* 데이터 기반 제품 개선(Product Science)이란?
* 데이터 과학자: 기계 학습의 형태로 사용자들의 경험을 개선

문제에 맞춰 가설 수립 > 데이터 수집 > 예측 모델 설계 > 테스트

짧은 사이클로 단순하게 시작해서 점차 고도화하는 방식으로 개발. 테스트는 A/B 테스트를 수행

* 데이터 과학자(Data Scientist)에게 필요한 Skillset

ML / AI에 대한 깊은 지식과 경험

Python, SQL을 이용해 코드를 짤 수 있는 능력

수학적, 통계학적 지식

끈기와 열정

* 데이터 기반 제품 개선에 있어서 데이터 과학자가 유의해야 할 점

1) AI, 딥러닝이 모든 문제의 해답이 아님을 명심해야 한다. 기계 학습이 더 나을 때도 있다.

2) 지표에 기반해, 모델의 목표와 측정 방법을 과학적으로 접근해야 한다.

3) 가장 중요한 것은 모델링을 위한 데이터가 존재하는지 여부

* 기계 학습 모델 개발 과정

문제 정의 > 훈련 데이터 수집 > 모델 설계, 테스트 > 모델 배포 > A/B테스트 >

(테스트 결과가 좋으면) 출시, (그렇지 못하면) 결과 분석 후 데이터 재수집

* 예시1. 개인화된 추천 엔진

인터넷 교육 업체가 규칙 기반에서 기계 학습 기반으로 추천 엔진 전환

* 예시2. 사기 결제 감지

실제 사례를 수집하고 이상값 탐지 실행, 사람이 검토해 패턴 발견

* 예시3. 환자의 이상 징후 예측

원격 환자 모니터링 – 환자의 다양한 측정 데이터를 기반으로 치료가 필요한 상황인지 예측

환자가 입원하거나 응급실까지 가는 사태가 발생하기 전에 적절한 치료를 받도록 하는 것이 목표

* 예시4. 의료 이미지 분석

딥러닝 알고리즘이 MRI, X-ray 이미지 분석에서 인간을 능가.

VoxelMorph 오픈소스 프레임워크의 경우 몇 초만에 MRI 이미지를 분석.

하지만 잘못된 결과를 내놓았을 때 책임 소재를 가리기 힘들다는 한계가 존재

* 기계 학습(Machine Learning)이란?
* A field of Study that gives computers the ability to learn without being explicitly programmed (컴퓨터에게 명시적인 프로그래밍 없이도 학습할 수 있는 능력을 주는 분야의 학문)

데이터의 패턴을 보고 **흉내**내는 방식. 컴퓨터가 학습할 수 있도록 알고리즘과 기술을 개발

딥 러닝은 기계 학습의 일부: 컴퓨터비젼, 자연어처리 등

* 기계 학습 모델

특정 방식의 예측을 수행하는 Blackbox(동작 메커니즘을 직관적으로 알기 힘듦). 선택한 알고리즘에 따라 내부가 달라짐.

일반적으로 입력한 데이터를 기반으로 예측하는 Supervised Learning (지도학습)

ML Model Training/Building

* 지도 기계 학습(Supervised Machine Learning)

명시적 예제 – Training Set – 을 통해 학습함. 정답이 존재하는 문제

분류(Classification), 회귀(Regression)

* 비지도 기계 학습(Unsupervised Machine Learning)

주어진 데이터를 몇 개의 그룹으로 나누기.

군집(Clustering), 집단화(Grouping),

**Semi-supervised ML**: GPT 같은 언어 모델의 훈련

* 강화 학습(Reinforcement Learning)

