Advancing Tabular Data Analysis

2024.12.6

2024020757 조용수

jys1537@korea.ac.kr

발표자 소개

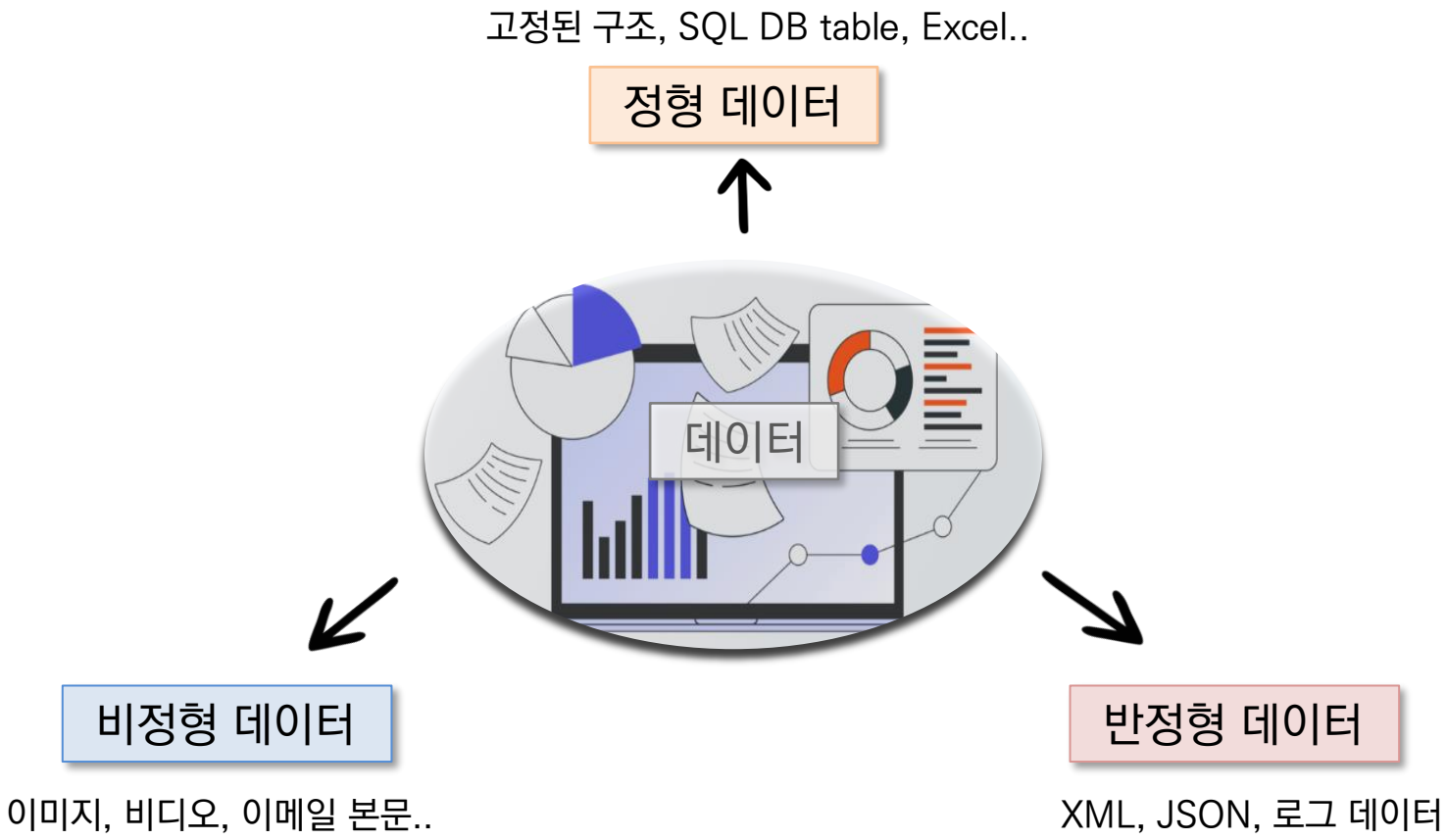
❖ **조용수 (Yongsu Jo)**
- 고려대학교 산업경영공학과 석사과정(2024.03 ~)
- Data Mining & Quality Analytics Labs. (김성범 교수님)

❖ **관심 연구 분야**
- Supervised Learning
- Tabular Data

❖ **E-Mail**
- jys1537@korea.ac.kr

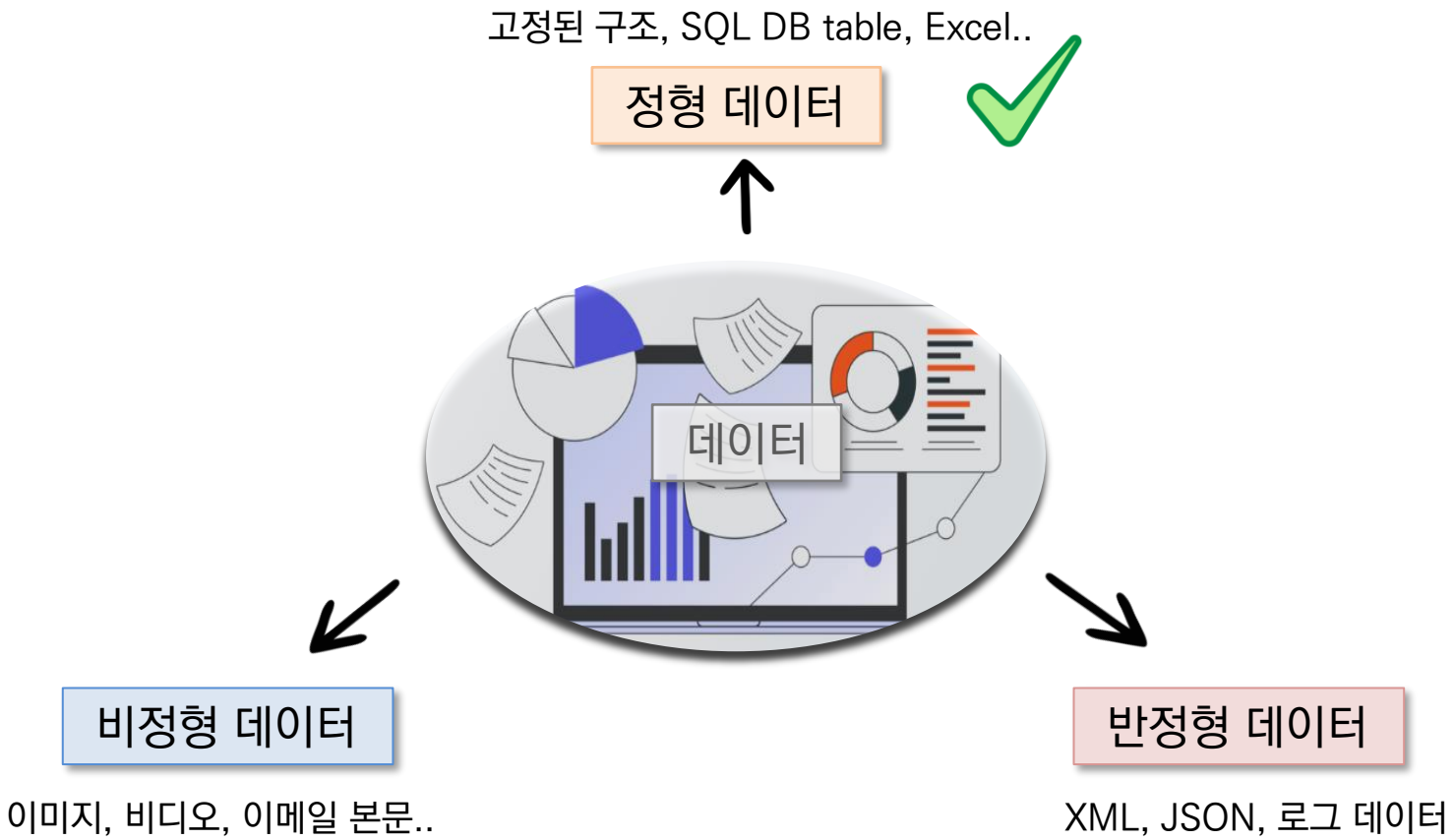
데이터

❖ 데이터
사실(fact)이나 관찰된 정보의 집합, 분석, 처리를 통해 유용한 정보로 변환될 수 있는 원재료



데이터

❖ 데이터
사실(fact)이나 관찰된 정보의 집합, 분석, 처리를 통해 유용한 정보로 변환될 수 있는 원재료



Tabular Data

❖ Tabular Data
– 행 (Row, Instance)과 열(Column, Feature)로 표현되는 데이터

학생 이름	과목	점수	마법 지팡이 재료	최애 주문
해리 포터	어둠의 방어술	95	호랑가시나무	익스펙토 페트로눔
헤르미온느 그레인저	변신술	100	포도나무	레비오사
론 위즐리	마법약	72	회양목	레둑토
말포이	마법 역사	50	호손	인센디오
네빌 롱바텀	식물학	85	체리나무	알로호모라

Tabular Data

❖ Tabular Data
– 행 (Row, Instance)과 열(Column, Feature)로 표현되는 데이터

특징 각 관측치는 어떤 특징을 가지고 있는가?

