

Term Project

:Campus ChatBot

충남대학교
지능소프트웨어연구실

Term Project | Tasks

Basic

- 질문 유형 분류기 (Question Classification) **40 score**
- 챗봇 모델 및 인터페이스 구현 (Chat Model & UI) **60 score**

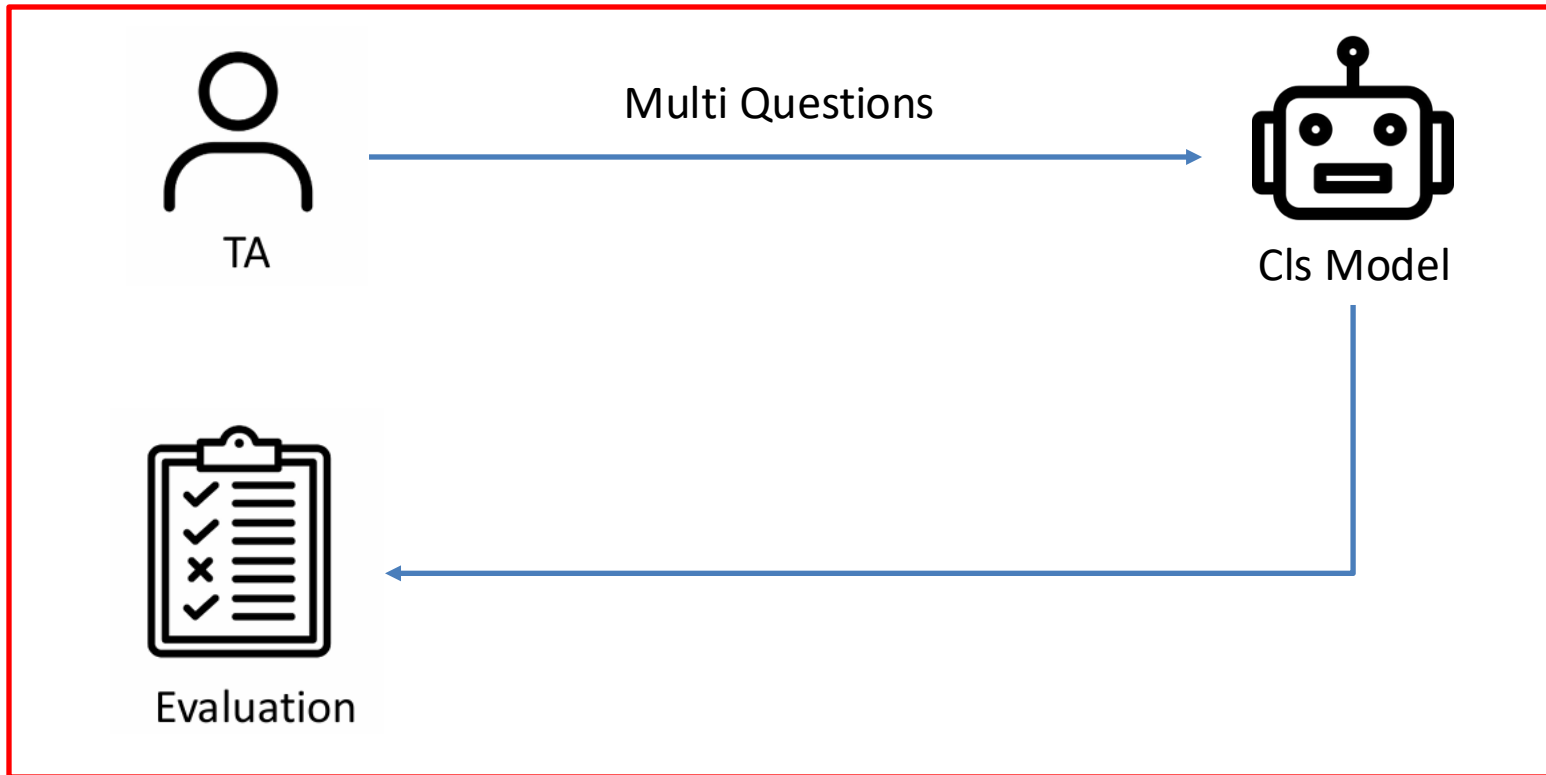
Optaional

- 실시간 정보 반영 **30 score**

Term Project | Tasks 1

❑ 질문 유형 분류기 (Question Classification)

