

Kaggle Competition 도전기

YBIGTA 9기 박성남, 10기 김상현, 10기 김태한

Kaggle에서 주최하는 Instacart Market Basket Analysis를 도전하며 겪었던 일련의 과정을 공유합니다. 데이터 분석을 진행하는 과정에서 어떤 문제가 발생했고 어떻게 해결해나가는지 간접적으로 체험할 수 있는 시간이 되길 바랍니다.

Index

01

Introduction

02

EDA &
Preprocessing

03

Methodology

04

Resampling &
Model Tuning

05

Conclusion

1. Introduction

■ Kaggle이란?

- 최근 구글의 인수로 화제가 된 실리콘밸리 스타트업
- 데이터 분석가들의 링이라고 불린다
- 회사는 캐글에 데이터와 함께 과제를 내고, 데이터 사이언티스트들은 문제를 풀고 상금을 받는다
- 심심풀이로 공개한 데이터부터 상금 몇 십만 달러의 컴페티션까지 그 종류가 매우 다양하다

■ Instacart Market Basket Analysis Competition

- Instacart는 온라인 기반 농작물 배송 업체로 웹사이트와 스마트폰으로 주문을 하면 Instacart가 제품을 구매하여 배송해준다
- Kaggle에 사용자의 미래 구매 상품들을 예측하는 컴페티션을 개최하였다
- 현재 800명의 참가자가 있으며, 상금은 2만 달러이다

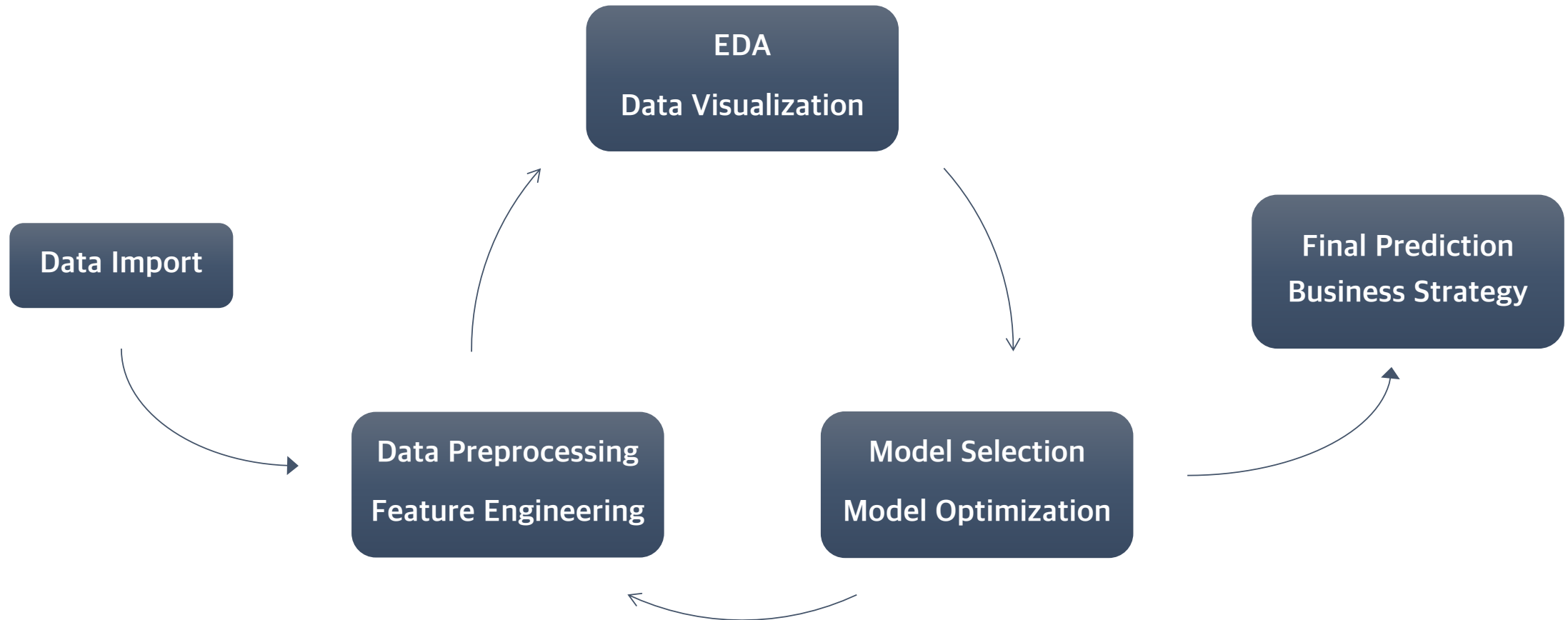
1. Introduction

■ 데이터 분석 과정

- 데이터 핸들링을 하기 쉽도록 데이터를 다듬는 전처리 과정
- 탐색적 자료 분석을 통해 데이터가 어느 특성을 가지고 있는지 다양하게 시각화하고 인사이트 찾아내기
- 찾아낸 인사이트를 바탕으로 추가 피처 생성 및 사용할 모델을 선정
- Train / Validate / Test Accuracy나 기타 Evaluation을 통해 모델의 성능을 평가
- 끊임없는 모델 최적화와 변수 생성 및 제거를 통한 모델의 성능을 개선

