

# AI Newsletter (Pilot)

연구관리처 AI연구센터 / (2022년 10월 25일)

K-water연구원 연구관리처 AI연구센터에서 정기적으로 발간하는 뉴스레터입니다.

AI 뉴스, Hands-on 프로젝트, 팁 등을 다양한 내용과 난이도로 담았습니다.

※ 코드 및 뉴스 등 외부링크 등이 다수 포함되어 있으므로 인터넷 환경 PC 권장

>> Hello, world!

#AI연구센터

#Hello, K-water

#직접 돌려보기

#AI 1학년 1반

- 안녕하세요. K-water연구원 AI연구센터입니다.
- 모두가 인식하고 있듯, 최근 AI 기술은 매우 빠른 속도로 발전하고 있으며, 모든 산업군에 빠짐없이 다양한 형태, 목적으로 적용되고 있습니다.
- 그럼에도 불구하고, AI 기술이 중요하다고는 하지만 너무 많은 정보들이 쏟아지고 있어 무슨 정보를 어디서 어떻게 습득해야 할지, 어떤 분야와 업무에 어떤 알고리즘이 적용되고 있는지 파악하기가 쉽지는 않습니다.
- 우리 K-water에서도 현업과 연구/기술 부서를 가리지 않고 많은 적용과 시도가 이루어지고 있는 이 시기에, 최신의 AI관련 S/W, H/W 기법과 동향을 공유함은 물론, 다양한 응용이 가능한 기본기술, 초중급자를 위한 코딩 Tip 등을 담아 공유하고자 합니다.

모든 실습 예제와 링크는 “AI연구센터” 누리집(Homepage, GitHub)에 게시되어 활용가능합니다.

## # CONTENTS

### ## 1. What's New in AI

### AI 분야 전반에 걸쳐 최근 어떤 일들이 일어나고 있는지 다양하게 담아보았습니다.

### ## 2. Hands-on AI Project

### AI 프로젝트의 End-to-end를 설명해 드립니다. 전체 코드는 colab 링크를 참고해주세요.

### 초급자를 위한 1건, 중-고급자를 위한 1건을 준비하고 있습니다.

### ## 3. TIPS

### AI의 기초적인 토막상식, 생산성을 올려주는 library, method, 코드 tip등을 소개합니다.

### ## 4. Meanwhile, in K-water AI Lab.

### K-water의 AI연구센터에서 AI연구 및 개발을 위해 무엇을 하고있는지 소개합니다.

> print(f“{YOUR\_NAME}, Please Have FUN\n :D”);

# 1. What's NEW in AI

#TTI

#DALL-E




#MULAN

#NVIDIA

#DeepNote

## □ (기술) **“말만 하면 그려드립니다”** Text-to-Image 기술

- SNS에서 ‘AI가 그린 해리포터’ 게시물을 본 적 있으신가요? 이 기술이 바로 TTI입니다. TTI는 글(텍스트)을 입력하면 그림으로 바꿔 줍니다. 관련 서비스로는 OpenAI의 DALL-E, Leap Motion의 MidJourney, Stability.AI의 Stable Diffusion 등이 있습니다.

	DALL-E2	MidJourney	Stable Diffusion
Prompt	말을 타고 있는 우주비행사를 사진같은 느낌이 나게 그려줘	a warrior robot astronaut, floral, posing for a fight intricate, elegant, highly detailed, digital painting, artstation, concept art, smooth, sharp focus, illustration, art by artgerm and gerg rutkowski and alponse mucha, 8K	
결과			
특징	<ul style="list-style-type: none"><li>- 해상도 높음</li><li>- 결과물 편집 가능</li><li>- 편향적인 결과 수정됨</li></ul>	<ul style="list-style-type: none"><li>- 디테일을 잘 살림</li><li>- 경험자에게 추천</li><li>- 오픈베타 중</li></ul>	<ul style="list-style-type: none"><li>- 보다 자연스러움</li><li>- 초보에게 추천</li><li>- 무료, 빠름</li></ul>

※Links: <https://openai.com/dall-e-2/>

<https://medium.com/codex/midjourney-vs-stable-diffusion-same-prompt-different-result-dd29ca4822>

35

## □ (논문) **“음악을 학습합니다”** MULAN: 오디오와 자연어의 Joint 임베딩 모델

- 오디오(음악)와 자연어를 연결하는 새로운 임베딩 모델에 대한 논문이 공개되었습니다.
- 일반적으로 음악에 태그를 걸고, 검색하고, 추천해주는 시스템은 음원에 대해 사전 정의된 온톨로지를 활용하여 구성되어 있습니다.
- 이 논문에서는 약 4,400만개의 음악(37만 시간)과, 각 음원에 연결된 주석을 활용하여 텍스트 임베딩 구조로 딥러닝 모델을 훈련하였습니다.
- 해당 모델을 통해 전이 학습, 제로샷 태깅, 음원 도메인에 대한 언어적 이해, 교차 검색 등의 다양한 어플리케이션에 대한 적용을 기대해봅니다.

