2주차

쿠글 9기 김나연, 이시은

〈CONTENTS〉

Kuggle

1. EDA

1. EDA

EDA의 의미

Exploratory Data Analysis의 약자로 탐색적 데이터 분석이라는 의미
말 그대로 수집한 데이터를 다양한 방면에서 탐색하고 이해하는
과정을 말함
쉽게 말해, 그래프나 통계적 방법을 사용하여
데이터의 특징을 알아보는 것

데이터 분석

탐색적 데이터 분석 (Exploratory Data Analysis, EDA)
확증적 데이터 분석 (Confirmatory Data Analysis, CDA)

두 접근법의 근본적인 차이는 EDA는 데이터를 보고 가설을 만들어내는 반면, CDA는 기존의 가설이 맞는지를 데이터를 통해 확인하는 것

1. EDA

사용되는 데이터

일변량 데이터

$$Y = \beta_0 + \beta_1 X + \varepsilon$$

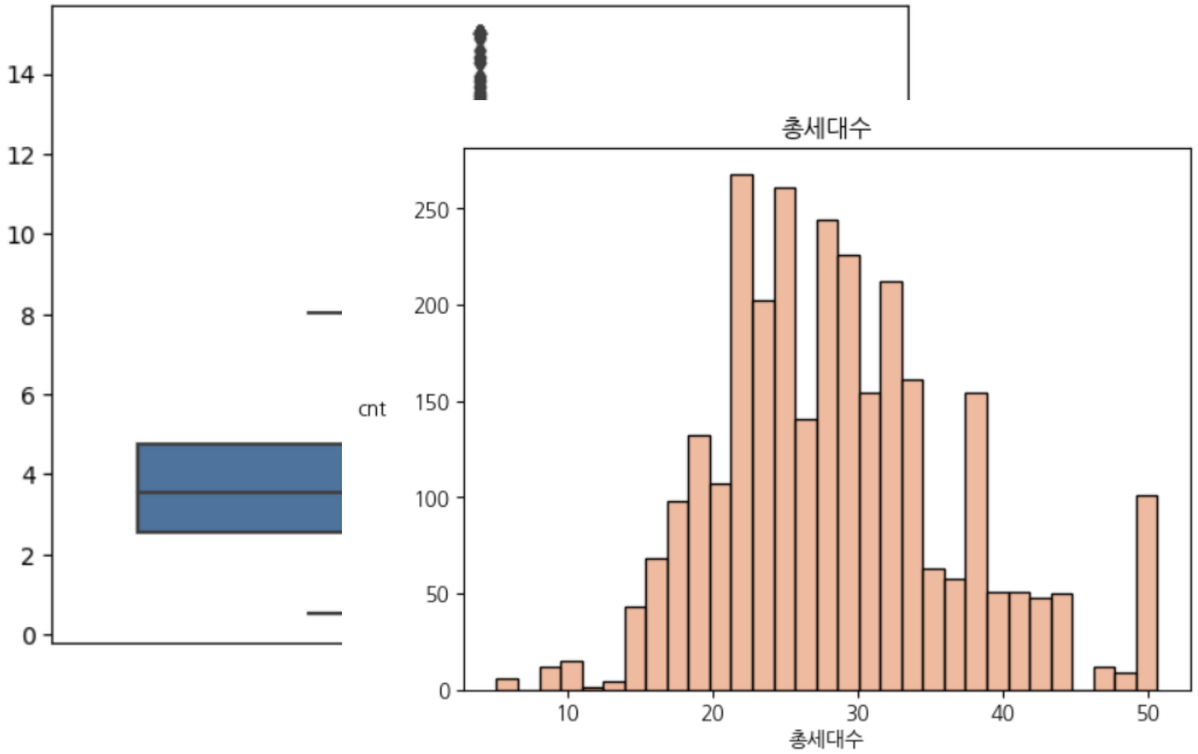
다변량 데이터

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p + \varepsilon$$

	AveRooms	HouseAge	Population	MedInc
0	6.984127	41.0	322.0	8.3252
1	6.238137	21.0	2401.0	8.3014
2	8.288136	52.0	496.0	7.2574
3	5.817352	52.0	558.0	5.6431
4	6.281853	52.0	565.0	3.8462
...	...	...	...	...
20635	5.045455	25.0	845.0	1.5603
20636	6.114035	18.0	356.0	2.5568
20637	5.205543	17.0	1007.0	1.7000
20638	5.329513	18.0	741.0	1.8672
20639	5.254717	16.0	1387.0	2.3886

1. EDA

EDA의 종류



	year	mnth_num	day	day2_num	Rating
count	701.000000	701.000000	701.000000	701.000000	701.000000
mean	2016.965763	6.857347	15.475036	2.958631	1.874465
std	3.231359	3.373369	8.956308	1.960248	1.400486
min	2010.000000	1.000000	1.000000	0.000000	1.000000
25%	2015.000000	4.000000	7.000000	1.000000	1.000000
50%	2017.000000	7.000000	15.000000	3.000000	1.000000
75%	2019.000000	10.000000	23.000000	5.000000	2.000000
max	2023.000000	12.000000	31.000000	6.000000	5.000000

Graphic(시각화)

데이터를 한눈에 파악하여 대략적인 형태 파악 가능

Non-Graphic(비시각화)