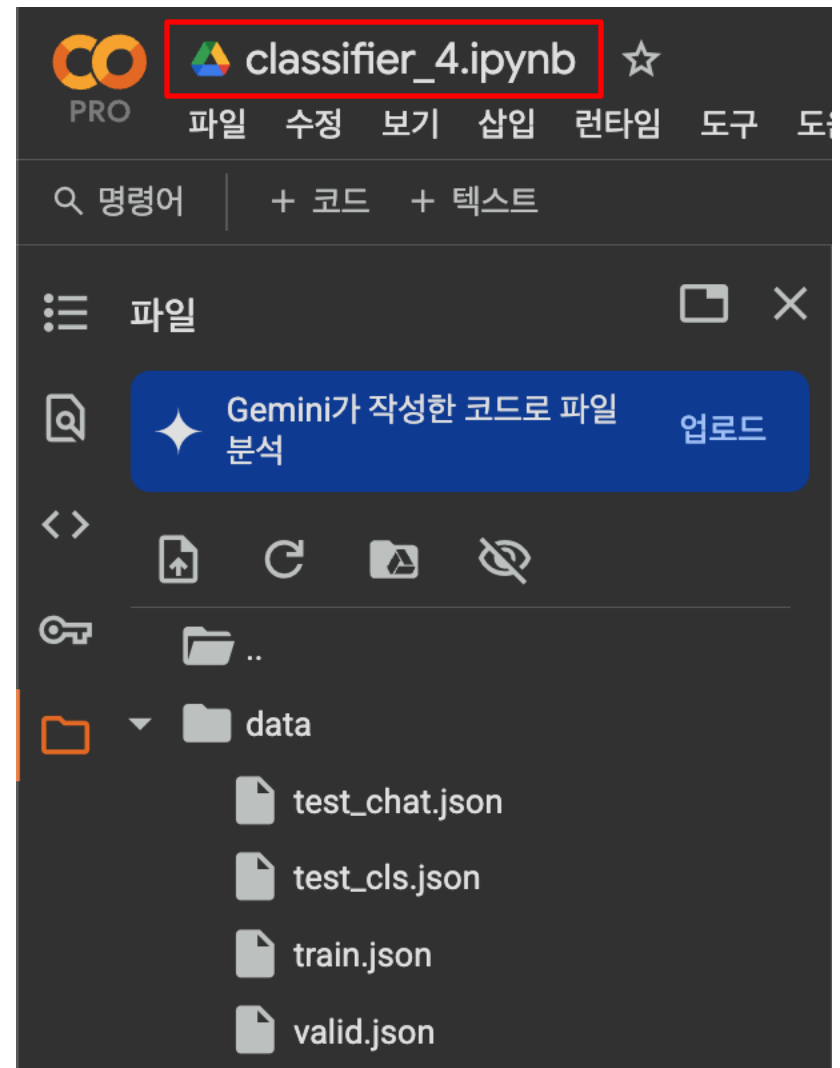
- 정량 평가 : F1 Score - 사용자의 질의를 5개의 카테고리 중 하나로 분류
- 테스트셋에 대한 모델의 예측 결과를 JSON 파일로 제출



