



The Apprentice Project

Lec03: Python Programming For the AI

충북대학교

문성태 (지능로봇공학과)

stmoon@cbnu.ac.kr

01

Introduction

AI 발전의 주 요인

- ❖ GPU
- ❖ Big Data
- ❖ Algorithm

With

Python and Open Source

Why Python?

01

Colab

What is google colab?



Google Colaboratory


- ❖ 구글이 제공하는 클라우드 기반 Jupyter Notebook
- ❖ 웹 브라우저에서 파이썬 코드를 작성하고 실행할 수 있는 웹 에디터
- ❖ 구글 계정만 있으면 무료로 사용 가능

<https://colab.research.google.com/>

Features

- ❖ Jupyter와 같이 Cell 기반으로 코드와 텍스트를 구분하여 사용
- ❖ 별도 파이썬 설치 필요 없음
- ❖ 데이터 분석에 활용되는 numpy, pandas, scikit-learn, matplotlib 등이 기본 설치됨
- ❖ GPU를 무료로 사용 가능
- ❖ 사용이 간편하여 교육용으로 활용 적합

Colab 초기화면

 Colaboratory에 오신 것을 환영합니다
파일 수정 보기 삽입 런타임 도구 도움말

목록

시작하기

데이터 과학

머신러닝

추가 리소스

주요 예시


섹션

+ 코드 + 텍스트

Drive로 복사

Colab 시작 페이지

Colab에 이미 익숙하다면 이 동영상을 통해 양방향 테이블, 코드 실행 기록 보기, 명령어 팔레트에 관해 알아보세요.



Colab이란?

Colaboratory(줄여서 'Colab'이라고 함)을 통해 브라우저 내에서 Python 스크립트를 작성하고 실행할 수 있습니다.

- 구성이 필요하지 않음
- 무료로 GPU 사용
- 간편한 공유

학생이든, 데이터 과학자든, AI 연구원이든 Colab으로 업무를 더욱 간편하게 처리할 수 있습니다. [Colab 소개 영상](#)에서 자세한 내용을 확인하거나 아래에서 시작해 보세요.

시작하기

지금 읽고 계신 문서는 정적 웹페이지가 아니라 코드를 작성하고 실행할 수 있는 대화형 환경인 **Colab 메모장**입니다. 예를 들어 다음은 값을 계산하여 변수로 저장하고 결과를 출력하는 간단한 Python 스크립트가 포함된 코드 셀입니다.

```
[ ] seconds_in_a_day = 24 * 60 * 60
seconds_in_a_day
```

단축키

- ❖ Ctrl + M A = 코드 셀 위에 삽입
- ❖ Ctrl + M B = 코드 셀 아래 삽입
- ❖ Ctrl + M D = 셀 지우기
- ❖ Ctrl + M Y = 코드 셀로 변경
- ❖ Ctrl + M M = 마크다운 셀로 변경
- ❖ Ctrl + M Z = 실행 취소

주의 사항

- ❖ 동시에 사용할 수 있는 구글 클라우드의 가상 서버 개수: 5개
 - 5개 이상의 노트북을 여는 경우 실행 중인 다른 노트북을 제거해야 함
- ❖ 12시간 이상 실행 불가능

02

numpy

Numpy

- 과학 계산을 위한 라이브러리
- 행렬/배열 처리 및 연산
 - Matrix 와 Vector 같은 Array 연산의 사실상의 표준

Numpy vs List

- 리스트는 여러 개의 값들을 저장할 수 있는 자료구조
 - 다양한 자료형의 데이터를 여러 개 저장할 수 있으며 데이터를 변경하거나 추가, 제거 가능

```
>>> scores = [10, 20, 30, 40, 50, 60]
```

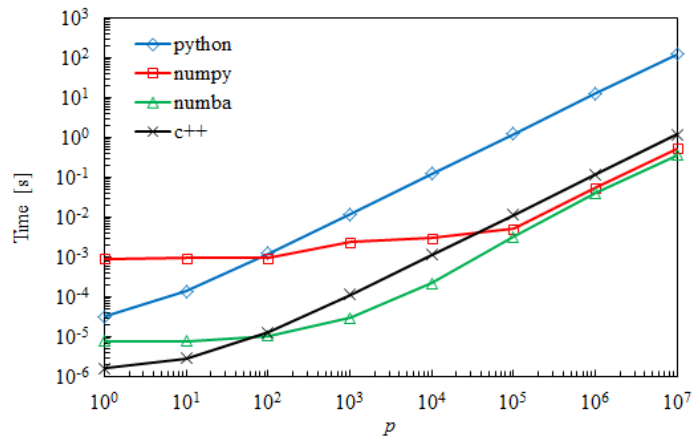
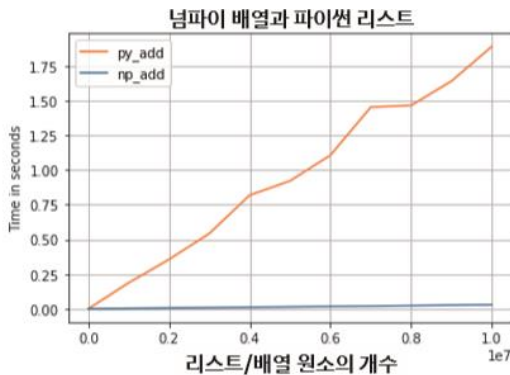
- 리스트 한계점
 - 리스트 간 연산 기능 부족
 - 연산 속도 저하
- 따라서, 데이터 과학자들은 리스트 대신 Numpy 선호

