2025년도 한국분석과학회 춘계 학술대회 튜토리얼

인공지능 기반 IR 스펙트럼 분석 방법 소개 및 기초 프로그래밍 실습

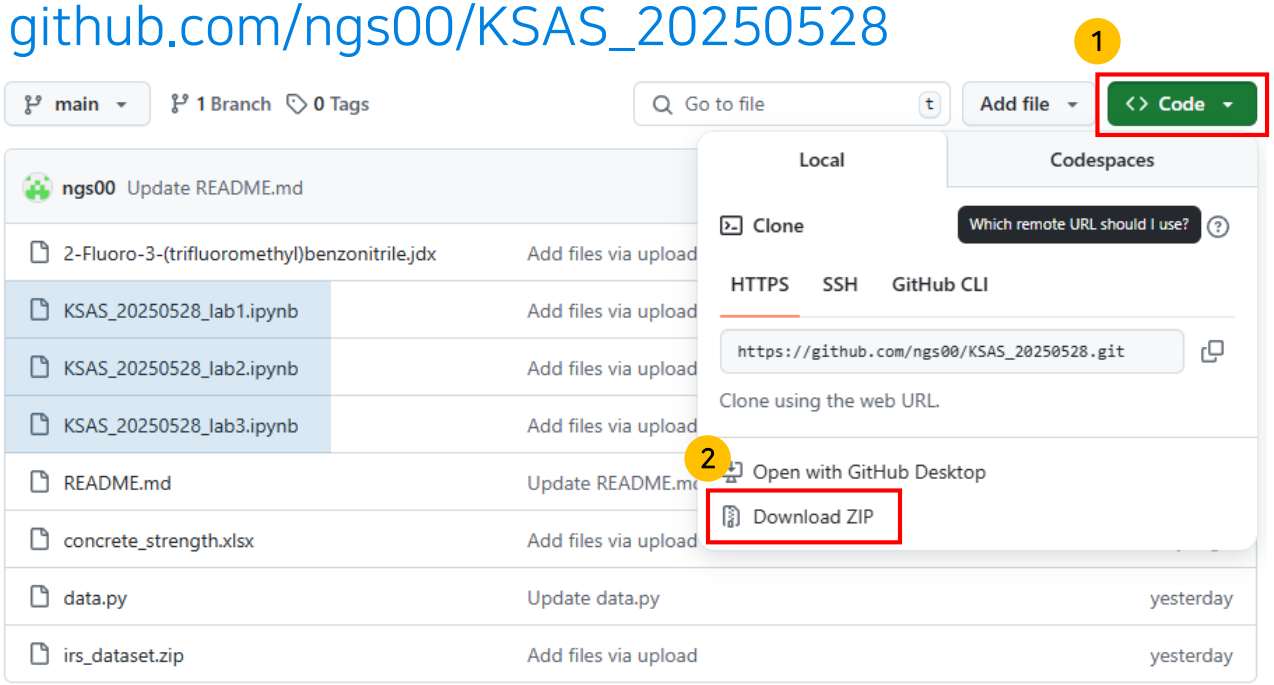
나경석 (Gyoung S. Na)
한국화학연구원 디지털화학연구센터
ngs0@krict.re.kr

실습 코드 및 참고 자료

[1] https://github.com/ngs00/KSAS_20250528
[2] https://ieeexplore.ieee.org/document/10607172?
[3] https://pubs.acs.org/doi/10.1021/acs.analchem.4c04786

실습 코드 및 데이터:

github.com/ngs00/KSAS_20250528



튜토리얼 순서

15:30 – 15:50 (20분)	▪ 인공지능 개념 소개 ▪ 화학 응용 적용 사례 소개
15:50 – 16:10 (20분)	▪ 실습 1: 개발 환경 설정 및 인공지능 예측 모델 구축 ▪ 개발 환경 설정 관련 추가 설명
16:10 – 16:30 (20분)	▪ 인공지능 기반 IR 스펙트럼 분석 모델 ▪ 생성형 인공지능 기반의 IR 스펙트럼 예측
16:30 – 17:00 (30분)	▪ 실습 2: IR 스펙트럼 시각화 및 데이터 전처리 ▪ 실습 3: 인공지능 기반 IR 스펙트럼 분석 모델 구축

기존 및 인공지능 기발 알고리즘 개발 방식

알고리즘 개발을 위한 응용수학 및 컴퓨터공학의 고전적인 방법론과 인공지능 기반 방식의 새로운 방법론

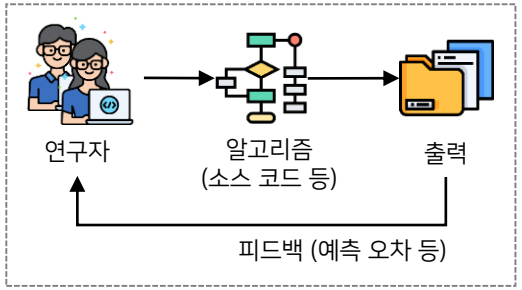
- 기존의 수학 및 컴퓨터공학 방법론에서는 연구자의 주관적인 경험과 직관을 바탕으로 특정 기능을 수행하는 컴퓨터 알고리즘을 개발
- 기존의 알고리즘 개발 방식은 시간과 비용이 많이 소모되고, 알고리즘의 성능이 연구자의 경험과 직관에 크게 의존하는 문제점이 존재
- 최근의 인공지능 기반 알고리즘 개발 방식에서는 주어진 관측 데이터 (= 학습 데이터셋)을 기반으로 인공지능이 자동으로 최적의 알고리즘을 학습
- 인공지능 기반 알고리즘 개발 방식에서 연구자들의 주요 업무 중 하나는 인공지능 학습을 위해 편향되지 않은 학습 데이터셋을 수집하고 이를 적절하게 변화하는 것
- 인공지능 기술의 발전에 따라 인공지능 학습 데이터셋 시장은 21조 600억원 규모까지 성장할 것으로 전망됨*

이미지 필터링을 위한 고전적 알고리즘

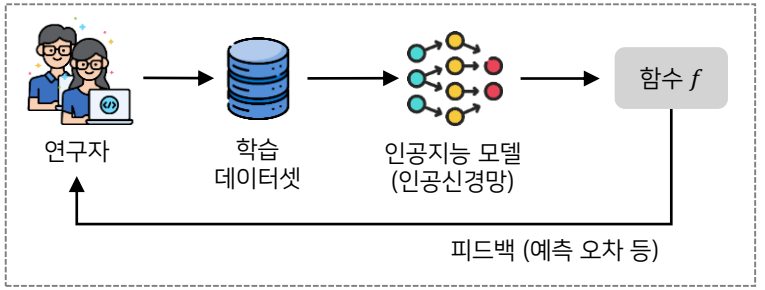
$$g(x,y) = \omega * f(x,y) = \sum_{i=-a}^{a} \sum_{j=-b}^{b} \omega(i,j) f(x-i, y-j)$$

```
mat3[3] region3x3(sampler2D sampler, vec2 uv)
{
    // Create each pixels for region
    vec4[9] region;

    for (int i = 0; i < 9; i++)
        region[i] = texture(sampler, uv + kpos(i));

    // Create 3x3 region with 3 color channels (red, green, blue)
    mat3[3] mRegion;

    for (int i = 0; i < 3; i++)
        mRegion[i] = mat3(
            region[0][i], region[1][i], region[2][i],
            region[3][i], region[4][i], region[5][i],
            region[6][i], region[7][i], region[8][i]
        );

    return mRegion;
}
```

고전적인 알고리즘 개발 방식



인공지능 기반의 알고리즘 개발 방식



사람의 직관으로 개발한 알고리즘과 인공지능 모델이 학습한 알고리즘 중 어떤 것이 더 정확한가? [Turing Test, 1950]

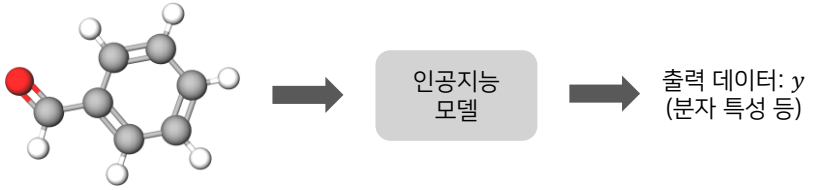
*인공지능 학습 데이터셋 시장은 2023년 3조 1500억원 규모에서 2030년 21조 600억원 규모까지 성장할 것으로 전망

2016년 구글 딥마인드 챌린지

인공지능 모델 구축을 위한 기본 요소

인공지능 모델 구축을 위한 머신러닝의 기본 요소: 예측 모델, 하드웨어, 학습 데이터셋

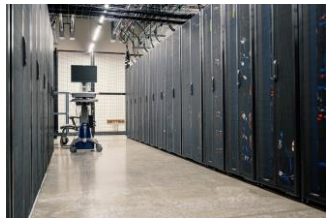
- 인공지능 모델의 근본적인 목적은 입력 데이터 x로부터 출력 데이터 y를 계산하는 것 (예: 현재 주식 가격→미래 주식 가격, 분자 구조→분자 특성)
- 입력 데이터 x는 어떠한 현상을 설명하고, 인공지능은 이를 바탕으로 주어진 현상에 대한 출력 데이터 y를 예측



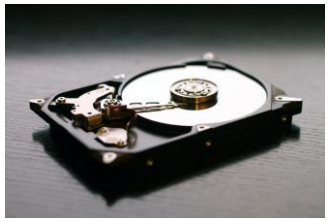
입력 데이터: x

- 인공지능 모델을 구축하는 주요 기술 중 하나인 머신러닝은 크게 인공지능 모델, 하드웨어, 학습 데이터셋의 세 가지 요소로 구성
- 인공지능 모델은 입력 데이터 x가 주어졌을 때 y를 예측하는 방식을 수학적으로 정의
- 학습 데이터셋은 어떠한 x가 입력되었을 때, 입력된 x에 대해 어떠한 y가 출력되어야 한다는 학습 정보를 제공

인공지능 예측 모델



하드웨어 자원



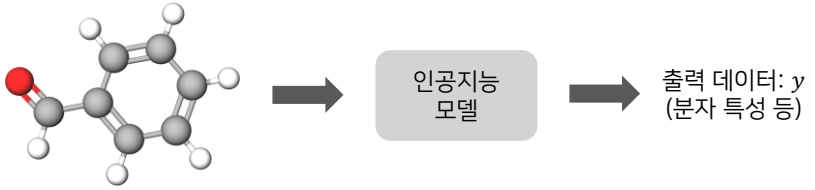
학습 데이터셋

$$\theta = \underset{\theta \in \mathcal{P}}{\operatorname{argmin}}\, L(y, f(x;\theta)) \rightarrow \underset{\theta \in \mathcal{P}}{\operatorname{argmin}} \sum_{(x,y)\in\mathcal{D}} (y - f(x;\theta))^2$$

