신용카드 고객 세그먼트 분류기 개발 및 활용성 검토

3조: 손효은 염기진 이우태 최다은

iM Bank

소개 목차

프로젝트 개요

- 기획배경
- 프로젝트 진행과정

프로젝트 팀 구성 및 역할

- 역할분담

프로젝트 수행절차 및 방법

- 프로젝트 진행과정 소개

II. 데이터 소개

프로젝트 수행 경과

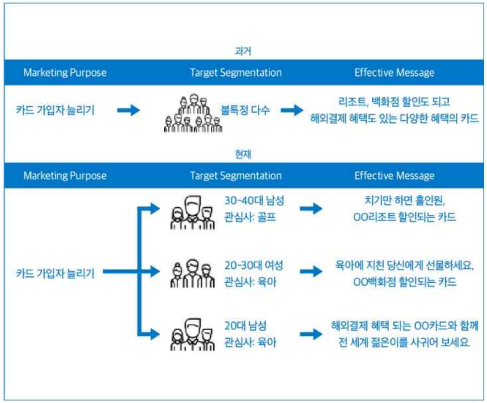
- 데이터 전처리 및 통합
- 분류 모델 구축

자체 평가 의견

- 결과해석
- 인사이트
- 자체 평가
- 출처 및 참고문헌

iM Bank

기획 배경



출처: 디지털 인사이트

기존 방식 (Before)

- 불특정 다수를 대상으로 카드 혜택 메시지 일괄 발송
- 높은 광고 비용 대비 낮은 반응률

개선된 방식 (After)

- 고객을 인구통계학적 기준으로 세분화
- 각 그룹의 관심사에 맞는 맞춤형 메시지 제공
- 타겟 고객의 가입률 증가
- 구매 가능성이 높은 고객에게만 접근 → 비용 효율성 향상

기획 배경

KB국민카드 AI 기반 마케팅 자동화 플랫폼

국민카드는 **AI 기반 마케팅 자동화 플랫폼 AIMs'를 도입**하여 고객의 **세그먼트를 세분화**하여 마케팅에 활용
→ 마케팅 효율을 높여 **비용 절감효과**를 거둠

개인화 마케팅을 통한 비용 축소

일반관리비 역시 지난해보다 줄어들면서 순이익 증가에 힘을 보탰다. 일반관리비는 올해 상반기 2896억원으로 전년동기대비 4.3% 감소했다. 일반관리비가 줄었다는 것은 마케팅, 모집 등 영업 관련 전반에 들어가는 비용을 효과적으로 축소했다는 의미다.

출처: 딜사이트https://dealsite.co.kr/articles/125815

프로젝트 진행과정

데이콘 데이터로 신용카드 고객 세그먼트 분류기 모델 제작

최종 분류기를 활용해 iM뱅크 데이터 고객 세그먼트 분류

iM Bank

역할 분담

이름	역할
손효은	EDA, 피처 엔지니어링, 샘플 모델링
염기진	EDA, 데이터 전처리, 모델 성능 개선
이우태(팀장)	데이터 전처리, 피처 엔지니어링, 샘플 모델링
최다은	EDA, 모델 성능 개선

프로젝트 진행과정 소개

단계	기간	주요 활동
사전 기획	4월 15일 ~ 4월 16일	프로젝트 기획서 작성
데이터 구조 이해 및 EDA 수행	4월 17일 ~ 4월 18일	데이터 구조 파악 및 타겟 변수 검토
데이터 전처리	4월 19일 ~ 4월 20일	주요 변수 전처리
모델링 및 초기 성능 평가	4월 21일 ~ 4월 22일	분류 모델 적용 및 베이스라인 성능 확보
모델 고도화 및 인사이트 도출	4월 23일 ~ 4월 24일	성능 개선 및 iM뱅크 데이터 테스트
최종 발표	4월 25일	최종 결과 시각화 및 발표 자료 작성

데이터 소개

■ DACON

- 데이터 구성:
  - 기간: 2018년 7월 ~ 2018년 12월(6개월)
  - 대상: 40만 명 고객
  - 분할: train / test (월별, 고객별 분리 구조)
  - 파일 수: 총 96개 (8개 유형 × 6개월 × 2 splits)
  - 데이터 유형:
    - 회원정보, 신용정보, 승인매출정보, 청구입금정보, 잔액정보, 채널정보, 마케팅정보, 성과정보
  - 데이터 형식: .parquet

