MQL 데이터 기반 B2B 영업기회 창출 예측 모델 개발

코드 리뷰

방국봉팀

목차

Table of Contents

EDA

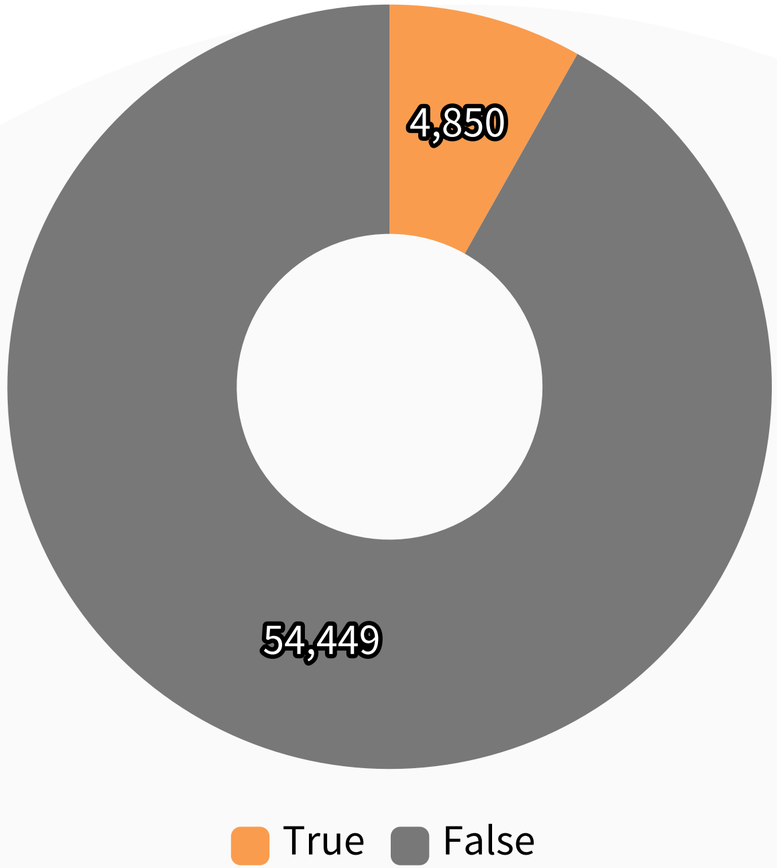
is_converted 불균형

데이터셋 내에서 'is_converted' 타겟 변수의 분포가 심각하게 불균형

- True: 4,850개
- False: 54,449개

해결방안

- 오버샘플링, 언더샘플링 적용
- 클래스 가중치를 계산 후 적용



EDA

고유값, 결측치가 많고 불순물이 포함된 특성

- 일부 특성들은 고유값의 개수가 매우 많고 데이터에 불순물(불필요한 문자 등)이 포함되어 있음

customer_country (15,399개)
customer_idx (35,112개)
customer_job (560개) 등

- 결측치는 다수의 특성에서 다양한 양으로 발견됨

com_reg_ver_win_rate (44,731개)
customer_type (43,961개)
historical_existing_cnt (45,543개) 등

해결방안

- 고유값 개수 축소
- 피처 엔지니어링
- fillna(0)을 사용하여 0으로 대체
- 결측치 많은 특성 제외

Preprocessing

새로운 특성 생성

as_strategic_ver 특성 추가

- business_unit의 중요성 및 AS 데이터의 충분한 양을 바탕으로, 특정 비즈니스 영역에 초점을 맞춘 새로운 특성 생성
- business_area가 'corporate / office' 또는 'hotel & accommodation'이고, business_unit이 'AS'인 경우를 식별하여 as_strategic_ver로 지정

has_historical 및 no_historical 특성 생성

- historical_existing_cnt의 유무만으로도 중요한 예측 정보를 제공할 수 있음을 발견하고, 이를 명확히 구분하는 특성 생성.
- historical_existing_cnt의 값이 있는 경우와 없는 경우를 각각 has_historical과 no_historical로 구분하여 표현.
- 여러 접근 방식을 시도한 결과, 이 간단한 구분이 모델 성능에 가장 긍정적인 영향을 미침을 발견.

Preprocessing

특성 변환 및 인코딩

lead_desc_length를 구간으로 나누고 원핫인코딩 적용

- lead_desc_length의 광범위한 숫자 범위의 복잡성을 감소시키고 모델이 중요한 정보를 더 쉽게 인식할 수 있도록 유용한 형태로 변환.
- 데이터의 분포도 등을 고려하여 다양한 방법으로 구간을 나누어 실험함.
- 결국, 단순한 5등분 전략이 데이터를 가장 잘 대표하고 모델 성능에 긍정적인 영향을 미침을 확인.
- [0, 252], [252, 504], [504, 756], [756, 1008], [1008, 1264]으로 구간 설정

문자열 타입의 특성에 대한 레이블 인코딩 수행.