인공지능 모델 구축을 위한 기본 요소

인공지능 모델 구축을 위한 머신러닝의 기본 요소: 예측 모델, 하드웨어, 학습 데이터셋

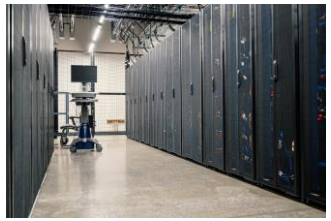
- 인공지능 모델의 근본적인 목적은 입력 데이터 x로부터 출력 데이터 y를 계산하는 것 (예: 현재 주식 가격→미래 주식 가격, 분자 구조→분자 특성)
- 입력 데이터 x는 어떠한 현상을 설명하고, 인공지능은 이를 바탕으로 주어진 현상에 대한 출력 데이터 y를 예측



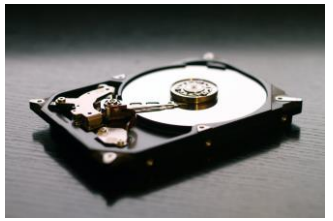
입력 데이터: x

- 인공지능 모델을 구축하는 주요 기술 중 하나인 머신러닝은 크게 인공지능 모델, 하드웨어, 학습 데이터셋의 세 가지 요소로 구성
- 인공지능 모델은 입력 데이터 x가 주어졌을 때 y를 예측하는 방식을 수학적으로 정의
- 학습 데이터셋은 어떠한 x가 입력되었을 때, 입력된 x에 대해 어떠한 y가 출력되어야 한다는 학습 정보를 제공

인공지능 예측 모델



하드웨어 자원



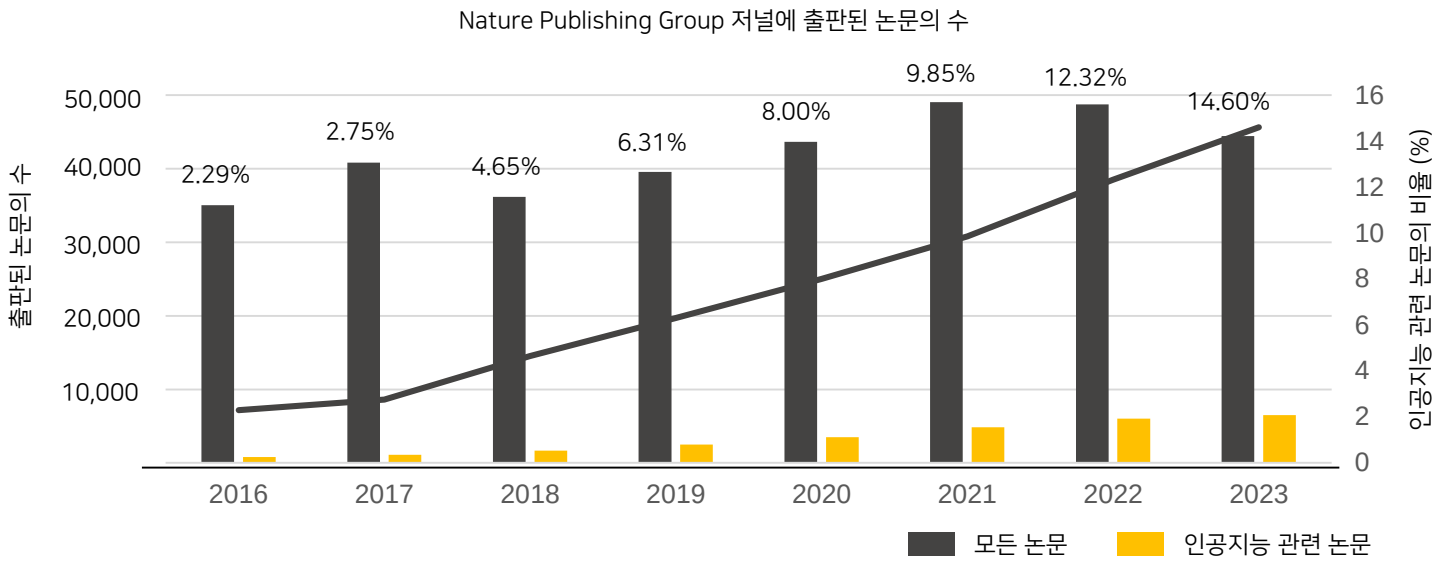
학습 데이터셋

$$\theta = \underset{\theta \in \mathcal{P}}{\operatorname{argmin}}\, L(y, f(x;\theta)) \rightarrow \underset{\theta \in \mathcal{P}}{\operatorname{argmin}} \sum_{(x,y)\in\mathcal{D}} (y - f(x;\theta))^2$$

학습 데이터 인공지능 모델

인공지능 기반 자연과학 연구 동향

출판 논문 통계 기반의 인공지능 기반 자연과학 연구 동향 분석



14.60%

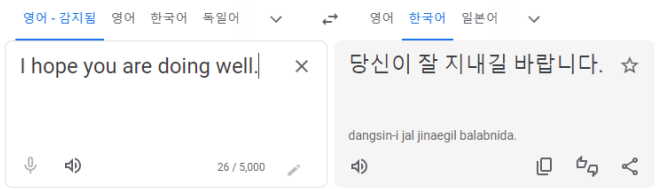
2023년도 기준

자연과학 분야에서 인공지능이 얼마나 다양하게 활용 및 연구되고 있는지를 분석하기 위해 Nature Publishing Group 저널에 출판된 인공지능 관련 논문의 수를 분석함. 인공지능 관련 논문의 수 (노란색)는 알파고 등장인 2016년 이후로 꾸준히 증가하고 있음. 또한 인공지능 관련 논문 수의 증가율도 꾸준히 증가하고 있음. 비록 2023년도에 출판된 전체 논문의 수는 감소하였지만, 인공지능 관련 논문의 수는 오히려 증가하여 전체 논문 대비 14.60%를 차지하였음.

인공지능의 발전 및 과학 응용 적용

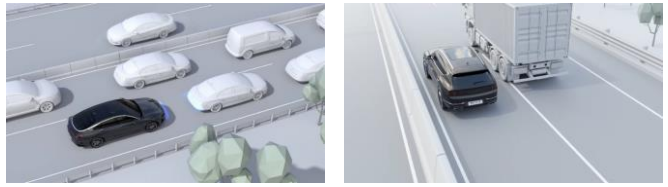
인공지능 방법론의 발전과 과학 응용에 대한 인공지능 방법론의 적용 사례

구글 한/영 번역기

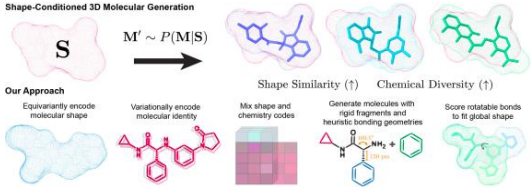


유튜브 영상 추천 알고리즘

인공지능 기반 자율주행 자동차

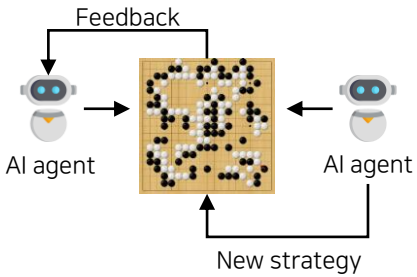


인공지능을 이용한 신약 개발 과정



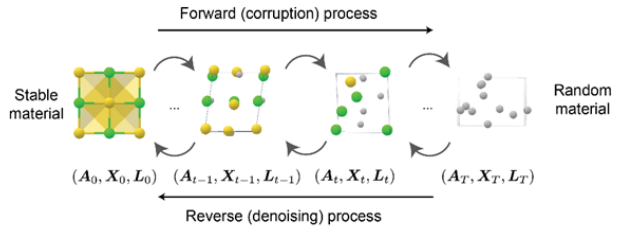
생성형 인공지능을 이용한 테마 지정 그림 생성
(https://arxiv.org/abs/1508.06576, arXiv, 2015)

AlphaGo Zero: 강화학습 기반의 바둑 인공지능
(Mastering the game of go without human knowledge, Nature, 2017)



Rank	Name	Nation	Elo
1	**AlphaGo Zero**	None	5,185
2	**AlphaGo Master**	None	4,858
3	Shin Jinseo	Korea	3,864
4	AlphaGo Lee	None	3,739
5	Ke Jie	China	3,677
6	Park Junghwan	Korea	3,675
7	Wang Xinghao	China	3,671

확산 모델 기반의 무기물 설계 인공지능
(MatterGen, Nature, 2025)