※Links: <https://arxiv.org/pdf/2208.12415.pdf>

## □ (뉴스) RTX 4000년대 제품 발표 (주경원)

- GPU 시장의 강자 NVIDIA에서 지난 9월 21일 RTX 4090, RTX 4080을 발표했습니다. 코드네임은 'Ada Lovelace'이며 기존 RTX 4090은 이전 세대 RTX 3090 Ti보다 1.5배 높은 성능을 나타낸다고 합니다.
- 하지만 환율 및 반도체 웨이퍼 생산 단가 상승으로 4090은 263만원, 4080은 192만원(16GB), 140만원(12GB)\*부터 시작합니다. 지난 세대인 3000년대에 비하면 가격 및 전력소모가 50%가까이 올랐습니다.

\*최근 NVIDIA는 4080 12GB 발매를 취소했습니다. 사실상 4070 급의 성능이라는 여론을 인식한 듯 합니다.

- 딥러닝 연산에 활용할 수 있는 CUDA 코어 수는 아래와 같습니다.

(딥러닝 전용 GPU를 활용하는게 유리할 수도 있지만, 게이밍 성능을 함께 원하면 RTX도 좋은 선택지입니다.)

Card	4090 24GB	4080 16GB	4080 12GB	3090 24GB	3080 10GB
CUDA Cores	16384	9728	7680	10496	8704

※Links: <https://nvidia.com/ko-kr/geforce/graphics-cards/40-series/>

## □ (소개) DeepNote: Python 노트북 클라우드 플랫폼

- 깃허브는 어렵고 코드 공유는 하고 싶을때?
  - 딥노트가 하나의 방법이 될 수 있습니다. 딥노트는 주피터 노트북 형태로 되어 있고 구글 문서처럼 내용 공유와 코멘트 기능이 있습니다. 또 노트 형태가 노션(Notion)과 유사하여 접근성이 좋습니다.
- 더불어 깃허브(GitHub), 노션, 슬랙(Slack), MySQL 등과 연결되어 있습니다.
- 하지만 활용하는 클라우드 컴퓨팅 리소스에 따라 요금이 세부적으로 책정되어 있어 무거운 작업이 필요할 때는 적합하지 않을 수도 있습니다.

The screenshot shows a web browser window with the URL <https://deepnote.com/workspace/newsletter-6539f630-36df-4091-a9b5-30b5941bb926/project/Daily-Stock-Price-Dashboard-fd85d6ea-8337-43f8-a9d4-7f853bc42c2c/notebook/notebook-72428635...>. The notebook is titled 'Daily Stock Price Dashboard' and contains a 'Time Series' section. The code cell shows the following imports: `from datetime import datetime, import pandas as pd, import plotly.graph_objects as go, import yfinance as yf`. The plot shows a line graph of Microsoft (MSFT) stock prices over time. The interface includes a sidebar with navigation options like Projects, Integrations, and Settings, and a right sidebar with Project, Integrations, Comments, and History tabs.

※Links: <https://deepnote.com/>

## 2. "10줄 코딩" Hands-on AI Project

#수질

#기사

#수치

#자연어

#BERT

#분류

### □ 먹을 수 있는 물인지 판단하기 (수질측정 자료) (초급, 분류, 수치)

※Link: [https://colab.research.google.com/drive/1p5cki\\_3KWsj1Ff52okNKWnb89Fgwt\\_pq?usp=sharing](https://colab.research.google.com/drive/1p5cki_3KWsj1Ff52okNKWnb89Fgwt_pq?usp=sharing)

○ Colab(무료버전) 개발환경에서 작업합니다. (하드웨어 가속 필요 없음)

○ Kaggle에서 제공하는 수질에 따른 음용가능 여부 데이터셋을 받습니다.

```
!pip install -q kaggle
# https://www.kaggle.com/datasets/adityakadiwal/water-potability
!kaggle datasets download -d adityakadiwal/water-potability
```

○ 데이터셋을 불러와서 어떤 항목을 가지고 있는지 확인해봅니다. 데이터 출처에서 각 변수의 뜻과 단위를 확인합니다.

