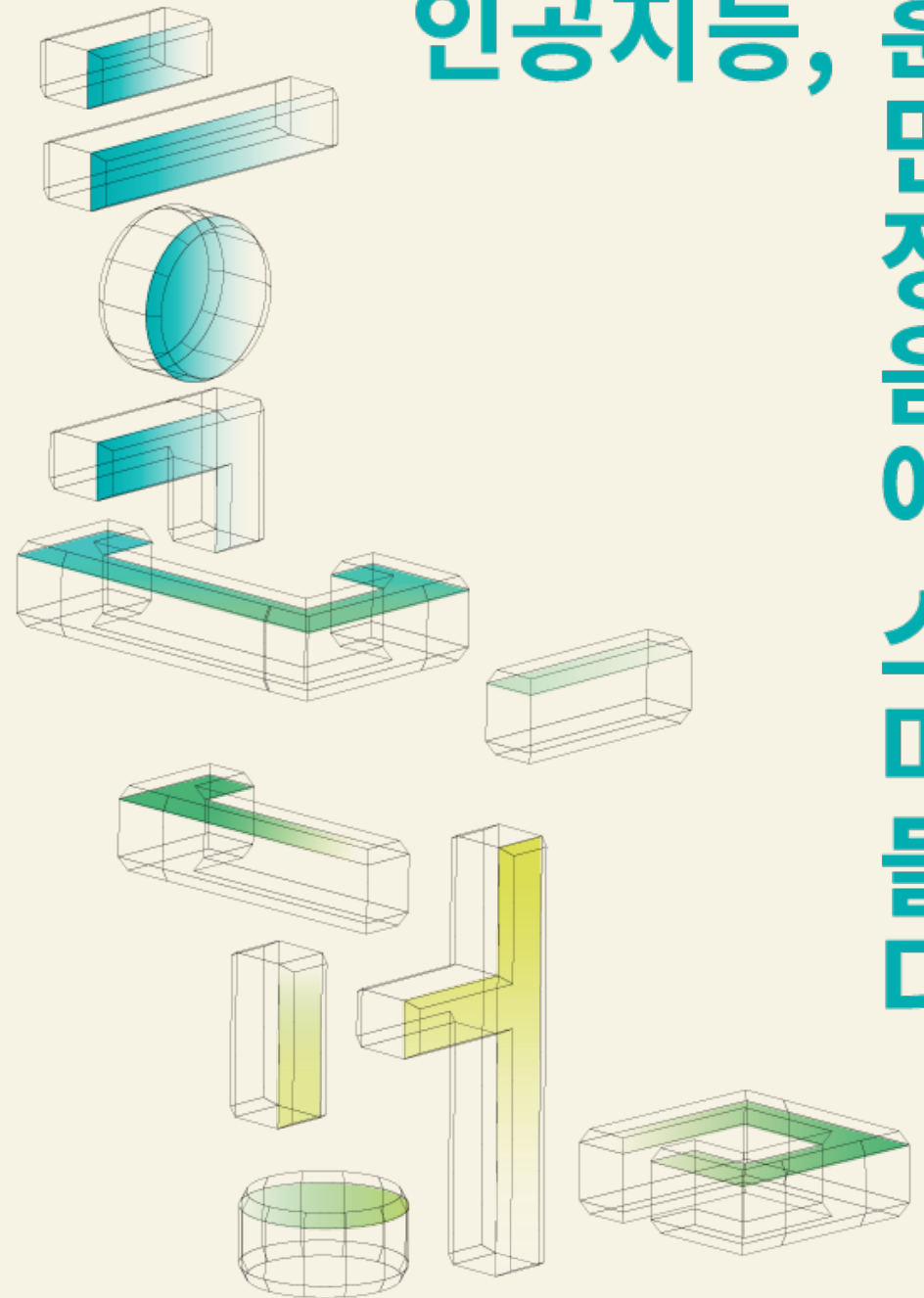


인공지능, 훈민정음에 스며들다

인공지능, 훈민정음에 스며들다

2021 한국어 음성·자연어 인공지능 경진대회

대회안내 및 오리엔테이션



대회 개요

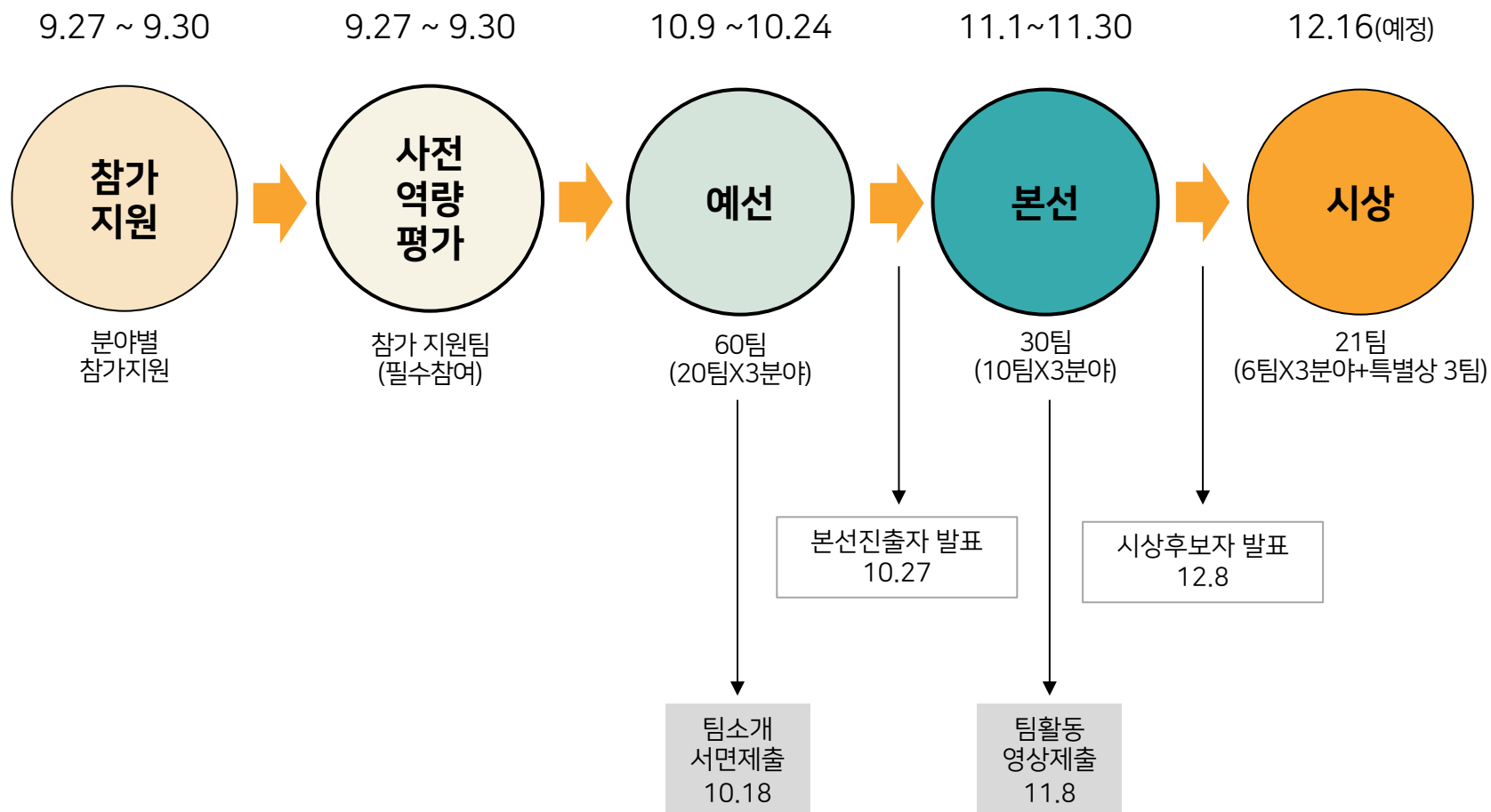
1. 대회 목적
2. 대회 일정
3. 참가 분야
4. 평가 방식
5. 상금/특전
6. 유의 사항

1. 대회 목적

- 2021 한국어 음성·자연어 인공지능 경진대회(부제: 인공지능, 훈민정음에 스며들다)는 AI-Hub에 구축된 한국어 음성·자연어 데이터셋을 활용하여 한국어 분야의 인공지능 기술고도화 및 서비스 발굴 등 민간과의 시너지 효과를 창출·확산하기 위해 기획되었습니다.
- 나아가, 본 대회를 통해 한국어 관련 인공지능 산업 인재 육성과 역량을 강화하고, 기술 아이디어 및 검증을 통해 산업 생태계를 활성화하고자 하는 목적을 추구하고자 합니다.

2. 대회 일정

- 진행 절차는 사전역량평가, 예선, 본선 3단계로 진행됩니다.



3. 참가 분야

• 분야별 설명

참가 분야는 음성인식, 대화요약, 화자인식의 3가지 부문입니다.

분야	음성인식	대화요약	화자인식
부제	한국어 음성 인식 성능을 향상하라	한국어 대화 요약 데이터를 활용하여 대화를 요약하라	회의 음성 데이터를 활용하여 화자를 인식하라
후원사	NIA 한국지능정보사회진흥원	NAVER	kt
데이터 및 문제	<ul style="list-style-type: none"> • 예선: 자유대화 3종, 명령어 3종 데이터를 활용하여한국어 음성 인식을 평가 • 본선: 자유대화 3종, 명령어 3종, 한국어 방언 발화 5종 데이터를 활용하여 한국어 음성 인식을 평가 	일상 대화, 토론 등 다양한 유형의 한국어 대화 원문 텍스트와 대응되는 한국어 대화 요약 데이터를 훈련하여 대화생성요약문을 작성하는 모델 개발	교육, 문화예술, 가족, 교양, 시사, 금융, IT 등 다양한 분야의 회의 음성 데이터와 대응되는 문자 및 발화자 정보가 결과로 제공되는 데이터를 토대로 화자를 인식하는 모델 개발
제공인프라	예선·본선 참가 1개 팀당 GPU(v100, 50GB) 2개 지원	예선·본선 참가 1개 팀당 GPU(v100, 50GB) 1개 지원	예선·본선 참가 1개 팀당 GPU(v100, 50GB) 1개 지원

4. 평가방식

• 평가기준 및 유의사항

평가방식은 음성인식, 대화요약, 화자인식 분야별로 아래와 같습니다.

