[강의 주제]

ㅇ 실전정보기술 : 최신기술 (인공지능/빅데이터, 블록체인, 클라우드)

SW공학, 기타 (윈도우, 모바일...)

파이썬 (VS Code, Colab)

ㅇ 서버프로그램구현 : 서버 구현 (백엔드/프론트엔드)

네트워크워, 테스트 (자동화)

ㅇ DB/SQL : DB/SQL, SQLD

Docker 프랫폼, 정보보안(암호화)

[시간표]

- 9시 ~ (출석)

- 점심 : 11시30분~오후1시.

- 끝 : 5시50분

\* 금요일 오후 : 평가, 설문

-----------------------------------------------------------------------------------------------------------------------------------

[파이썬 개발 환경을 구성]

\* Python 3.11 이상 + VSC (Extension)

\* Colab

\* <https://www.python.org/downloads/>

\* WIN+R : sysdm.cpl

\* [**https://code.visualstudio.com/**](https://code.visualstudio.com/)

\* 폴더 생성 : c:\kosa20250317

ㅇ 구글 Colab

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

[파이썬 언어의 특징]

\* 스크립트 언어

\* 동적 타이핑 (변수에 고유한 타입)

\* pypi 레포지토를 이용한 라이브러리 관리 : <https://pypi.org/> => pip 명령으로 설치/사용

\* 참고 링크 :

- <https://wikidocs.net/>

1) <https://wikidocs.net/book/1> (점프투파이썬)

2) <https://wikidocs.net/book/6353>

\* 들여쓰기를 이용한 블록 구조

if (a==0)

{

......

}

====>

if a==0:

...

...

[파이썬 언어의 기본 문법]

\* 모든 자료형, 연산자 (%, //)

리스트 : 여러 형식의 데이터를 모아서 사용

[‘hello’, 0]

딕셔러리 : JSON 타입

{‘name’ : ‘chris’, ‘age’ : 15}

data = ‘hello’ => ‘chris says “hello”’

“hello”

‘’’hello’’’

“””hello”””

\* if, while, for

\* 함수

def add(a,b):

return a+b

\* 모듈 중심의 언어

다른 코드를 내 코드에 포함해서 사용

import calc

pip install pandas ==> import pandas

==> 코드 예제

\* <https://wrtn.ai/>

\* 폴더 : ex-python

ㅇ 01-data.py

----------------

# 1. Number Types (int, float)

num\_int = 10

num\_float = 3.14

print("=== Numbers ===")

print(f"Integer: {num\_int}, Type: {type(num\_int)}")

print(f"Float: {num\_float}, Type: {type(num\_float)}")

print()

input("Press Enter to continue...")

# 2. String Type

text = "Hello Python!"

print("=== String ===")

print(f"String: {text}")

print(f"Length: {len(text)}")

print(f"Slicing: {text[0:5]}")

print()

input("Press Enter to continue...")

# 3. List Type (Mutable)

my\_list = [1, "Python", 3.14, [1, 2, 3]]

print("=== List ===")

print(f"Original list: {my\_list}")

my\_list.append("New Item")

print(f"After append: {my\_list}")

print(f"First item: {my\_list[0]}")

print(f"Last item: {my\_list[-1]}")

print(f"Nested list: {my\_list[3]}")

print(f"Nested item: {my\_list[3][1]}")

my\_list[1] = "Java"

print(f"After change: {my\_list}")

print(f"Slice: {my\_list[1:3]}")

print(f"Slice: {my\_list[:3]}")

print(f"Slice: {my\_list[3:]}")

print(f'lenght: {len(my\_list)}')

del my\_list[2:] # 2번째부터 끝까지 삭제

print(f"After delete: {my\_list}")

print(f'lenght: {len(my\_list)}')

print()

input("Press Enter to continue...")

# 4. Tuple Type (Immutable)

my\_tuple = (1, "Python", 3.14)

print("=== Tuple ===")

print(f"Tuple: {my\_tuple}")

print(f"First item: {my\_tuple[0]}")

print()

input("Press Enter to continue...")

# 5. Dictionary Type

my\_dict = {

"name": "John",

"age": 30,

"languages": ["Python", "JavaScript"],

10 : "number"

}

print("=== Dictionary ===")

print(f"Dictionary: {my\_dict}")

print(f"Name: {my\_dict['name']}")

print(f"Languages: {my\_dict['languages']}")

print(f'lenght: {len(my\_dict)}')

del my\_dict[10]

print(f"After delete: {my\_dict}")

print(f'lenght: {len(my\_dict)}')

my\_dict["languages"][1] = 'Java'

my\_dict.update({"name": "Jane", "age": 25})

print(f"After update: {my\_dict}")

print()

input("Press Enter to continue...")

# 6. list comprehension

print("=== List Comprehension ===")

my\_list = [x for x in range(10)]

print(f"List: {my\_list}")

my\_list = [x for x in range(10) if x % 2 == 0]

print(f"List: {my\_list}")

my\_list = [x\*\*2 for x in range(10)]

print(f"List: {my\_list}")

print()

input("Press Enter to continue...")

# 7. Shallow Copy vs Deep Copy

my\_list = [1, 2, 3]

your\_list = my\_list

your\_list[0] = 10

print("=== Shallow Copy ===")

print(f"Original list: {my\_list}")

print(f"Copied list: {your\_list}")

your\_list = my\_list.copy()

your\_list[1] = 20

print("=== Deep Copy ===")

print(f"Original list: {my\_list}")

print(f"Copied list: {your\_list}")

print()

input("Press Enter to continue...")

---------------

02-inout.py

---------------

# 1. Basic input and print examples

name = input("Enter your name: ")

age = input("Enter your age: ")

print(f"Hello {name}, you are {age} years old!")

print()

input("Press Enter to continue...")

# 2. File writing example using with statement

try:

with open('sample.txt', 'w') as file:

file.write("Hello, this is a test file!\n")

file.write(f"Written by {name}\n")

except IOError as e:

print(f"Error writing to sample.txt: {e}")

print()

input("Press Enter to continue...")

# 3. File writing example using explicit open() and close()

try:

file = open('sample2.txt', 'w')

try:

file.write("This is written using explicit open() and close()\n")

file.write(f"Written by {name}\n")

except IOError as e:

print(f"Error writing to file: {e}")

finally:

file.close()

except IOError as e:

print(f"Error opening sample2.txt: {e}")

print()

input("Press Enter to continue...")

# 4. File reading examples

try:

# Reading using with statement

try:

with open('sample.txt', 'r') as file:

content = file.read()

print("\nFile contents (sample.txt):")

print(content)

except IOError as e:

print(f"Error reading sample.txt: {e}")

# Reading using explicit open() and close()

try:

file = open('sample2.txt', 'r')

try:

content = file.read()

print("\nFile contents (sample2.txt):")

print(content)

except IOError as e:

print(f"Error reading file: {e}")

finally:

file.close()

except IOError as e:

print(f"Error opening sample2.txt: {e}")

# 6. Append to file

try:

file = open('sample2.txt', 'a')

try:

file.write("Additional line appended\n")

except IOError as e:

print(f"Error appending to file: {e}")

finally:

file.close()

except IOError as e:

print(f"Error opening file for append: {e}")

# 7. Read file line by line

print("\nReading file line by line:")

try:

file = open('sample2.txt', 'r')

try:

for line in file:

print(f"Line: {line.strip()}")

except IOError as e:

print(f"Error reading lines: {e}")

finally:

file.close()

except IOError as e:

print(f"Error opening file for reading lines: {e}")

except Exception as e:

print(f"Unexpected error occurred: {e}")

print()

input("Press Enter to continue...")

---------------

ㅇ 03-control.py

----------------------

# 1. if statement example

score = 85

if score >= 90:

print("Grade: A")

elif score >= 80:

print("Grade: B")

else:

print("Grade: C")

print()

input("Press Enter to continue...")

# 2. while loop example

count = 1

while count <= 5:

print(f"Count is: {count}")

count += 1

print()

input("Press Enter to continue...")

# 3. for loop examples

# Range based for loop

print("\nCounting from 1 to 5:")

for i in range(1, 6):

print(i)

print()

input("Press Enter to continue...")

# Iterating through a list

fruits = ["apple", "banana", "orange"]

print("\nFruit list:")

for fruit in fruits:

print(fruit)

print()

input("Press Enter to continue...")

# Nested loop example

print("\nMultiplication table (1-3):")

for i in range(1, 4):

for j in range(1, 4):

print(f"{i} x {j} = {i\*j}")

print()

input("Press Enter to continue...")

----------------------

ㅇ 04-function.py

-------------

# 기본적인 함수 정의와 호출

def greet(name):

return f"안녕하세요, {name}님!"

# 2개 반환값을 갖는 함수

def calc(a, b):

return a+b, a-b

# 기본값 매개변수를 가진 함수

def power(base, exponent=2):

return base \*\* exponent

# 가변 인자를 받는 함수

def sum\_all(\*args):

return sum(args)

# 키워드 인자를 받는 함수

def student\_info(\*\*kwargs):

for key, value in kwargs.items():

print(f"{key}: {value}")

# 원의 넓이를 구하는 함수

import math

def circle\_area(radius):

return math.pi \* radius \*\* 2

# 함수 사용 예제

if \_\_name\_\_ == "\_\_main\_\_":

# 기본 함수 호출

print(greet("chris"))

print()

input("Press Enter to continue...")

# 2개 반환값 함수 호출

add, sub = calc(10, 5)

print(f"10 + 5 = {add}, 10 - 5 = {sub}")

result = calc(10, 5)

print(f"10 + 5 = {result[0]}, 10 - 5 = {result[1]}")

print()

input("Press Enter to continue...")

# 기본값 매개변수 함수 호출

print(power(2)) # 2의 2제곱

print(power(2, 3)) # 2의 3제곱

print()

input("Press Enter to continue...")

# 가변 인자 함수 호출

print(sum\_all(1, 2, 3, 4, 5))

print()

input("Press Enter to continue...")

# 키워드 인자 함수 호출

student\_info(name="김윤수", age=20, grade="A")

print()

input("Press Enter to continue...")

# 사용자에게 값을 입력 받아 원의 넓이 계산

radius = float(input("반지름을 입력하세요: "))

print(f"반지름이 {radius}인 원의 넓이는 {circle\_area(radius)}입니다.")

print()

input("Press Enter to continue...")

-------------

ㅇ 05-lambda.py

----------

# Lamda

# 일반적인 함수

def add(a, b):

return a + b

result = add(1,2)

print(result)

print()

input("Press Enter to continue...")

# 간단한 람다함수

add = lambda a, b: a + b

result = add(1,3)

print(result)

print()

input("Press Enter to continue...")

# 람다 함수 정의와 함께 호출

result = (lambda a, b: a + b)(1,4)

print(result)

print()

input("Press Enter to continue...")

----------

ㅇ data.csv

--------------

name,age,grade

Alice,30,A

Bob,25,B

Charlie,35,C

-----------------

ㅇ calculator.py

--------

def add(a, b):

return a + b

def subtract(a, b):

return a - b

def multiply(a, b):

return a \* b

def divide(a, b):

if b == 0:

raise ValueError("Cannot divide by zero")

return a / b

---------

ㅇ 06-module.py

-------------------

import calculator

# pip install pandas

import pandas as pd

# pip install numpy

import numpy as np

if \_\_name\_\_ == "\_\_main\_\_":

a = 10

b = 5

print(f"{a} + {b} = {calculator.add(a, b)}")

print(f"{a} - {b} = {calculator.subtract(a, b)}")

print(f"{a} \* {b} = {calculator.multiply(a, b)}")

print(f"{a} / {b} = {calculator.divide(a, b)}")

print()

input("Press Enter to continue...")