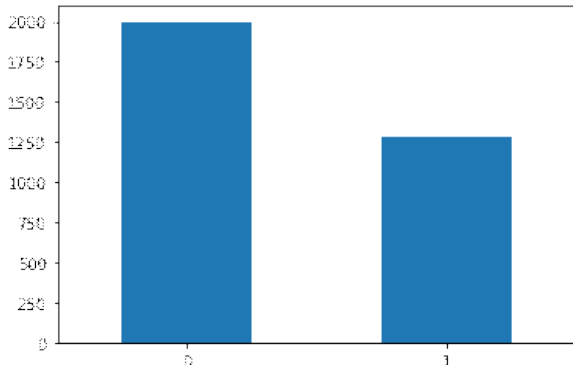
```
data = pd.read_csv('water_potability.csv')
data.info()
```

```
RangeIndex: 3276 entries, 0 to 3275
Data columns (total 10 columns):
#   Column                Non-Null Count  Dtype
---  -
0   ph                    2785 non-null   float64
1   Hardness              3276 non-null   float64
2   Solids               3276 non-null   float64
3   Chloramines          3276 non-null   float64
4   Sulfate              2495 non-null   float64
5   Conductivity         3276 non-null   float64
6   Organic_carbon       3276 non-null   float64
7   Trihalomethanes      3114 non-null   float64
8   Turbidity            3276 non-null   float64
9   Potability            3276 non-null   int64
```

0) pH: pH (0~14)  
1) Hardness: 경도, 물 100cc당 산화칼슘 함유량  
2) Solids: 용존 고형물(ppm)  
3) Chloramines: 클로라민(ppm)  
4) Sulfate: 황산염(mg/L)  
5) Conductivity: 전도도(μg/L)  
6) Organic\_carbon: 유기물(ppm)  
7) Trihalomethanes: 트리할로메탄(μg/L)  
8) Turbidity: 탁도(NTU)  
9) Potability: 음용가능여부 (0-불가능, 1-가능)

○ 수질 측정 자료를 바탕으로 음용가능 여부를 분류할 수 있는 모델을 만들면 좋을 것 같습니다. 이진분류(binary classification)에 해당하겠네요.

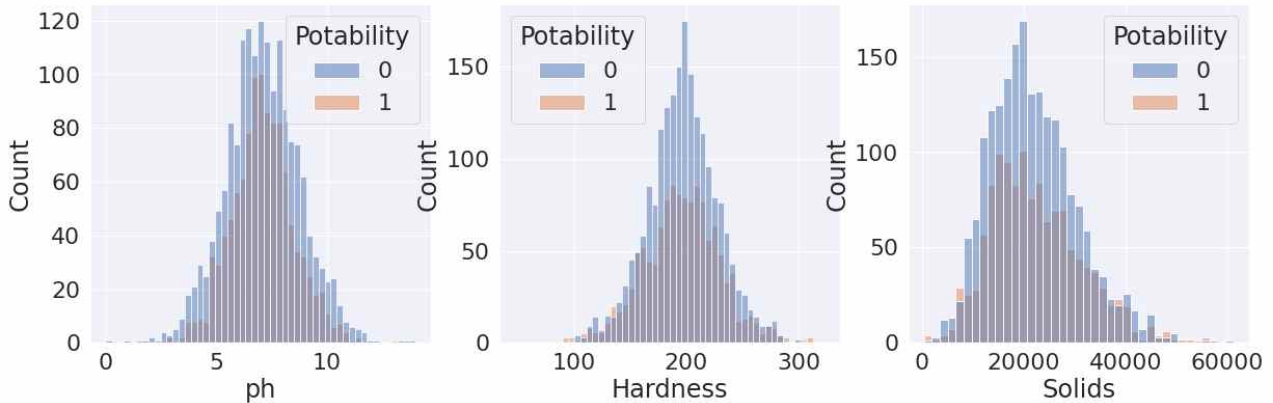
```
data['Potability'].value_counts().plot.bar(rot=0)
```



○ 0은 음용불가능, 1은 음용가능  
○ 음용불가 데이터가 2,000여개, 가능한 데이터가 1,300개 정도네요  
○ 6:4정도 비율로 조금은 불균형이 있지만 크게 무리는 없습니다.

- 수질자료가 음용불가능할 때, 가능할 때 어떻게 다른지 살펴보고 싶네요.  
9개씩 변수가 있으므로 3개씩 살펴봅니다. (전체는 link 참고)

```
fig, axes = plt.subplots(1,3, figsize=(20,6))
for idx, col in enumerate(list(data.columns[:3])):
    sns.histplot(data=data, x=col, hue='Potability', ax=axes[idx])
```



- 다음으로는 결측치를 확인합니다. 결측치는 중위값으로 채웠습니다.

