

Generative AI





Generative AI

생성형 AI는 인공지능망을 이용하여 새로운 데이터를 생성해내는 기술로 명령어(Prompt)를 통해 사용자의 의도를 스스로 이해하고, 주어진 데이터로 학습, 활용하여 텍스트, 이미지, 오디오, 비디오 등 새로운 콘텐츠를 생성해내는 인공지능이다.

ChatGPT



Bard



DALL-E 2



Stable Diffusion



Midjourney



<https://chat.openai.com/>

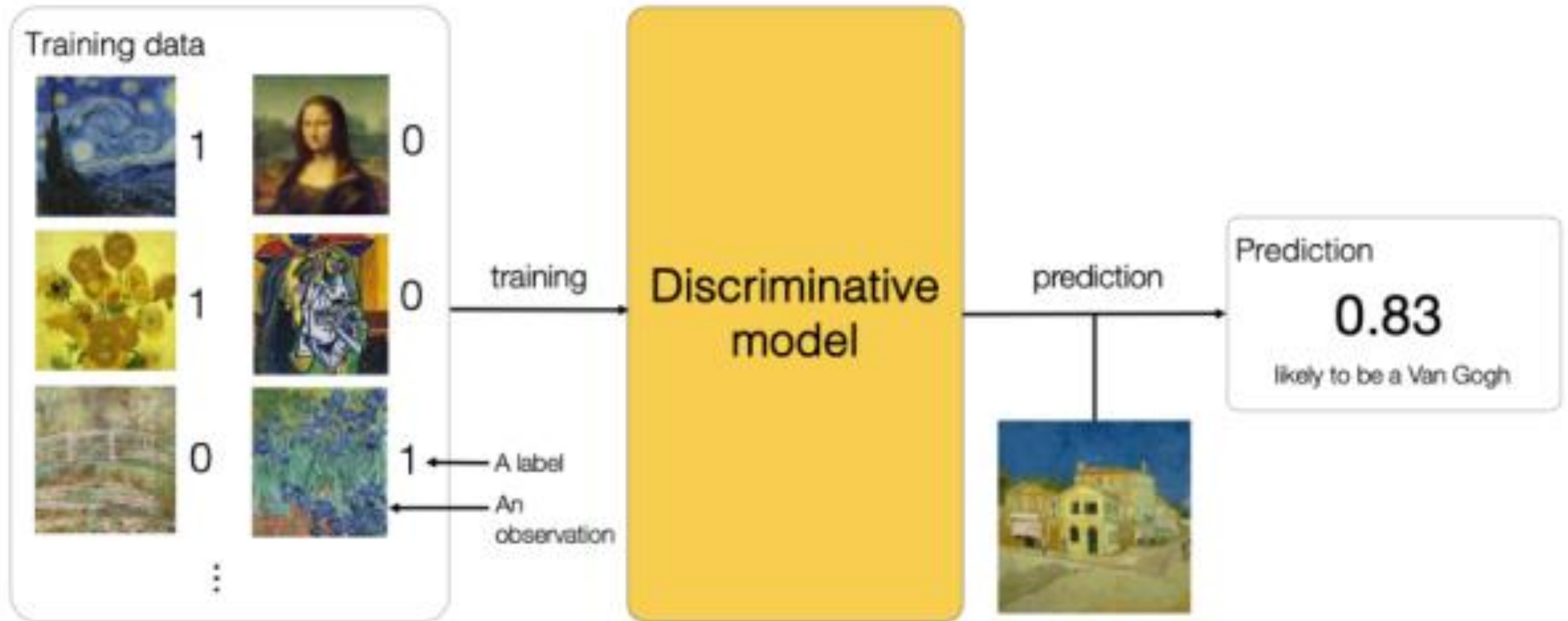
<https://www.midjourney.com/>

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

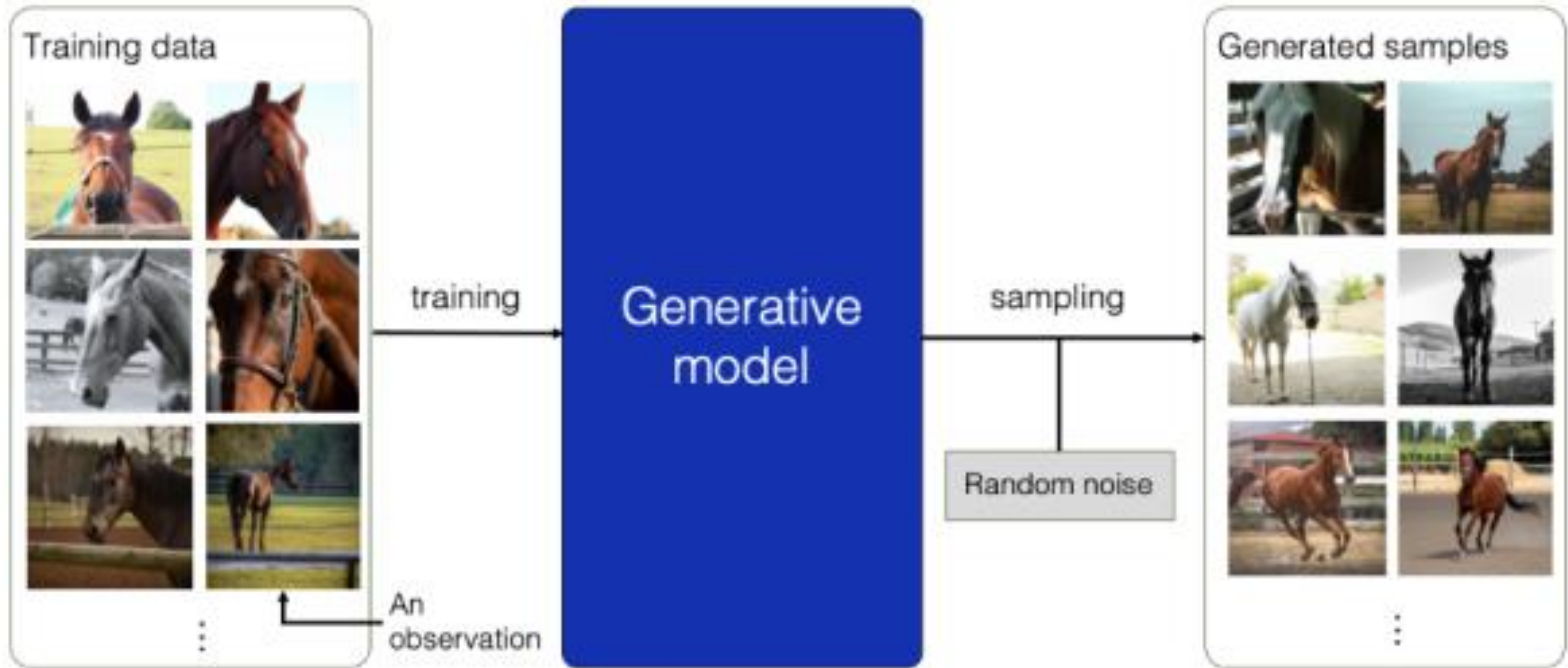
<https://stablediffusionweb.com/>

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

판별 모델링(Discriminative Modeling)



생성 모델링(Generative Modeling)



생성형 모델의 등장



2014



2015



2016



2017



2018



2020

ChatGPT

<https://chat.openai.com/>

텍스트 이해
질문에 대한 답변
콘텐츠 생성
리스트 생성
코드 작성/디버깅
장단점 비교 제시
교육 지원
창의적인 글쓰기
아이디어 기획
번역
단계별 지침 제공
가상 비서 역할

The screenshot shows the ChatGPT web interface. On the left is a dark sidebar with the following links: Reset Thread, Dark Mode, OpenAI Discord, Learn More, and Log out. The main content area has the title 'ChatGPT' at the top. Below it is a table with three columns: Examples, Capabilities, and Limitations. At the bottom of the main area is a text input field with a send button (a right-pointing arrow). Below the input field is a small disclaimer: 'Free Research Preview: ChatGPT is optimized for dialogue. Our goal is to make AI systems more natural to interact with, and your feedback will help us improve our systems and make them safer.'

