데이터 분석 기초
Fundamental of data analysis

기말 고사

0 기말고사 Intro

- 가상의 기상청 데이터 셋을 통해 주어진 상황에서 습도가 일정이상을 넘었는지 확인하는 로지스틱 회귀 모델을 학습시키고자 합니다.
- 문제는 총 3문제이고 전처리, 데이터 탐색 및 로지스틱 회귀 모델을 학습하는 과정으로 구성되어 있습니다.
- 각 문제의 점수 배점은 다음과 같습니다.

1-1	1-2	2-1	2-2	2-3	3-1	3-2
15	15	15	10	10	15	20

0 사용할 데이터 소개

	location	location_name	time	precipitation	humidity	vapor_pressure	local_pressure	cloudy	ground_temperature	sea_temperature	air_temperature
0	184.0	제주	2021-06-01 1:00	NaN	99.0	52.0	1005.0	4.0	25.0	24.0	35.0
1	184.0	제주	2021-06-01 2:00	NaN	82.0	60.0	999.0	5.0	26.0	23.0	50.0
2	184.0	제주	2021-06-01 3:00	NaN	97.0	50.0	1007.0	4.0	26.0	26.0	30.0
3	184.0	제주	2021-06-01 4:00	NaN	78.0	58.0	1002.0	8.0	26.0	24.0	40.0
4	184.0	제주	2021-06-01 5:00	NaN	92.0	50.0	1000.0	6.0	28.0	23.0	45.0

- **Location – 관측 지점**
- **Location name – 관측 지점명**
- **Time – 관측 시각**
- **Precipitation – 강수량**
- **Vapor_pressure – 수증기압**
- **Local pressure – 기압**
- **Cloudy – 구름량**
- **Ground_temperature – 지면 온도**
- **Sea_temperature – 해수 온도**
- **Air_temperature – 대기 온도**

1-1 데이터 구성과 전처리

	location	location_name	time	precipitation	humidity	vapor_pressure	local_pressure	cloudy	ground_temperature	sea_temperature	air_temperature
0	184.0	제주	2021-06-01 1:00	NaN	99.0	52.0	1005.0	4.0	25.0	24.0	35.0
1	184.0	제주	2021-06-01 2:00	NaN	82.0	60.0	999.0	5.0	26.0	23.0	50.0
2	184.0	제주	2021-06-01 3:00	NaN	97.0	50.0	1007.0	4.0	26.0	26.0	30.0
3	184.0	제주	2021-06-01 4:00	NaN	78.0	58.0	1002.0	8.0	26.0	24.0	40.0
4	184.0	제주	2021-06-01 5:00	NaN	92.0	50.0	1000.0	6.0	28.0	23.0	45.0

- **데이터에서 location, location_name, time 열을 제거해주세요.**

1-1 데이터 구성과 전처리 - 제출물

- **전처리가 완료된 데이터셋의 행의 일부를 print합니다.**

```
#채점을 위한 output입니다. 변수명에 주의해주세요.
for i in test_list :
  print(process_weather.iloc[i])
```

*** 채점 코드는 제공되니 변수명에 주의해주세요**

입력값

```
1
2
3
```

출력값

```
precipitation          NaN
humidity              82.0
vapor_pressure        60.0
local_pressure       999.0
cloudy                 5.0
ground_temperature    26.0
sea_temperature       23.0
air_temperature       50.0
Name: 1, dtype: float64
precipitation          NaN
humidity              97.0
```

1-2 데이터 구성과 전처리

[과정 1] 'rainy' 열을 생성하세요.

- Precipitation 열의 NaN은 0으로 변경하세요 (다른 열의 NaN은 처리하면 안됩니다)
- Precipitation이 0 은– rainy의 값은 0
- Precipitation이 0 보다 큰 경우 – rainy의 값은 1
- 완료된 후에는 기존 precipitation 열을 꼭 삭제하세요.

[과정 2] 'over_humidity' 열을 생성하세요.

- humidity가 90 이상일 경우 – over_humidity의 값은 1
- humidity가 90 미만일 경우 – over_humidity의 값은 0
- 완료된 후에는 기존 humidity 열을 꼭 삭제하세요.

[과정 3] [과정1, 2]를 거친 후, 결측치가 들어간 행은 삭제해 주세요.