학생 이름	과목	점수	마법 지팡이 재료	최애 주문
해리 포터	어둠의 방어술	95	호랑가시나무	익스펙토 페트로눔
헤르미온느 그레인저	변신술	100	포도나무	레비오사
론 위즐리	마법약	72	회양목	레둑토
말포이	마법 역사	50	호손	인센디오
네빌 롱바텀	식물학	85	체리나무	알로호모라

Tabular Data

❖ Tabular Data
– 행 (Row, Instance)과 열(Column, Feature)로 표현되는 데이터

특징

관측치

어떤 관측치가 있는가?

학생 이름	과목	점수	마법 지팡이 재료	최애 주문
해리 포터	어둠의 방어술	95	호랑가시나무	익스펙토 페트로눔
헤르미온느 그레인저	변신술	100	포도나무	레비오사
론 위즐리	마법약	72	회양목	레둑토
말포이	마법 역사	50	호손	인센디오
네빌 롱바텀	식물학	85	체리나무	알로호모라

Tabular Data

❖ Tabular Data
– 행 (Row, Instance)과 열(Column, Feature)로 표현되는 데이터

특징

관측치

학생 이름	과목	점수	마법 지팡이 재료	최애 주문
해리 포터	어둠의 방어술	95	호랑가시나무	익스펙토 페트로눔
헤르미온느 그레인저	변신술	100	포도나무	레비오사
론 위즐리	마법약	72	회양목	레둑토
말포이	마법 역사	50	호손	인센디오
네빌 롱바텀	식물학	85	체리나무	알로호모라

관계 관측치에서 특징간의 관계
ex) ‘말포이’의 ‘마법 지팡이 재료’는? 호손! ➔ Information

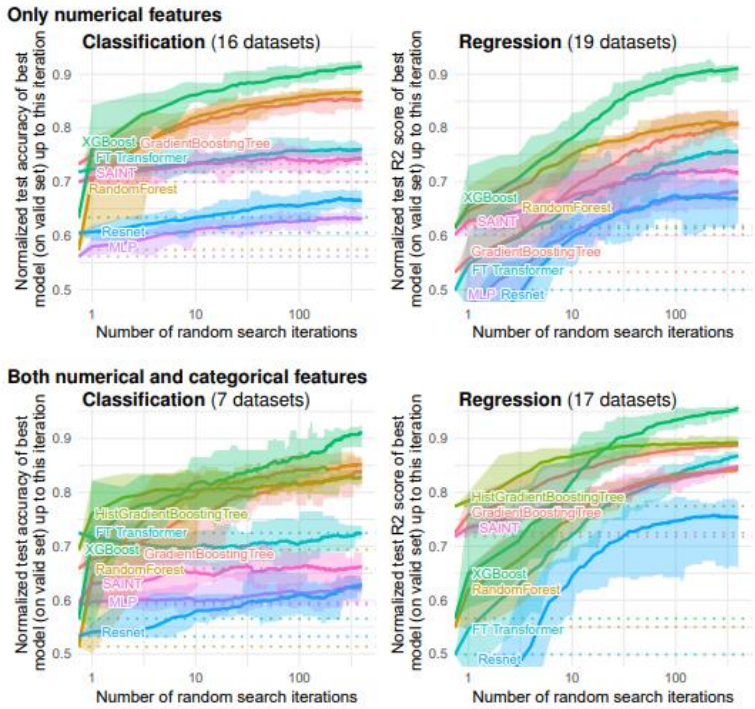
Why is deep learning not powerful for tabular data?

Why do tree-based models still outperform deep learning on typical tabular data?

Léo Grinsztajn
Soda, Inria Saclay
leo.grinsztajn@inria.fr

Edouard Oyallon
MLIA, Sorbonne University

Gaël Varoquaux
Soda, Inria Saclay

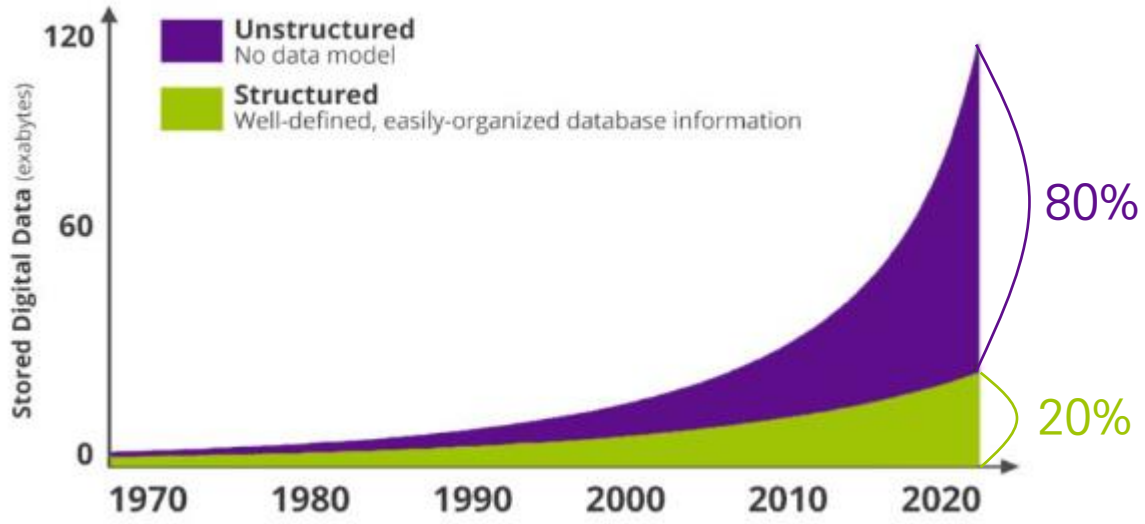


Why?
Heterogeneous , Small Size, Irregularity

Grinsztajn, L., Oyallon, E., & Varoquaux, G. (2022). Why do tree-based models still outperform deep learning on typical tabular data?. *Advances in neural information processing systems*, *35*, 507-520.

Why Tabular Data is Important?

❖ Tabular Data를 활용하기 위한 딥러닝 아키텍처가 지속적으로 개발되고 있음



Tabular data를 위한 딥러닝 모델

VIME: Extending the Success of Self- and Semi-supervised Learning to Tabular Domain

SAINT: Improved Neural Networks for Tabular Data via Row Attention and Contrastive Pre-Training

TabNet: Attentive Interpretable Tabular Learning

TABPFN: A TRANSFORMER THAT SOLVES SMALL TABULAR CLASSIFICATION PROBLEMS IN A SECOND

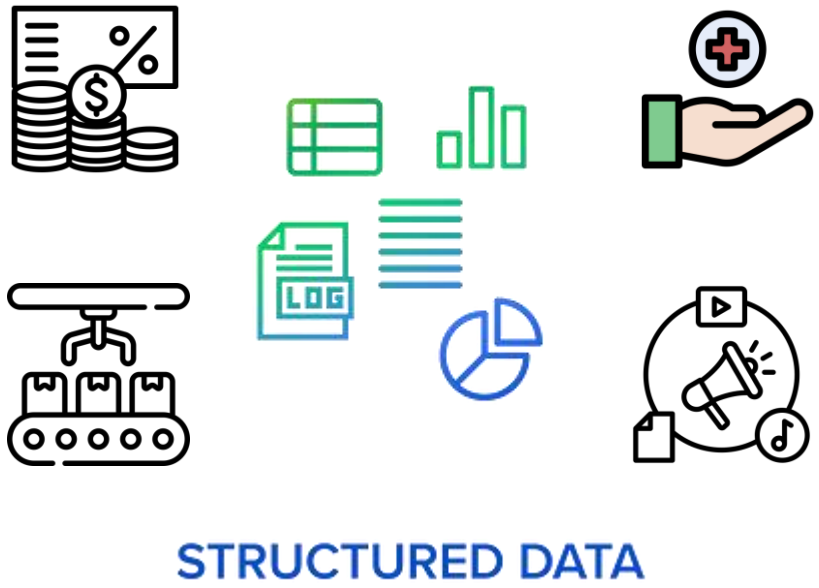
Noah Hollmann*,1,2 Samuel Müller*,1 Katharina Eggensperger[1] Frank Hutter[1,3]
[1] University of Freiburg, [2] Charité University Medicine Berlin
[3] Bosch Center for Artificial Intelligence * Equal contribution.
Correspondence to noah.hollmann@charite.de & muellesa@cs.uni-freiburg.de