정확한 값을 파악하기 좋음

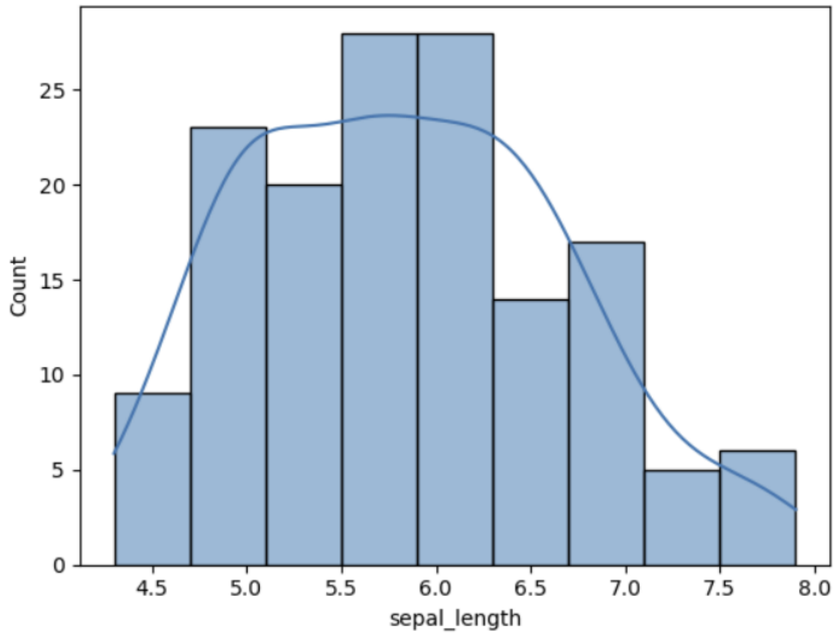
1. EDA

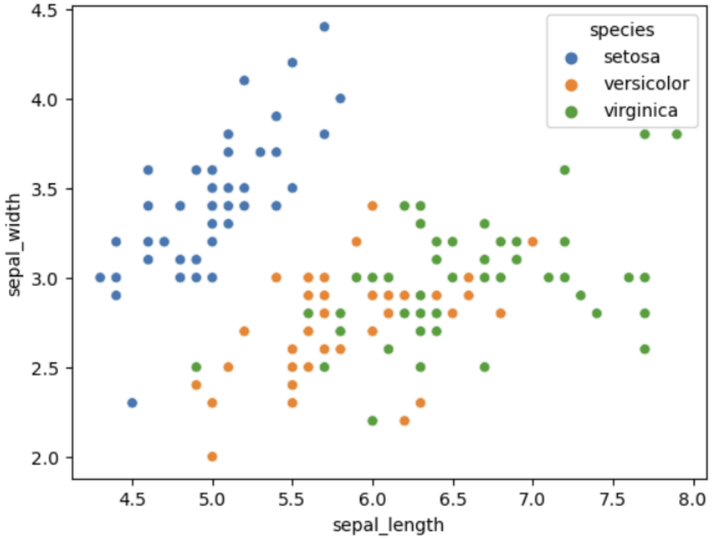
EDA의 종류 – 시각화

```
import seaborn as sns
import matplotlib.pyplot as plt
iris = sns.load_dataset('iris')
```

<u># 히스토그램</u>

```
sns.histplot(data=iris, x='sepal_length', kde=True)
plt.show()
```





<u># 산점도</u>

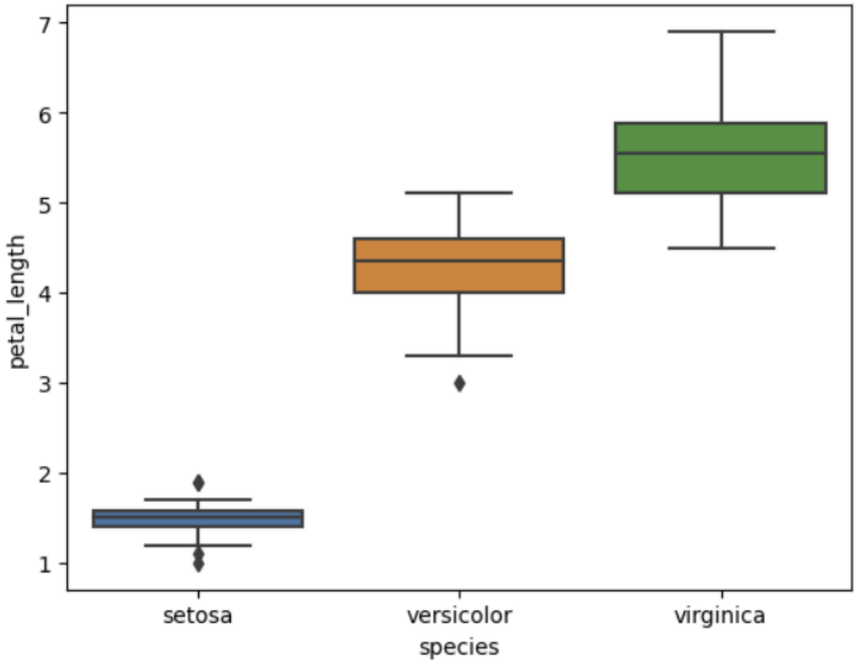
```
sns.scatterplot(data=iris, x='sepal_length', y='sepal_width', hue='species')
plt.show()
```

