



금융보안원  
FINANCIAL SECURITY INSTITUTE

DACON

# 생성형 AI를 활용한 이상금융거래 탐지

## FSI AIXData Challenge 2024

팀명 : GNOEYHEAT

팀원 : 김태형, 설근형, 서지원, 조주혜

# CONTENTS

---



## 01. 프로젝트 개요

- 1) 프로젝트 배경 및 주제
- 2) 데이터 명세 및 생성조건

## 02. 방법론

- 1) 합성 데이터 생성
- 2) 주요 속성 익명화
- 3) 분류모델 최적화

## 03. 실험 결과

## 04. 결론

## 프로젝트 배경 및 주제

- AI는 금융 산업에서 데이터 분석을 통한 서비스 개선, 이상금융거래 탐지 및 예방 등 다양한 방면으로 혁신을 이끌고 있음.
- 금융 데이터의 클래스 불균형 문제는 이상거래탐지의 대표적인 문제이며, 이를 해결하기 위한 방법으로 합성 데이터가 주목받고 있음.
- 합성 데이터는 이상 거래와 같은 소수 클래스의 비율을 조정하거나 새로운 학습 특성을 반영하여 클래스 불균형 문제를 개선할 수 있음.
- 또한, 생성형 AI를 활용한 고품질의 합성 데이터는 개인정보 보호 등의 사유로 실제 데이터의 활용이 어려운 경우에 활용 가치가 매우 높음.
- 합성 데이터 생성 및 활용 시 개인정보 보호와 보안을 고려하는 유용성과 안정성이 검증된 프레임워크가 필요함.

## 프로젝트 목표

- 본 프로젝트에서는 생성형 AI를 활용하여 보안을 강화한 고품질의 합성 데이터를 생성하고자 함.
- 합성 데이터를 분류 AI모델에 효과적으로 학습하여 클래스 불균형 문제를 해결하는 방법론을 제안함.

# 01. 프로젝트 개요 (데이터 명세 및 생성조건)

Identification & Personal Info (식별 및 개인 정보)	Demographics (인구 통계 정보)	Account Information (계좌 정보)
ID (샘플 식별자 번호)	Customer_Gender (고객 성별)	Account_account_number (계좌번호)
Customer_Birthyear (고객 출생년도)		Account_account_type (계좌 유형)
Customer_personal_identifier (고객명)		Account_initial_balance (거래 전 잔액)
Customer_identification_number (주민번호)		Account_balance (거래 후 잔액)
Customer_credit_rating (고객 등급)		Account_amount_daily_limit (1일 거래 한도)
Transaction Information (거래 정보)	Error Information (오류 정보)	Account remaining amount daily limit exceeded (1일 거래 한도 잔여액)
Transaction_Datetime (거래일자)	Error_Code (오류 코드)	
Transaction_Amount (이체 금액)	Transaction_Failure_Status (거래 성공 여부)	
Transaction_resumed_date (거래 재개 일자)		
Last_atm_transaction_datetime (최근 ATM 거래 일시)	Location & Device Info (위치 및 기기 정보)	Authentication & Security (인증 및 보안 정보)
Last_bank_branch_transaction_datetime (최근 은행 지점 거래 일시)	IP_Address (거래에 사용한 단말기 IP 주소)	Customer_flag_change_of_authentication (인증 여부)
	MAC_Address (거래에 사용한 단말기 MAC 주소)	Customer_rooting_jailbreak_indicator (탈옥 및 루팅 여부)
	Location (거래 발생 위치)	Customer_mobile_roaming_indicator (모바일 로밍 여부)
		Customer_VPN_Indicator (VPN 사용 여부)