Examples	Capabilities	Limitations
"Explain quantum computing in simple terms"	Remembers what user said earlier in the conversation	May occasionally generate incorrect information
"Got any creative ideas for a 10 year old's birthday?"	Allows user to provide follow-up corrections	May occasionally produce harmful instructions or biased content
"How do I make an HTTP request in Javascript?"	Trained to decline inappropriate requests	Limited knowledge of world and events after 2021

ChatGPT

+ New Chat

Today

OpenCV

Yesterday

Computer Vision

Previous 30 Days

AI Biz Idea Projects

September

프로그램개발

August

SWOT분석

모델링

ChatGPT Tips for Beginners

TikTok Video Ideas

Upgrade to Plus

KG kgpark88@naver.com

chat.openai.com/c/178d9fd2-f363-4c10-aa99-cb867d23d5f3

Default (GPT-3.5)

KG haarcascades 에 대해 설명

Haar cascades는 객체 감지(object detection)에 사용되는 컴퓨터 비전 알고리즘 중 하나입니다. 이 알고리즘은 얼굴, 눈, 차량 등 다양한 객체를 탐지하는 데 사용됩니다. Haar cascades는 기존의 이미지나 비디오 데이터에서 특정 객체를 자동으로 인식하고 추적하는 데 도움이 됩니다.

Haar cascades의 주요 특징과 동작 원리는 다음과 같습니다.

1. 하르 특징(Haar Features): Haar cascades는 이미지의 픽셀 값을 비교하여 특징을 추출합니다. 이러한 하르 특징은 객체의 에지, 색상 변화, 질감 등과 관련된 정보를 포함하며, 객체와 객체가 아닌 영역을 구별하는 데 사용됩니다.

2. 적응적 학습(Adaptive Learning): Haar cascades는 초기에 학습 단계에서 양성(타겟 객체가 있는 이미지)과 음성(타겟 객체가 없는 이미지) 예제를 사용하여 모델을 하르 특징을 기반으로 객체를 감지하는 분류기로 학습시킵니다.

3. Cascade Classifier: Haar cascades는 분류기를 계층화하여 사용합니다. 이것을 "Cascade Classifier"라고 합니다. Cascade Classifier는 순차적으로 객체의 존재 여부를 판별하며, 각 단계에서 거짓 양성 비율을 최소화하도록 조정됩니다. 이러한 방식으로 빠른 객체 감지를 가능하게 합니다.

Send a message

Free Research Preview. ChatGPT may produce inaccurate information about people, places, or facts. ChatGPT September 25 Version

ChatGPT에 대해 설명해줘

부산 여행 1일 여행계획표로 만들어줘

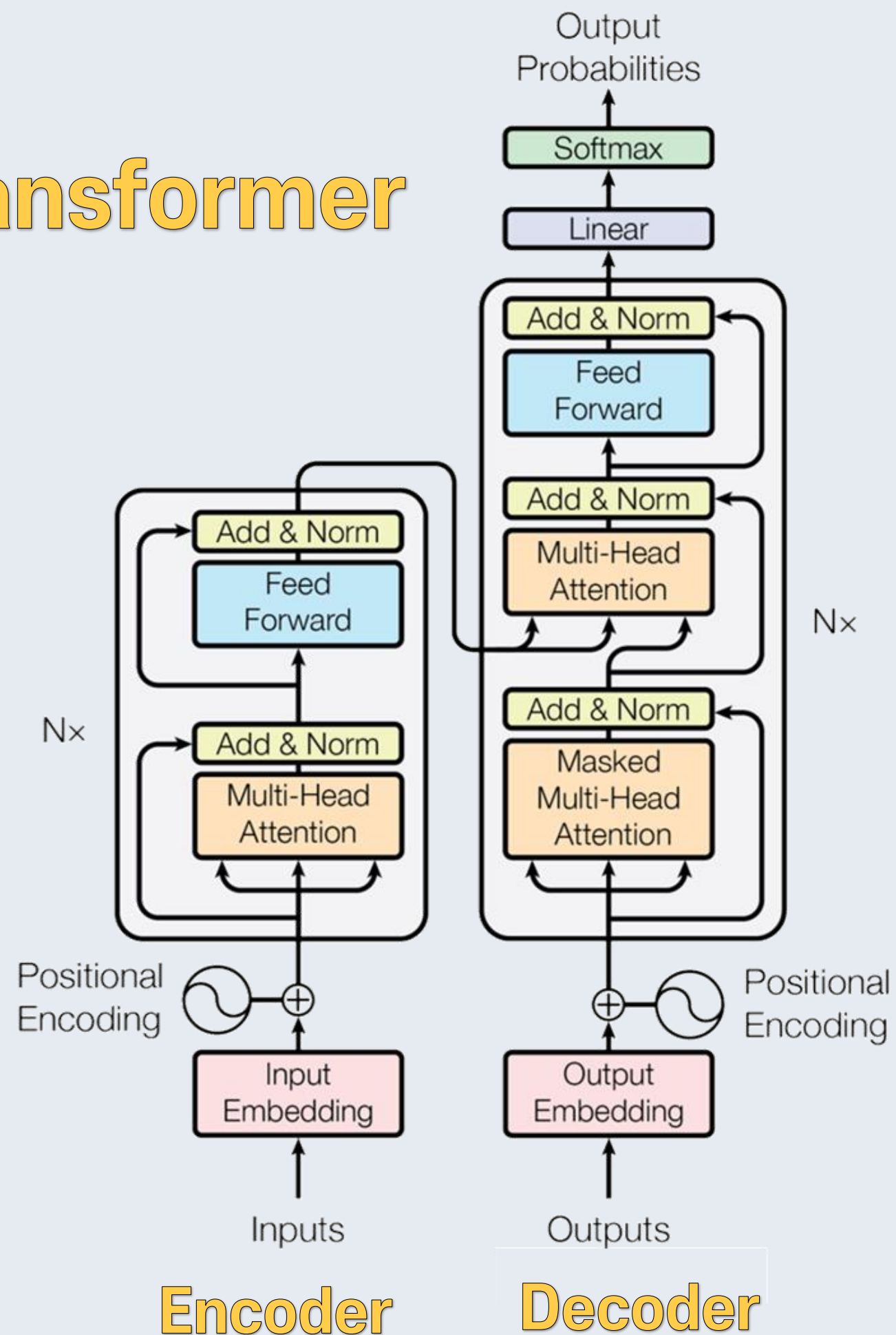
호텔에 Late check-in 하겠다는 영문 메일 써줘

8

GPT

GPT = Generative
Pre-trained
Transformer

Transformer



임베딩(Embedding)

원핫 인코딩(one hot encoding)

Human-Readable

Pet
Cat
Dog
Turtle
Fish
Cat

Machine-Readable

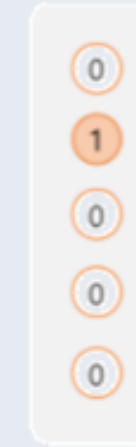
Cat	Dog	Turtle	Fish
1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1
1	0	0	0

희소표현(sparse representation)

고양이

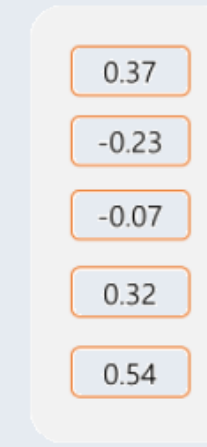


강아지

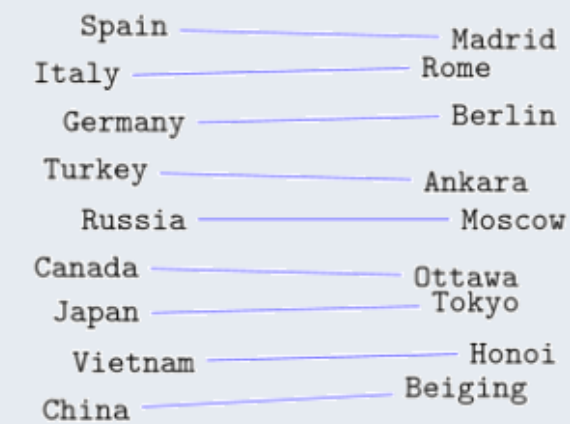
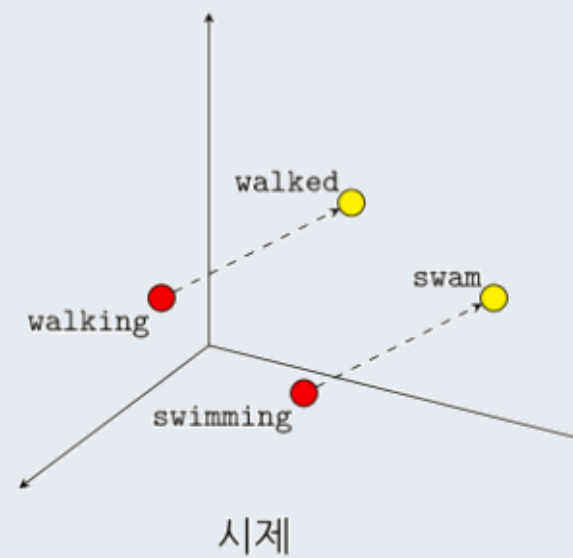
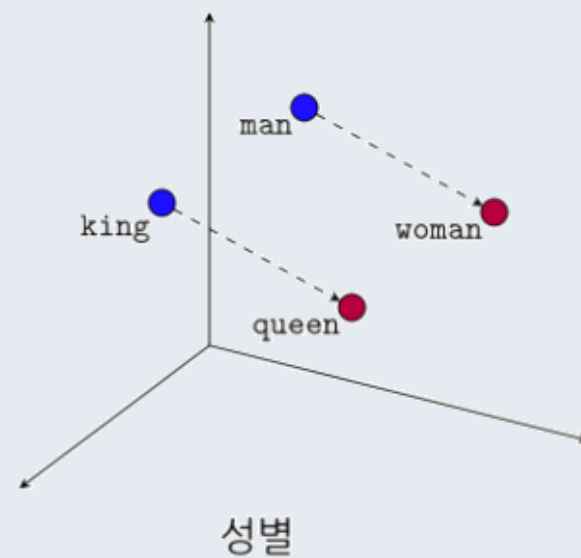
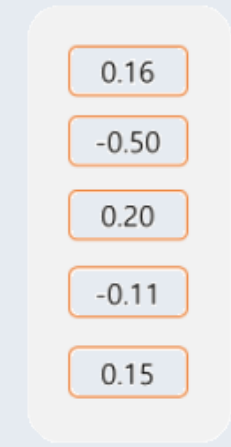


