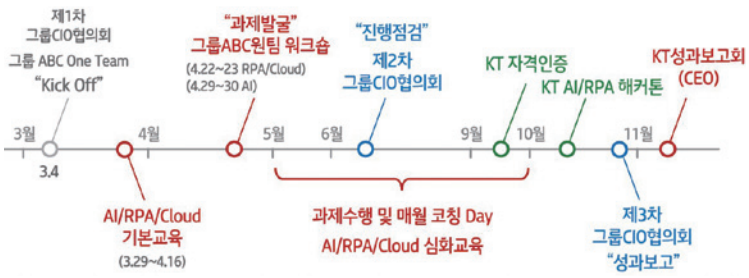
AIDU, AI? 내가 한다!(I DO)

글. 김희동 KT스카이라이프 AI/DX팀

최근 들어 모든 기업의 미래 먹거리 화두는 단연 AI, Bigdata, Cloud일 것이다. 관건은 남들보다 빨리 이를 적극 활용하는데 있고 성패는 이를 위한 디지털 트랜스포메이션 DX를 어떻게 남들과 달리 차별화된 비즈니스 경쟁력 DNA에 접목할 수 있을 것인가에 달렸다. KT는 이미 지난 2020년 AI를 미래 핵심 먹거리로 정하고 인재육성을 위해 내부 인력을 적극 활용하기로 방침을 정하였다. 그 일환으로 작년에는 그룹 미래인재 육성 프로젝트에 참가할 많은 지원자를 모집하기 위해 나이, 직급, 유사 직무 연관 등의 부서 조건을 없애니 50대 최고령자부터 AI와 평소 전혀 무관한 직무에 종사했던 사원들까지 스펙트럼을 넓힐 수 있었다. 지원자 중 대부분은 현장 AI 인력으로 육성되어 파트타임으로 6개 광역본부별 프로젝트에 참여하고, 후술할 KT 전용 AI 교육 플랫폼 AIDU를 활용하여 사내 코치지도 및 AI 분석 개발에 힘을 실었다. 올 초에는 이렇게 양성된 인력들이 AI 개발과 광역본부에 배치되고, 수십 개에 달하는 AI 혁신 프로젝트에 참여하는 등 관련 자격취득과 사내 AI 해커톤 경진대회를 통해 현장에 즉시 적용 가능한 솔루션을 선보이기도 하였다. 또한 KT는 이와 별도로 제4차 산업혁명시대의 대한민국의 성공적인 디지털 트랜스포메이션을 위한 AI One Team 프로젝트를 추진하고 있다.

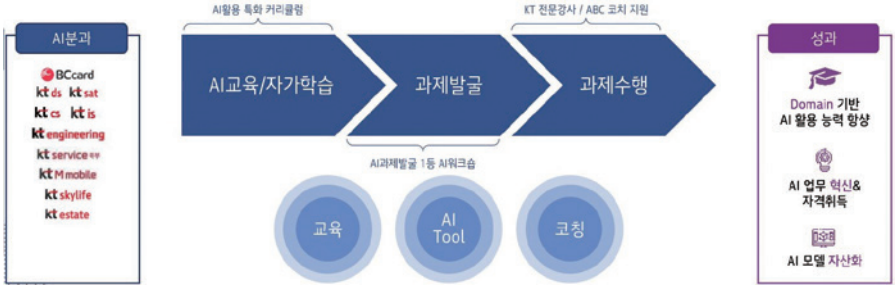
AI One Team은 국내 대표 산학연이 모여 대한민국의 AI 역량을 강화하고, 기업 및 산업의 AI 경쟁력을 드높이고자 2020년 출범한 협력체이다. AI open R&D와 인재양성을 주축으로 기업-스타트업을 잇는 AI 생태계 조성을 목표로 하고 있다. 이런 AI One Team은 KT, 현대중공업그룹, ETRI, 카이스트, 한양대 5개 참여기관으로 출범하여 코로나19 극복 등 사회문제 해결을 위한 연구개발(R&D)을 함께 진행하고 있다. 그 첫걸음으로 지난해 3월에는 KAIST와 함께 해외로부터 유입되는 감염위험도를 예측하여 국내 확산지역 예측 모델을 만드는 '코로나19 확산예측 연구 얼라이언스'를 구축하였고, 6월에는 LG전자 및 유플러스와 감염병 확산방지 모델을 개발한 바 있다. 결성 10개월만인 올 초에는 AI One Team의 공동 R&D를 통한 첫 성과로 ▲딥러닝 음성합성(P-TTS) ▲E2E 음성인식 ▲무빙 픽처(Moving Picture) 기술을 발표하기도 하였다. 5개 기관으로 출범한 프로젝트는 작년 6월 LG전자와 LG U+를 시작으로 한국투자증권, 동원그룹, 우리은행의 동참 릴레이가 이어져 제조, 중공업, 통신, 전자, 금융, 유통 등으로 산업 영역이 점차 확장되고 있다. AI One Team에 참여하고 있는 기관들은 그림과 같다.

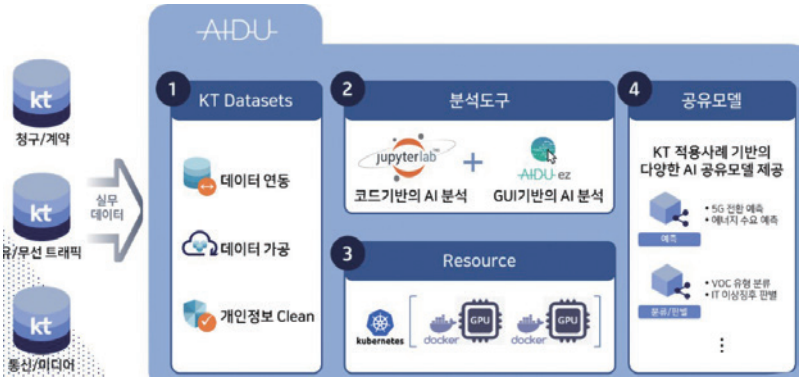
2021년에도 KT는 그룹 ABC OneTeam 2호를 출범하였다. 2020년 AI 역량강화 경험을 기반으로 교육이 실제 DX 업무혁신까지 이어지기 위한 KT ABC 추진 방법론을 그룹사 차원으로 확대 적용하자는 것이 그 추진 배경이다. 이를 통해 KT ABC 플랫폼의 그룹 Reference 확보와 그룹 적용사례의 B2B 사업화를 돕고 협업을 강화해 나갈 수 있는 토대가 마련될 것으로 기대된다.



