

Homework #2

Sept. 2023

<http://link.koreatech.ac.kr>

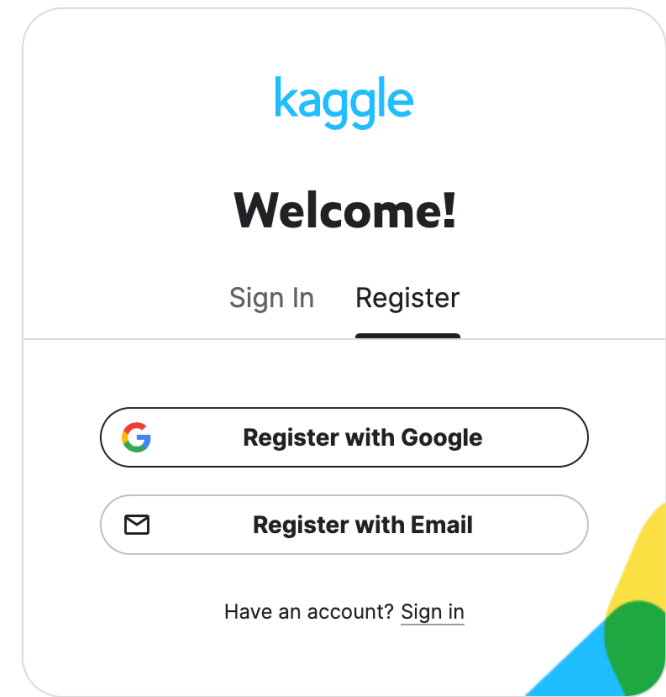
HW2. Kaggle Competition

◆ Kaggle 회원 가입

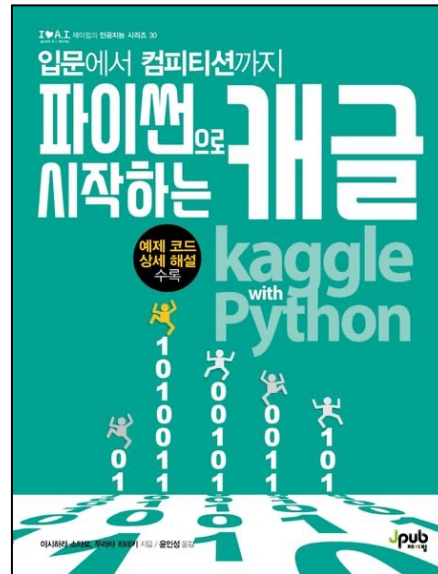
- <https://www.kaggle.com/>
- 가급적 구글 계정으로 회원 가입 추천

◆ Kaggle 둘러보기/알아보기

- <https://musma.github.io/2019/03/04/about-kaggle.html>



◆ 추천서



HW2. Kaggle Competition

◆ Titanic - Machine Learning from Disaster @ Kaggle

- <https://www.kaggle.com/competitions/titanic>
- This is the legendary Titanic ML competition: the best, first challenge for you to dive into ML competitions and familiarize yourself with how the Kaggle platform works.
- History
 - The sinking of the Titanic is one of the most infamous shipwrecks in history.
 - On April 15, 1912, during her maiden voyage, the widely considered "unsinkable" RMS Titanic sank after colliding with an iceberg.
 - Unfortunately, there weren't enough lifeboats for everyone onboard, resulting in the death of 1502 out of 2224 passengers and crew.
 - While there was some element of luck involved in surviving, it seems some groups of people were more likely to survive than others.
 - In this challenge, we ask you to build a predictive model that answers the question:
 - "what sorts of people were more likely to survive?" using passenger data (i.e. age, gender, socio-economic class, etc
- 한글 설명 (반드시 읽어보기)
 - <https://developers.ascentnet.co.jp/2017/11/24/kaggle-process-review/>

HW2. Kaggle Competition

◆ Titanic - Machine Learning from Disaster @ Kaggle

— Target Feature

- Survived: 0 = 사망, 1 = 생존

— Feature

- PassengerId: 승객 번호
- Name: 이름
- Pclass: 티켓 클래스
 - 1 = 1등석, 2 = 2등석, 3 = 3등석
- Sex: 성별
 - male = 남성, female = 여성
- Age: 나이
- SibSp: 동승한 자매 / 배우자의 수
- Parch: 동승한 부모 / 자식의 수
- Ticket: 티켓 번호
- Fare: 승객 요금
- Cabin: 방 호수
- Embarked: 탑승지
 - C = 세르부르, Q = 퀸즈타운, S = 사우샘프턴

