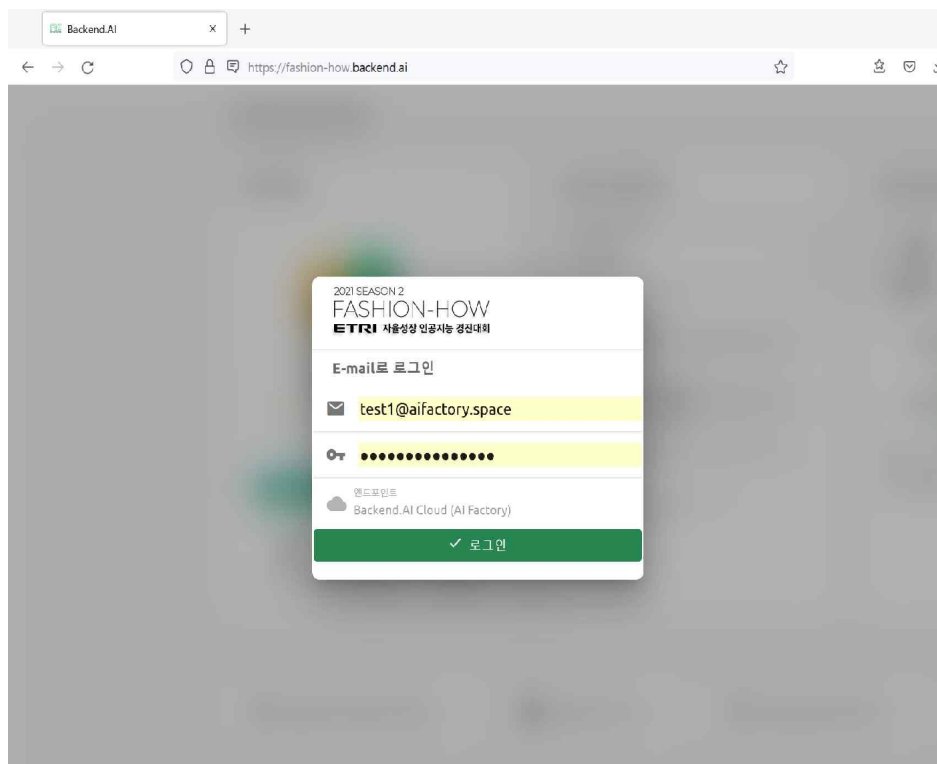


ETRI 자율성장 인공지능 경진대회 제출방법 가이드 (ver.2022)

- 꼼꼼히 순차적으로 읽고 따라하시면 됩니다! -

[로그인]

- <https://fashion-how.backend.ai>로 접속합니다.
- 팀별 email로 전달된 계정정보를 사용하시면 됩니다.



[로그인용 계정]

- 팀별로 각 태스크마다 서로 다른 계정을 사용하도록 안내가 되었습니다.
 - 각 팀은 1,2번 중 하나, 3,4번 중 하나로 총 2개의 태스크를 참가할 수 있습니다.
 - 따라서 4개의 계정 가운데 **태스크 번호에 맞는 2개를** 사용하면 됩니다.

