KUBIG Conference

- 연합학습을 통한 사기 탐지 시스템

KUBIG
DATA SCIENCE & AI

Team | 사기꾼연합단체

Members | 강지윤, 김채원, 이유진, 김수환

KUBIG
DATA SCIENCE & AI

CONTENTS

KUBIG
DATA SCIENCE & AI

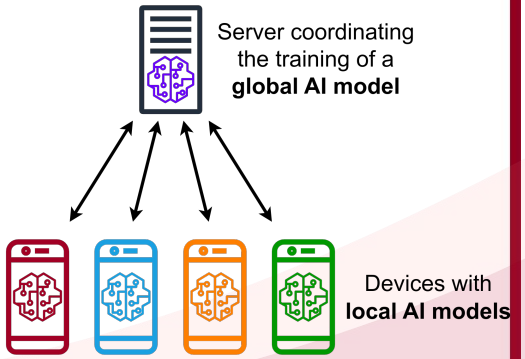
01. Introduction

01. Federated Learning

- **“Federated Learning”**
  - 여러 로컬 기기나 기관에 분산된 데이터를 직접 공유하지 않고, 각 기기에서 학습된 모델의 가중치를 중앙 서버로 모아 글로벌 모델을 학습시키는 분산형 머신러닝 시스템

- **학습 Flow**

1. 각 로컬 기기는 자체 데이터로 모델 학습
2. 학습 파라미터만 중앙 서버로 전송
3. 서버에서 모든 업데이트를 통합하여 글로벌 모델 생성
4. 업데이트된 글로벌 모델을 각 기기로 전달



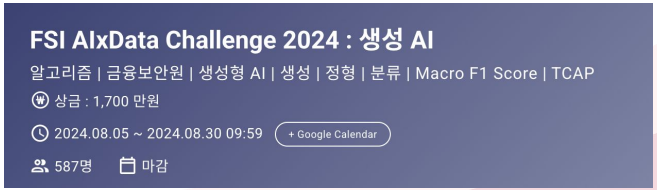
01. 주제 선정 배경

🏆 U.S. PETs Prize Challenge 대회

- 미국 NIST/NSF와 영국이 주최한 Privacy-Enhancing Technologies (PETs) 경진대회

수상팀들의 논문, 코드, 구조를 분석해본 후,

이를 *실제 금융 시나리오*에 적용해도 성능이 유지될까?



KUBIG
DATA SCIENCE & AI

02. Fraud Detection

02. Dataset & Preprocessing

- **FSI AIxData Challenge 2024**에서 공개한 dataset 활용
- 데이터 불균형이 매우 심하고, 고차원 구조
- **Target : 다중분류(사기유형 a~m) → 이진분류(fraud or not)**
- 송금인, 수취인 변수 기반 파생변수 생성 but 이후 전부 drop

▶ 발견 1 : **Only 송금인 → (102건) 100명 모두 사기(binary_label=1) (모두 binary_ratio=1.0)⇒ Only 송금인=사기용 생성된 계좌 (+: Fraud_Type은 e나 f) (e 단독: 50명, e,f: 2명, f 단독: 48명)** (e,f: 'VVVCSgSZhQ', 'hstjhgPrzV') (e,f type의 transaction count가 낮았던 이유)

▶ 발견 1.5: 송금인&수취인 동시 → 송금인 기준: 1098건 사기(중복포함), 수취인 기준: 640건 (중복포함), 송금인이 1회만 거래(1건), 'oRgEFlFEsl'가 송금인일 때도 사기, 수취인일 때도 사기

▶ 발견 2: **Only 수취인** → 사기 560건, **단건 사기용 수취인 계좌 514건**, 여러거래 사기용 수취인 계좌 46건(중복포함)

