사전학습 모델 기반 C/C++ 취약점 탐지 모델 개발

CCIT Project
Team 좋은데이

CCIT
CULTURE CONTENTS & INFORMATION TECHNOLOGY

목차

팀원 소개

92014980 김명규

92015295 이경재

92015051 김정욱

92015154 박준형

주제 선정 이유

코드 해석에 특화된 사전학습 모델을 활용함으로써 취약점 탐색에 특화된 모델을 개발하여 배포하고, 개발자나 화이트해커의 코드 리뷰 및 취약점 분석 과정에서 소스코드 상의 취약점 유형, 취약한 라인, 취약점 패치를 제공하는 모델을 개발한다.

프로젝트 관리

일정 관리

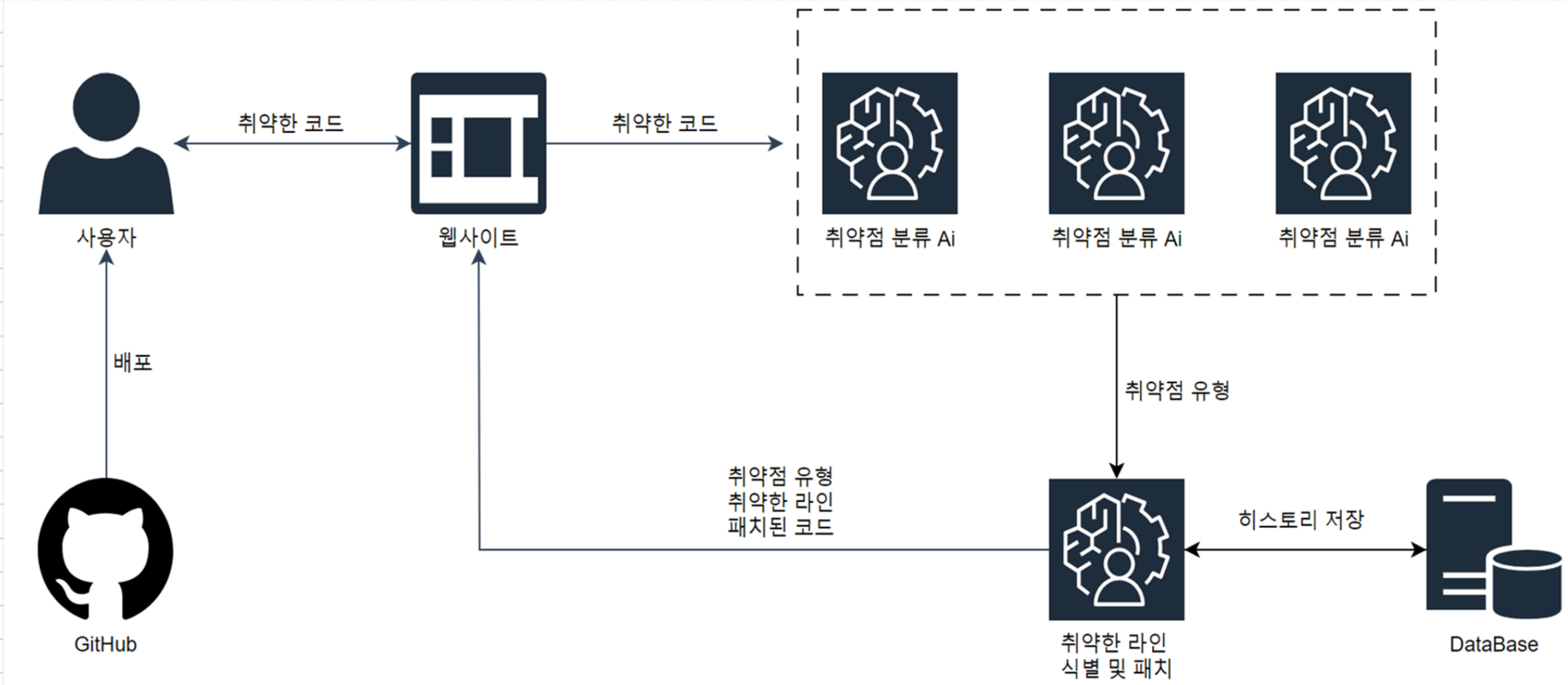
머신 러닝

Hugging Face