Numpy vs List

❖ Numpy

- 대용량의 배열과 행렬 연산을 빠르게 수행
- 고차원적인 수학 연산자와 함수 포함

❖ numpy의 배열은 주황색으로 표시된 파이썬의 리스트에 비하여 처리속도가 매우 빠름



Numpy vs List

- ❖ numpy 는 성능이 우수한 ndarray 객체 제공
- ❖ ndarray의 장점

- ndarray 는 C 언어에 기반한 배열 구조이므로 메모리를 적게 차지하고 속도가 빠르다.
- ndarray 를 사용하면 배열과 배열 간에 수학적 연산을 적용할 수 있다.
- ndarray 는 고급 연산자와 풍부한 함수들을 제공한다.

03

pandas

Pandas 소개

- **Panel Data System**
- 데이터 분석을 위한 파이썬 기반의 라이브러리
 - 특히, 2차원 데이터를 효율적으로 가공 및 처리할 수 있는 강력한 라이브러리

Pandas 특징

1. 빠르고 효율적이며 다양한 표현력을 갖춘 자료구조.

실세계 데이터 분석을 위해 만들어진 파이썬 패키지

2. 다양한 형태의 데이터에 적합

이종heterogeneous 자료형의 열을 가진 테이블 데이터

시계열 데이터

레이블을 가진 다양한 행렬 데이터

다양한 관측 통계 데이터

3 핵심 구조

시리즈Series : 1차원 구조를 가진 하나의 열

데이터프레임DataFrame : 복수의 열을 가진 2차원 데이터

4. 판다스가 잘 하는 일

결측 데이터 처리

데이터 추가 삭제 (새로운 열의 추가, 특정 열의 삭제 등)

데이터 정렬과 다양한 데이터 조작

Pandas 기능

❖ 데이터 보기 및 검사

- `mean()`로 모든 열의 평균을 계산할 수 있다.
- `corr()`로 데이터 프레임의 열 사이의 상관 관계를 계산할 수 있다.
- `count()`로 각 데이터 프레임 열에서 null이 아닌 값의 개수를 계산할 수 있다.

❖ 필터, 정렬 및 그룹화

- `sort_values()`로 데이터를 정렬할 수 있다.
- 조건을 사용하여 열을 필터링할 수 있다.
- `groupby()`를 이용하여 기준에 따라 몇 개의 그룹으로 데이터를 분할할 수 있다.

❖ 데이터 정제

- 데이터의 누락 값을 확인할 수 있다.
- 특정한 값을 다른 값으로 대체할 수 있다.

Pandas 데이터 구조

- ❖ 판다스는 데이터 저장을 위하여 다음과 같은 2가지의 기본 데이터 구조 제공
 - 데이터 구조는 모두 넘파이 배열을 이용하여 구현 \Rightarrow 속도가 빠르다
- ❖ 모든 데이터 구조는 값을 변경할 수 있으며, 시리즈를 제외하고는 크기도 변경할 수 있다. 각 행과 열은 이름이 부여되며,
- ❖ 행의 이름을 인덱스 index,
- ❖ 열의 이름을 컬럼스 columns

데이터 구조	차원	설명
시리즈	1	레이블이 붙어있는 1차원 벡터
데이터프레임	2	행과 열로 되어있는 2차원 테이블, 각 열은 시리즈로 되어 있다.

