 쉽게 설명해주셔서 만족할만한 결과물을 낼 수 있었던 것 같습니다.

4-2. Discussion

프로젝트 결과물 자체로 봤을 때는 첫 미팅 당시에 생각했던 그림과 현재를 비교해 봤을 때의 차이는 있지만 실무 활용 가능 정도나 완성도적인 측면에서 봤을 때 기업의 문제를 해결 하는데 있어서 도움을 줄 수 있을 것 입니다.

실제 현업 데이터를 다뤄볼 수 있다는 경험은 개인 및 팀원의 성장에 있어서 아주 큰 도움을 주었고, 특히 비지니스 환경에서 중요시 여겨지는 시계열 데이터를 다룰 수 있어서 향후에 기업의 데이터를 만났을 때 큰 도움이 될 수 있을 거라고 생각합니다. 또한 프로젝트를 진행하며 클러스터링, DTW, Apriori 알고리즘 등 여러 가지 실험적인 시도를 해볼 수 있었습니다. 이러한 과정은 개인의 큰 성장과 팀원간의 협동력을 기를 수 있었던 값진 경험이였습니다

개선할 부분은 현재로써는 신선식품의 카테고리에 한정된 모델이었다는 점이 아쉽다고 생각합니다. 추후 공산품, 수입상품 기타상품로 확대하는 것이 최종적인 결과물이 될 것이라고 생각합니다.

4-3. 프로젝트 진행/개발환경







개발 환경

Language	Library	Tool
Python	Numpy Pandas Seaborn Prophet Sklearn LightGBM Scipy	VSCode Jupyter Notebook Nbviewer Slack GitHub Notion Google Colab

Q & A

YEAR-DREAM SCHOOL

감사합니다