LG Aimers

MQL 데이터 기반 B2B 영업기회 창출 예측 모델 개발

Data Raiders

(고경준, 안현준, 진실, 유승태, 문정현)

Contents.

01. EDA

02. Pre-Processing

03. Model Selection

04. Ensemble

05. Appendix

Data Raiders

고경준

유승태

진실

안현준

문정현

01 EDA

"데이터 탐색 및 Insight 발견하기"

데이터 개요

변수별 target 분포

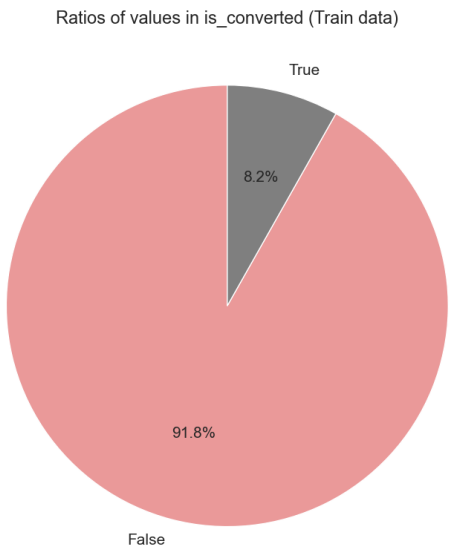
주요 변수 EDA & Insight

1. 데이터 개요

: MQL 고객의 38개 정보와 영업 전환 성공여부를 포함한 데이터

1. train.csv shape : (59299, 39)
   row : 59299 개
   column : 39 개 → 영업 전환 성공 여부(target) + 고객정보, 요청사항, 상품 정보, 문의 정보 등

2. Target ('is_converted', 영업 성공 전환 여부) 비율
   : True (8.2%) , False (91.8%) 로 데이터 불균형



1. 데이터 개요

결측치 분포

결측치 0%

bant_submit
business_unit
customer_idx
customer_history
enterprise
lead_desc_length
customer_position
response_corporate
ver_cus
ver_pro
lead_owner
is_converted

11개

결측치 1~79%

customer_country
com_reg_ver_win_rate
customer_type
historical_existing_cnt
customer_job
inquiry_type
product_category
customer_country.1
expected_timeline
ver_win_rate_x
ver_win_ratio_per_bu
business_area

12개

결측치 80%~

id_strategic_ver
it_strategic_ver
idit_strategic_ver
product_subcategory
product_modelname
business_subarea

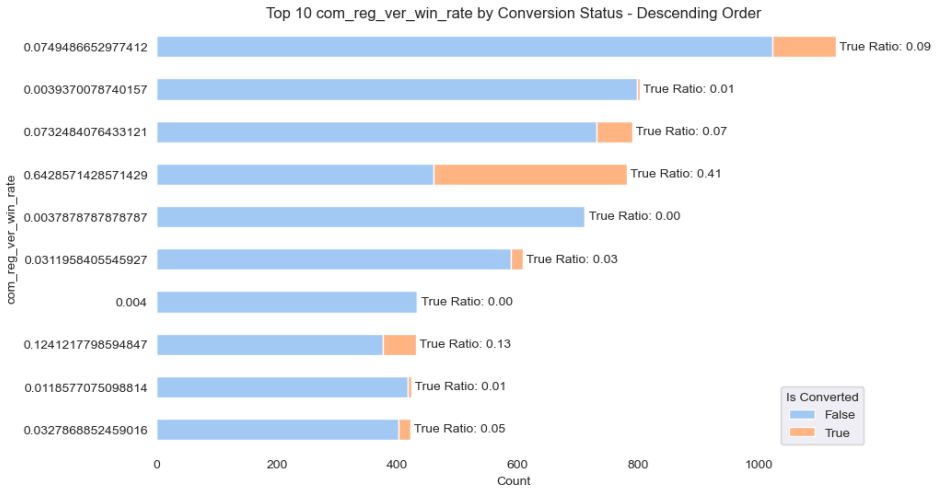
6개

결측치 0~79% 각 변수 내 카테고리 축소, 변수 활용하여 새로운 변수 생성 아이디어

결측치 80% 결측치 채울 아이디어, 변수 제거 타당성 확인

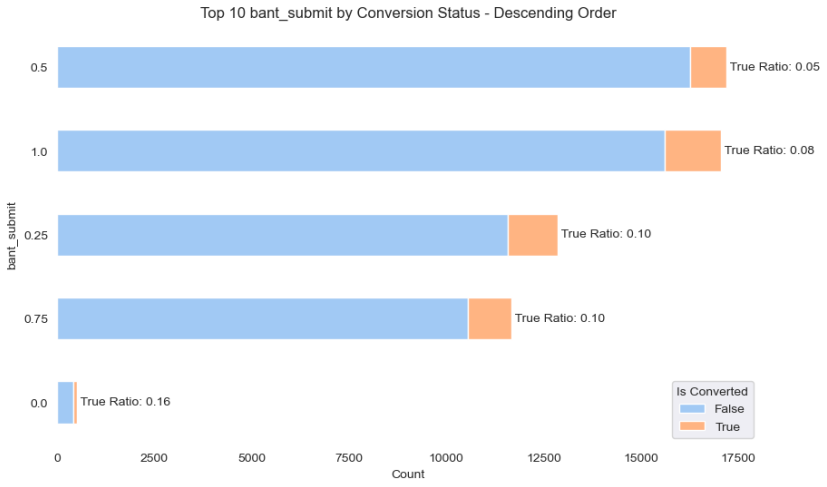
2. 변수별 Target 분포

Target 분포 차이 큰 변수 (Good)



'com_reg_ver_win_rate', 'product_modelname',
'ver_win_rate_x', 'business_area', 'business_subarea'

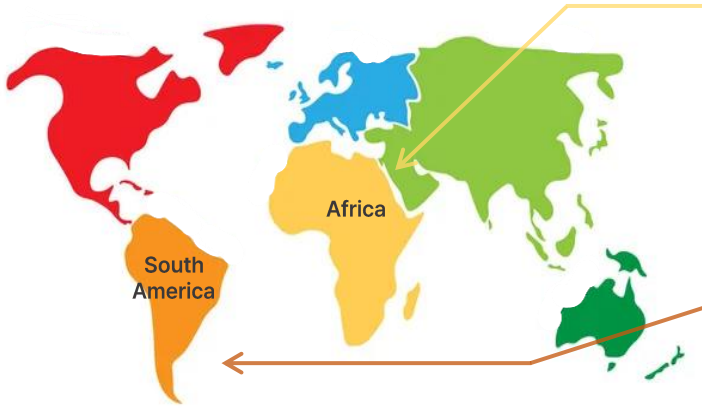
Target 분포 차이 작은 변수 (Bad)



