<제2회 신약개발 AI 경진대회>
IRAK4 IC50 활성 예측 모델 개발

Team Nabi (구나영, 전재영, 정한영, 최윤영, 권순준)

히브리어 נָבִיא (나비; 예언자라는 뜻)

Date: 2024.10.11

Private score 3등

목차

IRAK4 IC50 활성 예측 모델 개발

#Machine Learning #Deep Learning # Smiles data
#Feature extraction & selection #IRAK4

Team Nabi

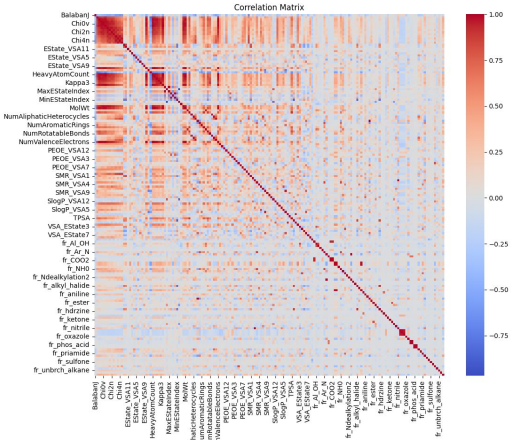
1. Exploratory Data Analysis

1. 데이터에서 주어진 Label 특성 살펴보기
2. EDA를 통해 세운 Validation Methods
3. 그 외 시도해본 EDA

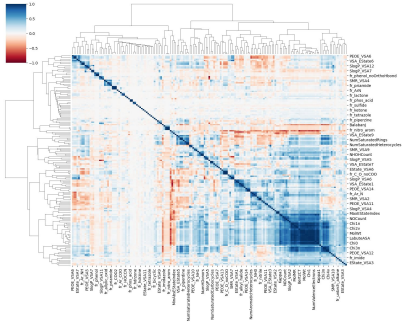
1. > EDA
2. 데이터 전략
3. 모델링 전략
4. 실험 관리
5. 범용성

EDA

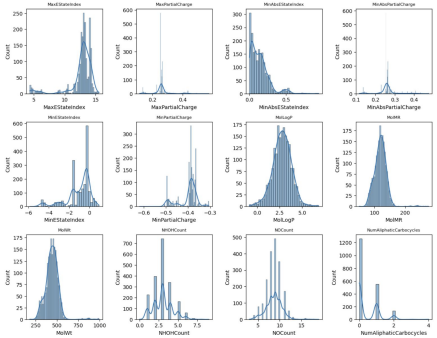
Correlation between each descriptor



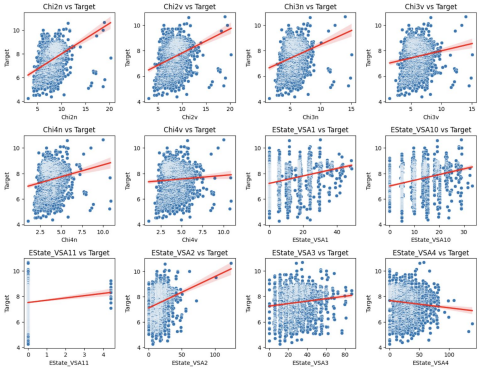
Cluster map of descriptor



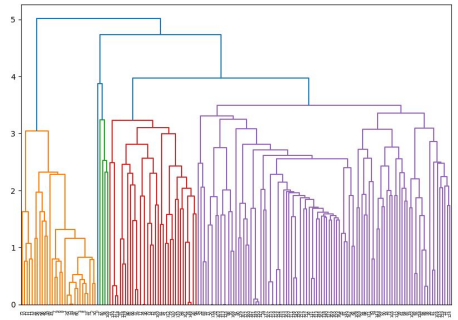
Histogram for each descriptor



Scatter plot of each descriptor vs target



Dendrogram of descriptor features

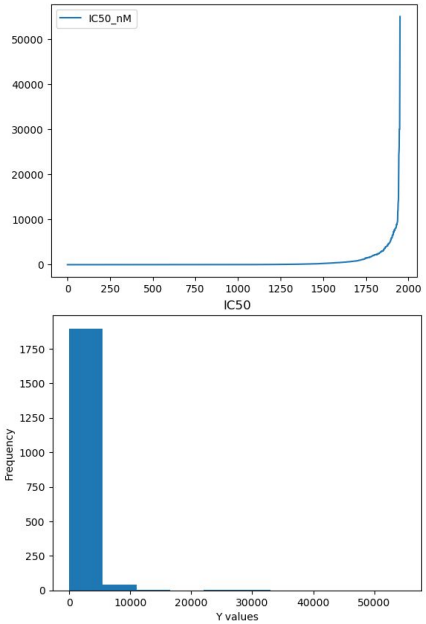


Exploratory Data Analysis, Target values

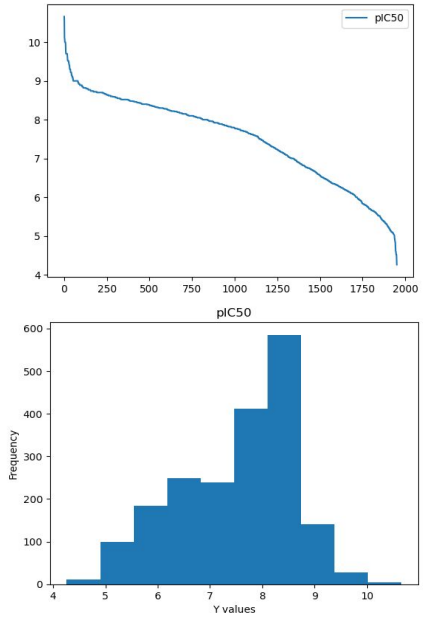
Target 선정: log(IC50)

- 분포의 균형
- Score의 pic50보다 nRMSE를 줄이기 위해 log(IC50)이 유리하다고 판단

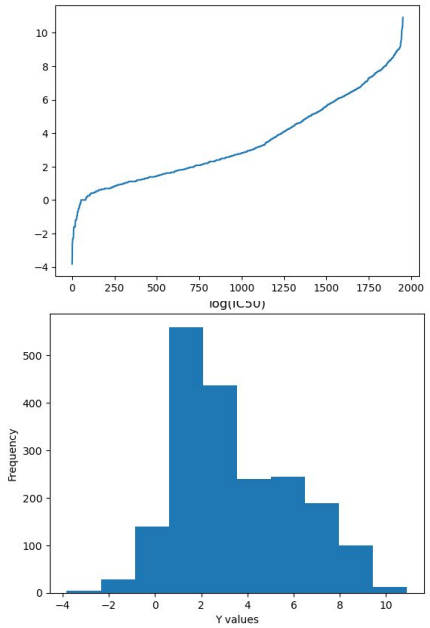
IC50



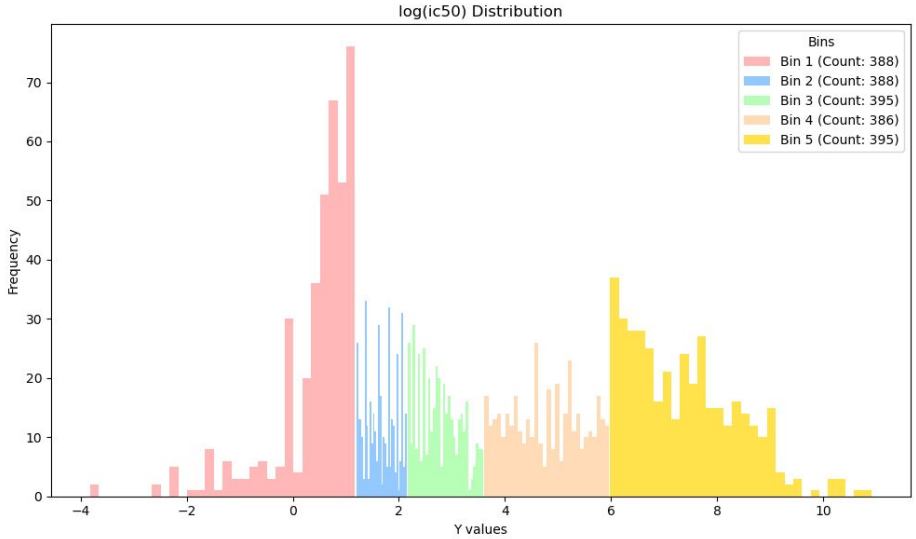
pIC50



log(IC50)



Validation Methods

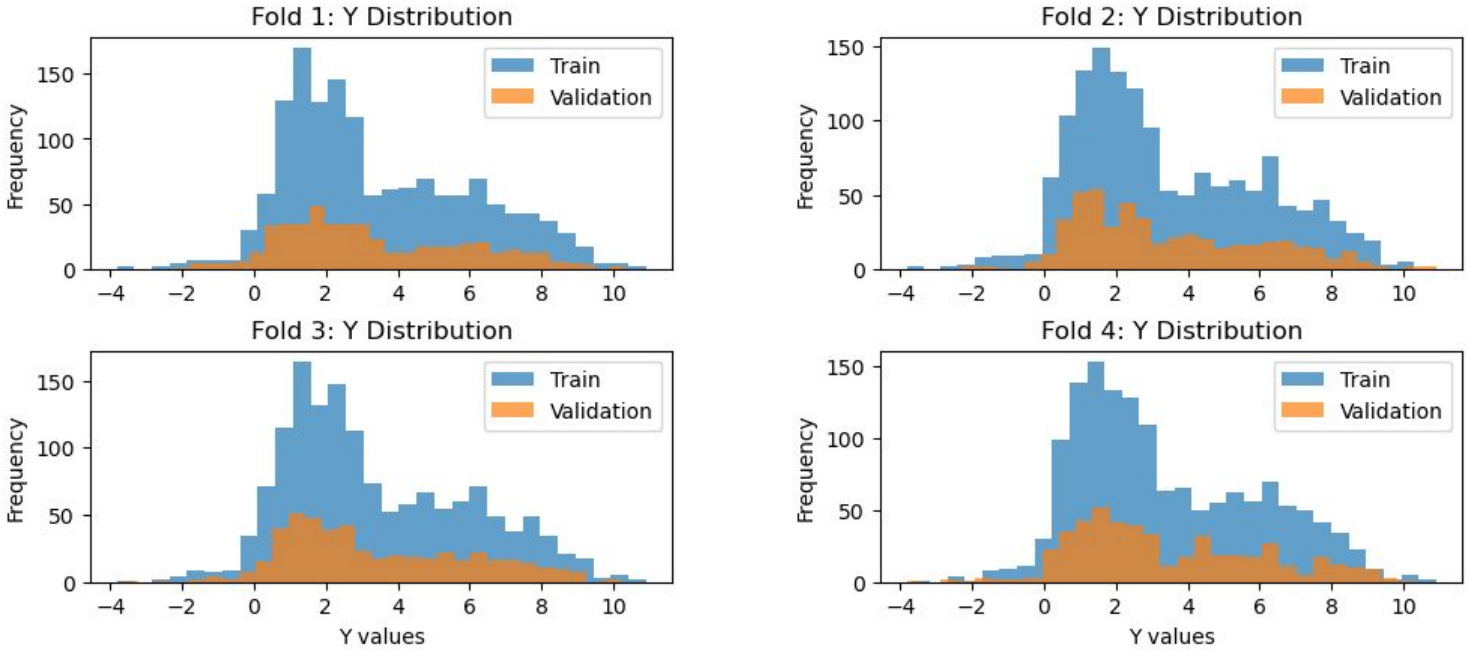


Stratified 5-Fold Cross Validation

- **잠재적인 위험:** train/validation 세트를 나눌 때, 큰 값이 train 세트에 집중되고 작은 값이 validation 세트에 집중될 수 있음
- **적용:** log(IC50) 을 기준으로 5개의 bin을 할당, 이를 이용하여 Stratified K-fold 수행
  - log(IC50) 값이 각 Fold에 골고루 분포되도록 함

