1강. Data handling with Python

◈ 담당교수 : 김 동 하

■ 학습개요

이번 강의는 다음 차시부터 배울 다양한 머신러닝 방법론을 구현하기 위해서 필수적으로 익혀야 하는 프로그램인 파이썬에 대해서 다루도록 한다. 특히, 파이썬의 변수 및 자료의 형태에 대해서 학습하며, 데이터를 입력하고 출력하는 방법에 대해서도 배우도록 한다. 더 나아가서, 주어진 데이터들을 병합 및 추출하고, 칼럼의 값을 기준으로 재정렬하는 등의 데이터 핸들링 기법에 대해서 집중적으로 알아보고자 한다.

■ 학습목표

1	파이썬의 변수 및 자료의 형태에 대해 학습한다.
2	데이터의 입출력 방법에 대해 학습한다.
3	주어진 데이터의 병합, 추출, 정렬 등의 데이터 핸들링 기법에 대해 학습한다.

■ 주요용어

용어	해설
파이썬 (Python)	세계에서 가장 많은 사용자를 가지고 있는 오픈 소스 기반 프로그램으로, 머신 러닝의 구현을 위해 필수적이다. 쉬운 문법과 높은 가독성, 그리고 풍부한 라이브러리를 보유하고 있어 사용자가 손쉽게 데이터 분석을 할 수 있다.
Pandas와 Numpy	파이썬 내에서 데이터 핸들링을 위해 필요한 다양한 명령어를 제공하는 대표적인 라이브러리. 본 강의에서는 대부분의 데이터 핸들링을 pandas를 이용한다.
데이터 핸들링	말 그대로 데이터를 가공하는 작업을 말한다. 데이터를 입력받아서 병합하고, 원하는 부분만을 출력하고, 가공한 자료를 다시 새롭게 저장하는 일련의 모든 과정을 뜻한다.

■ 학습하기
‘

01. Python 프로그램

파이썬이란?

- 네덜란드 프로그래머인 귀도 반 로섬이 발표한 고급 프로그래밍 언어.
- 비영리의 파이썬 소프트웨어 재단이 관리하는 개방형, 공동체 기반 개발 모델.
- 가장 많은 프로그램 사용자들이 사용하고 있음.
- 인공지능(AI) 산업의 성장과 더불어 높은 사용 증가율을 보이고 있음.

파이썬의 장단점

- 쉬운 문법, 높은 가독성, 풍부한 라이브러리를 보유하고 있음.
- 다양한 플랫폼에서 사용 가능, 메모리 자동 관리 기능 탑재
- 빠른 속도로 처리 불가 (특히, loop문의 경우 느리다는 단점)
- 하드웨어를 직접 건드려야 하는 일에는 적합하지 않음.

02. 데이터 입출력하기

데이터 입력하기

- 주로 pandas 패키지의 read_csv, read_table, read_excel을 사용.
- read_csv
  > 파일 혹은 URL 등으로부터 데이터를 읽어오는 함수.
  > 데이터 구분자는 쉼표(,)를 기본으로 함.
- read_table
  > read_csv와 같은 역할.
  > 데이터 구분자를 탭(₩t)으로 한다는 점에서 read_csv와 차이가 있음.
- read_excel
  > 엑셀 파일 (.xls, .xlsx)의 데이터를 읽어오는 함수.

CSV 파일 불러오기

- 데이터가 존재하는 디렉토리 설정
- read_csv 또는 read_table을 이용하여 데이터 불러오기.

```
[1]: import pandas as pd                                    # pandas 패키지 불러오기
     import os                                              # os 패키지 불러오기

[2]: os.getcwd()

[2]: '/home/dongha0718/KNOU_Machine_Learning/chap1'

[3]: data_path = './data'                                   # Data 경로
     os.chdir(data_path)                                    # 작업 디렉토리 변경

[4]: os.getcwd()

[4]: '/home/dongha0718/KNOU_Machine_Learning/chap1/data'
```

- 칼럼명이 없는 데이터 불러오기

```
[10]: pd.read_csv('ex2.csv')
```

[10]:

	1	2	3	4	hello
0	5	6	7	8	world
1	9	10	11	12	foo

엑셀 파일 불러오기

- openpyxl, xlrd를 설치해야 한다.