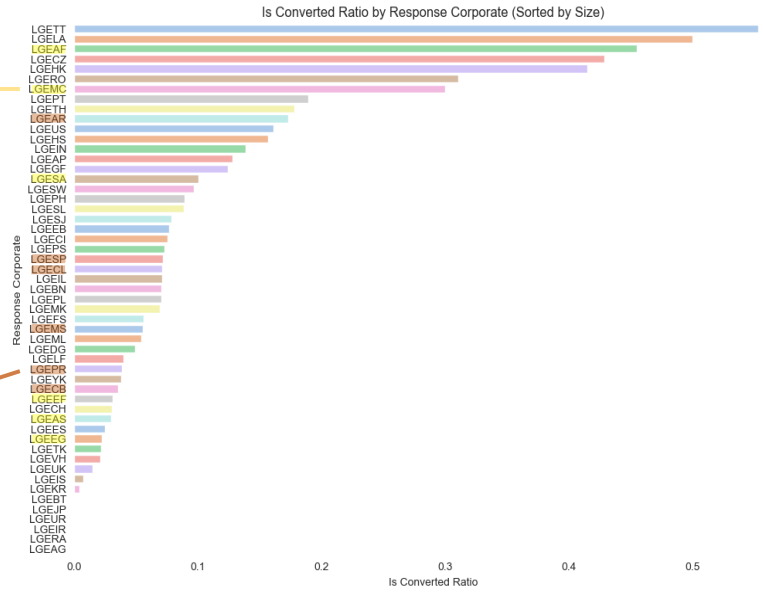
'bant_submit', 'customer_country', 'business_unit',
'customer_idx', 'customer_type','enterprise ' ,
historical_existing_cnt', 'customer_job ' ,
'lead_desc_length ' , 'inquiry_type ' , 'product_category',
'product_subcategory','customer_country.1 ' ,
'customer_position', 'response_corporate ' ,
'expected_timeline','ver_cus','ver_pro ' ,
'ver_win_ratio_per_bu', 'lead_owner'

3. 주요 변수 EDA & Insight

1. Response_corporate & Customer_country



< Customer_country & response corporate (groupby) >



Insight

- ‘response_corporate’ 변수는 총 53개의 자사 법인명으로, 법인에 따라 구매 전환율이 크게 달라진다.
- ‘customer_country’ 변수를 기준으로 법인명의 지역이 잘 구분된다.

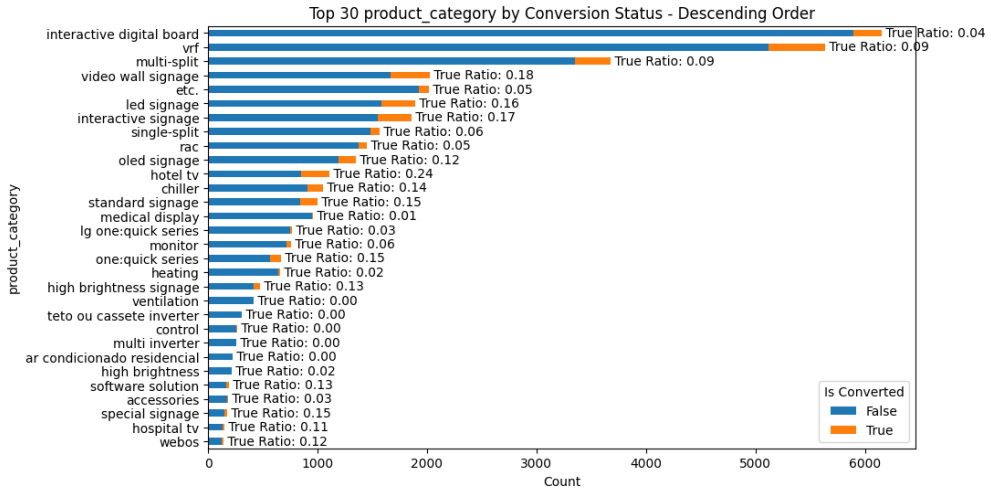
3. 주요 변수 EDA & Insight

2. Product_category

Product_category
'multi-split'
'single-split'
...
'solar, tv'
'monitor signage, tv'

' solar , tv '

: Product_category 값 중
컴마(,)를 사이로
여러 값이 있는 데이터 존재



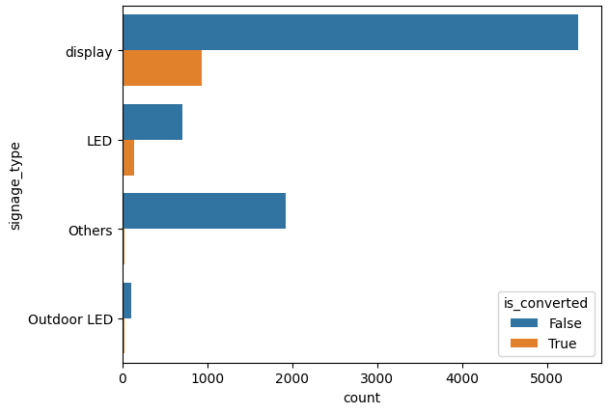
Insight

- ‘product_category’ 변수 내 각 클래스가 하나의 정보가 아닌 여러 정보를 포함하여 총 357개의 많은 클래스 값을 가진다.
- 데이터 수를 기준으로 정렬한 클래스에서 357개 중 상위 30개에 대부분의 데이터가 몰렸다.
  → 지엽적인 데이터로 overfitting 우려된다.

3. 주요 변수 EDA & Insight

3. Product_modelname

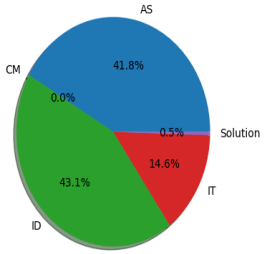
Product_modelname
22SM3G-B
55VSM5J
...
LAA015F
GSCD100

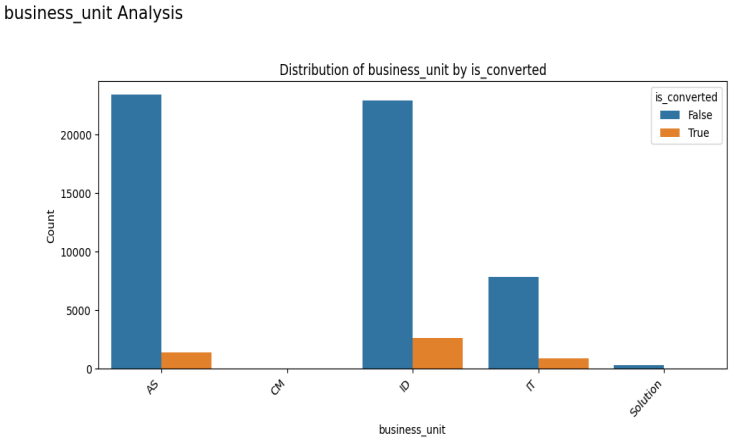


Insight

제품 사이트 참고하여 제품명이 가지는 규칙 세워,
새로운 변수(sinage_type) 생성 가능해 보인다.