1. Introduction

■ 데이터 분석 과정



1. Introduction

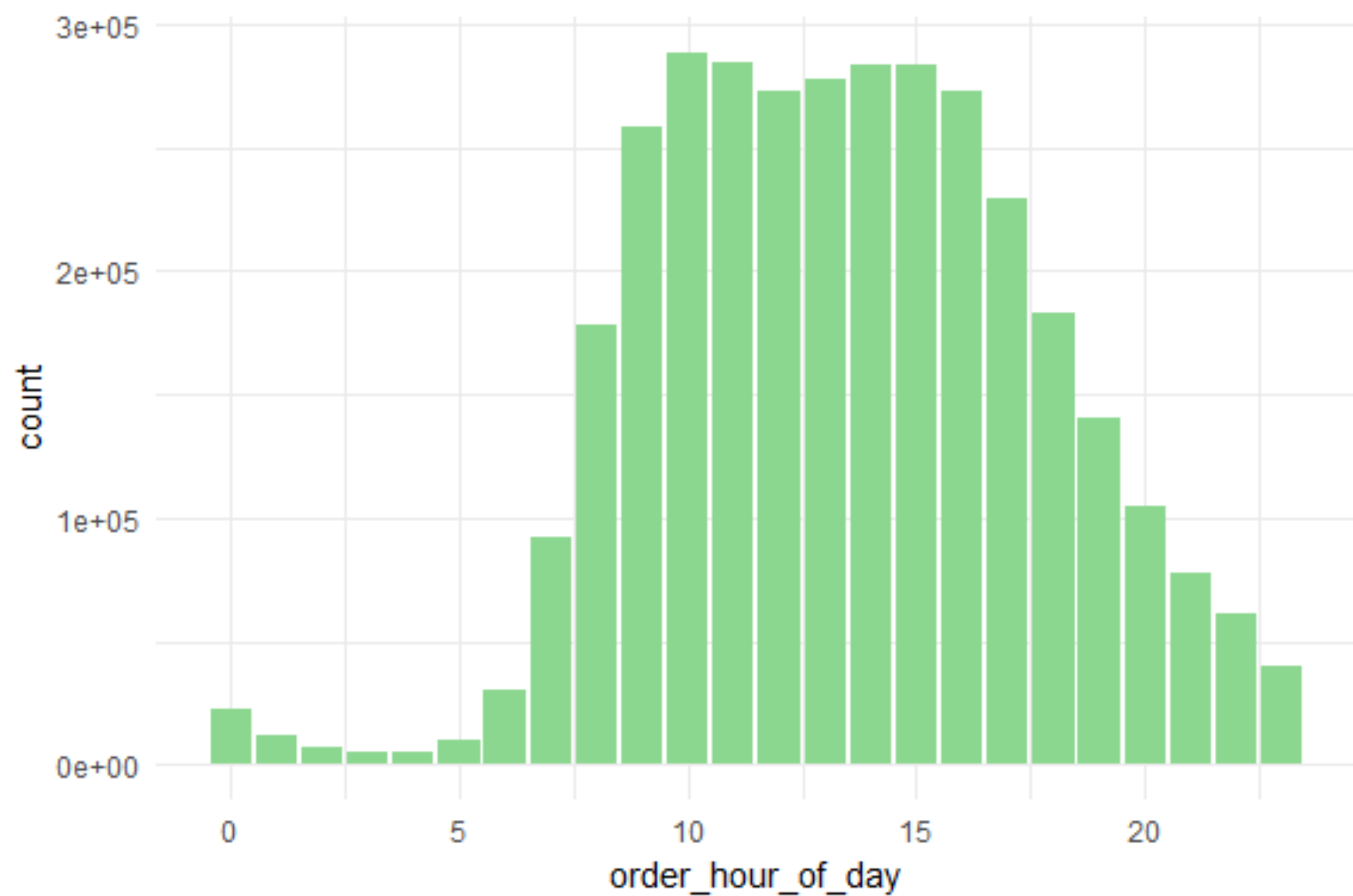
■ 데이터 구조

- 20만 여명의 사용자들이 5만 종류의 제품들을 구매한 330만 개의 주문 데이터로 구성
- 70%의 train 유저와 30%의 test 유저로 나뉘어져 있음
- 70%의 유저와 30%의 유저 모두 직전 구매 이전까지의 모든 주문에 대한 정보는 있음
- Test 유저는 마지막 주문이 언제 이뤄졌는지는 있지만 뭘 샀는 지에 대한 정보는 없음