분야	음성인식	대화요약	화자인식
평가기준	글자오류율-CER(% 제1지표) 단어오류율-WER(% 제2지표)	ROUGE-L 점수(제1지표) ROUGE-2 점수(제2지표) ROUGE-1 점수(제3지표)	EER(% Equal Error Rate)
외부자원	외부데이터/Pre-trained모델 허용	외부데이터/Pre-trained모델 불허	외부데이터/Pre-trained모델 불허
GPU제한	150GB 이내	100GB 이내	100GB 이내
사용언어	Python		
제출방식	<ul style="list-style-type: none"> 리더보드상 제출횟수는 1시간에 1회로 제한 1회제출시 인퍼런싱은 1시간 이내 모든 팀원이 동시접속 가능하나 1회 1명만 제출가능 		
리더보드	데이터셋당 1개 총 3개 리더보드 최 종순위는 3개 점수의 총합	리더보드 1개 운영	리더보드 1개 운영
부정행위	<ul style="list-style-type: none"> 대회기간 동안 전자적·물리적, 혹은 그 외의 방법으로 다른 참가팀의 도전문제 해결 시도 및 그 해결 결과를 방해한 경우 정상적인 대회 진행방법 외에 문제 데이터셋을 전자적·물리적으로 확보하여 도전문제를 해결하였거나, 시도한 경우 대회에서 허용되지 않은 방법으로 참가팀 간에 결탁하였거나, 결탁을 시도한 경우 대회에서 허용되지 않은 방법으로 여러 팀에 참여한 경우 비정상적인 방법으로 데이터셋 등 정보 탈취, 평가 플랫폼의 공격 또는 오류 발생 시도 등의 행위를 한 경우 그 외 대회 운영에 고의로 심각한 피해를 유발하였다고 판단되는 경우 		

5. 상금/특전

- **상금:** 총 21팀에 총 1.2억원 상금 시상

분야		훈격		규모(팀당)
음성인식 성능평가	NIA	최우수상(1팀)	과기정통부장관상	1,500만원
		우수상(2팀)	NIA 원장상	1,000만원
		장려상(3팀)	NIA 원장상	500만원
대화요약	네이버	최우수상(1팀)	네이버 대표상	1,000만원
		우수상(2팀)	NIA 원장상	700만원
		장려상(3팀)	NIA 원장상	300만원
화자인식	KT	최우수상(1팀)	KT융합기술원장상	1,000만원
		우수상(2팀)	NIA 원장상	700만원
		장려상(3팀)	NIA 원장상	300만원
특별상(중복수상가능)		품질검증상(3팀)	NIA 원장상	100만원

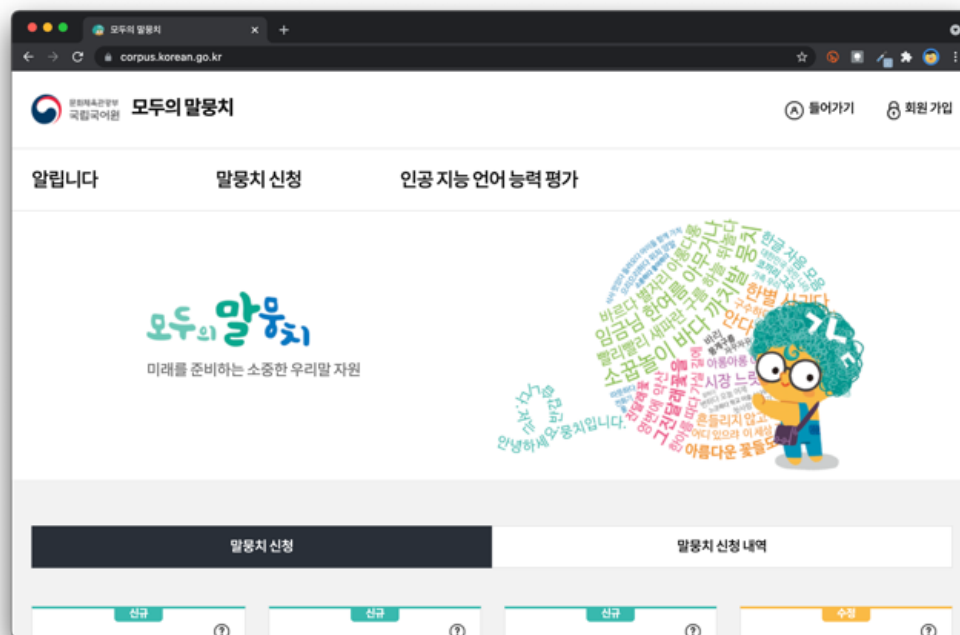
- **대회특전:**
 - 본선 진출팀에게 소정의 참가지원금(20만원)
 - 네이버 채용 시 서류전형 면제, KT 채용 시 서류전형 가산점 제공
 - 업스테이지 채용 시 서류전형 면제

6. 유의 사항

- 음성인식 분야의 경우 외부 데이터 및 Pre-trained 모델을 사용할 수 있으나, 이는 저작권에 문제가 없어야 하며 주최측은 이에 대해 어떠한 책임도 지지 않습니다.

❖ 모두의 말뭉치 사용 관련

- 데이터 신청부터 승인까지 1~2 일 정도가 소요될 수 있음
- 문체부 ‘모두의 말뭉치’ 중 금번 경진대회에 사용되는 일상대화 음성 데이터의 경우, 개인식별, 음성합성 및 변환에 사용될 수 없음(세부내용은 모두의말뭉치 사이트 참조)
- <https://corpus.korean.go.kr/>

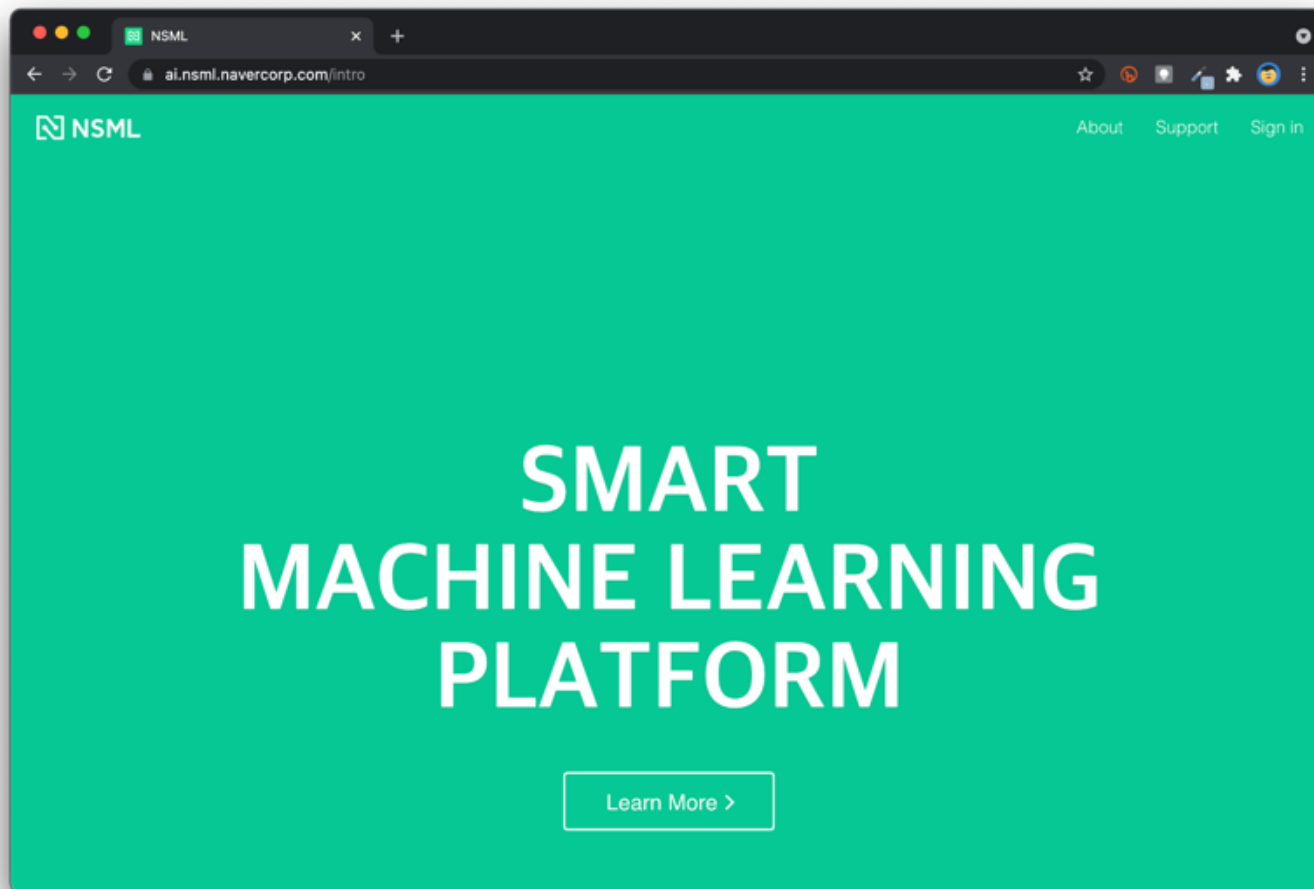


대회 플랫폼

1. ML플랫폼
2. NSML Documents
3. NSML 설치
4. Login
5. NSML 대시보드
6. Baseline 코드
7. Baseline 구성
8. Datasets
9. 실행
10. 실행 검토
11. 제출 및 확인
12. 운영 기술 문의