앞으로 비정형 데이터 분석이 비중이 더 커질텐데 왜 Tabular Data를?

https://www.pecan.ai/blog/what-is-structured-data/
https://www.komprise.com/glossary_terms/unstructured-data/

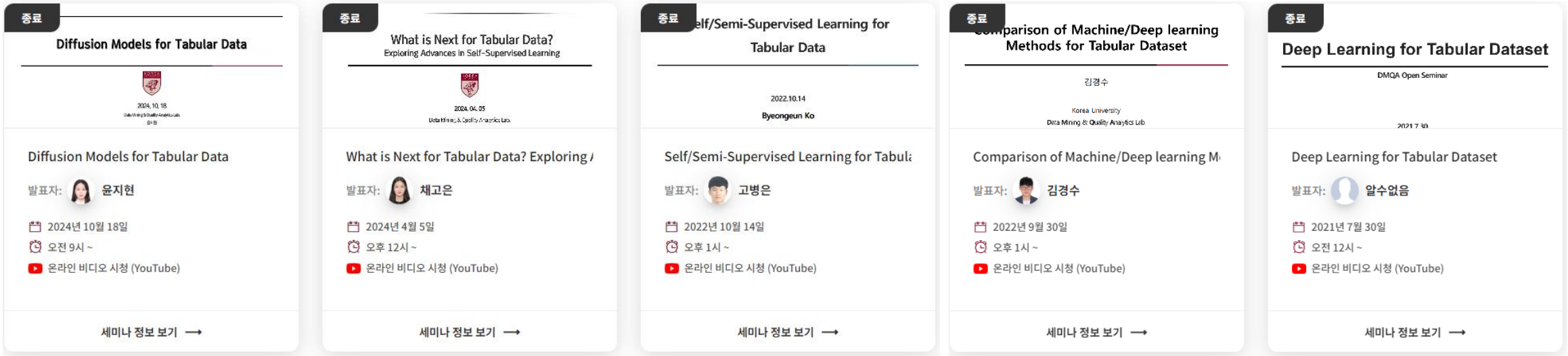
Why Tabular Data is Important?

❖ Tabular Data가 중요한 이유



Tabular Data는 여전히 전통적인 산업에서 매우 중요함

Tabular Data Seminar @ DMQA



Diffusion Models for Tabular Data
1) Multinomial Diffusion 2) TabDDPM 3) Tab-CSDI

What is Next for Tabular Data? Exploring Advances in Self-Supervised Learning
1) VIME 2) STUNT 3) SCARF 4) TransTab 5) MambaTab

Self/Semi-Supervised Learning for Tabular Data
1) VIME 2) SubTab 3) SCARF 4) Contrastive Mixup

Comparison of Machine/Deep learning Methods for Tabular Dataset
1) Deep Neural Networks and Tabular Data: A Survey
2) Tabular Data: Deep Learning is Not All You Need
3) Why do tree-based models still outperform deep learning on tabular data?

Deep Learning for Tabular Dataset
1) TabNet, 2) Tabular Data: Deep Learning is Not All You Need

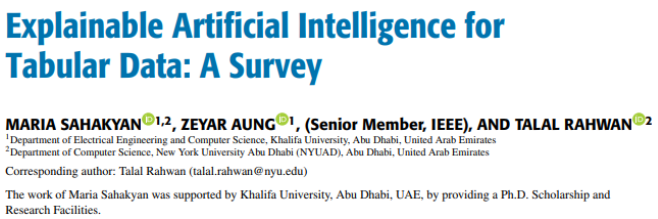
When Do Neural Nets Outperform Boosted Trees on Tabular Data?

Duncan McElfresh[*1,2], **Sujay Khandagale**[3], **Jonathan Valverde**[4], **Vishak Prasad C**[5], **Ganesh Ramakrishnan**[5], **Micah Goldblum**[6], **Colin White**[1,7]

[1] Abacus.AI, [2] Stanford, [3] Pinterest, [4] University of Maryland,
[5] IIT Bombay, [6] New York University, [7] Caltech

언제 GBDT 계열보다 딥러닝 계열이 성능이 우수한가?
(NeurIPS 2020, 212회 인용)

Introduction



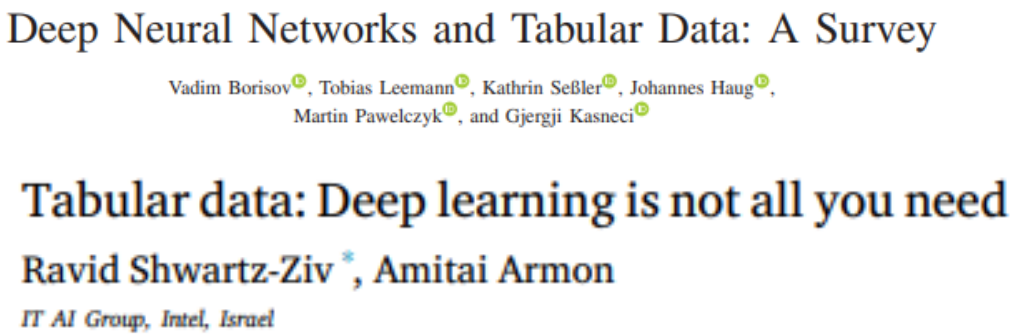
Explainable Artificial Intelligence for Tabular Data: A Survey

MARIA SAHAKYAN[1,2], ZEYAR AUNG[1], (Senior Member, IEEE), AND TALAL RAHWAN[2]

[1]Department of Electrical Engineering and Computer Science, Khalifa University, Abu Dhabi, United Arab Emirates
[2]Department of Computer Science, New York University Abu Dhabi (NYUAD), Abu Dhabi, United Arab Emirates

Corresponding author: Talal Rahwan (talal.rahwan@nyu.edu)

The work of Maria Sahakyan was supported by Khalifa University, Abu Dhabi, UAE, by providing a Ph.D. Scholarship and Research Facilities.



Deep Neural Networks and Tabular Data: A Survey

Vadim Borisov, Tobias Leemann, Kathrin Seßler, Johannes Haug, Martin Pawelczyk, and Gjergji Kasneci

Tabular data: Deep learning is not all you need

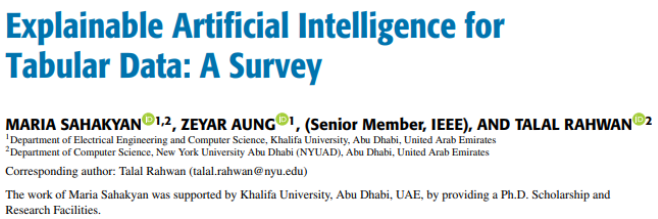
Ravid Shwartz-Ziv *, Amitai Armon

IT AI Group, Intel, Israel

Tabular Data에서 딥러닝은 정답이 아니다?

대부분 연구에서 50개 미만 데이터에서 혹은 충분한 Tunning 없이 성능의 **평균 순위 비교**에만 집중

Introduction



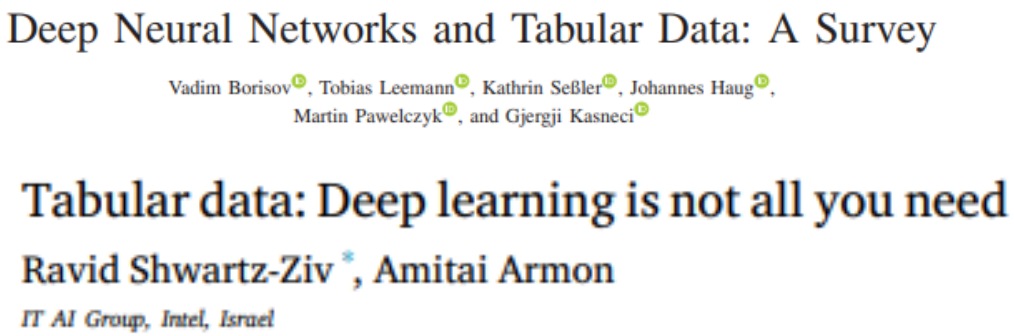
Explainable Artificial Intelligence for Tabular Data: A Survey

MARIA SAHAKYAN[1,2], ZEYAR AUNG[1], (Senior Member, IEEE), AND TALAL RAHWAN[2]

[1]Department of Electrical Engineering and Computer Science, Khalifa University, Abu Dhabi, United Arab Emirates
[2]Department of Computer Science, New York University Abu Dhabi (NYUAD), Abu Dhabi, United Arab Emirates

Corresponding author: Talal Rahwan (talal.rahwan@nyu.edu)

The work of Maria Sahakyan was supported by Khalifa University, Abu Dhabi, UAE, by providing a Ph.D. Scholarship and Research Facilities.