4. Business_unit



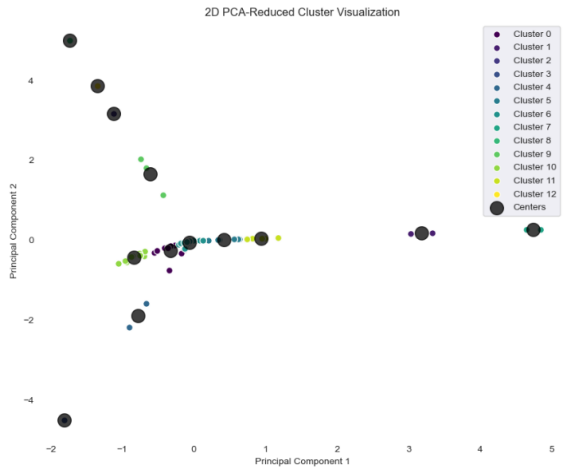


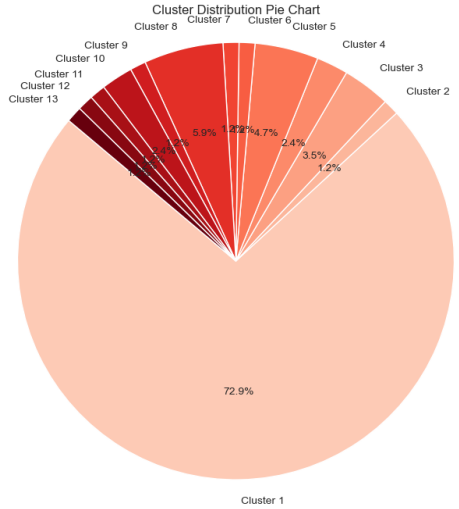
Insight

'business_unit' 의 클래스는 총 5개로, 대부분 데이터가
AS, ID, IT로 분류되어 클래스 개수를 줄여볼 수 있다.

3. 주요 변수 EDA & Insight

5. Business_area & Business_subarea





['business_area', 'business_subarea'] 은 다대일 대응 (1 : n)

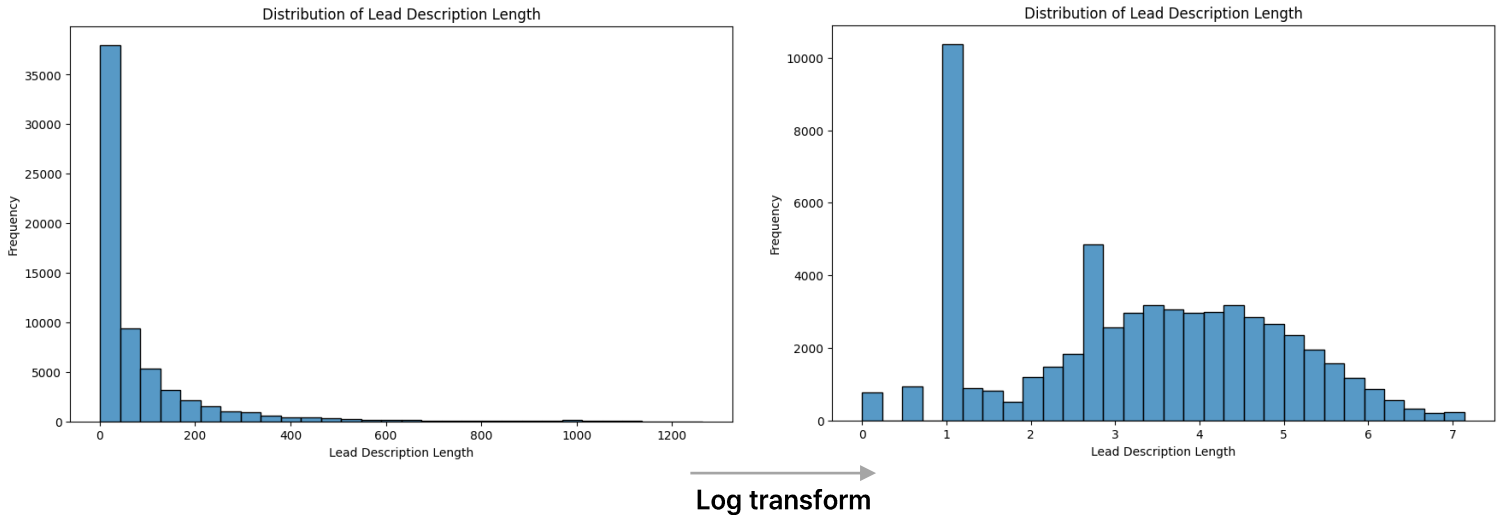
['business_area', 'business_subarea'] 간의 관계파악을 위해 business_subarea에 따른 business_area의 패턴을 Clustering

Insight

- Clustering 결과 : 13개 Cluster로 묶인다.
- 'business_area'의 클래스 개수가 12개인 점을 감안하면, other를 새로운 클러스터라 할 경우 거의 같다.
  → 'business_subarea' 는 'business_area'의 부분집합으로 볼 수 있다. (business_subarear ⊂ business_area)