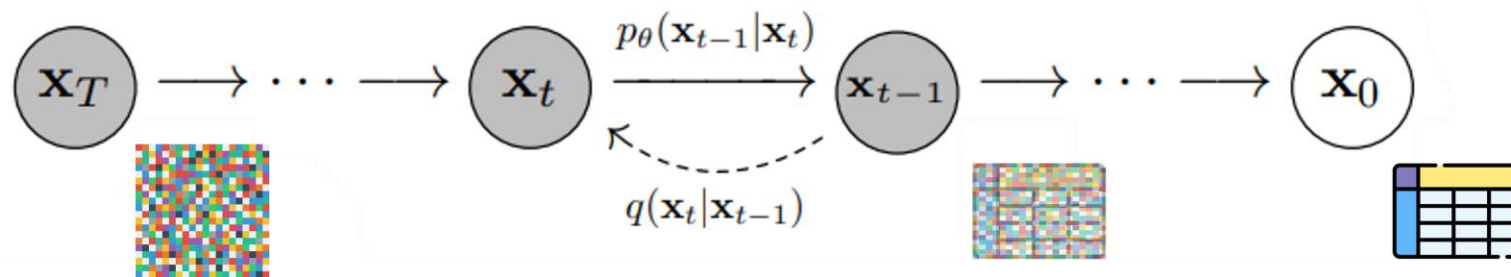
- ✓ **예측 목표 변수** : Fraud Type(사기 시나리오) - 총 13개의 범주 (정상 시나리오 1개, 이상 시나리오 12개, 1188:1로 매우 불균형 함.)
- ✓ **보안 조치(Masking) 된 변수** : Customer\_personal\_identifier (고객명), Customer\_identification\_number (주민번호), Account\_account\_number (계좌번호), IP\_Address (거래에 사용한 IP주소), MAC\_Address (거래에 사용한 MAC 주소), Location (거래 위치), Recipient\_Account\_Number (수취인 계좌번호)
- ✓ **학습에 제외된 변수** : ID (샘플 식별자 번호), Customer\_personal\_identifier (고객명), Customer\_identification\_number (주민번호), Account\_account\_number (계좌번호), Recipient\_Account\_Number (수취인 계좌번호), Account\_initial\_balance (거래 전 잔액), Account\_balance (거래 후 잔액), Account\_amount\_daily\_limit (1일 거래 한도), Account\_remaining\_amount\_daily\_limit\_exceeded (1일 거래 한도 잔여액), IP\_Address (거래에 사용한 단말기 IP 주소), MAC\_Address (거래에 사용한 단말기 MAC 주소), Location (거래 발생 위치), Customer\_registration\_datetime (고객 등록 일자), Account\_creation\_datetime (계좌 개설 일자), Transaction\_Datetime (거래일자), Last\_atm\_transaction\_datetime (마지막 ATM 거래 일자), Last\_bank\_branch\_transaction\_datetime (마지막 영업점 거래 일자), Transaction\_resumed\_date (거래 재개 일자)

### 합성 데이터 생성의 필요성

- 합성 데이터 생성이란 생성형 AI의 한 분야로서, 실제 데이터와 통계적 특성이 유사하도록 가상의 데이터를 생성해내는 과정을 의미함.
- 금융 도메인과 같이 이상 거래 데이터의 양이 부족하거나 클래스 불균형 문제가 존재하는 상황에서 유용하게 활용될 수 있음.

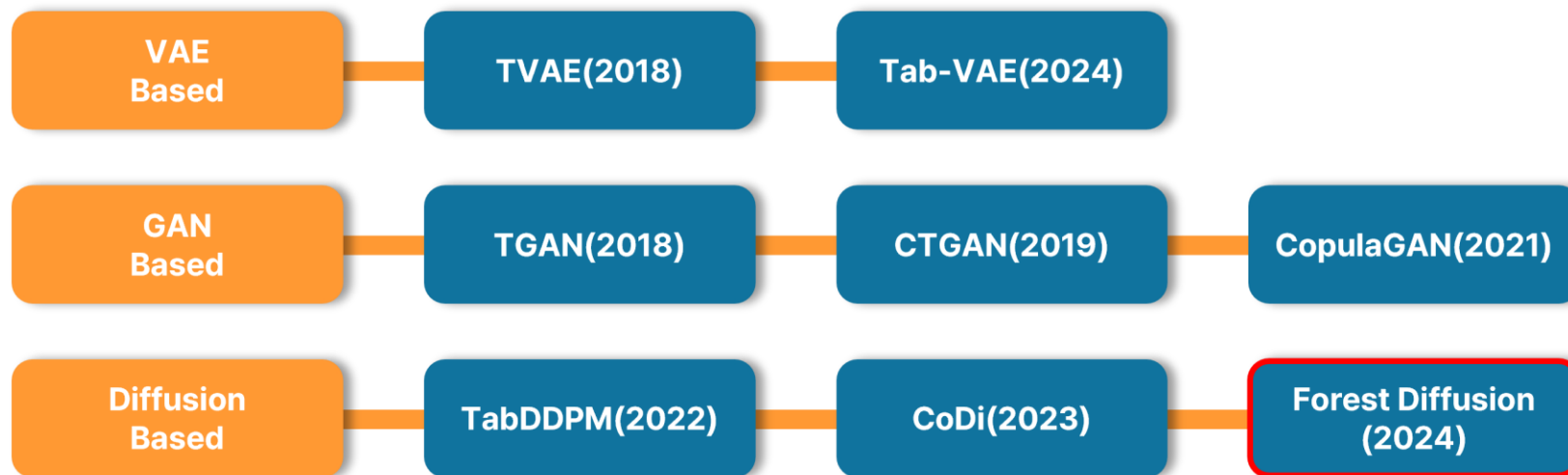
### 합성 데이터 생성의 목표

- 실제 데이터의 통계적 특성을 유지하면서 다양성을 반영한 데이터를 생성하여 분류 모델 학습 시 일반화 능력을 향상시키고자 함.
- 고품질의 합성 데이터 생성을 위한 방법론으로 확산 기반의 생성 모델(Forest Diffusion)을 활용함.



### Forest Diffusion

- 확산 모델은 원본 데이터를 노이즈로 변환한 뒤, 다시 복원하는 과정을 통해 분포를 학습하고 새로운 데이터를 생성함.
- 확산 모델은 GAN, VAE 등 기존의 방법론에 비해 더 높은 데이터 품질과 안정적인 학습 과정을 제공하는 장점을 가짐.
- 본 프로젝트에서는 확산 기반의 표 데이터 생성 State-of-the-Art 모델인 “Forest Diffusion”을 활용함.