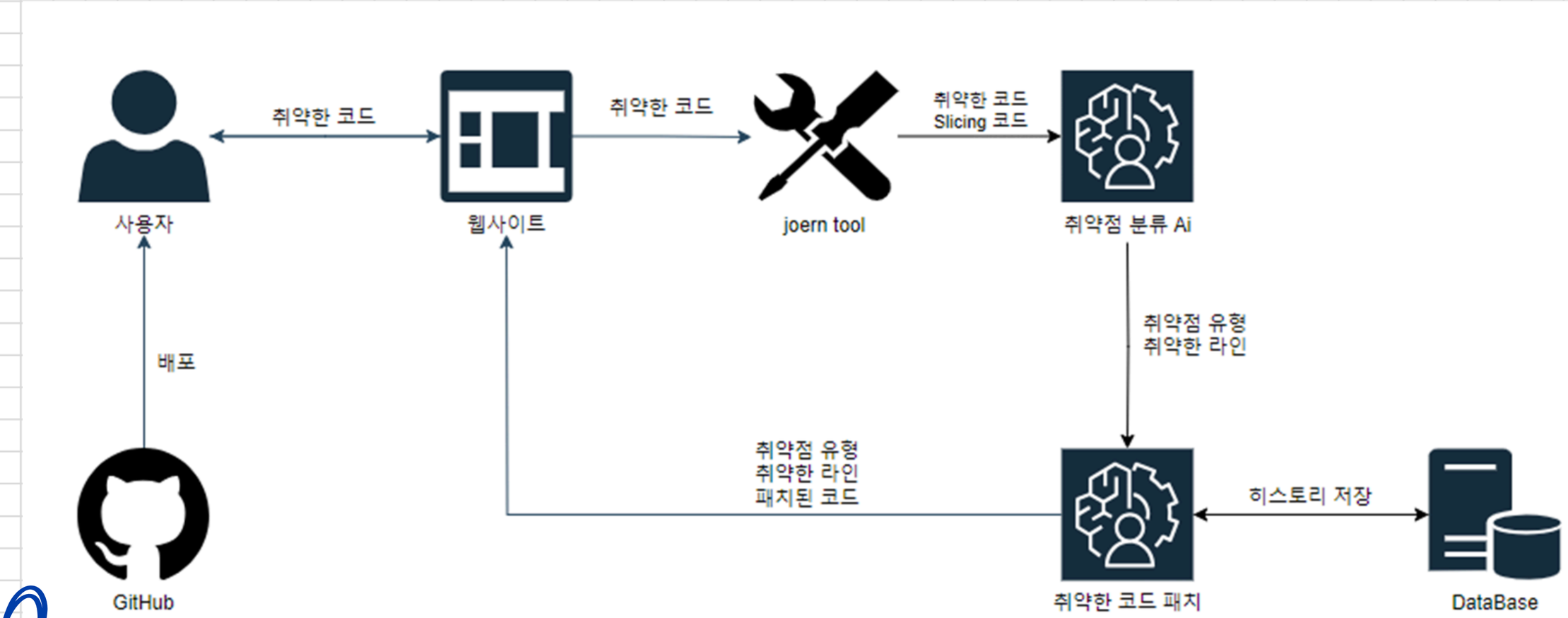
웹 사이트 구축

코드 관리

프로젝트 구성도(초안)


취약한 코드
취약한 코드
사용자
웹사이트
취약점 분류 Ai
취약점 분류 Ai
취약점 분류 Ai
배포
취약점 유형
취약점 유형
취약한 라인
패치된 코드
히스토리 저장
GitHub
취약한 라인
식별 및 패치
DataBase

프로젝트 구성도(최종)


취약한 코드
취약한 코드
취약한 코드
Slicing 코드
사용자
웹사이트
joern tool
취약점 분류 Ai
취약점 유형
취약한 라인
배포
취약점 유형
취약한 라인
패치된 코드
히스토리 저장
GitHub
취약한 코드 패치
DataBase

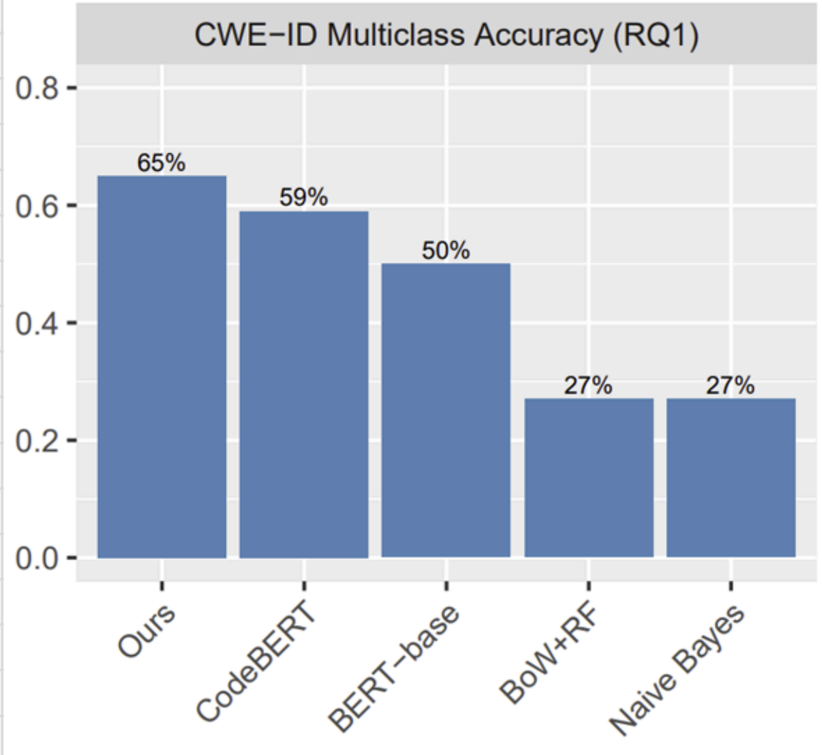
프로젝트 구성도(최종)

