

# 2023 년도 SW 마에스트로 과정 제 14 기 프로젝트 최종보고서

## 1. 프로젝트 요약

<b>프로젝트명</b>	(PyViz) 코딩보다 직관적인 데이터 중심 인공지능 교육 도구 사이트		
<b>프로젝트 소개</b>	드래그 앤 드롭 방식으로 데이터를 조작하고 신경망을 구축할 수 있는 웹 기반 플랫폼. 사용자 친화적인 인터페이스를 통해 머신러닝 모델의 설계와 훈련 과정을 시각적으로 이해하고 체험할 수 있도록 도움.		
<b>기술 키워드</b>	웹, 데이터 시각화, 인공지능		
<b>ICT 연구개발 기술 분류</b>	융합 SW - 인터넷 · 서비스 SW		
<b>팀 명</b>	LucidAI	<b>팀원</b>	김창연, 양석훈, 장율
		<b>멘토</b>	김현, 조재화, 정좌연
<b>목적 및 필요성</b>	프로그래밍은 데이터를 조작하는 방법을 작성하는 과정이며, 이 과정에서 프로그래머는 데이터가 조작되는 과정을 직접 상상해야 함. 이 때문에 일부 학생들은 프로그래밍을 이해하는 데 어려움을 겪음. 이를 해결하기 위해 현재 시장에 나와 있는 시각적 프로그래밍 도구들은 주로 코드의 절차를 시각화하는 데 집중되어 있어, 사용자들이 실제 데이터의 형태를 인식하기 어렵다는 기존 프로그래밍 방식의 문제를 해결해주지 못함.		
<b>프로젝트 개요</b>	PyViz 는 사용자가 데이터를 중심으로 인식하고 이를 바탕으로 인공지능을 습하고 조작할 수 있도록 함으로써, 인공지능의 작동 원리를 보다 빠르고 깊이 있게 이해할 수 있게 도와줌.		
<b>프로젝트 수행 방안</b>	프레임워크 개발, 웹사이트 개발		
<b>개발 결과물</b>	<a href="https://pyviz.site">https://pyviz.site</a>		
	Python 객체 렌더링 프레임워크 PyX Python 드래그 앤 드롭 웹 인터페이스 사이트 PyViz PyViz 용 Pytorch 호환 모듈(개발중) PyViz 시범용 웹사이트 <a href="https://pyviz.site/">https://pyviz.site/</a>		
<b>결과물 활용방안 및 기대효과</b>	인식구조와 밀접한 시각화를 통한 직관적인 인터페이스를 통해 데이터와 머신러닝 모델을 쉽게 다룰 수 있게 하는 사이트로 제공하려 함. 협업과 교육에도 유용한 도구가 될 것으로 예상함.		

## 【 목 차 】

- **목적 및 필요성** . . . . . 1
  - 문제인식 . . . . . 1
  - 기획 의도(문제해결) . . . . . 2
  
- **프로젝트 개요** . . . . . 2
  - 프로젝트 소개 . . . . . 2
  - 주요 기능 . . . . . 2
  - 시장/소비자 동향/분석, 경쟁 제품 비교/차별화 요소 등 . . . . . 3
  - 시스템 구성도 . . . . . 7
  - 개발 환경 . . . . . 8
  
- **프로젝트 수행 방안** . . . . . 8
  - 역할 분담 . . . . . 8
  - 추진 일정 . . . . . 8
  - 수행 방법 . . . . . 9
  - 문제점 및 해결방안 . . . . . 9
  
- **개발 결과물** . . . . . 10
  - 결과물 링크 . . . . . 10
  - 결과물 형태 . . . . . 10
  - 프로젝트 주요 결과물김 . . . . . 11
  
- **결과물 활용방안 및 기대효과** . . . . . 17
  - 결과물 활용 방안 . . . . . 17
  - 기대효과 . . . . . 17
  
- **담당멘토 의견** . . . . . 18
  
- **중간점검 주요 의견 반영 내역** . . . . . 18

## 2. 프로젝트 최종보고서

### □ 목적 및 필요성

#### ○ 문제인식

- 프로그래밍이 현대사회에서 중요한 요소로 부각됨에 따라 코딩 교육이 활성화되고 있음. 그리고 이를 돕기 위한 툴과 프로그램이 다수 나와있으며, 관련 강의도 인기가 많음. 하지만 **프로그래밍은 학생들에게 여전히 어려운 과목으로 인식**되고 있으며, 학생들은 프로그래밍 학습에 많은 어려움을 겪고 있음. 한 연구에서는 컴퓨터 프로그래밍 기초 과정에서 학생들의 실패율이 28%에 달한다는 연구 결과도 있음.
- 프로그래밍은 데이터 조작에 대한 추상적인 사고를 필요로 하며, 이 과정에서 데이터가 변화하는 과정을 프로그래머가 직접 상상하며 프로그램을 작성해야 함. 일반적인 **학습자들의 인식 구조와 맞지 않는 프로그래밍 방식** 때문에, 대부분의 사람들이 업무 시에 코드를 통한 데이터 조작보다는 엑셀과 같은 도구에 의존함. 한 연구에 따르면, **‘프로그램 실행 중 프로그램 상태의 시각화 부족이 학생의 프로그래밍 학습에 어려움을 주는 이유 중 하나이다.’** 라고 언급하였음.<sup>1</sup>
- 프로그래밍의 복잡성은 인공지능, 데이터 과학, 시뮬레이션 등의 **고급 분야로의 진입 장벽을 높이며**, 관련 학습에도 부담을 주고 있음.
- 이러한 문제 해결을 위해 Scratch, 엔트리, Blockly 와 같은 블록코딩 플랫폼이나, Azure Machine Learning Studio 등과 같은 GUI 기반 도구들이 나와있으나, 이러한 도구들 대부분은 프로그래밍의 절차적인 측면만을 시각화함. 결과적으로 이들은 **인간의 자연스러운 인식 방식과는 거리가 멀어**, 학생들이 프로그래밍의 본질적인 부분을 이해하는 데 큰 도움이 되지 못함.
- 따라서 학생들은 여전히 프로그래밍을 어려워하며, 이로 인해 **인공지능과**

<sup>1</sup> Ali, Mohamad Bilal and Ahmad Termizi Bin Zamzuri. "Difficulties In Learning Programming:views Of Students." (2012).

같은 기술 분야에서의 학습 역시 어려움을 겪고 있음.

○ 기획 의도(문제해결)

- PyViz 는 복잡한 프로그래밍 언어 대신 객체 중심으로 설계된 시각적인 인터페이스를 사용하여 데이터의 변화와 조작을 직관적으로 보여주며, 이를 통해 사용자들이 **데이터를 중심으로 인공지능 모델의 작동 원리를 자연스럽게 이해** 할 수 있도록 하고자 함.