밀집표현(distributed representation)

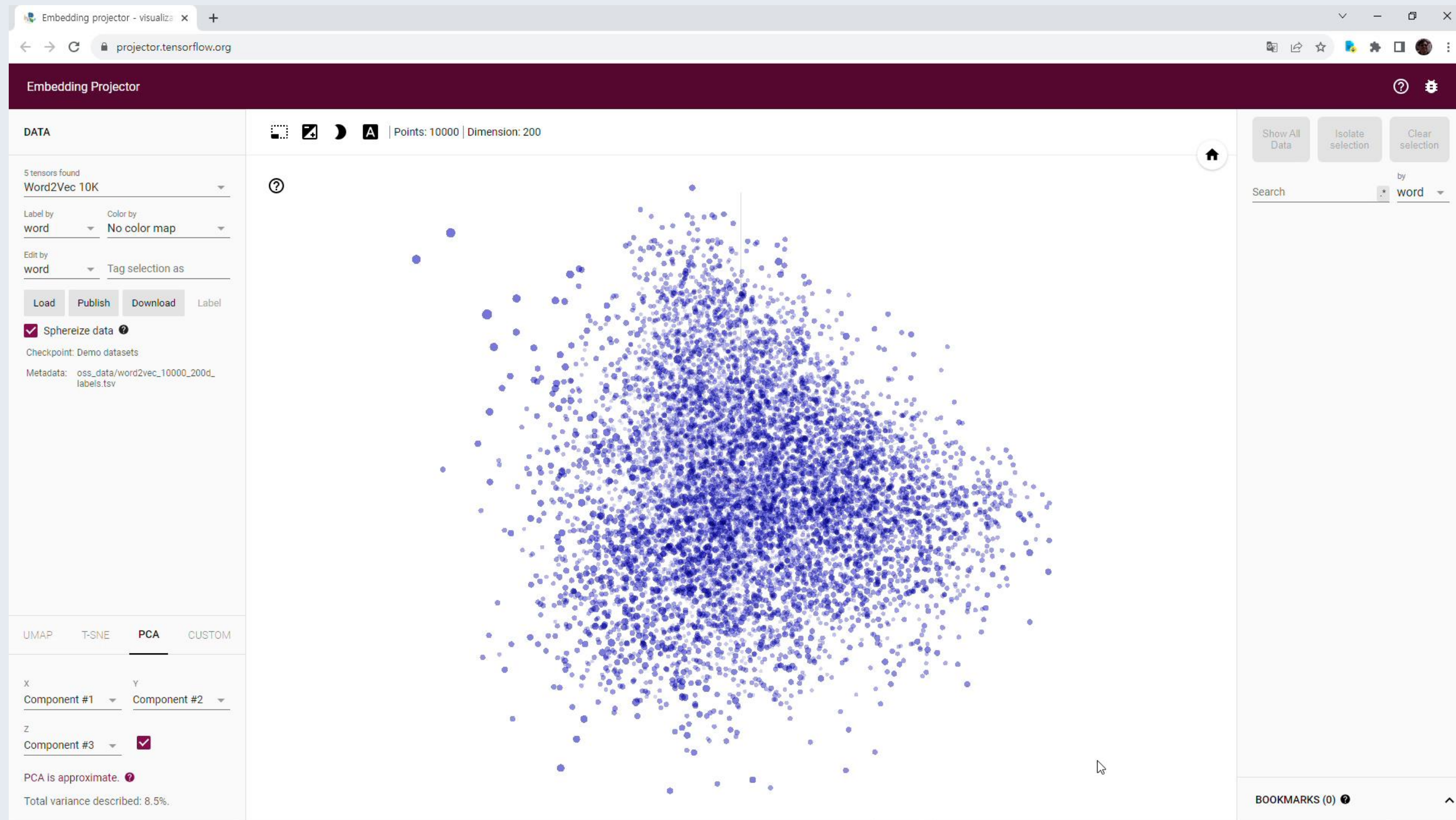
고양이



강아지



임베딩(Embedding)