Deep Neural Networks and Tabular Data: A Survey

Vadim Borisov, Tobias Leemann, Kathrin Seßler, Johannes Haug, Martin Pawelczyk, and Gjergji Kasneci

Tabular data: Deep learning is not all you need

Ravid Shwartz-Ziv *, Amitai Armon

IT AI Group, Intel, Israel

Tabular Data에서 딥러닝은 정답이 아니다?

대부분 연구에서 50개 미만 데이터에서 혹은 충분한 Tunning 없이 성능의 **평균 순위 비교**에만 집중

- 여러 Dataset에서 다양한 알고리즘의 Performance 비교 분석
  19 알고리즘, 176 Datasets, Hyper Para. Opt (〈 30) ← 최대 규모
- 알고리즘 선택 / Hyper Para. Opt. 중요도 분석
- 메타특성 분석을 통한 알고리즘 선택 가이드라인

No individual algorithm dominates

절대적인 알고리즘은 없다

Algorithm	Rank min	Rank max	Rank mean	Rank med.	Mean Acc. mean	Mean Acc. med.	Std. Acc. mean	Std. Acc. med.	Time /1000 inst. mean	Time /1000 inst. med.
TabPFN	1	18	4.88	3	0.84	0.93	0.35	0.26	0.00	0.00
CatBoost	1	18	5.37	4	0.85	0.91	0.39	0.30	26.22	2.75
ResNet	1	19	6.75	6	0.77	0.79	0.42	0.30	23.67	13.87
RandomForest	1	18	7.65	7	0.76	0.82	0.40	0.29	0.47	0.32
SAINT	1	19	7.67	6	0.74	0.87	0.42	0.31	197.41	181.62
FTTransformer	1	18	7.93	7	0.75	0.78	0.42	0.32	32.93	26.39
XGBoost	1	17	8.30	8	0.74	0.80	0.42	0.30	0.95	0.61
NODE	1	19	8.35	8	0.73	0.75	0.36	0.28	173.55	144.45
SVM	1	18	9.54	11	0.68	0.72	0.35	0.28	23.90	0.42
MLP-rtdl	1	19	9.77	10	0.64	0.69	0.39	0.31	21.48	12.21
LightGBM	1	19	10.00	10	0.68	0.71	0.45	0.38	0.64	0.23
LinearModel	1	19	10.21	11	0.61	0.71	0.38	0.29	0.06	0.05
DANet	1	18	10.74	10	0.68	0.69	0.41	0.34	83.57	71.19
DecisionTree	1	19	11.44	13	0.60	0.67	0.45	0.32	0.02	0.01
MLP	1	19	11.49	13	0.57	0.54	0.38	0.30	27.88	16.81
STG	1	19	11.49	12	0.57	0.64	0.40	0.34	21.22	18.24
KNN	1	19	13.12	15	0.46	0.51	0.38	0.32	0.00	0.00
TabNet	3	19	14.54	16	0.42	0.40	0.52	0.49	41.83	34.35
VIME	2	19	14.88	17	0.33	0.27	0.36	0.29	18.95	16.43

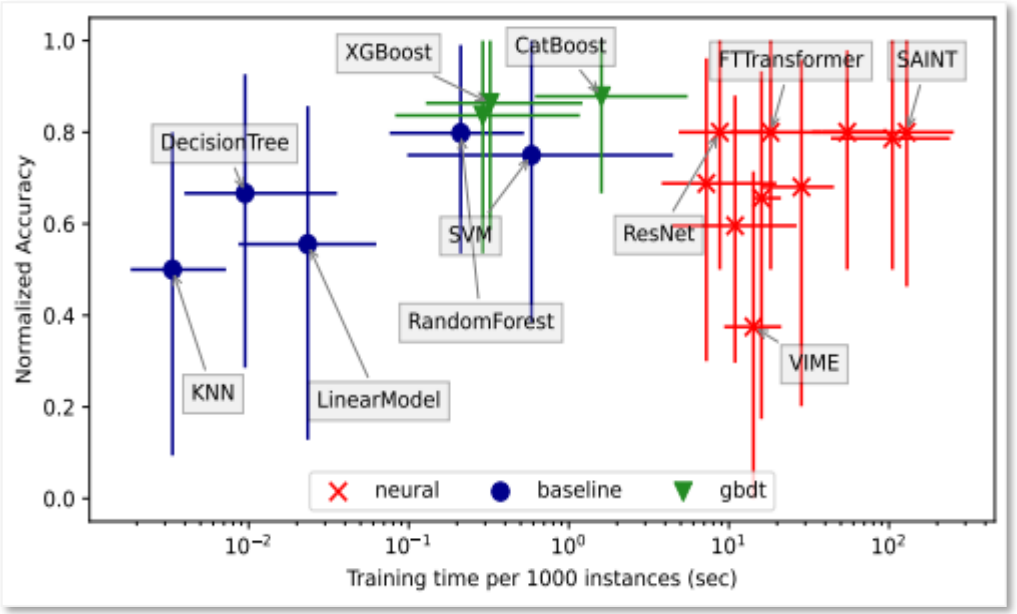
대부분 알고리즘이 적어도 1 개의 Dataset 이상에서 1위와 최하위를 기록

➔ 모든 Tabular 데이터에서 성능이 평균 이상을 하는 알고리즘은 없다.