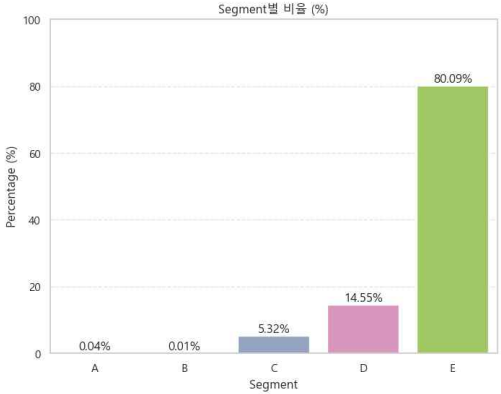
- 타겟 변수:
  - Segment (회원정보 테이블 내 포함)
  - 고객을 A~E 그룹으로 분류 (다중 분류 문제)

데이터 소개

■ iM뱅크

- 제공자: iM뱅크
- 데이터 수집 기간: 2021년 1월 ~ 2023년 12월
- 데이터 파일 구성: 총 6개 파일
  - 고객 데이터: 연도별 고객 정보 파일 (3개)
  - 카드 데이터: 연도별 카드 승인 거래 정보 파일 (3개)
- 데이터 규모:
  - 고객 정보: 연간 약 2천만 건 (총 약 6천만 건)
  - 카드 승인: 연간 약 6천만 건 (총 약 1억 8천만 건)
- 주요 변수:
  - 고객 정보: 연령대, 성별, 고객등급, 자택_시도, 수신_요구불예금, 대출금액 등 금융자산 속성
  - 카드 정보: 승인건수, 승인금액, 가맹점 업종명, 가맹점 지역, 거래년월 등 소비 패턴 속성

EDA



<Segment 분포 분석>

- E가 전체의 약 80%를 차지함
- A,B는 매우 희소(0.05% 미만)하며, **불균형**이 큼
- **세그먼트 별 분포를 고려**해야 함

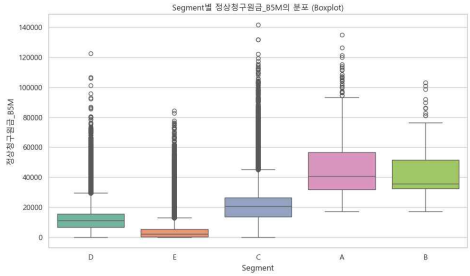
EDA

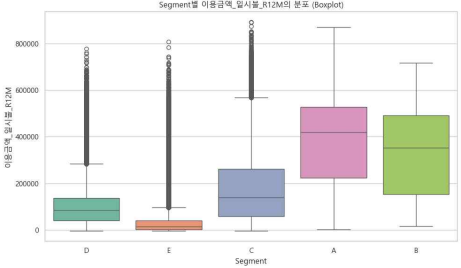
<Segment 상관계수 상위 10개 피처>

- 상관계수가 **0.5 이상**으로 이루어짐
- 세그먼트와 상관계수가 높은 피처들은 **금액과 관련된 피처**라는 특징을 가짐

iM Bank

EDA





<상관계수가 높았던 피처의 Segment 분포 분석>

- **E는 낮은 금액대에 밀집**된 형태를 띄고, 나머지는 더 넓고 높은 금액대에 분포
- E -> A로 갈수록 전반적으로 높은 금액대에 분포함
- A,B에 비해 인스턴스가 많은 C,D,E에서는 이상치가 많음

결측치

컬럼명	비율(소수 첫째자리에서 반올림)	처리 방법
OS구분코드	68%	'그외'로 대체
가입통신회사코드	17%	'그외'로 대체
직장시도명	10%	'없음'으로 대체
_1순위신용체크구분	0.5%	'없음'으로 대체
Rv전환가능여부	0.01%	인스턴스 제거
연체일자_B0M	99%	0으로 대체
혜택수혜율_B0M	23%	0으로 대체
혜택수혜율_R3M	18%	0으로 대체

iM Bank

컬럼명	비율(소수 첫째자리에서 반올림)	처리 방법
RV신청일자	81%	0으로 대체
최종유효년월_신용_이용가능	8%	0으로 대체
최종카드발급일자	0.8%	0으로 대체
최종유효년월_신용_이용	20%	0으로 대체
최종카드론_대출일자	83%	0으로 대체
최종카드론_신청경로코드 최종카드론_금융상환방식코드	82%	0으로 대체
_3순위여유업종	98%	'없음'으로 채움
_3순위쇼핑업종	54%	
_3순위납부업종	96%	
-3순위교통업종	84%	
_3순위업종	44%	
_2순위신용체크구분	39%	
_2순위여유업종	92%	
_2순위쇼핑업종	46%	
2순위납부업종	84%	
_2순위교통업종	67%	
_2순위업종	35%	
_1순위여유업종	74%	
_1순위쇼핑업종	36%	
_1순위납부업종	49%	
_1순위교통업종	47%	
_1순위업종	20%	

데이터 전처리

컬럼명	처리 방법
이용카드수_체크_가족 이용금액_R3M_체크_가족 연회비할인카드수_B0M 할인금액_기본연회비_B0M 할인금액_제휴연회비_B0M ... 할부건수_부분_6M_R12M 할부건수_부분_14M_R12M 컨택건수_FDS_B0M 컨택건수_FDS_R6M	하나의 값만 갖는 데이터라서 삭제 → 참고 문헌에 따르면, 모든 샘플에 값이 동일한 피처는 분류 과정에 아무런 정보를 제공하지 않는다고 명시(Teodorescu, M. H.).