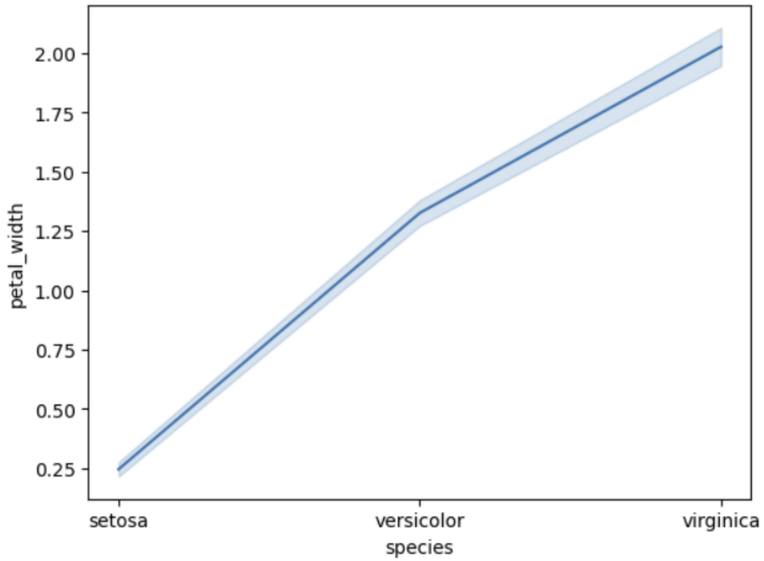
1. EDA

EDA의 종류 – 시각화

상자 그림

```
sns.boxplot(data=iris, x='species', y='petal_length')
plt.show()
```





선 그래프

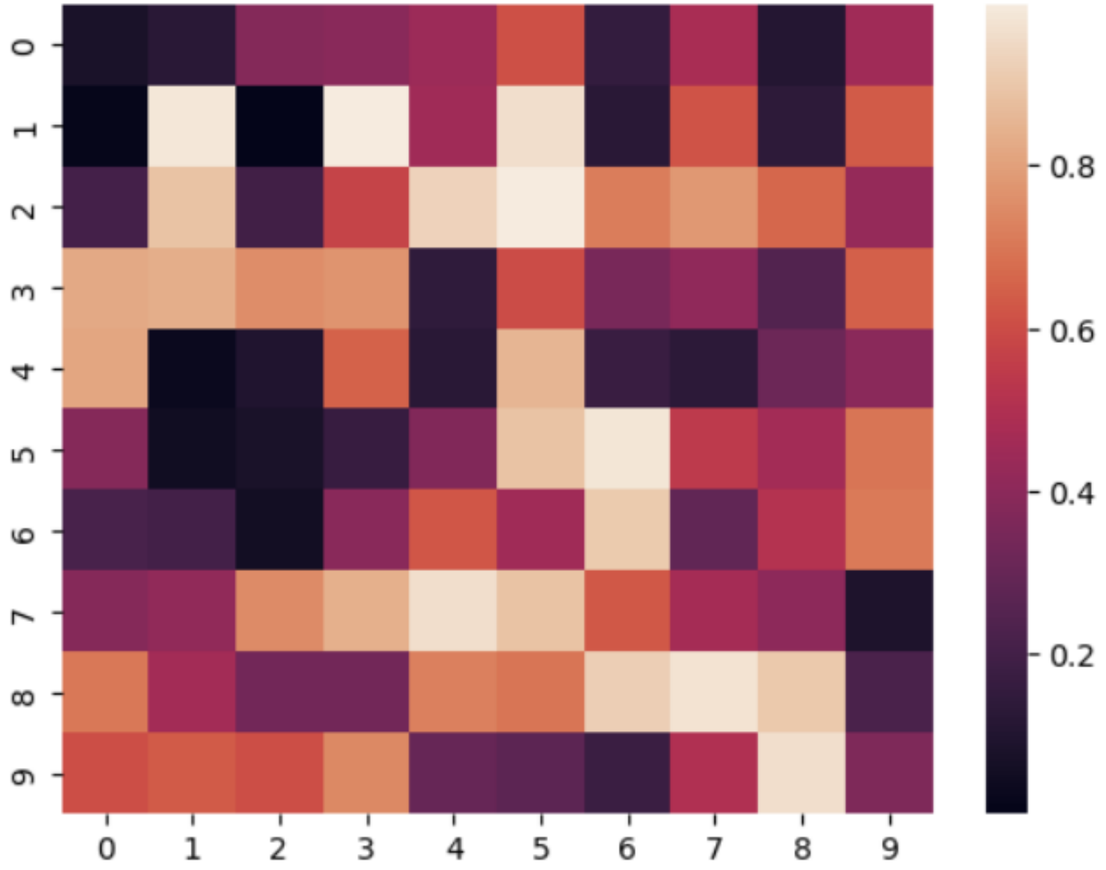
```
sns.lineplot(data=iris, x='species', y='petal_width')
plt.show()
```