Performance

Accuracy / Loss

- Accuracy

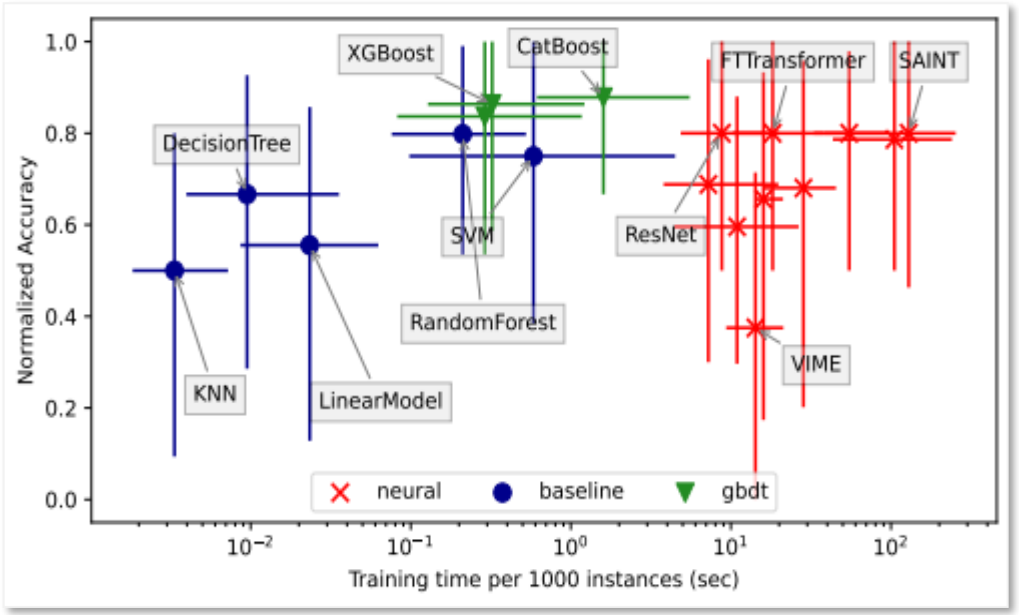


- Accuracy : 일반적으로는 GBDT 계열이 NN 계열보다 성능 및 학습 시간이 양호함

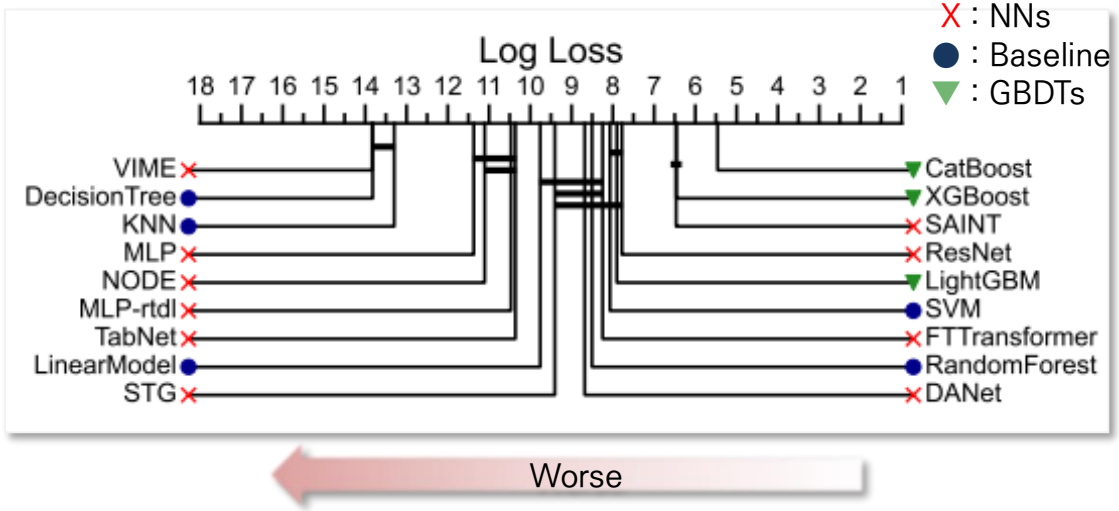
Performance

Accuracy / Loss

- Accuracy



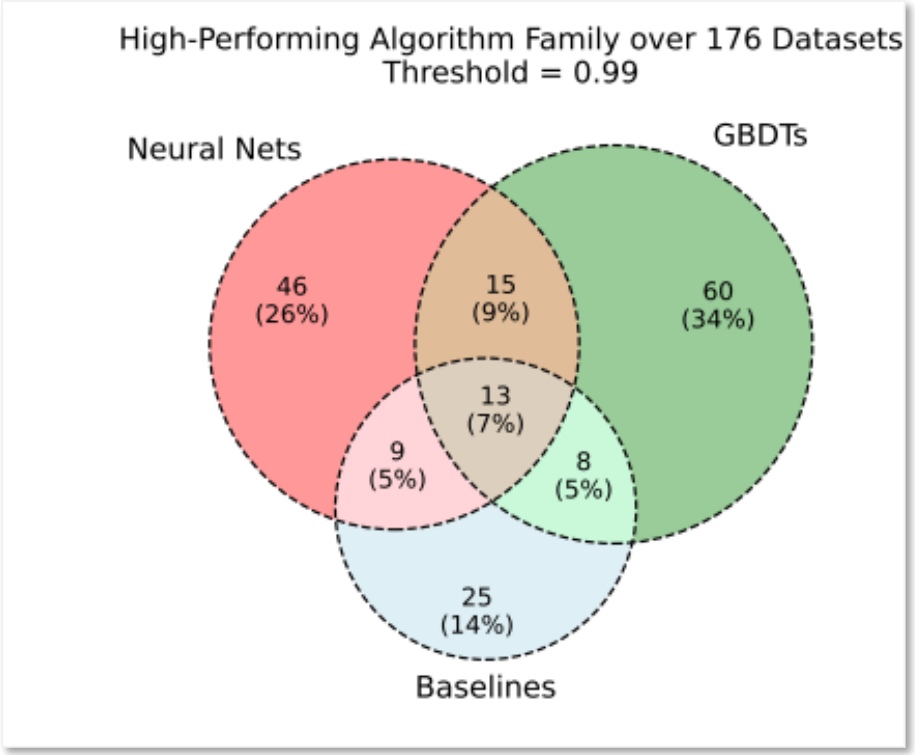
- Log Loss



- Accuracy : 일반적으로는 GBDT 계열이 NN 계열보다 성능 및 학습 시간이 양호함
- Log loss : Catboost가 평균적으로 가장 양호, 그 외 일부 알고리즘은 GBDT 계열과 유사 수준

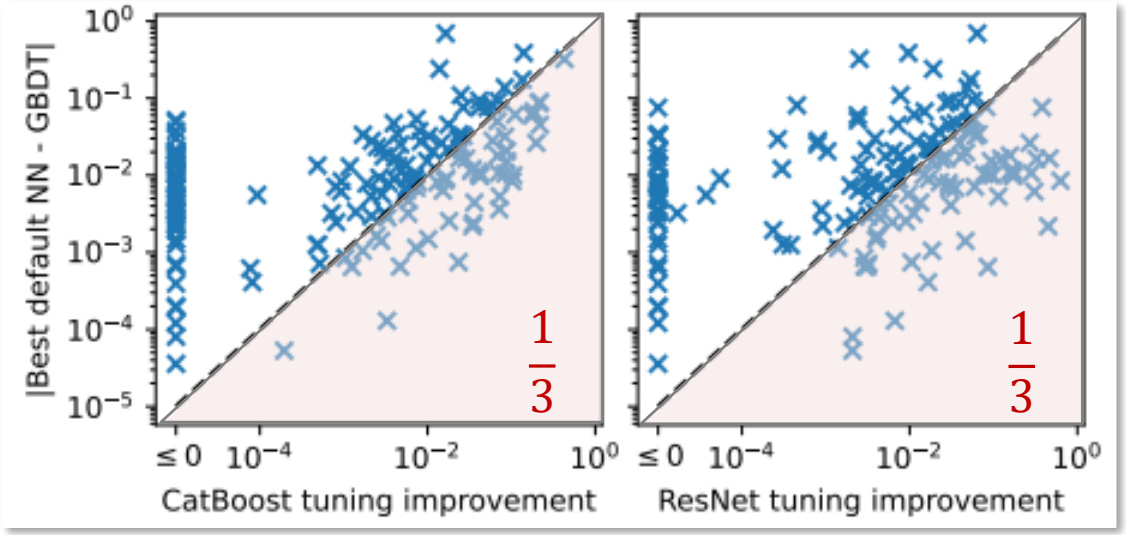
Performance

High Perf. Datasets by Algorithms



- Algorithms 종류별 High Performance Dataset
  1. 데이터셋에 따라 고성능 알고리즘 종류가 다름
  2. NNs, GBDTs, Baseline 중 '하나만' High Performance 보이는 Dataset이 다수 (74%)

Importance of Hyper Parameter Optimization



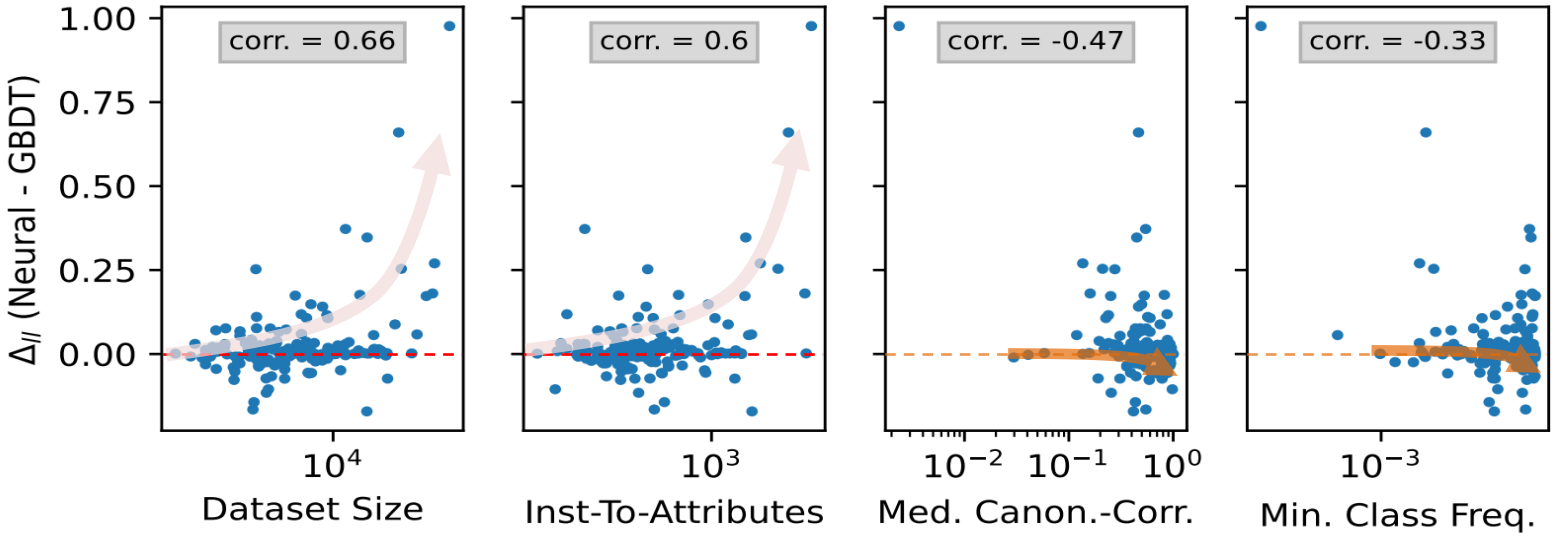
X axis – Hyper Parameter Tunning 의 개선 성능
Y axis – CatBoost와 ResNet의 기본 모델의 성능 차이

- 1/3 Dataset에서 모델간 차이보다 Hyper Parameter Tunning의 성능 개선이 더 크다.
  ➔ 많은 경우에 Algorithms의 차이보다 충분한 Hyper Paramter Tunning 이 더 효과적임

Metafeature Analysis

데이터의 '어떤 특성'이 '특정 알고리즘'을 성능이 좋도록 만들까?