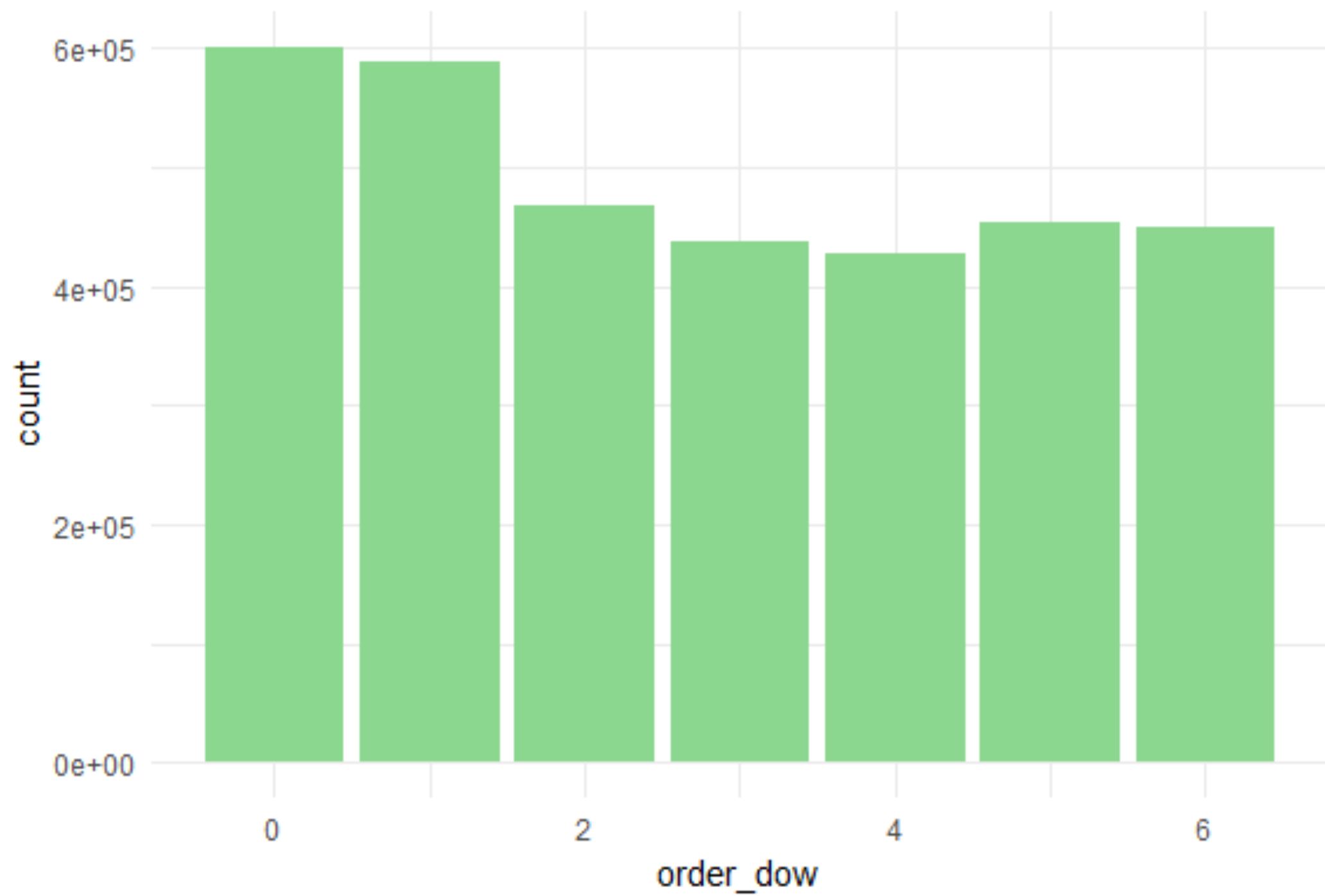
2. EDA & Preprocessing

- 시간에 따른 주문 수



2. EDA & Preprocessing

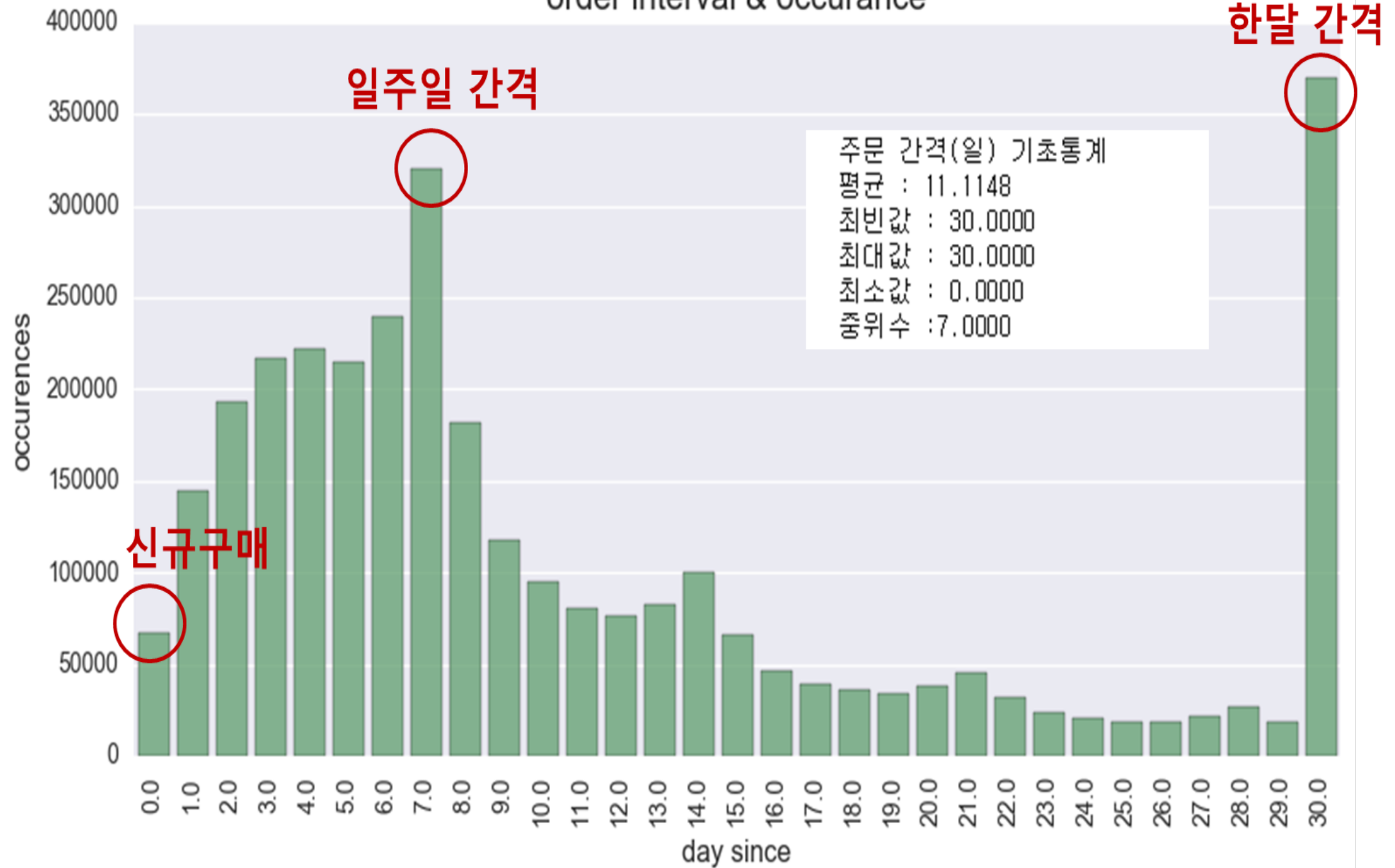
- 요일에 따른 주문 수



2. EDA & Preprocessing

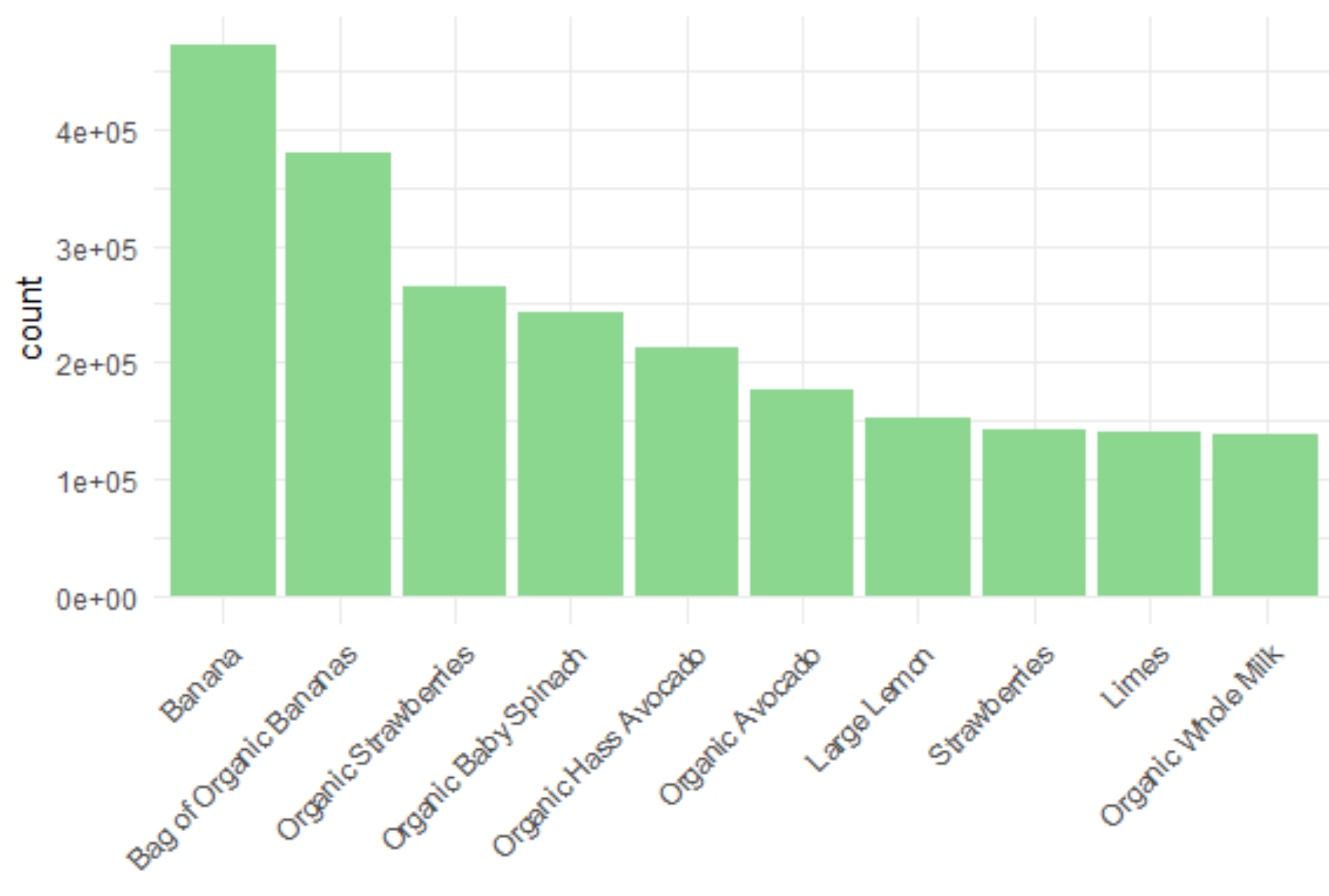
- 다음 구매까지의 시간 간격 분포

order interval & occurance



2. EDA & Preprocessing

- 가장 많이 구매하는 제품