필자가 몸담고 있는 KT스카이라이프의 AI/DX팀은 AI 분과에 지원하여 3월 말부터 동영상 기본교육을 이수하였고 4월 말부터는 현업에 적용 가능한 과제발굴을 위한 '그룹 ABC Oneteam 워크숍'에 참가하였다. 당초 그룹사 구성원들이 모여 이틀간 합숙하며 시너지를 도모할 계획이었으나 코로나로 인하여 팀즈(Teams)와 무랄(Mural) 플랫폼 등을 활용한 온라인방식으로 진행되었다.

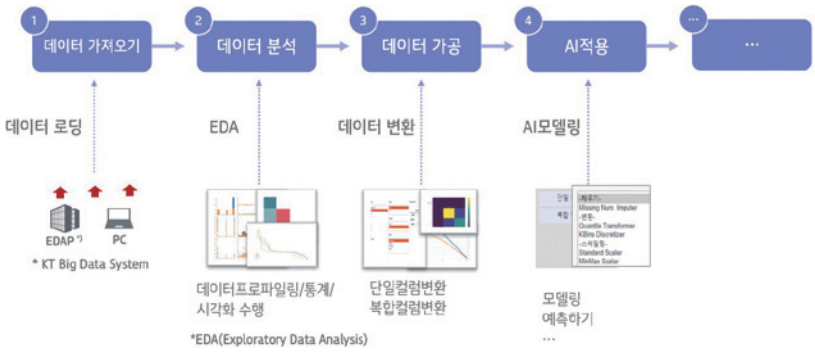
이를 지원하기 위해 KT는 그룹 ABC OneTeam 추진 TF를 구성하고 5개 조직(IT/AIDX/그룹인재개발실/그룹경영실/KTDS)으로부터 20여 명이 선발되었다. 과제실행을 위한 티칭도구로는 AI 분과에서는 AIDU, RPA에서는 개발 툴인 RPADU, 그리고 Cloud에서는 VM/PaaS, SaaS 등 KT Cloud가 활용되었다. 그럼 이제부터는 AI 분과에서 활용 중인 교육플랫폼 AIDU에 대해 자세히 기술하고자 한다.





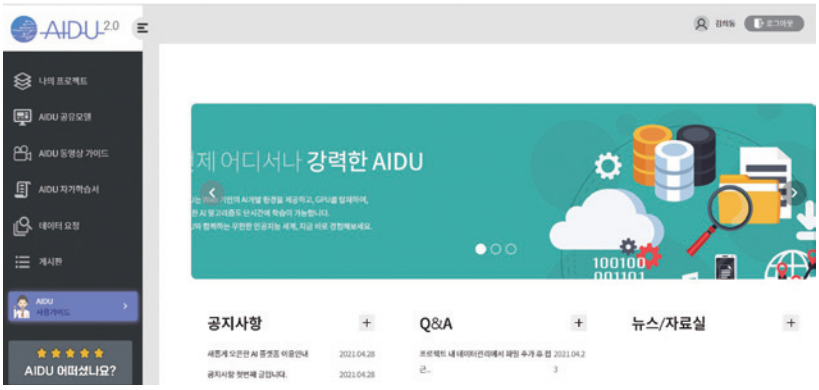
AIDU는 AI 분석 및 개발 프로젝트 수행이 가능한 Cloud 기반의 AI 플랫폼으로서 KT가 자체 개발하였다. 여기서 AIDU는 AI가 한다(AI do), 내가 한다(I do)의 의미로 누구나 쉽게 업무에 AI를 적용할 수 있도록 지원하는 툴을 뜻한다. 지금까지 AI를 배우고 또 개발하려면 Python과 같은 프로그래밍 언어를 습득함에 앞서 다운로드받아야 하고 또 Numpy, Pandas, Sckit-learn과 같은 라이브러리 형태의 머신러닝 프레임워크를 파이썬에서 실행하기 위해 명령프롬프트(cmd)에서 PIP로 인스톨해야하는 번거로움이 있었던 것이 사실이다. 다행히도 AIDU는 별도의 설치가 필요 없다.

왜냐면 클라우드 웹방식으로 데이터 분석부터 AI 모델링 및 운영관리까지 One-Stop으로 제공해 주는 No-Coding GUI 기반이기 때문이다. 또한 전사적으로 BI/DW, BDAP 등의 시스템 데이터와 Jupyterlab, AIDU ez와 같은 Web 기반의 개발 분석 도구를 제공하며 그런 공유 자원을 활용할 수 있는 GPU 환경에서 프로젝트 참여자끼리 기 개발된 AI 모델 소스코드를 참고하니 전문가가 아니더라도 친숙하게 다가갈 수 있는 장점이 있다.

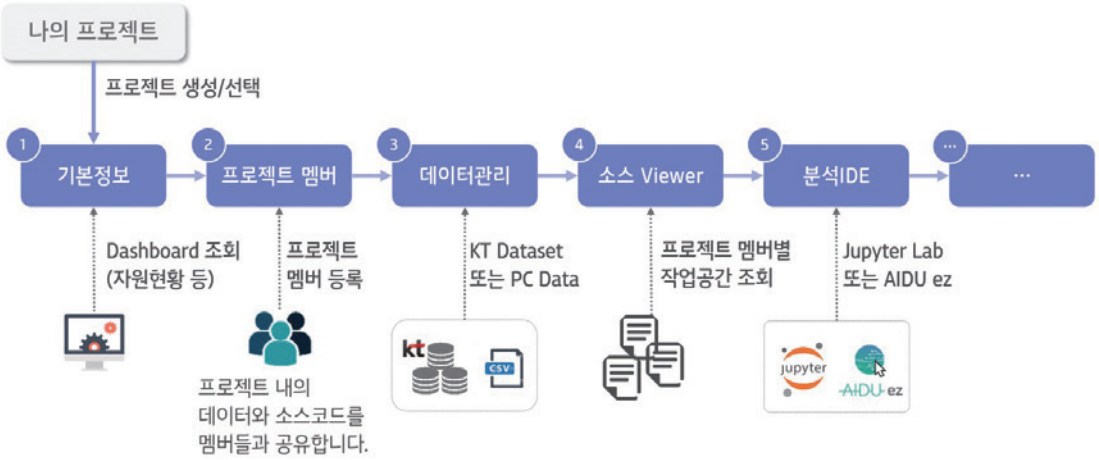


이런 AIDU 플랫폼에 접속하기 위해서는 일단 크롬 브라우저에서 해당 URL(lab.aidu.kt.com)로 로그인을 하면 된다. 하지만 AIDU 운영 사무국에서 사전에 인증받은 사용자에 한하여 ID와 PW가 부여되니 체험 버전을 바로 접해볼 수 없다는 점은 아쉽다.