# CSV 파일 읽기, pandas dataframe 예제

df = pd.read\_csv('data.csv')

print("\nCSV 파일 내용:")

print(df)

print()

input("Press Enter to continue...")

print("\nDataFrame 예제:")

data = {

'Name': ['John', 'Anna', 'Peter', 'Linda'],

'Location': ['New York', 'Paris', 'Berlin', 'London'],

'Age': [24, 13, 53, 33]

}

df = pd.DataFrame(data)

print(df)

print(df.describe())

print()

input("Press Enter to continue...")

print(df[df.Age > 30])

print(df[df.Age > 30].Name)

print()

input("Press Enter to continue...")

print(df['Name'])

print(df[['Name', 'Age']])

df.to\_csv('output.csv', index=False)

print()

input("Press Enter to continue...")

# numpy 예제

print("\nNumpy 예제:")

a = np.array([1, 2, 3])

print(a)

print(type(a))

print()

input("Press Enter to continue...")

b = np.array([[1, 2, 3], [4, 5, 6]])

print(b)

print(type(b))

print(b.shape)

print()

input("Press Enter to continue...")

data = np.random.randn(2, 3)

print(data)

# data 값을 계산하는 예제

print(data \* 10)

print(data + data)

print()

# data 형식을 변경하는 예제

data = data.astype(int)

print(data)

data = data.reshape(3, 2)

print(data)

print()

input("Press Enter to continue...")

print(np.zeros((2, 3)))

print()

input("Press Enter to continue...")

-------------------

ㅇ 블록체인

- 블록체인, 비트코인?

- 분산된 환경에서 참여자들의 상호 협동에 의해서 관리. (거래원장)

- 해시함수 : 대표적인 단방향 암호화, 무결성

1) 단방향 : 암호화는 가능한데, 수학적인 방법으로 복호화할 수 없다. (원문x)

2) 가변길이 입력을 고정길이의 출력

3) 입력값이 다르면 출려값도 다르다. => 출력값이 같으면 입력값도 같다

MD1 == MD2 ====> 입력값1 == 입력값2

- 머클트리, 머클트리 루트

\* 블록체인 ==> 블럭 + 체인

==> (헤더+데이터) + 체인

[빅데이터]

\* 특징 : 정형/비정형 데이터 (다양한 데이터)

매우 큰 볼륨의 데이터 (대용량)

빠르게 증가하는 데이터의 양

=> 기술과 인프라, 도구가 필요하다.

\* 비정형 : 1, 2, 1, 4 ........ {‘온도’ : 1.....}

1.2, 2.3. 4,1 .... {‘온도’ : 1.2}

1.2 + 45% {‘온도’ : 1.2, ‘습도: 45}

1) 데이터를 수집

2) 전송 / 저장 (N/W)

3) 전처리/가공

= 월 화 수 목 금 토 일

10 10 11 10 10 ---- 20 : 이상치

결측치

ㅇ 머신러닝(딥러닝) 어떤 문제를 해결한다?

1) 학습을 위한 데이터를 준비 (별, 동그라미)

2) 데이터에서 특징을 추출 (특징 공학)

3) 학습위한 데이터 (학습용), 검증위한 테스트 (테스트용)

4) 머신러닝(딥러닝) 모델을 준비

5) 학습을 진행한다. (결정경계를 찾는다. 오차를 최소하는 방법으로 가중치를 변경하며 계산)

6) 학습 완료 => 가중치

7) 검증용 데이터를 이용해서 평가 (99.0 %, 0.99)

8) 사용 : 우리가 알고 싶은 데이터를 입력해서... ===> 모델 + 가중치 ==> 결과 (신뢰)

※ 신뢰하고 사용한다. - 100% X

Q) 교수님 => 학점을 계산하는 프로그램을 작성하라.

국어, 수학 ==> 평균이 90 (A), 80 (B), 기타 (C)

String getGrade(국어, 수학)  
 {

평균 = (국어 + 수학) / 2

if (90<=평균)

return ‘A’

..  
 }

**==> 머신러닝/인공지능.**

1) 학점에 영향을 주는 데이터와 속성 : 국어, 수학 => A, B, C

2) 교수님, 지난 3년간 학생들의 학점 평가 테이블 (2만)

----------------------------------------------------------------------------------

국어 수학 학점

----------------------------------------------------------------------------------

90 90 A

10 20 B

...

..

----------------------------------------------------------------------------------

3) 16000(학습), 4000건 (검증)

4) SVM, DNN (심층신경망) --> 가중치 (Weight)

5) 검증 데이터 4000건을 이용해서 모델을 평가 ==> 99.9%

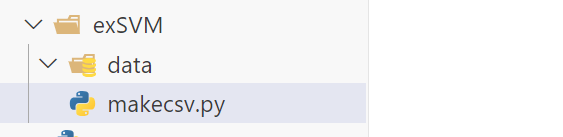
6) SVM / DNN + 가중치 (Weight)를 교수님께 S/W 구성해서 드린다.

<실습 준비>

ㅇ Visual Studio Code 를 실행 => kosa20250317 폴더 오픈

ㅇ Colab => kosa20250317 ipynb 파일 오픈

ㅇ 폴더 : testSVM 폴더



\* <https://wrtn.ai/>

# 점수를 계산해서 평균에 따른 학점을 반환

# 편의상 A, B, C로 구분함.

def get\_grade(kor, math):

mean = int((kor + math) / 2)

grade = "C"

if 70 <= mean:

grade = "A"

elif 40 <= mean:

grade = "B"

return grade

Q) makecsv.py를 위 소스코드(함수)를 이용해서 코딩

- data 폴더 안에 grade.csv 파일을 만든다.

- grade.csv의 내용:

KOR,MATH,GRADE

90 ,90,A

... 2만건.

---------------

# 데이터 준비

import random

import os

# 데이터가 저장될 폴더 (data 폴더가 없으면 생성)

DATA\_FILE\_PATH = os.path.join(os.path.dirname(\_\_file\_\_), "data")

if not os.path.exists(DATA\_FILE\_PATH):

os.makedirs(DATA\_FILE\_PATH)

# 점수를 계산해서 평균에 따른 학점을 반환

# 편의상 A, B, C로 구분함.

def get\_grade(kor, math):

mean = int((kor + math) / 2)

grade = "C"

if 70 <= mean:

grade = "A"

elif 40 <= mean:

grade = "B"

return grade

# 20000개 데이터를 만들어 CSV에 기록한다.

csvname = os.path.join(DATA\_FILE\_PATH, "grade.csv")

fp = open(csvname,"w",encoding="utf-8")

fp.write("KOR,MATH,GRADE\r\n")

for i in range(20000):

kor = random.randint(10, 100)

math = random.randint(10, 100)

grade = get\_grade(kor, math)

fp.write("{0},{1},{2}\r\n".format(kor, math, grade))

fp.close()

print("Done")

----------------------------

ㅇ testSVM.py

-------------

# pip install scikit-learn pandas

from sklearn import svm, metrics

from sklearn.model\_selection import train\_test\_split

import pandas as pd

import os

# ImportError: cannot import name 'joblib' from 'sklearn.externals'

# https://gaussian37.github.io/ml-sklearn-saving-model/

import joblib

DATA\_FILE\_PATH = os.path.join(os.path.dirname(\_\_file\_\_), "data")

# weight를 만들어주는 함수

def makeweights(csvFile):

grade = pd.read\_csv(csvFile)

# 데이터 전처리 (학습을 위하여 데이터를 분리하고 가공-정규화)

label = grade["GRADE"]

kor = grade["KOR"] / 100

math = grade["MATH"] / 100

data = pd.concat([kor, math], axis=1)

# 학습 및 테스트 데이터로 분리

data\_train, data\_test, label\_train, label\_test = \

train\_test\_split(data, label)

# 학습하기

model = svm.SVC()

model.fit(data\_train, label\_train)

# 모델 (weight) 저장

weightsname = os.path.join(DATA\_FILE\_PATH, "model\_weights.pkl")

joblib.dump(model, weightsname)

# model = joblib.load(weightsname)

# 테스트 데이터로 에측하기 (평가를 위함)

predict = model.predict(data\_test)

# 결과 확인하고 평가하기

ac\_score = metrics.accuracy\_score(label\_test, predict)

cl\_report = metrics.classification\_report(label\_test, predict)

print("Accuracy =", ac\_score)

print("Report =\n", cl\_report)

def runsvm(kor, math):

weightsname = os.path.join(DATA\_FILE\_PATH, "model\_weights.pkl")

if not os.path.exists(weightsname):

return "모델이 없습니다. 학습을 먼저 해주세요."

model = joblib.load(weightsname)

# 테스트 데이터로 에측하기

new\_data = pd.DataFrame({

'KOR': [kor/100],

'MATH': [math/100]

})

predict = model.predict(new\_data)

return predict[0]

if \_\_name\_\_ == "\_\_main\_\_":

csvname = os.path.join(DATA\_FILE\_PATH, "grade.csv")

if not os.path.exists(csvname):

print("데이터 파일이 없습니다. makecsv.py를 먼저 실행하세요.")

exit()

makeweights(csvname)

kor = 90

math = 80

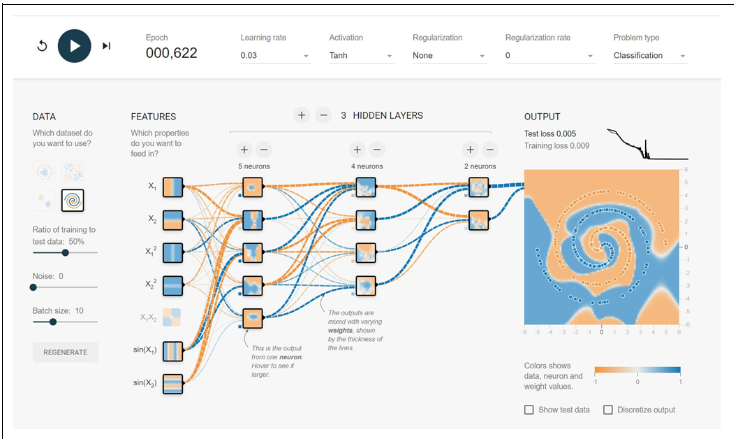
grade = runsvm(kor, math)

print(f'kor={kor}, math={math} => grade={grade}')

print("Done")

--------------

<https://playground.tensorflow.org/>



==> 점심 맛있게 드세요~~

<CSV 준비>

--------------------

#데이터 준비

import random

# 점수를 계산해서 평균에 따른 학점을 반환

# 편의상 A, B, C로 구분함.

def get\_grade(kor, math):

mean = int((kor + math) / 2)

grade = "C"

if 70 <= mean:

grade = "A"

elif 40 <= mean:

grade = "B"

return grade

# 20000개 데이터를 만들어 CSV에 기록한다.

fp = open("grade.csv","w",encoding="utf-8")

fp.write("KOR,MATH,GRADE\r\n")

for i in range(20000):

kor = random.randint(10, 100)

math = random.randint(10, 100)

grade = get\_grade(kor, math)

fp.write("{0},{1},{2}\r\n".format(kor, math, grade))

fp.close()

----------------

<데이터의 특징 보기>

----------------

# 특징 보기

import matplotlib.pyplot as plt

import pandas as pd

# pands로 csv 파일 읽기

grade = pd.read\_csv("grade.csv", index\_col=2)

# 그래프로 표시하기

fig = plt.figure()

ax = fig.add\_subplot(1, 1, 1)

# Lable 색상 표시하는 함수

def scatter(label\_text, color):

b = grade.loc[label\_text]

ax.scatter(b["KOR"],b["MATH"], c=color, label=label\_text)

scatter("A", "red")

scatter("B", "blue")

scatter("C", "yellow")

ax.legend()

----------------

[딥러닝]

----------------

#딥러닝

from keras.models import Sequential

from tensorflow.keras.layers import Dense, Dropout, Activation

from keras.callbacks import EarlyStopping

import pandas as pd, numpy as np

# grade가 저장된 CSV 파일 읽기

grade = pd.read\_csv("grade.csv")

#데이터 전처리 (MLP 학습을 위해 데이터를 가공)

grade["KOR"] /= 100 # 점수는 최대 100점

grade["MATH"] /= 100

X = grade[["KOR", "MATH"]].to\_numpy() # 입력데이터 (독립변수)

lable\_class = {"A":[1,0,0], "B":[0,1,0], "C":[0,0,1]} # Label (종속변수)

Y = np.empty((20000,3))

for i, j in enumerate(grade["GRADE"]):

Y[i] = lable\_class[j]

# 학습 데이터와 테스트 데이터 분할

X\_train, Y\_train = X[1:15001], Y[1:15001]

X\_test, Y\_test = X[15001:20001], Y[15001:20001]

# 학습을 위한 모델 구성하기

model = Sequential()

model.add(Dense(512, input\_shape=(2,)))

model.add(Activation('relu'))

model.add(Dropout(0.1))

model.add(Dense(512))

model.add(Activation('relu'))

model.add(Dropout(0.1))

model.add(Dense(3))

model.add(Activation('softmax'))

model.compile(

loss='categorical\_crossentropy',

optimizer="rmsprop",

metrics=['accuracy'])

model.summary()

# 모델을 이용한 학습

model.fit(

X\_train, Y\_train,

batch\_size=100,

epochs=20,

validation\_split=0.1,

callbacks=[EarlyStopping(monitor='val\_loss', patience=2)],

verbose=1)

# 테스트 데이터를 이용한 평가

score = model.evaluate(X\_test, Y\_test)

print('loss=', score[0])

print('accuracy=', score[1])

# 생성한 모델 이용

X\_Data = X[0:10]

result = model.predict(X\_Data)

print(result)

predicted = result.argmax(axis=-1)

print(predicted)

--------------------------

1시 50분~ 시작하겠습니다.