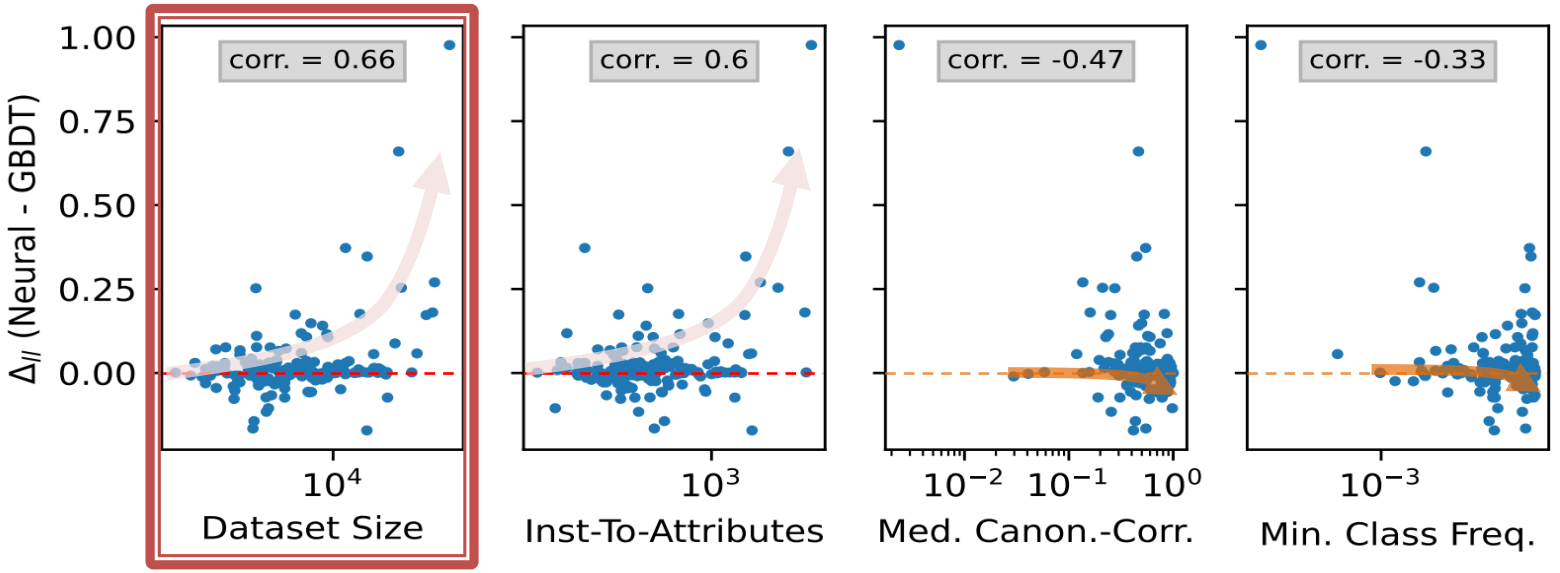
❖ Best GBDT계열 (XGBoost, Light GBM) / NN계열 (SAINT, ResNet) 간 Loss 차이와 Metafeature 분석



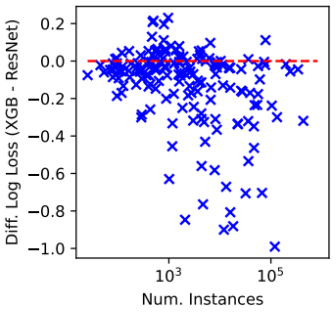
Metafeature Analysis

데이터의 '어떤 특성'이 '특정 알고리즘'을 성능이 좋도록 만들까?

❖ Best GBDT계열 (XGBoost, Light GBM) / NN계열 (SAINT, ResNet) 간 Loss 차이와 Metafeature 분석



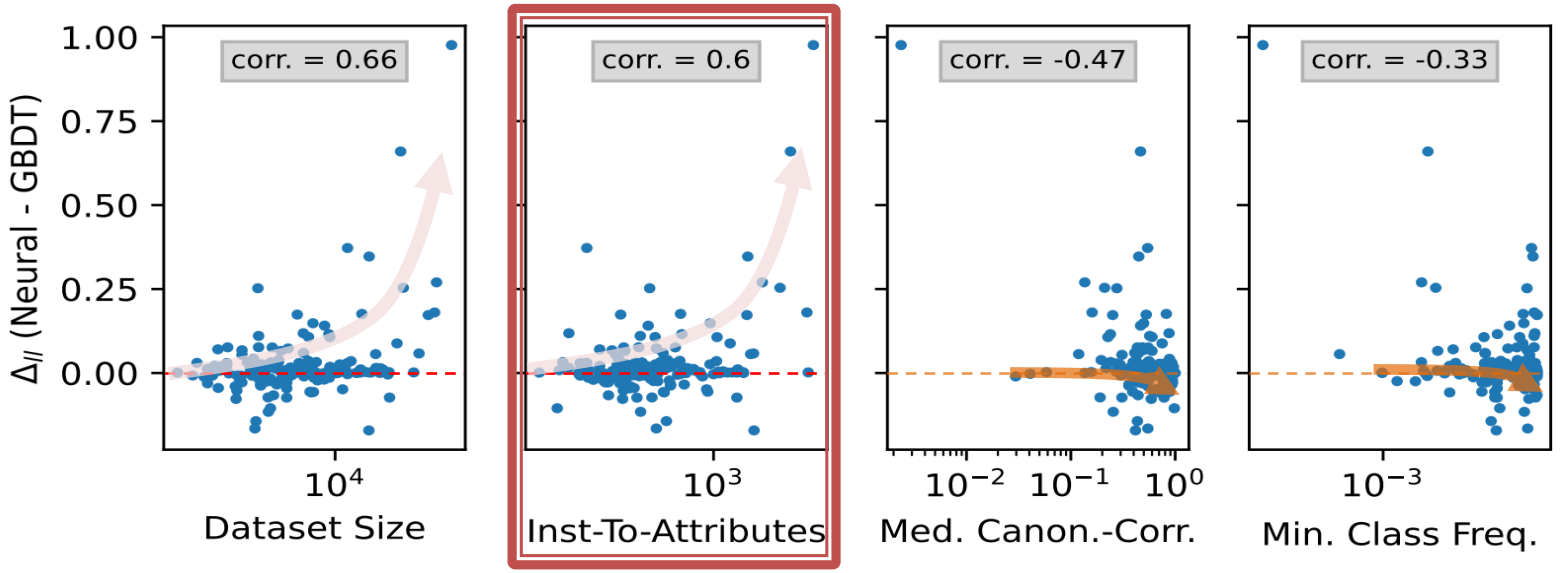
1. Dataset이 클수록 GBDT 계열이 상대적으로 성능이 좋다.



Metafeature Analysis

데이터의 '어떤 특성'이 '특정 알고리즘'을 성능이 좋도록 만들까?

❖ Best GBDT계열 (XGBoost, Light GBM) / NN계열 (SAINT, ResNet) 간 Loss 차이와 Metafeature 분석

