1. 데이터 준비

1 데이터 불러오기

os 모듈

os 모듈은 운영체제에서 제공하는 여러 기능을 파이썬에서 수행할 수 있게 하는 모듈입니다.

함수	입력	출력
getcwd	없음	현재 경로를 반환
chdir	path	현재 경로를 path로 수정
listdir	path	path 내에 있는 모든 파일명 반환

현재 경로 확인 및 설정

현재 경로는 os 모듈의 getcwd 함수를 이용해 확인할 수 있습니다.

경로 설정 예제

```
1 import os
2 print(os.getcwd())
```

```
C:\Users\<사용자명>\Desktop\파이썬을 이용한 머신러닝 자동화 시스템 구축\code\02. 파이썬을 이용한 머신러닝 모델 학습
```

os 모듈의 chdir(path) 함수를 사용하면 현재 경로를 다음과 같이 바꿀 수 있습니다.

경로 수정 예제

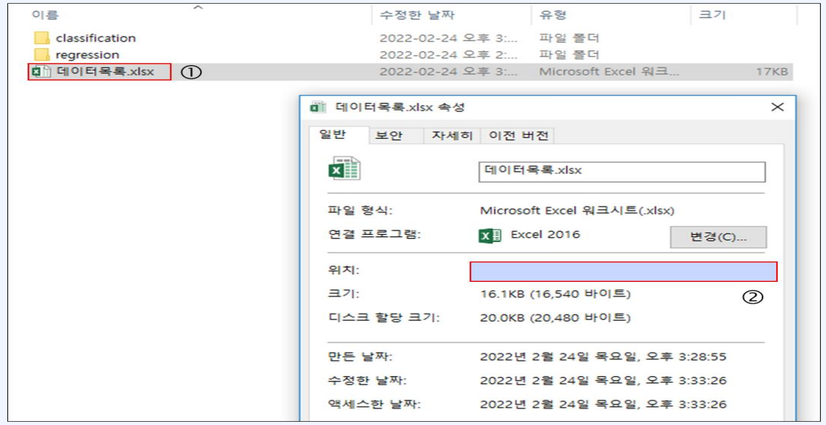
```
1 os.chdir("C:/Users/<사용자명>/Desktop/파이썬을 이용한 머신러닝 자동화 시스템 구축/data")
2 print(os.getcwd())
```

```
C:\Users\<사용자명>\Desktop\파이썬을 이용한 머신러닝 자동화 시스템 구축\data
```

현재 경로를 기준으로 파일을 불러올 수 있으므로, getcwd 함수를 이용해 현재 경로를 확인하고 chdir 함수를 이용해 현재 경로를 수정해야 함

경로 설정 방법

경로를 직접 입력해도 되지만, 속성 창에 있는 폴더 경로를 복사해서 사용하는 것이 편합니다.



- 파일(①)을 우클릭한 뒤 [속성]을 클릭하여 속성 창을 열기
- 속성 창에서 위치(②) 항목의 오른쪽에 있는 폴더 경로를 복사

경로 수정 방법

복사한 경로 예시: "C:\Users\<사용자명>\Desktop\파이썬을 이용한 머신러닝 자동화 시스템 구축\data"

파이썬에서 역슬래시는 이스케이프 문자(escape string)를 표현하는데 사용하므로, 경로 내 역슬래시를 아래 중 하나로 변환해야 합니다.

- 방법 (1) 역슬래시를 슬래시로 변환: "C:/Users/<사용자명>/Desktop/파이썬을 이용한 머신러닝 자동화 시스템 구축/data"
- 방법 (2) 역슬래시 두 개 사용: "C:\\Users\\<사용자명>\\Desktop\\파이썬을 이용한 머신러닝 자동화 시스템 구축\\data"
- 방법 (3) r 사용: r"C:\Users\<사용자명>\Desktop\파이썬을 이용한 머신러닝 자동화 시스템 구축\data"

상대 경로 설정

파이썬에서는 절대 경로뿐만 아니라 상대 경로를 바탕으로 데이터를 불러올 수 있습니다.

상대 경로 예시

폴더 0
- 폴더 1
  - 현재 폴더
    - 폴더 1-1
      - 파일 1
- 폴더 2
  - 폴더 2-1
    - 파일 2

절대 경로	상대 경로
폴더 0	../../
폴더 1	../
폴더 1-1	폴더 1-1
파일 1	폴더 1-1/파일 1
폴더 2	../../폴더 2
폴더 2	../../폴더 2/폴더 2-1
파일 2	../../폴더 2/폴더 2-1/파일 2

read_csv 함수

pandas의 read_csv 함수는 csv 파일 을 불러오는데 사용하는 함수입니다.