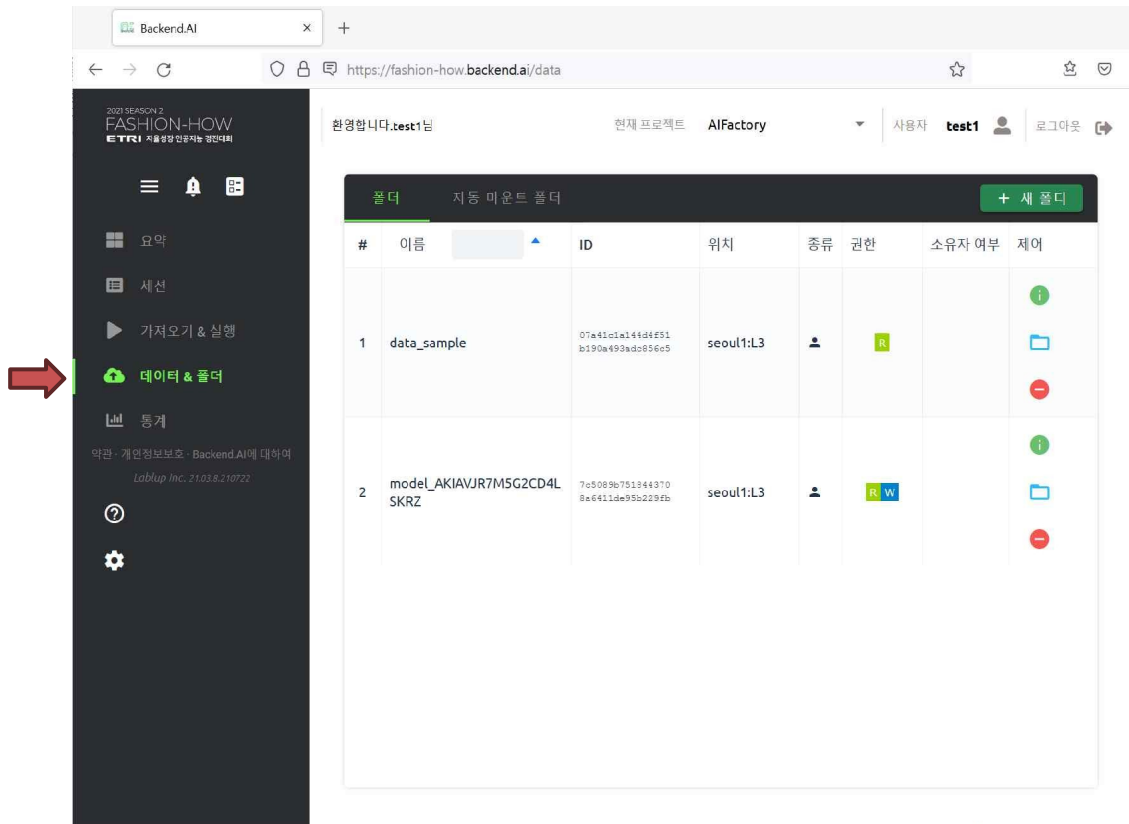
※ 각 계정에서는 지정된 태스크 외에는 제출이 정상적으로 이루어지지 않으니 유의 바랍니다!!

- 계정별 태스크 번호는 계정명에 있는 일련 번호로 확인하실 수 있습니다.
 - 예) aiXX_01 → 1번 태스크
 - aiXX_02 → 2번 태스크

....

[데이터&폴더]

- 로그인이 이루어지면 가장 먼저 화면 좌측 메뉴의 [데이터&폴더] 탭을 확인합니다.



- 위 그림과 같이 [데이터&폴더] 탭에는 두 가지 폴더가 있습니다.

1) data_sample

- 샘플 평가 데이터가 있는 폴더입니다. (읽기 전용)
- 샘플 데이터는 실제 평가 데이터와 동일한 이름과 양식으로 이루어져 있습니다.
- 샘플데이터는 학습데이터에도 함께 제공됩니다.
- 폴더 하위에는 task1, task2, task3, task4 4개의 샘플 데이터 폴더가 있습니다.
- 또한 폴더 하위에는 baseline 폴더가 있어서 이곳에서 baseline 모델을 가져올 수도 있습니다.