37	Account_dawn_one_month_std_dev	use
38	Transaction_Datetime	파생변수: (1.요일, 2.월, 3.time_ratio, 4.month_ratio)
39	Trnsaction_Amount	use
40	Channel	원핫인코딩
41	Operating_System	원핫인코딩
42	Error_Code	원핫인코딩
43	Transaction_Failure_Status	use
44	Type_General_Automatic	원핫인코딩
45	IP_Address	drop
46	MAC_Address	drop
47	Access_Medium	원핫인코딩
48	Location	1. 원핫인코딩(지역명) 2. 파생변수: 위도/경도 숫자 분리
49	Recipient_Account_Number	drop
50	Transaction_num_connection_failure	use

02. Modeling

- 이진 분류를 위해

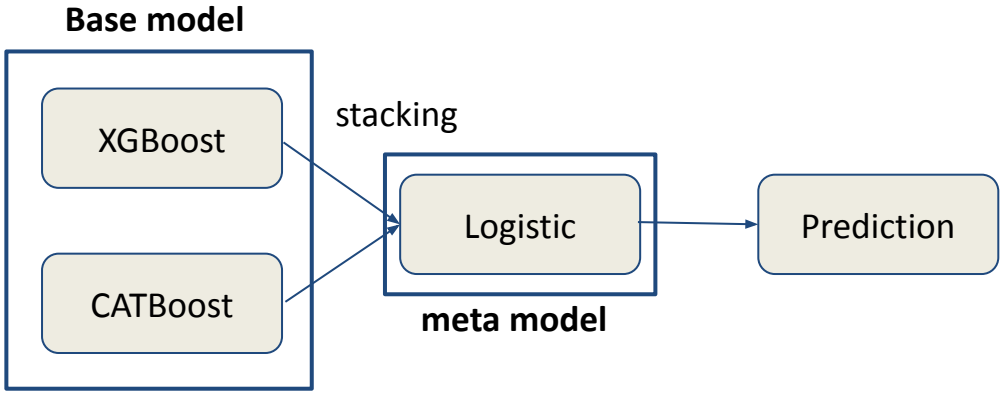
  Catboost, Xgboost, One-class SVM, AutoEncoder 튜닝

- **단일 모델별 최고 성능**

	Catboost	XGboost	One-Class SVM	AutoEncoder
(Binary) F1	0.82	0.81	0.54	0.53

02. Results

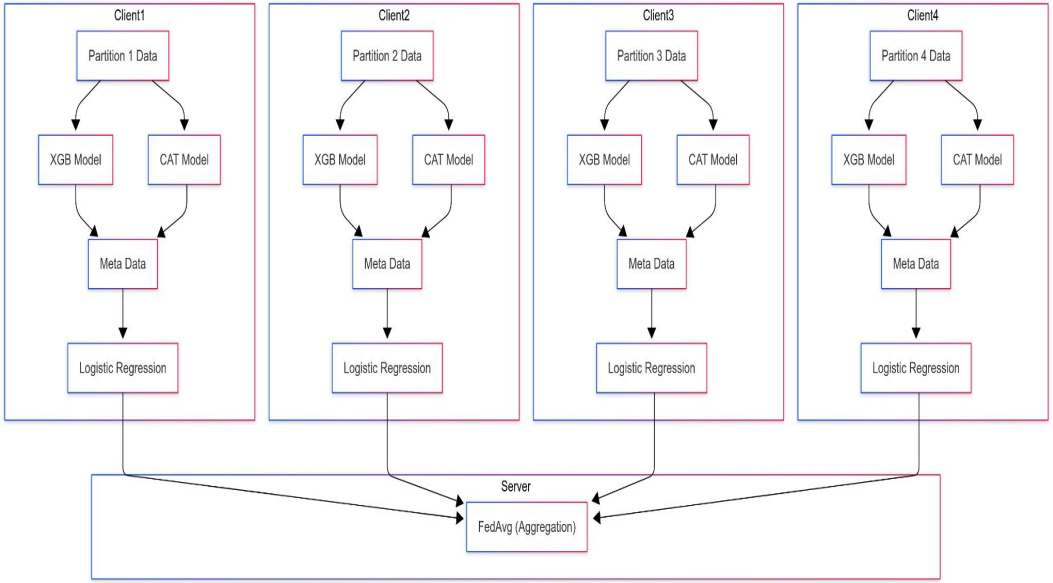
- 예측 성능을 높이고자, Xgb + catboost stacking, soft-voting 시도
- Soft voting : 0.82 / Stacking (meta model : Logistic) : 0.83
- **→ Centralized learning에서의 최종 성능 : 0.83 (binary f1 기준)**