이상치

컬럼명	비율(소수 첫째자리에서 반올림)	처리 방법
rv최초시작후경과일	81%	99999999 -> 0으로 대체
연체일수_B1M 연체일수_B2M	87%	-999999 -> 0으로 대체
최초카드론이용경과월 최종카드론이용경과월	83%	999 -> 0으로 대체
최종이용일자_기본 최종이용일자_신판 최종이용일자_CA 최종이용일자_카드론 최종이용일자_체크 최종이용일자_일시불 최종이용일자_할부	4% 4% 68% 83% 63% 4% 28%	10101-> 0으로 대체 후 라벨인코딩

형변환

컬럼명	변경 전 타입	변경 후 타입
기준년월	int64	datetime64[ns]
입회일자_신용	int64	int32
최종유효년월_신용_이용가능	float64	
최종유효년월_신용_이용	float64	
최종카드발급일자	float64	
RV신청일자	float64	
연체일자_B0M	float64	str
청구서발송여부_B0 청구서발송여부_R3M 청구서발송여부_R6M	int64	category

iM Bank

형변환

컬럼명	변경 전 타입	변경 후 타입
남녀구분코드 회원여부_이용가능 회원여부_이용가능_CA 회원여부_이용가능_카드론 소지여부_신용 회원여부_연체 이용거절여부_카드론 동의여부_한도증액안내 수신거부여부_TM 수신거부여부_DM 수신거부여부_메일 수신거부여부_SMS 마케팅동의여부 보유여부_해외겸용_본인 이용가능여부_해외겸용_본인 이용여부_3M_해외겸용_본인 보유여부_해외겸용_신용_본인 이용가능여부_해외겸용_신용_본인 이용여부_3M_해외겸용_신용_본인 일시불ONLY전환가능여부 연체감액여부_R3M	int64	category

형변환

컬럼명	변경 전 타입	변경 후 타입
연회비발생카드수_B0M	object	int32
자발한도감액횟수_R12M		
한도심사요청건수		
최종카드론_대출일자	int64	str
최종카드론_금융상환방식코드	float64	category
회원여부_이용가능	object	int64
회원여부_이용가능_CA		
회원여부_이용가능_카드론		
소지여부_신용		
회원여부_연체		
이용거절여부_카드론		
일시불ONLY전환가능여부		

iM Bank

형변환

컬럼명	변경 전 타입	변경 후 타입
연령	object	int64
연회비발생카드수_B0M		
한도증액횟수_R12M		
한도심사요청건수		
이용금액대		
할인건수_R3M		
할인건수_B0M		
인입횟수_ARS_R6M		
이용메뉴건수_ARS_R6M		
방문횟수_PC_R6M		
방문일수_PC_R6M		
방문횟수_앱_R6M		
캠페인접촉건수_R12M		
캠페인접촉일수_R12M		

iM Bank

인코딩

컬럼명	처리 방법
최종카드론_금융상환방식코드 남녀구분코드 회원여부_이용가능 회원여부_이용가능_CA 회원여부_이용가능_카드론 소지여부_신용 회원여부_연체 이용거절여부_카드론 동의여부_한도증액안내 수신거부여부_TM 수신거부여부_DM 수신거부여부_메일 수신거부여부_SMS 마케팅동의여부 보유여부_해외겸용_본인 이용가능여부_해외겸용_본인 이용여부_3M_해외겸용_본인 보유여부_해외겸용_신용_본인 이용가능여부_해외겸용_신용_본인 이용여부_3M_해외겸용_신용_본인 일시불ONLY전환가능여부 연체감액여부_R3M 시장단기연체여부_R6M 시장단기연체여부_R3M 청구서발송여부_B0 청구서발송여부_R3M 청구서발송여부_R6M	수치형이지만 범주의 의미를 가져 category 타입으로 변환 후, One-hot Encoding 진행

iM Bank

인코딩

컬럼명	처리 방법
가입통신회사코드 거주시도명 직장시도명 _1순위신용체크구분 _2순위신용체크구분 Life_Stage 카드론동의여부 RV전환가능여부 _1순위업종 _2순위업종 _3순위업종 _1순위소핑업종 _2순위소핑업종 _3순위소핑업종 _1순위교통업종 _2순위교통업종 _3순위교통업종 _1순위여유업종 _2순위여유업종 _3순위여유업종 _1순위납부업종 _2순위납부업종 _3순위납부업종 최종카드론_신청경로코드 대표청구지고객주소구분코드 대표청구서수령지구분코드 청구서수령방법 OS구분코드	One-hot Encoding