Stratified K-Fold: Fold 별 데이터 분포

훈련/검증 전에
실제로 fold별 분포도 확인

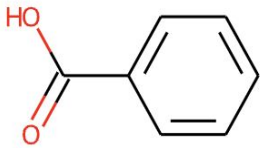
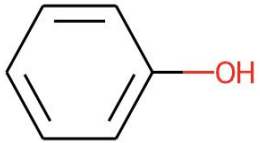


2. 데이터 전략: Features Extraction

1. Descriptors
2. Morgan Fingerprint
3. Infomax
4. Deep Purpose
5. Language Models

SMILES representation

SMILES: Simplified Molecular Input Line Entry System

Example Molecules	 Benzoic Acid	 Phenol
SMILES	c1cc(C(O)=O)ccc1	c1cc(O)ccc1

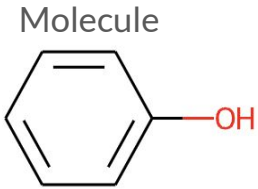
Molecule ChEMBL ID	Smiles
CHEMBL4443947	CN[C@@H](C)C(=O)N[C@H](C(=O)N1C[C@@H](NC(=O)CCO...
CHEMBL4556091	CC(C)(O)[C@H](F)CN1Cc2cc(NC(=O)c3cnn4cccnc34)c(N3C...
CHEMBL4566431	CC(C)(O)[C@H](F)CN1Cc2cc(NC(=O)c3cnn4cccnc34)c(N3C...
CHEMBL4545898	CC(C)(O)[C@H](F)CN1Cc2cc(NC(=O)c3cnn4cccnc34)c(N3C...
CHEMBL4448950	COc1cc2c(OC[C@@H]3CCC(=O)N3)ncc(C#CCCCCCCCCCC...
CHEMBL4445098	CC(C)(O)[C@H](F)CN1Cc2cc(NC(=O)c3cnn4cccnc34)c(N3C...
CHEMBL5190644	COc1cc2nn([C@H]3CC[C@@]4(CCC(=O)N4C)CC3)cc2cc1C(...
CHEMBL4066705	CC[C@H]1[C@@H](COc2nccc3cc(C(N)=O)c(OC)cc23)NC(=O...
CHEMBL4568894	CC(C)(O)[C@H](F)CN1Cc2cc(NC(=O)c3cnn4cccnc34)c(N3C...
CHEMBL4472406	CC(C)(O)CCN1Cc2cc(NC(=O)c3cnn4cccnc34)c(N3CCOCC3)...

Molecular Featurization

- **Molecular Featurization:** 분자를 벡터로 변환하는 방법; 보편적으로 신뢰할 수 있는 기본 방법이 존재하지는 않음.
- **Task-Specific Representation:** 과제의 특성에 따라 분자 표현 방식이 다르므로, 상황에 맞는 접근 필요
- **Various Approaches:**
  - Structural Fingerprints
  - Physicochemical Descriptors
  - Pre-trained Embeddings

Dataset	Metric	Representation	Score	Rank
Lipophilicity	MAE ↓	ECFP	0.727	1
		Mordred	0.579	0
		ChemBERTa	0.740	2
ClinTox	AUROC ↑	ECFP	0.535	2
		Mordred	0.563	1
		ChemBERTa	0.665	0

Representations



SMILES
c1cc(O)ccc1

A Expert-based Representations

A1 Fingerprints

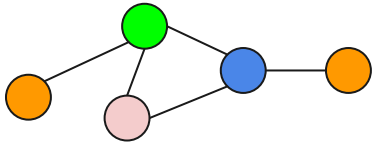
0	0	1	0	0	...

A2 Descriptors

LogP	tSP	MolWt	HBA	HBD	...
2.1	12	34.03	4	0	...

B Learned Representations

B1 Graphs



B2 Language Models (e.g. Transformers)

2.3	1.1	0.2	4.6	7.0	...

Embedding

Generating Features Using SMILES

대회 제공
데이터

Molecule ChEMBL ID	Smiles
CHEMBL4443947	CN[C@@H](C)C(=O)N[C@H](C(=O)N1C[C@@H](NC(=O)CCO...
CHEMBL4556091	CC(C)(O)[C@H](F)CN1Cc2cc(NC(=O)c3cnn4cccnc34)c(N3C...
CHEMBL4566431	CC(C)(O)[C@H](F)CN1Cc2cc(NC(=O)c3cnn4cccnc34)c(N3C...
CHEMBL4545898	CC(C)(O)[C@H](F)CN1Cc2cc(NC(=O)c3cnn4cccnc34)c(N3C...
CHEMBL4448950	COc1cc2c(OC[C@@H]3CCC(=O)N3)ncc(C#CCCCCCCCCCC...
CHEMBL4445098	CC(C)(O)[C@H](F)CN1Cc2cc(NC(=O)c3cnn4cccnc34)c(N3C...
CHEMBL5190644	COc1cc2nn([C@H]3CC[C@@]4(CCC(=O)N4C)CC3)cc2cc1C(...
CHEMBL4066705	CC[C@H]1[C@@H](COc2nccc3cc(C(N)=O)c(OC)cc23)NC(=O...
CHEMBL4568894	CC(C)(O)[C@H](F)CN1Cc2cc(NC(=O)c3cnn4cccnc34)c(N3C...
CHEMBL4472406	CC(C)(O)CCN1Cc2cc(NC(=O)c3cnn4cccnc34)c(N3CCOCC3)...

SMILES

A2 **Descriptors**
: chemical features of each molecules

A1 **Morgan Fingerprint**
: binary vector containing molecular structures

B1 **Infomax**
: fingerprints generated using a pre-trained Deep Graph

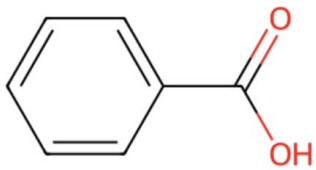
B2 **Language Models**
: transformer for generative molecular modeling

C **DeepPurpose**
: binding affinity value with IRAK4

Feature 1. Descriptors

- 분자의 특정 성질 또는 종합적인 구조/성질을 숫자로 나타내는 것
- RDKit Library에서 제공하는 195개 descriptor중, 모든 값이 0인 column 을 제외한 170개만 사용

<Descriptor 예시>



c1ccccc1C(=O)O

ring의 개수

```
Descriptors.RingCount(m)
```

```
1
```

sp3 carbon의 fraction

```
Descriptors.FractionCSP3(m)
```

```
0.0
```

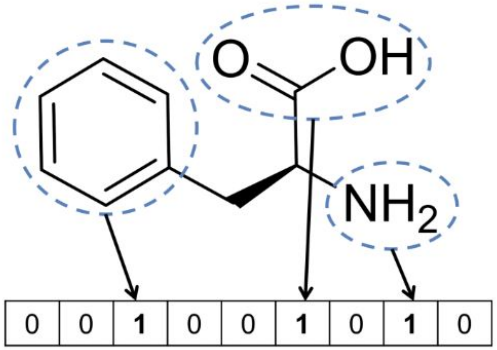
<실제 사용한 descriptor 데이터>

	BalabanJ	BertzCT	Chi0	Chi0n	Chi0v	Chi1	Chi1n	Chi1v	Chi2n	Chi2v	...
0	-2.586823	5.382888	7.337746	7.621311	7.531018	7.619835	7.640503	7.121666	6.305646	5.727457	...
1	-1.002407	0.968938	1.289540	1.475256	1.365800	1.333442	1.627839	1.380010	1.885683	1.556296	...
2	-1.002407	0.981288	1.289540	1.380477	1.270726	1.333442	1.411350	1.173279	1.656705	1.340207	...
3	-0.457864	0.873469	1.439189	1.266490	1.156383	1.264314	1.159783	0.933050	1.345021	1.046068	...
4	-2.456645	5.316561	6.597573	6.887633	7.054800	6.532472	6.642241	6.598663	6.126237	6.027605	...
...	...	...	...	...	...	...	...	...	...	...	...
1947	0.993376	-1.675671	-1.901542	-2.235206	-2.356231	-1.977071	-2.267563	-2.339815	-2.256723	-2.352932	...
1948	1.312270	-1.372236	-1.540258	-1.501693	-1.379950	-1.568864	-1.390196	-1.317120	-1.575693	-1.470538	...
1949	1.105465	-1.128628	-1.456906	-2.003264	-2.123565	-1.665731	-2.089609	-2.169882	-2.019948	-2.129485	...
1950	-0.201819	0.056209	-0.047073	-0.179904	-0.034771	0.146499	-0.008411	0.216884	-0.149697	0.201543	...
1951	5.082167	-3.584345	-3.172408	-3.131828	-3.255648	-3.533506	-3.205962	-3.235918	-2.953110	-3.010119	...

1952 rows × 170 columns

Feature 2. Morgan Fingerprint

- 분자구조를 정해진 차원의 정수 벡터로 변환
- 장점: 동일하거나 비슷한 분자 구조를 빠르게 검색 가능



Bajusz, D., Rácz, A., & Héberger, K. (2017). Chemical Data Formats, Fingerprints, and Other Molecular Descriptions for Database Analysis and Searching, Compr. Med. Chem. III, 3, 8.

```python
from rdkit import Chem
from rdkit.Chem import AllChem

# SMILES 데이터를 분자 지문으로 변환
def smiles_to_fingerprint(smiles):
    mol = Chem.MolFromSmiles(smiles)
    if mol is not None:
        fp = AllChem.GetMorganFingerprintAsBitVect(mol, 2, nBits=CFG['NBITS'])
        return np.array(fp)
    else:
        return np.zeros((CFG['NBITS'],))
```

RDKit Library 사용 (Cheminformatics)

	0	1	2	3	4	5	6	7	8	9	...	2038	2039	2040	2041	2042	2043	2044	2045	2046	2047
0	1	1	1	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	1	...	1	0	0	0	0	0	0	0	0	0
2	0	1	0	0	0	0	0	0	0	1	...	1	0	0	0	0	0	0	0	0	0
3	0	1	0	0	0	0	0	0	0	1	...	1	0	0	0	0	0	0	0	0	0
4	1	1	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1947	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1948	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1949	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1950	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1951	0	1	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0

1952 rows × 2048 columns

Feature 3. Infomax

- 300 bit 길이
- **Molecular Graph** 기반 Feature 생성
- Regression Performance 향상에 효과적이라는 선행 연구 존재 (Zagidullin et al., 2021)

	0	1	2	3	4	5	6	7	8	9	...
0	0.126515	-0.070788	-0.125801	-0.101101	-0.511209	-0.032890	0.021499	-0.063034	-0.106494	-0.069529	...
1	0.007390	-0.191097	0.201529	-0.104568	0.019750	0.009533	0.030039	-0.001289	0.039038	0.182121	...
2	0.117358	-0.079716	0.311567	-0.078423	-0.129427	0.011311	-0.029453	-0.068265	0.053194	0.196659	...
3	0.001371	-0.177162	0.069632	0.031912	0.153429	0.009424	-0.039644	0.020556	0.055848	0.070463	...
4	-0.074349	-0.083011	0.082345	-0.000260	0.072067	0.006270	-0.003263	-0.017865	0.087652	0.081751	...
...	...	...	...	...	...	...	...	...	...	...	...
108	0.016622	-0.210138	0.202166	-0.176142	0.079797	0.010415	0.066732	0.005604	0.018268	0.153305	...
109	0.041986	-0.051799	-0.029769	-0.347117	-0.378878	-0.033246	-0.107170	-0.103579	0.035946	0.174398	...
110	0.099336	-0.044724	0.323531	-0.057606	0.082990	0.010310	0.033376	-0.054251	0.093466	0.147529	...
111	0.029937	-0.029792	0.226532	-0.049432	0.100409	0.020880	0.009613	0.032810	0.187503	0.164056	...
112	0.053191	-0.153356	0.090668	-0.034836	0.128956	0.017058	-0.110226	-0.022378	0.034524	0.165198	...