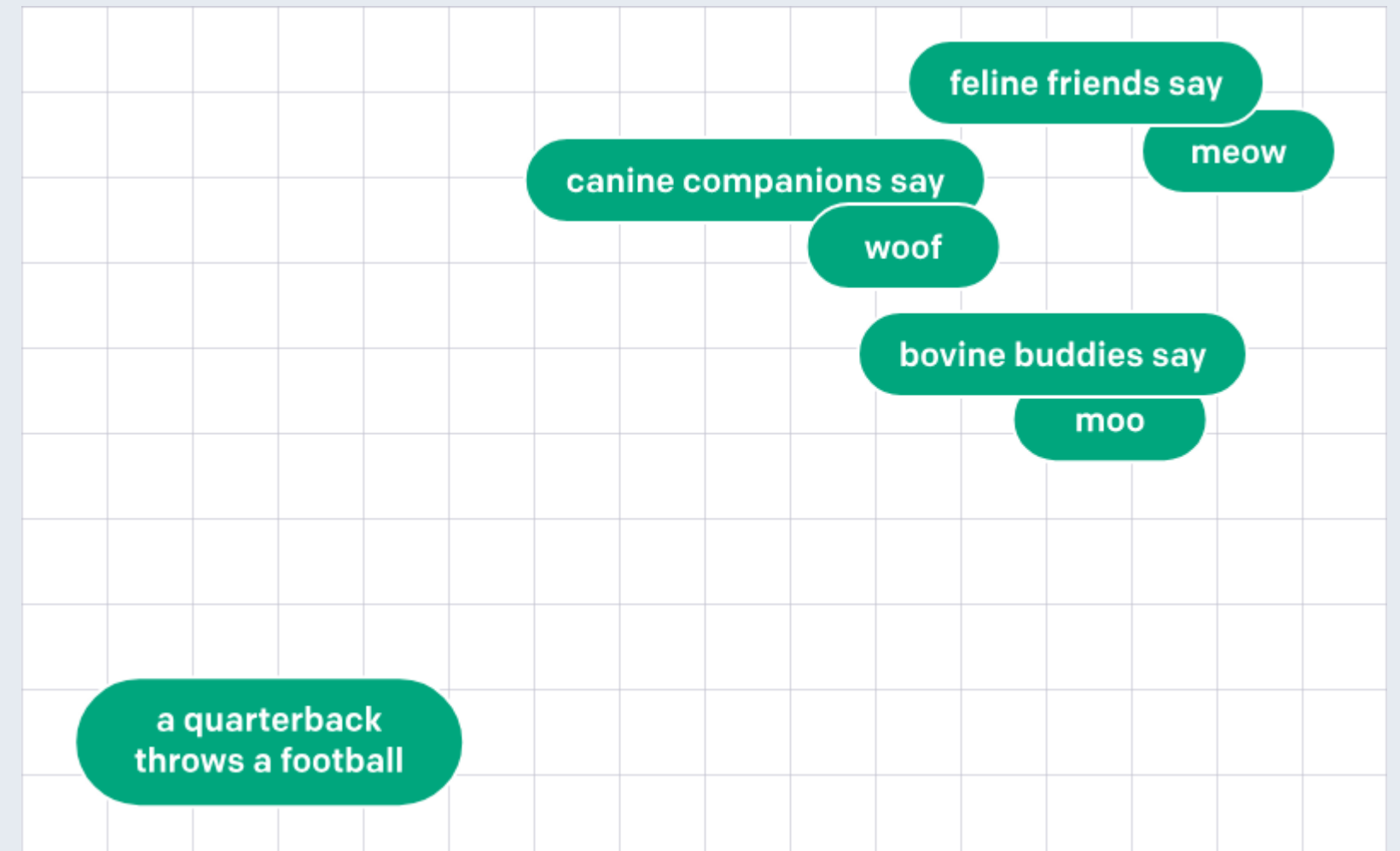
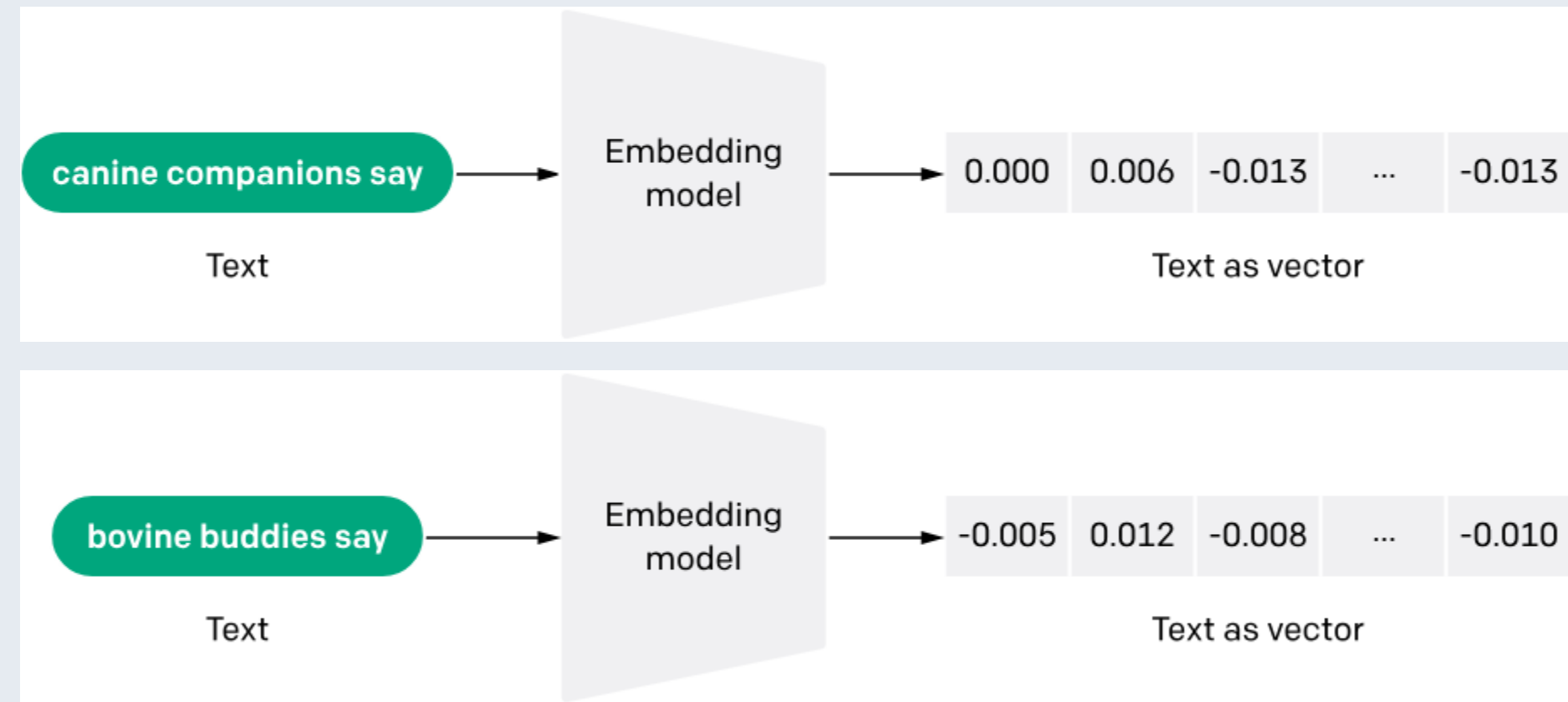


출처 : <https://projector.tensorflow.org/>

임베딩(Embedding)

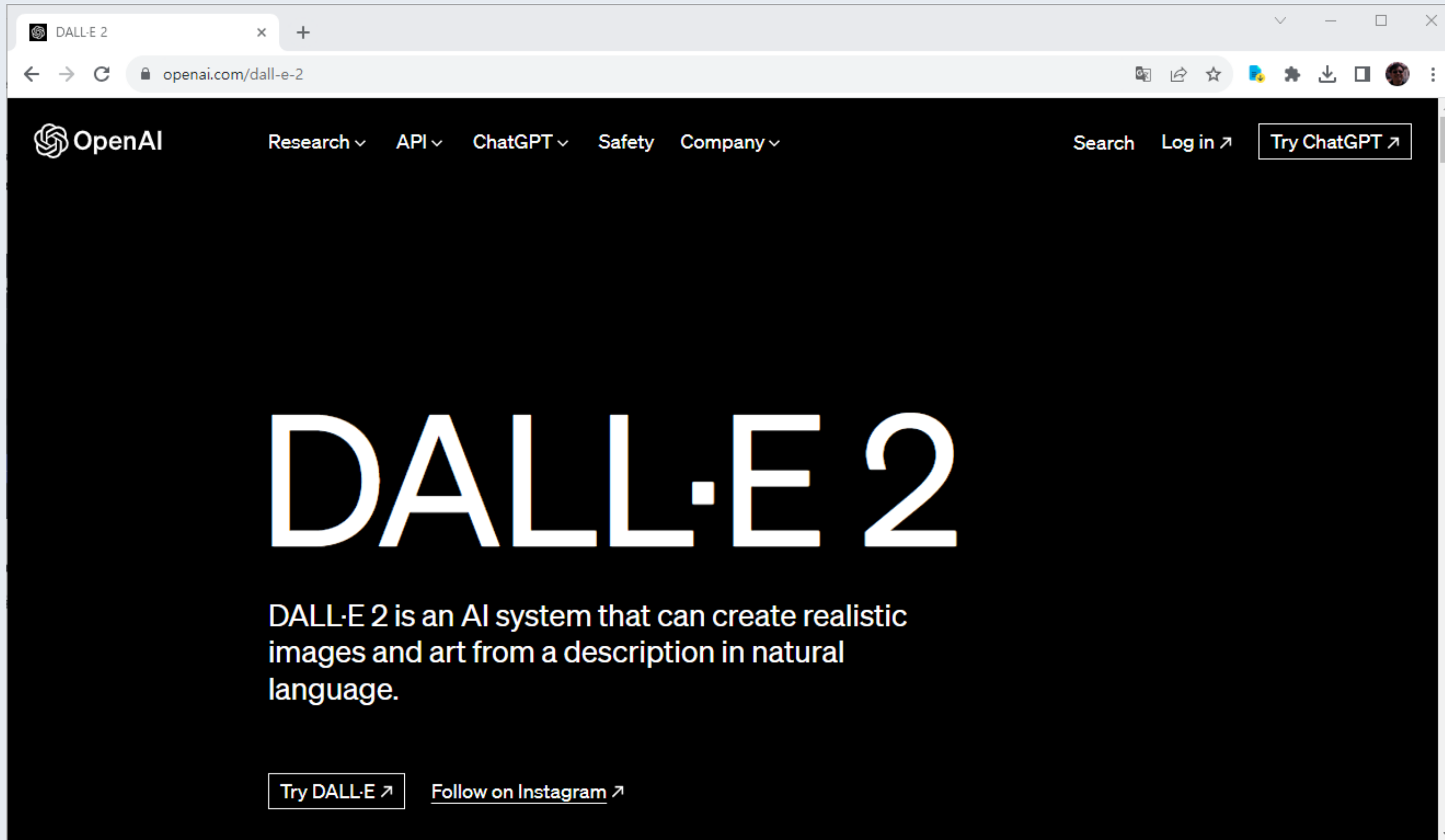
```
import openai
response = openai.Embedding.create(
    input="canine companions say",
    engine="text-similarity-davinci-001")

print(response)
{
  "data": [
    {
      "embedding": [
        0.000108064,
        0.005860855,
        -0.012656143,
        ...,
        -0.006642727,
        0.002583989,
        -0.012567150
      ],
      "index": 0,
      "object": "embedding"
    }
  ],
  "model": "text-similarity-davinci-001",
  "object": "list"
}
```