인코딩

컬럼명	처리 방법
연령 연회비발생카드수_B0M 자발한도감액횟수_R12M 한도증액횟수_R12M 한도심사요청건수 이용금액대 할인건수_R3M 할인건수_B0M 인입횟수_ARS_R6M 이용매뉴건수_ARS_R6M 방문횟수_PC_R6M	Label Encoding

iM Bank

1차 피처 선정 기준

<관계의 크기를 설명하는 기준>

Hopkins (1997)

Value or r	Description
0.9 - 1.0	Nearly perfect, distinct
0.7 - 0.9	Very large, very high
0.5 - 0.7	High, large, major
0.3 - 0.5	Moderate, medium
0.1 - 0.3	Low, small, minor
0.0 - 0.1	Trivial, very small, insubstantial

출처: New view of statistics: Effect magnitudes
https://www.sportsci.org/resource/stats/effectmag.html

전체 피처로 모델링시 Test성능이 낮게 나타남

→ 참고 논문을 기반으로, 목표 변수인 Segment와의 상관계수가 0.1 이상인 피처만 선별하여 모델을 재구성

피처선정별 모델 성능

- 전체 피처를 사용한 Xgboost성능

Train_F1 Score: 0.93

Test_F1 Score: 0.22

- Segment와의 상관계수 ≥ 0.1 피처사용 Xgboost성능

Train_F1 Score: 0.92

Test_F1 Score: 0.49

- Segment와의 상관계수 ≥ 0.3 피처사용 Xgboost성능

Train_F1 Score: 0.9

Test_F1 Score: 0.46

'Test 성능이 가장 높은 상관계수 0.1 이상 피처를 최종 선정'

iM Bank

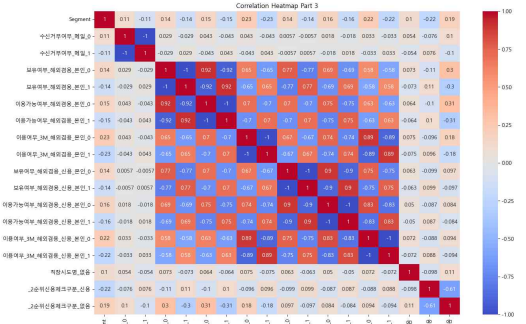
상관계수 도출

```
# 제외할 컬럼 목록
exclude_cols = ['ID', '기준년월']

# 상관계수 계산 (Segment 기준)
correlations = final_df.drop(columns=exclude_cols).corr(numeric_only=True)['Segment'].drop('Segment')

# 상관계수 0.1 이상인 피처 추출
high_corr_features = correlations[correlations.abs() >= 0.1]

# 결과 출력
print("Segment와 상관계수(|r| ≥ 0.1)인 피처 수:", len(high_corr_features))
print("\n해당 피처들:\n", high_corr_features.sort_values(ascending=False))
```



Segment와 상관계수(|r| ≥ 0.1)인 피처 수: 432개

iMBank

샘플링

<resample + SMOTE >

Test Classification Report:

	precision	recall	f1-score	support
A	0.65	0.83	0.73	291
B	0.92	0.84	0.88	43
C	0.57	0.83	0.68	38277
D	0.55	0.76	0.64	104773
E	0.98	0.88	0.93	576616
accuracy			0.86	720000
macro avg	0.74	0.83	0.77	720000
weighted avg	0.90	0.86	0.87	720000

<resample>

Test Classification Report:

	precision	recall	f1-score	support
A	0.54	0.92	0.68	291
B	0.85	0.77	0.80	43
C	0.57	0.83	0.67	38277
D	0.55	0.76	0.64	104773
E	0.98	0.88	0.93	576616
accuracy			0.86	720000
macro avg	0.70	0.83	0.74	720000
weighted avg	0.90	0.86	0.87	720000