113 rows × 300 columns → 300 bits long

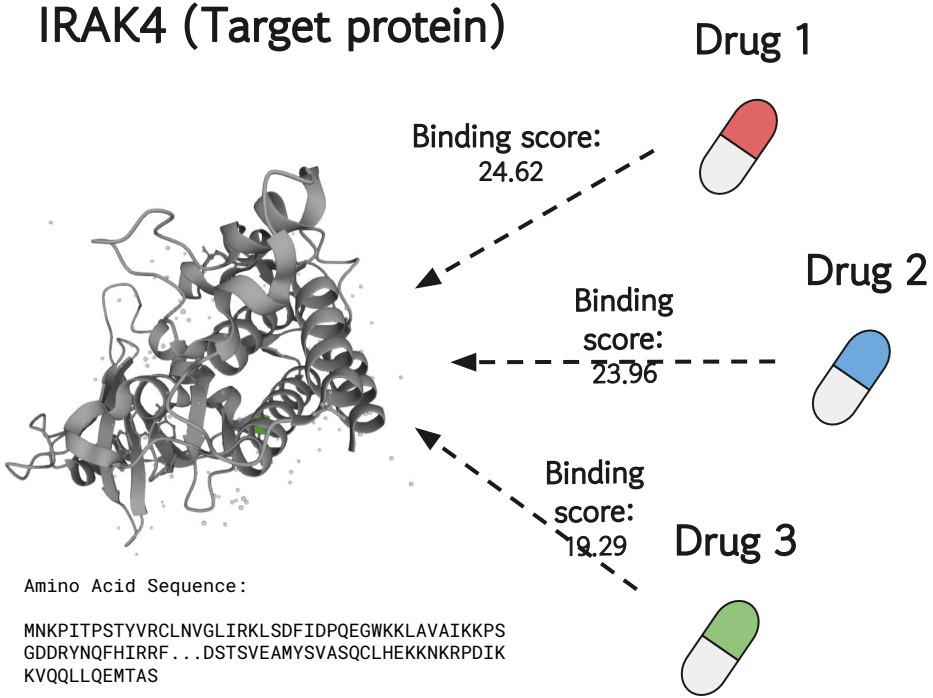
```
def mol2graph(smiles):
    graphs = []
    for smi in smiles:
        try:
            mol = Chem.MolFromSmiles(smi)
            if mol is None:
                continue
            g = mol_to_bigraph(mol, add_self_loop=True,
                               node_featurizer=PretrainAtomFeaturizer(),
                               edge_featurizer=PretrainBondFeaturizer(),
                               canonical_atom_order=True)
            graphs.append(g)

        except:
            continue
    return graphs
```

```
[Graph(num_nodes=72, num_edges=228,
       ndata_schemes={'atomic_number': Scheme(shape=(), dtype=torch.int64), 'chirality_type': Scheme(shape=(), dt
ype=torch.int64)}
       edata_schemes={'bond_type': Scheme(shape=(), dtype=torch.int64), 'bond_direction_type': Scheme(shape=(), d
type=torch.int64)}),
 Graph(num_nodes=39, num_edges=127,
       ndata_schemes={'atomic_number': Scheme(shape=(), dtype=torch.int64), 'chirality_type': Scheme(shape=(), dt
ype=torch.int64)}
       edata_schemes={'bond_type': Scheme(shape=(), dtype=torch.int64), 'bond_direction_type': Scheme(shape=(), d
type=torch.int64)})]
```

참고 코드: https://github.com/NetPharMedGroup/publication_fingerprint/tree/main

Feature 4. DeepPurpose



https://github.com/kexinhuang12345/DeepPurpose

DeepPurpose

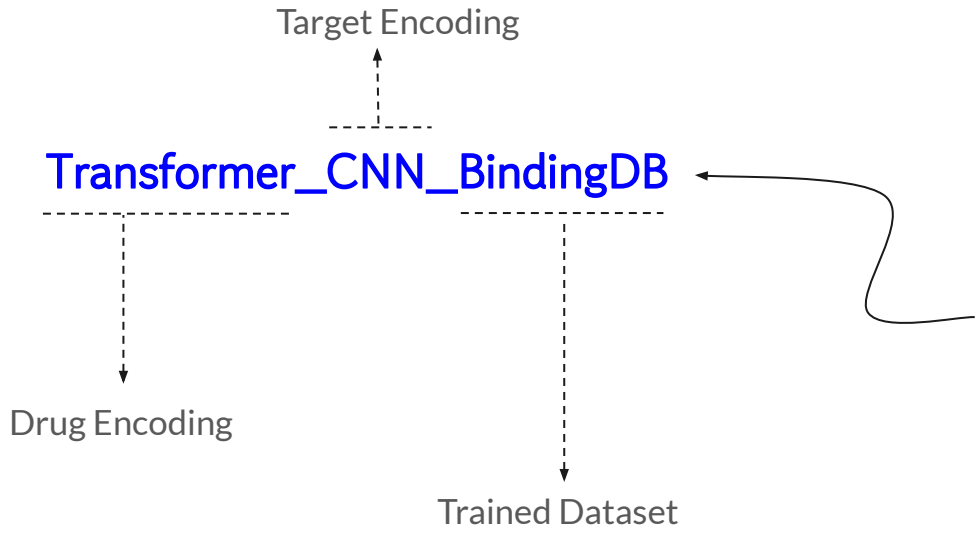
A Deep Learning Library for Compound and Protein Modeling
DTI, Drug Property, PPI, DDI, Protein Function Prediction

Applications in Drug Repurposing, Virtual Screening, QSAR, Side Effect Prediction and More

Virtual Screening Result

Rank	Drug Name	Target Name	Binding Score
1	CHEMBL4473472	IRAK4	24.62
2	CHEMBL4112564	IRAK4	23.96
3	CHEMBL5281809	IRAK4	23.83
4	CHEMBL4632511	IRAK4	23.75
5	CHEMBL3960537	IRAK4	23.20
6	CHEMBL3973967	IRAK4	23.09
7	CHEMBL4113046	IRAK4	21.79
8	CHEMBL4115314	IRAK4	20.93
9	CHEMBL4114127	IRAK4	20.17
10	CHEMBL4779496	IRAK4	19.29

Deep Purpose: Total 19 Pre-trained Models



1. CNN_CNN_BindingDB_IC50
2. Morgan_CNN_BindingDB_IC50
3. Morgan_AAC_BindingDB_IC50
4. MPNN_CNN_BindingDB_IC50
5. Daylight_AAC_BindingDB_IC50
6. CNN_CNN_DAVIS
7. CNN_CNN_BindingDB
8. Morgan_CNN_BindingDB
9. Morgan_CNN_DAVIS
10. MPNN_CNN_BindingDB
11. MPNN_CNN_KIBA
12. MPNN_CNN_DAVIS
13. **Transformer_CNN_BindingDB**
14. Daylight_AAC_DAVIS
15. Daylight_AAC_KIBA
16. Daylight_AAC_BindingDB
17. Morgan_AAC_BindingDB
18. Morgan_AAC_KIBA
19. Morgan_AAC_DAVIS

Feature 5. Language Models

- **ChemBERTa-77M-MTR:**
  - a pre-trained language model for molecules based on (Ro)BERT(a) trained on PubChem compounds.
  - 384 features

- **GPT2-Zinc480M-87M:**
  - a GPT2 style autoregressive language model trained on ~480m SMILES strings from the ZINC database available.
  - 768 features

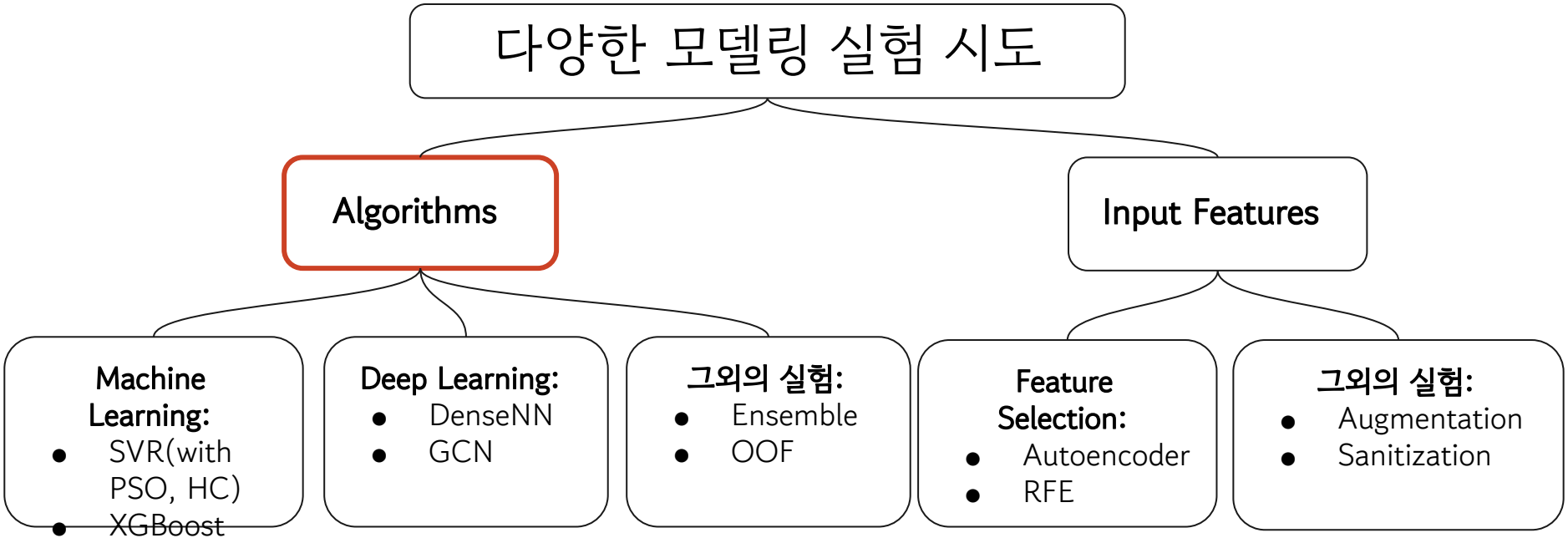
- **ChemGPT-19M:**
  - a transformers model for generative molecular modeling, which was pretrained on the PubChem10M dataset.
  - 256 features

3. Modeling

1. Algorithms
2. Input features

1. EDA
2. 데이터 전략
3. **> 모델링 전략**
4. 실험 관리
5. 범용성

모델링 전략


다양한 모델링 실험 시도
Algorithms
Input Features
Machine Learning:
SVR(with PSO, HC)
XGBoost
Deep Learning:
DenseNN
GCN
그외의 실험:
Ensemble
OOF
Feature Selection:
Autoencoder
RFE
그외의 실험:
Augmentation
Sanitization

PyCaret

- 짧은 시간 안에 여러가지 알고리즘을 한 번에 돌려보고 성능 좋은 몇 알고리즘만 선택
  - Catboost
  - Extra Trees
  - Gradient Boosting
  - Light GBM
  - Random Forest
  - XGBoost