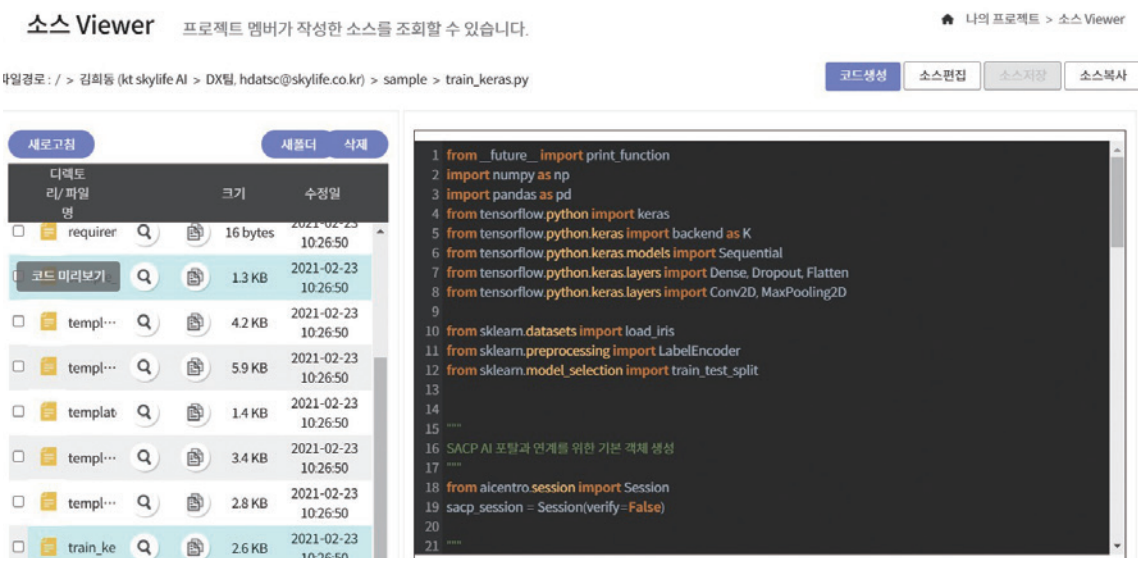
참고로 KT는 지난 5월 한양대의 소프트웨어 심화 교육 과정에 AIDU를 적용, 산업현장에서 사용되고 있는 AI 실습플랫폼을 경험할 수 있는 기회를 제공한 바 있다.



처음 로그인을 하게 되면 위와 같이 '나의 프로젝트'에서 '게시판'까지 총 6개의 GNB(Global Navigation Bar) 메뉴를 확인할 수 있다. 제일 중요한 AI 학습과제 작업공간인 '나의 프로젝트' 메뉴에서 실제 프로젝트를 생성하고 상세 분석 환경 설정이 이루어진다. 'AIDU 공유모델'은 다른 사용자가 생성한 AI 모델과 소스를 공유할 수가 있는데 분류, 회귀, 군집, 추천 등의 모델 유형과 AIDU Jupyter(코드)/ez(노코드) 도구별로 쉽게 검색이 가능하다. 또한 날짜, 조회, 인기, 댓글, 레벨순으로도 정렬이 가능하니 단순 조회는 물론 복사도 용이하다. 나머지 메뉴들의 기능은 이름 그대로라 추가 설명은 필요 없으니 핵심인 '나의 프로젝트' 메뉴에 대해 자세히 설명하고자 한다.



'나의 프로젝트' 메뉴에 있는 '기본정보' 탭에서는 프로젝트의 기본적인 정보와 자원 사용 현황을 대쉬보드 형태로 한눈에 알아볼 수 있다. 또한 프로젝트에서 사용 중인 Disk 현황과 참여 구성원, 그리고 분석에 필요한 통합개발환경(IDE)이 상세히 기록되어 있다. '프로젝트 멤버'는 AIDU 개발 컨셉이 클라우드 웹 접속으로 기획되었기에 원격에서 프로젝트에 참여하고 있는 멤버를 확인할 수 있으며, 구성원 추가 및 제거가 가능하다. 다만 주의할 점은 후술할 '소스 Viewer' 탭의 기능과 같이 서로 소스코드를 공유하고 있을 경우, 멤버 삭제가 진행된다면 소스도 함께 제거될 수 있으니 주의가 필요하다. '데이터관리' 탭에서는 AI 모델링 및 분석/예측에 활용할 데이터를 불러올 수가 있다. 이때, KT가 그룹사와 통합한 KOS, BI/DW 등 개인정보가 클렌징된 전사 시스템으로부터 제공하는 데이터셋을 가져올지, 아니면 이미 PC에 저장된 파일을 업로드할 지 선택하게 된다. AIDU ez 버전인 만큼 별도의 파일경로지정은 필요 없고 'Drag & Drop'으로도 가능하다. 업로드된 파일을 쉽게 확인하기 위해 미리보기 기능과 크기, 수정일 등도 제공하여 데이터가 잘못 업로딩 되는 실수를 사전에 막을 수 있다.

