

Homework I

- ▶ 제출기한 : 4월 22일 23:59
- ▶ 지각시 1시간마다 1점 감점
- ▶ 각 문제에 대한 답을 작성한 파이썬 파일 “HW1_학번.ipynb”를 LMS 과제함에 제출
- ▶ 한 셀에 한 문제에 해당하는 코드를 작성
- ▶ 각 셀마다 주석으로 문제번호 작성
- ▶ 첫 번째 셀 처음부분에는 주석으로 학번/이름 작성
- ▶ for문과 같은 명령어를 사용하는 것이 아닌 수업시간에 배운 함수 및 연산식을 사용하여 코드를 작성
- ▶ 상수형 숫자 사용시 감점(예. 평균 계산시 90/40)
- ▶ 데이터 파일의 경로는 “C:/ml_workspace”하위에 있어야함(경로 틀리면 감점)

- 모든 문제는 업로드한 “admission.csv” 파일을 읽고 문제 해당하는 코드를 작성
- 데이터 속성은 아래와 같음

피쳐	설명	데이터 분류	
GRE_Score	대학원 입학 평가 시험 점수	연속형	Input
TOEFL_Score	TOEFL 점수	연속형	
University_Rating	대학 평가 지수	범주형	
SOP	입학서 점수	연속형	
LOR	추천서 점수	연속형	
CGPA	학부 성적	연속형	
Research	연구 경험	범주형(1:있음/ 0: 없음)	
Chance_of_Admit	대학원 합격 확률	연속형	Output

(6점) 1. 파일을 읽고 데이터의 앞 부분을 출력하는 코드

	GRE_Score	TOEFL_Score	University_Rating	SOP	LOR	CGPA	Research	Chance_of_Admit
0	337	118	4	4.5	4.5	9.65	1	0.92
1	324	107	4	4.0	4.5	8.87	1	0.76
2	316	104	3	3.0	3.5	8.00	1	0.72
3	322	110	3	3.5	2.5	8.67	1	0.80
4	314	103	2	2.0	3.0	8.21	0	0.65

(9점) 2. 대학 평가 지수에 따른 학부 성적의 평균을 구하는 코드

```
1    7.745769
2    8.183738
3    8.552256
4    9.021622
5    9.291167
```

[결 과]

(9점) 3. 연구 경험 유무에 따른 비율을 구하는 코드

```
연구경험 유: 0.5475
연구경험 무: 0.4525
```

[결 과]

* 참고 코드

- 문자열 + 숫자 출력

```
a = 1
b, c = 2, 3
d = b + c
print("a =", a)
print("d =", d)
```

(8점) 4. 결측치 비율을 확인하는 코드

```
GRE_Score      0.02
TOEFL_Score    0.00
University_Rating 0.00
SOP            0.00
LOR            0.00
CGPA           0.00
Research        0.00
Chance_of_Admit 0.00
dtype: float64
```

[결 과]

(8점) 5. 결측치가 있는 열에 대하여 대학 평가 지수에 따른 평균값으로 대체하고 다시 결측치 개수 확인하는 코드

```
GRE_Score      0
TOEFL_Score    0
University_Rating 0
SOP            0
LOR            0
CGPA           0
Research        0
Chance_of_Admit 0
dtype: int64
```

[결 과]

(10점) 6. 대학 평가 지수를 원 핫 인코딩하여 원본 데이터에 열 방향으로 합하고 앞 일부를 출력하는 코드
- prefix는 “UR”로 지정

	GRE_Score	TOEFL_Score	University_Rating	SOP	LOR	CGPA	Research	Chance_of_Admit	UR_1	UR_2	UR_3	UR_4	UR_5
0	337.0	118	4	4.5	4.5	9.65	1	0.92	False	False	False	True	False
1	324.0	107	4	4.0	4.5	8.87	1	0.76	False	False	False	True	False
2	316.0	104	3	3.0	3.5	8.00	1	0.72	False	False	True	False	False
3	322.0	110	3	3.5	2.5	8.67	1	0.80	False	False	True	False	False
4	314.0	103	2	2.0	3.0	8.21	0	0.65	False	True	False	False	False

[결 과]

(10점) 7. 대학원 입학 평가 시험 점수는 최솟값-최댓값 정규화하여 ‘GRE’열을 생성하고 TOEFL 점수는 z-스코어로 정규화하여 ‘TOEFL’열을 생성하여 앞 일부를 출력하는 코드

	GRE_Score	TOEFL_Score	University_Rating	SOP	LOR	CGPA	Research	Chance_of_Admit	UR_1	UR_2	UR_3	UR_4	UR_5	GRE	TOEFL
0	337.0	118	4	4.5	4.5	9.65	1	0.92	False	False	False	True	False	0.94	1.744786
1	324.0	107	4	4.0	4.5	8.87	1	0.76	False	False	False	True	False	0.68	-0.067551
2	316.0	104	3	3.0	3.5	8.00	1	0.72	False	False	True	False	False	0.52	-0.561824
3	322.0	110	3	3.5	2.5	8.67	1	0.80	False	False	True	False	False	0.64	0.426723
4	314.0	103	2	2.0	3.0	8.21	0	0.65	False	True	False	False	False	0.48	-0.726582