	Model	MAE	MSE	RMSE	R2	RMSLE	MAPE	TT (Sec)
catboost	CatBoost Regressor	0.4399	0.3741	0.6111	0.6938	0.0763	0.0627	14.4760
et	Extra Trees Regressor	0.4382	0.3846	0.6196	0.6852	0.0771	0.0623	1.6200
br	Bayesian Ridge	0.4560	0.3840	0.6186	0.6845	0.0773	0.0647	0.1000
gbr	Gradient Boosting Regressor	0.4636	0.4029	0.6336	0.6718	0.0787	0.0656	2.5780
knn	K Neighbors Regressor	0.4599	0.4108	0.6402	0.6631	0.0795	0.0654	0.0500
lightgbm	Light Gradient Boosting Machine	0.4731	0.4283	0.6535	0.6504	0.0816	0.0674	0.6760
rf	Random Forest Regressor	0.4715	0.4339	0.6573	0.6463	0.0817	0.0671	4.0020
xgboost	Extreme Gradient Boosting	0.4779	0.4478	0.6688	0.6324	0.0832	0.0680	1.1800
par	Passive Aggressive Regressor	0.5033	0.4512	0.6700	0.6305	0.0832	0.0715	0.0660
omp	Orthogonal Matching Pursuit	0.5080	0.4636	0.6802	0.6199	0.0844	0.0715	0.0440
huber	Huber Regressor	0.4900	0.4621	0.6779	0.6183	0.0848	0.0698	0.2180
ridge	Ridge Regression	0.5036	0.4636	0.6797	0.6161	0.0848	0.0712	0.1680
ada	AdaBoost Regressor	0.5385	0.4989	0.7053	0.5930	0.0872	0.0757	0.7380
dt	Decision Tree Regressor	0.6834	0.9017	0.9489	0.2576	0.1182	0.0964	0.1120
lasso	Lasso Regression	0.9323	1.2361	1.1097	-0.0061	0.1372	0.1337	0.1780
en	Elastic Net	0.9323	1.2361	1.1097	-0.0061	0.1372	0.1337	0.0360
llar	Lasso Least Angle Regression	0.9323	1.2361	1.1097	-0.0061	0.1372	0.1337	0.0500
dummy	Dummy Regressor	0.9323	1.2361	1.1097	-0.0061	0.1372	0.1337	0.0500
lr	Linear Regression	2.9363	786.3027	14.0201	-534.9048	0.3279	0.3957	0.3460
lar	Least Angle Regression	23.1837	1644.5144	34.2167	-1299.8193	1.1512	3.2619	0.0720

```
CPU times: user 1min 44s, sys: 2.33 s, total: 1min 47s
Wall time: 2min 31s
```

Traditional ML with Feature Selection: Overall Workflow

1. Data

- Descriptor
- Descriptor + Morgan Fingerprint
- Descriptor + Morgan Fingerprint + infomax + binding affinity

2. Scaler

- Standard Scaler
- MinMax Scaler
- Robust Scaler
- No Scaling

3. Feature Selection

- PSO
- PSO + HC
- HC
- CFS
- Optuna

4. Regressor

- SVM
- Random Forest
- LightGBM
- ElasticNet

4-a. Hyperparameter Tuning

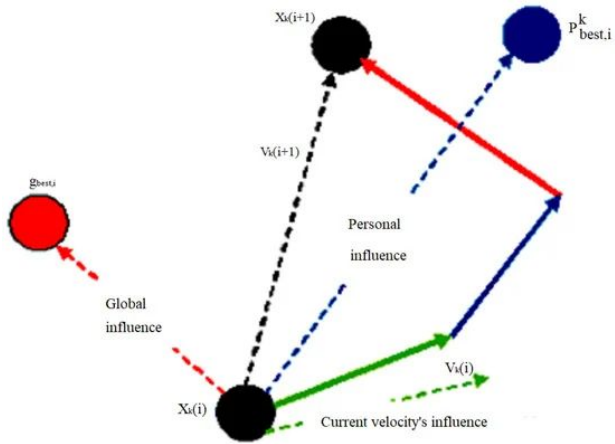
- Grid Search
- Optuna

4-b. Ensemble

- Bagging

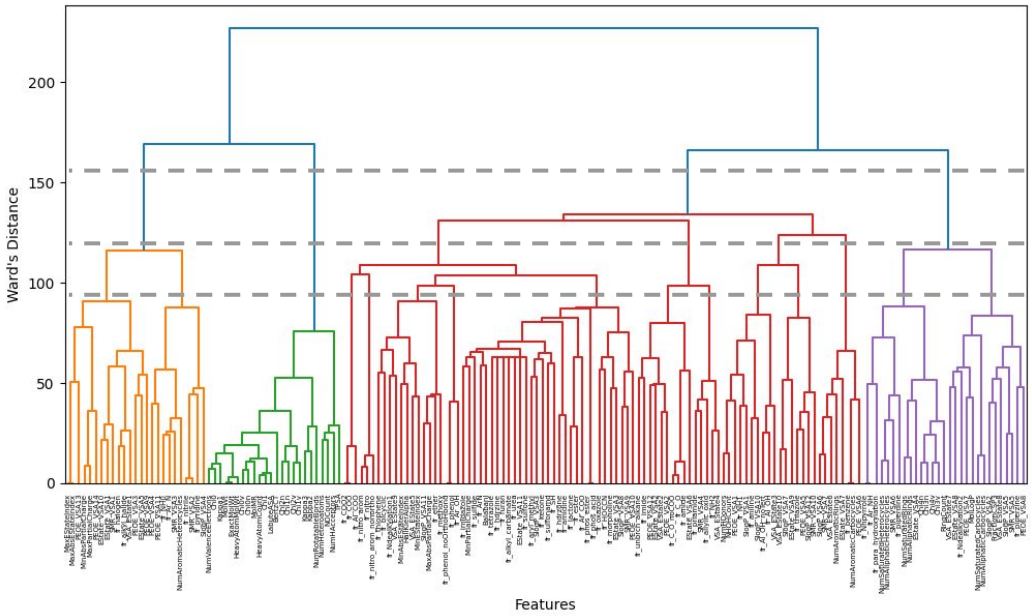
Feature Selection via PSO

- **PSO (Particle Swarm Optimization):**
  - 군집 지능 기반 최적화 알고리즘; 입자(particle)들이 최적해를 찾기 위해 군집 내 정보를 공유하며 이동
  - 이진 벡터를 사용해 feature을 선택, 입자가 최적의 feature 조합을 탐색
    - 입자가 0.5 이상이면 해당 특징 선택
    - 선택된 특징으로 머신러닝 모델 학습 후, RMSE로 성능 평가
  - 입자의 업데이트: 자신의 최적 위치와 군집 내 최적 위치를 기반으로 이동
  - 병렬 처리 가능



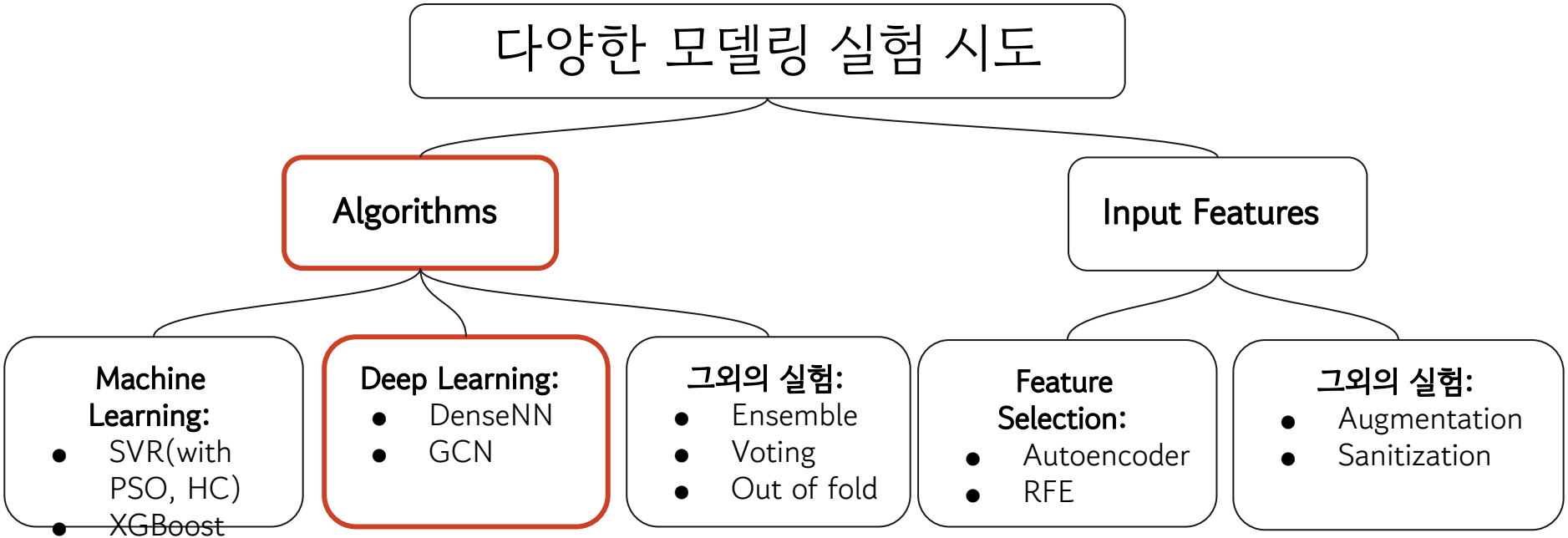
General scheme of the particle (Jain et al., 2022)

Feature Selection via Hierarchical Clustering



	feature	cluster	corr	abs_corr
0	FractionCSP3	1	-0.210268	0.210268
1	SlogP_VSA2	1	-0.193023	0.193023
2	SMR_VSA5	1	-0.184628	0.184628
3	PEOE_VSA8	1	-0.163672	0.163672
4	EState_VSA8	1	-0.154348	0.154348
...	...	...	...	...
165	SMR_VSA1	10	-0.036344	0.036344
166	MaxPartialCharge	10	-0.028379	0.028379
167	MinAbsPartialCharge	10	-0.024589	0.024589
168	PEOE_VSA14	10	-0.020717	0.020717
169	PEOE_VSA13	10	0.017000	0.017000

모델링 전략


다양한 모델링 실험 시도
Algorithms
Input Features
Machine Learning:
SVR(with PSO, HC)
XGBoost
Deep Learning:
DenseNN
GCN
그외의 실험:
Ensemble
Voting
Out of fold
Feature Selection:
Autoencoder
RFE
그외의 실험:
Augmentation
Sanitization

Multi-Layer Perceptron (Dense Neural Network)

Input features

- Morgan Fingerprint
- Infomax
- Descriptors
- Protein Binding Affinity
- Features from LLM

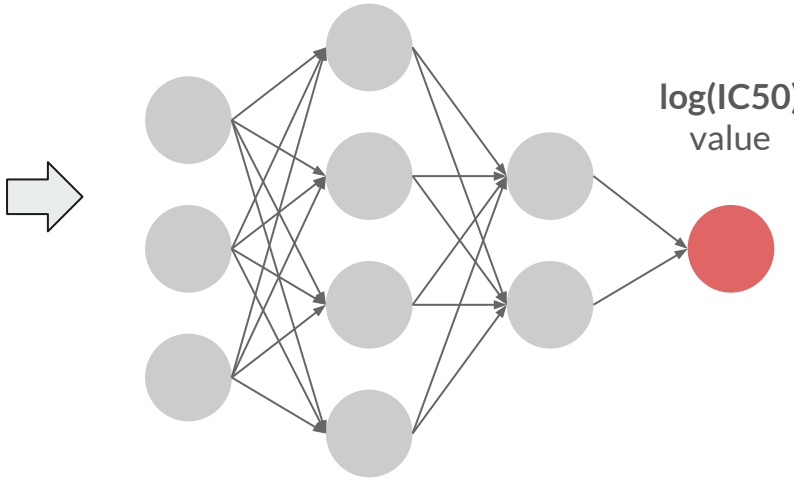
Trial 1

Trial 2

Trial 3

Trial 4

...