출처: Handling Method of Imbalance Data for Machine Learning

관련 문헌에 따르면, 불균형이 심한 데이터의 경우 **resample** 방식으로 샘플링하는 것을 추천한다고 함.
이를 참고하여 두 가지 방식의 샘플링을 진행한 결과, 테스트는 **SMOTE** 샘플링한 모델의 **F1-score** 값이 더 높았으나 실제 데이콘 제출 결과 **SMOTE**의 **F1-score**값이 더 낮아 **resample**을 최종 샘플링 방식으로 선정함

iM Bank

샘플링

샘플링 기준	F1-score
A	0.3831
B	0.2869
C	0.4010
D	0.4015
E	0.4021

<E 기준 샘플링 결과>

Classification Report:

	precision	recall	f1-score	support
A	0.89	0.95	0.92	1922052
B	0.95	1.00	0.97	1922052
C	0.73	0.63	0.68	1922052
D	0.67	0.67	0.67	1922052
E	0.82	0.83	0.83	1922052
accuracy			0.82	9610260
macro avg	0.81	0.82	0.81	9610260
weighted avg	0.81	0.82	0.81	9610260

<샘플링 기준 선정>

- resample 방법으로 각 기준에 따라 기본 모델로 샘플링 진행
- 데이콘 F1-score 기준, **E로 샘플링**을 했을때 가장 높은 성능이 나타났으나 C,D,E 값이 매우 유사함

→ **C,D,E를 기준으로 resample 방법으로 샘플링** 해보기로 함

iM Bank

피처 엔지니어링

Table 1. OLS and LR Performance by Multicollinearity Condition

λ_x	VIF Range (rounded)	Mean Performance $\rho^2 = .25$ OLS	$\rho^2 = .25$ LR	$\rho^2 = .75$ OLS	$\rho^2 = .75$ LR
.30	340 to 2000	.6136	.6127	.7841	.7808
.40	49 to 232	.6102	.6103	.7820	.7807
.50	12 to 39	.6082	.6077	.7821	.7769
.65	3 to 6	.6121	.6115	.7857	.7813
.80	1 to 2	.6085	.6083	.7809	.7767
.95	< 1	.6064	.6060	.7761	.7723
F(5,66) =		0.22	0.19	0.26	0.19

출처: The Precise Effect of Multicollinearity on Classification Prediction

<다중공선성 파악>

- 현재 1차적으로 목표변수와 상관계수가 0.1미만인 피처 제거된 상태
- 관련 문헌에 따르면, **분류 모델에서는 다중공선성이 모델 성능 향상에 큰 영향을 미치지 않는다**고 보고됨
- VIF 값이 큰 값이 매우 많았으나, 실제로 VIF 기준으로 피처를 제거하니 오히려 성능이 떨어짐

→ **다중공선성 피처는 제거하지 않음**

iM Bank

피처 엔지니어링

	Feature_1	Feature_2	Correlation
0	수신거부여부_TM_0	수신거부여부_TM_1	-1.0
3	보유여부_해외겸용_본인_0	보유여부_해외겸용_본인_1	-1.0
4	이용가능여부_해외겸용_본인_0	이용가능여부_해외겸용_본인_1	-1.0
5	이용여부_3M_해외겸용_본인_0	이용여부_3M_해외겸용_본인_1	-1.0
6	보유여부_해외겸용_신용_본인_0	보유여부_해외겸용_신용_본인_1	-1.0
8	이용여부_3M_해외겸용_신용_본인_0	이용여부_3M_해외겸용_신용_본인_1	-1.0
9	청구서발송여부_B0_0	청구서발송여부_B0_1	-1.0
10	청구서발송여부_R3M_0	청구서발송여부_R3M_1	-1.0
11	청구서발송여부_R6M_0	청구서발송여부_R6M_1	-1.0

<피처간 상관계수 파악>

- **전체 피처간 상관계수가 절댓값 1**인 피처쌍 중 하나 제거(12개)
- 원핫인코딩으로 생겨난 피처들임
- 중복 피처로 인한 과적합 위험을 줄이기 위함

-> 실제로 성능이 향상됨

모델 선정

< 부스팅 계열 모델(XGBoost, LightGBM, CatBoost) 사용 이유>

① 클래스 불균형에 강한 성능

부스팅 모델은 각 단계에서 이전 단계가 틀린 샘플에 더 집중하는 방식이라, 소수 클래스(A,B)에 대한 학습 효과가 상대적으로 좋음.

-> 참고 문헌에 따르면, 그라디언트 부스팅은 다른 머신러닝 알고리즘에 비해 불균형한 분류 작업에서도 좋은 성과를 거둠(Benkendorf, D. J.).

② 고차원의 많은 피처 처리에 강함

수백 개의 피처가 존재하는 상황에서 부스팅 모델은 불필요한 피처는 자동으로 무시하거나 가중치를 낮춰줌.

-> 참고 문헌에 따르면, GBDT는 불균형한 분류 문제에서도 다른 알고리즘보다 우수한 성능을 보임(Si, Si).