Basic | 파일 명

- 파일 명

classifier_{조}.ipynb



Basic | 학습 데이터 예시

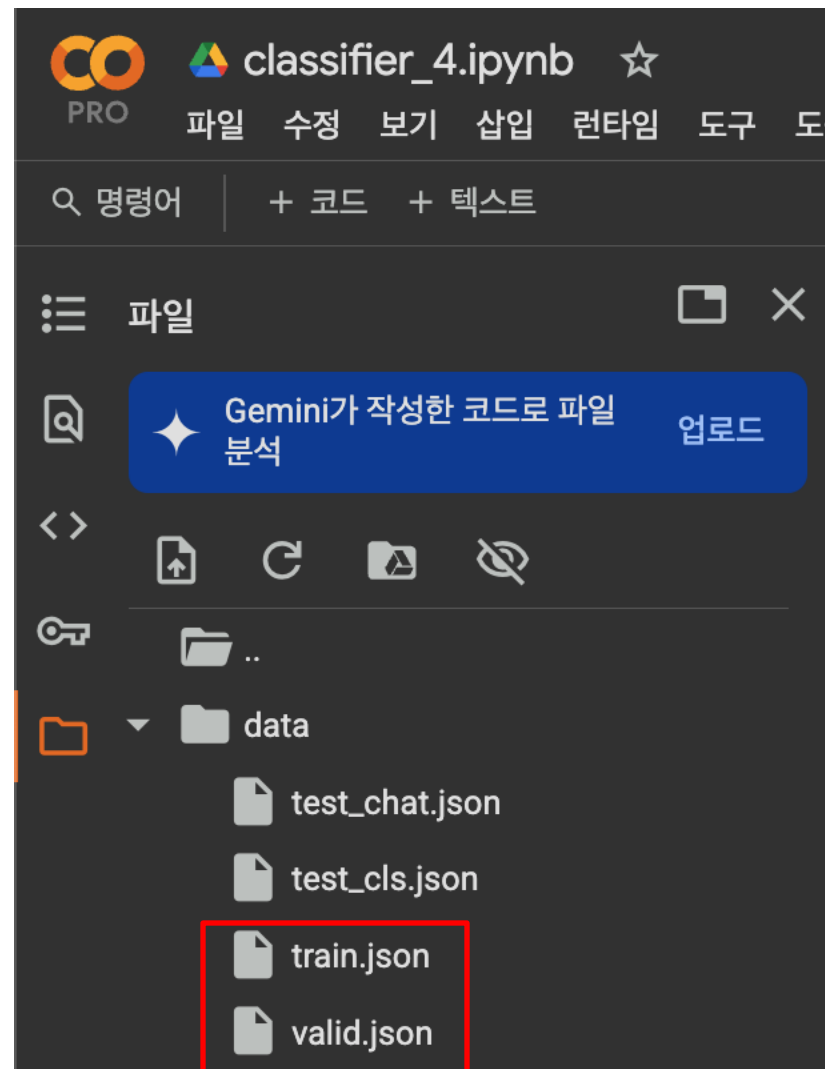
■ Train/ valid.json 파일

➤ 출력 라벨 준수

- 졸업요건 : 0
- 학교 공지사항 : 1
- 학사일정 : 2
- 식단 안내 : 3
- 통학/ 셔틀 버스 : 4

➤ Valid 데이터 미제공

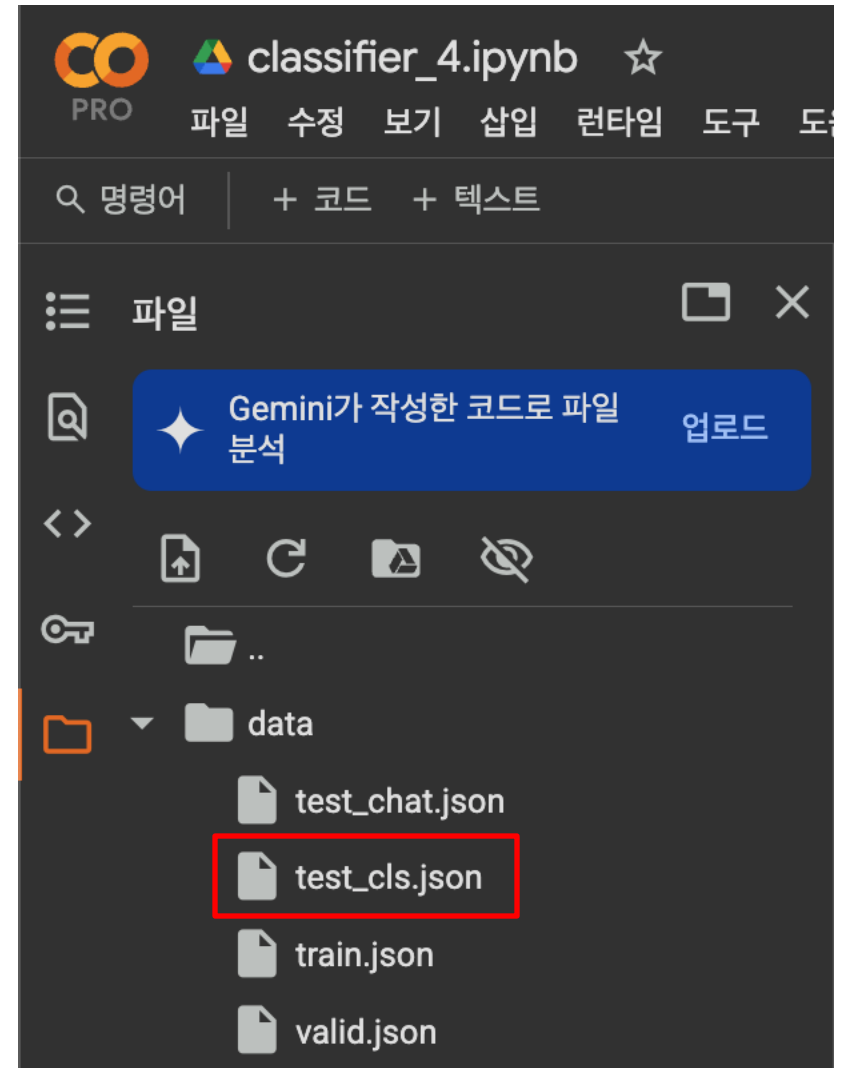
```
1 {
2   "question": "졸업까지 몇 학점을 들어야 하나요?",
3   "label": 0
4 },
5 {
6   "question": "이번 학기 수강신청은 언제 시작하나요?",
7   "label": 2
8 },
9 {
10  "question": "오늘 학식 뭐 나와요?",
11  "label": 3
12 },
13 {
14  "question": "다음주에 셔틀버스는 정상 운행하나요?",
15  "label": 4
16 },
17 {
18  "question": "이번에 올라온 공지사항 어디서 볼 수 있어요?",
19  "label": 1
20 },
21 }
```