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	0	3	Braund, Mr. Owen Harris	male	22	1	0	A/5 21171	7.25		S
2	1	1	Cumings, Mrs. John Bradley (Florence	female	38	1	0	PC 17599	71.2833	C85	C
3	1	3	Heikinen, Miss. Laina	female	26	0	0	STON/O2. 3101282	7.925		S
4	1	1	Futrelle, Mrs. Jacques Heath (Lily May	female	35	1	0	113803	53.1	C123	S
5	0	3	Allen, Mr. William Henry	male	35	0	0	373450	8.05		S
6	0	3	Moran, Mr. James	male		0	0	330877	8.4583		Q
7	0	1	McCarthy, Mr. Timothy J	male	54	0	0	17463	51.8625	E46	S
8	0	3	Palsson, Master. Gosta Leonard	male	2	3	1	349909	21.075		S
9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhe	female	27	0	2	347742	11.1333		S
10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14	1	0	237736	30.0708		C
11	1	3	Sandstrom, Miss. Marguerite Rut	female	4	1	1	PP 9549	16.7	G6	S
12	1	1	Bonnell, Miss. Elizabeth	female	58	0	0	113783	26.55	C103	S
13	0	3	Saunderscock, Mr. William Henry	male	20	0	0	A/5. 2151	8.05		S
14	0	3	Andersson, Mr. Anders Johan	male	39	1	5	347082	31.275		S
15	0	3	Vestrom, Miss. Hulda Amanda Adolfina	female	14	0	0	350406	7.8542		S
16	1	2	Hewlett, Mrs. (Mary D Kingcome)	female	55	0	0	248706	16		S
17	0	3	Rice, Master. Eugene	male	2	4	1	382652	29.125		Q
18	1	2	Williams, Mr. Charles Eugene	male		0	0	244373	13		S
19	0	3	Vander Planke, Mrs. Julius (Emelia Mari	female	31	1	0	345763	18		S
20	1	3	Masselmani, Mrs. Fatima	female		0	0	2649	7.225		C
21	0	2	Fynney, Mr. Joseph J	male	35	0	0	239865	26		S

HW2. Kaggle Competition

◇ [요구사항 1] titanic_dataset.py 분석 (1/6)

```
import os, torch
import pandas as pd
from torch.utils.data import Dataset, DataLoader, random_split

class TitanicDataset(Dataset):
    def __init__(self, X, y):
        self.X = torch.FloatTensor(X)
        self.y = torch.LongTensor(y)

    def __len__(self):
        return len(self.X)

    def __getitem__(self, idx):
        feature = self.X[idx]
        target = self.y[idx]
        return {'input': feature, 'target': target}

    def __str__(self):
        str = "Data Size: {0}, Input Shape: {1}, Target Shape: {2}".format(
            len(self.X), self.X.shape, self.y.shape
        )
        return str
```

HW2. Kaggle Competition

◇ [요구사항 1] titanic_dataset.py 분석 (2/6)

```
class TitanicTestDataset(Dataset):
    def __init__(self, X):
        self.X = torch.FloatTensor(X)

    def __len__(self):
        return len(self.X)

    def __getitem__(self, idx):
        feature = self.X[idx]
        return {'input': feature}

    def __str__(self):
        str = "Data Size: {0}, Input Shape: {1}".format(
            len(self.X), self.X.shape
        )
        return str
```

HW2. Kaggle Competition

◆ [요구사항 1] titanic_dataset.py 분석 (3/6)

```
def get_preprocessed_dataset():  
    CURRENT_FILE_PATH = os.path.dirname(os.path.abspath(__file__))  
  
    train_data_path = os.path.join(CURRENT_FILE_PATH, "train.csv")  
    test_data_path = os.path.join(CURRENT_FILE_PATH, "test.csv")  
  
    train_df = pd.read_csv(train_data_path)  
    test_df = pd.read_csv(test_data_path)  
  
    all_df = pd.concat([train_df, test_df], sort=False)  
    all_df = get_preprocessed_dataset_1(all_df)  
    all_df = get_preprocessed_dataset_2(all_df)  
    all_df = get_preprocessed_dataset_3(all_df)  
    all_df = get_preprocessed_dataset_4(all_df)  
    all_df = get_preprocessed_dataset_5(all_df)  
    all_df = get_preprocessed_dataset_6(all_df)  
    ...
```

