빅데이터분석 실습

1주 과목소개

데이터 사이언스 전공
담당교수: 곽철완

과목소개

- 강의 개설의 목적
- 무엇을 학습할 것인가?
- 수강생들에게 어떤 역량이 생길 것인가?
- 강의 진행 안내
- 사이킷런 및 주요 라이브러리 안내

1. 강의 개설의 목적

- Python 라이브러리 scikit-learn 활용

  - 머신러닝의 주요 알고리즘인 지도학습 및 비지도학습 학습
  - 빅데이터 분석 역량 강화

- 주요 내용

  - 지도학습: 결정트리, 결정트리의 앙상블, 서포트 벡터 머신
  - 비지도학습: 주성분분석, 군집분석
  - Kaggle(캐글) 경진대회 참가를 통한 빅데이터 분석 역량 증진

2. 무엇을 학습할 것인가?

- 사이킷런 라이브러리의 핵심 알고리즘
  - 머신러닝의 지도학습과 비지도학습 연습
  - 이론적인 내용에 대한 이해
- 데이터 분석 기획과 분석 마스트 플랜
  - 분석 방법론(예, 빅데이터 분석 방법론)
  - 분석 마스터 플랜
- 경진대회 참여와 빅데이터 분석 방법 공유

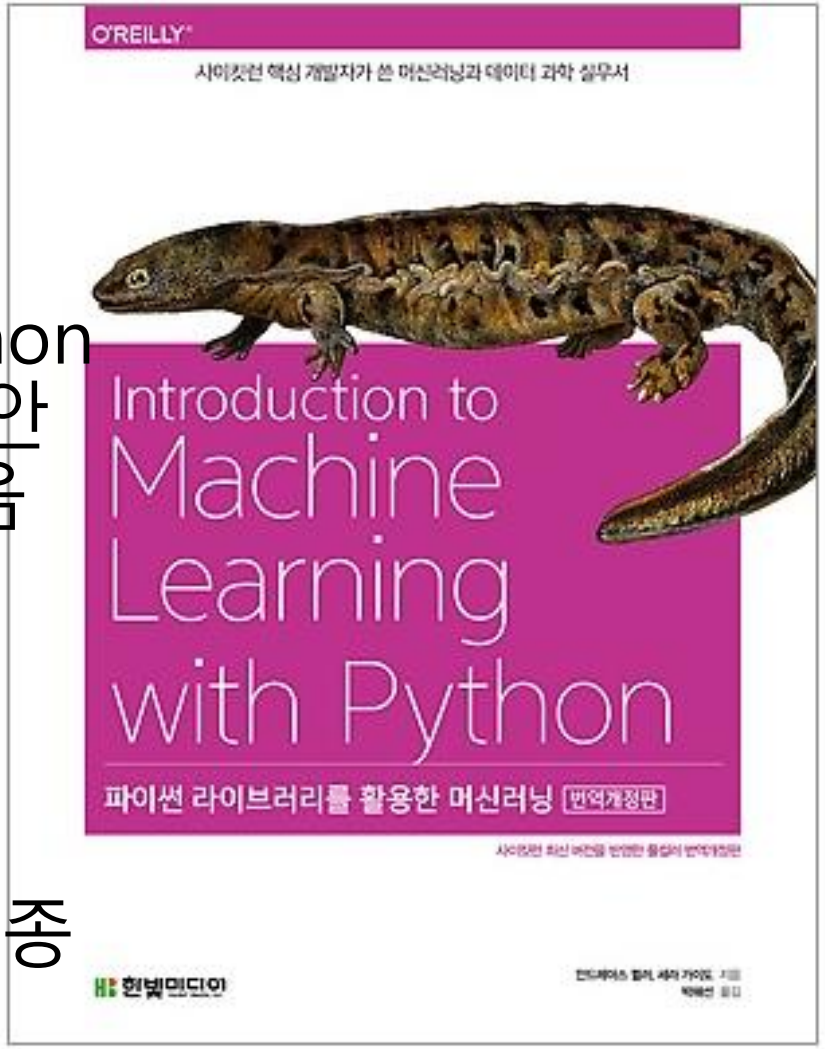
3. 수강생들에게 어떤 역량이 생길 것인가?