2. EDA & Preprocessing

■ Feature Engineering

- 끊임없이 질문하고, 질문에 대한 답을 찾기 위해 새로운 feature를 만들고 시각화하여 확인한다

장바구니에 담은 순서와 재구매 여부의 관계?

주기적 사게 되는 제품들, 항상 묶어서 구매되는 제품들이 있지 않을까?

train에 속해있는 product들은 이전에 구매했던 제품인 경우가 많을까?

제품마다 재구매되기까지 걸리는 시간이 다르지 않을까?

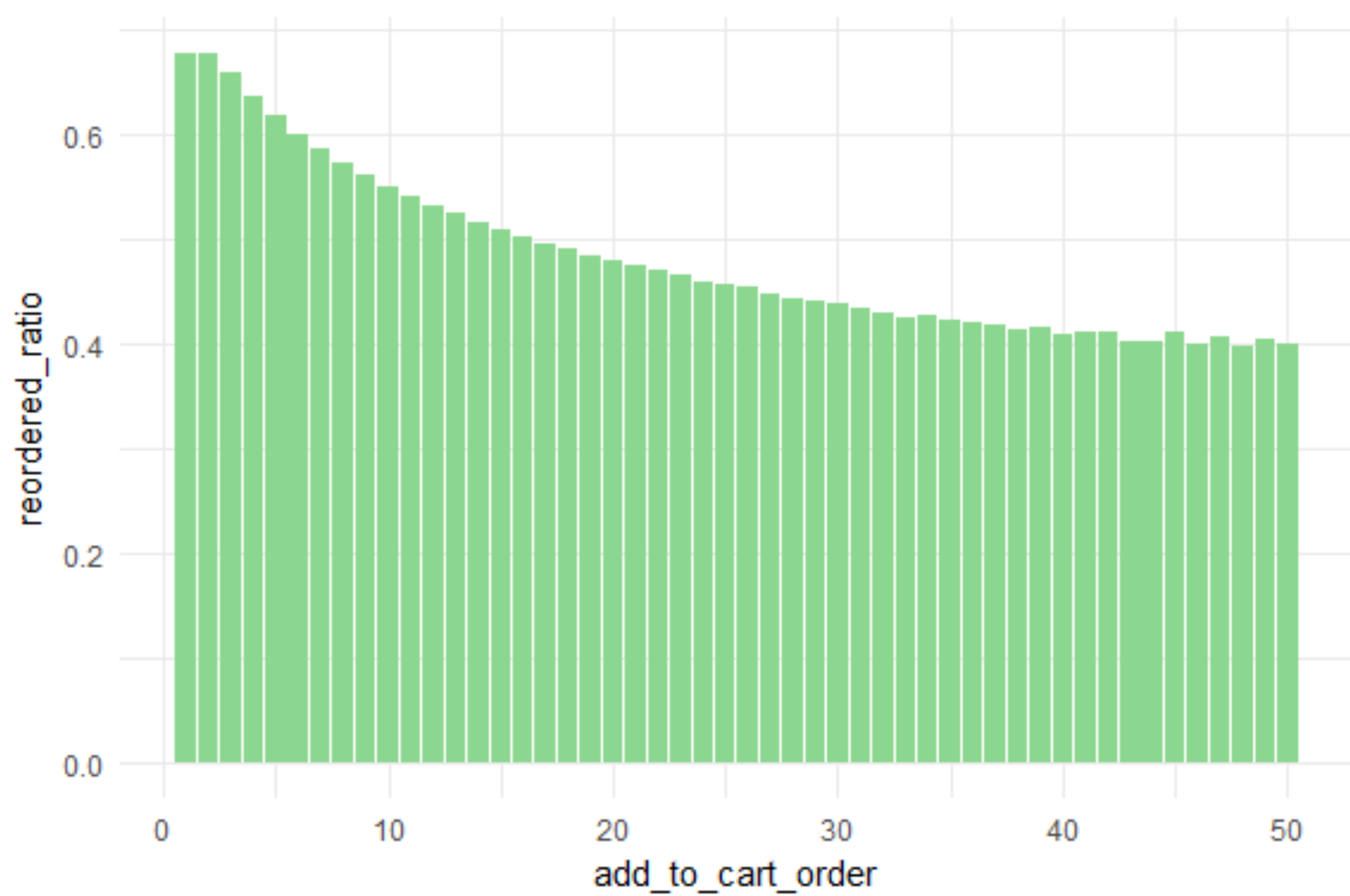
새롭게 구매되는 제품들은 어떤 특성이 있을까?