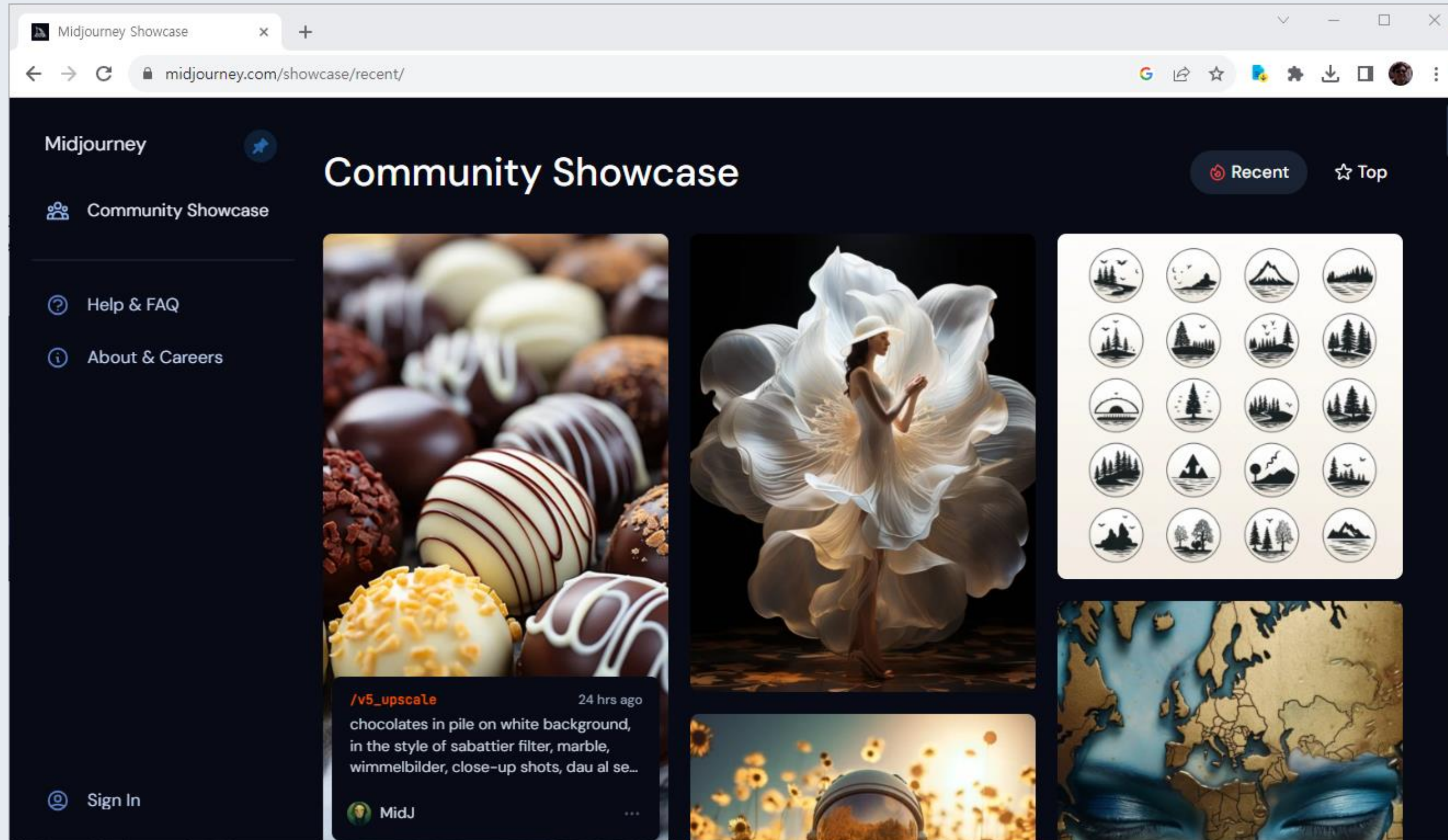
달리2

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



미드저니

<https://www.midjourney.com/>



빙 이미지 크리에이터

<https://www.bing.com/create>

Microsoft Bing

Image Creator
제공: DALL·E 3

PREVIEW

만들기 이미지 - 단어를 기반으로 AI를 사용하여 생성

만들고 싶은 이미지 설명





가입 및 만들기

Image Creator는 텍스트를 기반으로 AI 이미지를 생성합니다. 자세히 알아보세요.

Microsoft 및 파트너 제품에 대한 혜택이 포함된 Microsoft Rewards 관련 메일을 받게 됩니다. 또한 Bing Image Creator 관련 알림도 받게 됩니다. 계속 진행하면 아래의 Rewards 약관 및 Image Creator 약관에 동의하는 것으로 간주합니다.

[개인 정보](#) | [Rewards 약관](#) | [Image Creator 약관](#)

검색, 채팅 및 만들기를 모두 한 곳에서 사용할 수 있습니다.
새 Bing에서 Image Creator를 사용해 보세요.