Basic | 테스트 예시

- **평가**
 - /data/test_cls.json 파일에 대해 작동하도록 코드작성
- **test_cls.json 파일**
 - 평가 데이터 (조교 평가로만 사용)
 - 샘플 구조는 아래와 같음

```
1 [
2   {
3     "question": "졸업까지 몇 학점을 들어야 하나요?"
4   },
5   {
6     "question": "이번 학기 수강신청은 언제 시작하나요?"
7   },
8   {
9     "question": "오늘 학식 뭐 나와요?"
10  },
11  {
12    "question": "다음주에 셔틀버스는 정상 운행하나요?"
13  },
14  {
15    "question": "이번에 올라온 공지사항 어디서 볼 수 있어요?"
16  }
17 ]
```



Basic | system

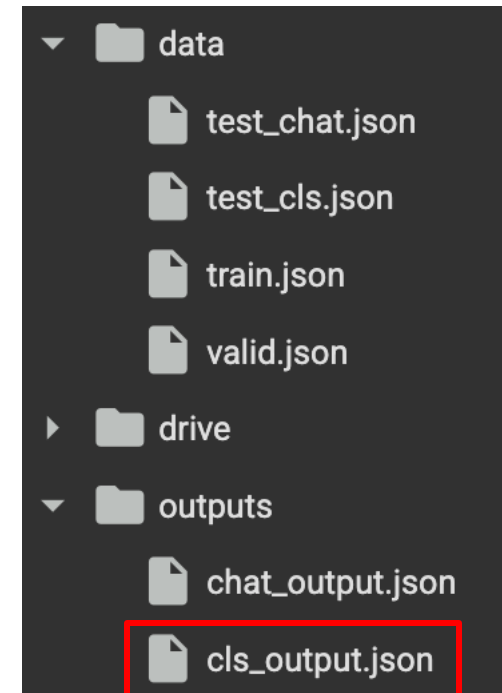
▪ *cls_output.json*

- *question* : 질문 (JSON 파일)
- *label* : 질문에 맞는 문제유형(라벨) 생성
- 코드 실행 시 /outputs 폴더 아래에 다음과 같은 구조의 JSON 파일 생성

cls_output.json

```
2  {
3  "question": "졸업까지 몇 학점을 들어야 하나요?",
4  "label": 0
5  },
6  {
7  "question": "이번 학기 수강신청은 언제 시작하나요?",
8  "label": 1
9  },
10 {
11 "question": "이번에 올라온 공지사항 어디서 볼 수 있어요?",
12 "label": 1
13 },
```

question → *label*



Term Project | Tasks 2

□ 챗봇 모델 및 인터페이스 구현 (Chat Model & UI)

- 정성 평가
 - UI 구동 여부 **10 socre**
 - Chat Iteration 형식 구현 여부 **10 socre**
 - 질문의 맥락에 알맞은 응답 **40 socre**