KUBIG
DATA SCIENCE & AI

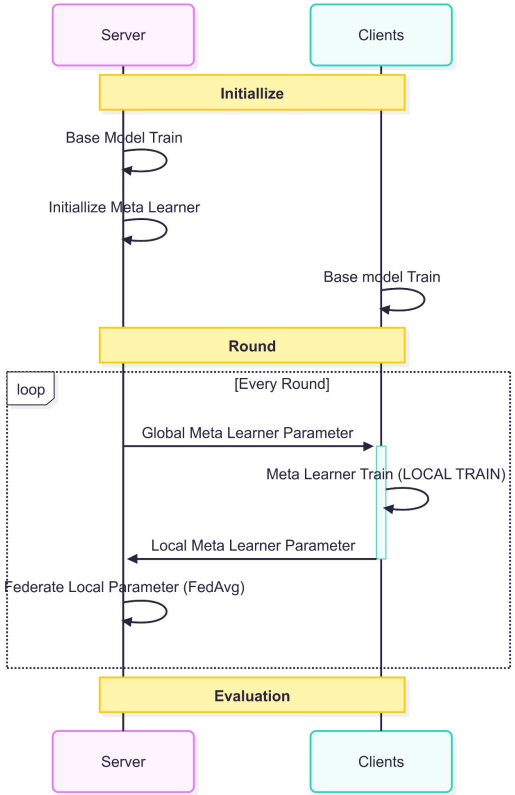
03. Federated Learning

03. Implantation



- 기존 **중앙 집중형 Stacking 구조**를 연합 학습 환경에 맞게 **분산형 구조로 구현**
- 각 클라이언트는 **Base 모델 학습 및 메타데이터 생성**
- 서버는 **Logistic Regression 기반 메타 러너**를 연합 방식으로 **통합 학습**

03. Federate Learning Flow



초기화 단계

- 서버와 클라이언트가 각각 Stacking Classifier의 **Base 모델을 학습**
- 서버는 **메타 러너(Logistic Regression)**의 초기 설정 수행

학습 단계

- 각 라운드마다 서버가 **글로벌 메타 러너 파라미터를 배포**
- 클라이언트는 이를 바탕으로 **메타 러너 로컬 학습**
- 업데이트된 파라미터를 서버로 전송 → **서버는 이를 통합(FedAvg)**

평가 단계

- 통합된 메타 러너를 기반으로 **모델 성능 평가 수행**

03. Try 1 : Logistic Regression

1️⃣시도 :

