

# AI 프로그래밍

- 2024



2주차

## ■ 주 차 별 수업 내용

- 1주차 : 인공지능, 딥러닝, 머신러닝의 정의와 적용 분야
- **2주차 : 머신 러닝 개념, 회귀 (regression), 선형 회귀**
- 3 주차 : 선형 회귀, 경사 하강법 , 다중 선형 회귀
- 4주차 : 넘파이(Numpy), 맷플롯립(Matplotlib) 라이브러리 실습
- 5주차 : 분류: 회귀 군집, 로지스틱 회귀
- 6주차 : 퍼셉트론, 활성화 함수, 다중 퍼셉트론
- 7주차: 다중 퍼셉트론, 딥러닝 개념
- 8주차 : 중간 평가

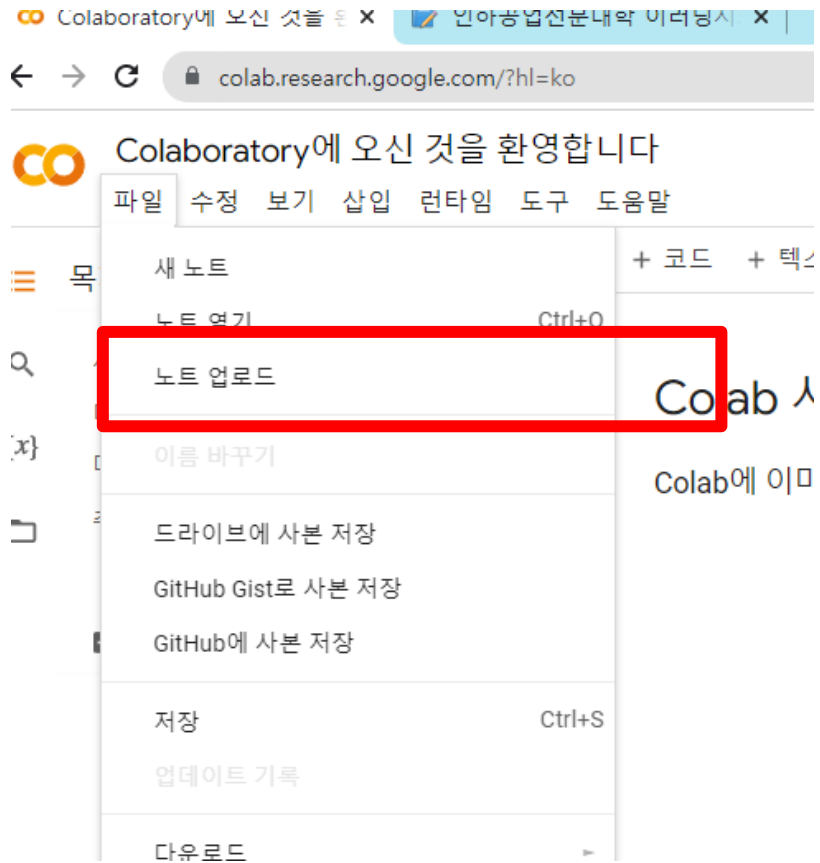
## ■ 주 차 별 수업 내용

- 9 주차 : 딥러닝 개념, 손실 함수
- 10-11주차 : 딥러닝 구현을 위한 개념
- 12 -13주차 : 컨볼루션 신경망
- 14주차: 순환 신경망
- 15주차 : 기말 평가