3. 주요 변수 EDA & Insight

6. Lead_desc_length



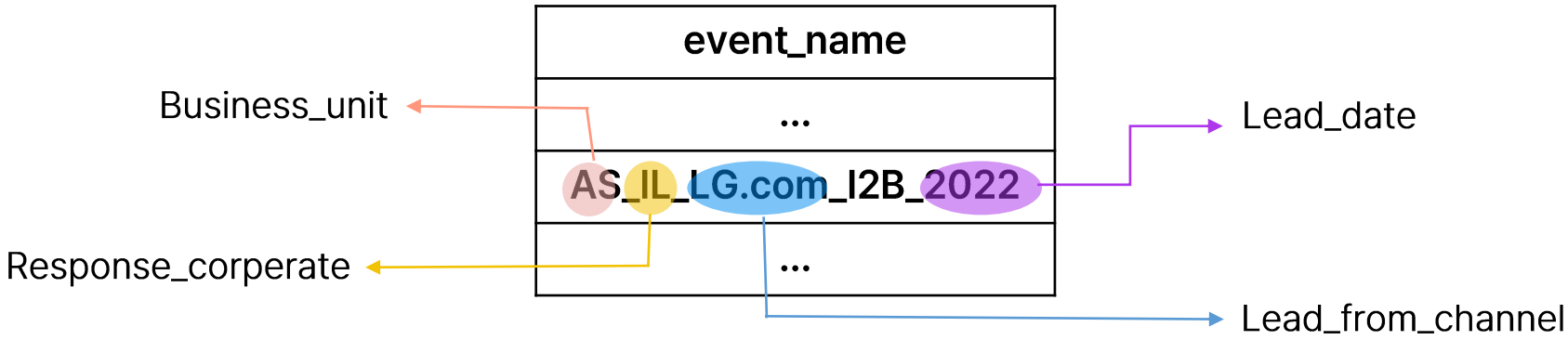
	False	True	Data_count	True(%)
0-1.9	12959	557	13516	4.12
2-3.4	14763	647	15410	4.20
3.5-4.5	13213	1391	14604	9.52
4.5+	12748	2250	14998	15.00

Insight

- 데이터의 값들이 넓은 범위에서 continuous하게 분포해 있지만, 대부분의 데이터는 특정 임계치를 넘어서지 않는다.
- 영업 설명 길이가 길어질수록 영업 성공 전환 비율은 높아진다.

3. 주요 변수 EDA & Insight

7. Event_name

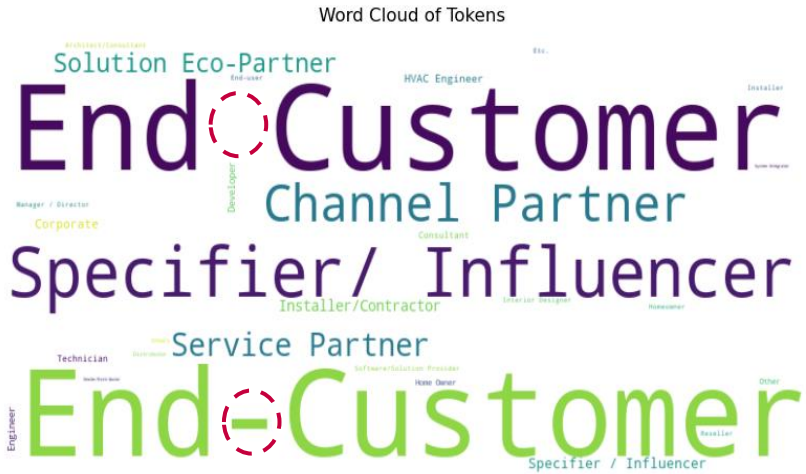


Insight

- 'event_name' 에는 <'besiness_unit', 'response_corperate', 'lead_date', 'lead_from_channel'> 과 같이 다양한 변수의 정보들이 들어있다.
  → '_'를 기준으로 split 하여 파생 변수를 생성

3. 주요 변수 EDA & Insight

8. Etc.



'تافييكت'
'הבורם לוציפ'
'醫院電視'
'điều hòa cục bộ'
...

다양한 외국어 존재

'customer_country' 변수에서
나라만 추출하여 워드클라우드 결과,
인도가 가장 많음

→ 하지만 영어 외 외국어 중 인도어는
거의 없는 것으로 보아
인도는 영어를 사용한다 추측

Insight

- 데이터 중 인도 고객이 가장 많지만, 영어 외 외국어 중 인도어는 거의 없는 것으로 보아 번역을 하지 않아도 되어 보인다.
- 텍스트 클래스를 갖는 변수 내에 같은 의미지만, [기호, 알파벳 대소문자, 동의어]로 다른 클래스로 간주되는 값이 있다.

02 Pre-Processing

"전처리로 데이터 품질 높이기"

Data Reduction

Data Transformation

Data Imputation

1. Data Reduction

💡 Input feautre의 수가 38개이며, null값을 많이 포함한 feature들이 존재.
중복된 feature, feature importance가 낮거나 불필요한 feature 제거.

1. 11개 feature 제거

   removal_features = {

   'id', 'bant_submit', 'id_strategic_ver', 'it_strategic_ver', 'idit_strategic_ver', 'customer_country', 'customer_country.1', 'business_subarea', 'business_area', 'lead_date', 'lead_description'