1. EDA

EDA의 종류 – 시각화

히트맵
```
data = np.random.rand(10, 10)
sns.heatmap(data)
plt.show()
```

```
array([[0.07453047, 0.12261998, 0.38063424, 0.39087818, 0.440226  ,
        0.60662033, 0.15276928, 0.47884783, 0.10887533, 0.45397978],
       [0.01226453, 0.98604108, 0.00418287, 0.99548895, 0.45283595,
        0.96335645, 0.12080988, 0.61707846, 0.13391249, 0.6363761 ],
       [0.20601292, 0.88701217, 0.19254633, 0.57454087, 0.93103778,
        0.99843577, 0.71196851, 0.77988546, 0.66066524, 0.42141097],
       [0.82322825, 0.83398099, 0.75311992, 0.76865551, 0.14349425,
        0.59602925, 0.34916952, 0.40951015, 0.24396424, 0.64752625],
       [0.81575907, 0.03105539, 0.09480986, 0.65088839, 0.1230626 ,
        0.85224505, 0.16876542, 0.12953086, 0.3117928 , 0.39396023],
       [0.38360325, 0.0490771 , 0.07489852, 0.16635815, 0.3711949 ,
        0.88779955, 0.98031874, 0.54180626, 0.46205841, 0.69401148],
       [0.21756637, 0.19997419, 0.05429885, 0.39137809, 0.62533544,
        0.45361315, 0.90820385, 0.28519084, 0.51214888, 0.70460641],
       [0.38142432, 0.41468832, 0.7479758 , 0.83738788, 0.96023926,
        0.88743042, 0.6282084 , 0.46777864, 0.40194976, 0.08684907],
       [0.70055903, 0.45998378, 0.32834933, 0.33023501, 0.71983995,
        0.69400233, 0.91842798, 0.97141952, 0.90182574, 0.21988347],
       [0.6058977 , 0.6358755 , 0.60268574, 0.74135176, 0.29704679,
        0.27036961, 0.17292534, 0.50111096, 0.96190745, 0.36529508]])
```



1. EDA

EDA의 과정

- 연구 목적 및 분석 데이터 확인
- 데이터 살펴보기 (이상치 결측치 확인 등)
- 데이터 시각화를 통해 분포 확인
- 데이터 속성 간의 관계에 초점을 맞추어, 개별 데이터 속성 관찰에서 찾아내지 못했던 패턴 발견

Kuggle

2. 머신러닝 & 딥러닝

2. 머신러닝 & 딥러닝

머신러닝의 의미

애플리케이션을 수정하지 않고도 데이터를 기반으로 패턴을 학습하고 결과를 예측하는 알고리즘 기법

데이터를 기반으로 통계적인 신뢰도를 강화하고 예측 오류를 최소화하기 위한 다양한 수학적 기법을 적용해 데이터 내의 패턴을 스스로 인지하고 신뢰도 있는 예측 결과를 도출해 냄

2. 머신러닝 & 딥러닝

머신러닝의 의미

데이터 분석 영역

머신러닝 기반의 예측 분석으로 재편되고 있음
많은 데이터 분석가와 데이터 과학자가 머신러닝 알고리즘 기반의 새로운 예측 모델을 이용해 더욱 정확한 예측 및 의사 결정을 도출함