# Colab 사용 방법

인하공전 컴퓨터 정보공학과

노트북 업로드 : \*.ipynb 파일 upload

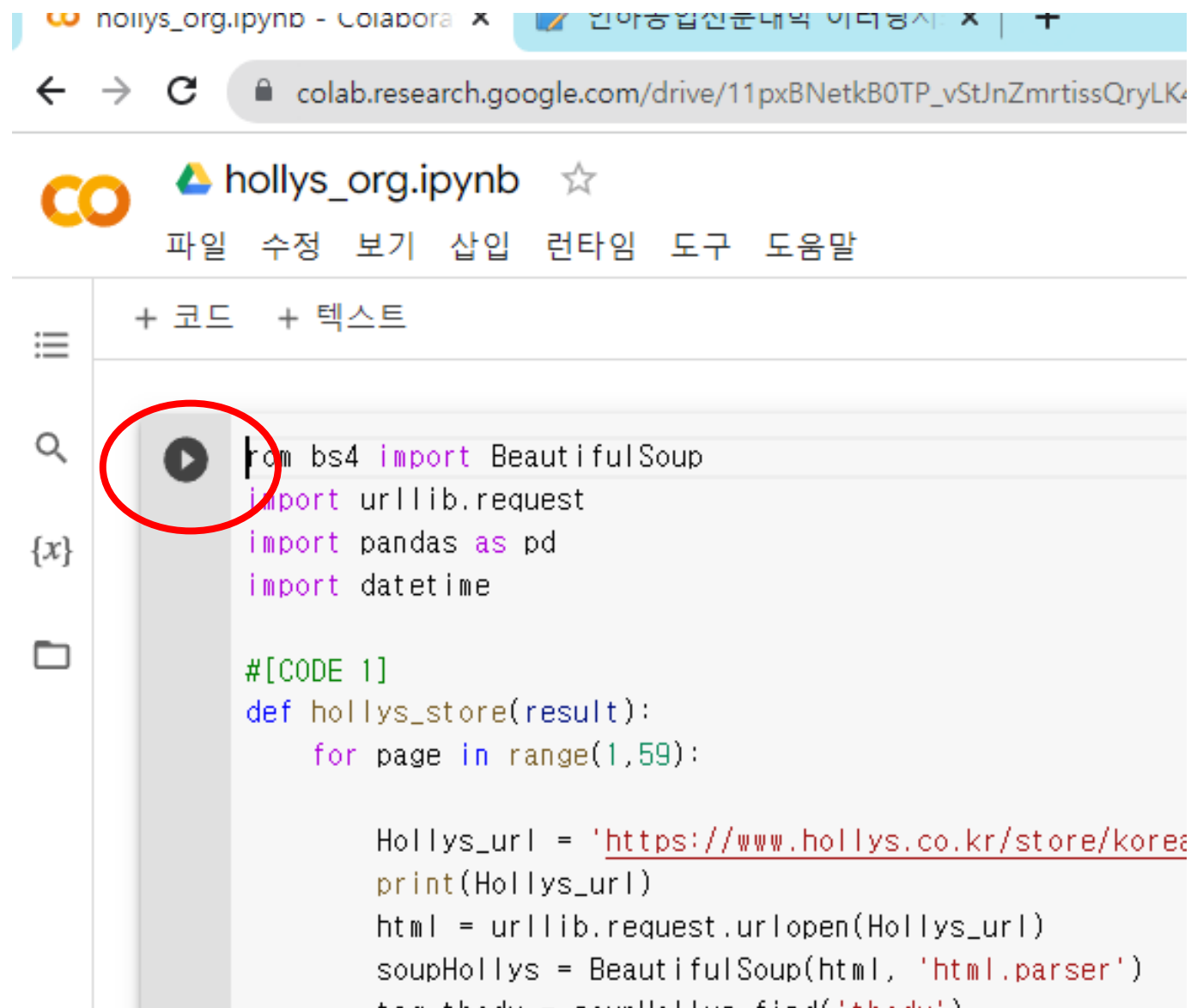


# Colab 사용 방법

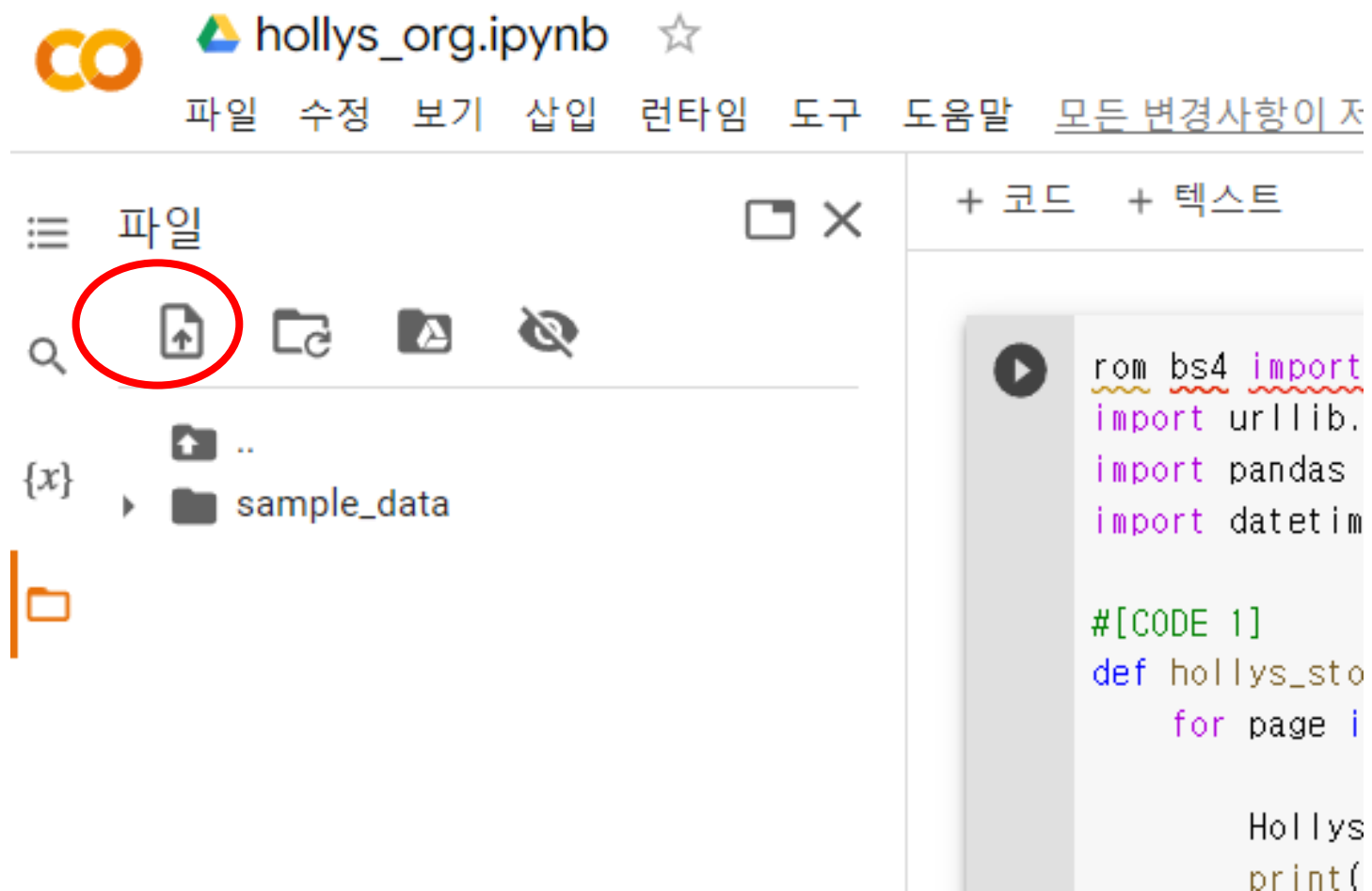
인하공전 컴퓨터 정보공학과

실행

=> 재생 버튼 클릭



데이터 파일 upload



The screenshot displays the Google Colab web interface. At the top, the Colab logo is followed by the file name 'hollys\_org.ipynb' and a star icon. Below this is a navigation bar with tabs: '파일' (File), '수정' (Edit), '보기' (View), '삽입' (Insert), '런타임' (Runtime), '도구' (Tools), '도움말' (Help), and '모든 변경사항이 저장됨' (All changes saved). The '파일' tab is active, showing a file explorer. In the file explorer, the 'upload' icon (a document with an upward arrow) is circled in red. Below the icons, a folder named 'sample\_data' is visible. To the right of the file explorer, there are buttons for '+ 코드' (Add code) and '+ 텍스트' (Add text). The main area on the right shows a code cell with a play button icon and the following Python code:

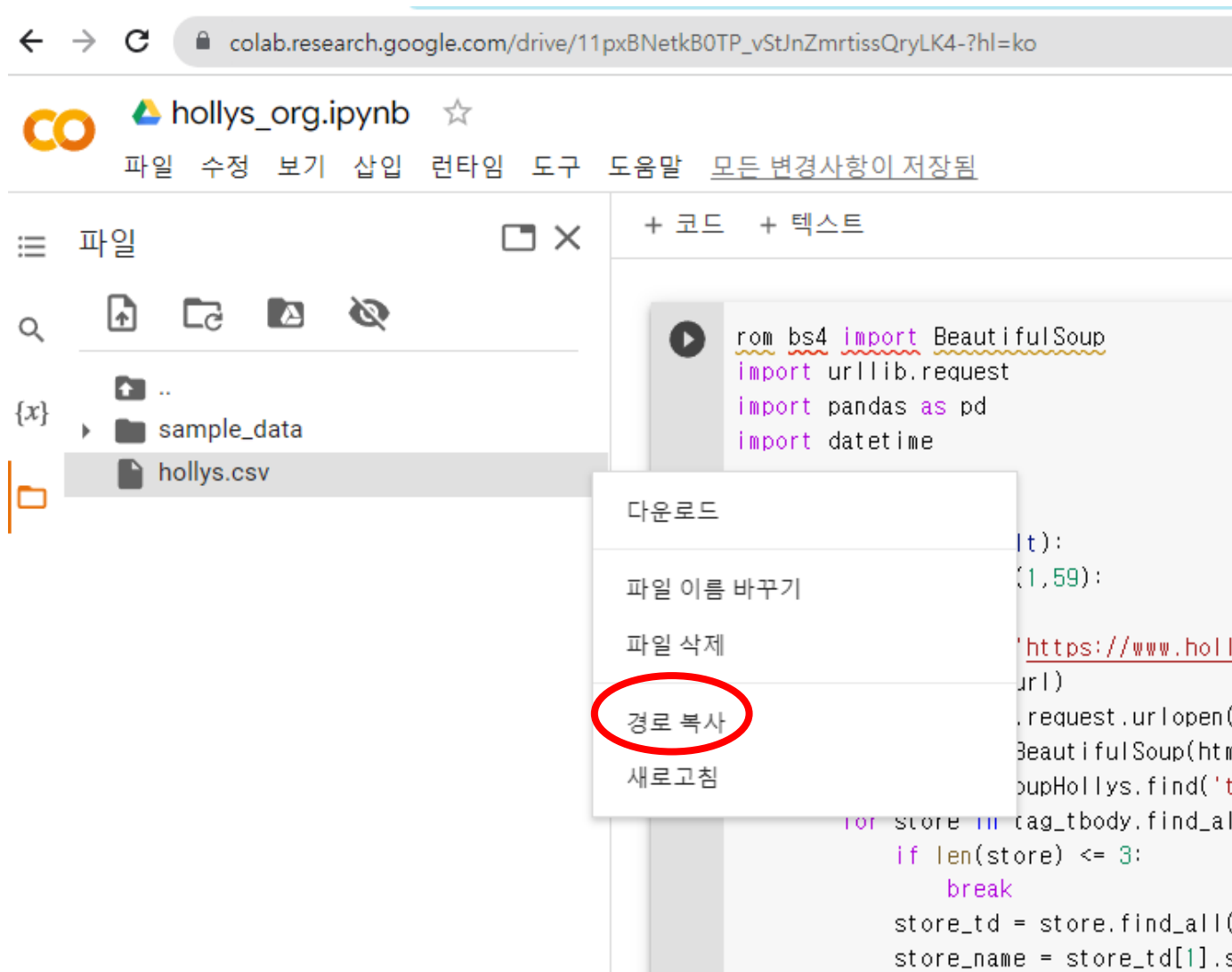
```
from bs4 import
import urllib.
import pandas
import datetim

#[CODE 1]
def hollys_sto
    for page i

Hollys
print(
```