사용자별 구매 패턴을 한눈에 보고 싶다

특정 요일/시간대에 많이 구매되는 제품이 있을까?

2. EDA & Preprocessing

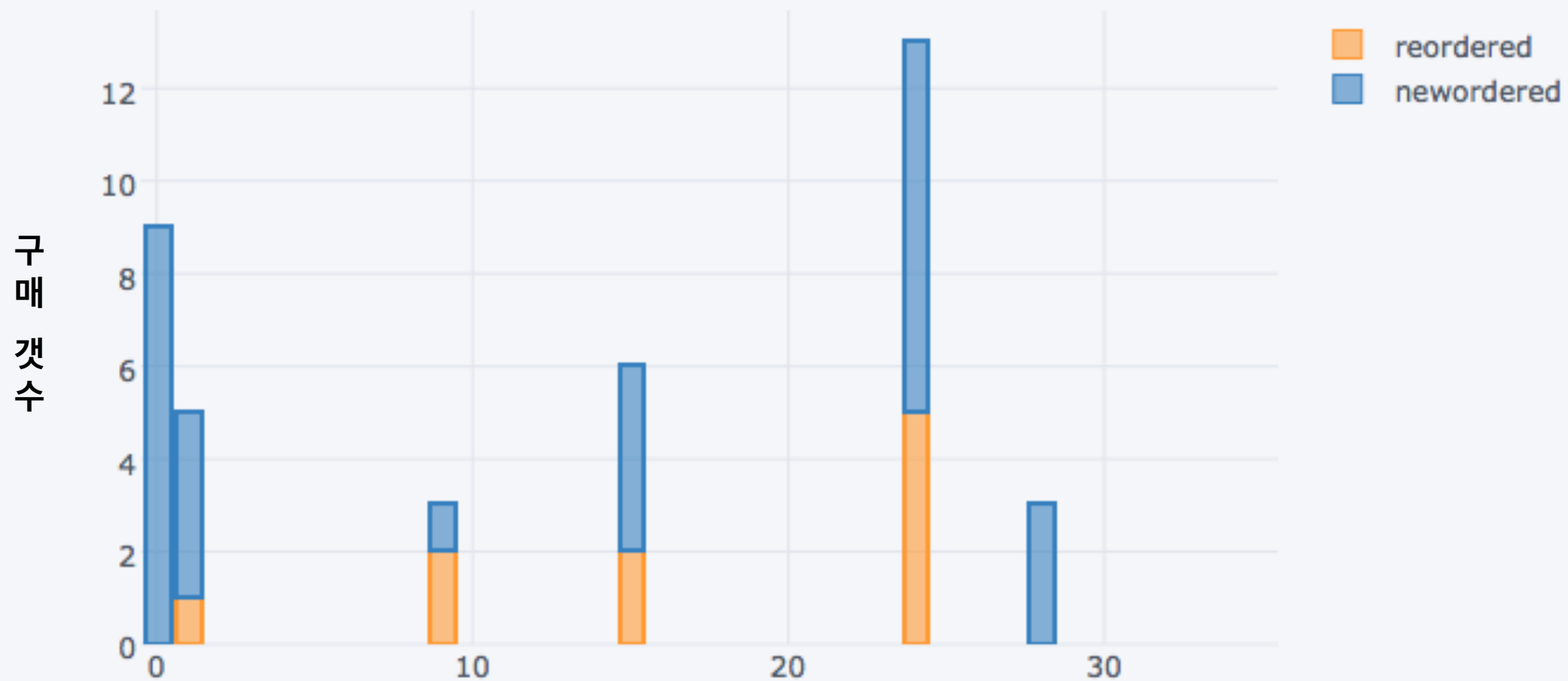
- 장바구니에 담은 순서와 reorder 여부의 관계



2. EDA & Preprocessing

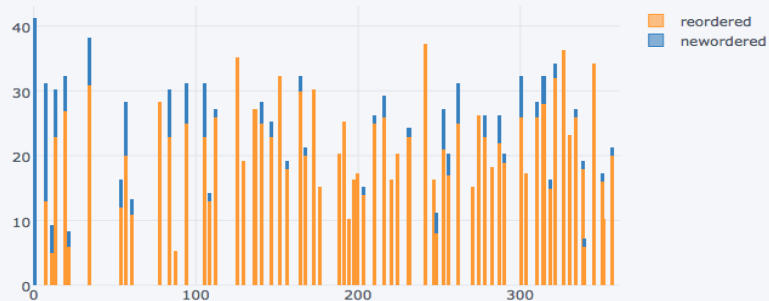
- Feature Engineering

User Pattern : 18

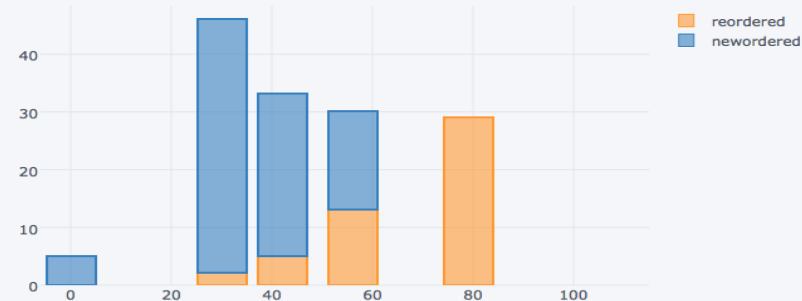


첫 구매로부터 지난 날짜

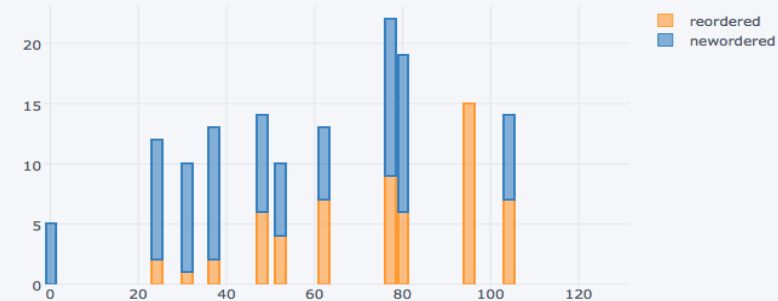
User Pattern : 120572



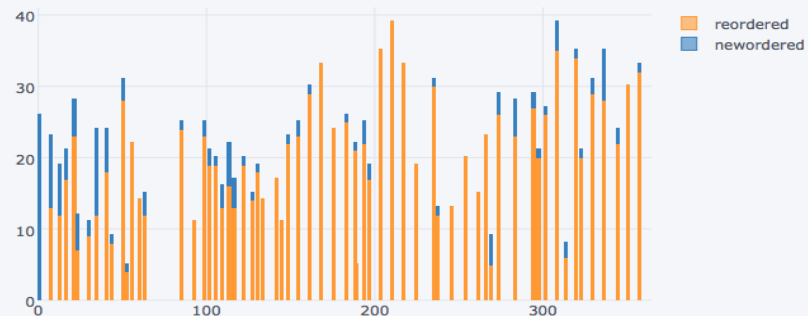
User Pattern : 10



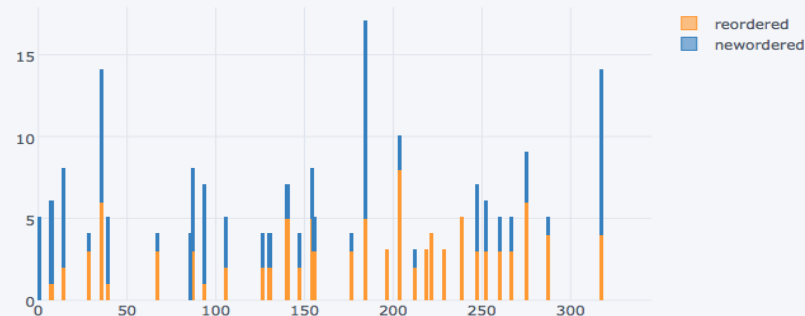
User Pattern : 43



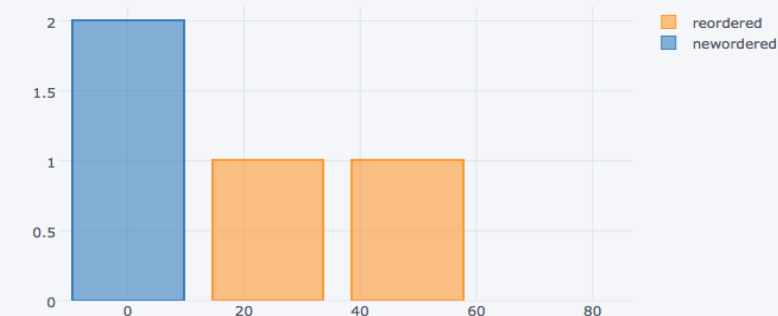
User Pattern : 153204



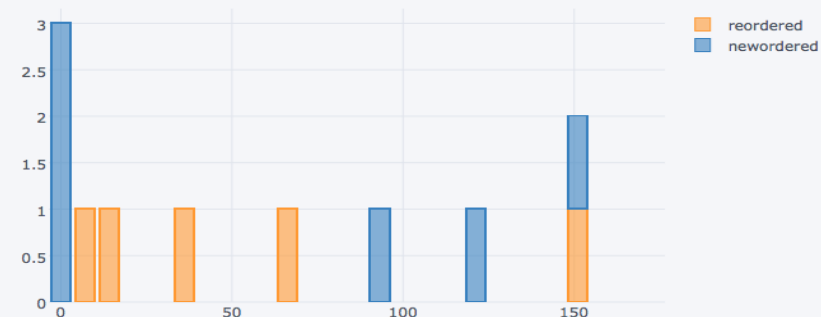
User Pattern : 21



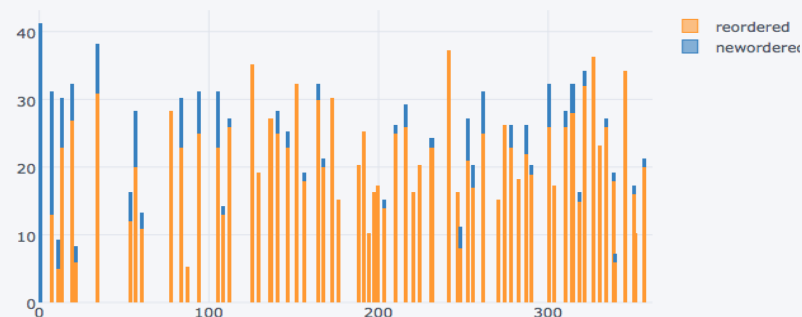
User Pattern : 103597



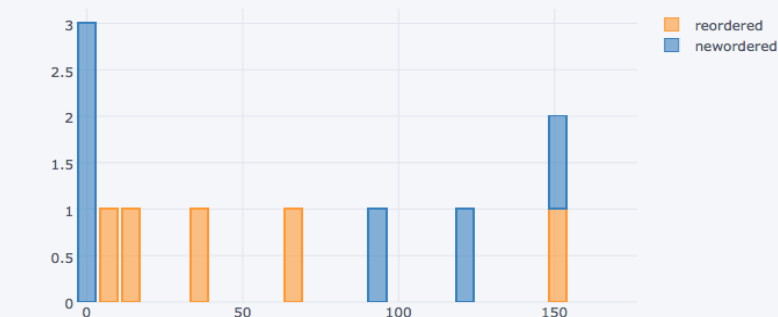
User Pattern : 30



User Pattern : 120572

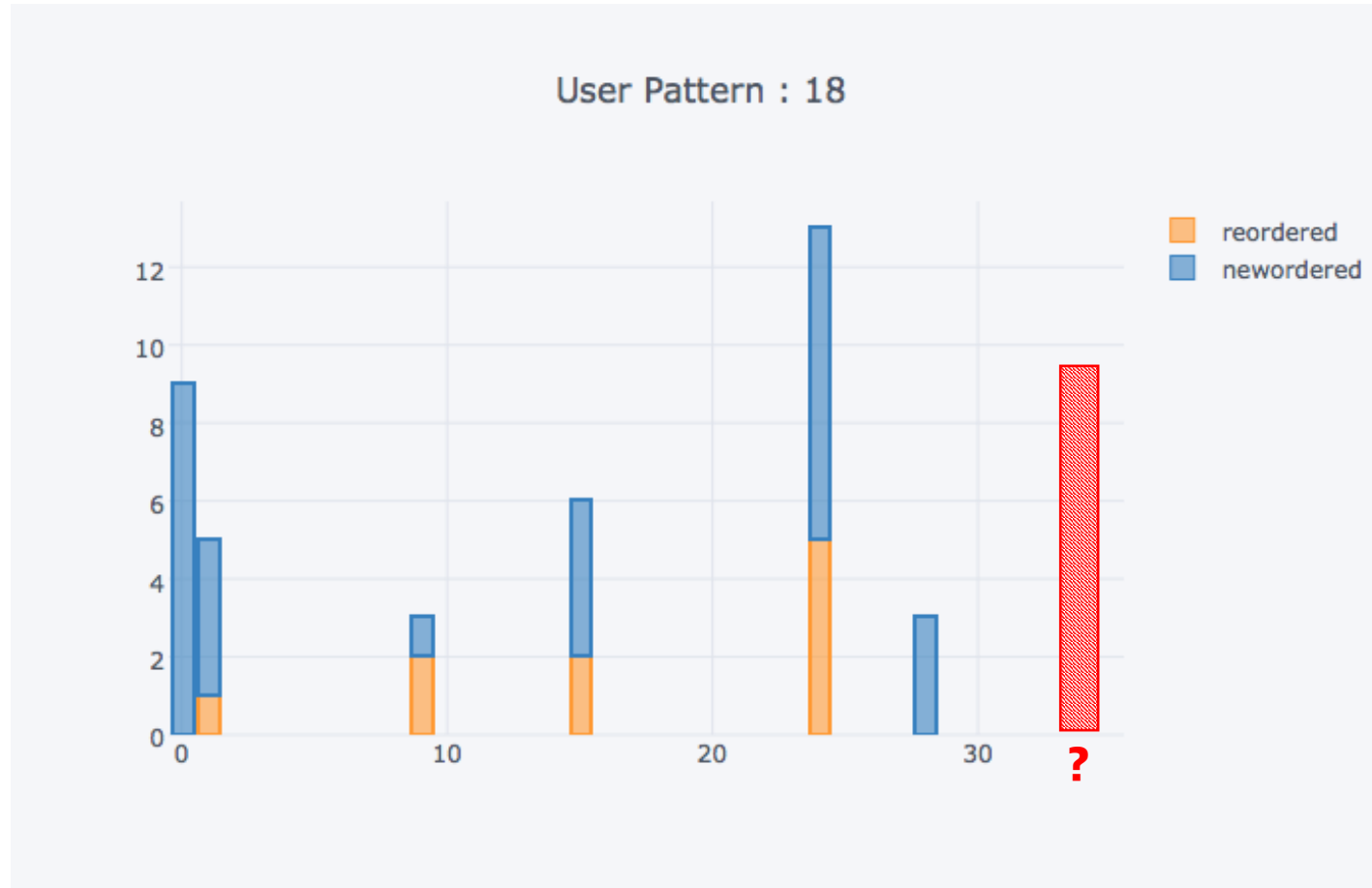


User Pattern : 30



3. Methology

■ 모델 전략



3. Methology

■ 모델 전략

새롭게 구매될 제품들

- 5만 개의 제품들 중 새롭게 갑자기 구매될 제품들을 찾기는 쉽지 않다
- 지금까지 가장 비슷하게 구매한 유저들을 찾아서 그 유저들의 신규 구매 제품들을 보자
- 모든 제품들 중 해당 유저의 환경에서 가장 많이 신규 구매된 제품들을 보자

재구매 비율 예측

재구매될 제품들

- 지금까지 구매한 제품들이 후보군
- 모든 주문 데이터를 기반으로 이 제품이 재구매될 확률을 예측
- 하이퍼파라미터 조정을 통해 기준값을 정하여 그 값 이상이면 살 것이라고 예측



?

3. Methology

■ Xgboost

- Tianqi Chen에 의해 개발
- 진화된 Gradient Boosting 기법, 병렬 처리를 통한 빠른 연산 가능
- 다양한 커스텀 최적화 옵션을 제공함
- kaggle competition의 고득점자들이 애용하는 기법

dmlc
XGBoost

■ Light Gradient Boosting Model

- Microsoft가 최근 발표한 가벼운 Gradient Boost Model
- 트리 전개 방식 : leaf wise 방식 (XG BOOST는 level wise 방식)
- level - wise 알고리즘 보다 loss를 더 많이 줄여줌 -> 높은 accuracy