[결 과]

- 8~11번은 데이터를 사용하여 아래 그림과 같이 지도학습을 진행함
- Input 데이터 열은 아래 벡터 열 이름을 사용함

```
x_col=['University_Rating','SOP','LOR','CGPA','Research','UR_1','UR_2','UR_3','UR_4','UR_5','GRE','TOEFL']
```



[머신러닝 지도 학습 구조]

(10점) 8. 「Output 데이터 만들기」 대학원 합격 확률을 아래 표처럼 구간을 나누어 범주형 데이터로 생성하고 앞 일부를 출력하는 코드

구간	라벨	admit_Bad	admit_NotBad	admit_Okay	admit_Good	admit_Great	admit_Excellent
0-7.5	Bad	True	False	False	False	False	False
7.5-8.0	NotBad	True	False	False	False	False	False
8.0-8.5	Okay	True	False	False	False	False	False
8.5-9.0	Good	True	False	False	False	False	False
9.0-9.5	Great	True	False	False	False	False	False
9.5-10	Excellent	True	False	False	False	False	False

[결과]

(8점) 9. 「데이터 분리」 테스트 데이터는 전체 데이터의 25%로 사용하여 훈련 데이터와 테스트 데이터를 분리하는 코드

- 코드 분리 후, 넘파이 배열로 지정 후, “np.float32”로 데이터 타입 지정하는 것이 안정적

```
X_train = np.asarray(X_train).astype(np.float32)
y_train = np.asarray(y_train).astype(np.float32)
X_test = np.asarray(X_test).astype(np.float32)
y_test = np.asarray(y_test).astype(np.float32)
```

(10점) 10. 「모델 생성」 그림 “머신러닝 지도 학습 구조”를 기반으로 모델 생성하여 summary함수 호출하는 코드

```
Model: "model"

Layer (type)                 Output Shape              Param #
=====
input_1 (InputLayer)         [(None, 12)]              0
dense (Dense)                 (None, 10)                130
dense_1 (Dense)               (None, 8)                 88
dense_2 (Dense)               (None, 6)                 54
=====
Total params: 272
Trainable params: 272
Non-trainable params: 0
```

[결 과]

(6점) 10. 「학습」 배치 사이즈는 5, 에폭은 10으로 하여 학습 시키는 코드

```
Epoch 1/10
48/48 [=====] - 0s 3ms/step - loss: 9.8651e-05 - acc: 1.0000 - val_loss: 1.0569e-04 - val_acc: 1.0000
Epoch 2/10
48/48 [=====] - 0s 2ms/step - loss: 9.4169e-05 - acc: 1.0000 - val_loss: 1.0137e-04 - val_acc: 1.0000
Epoch 3/10
48/48 [=====] - 0s 2ms/step - loss: 9.0126e-05 - acc: 1.0000 - val_loss: 9.6351e-05 - val_acc: 1.0000
Epoch 4/10
48/48 [=====] - 0s 2ms/step - loss: 8.5986e-05 - acc: 1.0000 - val_loss: 9.2379e-05 - val_acc: 1.0000
Epoch 5/10
48/48 [=====] - 0s 2ms/step - loss: 8.2292e-05 - acc: 1.0000 - val_loss: 8.8354e-05 - val_acc: 1.0000
Epoch 6/10
48/48 [=====] - 0s 2ms/step - loss: 7.8796e-05 - acc: 1.0000 - val_loss: 8.4485e-05 - val_acc: 1.0000
Epoch 7/10
48/48 [=====] - 0s 2ms/step - loss: 7.5390e-05 - acc: 1.0000 - val_loss: 8.1075e-05 - val_acc: 1.0000
Epoch 8/10
48/48 [=====] - 0s 2ms/step - loss: 7.2265e-05 - acc: 1.0000 - val_loss: 7.7611e-05 - val_acc: 1.0000
Epoch 9/10
48/48 [=====] - 0s 2ms/step - loss: 6.9218e-05 - acc: 1.0000 - val_loss: 7.4647e-05 - val_acc: 1.0000
Epoch 10/10
48/48 [=====] - 0s 2ms/step - loss: 6.6426e-05 - acc: 1.0000 - val_loss: 7.1447e-05 - val_acc: 1.0000
```

[결 과 예 시]

(6점) 11. 「평가」 Loss율과 정확도를 보여주는 코드

```
Test Loss: 0.0022197342477738857
Test Accuracy: 1.0
```

[결 과 예 시]