주요 인자

인자	설명	기본 값
filepath	읽어올 파일의 경로	없음
sep	구분자	“,”
header	칼럼의 위치	0

데이터 불러오기 예제

```
1  import pandas as pd
2  df = pd.read_csv("../../data/classification/adult.csv")
3  display(df.head())
```

	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11	x12	x13	x14	y
0	25	Private	226802	11th	7	Never-married	Machine-op-inspct	Own-child	Black	Male	0	0	40	United-States	<=50K
1	38	Private	89814	HS-grad	9	Married-civ-spouse	Farming-fishing	Husband	White	Male	0	0	50	United-States	<=50K
2	28	Local-gov	336951	Assoc-acdm	12	Married-civ-spouse	Protective-serv	Husband	White	Male	0	0	40	United-States	>50K
3	44	Private	160323	Some-college	10	Married-civ-spouse	Machine-op-inspct	Husband	Black	Male	7688	0	40	United-States	>50K
4	18	NaN	103497	Some-college	10	Never-married	NaN	Own-child	White	Female	0	0	30	United-States	<=50K

(잠깐) print와 display 함수

- print 함수: 문자열로 바꿔 출력하는 함수

문자열 출력

Hello World

데이터프레임 출력

```
   A B
0  1 4
1  2 5
2  3 6
```

ndarray 출력

[1 2 3]

- display 함수: 원본 모양 그대로 출력하는 함수

문자열 출력

'Hello World'

데이터프레임 출력

	A	B
0	1	4
1	2	5
2	3	6

ndarray 출력

array([1,2,3])

- 주피터 노트북에선 굳이 두 함수를 사용하지 않아도 맨 아래 실행된 값이 출력됨

1. 데이터 준비

2 데이터 확인하기

shape 메서드

shape 메서드는 아래와 같이 데이터프레임의 모양을 확인하는 데 사용합니다.

```
1  print(df.shape)
```

```
(48842, 15)
```

shape 메서드의 출력 결과는 (행 개수, 열 개수)로 표현된 데이터의 모양으로, 샘플 수와 특징 개수를 알 수 있음

info 메서드

info 메서드는 데이터프레임을 구성하는 칼럼별 데이터 타입과 결측을 제외한 샘플 수를 반환합니다.

```
1    print(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 48842 entries, 0 to 48841
Data columns (total 15 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   x1      48842 non-null  int64
 1   x2      46043 non-null  object
 2   x3      48842 non-null  int64
 3   x4      48842 non-null  object
 4   x5      48842 non-null  int64
 5   x6      48842 non-null  object
 6   x7      46033 non-null  object
 7   x8      48842 non-null  object
 8   x9      48842 non-null  object
 9   x10     48842 non-null  object
 10  x11     48842 non-null  int64
 11  x12     48842 non-null  int64
 12  x13     48842 non-null  int64
 13  x14     47985 non-null  object
 14  y       48842 non-null  object
dtypes: int64(6), object(9)
memory usage: 5.6+ MB
None
```

칼럼 x1은 데이터 타입이 int64이고 결측을 제외한 샘플 수가 48,842개이며,
칼럼 x2는 데이터 타입이 object이고 결측을 제외한 샘플 수가 46,043개임

describe 메서드

describe 메서드는 숫자형 칼럼의 기술 통계량을 반환합니다.

```
1  display(df.describe())
```

	x1	x3	x5	x11	x12	x13
count	48842.000000	4.884200e+04	48842.000000	48842.000000	48842.000000	48842.000000
mean	38.643585	1.896641e+05	10.078089	1079.067626	87.502314	40.422382
std	13.710510	1.056040e+05	2.570973	7452.019058	403.004552	12.391444
min	17.000000	1.228500e+04	1.000000	0.000000	0.000000	1.000000
25%	28.000000	1.175505e+05	9.000000	0.000000	0.000000	40.000000
50%	37.000000	1.781445e+05	10.000000	0.000000	0.000000	40.000000
75%	48.000000	2.376420e+05	12.000000	0.000000	0.000000	45.000000
max	90.000000	1.490400e+06	16.000000	99999.000000	4356.000000	99.000000

- count: 결측을 제외한 값의 개수
- mean: 평균
- std: 표준편차
- min: 최솟값
- 25%: 값이 작은 상위 25% 값(1사분위수)
- 50%: 값이 작은 상위 50% 값(중위수)
- 75%: 값이 작은 상위 75% 값(3사분위수)
- max: 최댓값

1. 데이터 준비

3 데이터 분할하기

drop 메서드

drop 메서드는 인덱스를 기준으로 데이터프레임의 행(axis = 0) 혹은 열(axis = 1)을 제거합니다.

