

STEP. 10

파이썬 라이브러리 개념

파이썬 라이브러리 활용



01 Python

- 과학 분야를 위한 표준 프로그래밍 언어
- MATLAB, R과 같은 도메인 특화 언어와 Java, C 같은 범용 언어의 장점을 고루 갖춘
- 통계, 머신러닝, 자연어, 이미지, 시각화 등을 포함한 풍부한 라이브러리를 지님
- 브라우저 기반의 인터랙티브 프로그래밍 환경인 Jupyter Notebook으로 쉽게 구현
- 파이썬 딥 러닝 라이브러리인 Tensorflow, Pytorch, Keras 등이 존재



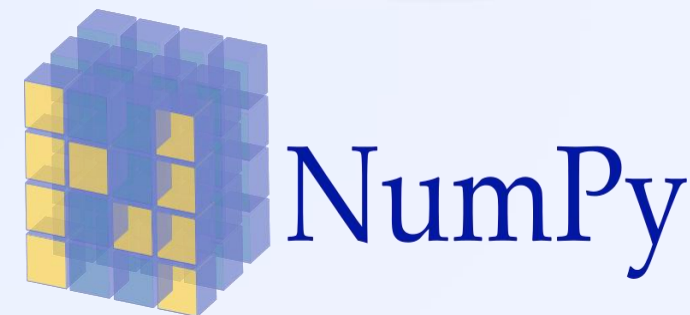
02 Python과 Machine Learning

- 파이썬은 뛰어난 확장성과 연계 호환성을 가짐
 - 분석 영역을 넘어 ML 기반의 다양한 Application 개발 용이
 - 기존 Application과 연계가 쉬움
- 딥 러닝 프레임워크들이 파이썬 기반으로 작성
 - 대부분의 딥 러닝 관련 튜토리얼, 설명 자료들이 파이썬으로 작성되어 제공
 - 현 시점에서 딥 러닝을 학습하기 위한 최적의 언어는 파이썬!



03 수치 계산 라이브러리(NumPy)

- Numerical Python의 약자로 대표적인 파이썬 기반 수치 해석 라이브러리
- 선형대수 연산에 필요한 다차원 배열과 배열 연산을 수행하는 다양한 함수 제공
- 설치 방법
 - `pip install numpy`
- 사용 방법
 - `import numpy as np`
- <http://www.numpy.org> 에서 numpy에 대한 다양한 설명 참고



04 데이터 처리 라이브러리(Pandas)

- 파이썬에서 사용하는 데이터 분석 라이브러리
- 행과 열로 이루어진 2차원 데이터를 효율적으로 가공할 수 있는 다양한 기능 제공
- 설치 방법
 - `pip install pandas`
- 사용 방법
 - `import pandas as pd`
- <https://pandas.pydata.org/> 에서 pandas에 대한 다양한 설명 참고



05 데이터 시각화 라이브러리(Matplotlib)

- matplotlib은 파이썬에서 데이터를 차트나 플롯으로 시각화하는 라이브러리
- matplotlib.pyplot 모듈의 함수를 이용하여 간편하게 그래프를 만들고 변화를 줄 수 있음