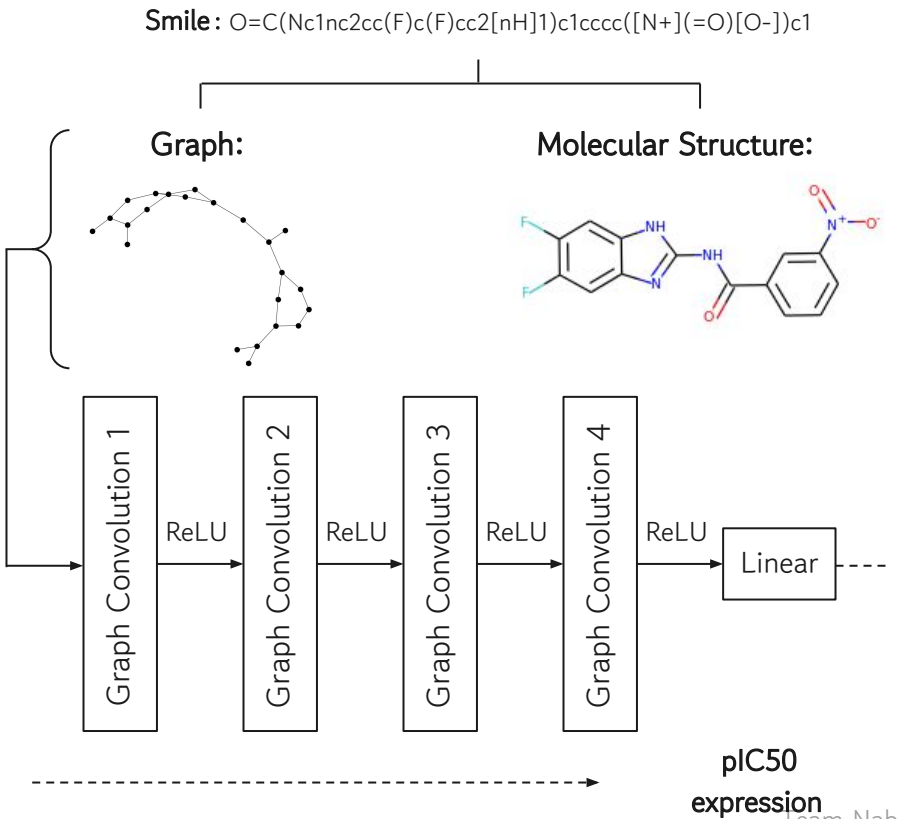


Hyperparameters

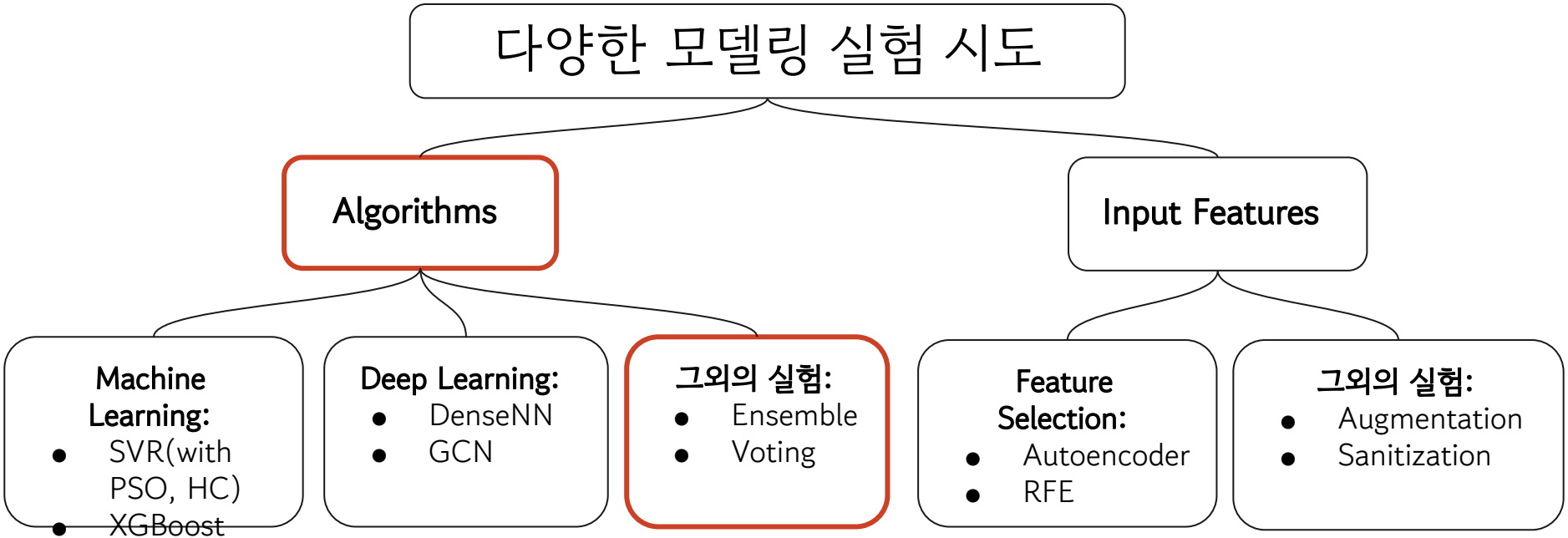
- Batch size
- Learning rate (+scheduler)
- Number of nodes/layers
- Activation functions (RELU, GELU, SELU, MISH)
- Epochs
- Dropout rate
- Optimizer
- ...

Graph Convolutional Network

- Smiles 데이터 graph 데이터로 변환하기
- Graph 데이터 convolutional layer들에 넣고 structural features 뽑아내기
- 마지막에 linear로 pIC50 예측
- 장점: Molecular structure를 고려하고 structural features를 사용하여 예측
- RMSE: 0.7768

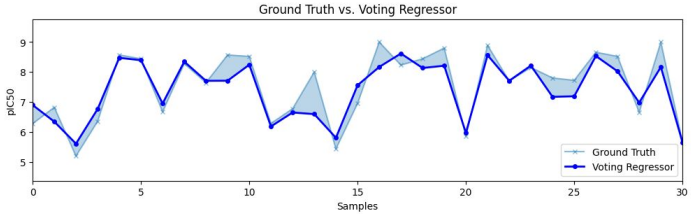


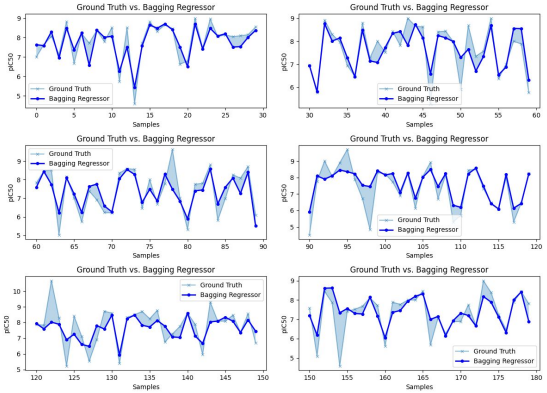
모델링 전략

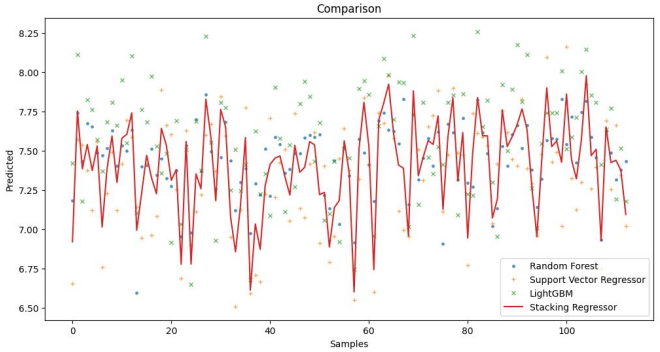

다양한 모델링 실험 시도
Algorithms
Input Features
Machine Learning:
SVR(with PSO, HC)
XGBoost
Deep Learning:
DenseNN
GCN
그외의 실험:
Ensemble
Voting
Feature Selection:
Autoencoder
RFE
그외의 실험:
Augmentation
Sanitization

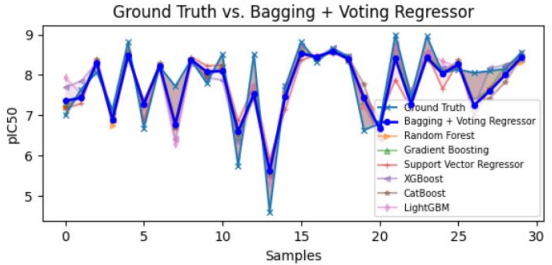
Ensemble : Bagging, Voting, Stacking, Bagging+Voting

- Random Forest
- Gradient Boosting
- Support Vector Regressor
- XGBoost
- CatBoost
- LightGBM