   }

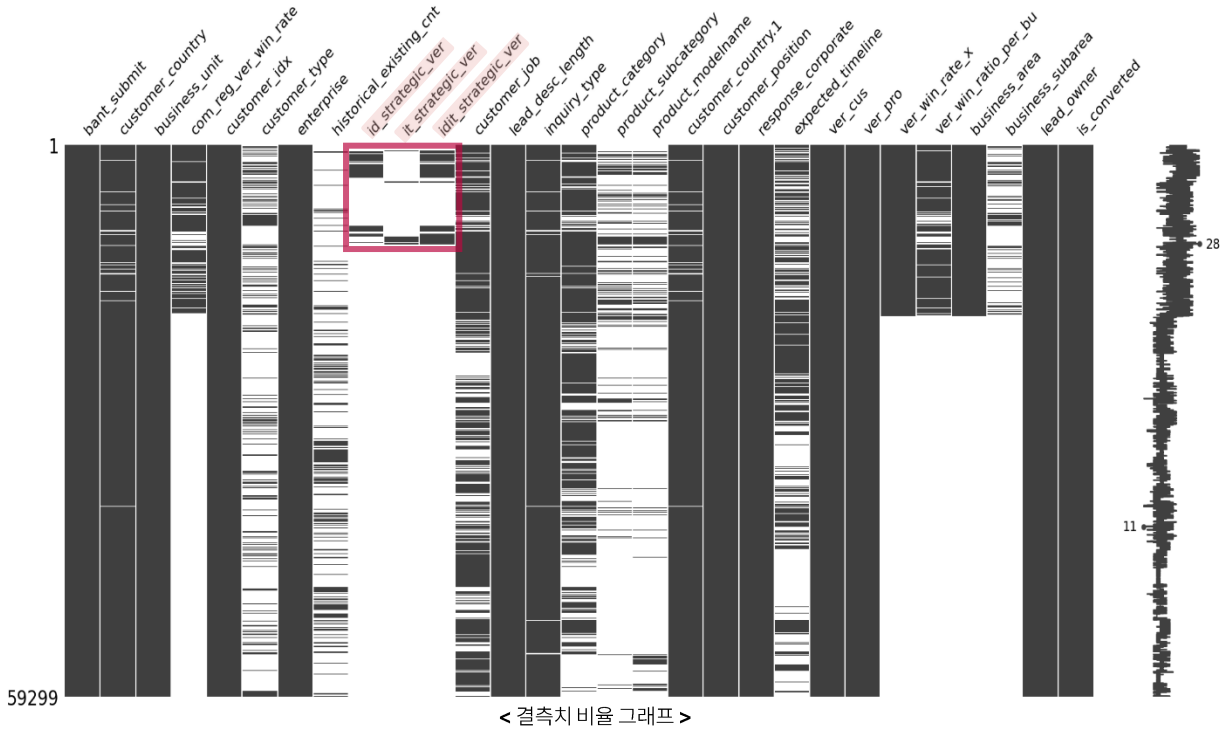
02 Pre-processing

2. 변수 제거

[id_strategic_ver, it_strategic_ver, idit_strategic_ver] 변수 제거

: 결측치가 많으며 세 변수끼리 만 분포가 같고,
다른 컬럼과 연관성 보이지 않음

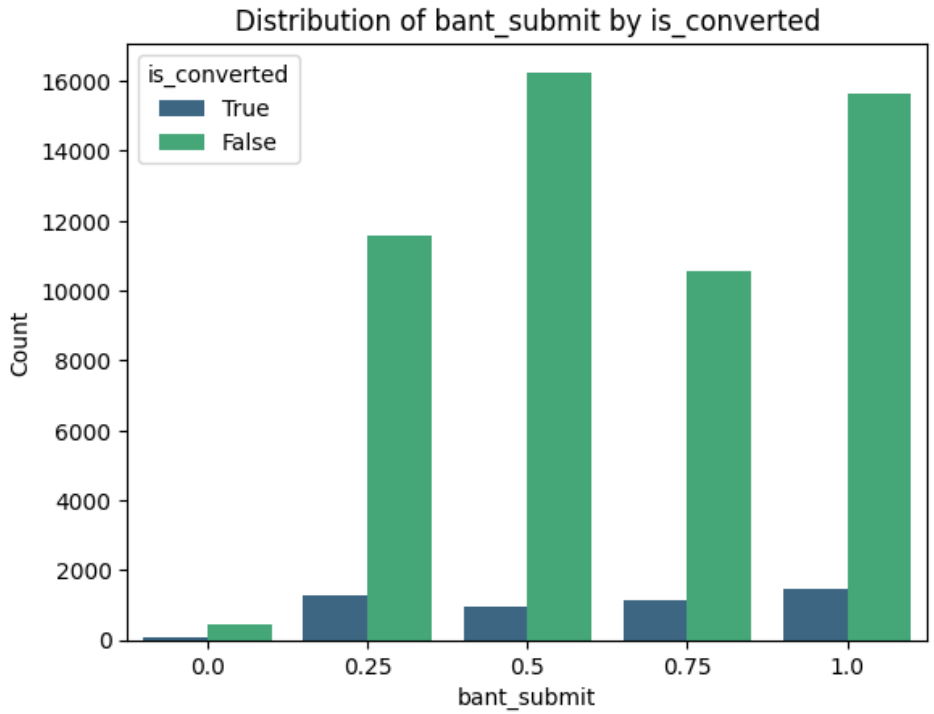
변수명	id_strategic_ver	it_strategic_ver	idit_strategic_ver
결측치 비율	94.19%	98.11%	92.30%



< 결측치 비율 그래프 >

[bant_submit] 변수 제거

: 분산 0.082 로 각 class 별 분포의 차이 크지 않음
target (is_converted) 의 비율 차이도 없다고 판단



02 Pre-processing

2. 변수 제거

[customer_country.1] 변수 제거

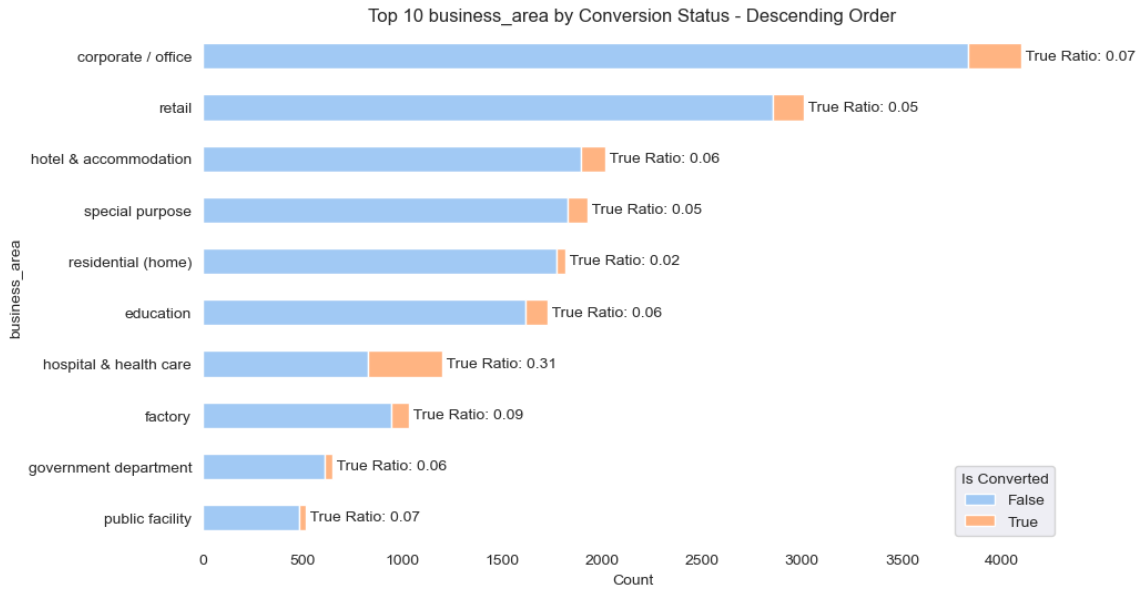
: customer_country 변수와 중복됨

	customer_country	customer_country.1
0	/Quezon City/Philippines	/Quezon City/Philippines
1	/PH-00/Philippines	/PH-00/Philippines
2	/Kolkata /India	/Kolkata /India
3	/Bhubaneswar/India	/Bhubaneswar/India
4	/Hyderabad/India	/Hyderabad/India
...	...	...
64565	/São Paulo/Brazil	/São Paulo/Brazil
64566	General / / United States	General / / United States
64567	/ OURO BRANCO / Brazil	/ OURO BRANCO / Brazil
64568	/ / Germany	/ / Germany
64569	/ Ongole / India	/ Ongole / India