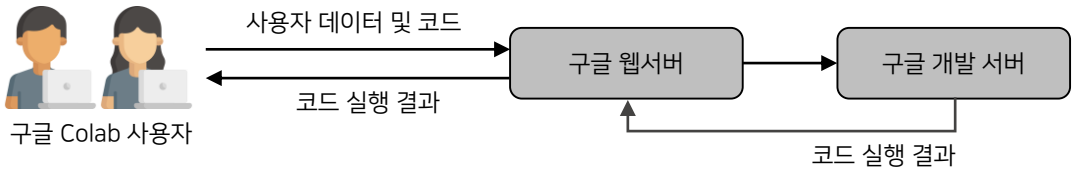


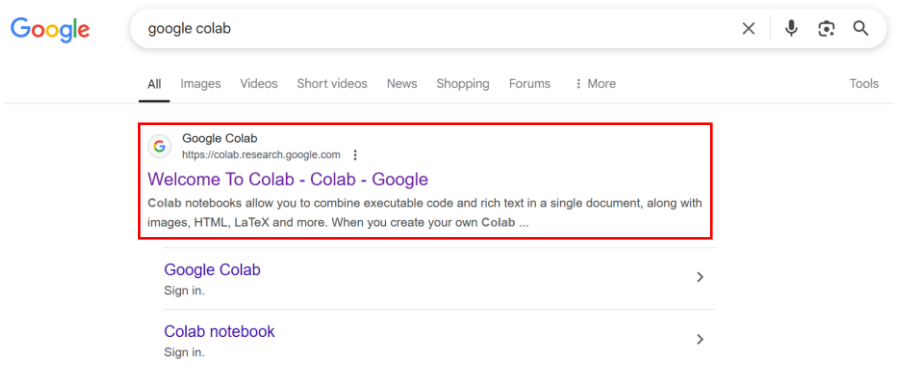
실습 1: 머신러닝 기반 예측 모델 구축

구글 Colab을 이용한 개발 환경 구축 및 머신러닝 기반 예측 모델 구축 실습

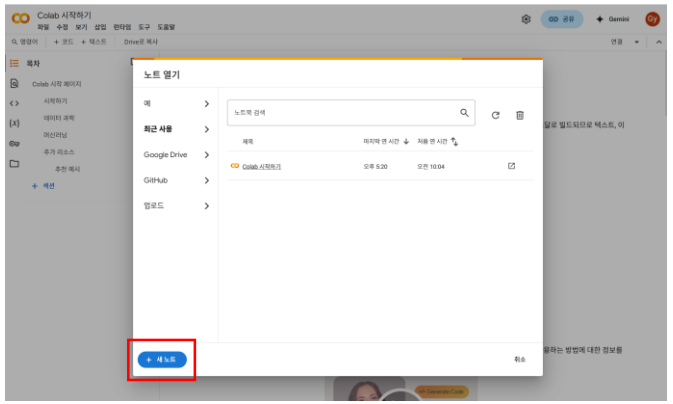
구글 Colab과 로컬 개발환경의 특징 비교

	구글 Colab	로컬 개발환경
파이썬 개발 환경	최신 파이썬 개발 환경 제공 (+)	사용하고자 하는 파이썬 개발 환경 선택 및 설치 필요 (-)
프로그램 설치	파이썬 엔진, 프로그래밍 에디터 설치 필요 없음 (+)	파이썬 엔진, 프로그래밍 에디터의 설치 필요 (-)
사용료	구글 정책에 따라 사용료 부과 (-)	없음 (+)
정보 보호	데이터 및 개발 코드가 구글 서버로 이동됨 (-)	개인 컴퓨터 내에서만 데이터와 코드가 저장됨 (+)
인터넷 사용	구글 서버에 접속하기 위해 인터넷 연결 필요 (-)	프로그래밍 작업에 대해서는 인터넷 연결 필요 없음 (+)





https://colab.research.google.com



실습 1: 머신러닝 기반 예측 모델 구축

구글 Colab을 이용한 개발 환경 구축 및 머신러닝 기반 예측 모델 구축 실습

Concrete compressive strength 데이터셋
(https://archive.ics.uci.edu/dataset/165/concrete+compressive+strength)

Cement (kg in a m^3 mixture)	Blast Furnace Slag (kg in a m^3 mixture)	Fly Ash (kg in a m^3 mixture)	Water (kg in a m^3 mixture)	Superplasticizer (kg in a m^3 mixture)	Coarse Aggregate (kg in a m^3 mixture)	Fine Aggregate (kg in a m^3 mixture)	Age (day)	Concrete compressive strength (MPa, megapascals)
540.0	0.0	0.0	162.0	2.5	1040.0	676.0	28	79.99
540.0	0.0	0.0	162.0	2.5	1055.0	676.0	28	61.89
332.5	142.5	0.0	228.0	0.0	932.0	594.0	270	40.27
332.5	142.5	0.0	228.0	0.0	932.0	594.0	365	41.05
198.6	132.4	0.0	192.0	0.0	978.4	825.5	360	44.30
266.0	114.0	0.0	228.0	0.0	932.0	670.0	90	47.03
380.0	95.0	0.0	228.0	0.0	932.0	594.0	365	43.70
380.0	95.0	0.0	228.0	0.0	932.0	594.0	28	36.45
266.0	114.0	0.0	228.0	0.0	932.0	670.0	28	45.85
475.0	0.0	0.0	228.0	0.0	932.0	594.0	28	39.29
198.6	132.4	0.0	192.0	0.0	978.4	825.5	90	38.07
198.6	132.4	0.0	192.0	0.0	978.4	825.5	28	28.02
427.5	47.5	0.0	228.0	0.0	932.0	594.0	270	43.01
190.0	190.0	0.0	228.0	0.0	932.0	670.0	90	42.33
304.0	76.0	0.0	228.0	0.0	932.0	670.0	28	47.81

데이터

입력 변수: x

출력 변수: y

Cement: This feature represents the amount of cement used in the concrete mix. Cement is a crucial component of concrete and plays a significant role in determining its strength and durability.

Blast Furnace Slag: This feature represents the proportion of blast furnace slag in the concrete mix. Blast furnace slag is a byproduct of the iron and steel manufacturing process and can be used as a supplementary cementitious material in concrete.

Fly Ash: This feature represents the amount of fly ash in the concrete mix. Fly ash is another supplementary cementitious material that can improve the workability and durability of concrete.

Water: This feature represents the amount of water used in the concrete mix. The water-cement ratio is a critical factor in concrete design, influencing both strength and workability.

Superplasticizer: Superplasticizers are chemical additives used to improve the flow and workability of concrete. This feature indicates the amount of superplasticizer added to the mix.

Coarse Aggregate: Coarse aggregate consists of larger particles, such as gravel or crushed stone, and is used in concrete to provide bulk and strength.

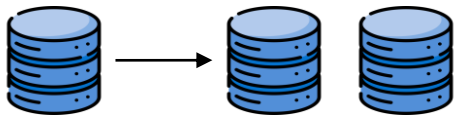
Fine Aggregate: Fine aggregate consists of smaller particles, such as sand, and is used to fill the voids between coarse aggregate particles and improve the workability of the mix.

Age: This feature represents the age of the concrete samples. The age of concrete can influence its strength, with strength typically increasing with time due to curing and hydration processes.

Target Variable:

Strength: The target variable, strength, represents the compressive strength of the concrete. Compressive strength is a critical property of concrete and measures its ability to withstand axial loads (e.g., weight or pressure) without failing. It is a crucial indicator of the quality and performance of concrete in various applications.

데이터셋으로부터 학습 및 평가 (test) 데이터셋 구성




```
dataset = pandas.read_excel('concrete_strength.xlsx').values.tolist()
random.shuffle(dataset)
dataset_train = numpy.vstack(dataset[:900])
dataset_test = numpy.vstack(dataset[900:])

print('Shape of the training dataset: {}'.format(dataset_train.shape))
print(dataset_train)
```

```
Shape of the training dataset: (900, 9)
[[251.37         0.         118.27       ... 757.73         3.
   17.22311048]
 [290.35         0.          96.18       ... 865.          56.
   45.08483564]
 [424.          22.         132.         ... 750.          3.
   32.01138572]
 ...
 [165.         128.5        132.1        ... 746.6         3.
   19.41564416]
 [159.8        250.           0.         ... 688.2        28.
   39.45595358]
 [251.81         0.          99.94       ... 899.76      100.
   45.3675208 ]]
```

실습 1: 머신러닝 기반 예측 모델 구축

구글 Colab을 이용한 개발 환경 구축 및 머신러닝 기반 예측 모델 구축 실습

Concrete compressive strength 데이터셋
(https://archive.ics.uci.edu/dataset/165/concrete+compressive+strength)

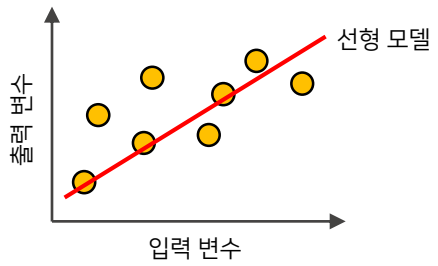
Cement (kg in a m^3 mixture)	Blast Furnace Slag (kg in a m^3 mixture)	Fly Ash (kg in a m^3 mixture)	Water (kg in a m^3 mixture)	Superplasticizer (kg in a m^3 mixture)	Coarse Aggregate (kg in a m^3 mixture)	Fine Aggregate (kg in a m^3 mixture)	Age (day)	Concrete compressive strength (MPa, megapascals)
540.0	0.0	0.0	162.0	2.5	1040.0	676.0	28	79.99
540.0	0.0	0.0	162.0	2.5	1055.0	676.0	28	61.89
332.5	142.5	0.0	228.0	0.0	932.0	594.0	270	40.27
332.5	142.5	0.0	228.0	0.0	932.0	594.0	365	41.05
198.6	132.4	0.0	192.0	0.0	978.4	825.5	360	44.30
266.0	114.0	0.0	228.0	0.0	932.0	670.0	90	47.03
380.0	95.0	0.0	228.0	0.0	932.0	594.0	365	43.70
380.0	95.0	0.0	228.0	0.0	932.0	594.0	28	36.45
266.0	114.0	0.0	228.0	0.0	932.0	670.0	28	45.85
475.0	0.0	0.0	228.0	0.0	932.0	594.0	28	39.29
198.6	132.4	0.0	192.0	0.0	978.4	825.5	90	38.07
198.6	132.4	0.0	192.0	0.0	978.4	825.5	28	28.02
427.5	47.5	0.0	228.0	0.0	932.0	594.0	270	43.01
190.0	190.0	0.0	228.0	0.0	932.0	670.0	90	42.33
304.0	76.0	0.0	228.0	0.0	932.0	670.0	28	47.81

데이터

입력 변수: x 출력 변수: y

```
dataset = pandas.read_excel('concrete_strength.xlsx').values.tolist()
random.shuffle(dataset)
dataset_train = numpy.vstack(dataset[:900])
dataset_test = numpy.vstack(dataset[900:])

print('Shape of the training dataset: {}'.format(dataset_train.shape))
print(dataset_train)
```

```
Shape of the training dataset: (900, 9)
[[251.37        0.        118.27      ... 757.73        3.
  17.22311048]
 [290.35        0.         96.18      ... 865.         56.
  45.08483564]
 [424.         22.        132.        ... 750.          3.
  32.01138572]
 ...
 [165.        128.5       132.1       ... 746.6         3.
  19.41564416]
 [159.8       250.         0.        ... 688.2        28.
  39.45595358]
 [251.81        0.        99.94      ... 899.76      100.
  45.3675208 ]]
```

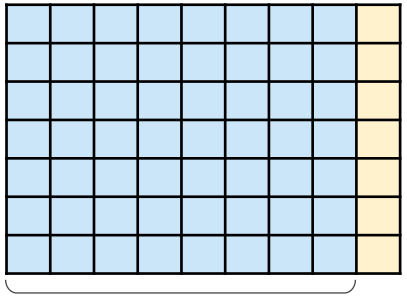


Scikit-learn

scikit-learn (https://scikit-learn.org) is a free and open-source machine learning library for Python programming environments.

```
lin_model = LinearRegression()
lin_model.fit(dataset_train[:, :-1], dataset_train[:, -1])
```

$$\theta = \underset{\theta \in \mathcal{P}}{\operatorname{argmin}}\, L(y, f(x;\theta)) \rightarrow \underset{\theta \in \mathcal{P}}{\operatorname{argmin}} \sum_{(x,y)\in\mathcal{D}} (y - f(x;\theta))^2$$