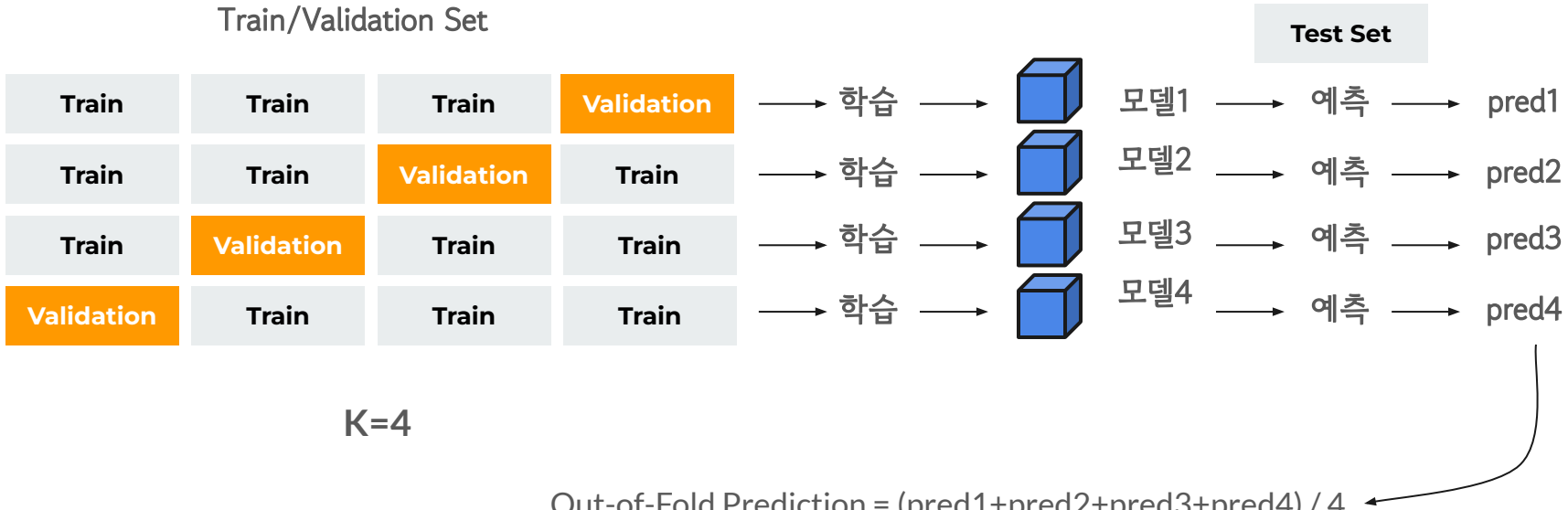




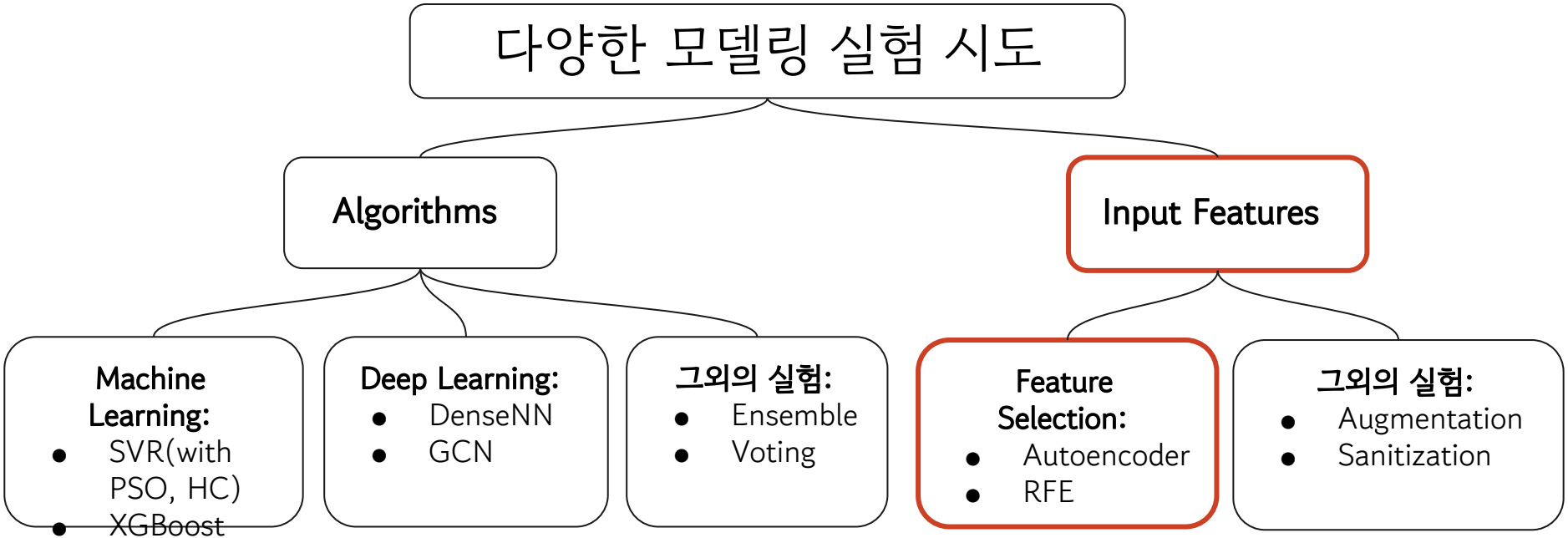




Out-of-Fold Prediction



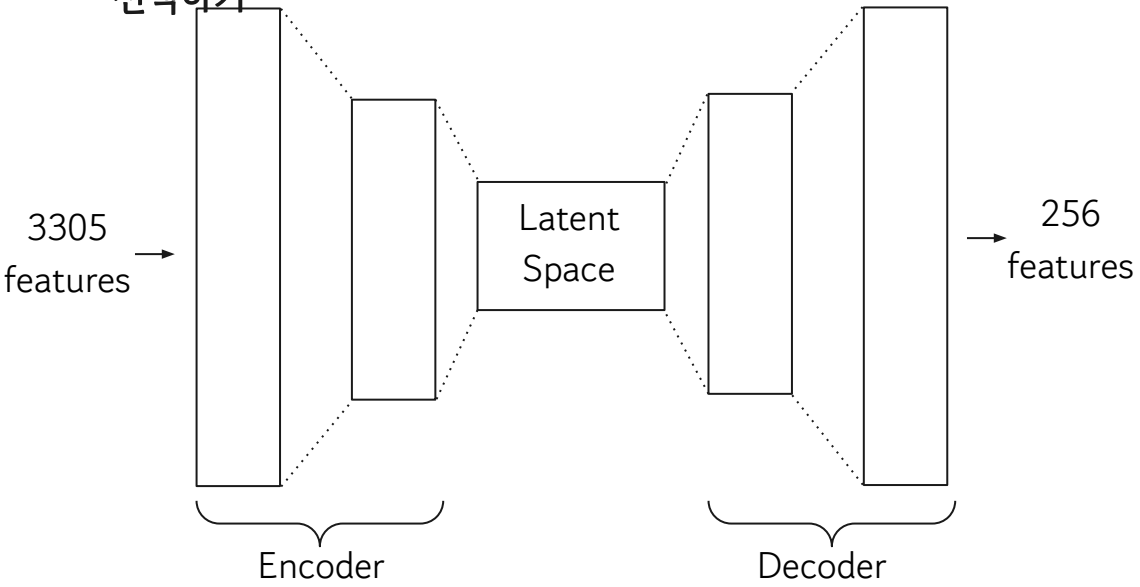
모델링 전략


다양한 모델링 실험 시도
Algorithms
Input Features
Machine Learning:
SVR(with PSO, HC)
XGBoost
Deep Learning:
DenseNN
GCN
그외의 실험:
Ensemble
Voting
Feature Selection:
Autoencoder
RFE
그외의 실험:
Augmentation
Sanitization

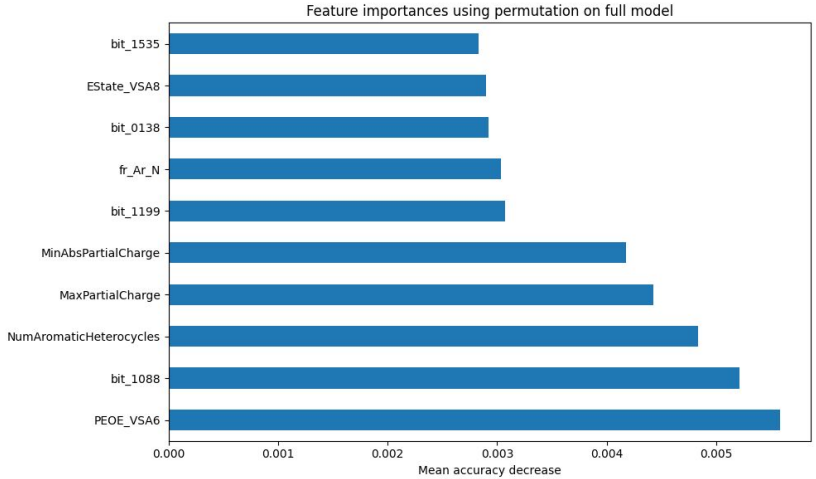
Feature Selection through Autoencoder

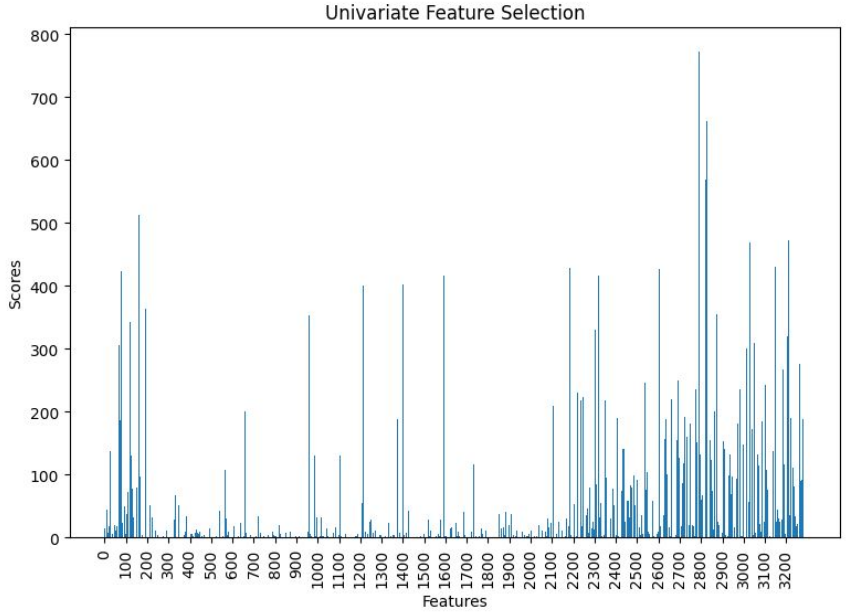
- 모든 features를 합쳤을 때, 3305 features가 나옴
- Autoencoder는 input data의 feature selection을 위해 사용되기도 함
- 일반화가 중요하기에 Autoencoder를 사용하여 3305 features 중 중요한 features들을 고르는 시도를 해봄

모든 features 중 Autoencoder로 중요한 features 선택하기



Other Feature Selection Related Work



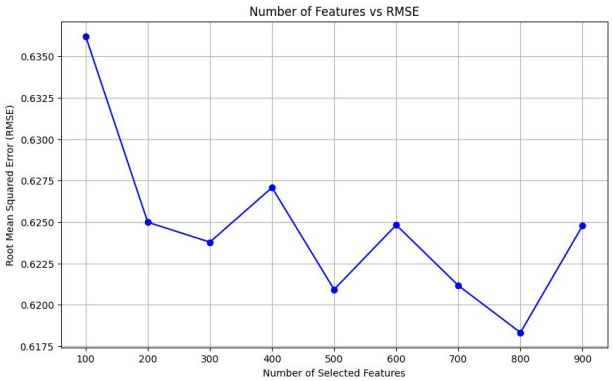


VIF to avoid multicollinearity

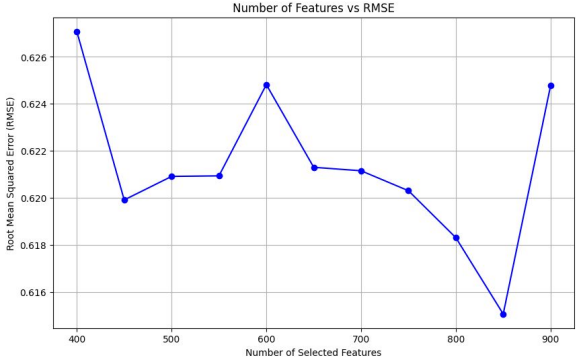
Other Feature Selection Related Work

- Feature selection via Recursive Feature Elimination (RFE)

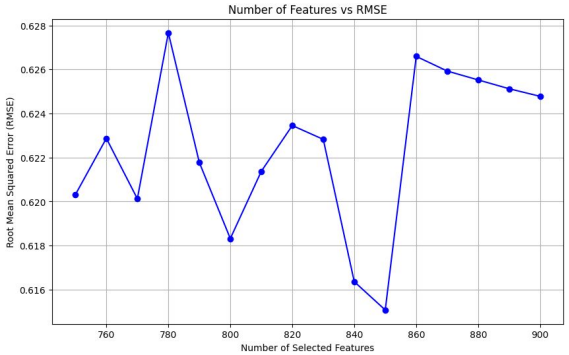
In range of 100 to 900, increment by 100



In range of 400 to 900, increment by 50



In range of 750 to 900, increment by 10



RFE 전 리더보드 점수

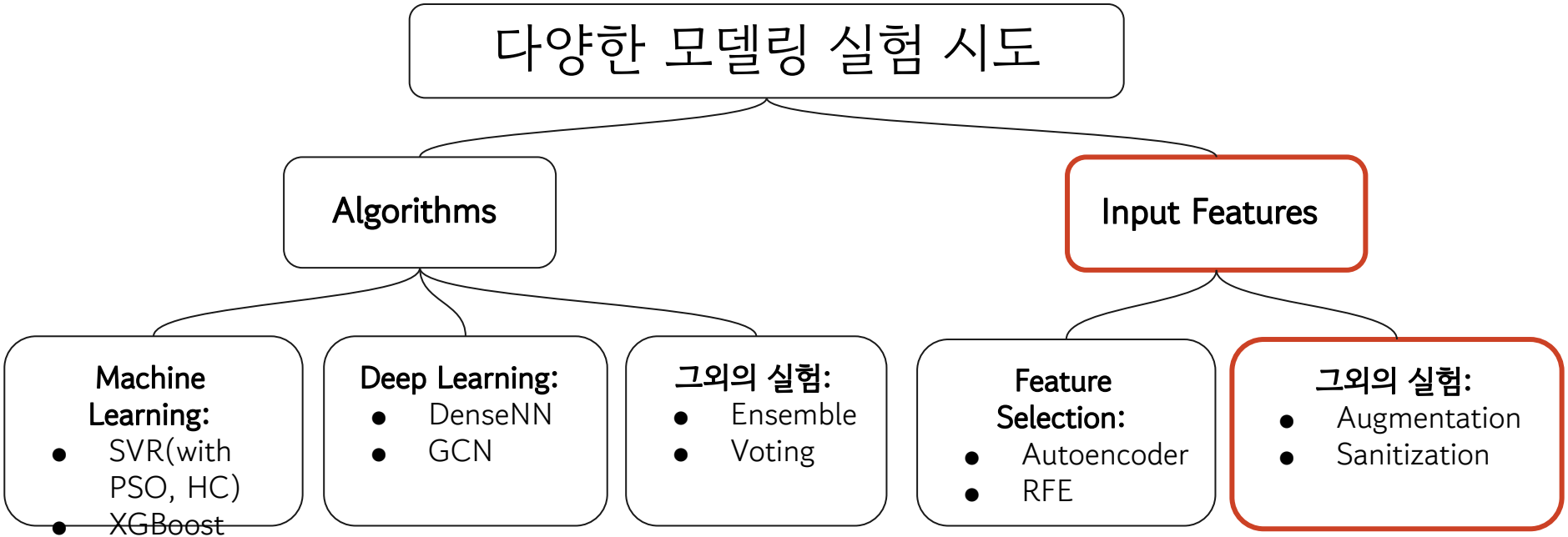
0.5452074148

RFE로 917 features 중 850개 골랐다.