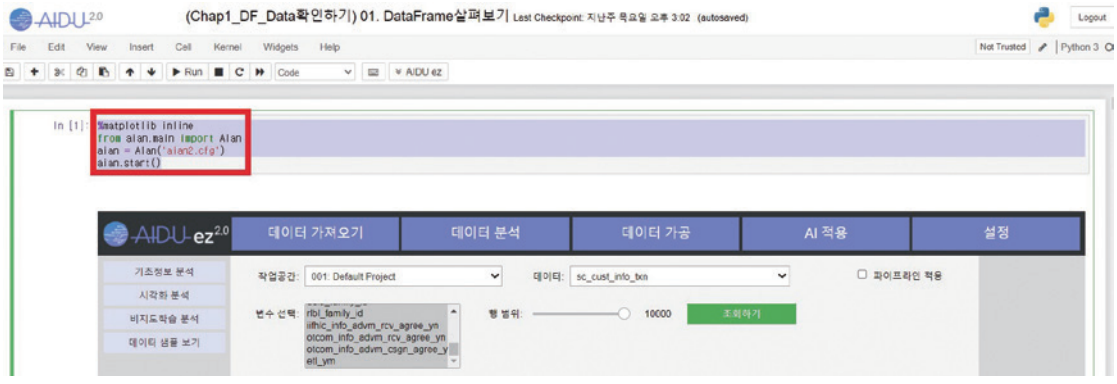


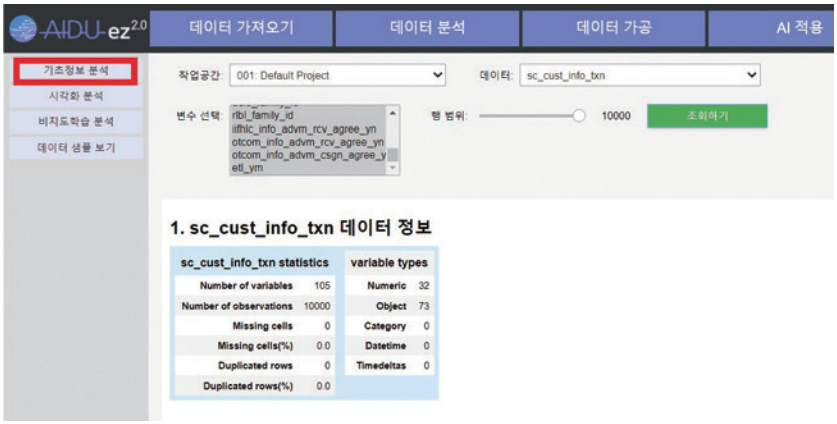
'소스 Viewer' 탭에서는 프로젝트의 참여하고 있는 구성원의 소스를 확인하고 필요시에는 복사 후, 편집이 가능하다. 소스 확인 및 복사, 편집 등이 하나의 플랫폼에서 원스톱으로 이루어지니 편할 듯하다. '분석IDE' 탭에서는 실제 AI 분석을 위한 통합개발환경(Integrated Development Environment) 설정을 할 수가 있는데 개발도구선택 → 시스템자원선택 → 분석환경선택 → 최종확인 순으로 되어있다. 먼저 개발도구에서는 Python과 R을 활용한 소스 코드 기반의 AI 모델링을 지원하는 Jupyterlab 또는 AI 개발 역량이 없어도 클릭 기반의 AI 모델링이 가능한 AIDU ez 중 하나를 선택하게 된다. 평소 Jupyter 개발환경에 익숙하며 코딩이 자유롭다면 Jupyterlab을, 그렇지 않다면 AIDU ez가 적합하다. 다음 시스템자원선택에서는 개발수준 또는 플랫폼 버전에 따라 상이하지만 보통 Basic(1core/2GB) - Expert(2core/16GB) - Special(4core/32GB)로 구분되며 괄호 안은 CPU와 Memory 할당을 각각 의미한다. 처음 AIDU ez 플랫폼이 개발되기 시작하면서 1.0 → 2.0, Beta → Stable로 시스템 안정화와 다양한 기능이 추가되고 현재는 AIDU ez 2.0 Stable이 배포되어 KT Auto-ML을 지원하는 AI 모델링 솔루션을 선택할 수 있다. 끝으로 최종확인 선택에서는 지금까지 설정했던 IDE 환경을 한 화면에서 확인이 가능한데 이마저도 번거로운 유저를 위해 Jupyterlab과 AIDU ez 빠른 생성 탭이 활성화되어 있어 통합개발환경 선택을 버튼 하나로 해결해준다. 다만 디폴트값이 기본 베이직으로만 세팅되어있다는 단점이 있지만 그럼에도 UI/UX를 고민했던 흔적이 역력하다.

분석IDE의 실제 UI는 위와 같은데, 앞서 글로 설명한 AIDU ez가 얼마나 편한지 보여드리자면, 먼저 코드 기반의 Jupyterlab(작업ID1798)으로 연결하고 그다음 판다스 라이브러리 호출을 시작(import pandas as pd)으로 최종 임의의 변수(df)를 지정하고 그 변수에 csv 데이터를 불러오는 코드를 입력해야 한다. 이후, 불러온 데이터가 맞는지 확인 작업을 거쳐야 하는데 그림과 같이 변수 df에 info() 함수를 사용하여 요약정보를 비교할 수가 있다.

```
[49]: import pandas as pd

from aicentro.session import Session
from aicentro.framework.keras import Keras as AiduFrm

aidu_session = Session(verify=False)
aidu_framework = AiduFrm(session=aidu_session)

df = pd.read_csv(aidu_framework.config.data_dir + '/sc_cust_info_txn.csv')

[51]: df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Columns: 105 entries, base_ym to etl_ym
dtypes: float64(21), int64(11), object(73)
memory usage: 8.0+ MB
```

그 결과값을 본다면, 0부터 9999까지 총 10000개의 행 또는 row 엔트리와 105개의 컬럼 또는 피처 엔트리로 이루어져있고 105개 변수데이터 타입은 실수형(float) 21개와 정수형(integer) 11개 그리고 오브젝트타입(문자열 또는 String) 73개로 구성되어 있음을 알 수 있다. 그럼 이 모든 과정이 AIDU ez에서는 어떻게 처리되는지 보자면, 먼저 편의상 빠른 IDE 생성을 해둔 작업ID 1807에 접속하기 위해 연결버튼을 클릭한다. 그러면 아래 붉은 박스로 표시된 코드와 같이 자동으로 %matplotlib이 실행되면서 미리 저장해둔 AIDU ez 2.0플랫폼으로 연결되는 결과값이 출력되는 것을 확인할 수 있다. 맷플롭립은 파이썬에서 시각화를 위해 자주 이용되는 라이브러리로 결국 AIDU 자체도 노코드 개발의 편이성을 위해 맷플롭립의 시각화된 결과값인 셈이다. AIDU는 파이썬에서 자주 사용되는 판다스, 넘파이, 사이킷런 등의 라이브러리가 자동으로 제공되므로 Jupyterlab의 import 명령어 코드는 단연 필요가 없고 데이터를 불러오기 위해 새로운 변수를 생성하고 그 변수에 데이터 디렉토리 지정 또한 '데이터 가져오기'란 탭으로 쉽게 해결된다.





'데이터 가져오기'는 다시 'PC에서 가져오기', 'AIDU에서 가져오기', '가공한 데이터 가져오기' 중 선택할 수 있는데 모두 마우스로 불러올 수 있게 설정되어 있다. 마지막 '가공한 데이터 가져오기'는 AIDU에서도 판다스를 활용한 데이터 전처리 기능을 제공하고 있으며 그렇게 해서 최종 가공된 데이터를 불러올 수도 있다. 사실 AI 모델링에서 가장 많은 시간과 노력이 할애되는 부분이 바로 데이터 전처리 과정인데 이를 다루는 'Feature Engineering'이란 학문이 따로 있을 정도로 매우 중요하다. 하지만 지면상으로 보여주기에는 후술할 시각화분석 부분이 적합하기에 이 부분은 아쉽게도 본 고에는 일부 정도만 설명하고 데이터 타입변경, 결측치처리, 카테고리/수치형별 이상치 처리 등 나머지는 생략되었다는 점을 미리 밝혀둔다.