04 프로젝트 수행경과

iM Bank

모델링

<Soft Voting에 활용한 모델 구성>

```
# 개별 모델 정의
lgbm = LGBMClassifier(random_state=42)
xgb = XGBClassifier(use_label_encoder=False, eval_metric='logloss', random_state=42)
cat = CatBoostClassifier(verbose=0, random_state=42)

# 앙상블 모델 정의 (soft voting = 확률 기반 평균)
voting_dlf = VotingClassifier(
    estimators=[
        ('lgbm', lgbm),
        ('xgb', xgb),
        ('cat', cat)
    ],
    voting='soft'
)
```

<C샘플링 모델 성능 비교>

XGBoost : 0.62
LightGBM : 0.62
CatBoost : 0.62
Logistic : 0.4
Decision: 0.41

Segment와의 상관계수 0.1 이상의 피처들로만 모델링 했을 때 XGBoost, LightGBM,CatBoost의 성능이 가장 高

→ 위 3개 모델을 활용한 Soft Voting Test_F1_Score: 0.629

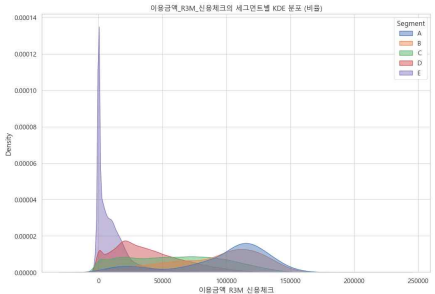
모델링

Soft Voting

여러 개의 모델이 각각 예측한 확률을 평균내서, 가장 확률이 높은 클래스 선택

Hard Voting

각 모델이 예측한 클래스(정답 레이블) 중 다수결로 결정



<Soft Voting 선정이유>

Hard Voting(다수결)보다 **클래스 간 경계가 불확실한 경우** 더 유연하고 **안정적인 예측** 가능

클래스 불균형 상황일 경우, **소수 클래스에 대한 예측 확률 정보**를 반영할 수 있어 f1score 개선에 도움을 줌.

iM Bank

모델링

<Soft Voting 샘플링 별 모델 성능 비교>

앞서 C,D,E샘플링의 성능에 유의미한 차이가 없었기 때문에 최종 모델로 각 샘플링별 모델 성능 테스트

C_Test_F1_Score: 0.642
D_Test_ F1_Score: 0.649
E_Test_F1_Score: 0.640

<최종 모델 Classification Report>

```
Confusion Matrix:
[[127590      0      0      0      0]
 [     0 127590      0      0      0]
 [   271      0 110057  15235   2027]
 [    28      0  15701 102537   9324]
 [     2      0   1402  11136 115050]]
Classification Report:
              precision    recall  f1-score   support

           A       1.00      1.00      1.00    127590
           B       1.00      1.00      1.00    127590
           C       0.87      0.86      0.86    127590
           D       0.80      0.80      0.80    127590
           E       0.91      0.90      0.91    127590

    accuracy                           0.91    637950
   macro avg       0.91      0.91      0.91    637950
weighted avg       0.91      0.91      0.91    637950
```

Test _F1_Score: 0.649

iM Bank

모델 튜닝

<튜닝 방법>

```
# 1. 개별 모델 정의 (튜닝 포함)
lgbm = LGBMClassifier(n_estimators=200, learning_rate=0.05, max_depth=7, random_state=42)
xgb = XGBClassifier(n_estimators=200, learning_rate=0.05, max_depth=6,
                    use_label_encoder=False, eval_metric='logloss', random_state=42)
cat = CatBoostClassifier(iterations=200, learning_rate=0.05, depth=6, verbose=0, random_state=42)

# 2. Soft Voting 앙상블 (가중치 부여)
voting_clf = VotingClassifier(
    estimators=[('lgbm', lgbm), ('xgb', xgb), ('cat', cat)],
    voting='soft',
    weights=[1.5, 1, 1.2]  # lgbm > catboost > xgb
)
```

<튜닝 모델 Classification Report>

```
✅ F1 Score(macro): 0.8867156611803685
✅ Confusion Matrix:
[[349242      0      0      0      0]
 [     0 349242      0      0      0]
 [  6024     31 280744  54121   8322]
 [   328     12  53922 261709  33271]
 [    27      1   4481  36826 307907]]
✅ Classification Report:
              precision    recall  f1-score   support

           0       0.98      1.00      0.99    349242
           1       1.00      1.00      1.00    349242
           2       0.83      0.80      0.82    349242
           3       0.74      0.75      0.75    349242
           4       0.88      0.88      0.88    349242

    accuracy                           0.89   1746210
   macro avg       0.89      0.89      0.89   1746210
weighted avg       0.89      0.89      0.89   1746210
```