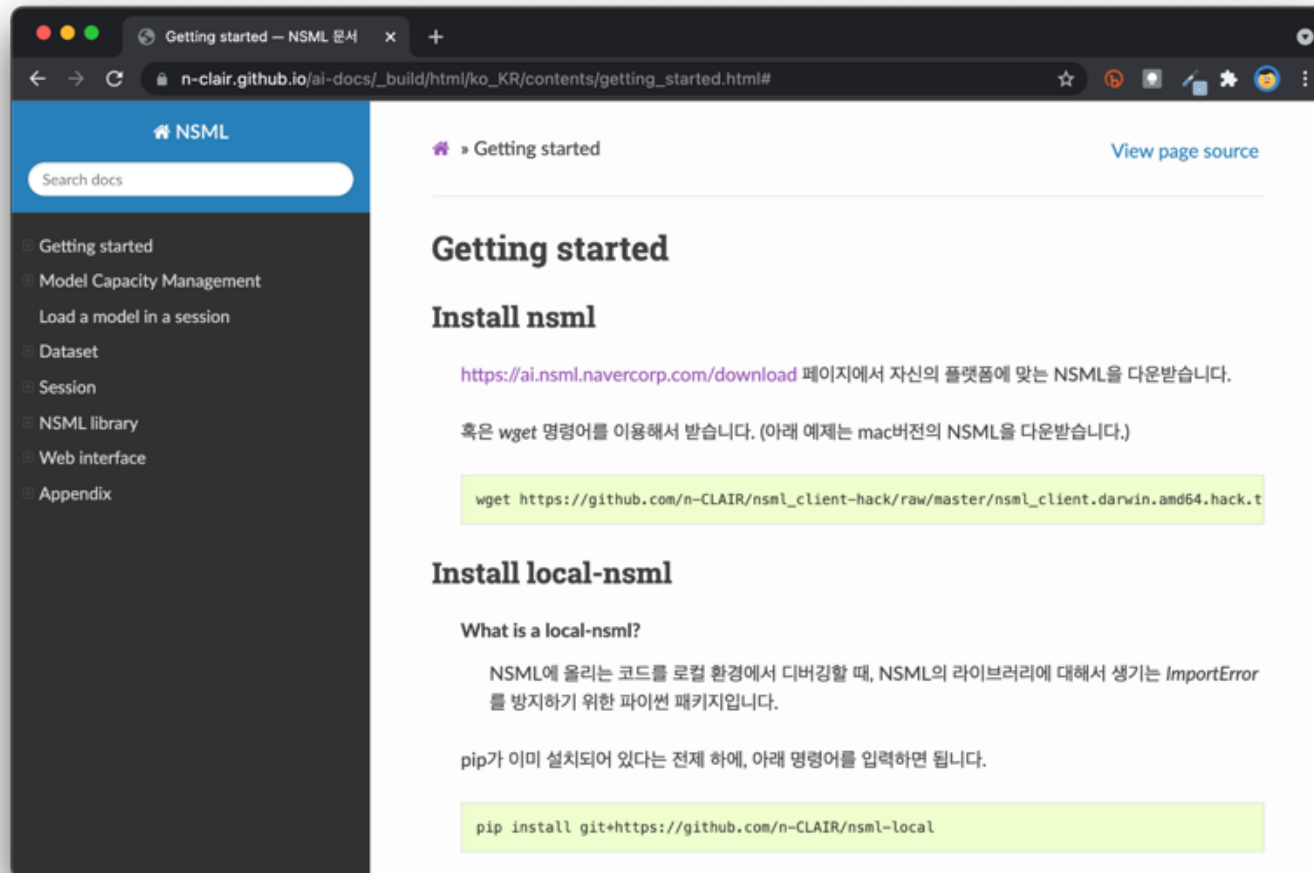
1. ML플랫폼

- ML플랫폼: 네이버클라우드의 NSML
- URL: <https://ai.nsml.navercorp.com/intro>
- Username과 Password는 github계정과 동일하게 사용 (참가신청시 제출한 github profile 등록됨)



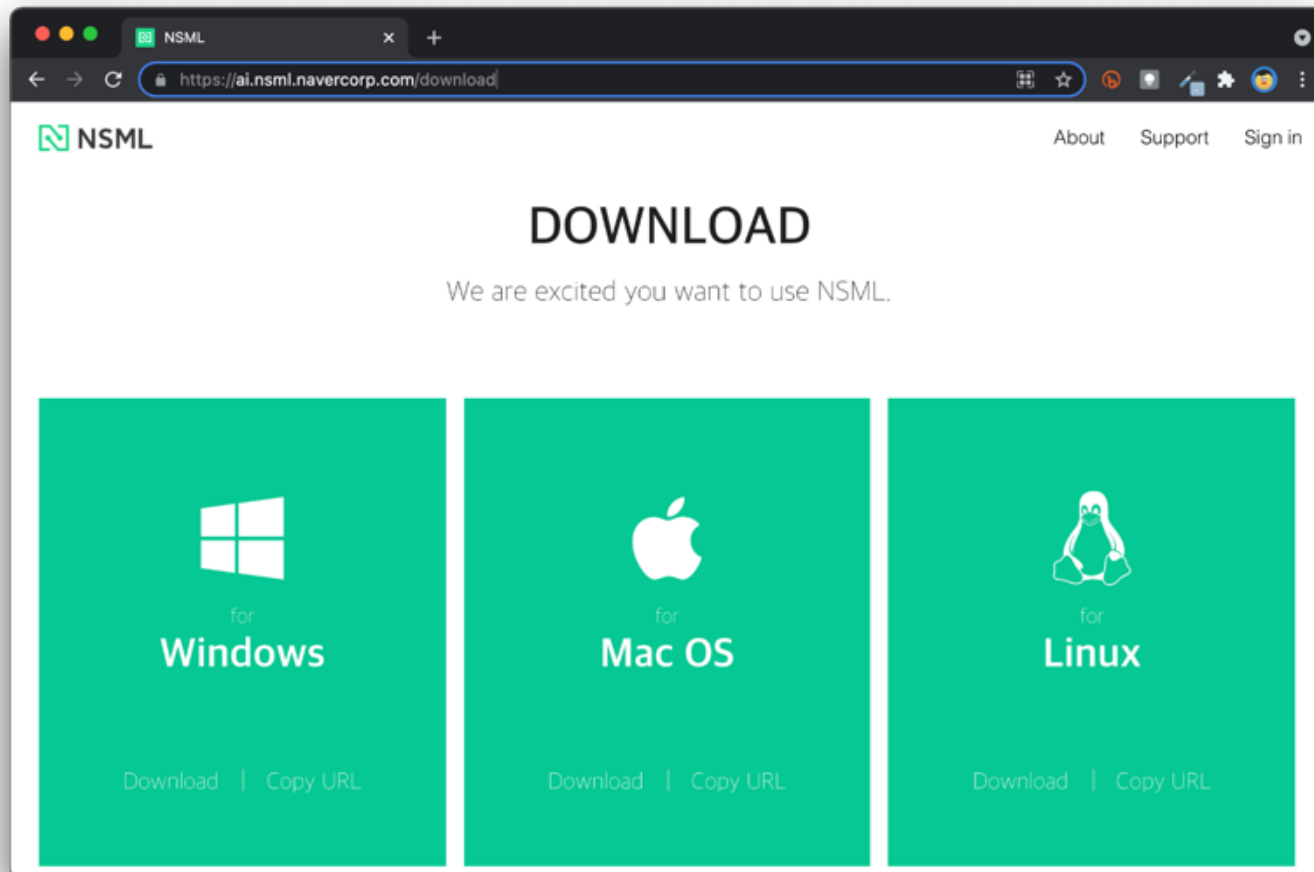
2. NSML Documents

- NSML CLI commands 또는 NSML python 함수 관련 내용 최우선적으로 NSML document를 참조
- URL: https://n-clair.github.io/ai-docs/_build/html/ko_KR/index.html



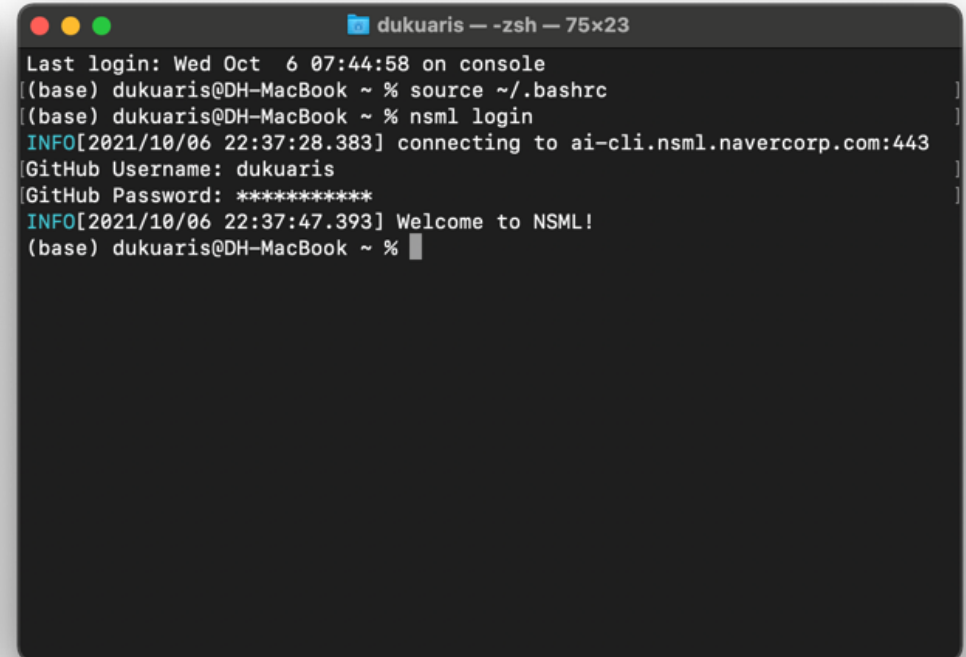
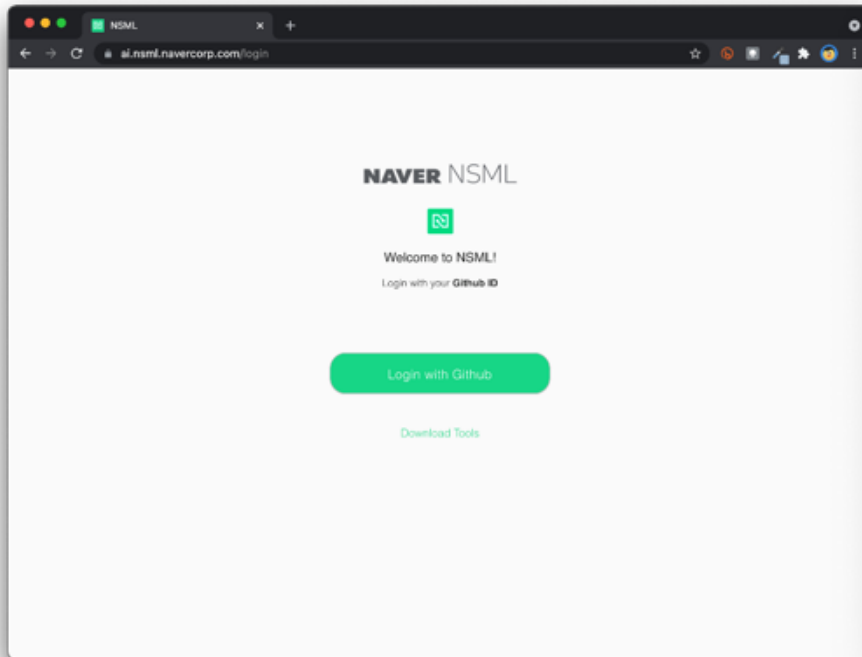
3. NSML 설치