HW2. Kaggle Competition

◇ [요구사항 1] titanic_dataset.py 분석 (4/6)

```
from torch import nn, optim
class MyModel(nn.Module):
    def __init__(self, n_input, n_output):
        super().__init__()

        self.model = nn.Sequential(
            nn.Linear(n_input, 30),
            nn.ReLU(),
            nn.Linear(30, 30),
            nn.ReLU(),
            nn.Linear(30, n_output),
        )

    def forward(self, x):
        x = self.model(x)
        return x
```


HW2. Kaggle Competition

◆ [요구사항 1] titanic_dataset.py 분석 (5/6)

```
if __name__ == "__main__":
    train_dataset, validation_dataset, test_dataset = get_preprocessed_dataset()

    print("train_dataset: {0}, validation_dataset.shape: {1}, test_dataset: {2}".format(
        len(train_dataset), len(validation_dataset), len(test_dataset)
    ))
    print("#" * 50, 1)

    for idx, sample in enumerate(train_dataset):
        print("{0} - {1}: {2}".format(idx, sample['input'], sample['target']))

    print("#" * 50, 2)

    train_data_loader = DataLoader(dataset=train_dataset, batch_size=16, shuffle=True)
    validation_data_loader = DataLoader(dataset=validation_dataset, batch_size=16, shuffle=True)
    test_data_loader = DataLoader(dataset=test_dataset, batch_size=len(test_dataset))
```

HW2. Kaggle Competition

◇ [요구사항 1] titanic_dataset.py 분석 (6/6)

```
if __name__ == "__main__":
    ...

    print("[TRAIN]")
    for idx, batch in enumerate(train_data_loader):
        print("{0} - {1}: {2}".format(idx, batch['input'].shape, batch['target'].shape))

    print("[VALIDATION]")
    for idx, batch in enumerate(validation_data_loader):
        print("{0} - {1}: {2}".format(idx, batch['input'].shape, batch['target'].shape))

    print("#" * 50, 3)

    print("[TEST]")
    batch = next(iter(test_data_loader))
    print("{0}".format(batch['input'].shape))
    my_model = MyModel(n_input=11, n_output=2)
    output_batch = my_model(batch['input'])
    prediction_batch = torch.argmax(output_batch, dim=1)
    for idx, prediction in enumerate(prediction_batch, start=892):
        print(idx, prediction.item())
```

HW2. Kaggle Competition

◆[요구사항 1] titanic_dataset.py 분석 리포트 작성법

– titanic_dataset.py에 작성된 코드를 적절한 조각으로 나누어 jupyter notebook에 셀로 넣고 실행

– 코드 조각 (셀)

- class TitanicDataset(Dataset)
- class TitanicTestDataset(Dataset)
- def get_preprocessed_dataset()
- def get_preprocessed_dataset_1(all_df)
- ...
- def get_preprocessed_dataset_6(all_df)
- class MyModel(nn.Module)
- def test(test_data_loader)
- if __name__ == "__main__":

– 각 코드 조각 별로 코드 분석 후 주석을 충분히 작성하기

HW2. Kaggle Competition

◆[요구사항 2] titanic 딥러닝 모델 훈련 코드 및 Activation Function 변경해보기

- `_01_code/_05_fcn_and_training/f_my_model_training_with_argparse_wandb.py` 코드를 그대로 활용하되 titanic 데이터에 맞게 수정하여 코딩하기
- Wandb로 훈련 과정 데이터 올려 그래프 얻어 내기
 - Training loss
 - Validation loss
 - 위 두 그래프를 보여주는 Wandb URL 얻어내기
- 모델 구성 내에 Activation Function를 변경하여 더 나은 성능을 산출하는 Activation Function 이 있는지 조사하기
 - ReLU
 - ELU
 - Leaky ReLU
 - PReLU
 - ...