2. 머신러닝 & 딥러닝

기존 프로그래밍과의 차이

<기존 프로그래밍>
데이터를 입력해서 답을 구하는 데 초점이 맞춰져 있음

데이터
|
V
일반 프로그래밍
|
V
답

<머신러닝>
데이터 안에서 규칙을 발견하고 그 규칙을 새로운 데이터에 적용해서 새로운 결과를 도출하는 데 초점이 맞춰져 있음

데이터 + 답
|
V
머신러닝
|
V
규칙

=> 머신러닝은 기존 데이터를 이용해 아직 일어나지 않은 미지의 일을 예측하기 위해 만들어진 기법

2. 머신러닝 & 딥러닝

머신러닝의 분류

- 지도학습

기계 학습 중 컴퓨터가 입력값과 그에 따른 출력값이 있는 데이터를 이용하여 주어진 입력에 맞는 출력을 찾는 학습 방법

분류 / 회귀 / 추천 시스템 / 시각(음성) 감지(인지) / 텍스트 분석, NLP

- 비지도학습

계 학습 중 컴퓨터가 입력값만 있는 훈련 데이터를 이용하여 입력들의 규칙성을 찾는 학습 방법

클러스터링 / 차원 축소

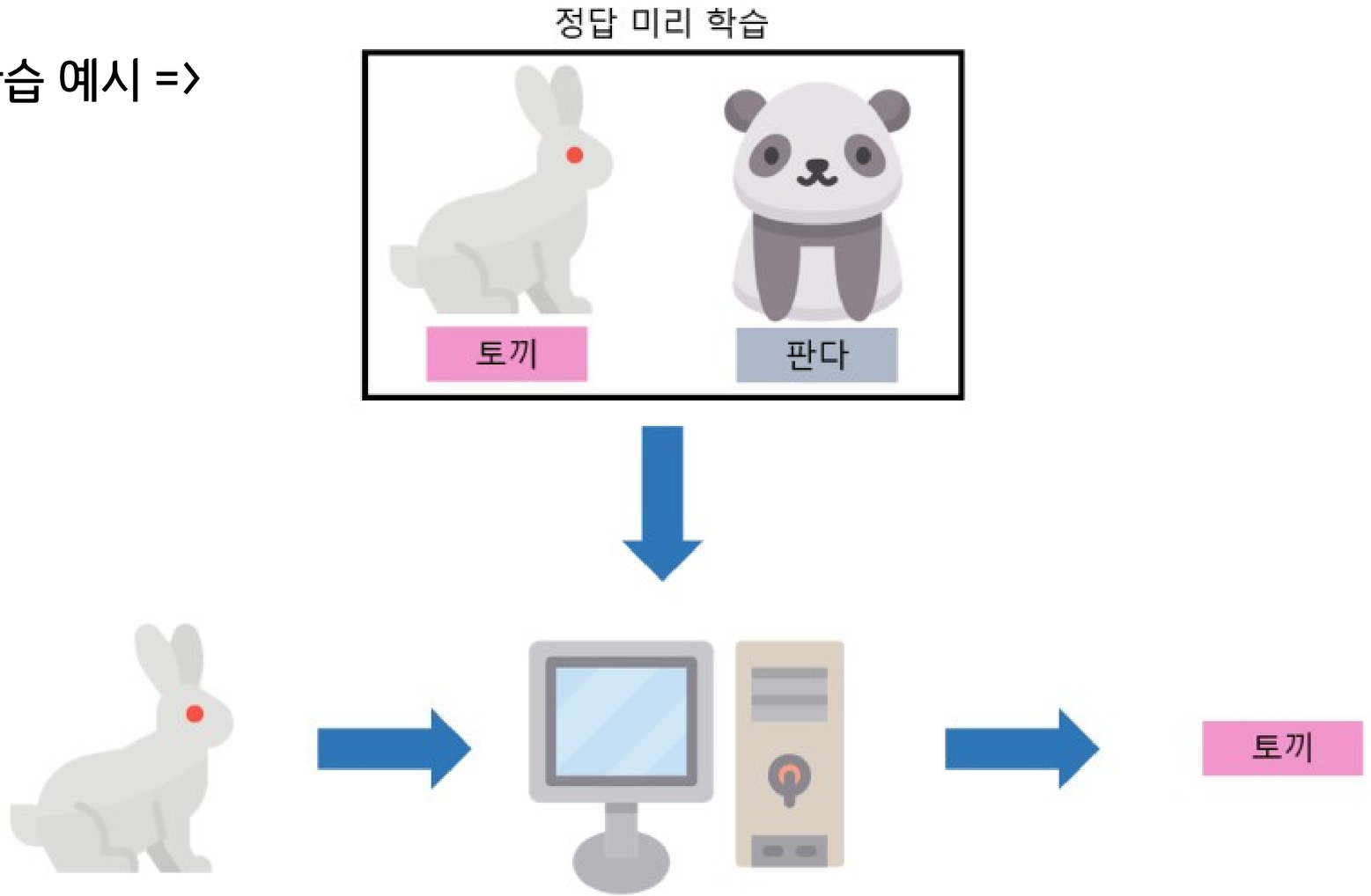
- 강화학습

기계 학습 중 컴퓨터가 주어진 상태에 대해 최적의 행동을 선택하는 학습 방법

2. 머신러닝 & 딥러닝

머신러닝의 분류

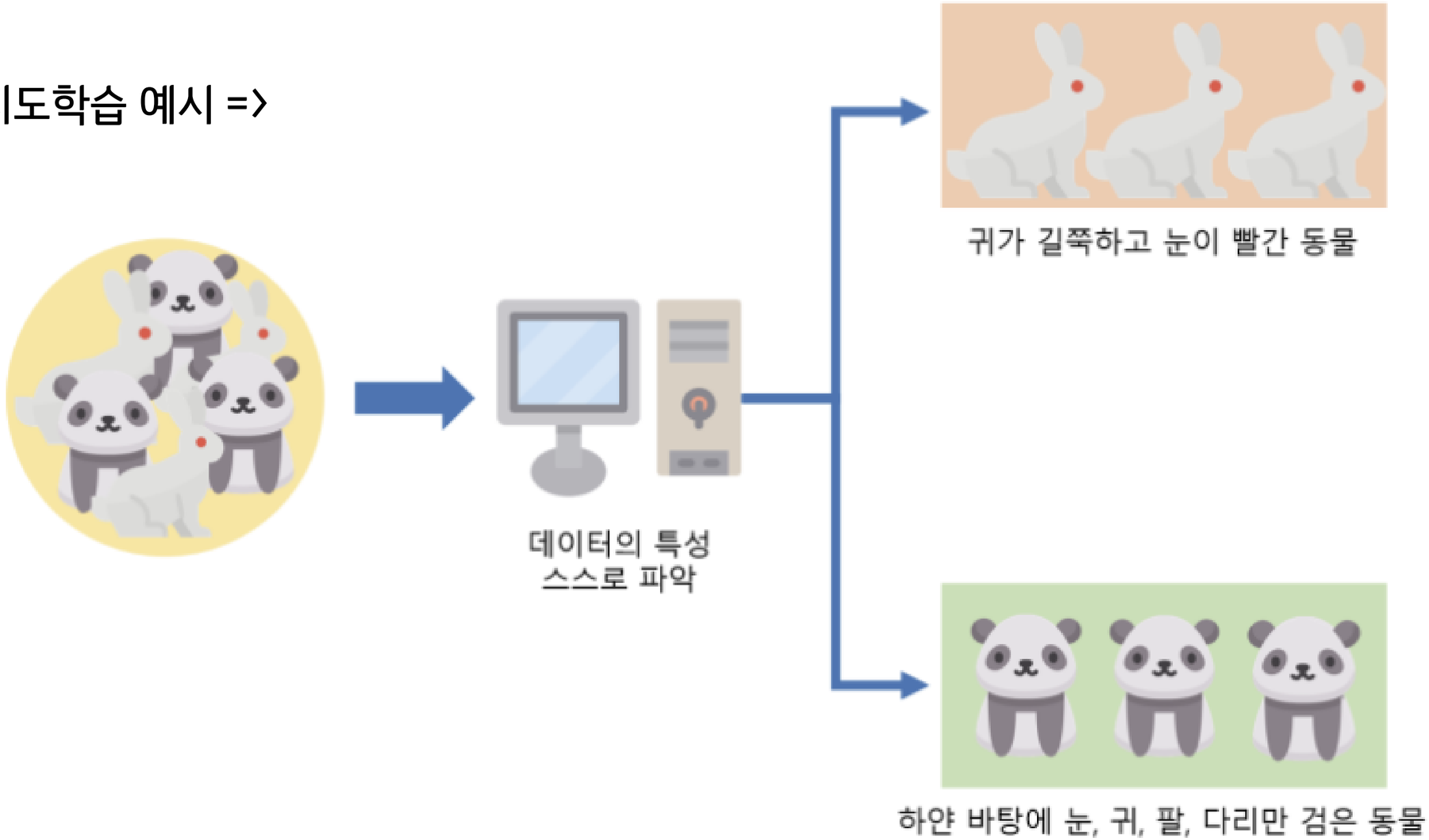
지도학습 예시 =>



2. 머신러닝 & 딥러닝

머신러닝의 분류

비지도학습 예시 =>

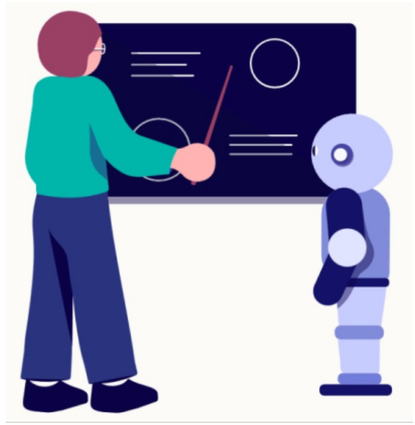


2. 머신러닝 & 딥러닝

과정

데이터 수집