# Colab 사용 방법

인하공전 컴퓨터 정보공학과



The screenshot shows the Google Colab web interface. The browser address bar displays `colab.research.google.com/drive/11pxBNetkB0TP_vStJnZmrtissQryLK4-?hl=ko`. The Colab logo and the file name `hollys_org.ipynb` are visible. Below the file name, there are tabs for `파일` (File), `수정` (Edit), `보기` (View), `삽입` (Insert), `런타임` (Runtime), `도구` (Tools), `도움말` (Help), and `모든 변경사항이 저장됨` (All changes are saved). The left sidebar shows a file explorer with a folder named `sample_data` and a file named `hollys.csv`. The main area contains a code cell with the following Python code:

```
from bs4 import BeautifulSoup
import urllib.request
import pandas as pd
import datetime
```

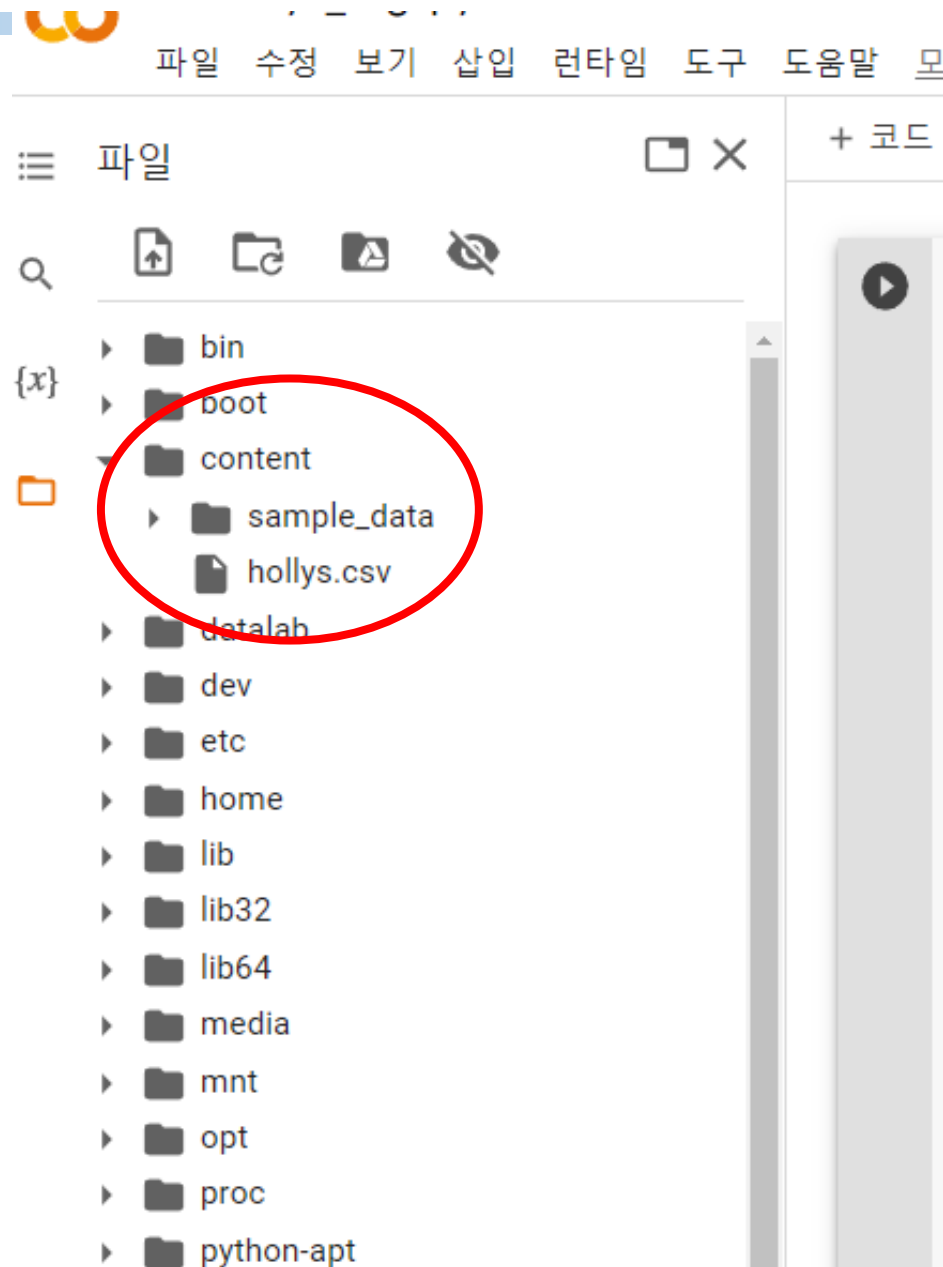
A context menu is open over the code cell, showing options: `다운로드` (Download), `파일 이름 바꾸기` (Rename), `파일 삭제` (Delete), `경로 복사` (Copy path), and `새로고침` (Refresh). The `경로 복사` option is highlighted with a red circle.