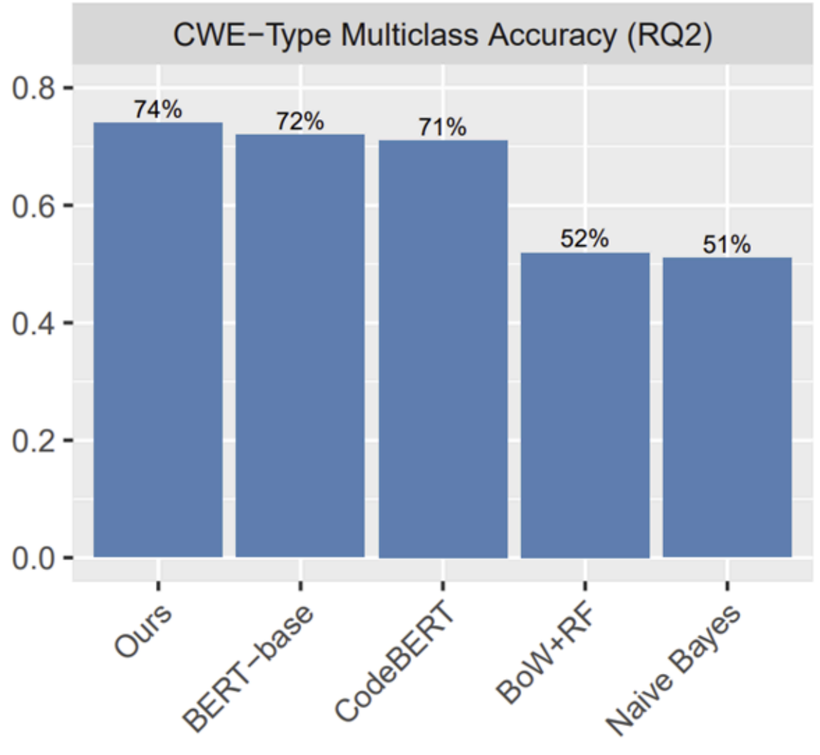

취약한 코드
취약한 코드
취약한 코드
Slicing 코드
사용자
웹사이트
joern tool
취약점 분류 Ai
취약점 유형
취약한 라인
배포
$git clone https://github.com/rlaaudrb1104/Ai.git
$pip install -r requirement.txt
$python3 joern-install.py
$python3 init_db.py
히스토리 저장
GitHub
취약한 코드 패치
DataBase

기존 연구 분석

AIBugHunter: A Practical Tool for Predicting, Classifying and Repairing Software Vulnerabilities

26 May 2023 · Michael Fu, Chakkrit Tantithamthavorn, Trung Le, Yuki Kume, Van Nguyen, Dinh Phung, John Grundy · Edit social preview





기존 연구 분석

Rank	CWE-ID	Name	Accuracy	Proportion
1	CWE-787	Out-of-bounds Write	43%	9/21
2	CWE-79	Cross-site Scripting	29%	2/7
3	CWE-125	Out-of-bounds Read	67%	44/66
4	CWE-20	Improper Input Validation	66%	71/107
7	CWE-416	Use After Free	52%	15/29
8	CWE-22	Path Traversal	0%	0/4
9	CWE-352	Cross-Site Request Forgery	0%	0/1
12	CWE-190	Integer Overflow	68%	21/31
14	CWE-287	Improper Authentication	0%	0/2
15	CWE-476	NULL Pointer Dereference	41%	7/17
17	CWE-119	Improper Restriction	79%	180/228
18	CWE-862	Missing Authorization	0%	0/1
20	CWE-200	Exposure of Sensitive Info	62%	26/42
22	CWE-732	Incorrect Permission Assignment	86%	6/7
23	CWE-611	Improper Restriction	50%	1/2
25	CWE-77	Improper Neutralization	0%	0/1
			67%	382/566

기존 연구 분석

Function Length (Tokens)	CWE-ID Accuracy	CWE-Type Accuracy
0-100	84%	85%
101-200	72%	81%
201-300	69%	71%
301-400	60%	69%
401-500	61%	70%
>500	57%	72%

프로젝트 진행 과정

데이터세트 수집

데이터 전처리

모델 선정 및 학습

데이터세트 수집

	code	lang	CWE ID	vul	code_y	labels
0	void char_01()₩n{₩n char * data;₩n /* In...	0	CWE-415	1	data = NULL;₩ndata = (char *)malloc(100*sizeof...	1
1	void char_02()₩n{₩n char * data;₩n /* In...	0	CWE-415	1	data = NULL;₩ndata = (char *)malloc(100*sizeof...	1
2	void char_03()₩n{₩n char * data;₩n /* In...	0	CWE-415	1	data = NULL;₩nif(5==5)₩ndata = (char *)malloc(...	1
3	void char_04()₩n{₩n char * data;₩n /* In...	0	CWE-415	1	data = NULL;₩ndata = (char *)malloc(100*sizeof...	1
4	void char_05()₩n{₩n char * data;₩n /* In...	0	CWE-415	1	data = NULL;₩ndata = (char *)malloc(100*sizeof...	1
...	...	...	...	...	...	...
9196	clump_splay_walk_init_mid(clump_splay_walker *...	0	safe	0	sw->from = SPLAY_FROM_LEFT;₩nsw->cp = cp;₩nsw-...	0
9197	free_all_allocator(clump_t *cp, void *arg)₩n{₩...	0	safe	0	struct free_data *fd = (struct free_data *)arg...	0
9198	free_all_not_allocator(clump_t *cp, void *arg)...	0	safe	0	struct free_data *fd = (struct free_data *)arg...	0
9199	gs_id get_mem_hdr_id (void *ptr)₩n{₩n retur...	0	safe	0	return (*((hdr_id_t *)((byte *)ptr) - HDR_ID_O...	0
9200	gs_memory_gc_status(const gs_ref_memory_t * me...	0	safe	0	*pstat = mem->gc_status;	0

9201 rows × 6 columns

```
CWE ID   labels
safe     0         1907
CWE-415  1         1658
CWE-119  2         1236
CWE-20   3          813
CWE-125  4          808
CWE-787  5          717
CWE-416  6          588
CWE-476  7          519
CWE-399  8          503
CWE-190  9          452
Name: count, dtype: int64
```