[business_area, business_subarea] 변수 제거

: hospital & healthcare만 target 분포 차이 있음
결측치 비율이 높음

변수명	business_area	business_subarea
결측치 비율	68.19%	90.68%

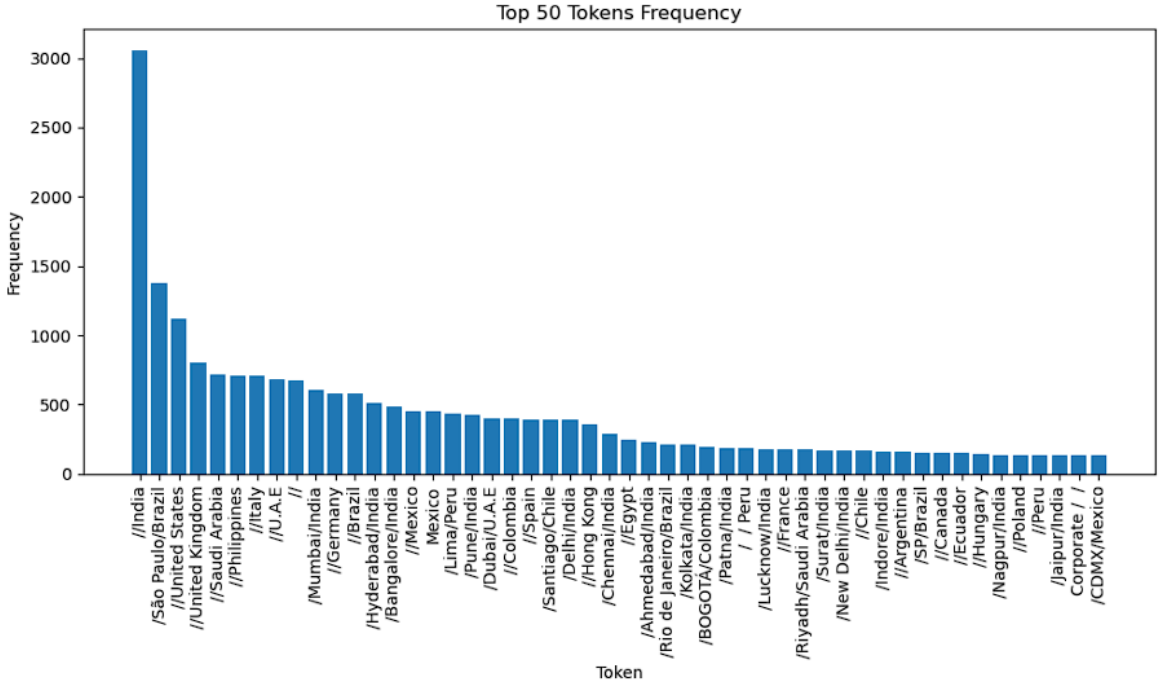


02 Pre-processing

2. 변수 제거

[customer_country] 변수 제거

- unique 값 수가 많음 -> 정제 되어있지 않음
- 형식 통일되어 있지 않음 (e.g. //나라, 도시/나라 등)
- 같은 값이지만 다르게 표시된 경우 다수 (e.g. 터키, 튀르키예)



2. Data Transformation

💡 영어로 작성된 문자형 feature들 존재.
특수문자가 들어가 있거나, 대 소문자의 차이로 서로 다른 변수 값으로 나타남.

1. 변수 값 통일
   → 모든 문자 소문자화, 쉼표 및 마침표 제거, 밑줄 및 하이픈 공백으로 변환.
   → Feature에서 'other', 'others', 'etc' 값을 'Others'로 통일.

2. 변수 값 대체
   → ['business_unit'] feature에서 'Solution', 'CM' 값은 'Others'로 변경.

2. Data Transformation

💡 범주형 feature 중 변수값 종류가 너무 다양하고, 소수의 특정 변수 값들이 feature의 대다수를 차지하고 있음.

1. Feature 내 변수 category 축소를 통한 새로운 feature 생성
   → ['continent'] : ['response_corporate'] feature를 활용하여 'Asia', 'Africa', 'NorthAmerica', 'SouthAmerica', 'Europe', 'Oceania' 값을 가진 대륙 feature 생성
   → ['business_area_group'] : ['business_area'] feature를 활용하여 'Office', 'Public', 'Amenity', 'Industry', 'Cur_area' 값을 가진 feature 생성
2. 변수 값에서 등장하는 특정 패턴 counting을 통한 새로운 feature 생성
   → ['product_category_count'] : ['product_category'] feature에서 쉼표(,)로 분리하여 그 개수를 세어 feature 생성
   → ['lead_owner_count'] : ['lead_owner'] feature에서 값의 등장 횟수를 계산한 feature 생성

2. Data Transformation

💡 수치형 feature임에도 불구하고 변수 값의 다양성이 매우 적거나, 수치의 높고 낮음이 유의미하지 않은 feature 존재.

1. 수치형 feature 범주화

   → ['customer_idx', 'lead_owner', 'ver_cus', 'ver_pro']

2. expected_budget (미사용)

💡 문자열에서 숫자 추출 및 화폐 통일 필요성에 대한 분석

1. 영어로 번역
   → ['weniger als', 'menos de' -> 'less than']
   → ['uber' -> 'more than']
2. 문자열에서 숫자 추출
   → lower_budget과 upper_budget 정의
   → 예: less than 200,000 -> [0, 200000]
3. 달러로 통화 통일
   → forex-python의 CurrencyRates() 이용해서 '2021-01-01'의 환율 계산
   → lower_budget_usd과 upper_budget_usd로 저장

2. lead_from_channel (미완성)

💡 Lead로 연락되는 경로에서 인사이트 찾기

1. 1% 이상 등장한 channel 경로 필터링
   → unique 개수 약 60개

2. 텍스트 클러스터링
   → 의미 유사도가 아니라 텍스트 유사도를 이용
   → edit distance 사용

3. Data Imputation

💡 24개의 Input feature 중 12개의 feature가 null값 존재.

1. 범주형 feature
   → String 'UNK'로 결측치 처리.

2. 수치형 feature
   → 전처리 이후 존재할 수 없는 값(-1)으로 결측치 처리.

03 Model Selection

"베이스라인 모델 선택 및 학습"

베이스라인 모델 선택 기준

Catboost

학습 파라미터

1. 베이스라인 모델 선택 기준 (1)

💡 Which model is good?