- 파이썬을 활용한 머신러닝 알고리즘 활용 역량
- 경진대회 참가를 통한 빅데이터 분석 역량

4. 강의 진행 안내

▪ 주교재

- Introduction to machine learning with Python = 파이썬 라이브러리를 활용한 머신러닝 / 안드레이스 뮐서, 세라 가이도 지음 ; 박해선 옮김. 번역개정판. 한빛미디어, 2019.



▪ 부교재

- 2018 데이터분석 준전문가: ADsP / 저자: 윤종식. 데이터 에듀, 2018.

- Kaggle 경진대회 참가

  - www.kaggle.com
  - 팀 단위로 참가 권장(3명)
  - 기존의 참가 팀 분석 방법 조사
  - 진행중인 competitions에 참가(지속적으로 업데이트 필요)
    - Digit Recognizer
    - Titanic: Machine Learning from Disaster

▪ 발표

- 1차 발표
  - 기간: 4주차(3/24) ~ 7주(4/14)
  - 내용: 기존 참가팀 결과 분석

- 2차 발표
  - 기간: 9주차(4/28) ~ 12주(5/19)
  - 내용: competition 참가 내용

사이킷런 scikit-learn 설치와 필수 라이브러리

▪ 기본 프로그램

- scikit-learn은 NumPy와 SciPy 사용
- 그래프 작성, 대화식 개발을 위한 라이브러리 필요

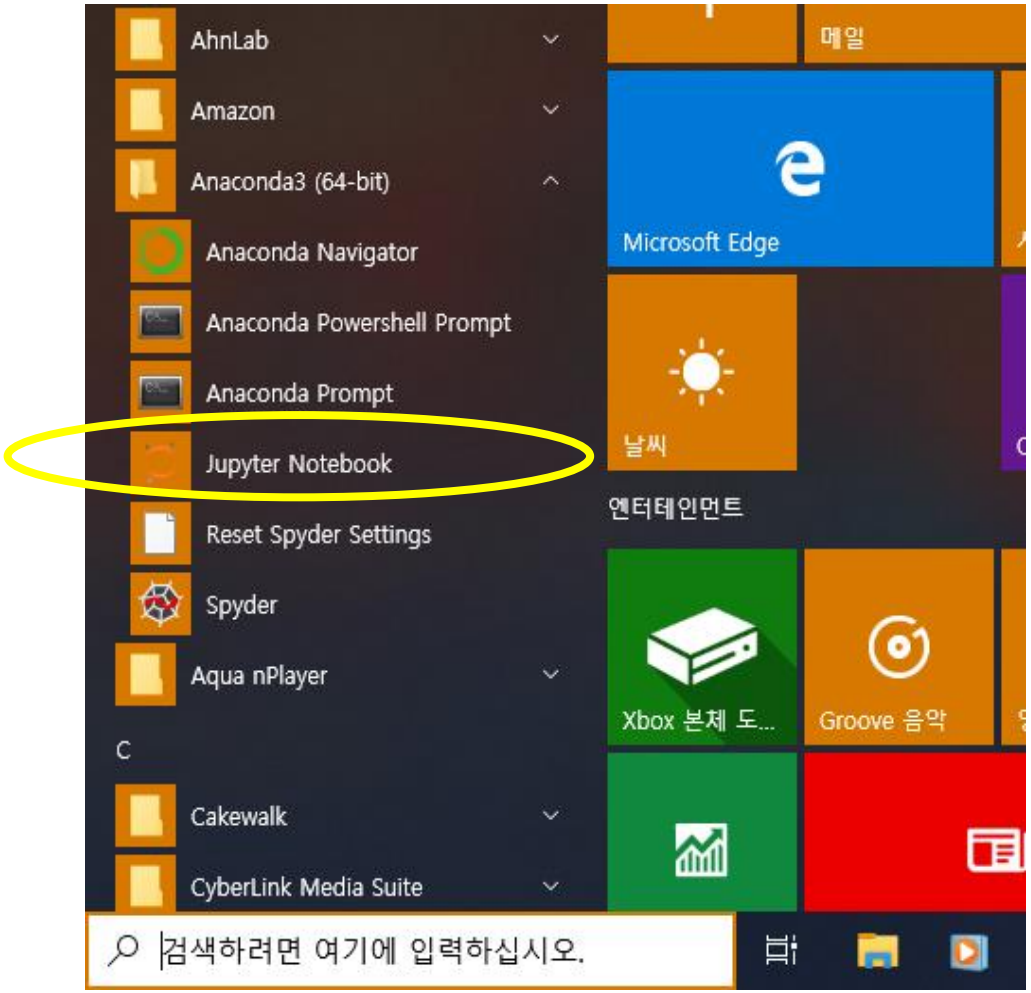
▪ Anaconda (https: //www.anaconda.com/)