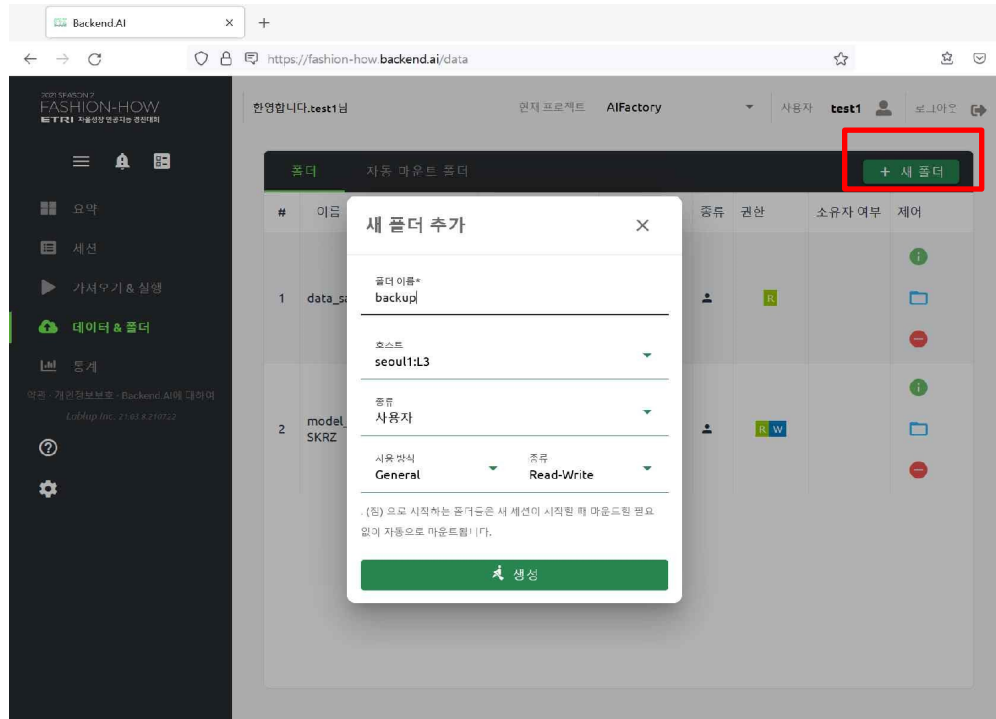
2) model_*

- 사용자가 제출할 모델을 업로드 하기 위한 폴더입니다.
- 초기에는 완전히 비어있으며 이 경로로 제출할 모델을 업로드합니다.

※ 위 두 폴더는 반드시 유지해야 함

[새 폴더]

- 아래 그림과 같이 우측 상단의 새 폴더 버튼을 클릭하면 사용자 전용 폴더를 만들 수 있습니다



- 폴더를 생성하여 마운트하면 세션에 마운트하여 사용할 수 있습니다
- ※ 단, 사용자가 생성한 폴더는 [제출→모델가동→결과생성] 과정에서는 사용되지 않음에 유의바랍니다. 즉, 모델을 가동하여 추론하는 데 필요한 경로는 모두 **model_*** 폴더 하위에 위치해야 합니다.
- model_*** 폴더 및 사용자 생성 폴더의 내용은 세션이 종료되어도 보존되므로 중요한 파일 백업은 생성한 폴더에 저장합니다.

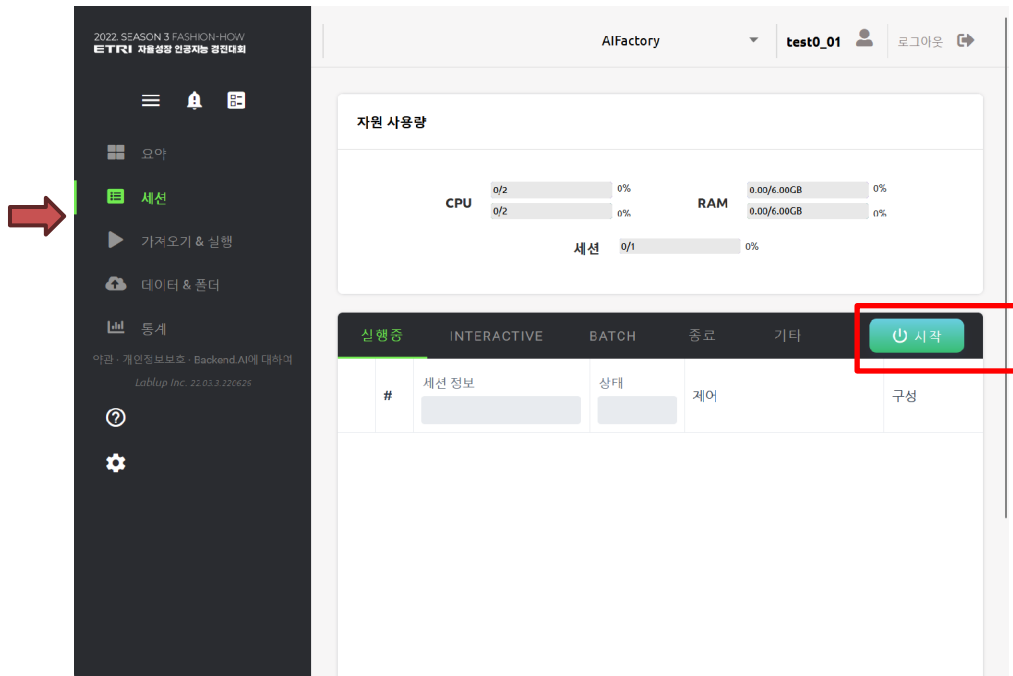
[파일 업로드]