  > pip install openpyxl xlrd

- 첫번째 sheet에 있는 데이터 불러오기.

```
[17]: xls_filename = "ex1.xlsx"
      pd.read_excel(xls_filename, sheet_name = "Sheet1", engine='openpyxl')
```

[17]:

	Unnamed: 0	a	b	c	d	message
0	0	1	2	3	4	hello
1	1	5	6	7	8	world
2	2	9	10	11	12	foo

결측값 다루기

- 특정 값에 대해서 결측값으로 처리할 수 있다.
- 특정 갈럼을 선택해서 결측치 처리를 할 수도 있다.

```
[18]: sentinels = {'message':['foo', 'NA'], 'something':['two']}
      pd.read_csv('ex3.csv', na_values=sentinels)
```

[18]:

	something	a	b	c	d	message
0	one	1	2	3.0	4	NaN
1	NaN	5	6	NaN	8	world
2	three	9	10	11.0	12	NaN

데이터 출력하기

- 기본적으로 to 함수를 사용한다.

> to 함수: 주어진 데이터를 원하는 형태, 원하는 이름의 파일로 내보내는 함수.

- to 함수를 이용하여 파일을 저장해보자.

```
[31]: data = pd.read_csv('ex1.csv')
      data.to_csv('out.csv')
```

```
[32]: pd.read_csv('out.csv')
```

[32]:

	Unnamed: 0	a	b	c	d	message
0	0	1	2	3	4	hello
1	1	5	6	7	8	world
2	2	9	10	11	12	foo

```
[33]: data
```

[33]:

	a	b	c	d	message
0	1	2	3	4	hello
1	5	6	7	8	world
2	9	10	11	12	foo

03. 데이터 핸들링하기

Pandas의 데이터 구조

- 크게 Series와 DataFrame 형태가 존재한다.
- Series

  > 모든 데이터 유형 (정수, 문자 등)을 저장할 수 있는 1차원 배열.

- DataFrame

  > 잠재적으로 다른 유형의 열이 있는 2차원 데이터 구조.

  > 스프레드 시트나 SQL 테이블 등으로 생각할 수 있음.

  > R의 data.frame()과 같은 기능.

Pandas Series

- Series 벡터를 생성해보자.

```
[2]: import pandas as pd
     obj = pd.Series([4,7,-5,3])
     obj
```

```
[2]: 0    4
     1    7
     2   -5
     3    3
     dtype: int64
```

```
[3]: obj.values
```

```
[3]: array([ 4,  7, -5,  3])
```

Pandas DataFrame

- 3개의 DataFrame (user1~3)을 생성해보자.
- 칼럼의 이름을 다양한 방식으로 입력할 수 있다.

[24]:
```
columns = ['name', 'age', 'gender', 'job']

user1 = pd.DataFrame([['alice', 19, "F", "student"],
                      ['john', 26, "M", "student"]],
                     columns=columns)
user1
```

[24]:

	name	age	gender	job
0	alice	19	F	student
1	john	26	M	student

[43]:
```
user2 = pd.DataFrame([['eric', 22, "M", "student"],
                      ['paul', 58, "F", "manager"]],
                     columns=columns)
user2
```

[43]:

	name	age	gender	job
0	eric	22	M	student
1	paul	58	F	manager

[25]:
```
user3 = pd.DataFrame(dict(name=['peter', 'julie'],
                          age=[33, 44],
                          gender=['M', 'F'],
                          job=['engineer', 'scientist']))
user3
```

[25]:

	name	age	gender	job
0	peter	33	M	engineer
1	julie	44	F	scientist

데이터 합치기: combine

- Pandas의 append와 concat 함수를 이용하여 데이터를 합칠 수 있다.

[8]:
```
# Combining DataFrames
user1.append(user2)
```

[8]:

	name	age	gender	job
0	alice	19	F	student
1	john	26	M	student
0	eric	22	M	student
1	paul	58	F	manager

[16]:
```
users = pd.concat([user1, user2, user3])
users
```

[16]:

	name	age	gender	job
0	alice	19	F	student
1	john	26	M	student
0	eric	22	M	student
1	paul	58	F	manager
0	peter	33	M	engineer
1	julie	44	F	scientist

데이터 합치기: merge

- merge 함수는 기본적으로 내부조인 (inner join)을 수행하여 교집합인 결과를 반환한다.
- 조인할 때의 key값은 on 옵션을 통해 설정할 수 있다.
- 아래의 결과는 users (왼쪽 위)와 user4 (오른쪽 위)를 merge한 결과 (아래)이다.

	name	age	gender	job
0	alice	19	F	student
1	john	26	M	student
0	eric	22	M	student
1	paul	58	F	manager
0	peter	33	M	engineer
1	julie	44	F	scientist

	name	height
0	alice	165
1	john	180
2	eric	175
3	julie	171

[19]:
```
# Use union of keys from both frames
users2 = pd.merge(users, user4, on="name")
users2
```

[19]:

	name	age	gender	job	height
0	alice	19	F	student	165
1	john	26	M	student	180
2	eric	22	M	student	175
3	julie	44	F	scientist	171

칼럼 및 로우 선택하기

- iloc 및 loc를 이용하여 칼럼 및 로우를 선택할 수 있으며, 원하는 원소만 뽑을 수도 있다.