# Colab 사용 방법

인하공전 컴퓨터 정보공학과

실행 directory

./content





# 오늘 수업 순서

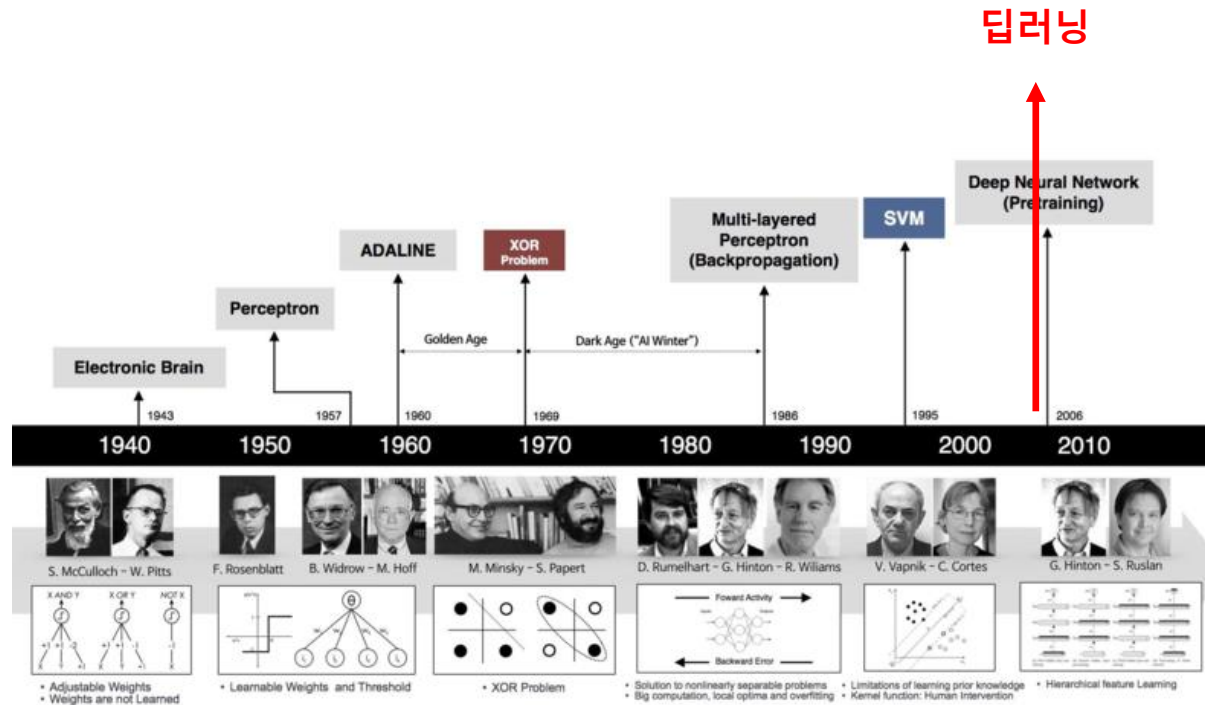
인하공전 컴퓨터 정보공학과

- **인공 지능 개념 복습**
- **머신 러닝 개념**
  - Iris 예제 실습
  - Mnist 예제 실습
- **선형 회귀 실습**

# 인공지능의 역사

인하공전 컴퓨터 정보공학과

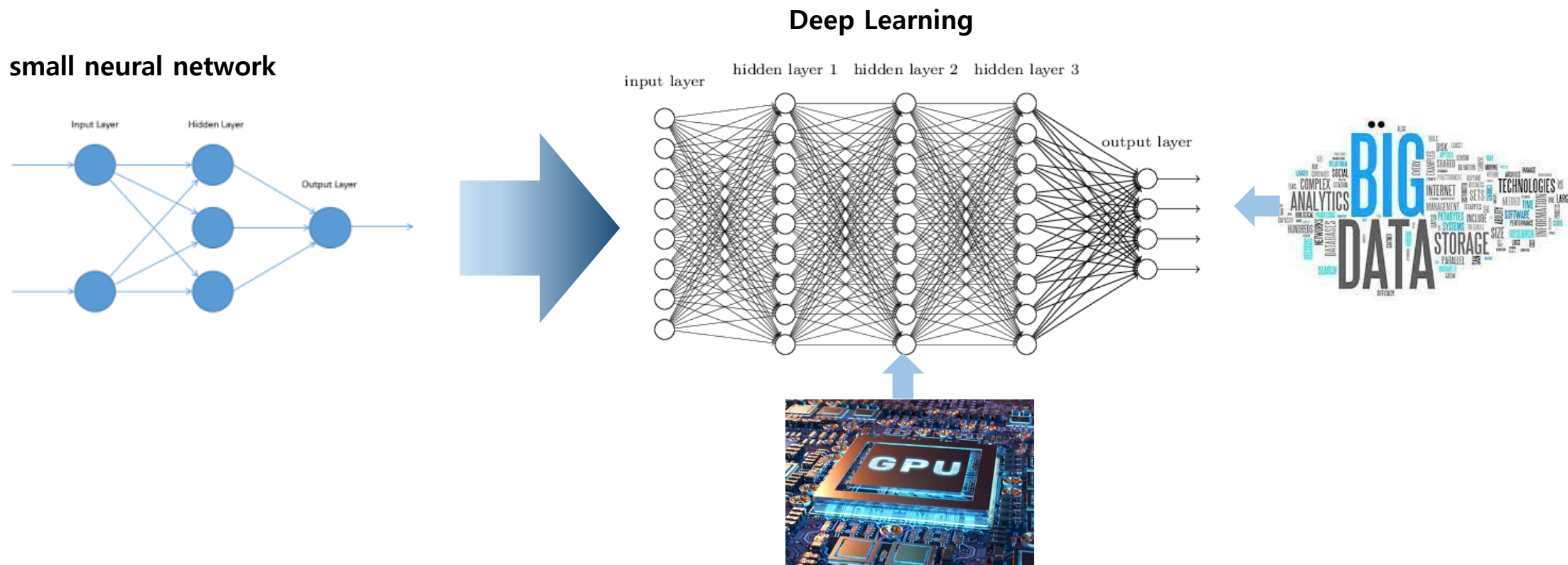
- 1950년대에 **인공 지능의 개념**과 신경망의 기본 개념인 **퍼셉트론** 개념이 성립됨.
- 2000년까지 신경망을 사용하지 않은 머신 러닝 사용됨.
- 2006 년부터 신경망을 이용한 딥 러닝 (Deep Learning)이 급속히 발전함.