- Light GBM은 XGBoost 보다 복잡, 오버피팅 가능성 높음

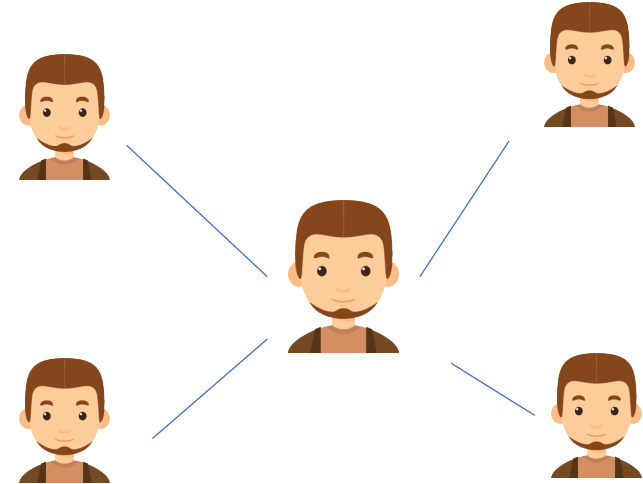


Light **GBM**

3. Methology

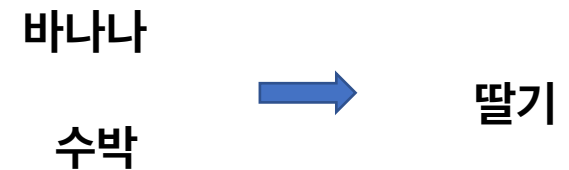
■ k-nearest Neighbors

- 나랑 가장 비슷한 사용자는 어떤 물건을 살 것인가?
- 각 유저의 구매 과거를 정규화하 + 벡터화
- 나와 가장 비슷한 cosine similarity를 갖는 유저 k명 찾기
- k수를 계속 바꿔가면서 최적화



■ Association Rules 연관규칙

- 전통적이고 통계적으로 입증된 방법
- “A와 B를 사는 사람은 C도 사더라”
- 머신러닝을 통해 자동으로 Rule을 generate시키고 통계적 검정의 반복



4. Resampling & Model Tuning

■ Computing Power의 부족



20만 명



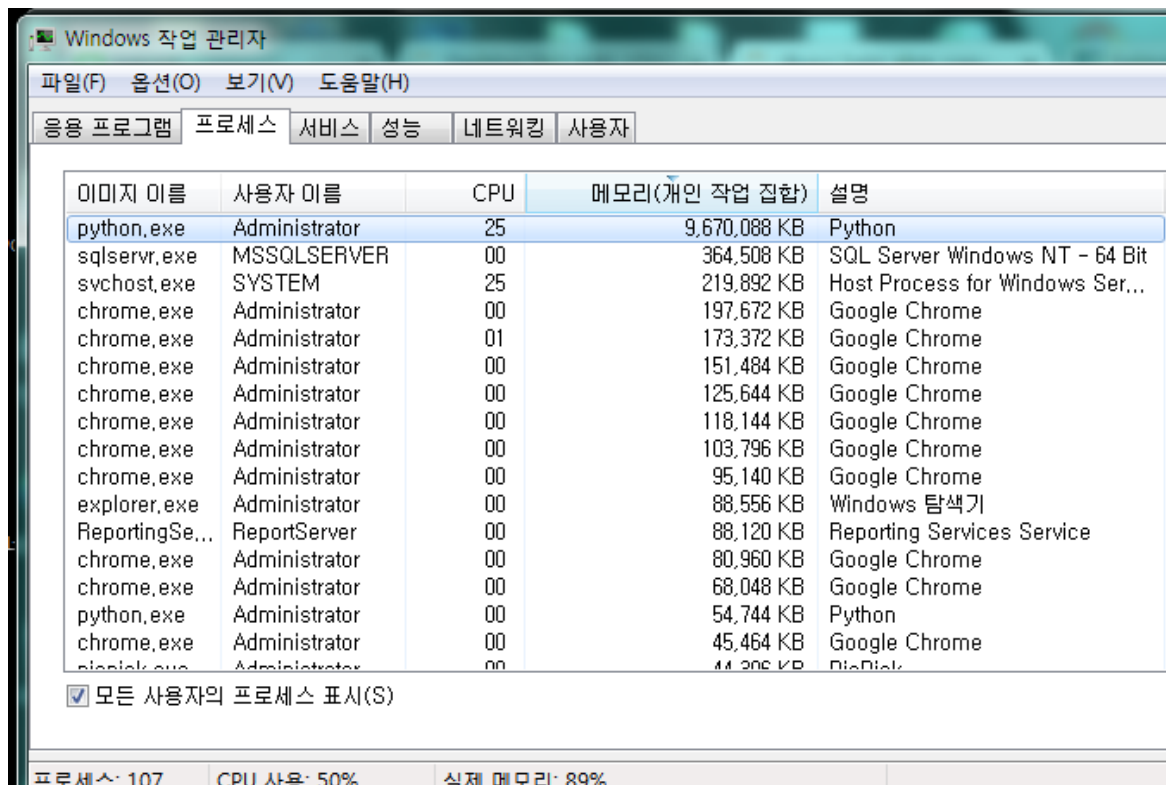
5만 종류



330만 주문

4. Resampling & Model Tuning

■ Computing Power의 부족



이미지 이름	사용자 이름	CPU	메모리(개인 작업 집합)	설명
python.exe	Administrator	25	9,670,088 KB	Python
sqlservr.exe	MSSQLSERVER	00	364,508 KB	SQL Server Windows NT - 64 Bit
svchost.exe	SYSTEM	25	219,892 KB	Host Process for Windows Ser...
chrome.exe	Administrator	00	197,672 KB	Google Chrome
chrome.exe	Administrator	01	173,372 KB	Google Chrome
chrome.exe	Administrator	00	151,484 KB	Google Chrome
chrome.exe	Administrator	00	125,644 KB	Google Chrome
chrome.exe	Administrator	00	118,144 KB	Google Chrome
chrome.exe	Administrator	00	103,796 KB	Google Chrome
chrome.exe	Administrator	00	95,140 KB	Google Chrome
explorer.exe	Administrator	00	88,556 KB	Windows 탐색기
ReportingSe...	ReportServer	00	88,120 KB	Reporting Services Service
chrome.exe	Administrator	00	80,960 KB	Google Chrome
chrome.exe	Administrator	00	68,048 KB	Google Chrome
python.exe	Administrator	00	54,744 KB	Python
chrome.exe	Administrator	00	45,464 KB	Google Chrome
disks.sys	Administrator	00	44,306 KB	Disks.sys