- NSML 구동(CLI 기반)을 위해서는 아래 웹사이트에서 자신의 OS에 맞는 설치파일을 다운받고 설치
- URL: <https://ai.nsml.navercorp.com/download>



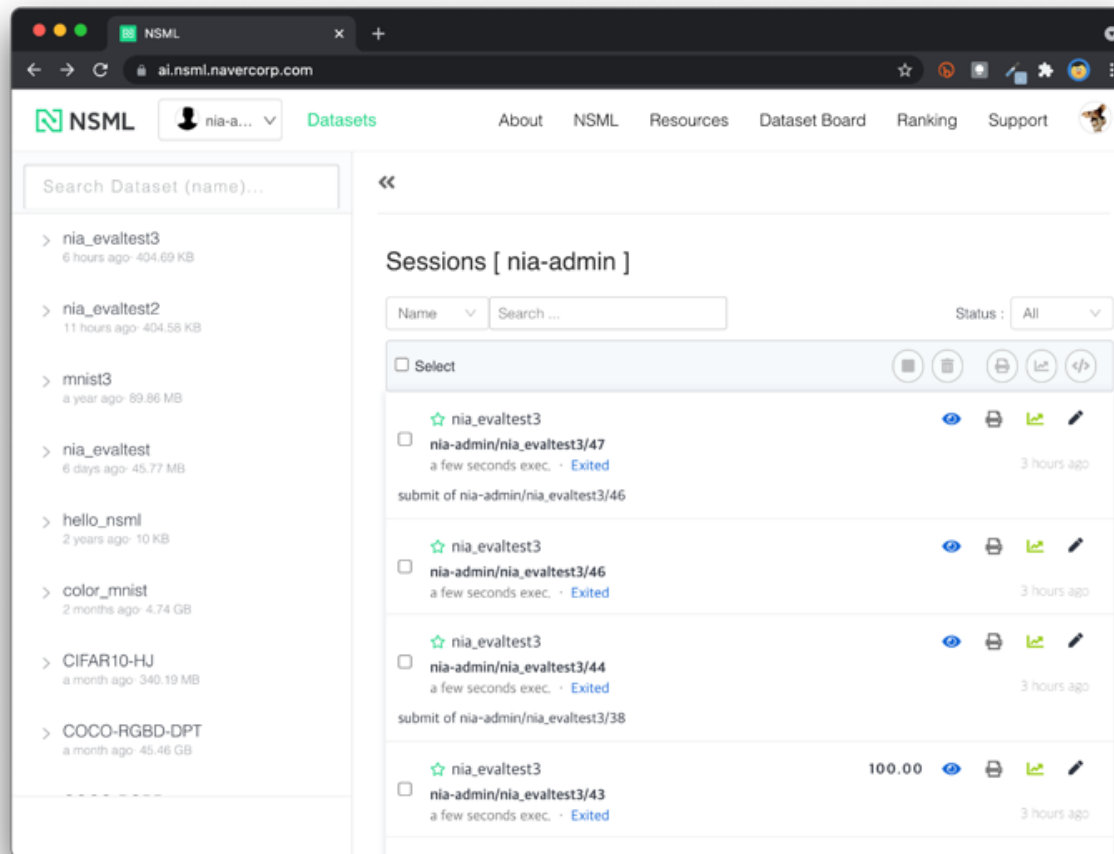
4. Login

- 웹사이트 로그인
 - NSML 홈페이지에서 로그인
 - 홈페이지는 데이터셋, GPU자원 현황, 리더보드 등을 GUI환경에서 확인 검토용으로 사용
- 터미널 로그인
 - 명령어: `nsml login`
 - NSML 명령어 구동을 위해서는 path 설정 필요
 - 대부분의 실행은 터미널에서 운영



5. NSML 대시보드: 첫화면

- 데이터셋: 좌측 목록으로 대회 데이터셋을 확인 참가자에게는 해당하는 데이터만 보임
(데이터셋이 하나인 대회는 하나, 데이터셋이 둘인 경우 둘만 보임)
- Sessions: 우측 목록으로 실행기록을 차수별로 보여줌



5. NSML 대시보드: 데이터셋별 화면

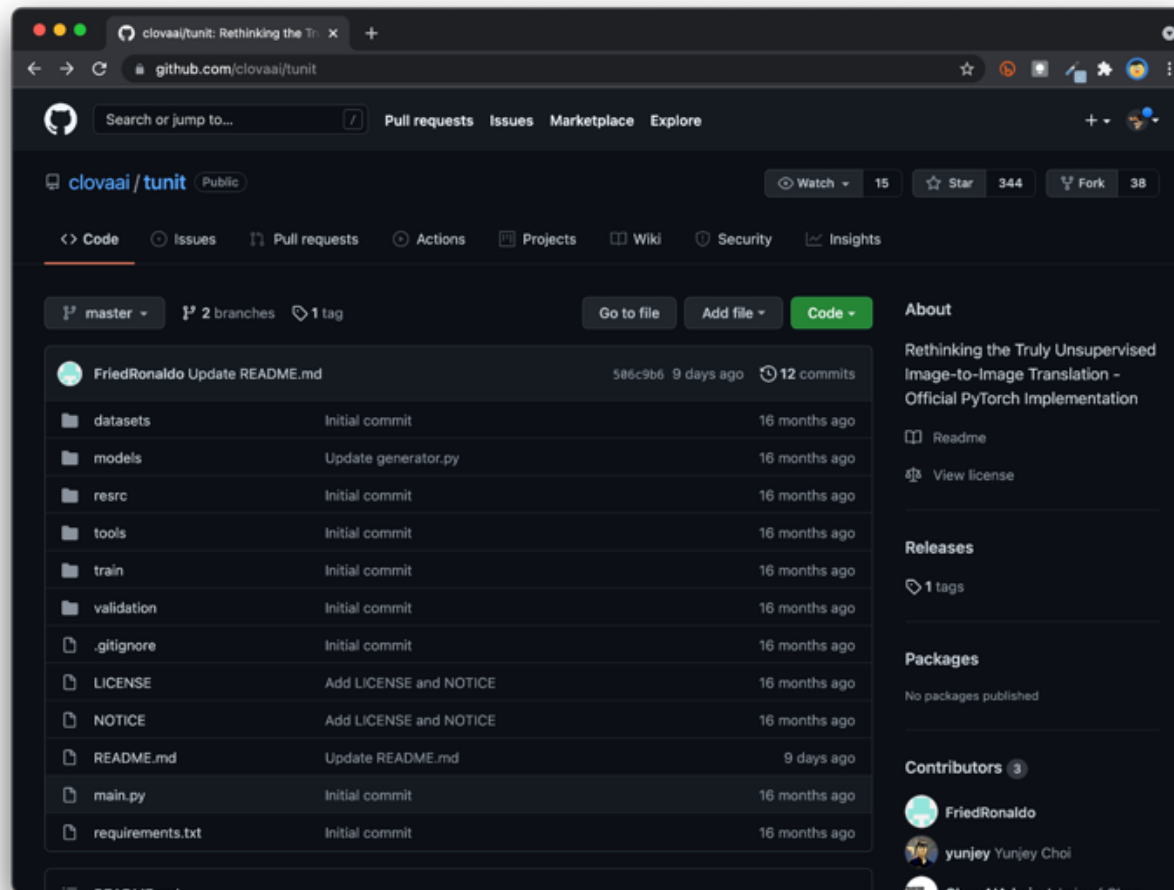
- 데이터셋 이름을 클릭하면 해당되는 리더보드와 파일시스템(데이터 목록) 탭이 나타남
- 리더보드에서 제출에 대한 평가가 점수와 순위로 표시됨
- 데이터셋당 하나의 리더보드만 표시

The screenshot shows the NSML dashboard interface. On the left, a list of datasets is displayed, with 'mnist3' selected and highlighted in green. The main area shows the 'mnist3' leaderboard. The leaderboard is titled 'mnist3' and has tabs for 'Leaderboard' (selected) and 'File System'. Below the title, there is a dropdown menu for 'custom - descending' and a 'Public' filter. The leaderboard table lists the top two entries:

Rank	Name	Score	Model	Count	Last Submit	Recorded
1	nia-admin	0.9867	nia-admin/mnist3/1/9	1	19 days ago	19 days ago
2	nsmteam	0.9845	nsmteam/mnist3/7/8	2	a year ago	a year ago