- 칼럼 선택하기

[66]:
```
users.iloc[:,1:3]
```

[66]:

	age	gender
0	19	F
1	26	M
0	22	M
1	58	F
0	33	M
1	44	F

[67]:
```
users.loc[:,['age','gender']]
```

[67]:

	age	gender
0	19	F
1	26	M
0	22	M
1	58	F
0	33	M
1	44	F

- 로우 선택하기

[57]:
```
df = users.copy()
df.iloc[0] # first row
```

[57]:
```
name         alice
age             19
gender           F
job        student
Name: 0, dtype: object
```

[47]:
```
df.iloc[2:4]
```

[47]:

	name	age	gender	job
0	eric	22	M	student
1	paul	58	F	manager

[86]:
```
print(df.iloc[0, 0]) # first item of first row
```
```
alice
```

[56]:
```
df.loc[0]
```

[56]:

	name	age	gender	job	1
0	alice	55	F	student	55.0
0	eric	22	M	student	55.0
0	peter	33	M	engineer	55.0

[55]:
```
print(df.loc[0, "age"])
```
```
0    55
0    22
0    33
Name: age, dtype: int64
```

■ 연습문제

(객관식)1. 파이썬의 특징을 잘못 설명한 문항을 고르시오.

① 윈도우, 리눅스, Mac OS X 등 다양한 시슽메에서 사용 가능하다.

② 인공지능(AI) 산업의 성장과 더불어 높은 사용 증가율을 보이고 있는 추세이다.

③ C, JAVA 등의 프로그램과 함께 빠른 속도로 계산 처리가 가능한 프로그램이다.

④ 다양한 플랫폼에서 사용이 가능하며, 메모리 자동 관리 기능이 있다.

정답 : ③

해설 : 파이썬은 C 프로그램과 JAVA 프로그램보다 계산 처리 속도가 늦다.

(객관식)2. 파이썬을 이용한 데이터의 입출력에 대해서 잘못 설명한 것을 고르시오.

① read_csv는 데이터 구분자의 기본 옵션이 탭(₩t)으로 지정되어 있다.

② 엑셀 파일을 불러올 때는 read_excel 명령어를 사용한다.

③ to 명령어를 이용하여 데이터를 저장할 수 있다.

④ 다양한 플랫폼에서 사용이 가능하며, 메모리 자동 관리 기능이 있다.

정답) ①

해설) read_csv는 데이터 구분자의 기본 옵션이 쉼표(,)로 지정되어 있다.

(단답형)3. Pandas 라이브러리에서 데이터를 특정 칼럼을 기준으로 오름차순 또는 내림차순으로 자료를 재정렬할 때 사용하는 명령어의 이름은 무엇인가?

정답 : sort_values

해설 : 해설 없음.

■ 정리하기

1. 파이썬은 머신 러닝 분석을 구현하는데 있어서 필수적인 오픈 소스 기반의 무료 프로그램으로, anaconda에서 다운 받아 사용하거나 구글의 colab에서 구글 계정을 통해 사용할 수 있다.

2. Pandas 라이브러리의 다양한 명령어로 데이터를 불러오고, 저장할 수 있다. 데이터를 불러올 때는 주로 read_csv, read_table, read_excel 등의 명령어를 사용하며, 저장할 때는 to 명령어를 주로 사용한다.

3. Pandas 라이브러리와 Numpy 라이브러리를 이용하여 데이터를 자유롭게 핸들링할 수 있다. 가장 많이 사용하는 기법은 병합, 추출, 정렬로, 모두 간편한 명령어를 통해 손쉽게 구현해볼 수 있다.

■ 참고자료 (참고도서, 참고논문, 참고사이트 등)

1. 참고 사이트: WikiDocs (점프 투 파이썬, https://wikidocs.net/book/1)
=> WikiDocs (점프 투 파이썬) 웹사이트는 데이터 핸들링 뿐만 아니라 파이썬 문법을 친절하고 자세하게 설명해주고 있다.