데이터 전처리

소스코드 상에서 취약점 탐지에 방해되는 부분 제거

- 싱글, 멀티 라인 주석 제거
- 전처리문
- 전처리 지시문 제거
- 개행 문자 제거

모델 선정

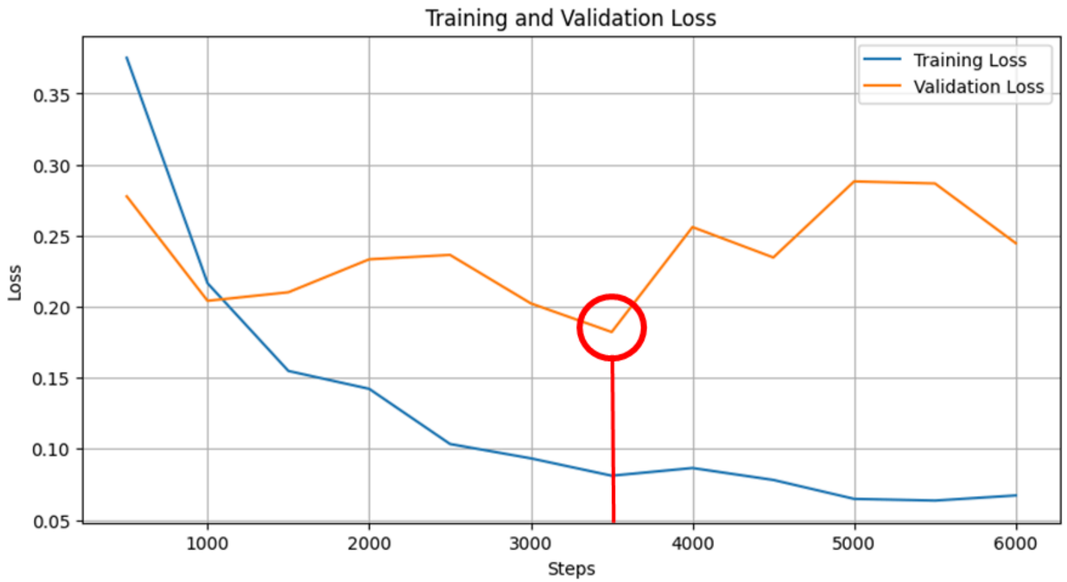
microsoft/codebert-base	프로그래밍 언어용 사전 학습 모델
microsoft/graphcodebert-base	코드의 고유 구조, 즉 데이터 흐름을 고려하는 프로그래밍 언어용 사전 학습 모델
microsoft/unixcoder-base	코드 관련 이해와 생성 작업을 모두 지원하는 프로그래밍 언어용 통합 크로스 모달 사전 학습 모델

모델 학습(이진 분류)

microsoft/codebert-base

Step	Accuracy	Recall Weighted	F1 Weighted	F1 Macro	F1 Micro
500	0.884271	0.884271	0.887510	0.812492	0.884271
1000	0.922421	0.922421	0.917561	0.850531	0.922421
1500	0.938406	0.938406	0.938420	0.894194	0.938406
2000	0.936488	0.936488	0.934784	0.885089	0.936488
2500	0.932438	0.932438	0.933626	0.888024	0.932438
3000	0.939685	0.939685	0.941161	0.901515	0.939685
3500	0.945013	0.945013	0.945805	0.908251	0.945013
4000	0.936488	0.936488	0.936120	0.889587	0.936488
4500	0.942242	0.942242	0.942445	0.901435	0.942242
5000	0.937127	0.937127	0.936368	0.889357	0.937127
5500	0.937340	0.937340	0.936695	0.890111	0.937340
6000	0.947997	0.947997	0.947355	0.908440	0.947997

과적합 방지를 위해 Early Stopping 적용



모델 학습(다중 분류)

모델명	Accuracy	f1-score-Weighted
Codebert	0.703283	0.701462
GraphCodebert	0.702899	0.699920
Unixcoder	0.710145	0.708684