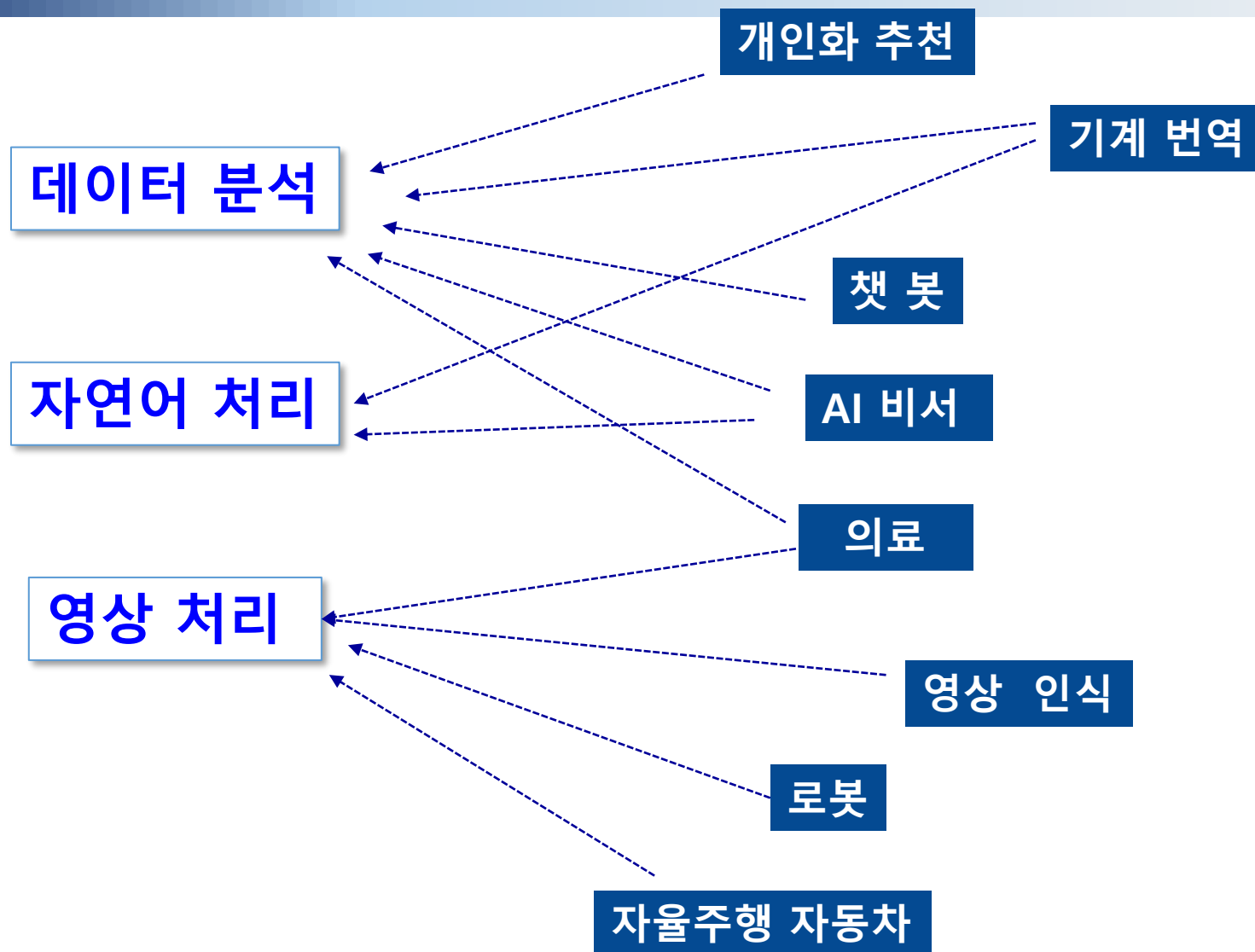


# 딥 러닝 (Deep Learning) 발전 이유

인하공전 컴퓨터 정보공학과

- **하드웨어의 발전.** 강력한 GPU는 딥러닝에서 복잡한 연산에 소요되는 시간을 크게 단축
- **빅 데이터.** 대량으로 쏟아져 나오는 데이터들을 학습에 활용 가능
- **신경망 이론의 단점 해결**





# 머신 러닝 라이브러리

인하공전 컴퓨터 정보공학과

## 머신러닝

 TensorFlow

 Keras

 PYTORCH  
pytorch

 scikit  
learn  
Scikit learn

## 데이터 분석

 pandas

 seaborn

 Matplotlib

## 자연어 처리

 spaCy

 Transformers  
Hugging face

## 영상 처리

 OpenCV

 pillow

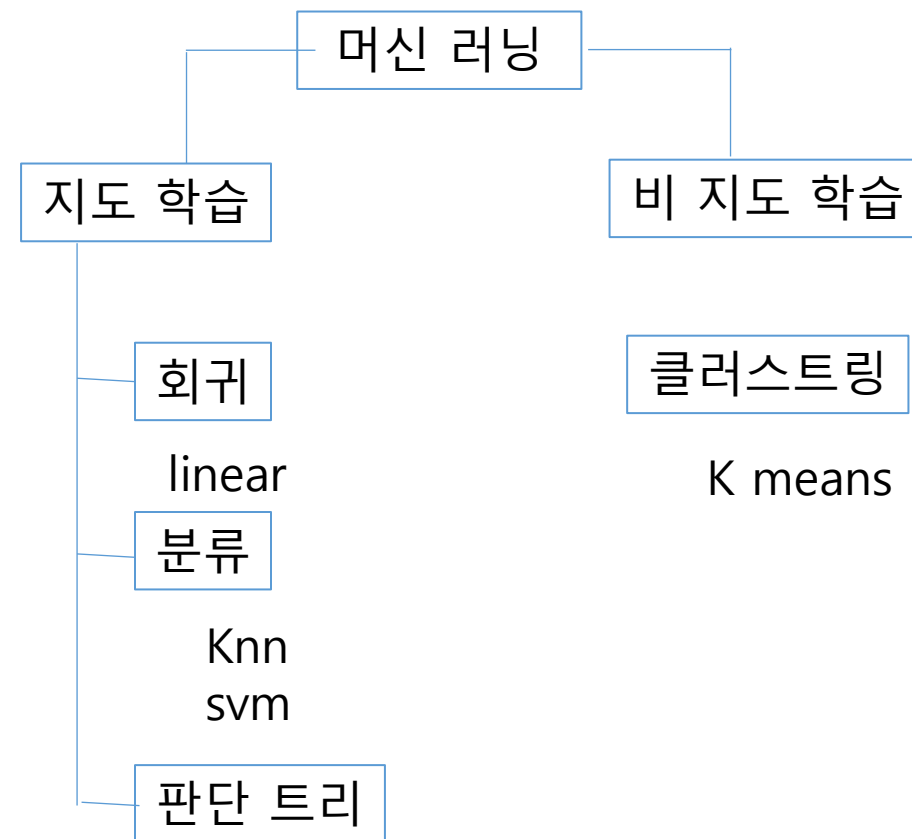
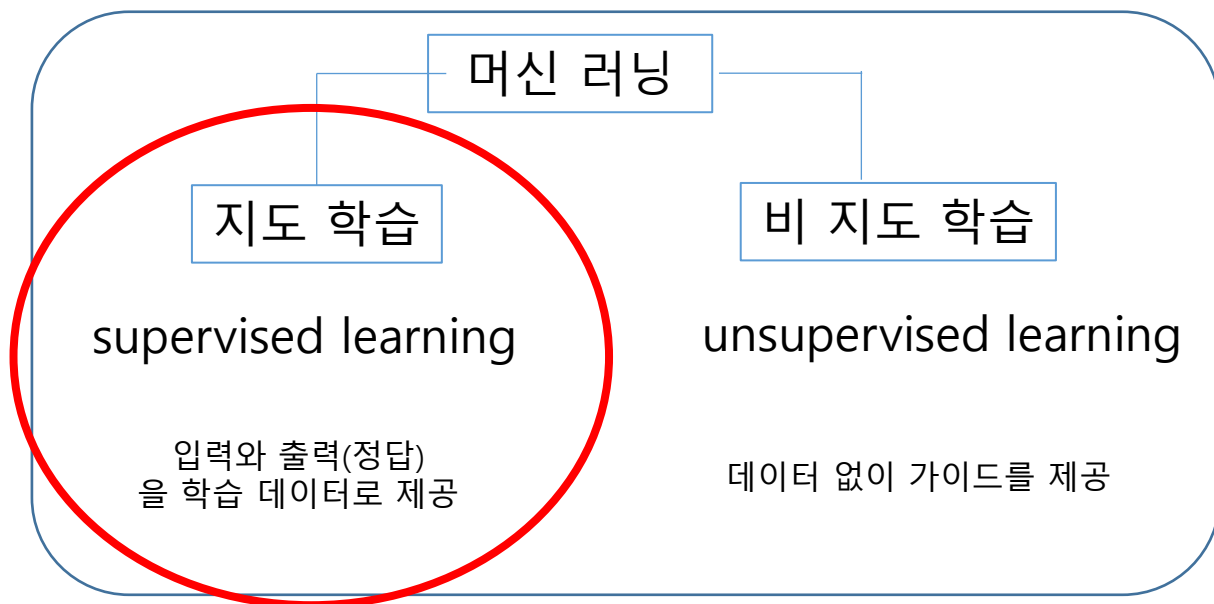
# 머신러닝과 vs 기존 프로그래밍