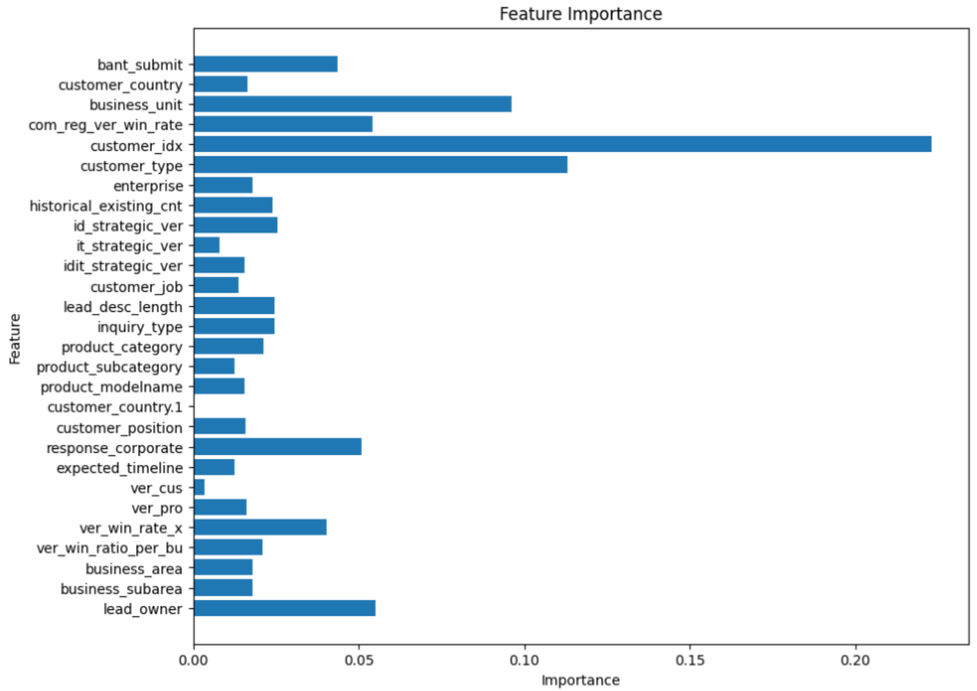
- 문자열 형태의 데이터를 모델이 이해할 수 있는 숫자형 데이터로 변환.

결측치 처리

- 일반 결측치 처리
  대부분의 결측치를 fillna(0)을 사용해 0으로 대체
- 특수 경우 처리
  com_reg_ver_win_rate 같은 특정 특성에서는 기존 0값과 구분하기 위해 결측치를 -999로 대체

Model

최적 특성 조합을 통한 모델 성능 최적화



배경

전체 특성을 사용했을 때와 비교하여, 적절한 특성 조합이 모델 성능에 미치는 영향 평가 필요

Feature Importance 분석 비교

XGBoost를 이용해 초기 특성 중요도 평가 후 피처 중요도가 높은 특성 위주로 여러 조합을 실험하여, 모델 성능 비교 분석

분석 결과

전체 특성을 사용했을 때보다 성능이 향상되는 최적의 특성 조합 발견.
너무 많은 특성 또는 너무 적은 특성 모두 성능 저하의 원인

최종 선택된 특성

"com_reg_ver_win_rate", "customer_idx", "customer_type", "inquiry_type", "it_strategic_ver", "has_historical", "no_historical", "response_corporate","as_strategic_ver", "lead_owner", "desc_length_Short", "desc_length_Medium", "desc_length_Long", "desc_length_Very Long", "desc_length_Very Short"

Model

앙상블 모델 최적화 전략 및 성능 개선

앙상블 모델 선정 과정

모델 조합 실험
여러 앙상블 모델(랜덤 포레스트, XGBoost, CatBoost 등)
다양한 조합으로 실험

앙상블 방식 탐색
보팅, 스태킹, 배깅 등 다양한 앙상블 방식을 적용해 성능 비교.

결정 과정
- XGBoost와 CatBoost 조합이 다른 모델 조합에 비해 가장 우수한 성능을 보임
- 앙상블 기법 중 보팅 기법이 가장 효과적
- 보팅 방식에서는 soft 방식이 hard 방식보다 성능이 더 우수함을 확인

파라미터 최적화

Optuna 활용
Optuna를 사용하여
XGBoost와 CatBoost의
파라미터 최적화 수행

성능 개선
최적화된 파라미터를 적용한
결과, 모델의 F1 점수와
정확도에서
눈에 띄는 개선 확인.

최종 앙상블 모델

모델 구성
VotingClassifier를 이용한
XGBoost와 CatBoost의 소프트 보팅 앙상블 모델

기대효과
- 서로 다른 특징을 가진 두 모델의 조합으로 다양한 데이터 패턴을 효과적으로 포착.
- 최적화된 파라미터를 통한 성능 극대화로 더 정확하고 신뢰도 높은 예측 가능.

Model

불균형 데이터 처리를 위한 클래스 가중치 적용

XGBoost

- XGBoost는 불균형 데이터를 처리하기 위한 자체적인 가중치 조정 기능이 없음.

- 이에 따라, 클래스 가중치를 수동으로 계산하고 모델 학습 시 적용해야 함.

- 가중치 계산 방법
  df_train_prepared['is_converted']를 기준으로 각 클래스 별 가중치를 계산하고, 모든 샘플에 적용하여 모델의 학습 과정에서 불균형을 조정.

CatBoost

- CatBoost는 불균형 데이터를 자동으로 처리할 수 있는 auto_class_weights='Balanced' 옵션을 제공함

- 이 옵션을 사용하면, CatBoost가 데이터의 불균형을 감지하고 자동으로 가중치를 조정함.

Model

교차 검증과 최종 모델 학습

교차 검증

- 선택된 앙상블 모델에 대해 5-폴드 교차 검증을 적용하여, 다양한 데이터 분할에 대한 모델의 성능 평가

최종 모델 학습

- 교차 검증 결과를 바탕으로 최적화된 파라미터와 클래스 가중치를 적용한 최종 모델 학습

검증 결과

교차 검증 F1 점수
[0.87751938 0.62544031 0.55123675 0.92956243 0.55297863]

평균 F1 점수
0.7073475004038701

Public Score
0.759836