□ 프로젝트 개요

○ 프로젝트 소개

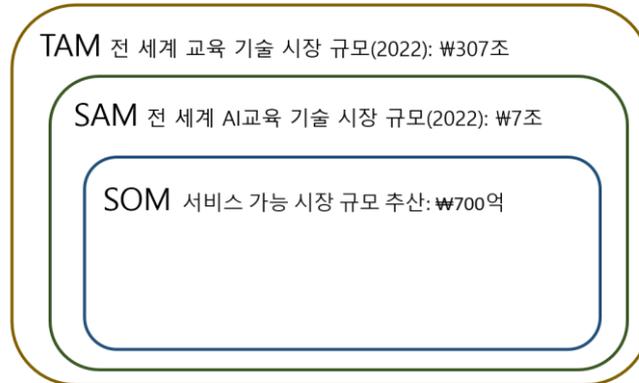
- PyViz 는 코딩 기술이 없는 사용자들도 **인공지능과 데이터 과학을 쉽게 배울 수 있도록 설계된 교육 도구**임.
- PyViz 는 복잡한 프로그래밍 언어 대신 **직관적인 드래그 앤 드롭 인터페이스**를 사용하여, 데이터의 변화와 인공지능 모델의 작동을 시각적으로 보여주는 것이 특징임.
- 사용자는 **실시간으로 데이터를 조작**하고, 그 **결과를 바로 확인**할 수 있어, 데이터 과학과 인공지능의 핵심 개념을 더욱 쉽고 빠르게 이해할 수 있음.
- 또한 **파이썬 친화적인 PyX 프레임워크** 위에 구축되어 있어, 기존 Python 라이브러리와 모듈을 쉽게 통합하고 활용할 수 있음. 이는 개발자가 PyViz 에 새로운 기능과 도구를 추가할 때 **기존의 코드를 재사용하고 확장할 수 있는 유연성을** 제공함.

○ 주요 기능

- **파이썬 객체 시각화**: 파이썬 객체를 HTML 요소로 보여줌. 이를 통해 사용자가 데이터 구조를 직관적으로 인식하도록 도움.
- **드래그 앤 드롭 데이터 조작**: 파이썬 객체를 드래그 앤 드롭으로 대입할 수 있음. 이는 데이터의 상태를 직관적으로 접근할 수 있게 함.
- **인공지능 설계 및 학습**: 설계한 모델을 버튼을 눌러 학습시킬 수 있는 도구를 제공함.
- **확장 틀 제공**: 개발자가 Python 코드를 작성하여 새로운 컴포넌트를 쉽게 작성할 수 있는 프레임워크(PyX)를 제공함. 전처리 라이브러리 등을

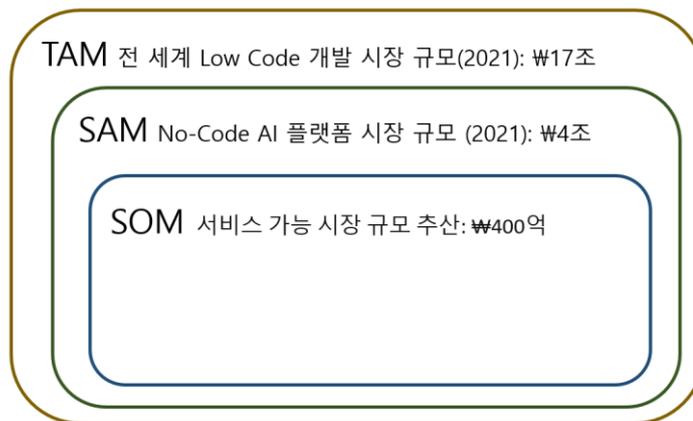
Python 만을 이용하여 PyViz 사이트에 호환되게 개발할 수 있음.

- 시장/소비자 동향/분석, 경쟁 제품 비교/차별화 요소 등
  - 교육 관련 시장 분석



전 세계 교육 기술 시장은 2022년 기준 한화 307조원으로 매우 크며<sup>2</sup>, AI 교육 기술 시장도 한화 약 7조원임<sup>2</sup>. 또한 이 분야는 매우 빠르게 성장하고 있어, TAM이 2030년까지 1100조 규모로 성장할 것으로 예상됨<sup>3</sup>.

- Low-Code 툴 관련 시장 분석



전 세계 Low Code 개발 시장은 한화 17조원이며<sup>4</sup>, No-Code AI 플랫폼 시장 규모는 한화 약 4조원임<sup>5</sup>. 이 분야 또한 매우 빠르게 성장하고 있으며,

<sup>2</sup> Yahoo finance - Global Education Technology Market Report 2022 <https://finance.yahoo.com/news/global-education-technology-market-report-135300840.html?guccounter=1#:~:text=The%20global%20education%20technology%20market,With%20the%20development%20of%20technologies>

<sup>3</sup> Yahoo finance - AI in Education Market Size In 2023 <https://www.linkedin.com/pulse/ai-education-market-size-2023-share-trends>

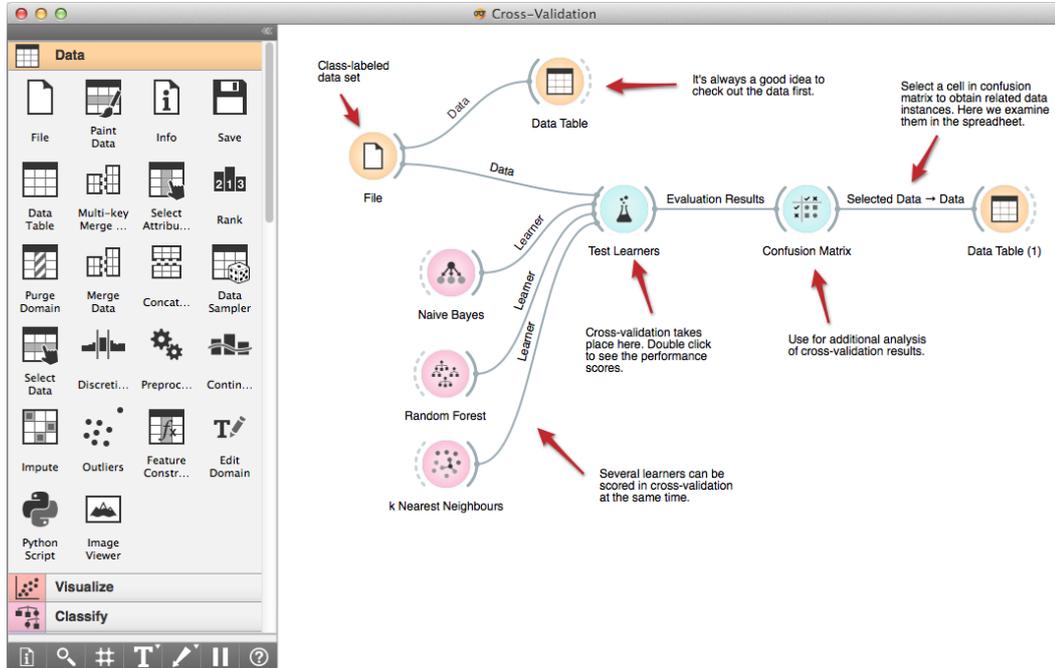
<sup>4</sup> Gartner-Gartner Forecasts Worldwide Low - Code Development Technologies Market <https://www.gartner.com/en/newsroom/press-releases/2021-02-15-gartner-forecasts-worldwide-low-code-development-technologies-market-to-grow-23-percent-in-2021#:~:text=Low,9%20billion%20in%202030>

<sup>5</sup> Grand View Research - No-code AI Platform Market Size & Trends <https://www.grandviewresearch.com/industry-analysis/no-code-ai-platform-market-report#:~:text=The%20global%20no,need%20to%20implement%20AI%20projects>

연 평균 성장률(CAGR) 약 27.3%로 2030년까지 성장할 것으로 예상됨.