### Forest Diffusion 모델의 장점

- 안정적인 학습
  - 확산 모델은 점진적으로 데이터를 복원하는 방식으로 학습이 이루어지기 때문에 모드 붕괴<sup>1)</sup> 문제에 강건함.
- 고품질 합성 데이터
  - 원본 데이터와 유사한 통계적 특성을 유지하면서도 다양성이 보장됨.
- 표 형태 데이터 생성에 특화
  - 신경망이 아닌 XGBoost를 사용하여 표 형태 데이터 생성에 특화됨.
- 낮은 계산 복잡성
  - 확산 모델에 조건부 흐름 매칭<sup>2)</sup>을 결합하여 데이터 복원 경로를 결정론적으로 정의하여 복잡성을 줄이는데 기여함.

“위와 같은 장점으로 합성 데이터를 생성하는 VAE, GAN 등 방법론에 비해 제안하는 Diffusion 기반의 방법론은 활용 가능성이 높음.”

1) 생성자와 판별자가 경쟁하는 GAN의 학습에서 생성자가 판별자를 속이는 과정에만 집중하게 되는 현상

2) 특정 조건을 통해 데이터 복원을 제어하여 표 데이터의 복잡한 데이터 구조와 특정 변수의 형태를 유지한 채로 변환을 수행

### 주요 속성 익명화의 필요성

- 주요 속성(변수)의 익명화는 보안 위협 상황으로부터 개인정보를 보호하는 중요한 역할을 함.
- 금융 데이터는 고객의 민감한 개인 정보를 포함하기 때문에 주요 속성 익명화는 필수적인 보안 조치임.

### 주요 속성 익명화의 목표

- 익명화 된 합성 데이터를 통해 원본 데이터의 주요 속성 값을 유추할 가능성을 차단하고자 함.
- 생성모델 익명성을 평가하기 위한 지표로  $TCAP$ 를 사용함.
- $TCAP$ 는 원본 데이터의 주요 속성 값을 알고 있을 때, 합성 데이터를 통해 원본 데이터 내 개인의 민감한 정보를 유추해낼 위험성을 평가함.

$$TCAP_j = \frac{\sum_{i=1}^n I(T_{obs,i} = T_{syn,j}, K_{obs,i} = K_{syn,j})}{\sum_{i=1}^n K_{obs,i} = K_{syn,j}}$$

$I(A)$  : 조건 A를 만족할 경우 1, 만족하지 않을 경우 0 값을 갖는 지시 함수

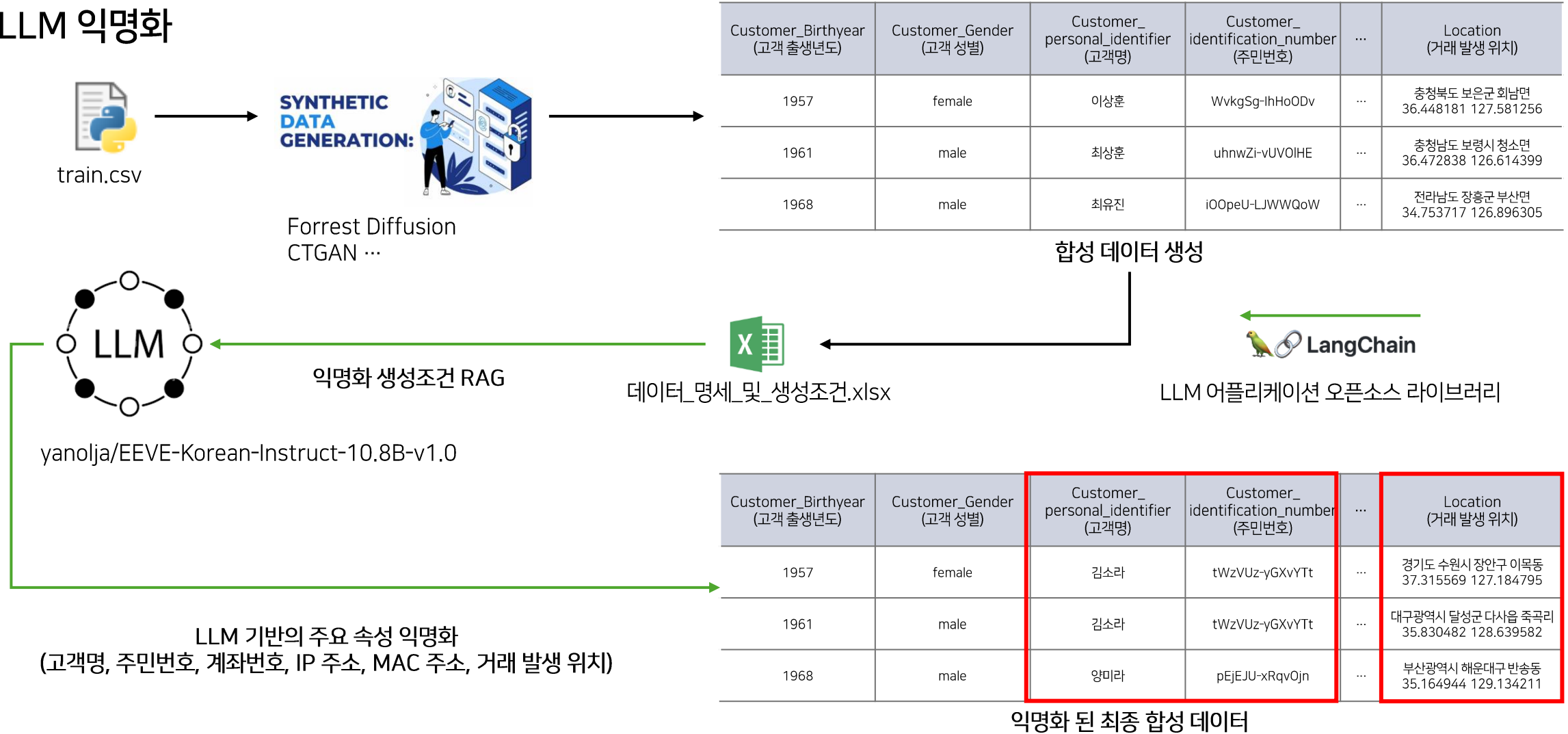
$K_{obs,i} / K_{syn,i}$  : 원 데이터 / 합성데이터의  $i$ 번째 레코드의 주요 속성 값