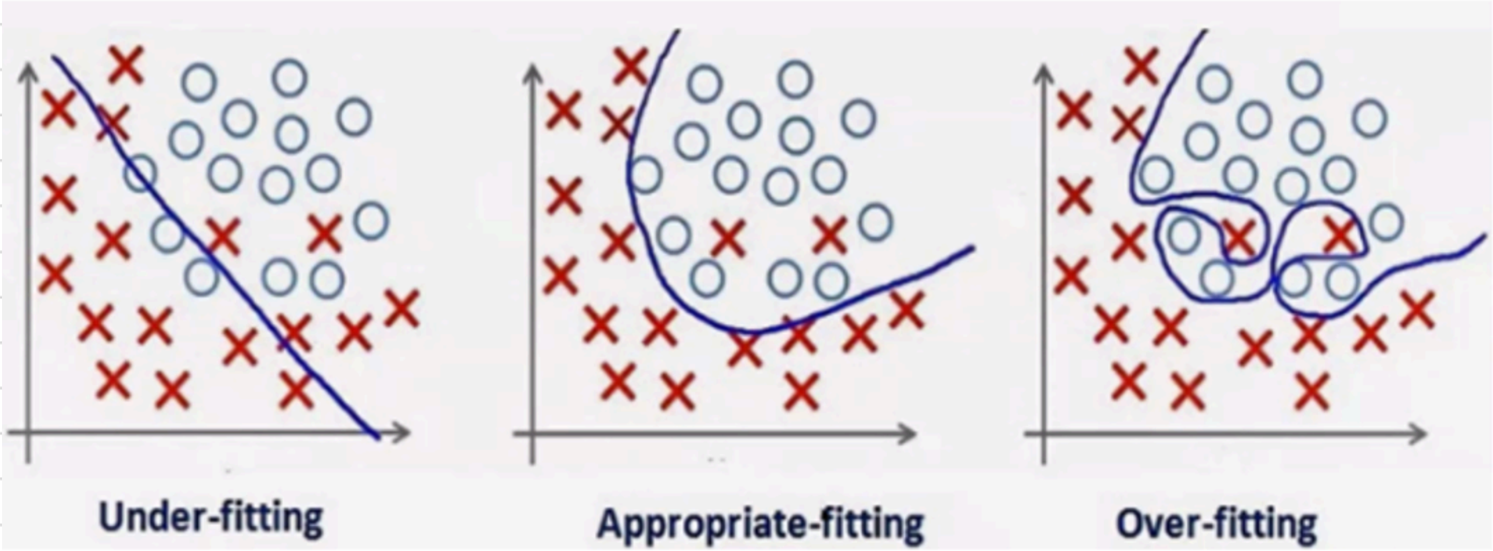
성능향상을 위한 시도

1. 규제화 및 오버샘플링을 통한 성능 향상

2. Program Slicing 기술을 통한 성능 향상

3. 피쳐 엔지니어링을 통한 성능향상

모델의 과적합을 방지하기 위해 규제화 시도



일반화 성능이 최대화 되는 모델을 찾는 것이 목표

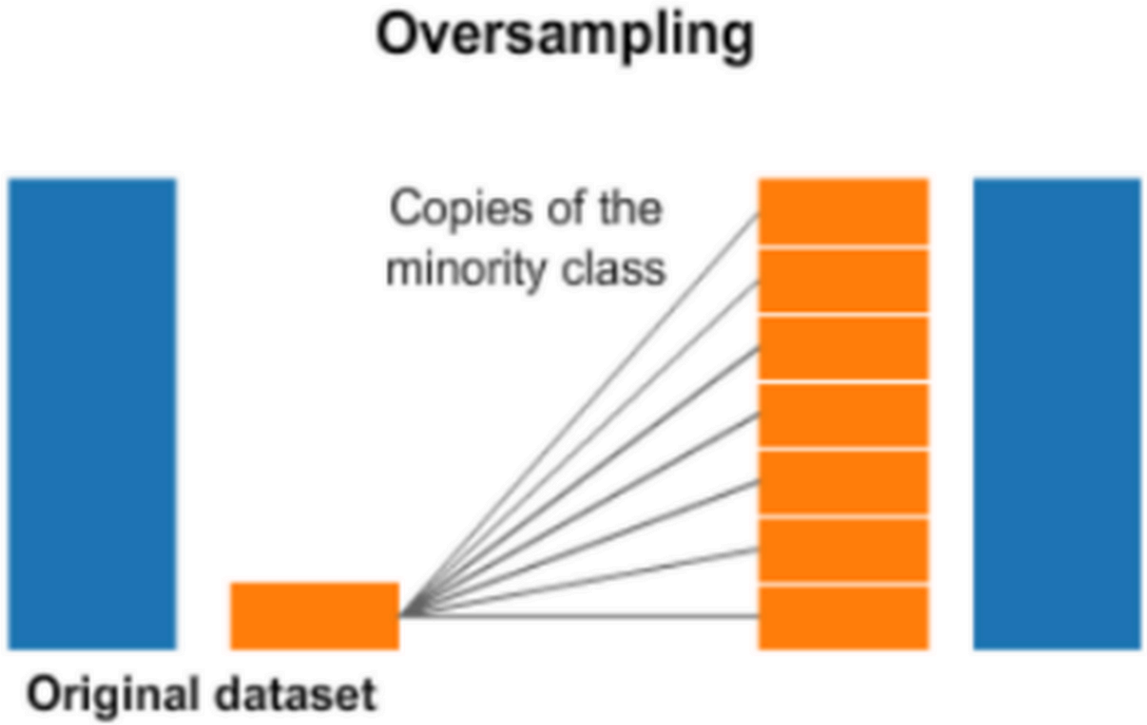
과대적합(Overfitting)

· 너무 상세하고 복잡한 모델링을 하여 훈련데이터에만 과도하게 정확히 동작하는 모델

과소적합(Underfitting)

· 모델링을 너무 간단하게 하여 성능이 제대로 나오지 않는 모델

오버샘플링을 통한 데이터 불균형 문제 해소



오버샘플링을 통한 데이터 불균형 문제 해소

CWE ID	labels	
safe	0	1907
CWE-415	1	1658
CWE-119	2	1236
CWE-20	3	813
CWE-125	4	808
CWE-787	5	717
CWE-416	6	588
CWE-476	7	519
CWE-399	8	503
CWE-190	9	452