☒ 모든 사용자의 프로세스 표시(S)

프로세스: 107 CPU 사용: 50% 실제 메모리: 89%

문제 푸는 데 걸리는 시간

1개 : 13분

75,000개 : 975,000분
= 16,250시간
= 667일
= 22개월

4. Resampling & Model Tuning

■ 해결방법 1. AWS 이용

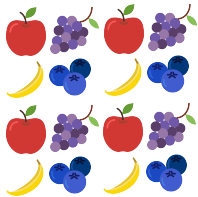
- 아마존닷컴이 제공하는 각종 원격 컴퓨팅 서비스
- 64GB의 CPU와 256GB의 RAM을 가진 컴퓨터를 사용
- 시간당 비용이 발생하여 계속 돌리기 어렵다
- 확실하게 모델을 정하고 돌리는 것이 더 효율적

■ 해결방법 2. Data Sampling

- 기존 Train / Test 비율과 동일하게 조그만 샘플 데이터셋 생성
- 샘플 데이터셋의 Test Set에 대한 답지와 채점 함수 생성
- Submission을 제출할 경우 자동으로 점수와 오답 확인하는 함수 생서

4. Resampling & Model Tuning

■ 해결방법 2. Data Sampling



예측값

kaggle™

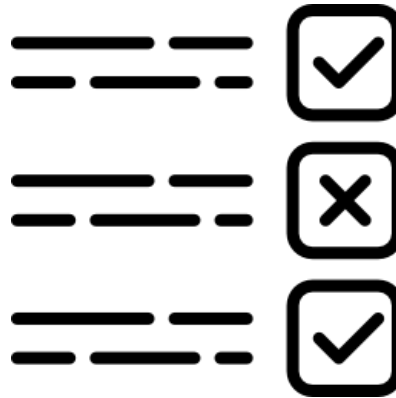
Mean F1 Score
: 0.308

4. Resampling & Model Tuning

■ 해결방법 2. Data Sampling



예측값



샘플 데이터의
답지로 채점

order_id	product_id	answer	TP	FP	F1_score	user_id
1342	3798 5058 8048 12341 13176 14966 14879 21137 2...	13176 30927 14966 21137 45129 3798 33081 7982	5	3	8	0.476190 156818
2869	34578 37067 46802	17779 283 14511 10246 6141 34824 43961 42295	1	11	2	0.133333 68146
2889	13176 13535 21616 28985 46979 47766	35951 22142 21616 20889 19508 2190 22968 22395...	1	11	5	0.111111 182107
3179	35221	35221 38522 3634 44632 14867 47990 5757 34521	1	7	0	0.222222 54174
3349	None	33188 25064 48455 17038 30391	0	5	1	0.000000 134766
3957	16787 21616 21603 29993 35028 47766	2187 13225 24235 13000 14218 42368 45066 24802...	3	23	3	0.187500 158235
3971	2295 2435 5183 13733 21573 21781 23452 31485 3...	5183 13733 23452 38730 40408 2436 47605 45333 ...	11	2	5	0.758621 127382
4290	52899	5994 11365 30927	0	3	1	0.000000 96792
4201	651 8571 11759 12341 16281 16797 18227 20840 2...	19565 4820 21137 20940 32079 12341 11759 38928...	7	11	4	0.482759 154884
4267	7781 16696 23909 25830 29432 30274 31183 31735...	31683 31730 37158 25830 37215 34222 6128 6448 ...	6	7	10	0.413793 107978
4284	4600 7559 10814 21137 21903 24852 27845 37646	19678 45123 24852 45007 18811 25005 35621 2784...	4	10	4	0.363636 165282
4305	195 9387 12145 13504 20788 25584 28309 30055 3...	195 20788 28309 40516 10145 13504 34504 36055 ...	8	1	5	0.727273 177727
4309	5385 6052 6532 10490 13176 13774 22825 27582 2...	33232 45371 1483 29584 5385 13774 22128 30731 ...	5	19	7	0.277778 155465
4494	2825 5303 7781 11352 13083 13176 22963 29465 3...	38838 34 17980 45007 48745 48679 47209 30169 1...	1	21	10	0.060606 115015

점수뿐만 아니라
무엇을 어떻게 틀렸는지 파악 가능

4.

■ 하

order_id	예측값	product_id	실제값	answer	TP	FN	FP	점수	user_id
1342	3798 5068 8048 12341 13176 14966 14979 21137 2...		13176 30827 14966 21137 46129 3798 33081 7862		5	3	8	0.476190	156818
2869	34578 37067 46802		17779 283 14511 10246 6141 34824 43961 42265 4...		1	11	2	0.133333	68146
2889	13176 13535 21616 28985 46979 47766		35951 22142 21616 20889 19508 2180 22888 22395...		1	11	5	0.111111	182107
3179	35221	35221	35221 39322 3634 44632 14867 47990 5757 34521		1	7	0	0.222222	54174
3349	None	None	33198 25064 48455 17038 30391		0	5	1	0.000000	134766
3957	16797 21616 21903 29993 35628 47766		2187 13225 24235 13500 14218 42368 45066 24852...		3	23	3	0.187500	159235
3971	예측 점수가 특히 높거나 낮은 유저들의 구매 패턴 확인								2
4090	12899		5994 11365 30827		0	3	1	0.000000	96792
4201	651 8571 11759 12341 16281 16797 18027 20940 2...		19565 4920 21137 20940 32079 12341 11759 38928...		7	11	4	0.482759	154864
4267	7781 16696 23909 25830 29432 30274 31683 31730...		31683 31730 37158 25830 37215 34222 6128 6448 ...		6	7	10	0.413793	107978
4284	4920 7559 10814 21137 21903 24852 27845 37646		19678 45123 24852 45007 18811 25005 35921 2784...		4	10	4	0.363636	163282
4305	195 9387 10145 13504 20788 25584 29309 32605 3...		195 20788 29309 40516 10145 13504 34604 36095 ...		8	1	5	0.727273	177727
4309	5385 6052 6532 10490 13176 13774 22825 27592 2...		33232 45371 1463 29584 5385 13774 22128 30731 ...		5	19	7	0.277778	155465
4494	2825 5303 7781 11352 13083 13176 22963 28465 3...		38838 34 17980 45007 48745 48679 47209 30169 1...		1	21	10	0.060606	115015