그다음 불러들인 데이터의 정보를 확인하기 위해 Jupyterlab에서는 info() 함수를 사용하였는데 AIDU에서는 '데이터 분석' 탭의 '기초정보 분석' 메뉴를 선택하면 쉽게 가능하다. 다만 차이점은 row 엔트리는 관측숫자로, column 엔트리는 변수숫자로 대체되었고, 변수타입 또한 실수형(float) 21개와 정수형(int) 11개가 더해져 숫자형(numeric) 32개로 세분화되지 않았다는 점이다. 그리고 행 범위를 정할 때, 첫 번째 행은 변수명이 들어가고 두 번째 행부터 0번 데이터가 입력되므로 10000개 행이라면 0~9999까지 9999행에 변수명 자리를 위한 1을 더해10000으로 설정해야 한다. 글로는 다소 모호할 수 있기에 이번에는 위 예제보다 적은 수의 행들로 이뤄진 다른 파일을 불러와 Jupyterlab과 AIDU를 비교하여 보았다.

```
[9]: df = pd.read_csv(sacp_framework.config.data_dir +'/' + 'TrainData_0506.csv',delimiter=',')
[10]: df
```

[10]:

	url_len	url_num_hyphens_dom	url_path_len	url_domain_len	url_hostname_len	url_num_dots	url_num_underscores	url_query_len	url_num_query_para	url_ip_present
0	15	0	0.0	15.0	15	3	0.0	0	0	1
1	96	0	82.0	14.0	14	3	2.0	0	0	0
2	12	0	0.0	12.0	12	2	0.0	0	0	0
3	35	0	18.0	17.0	17	3	0.0	0	0	0
4	38	0	17.0	21.0	21	3	0.0	0	0	0
...	...	...	...	...	...	...	...	...	...	...
6103	37	0	24.0	13.0	13	3	0.0	0	0	0
6104	37	0	24.0	13.0	13	3	0.0	0	0	0
6105	37	0	24.0	13.0	13	3	0.0	0	0	0
6106	37	0	24.0	13.0	13	3	0.0	0	0	0
6107	37	0	24.0	13.0	13	3	0.0	0	0	0

6108 rows × 24 columns

우선 Jupyterlab에서 판다스 라이브러리를 활용하여 0~6107개 레코드를 확인하였다. AIDU에서는 자동으로 'First rows'와 'Last rows' 각각 5개씩 화면에 보여지는데 Jupyterlab에서 맨 처음 5개와 마지막 5개 정보를 따로 출력하려면 각각, head()와 tail() 함수를 사용하면 된다. AIDU에서는 '데이터 분석' 탭의 '데이터 샘플 보기' 메뉴를 이용하면 아래와 같이 표시될 변수를 마우스로 지정하여 선택할 수도 있고 행의 개수 또한 '행 범위'를 정할 수 있다. 만약 Jupyterlab과 같이 맨 앞(0~4)과 맨 뒤(6103~6107) 5개 행을 보고 싶다면 '행 범위'가 5가 아닌 변수명 자리를 위해 6으로 설정해야 Jupyterlab의 결과값과 같게 된다는 것을 알 수 있다.

데이터 확인(Sample)

First rows

	url_len	url_num_hyphens_dom	url_path_len	url_domain_len	url_hostname_len	url_num_dots	url_num_underscores	url_query_len	url_num_query_para
0	15	0	0.0	15.0	15	3	0.0	0	0
1	96	0	82.0	14.0	14	3	2.0	0	0
2	12	0	0.0	12.0	12	2	0.0	0	0
3	35	0	18.0	17.0	17	3	0.0	0	0
4	38	0	17.0	21.0	21	3	0.0	0	0

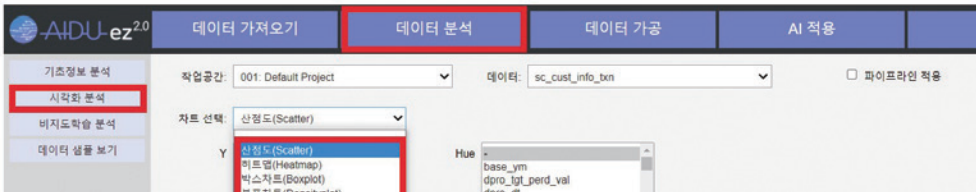
Last rows

	url_len	url_num_hyphens_dom	url_path_len	url_domain_len	url_hostname_len	url_num_dots	url_num_underscores	url_query_len	url_num_query_para
0	15	0	0.0	15.0	15	3	0.0	0	0
1	96	0	82.0	14.0	14	3	2.0	0	0
2	12	0	0.0	12.0	12	2	0.0	0	0
3	35	0	18.0	17.0	17	3	0.0	0	0
4	38	0	17.0	21.0	21	3	0.0	0	0

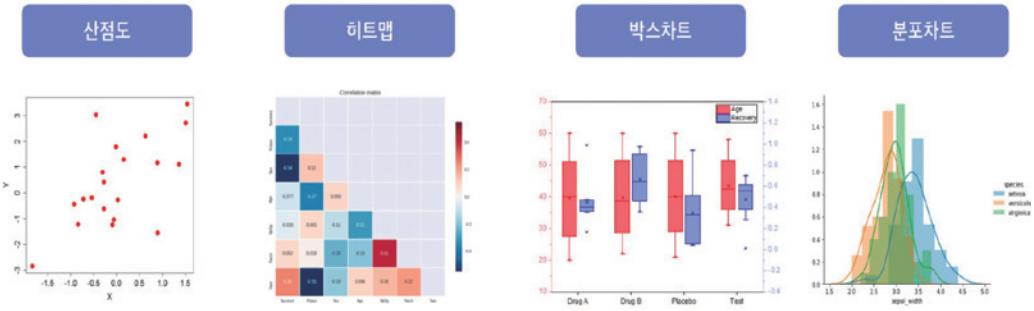
앞에서 AIDU의 '행 범위'를 설정하면서 분석에 필요한 핵심 변수를 선택할 때도 마우스로 쉽게 지정만 해주면 된다고 하였는데 이런 기능 모두가 Jupyterlab에서는 코드로 실행해야만 하니 이 역시 얼마나 불편한 작업인지 굳이 시연이 없더라도 짐작할 수가 있다. 그것보다 이 작업이 왜 필요한지 인지하는 것이 더 중요한데 AI Life-cycle(문제정의-자료수집-전처리-모델링-검증-최적화&배포)로 보자면 첫 번째 문제정의와 관련이 있다. 이는 문제를 정의하여 이를 해결해 줄 수 있는 꼭 필요한 변수를 선택하여야 하고 필요에 따라 삭제 또는 새롭게 가공하여야하기 때문이다. 학습에 필요한 변수(피처) 데이터가 많으면 많을수록 일단은 좋다. 없어서 의미 있는 결과값이 나오지 못하는 경우보다는 그렇다. 하지만 무턱대고 모두 학습시키기에는 아무리 GPU 기반이라 할지라도 그 양에 비례하여 시간은 증가하기 마련이다. 전체 데이터를 학습에 적합한 일정 크기로 나누는 Batch size, 처음부터 끝까지 몇 번 반복학습을 할지 정하는 Epoch, 끝으로 한 번의 Epoch를 학습시키기 위해 필요한 Batch의 수인 Iteration 과정을 고려해야하기 때문이다. 이 부분은 학습데이터가 적어 일반화하지 못해 특정 패턴을 찾지 못하는 과소적합(Underfitting)과 데이터의 Noise까지 디테일하게 학습하여 테스트 데이터에 적용하였을 때 오차가 증가하는 과적합(Overfitting) 사이에서 어떻게 최적화를 이룰지에 관한 문제이다. 다시 말해, 모델 복잡도가 증가할수록 Bias는 줄일 수 있지만 그에 따라 Variance가 증가하는 Bias-Variance Trade-off 문제를 어떻게 조화하여 최고의 모델을 위한 최적의 복잡도 지점을 찾느냐인 것이다. 따라서 기계에게 학습시킬 데이터셋을 정하는 하이퍼파라미터 값은 사람의 몫이기에 전처리 과정이 무엇보다 중요하다. 또한 변수 간 상관관계가 높다면, 회귀(Regression)모델의 경우 계수의 분산이 증가하기에 유의미한 분석에 왜곡을 일으키는 원인이 되므로 변수가공 또는 제외시키는 것이 바람직하다. 예를 들자면 집값을 예측하는 회귀모델에서 아파트 평수와 방의 개수는 강한 양의 상관관계가 있을 것이 자명하기에 둘 중 하나는 전처리 과정에서 빼는 것이 바람직하다. 그런데 대부분은 위 예와 달리 변수 간 가늠이 어려운 경우가 많다. 이럴 때는 분산팽창요인(Variance Inflation Factor)을 구하여 해결이 가능하다. 독립변수들의 VIF값이 10을 넘지 않는다면 문제가 없다고 보는 것이다. 그렇기 때문에 모