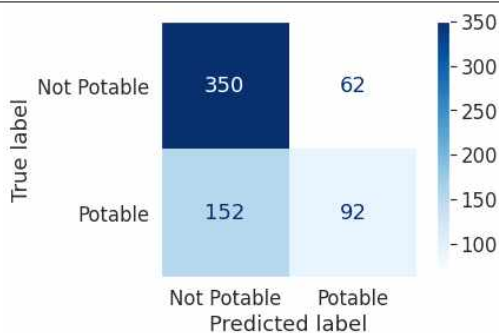
```
data.isnull().sum()
data.isnull().any(axis=1).sum()
data['pH'].fillna(value=data['pH'].median(), inplace=True)
```

- 모델링은 Random Forest(RF)를 활용했습니다. RF는 일반적으로 사용하기 편리하며, 별다른 튜닝이 없어도 보통 우수한 성능을 나타냅니다.

```
# Train/Test Split
X_train, X_test, y_train, y_test = train_test_split(data[cols_X], data['Potability'])
# Modeling
rf_clf = RandomForestClassifier(random_state=42)
rf_clf.fit(X_train, y_train)
rf_clf.score(X_test, y_test)
> 0.6738
```

- 전체적으로는 67% 정도의 정확도를 보여줍니다. 아래는 Test 셋의 분류결과 행렬입니다.

```
plot = plot_confusion_matrix(rf_clf, X_test, y_test)
```



- 정답(좌상단, 우하단) 442개
- 오답(우상단, 좌하단) 214개
- 정확도 =  $442 / (442 + 214) = 67.4\%$
- 왼쪽의 행렬을 조금 더 이해하기 위해서는 Precision과 Recall, F1-score에 대한 개념을 이해하면 좀 더 도움이 됩니다.

- 먹을 수 없는 물을 분류해 내는 성능(재현율)은 85%(350/412)이며, 먹을 수 없다고 판단한 물 중 진짜 먹을 수 없는 물(정밀도)은 70%(350/502)입니다

## □ 가짜뉴스 판단하기 (BERT 모델 활용) (중급, 분류, 자연어)

※Link: [https://colab.research.google.com/drive/19U5YHHWn-IQaA\\_XtF8tTiPC7UnErATLP?usp=sharing](https://colab.research.google.com/drive/19U5YHHWn-IQaA_XtF8tTiPC7UnErATLP?usp=sharing)

○ Colab(무료버전) GPU Runtime 개발환경에서 작업합니다.

○ github에 공개되어 있는 가짜/진짜 뉴스 데이터를 활용합니다.

```
!wget https://github.com/muttinenisairohith/FakeNewsDetection/raw/main/Data/Fake.csv.zip
!unzip /content/Fake.csv.zip
```

○ 이번 프로젝트에서 핵심이 되는 라이브러리는 keras, transformers입니다.

```
from keras models import Model, Sequential
from keras layers import Input, Dense, Dropout, Embedding
from transformers import AutoTokenizer, TFBertModel
```

○ 데이터셋을 불러와서 어떻게 생겼는지 확인해보겠습니다.

```
df_fake = pd.read_csv("Fake.csv")
df_true = pd.read_csv("True.csv")
df_fake['text'].iloc[2]
df_true['text'].iloc[2]
```

Fake News	True News
On Friday, it was revealed that former Milwaukee Sheriff David Clarke, who was being considered for Homeland Security Secretary in Donald Trump's administration, has an email scandal of his own. In January, there was a brief run-in on a plane between Clarke and fellow passenger Dan Black, who he later had detained by the police for no reason whatsoever, except that maybe his feelings were hurt. (...) #MAGA pic.twitter.com/XtZW5PdU2b David A. Clarke, Jr. (@SheriffClarke) (...) (@SheriffClarke) (...) https://t.co/zcbyc4Wp5b KeithLeBlanc (@KeithLeBlanc63) (...) Kirk Ketchum (@kirkketchum) (...)	WASHINGTON (Reuters) - The special counsel investigation of links between Russia and President Trump's 2016 election campaign should continue without interference in 2018, despite calls from some Trump administration allies and Republican lawmakers to shut it down, a prominent Republican senator said on Sunday. (...) Trump and his allies deny any collusion with Russia during the campaign, and the Kremlin has denied meddling in the election. (...) "As a matter of fact, it would hurt us if we ignored it," he said.
전(前) 밀워키 보안관이자 도널드 트럼프 정부의 국토안보부장관으로 거론되고 있는 데이비드 클라크(David Clarke)가 금요일, 자신과 관련된 스캔들을 이메일로 가지고 있다는 것이 밝혀졌습니다. 지난 1월, 비행기에서 클라크와 그의 동료 댄 블랙 사이에 작은 언쟁이 있었습니다. 그 후 댄 블랙은 단지(클라크의) 감정이 상했을지도 모른다는 이유만으로 경찰에 구금되었습니다. (...) #MAGA pic.twitter.com/XtZW5PdU2b David A. Clarke, Jr. (@SheriffClarke) (...) (@SheriffClarke) (...) https://t.co/zcbyc4Wp5b KeithLeBlanc (@KeithLeBlanc63) (...) Kirk Ketchum (@kirkketchum) (...)	워싱턴(로이터) - 일부 트럼프 행정부 지지자들과 공화당 의원들의 중단 요구에도 불구하고, 러시아와 트럼프 대통령의 2016년 선거운동 관련 특검 수사는 2018년에도 간섭 없이 계속돼야 한다고 공화당 상원의원이 일요일에 말했습니다. (...) 트럼프 대통령과 관계자들은 선거 기간 동안 러시아와의 유착을 부인하고 있으며, 크렘린궁(Kremlin, 러시아) 또한 선거 개입을 부인하고 있습니다. (...) 혹자는 "만약 우리가 (이 사건을) 무시한다면 이것은 우리에게 상처가 될 것입니다," 라고 말했습니다.