– 경쟁 제품 비교/차별화 요소

1) Orange Data Mining<sup>6</sup>

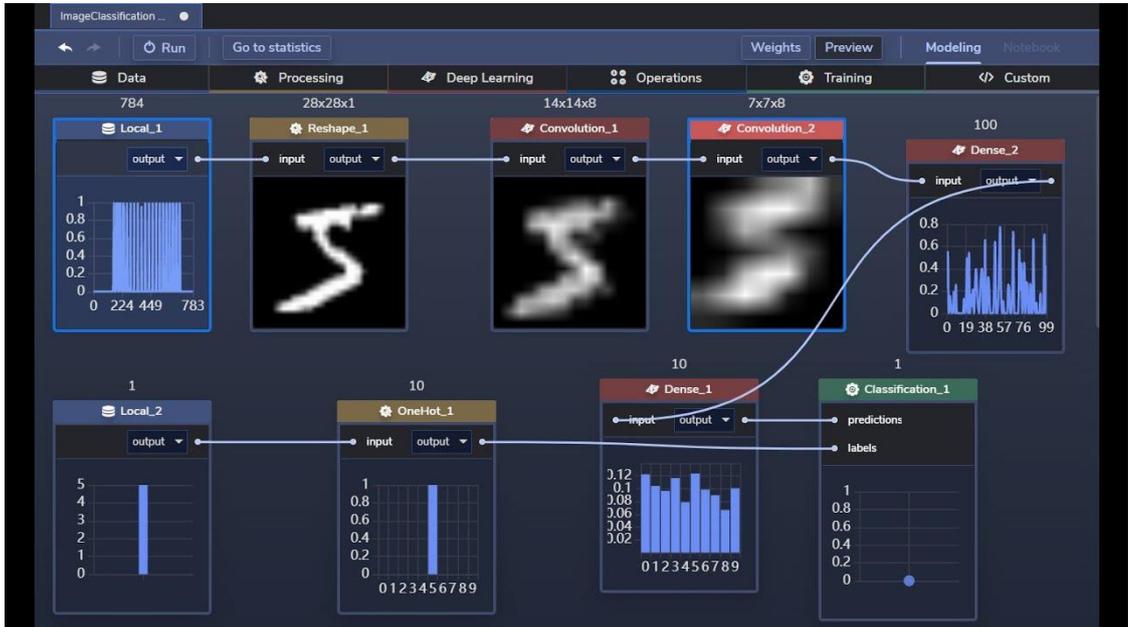


Orange Data Mining 은 데이터 시각화, 머신러닝, 데이터 마이닝, 데이터 분석을 위한 구성 요소를 중심으로 개발된 오픈 소스 시각 프로그래밍 소프트웨어 패키지로, **사용자 친화적인 그래픽 인터페이스**를 통해 **코딩보다는 탐색적 데이터 분석에 집중**할 수 있게 해주며, 빠른 프로토타이핑과 정성적 분석을 위한 인터랙티브한 데이터 탐색 기능을 제공함.

하지만 Orange Data Mining 은 절차를 시각화하는 자유도가 낮은 틀임. 데이터의 흐름은 노드 간의 선으로 표현되며, 노드와 독립적인 데이터는 볼 수 없음. **고로 프로그래밍이 데이터의 흐름이라는 개념이 부족한 비전문가에게는 크게 도움이 되지 않는 방식**일 수 있음. 이에 반해 PyViz 는 데이터를 시각화하여, '절차'를 노드로 하는 비직관적인 방식 대신 '객체'를 노드로 하도록 설계됨.

<sup>6</sup> Orange Data Mining, <https://orangedatamining.com/#:~:text=,prototyping%20of%20a%20data>

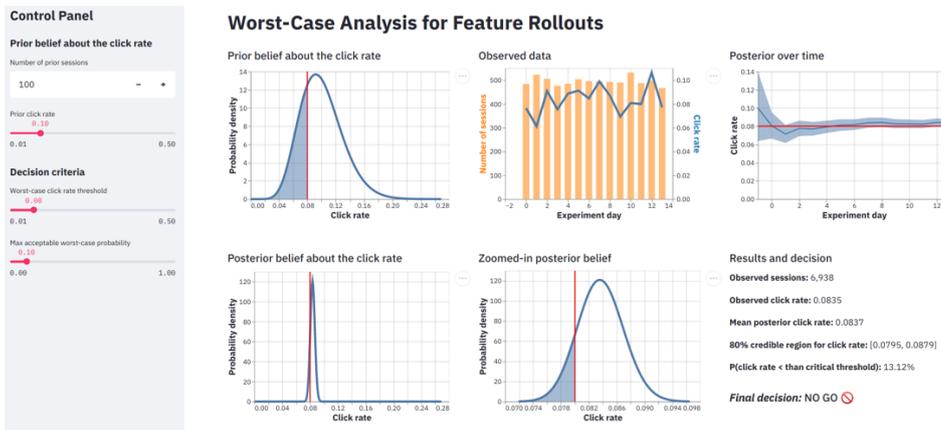
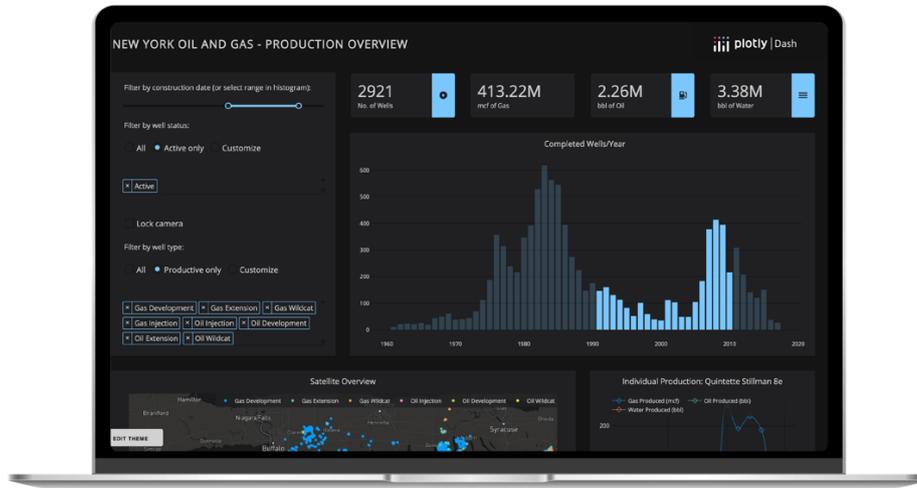
## 2) Perceptilabs<sup>7</sup>