HW2. Kaggle Competition

◆[요구사항 3] 테스트 및 `submission.csv` 생성

- 요구사항 2에서 살펴본 가장 좋은 성능을 보이는 `Activation Function`으로 모델 구성하기
- 훈련과정 중 어느 `Epoch` 시점에 테스트를 수행하여 `submission.csv` 를 구성해야 하는지 고찰하기
 - 테스트 데이터 (즉, `test_data_loader`) 활용 필요
- 고찰한 내용에 대한 추가 코딩 수행
- `submission.csv` 생성하기

HW2. Kaggle Competition

◆ [요구사항 4] submission.csv 제출 및 등수확인


- Kaggle에 로그인 후 "Submit Prediction" 기능을 통한 submission.csv 제출
- LeaderBoard에 등록된 나의 점수 및 위치 스크린 캡처하여 Jupyter Notebook에 넣기
- 캡처 이미지를 클라우드에 업로드하여 해당 그림의 URL 생성필요

✕

Submit to Competition


File Upload

Notebook



Titanic - Machine Learning from Disaster

You have 10 submissions remaining today. This resets in 17 hours.



Drag and drop file to upload


(e.g., .csv, .zip, .gz, .7z)

or

Browse Files

10981


Youn-HeeHan



0.76555

1

42s



Your Best Entry!

Your submission scored 0.76555, which is not an improvement of your previous score. Keep trying!

HW2. Kaggle Competition

◆ [요구사항 4] submission.csv 제출 및 등수확인

9. IMAGE

마크다운에 이미지를 삽입하는 방법은 다음과 같다.

기본 문법은 `![이미지 이름](이미지 주소)` 이다.

이미지 이름은 만약, 이미지의 주소가 변한다거나 등의 이유로 문제가 생겼을 때 뜨는 이름이다.

이미지 주소는 웹상의 주소를 넣어도 되고, local에 저장되어있는 이미지라면, 그 path를 넣어도 된다.

```
![sk_logo](https://upload.wikimedia.org/wikipedia/commons/b/b4/SK_logo.svg)
```



HW2. Kaggle Competition

◇[요구사항 5] Wandb 페이지 생성 및 URL 제출

- x축 Epoch에 대하여, y축에 Training loss 변화를 보여주는 그래프 제시
- x축 Epoch에 대하여, y축에 Validation loss 변화를 보여주는 그래프 제시
- 위 두 그래프를 포함하고 있는 Wandb URL 제출
- *주의*: Overview 메뉴 페이지

Privacy **PRIVATE**

Last active 2023. 10. 3. 오후 1:23:55

Author **K** link-koreatech

Contributors 1 user

Total runs 2

Total compute 40 minutes



Project Access

Privacy settings affect your whole project, including runs, reports, artifacts, etc.

☐ Private **DEFAULT**
Only you can view and contribute

☒ Public
Anyone can view

+ Show advanced settings

Create a team to collaborate on private projects.

Create Team

Cancel Save

What problem does this project tackle? ✎

Privacy **PUBLIC**

Last active 2023. 10. 3. 오후 1:23:55

Author **K** link-koreatech

Contributors 1 user

Total runs 2

Total compute 40 minutes

요구사항

◆ 보고서(Report) 내용에 대한 요구사항

- 프로그램 코드가 실행되는 것을 확인할 수 있도록 `jupyter notebook` 내 셀별로 출력 값들을 정확하게 나타낼 것
 - 모든 코드에 대한 출력이 잘 나와 있는지 확인함
- 핵심 코드라고 생각되는 것들에 대해서 주석(한글 또는 영문)을 넣을 것
 - 다다익선
- 코딩을 통하여 자신이 취득한 기술적 사항/고찰 내용을 생각한 바를 제시할 것
 - 다다익선
- [NOTE] 숙제 마지막에는 "숙제 후기" 라는 섹션 제목으로 본 숙제를 한 이후의 느낀점, 하고싶은 말, 또는 불평 등을 반드시 넣을 것
 - "숙제 후기" 섹션이 없으면 감점 처리

요구사항

◆ 제출형태 및 방법

- Jupyter Notebook 파일명: [hw2.ipynb](#)
 - 각 4가지 요구사항에 대해 섹션 제목을 정확히 넣고, 섹션이 잘 구분되도록 작성하기
- 숙제 제출 방식
 - <http://el2.koreatech.ac.kr> 의 "딥러닝및실습" 교과목 "과제" 메뉴
 - 게시물 본문에는 앞 페이지에서 설명한 <https://nbviewer.org/> 로 시작하는 URL을 넣기
 - 게시물 본문에 Wandb URL 넣기
 - 총 2개 URL 제시
- 숙제에 관한 질의/응답
 - <http://el2.koreatech.ac.kr> 의 "딥러닝및실습" 교과목 "Q&A" 게시판 활용

◆ 제출기한

- 2023년 10월 6일 (금) 23시 59분
- 지각 제출은 받지 않습니다 (0점 처리)