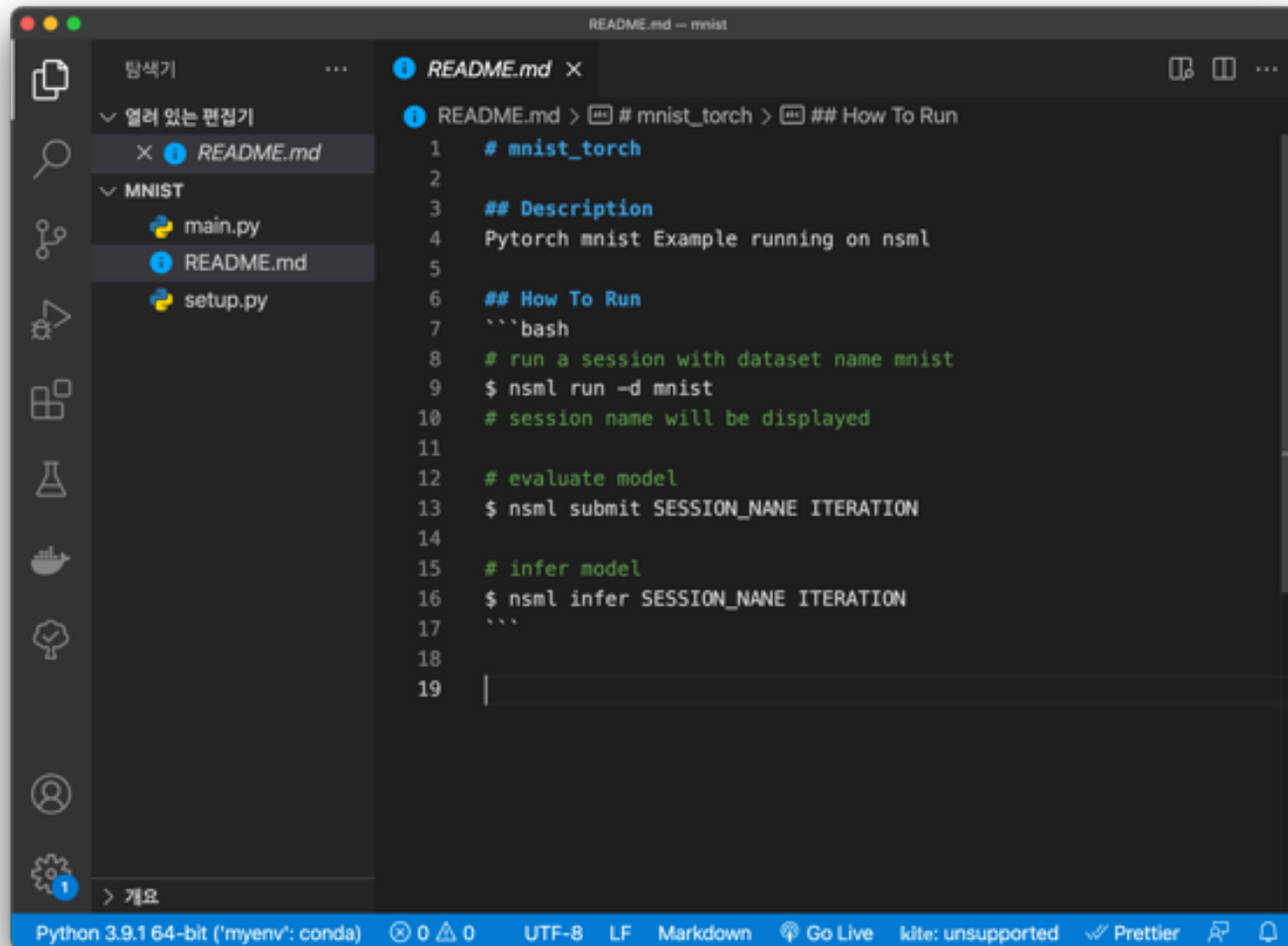
6. Baseline 코드: download

- 제공되는 github repository에서 다운로드
- 예시: `git clone https://github.com/clovaai/tunit.git`



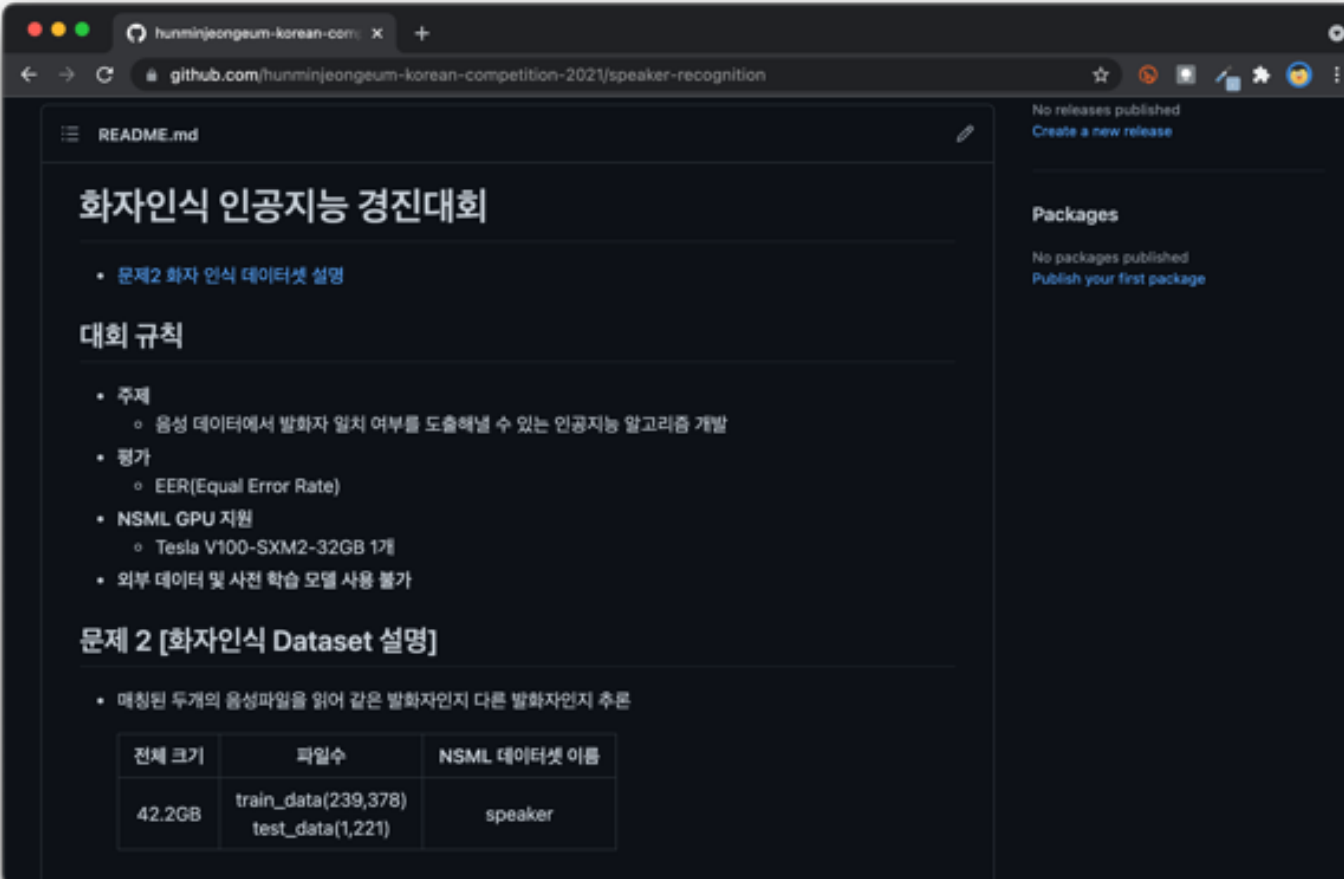
6. Baseline 코드: 실행 준비

- 터미널에서 baseline코드가 다운로드된 local 디렉토리로 이동
- 주요파일구성확인: README.md, main.py, setup.py



7. Baseline 구성: README.md

- 문제 개요: 개요, 규칙 등 설명
- NSML 에서 실행방법 설명



The screenshot shows a GitHub repository page for 'speaker-recognition' by 'hunminjeongeum-korean-competition-2021'. The README.md file is displayed in a dark theme. It contains the following sections:

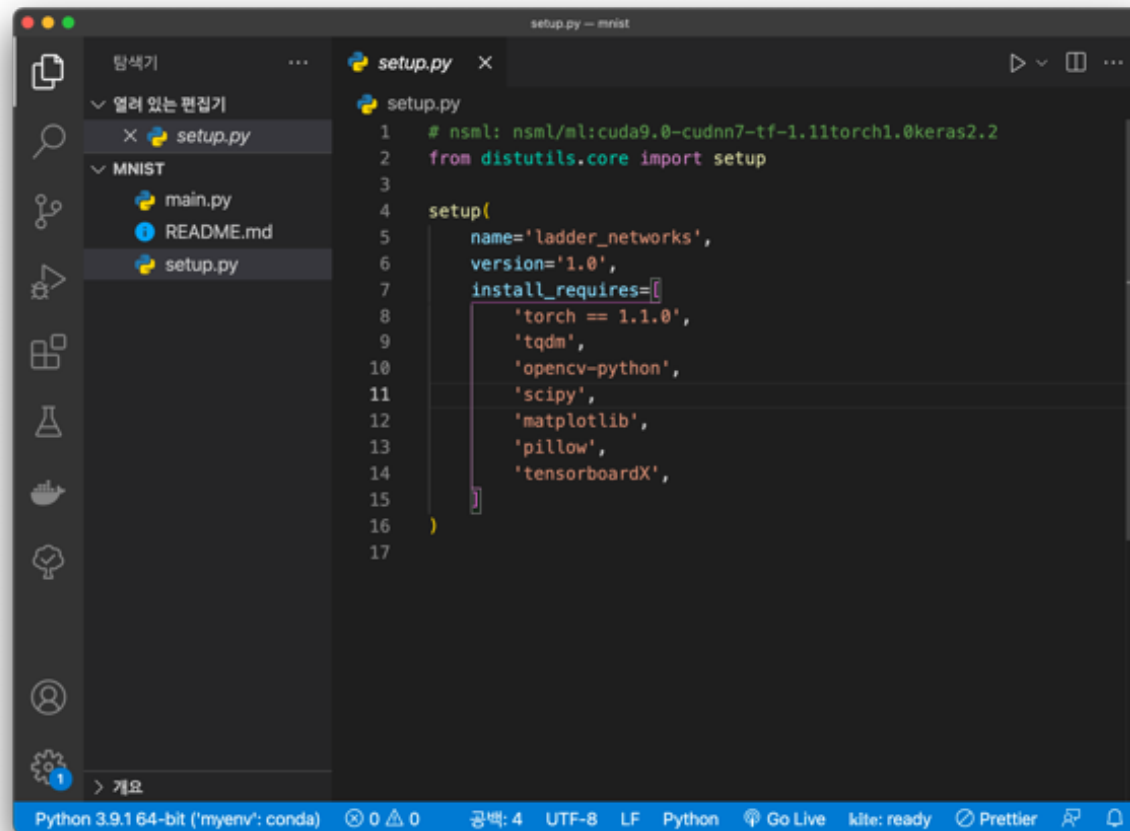
- 화자인식 인공지능 경진대회**
 - 문제2 화자 인식 데이터셋 설명
- 대회 규칙**
 - 주제
 - 음성 데이터에서 발화자 일치 여부를 도출해낼 수 있는 인공지능 알고리즘 개발
 - 평가
 - EER(Equal Error Rate)
 - NSML GPU 지원
 - Tesla V100-SXM2-32GB 1개
 - 외부 데이터 및 사전 학습 모델 사용 불가
- 문제 2 [화자인식 Dataset 설명]**
 - 매칭된 두개의 음성파일을 읽어 같은 발화자인지 다른 발화자인지 추론