Test_F1_Score: 0.596

iM Bank

최종 모델 선정

<기본 모델 Classification Report>

```
Confusion Matrix:
[[127590      0      0      0      0]
 [     0 127590      0      0      0]
 [   271      0 110057  15235   2027]
 [    28      0  15701 102537   9324]
 [     2      0   1402  11136 115050]]
Classification Report:
              precision    recall  f1-score   support

           A       1.00      1.00      1.00    127590
           B       1.00      1.00      1.00    127590
           C       0.87      0.86      0.86    127590
           D       0.80      0.80      0.80    127590
           E       0.91      0.90      0.91    127590

    accuracy                           0.91    637950
   macro avg       0.91      0.91      0.91    637950
weighted avg       0.91      0.91      0.91    637950
```

<최종 모델링 전략 >

- **Soft Voting**(LightGBM, XGBoost,CatBoost) 모델 사용
- Segment와의 상관계수 ≥ 0.1 피처 선정
- 피처간 상관관계 = 1 제거
- Segment가 'D'인 클래스 기준으로 Resampling

#	팀	팀 멤버	점수	제출수	등록일
46	최다은조	손 염 최 이	0.64902	17	8시간 전

iM뱅크 전처리

<card data>

컬럼명	처리 방법
가맹점 시군구	결측치 존재 행 제거

<customer data>

컬럼명	처리 방법
자택 시군구	결측치 존재 행 제거
기준년월	칼럼 삭제(거래년월과 겹치므로)

고객ID와 거래년월을 기준으로 card data에 left merge

iM뱅크 전처리

<merged data>

컬럼명	처리 방법
고객등급 성별 수신_외화예금 수신_펀드 연령대 수신_적립식예금 수신_거치식예금 수신_요구불예금 자택_시군구 자택_시도 대출금액	결측치 존재 인스턴스 제거 (card data 에 있음, customer data 에 없음)

iM뱅크 전처리

<merged data>

컬럼명	처리 방법
가맹점업종명 가맹점_광역시도 가맹점_시군구 수신_거치식예금 수신_적립식예금 수신_외화예금 수신_펀드 자택_시군구	컬럼 제거 (데이콘 데이터에 존재 X)

iM뱅크 전처리

<merged data>

인스턴스명	처리 방법
'경상북도': '경북', '울산광역시': '울산', '대구광역시': '대구', '서울특별시': '서울', '대전광역시': '대전', '충청북도': '충북', '경상남도': '경남', '부산광역시': '부산', '경기도': '경기', '충청남도': '충남', '강원특별자치도': '강원', '인천광역시': '인천', '제주특별자치도': '제주', '전북특별자치도': '전북', '광주광역시': '광주', '전라남도': '전남'	거주시도명 칼럼 고유값 수정 (데이콘 데이터와 이름 맞춰주기)

iM뱅크 전처리

<merged data>

컬럼명	처리 방법
거래년월	datetime으로 데이터 타입 변경

컬럼명	처리 방법
승인건수 연령대 고객등급	라벨 인코딩(지정인코딩)

컬럼명	처리 방법
남녀구분코드	성별 -> 남녀구분코드로 칼럼명 변경 후 원핫인코딩

iM뱅크 전처리

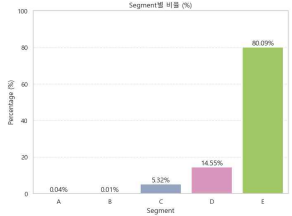
<merged data>

컬럼명	처리 방법
거주시도명	원핫인코딩

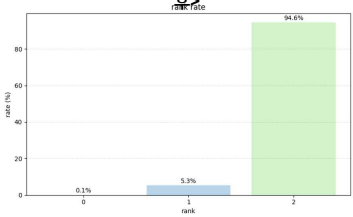
컬럼명	처리 방법
수신_요구불예금 대출금액	데이터 타입 변경(Dacon 데이터와 타입 통일) (float -> int)

세그먼트 분포 비교

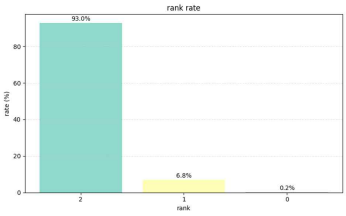
<데이콘 세그먼트 별 분포 비율_조정 전>



<데이콘 세그먼트 별 분포 비율_조정 후>



<iM뱅크 고객등급 별 분포 비율>



iM뱅크 데이터와 데이콘 데이터의 세그먼트 별 분포 비율을 맞춰주기 위해,

A/B → 최우수, C → 우수등급, D/E → 일반등급으로 조정

iM뱅크 모델링

```
Accuracy: 0.230049950241020 35
Classification Report:
              precision    recall  f1-score   support

           0       0.00      0.01      0.00    358911
           1       0.08      0.92      0.14  12364130
           2       0.97      0.18      0.30 167762975

    accuracy                           0.23 180486016
   macro avg       0.35      0.37      0.15 180486016
weighted avg       0.91      0.23      0.29 180486016
```