stacart

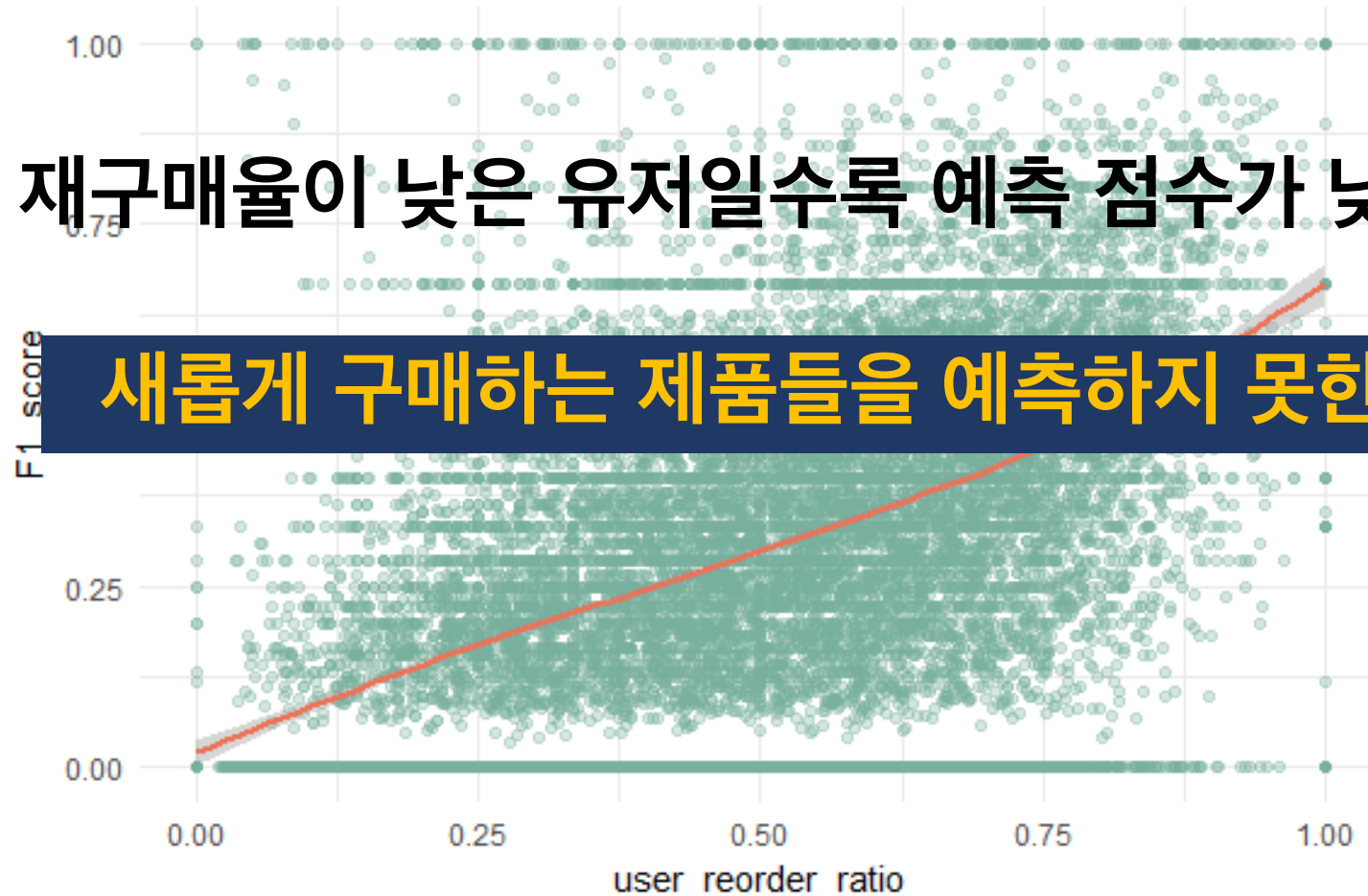
나라
파악 가능

4. Resampling & Model Tuning

■ 모델 자체의 한계

재구매율이 낮은 유저일수록 예측 점수가 낮았다

새롭게 구매하는 제품들을 예측하지 못한다



4. Resampling & Model Tuning

■ 모델 전략

새롭게 구매될 제품들

- 5만 개의 제품들 중 새롭게 갑자기 구매될 제품들을 찾기는 쉽지 않다
- 지금까지 가장 비슷하게 구매한 유저들을 찾아서 그 유저들의 신규 구매 제품들을 보자
- 모든 제품들 중 해당 유저의 환경에서 가장 많이 신규 구매된 제품들을 보자

재구매 비율 예측

재구매될 제품들

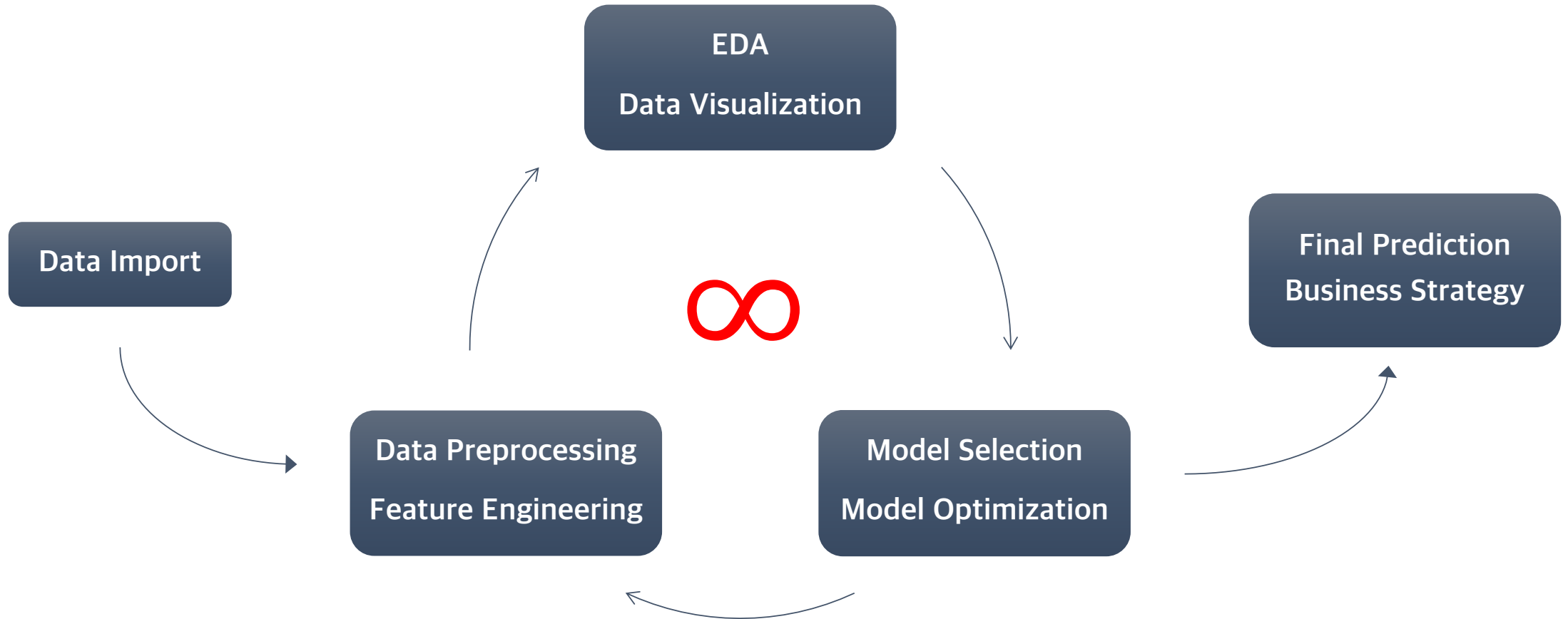
- 지금까지 구매한 제품들이 후보군
- 모든 주문 데이터를 기반으로 이 제품이 재구매될 확률을 예측
- 하이퍼파라미터 조정을 통해 기준값을 정하여 그 값 이상이면 살 것이라고 예측



?

5. Conclusion

■ 데이터 분석은 문제 해결의 무한한 연속



5. Conclusion

■ 최신 기술만이 정답은 아니다

- 쉽게 불러올 수 있는 좋은 라이브러리는 이미 많다
- 적절한 라이브러리를 적절하게 사용하는 것이 중요
- 데이터에 대한 충분한 이해와 탐색이 필요하다

dmlc
XGBoost



Light **GBM**

■ 데이터에 대한 이해와 정확한 설계가 시간을 획기적으로 줄인다

- 모델을 수정하기 위한 최적화 설계가 잘 갖춰져야 한다
- 내 모델이 어떻게 왜 틀렸는지 직접 확인하는 과정이 필요하다
- 끊임없이 데이터를 탐색하고 문제를 해결하는 과정이 반복되어야 한다

감사합니다

YBIGTA 9기 박성남, 10기 김상현, 10기 김태한

Q&A

그 어느 질문이든 환영합니다