- 주피터 노트북 (jupyter notebook): 브라우저에서 프로그램 코드 실행
- NumPy: 선형대수, 고수준의 수학 함수와 유사 pseudo 난수 생성기 포함

- SciPy: 고성능 선형대수, 함수 최적화, 신호 처리, 통계 분포 등 제공
- matplotlib: 그래프 라이브러리
- pandas: 데이터 처리와 분석
- mglean: 간단하게 그림을 그리거나 필요한 데이터를 바로 불러들이기 위해 사용

주피터 노트북

- 열기
  - anaconda navigator에서 Jupyter Notebook 선택 or;
  - 시작 화면에서 Jupyter Notebook 선택



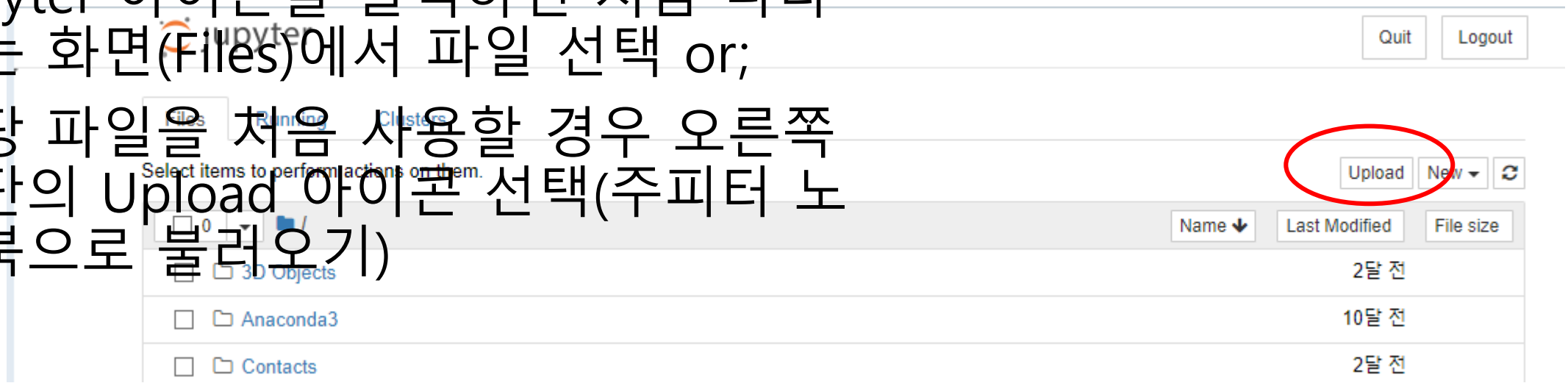
- Anaconda 를 설치한 후에 다음 내용을 듣기 바랍니다.