$T_{obs,i} / T_{syn,i}$  : 원 데이터 / 합성데이터의  $i$ 번째 레코드의 목표 속성 값



## 02. 방법론 (주요 속성 식명화)

### LLM 식명화



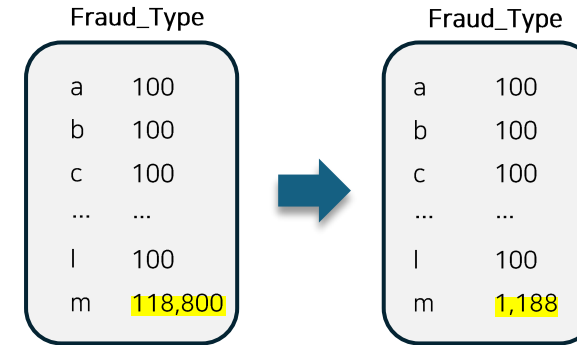
### LLM 익명화 기법의 장점

- 문맥을 고려한 자연스러운 익명화
  - LLM은 데이터의 문맥을 이해하고 속성의 생성조건을 반영하여 익명화 된 새로운 값을 생성함.
  - 고객명, 주민번호 등 민감한 개인 정보를 유사한 맥락의 새로운 값으로 대체하여 익명성을 강화함.
  - 기존의 대체 기반의 익명화 방식 대비 합성 데이터의 유용성이 유지됨.
- 다양한 생성조건에 유연한 적용
  - LLM은 자연어 지시를 통해 생성조건을 전달받아 새로운 속성(텍스트)을 생성할 수 있음.
  - LangChain 라이브러리를 활용하여 데이터 명세서 내 생성 조건을 쉽게 적용할 수 있음.
  - 이는, 복잡한 생성조건을 쉽게 추가할 수 있기 때문에 자유도와 확장성이 높은 장점을 가지며 지속(개선) 가능성이 우수함.

“위와 같은 장점으로 제안하는 LLM 익명화 기법은 생성 조건이 추가되어도 쉽게 반영할 수 있기 때문에 지속(개선) 가능성이 우수함.”

### Under Sampling

- 클래스 간의 불균형을 해결하기 위해 under sampling 기법을 적용함.
- 다수 클래스인 'm'에 해당하는 데이터의 1%를 샘플링 함.
















### Feature Engineering

- 날짜 및 시간 변수를 datetime 객체로 변환 후 시간 차이를 나타내는 파생 변수를 생성함.
- 예를 들어, Time Difference 변수는 직전 거래와의 시간 차이를 나타내며 추가적인 파생 변수는 다음과 같음.

파생 변수	사용된 조건 변수	의미
Age_Group	Customer_Birthyear	나이 그룹
Total_change_of_authentication	Customer_flag_change_of_authentication_1 ~ 4	인증 변경 횟수
rooting_jailbreak_Transaction_Amount	Customer_rooting_jailbreak_indicator, Transaction_Amount	루팅/탈옥 거래 금액
mobile_roaming_Transaction_Amount	Customer_mobile_roaming_indicator, Transaction_Amount	로밍 거래 금액
Balance_change	Account_balance, Account_initial_balance	잔액 변동
remaining_daily_limit	Account_amount_daily_limit, Account_remaining_amount_daily_limit_exceeded	일일 한도 사용 금액
Transaction_Amount_to_Daily_Limit_Ratio	Transaction_Amount, Account_amount_daily_limit	일일 한도 대비 거래 비율
Transaction_UCL	Transaction_Amount, Account_one_month_std_dev	거래 금액 상한선
Over_UCL	Account_one_month_max_amount, Transaction_UCL	상한선 초과 여부
creation_to_registration_timedelta	Account_creation_datetime, Customer_registration_datetime	계좌 생성 - 고객 등록 시간 차
transaction_to_creation_timedelta	Transaction_Datetime, Account_creation_datetime	계좌 생성 - 거래 발생 시간 차
last_atm_timedelta	Transaction_Datetime, Last_atm_transaction_datetime	마지막 ATM 거래 후 경과 시간
last_bank_branch_timedelta	Transaction_Datetime, Last_bank_branch_transaction_datetime	마지막 지점 거래 후 경과 시간
resumed_date_timedelta	Transaction_Datetime, Transaction_resumed_date	거래 재개 후 경과 시간