0~7열 (X[:, :-1])

8열 X[:, -1]

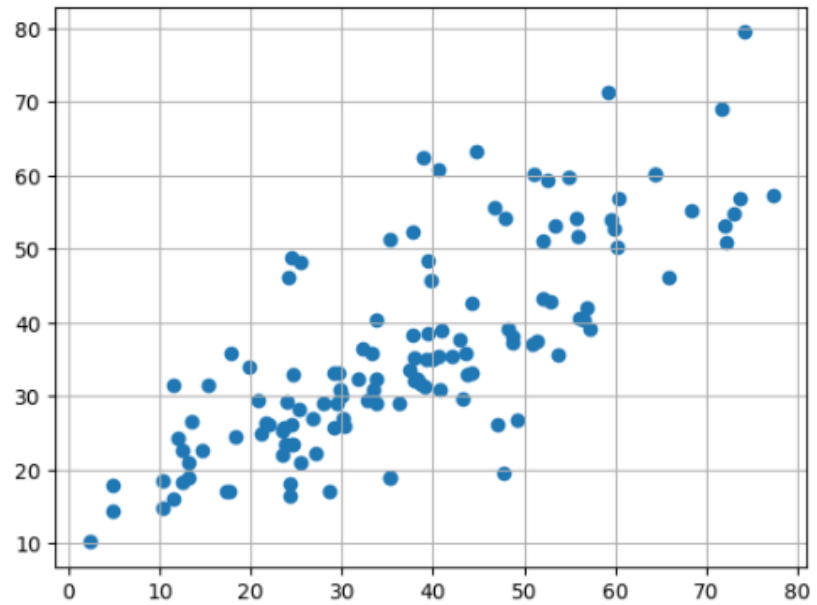
실습 1: 머신러닝 기반 예측 모델 구축

구글 Colab을 이용한 개발 환경 구축 및 머신러닝 기반 예측 모델 구축 실습

선형 회귀 모델 학습 및 concrete strength 예측 결과

```
preds_test = lin_model.predict(dataset_test[:, :-1])
mae_test = mean_absolute_error(dataset_test[:, -1], preds_test)
r2_test = r2_score(dataset_test[:, -1], preds_test)

print('Test MAE: {:.3f}₩tTest R2-score: {:.3f}'.format(mae_test, r2_test))
plt.scatter(dataset_test[:, -1], preds_test)
plt.grid()
plt.show()
plt.close()
```

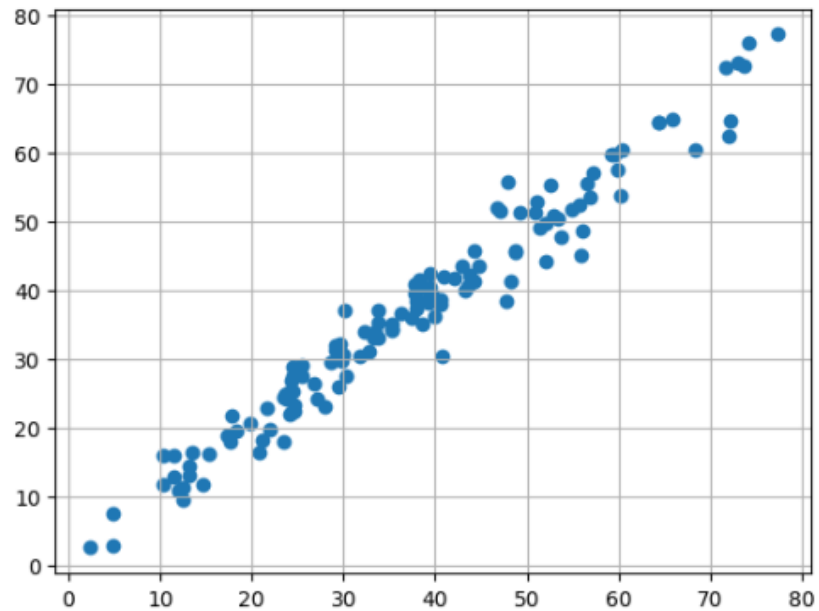
```
Test MAE: 8.504 Test R2-score: 0.604
```



XGBoost 모델 학습 및 concrete strength 예측 결과

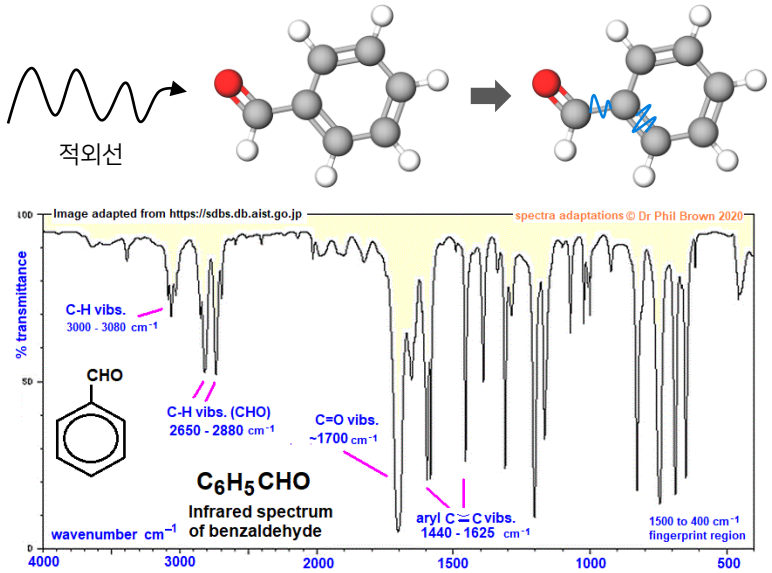
```
preds_test = xgb_model.predict(dataset_test[:, :-1])
mae_test = mean_absolute_error(dataset_test[:, -1], preds_test)
r2_test = r2_score(dataset_test[:, -1], preds_test)

print('Test MAE: {:.3f}₩tTest R2-score: {:.3f}'.format(mae_test, r2_test))
plt.scatter(dataset_test[:, -1], preds_test)
plt.grid()
plt.show()
plt.close()
```

```
Test MAE: 2.494 Test R2-score: 0.961
```



적외선 분광법 기반의 분자 구조 분석

적외선 분광법의 개념과 적외선 분광법 기반의 화합물 분석의 장점 및 단점



	Stretching Vibrations			Bending Vibrations		
Functional Class	Range (cm^{-1})	Intensity	Assignment	Range (cm^{-1})	Intensity	Assignment
Alkanes	2850-3000	str	CH_3, CH_2 & CH 2 or 3 bands	1350-1470 1370-1390 720-725	med med wk	CH_2 & CH_3 deformation CH_3 deformation CH_2 rocking
Alkenes	3020-3100 1630-1680 1900-2000	med var str	=C-H & $=CH_2$ C=C C=C asymmetric stretch	880-995 780-850 675-730	str med med	=C-H & $=CH_2$ (out-of-plane bending) cis-RCH=CHR
Alkynes	3300 2100-2250	str var	C-H (usually sharp) C≡C (symmetry reduces intensity)	600-700	str	C-H deformation
Arenes	3030 1600 & 1500	var med-wk	C-H C=C (in ring) (2 bands) (3 if conjugated)	690-900	str-med	C-H bending & ring puckering
Alcohols & Phenols	3580-3650 3200-3550 970-1250	var str str	O-H (free), usually sharp O-H (H-bonded), usually broad C-O	1330-1430 650-770	med var-wk	O-H bending (in-plane) O-H bend (out-of- plane)
Amines	3400-3500 (dil. soln.) 3300-3400 (dil. soln.) 1000-1250	wk wk med	N-H, 2 bands N-H (2°-amines) C-N	1550-1650 660-900	med-str var	NH_2 scissoring (1°- amines) NH_2 & N-H wagging (shifts on H-bonding)

- 적외선 분광법 (Infrared spectroscopy)는 적외선 투과하고 이때 흡수된 빛을 기반으로 화합물의 원자 구조를 분석하는 방법이며, 유기 화합물을 분석하는 데 유용하게 활용 가능
- 적외선 분광법은 유기 화합물을 분석하는 데 매우 유용하지만, infrared spectroscopy absorption table을 참조하여 분석 전문가가 직접 스펙트럼의 각 영역을 해석해야하는 한계점이 있음

장점	단점
▪ 빠르게 분석 결과 (스펙트럼)를 얻을 수 있음 ▪ 비교적 높은 해상도 ▪ 분자 구조에 대한 다양한 정보 제공 ▪ 비교적 샘플을 원상태로 유지할 수 있음	▪ 혼합물에 대해서는 분석 한계점이 존재 ▪ IR 스펙트럼의 복잡도 ▪ 높은 IR 스펙트럼 해석 비용

MOSGEN: 미지 시료 분석을 위한 IR 스펙트럼 분석 인공지능

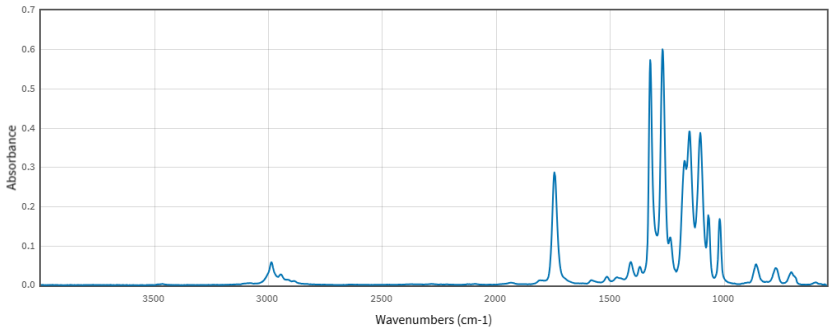
IR 스펙트럼 기반 미지 시료 분석을 위한 M-Order 스펙트럼 그래프 임베딩 네트워크

Paper: Na, G. S., & Rho, Y. C. (2024, March). In *2024 9th International Conference on Big Data Analytics (ICBDA)* (pp. 134-143). IEEE.