데이터 전처리



모델 학습

예측 및 평가

*학습 : 모델에 데이터가 입력되고 패턴이 분석되는 과정

머신러닝의 가장 큰 단점 : 데이터에 매우 의존적

좋은 품질의 데이터를 갖추지 못한다면 머신러닝의 수행 결과도 좋을 수 없음

2. 머신러닝 & 딥러닝

패키지 소개 pandas / numpy / scikit-learn / seaborn&matplotlib

Pandas

파이썬의 데이터 분석을 위한 라이브러리
수치형 테이블과 시계열 데이터를 조작하고 운영하기 위한
데이터 제공

NumPy

수학 및 과학 연산을 위한 파이썬 패키지
파이썬으로 수치해석, 통계 관련 기능을 구현한다고 할 때 가
장 기본이 되는 모듈

2. 머신러닝 & 딥러닝

패키지 소개 pandas / numpy / scikit-learn / seaborn&matplotlib

Scikit-Learn은 Sklearn이라고도 불리며, Python을 기반으로 작동하는, 기계 학습 분야에서 주로 활용되는 오픈 소스 소프트웨어 라이브러리

matplotlib : 데이터의 시각화에 주로 활용되는 오픈 소스 라이브러리

seaborn : python에서 통계 그래프를 만들기 위한 라이브러리

2. 머신러닝 & 딥러닝

딥러닝

딥러닝은 사람을 닮은 인공지능을 만들기 위해 나온 것
사람이 할 수 있는 것과 유사한 판단을 컴퓨터가 해 낼 수 있게끔 하는 인공지능 기법 중 '머신러닝' 기법이 효과적. 이 머신러닝 안에는 여러 알고리즘들이 있는데 그 중에서도 딥러닝이 가장 좋은 효과를 냄