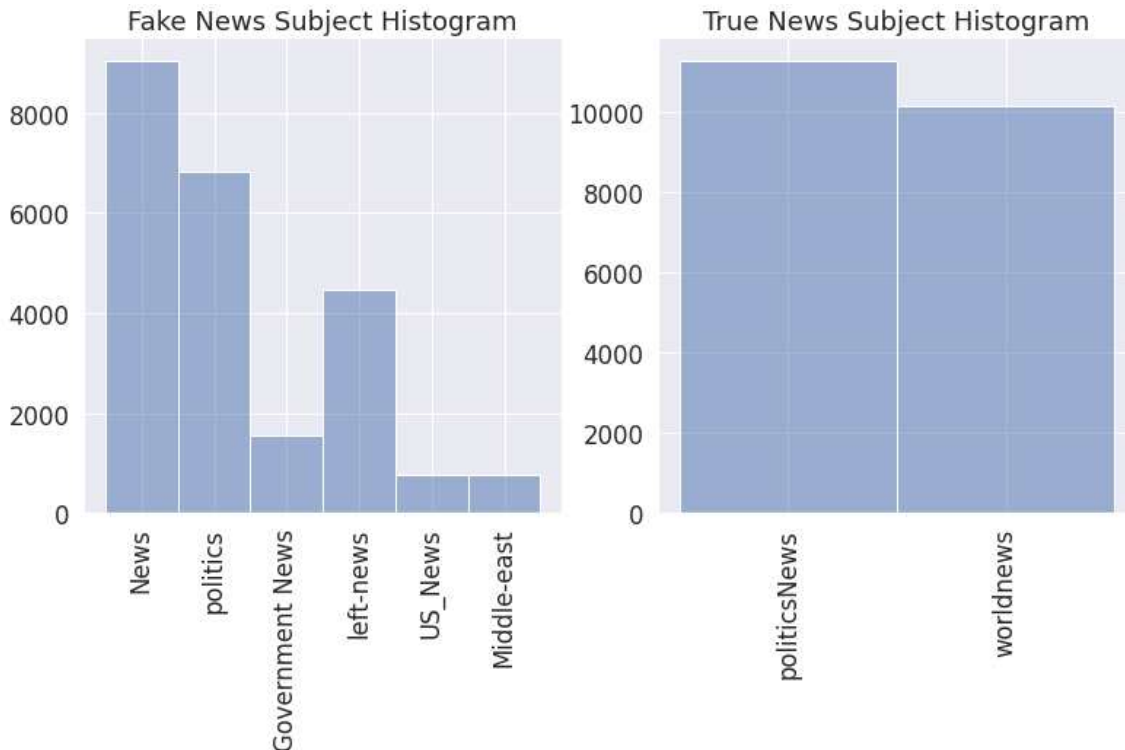
○ 가짜/진짜 열을 추가하고, 데이터프레임을 합치고, 섞어줍니다.

```
df_fake['Label'] = 'Fake'
df_true['Label'] = 'True'
df = pd.concat([df_fake, df_true])
df = df.sample(frac=1).reset_index(drop=True) # 이렇게 하지 않고도 섞는 방법은 많습니다
```



○ 'subject' 라는 변수가 있어서 어떻게 생겼는지 살펴봅니다.

```
sns.histplot(df_fake['subject'], alpha=0.5, ax=axes[0])
sns.histplot(df_true['subject'], alpha=0.5, ax=axes[1])
```



- 가짜뉴스랑 진짜뉴스는 카테고리가 달라 사용하기가 어렵겠네요. 보통 이런 카테고리 형태의 데이터는 one-hot-encoding을 활용하여 추가적으로 변수로 사용할 수 있지만 이 데이터에는 사용이 어렵겠습니다.