폴더 자동 마운트 폴더								
#	이름	ID	위치	가용량	종류	권한	소유자 여부	제어
1	data_sample	07a41e1a114d5f51b190e493ade056c5	seoul1:1.3	-	사용자	R		<div> <div></div> <div></div> <div></div> </div>
2	model_AKIAWS6IDDC2AN4EXRBU	037e274242744ccb8a22553c61d68fe9	seoul1:1.3	-	사용자	R W		<div> <div></div> <div></div> <div></div> </div>

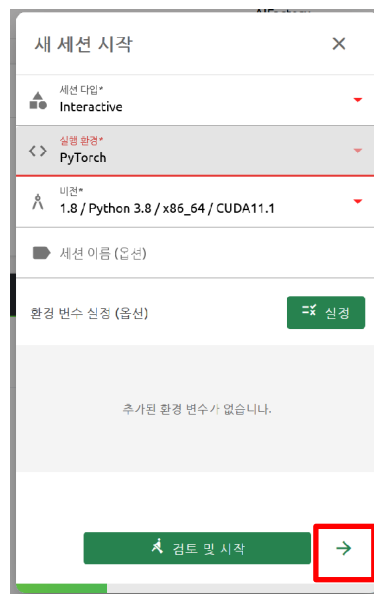
- 위 이미지의 붉은 박스는 각 폴더를 여는 버튼입니다.
- 각 폴더를 열어 드래그 앤 드롭으로 파일을 업로드할 수 있습니다.
- 또한 폴더를 열어 폴더 내에서 파일브라우저나 업로드 메뉴 등을 사용할 수 있습니다.
- 업로드 가능 단일 파일 용량을 초과하는 경우 분할압축을 이용합니다.

[작업/제출용 세션 생성]

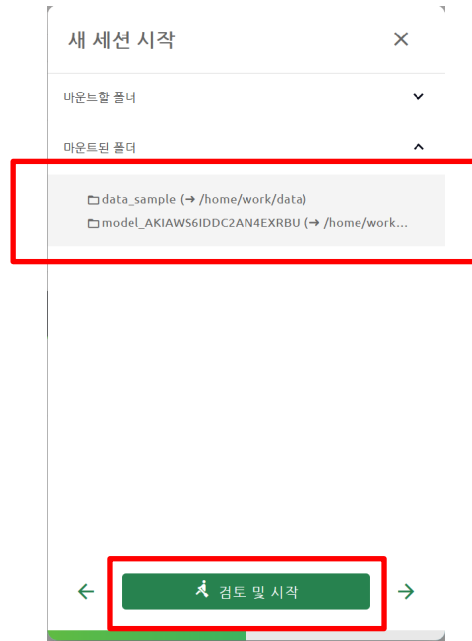
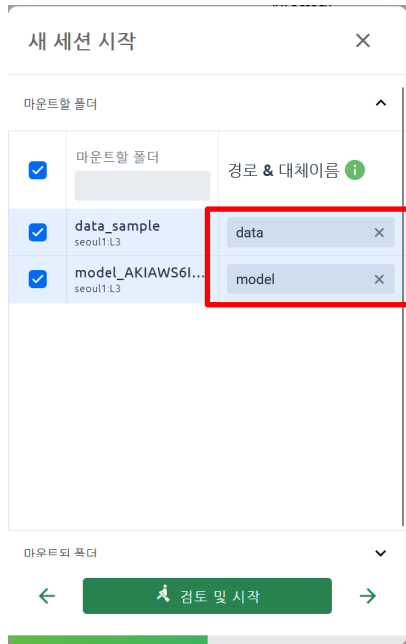
- 아래 그림과 같이 [세션] 탭을 눌러 현재 세션 정보를 확인한 다음
- 우측의 [시작] 버튼을 눌러 세션을 생성합니다.



- 그러면 아래 좌측 그림과 같이 [새 세션 시작] 창이 나타납니다.