개인정보처리방침 및 위치정보이용약관

콘텐츠 정책

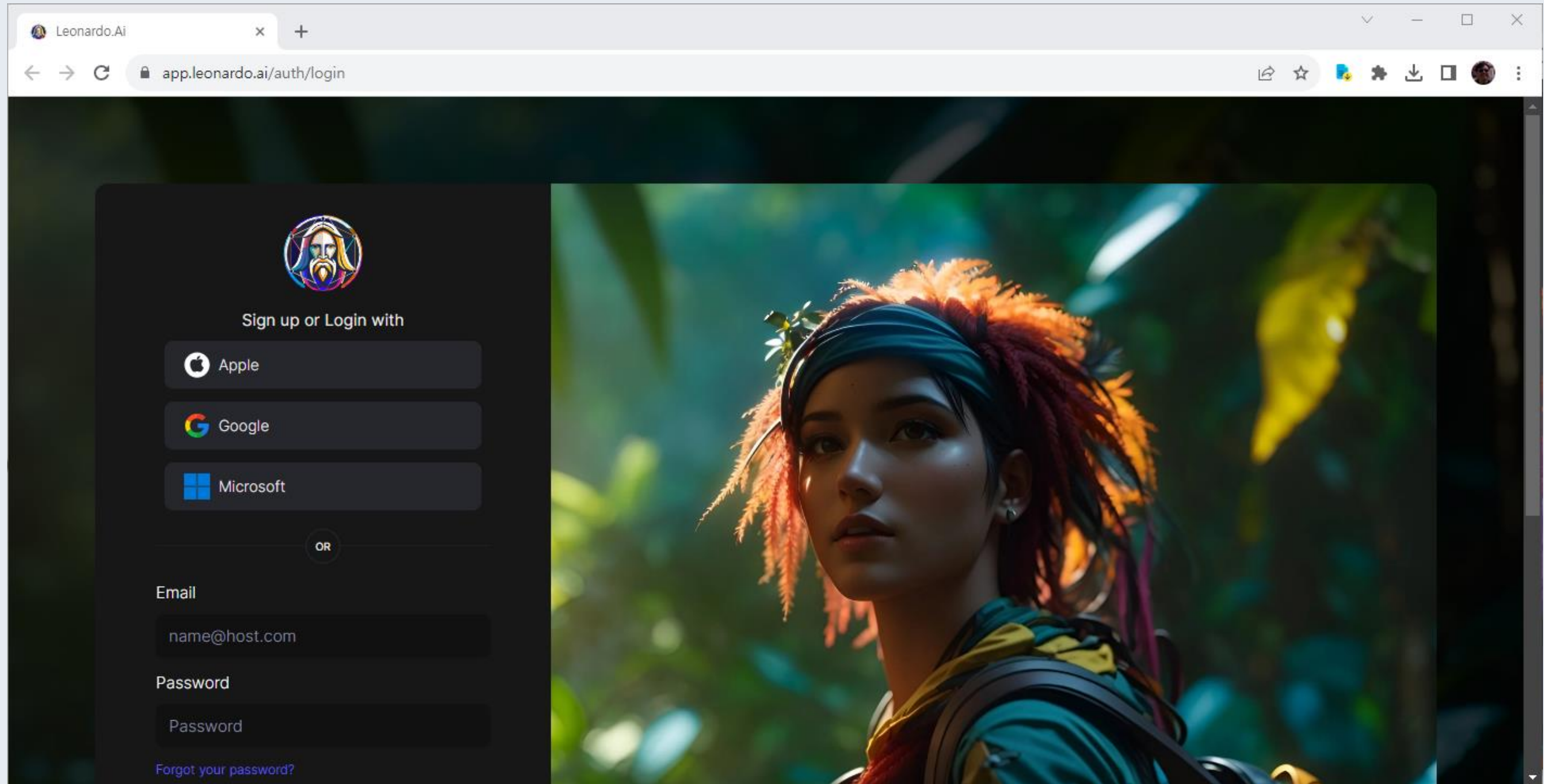
사용 약관

의견

피드백

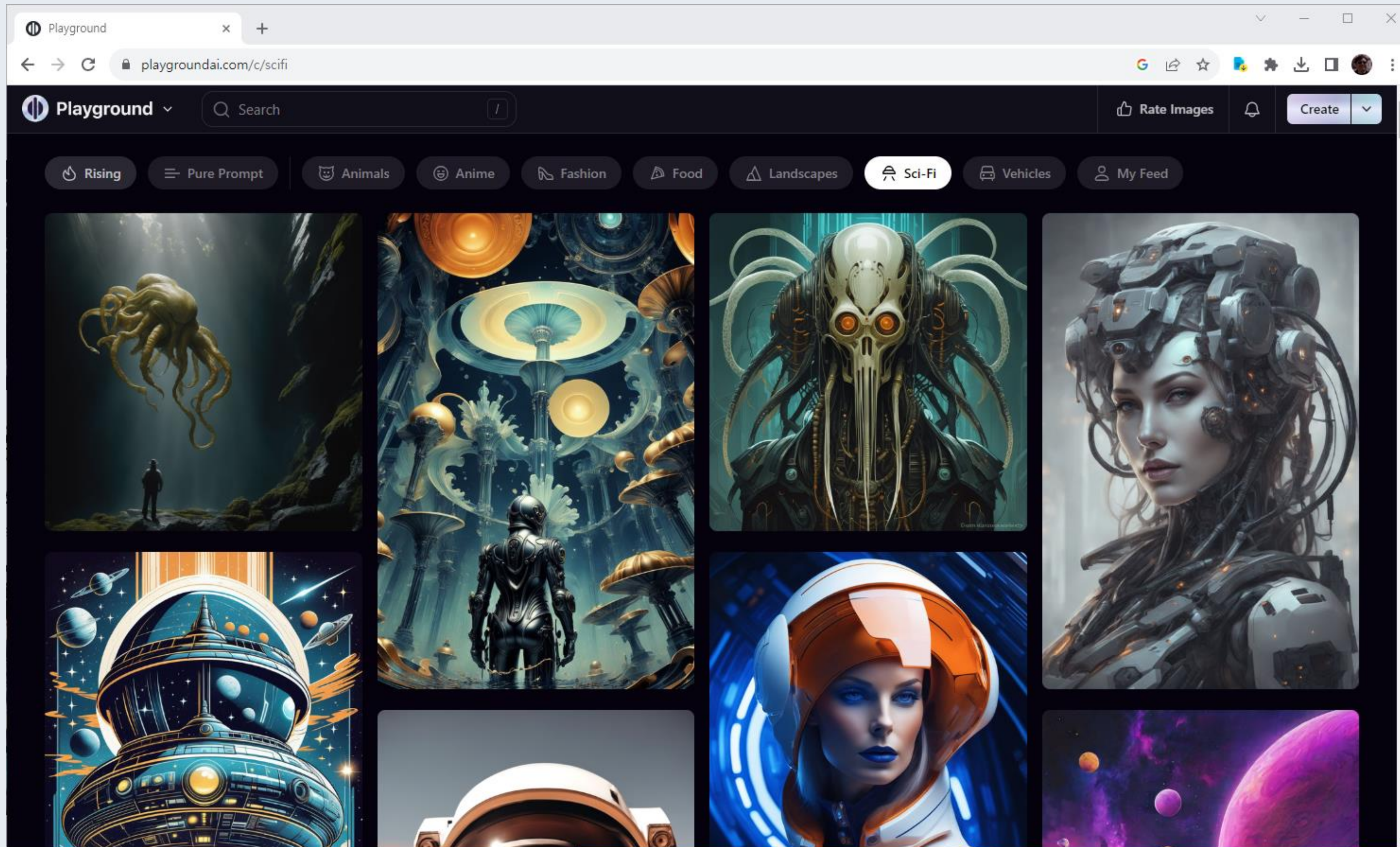
레오나르드 AI

<https://app.leonardo.ai/>



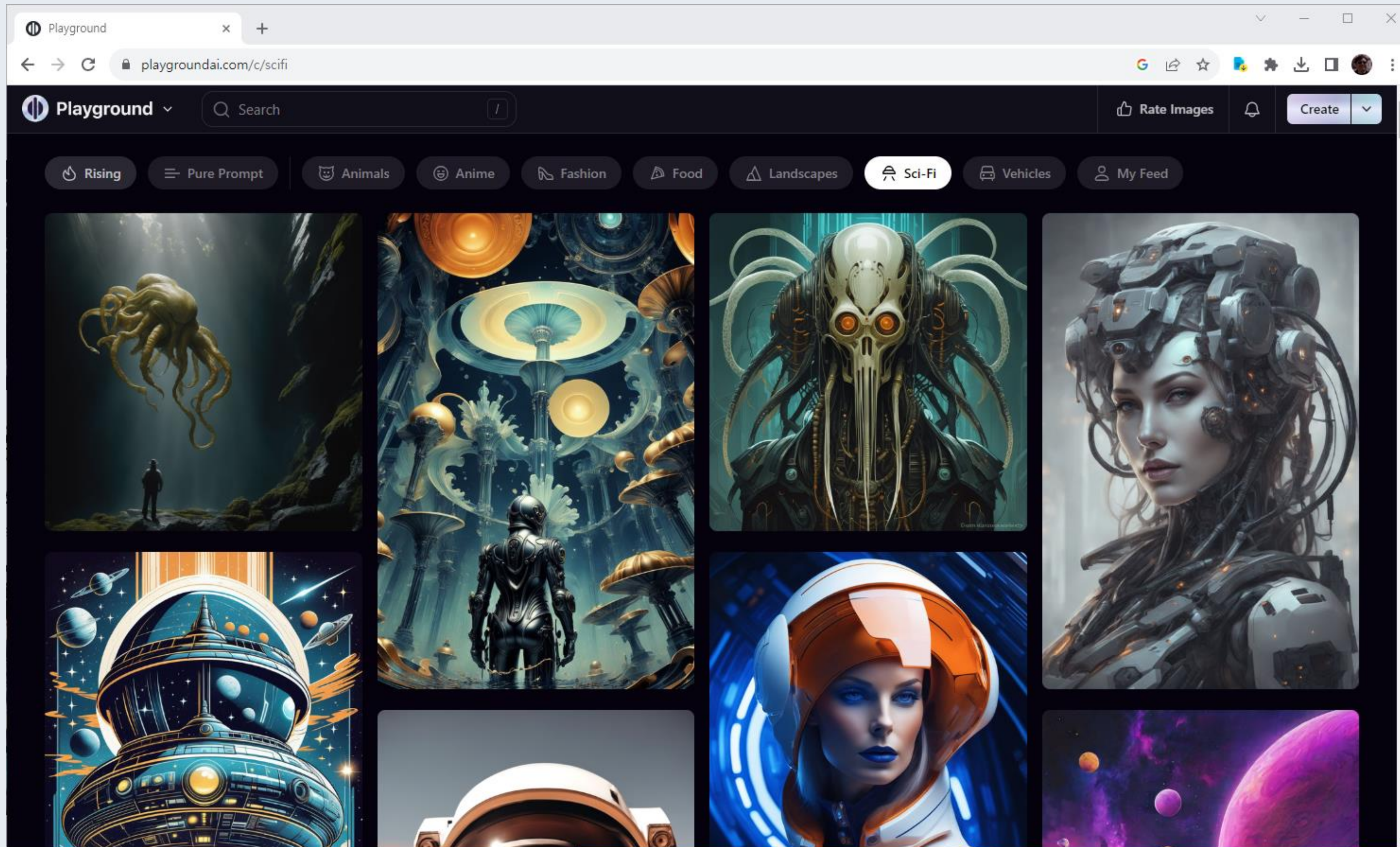
플레이그라운드AI

<https://playgroundai.com/>

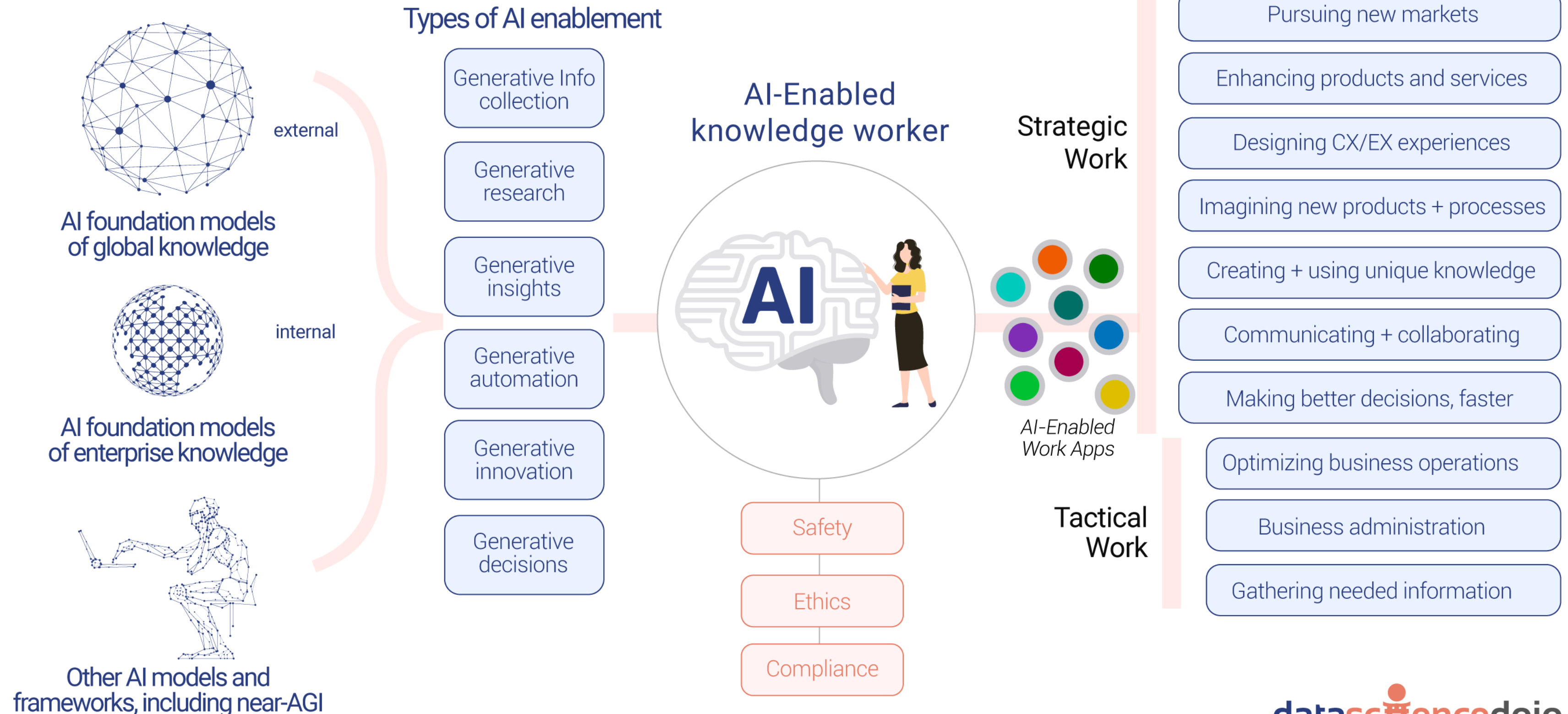


플레이그라운드AI

<https://playgroundai.com/>

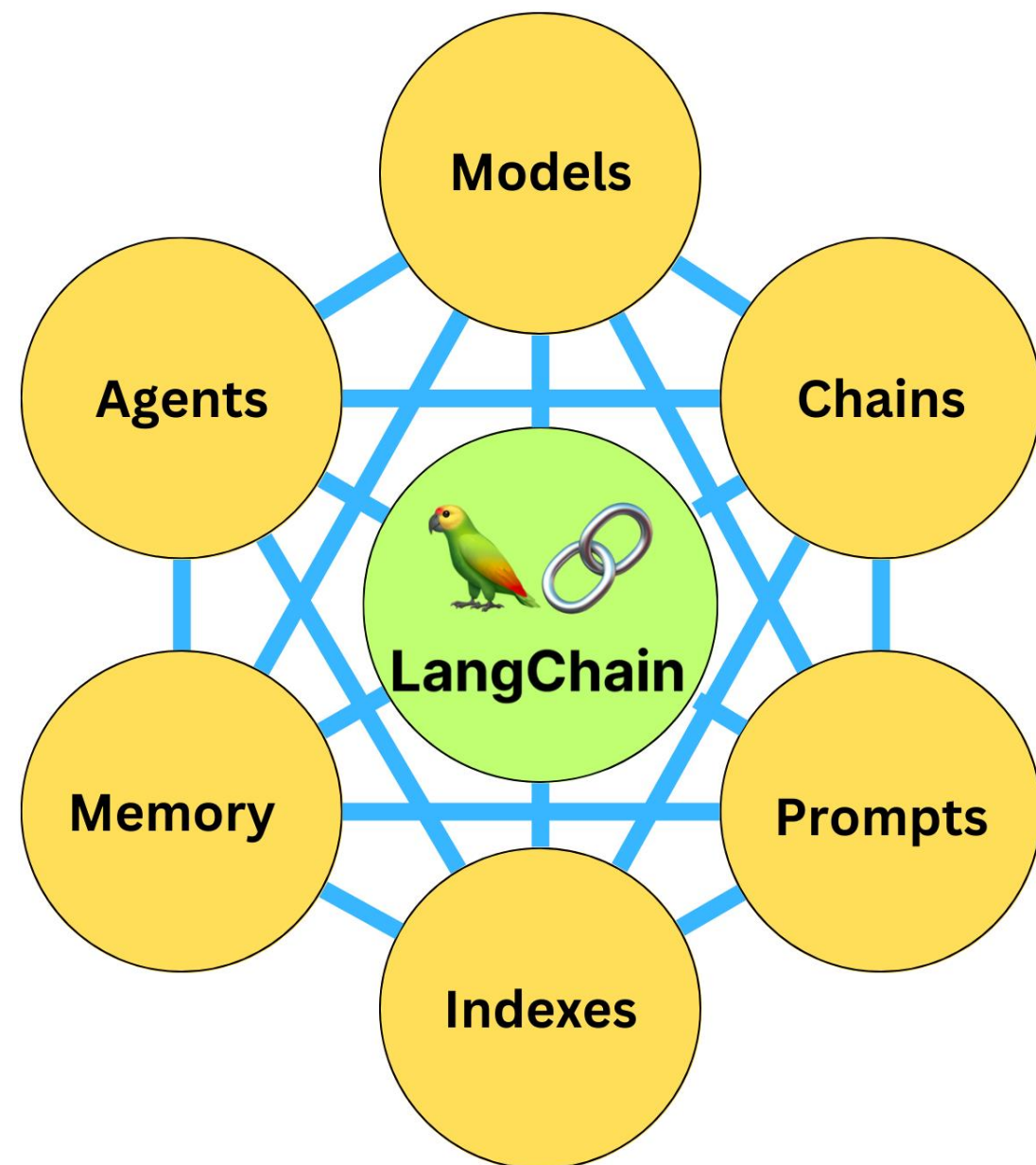


Generative AI Reshaping the future



How Artificial Intelligence will reshape the workplace and employee experience

Power your applications with Large Language Models



체인(Chains) : 일반적인 유스케이스를 달성하기 위한 컴포넌트들의 시퀀스인 체인의 표준 인터페이스와 일반적인 체인 구현을 제공

에이전트(Agents) : 체인에는 일련의 도구에 액세스할 수 있는 에이전트가 있으며, 에이전트는 사용자 입력에 따라 호출할 도구를 결정

도구(Tools) : 에이전트가 세상과 상호 작용하는 데 사용할 수 있는 기능, Python REPL, Google Search API, Wikipedia, Terminal 등

프롬프트(Prompts) : LLM에 적용할 수 있는 동적 프롬프트 작성

모델(Models) : 대부분의 LLM API에 연결하기 위한 추상화 계층

메모리(Memory) : 애플리케이션 상태 유지, 대화기록

인덱스(Indexes) : 문서 작업과 다른 벡터DB와의 통합을 위한 유틸리티 기능

LangChain



LangChain 실습

■ 데이터 <https://www.kaggle.com/datasets/sdolezel/black-friday>

LangChain 으로 자연어 질의 데이터 분석 프로그램 구현

분석 대상 데이터 : <https://www.kaggle.com/datasets/sdolezel/black-friday>

LangChain과 OpenAI 라이브러리 설치

```
!pip -q install langchain openai
```

라이브러리 로드

```
from dotenv import dotenv_values
from langchain.llms import OpenAI
from langchain.agents import create_csv_agent
```