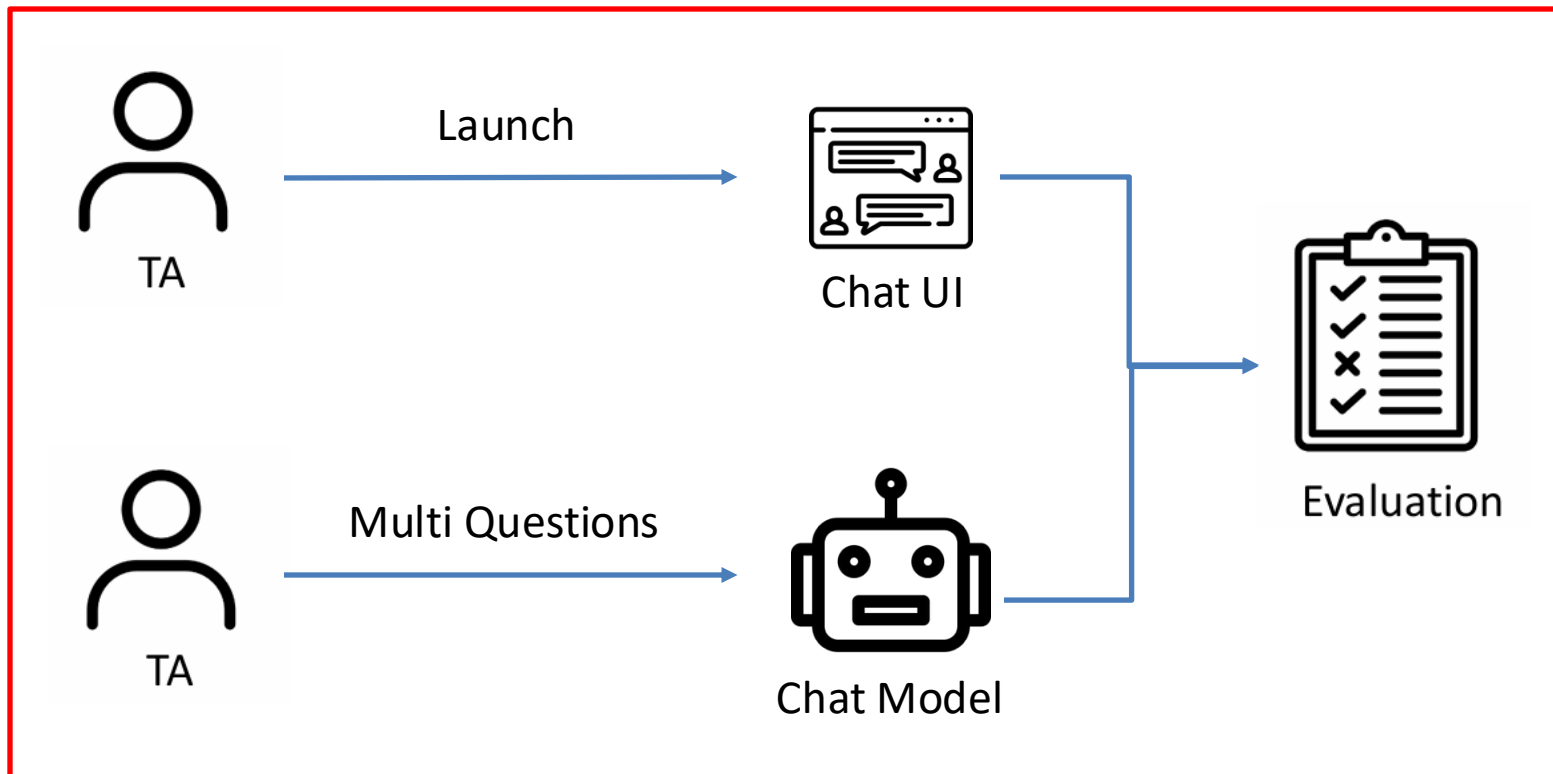
Term Project | Tasks 2

□ UI

- 사용자가 직접 질문을 입력하고 응답을 확인할 수 있는 UI 제공

□ Chat Model

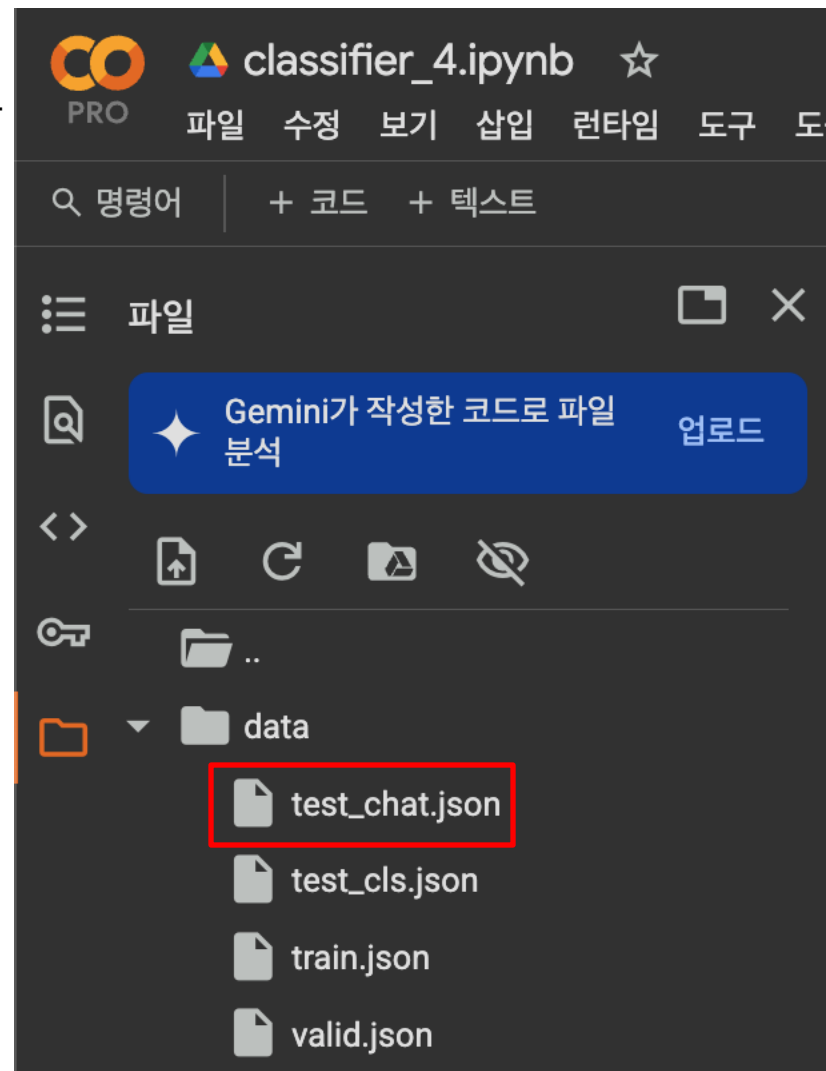
- UI 없이, 테스트셋에 대한 모델의 예측 결과를 JSON 파일로 제출