전체 크기	파일수	NSML 데이터셋 이름
42.2GB	train_data(239,378) test_data(1,221)	speaker

7. Baseline 구성: setup.py

- setup.py에 pip install이 가능한 dependencies를 install_requires 에 목록으로 입력
- 특히, Pre-trained model이나 외부데이터를 사용해야할 경우 Docker Image 제작 후 Docker Hub에 업로드 후 setup.py 최상단에 Docker Image 경로를 포함

예) #nsm1: nsm1/ml:cuda9.0-cudnn7-tf-1.11torch1.0keras2.2



7. Baseline 구성: entry file

- 핵심 실행 파일로서 통상 main.py로 명명
- NSML의 함수를 활용하여 전처리-훈련-추론 전과정을 포함

```

1  """ 02: MNIST PyTorch
2  =====
3
4
5  Implementation of a CNN on PyTorch using the MNIST dataset.
6  |
7  """
8
9  import argparse
10 import os
11 import torch
12 import torch.nn as nn
13 import torch.nn.functional as F
14 import torch.optim as optim
15 from PIL import Image
16 from torch.autograd import Variable
17 from torch.utils.data import DataLoader
18 from torchvision import transforms
19 from torchvision.datasets.mnist import MNIST, read_image_file, read_label_file
20
21 import nsml
22 from nsml import HAS_DATASET, DATASET_PATH
23
24
  
```

```

24
25 def bind_model(model, optimizer):
26     def save(dir_path):
27         state = {
28             'model': model.state_dict(),
29             'optimizer': optimizer.state_dict()
30         }
31         torch.save(state, os.path.join(dir_path, 'model.pt'))
32
33     def load(dir_path):
34         state = torch.load(os.path.join(dir_path, 'model.pt'))
35         model.load_state_dict(state['model'])
36         if 'optimizer' in state and optimizer:
37             optimizer.load_state_dict(state['optimizer'])
38             print('optimizer loaded!')
39         print('model loaded!')
40
41     def infer(input_data, top_k):
42         model.eval()
43         # from list to tensor
44         image = torch.stack(preprocess(None, input_data))
45         image = Variable(image.cuda())
46         clean_state, _ = model(image)
47         batch_size, all_cls = clean_state.size()
  
```

8. Dataset

- 데이터셋에 대한 세부내용은 github repository의 README.md 문서에서 확인
- 데이터셋은 NSML 플랫폼에 업로드되며 entry file(main.py)내에서 NSML 함수를 통해 load

The screenshot shows a web browser displaying the README.md file for the 'dialog-summarization' repository on GitHub. The page title is 'Train Dataset'. It lists two bullet points: the first points to the data path 'root_path/train/train_data/' with examples (Edu, Food, Life, Per_Rel, Work); the second states that 'train_label' is not present and 'summary' from 'train_data' is used as the label. Below this, a JSON structure for 'train_data' is shown with various fields like 'numberOfItems', 'data', 'header', 'dialogueInfo', 'participantsInfo', 'age', 'residentialProvince', 'gender', 'participantID', 'body', 'dialogue', 'utterance', 'utteranceID', 'participantID', 'date', 'turnID', and 'size'.

```

train_data (json): 학습용 데이터셋
├── numberOfItems (ex. 71408)
├── data
│   ├── header
│   │   ├── dialogueInfo
│   │   │   ├── numberOfParticipants (ex. 2)
│   │   │   ├── numberOfUtterances (ex. 16)
│   │   │   ├── numberOfTurns (ex. 6)
│   │   │   ├── type (ex. 일상 대화)
│   │   │   ├── topic (ex. 개인 및 관계)
│   │   │   ├── dialogueID (ex. b6d7d466-25f2-5b3e-a55d-fe5ce1e32687)
│   │   │   ├── participantsInfo (List)
│   │   │   │   ├── age (ex. 20대)
│   │   │   │   ├── residentialProvince (ex. 부산광역시)
│   │   │   │   ├── gender (ex. 여성)
│   │   │   │   └── participantID (P01)
│   │   └── ...
│   └── body
│       ├── dialogue (List)
│       │   ├── utterance (ex. 양치하고 물먹)
│       │   ├── utteranceID (ex. U1)
│       │   ├── participantID (ex. P01)
│       │   ├── date (ex. 2019-11-08)
│       │   ├── turnID (ex. T1)
│       │   └── size (ex. 13+33+00)

```

9. 실행

• 실행

- 명령어: `nsml run -e main.py -d mnist3`
- -e: entry file
- -d datasets

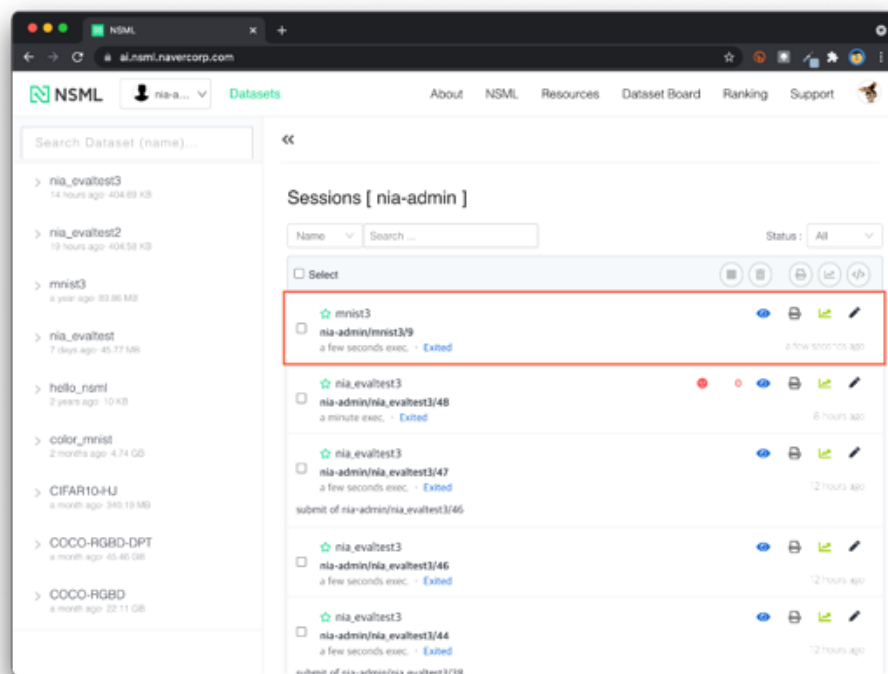
• NSML대시보드

- 터미널에서 NSML 명령어 수행 후 NSML 대시보드 상에서 실행(Session)상황을 파악
- Session 목록을 클릭하여 세부내용 탐색

```

mnist - zsh - 83x29
Last login: Thu Oct  7 05:36:37 on console
(base) dukuaris@DH-MacBook ~ % source ~/.bashrc
(base) dukuaris@DH-MacBook ~ % cd nsml/mnist
(base) dukuaris@DH-MacBook mnist % ls
README.md      main.py      requirements.txt
(base) dukuaris@DH-MacBook mnist % nsml run -e main.py -d mnist3
INFO[2021/10/07 07:23:23.412] .nsmlignore check - start
INFO[2021/10/07 07:23:23.414] .nsmlignore check - done
INFO[2021/10/07 07:23:23.522] file integrity check - start
INFO[2021/10/07 07:23:23.523] file integrity check - done
INFO[2021/10/07 07:23:23.529] README.md 296 B - start
INFO[2021/10/07 07:23:23.530] README.md 296 B - done (1/3 33.33%) (296 B/10 KiB 2.8
1%)
INFO[2021/10/07 07:23:23.530] main.py 9.9 KiB - start
INFO[2021/10/07 07:23:23.531] main.py 9.9 KiB - done (2/3 66.67%) (10 KiB/10 KiB 98
.84%)
INFO[2021/10/07 07:23:23.531] requirements.txt 122 B - start
INFO[2021/10/07 07:23:23.531] requirements.txt 122 B - done (3/3 100.00%) (10 KiB/1
0 KiB 100.00%)
.....
Building docker image. It may take a while
.....
Session nia-admin/mnist3/9 is started
(base) dukuaris@DH-MacBook mnist %

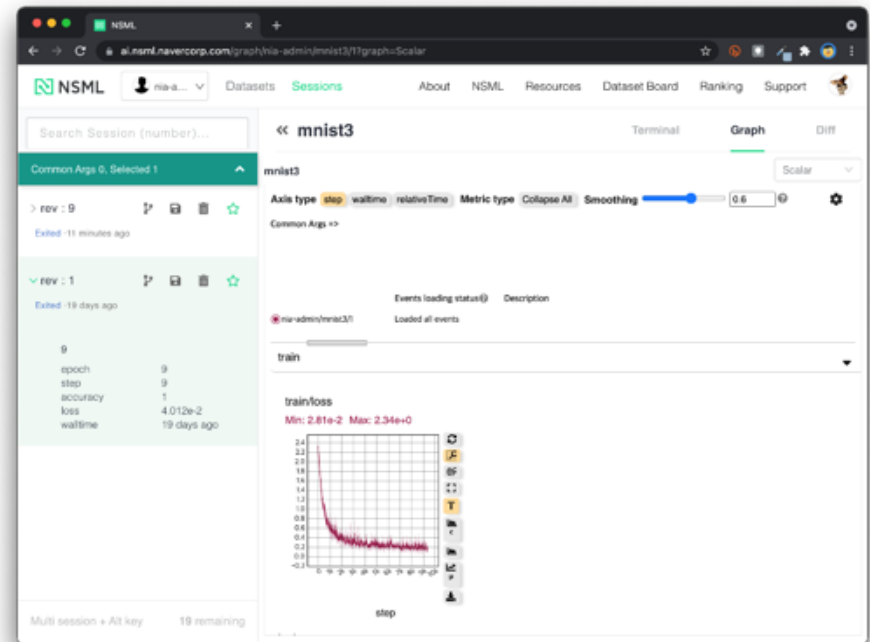
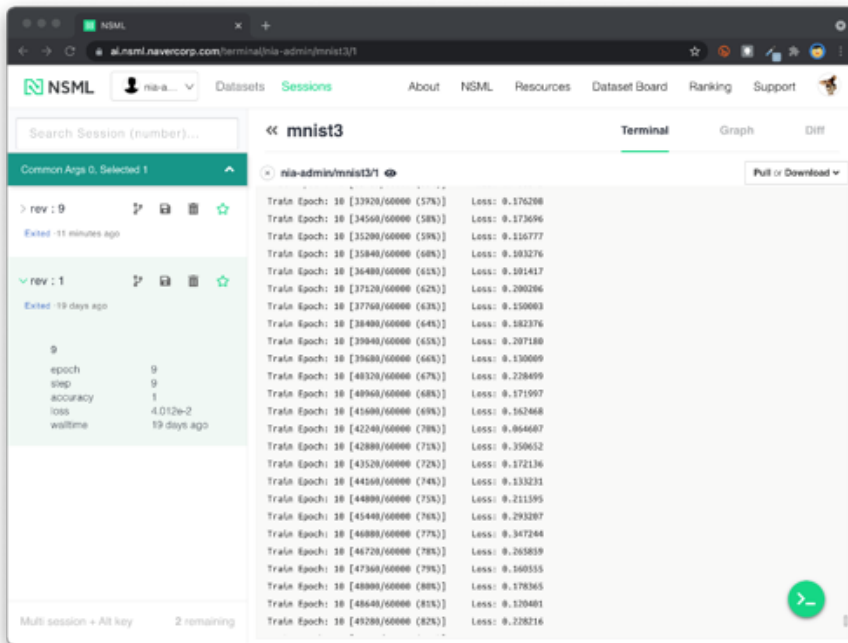
```