1-2 데이터 구성과 전처리

- **정답이 존재하는 문제입니다. 아래 head()를 참고하세요.**

전처리 전 데이터의 .head()

	location	location_name	time	precipitation	humidity	vapor_pressure	local_pressure	cloudy	ground_temperature	sea_temperature	air_temperature
0	184.0	제주	2021-06-01 1:00	NaN	99.0	52.0	1005.0	4.0	25.0	24.0	35.0
1	184.0	제주	2021-06-01 2:00	NaN	82.0	60.0	999.0	5.0	26.0	23.0	50.0
2	184.0	제주	2021-06-01 3:00	NaN	97.0	50.0	1007.0	4.0	26.0	26.0	30.0
3	184.0	제주	2021-06-01 4:00	NaN	78.0	58.0	1002.0	8.0	26.0	24.0	40.0
4	184.0	제주	2021-06-01 5:00	NaN	92.0	50.0	1000.0	6.0	28.0	23.0	45.0

전처리 후 데이터의 .head()

	vapor_pressure	local_pressure	cloudy	ground_temperature	sea_temperature	air_temperature	rainy	over_humidity
0	52.0	1005.0	4.0	25.0	24.0	35.0	0	1
1	60.0	999.0	5.0	26.0	23.0	50.0	0	0
2	50.0	1007.0	4.0	26.0	26.0	30.0	0	1
3	58.0	1002.0	8.0	26.0	24.0	40.0	0	0
4	50.0	1000.0	6.0	28.0	23.0	45.0	0	1

1-2 데이터 구성과 전처리 - 제출물

- **전처리가 완료된 데이터셋의 행의 일부를 print합니다.**

```
#채점을 위한 output입니다. 변수명에 주의해주세요.
for i in test_list :
  print(process_weather.iloc[i])
```

*** 채점 코드는 제공되니 변수명에 주의해주세요**

입력값

```
1
2
3
```

출력값

```
vapor_pressure        60.0
local_pressure       999.0
cloudy                 5.0
ground_temperature    26.0
sea_temperature       23.0
air_temperature       50.0
rainy                  0.0
over_humidity          0.0
Name: 1, dtype: float64
```

2-1 데이터 탐색

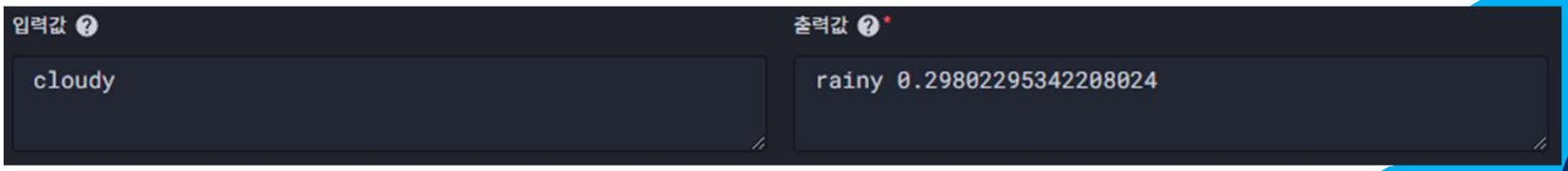
- 1번 전처리 과정이 적용된 dataset이 주어집니다.
- 아래의 순서로 동작하도록 프로그램을 구성하세요.

  (1) 종속 변수 1개를 입력합니다 (예시: cloudy)

  (2) 프로그램은 나머지 변수들 중 입력된 변수와의 양의 상관계수값이 가장 큰 변수명을 출력합니다 (예시: rainy)

  (3) 이어서, 상관계수를 함께 출력합니다 (예시: cloudy와 rainy의 상관계수값인 0.298)

예시



2-2 데이터 탐색

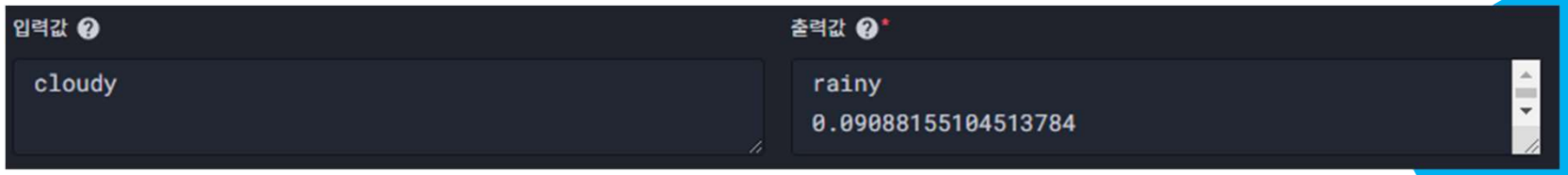
- **1번 전처리 과정이 적용된 dataset이 주어집니다.**
- **아래의 순서로 동작하도록 프로그램을 구성하세요.**

  (1) 종속 변수 1개를 입력합니다 (예시: cloudy)

  (2) 프로그램은 나머지 변수들 중 입력된 변수와의 양의 상관계수값이 가장 큰 변수명을 출력합니다 (예시: rainy)

  (3) 출력한 변수의 평균값을 출력합니다 (예시: rainy의 평균값이 0.0908)

예시



2-3 데이터 탐색

- 1번 전처리 과정이 적용된 dataset이 주어집니다.
- 아래의 순서로 동작하도록 프로그램을 구성하세요.