형의 절약성(Parsimony) 관점에서 두 모형의 설명역이 비슷하다면 적은 독립변수들로 구성된 쪽이 더 경제적이기에 선호된다. 이는 다뤄야할 변수 데이터가 많을수록 페널티를 부과하여 과적합을 방지하는 능형회귀모델(Ridge)이 탄생한 이유이기도 하다.

```
In [1]: %matplotlib inline
from alan.main import Alan
alan = Alan('alan2.cfg')
alan.start()
```

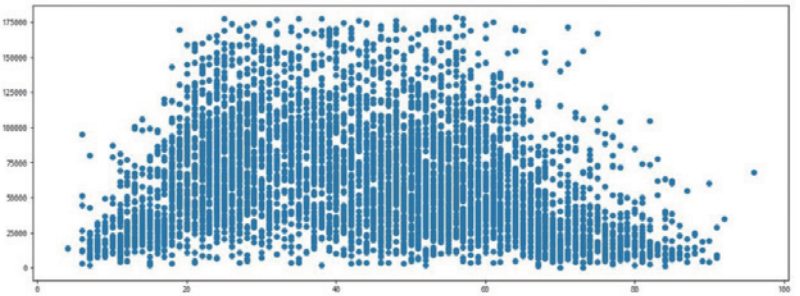


끝으로 시각화(Visualization)는 데이터를 분석하는데 있어 고도의 스킬 없이 직관적으로 데이터 사이의 패턴이나 특징을 보다 쉽게 파악하여 다양한 인사이트를 도출할 수 있는 장점이 있기에 많은 분야에서 활용되고 있다. 이 역시 많은 시각화 종류 중 하나를 선택하고, 또 여기에 적용할 데이터 필터링 등 파이썬에서 코드로 실행시키는 것조차 트레이닝이 없다면 처음부터 배우면서 시작해야 한다. 하지만 처음 시각화 분석을 해야 하는 사람도 AIDU만 있다면 걱정이 없을 정도로 심플하다. '데이터 분석' → '시각화 분석' → '시각화종류 및 변수 선택'만 지정해주면 된다. AIDU에서 제공하는 시각화 종류는 가장 많이 사용되는 산점도(Scatter), 히트맵(Hitmap), 박스차트(Boxplot), 분포차트(Densityplot) 4가지를 제공하고 있다.

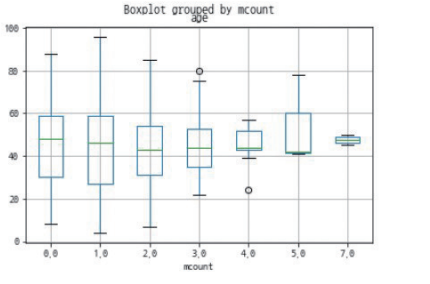


산점도는 2개의 수치형 변수 사이에서의 상관관계를 보여주며, 히트맵은 그 상관관계 값을 색으로 나타내 준다. 박스차트는 범주형 데이터를 X축에 두고 그에 따른 수치형 값을 Y축에 표현하여 한 눈에 알기 쉽게 해주며 분포차트는 수치형 변수 표현에 적합하다.

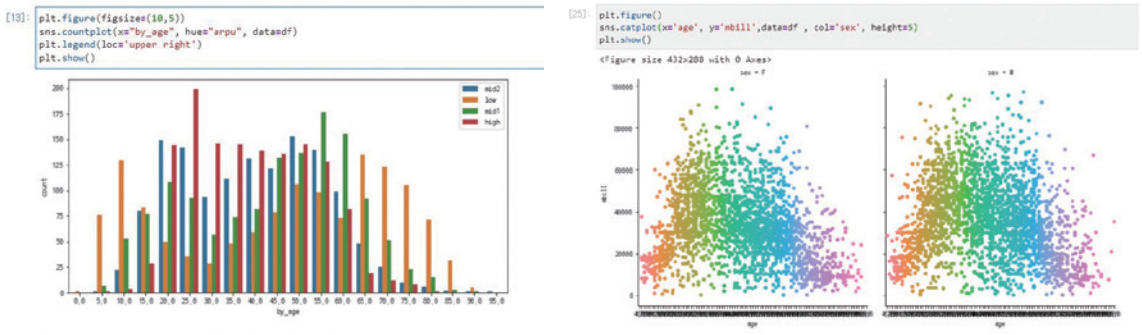
```
[7]: plt.figure(figsize=(16,6))
plt.scatter(y=df["tbill"], x=df["age"])
plt.show()
```



```
[10]: df.boxplot(by="mcount", column="age")
[10]: <matplotlib.axes._subplots.AxesSubplot at 0x7fae490f2630>
```



Jupyterlab에서는 AIDU에서 제공하는 시각화 종류보다 훨씬 더 많이 제공해준다. 다만 위와 같이 간단한 코드라 할지라도 누군가에게는 부담이 될 수도 있다. 그럼에도 시각화에 특화된 Seaborn 툴이 가능하다라는 점은 아무리 AIDU가 편하긴 해도 기능적인 측면에서 특장점이 될 수 있다. 아래는 Seaborn 모듈을 PIP로 인스톨한 후 나타낸 Countplot과 Catplot이다. 다양한 옵션들을 바탕으로 다채로운 시각화가 가능하니 AIDU가 제공하는 산점도와 확연히 차이가 남을 확인할 수 있다. 특히, Countplot의 경우 히스토그램과 같이 항목의 개수분포를 알 수 있지만, 히스토그램이 연속형 변수에만 한정적인 것에 반해 범주형에도 사용될 수 있다. 따라서 우선 노코드기반의 AIDU ez로 다양한 모델링을 접해본 다음 어느 정도 AI 모델링에 익숙해지면 마우스로 동작시켰던 명령 중, 쉬운 것부터 차근차근 Jupyterlab 환경에서 파이썬으로 실행시켜 보는 것을 권해본다.