Source code: https://github.com/ngs00/mosgen

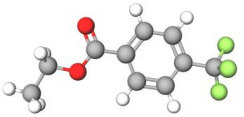
IR 스펙트럼 분석 문제	입력 데이터	출력 데이터	기존 방법론
소재군 분류 (Material class classification)	▪ IR 스펙트럼 ▪ 미지 시료에 대한 메타데이터	▪ 소재군 분류 확률 ▪ 소재군 분류 multiclass label	분석 전문가에 경험과 직관에 기반한 소재군 분류 [1, 2, 3]
하부 구조 감지 (Functional group detection)	▪ IR 스펙트럼 ▪ 미지 시료에 대한 메타데이터 ▪ 목적 하부 구조의 SMILES	▪ 목적 하부 구조의 존재 여부에 대한 binary label	분석 전문가의 IR spectrum table과 기존 문헌 분석을 통한 하부 구조 식별 [2, 4]
미지 시료 식별 (Compound identification)	▪ IR 스펙트럼 ▪ 미지 시료에 대한 메타데이터 ▪ 쿼리 값	▪ 미지 시료의 식별자	IR 스펙트럼 라이브러리 및 데이터베이스 검색 [2, 5]

미지 시료의 IR 스펙트럼



인공지능 모델 (MOSGEN)

- 소재군 분류: organic dyes, oils, mixtures, …
- 하부 구조 감지: aromatic, trifluoromethyl, …
- 미지 시료 식별: CCOC(=O)C1=CC=C(C=C1)C(F)(F)F

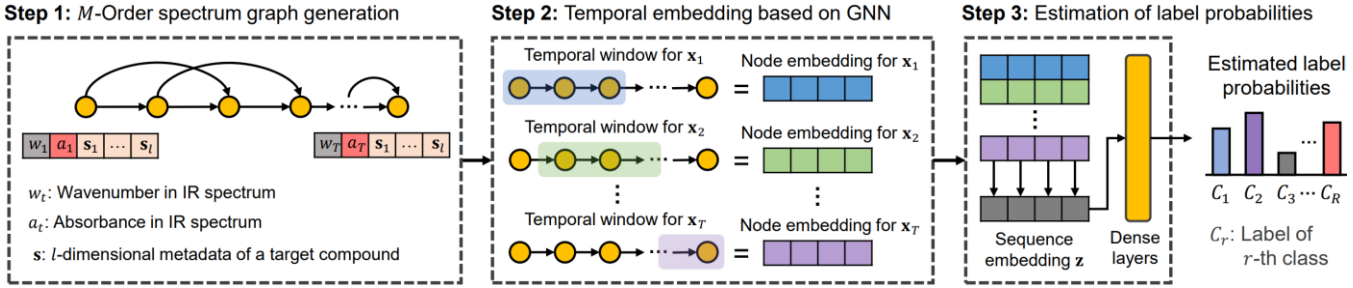


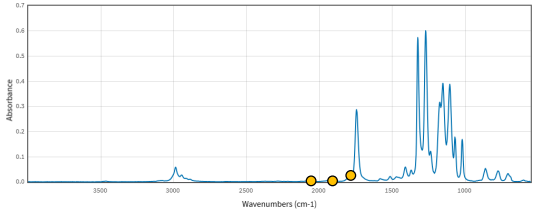
[1] Boulet-Audet, M., Vollrath, F., & Holland, C. (2015). Identification and classification of silks using infrared spectroscopy. *Journal of Experimental Biology*, *218*(19), 3138-3149.
[2] Ng, L. M., & Simmons, R. (1999). Infrared spectroscopy. *Analytical chemistry*, *71*(12), 343-350.
[3] Jacox, M. E. (2003). Vibrational and electronic energy levels of polyatomic transient molecules. Supplement B. *Journal of Physical and Chemical Reference Data*, *32*(1), 1.
[4] Jiang, W., Saxena, A., Song, B., Ward, B. B., Beveridge, T. J., & Myneni, S. C. (2004). Elucidation of functional groups on gram-positive and gram-negative bacterial surfaces using infrared spectroscopy. *Langmuir*, *20*(26), 11433-11442.
[5] Jamrógiewicz, M. (2012). Application of the near-infrared spectroscopy in the pharmaceutical technology. *Journal of pharmaceutical and biomedical analysis*, *66*, 1-10.

MOSGEN: 미지 시료 분석을 위한 IR 스펙트럼 분석 인공지능

IR 스펙트럼 기반 미지 시료 분석을 위한 M-Order 스펙트럼 그래프 임베딩 네트워크

M-order spectrum graph embedding network (MOSGEN)의 구조와 IR 스펙트럼 분석 과정

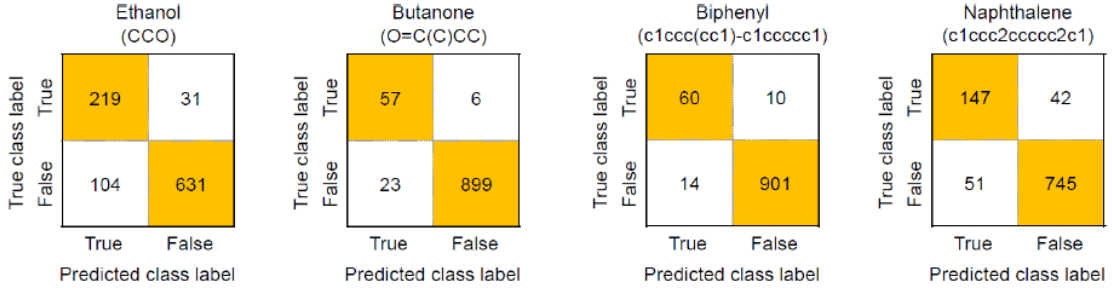




잠재 임베딩 (Latent embedding) 계산:

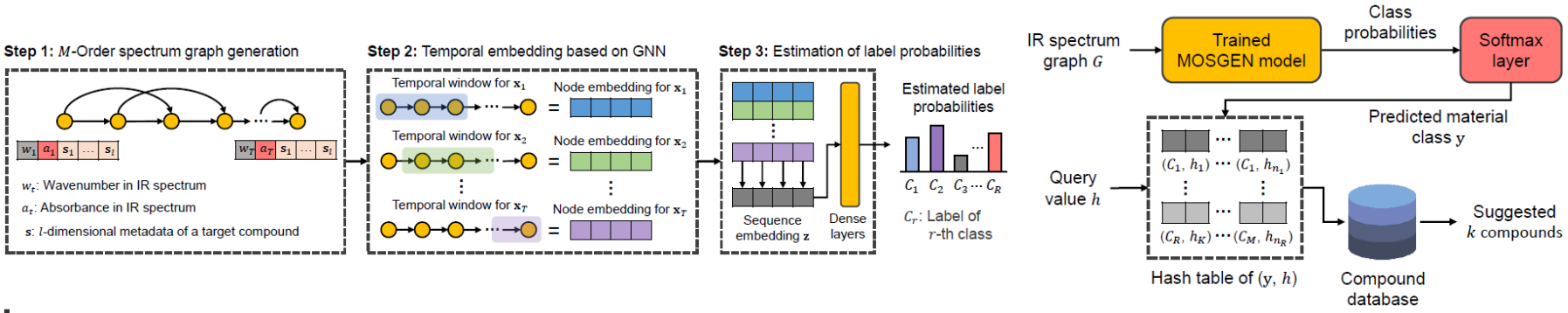
$$\mathbf{H}_i^{(k)} = \mathbf{W}_r^{(k)}\mathbf{H}_i^{(k-1)} + \sum_{j\in\mathcal{N}_i} \alpha_{i,j}^{(k)} \left(\mathbf{W}_a^{(k)}\mathbf{H}_j^{(k)} + \mathbf{W}_e^{(k)}\mathbf{E}_{i,j}\right)$$

- 미지 시료의 IR 스펙트럼으로부터 소재군 (organic dyes, minerals, carbohydrates, oils 등)을 분류하는 문제에서 MOSGEN은 88.60%의 예측 정확도를 달성
- 미지 시료의 IR 스펙트럼으로부터 특정 하부 구조 (ethanol, biphenyl 등)의 존재 여부를 예측하는 문제에서 MOSGEN은 76-83%의 예측 정확도를 달성



MOSGEN: 인공지능 기반 미지 시료 식별

미지 시료의 IR 스펙트럼으로부터 미지 시료를 식별하기 위한 인공지능



MOSGEN 기반 미지 시료 식별 과정

1. 학습된 MOSGEN을 이용하여 미지 시료의 IR 스펙트럼으로부터 미지 시료의 소재군을 예측
2. 분자량 등과 같은 미지 시료의 메타데이터에 해당하는 쿼리 값 (query value, h)을 입력
3. 예측된 소재군 label와 입력된 쿼리 값을 기반으로 IR 스펙트럼 데이터베이스에서 k개의 미지 시료 후보군을 도출

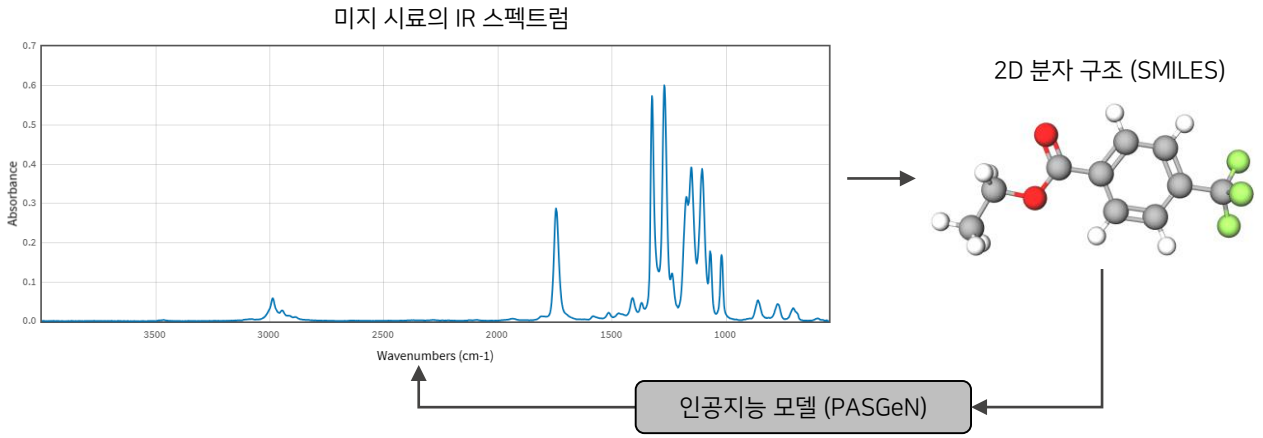
Material Class	# of materials	Baseline Accuracy	Compound Identification Accuracy (%)			
			$k=1$	$k=3$	$k=5$	$k=7$
Organic dyes & pigments (OD)	699	0.14%	28.65 (±2.38)	41.82 (±1.38)	48.97 (±1.36)	53.80 (±1.87)
Minerals & pigments (MP)	287	0.35%	71.17 (±6.25)	85.10 (±1.84)	89.33 (±4.09)	91.14 (±3.82)
Natural polymers (NP)	41	2.44%	61.45 (±7.08)	63.67 (±9.99)	66.39 (±6.17)	66.39 (±6.17)
Wax materials (WX)	13	7.69%	41.11(±8.39)	47.78 (±13.47)	47.78 (±13.47)	47.78 (±13.47)
Carbohydrates (CB)	29	3.45%	64.81 (±13.98)	75.13 (±9.80)	79.89 (±3.00)	79.89 (±3.00)
Oils & fats (OF)	44	2.27%	58.33 (±14.14)	60.29 (±16.84)	64.52 (±10.77)	68.23 (±7.59)
Proteinaceous materials (PR)	12	8.33%	62.70 (±11.25)	68.25 (±2.75)	71.03 (±4.18)	71.03 (±4.18)
Synthetic polymers (SP)	51	1.96%	73.35 (±5.80)	78.86 (±8.57)	78.86 (±8.57)	84.46 (±6.53)
Mixtures (MX)	62	1.61%	66.23 (±8.72)	74.98 (±5.72)	80.99 (±5.21)	80.99 (±5.21)