- 첫 시도로 **Logistic Regression**을 메타 러너로 사용
- 그러나 **라운드마다 결과 변화 없음**

```
INFO :      [ROUND 1]
INFO :      configure_fit: strategy sampled 3 clients (out of 3)
INFO :      aggregate_fit: received 3 results and 0 failures
서버 : 클라이언트로 부터 받은 매개변수의 1번째 AVG값 :
  coef : [[-0.66597141  0.00609968 -0.66498208  0.00511036]],
  intercept : [-0.65987173]

서버 라운드 1 중앙 집중식 평가 :
  정확도 : 0.0100
  클래스 0 정밀도 : 0.0000, 재현율 : 0.0000, F1: 0.0000
  클래스 1 정밀도 : 0.0100, 재현율 : 1.0000, F1: 0.0197
  (가중 평균) 정밀도 : 0.0001, 재현율 : 0.0100, F1: 0.0002

INFO :      [ROUND 5]
INFO :      configure_fit: strategy sampled 3 clients (out of 3)
INFO :      aggregate_fit: received 3 results and 0 failures
서버 : 클라이언트로 부터 받은 매개변수의 5번째 AVG값 :
  coef : [[-0.66597141  0.00609968 -0.66498208  0.00511036]],
  intercept : [-0.65987173]

서버 라운드 5 중앙 집중식 평가 :
  정확도 : 0.0100
  클래스 0 정밀도 : 0.0000, 재현율 : 0.0000, F1: 0.0000
  클래스 1 정밀도 : 0.0100, 재현율 : 1.0000, F1: 0.0197
  (가중 평균) 정밀도 : 0.0001, 재현율 : 0.0100, F1: 0.0002
```

🧠 원인 분석:

- 로컬 학습 시 fit() 함수가 **최적 파라미터로 완전히 수렴**
- 클라이언트가 자체 데이터에 대해 **과최적화 (Overfitting)**
- 결과적으로 **Federated Learning이 라운드별로 진행되지 않음**

03. Try 2 : Stochastic Gradient Descent

2시도 :

- **SGD(Stochastic Gradient Descent)**를 메타 러너로 사용
- partial_fit()을 통해 **점진적 학습이 가능**
- loss='log_loss' 설정 시, **Logistic Regression과 유사한 동작**

🧠 Federated Learning에 적합한 이유:

- 각 라운드마다 **소폭의 파라미터 업데이트 유지**
- **과적합 없이 반복적인 연합 학습 구조** 구현 가능
- **클라이언트 간 파라미터 통합이 자연스러움 (FedAvg 호환)**

```
INFO :      [ROUND 1]
INFO :      configure_fit: strategy sampled 4 clients (out of 4)
INFO :      aggregate_fit: received 4 results and 0 failures
WARNING :   No fit_metrics_aggregation_fn provided
서버 : 서버 평가 시작 시 X_combined_test 형태 : (24002, 124)
서버 : 서버 평가를 위해 생성된 메타 특성 형태 : (24002, 4)
서버 : 클라이언트 매개변수 설정 후 모델 coef_ 형태 : (1, 4)
서버 : 클라이언트로 부터 받은 매개변수의 1번째 AVG값 :
coef : [[-3.4393921   2.65397963 -3.17785403  2.36523188]],
intercept : [-1.05724504]

서버 라운드 1 중앙 집중식 평가 :
  정확도 : 0.9740
  클래스 0 정밀도 : 0.9933, 재현율 : 0.9803, F1: 0.9868
  클래스 1 정밀도 : 0.1509, 재현율 : 0.3473, F1: 0.2104
  (가중 평균) 정밀도 : 0.9850, 재현율 : 0.9740, F1: 0.9791
```

```
INFO :      [ROUND 5]
INFO :      configure_fit: strategy sampled 4 clients (out of 4)
INFO :      aggregate_fit: received 4 results and 0 failures
서버 : 서버 평가 시작 시 X_combined_test 형태 : (24002, 124)
서버 : 서버 평가를 위해 생성된 메타 특성 형태 : (24002, 4)
서버 : 클라이언트 매개변수 설정 후 모델 coef_ 형태 : (1, 4)
서버 : 클라이언트로 부터 받은 매개변수의 5번째 AVG값 :
coef : [[-3.51054691  3.03971853 -3.10246514  2.62125423]],
intercept : [-1.32386557]

서버 라운드 5 중앙 집중식 평가 :
  정확도 : 0.9737
  클래스 0 정밀도 : 0.9934, 재현율 : 0.9799, F1: 0.9866
  클래스 1 정밀도 : 0.1497, 재현율 : 0.3515, F1: 0.2100
  (가중 평균) 정밀도 : 0.9850, 재현율 : 0.9737, F1: 0.9789
```