(1) 종속 변수 1개를 입력합니다 (예시: cloudy)

(2) Grouping 기준값을 입력합니다 (예시: 3)

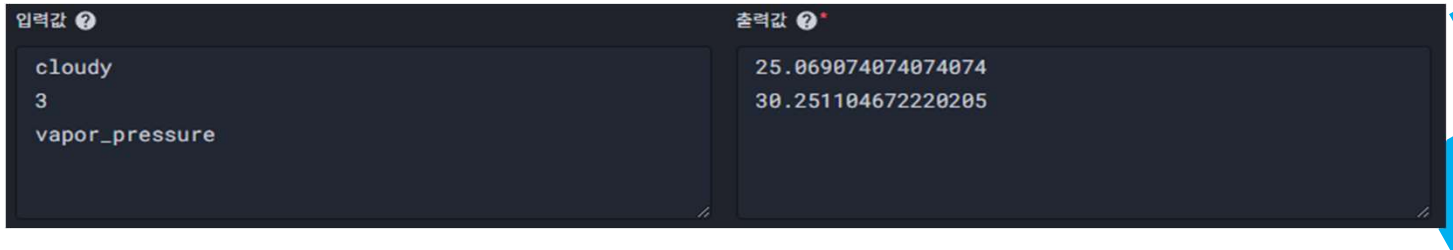
- 이 경우, 데이터는 cloudy가 3 미만과 3 이상인 두 그룹을 나누기 위한 기준이 됩니다.

(3) 추가변수 1개를 입력하고 (예: vapor_pressure),

(2)의 기준으로 나뉜 두 그룹에 대해 추가변수의 평균을 출력하세요.

- 이 경우, 25.06은 cloudy가 3 미만인 데이터의 vapor_pressure 평균입니다.
- 이 경우, 30.25는 cloudy가 3 이상인 데이터의 vapor_pressure 평균입니다.

예시



3-1 회귀 모델 학습

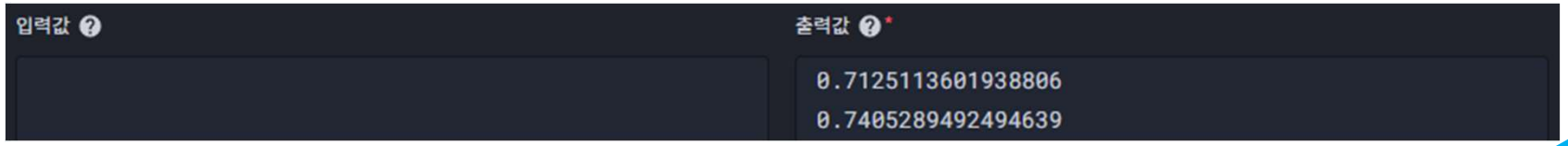
- 로지스틱 회귀 모델을 학습하려고 합니다.
- 1번 전처리 과정이 적용된 dataset이 주어집니다.
- 종속변수: 'over_humidity'
- 독립변수: 'over_humidity' 제외 다른 변수들
- 추가 전처리 과정 없이 모든 독립변수를 사용하여 학습하세요.

* 본 문제는 채점을 원활히 하기 위해 warning 메시지가 출력되지 않게 설정하였습니다.
* 학습을 위해 Max_iter 값을 1000 으로 설정하세요.

3-1 회귀 모델 학습 - 제출

- 학습이 완료된 모델의 score를 train_set/test_set에 대하여 출력합니다.
- 별도의 전처리 과정을 하지 않고 모든 독립변수를 사용하여 학습 시키면, 아래와 같이 제공되는 채점 코드로 score 점수가 출력됩니다.

```
print(lr.score(train_X, train_Y))
print(lr.score(test_X, test_Y),"\n")
```



출력 예시

3-2 회귀 모델 학습(도전)

- 3-1 문제와 동일한 목적의 모델을 만드세요. 여러 시도를 통해 test score를 더욱 높이는 것이 목적입니다.
- 시각화를 통한 데이터 탐색, 전처리, 변수조절 등을 통해 test score를 높여주세요.

  * 이상치 제거 가이드: vapor_pressure < 10, vapor_pressure >= 50, ground_temperature >=45.(추천 사항. 수정 적용 가능)
- 결측치 처리 이후의 학습 또한 모델의 성능에 영향을 줄 수 있습니다.
- 등급에 따른 채점 기준은 아래와 같습니다.

  * train score가 0.65 이하면 일괄적으로 0점 처리됩니다.

Test score 기준	점수
0.83이상	20점
0.81 이상 0.83 미만	15점
0.79 이상 0.81미만	10점
0.75 이상 0.79 미만	5점
0.75 미만	0점

3-2 회귀 모델 학습(도전) - 제출

- 모델의 score를 확인합니다. 채점 함수가 복잡하니 변수명을 꼭 지켜주세요.
- 채점함수는 제공됩니다.

```
calculation(len train_X.keys()), lr.score train_X, train_Y), lr.score(test_X, test_Y))
```

Train set model Train set, testset

Thank you