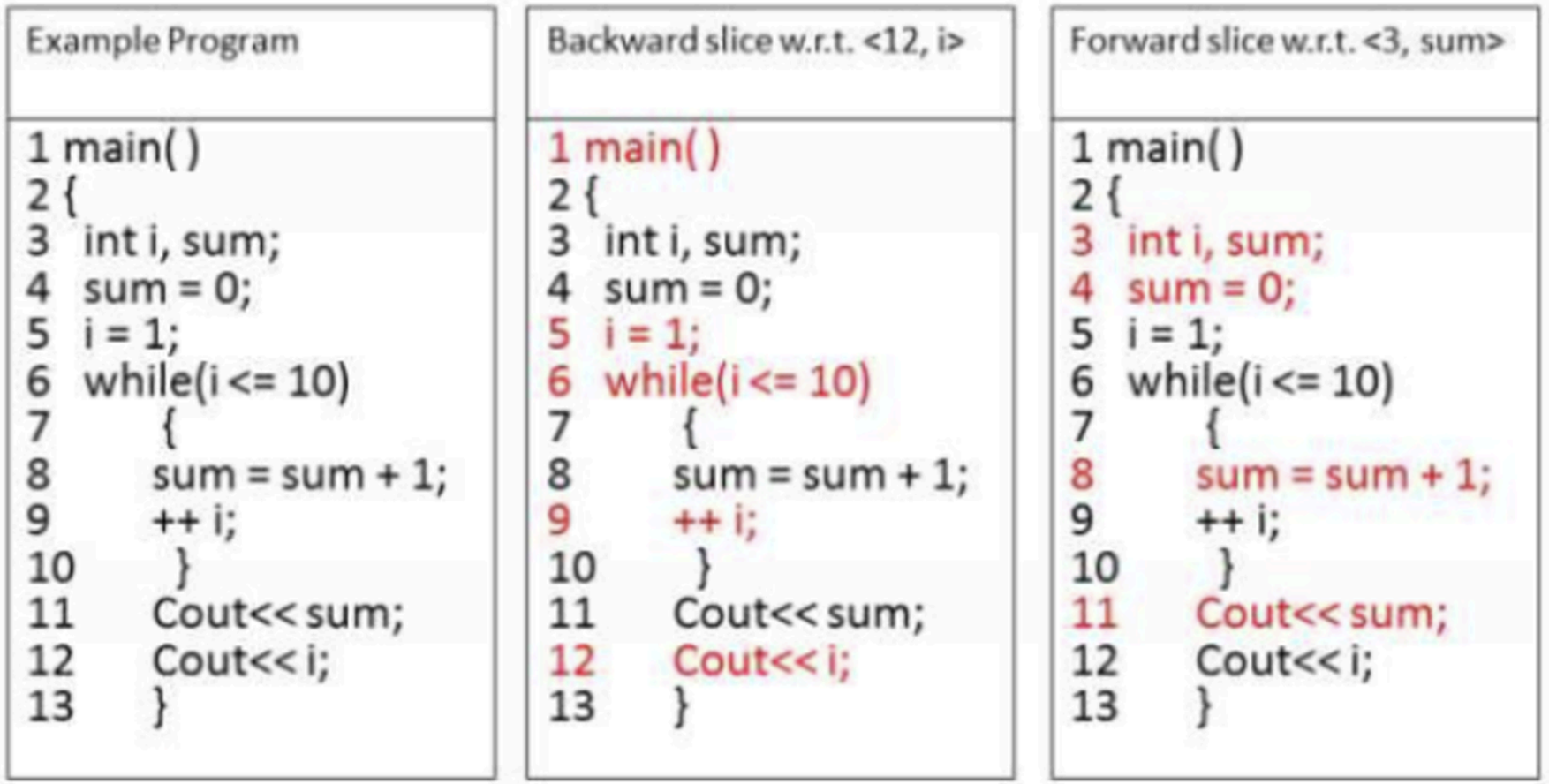
Name: count, dtype: int64

CWE ID	labels	count
safe	0	1907
CWE-415	1	1907
CWE-119	2	1907
CWE-20	3	1907
CWE-125	4	1907
CWE-787	5	1907
CWE-416	6	1907
CWE-476	7	1907
CWE-399	8	1907
CWE-190	9	1907

모델 학습(규제화 및 오버샘플링)

모델 이름	Accuracy	f1-score-Weighted
Codebert	0.867746	0.867256
GraphCodebert	0.905742	0.906381
Unixcoder	0.899471	0.900806

Proggram Slicing 기술을 통한 성능향상



모델 학습(Program Slicing)

모델명	Accuracy	f1-score-Weighted
Codebert	0.926067	0.926359
GraphCodebert	0.918025	0.917683
Unixcoder	0.927183	0.928029

피쳐 엔지니어링을 통한 성능향상

피쳐 엔지니어링 : 기존 데이터세트에 있는 피쳐를 재조합하거나 변형하여 의미 있는 피쳐를 만드는 과정

- 취약점 유형 마다 별도의 피쳐 추출
  - CWE-119 : 동적 할당된 메모리 사이즈와 버퍼 사용 인자 비교
  - CWE-190 : 정수 자료형 개수 확인
  - CWE-416 : malloc, calloc, realloc 함수와 free 함수의 갯수 비교