## Feature Engineering

- 범주형 변수를 수치형 변수로 변환하기 위해 총 3가지의 encoding 방법을 모두 적용함.

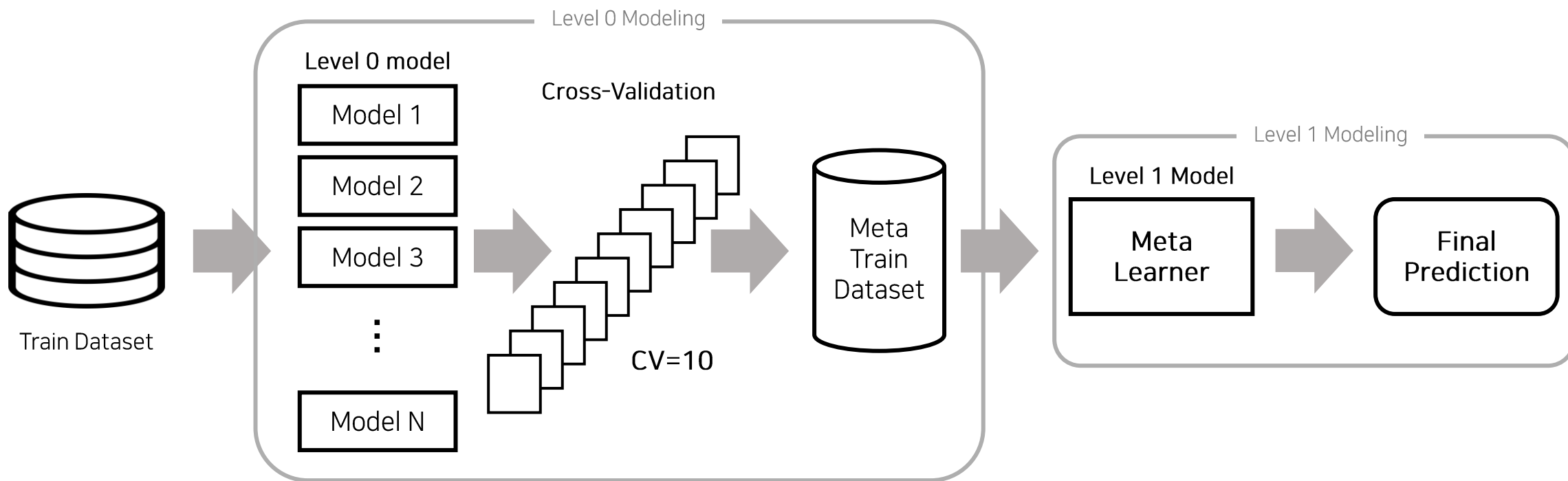
방법	Label Encoding	Target Encoding	One-hot Encoding	
설명	범주형 데이터를 숫자로 변환	범주형 데이터를 범주의 타겟 평균 값과 전체 데이터의 타겟 평균을 혼합하여 계산된 최종 값으로 변환	범주형 데이터를 이진 벡터로 변환	
예시	<div><div>Gender</div><div>   </div><div>Gender</div><div>1 0 1 1</div></div> <div></div>	<div><div>Gender</div><div>   </div><div>Target</div><div>1 1 0 1</div></div> <div>→</div> <div><div>Gender_te</div><div>0.67 1 0.67 0.67</div></div>	<div><div>Gender</div><div>   </div><div>Gender_Female</div><div>0 1 0 0</div></div> <div>→</div> <div><div>Gender_Male</div><div>1 0 1 1</div></div>	
사용변수	Fraud_Type, Customer_Gender	Customer_credit_rating, Customer_loan_type, Account_account_type, Channel, Operating_System, Error_Code, Type_General_Automatic, Access_Medium		

- 분류 AI 모델 학습 시 불필요하거나 상관관계가 낮은 변수를 학습에서 제외함.