10. 실행 검토

- 실행경과/결과 검토
 - NSML웹사이트 로그인
 - 대시보드 Sessions 창에서 실행목록을 클릭 이동
 - Terminal 탭 화면

- 그래프 검토
 - Sessions 화면에서 Graph탭 클릭



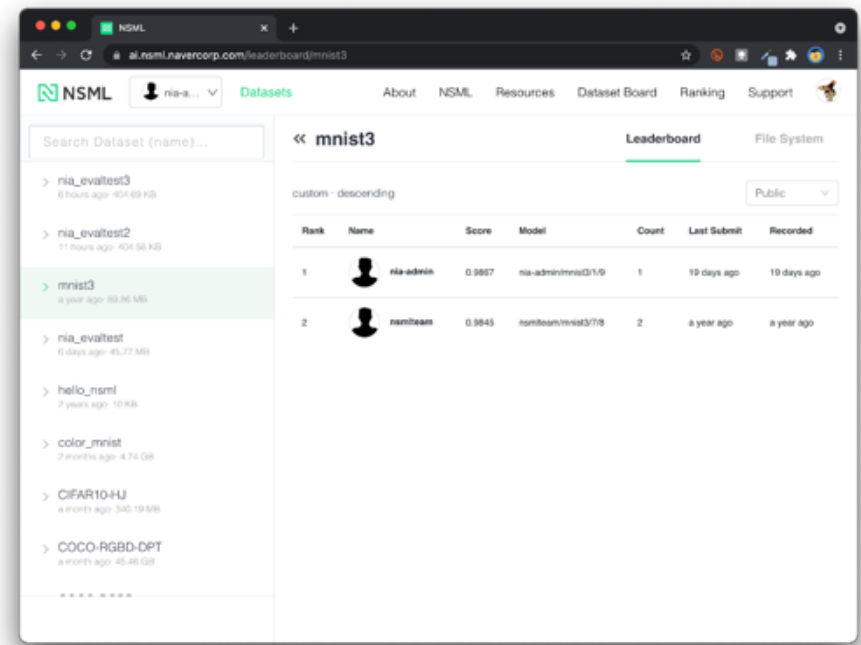
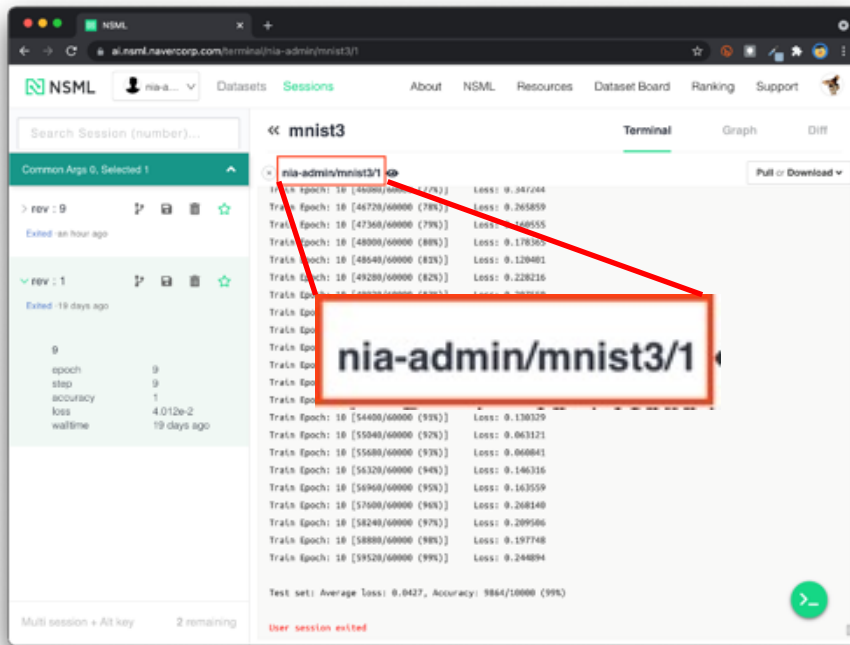
11. 제출 및 확인

• 제출

- 명령어: `$ nsml submit Options SESSION_NAME CHECKPOINT`
- 명령어 예시: `$ nsml submit -t nia-admin/mnist3/1`

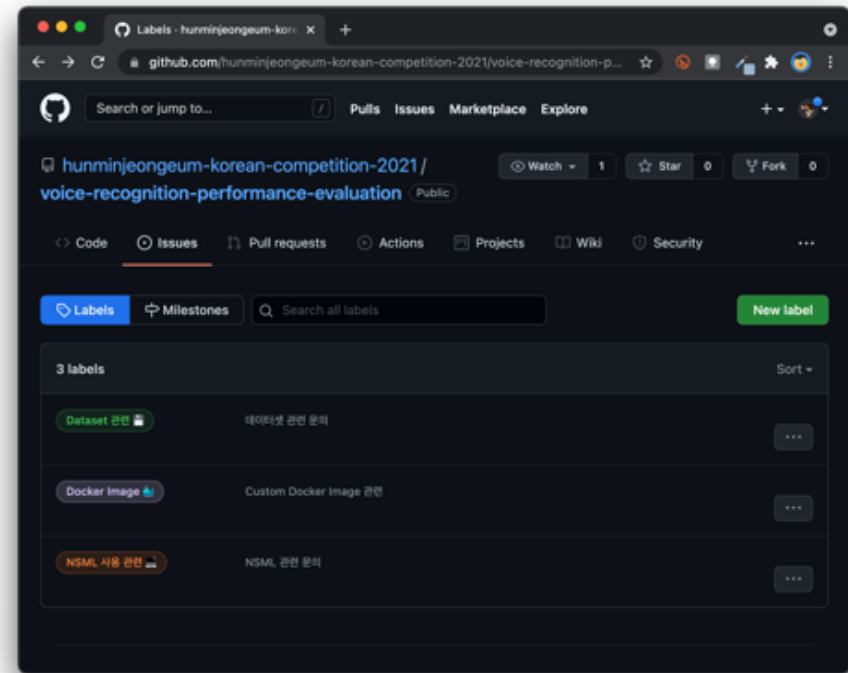
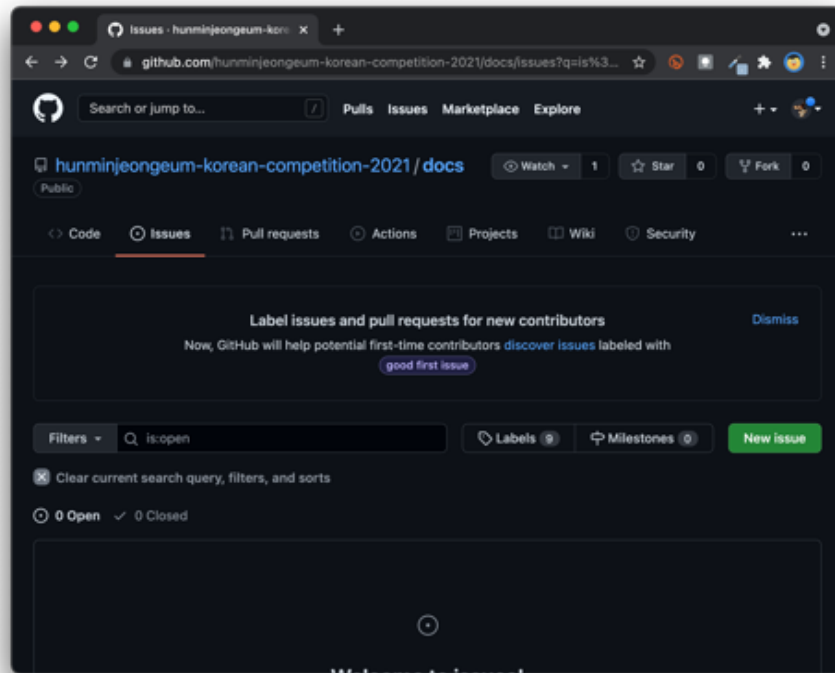
• 리더보드 확인

- 제출 후 리더보드에서 결과 확인



12. 운영 기술 문의

- 대회 중 문의사항은 전문적이고 빠른 지원을 위해 주제별 repository의 issues탭 이용
- github 주소: <https://github.com/hunminjeongeum-korean-competition-2021>