○ 데이터를 이제 훈련/테스트 자료로 분리하고 tokenize해줍니다.

```
X_train, X_test, y_train, y_test = train_test_split(df['text'], df['Label'], test_size=0.2)
X_train_tokens = tokenize(X_train)
X_test_tokens = tokenize(X_test)
```

○ 모델의 구조 및 학습방법, 결과는 아래와 같습니다.

<b>모델</b>	<ul style="list-style-type: none"> <li>○ Dropout rate: 0.2</li> <li>○ 레이어: 임베딩벡터 &gt; Dense(64) &gt; Dense(1)</li> <li>○ Optimizer: Adam</li> <li>○ Loss: 'binary_crossentropy'</li> <li>○ Metric: 정확도 (accuracy)</li> </ul>	<pre> graph TD     input_ids[input_ids] --&gt; InputLayer1[InputLayer]     input_mask[input_mask] --&gt; InputLayer2[InputLayer]     InputLayer1 --&gt; tf_bert_model[tf_bert_model]     InputLayer2 --&gt; TFBertModel[TFBertModel]     tf_bert_model --&gt; dropout_49[dropout_49]     TFBertModel --&gt; dropout_49     dropout_49 --&gt; dense_12[dense_12]     dense_12 --&gt; dropout_50[dropout_50]     dropout_50 --&gt; dense_13[dense_13]     dense_13 --&gt; output[output]     </pre>
<b>학습</b>	<ul style="list-style-type: none"> <li>○ epochs: 3</li> <li>○ validation split: 0.2</li> <li>○ batch_size: 32</li> <li>○ callbacks: early stopping</li> </ul>	
<b>결과</b>	<p style="text-align: center;"><b>100.0%</b></p> <p>*이정도 성능이라면, 진짜 뉴스와 가짜 뉴스를 구분하기 쉬웠나 봅니다. 다음 호에는 다시 데이터로 돌아가서 NLP를 활용하여 뉴스별 특징을 확인해보겠습니다.</p>	