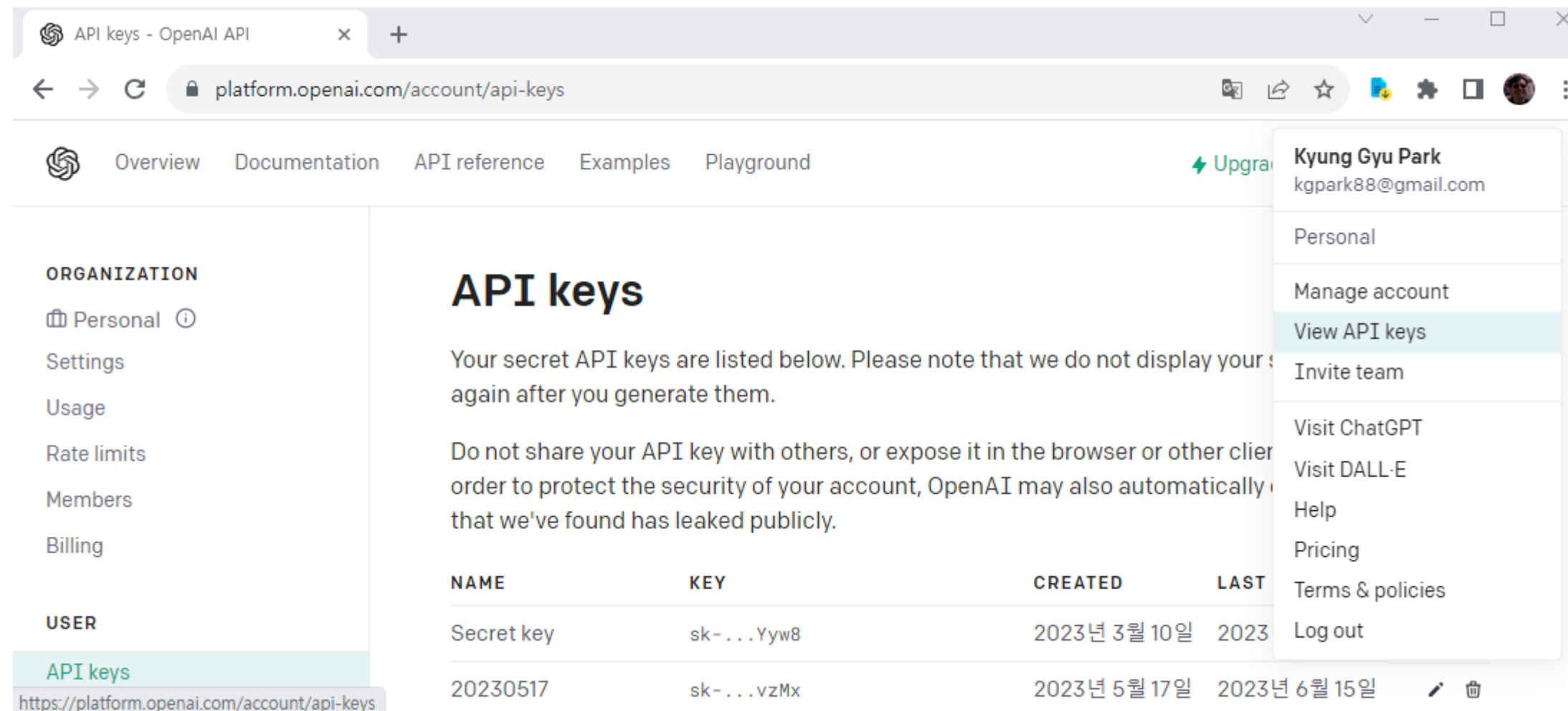
OpenAI API키 설정

- <https://platform.openai.com/account/api-keys> 에서 OpenAI API Key 확인

```
#setup the api key
# config = dotenv_values(".env")
# openai_api_key = config["OPENAI_API_KEY"]
openai_api_key = "sk-F4vrSYMEvrNfT123456789B1bkFJjk5xYxJiKyGpNasUGw6Z"
```


LangChain 실습

■ OpenAI API Key <https://platform.openai.com/account/api-keys>



The screenshot shows the OpenAI API keys management page. The page has a sidebar with navigation links: Overview, Documentation, API reference, Examples, and Playground. The main content area is titled 'API keys' and contains a warning about not sharing API keys. Below the warning is a table of API keys.

NAME	KEY	CREATED	LAST
Secret key	sk-...Yyw8	2023년 3월 10일	2023년 3월 10일
20230517	sk-...vzMx	2023년 5월 17일	2023년 6월 15일

A dropdown menu is open, showing options: Personal, Manage account, View API keys (highlighted), Invite team, Visit ChatGPT, Visit DALL-E, Help, Pricing, Terms & policies, and Log out.

LangChain 실습

Agent 실행

- SQL이나 Python이 아닌 자연어(프롬프트)로 데이터 세트를 쿼리할 수 있습니다.
- Agent가 파이썬 코드를 작성하고 실행합니다.

```
agent.run("데이터 행이 몇 개지?")
```

> Entering new AgentExecutor chain...

Thought: 데이터 프레임의 크기를 확인해야 합니다.

Action: python_repl_ast

Action Input: df.shape

Observation: (550068, 12)

Thought: 행의 개수는 첫 번째 숫자입니다.

Final Answer: 데이터 행은 550068개입니다.

> Finished chain.

'데이터 행은 550068개입니다.'

LangChain 실습

```
agent.run("현재 지역에 3년 이상 거주한 사람의 수는?")
```

```
> Entering new AgentExecutor chain...
```

```
Thought: 데이터프레임의 Stay_In_Current_City_Years 열을 사용해야 한다.
```

```
Action: python_repl_ast
```

```
Action Input: df['Stay_In_Current_City_Years'].value_counts()
```

```
Observation: 1      193821
```

```
2      101838
```

```
3       95285
```

```
4+      84726
```

```
0       74398
```

```
Name: Stay_In_Current_City_Years, dtype: int64
```

```
Thought: 4+ 카테고리를 계산해야 한다.
```

```
Action: python_repl_ast
```

```
Action Input: df[df['Stay_In_Current_City_Years'] == '4+']['Stay_In_Current_City_Years'].count()
```

```
Observation: 84726
```

```
Thought:
```

```
Retrying langchain.llms.openai.completion_with_retry.<locals>._completion_with_retry in 4.0 seconds as it raised RateLimitError: Rate limit reached for default-text-davinci-003 in organization org-88vQ4lhs2MyMLYzg0wpYnWH2 on requests per min. Limit: 3 / min. Please try again in 20s. Contact us through our help center at help.openai.com if you continue to have issues. Please add a payment method to your account to increase your rate limit. Visit https://platform.openai.com/account/billing to add a payment method..
```

```
이것이 최종 답이다.
```

```
Final Answer: 84726
```

```
> Finished chain.
```

```
'84726'
```

LangChain 실습

```
agent.run("A 지역에서 Product_Category_1을 구매한 여성은 몇 명?")
```

> Entering new AgentExecutor chain...

Thought: 이 문제를 해결하기 위해 데이터 프레임의 필터링 기능을 사용해야 합니다.

Action: python_repl_ast

Action Input: df[(df['City_Category'] == 'A') & (df['Gender'] == 'F')]['Product_Category_1'].count()

Observation: 35704

Thought: 이것이 우리가 찾는 답입니다.

Final Answer: A 지역에서 Product_Category_1을 구매한 여성은 35704 명입니다.

> Finished chain.

'A 지역에서 Product_Category_1을 구매한 여성은 35704 명입니다.'

```
agent.run("B 도시의 0-17세 사이의 여성이 구매한 Product_Category_1 수량의 총합은 몇 개이니?")
```

> Entering new AgentExecutor chain...

Thought: 먼저 데이터프레임에서 우리가 찾고자 하는 조건을 만족하는 데이터를 찾아야 합니다.

Action: python_repl_ast

Action Input: df[(df['City_Category'] == 'B') & (df['Age'] == '0-17') & (df['Gender'] == 'F')]['Product_Category_1'].sum()

Observation: 8240

Thought:

이제 총합을 구할 수 있습니다.

Final Answer: B 도시의 0-17세 사이의 여성이 구매한 Product_Category_1 수량의 총합은 8240개입니다.

> Finished chain.

'B 도시의 0-17세 사이의 여성이 구매한 Product_Category_1 수량의 총합은 8240개입니다.'

LangChain 실습

RFM 분석 : Recency(최근성), Frequency(빈도), Monetary(구매금액)로 고객 가치 분석

```
agent.run("Calculate recency, frequency and monetaty value for each customer?")
```

> Entering new AgentExecutor chain...

Thought: I need to calculate the recency, frequency and monetary value for each customer.

Action: python_repl_ast

Action Input: df.groupby('User_ID').agg({'Purchase': ['count', 'sum', 'max']})

Observation:

		Purchase		
	count	sum	max	
User_ID				
1000001	35	334093	19219	
1000002	77	810472	20657	
1000003	29	341635	19631	
1000004	14	206468	19693	
1000005	106	821001	21021	
...	
1006036	514	4116058	23525	
1006037	122	1119538	23714	
1006038	12	90034	13565	
1006039	74	590319	23920	
1006040	180	1653299	23251	

[5891 rows x 3 columns]

Thought: I now know the final answer

Final Answer: The recency, frequency and monetary value for each customer can be calculated using the groupby and agg functions in pandas.

> Finished chain.

'The recency, frequency and monetary value for each customer can be calculated using the groupby and agg functions in pandas.'

THANK YOU

kgpark88@gmail.com