Basic | 테스트 예시

- 평가
 - /data/test_chat.json 파일에 대해 작동하도록 코드작성
- test_chat.json 파일
 - 평가 데이터 (조교 평가로만 사용)
 - 샘플 구조는 아래와 같음

```
1 [
2   {
3     "user": "졸업까지 몇 학점을 들어야 하나요?"
4   },
5   {
6     "user": "이번 학기 수강신청은 언제 시작하나요?"
7   },
8   {
9     "user": "오늘 학식 뭐 나와요?"
10  },
11  {
12    "user": "다음주에 셔틀버스는 정상 운행하나요?"
13  },
14  {
15    "user": "이번에 올라온 공지사항 어디서 볼 수 있어요?"
16  }
17 ]
```



Basic | system

▪ *chat_output.json*

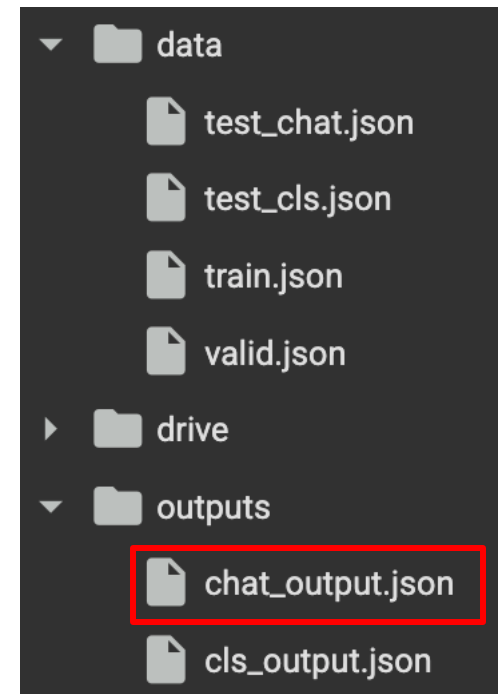
- *user* : 질문 (JSON 파일)
- *model* : 질문에 맞는 챗봇의 응답생성
- 코드 실행 시 /outputs 폴더 아래에 다음과 같은 구조의 JSON 파일 생성

chat_output.json

```
1 [
2   {
3     "user": "졸업까지 몇 학점을 들어야 하나요?",
4     "model": "졸업 요건은 전공에 따라 다르지만, 일반적으로 총 130학점 이상 이수해야 졸업할 수 있습니다."
5   },
6   {
7     "user": "이번 학기 수강신청은 언제 시작하나요?",
8     "model": "이번 학기 수강신청은 8월 7일(수) 오전 10시부터 시작됩니다."
9   },
10  {
11    "user": "오늘 학식 뭐 나와요?",
12    "model": "오늘 점심은 제육볶음, 계란국, 나물무침, 깍두기가 제공됩니다."
13  },
14  {
15    "user": "다음주에 셔틀버스는 정상 운행하나요?",
16    "model": "다음 주 셔틀버스는 정상 운행될 예정입니다."
17  },
18  {
19    "user": "이번에 올라온 공지사항 어디서 볼 수 있어요?",
20    "model": "학교 홈페이지 > 공지사항 게시판에서 확인하실 수 있습니다."
21  }
22 ]
```