김윤수 교수 / 인공지능학과이(가) 예약된 Zoom 회의에 귀하를 초대합니다.

Zoom 회의 참가

<https://sjcu-ac-kr.zoom.us/j/81463749747?pwd=jxS4L1XzNoGaaoEN4Sxfbzhrh1ZPhx.1>

회의 ID: 814 6374 9747

암호: 809415

ㅇ Color Space (색상공간)

- RGB

- YCbCr (YUV) => Y(8, 휘도) + CbCr (8+8, 색차)

Y 1/2, 1/2 ==> YCbCr 4:2:0

ㅇ 폴더 : imageVideo

- <https://bit.ly/KOSA20250317>

pip install opencv-python

ㅇ ex01-image.py

----------------------------------------

# pip install opencv-python (이것 사용하기)

# pip install opencv-contrib-python

import cv2

import numpy as np

import os

# 데이터 관리를 위한 폴더 (data 폴더가 없으면 생성)

DATA\_FILE\_PATH = os.path.join(os.path.dirname(\_\_file\_\_), "data")

if not os.path.exists(DATA\_FILE\_PATH):

os.makedirs(DATA\_FILE\_PATH)

# 원본 이미지 읽고 표시

imageFile = os.path.join(DATA\_FILE\_PATH, 'lena.jpg')

img = cv2.imread(imageFile)

cv2.imshow('Lena color',img)

# 다양한 형식으로 저장

imageFile = os.path.join(DATA\_FILE\_PATH, 'Lena.bmp')

cv2.imwrite(imageFile, img)

imageFile = os.path.join(DATA\_FILE\_PATH, 'Lena.png')

cv2.imwrite(imageFile, img)

imageFile = os.path.join(DATA\_FILE\_PATH, 'Lena2.jpg')

cv2.imwrite(imageFile, img, [cv2.IMWRITE\_JPEG\_QUALITY, 90])

# GRAYSCALE로 변환하여 읽고 표시

imageFile = os.path.join(DATA\_FILE\_PATH, 'Lena.png')

img = cv2.imread(imageFile, cv2.IMREAD\_GRAYSCALE)

cv2.imshow('Lena gray',img)

# 사용자가 키를 입력하면 창을 닫고 종료하기

cv2.waitKey()

cv2.destroyAllWindows()

--------------------------------------

ㅇ ex02-image.py (코드) :

---------------------

lena.jpg을 읽는다.

- 선, 사각형 그린다

- 임의의 영역을 검은색으로 채운다.

- 텍스트를 이미지 위에 출력한다.

=> 편집한 데이터를 lena2.jpg로 저장한다.

=> 원본, 편집한 이미지를 화면에 표시한다

-----------------------------

import cv2

import numpy as np

import os

# 데이터 관리를 위한 폴더 (data 폴더가 없으면 생성)

DATA\_FILE\_PATH = os.path.join(os.path.dirname(\_\_file\_\_), "data")

if not os.path.exists(DATA\_FILE\_PATH):

os.makedirs(DATA\_FILE\_PATH)

# 원본 이미지를 읽고 화면에 표시

imageFile = os.path.join(DATA\_FILE\_PATH, 'lena.jpg')

img = cv2.imread(imageFile)

cv2.imshow('Lena Original',img)

# 선과 사각형 그리기

pt1 = 100, 100

pt2 = 300, 300

cv2.rectangle(img, pt1, pt2, (0, 255, 0), 2)

cv2.line(img, (0, 0), (400, 0), (255, 0, 0), 10)

cv2.line(img, (0, 0), (0, 400), (0,0,255), 10)

# 특정 영역의 색상 변경

img[10:50, 10:50] = 0

# 텍스트 출력

font = cv2.FONT\_HERSHEY\_SIMPLEX

text = "Hello"

cv2.putText(img, text, (300, 300), font, 1, (255, 255, 255), 4)

# 편집한 이미지를 저장하고 다시 읽기 (단, 저장시 낮은 품질로 저장해 보기)

cv2.imshow('Lena Edited',img)

imageFile = os.path.join(DATA\_FILE\_PATH, 'Lena2.jpg')

cv2.imwrite(imageFile, img, [cv2.IMWRITE\_JPEG\_QUALITY, 2])

img = cv2.imread(imageFile)

cv2.imshow('Lena 2nd',img)

# 임의의 키를 누르면 창을 닫고 종료

cv2.waitKey()

cv2.destroyAllWindows()

-------------------------

\* 이미지의 특징? 어떤 것이 이미지의 특징?

- RGB Pixel

- 히스토그램

- Edge (외곽선)

- OCR (Text)

ㅇ ex03-image.py

---------------------

# Hough Line Transform을 이용한 직선 검출

import numpy as np

import cv2 as cv

import os

# 데이터 관리를 위한 폴더 (data 폴더가 없으면 생성)

DATA\_FILE\_PATH = os.path.join(os.path.dirname(\_\_file\_\_), "data")

if not os.path.exists(DATA\_FILE\_PATH):

os.makedirs(DATA\_FILE\_PATH)

# 도로 이미지 읽어서 표시

imgfile = os.path.join(DATA\_FILE\_PATH, 'Roads.png')

cimg = cv.imread(imgfile)

cv.imshow('Roads - original', cimg)

# 도로이미지를 GRAY로 변환 후 이진화, 에지 추출 후 표시

img = cv.cvtColor(cimg, cv.COLOR\_BGR2GRAY)

\_, img = cv.threshold(img, 100, 255, cv.THRESH\_BINARY)

img = cv.Canny(img, 50, 200, None, 3)

cv.imshow('Image - binary & edge', img)

# 사용자 입력키 대기 및 종료

cv.waitKey()

cv.destroyAllWindows()

-------------------------

ㅇ ex04-movie.py

----------------------------

import cv2

import numpy as np

import os

# 데이터 관리를 위한 폴더 (data 폴더가 없으면 생성)

DATA\_FILE\_PATH = os.path.join(os.path.dirname(\_\_file\_\_), "data")

if not os.path.exists(DATA\_FILE\_PATH):

os.makedirs(DATA\_FILE\_PATH)

dataFile = os.path.join(DATA\_FILE\_PATH,'road.mp4')

# 동영상 파일을 읽어서 표시

cap=cv2.VideoCapture(dataFile)

while(True):

retval, frame = cap.read()

if not retval:

break

cv2.imshow('frame',frame)

key = cv2.waitKey(25)

if key == 27: # Esc

break

cap.release()

cv2.destroyAllWindows()

--------------------------

ㅇ ex05-movie.py

----------

import cv2

import numpy as np

import os

# 데이터 관리를 위한 폴더 (data 폴더가 없으면 생성)

DATA\_FILE\_PATH = os.path.join(os.path.dirname(\_\_file\_\_), "data")

if not os.path.exists(DATA\_FILE\_PATH):

os.makedirs(DATA\_FILE\_PATH)

dataFile = os.path.join(DATA\_FILE\_PATH,'road.mp4')

# 동영상 파일을 읽어서 표시

cap=cv2.VideoCapture(dataFile)

while(True):

retval, frame = cap.read()

if not retval:

break

cv2.imshow('frame',frame)

# 이미지를 GRAY로 변환, 에지 추출 후 표시

gray = cv2.cvtColor(frame, cv2.COLOR\_BGR2GRAY)

edges = cv2.Canny(gray,100,200)

cv2.imshow('edges',edges)

key = cv2.waitKey(25)

if key == 27: # Esc

break

cap.release()

cv2.destroyAllWindows()

------------------

ㅇ ex07-youtube.py

------------------

import cv2

import numpy as np

# https://pypi.org/project/cap-from-youtube/

# pip install cap-from-youtube

from cap\_from\_youtube import cap\_from\_youtube

# YouTube 동영상을 읽어서 표시

url = 'https://www.youtube.com/watch?v=eA-7Bu1yCJQ'

cap = cap\_from\_youtube(url, '480p')

while(True):

retval, frame = cap.read()

if not retval:

break

cv2.imshow('frame',frame)

# 이미지를 GRAY로 변환, 에지 추출 후 표시

gray = cv2.cvtColor(frame, cv2.COLOR\_BGR2GRAY)

edges = cv2.Canny(gray,100,200)

cv2.imshow('edges',edges)

key = cv2.waitKey(25)

if key == 27: # Esc

break

cap.release()

cv2.destroyAllWindows()

-----------------

ㅇ ex08-matching.py

---------------------------

import cv2

import os

# 데이터 관리를 위한 폴더 (data 폴더가 없으면 생성)

DATA\_FILE\_PATH = os.path.join(os.path.dirname(\_\_file\_\_), "data")

if not os.path.exists(DATA\_FILE\_PATH):

os.makedirs(DATA\_FILE\_PATH)

# 사람 이미지를 읽고 가로/세로 크기를 구함

image\_file = os.path.join(DATA\_FILE\_PATH,'wili.png')

template = cv2.imread(image\_file)

th, tw = template.shape[:2]

cv2.imshow('template', template)

# 배경 이미지를 준비

image\_file = os.path.join(DATA\_FILE\_PATH,'original.jpg')

image = cv2.imread(image\_file)

# 이미지 특징 매칭

result = cv2.matchTemplate(image, template, cv2.TM\_CCOEFF\_NORMED)

# 매칭 좌표를 이미지에 그리기

minVal, maxVal, minLoc, maxLoc = cv2.minMaxLoc(result)

top\_left = maxLoc

match\_val = maxVal

bottom\_right = (top\_left[0] + tw, top\_left[1] + th)

cv2.rectangle(image, top\_left, bottom\_right, (0, 0, 255), 2)

cv2.imshow('Result', image)

# 키 입력시 창을 닫고 종료

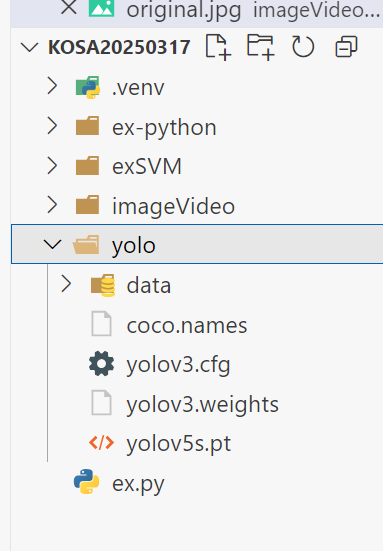
cv2.waitKey(0)

cv2.destroyAllWindows()

------------------------

[Code\_Data](https://drive.google.com/drive/folders/18A-qLdo-6Fs34QFBABEkVjr7DKAQ3ybc)

\* 악성코드 탐지.