▪ 새로 만들기

- 오른쪽 상단의 New 아이콘 선택

▪ 파일 불러오기

- Jupyter 아이콘을 클릭하면 처음 나타나는 화면(Files)에서 파일 선택 or;
- 해당 파일을 처음 사용할 경우 오른쪽 상단의 Upload 아이콘 선택(주피터 노트북으로 불러오기)



NumPy

- 특징
  - 과학, 수학 계산을 위해 필수적인 라이브러리
  - 사이킷런에서 NumPy 배열이 기본 데이터 구조
  - 사용할 데이터는 모두 NumPy 배열로 변환
- NumPy 배열 예

```
In [1]: import numpy as np
        x=np.array([[1, 2, 3], [4, 5, 6]])
        print("x:\n", x)

        x:
         [[1 2 3]
         [4 5 6]]
```

SciPy

▪ 특징

- 과학 계산용 함수를 모아놓은 라이브러리
- 선형대수, 함수 최적화, 신호 처리, 통계 분포 등의 기능 제공
- 사이킷런은 알고리즘을 구현할 때, SciPy의 여러 함수 사용
- 그중 가장 중요한 기능이 scipy.sparse
  - scipy.sparse 모듈은 희소 행렬 기능 제공
  - 희소 행렬은 0을 많이 포함한 2차원 배열 저장에 사용

▪ Sparse matrix 희소 행렬 이해

◦ NumPy의 eye, zeros, ones 함수

```
In [2]: from scipy import sparse

        #대각선 원소는 1이고 나머지는 0인 2차원 NumPy 배열 작성
        eye = np.eye(4)
        print("NumPy 배열:\n", eye)

NumPy 배열:
 [[1. 0. 0. 0.]
 [0. 1. 0. 0.]
 [0. 0. 1. 0.]
 [0. 0. 0. 1.]]
```

◦ np.zeros((4,4))는?

◦ np.ones((4,4))는?

```
zeros:
 [[0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]]
ones:
 [[1. 1. 1. 1.]
 [1. 1. 1. 1.]
 [1. 1. 1. 1.]
 [1. 1. 1. 1.]]
```

▪ 희소 행렬 sparse.csr_matrix

- NumPy 배열을 CSR 포맷의 SciPy희소 행렬로 변환
  - 사이킷런에서 또 하나 데이터 표현 방법

```
In [3]: # NumPy 배열을 CSR 포맷의 SciPy 희소행렬로 변환
        # 0이 아닌 원소만 저장
        sparse_matrix = sparse.csr_matrix(eye)
        print("\nSciPy의 CSR 행렬:\n", sparse_matrix)

SciPy의 CSR 행렬:
   (0, 0)        1.0
  (1, 1)        1.0
  (2, 2)        1.0
  (3, 3)        1.0
```

```
NumPy 배열:
 [[1. 0. 0. 0.]
 [0. 1. 0. 0.]
 [0. 0. 1. 0.]
 [0. 0. 0. 1.]]
```

※ CSR: Compressed Sparse Row
행(가로)으로 재정리한 형태

- 왼쪽의 (0, 0)은 첫째 행, 첫째 열, (1, 1)은 둘째 행, 둘째 열 표시

- CRS(compressed sparse row)

$$A_{IJ} = \begin{bmatrix} 10 & 0 & 0 & 12 & 0 \\ 0 & 0 & 11 & 0 & 13 \\ 0 & 16 & 0 & 0 & 0 \\ 0 & 0 & 11 & 0 & 13 \end{bmatrix}$$

※ 행 압축정보 배열은
'최초시작행번호'
'시작행에서의 데이터 개수'
'두번째 행에서 데이터 누적 개수'
…
'마지막 행에서 데이터 누적 개수 '

데이터(A) =

10	12	11	13	16	11	13
(0,0)	(0,3)	(1,2)	(1,4)	(2,1)	(3,2)	(3,4)

열인덱스(JA) =

0	3	2	4	1	2	4
(0	)	(1	4)	(2)	(3	)

행압축정보(IA) =

0	2	4	5	7
(0)	(1)	(2)	(3)	(4)