RFE 후 리더보드 점수

0.5550250157

모델링 전략


다양한 모델링 실험 시도
Algorithms
Input Features
Machine Learning:
SVR(with PSO, HC)
XGBoost
Deep Learning:
DenseNN
GCN
그외의 실험:
Ensemble
Voting
Feature Selection:
Autoencoder
RFE
그외의 실험:
Augmentation
Sanitization

Data Augmentation

- 논문에서 제공하는 source 코드 사용해서 data augmentation 진행
- 총 데이터의 3배와 5배 augmentation 진행

<예시. 총 데이터 양 3배로 만들기>

	Molecule ChEMBL ID	Molecule_ID	Smiles	pIC50
0	CHEMBL4443947	0	CN[C@@H](C)C(=O)N[C@H](C(=O)N1C[C@@H](NC(=O)CC...	10.66
1	CHEMBL4443947	0	O=C(N[C@H](C(=O)N1[C@H](C(=O)N[C@H]2c3c(cccc3)...	10.66
2	CHEMBL4443947	0	C1CC[C@@H](NC(=O)[C@@H]2C[C@H](NC(=O)CCOCCOCCO...	10.66
3	CHEMBL4556091	1	CC(C)(O)[C@H](F)CN1Cc2cc(NC(=O)c3cnn4cccnc34)c...	10.59
4	CHEMBL4556091	1	N(C(=O)c1cnn2cccnc12)c1c(N2CCC(N3CCC3)CC2)cc2c...	10.59
...	...	...	...	...
5851	CHEMBL3403453	1950	c1(Nc2ncc(-c3nc4ccccc4s3)c(N[C@H]3[C@H](O)[C@H...	4.38
5852	CHEMBL3403453	1950	c1(Nc2ccccc2)nc(N[C@@H]2C[C@H](CO)[C@@H](O)[C@...	4.38
5853	CHEMBL4093989	1951	CC(C)Oc1ccccc1C(N)=O	4.26
5854	CHEMBL4093989	1951	NC(c1c(OC(C)C)cccc1)=O	4.26
5855	CHEMBL4093989	1951	NC(=O)c1ccccc1OC(C)C	4.26

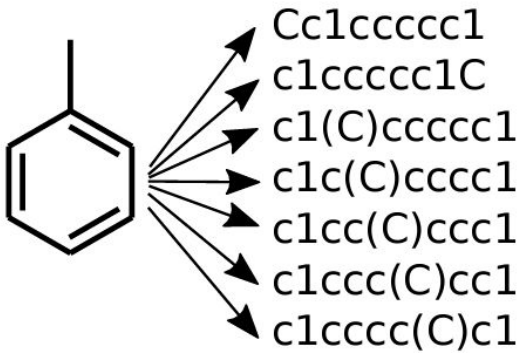
5856 rows × 4 columns

SMILES Enumeration as Data Augmentation for Neural Network Modeling of Molecules

Esben Jannik Bjerrum*

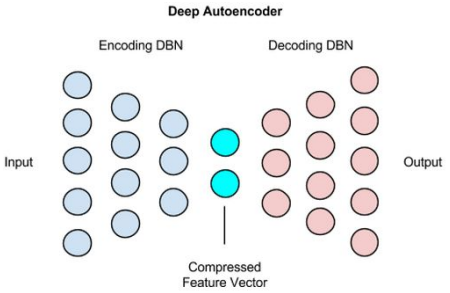
Wildcard Pharmaceutical Consulting, Frødings Allé 41, 2860 Søborg, Denmark
*) esben@wildcardconsulting.dk

예시. Smiles augmentation



(Bjerrum, E. J. (2017))

Deep Tabular Augmentation (DTA)



- pIC50 value값 중 양 극단에 있는 값에 대한 데이터가 부족
- Denoising autoencoder를 사용하여 원래 값에 약간의 noise를 추가하여 **가상의 데이터 생성**

<원본 데이터>

917	918	919	920	921
0.414871	0.723180	0.286540	0.260722	0.571070
0.479593	0.632385	0.455841	0.389242	0.230823
0.441555	0.648021	0.231791	0.432511	0.478430
0.487806	0.536121	0.452742	0.558532	0.225710
0.597747	0.461520	0.490433	0.468835	0.416154

+

<가상 데이터>

917	918	919	920	921
0.446992	0.520247	0.519448	0.475026	0.411388
0.443763	0.505544	0.521020	0.456339	0.421295
0.468355	0.543052	0.545958	0.458468	0.439213
0.479693	0.533197	0.534598	0.490743	0.429147
0.468620	0.555850	0.514132	0.482099	0.433300

Molecular Sanitization

the process of ensuring that the molecules in your dataset *are realistic*

	ID	Smiles	datamol_smiles
0	TEST_000	O=C(C1=CSC(C2=CC=CN=C2)=N1)NC3=CC(NC4CCN(C)CC4...	CN1CCC(Nc2ccc(OC3CCCC3)c(NC(=O)c3csc(-c4cccnc4...
1	TEST_001	N#CC1=CC(C=C2)=C(C=C1)N2C(N=C3)=NC(NC4CCCCC4)=...	N#Cc1ccc2c(ccn2-c2ncc(-c3cnn(C4CCNCC4)c3)c(NC3...
2	TEST_002	N#CC(C=C1)=C(N[C@@H]2CCNC2)C=C1NC(N=C3)=NC=C3C...	N#Cc1ccc(Nc2ncc(-c3cnn([C@H]4CCNC4)c3)cn2)cc1N...
3	TEST_003	N#CC(C=C1)=CC=C1NC(N=C2)=NC(NC3CC(NC(C=C)=O)CC...	C=CC(=O)NC1CCC(Nc2nc(Nc3ccc(C#N)cc3)ncc2-c2cnn...
4	TEST_004	N#CC(C=C1)=CC=C1NC(N=C2)=NC(NC3CC(N)CC3)=C2C(C...	N#Cc1ccc(Nc2ncc(-c3cnn(C4CCOCC4)c3)c(NC3CCC(N)...
...	...	...	...
108	TEST_108	N#CC1=CC(C=C2)=C(C=C1)N2C(N=C3)=NC(N4CCOCC4)=C...	N#Cc1ccc2c(ccn2-c2ncc(-c3cnn(C4CCNCC4)c3)c(N3C...
109	TEST_109	O=C(C1=CSC(C2=CC=NC=C2)=N1)NC3=CC(NC4CCN(C(C)C...	CC(=O)NC1CCN(c2ccc(NC3CCN(C(C)C)CC3)cc2NC(=O)c...
110	TEST_110	N#Cc1ccc(Nc2ncc(cn2)c3cnn(c3)C4CCNCC4)cc1N[C@@...	N#Cc1ccc(Nc2ncc(-c3cnn(C4CCNCC4)c3)cn2)cc1N[C@...
111	TEST_111	O=C(C)N(CC1)CCC1N2N=CC(C3=CN=C(N4C(C=CC(C#N)=C...	CC(=O)N1CCC(n2cc(-c3cnc(-n4ccc5cc(C#N)ccc54)nc...
112	TEST_112	N#CC(C=C1)=CC=C1NC(N=C2)=NC(NC3CNCCC3)=C2C(C=N...	N#Cc1ccc(Nc2ncc(-c3cnn(C4CCOCC4)c3)c(NC3CCCNC3...

113 rows × 3 columns

대회에서 제공한 SMILES

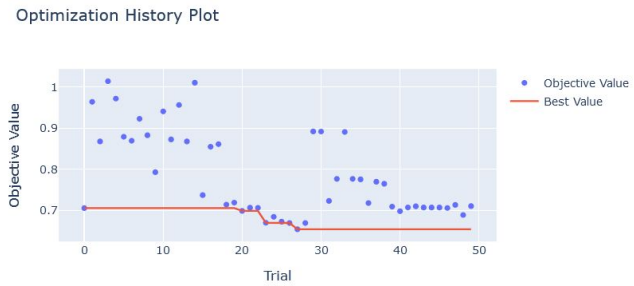
Sanitized SMILES

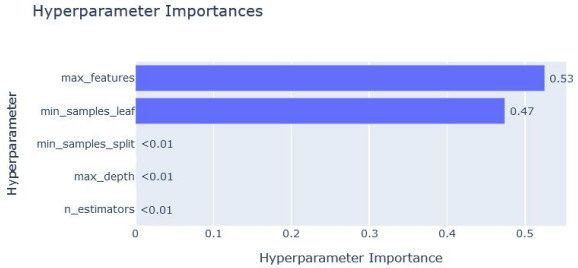
4. 실험관리

1. Optuna
2. MLflow
3. Score metrics
4. 최종 제출 파일 선택 기준

1. EDA
2. 데이터 전략
3. 모델링 전략
4. **> 실험 관리**
5. 범용성

Optuna





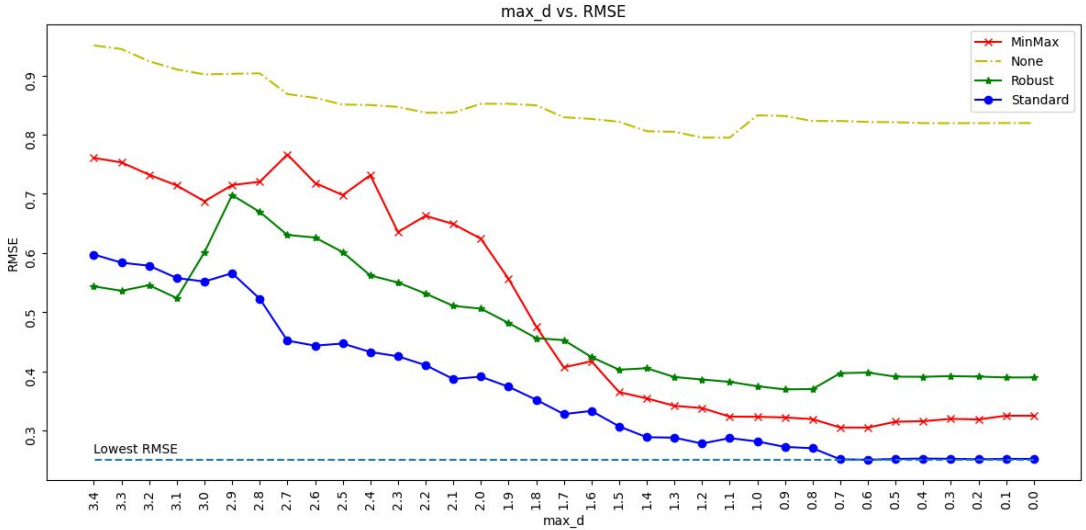
OPTUNA **Hyperparameter Optimization**

- Input Features
- Number of Hidden Layers
- Hidden Layer Dimension
- Learning Rate
- Batch Size
- Dropout Rate

- Optuna를 실행해서 optimal한 hyperparameter들을 찾아낸 후, 이 중에서 중요하다고 판단된 hyperparameter에 집중해서 다시 최적화 수행

MLflow API & 시각화

MLOps and Experiment Management



실험을 하면서 MLflow 저장소에 저장한 정보들을 기반으로, MLflow API를 사용해서 시각화한 모습.