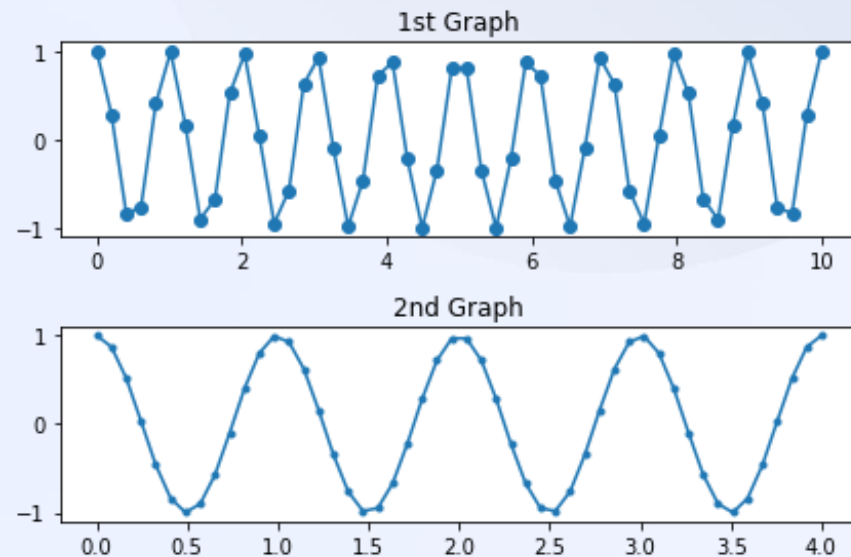
- 설치 방법

- `pip install matplotlib`

- 사용 방법

- `import matplotlib.pyplot as plt`

- <https://matplotlib.org/>에서 matplotlib에 대한 다양한 설명 참고

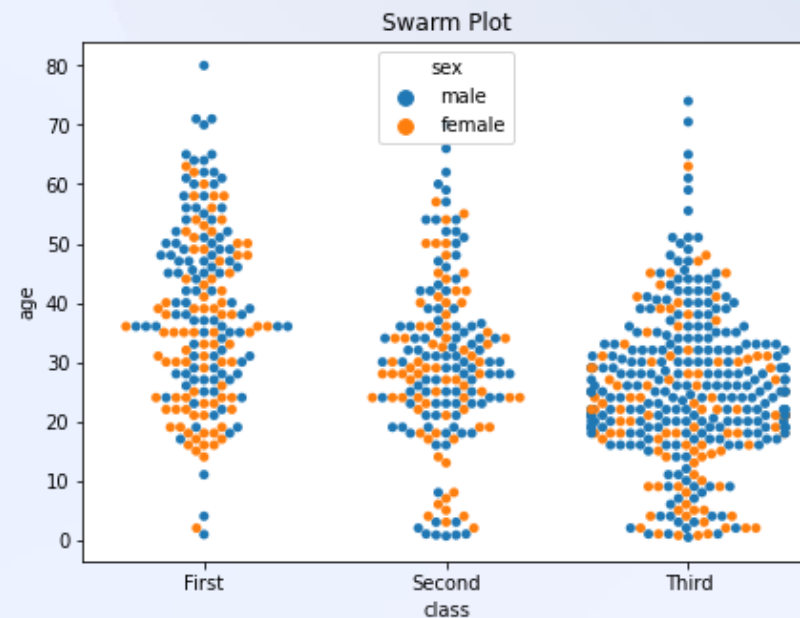


matplotlib



06 데이터 시각화 라이브러리(Seaborn)

- seaborn은 matplotlib 기반의 시각화 라이브러리
- 유익한 통계 기반 그래픽을 그리기 위한 고급 인터페이스를 제공
- 설치 방법
 - `pip install seaborn`
- 사용 방법
 - `import seaborn as sns`
- <https://seaborn.pydata.org/>에서 seaborn에 대한 다양한 설명 참고



seaborn



07 웹 데이터 수집 라이브러리(BeautifulSoup)

- HTML 및 XML에서 데이터를 쉽게 처리하는 파이썬 라이브러리
- HTML은 태그로 이루어져 있고, 수많은 공백과 변화하는 소스들 때문에 오류가 있을 가능성이 높지만 BeautifulSoup을 이용하면 이러한 오류를 잡아서 고친 후 데이터를 전달해줌

- 설치 방법
 - `pip install beautifulsoup4`
- 사용 방법
 - `from bs4 import BeautifulSoup`

종가	전일비	시가	고가	저가	거래량
70600.0	100.0	70700.0	71000.0	70300.0	4735065.0
70700.0	600.0	71300.0	71600.0	70600.0	11027606.0
71300.0	100.0	71500.0	72000.0	71300.0	10919239.0
71400.0	800.0	71700.0	71900.0	70900.0	12420710.0
70600.0	700.0	70200.0	70900.0	69900.0	10087450.0
...
84600.0	200.0	84800.0	85400.0	83400.0	22112205.0
84400.0	1400.0	84100.0	86400.0	83700.0	26302077.0
83000.0	1000.0	81700.0	83400.0	81000.0	28046832.0
82000.0	1700.0	84500.0	85000.0	82000.0	39615978.0
83700.0	1900.0	83200.0	85600.0	83200.0	31859808.0

[그림. 삼성전자 일별 시세 정보 웹 크롤링 결과]

- <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>에서 BeautifulSoup에 대한 다양한 설명 참고



08 머신러닝 라이브러리(Scikit Learn)

- 파이썬 기반의 머신러닝을 위한 가장 쉽고, 효율적인 개발 라이브러리
- 머신러닝은 곧 Scikit-Learn을 의미할 정도로 오랜 기간 파이썬 영역에서 인정 받아 옴
- R에 비견될 수 있는 Python 세계의 분석 라이브러리
- 머신러닝을 위한 다양한 알고리즘과 개발을 위한 프레임워크와 API를 제공



08 머신러닝 라이브러리(Scikit Learn)

- 회귀 모델, 분류 모델, 데이터 전처리, 성능 측정등 머신러닝에 필요한 도구를 두루 갖추

- 설치 방법

- `pip install scikit-learn`

- 사용 방법

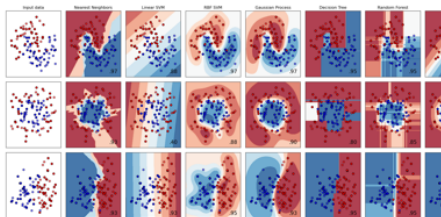
- `import scikit-learn`

Classification

Identifying which category an object belongs to.

Applications: Spam detection, image recognition.

Algorithms: SVM, nearest neighbors, random forest, and more...