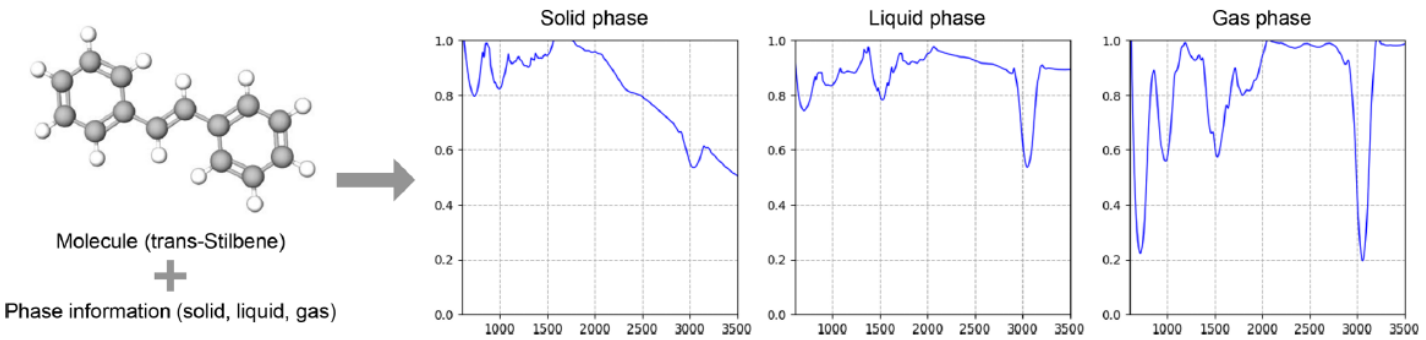
PASGeN: 인공지능 기반 IR 스펙트럼 생성

시료의 상태 정보를 고려한 인공지능 기반 IR 스펙트럼 생성 모델

Paper: Na, G. S. (2024). Deep Learning for Generating Phase-Conditioned Infrared Spectra. *Analytical Chemistry*, *96*(49), 19659-19669.

Source code: https://github.com/ngs00/pasgen



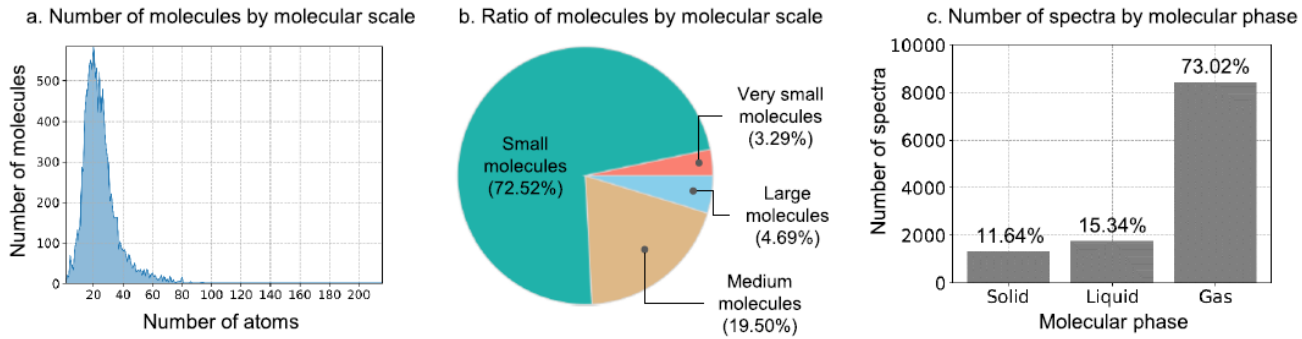


단일 분자 (CAS 번호: 103-30-0)의 상태 (기체, 액체, 고체)에 따른 IR 스펙트럼의 변화

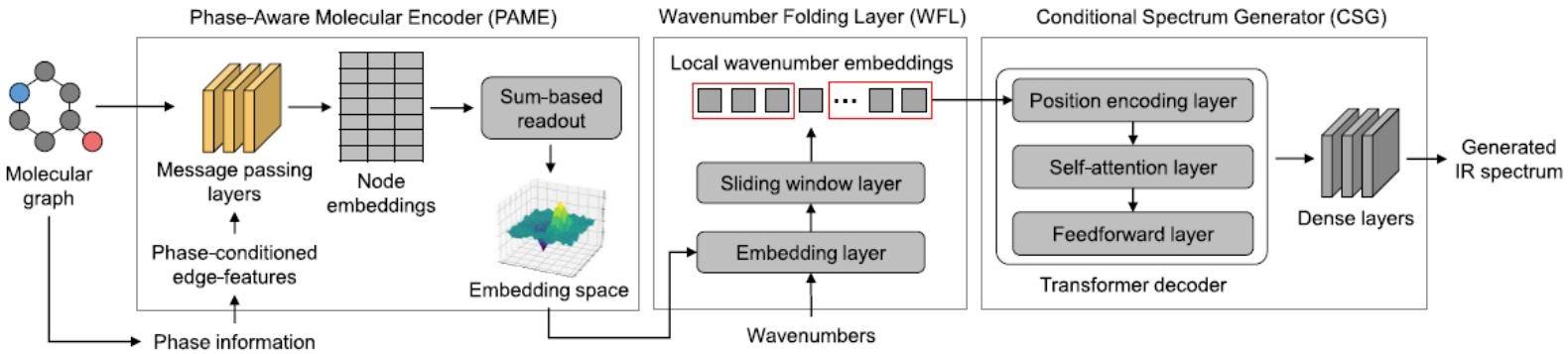
- 분자의 IR 스펙트럼은 분자 구조뿐만 아니라 분자의 상태에도 영향을 받음
- 기존 머신러닝 방법론들은 분자의 상태를 고려할 수 없는 한계점이 존재
- Multimodal 모델을 기반으로 구조와 상태를 모두 고려한 IR 스펙트럼 생성형 모델을 개발

PASGeN: 인공지능 기반 IR 스펙트럼 생성

시료의 상태 정보를 고려한 인공지능 기반 IR 스펙트럼 생성 모델



- NIST Chemistry Webbook 데이터베이스에서 10,288 분자에 대한 실험적으로 측정된 11,546개의 IR 스펙트럼을 수집하여 데이터셋을 구축
- 수집된 데이터셋은 (분자 SMILES, 분자 상태, IR 스펙트럼)의 데이터로 구성되었으며, 입력 데이터는 분자 SMILES와 분자 상태, 출력 데이터는 IR 스펙트럼
- 분자의 상태를 기체, 액체, 고체로 나누어서 데이터셋을 구축하였으며, 향후 연구에서는 상태의 정보를 더욱 세분화하는 것이 목표



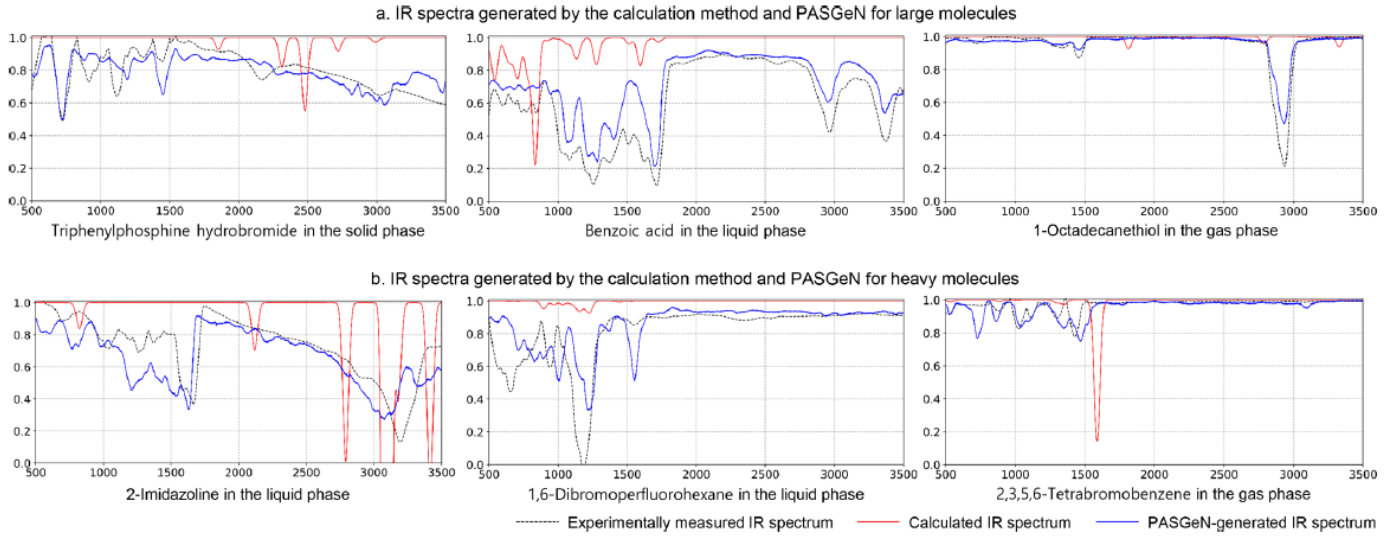
PASGeN의 구조 및 IR 스펙트럼 생성 과정

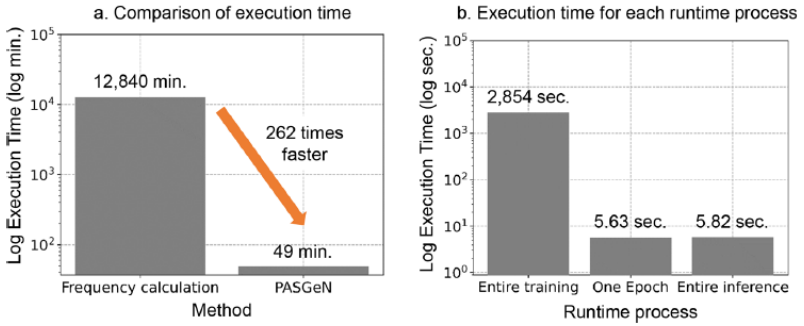
PASGeN: 인공지능 기반 IR 스펙트럼 생성

시료의 상태 정보를 고려한 인공지능 기반 IR 스펙트럼 생성 모델

	non-sequential model					sequential model		phase-aware sequential model
metric	GAT_{fc}	$D\text{-}MPNN_{fc}$	$UniMP_{fc}$	$MPNN_{fc}$	$AttFP_{fc}$	$MPNN_{tf}$	$AttFP_{tf}$	PASGeN
RMSE	0.121 (0.032)	0.117 (0.033)	0.123 (0.034)	0.118 (0.032)	0.121 (0.032)	0.112 (0.032)	0.114 (0.033)	**0.079 (0.024)**
RMSLE	0.109 (0.035)	0.105 (0.034)	0.108 (0.037)	0.096 (0.035)	0.099 (0.037)	0.094 (0.034)	0.096 (0.035)	0.063 (0.028)
Corr.	0.752 (0.057)	0.770 (0.054)	0.751 (0.060)	0.772 (0.061)	0.746 (0.060)	0.781 (0.058)	0.778 (0.055)	**0.895 (0.047)**