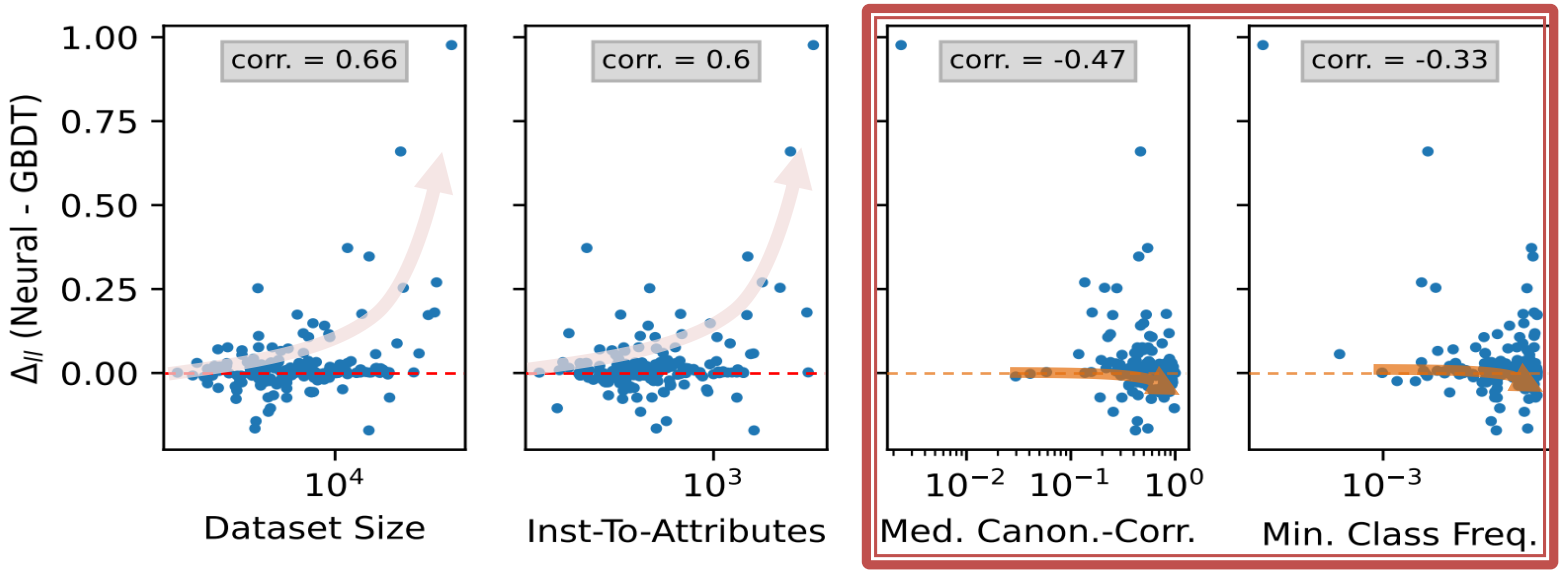


1. Dataset이 클수록 GBDT 계열이 상대적으로 성능이 좋다.
2. Instance의 수가 Feature 수에 비해 클수록 GBDT 계열의 성능이 좋다.

Metafeature Analysis

데이터의 '어떤 특성'이 '특정 알고리즘'을 성능이 좋도록 만들까?

❖ Best GBDT계열 (XGBoost, Light GBM) / NN계열 (SAINT, ResNet) 간 Loss 차이와 Metafeature 분석

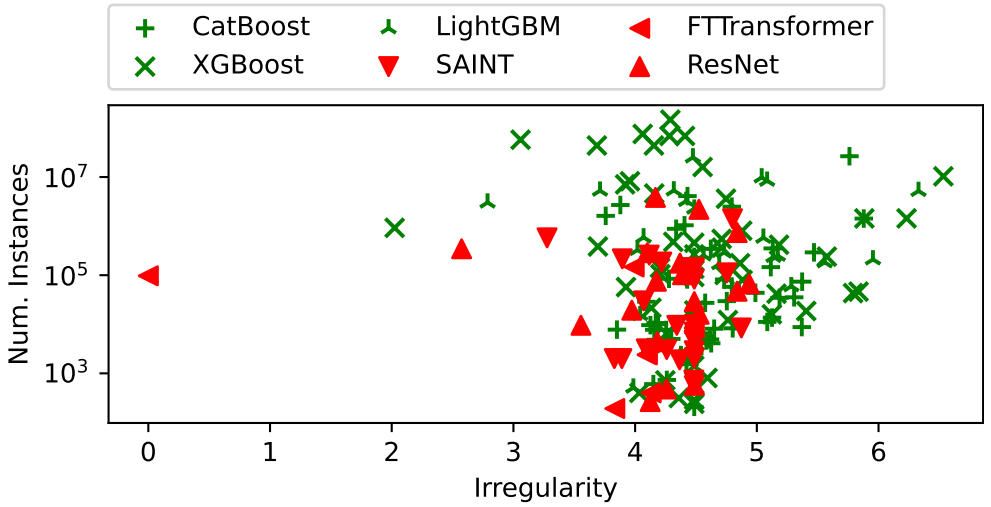


1. Dataset이 클수록 GBDT 계열이 상대적으로 성능이 좋다.
2. Instance의 수가 Feature 수에 비해 클수록 GBDT 계열의 성능이 좋다.
3. Feature와 Label간의 상관관계가 클수록 NN 계열이 성능이 좋다.
4. 최소로 나타나는 Class의 빈도가 클수록 NN 계열이 성능이 좋다.

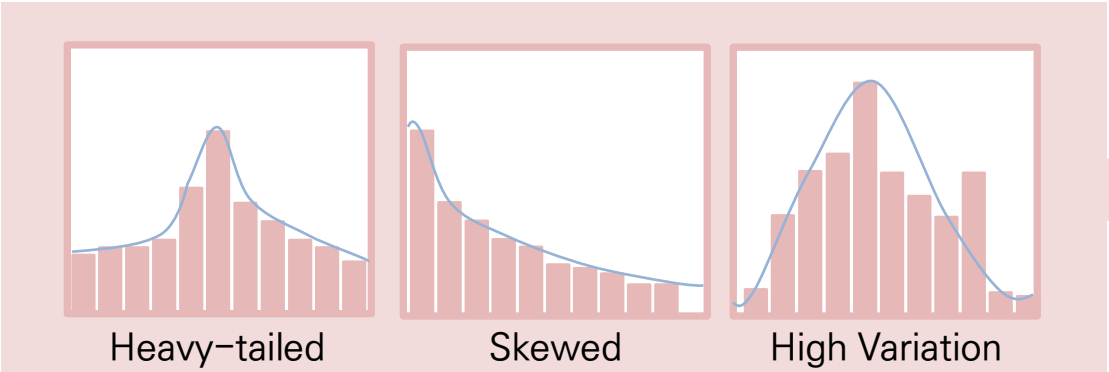
Metafeature Analysis

데이터의 '어떤 특성'이 '특정 알고리즘'을 성능이 좋도록 만들까?

❖ Irregularity



Alg. 1	Alg. 2	Corr.	Attribute Name
CatBoost	ResNet	-0.25	Maximum skewness of all features.
CatBoost	ResNet	-0.24	Range of the skewness of all features.
CatBoost	ResNet	-0.23	Log of the standard deviation of the kurtosis of all features.
CatBoost	ResNet	-0.23	Log of the standard deviation of the skewness of all features.
CatBoost	ResNet	0.22	Log of the median of the absolute value of the covariance between all feature pairs.
CatBoost	ResNet	0.21	Log of the median of the standard deviation of all features.
CatBoost	ResNet	0.21	Log of the median of the variance of all features.
CatBoost	ResNet	0.20	Log of the median of the maximum value of all features.
CatBoost	ResNet	0.20	Best performance of a naive Bayes classifier trained over 10-fold CV.

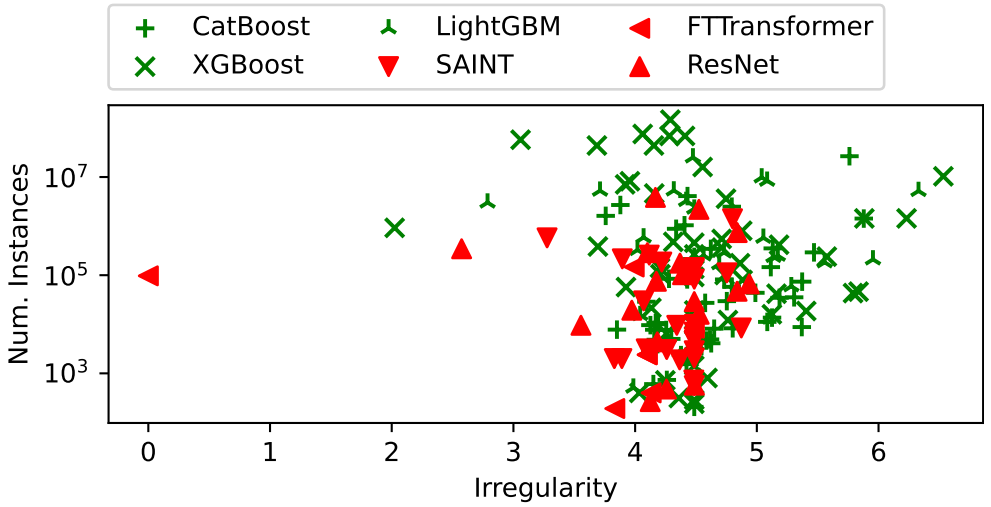


"Irregular"

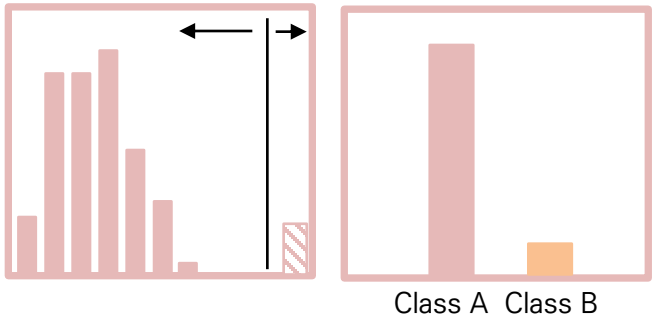
Metafeature Analysis

데이터의 '어떤 특성'이 '특정 알고리즘'을 성능이 좋도록 만들까?