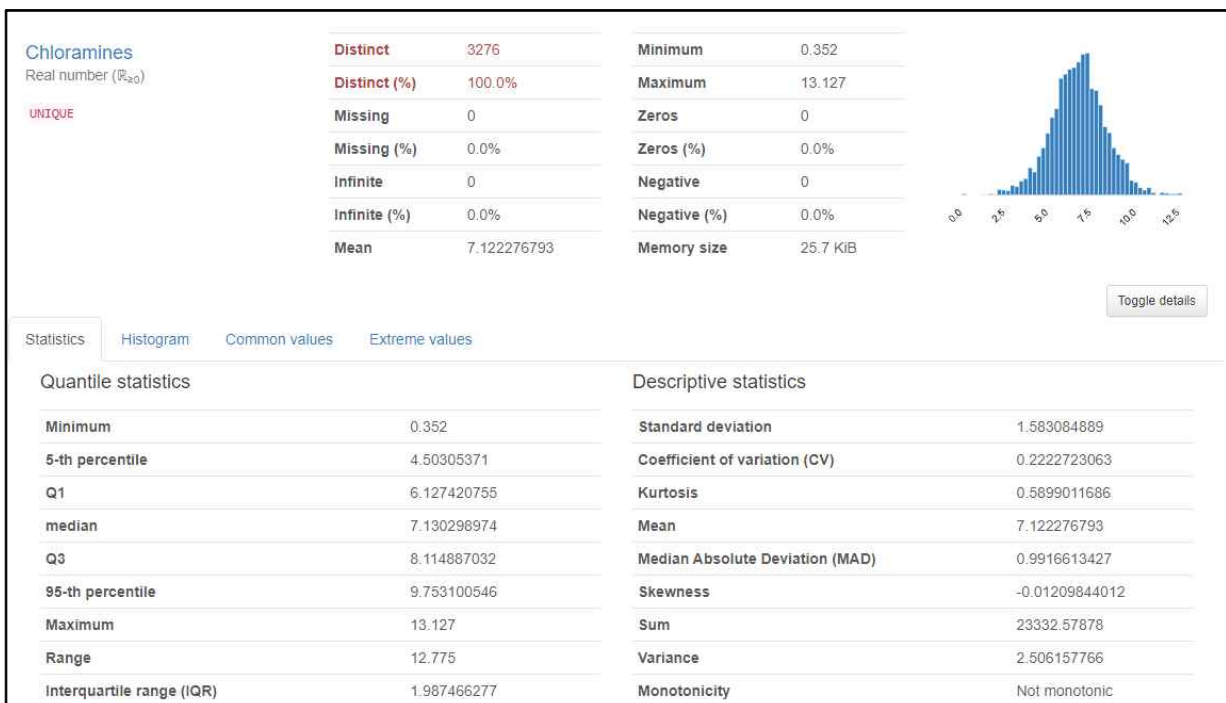
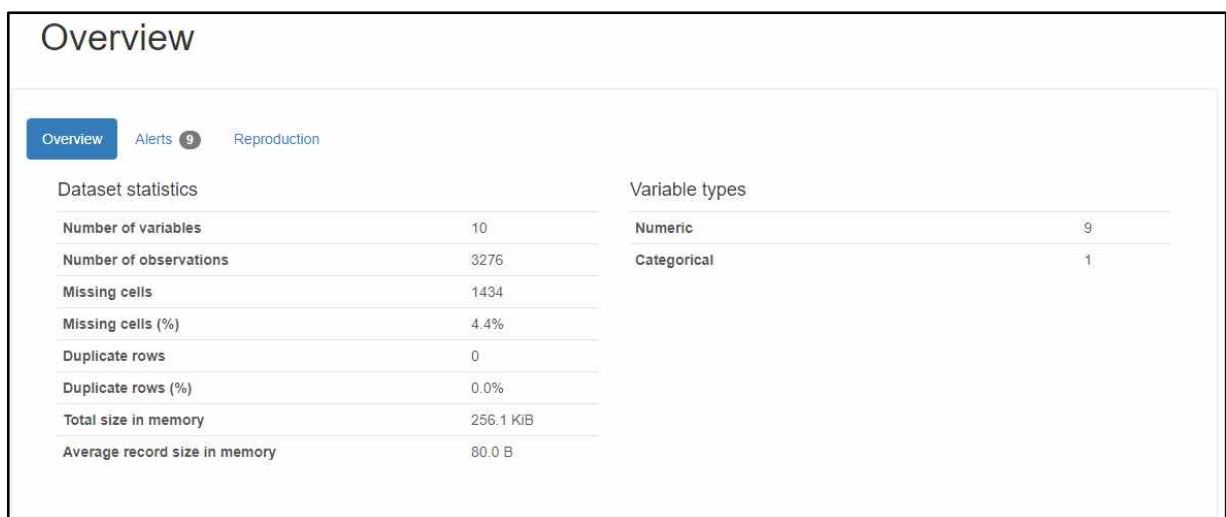
### 3. TIPS

#### □ pandas.profile

- 데이터를 핸들링하기 위해 주로 사용하는 pandas의 애드온 라이브러리

```
# 최초 라이브러리 설치 시
!pip install pandas_profiling # terminal에서 실행 시에는 맨 앞 '!' 제외
import pandas as pd
import pandas_profiling
df = pd.read_csv('my_file.csv')
profile = df.profile_report()
profile
```

- 실행 시 overview, 통계량, 상관관계수, 결측치 등의 report 등을 생성해줌





## □ List comprehension

- Python답게 코딩하는 방법 중 대표적으로 list comprehension이 있습니다. 간단한 반복문은 이를 활용하여 조금 있어보이게 만들어줄 수 있습니다.

# 0부터 9까지 정수의 제곱 리스트를 만들 경우	
# Old squares = [] for i in range(10): squares.append(i^2)	# New squares = [i^2 for i in range(10)]
squares [0, 1, 4, 9, 16, 25, 36, 49, 64, 81]	

※Link: [https://www.w3schools.com/python/python\\_lists\\_comprehension.asp](https://www.w3schools.com/python/python_lists_comprehension.asp)

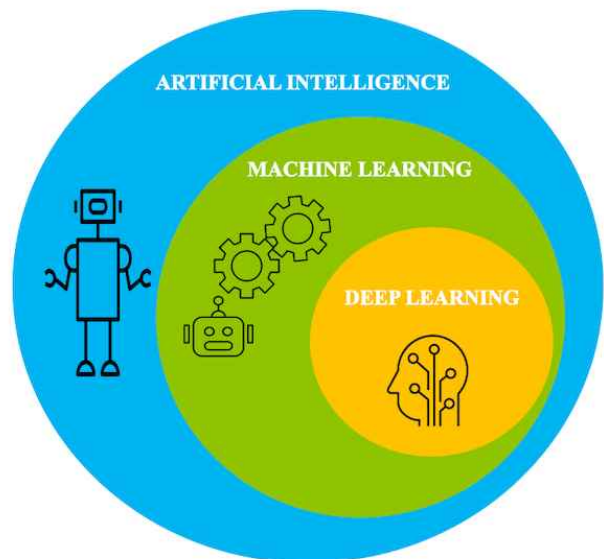
## □ Variable swap

- a와 b에 들어있는 내용을 바꿔주고 싶을 때는 간단하게 swap할 수 있습니다.