- 이제 실행 환경을 선택합니다.
1) PyTorch, 2) Tensorflow 두 환경 가운데 하나를 사용
※ 베이스라인은 모드 PyTorch를 사용하므로 베이스라인 사용 시에는 PyTorch를 선택
- 나머지 옵션은 그대로 두고 우측 하단 → 화살표를 눌러 폴더 마운트 설정으로 이동합니다.



(※중요!) • 이어서 폴더를 마운트합니다.

- 먼저 각 마운트할 폴더를 우측의 대체이름(alias)으로 지정합니다.

- 대체 이름: data_sample → **data**

model_*... → **model** 로 대체이름을 지정하여 마운트해야 함(필수)

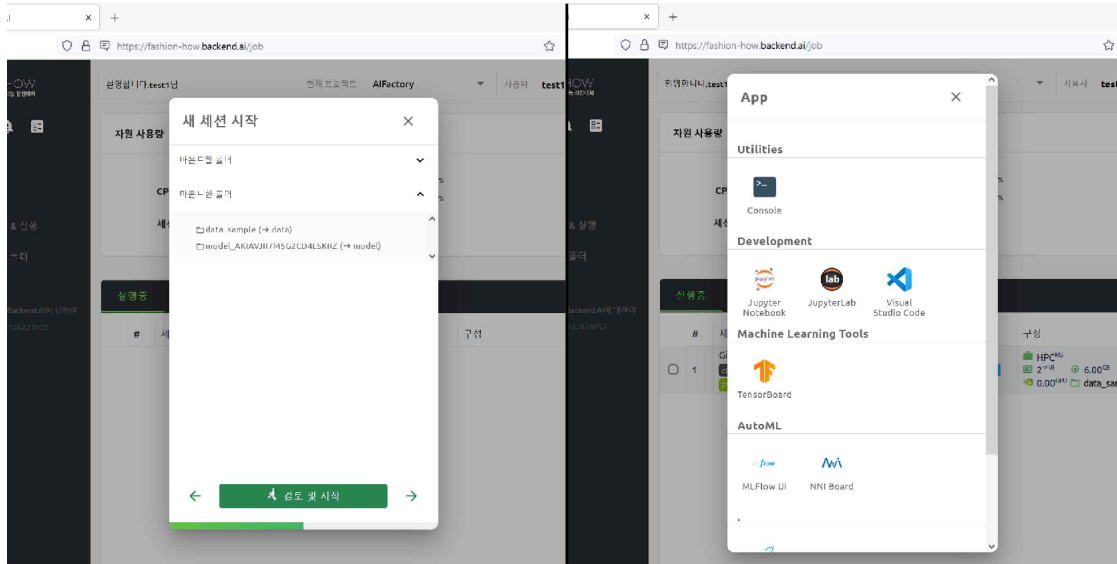
※ 마운트한 폴더는 세션에서 각각 /home/work/data, /home/work/model의 경로로 인식되며 사용자가 폴더를 추가하여 마운트하면 역시 /home/work/ 아래에 인식

- 즉, 제출할 모든 모델은 /home/work/model에 위치하게 되며
모델이 사용할 추론 데이터(샘플/테스트)는 /home/work/data/task1~4 에 위치하게 됩니다.
- 베이스라인이 이미 해당 경로를 사용하도록 구성되어 있으므로 직접 모델을 작업하실 때에는 각 태스크별 베이스라인의 파일 경로, 실행 경로, 결과 생성 경로 등을 참고바랍니다.

※ !주의!



- 폴더 마운트시 그림과 같이 [마운트한 폴더] 메뉴를 클릭하여 지정된 이름으로 마운트되었는지 반드시 **확인**이 필요합니다.
- 간혹 대체 이름이 적용되지 않은 경우 대체 이름을 삭제하고, 다시 입력한 후 마운트해서 확인합니다.
- 대체이름이 입력되지 않는 경우 브라우저의 입력이 한글로 설정되어있는지 확인 → 영어로



- 마운트 설정이 마무리되었으면 [검토 및 시작] → [시작]을 눌러서 세션을 생성합니다.
- 세션이 정상적으로 생성되면 우측과 같이 App 메뉴가 나타나며 Console 및 여타의 개발환경 앱(VS Code, Jupyter Notebook 등)을 열어서 세션에 접근할 수 있습니다.
- 환경 및 상황에 따라 세션 개설까지 약간의 딜레이가 있을 수 있습니다.