이제 남은 것은 AI 적용뿐인데 이 부분은 실제 고객의 세부 데이터와 모델링을 통한 결과값이 공개될 수 있기에 본 고에서는 생략되었다. 이 결과값에는 민감한 고객정보가 담길 수 있어 항상 조심해야 하기 때문이다. 아무리 개인을 식별할 수 없는 데이터는 마음대로 사용할 수 있는 데이터 3법이 통과되었다지만 추가 정보 가공으로 식별이 가능한 경우는 법적 처벌 대상이어서 항상 각별한 주의가 필요하다. 그럼 이것으로 KT가 자체개발한 Jupyterlab과 AIDU ez의 설명을 마치도록 하고 그렇다면 왜 도대체 이 시점에 KT는 AI 자강론 카드를 꺼냈을까에 대한 의미를 되짚어보고 방송기술에 들이닥칠 AI 파고에 대해 예측해보며 끝마쳐볼까 한다.

주지하디시피 KT는 일찍이 Digico(디지털 플랫폼 기업)로의 도약을 천명하며, 더 이상 통신만 잘하는 회사가 아닌 최신 글로벌 트렌드에 발맞춰 AI, Bigdata, Cloud 등, IT기술을 중심으로 각 산업의 디지털 변혁(DX)을 이끌어가는 선도 기업으로서의 청사진을 대내외에 밝힌 바 있다. 그러나 국내 현실은 녹록지 않다. 점차 늘어가는 디지털 인력 수요를 질적으로나 양적으로 뒷받침해주지 못하고 있는 것이 사실이다. 따라서 그 첫 발걸음으로 '미래인재육성 프로젝트'가 지난해 구현모 대표 취임 직후 시행되었다. 프로젝트 1기 결과로는 AI 기반 고객센터 'AICC', 가상 상담 'VoiceBot', 앞서 AIDU에서 소개한 고객 데이터 분석을 통한 '맞춤 서비스 추천' 등 전체 추진되었던 프로젝트 중 60% 정도가 상용화 과제로 채택되었을 만큼 괄목할 만한 성과를 거뒀다. 올해도 인재육성 프로젝트 2기 모집을 통해 선발된 인원이 전일제로 5개월간의 교육과 AI/DX 프로젝트 실무과제 개발에 투입되었다. 이처럼 외부 수혈론이 아닌 내부 자강론이 주목을 받는 이유에 대한 해답은 AI Life-cycle에서 찾을 수 있다. 아무리 AI 모델링 전문가라 할지라도 해당 산업 분야의 도메인 지식이 없다면 첫 단계인 문제정의에서부터 그릇될 개연성이 매우 높기 때문이다. 문제정의는 전체 프로젝트의 성패를 좌우할 첫 단추로 우리 몸에 비유하자면 어디가 아픈지에 대한 진단이다. 제대로 된 진단이 우선되어야 그에 알맞은 처방을 내릴 수 있다. 물론 영입으로도 이 부분을 해결할 수도 있겠지만 도메인 전문가와 항상 원활한 협업을 기대할 수는 없다. 특히 프로젝트의 대부분의 시간과 노력이 소요되는 데이터 전처리 과정은 도메인 지식이 왜 절대적인지 말해준다. 결측치를 채울 때도, 또 이상치를 없앨 때도 단순 편의를 위한 접근은 자칫 잘못된 데이터를 가공해내는 결

과를 초래할 수 있다. 단순 시스템 오류 또는 작업과 같은 인위적 행위에 따른 기록인지 아니면 필연적으로 발생했으며 통계학적으로 유의미한 아웃라이어인지에 대한 안목은 철저하게 오랜 기간 습득된 도메인 지식으로부터 대부분 얻을 수 있기 때문이다. 오류가 있는 데이터를 학습한 모델링의 결과 역시 정확하다. AI는 거짓 또한 있는 그대로 학습하기에 해당 데이터를 정확하게 분석하게 되며, 때문에 잘못 가공된 데이터 외의 실제 문제에 적용했을 경우 오차는 클 수밖에 없다. 모두에서 언급한 바와 같이 필자는 '그룹 ABC OneTeam' AI 분과에 편입되어 지금은 프로젝트 진행이 한창이다. 10월 전사적 해커톤 경시대회에서 시연을 목표로 문제정의를 마치고 데이터 수집 방안에 대해 관계 부서와 협의 중에 있다. 아직은 구체적으로 밝힐 수는 없으나 프로젝트명은 'AI 비전을 활용한 장애 인지'다. 평소 장애 인지를 위해 24시간 MCR 교대근무자의 단순 반복적으로 행해지는 점검의 부담을 덜기 위해 고안되었다.

단계	정의	장애변수 감지	장애제어 주체	관제 주체	비고
0	**레가시 관제** : 엔지니어가 모든 것을 통제. 시스템 경고와 일시적 개입 허용	인간	인간	인간	아날로그(초기)
1	**부분 보조 관제** : A/V Signal Loss 등 시스템이 제한된 일정부분 개입	인간 & 시스템	인간	인간	디지털(초중기)
2	**시스템 보조 관제** : 엔지니어가 설정해 놓은 룰대로 유사시 시스템 자동 절체	시스템	인간	인간	디지털(~현재)
3	**부분 AI 관제** : 일부 시스템에 AI 적용. 선 AI 보고, 후 엔지니어 조치	AI	AI &인간	인간	**DX 초기(~2022)** - 프로젝트명 1) AI-Vision (LOGO)을 활용한 장애인지(가칭)
4	**고도 AI 관제** : 특정 예외 상황을 제외한 대부분의 시스템 AI 적용	AI	AI	AI &인간	**DX 중기(~2025)** 1) EPG 키워드중심 토픽 모델링 및 프로그램 객체기반 매칭 2) 기존 코덱의 압축 성능을 개선하기 위해 보조적 사용
5	**완전 AI 관제** : 모든 시스템 AI 적용. 선 AI 조치, 후 엔지니어 보고	AI	AI	AI	**DX 후기(~2030)** 1) A/V 장애발생 요인 Feature VSM 모델링 기반 이상 탐지 2) Post 코덱(H.266)을 넘어 AE (Autoencoder) 메인 처리