	all phases		solid phase		liquid phase		gas phase	
method	RMSE	corr.	RMSE	corr.	RMSE	corr.	RMSE	corr.
calculation method	0.146 (0.137)	0.239 (0.137)	0.385 (0.109)	0.227 (0.102)	0.267 (0.112)	0.260 (0.137)	0.090 (0.087)	0.235 (0.139)
PASGeN	0.085 (0.055)	0.908 (0.116)	0.129 (0.058)	0.749 (0.160)	0.103 (0.057)	0.873 (0.113)	0.070 (0.046)	0.924 (0.106)





- Corr. 기준 280% 향상된 IR 스펙트럼 생성 정확도
- 분자의 상태에 상관없이 0.75 이상의 IR 스펙트럼 생성 정확도 달성
- 262배 빨라진 IR 스펙트럼 생성 속도
- 인공지능 기반의 IR 스펙트럼 생성 완전 자동화

실습 2: 머신러닝 기반 IR 스펙트럼 분석을 위한 데이터 전처리

머신러닝을 위한 IR 스펙트럼 데이터셋 구축: JDX 파일 읽기, 빈 값 제거, 내삽 (Interpolation) 방법론

Read JDX file of 2-Fluoro-3-(trifluoromethyl)benzonitrile.

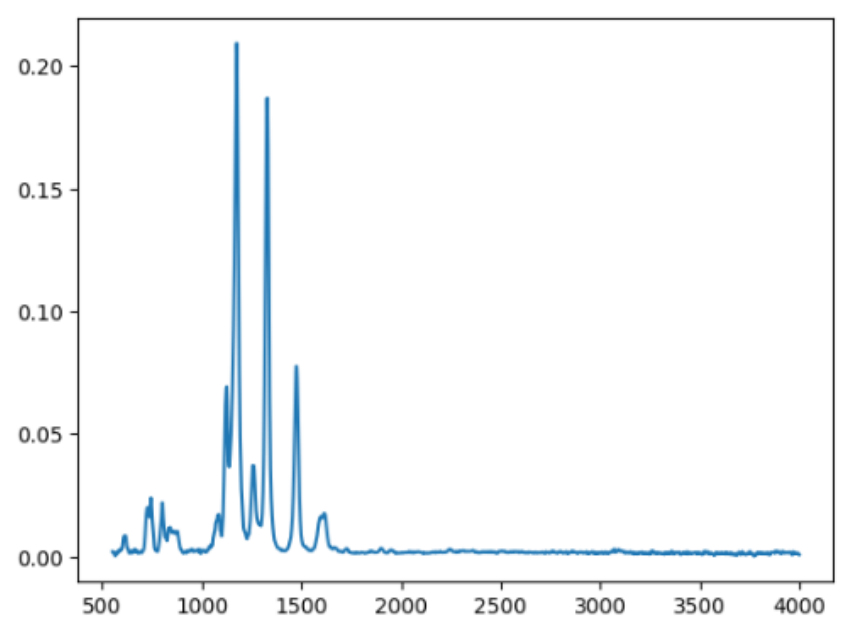
```
[3] jdx_file = 'dataset/2-Fluoro-3-(trifluoromethyl)benzonitrile.jdx'
    irs = jcamp.jcamp_readfile(jdx_file)
    for key in irs.keys():
        print('{}: {}'.format(key, irs[key]))
    wavenumber = irs['x']
    absorbance = irs['y']
```

```
title: 2-Fluoro-3-(trifluoromethyl)benzonitrile
jcamp-dx: 4.24
data type: INFRARED SPECTRUM
origin: EPA-IR VAPOR PHASE LIBRARY
owner: SRD/NIST
Collection (C) 2018 copyright by the U.S. Secretary of Commerce
on behalf of the United States of America. All rights reserved.
cas registry no: 146070-35-1
molform: C 8 H 3 F 4 N
$nist source: MSDC-IR
state: gas
xunits: 1/CM
yunits: ABSORBANCE
xfactor: 1.0
yfactor: 0.001
deltax: 1.92898
firstx: 549.8
lastx: 4000.7
firsty: 2.0496
maxx: 4000.7
minx: 549.759
maxy: 0.21129
miny: 0
npoints: 1790
xydata: (X++(Y..Y))
end:
x: [ 549.759  551.688  553.617 ... 3996.842 3998.771 4000.7  ]
y: [0.0020496 0.00212   0.0018984 ... 0.001088  0.000747  0.000762 ]
filename: dataset/2-Fluoro-3-(trifluoromethyl)benzonitrile.jdx
```

Perform an imputation method to fill missing values.

```
[7] absorbance = numpy.nan_to_num(absorbance, nan=0)
    f_interpol = interp1d(wavenumber, absorbance, kind='linear', fill_value='extrapolate')
    wavenumber = numpy.arange(550, 4000 + 2, step=2)
    absorbance = f_interpol(wavenumber)
    plt.plot(wavenumber, absorbance)
    plt.show()
    plt.close()
```

plt.gca().invert_xaxis()



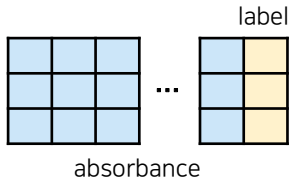
실습 3: 머신러닝 기반 IR 스펙트럼 분석

미지 시료의 IR 스펙트럼으로부터 Amide 작용기를 탐지하기 위한 인공지능 예측 모델 구축

Read the IR spectrum dataset.

```
[4]  with open('metadata.json', 'r') as f:
         metadata = json.load(f)

     # Target functional group: amide.
     target_fg = '[NX3][CX3](=[OX1])[#6]'
     train_x, train_y = load_dataset(metadata, 'dataset_train', target_fg)
     test_x, test_y = load_dataset(metadata, 'dataset_test', target_fg)
     print('Shape of the training dataset: {}'.format(train_x.shape))
     print('Shape of the test dataset: {}'.format(test_x.shape))
```

```
Shape of the training dataset: (97, 1726)
Shape of the test dataset: (18, 1726)
```



Train a classification model.

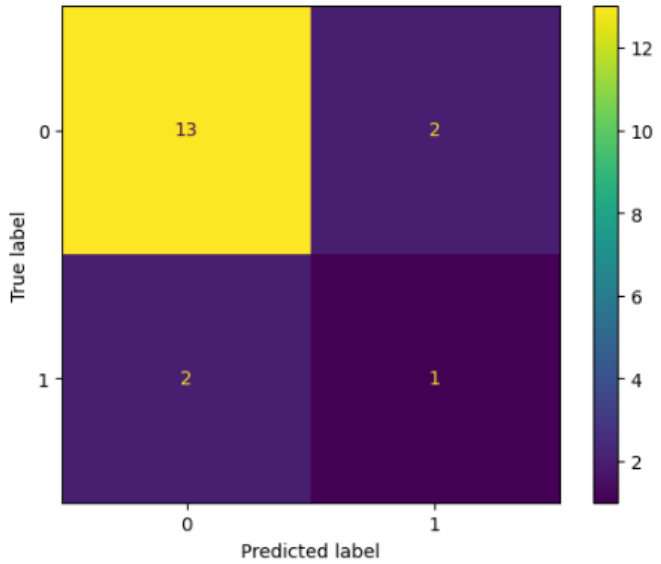
```
model = MLPClassifier(hidden_layer_sizes=128, batch_size=16, max_iter=500, verbose=True)
model.fit(train_x, train_y)
```

```
Iteration 1, loss = 0.69780196
Iteration 2, loss = 0.48049320
Iteration 3, loss = 0.43434489
Iteration 4, loss = 0.39896640
Iteration 5, loss = 0.35189613
Iteration 6, loss = 0.34553184
Iteration 7, loss = 0.30065601
Iteration 8, loss = 0.28947499
Iteration 9, loss = 0.26959577
Iteration 10, loss = 0.24446824
```

Evaluate the trained classification model.

```
preds = model.predict(test_x)
acc = numpy.round(numpy.mean(preds == test_y), 2)
f1 = numpy.round(f1_score(test_y, preds), 2)
print('Detection accuracy: {:.2f}'.format(acc))
print('F1-score: {:.2f}'.format(f1))

import matplotlib.pyplot as plt
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay
cm = confusion_matrix(test_y, preds)
disp = ConfusionMatrixDisplay(cm)
disp.plot()
plt.show()
plt.close()
```