- 아래는 정상적으로 세션이 생성된 경우의 콘솔 화면입니다.
- **data, model, 사용자 생성 폴더 이외의 파일/환경을 수정할 경우 정상 제출이 되지 않을 수 있습니다!**

```
work@main1[ISiW80fh-session]:~$ ls
DO_NOT_STORE_PERSISTENT_FILES_HERE.md  data  id_container  model
work@main1[ISiW80fh-session]:~$
```

[0] 0:* "main1" 06:48 10-Aug-22

[경로] (중요!)

- 본 세션은 사용자가 각자의 환경에서 개발한 모델이 평가 세션에서 정상적으로 실행-추론-결과생성을 할 수 있는지 확인하는 목적으로 제공됩니다.
- 사용자가 생성한 세션에서 이하의 경로를 맞추어 작성한 모델이 평가 데이터 샘플을 이용하여 정상적으로 가동, 결과를 생성하는 경우 제출 이후 실제 평가 데이터를 이용한 추론 과정 역시 정상적으로 작동합니다.
- 제공되는 평가 데이터 샘플로 가동할 때 문제가 생기는 경우 실제 평가 데이터로도 정상 작동하지 않기 때문에 본 환경 내에서 가능한 오류를 파악하고 제출할 것이 권장됩니다.
- 정상 작동을 위해서는 이하의 경로를 반드시 준수해야 합니다.
- 사용자가 제출을 실행할 경우 아래와 같은 과정으로 평가가 수행됩니다.

- 사용자 세션과 독립된 별도의 임시 평가용 세션 생성
- 평가용 세션은 /home/work/data/task1~4에 샘플 평가 데이터가 아닌 **실제 평가 데이터**를 불러옴. 샘플과 실제 평가 데이터 사이의 양식은 동일
- 평가용 세션은 **사용자의 model_* 폴더를 /home/work/model** 경로로 마운트 (그밖의 폴더는 마운트되지 않음에 유의)
- 평가용 세션은 model 폴더 내의 **/home/work/model/predict.py** 실행
- **/home/work/model/prediction.csv** 생성
- 생성된 prediction.csv를 평가서버로 전송 → 평가

- 따라서 사용자는 **/home/work/model/predict.py**를 실행하면 모델의 추론이 실행되고 **/home/work/model/prediction.csv**로 결과를 저장하도록 모델의 코드를 작성해야 합니다.
- 또한 pip 등을 이용하여 추가로 라이브러리를 설치하는 경우, 해당 라이브러리가 실제 평가 세션에는 설치되어있지 않다는 점에 주의하고 **predict.py** 실행 시 필요한 라이브러리 설치가 함께 이루어질 수 있게 해야 함
- 평가용 세션에 마운트되는 폴더는 /model_* 하나이기 때문에, 모델이 정상 동작하는 데에 필요한 모든 리소스는 **/model_*** 폴더 하위에 저장하고 경로를 **/home/work/model/이하로** 지정해야 함

※ 이상의 경로는 반드시 준수해야 하며 (절대경로지정 권장)

※ prediction.csv 양식 또한 준수해야 합니다.

(베이스라인에서 predict.py를 수행하여 생성 결과를 확인합니다.)

[베이스라인 테스트]

```
_requirement._memn2n._B
_requirement._memn2n._C
_requirement._memn2n._H
_requirement._memn2n._W
_policy._mlp_eval.layer1_linear.weight
_policy._mlp_eval.layer1_linear.bias
_policy._mlp_eval.layer2_linear.weight
_policy._mlp_eval.layer2_linear.bias
_policy._mlp_eval.layer3_linear.weight
_policy._mlp_eval.layer3_linear.bias
_policy._mlp_eval.layer4_linear.weight
_policy._mlp_eval.layer4_linear.bias
_policy._mlp_rnk.layer1_linear.weight
_policy._mlp_rnk.layer1_linear.bias
_policy._mlp_rnk.layer2_linear.weight
_policy._mlp_rnk.layer2_linear.bias
_policy._mlp_rnk.layer3_linear.weight
_policy._mlp_rnk.layer3_linear.bias

<Predict>
[*] load success: /home/work/model/gAia_model/gAia-10.pt

-----
Prediction Time: 2.88sec
# of Test Examples: 200
-----
work@main1[sOeitKyK-session]:~/model$ ls
LICENSE.txt  data      gaia.py  pip.txt  prediction.csv  run_test.sh  submit.sh
README.pdf  file_io.py  log.txt  policy.py  requirement.py  run_train.sh
__pycache__  gAia_model  main.py  predict.py  run_pred.sh    sstm_v0p5_deploy
work@main1[sOeitKyK-session]:~/model$
```