user

model



Term Project | Tasks 3

□ 실시간 정보 반영

- 정성 평가 : 실제 정보와의 동기화 정확성

Optional | 테스트 예시

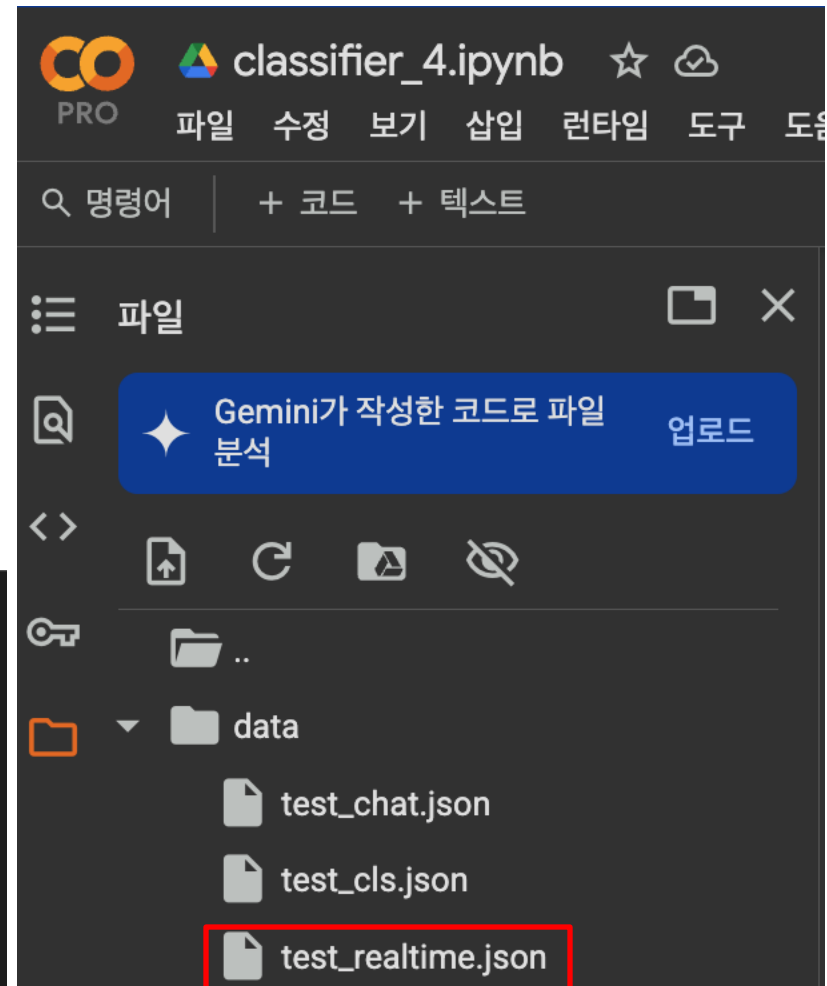
■ 평가

- /data/test_realtime.json 파일에 대해 작동하도록 코드작성

■ test_realtime.json 파일

- 평가 데이터 (조교 평가로만 사용)
- 샘플 구조는 아래와 같음

```
1 [
2   {
3     "user": "새로 업데이트된 셔틀버스 정류장이 있을까요?"
4   },
5   {
6     "user": "5월 이후로 변동된 학사일정이 있을까요?"
7   },
8   {
9     "user": "다음주 학식 뭐 나와요?"
10  },
11  {
12    "user": "가장 최근에 올라온 공지사항은 언제 게시되었나요?"
13  }
14 ]
```



Optional | system

▪ *realtime_output.json*

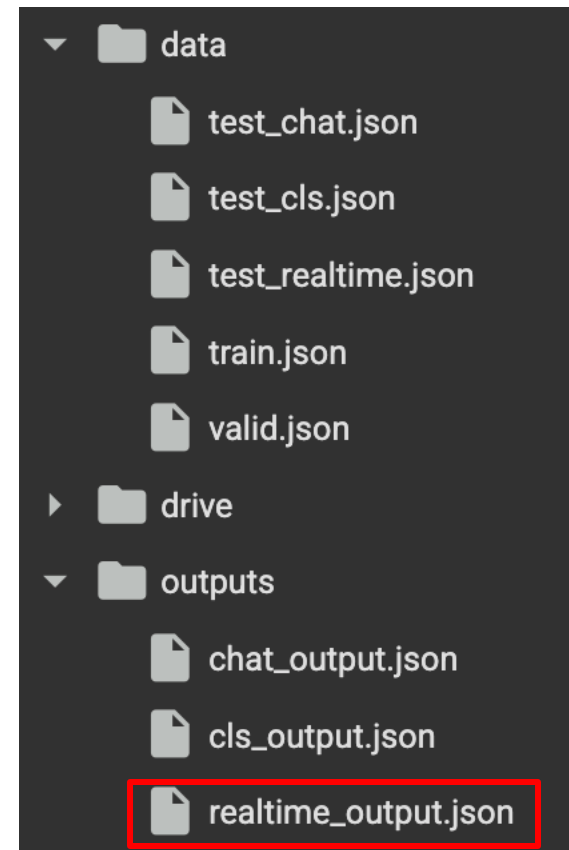
- *user* : 질문 (JSON 파일)
- *model* : 질문에 맞는 챗봇의 응답생성
- 코드 실행 시 /outputs 폴더 아래에 다음과 같은 구조의 JSON 파일 생성