Perceptilabs 는 **순수 코드의 유연성과 드래그 앤 드롭 사용자 인터페이스의 편리함을 결합한** 새로운 시각적 접근 방식을 제공하는 **머신러닝 모델링 도구**임. 이 도구는 컴포넌트 연결, 모델 아키텍처 생성, 설정 및 하이퍼파라미터 조정에 있어 일부 자동화 기능을 포함하고 있음. Perceptilabs 는 모델 디자인을 위한 모델링 도구와 모델 허브를 포함하여, 사용자에게 데이터 과학과 머신러닝 작업을 위한 시각적인 환경을 제공함. Perceptilabs 는 **데이터를 시각화 하는 적절한 접근**을 통해 사용자에게 데이터의 상태에 대한 직관적인 접근을 가능하게 하였으나, 여전히 **데이터를 객체로 다루지는 않고 있으며 자유도가 낮음**. 이에 비해 PyViz 는 자유도 높은 조작을 통해 인공지능 분야에 국한되지 않은 직관을 제공함.

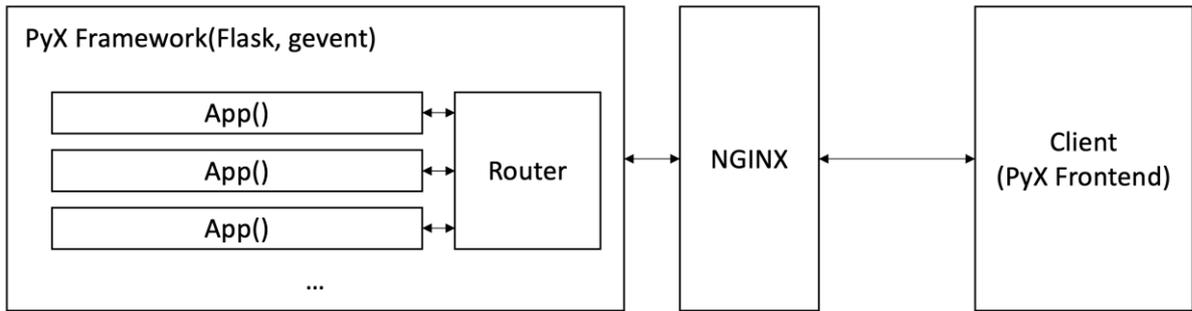
<sup>7</sup> Perceptilabs, <https://www.perceptilabs.com/>

### 3) Dash, Streamlit

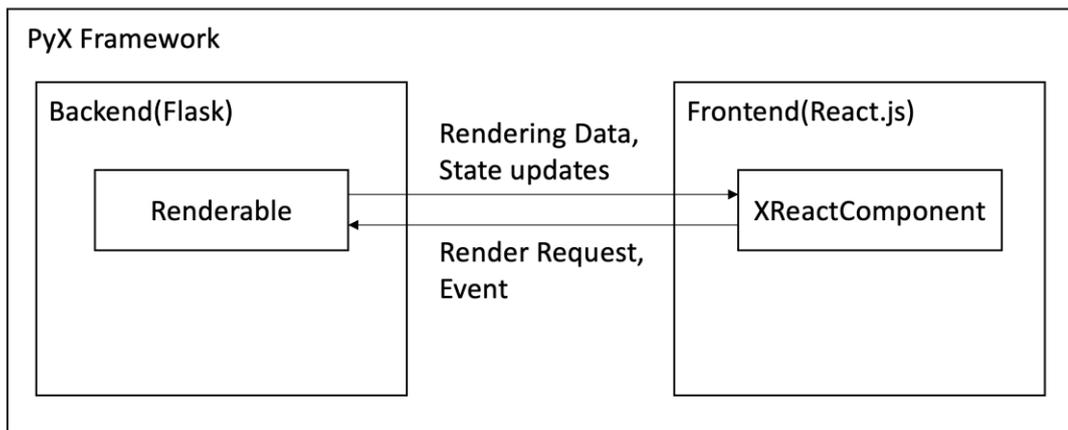


Dash 와 Streamlit 은 인공지능 교육과 큰 상관은 없지만, PyX 프레임워크처럼 파이썬으로 빠르게 웹 프로그램을 작성할 수 있다는 점에서 유사함. 또한 머신러닝 등의 분야나 데이터 분야의 전문가들이 파이썬에 익숙하며 웹 관련 지식이 부족하다는 것을 고려할 때, 대쉬보드를 웹 기반 지식 없이 만들 수 있다는 점에서 이점이 있음. 다만 둘 다 데이터를 시각화하는 데에 집중되어 있어서, PyX 와 같이 일반적인 목적으로 사용하기에는 적합하지 않음. 또한 개발자들이 Dash, Streamlit 에 특화된 문법을 배워야 한다는 단점이 있음.

○ 시스템 구성도



- **PyX 프레임워크:** Python Renderable 객체를 호스팅하는 프레임워크. 사용자가 새로운 방(Room)에 접속할 때, App() 객체를 새로 생성함. 같은 방에 접속한 유저끼리는 실시간으로 같은 화면을 볼 수 있음.
- **NGINX:** Let's Encrypt 를 이용한 SSL 인증 및 확장성, 성능을 위해 NGINX 를 활용하여 서버를 제공함.
- **React.js:** PyX 에 해당하는 프론트엔드를 제작하기 위해 React.js 를 사용함.



- Python 의 객체 (Renderable)의 렌더링이 요청되면, PyX 프레임워크는 해당 객체의 \_\_render\_\_ 함수를 호출하여 그 구조를 json 형태로 Frontend 에 보냄. 그 객체에 해당하는 XReactComponent 는 이를 그대로 React.createElement 에 넣어서, React 의 최적화 기능을 잃지 않으며 똑같은 문법으로 사용할 수 있음.