- /home/work/model 에서
> **python predict.py** 로 실행
- 실행 종료되면 그림과 같이 **prediction.csv**가 생성되는 것을 확인할 수 있습니다.
- 베이스라인 상세 내용은 함께 제공되는 베이스라인 설명서를 참고바랍니다.

[제출]

- 1일 3회 제출 제한이 있음에 유의 바랍니다.
- 이전에 제출한 결과가 아직 추론중일 경우 끝날 때까지 제출이 되지 않습니다.
- 추론중 모델의 소스코드, 경로, 결과 양식 실수로 인하여 오류가 발생하는 경우 제출 회수에 가산되지 않으므로 다시 제출할 수 있습니다.

- 베이스라인에 포함되어 있는 submit.sh를 /home/work/model에 업로드하여 실행
> bash /home/work/model/submit.sh

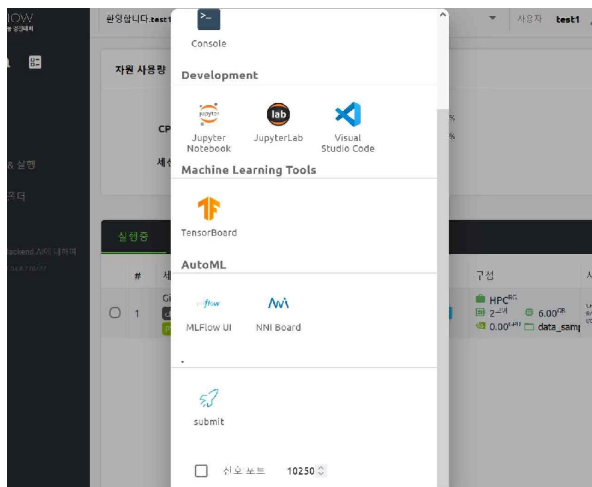
```
work@main1[s0eitKyK-session]:~/model$ bash submit.sh
{
  "message": "Submitted and queued the evaluation task",
  "data": {
    "evaluation_status": "RUNNING",
    "timestamp": "2021-08-08T09:32:10.774130+00:00"
  }
}
work@main1[s0eitKyK-session]:~/model$


{"message": "Endpoint request timed out"}
work@main1[s0eitKyK-session]:~/model$
```

- 정상 제출 시 좌측과 같은 메시지 출력

- “Endpoint request time out”이 발생하는 경우 일시적 장애이므로 곧바로 바로 재시도하면 됨

- 세션 메뉴의 submit 아이콘 실행



- 세션-제어 아래 메뉴 버튼() 클릭

- 스크롤을 내리면 submit 아이콘이 있음



- 클릭하면 bash 명령과 동일하게 제출됨

- 출력 메시지 및 오류:
- 추론이 모두 종료되면 생성-전송된 결과는 삭제되고, 아래와 같은 파일이 생성됨
 - /home/work/model/log.txt (화면 출력, 에러 포함 전체 메시지)
 - /home/work/model/error_log.txt (에러가 있을 시)

- 정상 제출 완료시 log.txt에서 다음과 같은 메시지를 확인할 수 있음

```
-----
100% | ██████████ | 1.37k/1.37k [00:00<00:00, 73.6kB/s]
success: 200
```

※ 결과가 정상 제출되더라도 생성한 prediction.csv의 양식이 잘못된 경우 평가서버의 점수 계산과정에서 오류가 발생할 수 있으므로 양식 준수가 필요합니다.