ID, Customer\_personal\_identifier, Customer\_identification\_number, Account\_account\_number, Account\_initial\_balance, Account\_balance, Account\_amount\_daily\_limit, Account\_remaining\_amount\_daily\_limit\_exceeded, IP\_Address, MAC\_Address, Location, Recipient\_Account\_Number, Another\_Person\_Account, Customer\_registration\_datetime, Account\_creation\_datetime, Transaction\_Datetime, Last\_atm\_transaction\_datetime, Last\_bank\_branch\_transaction\_datetime, Transaction\_resumed\_date

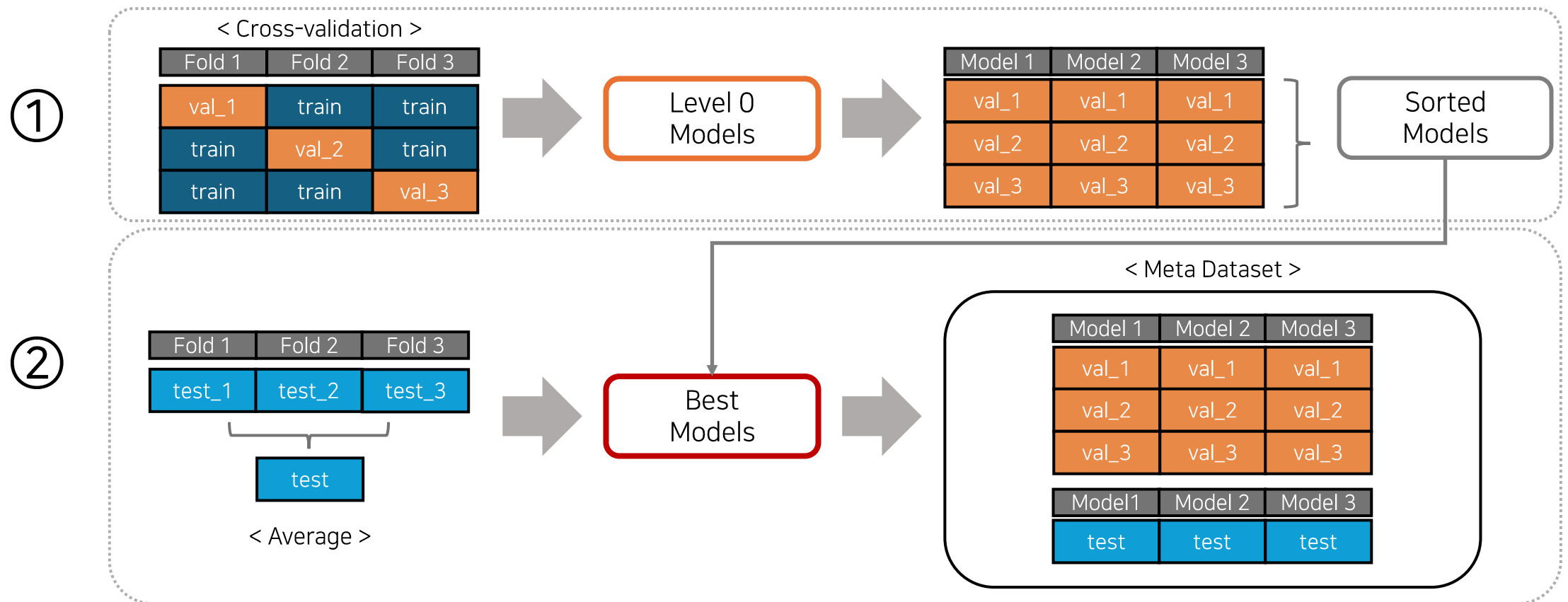
### Cross Validation Stacked Generation (Stacking Ensemble)

- Stacking은 다양한 예측 모형의 출력 값을 최종 예측 모형의 입력 값으로 사용하는 앙상블 기법임.
- Level 0 Modeling을 통해 선택된 모형의 예측 값으로 Meta Train Dataset을 구축함.
- 그 후, Level 1 Modeling을 통해 Meta Learner을 학습하여 최종 예측을 진행함.