인하공전 컴퓨터 정보공학과

- 전통적인 접근 방식은 프로그래밍.
- 머신러닝은 데이터를 주고 학습시킴
  - 입력데이터와 출력 데이터 제공

# 머신 러닝

인하공전 컴퓨터 정보공학과



지도학습 (Supervised Learning)

: 정답(입력 과 원하는 출력)을 제공 받음

비지도 학습(Unsupervised learning)

: 정답이 주어지지 않음. 입력 데이터에서  
패턴을 찾는 학습



회귀 (regression)

: 입력과 출력이 실수

분류(classification)

: 입력을 레이블로 분할. 출력이 이산적(discrete)

1. 데이터 확보
2. 데이터 정제
3. 모델 학습
4. 모델 평가
5. 예측에 사용

# 학습 데이터와 테스트 데이터

인하공전 컴퓨터 정보공학과

데이터

훈련 데이터

테스트 데이터

## 특성

- 꽃받침(sepal) width
- 꽃받침(sepal) length
- 꽃잎(petal) width
- 꽃잎(petal) length

## 종류

- iris setosa
- iris versicolor
- iris virginica

## 인하공전 컴퓨터 정보공학과

[illegible]

# 훈련 데이터와 테스트 데이터 분리

인하공전 컴퓨터 정보공학과

```
from sklearn.model_selection import train_test_split

X = iris.data
y = iris.target

# (80:20)으로 분할한다.
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=4)

print(X_train.shape)
print(X_test.shape)
```

```
(120, 4)
(30, 4)
```

```
from sklearn.neighbors import KNeighborsClassifier  
  
knn = KNeighborsClassifier(n_neighbors=6)  
knn.fit(X_train, y_train)
```

```
y_pred = knn.predict(X_test)
from sklearn import metrics

scores = metrics.accuracy_score(y_test, y_pred)
```

```
0.9666666666666667
```



```
classes = {0:'setosa',1:'versicolor',2:'virginica'}
```

```
# 새로운 데이터를 제시해보자.
```

```
x_new = [[3,4,6,2], [7,4,2,2]]
```

```
y_predict = knn.predict(x_new)
```

```
print(classes[y_predict[0]])
```

```
print(classes[y_predict[1]])
```

```
virginica
```

```
setosa
```

과제 1)

```
code :iris_knn.ipynb
```

입력 데이터가 [5,1,2,1] , [3,5,2,2] 일때 class 예측값은?

- MNIST 필기체 숫자 이미지

```
y_pred = knn.predict(X_test)

scores = metrics.accuracy_score(y_test, y_pred)
print(scores)
```

```
0.9532814238042269
```

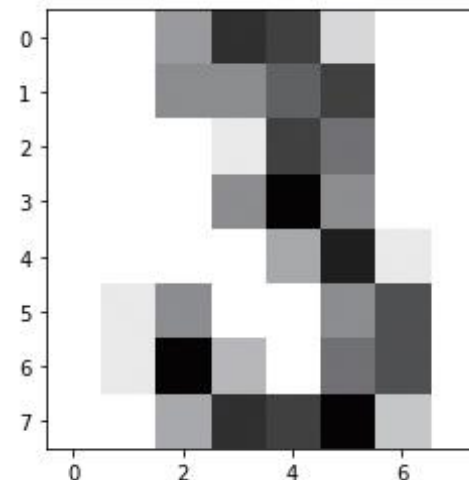
```
# 이미지를 출력하기 위하여 평탄화된 이미지를 다시 8×8 형상으로 만든다.  
plt.imshow(X_test[10].reshape(8,8), cmap=plt.cm.gray_r, interpolation='nearest')  
  
y_pred = knn.predict([X_test[10]]) # 입력은 항상 2차원 행렬이어야 한다.  
print(y_pred)
```

[3]

딥러닝 익스프레스, 생능 출판사

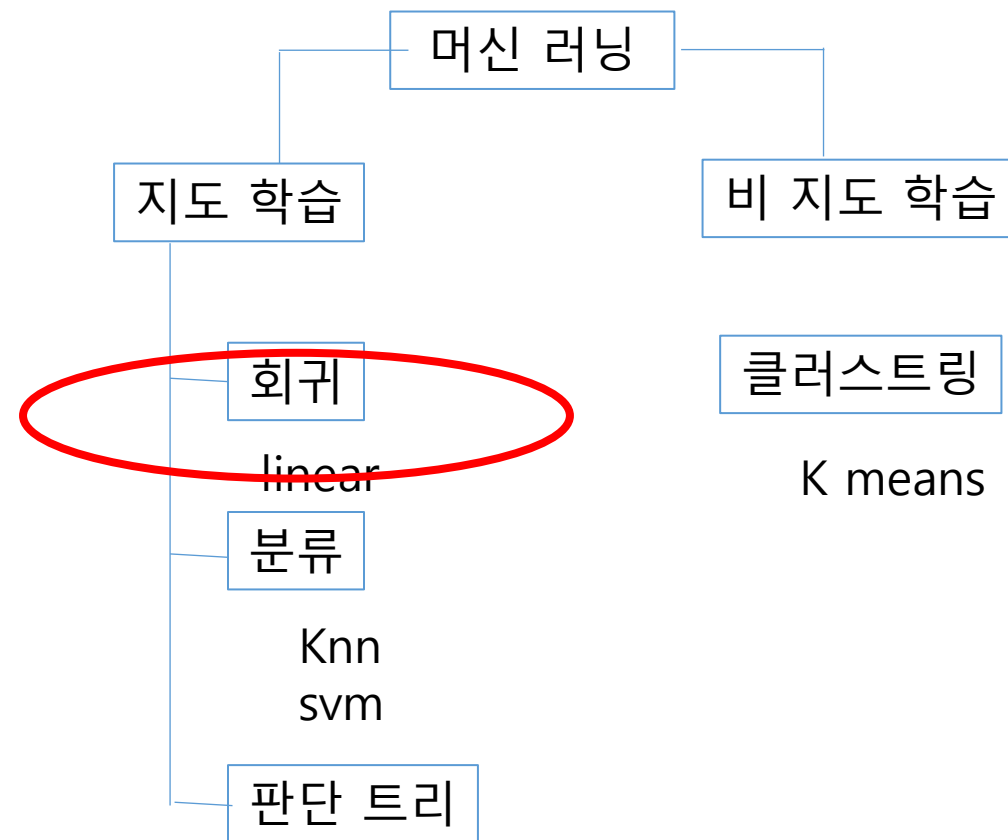
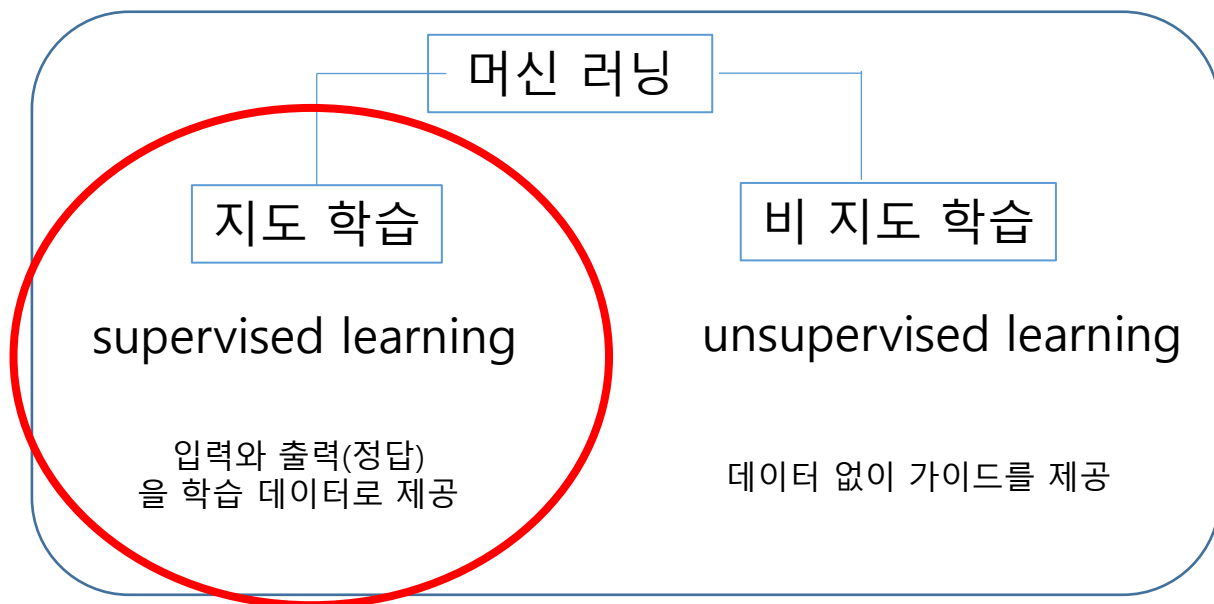
과제 2)