게임과 같이 일정한 규칙 내에서 어떻게 하면 이기는지를 학습

자율주행 모델을 학습시킬 때 사용

* ML 모델 개발 시 고려할 점
* 데이터 과학자와 데이터 엔지니어 사이의 마찰이 생기는 지점

데이터 과학자가 개발한 모델을 데이터 엔지니어에게 이양할 때

많은 데이터 과학자들은 R을 비롯한 다양한 도구로 모델을 개발

하지만 실제 환경은 다양한 모델을 지원하지 못함

* 모델 개발부터 출시까지 총괄적으로 책임질 사람이 필요

이 과정에 참여하는 모든 사람들이 기여도를 인정받아야 협업이 쉬워

데이터 과학자와 데이터 엔지니어들 사이 소통이 수시로 이루어지는 것이 중요

* 모델 개발 초기부터 개발 및 출시 과정을 구체화하고 소통해야

모델의 검증 방법, 모델의 형태과 규격, A/B 테스트상에서 성공 판단 지표 등에 대한 논의가 필요

* 개발된 모델을 바로 출시할 수 있는 프레임워크의 필요

예시) R로 개발된 모델은 바로 제품 출시가 불가능하기에 Python 라이브러리를 이용

AWS SageMaker 등 ML 전반에 걸친 개발/배포 프레임워크의 사용

ML 모델 개발, 검증, 배포까지 하나의 프레임워크에서 수행할 수 있어

Google Cloud, Microsoft Azure도 비슷한 프레임워크를 지원하며 많은 IT 기업들이 자체 ML 개발/배포 프레임워크를 개발해 사용하고 있다.

MLflow나 KubeFlow 등의 오픈 소스 프레임워크를 사용하기도 함

* 모델 출시는 시작일 뿐, 운영 과정에서 점진적으로 개선되는 것이 중요

운영 과정에서 생기는 데이터, 통찰을 통해 개선점을 만들어 나가는 피드백 루프가 필요하다.

주기적으로 모델을 재구성하고 배포해야 한다.

Continuous Model Update and Monitoring

이로 인해 탄생한 직군이 **MLOps**

* MLOps란?
* 모델 빌드, 배포, 모니터링 전체 프로세스를 자동화하는 직군
* Data Drift로 인한 모델 성능 저하를 방지

**시간이 흐르며 훈련 데이터와 실제 환경 데이터가 다르게 변화**함. 이를 Data Drift(데이터 표류)라고 부르며 관찰, 감시하는 것이 중요. 즉, **주기적으로 기계 학습 모델을 다시 빌드해야** 함

* MLOps vs DevOps

DevOps: Deliver software faster and more reliably in automated fashion

개발자가 만든 코드를 시스템에 반영(Continuous Integration/Continuous Delivery)[[1]](#footnote-1)

시스템 동작 모니터링 및 이슈 감지 시 Escalation Process[[2]](#footnote-2) 수행

On-call Process

MLOps: Deliver ML models faster and more reliably in automated fashion

기계 학습 모델을 시스템에 반영하고 모니터링 및 이슈 감지

모델 구동 환경과 성능 저하를 모니터링하고 필요시 Escalation Process 진행

모델을 계속적으로 빌드, 배포, 성능 모니터링 – Continuous Training

Latency 중요, Data Drift 측정

* MLOps Engineer가 알아야 하는 기술

데이터 엔지니어로서 알아야 하는 기술:

Python, Scala, Java

Data Pipeline, Data Warehouse

DevOps 엔지니어로서 알아야 하는 기술:

CI/CD, 서비스 모니터링, 컨테이너 기술(K8S, Docker),

Cloud(AWS, GCP, Azure)

Infrastructure As Code (Configuration As Code)

Machine Learning 관련 경험과 지식:

ML 모델 빌드와 배포, 프레임워크(SageMaker, Kubeflow, MLflow 등) 경험

* 기계 학습 사용 시 고려할 점: Bias, ML Explainability, Data Rights
* 데이터 기반 AI는 완벽한가?

Training Set의 품질은 어떤가? Data의 양도 중요하지만 품질도 중요. GIGO.

그럴듯한 시스템에서 내려받은 Data라고 해서 절대적으로 신뢰하면 안돼

* AI도입 시 가능한 문제를 어떻게 해결할 것인가?

알고리즘의 작동 원리를 설명할 수 있는가? 알고리즘 자체에 편향성이 있지는 않는가?

* EU Trustworthy AI Guideline

AI에 대한 인간의 감독 / AI 모델의 견고성과 안정성

개인정보 보호 및 Data Governance / 알고리즘 투명성

다양성, 비차별성 및 공정성 / 사회친화적 모델, 환경친화적 모델

AI로 인한 문제 발생 시 책임 소재 명확화

* 잘못된 개인정보 보존으로 인한 패널티를 고려해야

**미국 HIPAA(Health Insurance Portability and Accountability Act)**

개인 식별 정보 18개 를 보호하려는 목적으로 1996년 발표

(이름, 주소, 생년월일, 전화번호, 이메일 주소, 사회보장번호, 라이선스번호, IP주소 등)

**유럽연합 GDPR(General Data Protection Regulation)**

불필요한 개인 정보 저장 금지, 불가피하게 개인 정보를 저장할 경우 암호화 필수.