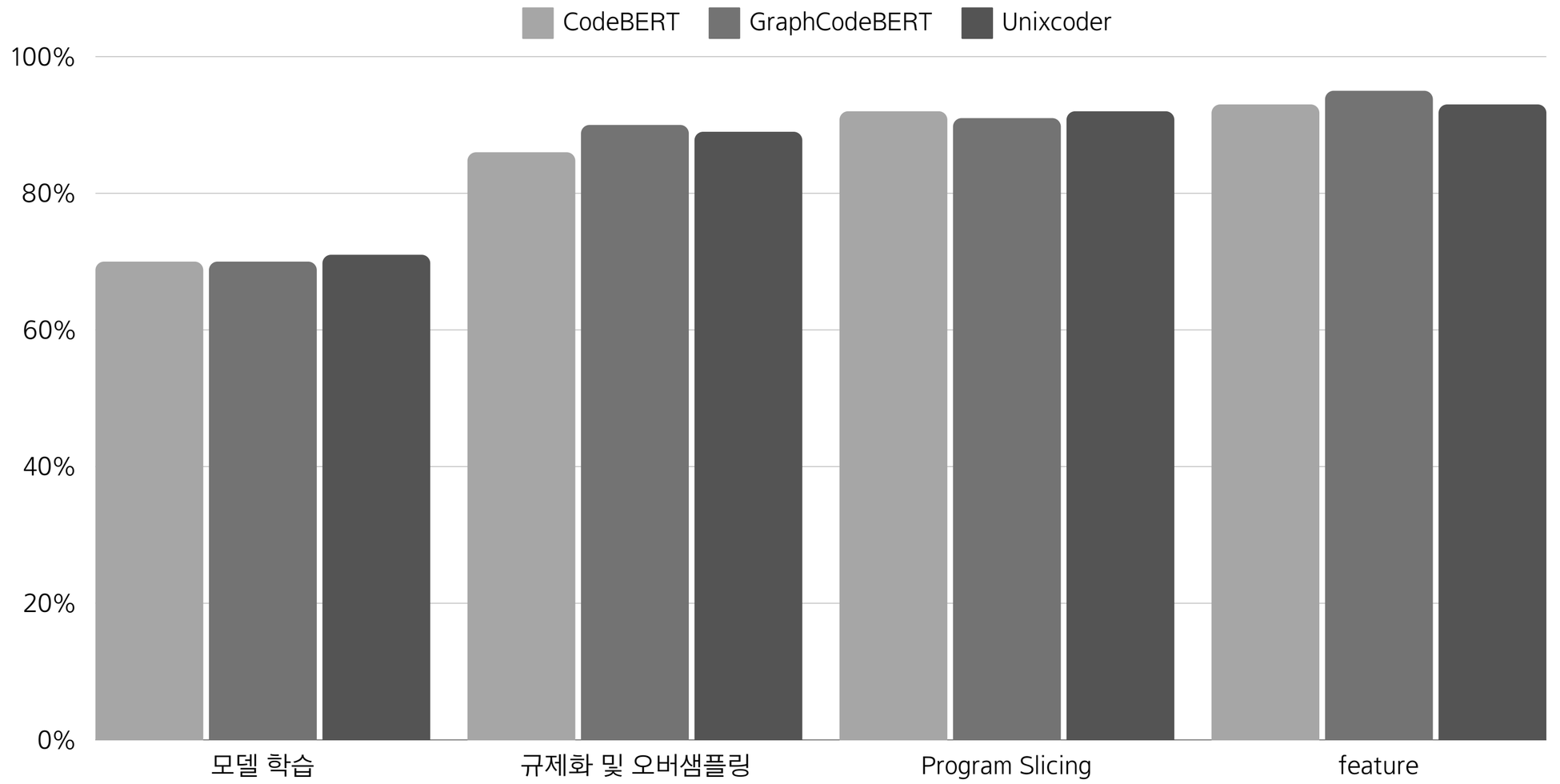
모델 학습(Program Slicing)

모델명	Accuracy	f1-score-Weighted
Codebert	0.926067	0.926359
GraphCodebert	0.918025	0.917683
Unixcoder	0.927183	0.928029

Program Slicing 코드를 피쳐로 설정

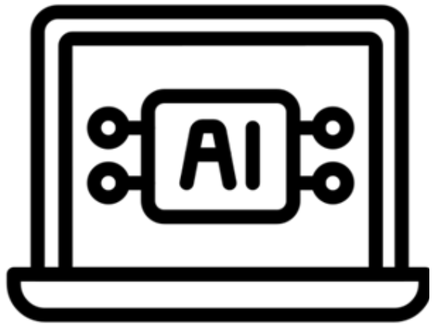
모델명	Accuracy	f1-score-Weighted
Codebert	0.939698	0.939934
GraphCodebert	0.951424	0.951522
Unixcoder	0.938693	0.938665

전체 결과


CodeBERT
GraphCodeBERT
Unixcoder
100%
80%
60%
40%
20%
0%
모델 학습
규제화 및 오버샘플링
Program Slicing
feature