○ 개발환경

구분		세부 내용
S/W 개발환경 환경	OS	서버: Ubuntu 22.04 LTS
	개발환경(IDE)	VS Code(code-server)
	개발도구	React.js, Flask
	개발언어	Python, Typescript
프로젝트 관리환경	디바이스	Web
	형상관리	Github, Git
	의사소통관리	Notion
기타(HW 구성장비 등)		Vultr

□ 프로젝트 수행 방안

○ 역할 분담

구분		담당 역할 및 주요 활동
연수생	김창연	팀장, 기획, 통합 개발 및 리서치
	장울	서류 작성, 리서치 및 기획, 백엔드
	양석훈	서류 작성, 리서치 및 기획, 프론트엔드
멘토	김현	팀 관리, 기술 자문
	조재화	프로젝트 관리, 프로젝트 기획 피드백
	정좌연	기술적 지원

○ 추진 일정

기능명	구분	추진 내용	추진 일정(월)					
			6	7	8	9	10	11
PyX Framework	계획	PyX 프레임워크 기획(Renderable)						
	분석	필요 기술 분석						
	설계	PyX 프레임워크 구조 설계						
	개발	개발 및 디버깅						

기능명	구분	추진 내용	추진 일정(월)					
			6	7	8	9	10	11
PyViz Website	분석	시장 분석 및 필요 기술 분석						
	설계	PyViz 사이트 구조 설계						
	개발	PyViz 사이트 개발						
	설계 2	PyX 기반 PyViz 사이트 설계						
	개발 2	PyX 기반 PyViz 사이트 개발						
		PyViz 사이트용 모듈 개발						
	완성	서버 설정 및 완성도						

○ 수행 방법

구분		상세 내용
기획 단계		인터넷을 통한 시장 조사 및 경쟁 서비스 분석.
개발 단계	PyX Framework	PyX Framework 개발을 위해 프론트/백엔드 지식 학습 (React.js, Flask 등) 인터넷 강의 및 인터넷 검색 해당 모듈이 존재하지 않아서 React.js 와 Flask 를 활용해 직접 개발
	PyViz Website	안정적인 서버 제공을 위해 NGINX, WSGI 지식 학습. 인터넷 강의 및 인터넷 검색, 직접 개발.

○ 문제점 및 해결방안

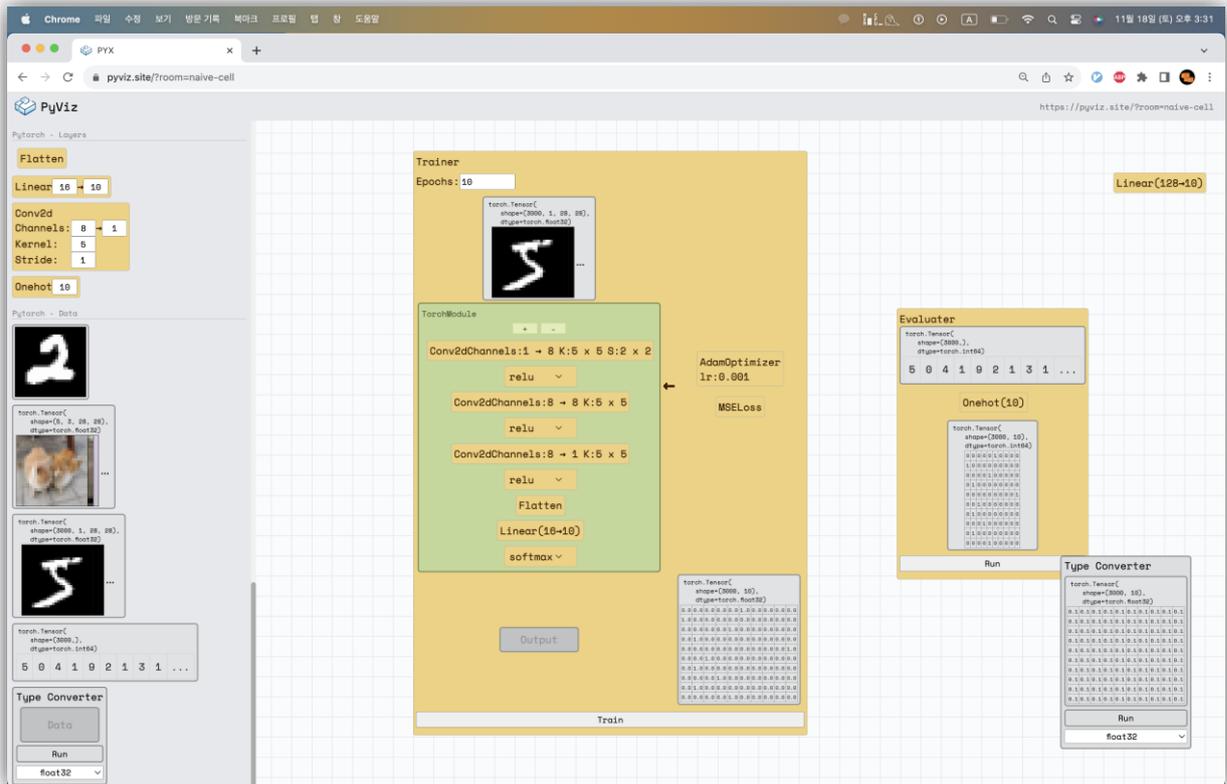
구분	문제점	해결 방안
관리 측면	피봇팅 이후 개발 기간이 부족함	개발 역량 내에서 가능한 범위로 목표 범위를 수정함. 중간평가의 피드백을 반영하여 데이터 시각화 방향에 집중하여 개발.
개발 측면	관련 기술이 부족함	NGINX, React.js 등의 지식을 공부함.
	피봇팅 이후 개발 기간이 부족함	개발을 빠르게 해줄 수 있는 프레임워크를 기획하여 개발하고, 이를 바탕으로 사이트를 개발함.

## □ 개발 진행현황

○ 결과물 링크

– <https://pyviz.site/>

○ 결과물 형태



– PyViz Website

PyViz 페이지를 제공하고 있음. 서비스를 제공할 수 있는 단계로는 아직 개발하지 못하였으나, 시연을 위해 사용자마다 사용할 수 있는 룸을 제공함. 사이트에 접속하면 임의의 룸을 지정해주며, 링크 공유 시 같은 룸에 접속할 수 있음. 모든 조작이 실시간으로 동기화됨.