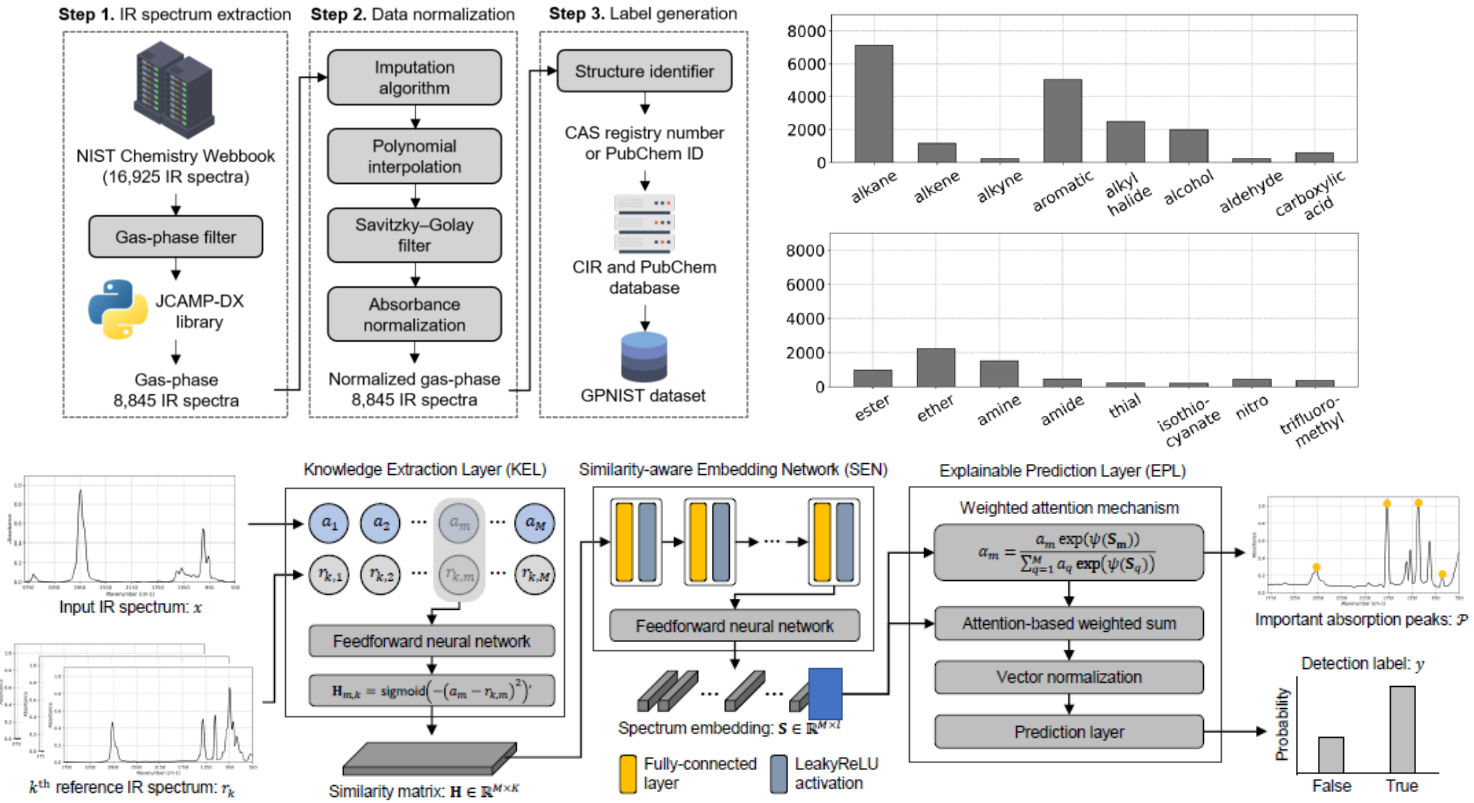
```
Detection accuracy: 0.78
F1-score: 0.33
```



SSIN: IR 스펙트럼 분석을 위한 해석 가능한 인공지능

해석 가능한 인공지능 기반의 IR 스펙트럼 분석 및 IR 스펙트럼 분석 보고서 생성

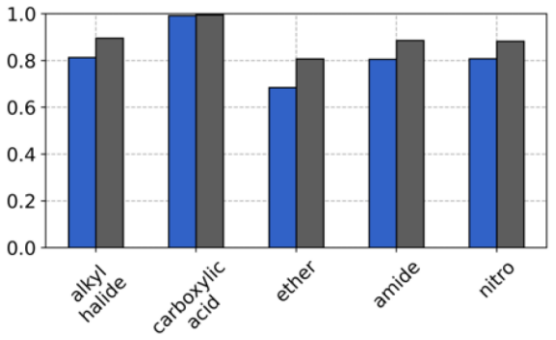
- 기존 인공지능의 한계점을 극복하고 해석 가능한 IR 스펙트럼 분석 결과를 제공하기 위한 substructure-directed spectrum interpreter network (SSIN)을 개발
- SSIN의 학습과 평가를 위해 NIST와 PubChem에서 8,814개의 IR 스펙트럼을 포함하는 GPNIST 데이터셋을 구축
- Alkyl halide, aldehyde, trifluoromethyl 등, 총 16개의 기능기에 대해 미지 시료의 IR 스펙트럼으로부터 특정 기능기의 존재 여부를 예측하는 SSIN을 개발



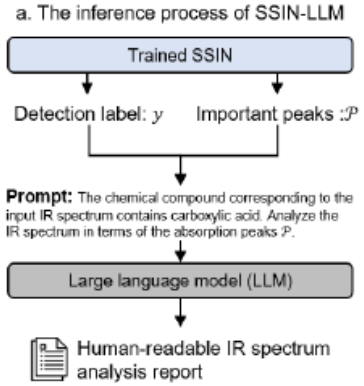
SSIN: IR 스펙트럼 분석을 위한 해석 가능한 인공지능

해석 가능한 인공지능 기반의 IR 스펙트럼 분석 및 IR 스펙트럼 분석 보고서 생성

Functional Group	SMARTS Pattern	Accuracy	Precision	Recall	F1-Score
alkane	[CX4;H3,H2]	0.972 (±0.003)	0.986 (±0.002)	0.980 (±0.004)	0.983 (±0.002)
alkene	[CX3]=[CX3]	0.967 (±0.004)	0.896 (±0.033)	0.822 (±0.022)	0.853 (±0.015)
alkyne	[CX2]#[CX2]	0.995 (±0.002)	0.938 (±0.029)	0.874 (±0.085)	0.912 (±0.050)
aromatic	[$([cX3](:*):*),$([cX2+](:*):*)]	0.977 (±0.004)	0.978 (±0.003)	0.981 (±0.007)	0.979 (±0.003)
alkyl halide	[#6][F,Cl,Br,I]	0.927 (±0.009)	0.889 (±0.023)	0.848 (±0.015)	0.858 (±0.017)
alcohol	[#6][OX2H]	0.979 (±0.003)	0.956 (±0.007)	0.952 (±0.008)	0.954 (±0.007)
aldehyde	[CX3H1](=O)[#6,H]	0.995 (±0.002)	0.922 (±0.045)	0.894 (±0.045)	0.908 (±0.040)
carboxylic acid	[CX3](=O)[OX2H]	0.987 (±0.002)	0.911 (±0.027)	0.899 (±0.032)	0.905 (±0.017)
ester	[#6][CX3](=O)[OX2H0][#6]	0.986 (±0.001)	0.930 (±0.014)	0.948 (±0.014)	0.938 (±0.004)
ether	[OD2]([#6])[#6]	0.967 (±0.004)	0.934 (±0.006)	0.936 (±0.009)	0.935 (±0.007)
amine	[NX3;H2,H1,H0;!$(NC=O)]	0.973 (±0.003)	0.934 (±0.014)	0.908 (±0.008)	0.920 (±0.006)
amide	[NX3][CX3](=[OX1])[#6]	0.984 (±0.004)	0.864 (±0.070)	0.842 (±0.043)	0.863 (±0.039)
thial	[CX3H1](=O)[#6,H]	0.996 (±0.001)	0.960 (±0.022)	0.880 (±0.056)	0.918 (±0.038)
difluoromethyl	[#6](-[#9])-[#9]	0.995 (±0.002)	0.925 (±0.027)	0.956 (±0.010)	0.940 (±0.017)
nitro	N(=O)(O)[#6]	0.995 (±0.001)	0.949 (±0.028)	0.957 (±0.023)	0.952 (±0.010)
trifluoromethyl	[#6](-[#9])(-[#9])-[#9]	0.994 (±0.003)	0.908 (±0.060)	0.963 (±0.008)	0.934 (±0.034)



- SSIN은 16개의 기능기를 감지하는 문제에서 85-98%의 정확도를 달성
- F, Cl, Br, I의 원소를 감지하는 문제에 대해서도 85%의 정확도를 달성
- Fingerprint region을 분석하는 문제에서도 65% 이상의 정확도를 달성



b. A human-readable IR spectrum analysis report and an evaluation process for the generated analysis report

A human-readable IR spectrum analysis report

Peak Range	Relevance	Reasons for Analysis
1784, 1816	Yes	Characteristic of the C=O stretching vibration in carboxylic acids.
1178, 1240	Yes	Corresponds to C-O stretching vibrations in carboxylic acids.
1050, 1068	No	Typically associated with C-O stretching in esters.

1. **[1784, 1816]**:
   - Characteristic absorption due to asymmetric C=O stretching in free carboxylic acids.
   - Observed peaks within this range confirm the presence of carboxylic acids.
2. **[1178, 1240]**:
   - This range is significant for C-O stretching vibrations typical in carboxylic acids.
   - The presence of peaks around these values suggests relevance to the C-O bond within the carboxyl group.
3. **[1050, 1068]**:
   - The range is typically associated with C-O stretching vibrations found in esters rather than carboxylic acids.
   - No observed peaks fall within this specific single-point range, indicating no direct relevance to the characteristic of a free carboxylic acid group.

An IR spectroscopy absorption table

Wavenumber (cm-1)	Bond	Functional Group
3300-2500	O-H stretch	carboxylic acid
1760-1690	C=O stretch	carboxylic acid
1320-1000	C-O stretch	carboxylic acid
950-910	O-H bend	carboxylic acid
⋮		
1175-1100	C-X bend	trifluoromethyl

Machine-generated reports: github.com/ngs00/SSIN/tree/main/save

Functional Group	Peak Identification Accuracy		Imp. (%)
	SSIN	SSIN-LLM	
alkane	0.823 (±0.003)	0.991 (±0.003)	20.413
alkene	0.911 (±0.002)	0.923 (±0.003)	1.317
aromatic	0.904 (±0.003)	0.962 (±0.003)	6.156
alkyl halide	0.813 (±0.011)	0.898 (±0.008)	10.455
alcohol	0.854 (±0.006)	0.993 (±0.003)	16.276
carboxylic acid	0.992 (±0.003)	0.998 (±0.003)	0.605
ester	0.879 (±0.016)	0.909 (±0.015)	3.413
ether	0.685 (±0.005)	0.879 (±0.008)	28.321
amine	0.811 (±0.016)	0.881 (±0.023)	8.631
amide	0.806 (±0.016)	0.815 (±0.025)	1.117
difluoromethyl	0.904 (±0.027)	0.921 (±0.032)	1.881
nitro	0.808 (±0.028)	0.932 (±0.016)	15.347
trifluoromethyl	0.946 (±0.021)	0.976 (±0.016)	3.171