ㅇ testyolo3.py

--------

import cv2

import numpy as np

import os

# 데이터 관리를 위한 폴더 (data 폴더가 없으면 생성)

WORKING\_PATH = os.path.dirname(\_\_file\_\_)

DATA\_FILE\_PATH = os.path.join(WORKING\_PATH, "data")

if not os.path.exists(DATA\_FILE\_PATH):

os.makedirs(DATA\_FILE\_PATH)

WEIGHTS\_FILE = os.path.join(WORKING\_PATH, "yolov3.weights")

CFG\_FILE = os.path.join(WORKING\_PATH, "yolov3.cfg")

# YOLO model configuration과 weights를 로드

net = cv2.dnn.readNet(WEIGHTS\_FILE, CFG\_FILE)

# Class labels를 Load (COCO names)

NAMES\_FILE = os.path.join(WORKING\_PATH, "coco.names")

with open(NAMES\_FILE, "r") as f:

classes = [line.strip() for line in f.readlines()]

# 분석 대상 파일 읽기

IMAGE\_FILE = os.path.join(DATA\_FILE\_PATH, "animal.jpeg")

img = cv2.imread(IMAGE\_FILE)

height, width, \_ = img.shape

# YOLO를 이용한 분석

blob = cv2.dnn.blobFromImage(img, 1/255.0, (416, 416), swapRB=True, crop=False)

net.setInput(blob)

# Get output layer names

output\_layers\_names = net.getUnconnectedOutLayersNames()

# Forward pass through YOLO

layer\_outputs = net.forward(output\_layers\_names)

# 점검 결과 분석 및 활용

boxes = []

confidences = []

class\_ids = []

for output in layer\_outputs:

for detection in output:

scores = detection[5:]

class\_id = np.argmax(scores)

confidence = scores[class\_id]

if confidence > 0.5: # Confidence threshold

center\_x = int(detection[0] \* width)

center\_y = int(detection[1] \* height)

w = int(detection[2] \* width)

h = int(detection[3] \* height)

x = int(center\_x - w / 2)

y = int(center\_y - h / 2)

boxes.append([x, y, w, h])

confidences.append(float(confidence))

class\_ids.append(class\_id)

# Apply non-maximum suppression to eliminate redundant overlapping boxes

indices = cv2.dnn.NMSBoxes(boxes, confidences, 0.5, 0.4)

# Draw bounding boxes and labels on the image

if len(indices) > 0:

for i in indices.flatten():

x, y, w, h = boxes[i]

label = str(classes[class\_ids[i]])

confidence = str(round(confidences[i], 2))

color = (0, 255, 0) # Green color for boxes

cv2.rectangle(img, (x, y), (x + w, y + h), color, 2)

cv2.putText(img, label + " " + confidence, (x, y - 10), cv2.FONT\_HERSHEY\_SIMPLEX, 0.5, color, 2)

# Display the image with detections

cv2.imshow("Image", img)

cv2.waitKey(0)

cv2.destroyAllWindows()

--------------------------

yolov3 (OpenCV를 이용해서) 실행. ‘코리리를 분석’

==> 자동차 동영상을 이용해서 탐지하는 것으로 수정 !!!

[수요일]

\* Text를 중심으로 서비스를 해야 한다.

1) 데이터 공학 : Text, 어떤 속성이 중요한가

2) Text 데이터는 어디에 있는가? 어떻게 수집/가져올 수 있는가?

- 회사의 DBMS (MariaDB, Oracle) : SQL 도구

- 파일 형태 : 자체적으로 분석

- 웹 : 웹 클로링

- 특정 시스템/서비스 안에 있다. : OpenAPI (공개된 API)

3) 특징 공학

- 형태소 --> 명사, 워드카운트

- 인코딩

....

ㅇ Visual Studio Code -> kosa~~~~~ 폴더 오픈

ㅇ 서브폴더 : textprocessing

\* HTML이 있다. ==> HTML에서 정보를 어떻게 가져올까?

\* GPT, 뤼튼, 웹 검색

ㅇ exbs.py

------------------------

# pip install beautifulsoup4

# https://www.crummy.com/software/BeautifulSoup/bs4/doc/

# http://www.w3school.com/

from bs4 import BeautifulSoup

html\_doc = """

<html>

<head><title>The Dormouse's story</title></head>

<body>

<p class="title"><b>The Dormouse's story</b></p>

<p class="story">Once upon a time there were three little sisters

<a href="http://example.com/elsie" class="sister" id="link1">Elsie</a>,

<a href="http://example.com/lacie" class="sister" id="link2">Lacie</a> and

<a href="http://example.com/tillie" class="sister" id="link3">Tillie</a>;

and they lived at the bottom of a well.</p>

<p class="story">...</p>

</body>

</html>

"""

soup = BeautifulSoup(html\_doc, 'html.parser')

print(soup.title)

print(soup.title.parent.name)

print(soup.p)

print(soup.a)

for link in soup.find\_all('a'):

print(link.get('href'))

-------------------------

ㅇ exreq.py

-----------

# pip install requests

#pip install beautifulsoup4

import requests

from bs4 import BeautifulSoup

import os

# 데이터 관리를 위한 폴더 (data 폴더가 없으면 생성)

DATA\_FILE\_PATH = os.path.join(os.path.dirname(\_\_file\_\_), "data")

if not os.path.exists(DATA\_FILE\_PATH):

os.makedirs(DATA\_FILE\_PATH)

# 네이버에 '개발자'를 검색한 결과의 URL

url = "https://search.naver.com/search.naver?query=개발자"

# 요청을 보내고 응답을 받아옴

response = requests.get(url)

response.raise\_for\_status() # 요청에 실패한 경우 예외를 발생시킴

# 파일 저장

file\_path = os.path.join(DATA\_FILE\_PATH, 'search\_result.html')

with open(file\_path, 'w', encoding='utf-8') as file:

file.write(response.text)

# 저장한 파일을 분석

with open(file\_path, 'r', encoding='utf-8') as file:

html = file.read()

soup = BeautifulSoup(html, 'html.parser')

# class 이름이 total\_tit에 해당하는 웹 페이지의 이름을 찾아서 출력

for title in soup.find\_all(class\_="total\_tit"):

print(title.get\_text())

-----------

ㅇ exWeather.py

- 네이버에 날씨를 검색

- 현재 온도를 출력

==> 출력

===> 10시 15분~~

-----------

import requests

from bs4 import BeautifulSoup

# URL 설정

url = "https://search.naver.com/search.naver?where=nexearch&sm=top\_hty&fbm=0&ie=utf8&query=%EB%82%A0%EC%94%A8"

# 웹 페이지 요청

headers = {"User-Agent": "Mozilla/5.0"} # 네이버는 User-Agent 헤더가 필요할 수 있습니다.

response = requests.get(url, headers=headers)

# BeautifulSoup으로 HTML 파싱

soup = BeautifulSoup(response.text, "html.parser")

# 현재 온도 추출

temperature\_div = soup.find("div", class\_="temperature\_text")

if temperature\_div:

temperature = temperature\_div.get\_text(strip=True)

print(f"현재 온도 텍스트 : {temperature}")

else:

print("온도 정보를 찾을 수 없습니다.")

----------

ㅇ exselenium.py

----------------------------------

# pip install selenium

# https://chromedriver.chromium.org/

from selenium import webdriver

from selenium.webdriver.chrome.options import Options

import os

# 데이터 관리를 위한 폴더 (data 폴더가 없으면 생성)

DATA\_FILE\_PATH = os.path.join(os.path.dirname(\_\_file\_\_), "data")

if not os.path.exists(DATA\_FILE\_PATH):

os.makedirs(DATA\_FILE\_PATH)

options = Options()

options.add\_argument('--headless')

driver = webdriver.Chrome(options=options)

url = "http://www.naver.com/"

driver.get(url)

driver.implicitly\_wait(2)

file\_path = os.path.join(DATA\_FILE\_PATH, 'webCapture.png')

driver.get\_screenshot\_as\_file(file\_path)

driver.quit()

----------------------------------

ㅇ exseleniumNaver.py

---------------------------------

# pip install selenium

import time

from selenium import webdriver

from selenium.webdriver.common.by import By

from selenium.webdriver.chrome.options import Options

options = Options()

options.add\_argument("user-agent=Mozilla/5.0")

options.add\_argument("--start-maximized")

driver = webdriver.Chrome(options=options)

url = "https://www.naver.com/"

driver.get(url)

time.sleep(2)

inputElement = driver.find\_element(By.NAME,"query")

inputElement.clear()

time.sleep(1)

inputElement.send\_keys("개발자")

time.sleep(1)

inputElement.submit()

time.sleep(1)

results = driver.find\_elements(By.CLASS\_NAME,'total\_tit')

time.sleep(1)

for one in results:

print(one.text)

time.sleep(1)

results[0].click()

time.sleep(2)

# WebDriver를 종료합니다. (브라우저 닫기)

driver.quit()

--------------------

ㅇ exsqlite.py

<요구사항>

- DB는 sqlite를 사용합니다. (DB 파일 경로는 임의로)

- 다음과 같이 데이터를 준비하여 sqlite에 저장

1) user 정보 : 이름(한국어), 나이(숫자), 이메일(영문)

=> 100개 생성

=> 테스트 데이터.

※ faker

2) sqlite에 저장

=> table 설계 / 만들어야 한다. (create~~~)

=> insert 구문으로 데이터 저장

※ sqlite3 (설치된 상태)

3) DB에 저장된 자료 중에서 30살 이상의 데이터를 검색해서 출력

SELECT ~ WHERE ....

=> 출력 형식은 원하는 대로 한다.

------------------

import sqlite3

import random

import string

from faker import Faker

import os

# 데이터 관리를 위한 폴더 (data 폴더가 없으면 생성)

DATA\_FILE\_PATH = os.path.join(os.path.dirname(\_\_file\_\_), "data")

if not os.path.exists(DATA\_FILE\_PATH):

os.makedirs(DATA\_FILE\_PATH)

DBNAME = os.path.join(DATA\_FILE\_PATH, 'users.db')

# 한국 이름과 영문 이메일을 생성하기 위한 Faker 초기화

fake\_kr = Faker('ko\_KR') # 이름을 위한 한국어 로케일

fake\_en = Faker('en\_US') # 이메일을 위한 영어 로케일

# 랜덤 사용자 데이터 생성 함수

def generate\_users(count=100):

users = []

for i in range(1, count+1):

user\_id = i

name = fake\_kr.name()

age = random.randint(10, 60)

email = fake\_en.email() # 영문 이메일 사용

users.append((user\_id, name, age, email))

return users

# 100명의 사용자 생성

users = generate\_users(100)

# 2. 처음 5명의 사용자 정보 출력

print("처음 5명의 사용자:")

for i in range(5):

print(f"ID: {users[i][0]}, 이름: {users[i][1]}, 나이: {users[i][2]}, 이메일: {users[i][3]}")

# 3. 모든 사용자 정보를 SQLite 데이터베이스에 저장

print("\n사용자 정보를 데이터베이스에 저장 중...")

conn = sqlite3.connect(DBNAME)

cursor = conn.cursor()

# 사용자 테이블이 없으면 생성

cursor.execute('''

CREATE TABLE IF NOT EXISTS users (

id INTEGER PRIMARY KEY,

name TEXT,

age INTEGER,

email TEXT

)

''')

# 사용자 정보 데이터베이스에 삽입

cursor.executemany('INSERT INTO users VALUES (?, ?, ?, ?)', users)

conn.commit()

print(f"데이터베이스에 {len(users)}명의 사용자 정보가 저장되었습니다.")

# 4. 나이가 30살 이상인 사용자 검색 및 출력

print("\n30세 이상의 사용자:")

cursor.execute('SELECT \* FROM users WHERE age >= 30')

older\_users = cursor.fetchall()

# 검색 결과 출력

for user in older\_users:

print(f"ID: {user[0]}, 이름: {user[1]}, 나이: {user[2]}, 이메일: {user[3]}")

# 연결 종료

conn.close()

-------------------

ㅇ nlp 폴더

ㅇ testOkt.py

-----------------------

# 형태소 분석

# pip install konlpy

from konlpy.tag import Okt

text = '''국립 한국방송통신대학교(고성환 총장/이하 방송대)는 지난 25일 타슈켄트 한국교육원(이영웅 원장/이하 한국교육원)과 교육 협력 체제 구축 및 활성화를 위한 MOU를 체결했다고 28일 밝혔다.