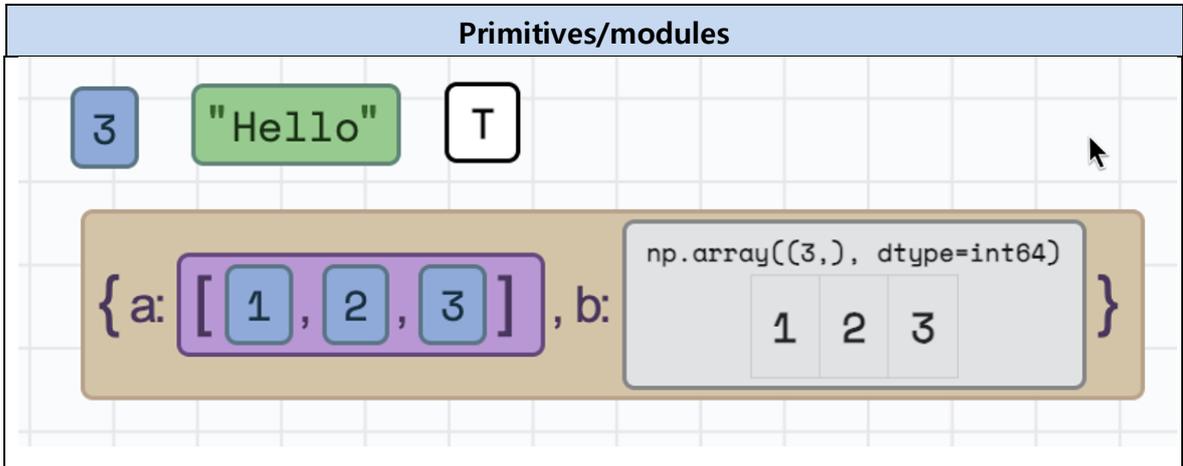
– 인공지능 모듈

PyTorch 모듈의 객체들이 제공됨. 드래그 앤 드롭과 클릭만으로 인공지능 모델을 설계 및 학습할 수 있음.

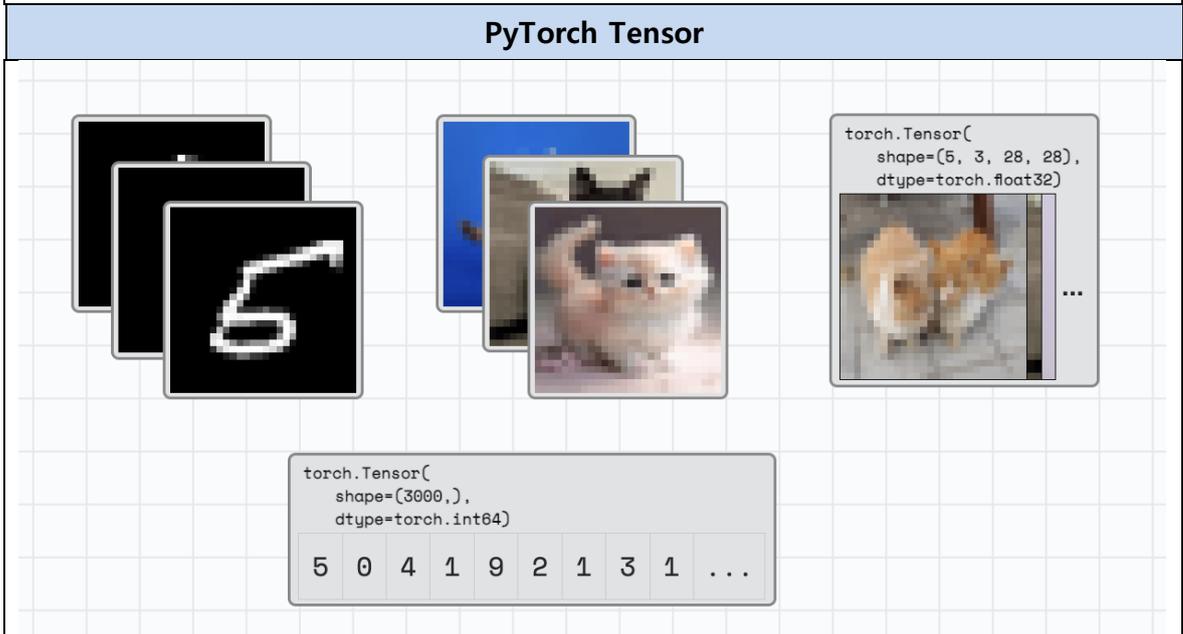
○ 프로젝트 주요 결과물

－ 객체 시각화

다양한 방식으로 객체를 시각화하여 보여줌. PyX 모듈은 데이터에 따라 다른 방식의 렌더링이 가능하며, 상황에 맞는 다양한 방식으로 렌더링함.



위는 다양한 객체를 시각화 한 모습. 숫자형, 문자형 등의 Primitive 와, 리스트, 딕셔너리, 모듈의 객체인 np.array 까지 위와 같이 시각화해서 보여줌.



위는 torch.Tensor 의 시각화된 모습. Tensor 의 모양에 따라 포맷을 자동으로 인식하여 서로 다른 방법으로 렌더링하여 보여줌. 왼쪽부터 흑백 이미지, 컬러 이미지, 다수의 컬러 이미지, 아래는 단순 숫자 데이터임.

**PyTorch Layers**

`Conv2dChannels:3 → 3 K:3 x 3`

`Conv2dChannels:1 → 8 K:5 x 5 S:2 x 2`

레이어 데이터에 따라 다른 방식의 렌더링을 보여주는 모습. 레이어는 개별적으로 다룰 수 있으며, 다른 모듈에 포함되어 Train 되어도 개별적으로 학습되어, 따로 쓸 수 있음. 이는 Auto Encoder 나 Encoder-Decoder 구조를 사용하는 등의 모델에 유용하게 사용할 수 있음.

– 드래그 앤 드롭

드래그 앤 드롭으로 객체를 직접 다룰 수 있음. 객체에 객체를 넣어 데이터를 원하는 대로 구성할 수 있음.

**Assignment**

객체를 드래그 앤 드롭을 통해 멤버 변수에 대입할 수 있음. 위는 [None, 2, 3]이라는 리스트의 0 번째 위치에 "Hello"라는 문자열을 대입하는 모습. 드래그 앤 드롭으로 대입을 하면, 백엔드의 Python 에 바로 적용이 되며 그 모습이 Rerendering 을 통해 프론트엔드에 보여짐.

### Building Model

TorchModule

+ -

Linear(8→64)

relu ▾

Linear(64→64)

relu ▾

Linear(64→1)

sigmoid ▾

TorchModule

+ -

Conv2dChannels:1 → 8 K:5 x 5 S:2 x 2

relu ▾

Conv2dChannels:8 → 8 K:5 x 5

relu ▾

Conv2dChannels:8 → 1 K:5 x 5

relu ▾

Flatten

Linear(16→10)

softmax ▾

왼쪽은 드래그 앤 드롭을 이용해 PyTorch의 sigmoid 레이어를 모델에 넣어서 인공지능 모델을 구성하는 예시임. 오른쪽은 CNN을 활용해 만든, 이미지를 입력받아 Binary Classification을 할 용도로 만들어진 모델임.