code :mnist\_knn.ipynb  
5번째 test 이미지의 예측 값은?  
출력 된 image는 첨부 하시오.



# 머신 러닝

인하공전 컴퓨터 정보공학과



# 회귀 (regression)

- 데이터를 학습하여 결과를 예측 하는 함수를 도출  $y=f(x)$

- 선형 회귀

- 입력 값  $x$ 가 1차원일때
- 직선의 방정식
- $Y=ax+b$

$\Rightarrow Y=wx+b$ ,  $w$ : weight,  $b$ : bias

예) 입력 : 공부시간, 출력 : 성적  
입력: 몸무게, 출력 : 키

```
import numpy as np
import matplotlib.pyplot as plt

X = np.array([0.0, 1.0, 2.0])
y = np.array([3.0, 3.5, 5.5])

w = 0      # 기울기
b = 0      # 절편

lr = 0.01  # 학습률
epochs = 1000 # 반복 횟수

n = float(len(X)) # 입력 데이터의 개수

# 경사 하강법
for i in range(epochs):
    y_pred = w*X + b
    dw = (2/n) * sum(X * (y_pred-y))
    db = (2/n) * sum(y_pred-y)
    w = w - lr * dw
    b = b - lr * db

# 기울기와 절편을 출력한다.
print(w, b)

# 예측값을 만든다.
y_pred = w*X + b

# 입력 데이터를 그래프 상에 찍는다.
plt.scatter(X, y)

# 예측값은 선그래프로 그린다.
plt.plot([min(X), max(X)], [min(y_pred), max(y_pred)], color='red')
plt.show()
```

# 선형 회귀 예측값  
# 넘파이 배열간의 산술 계산은 요소별로 적용  
# sum()은 모든 요소들의 합을 계산하는 내장 함수  
# 기울기 수정  
# 절편 수정

## regression1\_loss\_noprint .ipynb

```
import numpy as np
import matplotlib.pyplot as plt
```

```
def mse(y,y_hat):
    return ((y-y_hat)**2).mean()
```

```
def mse_val(y,predict_result):
    return mse(np.array(y),np.array(predict_result))
```

```
X = np.array([0.0, 1.0, 2.0])
y = np.array([3.0, 3.5, 5.5])
```

```
w = 0      # 기울기
b = 0      # 절편
```

```
lr = 0.01 # 학습률
epochs = 500 # 반복 횟수
```

```
n = float(len(X)) # 입력 데이터의 개수
```

```
# 경사 하강법
for i in range(epochs):
    y_pred = w*X + b      # 선형 회귀 예측값
    dw = (2/n) * sum(X * (y_pred-y)) # 넘파이 배열간의 산술 계산은 요소별로 적용
    db = (2/n) * sum(y_pred-y)      # sum()은 모든 요소들의 합을 계산하는 내장 함수
    w = w - lr * dw      # 기울기 수정
    b = b - lr * db      # 절편 수정
    if (i%50==0):
        print('iteration %3d: loss %4.2f w %3.2f b %3.2f'%(i,mse(y,y_pred),w,b))
```

```
# 기울기와 절편을 출력한다.
print ('##### final w,b',w, b)
```

```
# 예측값을 만든다.
y_pred = w*X + b
```

```
# 입력 데이터를 그래프 상에 찍는다.
plt.scatter(X, y)
```

```
# 예측값은 선그래프로 그린다.
plt.plot([min(X), max(X)], [min(y_pred), max(y_pred)], color='red')
plt.show()
```

과제 3: code : regression1\_loss\_noprint .ipynb  
첫번째 loss 값과 w,b 값은?



# 선형 회귀 예제 실습

regression2.ipynb

인하공전 컴퓨터 정보공학과

x	Y
0	3
1	3.5
2	5.5

```
import matplotlib.pyplot as plt
from sklearn import linear_model
```

```
# 선형 회귀 모델을 생성한다.
reg = linear_model.LinearRegression()
```

```
# 데이터는 파이썬의 리스트로 만들어도 되고 아니면 넘파이의 배열로 만들어도 됨
X = [[0], [1], [2]]          # 반드시 2차원으로 만들어야 함
y = [3, 3.5, 5.5]           #  $y = x + 3$ 
```

```
# 학습을 시킨다.
reg.fit(X, y)
```

```
print(reg.coef_)              # 직선의 기울기, w
print(reg.intercept_)         # 직선의 y-절편, b
print(reg.score(X, y))
```

```
print(reg.predict([[5]]))
```

```
# 학습 데이터와 y 값을 산포도로 그린다.
plt.scatter(X, y, color='black')
```

```
# 학습 데이터를 입력으로 하여 예측값을 계산한다.
y_pred = reg.predict(X)
```

```
# 학습 데이터와 예측값으로 선그래프로 그린다.
# 계산된 기울기와 y 절편을 가지는 직선이 그려진다.
plt.plot(X, y_pred, color='blue', linewidth=3)
plt.show()
```



# 선형 회귀 예제 실습

인하공전 컴퓨터 정보공학과

키(단위: cm)	몸무게(단위: kg)
174	71
152	55
138	46
128	38
186	88

```
import matplotlib.pyplot as plt
from sklearn import linear_model
```

```
reg = linear_model.LinearRegression()
```

```
X = [[174], [152], [138], [128], [186]]
y = [71, 55, 46, 38, 88]
```

```
reg.fit(X, y) # 학습
```

```
print(reg.predict([[165]]))
```

```
# 학습 데이터와 y 값을 산포도로 그린다.
plt.scatter(X, y, color='black')
```

```
# 학습 데이터를 입력으로 하여 예측값을 계산한다.
y_pred = reg.predict(X)
```

```
# 학습 데이터와 예측값으로 선그래프로 그린다.
# 계산된 기울기와 y 절편을 가지는 직선이 그려진다.
plt.plot(X, y_pred, color='blue', linewidth=3)
plt.show()
```

과제 4: code : regression3.ipynb  
키가 160일때의 몸무게 예측 값은?