▪ 희소 행렬: sparse.coo_matrix

◦ COO 포맷을 이용하여 희소 행렬 만들기

In [5]:
```
data = np.ones(4)
row_indices = np.arange(4)
col_indices = np.arange(4)
eye_coo = sparse.coo_matrix((data, (row_indices, col_indices)))
print("COO 표현:\n", eye_coo)
```

```
COO 표현:
   (0, 0)        1.0
  (1, 1)        1.0
  (2, 2)        1.0
  (3, 3)        1.0
```

matplotlib

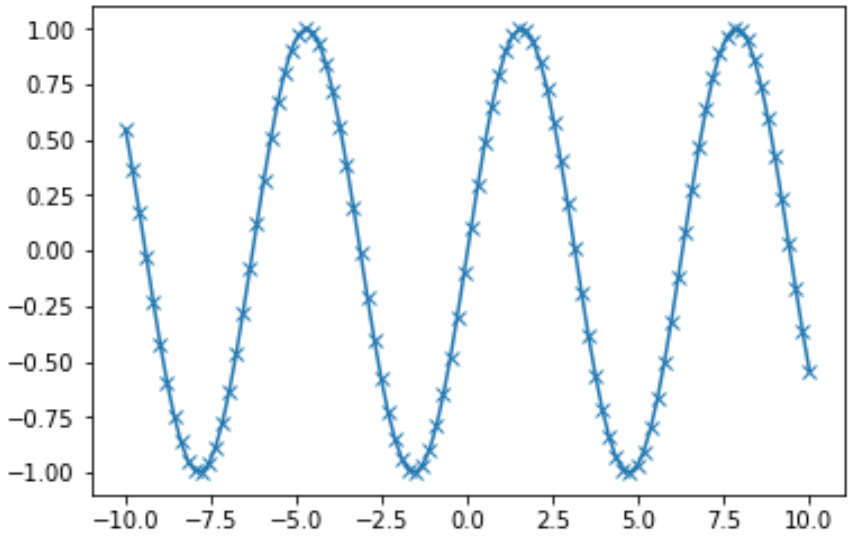
▪ 특징

- ◦ 과학 계산용 그래프 라이브러리
- ◦ %matplotlib inline 명령으로 브라우저에서 이미지 확인

In [6]:
```
%matplotlib inline
import matplotlib.pyplot as plt

# -10에서 10까지 100개의 간격으로 나뉘어진 배열 생성
x = np.linspace(-10, 10, 100)
# 사인 함수를 사용하여 y 배열을 생성
y = np.sin(x)
# plot 함수는 한 배열의 값을  다른 배열에 대응해서 선 그래프를 그림
plt.plot(x, y, marker="x")
```

Out [6]: [<matplotlib.lines.Line2D at 0x1e62ad0f828>]



pandas

- 특징
  - 데이터 처리와 분석을 위한 라이브러리
  - R의 data.frame과 유사

```
In [10]: import pandas as pd

         # 회원 정보가 들어간 간단한 데이터셋 생성
         data = {'Name' : ["John", "Anna", "Peter", "Linda"],
                 'Location' : ["New York",  "Paris", "Berlin", "London"],
                 'Age' : [22, 19, 21, 23]
                 }

         data_pandas = pd.DataFrame(data)
         # IPython.display는 주피터 노트북에서 Dataframe을 멋있게 출력해줌
         display(data_pandas)
```

	Name	Location	Age
0	John	New York	22
1	Anna	Paris	19
2	Peter	Berlin	21
3	Linda	London	23

▪ 질의 작성 display 함수 이용

In [11]:
```
# Age 열의 값이 20 이상인 모든 행을 선택
display(data_pandas[data_pandas.Age > 20])
```