KUBIG
DATA SCIENCE & AI

04. Conclusion

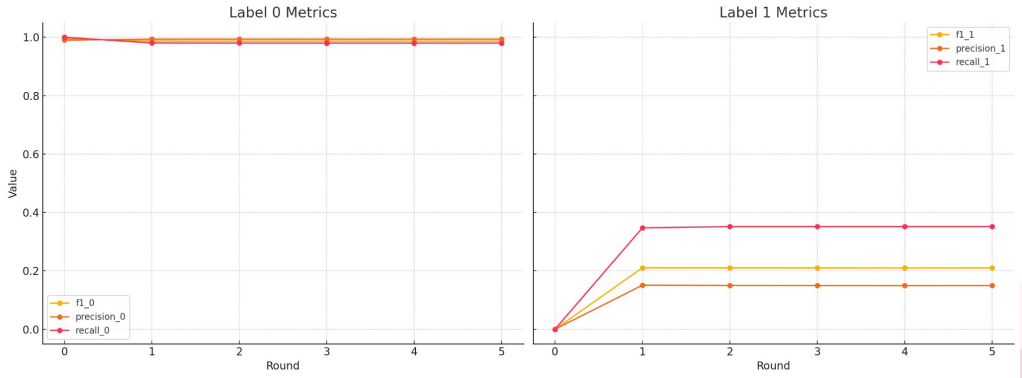
04. 결론

Centralized Learning

Label	Precision	Recall	F1-score
0	1.00	1.00	1.00
1	0.96	0.73	0.83

Federate Learning

Label (Round)	Precision	Recall	F1
0 (round 1)	0.993348	0.980348	0.986805
1 (round 1)	0.150909	0.347280	0.210393
0 (round 5)	0.993388	0.979927	0.986611
1 (round 5)	0.149733	0.351464	0.210000



04. Limitation & Future Work

❌ FL이 CL(중앙집중식)보다 비열등성 입증 실패

- **이유 1: 클래스 불균형**
  - 0/1 라벨 간 **심각한 비율 차이**
- **이유 2: Base 모델 불일치**
  - 클라이언트마다 서로 다른 XGB/CAT 모델
  - 메타 러너가 **일관된 메타데이터 학습 어려움**
  - 클라이언트별 predict_proba가 서로 다름
  - Base 모델이 각자 데이터에 과적합 → 극단적 분포 반영 → 글로벌 일반화 방해
- **이유 3 : 스태킹 모델의 적용**
  - 클라이언트별로 데이터가 제한적이므로, 메타 모델의 일반화 성능이 저하되었을 것으로 예상

🔧 Future Work (해결 방향)

1. **Base 모델 통일**
   - 모든 클라이언트에 **동일한 초기 Base 모델 및 시드 제공**
2. **서버 측 Base 모델 보완**
   - 서버도 클라이언트들의 Base 모델 모두 학습
3. **Aggregation 개선**
   - FedAvg 대신 **클래스별 가중치 보정 (weighted average)** 적용

05. One more thing

05. Try : CTGAN, Data Augmentation

- 라벨 불균형 문제를 해결하기 위해 CTGAN으로 class 1 데이터 생성

Flow

Step 1. 소수 클래스 샘플링

Step 2. 메타데이터 정의

Step 3. CTGAN 학습 → 소수 클래스 샘플 500건 생성

Results

원본 데이터에 비해 f1 score 0.02 상승, but 큰 변화 없음

```
서버 라운드 5 중앙 집중식 평가:
  정확도: 0.9750
  클래스 0 정밀도: 0.9937, 재현율: 0.9809, F1: 0.9873
  클래스 1 정밀도: 0.1688, 재현율: 0.3849, F1: 0.2347
  (가중 평균) 정밀도: 0.9855, 재현율: 0.9750, F1: 0.9798
```

KUBIG
DATA SCIENCE & AI

Thank You