1. Interpretability
   → 어떤 feature가 고객 전환에 영향을 미치는지 알아야 현업에서 사용 가능
2. Fast training and inference
   → 모델 학습/추론 속도가 일정 수준 이상
   → 다양한 실험을 빠르게 할 수 있음

💡 **Candidate Baseline Models**

1. Linear models (Linear regression, SVM, etc.)
2. Decision Trees (including Random Forests, Boosting-based, etc.)
3. Deep learning models are typically difficult to interpret.

1. 베이스라인 모델 선택 기준 (2)

💡 How to convert categorical features to real numbers?

1. One-hot encoding
   → 차원이 급격히 늘어나는 문제가 있음. (차원의 저주)
   → customer_idx, lead_owner, product_category 등은 cardinality가 매우 높음
2. Target encoding
   → class label의 통계량으로 변환 (i.e. 평균값, 중앙값)
   → 높은 Overfitting의 위험성
3. Unseen category
   → unseen category에 대해서도 모델이 대응 가능해야 함

2. Baseline model: Catboost

💡 Advantages of Catboost

1. Boosting-based Decision Tree
   → feature importance로 interpretability 확보 가능
   → 빠른 학습/추론 속도
2. Good handling of categorical features
   → high cardinality
   → avoid overfitting
   → unseen categorical features

2. Catboost

💡 Catboost Algorithm

1. High cardinalilty → average class label value
2. Avoid overfitting → random permutation
3. Unseen categorical features → using prior probability of class label

Average label value **Random permutation**

$$\frac{\sum_{j=1}^{p-1}\left[x_{\sigma_j,k}=x_{\sigma_p,k}\right]\cdot Y_{\sigma_j}+a\cdot P}{\sum_{j=1}^{p-1}\left[x_{\sigma_j,k}=x_{\sigma_p,k}\right]+a}$$

Prior probability of class label

Weight of the prior

04 Ensemble

"모델 앙상블로 Robustness 향상하기"

Ensemble 이유

Bagging vs Boosting

Ensemble 적용

1. Ensemble 이유

💡 Overfitting을 방지하고 Robust한 모델 예측 결과를 내기 위해 사용

1. 모델에 다양성 도입

   → Ensemble은 다양한 개별 모델로 구성되어 각 모델이 데이터의 서로 다른 측면을 학습할 수 있음

2. Overfitting(과적합) 감소

   → Ensemble은 여러 모델의 예측을 평균화하거나 가중치를 부여하여 종합함으로써 과적합을 줄일 수 있음

3. 최종 모델 오류 감소

   → 개별 모델이 서로 다른 유형의 오류를 만들 수 있는데, Ensemble을 통해 모델들의 서로 다른 오류를 상쇄시킬 수 있음

2. Bagging vs Boosting

💡 Boosting 기반 모델로 Bagging 기법 적용

1. Boosting 기반 모델 사용
   → Catboost 모델 사용
   → Boosting 기반 모델 사용하여 Bias 감소

2. Bagging 기법 적용
   → Random Seed Ensemble 사용
   → Catboost 모델 기반으로 Bagging 기법 적용하여 Variance 감소

Bagging과 Boosting을 동시에 사용하여
Bias와 Variance를 동시에 감소시킴

3. Ensemble 적용

💡 Random Seed Ensemble

1. Random Seed

   → 난수 생성에 영향을 주는 hyperparameter

   → 동일한 모델 및 hyperparameter 값을 사용해도 Random Seed 값에 따라 모델 예측 결과가 달라질 수 있음

2. Seed Ensemble

   → Random Seed 값을 다양하게 활용하는 Ensemble 기법

   → 각기 다른 예측을 하는 각각의 Random Seed 값을 갖는 모델들을 조합하여 Robust한 모델을 구축할 수 있음 (9개 값 사용)

05 Appendix

"미사용된 실험들 소개"

변수 분포 단순화하기
customer_country 변수 전처리
customer_idx 처리
Hyper-Parameter Tuning
의미있는 변수 생성 시도
True 비율을 유지하는 데이터 셋 조정

1. 변수 분포 단순화하기

Categorical 변수는 cardinality가 높으면서 빈도수가 적은 category가 많았고,
Numerical 변수는 분포가 매우 넓고 치우친 형태를 띄었음
이를 단순화하여 over/under fitting 을 방지해보자는 아이디어

1-1. 비슷한 category를 통합하기 (cat)

→ category를 통합하는 rule에 주관이 반영되어 데이터 오염이 된 것으로 추정.

1-2. 빈도수가 특정 값 (α) 미만인 변수는 'others'로 치환하기 (cat)

→ 주관이 반영되었고, 의미 있는 category 정보가 누락되었을 수 있을 것으로 추정.

1-3. category 별 영업 전환율에 laplace smoothig 적용 (cat)

→ 초기 영업 전환율이 이후 전환율에 유의미한 영향이 있는지 (?)

1-4. binning 적용 (num)

→ numerical 변수에 대한 CatBoost의 기본 rule이 더 강력한 것으로 추정.

2. customer_country 변수 전처리

customer_country 변수는 text가 정제되지 않아서 활용이 어려움
고객의 나라, 도시 등의 위치 정보를 제대로 반영하기 위해 전처리 진행

customer_country
/Quezon City/Philippines
/PH-00/Philippines
/Kolkata /India
/Bhubaneswar/India
/Hyderabad/India