개인 정보는 사람들의 것이지 보관하고 있는 기업의 소유가 아니다.

개인들의 허가 없이 개인 정보를 사용할 수 없게 명확화함.

* 집단 이기주의로 인한 반발이 있을 수 있다는 점 고려해야

예시. 지난 COVID-19 팬데믹 때 한시 허용되었던 원격진료가 의사집단의 반발로 표류

AI는 의사를 대체하기보다 의사의 효율성과 진단/치료의 정확성을 높이는 보조적 역할.

경험 많은 의사라 해도 실수를 할 수 있고, 의사마다 진료 편차가 다름.

AI는 진단 절차를 체계화하고 작업을 빠르고 정확하게 하는 데 사용할 수 있음.

의사들도 Data Literacy 교육을 받고, 또 환자와의 진료 및 대화, 공감에 더 시간을 할애해야

* 생성형 AI(Generative AI, Gen AI)란?
* 생성형 AI는 학습된 콘텐츠를 바탕으로 새로운 콘텐츠를 만드는 딥 러닝 기술[[3]](#footnote-3)
* 프롬프트의 내용을 바탕으로 대답을 출력하거나 새로운 콘텐츠를 생성함
* Deep Learning Model Type

Discriminative

분류/예측 수행. Label이 존재하는 데이터에 적용. Supervised Learning

Feature와 Label 간의 관계를 학습

**Generative**

훈련된 데이터와 비슷하지만 새로운 데이터를 생성 (훈련된 데이터의 통계적 특성을 이해)

Unsupervised Learning

* Gen AI 모델과 일반 ML 모델의 동작 방식 비교



일반 ML 모델에서 y는 보통 숫자, 카테고리, 확률 등을 의미

Gen AI 모델에서 y는 보통 자연어 문장, 이미지, 오디오 데이터 등을 의미

* Gen AI Foundation Model

광범위한 데이터 세트에 대해 학습된 대규모 기계 학습 모델

**Pre-trained, Unsupervised Learning/Self-supervised Learning**

**대용량 데이터로 학습**하기에 많은 시간과 비용, 인력이 필요

Transformer Model Architecture 사용 (⟪Attention is All You Need⟫ 논문에서 처음 발표됨)

특별한 학습 없이 다양한 작업에 적용할 수 있다는 특징.

GPT-3, GPT-4, BERT, T5, DALL-E 등의 모델이 있다.

* Gen AI Fine-Tuning

Foundation Model을 원하는 방향으로 동작하도록 특정 지식을 학습시켜 맞춤형으로 만드는 재훈련

* Hallucinations(환각) 현상

모델이 부정확하거나, 무의미하거나, 완전히 조작된 정보를 생성하는 경우가 있다.

따라서 Gen AI를 사용할 때는 항상 Fact-Checking이 중요하다.

**이런 현상이 발생하는 이유**는:

훈련 데이터의 불충분, 훈련 데이터의 최신성 부족, 훈련 데이터의 품질 이슈, 모델이 프롬프트로 충분한 Context를 받을 수 없는 경우

* Multimodal Foundation Model
* ChatGPT의 발전 과정
* GPT (Generative Pre-trained Transformer)

OpenAI에서 만든 LLM(초거대 언어 모델)

초기에는 Word Completion, Code Completion 두 가지 모델을 제공함

* GPT-3 vs GPT-4

|  |  |
| --- | --- |
| **GPT-3(2020)** | **GPT-4(2023)** |
| 1750억 개의 매개변수 (약 800GB) | 1조 개의 매개변수 |
| Context Window[[4]](#footnote-4) 2,048 + 1 | Context Window 8,192 + 1 |
| Word Vector 12,288개 | Word Vector 32,768개 |
|  | Multi-modal (이미지 인식 가능) |

GPT-3은 NVIDIA의 A100으로, GPT-4는 NVIDIA의 H100이 사용되었다.