타슈켄트 한국교육원에서 진행된 협약식에는 방송대 고성환 총장, 장영재 기획처장, 김재형 국제협력단장 등 관계자들이 참석해 우즈베키스탄 기묘국제대학교, 슈켄트 아주대학교를 비롯한 주요 교육 기관들과의 협력 확대도 논의했다.

2015년 11월 첫 협약을 맺은 양 기관은 한국어 교육 콘텐츠 공동 개발과 학술 연구 활동 지원 등 다양한 분야에서 긴밀한 협력을 해왔다. 이를 바탕으로 MOU수정 체결을 통해 한국학 콘텐츠 확대 보급을 위한 프로그램 개발 등 다양한 분야에서 협력을 강화해 학문적 교류를 확대하는 데 중점을 두고 있다.

특히, 중앙아시아 지역에서 외국인 유학생 유치 기반을 강화하고, 중앙아시아 대학들과 복수학위 프로그램 개설 가능성도 논의되어 향후 국제적 교육 교류가 더욱 확대될 전망이다.

타슈켄트 한국교육원은 우즈베키스탄의 수도 타슈켄트에 위치한 한국어 교육 및 한국 문화 보급을 위한 교육기관이다. 한국과 우즈베키스탄 간의 교육과 문화 교류를 촉진하기 위해 한국어 보급, 한국인 유학생 상담 및 지도, 외국인 유학생 유치 활동 지원, 한국어 강의 등을 운영하고 있다.

고성환 총장은 “이번 협약은 방송대가 국제적 교육 협력의 선두주자로 나아가는 중요한 발걸음이다. 특히, 중앙아시아 지역에서 유학생 유치를 통해 방송대의 글로벌 경쟁력을 강화할 것”이라고 말했다.

'''

okt = Okt()

# norm=True : 표준어로 변환, 잘못된 철자 교정 등 정규화 수행

# stem=True : 단어를 기본형으로 변환 (어간추출, stemming)

result = okt.pos(text, norm=True, stem=True)

print(result)

-----------------------

ㅇ testNounCount.py

-------------------

# 형태소 분석

# pip install konlpy

from konlpy.tag import Okt

from collections import Counter

text = '''국립 한국방송통신대학교(고성환 총장/이하 방송대)는 지난 25일 타슈켄트 한국교육원(이영웅 원장/이하 한국교육원)과 교육 협력 체제 구축 및 활성화를 위한 MOU를 체결했다고 28일 밝혔다.

타슈켄트 한국교육원에서 진행된 협약식에는 방송대 고성환 총장, 장영재 기획처장, 김재형 국제협력단장 등 관계자들이 참석해 우즈베키스탄 기묘국제대학교, 슈켄트 아주대학교를 비롯한 주요 교육 기관들과의 협력 확대도 논의했다.

2015년 11월 첫 협약을 맺은 양 기관은 한국어 교육 콘텐츠 공동 개발과 학술 연구 활동 지원 등 다양한 분야에서 긴밀한 협력을 해왔다. 이를 바탕으로 MOU수정 체결을 통해 한국학 콘텐츠 확대 보급을 위한 프로그램 개발 등 다양한 분야에서 협력을 강화해 학문적 교류를 확대하는 데 중점을 두고 있다.

특히, 중앙아시아 지역에서 외국인 유학생 유치 기반을 강화하고, 중앙아시아 대학들과 복수학위 프로그램 개설 가능성도 논의되어 향후 국제적 교육 교류가 더욱 확대될 전망이다.

타슈켄트 한국교육원은 우즈베키스탄의 수도 타슈켄트에 위치한 한국어 교육 및 한국 문화 보급을 위한 교육기관이다. 한국과 우즈베키스탄 간의 교육과 문화 교류를 촉진하기 위해 한국어 보급, 한국인 유학생 상담 및 지도, 외국인 유학생 유치 활동 지원, 한국어 강의 등을 운영하고 있다.

고성환 총장은 “이번 협약은 방송대가 국제적 교육 협력의 선두주자로 나아가는 중요한 발걸음이다. 특히, 중앙아시아 지역에서 유학생 유치를 통해 방송대의 글로벌 경쟁력을 강화할 것”이라고 말했다.

'''

okt = Okt()

# norm=True : 표준어로 변환, 잘못된 철자 교정 등 정규화 수행

# stem=True : 단어를 기본형으로 변환 (어간추출, stemming)

result = okt.pos(text, norm=True, stem=True)

print(result)

nouns = okt.nouns(text)

print(nouns)

print(len(nouns)) # 명사 개수

# 명사 빈도수 계산

noun\_counts = Counter(nouns)

# 상위 5개 명사 출력

top\_5\_nouns = noun\_counts.most\_common(5)

for noun, count in top\_5\_nouns:

print(f'{noun}: {count}')

--------------------

ㅇ testBasyes.py

----------------

# pip install scikit-learn

# 조건부 확률을 이용한 스팸(텍스트) 분류

from sklearn.naive\_bayes import BernoulliNB, GaussianNB, MultinomialNB

from sklearn.feature\_extraction.text import CountVectorizer

# 훈련 데이터 (텍스트)

corpus = ["광고 쇼핑몰 가격 선물 세일",

"투자 수익 소식 등급",

"쿠폰 선물 무료 익일 배송",

"백화점 상품 파격 세일",

"무료 긴급 당첨 투자 수익 대출",

"프로젝트 보고서 지급",

"회의 일정 변경"]

# 훈련라벨 ())0: 일반, 1: 스팸)

label = [1,1,1,1,1,0,0]

vector = CountVectorizer()

train = vector.fit\_transform(corpus).toarray()

# 나이브 베이즈 분류 모델 생성

model = BernoulliNB()

model.fit(train, label) # 학습

data = ["백화점 세일 무료 배송 알찬"]

test = vector.transform(data).toarray()

result = model.predict(test)

print(result)

---------------

ㅇ DNN을 이용한 스팸메일 분류

-----------

import numpy as np

from sklearn.feature\_extraction.text import CountVectorizer

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Dense, Dropout

from tensorflow.keras.optimizers import Adam

from tensorflow.keras.utils import to\_categorical

# 훈련 데이터 (텍스트)

corpus = ["광고 쇼핑몰 가격 선물 세일",

"투자 수익 소식 등급",

"쿠폰 선물 무료 익일 배송",

"백화점 상품 파격 세일",

"무료 긴급 당첨 투자 수익 대출",

"프로젝트 보고서 지급",

"회의 일정 변경"]

# 훈련라벨 (0: 일반, 1: 스팸)

labels = [1, 1, 1, 1, 1, 0, 0]

# 텍스트 데이터 벡터화

vectorizer = CountVectorizer()

X\_train = vectorizer.fit\_transform(corpus).toarray()

# 라벨 원-핫 인코딩 (DNN 입력용)

y\_train = to\_categorical(labels, num\_classes=2)

# 피처(단어) 개수 확인

feature\_count = X\_train.shape[1]

print(f"특성(단어) 개수: {feature\_count}")

# DNN 모델 생성

model = Sequential([

Dense(32, activation='relu', input\_shape=(feature\_count,)),

Dropout(0.2), # 과적합 방지

Dense(16, activation='relu'),

Dropout(0.2), # 과적합 방지

Dense(2, activation='softmax') # 이진 분류 (일반 vs 스팸)

])

# 모델 컴파일

model.compile(

optimizer=Adam(learning\_rate=0.001),

loss='categorical\_crossentropy',

metrics=['accuracy']

)

# 모델 요약

model.summary()

# 모델 학습

history = model.fit(

X\_train,

y\_train,

epochs=100, # 에포크 수 (작은 데이터셋이므로 많이 학습)

batch\_size=2, # 작은 데이터셋에 맞게 배치 크기 조정

verbose=1

)

# 테스트 데이터

test\_data = ["백화점 세일 무료 배송 알찬"]

X\_test = vectorizer.transform(test\_data).toarray()

# 예측

raw\_predictions = model.predict(X\_test)

predicted\_class = np.argmax(raw\_predictions, axis=1)

print("\n--- 예측 결과 ---")

print(f"입력 텍스트: {test\_data[0]}")

print(f"예측 확률: {raw\_predictions[0]}") # [일반 확률, 스팸 확률]

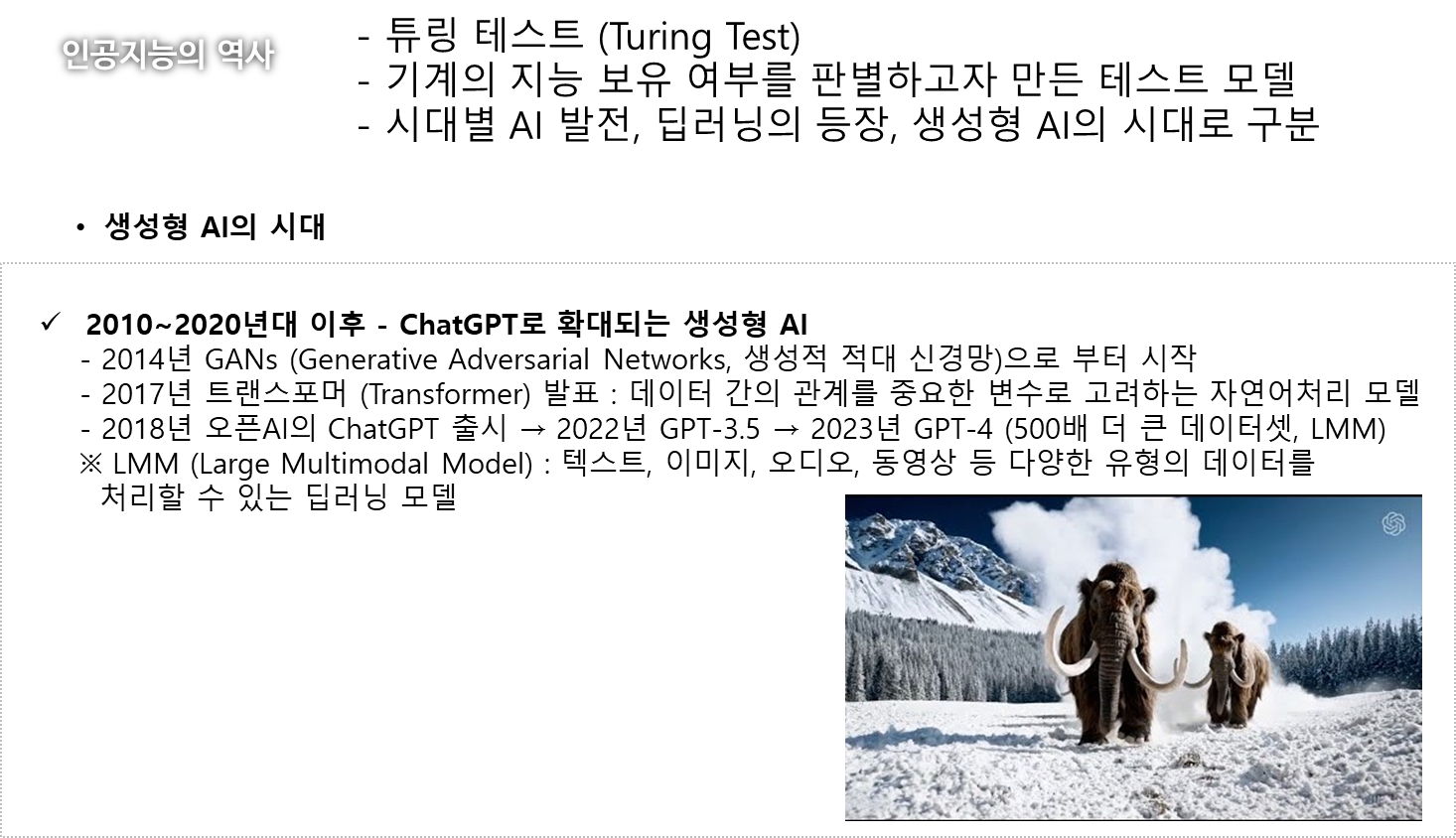
print(f"예측 클래스: {'스팸' if predicted\_class[0] == 1 else '일반'} ({predicted\_class[0]})")