	Name	Location	Age
0	John	New York	22
2	Peter	Berlin	21
3	Linda	London	23

mglearn

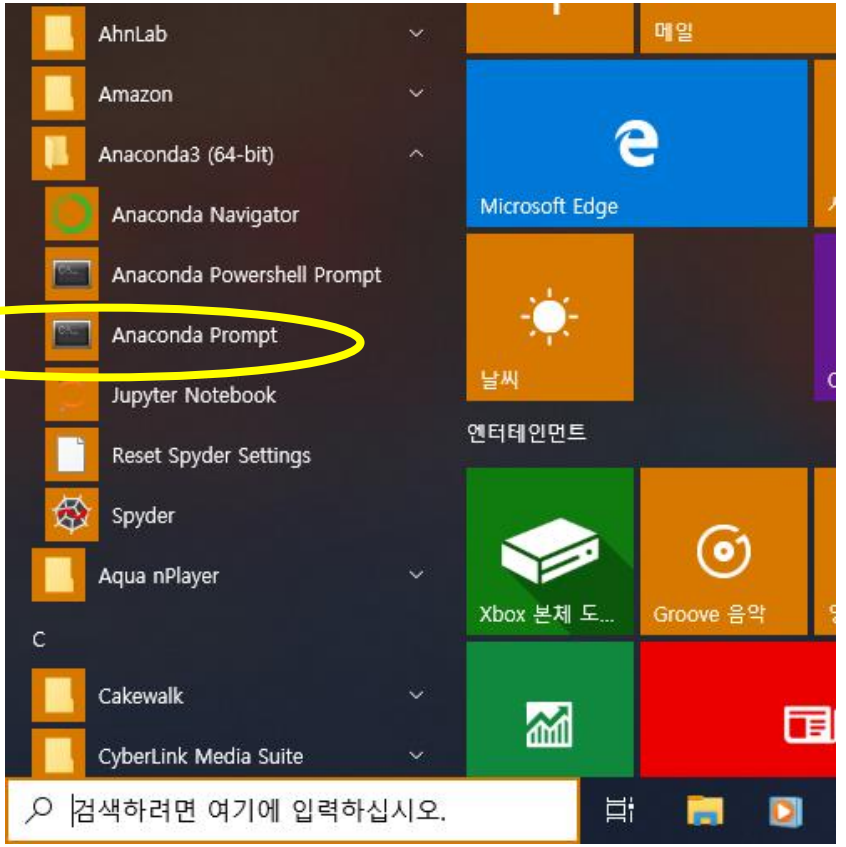
▪ 특징

- 간단한 그림을 그리거나 필요한 데이터를 바로 불러들이기 위해 사용

▪ 설치 방법

- 윈도우 시작화면에서 'Anaconda Prompt' 선택
- 아나콘다 프롬프트에서

  pip install mglearn 입력
- 사용할 경우 import mglearn 입력



라이브러리 버전 확인

In [4]:
```
import sys
print("Python 버전:", sys.version)

import pandas as pd
print("pandas 버전:", pd.__version__)

import matplotlib
print("matplotlib 버전:", matplotlib.__version__)

import numpy as np
print("NumPy 버전:", np.__version__)

import scipy as sp
print("SciPy 버전:", sp.__version__)

import IPython
print("IPython 버전:", IPython.__version__)

import sklearn
print("scikit-learn 버전:", sklearn.__version__)
```

```
Python 버전: 3.7.3 (default, Mar 27 2019, 17:13:21) [MSC v.1915 64 bit (AMD64)]
pandas 버전: 0.24.2
matplotlib 버전: 3.0.3
NumPy 버전: 1.16.2
SciPy 버전: 1.2.1
IPython 버전: 7.4.0
scikit-learn 버전: 0.20.3
```

오늘 수업 정리

- 강의에 대한 소개
- Kaggle 경진대회 안내
- 사이킷런 및 주요 라이브러리 안내

토론 및 과제

- 토론방
  - 캐글 경진대회 참가를 위한 팀 구성(3명)
  - 현재 파이썬에 대한 수준 이야기

- 과제
  - 오늘 이야기한 내용 중 마지막에 이야기한 Anaconda 에서 '라이브러리 버전' 화면을 캡쳐하여 과제에 올리기

- 마감: 3월 8일(일) 저녁 6시

다음 시간

- 데이터 분석 기획의 이해
  - 분석 방법론
  - 분석 과제 발굴