특징과 라벨 분리 예제

```
1  X = df.drop('y', axis = 1)
2  y = df['y']
```

- **라인 1**: 이름이 'y'인 칼럼을 제거한 데이터프레임을 변수 X에 저장합니다.
- **라인 2**: 이름이 'y'인 칼럼을 변수 y에 저장합니다.

머신러닝 모델을 개발할 때는 특징과 라벨을 분리하거나 불필요한 칼럼(예: ID)을 제거하는데 이 메서드를 주로 사용함

train_test_split 함수

train_test_split 함수는 하나 이상의 배열을 학습 데이터와 평가 데이터로 분할합니다.

주요 인자

인자	설명
*arrays	분할할 배열로 입력한 순서대로 학습 데이터와 평가 데이터로 분할됨
test_size	전체 데이터에서 평가 데이터의 비율
random_state	시드를 나타내며, 모델 재현을 위해 고정하는 것이 바람직함

학습 데이터와 평가 데이터 분리 예제

```
1  from sklearn.model_selection import train_test_split
2  X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.3, random_state = 2022)
```

- **라인 2**: X와 y를 학습 데이터(비율: 70%)와 평가 데이터(비율: 30%)로 분할합니다. X와 y를 순서대로 입력했으므로 출력을 X_train, X_test, y_train, y_test로 정의했습니다.

(잠깐) 시드

파이썬에서 생성하는 난수는 엄밀히 따지자면 난수가 아니며, 시드라는 특정한 숫자를 시작으로 정해진 알고리즘을 사용해 난수를 순서대로 생성합니다.
따라서 시드를 고정하면 같은 결과를 재현할 수 있습니다.

알고리즘 예시: 선형합동법

$$x_{n+1} = \mathrm{MOD}(1103515245 \times x_n + 12345, 2^{31})$$

코드 예시

```
np.random.seed(0)
print(np.random.random(4))
np.random.seed(0)
print(np.random.random(4))
```

```
[0.5488135  0.71518937 0.60276338 0.54488318]
[0.5488135  0.71518937 0.60276338 0.54488318]
```

KFold 클래스

KFold 클래스는 데이터의 인덱스를 서로 중복되지 않는 k개의 폴드로 분할하여 i(i=0, 1, 2, ⋯, k-1)번째 이터레이션에서 i번째 폴드를 평가 데이터의 인덱스로 나머지 폴드에 속한 모든 인덱스를 학습 데이터의 인덱스로 반환하는 인스턴스를 생성합니다.

주요 인자

인자	설명	기본값
n_splits	k-겹 교차 검증에서 폴드 개수	5
shuffle	데이터를 나누기 전에 섞을 것인지 여부	False

주요 메서드

메서드	설명
split	학습 데이터와 평가 데이터의 인덱스를 순서대로 반환

KFold를 이용한 k-겹 교차 검증 예제

```
1  from sklearn.model_selection import KFold
2  kf = KFold(n_splits = 5, shuffle = True)
3  for train_idx, test_idx in kf.split(X):
4      X_train, X_test = X.loc[train_idx], X.loc[test_idx]
5      y_train, y_test = y.loc[train_idx], y.loc[test_idx]
6      # model training and test
```

- **라인 2**: KFold 인스턴스를 생성합니다.
- **라인 3**: split 메서드에 X를 입력하여, X의 인덱스를 학습 데이터의 인덱스인 train_idx와 평가 데이터의 인덱스 test_idx로 나누어 순회합니다. **라인 2**에서 n_splits를 5로 설정했으므로, test_idx는 0번째 폴드부터 4번째 폴드의 인덱스를 순회합니다.
- **라인 4 – 6**: loc 인덱서 를 사용해 학습 데이터와 평가 데이터로 분할합니다. 실제로 머신러닝 모델을 개발하는 상황이라면 **라인 6**부터 X_train과 y_train로 모델을 학습하고 X_test와 y_test으로 학습한 모델을 평가합니다.