Pandas 데이터 구조

- ❖ Series : 각 열(Column) 단위의 데이터
- ❖ Dataframe : 각 열 단위(Series)가 모여 된 하나의 표
- ❖ Index : Series, Dataframe을 생성하면 인덱싱 번호가 따라다닌다. 인덱스는 Series가 아니다. 위 이미지에는 숫자로 되어있지만 내가 원하는 인덱스 형태로 변경할 수 있다

	이름	국어	영어	수학
0	YB	90	100	100
1	SW	70	80	70
2	EJ	60	80	70
3	HJ	50	80	80
.
.
.
.
	Sereies	Sereies	Sereies	Sereies



Dataframe

Series

pandas.Series

```
class pandas.Series(data=None, index=None, dtype=None, name=None,  
copy=False, fastpath=False) \[source\]
```

One-dimensional ndarray with axis labels (including time series).

Labels need not be unique but must be a hashable type. The object supports both integer- and label-based indexing and provides a host of methods for performing operations involving the index. Statistical methods from ndarray have been overridden to automatically exclude missing data (currently represented as NaN).

Operations between Series (+, -, /, *, **) align values based on their associated index values— they need not be the same length. The result index will be the sorted union of the two indexes.

DataFrame

pandas.DataFrame

```
class pandas.DataFrame(data=None, index=None, columns=None,  
dtype=None, copy=None)
```

[\[source\]](#)

Two-dimensional, size-mutable, potentially heterogeneous tabular data.

Data structure also contains labeled axes (rows and columns). Arithmetic operations align on both row and column labels. Can be thought of as a dict-like container for Series objects.

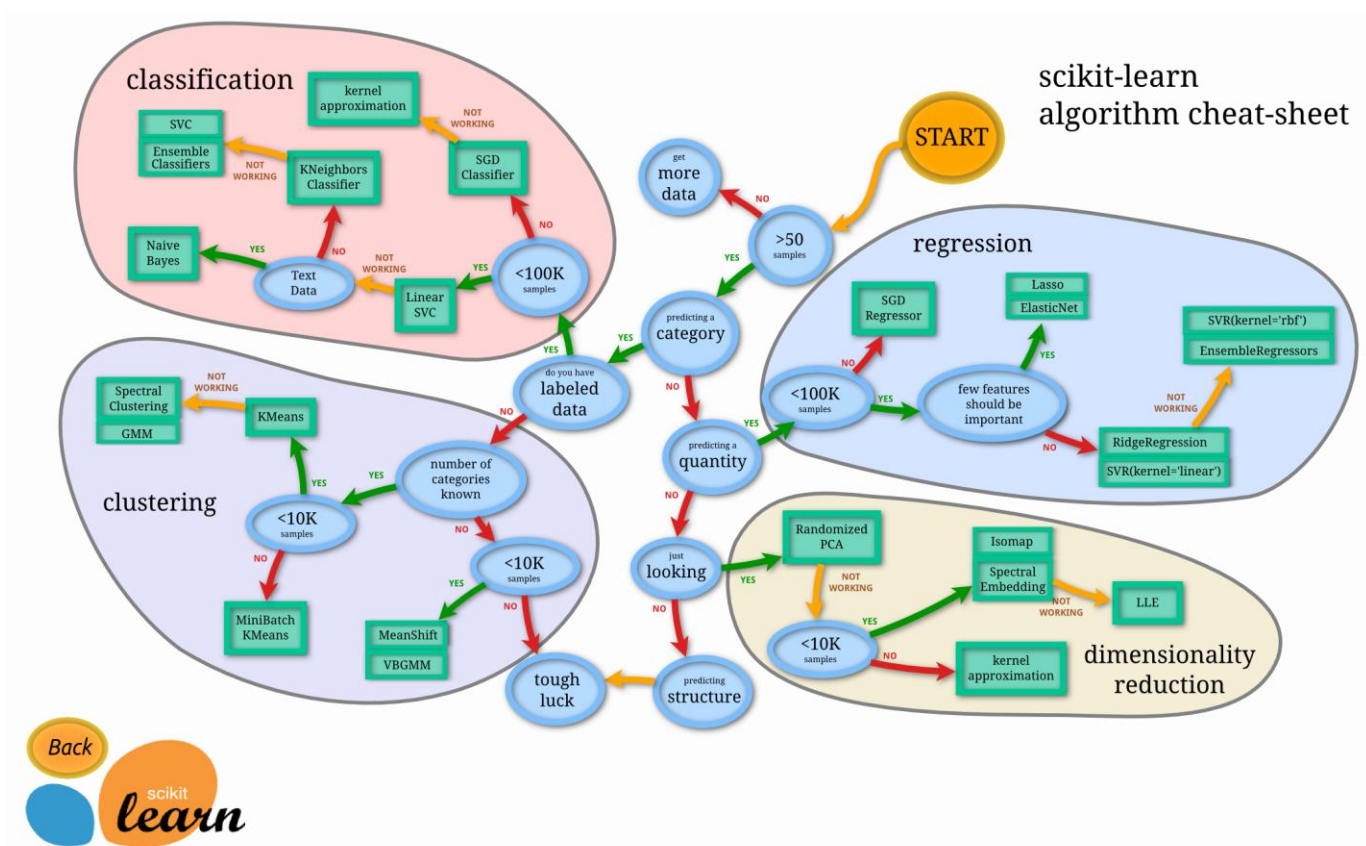
The primary pandas data structure.