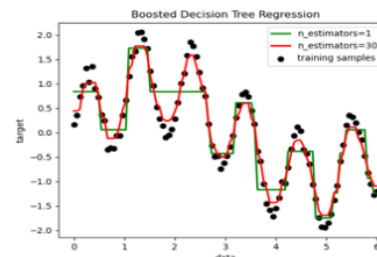
Examples

Regression

Predicting a continuous-valued attribute associated with an object.

Applications: Drug response, Stock prices.

Algorithms: SVR, nearest neighbors, random forest, and more...



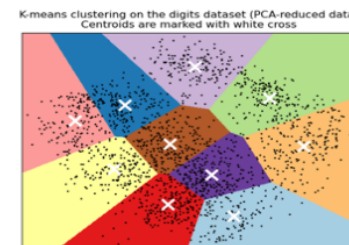
Examples

Clustering

Automatic grouping of similar objects into sets.

Applications: Customer segmentation, Grouping experiment outcomes

Algorithms: k-Means, spectral clustering, mean-shift, and more...



Examples

- <http://scikit-learn.org/stable/documentation>에서 scikit-learn에 대한 다양한 설명 참고



09 Python Jupyter Notebook

- 오픈 소스 기반의 웹 애플리케이션
- 파이썬으로 작성한 여러 개의 코드와 실행 결과를 하나의 문서처럼 관리 가능
- 즉, 프로그램 코드 + 결과 + 문서를 위한 대화식 개발 환경
- 기존의 파이썬 IDLE을 사용하는 것과 비교했을 때, 일부 코드만 실행하여 결과 확인 가능



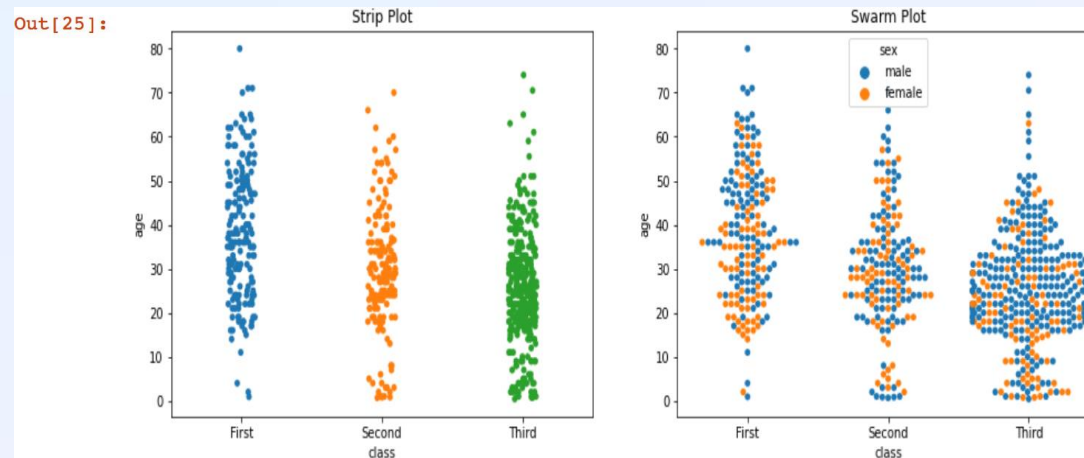
```
In [25]: # 그래프 객체 생성 (figure에 2개의 서브 플롯을 생성)
fig = plt.figure(figsize=(15, 5))
ax1 = fig.add_subplot(1, 2, 1)
ax2 = fig.add_subplot(1, 2, 2)

# 이산형 변수의 분포 - 데이터 분산 미고려
# x축 변수, y축 변수, 데이터 셋, axe 객체(1번째 그래프)
sns.stripplot(x='class', y='age', data=titanic, ax=ax1)

# 이산형 변수의 분포 - 데이터 분산 고려 (중복 X)
# x축 변수, y축 변수, 데이터 셋, axe 객체(2번째 그래프), 성별로 색상 구분
sns.swarmplot(x='class', y='age', data=titanic, ax=ax2, hue='sex')

# 차트 제목 표시
ax1.set_title('Strip Plot')
ax2.set_title('Swarm Plot')

plt.show()
```



10 Google Colaboratory (Colab)

- Google Colaboratory = Google Drive + Jupyter Notebook
- 구글 코랩은 주피터 노트북 기반의 오픈소스 프로젝트
- 구글 코랩의 장점
 - 대부분의 파이썬 패키지들이 설치 되어 있음
 - 구글 아이디와 인터넷만 있다면 하드웨어와 장소에 구애받지 않고 코딩 가능
 - 구글 드라이브와 연동하여 파일 불러오기
 - 무료로 GPU도 사용할 수 있어서 고성능 딥러닝 프로젝트 가능
 - 주피터 노트북 문서를 여러 사람이 동시에 열어서 함께 편집 가능

colab