* GPT-4 Turbo(2023.11)

Context Window가 1억 2800만 개로 확장되어 모델의 정확도가 개선되었다.

API 기능 역시 개선되어 JSON 모드와 시드 제어, 다수 함수 동시 호출이 가능해졌다.

RAG[[5]](#footnote-5) 기능을 통해 외부 문서나 데이터베이스를 가져올 수 있게 되었다.

기존 2021년 9월까지의 정보만을 담고 있던 것을 넘어 2023년 4월까지로 갱신되었다.

* 경량화 언어 모델 예시

**Meta의 LLaMA (2024.9 v3.2 공개됨) – 라이트 모델과 멀티모달 모델 두 가지로 출시됨**

Stanford University의 Alpaca – Meta LLaMA의 Fine-tuned 버전

Databricks의 Dolly

(이러한 모델들은 초거대 언어 모델에 비해 학습과 추론이 훨씬 더 빠르고 쉽다)

* RLHF (Reinforcement Learning from Human Feedback)
* ChatGPT를 쓰기 위한 좋은 프롬프트

**Act as a [Role], [Task], in a [Format] using a [Tone]. The main objective is to [Objective], don’t use [Restriction].**

예시) Your new **role** is article writer and your **task** is to write an introduction. Your main **objective** is to explain simply what ChatGPT is, to people who are uninformed about the subject. The **tone** you will use is fairly neutral, but still enthusiastic. The context in which you are creating this introduction is for an article that will explain how to generate effective prompts with ChatGPT. **First,** you’ll explain what ChatGPT is and then you’ll write a tagline. I would like this introduction to be **no longer than 8 lines.**

* ChatGPT 4.0 (2023.03) > ChatGPT 4 Turbo (2023.11)
* 생성형 AI의 적용 케이스
* Quizlet사: OpenAI의 ChatGPT로 구축된 Q-chat AI 개인 튜터 서비스를 제공.
* Duolingo사: GPT-4를 이용해 두 가지 기능을 구현했다. AI 대화 파트너인 Roleplay 기능과, 대화 과정에서 실수한 문법 규칙을 세분화해서 설명하는 Explain My Answer 기능이다.
* Morgan Stanley사: 자산관리와 관련된 방대한 내부 데이터에서 검색할 수 있는 챗봇 개발을 위해 GPT-4를 도입했다. PDF 등의 다양한 포맷으로 구성된 데이터 검색을 수행할 수 있다.
* Viable사: GPT-4를 사용해 CS 티켓과 같은 자연언어 데이터 분석을 수행했다.

1. CI/CD: Continuous Integration / Continuous Delivery(Deployment) 개발자들이 코드를 업데이트하는 순간 자동으로 테스트가 이뤄지고 테스트상 문제가 없으면 코드를 패키지로 묶어 배포하기 쉽게 만드는 것(CI)과, 그 결과물을 프로덕션 서버에 배포하는 것(CD)을 의미한다. 이를 전담하는 팀이 DevOps다. [↑](#footnote-ref-1)
2. Escalation Process: 서비스 사고가 생긴 것을 인지할 때 그걸 해결하기 위한 프로세스를 시작하는 것이다. [↑](#footnote-ref-2)
3. 딥 러닝은 인공신경망을 사용해서 기존 기계 학습 알고리즘이 처리하지 못하는 복잡한 패턴을 처리하는 기술이다. [↑](#footnote-ref-3)
4. Context Window는 시스템이 한 번에 처리할 수 있는 정보의 양을 의미한다. 100만 개 이상의 토큰을 다루는 ‘Long Context Window’는 매우 방대한 정보를 한 번에 처리할 수 있는 시스템이다. (<https://www.skelterlabs.com/blog/sustainable-rag>) [↑](#footnote-ref-4)
5. RAG (Retrieval-Augmented Generation)은 검색 증강 생성이라는 의미로, LLM의 단점 중 사실 관계 오류 가능성과 맥락 이해의 한계를 개선하는 데 초점을 맞춘 방법이다. RAG는 LLM에 외부 지식 베이스를 연결하여 모델의 생성 능력과 사실 관계 파악 능력을 향상시킨다. (<https://modulabs.co.kr/blog/retrieval-augmented-generation/>) [↑](#footnote-ref-5)