realtime_output.json

```
1 [
2   {
3     "user": "새로 업데이트된 셔틀버스 정류장이 있을까요?",
4     "model": "새로 업데이트 된 정류장은 없습니다."
5   },
6   {
7     "user": "6월 이후로 변동된 학사일정이 있을까요?",
8     "model": "6월 3일이 임시공휴일로 지정되었습니다."
9   },
10  {
11    "user": "다음주 학식 뭐 나와요?",
12    "model": "다음주 월요일 아침은 밥,미역국,계란말이,김치,불고기가 나오고
13  },
14  {
15    "user": "가장 최근에 올라온 공지사항은 언제 게시되었나요?",
16    "model": "5월 26일에 '2024년 교육연구 학생지도비 예결산 현황 및 지급
17  }
```

user

model



Requirements

- 지정된 라이브러리 설치

- Python 3.10.12
- torch 2.5.1
- Pytorch-lightning 2.4.0 (사용할 경우)

- 필요 패키지 설치 가능

- [제출] requirements.txt (본인 local에 설치한 패키지 정보)

- 터미널에서 아래 명령어를 통해 패키지 목록 requirements.txt로 생성 가능

```
pip freeze > requirements.txt
```

Directory Structure & Filename

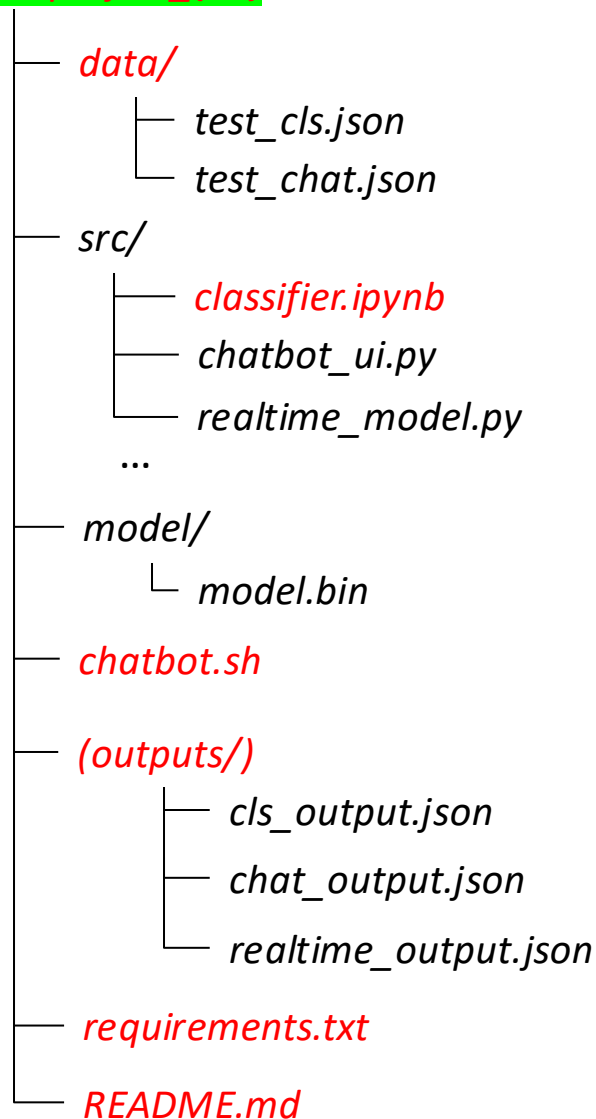
Example

src/classifier.ipynb 실행
↓
JSON파일 생성

chatbot.sh 실행
↓ ↓
UI 실행 JSON파일 생성

1. 오른쪽 경로 참고하여
유동적으로 구성할 것
2. 빨간색으로표기된 경로 구조는
평가를 위해 그대로 사용
3. 평가 시 classifier.ipynb와
chatbot.sh 만 실행!!

Termproject_{조}



Term Project Submission

텀프로젝트 최종 제출 안내	
제출 방법	사이버캠퍼스
파일 제출	소스코드, 모델 파일 및 다운로드 링크, 발표자료 Termproject_{조}.zip
제출 기한	6월 20일(금) 밤 12시

- 전체 코드 압축 후 제출
- Colab환경에서 실행가능한 .ipynb 또는 .py 형식으로 제출
- 용량 클 경우 google drive 에 올리고 공유 (다운로드 받을 수 있도록 설정)
- 제출 전 코드 정상 작동 확인
- 발표 자료 (5분 내외)
 - 구현 방법 + 챗 인터페이스 동작 여부