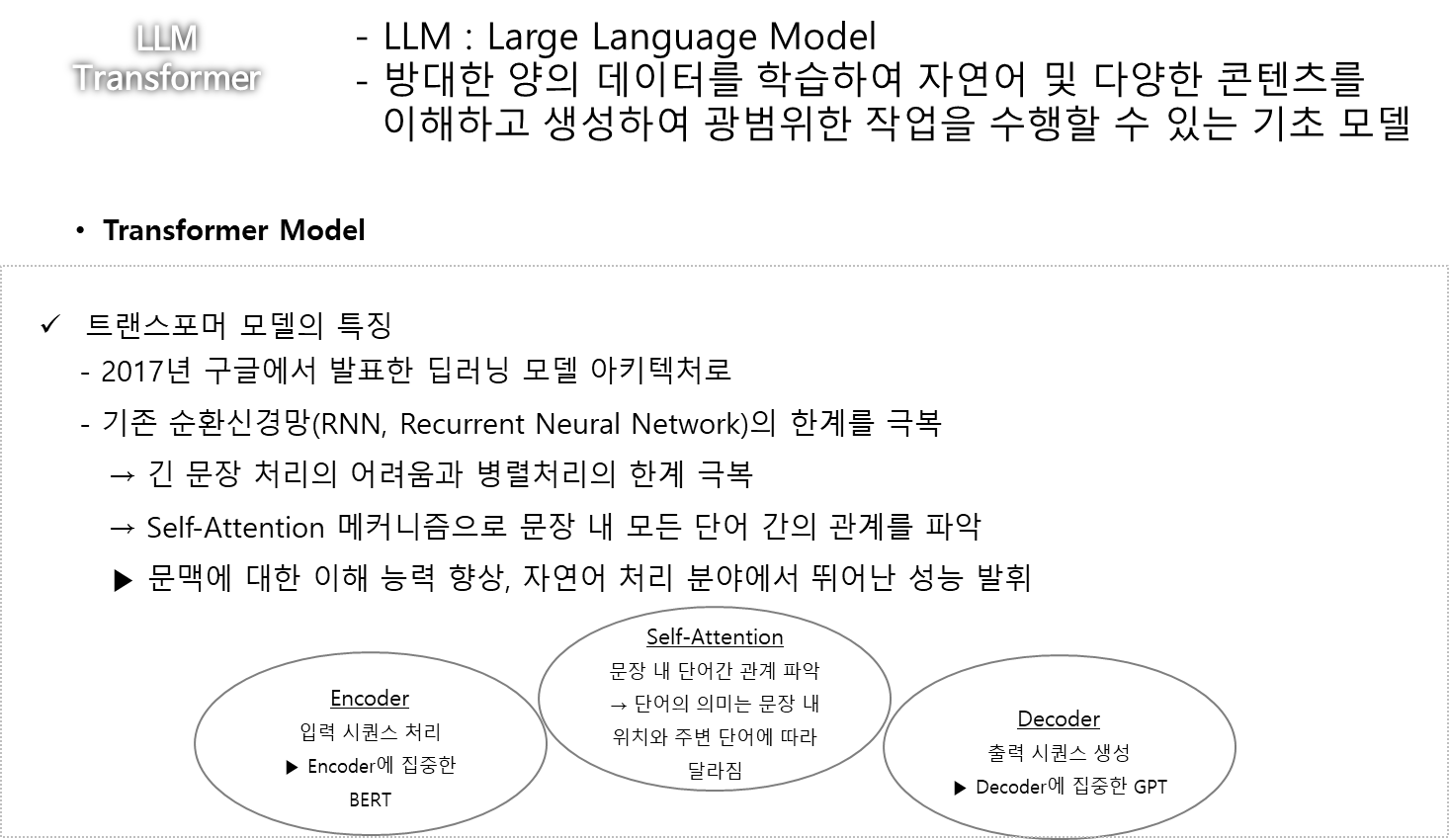
------------------------------

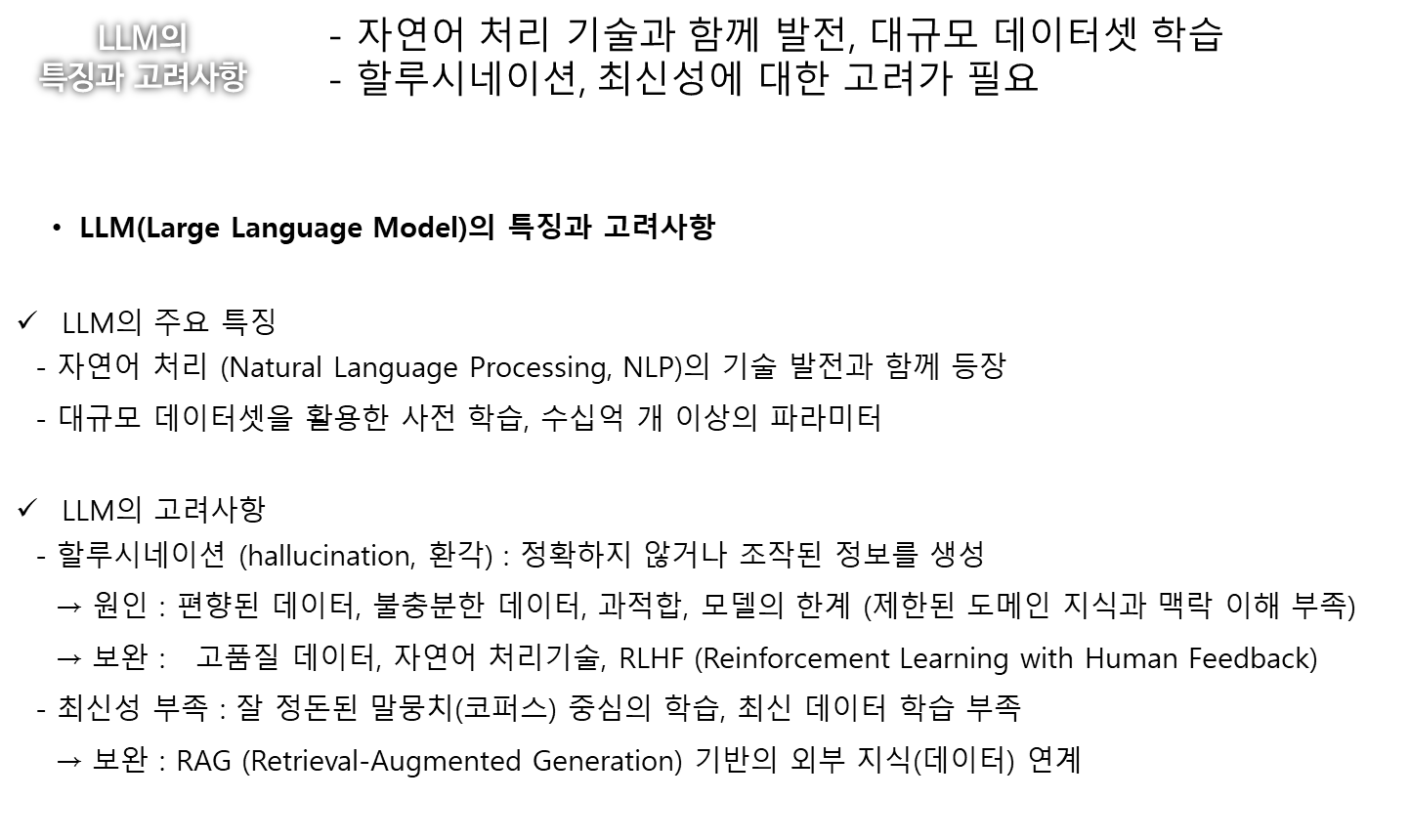
[교안 발췌]