프로젝트 산출물

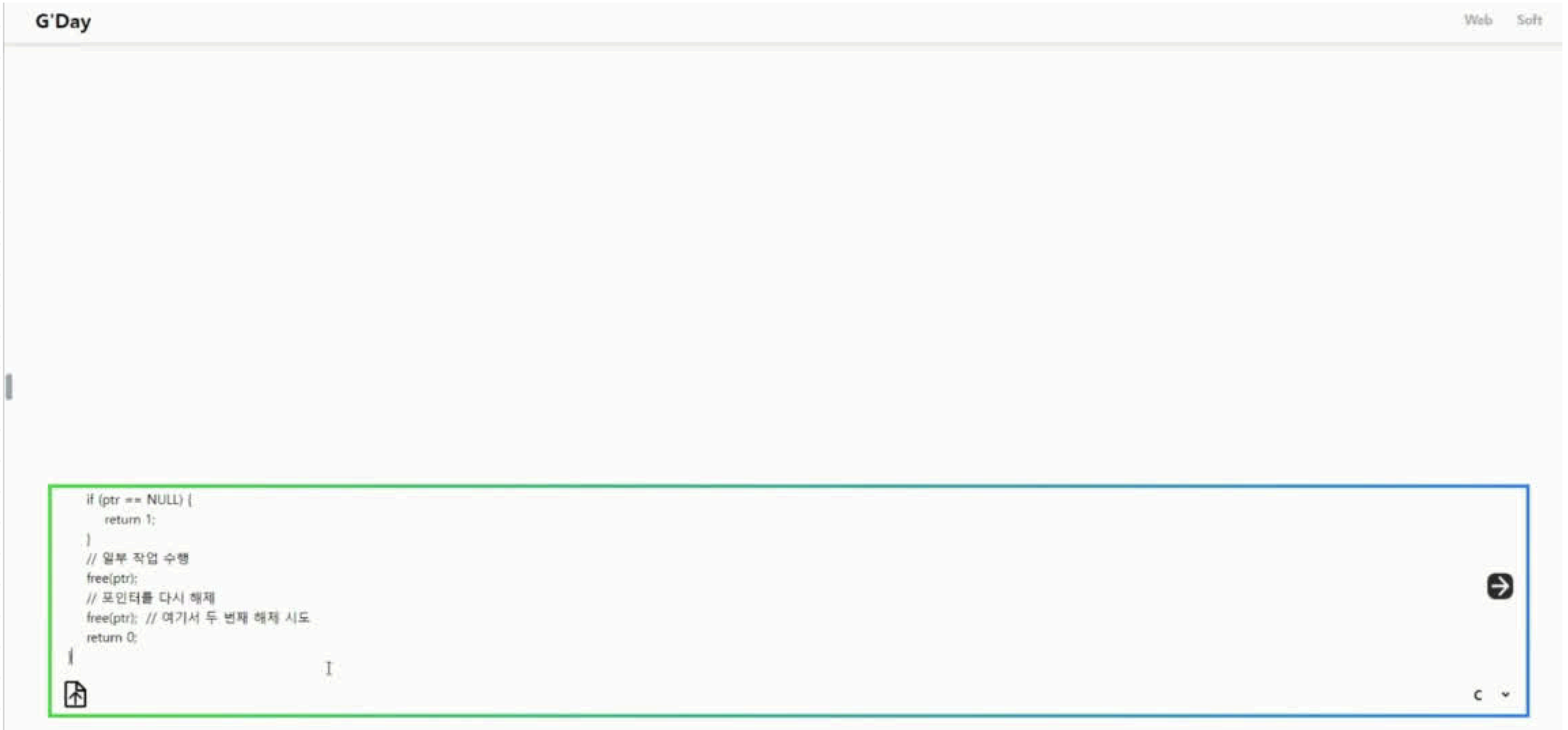
논문 작성



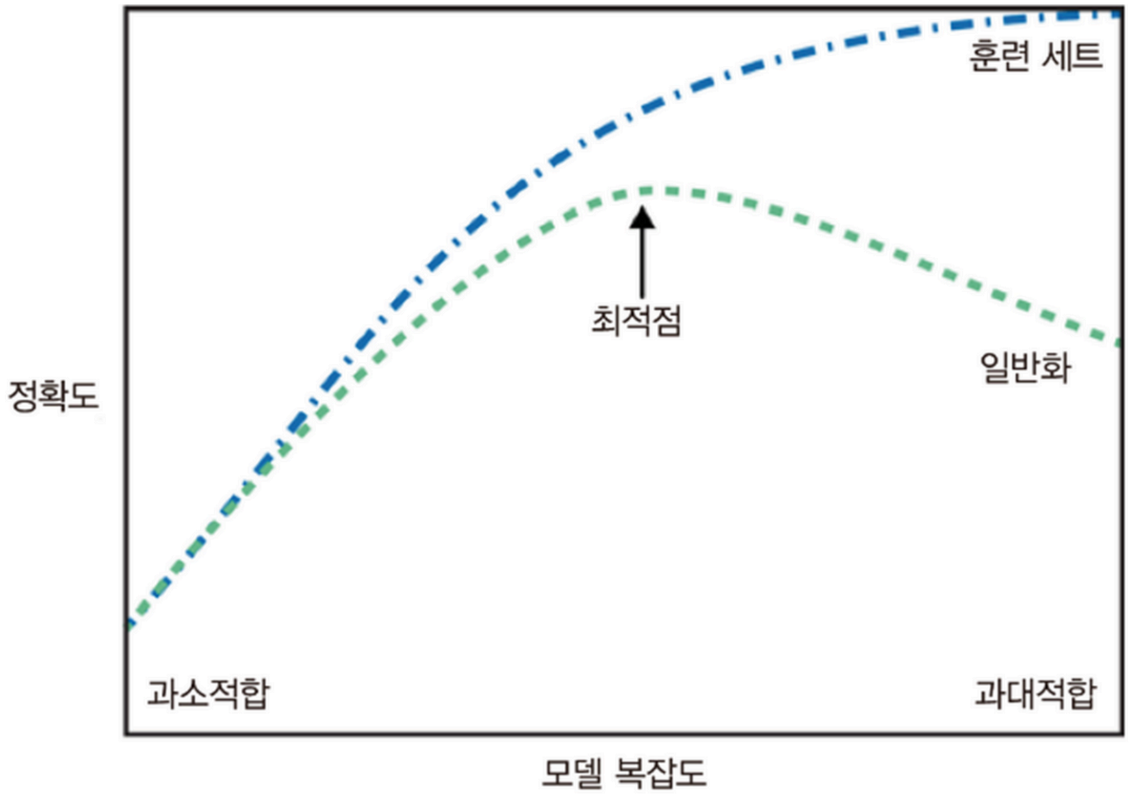
최종 Ai 모델 완성

웹사이트 배포

웹사이트 시연영상


G'Day
Web
Soft
if (ptr == NULL) {
return 1;
}
// 일부 작업 수행
free(ptr);
// 포인터를 다시 해제
free(ptr); // 여기서 두 번째 해제 시도
return 0;
}
C

프로젝트 개선사항



모델의 일반화 문제

Q&A