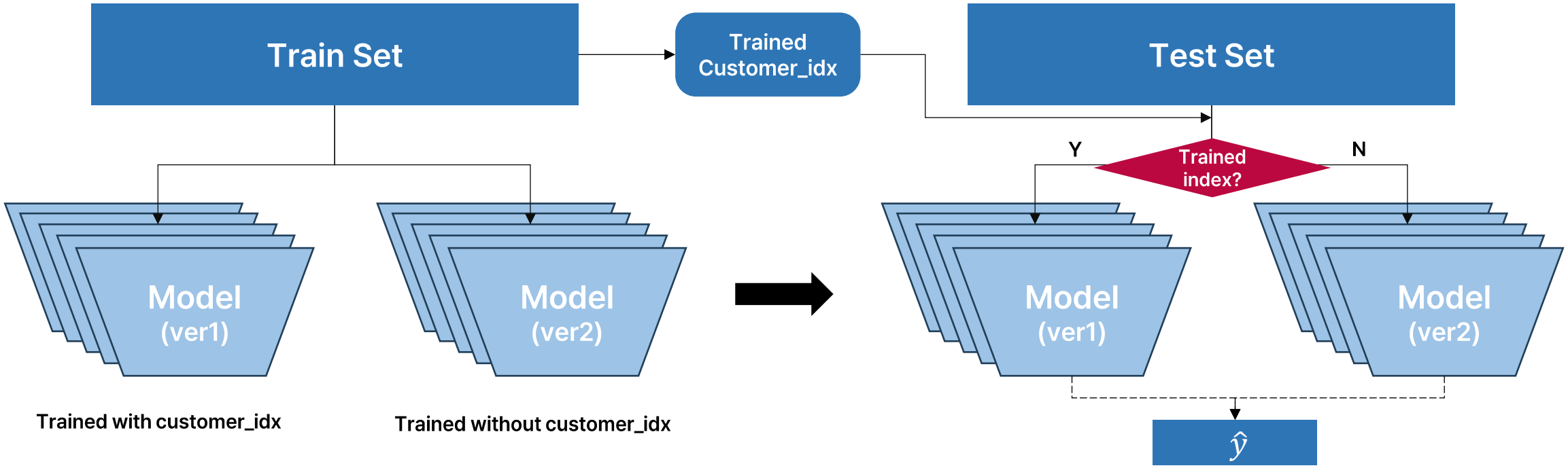
대부분 '/도시/나라' 형태로 저장된 string 자료
→ 소문자 변환, 공백 제거 등 일반적인 text 전처리 진행
→ '/'를 기준으로 split하여 -1, -2번 index를 각각 country, city로 추출

∴ 최종 country : 62,877개, city : 16,873개 non-null 값 추출

▶ 추출된 변수(country, city)의 결측값이 많아서 학습이 잘 되지 않았을 것으로 추정.

3. Unseen customer_idx 처리

customer_idx 변수는 항상 Feature Importance가 컸지만, train set에 속한 고객과 test set에 속한 고객이 달라서 성능 향상에 크게 도움이 되지 않았음
Train에서 학습되지 않은 customer_idx는 모델을 변경해주는 시도



4. Hyper-Parameter Tuning

💡 CatBoost 모형의 최적 hyper-parameter setting을 위한 시도
실험은 앨리스와 로컬, colab 등으로 병렬적으로 진행

4-1. Optuna 활용

: hyper-parameter optimization 프레임워크 optuna로 최적 setting을 적용했습니다.

→ train – test set 간 분포의 차이로 인해 적용이 안되는 것으로 판단.

4-2. Hyper-Parameter Ensemble

: hyper-parameter를 변경하며 5, 7, 9 개 모델을 앙상블

→ robustness 향상에는 도움이 되었을 것으로 추측하는데, 성능이 향상되지는 않음.

5. 의미 있는 변수 생성 시도

💡 EDA를 진행하며 구매 전환의 signal이 되는 변수를 생성하기 위해 노력
특히, 앙상블 모형의 Feature Importance를 참고하여 engineering을 진행

5-1. lead_owner

→ 영업 사원마다 구매 전환율이 다를 것이라고 추측

→ sales 횟수가 많은데도 전환율이 높은 사원들만 추출 (sales_king 변수)

5-2. inquiry_type

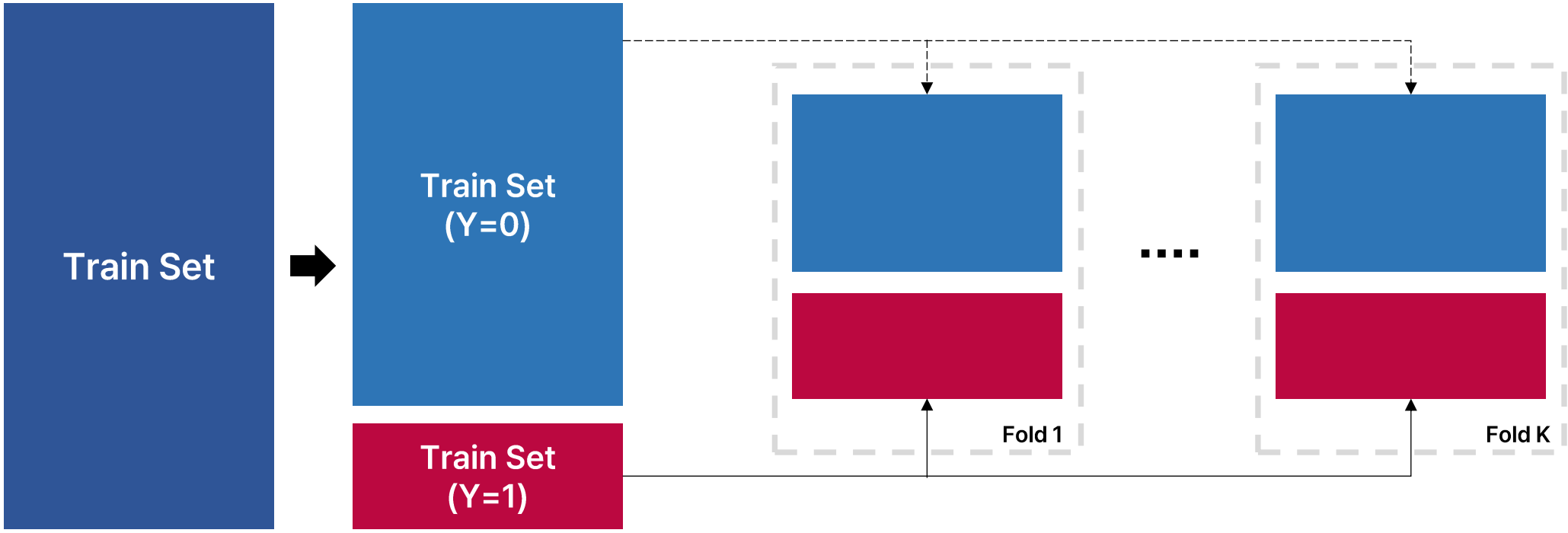
→ inquiry message 내용 중 구매 전환의 단서가 되는 keyword가 있을 것으로 추측

→ 구매 전환이 된 경우와 되지 않은 경우를 Document로 TF-IDF 진행

→ 구매 전환 시 자주 등장한 단어들(keyword)을 언급했는지 여부를 추출 (keyword 변수)

6. True 비율을 유지하는 데이터 셋 조정

Precision보다 Recall이 매우 낮은 점을 해결하기 위한 시도입니다.
True 판별 성능을 높이기 위해 K-fold 시 사용하는 데이터 셋을 변형했습니다.



Q&A

LG Aimers
Data Raiders
(고경준, 안현준, 진실, 유승태, 문정현)