a = 10; b = 20	
# Old tmp = a a = b b = tmp	# New a, b = b, a
print(f"a: {a}, b: {b}") a: 20, b:10	

## □ AI 기초: AI? 머신러닝? 딥러닝?

- 인공지능(Artificial Intelligent, AI)
  - AI는 인간이 갖고 있는 학습, 추론 및 판단 능력을 컴퓨터로 구현하는 가장 포괄적인 개념입니다. 규칙 기반의 프로그래밍(ex. 발신자 기준으로 메일함 분류) 등이 파란색 영역에 해당합니다.
- 머신러닝(Machine Learning, ML)
  - AI를 구현하는 하나의 방법으로, 데이터의 특성과 패턴을 학습해서 미지의 데이터에 대한 추정치를 계산하는 방법입니다.
- 딥러닝(Deep Learning, DL)
  - 딥러닝은, 기본적인 신경망(NN)을 여러 층으로 구성해 학습하는 구조와 알고리즘입니다.
  - 대표적인 신경망 구조로는 DNN(Deep NN, 심층신경망), CNN(Convolution NN, 합성곱신경망), RNN(Recurrent NN, 순환신경망) 등이 있으며, 이를 더욱 발전시켜 Attention, 강화학습 등에 활용할 수도 있습니다.



## 4. Meanwhile, in K-water AI Lab.

### □ 제2회 K-water AI 경진대회 개최

- AI 연구센터에서는 빠르게 진화하는 AI 알고리즘의 효율적인 확보를 위해 개방형 클라우드 소싱(crowd sourcing, 일명 해커톤)을 개최하고 있습니다.
- 이를 통해 필요 AI한 알고리즘을 빠르게 확보하고 내재화할 수 있으며, 일반 데이터 분석가나 관련 대학, 대학원 학생들에게 K-water의 기술분야에 대해 홍보할 수 있는 기회를 마련하고 있습니다
- 11월 초부터 개최예정인 제2회 AI 경진대회에서는 용수 특성이 다른 각각 3개의 배수지에서 향후 2주간 필요한 용수량을 정확하게 예측해야 합니다.

- (참고, 1회 대회) “낙동강 하굿둑 개방에 따른 수리·수질 영향 AI 예측” 제1회 AI 경진대회('21. 10월) 시행
  - 낙동강 하굿둑 실시간 운영시 신속하고 정확한 의사결정 지원을 위한 AI모형 확보



### □ K-water AI Lab Hub 런칭 (<https://github.com/Kwater-AILab>)

- AI연구센터는 깃허브를 통해 센터 내외에서 수행하고 있는 여러 프로젝트의 내용, 목적과 코드를 공유하고 있습니다.
- 저희 AI연구센터와 협업이 필요하시면 언제든지 연락주세요 :D

Kwater-AILab / README.md

K-water AILab GitHub에 오신 것을 환영합니다. 🙌

이곳은 K-water AILab에서 개발한 다양한 AI Source Code를 공유하는 곳입니다. 앞으로 물관리 분야에 대한 다양한 코드를 공유할 예정입니다. 많은 관심부탁드립니다.

🔬 Research 분야 🙌

1. 🏠 Digital Twin과 연계한 시기술 개발 (In progress)
2. 🛡️ 댐안전관리 플랫폼 개발을 위한 인공지능 알고리즘 선정평가 연구 (In progress)
3. 🌊 AI기반 위성영상을 활용한 강우자료 신뢰도 개선 (In progress, Launch Jupyter notebook [launch](#) [binder](#))
4. 🌊 AI기반 녹조 광역 모니터링 기술 개발 ([https://github.com/Kwater-AILab/algae\\_monitoring](https://github.com/Kwater-AILab/algae_monitoring))
  - Launch Jupyter notebook [launch](#) [binder](#) Launch Web App [Open in Streamlit](#)
5. ☁️ 정수장 유입 수질(망간 등) 예측 알고리즘 개발 (In progress)
6. 🏠 막여과공정 AI운용모델 개발 (In progress)
7. ⚡ AI를 활용한 성공용수(국가물관리기본계획, 목표년도 2035년) 수요패턴 예측 알고리즘 개발 ([https://github.com/Kwater-AILab/water\\_demand\\_forecast](https://github.com/Kwater-AILab/water_demand_forecast))
  - Launch Jupyter notebook [launch](#) [binder](#)

📢 Conference 발표 🙌