<최종 모델로 iM뱅크 데이터 분류>

데이콘 데이터에서 좋은 성능을 보였던 최종 모델은, iM뱅크 데이터에서는 기대만큼의 성능을 내지 못함

iM Bank

결론 도출 및 인사이트

<결론도출>

1. Segment 에서**'E'는 낮은 금액대에 밀집**되어 있고, **'A'로 갈 수록 높은 금액대**에 분포
2. C, D, E와 같은 중·하위 세그먼트에서는 이상치가 많고 인스턴스 수도 많음
3. Segment와 상관관계가 높은 피처들은 대부분 금액 관련 변수로 구성
4. 피처 간 상관계수 ≥ 1인 중복 피처는 제거하고, Segment와 상관계수 ≥ 0.1인 피처만 사용하여 정보성 강화

<인사이트>

1. Segment 구간별 금액 분포 특성이 비교적 뚜렷하며, 금액 관련 피처가 핵심 변수로 작용
2. 세그먼트 정의 및 비율 차이가 모델 일반화에 영향을 미침
   → 데이터 셋마다 적절한 등급 조정 및 클래스 균형 처리가 필요
3. 고차원, 불균형 데이터에 대해 Gradient Boosting 기반 **앙상블 모델이 안정적인 성능**을 보임
4. 그러나 데이터 도메인 특성이 다르면 성능이 크게 달라질 수 있음
   → 일반화 가능한 전처리 전략과 피처 선택 기준이 중요

자체 평가 의견

<잘한 점>

- 다양한 샘플링을 통해 클래스 불균형을 해결함
- 데이터 용량이 매우 커 반복 학습에 시간이 많이 소요되었기 때문에, **여러 연구를 참고**해 의미 있는 변수 중심으로 데이터를 선별한 뒤 모델링을 진행
- **피클로 모델을 저장**하여 재사용으로 인한 메모리 및 시간 절약

<추후 개선점이나 보완할 점>

- 학습 시간이 오래 걸려서, 모델 학습 과정을 더 빠르게 만드는 방법이 필요
- 다양한 모델을 실험해보기엔 시간이 부족했어서, 다음엔 여유 있는 일정이 필요
- 성능 튜닝을 충분히 못 해서, 다음엔 튜닝에 더 집중할 수 있는 시간 확보가 필요

<아쉬운 점>

- 기간이 짧아 튜닝에 시간을 많이 쓰지못함
- 데이터 용량이 커 모델링에 시간이 많이 소요되어 더 다양한 모델링을 시도해보지 못함
- iM뱅크 데이터와 데이콘 데이터가 많이 달라 제대로 비교가 안되서 아쉬움

<느낀 점이나 경험한 성과(경력 계획 등과 연관)>

- 다양한 샘플링 방법을 직접 적용해보며 **불균형 데이터 처리에 대한 이해도가 높아짐**
- 시간이 부족한 상황에서도 레퍼런스를 참고해 효율적으로 프로젝트를 진행한 점이 뿌듯했음
- 변수 선택부터 모델링까지 전체 과정을 경험하면서 모델 개발 흐름을 체계적으로 익힐 수 있었음

출처 및 참고문헌

Si, Si., Zhang, H., Keerthi, S. S., Mahajan, D., Dhillon, I. S., & Hsieh, C.-J. (2017). *Gradient boosted decision trees for high dimensional sparse output*. In Proceedings of the 34th International Conference on Machine Learning.

Benkendorf, D. J., Schwartz, S. D., Cutler, D. R., & Hawkins, C. P. (2023). Correcting for the effects of class imbalance improves the performance of machine-learning based species distribution models.

Warmbrod, J. R. (2001, December 11). *Conducting, interpreting, and reporting quantitative research*. National Agricultural Education Research Conference, New Orleans, LA.

Teodorescu, M. H. (2018). *Machine learning methods for strategy research* (HBS Working Paper No. 18-011). Harvard Business School.

Handling Method of Imbalance Data for Machine Learning

The Precise Effect of Multicollinearity on Classification Prediction

Boosting methods for multi-class imbalanced data classification

Q&A

3조: 손효은 염기진 이우태 최다은