– 모델 학습 및 평가

### Evaluation

Evaluator



TorchModule

+ -

Conv2dChannels:1 → 8 K:5 x 5 S:2 x 2

relu ▾

Conv2dChannels:8 → 8 K:5 x 5

relu ▾

Conv2dChannels:8 → 8 K:5 x 5

relu ▾

Flatten

Linear(128→10)

softmax ▾

torch.Tensor(  
shape=(1, 10),  
dtype=torch.float32)

Run

Evaluator



Conv2dChannels:1 → 3 K:5 x 5 S:2 x 2

torch.Tensor(  
shape=(1, 3, 12, 12),  
dtype=torch.float32)



Run

Evaluator 라는 객체를 통해 Evaluation(평가)을 하는 모습. Run 버튼을 누르면, 가운데의 TorchModule에 데이터를 대입한 결과를 아래에 반환함. 왼쪽은 MNIST 예제로, 10개의 라벨에 대한 softmax 값을, 오른쪽은 CNN 하나의 레이어를 통해 처리한 모습임. 결과 값을 이미지로 인식하고 보여주는 모습.

## Train

Trainer 라는 객체를 통해 Train(학습)을 하는 모습. Optimizer 와 Loss, Input, Target 데이터를 넣고 Train 을 누르면 모델이 업데이트 되며, 학습이 끝나면 Output 에 Input 에 대한 평가 결과가 나옴.

## Data Transforming

Grayscale, Downscale, Type Converter 등을 이용해 데이터의 형태를 바꾸어주는 객체임.

## Error displaying

실행 과정에서 생긴 에러를 화면 오른쪽 하단에 표시해줌.

– 추가 모듈 개발 확장성 지원(PyX Framework)

추가 모듈 개발을 쉽게 할 수 있도록 **React** 와 같은 문법을 사용하도록 개발.

아래는 PyViz 사이트의 Activation 블록을 Renderable 로 제작한 예시임.

PyX Framework Example(Activation.x.py)	
<pre> import torch from torch import nn  class Activation(nn.Module):     def __init__(self, activation='relu'):         super().__init__()         self.type = activation         self.activations = {             'relu': torch.sigmoid,             ...         }      def forward(self, x):         return self.activations[self.type](x)      def __render__(self, user):         def changeType(e):             self.type = e['target']['value']         return &lt;div style={{             display: 'block',             width: 'fit-content',             ...         }}&gt;         Activation(             &lt;select value={self.type}                 onChange={changeType}                 style={{                     display: 'block',                     width: 'fit-content',                     ...                 }}&gt;                 &lt;option value='relu'&gt;relu&lt;/option&gt;                 &lt;option value='sigmoid'&gt;sigmoid&lt;/option&gt;             &lt;/select&gt;         )         &lt;/div&gt; </pre>	<p><b>PyTorch</b> 모듈의 정의 방식대로 Activation 모델을 정의함.</p> <p><b>PyTorch</b> 의 <b>nn.Module</b> 을 상속받으며, <b>forward(self, x)</b> 메소드를 구현하여 모델의 평가 과정을 정의함. 확장성을 위해 <b>Renderable</b> 객체를 상속받지 않아도 되도록 구현함.</p> <p><b>__render__(self, user)</b> 를 정의하여 해당 모델을 PyX Framework 가 렌더링 할 수 있는 <b>Renderable</b> 로 만들.</p> <p><b>__render__</b> 함수 구현은 <b>React</b> 와 같은 방식으로 하되, <b>React</b> 에서 사용하는 <b>useState, useEffect</b> 등의 훅을 사용할 필요 없이 렌더링 과정에서 사용되는 멤버 변수의 변화를 <b>자동으로 인식</b>하여 프론트엔드와 동기화 해줌. 고로 같은 객체를 렌더링하는 <b>모든 클라이언트에 동일하게 적용</b>됨.</p> <p><b>onChange, className</b> 등 리액트에 특화된 문법을 동일하게 지원함. 내부적으로 해당 데이터를 프론트엔드의 <b>React</b> 프레임워크에 그대로 전달하도록 구현되어 있음.</p>

- PyX 파일의 확장자는 개발 시 Python 과의 호환성을 위해 "\*.x.py" 이며, 일정상 트랜스파일러는 GPT-3.5 API 를 활용함.

Transpiler Prompt
<p>You are now a transpiler that transpiles XReact to Python.            XReact is a library that allows you to write React-like code in Python.            The code below is a code snippet from XReact.pyx that you need to transpile to Python.            You should NOT generate any other text other than the resulting Python code.</p> <p>Example:</p> <p>[XReact.pyx]            &lt;div style={{width: '100%'}} onClick={clickHandler}&gt;{user.a}&lt;/div&gt;</p> <p>[XReact.py]            createElement("div", {"style": {"width": '100%'}, "onClick": clickHandler}, user.a)</p> <p># Path: xreact.pyx</p> <p>{ original code }</p>
Main.py
<pre>import pyx from components.App import App  host(App(), port=80)</pre>

- GPT 가 트랜스파일링을 잘 할 수 있도록 예시를 주어 One-Shot 방식으로 진행함. 약 75%의 확률로 정확하게 트랜스파일링을 진행하였음.
- 추후에 Parser 등을 활용하여 정확한 트랜스파일러를 작성할 예정.

## □ 결과물 활용 방안 및 기대효과

### ○ 결과물 활용 방안

– **교육 서비스 제공:** PyViz 를 활용하여 온라인 **교육 플랫폼**을 구축하고, 데이터 과학 및 인공지능 과정을 제공하여 학생과 전문가를 대상으로 수익을 창출할 수 있음. 아직 사이트가 서비스 가능한 단계로 완성되지 않아서 서비스를 시작하거나 홍보하지는 못하였지만, 추후에 할 진행할 예정임.

또한 중간평가 때의 의견을 반영하여 B2B **교육도구 및 업무 보조 도구**로의 개발도 염두에 두고 있음.

– **인공지능 연구에 활용:** 인공지능 연구시에 해당 프레임워크가 필요한 경우, 이를 활용해 효과적인 연구 및 시연을 할 수 있을 것으로 예상함.

– **PyX 프레임워크 오픈소스화:** **PyX 프레임워크를 오픈소스로 전환**하여, 전 세계 개발자 및 사용자 커뮤니티의 기여를 받을 계획임. 이를 통해 프레임워크의 기능 향상, 최적화, 버그 수정 등을 지속적으로 수행할 수 있을 것으로 기대함.

PyX 프레임워크는 확장성 및 호환성이 매우 좋으며 기존 Python & React 개발자들이 진입장벽이 거의 없이 다룰 수 있으므로, 이를 활용한 다양한 틀이 나올 것으로 기대함.

– **논문 등재:** PyViz 의 교육적 효과를 연구하고 정리하여 학술 논문으로 발표함으로써 기술의 신뢰성을 강화할 계획임.