이는 위 지능형 관제 레벨표에서 보듯, 3단계로서 가장 기초적인 부분 AI 관제에 해당한다. AIDU Jupyterlab 플랫폼에서 비전 인식을 위해 합성곱신경망(CNN) 알고리즘과 원래 채널이 가질 수 있는 로고와 구별하기 위해 지도학습(분류) 개념을 활용할 계획이다. 다행히 Python에서 오픈소스 컴퓨터 비전(OpenCV) 라이브러리를 불러(Import)올 수 있어 아직까지는 큰 무리 없이 데이터 확보 방안과 그 후 전처리 과정에 대해 논의 중에 있다. 이 프로젝트가 성공적으로 끝마쳐진다면 보다 더 정교해진 솔루션을 위해 EPG 또는 제공되는 시놉시스의 키워드를 추출, 이를 토픽 모델링하여 실제 화면의 객체들과 비교분석할 수 있는 모델개발에 착수할 예정이다. 그다음 단계인 마지막 레벨 5는 아직 구체화되지는 않았으나 개념은 장애요인별 A/V 변수들을 선정하게 되고 그러면 고차원의 평면에 매핑된 데이터 중에서 주요 피처들 위주로 차원축소하여 X-Y 평면에 배열시키는 것이다. 그 후 시계열로 분석하게 되면 정상 데이터들끼리는 서로 오밀조밀 군집화되어있을 것이고 그렇지 않은 이상치들은 LOF(Local Outlier factor)값이 상대적으로 높은값을 가질

것이기에 이 값이 지속적으로 기준치 이상을 유지하였을 경우 시스템 장애로 판단하여 경보를 발생시켜주는 원리이다. 물론 지금도 ACO라 하여 자동절체기가 있긴 하나 채널별 특성(ex: 바둑TV의 오랜 정지화면과 무음 기간, 지상파 방송사들의 정기적인 새벽 정파 등)과 특정 포맷의 오디오 감지 불가(ex: Dolby AC-3), 특정화면 불량(ex: 색 번짐 또는 모자이크 현상)시 감지 불가 등의 단점이 많았다. 지능화된 관제 시스템들은 기존의 문제점들을 해결해주면서 처음에는 보조적 수단으로 쓰이다 차츰 그 비중이 증대되어(ex: 시청률이 저조한 새벽시간대 우선 적용 후 확대 운용) 결국에는 엔지니어가 설정해주는 권한 내에서 스스로 처리하고 인간은 보고만 받는 형태로 발전될 것이라 예상된다.

그뿐만이 아니다. 통신이 방송과의 경계를 허물었듯이 AI의 불씨(가능성)는 점점 타오르고 있어 강한 기류(AI 신드롬)를 만나 통신이란 강을 넘어 방송영역에까지 문을 두드리고 있다. 이미 포스트 프로덕션 과정에서 편집, 색보정 및 복원, 화질 개선 등으로 그 쓰임새가 확장되고 있다. 또 하나의 도전 영역은 코덱 분야가 되지 않을까 싶다. 현재 스카이라이프의 주력이라 할 수 있는 비디오 코덱의 레퍼런스는 H.264(AVC)이다. 일반적인 HD(720P with 30fps)부터 Full-HD(1080P)까지 처리가 가능하다. 이와 함께 UHD 채널은 H.265(HEVC)로 4K(3840×2160) 신호를 압축하여 전송(10M대 bps)하고 있으며 작년 표준화 공표가 끝난 H.266(VVC)의 경우 8K(7680×4320)부터 16K(15360×8640), VR 등 대용량의 비디오 신호를 커버할 것으로 예상된다. 그러나 아무리 새로운 H.266 코덱이 기존 H.265 대비 비트레이트 효율성을 50% 정도 높였다지만 평가기관마다 다르긴 해도 HEVC 대비 인코딩 시간이 4~6.5배 늘어났고 디코딩 시간은 조금 적거나 많게는 1.5배 더 걸렸다는 점, 그리고 HEVC 때와 달리 여러 업체가 모여 서로의 특허가 공유된 복잡한 라이센스 구조를 갖는다는 점에는 전면 도입에 많은 의구심이 드는 것 또한 사실이다.

바로 이 부분을 심층신경망 계열이라 볼 수 있는 오토인코더(Autoencoder)가 공략할 것으로 보인다. 머신러닝의 기술 발전은 이미 지도학습에서 비지도학습을 거쳐 뉴럴넷 기반의 심층신경망으로 옮겨간 지 오래다. 관련 논문만 보더라도 월등하다라는 것을 알 수 있다. 오토인코더는 CNN의 피처 추출 방식과 같이 모든 데이터들을 대상으로 하지 않고 특정 입력값을 받아 차원을 축소(Manifold)하고 특징값(Latent Feature)으로 변환하는 신경망인 Encoder와 이 특징값을 출력값으로 변환시키는 신경망인 Decoder, 그리고 이 둘 사이에서 학습을 진행하는 은닉층(Hidden layer)으로 이뤄졌다. 그리고 입력값(X)과 출력값(X')의 차이를 의미하는 손실함수를 최소화하는 것이 성능의 관건이 된다. AE의 뿌리라 할 수 있는 심층신경망의 컨셉은 꽤 심플하다. 인간 뇌는 천억 개의 뉴런과 천조 개의 시냅스가 복잡하게 연결되어 서로 전기신호를 주고받는데 이를 인지과정이라 한다. 딥러닝은 그저 이런 뉴런과 시냅스의 병렬연산을 알고리즘으로 재현해주는 것이다. 예전에도 이미지 압축의 효율을 높이기 위해 딥러닝을 시용해 보려는 시도는 있었으나 SVM 등의 신기술에 밀려 크게 빛을 보지는 못하였다. 그러나 컴퓨팅 기술의 비약적 발전과 기계학습의 주를 이루는 딥러닝 기반으로 개발된 오토인코더는 다를 듯하다. 첨엔 기존 코덱의 인루프나 후처리 필터 등에 쓰여 압축 효율을 높여주는 보조적 형태로 사용되다 결국에는 핵심이 될 것으로 예상된다.

『단언컨대 종합해보면 기업의 사활적 이익 관점에서 볼 때, AI는 더 이상 방향 선택이 아닌 속도의 문제다. 이제 AI 구슬은 서 말이 아니라 차고도 넘친다. 도메인 지식은 충분하나 아직 꿰는 법을 몰라 망설이고 있는 수많은 엔지니어들에게 AIDU는 어쩌면 실타래를 풀어줄 실마리가 될 지도...』