Run Name	Created	avg_correct_ratio	avg_r2_ic50
trial_17	10 days ago	0.6731706997180...	0.095577919...
trial_10	10 days ago	0.6711246639123...	0.084929585...
trial_3	10 days ago	0.6644540625614...	0.081830012...
trial_16	10 days ago	0.6572771985048...	0.166957366...
trial_14	10 days ago	0.6562607384090...	0.193886148...

Score Metrics

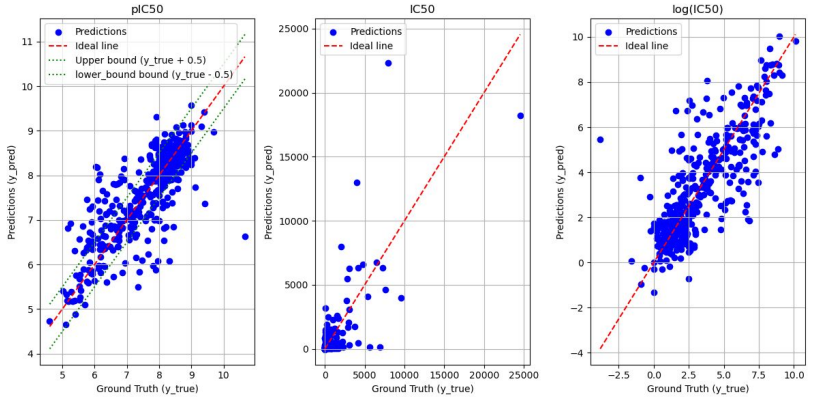
주어진 Score Metrics:

$$\text{Normalized RMSE} = \frac{\text{RMSE}}{\max(\text{IC50}_{\text{true}}) - \min(\text{IC50}_{\text{true}})}$$

$$\text{Correct Ratio} = \frac{1}{N}\sum_{i=1}^{N} I(\text{Absolute Error}_{\text{pIC50},i} \leq 0.5)$$

$$\text{Score} = 0.5 \times (1 - \min(\text{Normalized RMSE}, 1)) + 0.5 \times \text{Correct Ratio}$$

평가를 위한 시각화



우리가 평가를 위해 사용한 Score Metrics

- RMSE of pIC50
- RMSE of IC50
- MAPE of pIC50
- MAPE of IC50
- nRMSE of IC50
- R-squared of pIC50
- R-squared of IC50
- Correct ratio
- Score

결과 예시

fold	rmse_pic50	mape_pic50	rmse_ic50	nrmse_ic50	mape_ic50	correct_ratio	score
0	0.712715	7.06998	784.805	0.160239	953.547	0.641295	0.740528
1	0.560221	5.65182	697.222	0.123989	209.519	0.709536	0.792774
2	0.596321	5.98207	738.825	0.137576	172.59	0.701662	0.782043
3	0.639124	6.32008	955.93	0.162341	557.898	0.657895	0.747777
4	0.555675	5.76084	970.804	0.164869	201.866	0.682456	0.758794

모델링 전략 및 선택 기준

- **모델 선택:** SVR 모델과 MLP 모델을 활용하여 제출
- **성능 평가:** Public Leaderboard (LB) 점수와 내부 교차 검증 점수를 종합적으로 고려하여 모델 성능을 평가
- **일반화 성능 최적화:** 모델이 훈련 데이터에 과적합되지 않도록 주의하면서, 일반화 성능이 뛰어난 모델을 선별
- **Leaderboard Shake-up 대응:** Public LB에서의 점수가 다소 낮게 나왔지만, 최종 성적에서의 Shake-up 가능성을 염두에 두고 모델 선택.

모델링 전략 및 선택 기준

Descriptors

최종선택

1. SVR

- Traditional ML
- Descriptor | Morgan FP
- PSO
- Hierarchical clustering | Depth tuning with Optuna
- SVR bagging
- K-fold cross validation
- Target(y): pIC50

2. MLP

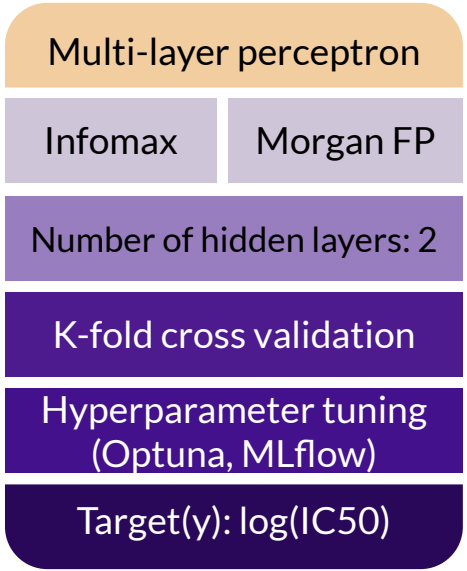
- Deep Learning
- Infomax | Morgan FP
- Dense network with Optuna
- K-fold cross validation
- Target(y): log(IC50)

- Algorithm
- Input features
- Feature selection
- Model architecture
- Validation
- Label (y) value

5. 범용성

1. MLP 모델의 장점, 응용분야
2. Smiles 관련 문제 해결 방법

MLP 모델 장점



1. **Morgan fingerprint**
   - 분자의 구조적 특징을 고차원 벡터로 변환
   - 다양한 화합물에 적용가능
2. **MLP**
   - MLP의 비선형 학습 능력은 일반화 성능이 우수
   - Hyperparameter tuning을 통한 최적화

→ 간단한 구조로 빠른 훈련 및 예측, 대규모 데이터에 적용 가능

→ 약물 설계, 재료 과학 등 다양한 화학 도메인에 적용 가능

Morgan Fingerprint 기반 MLP 모델의 응용 분야

1. **ADMET 특성 예측**
   a. 흡수, 분포, 대사, 배설, 독성 예측
   b. 약물의 체내 동태 분석에 활용

2. **재창출 약물 검색**
   a. 기존 약물의 새로운 질병 치료 가능성 탐색
   b. 예: 항바이러스제, 항암제의 새로운 응용 분야 발굴

3. **환경 독성 및 화학 물질 평가**
   a. 산업 화학 물질의 환경 독성, 생분해성 예측
   b. 환경 영향 평가 도구로 활용

4. **신소재 개발**
   a. 신소재의 특성 및 최적의 소재 조합 예측
   b. 전자 재료, 배터리 촉매 분야 응용 가능

범용성: Workflow / Smiles 관련 문제 해결 방법

1. 탐색적 데이터 분석 (EDA)

- 데이터 분포, 결측치, 이상치 파악
- 변수 간 상관관계 분석
- 시각화를 통한 패턴 및 트렌드 식별
- 도메인 지식을 활용한 데이터 해석

2. 특성 및 모델 탐색

- 다양한 특성 엔지니어링 기법 적용
- 베이스라인 모델 구축 및 성능 평가
- 여러 알고리즘 실험 (예: 랜덤 포레스트, XGBoost, 신경망)
- 앙상블 기법 고려 (예: 스태킹, 보팅)

3. 평가 지표 관리 및 교차 검증

- 문제에 적합한 평가 지표 선정 (예: RMSE,nRMSE, R2, mape, score...)
- K-fold 교차 검증
- 과적합 모니터링 및 방지 전략 수립
- MLflow를 통해 실험 관리

4. 특성 선택 및 데이터 증강

- 특성 중요도 분석
- 차원 축소 기법 적용
- 도메인 특화 데이터 증강 기법 개발
- 불균형 데이터 처리

5. 하이퍼파라미터 최적화

- 하이퍼파라미터 튜닝 (Optuna, GridSearch, Random Search)

범용성: Morgan FP + MLP + Optuna 조합

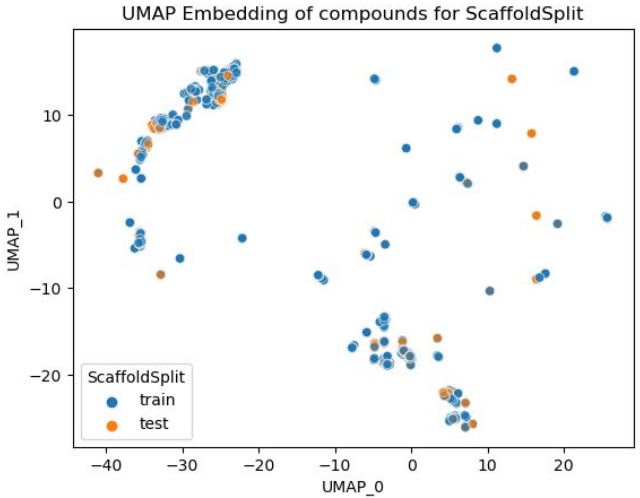
1. Morgan fingerprint
    - 분자의 구조적 특징을 고차원 벡터로 변환
    - 다양한 화합물에 적용가능
2. MLP
    - MLP의 비선형 학습 능력은 일반화 성능이 우수
    - Hyperparameter tuning을 통한 최적화

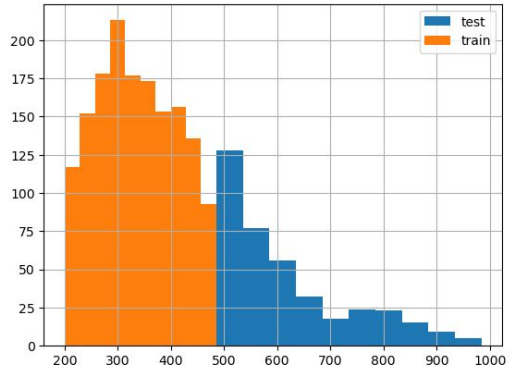
→ 간단한 구조로 빠른 훈련 및 예측, 대규모 데이터에 적용 가능

→ 약물 설계, 재료 과학 등 다양한 화학 도메인에 적용 가능

Future Work

- splito 라이브러리를 사용해서 train/validation 세트를 나눌 때, 화학 구조나, 분자들간의 거리, 무게 등을 기준으로 나누는 방법 시도해보기
- 경제학의 gini score를 성능평가 metrics로 사용해보기
- t-SMILES, mol2vec 같은 다른 molecular representation 사용해보기
- RDKit 라이브러리를 사용해서 Descriptor를 만들었는데, datamol이나 ChemoPy 등의 다른 라이브러리를 사용해서 만들어보고 비교해보기
- MLP와 SVR을 앙상블 시켜보기





Reference

- Bajusz, D., Rácz, A., & Héberger, K. (2017). Chemical Data Formats, Fingerprints, and Other Molecular Descriptions for Database Analysis and Searching, Compr. Med. Chem. III, 3, 8.
- Bjerrum, E. J. (2017). SMILES enumeration as data augmentation for neural network modeling of molecules. arXiv preprint arXiv:1703.07076.
- Jain, M., Saihjpal, V., Singh, N., & Singh, S. B. (2022). An Overview of Variants and Advancements of PSO Algorithm. Applied Sciences, 12(17), 8392. https://doi.org/10.3390/app12178392
- Zagidullin, B., Wang, Z., Guan, Y., Pitkänen, E., & Tang, J. (2021). Comparative analysis of molecular fingerprints in prediction of drug combination effects. Briefings in bioinformatics, 22(6), bbab291.

Thanks for Your Attention