### ○ 기대효과

<b>사용자 측면</b>	PyViz	객체를 중심으로 시각화하여, 사용자가 자연스러운 방식으로 객체를 인식하고 다룸으로써 인공지능을 쉽고 직관적으로 배울 수 있을 것으로 기대함.
		복잡한 코드를 사용하는 대신 드래그 앤 드롭으로 모델을 만들고 학습을 시킴으로써 인공지능에 대한 진입장벽을 하나 줄일 수 있을 것으로 기대함.
		다양한 구조를 수정하며 테스트해보는 인공지능 연구의 특성상, 간단한 조작으로 여러가지 실험을 할 수 있어 유용할 것으로 기대함.
PyX	PyViz 의 추가 모듈을 개발할 때, React 또는 HTML 경력이 있는 사용자는 쉽게 화면을 커스텀하여 사용할 수 있음. React 보다 간단한	

	<p>문법으로 개발할 수 있어, PyViz 콘텐츠 개발 진입장벽이 낮아짐.</p> <p>머신러닝, 딥러닝 등을 할 때 커스텀 로깅 사이트 및 대쉬보드를 빠르게 직접 작성할 수 있음. 파이썬의 print 문에 의존하는 것보다 양방향성이며 편리함.</p> <p>기존의 프로젝트 또는 모듈의 객체에 <code>_render_</code> 함수만 추가함으로써 PyViz에 적합한 형태로 개발할 수 있어 확장성이 좋음.</p>
비즈니스 측면	쉬운 방식의 인공지능 교육을 제공함으로써, Python, 코딩 등의 진입장벽에 막혀 인공지능 교육을 접하지 못하였던 사용자로 대상을 확장할 수 있음.
	PyViz와 같은 도구를 내부적으로 사용함으로써 연구 및 개발 과정에서 발생할 수 있는 시간과 비용을 절감할 수 있음.

### □ 담당멘토 의견

멘토	멘토 의견
김현	- <b>우수한 점</b> : 본 과제 결과물인 PyViz 를 통해 코딩 기술이 없는 사용자들도 인공지능과 데이터 과학을 쉽게 배울 수 있도록 설계된 것이 무척 우수함. 직관적으로 드래그 앤 드롭만으로 컴퓨터 내의 데이터의 변화와 인공지능 모델의 작동을 시각적으로 보여주는 것이 차별성이 있음
인공지능, 기획	- <b>기타 의견</b> : 실시간으로 데이터 변화를 바로 확인할 수 있어, 향후 데이터 과학과 인공지능 교육에 확대 적용한다면 핵심 개념을 더욱 쉽고 빠르게 이해할 수 있을 것으로 사료됨
조재화	- <b>우수한 점</b> : 중간에 AI 프로젝트를 어쩔 수 없이 변경하는 어려움이 있었지만 남은 기간 의미있는 프로젝트를 마무리하기 위해 노력함
프로젝트 관리 / 기획 피드백	- <b>기타 의견</b> : 개발 소요시간이 부족하여 실제 제품에 대한 고객반응을 보지 못했는데 기회가 된다면 고객 피드백을 통해 더 완성도 높은 프로젝트 되었으면 좋겠습니다.
정좌연	- <b>우수한 점</b> : 기존의 코딩 툴 대비 UX 측면에서의 편의성을 제공하고 있음. 드래그 앤 드롭 중심의 모델 학습으로 AI 접근을 좀 더 수월하게 진행할 수 있음.
프로젝트 관리 / 인공지능	- <b>기타 의견</b> : 실제 시나리오 위주의 다양한 테스트를 통해서 기능의 점검과 개선이 필요함

### □ 중간점검 주요 의견 반영 내역

중간점검 의견	반영 결과
- 피봇팅을 한 과제인 만큼 <b>개발 기간도</b> 짧고 진행 상황도 빠르지 않으니, ... <b>중요</b>	- 중간점검 피드백의 개발 기간 문제를 고려하여, 목표를 ' <b>코딩 교육 콘텐츠 개발</b> '에서 ' <b>인공지능 교육</b>

중간점검 의견	반영 결과
<p><b>주제에 집중</b>하여 개발하는 것이 좋겠음</p> <ul style="list-style-type: none"> <li>- 짧은 시간에 데이터 시각화와 교육 콘텐츠를 모두 설정하는 것보다 <b>하나의 목표를 설정하여 추진</b>하는 것을 추천함</li> <li>- ... 중간에 <b>가교 역할을 하는 템플릿을 만들고, 템플릿을 활용하여 교육 콘텐츠를 생성하는 부분을 위임</b>하여 템플릿을 개발하는 부분을 개발자가 하도록 하는 역할 분담을 고려할 필요가 있음.</li> <li>- <b>B2B 시장의 교육 교재</b>로 활용하는 방법도 좋은 사업 모델로 보임.</li> </ul>	<p><b>도구 개발'</b> 로 줄임.</p> <ul style="list-style-type: none"> <li>- '가교 역할을 하는 템플릿을 만들라'는 조언을 받아들여, <b>PyX 프레임워크 및 사이트의 확장성</b>에 신경써서 개발함. <b>React 와 구조가 호환</b>되도록 만들어, React 개발 경험이 있는 사람은 어려움 없이 추가 모듈을 만들 수 있도록 개발함.</li> </ul>
<ul style="list-style-type: none"> <li>- ... 현재의 개발화면 UI/UX 로 판단해 보면 <b>데이터 시각화 기능도 부족</b>하고, 이를 통한 <b>교육적인 효과도 미흡</b>한 부분이 있음.</li> <li>- ... 현재의 구성도 화면은 <b>직관성이 부족</b>함.</li> </ul>	<ul style="list-style-type: none"> <li>- 직관성이 부족하다는 피드백을 반영하여, <b>더 직관적인 렌더링 방식</b>(이미지 등)을 추가함.</li> </ul>
<ul style="list-style-type: none"> <li>- 현재 진행된 내용만으로는 <b>단순 데이터의 프린트문 수준의 내용만을 보여주고 있어</b> 실제 의도했던 <b>코드의 시각화 및 교육용 코딩 툴로서의 활용성</b>은 최종점검에서 검증이 필요함</li> <li>- 데이터 시각화를 기반으로 한 블록 코딩보다는 <b>디버거 Watch 기능</b>으로 보여져, ...</li> <li>- 블록 코딩이 아니면 <b>딥러닝을 위한 중간 결과 모니터링 솔루션</b>으로 발전 가능성도 있어 보임.</li> <li>- 코딩 교육 콘텐츠 제작을 목표로 하는 것보다 <b>데이터 시각화에 보다 더 집중</b>하여 진행하는 것을 추천함.</li> </ul>	<ul style="list-style-type: none"> <li>- 데이터 시각화로 발전의 가능성을 고려해보라는 피드백을 반영하여, 해당 기능인 PyX의 개발에 더욱 집중하였으며, 이후 PyViz는 <b>데이터 시각화 및 딥러닝을 위한 중간 결과 모니터링</b> 기능을 위주로 한 딥러닝 교육 솔루션으로 방향을 바꿈.</li> <li>- 단순 데이터의 프린트문 수준이라는 피드백을 반영하여, <b>화면에서 백엔드의 함수를 부를 수 있는 기능</b>을 추가함.</li> </ul>