# 선형 회귀 예제 실습 - 당뇨병 예제

인하공전 컴퓨터 정보공학과

당뇨병 데이터

데이터 개수 :442

특징(입력) 10개 (age,sex,bmi,bp,s1,s2,s3,s4,s5,s6)

출력(혈당)

Bmi와 혈당간의 관계 예측

```
diabetes_X, diabetes_y = datasets.load_diabetes(return_X_y=True)
```

```
print('diabetes_X',diabetes_X.shape )
```

# 하나의 특징(BMI)만 추려내서 2차원 배열로 만든다. BMI 특징의 인덱스가 2이다.

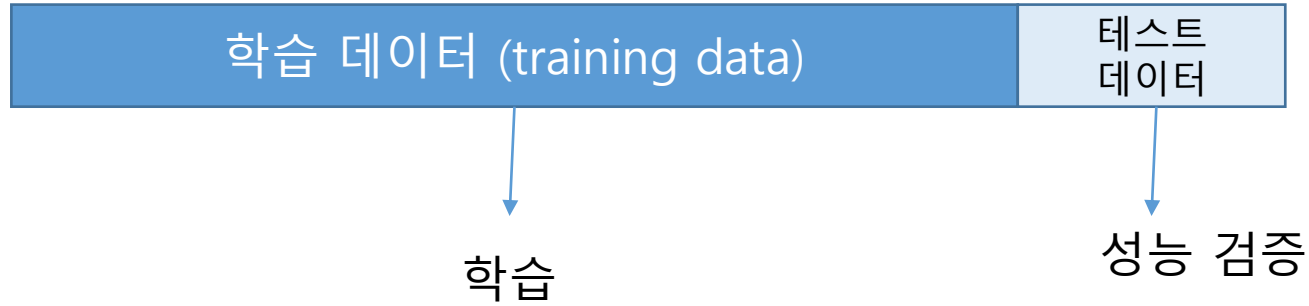
```
diabetes_X_new0 = diabetes_X[:, 2]
```

```
print('diabetes_X_new0',diabetes_X_new0.shape )
```

```
diabetes_X_new = diabetes_X_new0[:, np.newaxis]
```

```
print('diabetes_X_new',diabetes_X_new.shape )
```

```
diabetes_X (442, 10)  
diabetes_X_new0 (442,)  
diabetes_X_new (442, 1)
```



442개

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(diabetes_X_new, diabetes_y, test_size=0.1, random_state=0)
print('X_train',X_train.shape )
print('X_test',X_test.shape )
print('Y_train',y_train.shape )
print('Y_test',y_test.shape )
```

```
X_train (397, 1)
X_test (45, 1)
Y_train (397,)
Y_test (45,)
```

# 선형 회귀 예제 실습 - 당뇨병 예제

인하공전 컴퓨터 정보공학과

```
import matplotlib.pyplot as plt
import numpy as np
from sklearn.linear_model import LinearRegression
from sklearn import datasets, linear_model

# 당뇨병 데이터 세트를 적재한다.
diabetes_X, diabetes_y = datasets.load_diabetes(return_X_y=True)

print('diabetes_X,diabetes_X.shape )
# 하나의 특징(BMI)만 추려내서 2차원 배열로 만든다. BMI 특징의 인덱스가 2이다.
diabetes_X_new0 = diabetes_X[:, 2]
print('diabetes_X_new0,diabetes_X_new0.shape )
diabetes_X_new = diabetes_X[:, np.newaxis, 2]
print('diabetes_X_new,diabetes_X_new.shape )

# 학습 데이터와 테스트 데이터를 분리한다.
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(diabetes_X_new, diabetes_y, test_size=0.1,
random_state=0)
print('X_train,X_train.shape)
print('X_test,X_test.shape)
print('y_train,y_train.shape)
print('y_test,y_test.shape)

regr = linear_model.LinearRegression()
regr.fit(X_train, y_train)

# 테스트 데이터로 예측해보자.
y_pred = regr.predict(X_test)
print(regr.predict([[0.01]])) # bmi가 0.01일때 혈당 예측값

# 실제 데이터와 예측 데이터를 비교해보자.
# plt.plot(y_test, y_pred, '.')

plt.scatter(X_test, y_test, color='black')
plt.plot(X_test, y_pred, color='blue', linewidth=3)
plt.show()
```

과제 5: code : diabetes\_exe.ipynb  
Bmi가 0.025일때의 혈당의 예측 값

# 선형 회귀 예제 실습

인하공전 컴퓨터 정보공학과

다음은 집의 면적 당 가격을 정리한 표이다.

X 입력	면적	5	7	12	13	19
Y 출력	가격	12	19	28	37	48

자료실의 `regression1.ipnb` 프로그램을 이용하여 (**학습률은(lr) 0.001**로 변경하시오)

- 1) 선형회귀로 분석 하여 직선의 방정식  $y=wx+b$ 를 구하여라
- 2) 면적이 10일때의 가격을 예측해보자.

```
print(w*10+b)
```

과제 6: code : regression1.ipnb

w,b 와

면적이 10일때의 가격의 예측값은?

# 선형 회귀 예제 실습

인하공전 컴퓨터 정보공학과

다음은 CPU 속도와 프로그래밍 수행 시간을 정리한 표이다.

CPU	30	50	80	90	120
수행 시간	70	140	145	170	260

자료실의 `regression2.ipnb` 프로그램을 이용하여

- 1) 선형회귀로 분석 하여 직선의 방정식  $y=wx+b$ 를 구하여라
- 1) CPU 속도가 100일때의 수행 시간을 예측 하시오.

```
x = [[30], [50], [80], [90], [120]]  
y = [70, 140, 145, 170, 260]
```

```
print(reg.predict([[100]]))
```

과제 7: code : regression2.ipnb

w,b 와

w,b, CPU 속도가 100일때의 수행시간 예측 값은?

- 머신러닝(machine learning)은 인공지능의 한 분야로, 데이터로 알고리즘을 이용하여 학습
- 머신러닝은 지도 학습과 비지도 학습으로 나뉘어진다
- 지도 학습은 크게 회귀와 분류로 나눌 수 있다.
- 입력 (특징, 속성) => 출력(정답, 결과)



# 수고하셨습니다

---

[jhmin@inhatec.ac.kr](mailto:jhmin@inhatec.ac.kr)