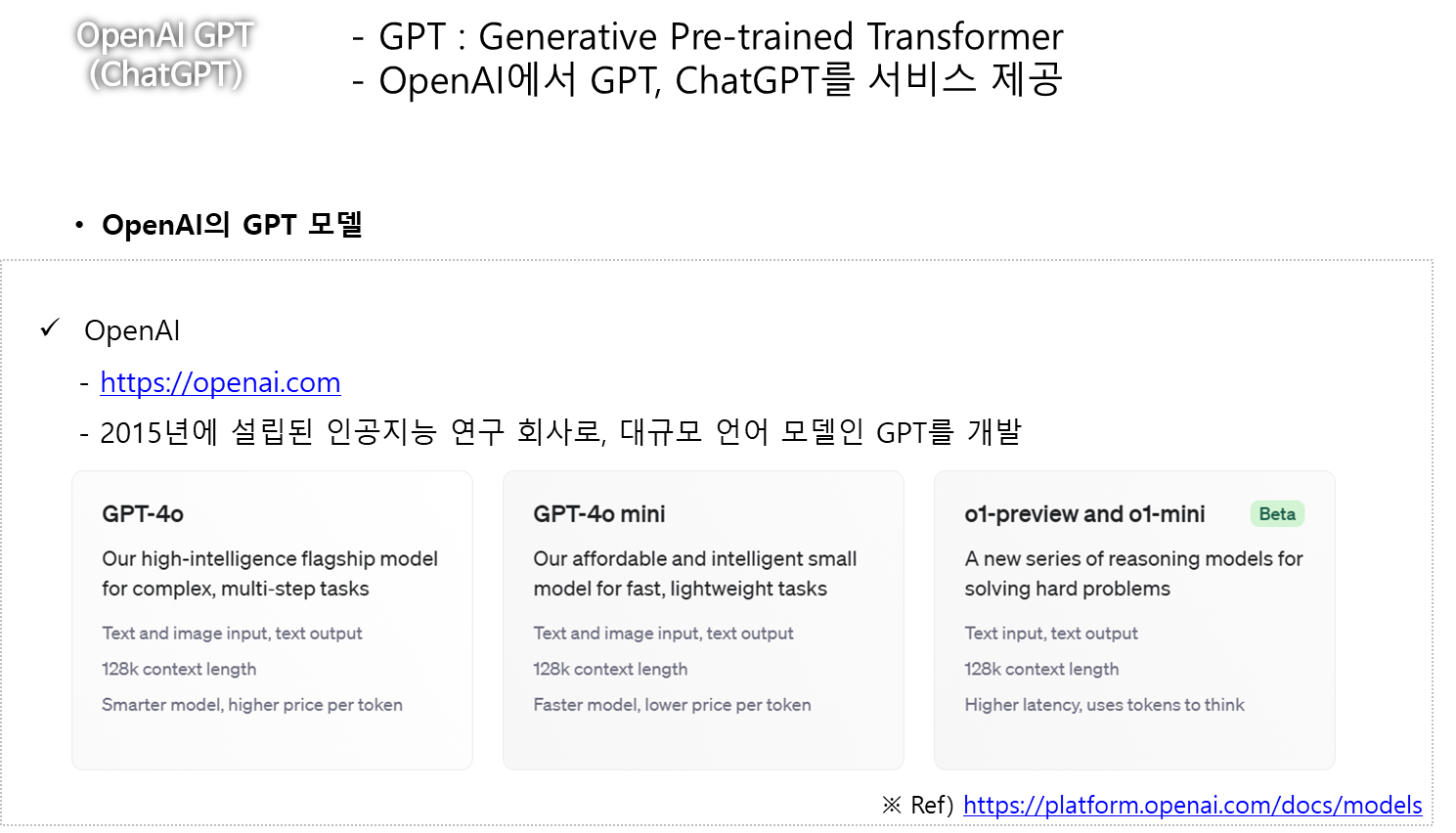


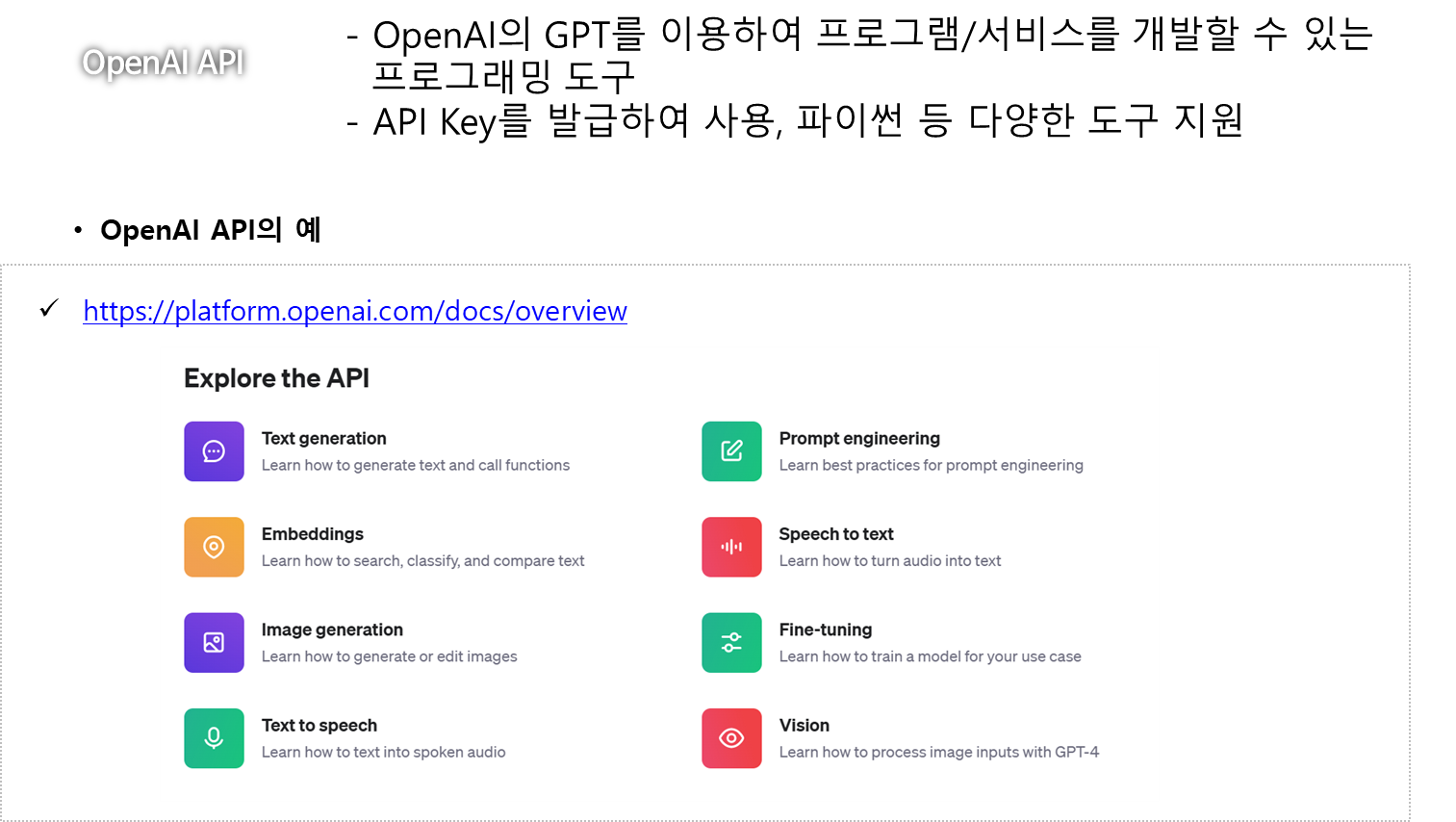


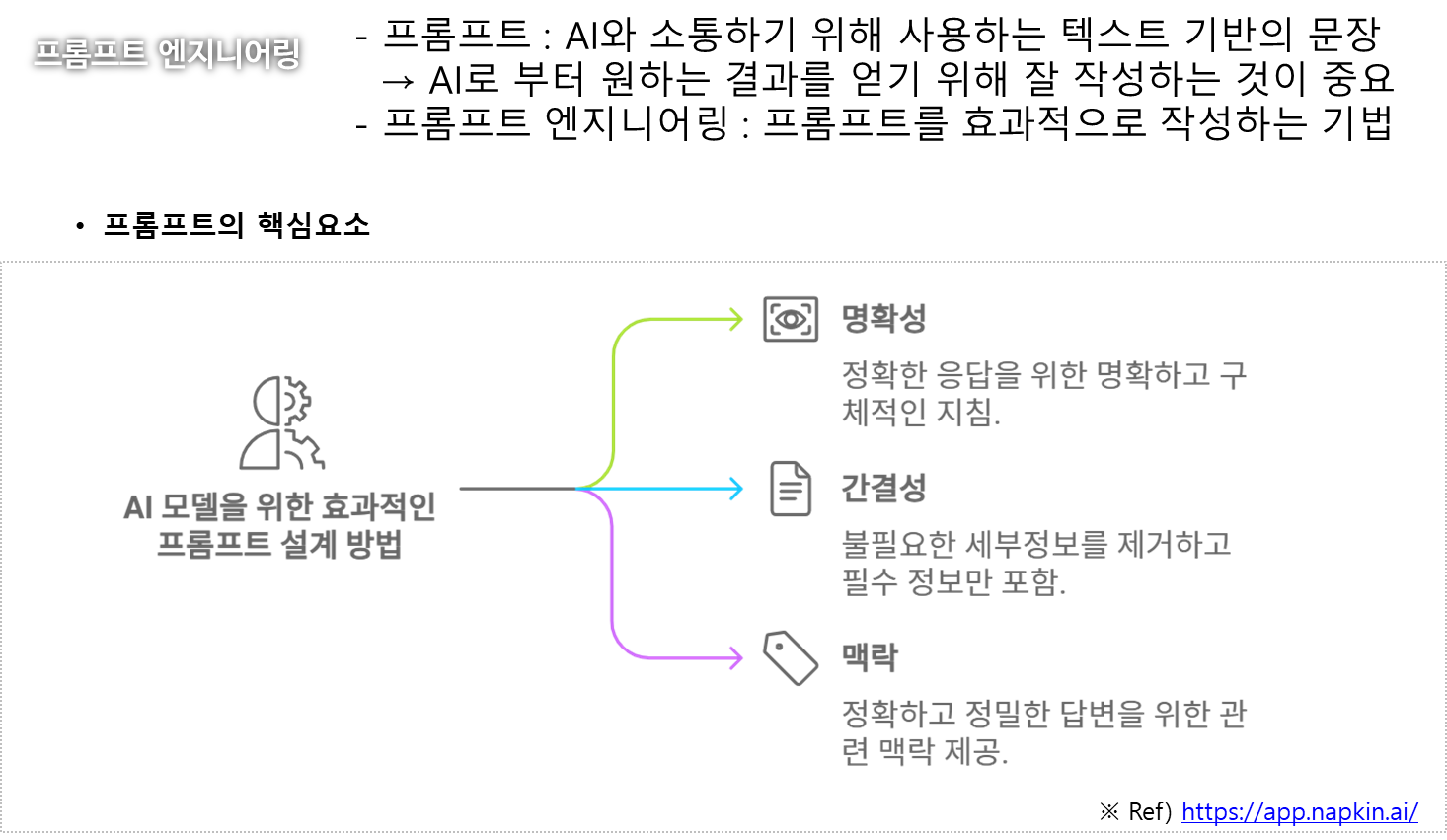












1) 자료를 검색/서베이

- 구글 : 이미지

- <https://www.perplexity.ai/>

2) 자료를 정리/요약 (컨텐트 생산)

질의/응답

- <https://claude.ai/>

3) 문서를 만들거나 컨텐트를 생산

ㅇ 프롬프트

- 대화다. (커뮤니케이션)

- 내가 알고 있는 정보, 전달하고 싶은 규칙이 있으면 구체적으로 가이드한다. (Few Shot L~~~)

- 내가 원하는 것을 명확히 하고, 결과물의 형식도 정확히 한다.

\* [KOSA20250317](https://drive.google.com/drive/folders/1NnE7z8bexu1Wft-3gFsRJ1n4oS7v6dQB)

=> <https://wrtn.ai/>

ㅇ ChatGPT는 검색엔지?

백과사전?

교과서

==> 추론엔진

===============

추강(秋江)에 밤이 드니 물결이 차노매라.

낚시 드리오니 고기 아니 무노매라.

무심(無心)한 달빛만 싣고 빈 배 저어 오노라.

===============

access.log (access.txt) : 아파치 웹서버 로그

=> 응답코드 404인 로그 파일에서 클라이언트 IP와 요청 URL을 추출해서

표시하는 것.

ㅇ 요구사항 : 잘 관리하고, 주의

- 기능적 : ID와 PW로 로그인한다.

- 품질적 제약사항 : 500명이 동시에 로그인해도 최대 1초 이내에는 응답

- 보안 : 연속적으로 5번 PW를 잘못 입력하면 LOCK을 건다.

- 관리적

ㅇ 요구사항을 확인/검증하는 기분

- 완전성 : 기능적, 품질적 제약, 보안.

- 구체성 : 구현 가능하고, 검증 가능

ㅇ 개발 방법론 (재사용)

- 구조적 : 함수 (기능)

- 객체지향 : Class (개체), 아키텍처 (프레임워크, 디자인 패턴)

- CBD

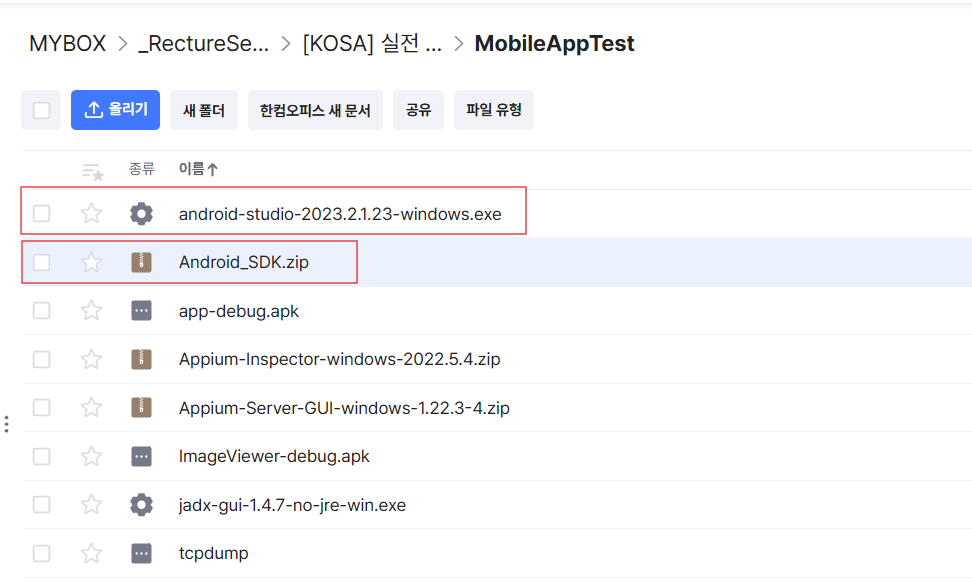
- Web Service (SOA)

- MSA (Micro Service Architecture)

------------------------------------------------------

[목요일]

\* [**https://naver.me/GdyzE46V**](https://naver.me/GdyzE46V)

****

ㅇ 일을 잘 해 보자 !!

ㅇ 성숙도 모델 (CMMi)

- 초기(Initial) : 그때 그때 다르다.

- 반복 : 해본 것은 그럭저럭 한다.

- 정의 : 전사 표준화

- 정량화 : 수치로 측정

- 최적화 : 스스로 성숙도를 높혀간다.