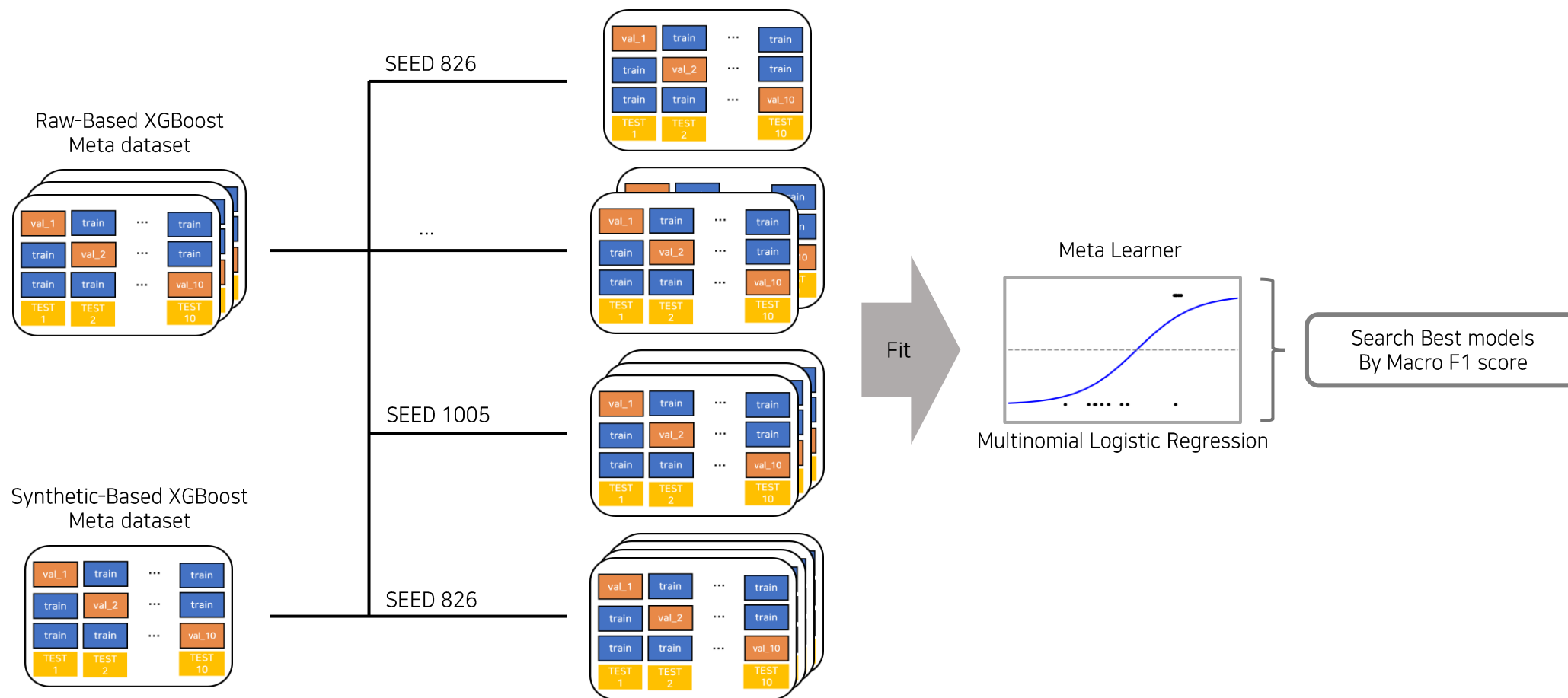
### Cross Validation Stacked Generation (Level 0)

- Fold 마다 Dataset에 대해 예측을 진행하고, train dataset 예측 값을 누적하고 test dataset 예측 값을 평균하여 Level 1 Modeling에 전달하는 Meta Dataset을 구성함.



### Cross Validation Stacked Generation (Level 1)

- 원본 데이터 기반의 예측 값과 합성 데이터 기반의 예측 값을 메타모델의 입력 값으로 모두 활용하여 합성 데이터의 영향력을 반영함.



### Bayesian Optimization

- 분류 AI 모델을 최적화하기 위해 Bayesian Optimization(베이지안 최적화) 기법을 사용함.
- Bayesian Optimization은 이전에 탐색된 하이퍼 파라미터의 결과를 바탕으로 다음에 탐색할 후보를 효과적으로 결정하는 장점을 가짐.
- 파이썬 오픈소스 라이브러리 Optuna를 사용함.

#### XGBoost Hyperparameters

n\_estimators(트리의 개수): 1976  
max\_depth(개별 트리의 최대 깊이): 6  
learning\_rate(학습률, 모델이 학습할 때 가중치 조정의 크기: 0.008294125648045027  
gamma(트리 분할 시 필요한 최소 손실 감소):0.11044398100317245  
min\_child\_weight(리프 노드에서 최소 가중치 합): 1  
subsample(각 트리에서 사용할 샘플의 비율): 0.9  
sampling\_method(샘플링 방식, gradient 기반): gradient\_based  
colsample\_bytree(트리를 구성할 때 사용할 피쳐의 비율): 0.8  
reg\_alpha(L1 규제, 가중치 절대값의 합을 최소화하는 패널티): 0.03296137174022581  
reg\_lambda(L2 규제, 가중치 제곱합을 최소화하는 패널티): 0.006095201538414734  
tree\_method(트리 구축 방법, GPU 가속을 사용하는 히스토그램 기반 알고리즘): gpu\_hist  
n\_jobs(병렬 처리에 사용할 CPU 코어 수): -1(가능한 모든 코어 사용)  
random\_state(난수 생성기의 시드 값, 모델의 재현성을 위한 설정): seed  
eval\_metric(모델 평가에 사용되는 메트릭): macro\_f1\_score



### 이상금융거래 탐지 성능 평가

- Score = 0.7 x Macro F1 Score + 0.3 x (1 - TCAP)

Model	Public Score	Private Score
Ours (Raw data + Syn data)	0.81623	0.81802
Ours (Raw data)	0.81403	0.81577