04

Scikit-learn

- ❖ Scikits.learn 또는 sklearn 이라고도 함
- ❖ 2007년 구글 썸머 코드에서 처음 구현
- ❖ 파이썬으로 구현된 가장 유명한 기계 학습 오픈 소스 라이브러리
 - 개인, 비즈니스 관계없이 누구나 무료로 사용가능
- ❖ 많은 머신러닝 알고리즘이 구현되어 있고, 샘플 데이터 셋(토이 데이터 셋)이 포함되어 있음
 - 초심자가 기계학습을 배우기 시작할 때 적합한 라이브러리

기능



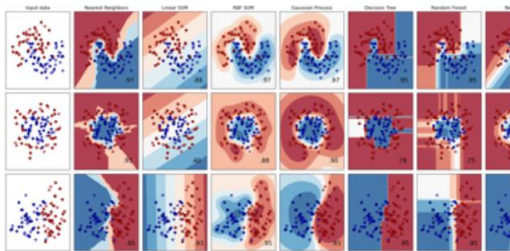
기능

Classification

Identifying which category an object belongs to.

Applications: Spam detection, image recognition.

Algorithms: Gradient boosting, nearest neighbors, random forest, logistic regression, and more...

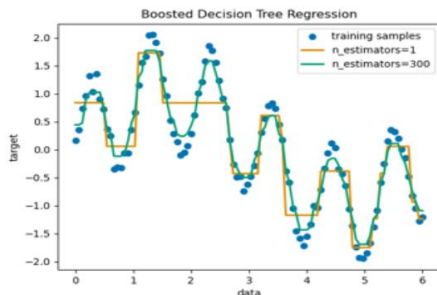


Regression

Predicting a continuous-valued attribute associated with an object.

Applications: Drug response, Stock prices.

Algorithms: Gradient boosting, nearest neighbors, random forest, ridge, and more...

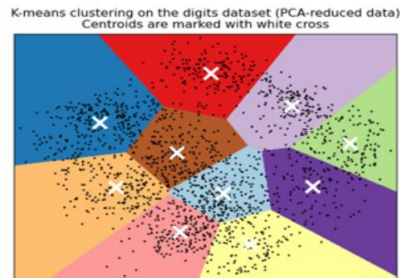


Clustering

Automatic grouping of similar objects into sets.

Applications: Customer segmentation, Grouping experiment outcomes

Algorithms: k-Means, HDBSCAN, hierarchical clustering, and more...



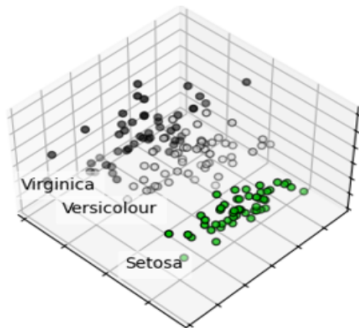
기능

Dimensionality reduction

Reducing the number of random variables to consider.

Applications: Visualization, Increased efficiency

Algorithms: PCA, feature selection, non-negative matrix factorization, and more...

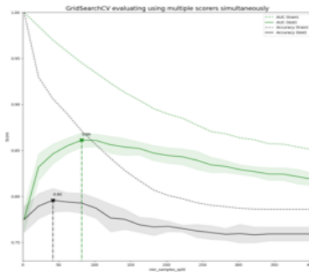


Model selection

Comparing, validating and choosing parameters and models.

Applications: Improved accuracy via parameter tuning

Algorithms: grid search, cross validation, metrics, and more...

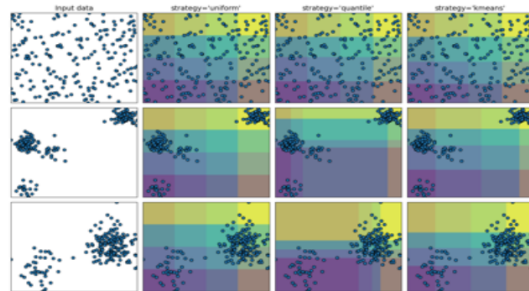


Preprocessing

Feature extraction and normalization.

Applications: Transforming input data such as text for use with machine learning algorithms.

Algorithms: preprocessing, feature extraction, and more...



방식