[task 3, 4 정답양식 및 평가방식 이해]

※ 작성해야 하는 답안 = 각 대화별로 제시된 3개의 코디 조합의 추천 순위 정하기

```

; 0
US   가을 축제 에 입고 갈 스타일 로 코디 해 주 세 요
CO   치마 바지 원피스 중 어떤 옷 이 포함_된 코디 를 추천_해 드릴_까 요
US   원피스 나 치마 로 추천_해 주 세 요
CO   가을 에 입 기 좋 은 적당_한 두께감 의 아우터 와 함께 추천_해 드릴_까 요
US   네
CO   상의 는 무늬 없 는 티 로 추천_해 드리 는 것 은 어떠 신가 요
US   좋 습니다
R1   JP-137 KN-008 SK-047 SE-042
R2   JP-137 KN-045 SK-047 SE-004
R3   JP-137 KN-045 SK-047 SE-052
    
```

← 입력데이터에 나열된 코디 조합의 순위를
순서대로 나열

- 각 대화에 제시된 코디의 순위를

1위 → 0

2위 → 1

3위 → 2로 인덱싱하고,

나열된 순위 순서 리스트를 랭크테이블(우측 이미지)

상의 인덱스로 대체하여 답안으로 기록

0	0	1	2
1	0	2	1
2	1	0	2
3	1	2	0
4	2	0	1
5	2	1	0

e.g. 문제 답안이 [1, 0, 2] 순서인 경우 → 2

[2, 1, 0] 순서인 경우 → 5

- 결과로 생성할 **prediction.csv**는 index, column명 없이 row 순서대로 랭크테이블 인덱스를 기록한 utf-8 인코딩의 텍스트 문서

※ numpy array 기준

`np.savetxt("/home/work/model/prediction.csv", "변수명", fmt="%d")`

명령으로 저장

※ **Weighted Kendall's Tau 계산**

- 두 순위 배열 사이의 Correlation을 -1 ~ 1 사이의 스코어로 산출
- WKT는 순위가 높을수록 큰 값이 들어오도록 되어있음 (인덱스와 반대)
- 따라서 점수를 계산할 때에는 0 → 2,
1 → 1,
2 → 0 으로 인덱스를 역으로 변환하여 계산함
- 실제 계산 함수에 대해서는 베이스라인에 포함된 score.py 참조
- **task3의 경우 6개 내부 태스크의 결과물을 순서대로 하나의 prediction.csv로 합하여 제출해야 합니다!**

[기타 주의사항]

- 사용자 세션과 마찬가지로 평가용 임시 세션은 GPU없이 CPU를 사용하여 추론을 수행하게 됩니다. 2CPU, 6GB 환경으로 4개 베이스라인은 최소3분 ~ 최대 7분 사이에 추론과 평가가 종료됩니다.
- 간혹 모델이 너무 크거나 데이터 배치를 너무 크게 잡으면 메모리오버로 세션이 죽는 문제가 발생할 수 있습니다. 특히 추론 시의 배치 사이즈는 베이스라인 추론 코드에 설정된 크기 이하로 정하는 것이 안전합니다.
 - task1 & 2 : 64 이하 권장
 - task 3 & 4 : 100 이하 권장
- 위와 같은 이유로 세션이 죽는 경우 서버가 추론이 끝나지 않은 것으로 판단하여 재제출이 안 될 수 있습니다. 해당되는 경우 문의하시면 조치해드립니다.
- 순위 상위를 기록하여 입상 대상이 되는 후보들의 경우
 - 코드분석
 - 재학습 및 재추론을 이용한 결과 동일성 검증을 거치게 됩니다.
 - 특히 task3의 경우 모델/소스코드가 연속학습의 조건을 충족하지 않으면 무효처리됩니다!
- 따라서 모델을 설계, 학습, 추론하는 과정에 결과 재현이 가능하도록 seed를 지정하여 무작위성을 최대한 통제해야 합니다.
- 다음은 1일 재제출 횟수 3회가 초과된 경우의 세션 출력 메세지입니다. 참고 바랍니다.

```
work@main1[czuPPodp-session]:~/model$ bash submit.sh
{
  "message": "Bad Request",
  "data": "{\"statusCode\":400,\"message\":\"You have exceeded the number of submissions in the specified period\", \"error\":\"Bad Request\"}"
}
```

- 또한 이전 제출결과의 추론이 아직 끝나지 않은 상태에서 재제출을 시도하는 경우 아래와 같은 메세지가 출력됩니다.

```
{
  "statusCode": 400,
  "message": "previous inference is not over yet",
  "error": "Bad Request"
}
```