** 인공지능 > 머신러닝 > 딥러닝

Kuggle

3. 데이터 전처리

3. 데이터 전처리

데이터 전처리 의미

데이터를 분석 및 처리에 적합한 형태로 만드는 과정

일반적으로 데이터는 비어있는 부분이 있거나 정합성이 맞지 않는 경우가 많음

데이터가 갖고 있는 본래의 정보를 왜곡시키거나 변형시키지 않으면서
데이터 사용 목적에 맞게 효과적으로 가공해야 함

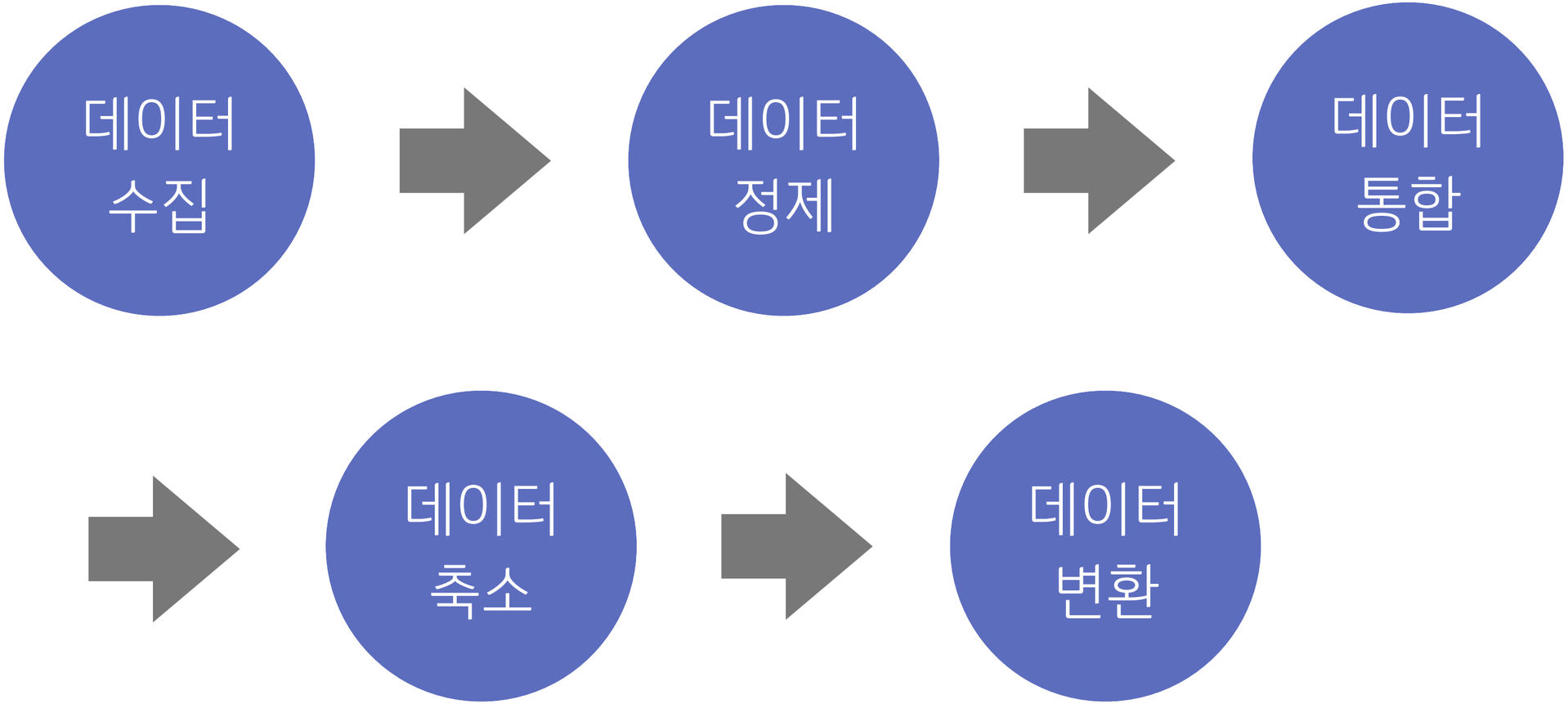
3. 데이터 전처리

데이터 전처리 종류

- 데이터 정제 (Cleansing)
- 데이터 변환 (Transformation)
- 데이터 필터링 (Filtering)
- 데이터 통합 (Integration)
- 데이터 축소 (Reduction)

3. 데이터 전처리

데이터 전처리 과정



3. 데이터 전처리

결측치 처리

결측치 확인 코드
: df.isna( ).sum( )

inplace=True
: 변경된 설정으로 덮어씌움

1. 결측치 제거

   - dropna( ) <-> fillna( ) `df.dropna( , inplace=True)`

2. 평균(mean), 중앙치(median), 최빈값(mode)으로 대체

```
df['cylinders']=df['cylinders'].fillna(df.cylinders.mean( ))
```