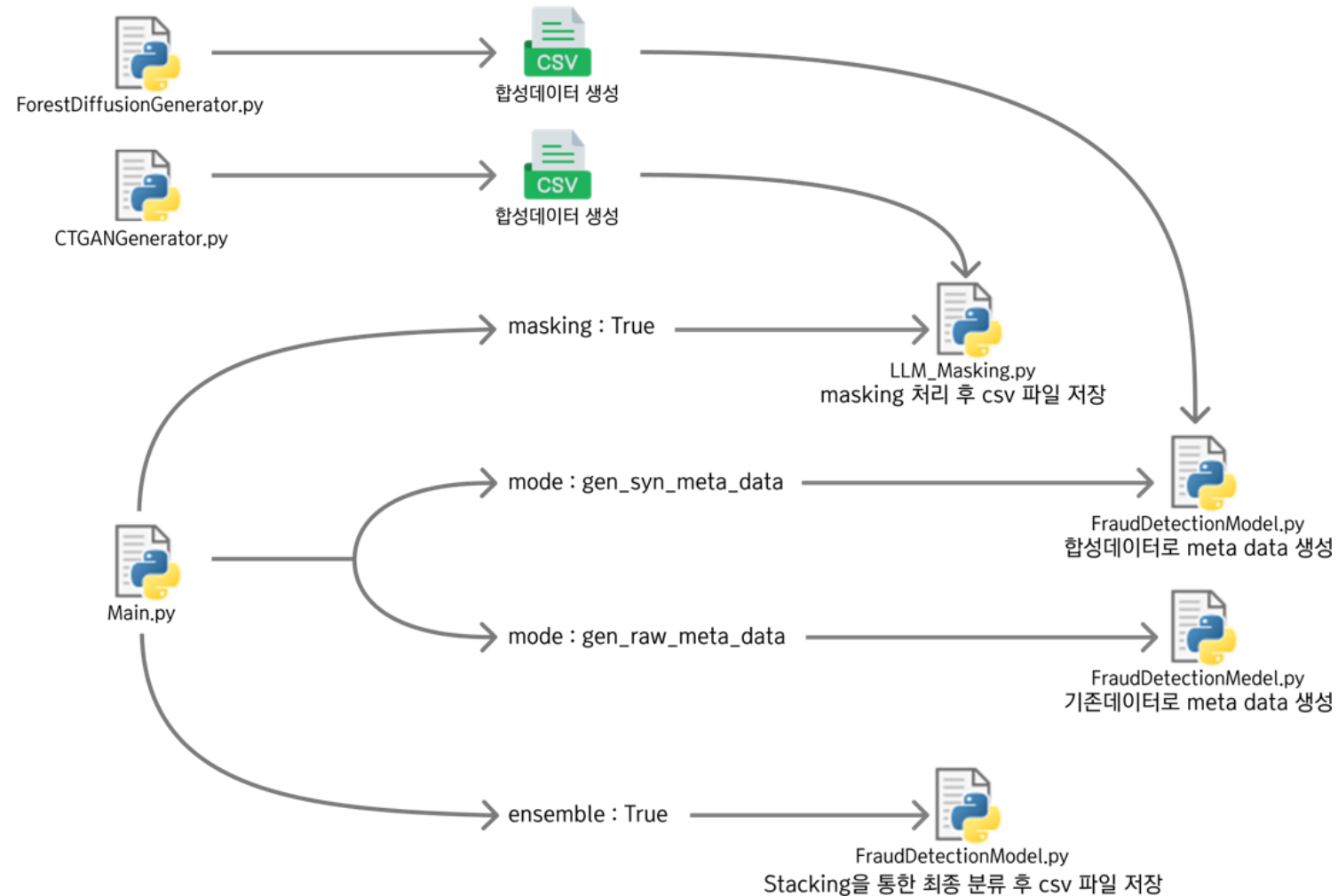
### Ablation Study – Under sampling

- Under sampling의 비율에 대한 5-Fold 검증 성능을 비교함.

Model	Val Macro F1
XGBoost-Optuna (Frac=0.01)	0.70742
XGBoost (Frac=0.01)	0.68532
XGBoost (Frac=0.1)	0.63969
XGBoost (Frac=1)	0.57486

- 본 프로젝트에서는 생성형 AI를 활용한 이상금융거래 탐지 프레임워크를 개발함.
- Forest Diffusion 기반의 합성 데이터는 CTGAN 등 다른 생성형 AI 모델에 비해 분류 모델 학습에서 사용 시 우수한 성능을 보임.
- LLM을 활용한 주요 속성 식명화 기법은 변화되는 복잡한 식명화 조건을 지속적으로 반영할 수 있으며 실용적임.
- 이상 거래 탐지의 성능을 높이기 위한 데이터 전처리 방법으로 Under Sampling과 Feature Engineering을 진행함.
- 분류 모델의 성능을 극대화하기 위해 최적화된 Stacked Generation 모델을 제안함.
- 제안하는 방법론을 적용한 이상금융거래 탐지 모델은 클래스 불균형 문제를 해결하고 우수한 성능을 도출함.
- 합성 데이터를 단순히 소수 범주의 데이터 수를 늘리는 것뿐만이 아닌 메타 데이터의 입력 변수로 활용하는 접근을 통해 분류 AI 모델 학습에 효과적으로 반영할 수 있음.
- LLM 식명화 기법은 금융 데이터 내 통계적 정보를 조건으로 합성 데이터를 직접 생성하는 방식으로 발전 가능함.

## 전체 파이프라인



**Thank You**