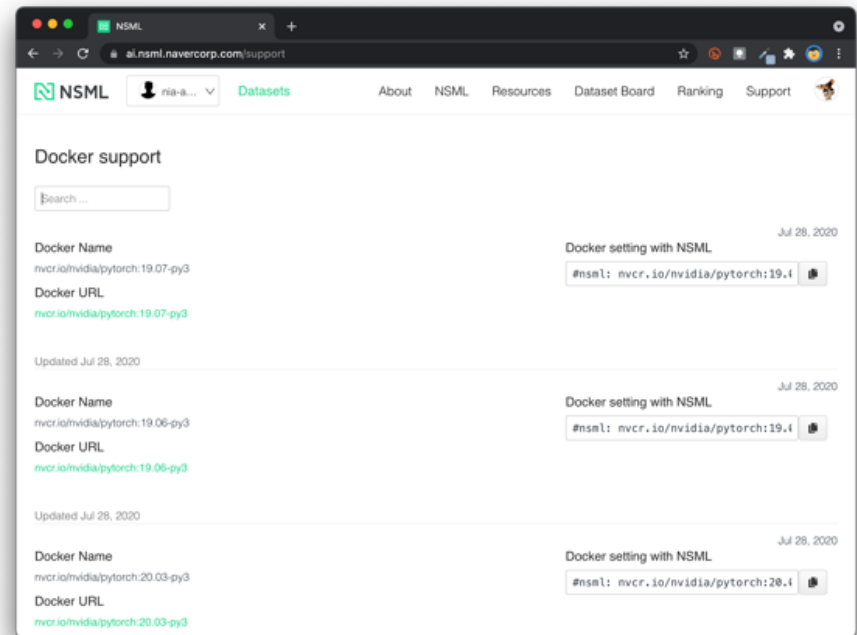
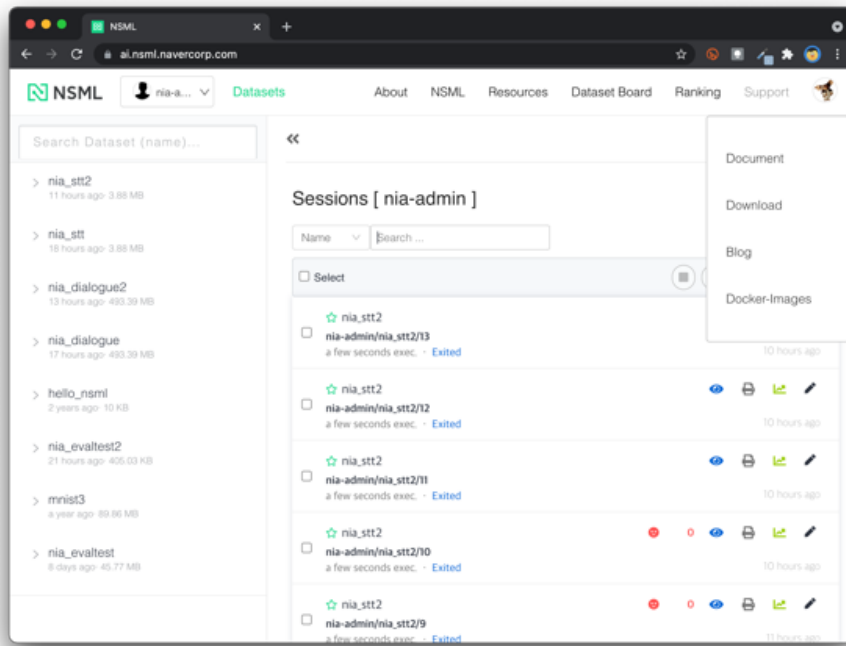


Docker for NSML

1. 기본 도커
2. 외부 라이브러리 사용
3. Dockerfile 구성
4. Build and Push

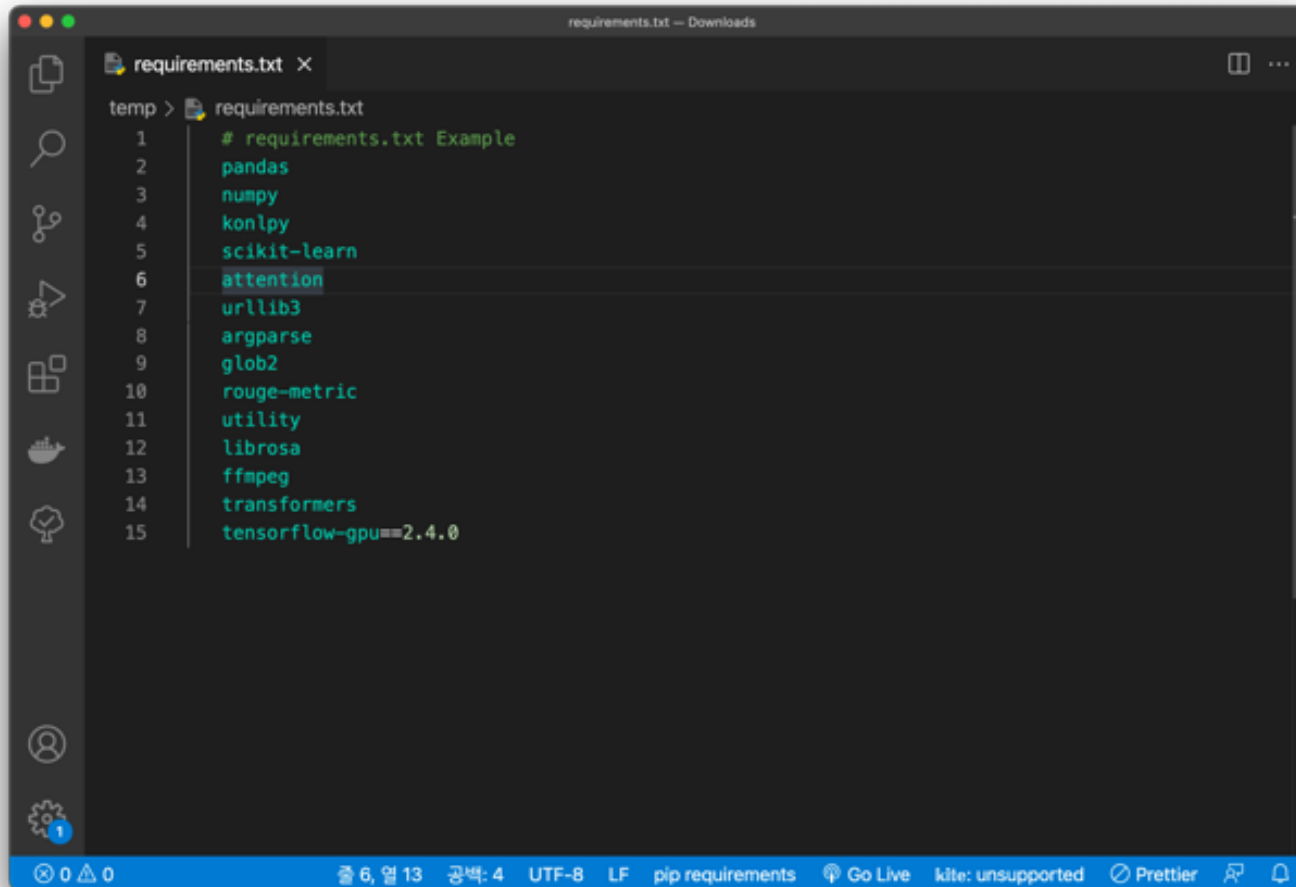
1. 기본 도커

- 로그인 후 첫화면에서 우상단 Support 탭에 마우스오버하고 드롭다운되는 메뉴에서 Docker-Images 선택
- NSML에서는 현재(2021년 10월) 기본 10개의 도커를 제공
- Docker list URL: <https://hub.docker.com/u/nsml>



2. 외부 라이브러리 사용

- NSML의 기본제공 Docker 이외의 `python library`를 사용할 경우, 사용할 `library`를 `requirements.txt`파일에 명시하여 Docker Image를 구축해 사용
- 외부 Pre-trained 모델 사용시 동 파일에 `kobert-transformers`(예시) 같이 명기



```
requirements.txt
temp > requirements.txt
1 # requirements.txt Example
2 pandas
3 numpy
4 konlpy
5 scikit-learn
6 attention
7 urllib3
8 argparse
9 glob2
10 rouge-metric
11 utility
12 librosa
13 ffmpeg
14 transformers
15 tensorflow-gpu==2.4.0
```

3. Dockerfile 구성

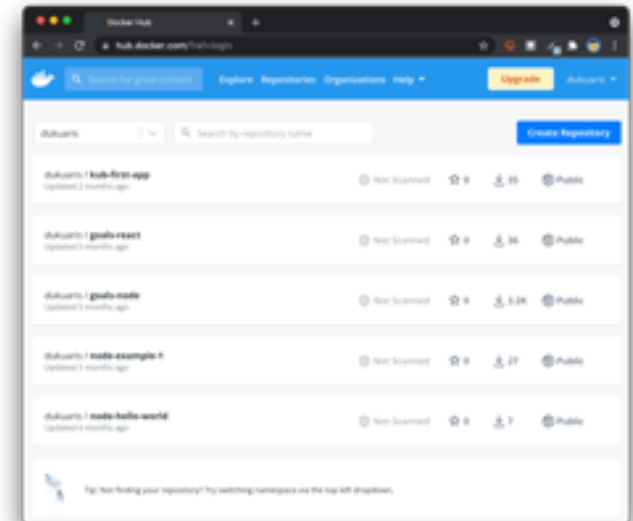
- Dockerfile(확장자 없음)은 Docker image를 구축하는 청사진 역할
- 구축시 `Dockerfile`과 `requirements.txt` 파일은 같은 디렉토리 안에 있어야 함

```
Dockerfile -- my-docker
Dockerfile > ...
1  # MANITANER Dacon_Dev_Team <dacon@dacon.io>
2  # Docker Base Image
3  FROM nsm1/default_ml:cuda9
4
5  # python lib
6  COPY requirements.txt ./
7
8  # pip3 install and apt-get update
9  RUN apt-get update && apt-get install -y vim libbz2-dev python3-pip
10
11 # install python3.6.3 version
12 RUN wget https://www.python.org/ftp/python/3.6.3/Python-3.6.3.tgz
13 RUN tar xvfz Python-3.6.3.tgz
14 RUN cd Python-3.6.3 && ./configure && make && make install
15
16 # pip update
17 RUN pip install --upgrade pip
18
19 # install Python Packages
20 RUN pip install -r requirements.txt
```

Python 3.9.1 64-bit ('myenv': conda) 0 0 0 공백: 4 UTF-8 LF Dockerfile Go Live k1te: unsupported Prettier

4. Build and Push

- Push 전에 dockerhub(<https://hub.docker.com/>)에 가입
- 터미널에서 Dockerfile이 있는 디렉토리로 이동 후 아래 명령어를 수행
 - Login to Dockerhub(docker image를 push하기 위함)
`$ docker login`
 - Build docker image build
`$ docker build -t daconDevTeam/nsml-nia-competiton .`
 - Push docker image to dockerhub
`$ docker push daconDevTeam/nsml-nia-competiton`



NSML 문서 : https://n-clair.github.io/ai-docs/_build/html/ko_KR/index.html

운영기술문의 : <https://github.com/hunminjeongeum-korean-competition-2021>