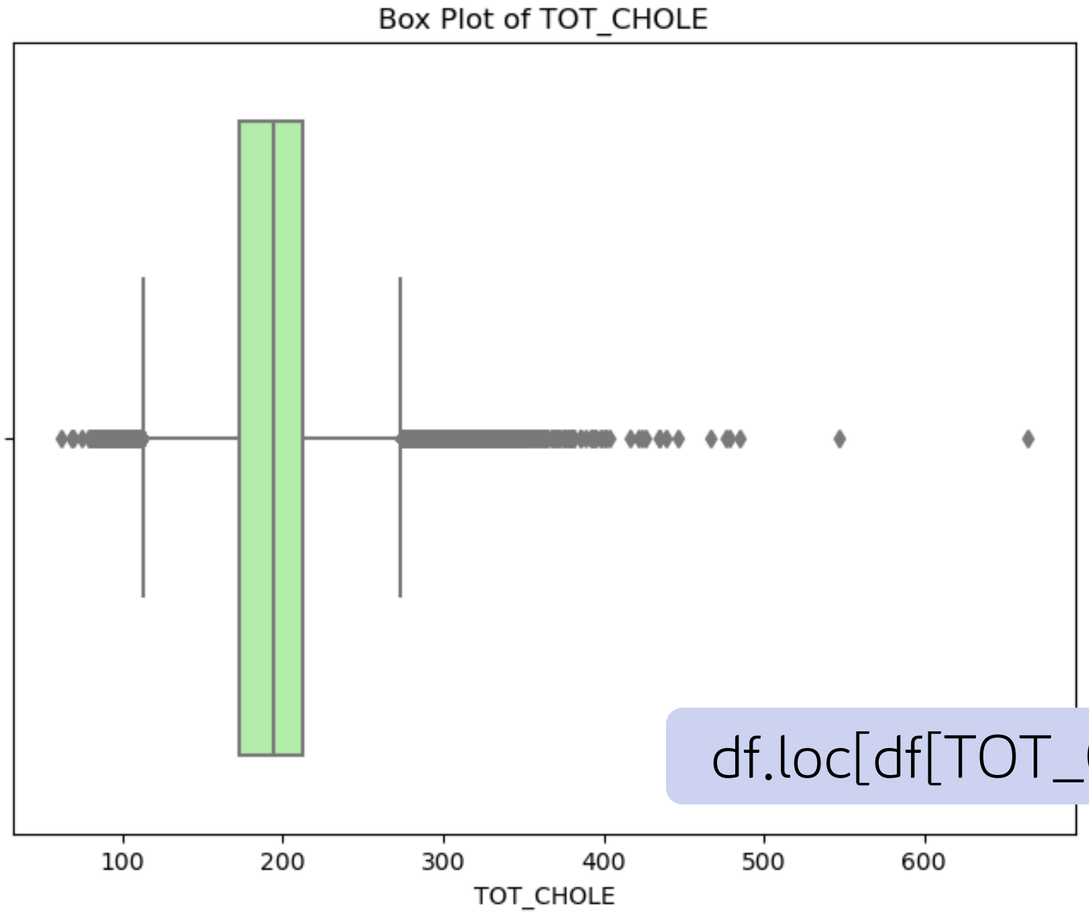
3. 보간법

   - 선형보간법 : interpolate(method='linear)

```
df['cylinders'] = df['cylinders'].interpolate(method='linear')
```

3. 데이터 전처리

이상치 처리



- dropna( ) 함수로 제거 후 원하는 값으로 채우기
- np.nan 값으로 바꾼 후 원하는 값으로 채우기

```
df.loc[df[TOT_CHOLE] >= 500, 'TOT_CHOLE'] = np.nan
```

3. 데이터 전처리

중복값 처리 및 데이터 분포 변환

<중복값 처리>

- drop_duplicates( )
- 데이터가 중복되어 들어간 것인지, 아니면 실제 데이터의 값이 같은 것인지에 대한 구분 필요

<데이터 분포 변환>

- 대부분의 모델 : 변수가 특정 분포를 따른다는 가정을 기반으로 함
- log( ), exp( ), sqrt( ) 등의 함수를 이용하여 데이터 분포 변환

3. 데이터 전처리

데이터 단위 변환

데이터의 스케일(측정 단위)가 다를 경우

모델에 부정적인 영향을 끼침

스케일링을 통해 단위를 일정하게 맞추는 작업 필요

- Scaling : 평균이 0, 분산이 1인 분포로 변환
- MinMax Scaling : 특정 범위 (ex) 0 ~ 1)로 모든 데이터를 변환

3. 데이터 전처리

데이터 선택

<loc 속성>

- 인덱스를 기준으로 행 데이터 추출
- df.loc[0], df.loc[[0,10,20]]
- df.loc[-1] -> 인덱스에 없는 값을 사용하면 오류 발생

```
In [7]: df.loc[-1]
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
File ~\anaconda3\Lib\site-packages\pandas\core\indexes\range.py:391, in RangeIndex.get_loc(self, key, method, tolerance)
    390 try:
--> 391     return self._range.index(new_key)
    392 except ValueError as err:

ValueError: -1 is not in range

The above exception was the direct cause of the following exception:

KeyError                                  Traceback (most recent call last)
Cell In[7], line 1
----> 1 df.loc[-1]
```

<iloc 속성>

- 행 번호를 기준으로 행 데이터 추출
- df.iloc[1], df.iloc[[0,10,20]]
- df.iloc[-1] -> 마지막 행 데이터를 추출

```
In [13]: df.iloc[-1]
```

```
Out[13]: mpg                31.0
         cylinders           4
         displacement        119.0
         horsepower          82.0
         weight              2720
         acceleration        19.4
         model_year          82
         origin              usa
         name                chevy s-10
         Name: 397, dtype: object
```

3. 데이터 전처리

데이터 선택

<그룹 연산>

- groupby( )
- ex) displacement 열을 cylinders 별로 그룹화하여 평균 계산

```
In [18]: df.groupby('cylinders')['displacement'].mean()

Out[18]: cylinders
         3     72.500000
         4    109.796569
         5    145.000000
         6    218.142857
         8    345.009709
         Name: displacement, dtype: float64
```

4. 과제

1. pandas 이용하여
데이터 전처리 해보기

2. 시각화 해보기