ㅇ 프로젝트를 어떻게 수행해야 성공

\* 범위 : 우리가 해야 할일이 어디~어디까지인가?

**- WBS (Work Breakdown System) : 과업을 top-down 방식으로 구조화하여 관리하는 체계**

**=> Work Package : 단위의 일을 모두 완료하면 전체가 완료.**

**1) 예상 일정**

**2) 담당자**

**3) 산출물 (이 일이 완료되었음을 객관적으로 확인할 수 있는 대상)**

**+**

**세부 절차.**

야유회

- 장소섭외

- 설문조사

- 현장조사

- 투표/선정

=> 2.5일

=> 크리스

=> 예약하고 선입금 영수증

- 식당섭외

- 참석대상

- 차편결정

- 행사내용

- 고객사에 안내

\* 규모 -> 비용

\* 일정 관리 : Critical Path

\* 의사소통 : 보고

1) 얼마나 자주

2) 어떤 수준으로

3) 누구한테

4) 어떠한 방법으로

\* 위험 : 이슈가 되지 않도록 잘 관리하는 것이 핵심

위험관리 절차 (프레임워크)

1) 위험 식별

- 체크리스트 : 과거의 경험을 이용

- 시나리오 기법 : 알려지지 않은 문제점(위험)을 발견

- 전문가 감정 : 경험이 있는, 해당 분야의 전문가에게 자문.

2) 위험 분석 / 명세화

- 위험을 정량화(수치), 비교(우선순위)

- 위험도 = 발생가능성 x 피해정도

3) 위험 대응 계획 (예방 활동은 선행)

- 회피

- 완화

- 수용

- 전가

4) 모니터링 / 통제

<윈도우 운영체제 및 응용 프로그램의 특징>

ㅇ 윈도우, 메시지 => 메시지 드리븐 (Message Driven)

- 모든 윈도우 (GUI) 응용 프로그램은

윈도우들의 모음으로 구성되고

이 개별 윈도우는 윈도우즈 운영체제가 관리하며

개별 윈도우에 메시지를 보내면 반응한다.

ㅇ PNP, 메시지 훅

=> 키보드를 누르면 왜 텍스트가 입력되는가?

- 디바이스 드라이버 => 기능, 필터 종류로 나뉘고 계층구조로 동작.

ㅇ NTFS는 ADS를 지원

- ADS : Alternative Data Stream

type c:\Windows\System32\calc.exe > test.txt:calc.exe

wmic process call create "c:\\_Temp\test.txt:calc.exe"

dir /R

1) Android Studio 설치 => 실행

2) Android\_SDK : 압축 해제

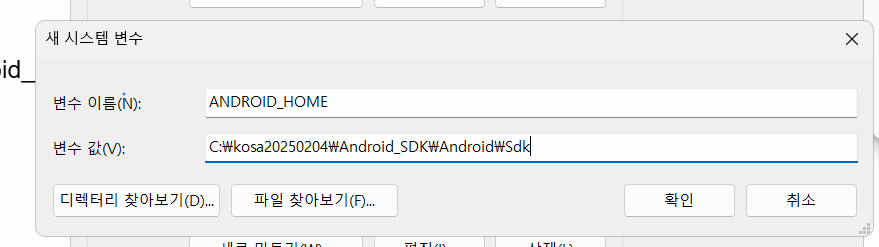
--------------------------------------------------------------------------------------

(1) SDK 환경 설정. (6.0)

- 설치경로 : C:\kosa20250317\Android\_SDK\Android\Sdk

환경 변수

ANDROID\_HOME=C:\kosa20250317\Android\_SDK\Android\Sdk



PATH에 등록

%ANDROID\_HOME%\platform-tools

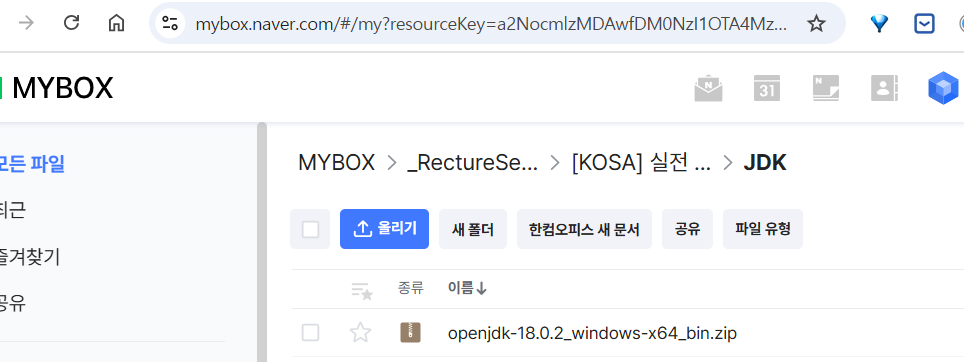
%ANDROID\_HOME%\tools

%ANDROID\_HOME%\tools\bin

%ANDROID\_HOME%\emulator

  
\* adb --version

ㅇ JDK 설치



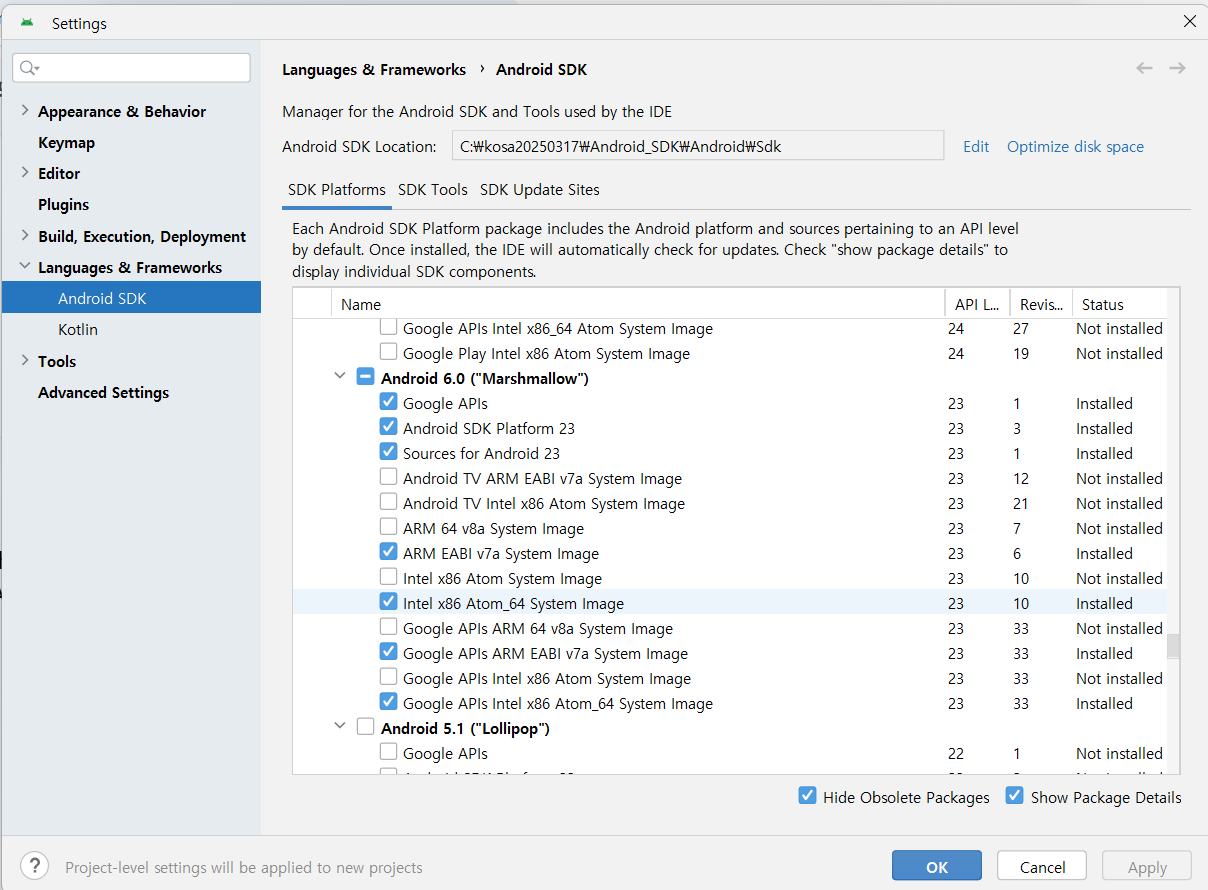
1) 압축 푼다

2) 압축 푼 경로를 적절한 곳으로 이동(복사) : c:\kosa20250317\

3) JAVA\_HOME을 설정한다.

4) %JAVA\_HOME%\bin 을 PATH에 등록한다.

ㅇ Android Studio를 실행하고 SDK 경로를 설정



<https://play.google.com/>

<https://apkpure.com/kr/>

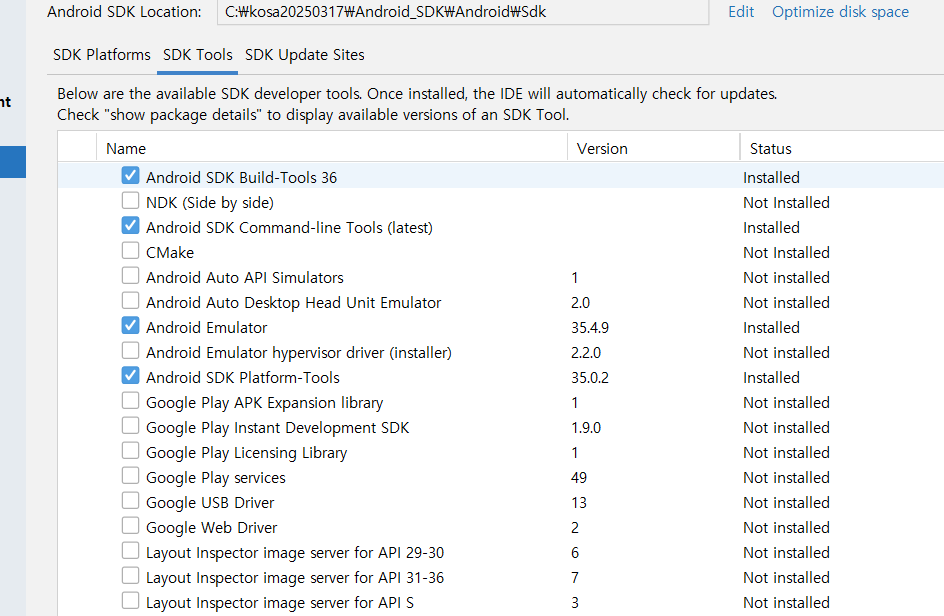
--------------------------

~~5시 50분

-------------------------

----------------------------------------------------------------------------------------------------------------------------

1) Android Emulator 가 동작.



==> Android Studio 를 최신버전으로 업데이트 ( <https://developer.android.com/studio?hl=ko> )

adb --version

2) JAVA\_HOME 환경 변수와 %JAVA\_HOME%\bin JAVA 설치

java --version

------------------------------------------------------------------------------------------------------------------------------

\* WBS, 위험관리 ===> PMP

-------------------------------------------------------------------------------

[마지막날]

\* 오후 1시~2시 : 한주간 학습내용 리뷰(복습)

\* 오후 2시~3시40분 : (100간) 평가

\* 3시50분~4시 : 안내, 설문관련해서 안내.

~~ 5시 50분 : 오후 과정.

--------------------------------------------------------------------------------

ㅇ 준비

-> Android Studio => Virtual Device Manager => 에뮬레이터 실행.

-> CMD 창 띄우시고 => adb --version

---------------------------------------------------------------------------------

[모바일 보안 테스트 - 계획서/결과보고서](https://docs.google.com/document/d/1CCsKWgtFB872ZjCxosn3Mu50r4gjOPZ6lND6QksyBEM/edit?tab=t.0#heading=h.txgcz2odqtfo)

<https://drive.google.com/drive/u/0/folders/1whovTmSAAcZY89lpnHKvbV7DiWpDULMP>

[**https://naver.me/GdyzE46V**](https://naver.me/GdyzE46V)