❖ Irregularity



Alg. 1	Alg. 2	Corr.	Attribute Name
CatBoost	ResNet	-0.25	Maximum skewness of all features.
CatBoost	ResNet	-0.24	Range of the skewness of all features.
CatBoost	ResNet	-0.23	Log of the standard deviation of the kurtosis of all features.
CatBoost	ResNet	-0.23	Log of the standard deviation of the skewness of all features.
CatBoost	ResNet	0.22	Log of the median of the absolute value of the covariance between all feature pairs.
CatBoost	ResNet	0.21	Log of the median of the standard deviation of all features.
CatBoost	ResNet	0.21	Log of the median of the variance of all features.
CatBoost	ResNet	0.20	Log of the median of the maximum value of all features.
CatBoost	ResNet	0.20	Best performance of a naive Bayes classifier trained over 10-fold CV.



"Irregular" Data set은
불균형 데이터, 극단값 등에 강건한 GBDT 계열이 유리

Tabzilla

❖ TabZilla Benchmark : 'Hardest' Tabular Datasets (36 Datasets Selected from 176 Datasets)

+) 기준

1) Baseline 알고리즘이 최고 성능에서 20% 이내의 성능 확인된 데이터셋 제외.

2) 소수의 알고리즘만 잘 작동하는 데이터셋.

3) GBDTs가 최고 성능보다 10% 이상 낮은 성능을 보이는 데이터셋.

Table 4: The TabZilla Benchmark Suite. Columns show the hardness metrics used as selection criteria, dataset attributes, and the top-performing algorithms. 'Std. Kurtosis' indicates the std. dev. of the kurtosis of all features. Hardness metrics that meet our selection criteria are shown in bold.

	Hardness Metrics			Dataset Attributes			Top 3 Algs.		
Dataset	base	4th-best	GBDT	*N*	# feats.	Std. Kurtosis	1st	2nd	3rd
credit-g	**0.26**	**0.13**	**0.12**	1 000	21	1.92	ResNet	FTTransformer	CatBoost
jungle-chess	**0.30**	**0.18**	**0.17**	44 819	7	0.08	SAINT	TabNet	LightGBM
MiniBooNE	**0.20**	**0.09**	0.00	130 064	51	12162.65	LightGBM	XGBoost	CatBoost
albert	**0.42**	**0.28**	0.00	425 240	79	1686.90	CatBoost	XGBoost	ResNet
electricity	**0.46**	**0.38**	0.00	45 312	9	2693.51	LightGBM	XGBoost	FTTransformer
elevators	**0.36**	**0.08**	0.05	16 599	19	2986.50	TabNet	XGBoost	CatBoost
guillermo	**0.35**	**0.60**	0.00	20 000	4 297	NaN	XGBoost	RandomForest	TabNet
higgs	**0.41**	**0.10**	0.07	98 050	29	15.53	ResNet	XGBoost	LightGBM
nomao	**0.22**	**0.18**	0.00	34 465	119	1100.34	LightGBM	XGBoost	CatBoost
100-plants-texture	**0.20**	**0.11**	0.00	1 599	65	17.66	CatBoost	XGBoost	ResNet
poker-hand	**0.58**	**0.98**	0.00	1 025 009	11	0.08	XGBoost	CatBoost	KNN
profb	**0.39**	**0.38**	0.00	672	10	0.95	CatBoost	DeepFM	MLP-rtdl
socmob	**0.24**	**0.10**	0.00	1 156	6	NaN	XGBoost	CatBoost	ResNet
audiology	**0.43**	0.03	0.00	226	70	NaN	STG	XGBoost	ResNet
splice	**0.30**	0.03	0.00	3 190	61	NaN	LightGBM	XGBoost	CatBoost
vehicle	0.05	**0.10**	**0.10**	846	19	15.16	TabPFN	SVM	DANet
Australian	0.15	**0.08**	0.00	690	15	2.00	CatBoost	XGBoost	TabPFN
Bioresponse	0.07	**0.07**	0.00	3 751	1 777	328.77	LightGBM	XGBoost	CatBoost
GesturePhase	0.08	**0.08**	0.00	9 872	33	52.18	LightGBM	XGBoost	CatBoost
SpeedDating	0.18	**0.14**	0.00	8 378	121	36.43	XGBoost	CatBoost	LightGBM
ada-agnostic	0.12	**0.11**	0.00	4 562	49	NaN	XGBoost	CatBoost	LightGBM
airlines	**0.20**	**0.18**	0.00	539 382	8	2.01	LightGBM	XGBoost	CatBoost
artificial-characters	0.13	**0.11**	0.00	10 218	8	0.63	XGBoost	LightGBM	CatBoost
colic	0.13	**0.11**	0.00	368	27	4.00	CatBoost	XGBoost	FTTransformer
credit-approval	0.12	**0.08**	0.00	690	16	74.77	CatBoost	TabPFN	XGBoost
heart-h	0.10	**0.07**	0.08	294	14	NaN	DeepFM	TabTransformer	NAM
jasmine	0.13	**0.13**	0.00	2 984	145	47.60	CatBoost	XGBoost	LightGBM
kc1	0.14	**0.07**	0.00	2 109	22	28.34	CatBoost	XGBoost	FTTransformer
lymph	0.14	**0.08**	0.00	148	19	17.04	XGBoost	DANet	SAINT
mfeat-fourier	0.00	**0.07**	0.07	2 000	77	0.64	SVM	SAINT	STG
phoneme	0.10	**0.15**	0.00	5 404	6	1.23	XGBoost	LightGBM	RandomForest
qsar-biodeg	0.08	**0.08**	0.05	1 055	42	93.24	TabPFN	CatBoost	SAINT
balance-scale	0.07	0.05	**0.16**	625	5	0.02	TabPFN	SAINT	MLP
cnae-9	0.11	0.04	**0.10**	1 080	857	NaN	TabTransformer	STG	MLP-rtdl
mfeat-zernike	0.00	0.04	**0.10**	2 000	48	1.42	SVM	DANet	ResNet
monks-problems-2	0.04	0.00	**0.17**	601	7	NaN	SAINT	ResNet	MLP-rtdl

Summary

❖ GBDT 계열 알고리즘이 평균적으로 NN 계열보다 Tabular Data에서는 나은 성능을 보임

1) “Irregular” 특성은 GBDT 계열에서 뚜렷한 양호한 성능

2) Instance가 많은, Feature 대비 Instance 비율이 높은 데이터에서도 GBDT 계열에서 상대적 양호한 성능

❖ Tabular Data 활용 시 제안 Framework

1) 간단한 Baseline 모델 시도 (Random Forest, Decision Tree 등) → 데이터 특징 파악

2) CatBoost를 가벼운 하이퍼파라미터 튜닝과 함께 시도 (간단한 설정 + 높은 성능)

3) 필요시 NN 계열 및 다른 GBDT 계열 알고리즘 시도

NEURAL OBLIVIOUS DECISION ENSEMBLES FOR DEEP LEARNING ON TABULAR DATA

Sergei Popov
Yandex
sapopov@yandex-team.ru

Stanislav Morozov
Yandex
Lomonosov Moscow State University
stanis-morozov@yandex.ru

Artem Babenko
Yandex
National Research University
Higher School of Economics
artem.babenko@phystech.edu

표형식 데이터의 딥러닝을 위한 신경망 망각 결정 나무
(ICLR 2020, 324회 인용)

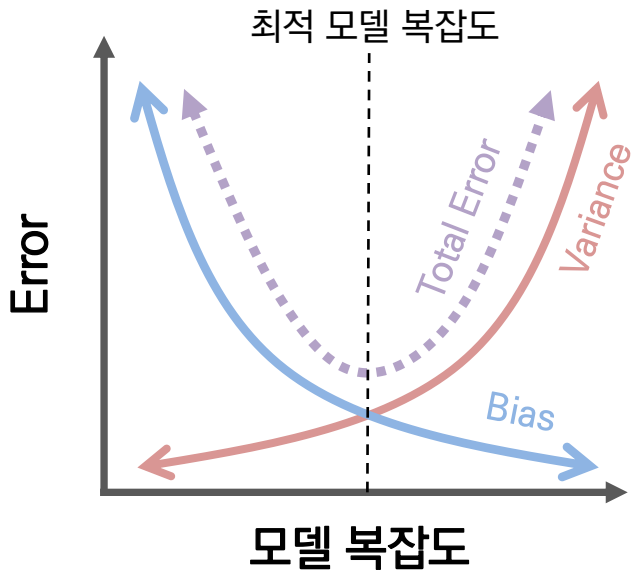
Ensemble

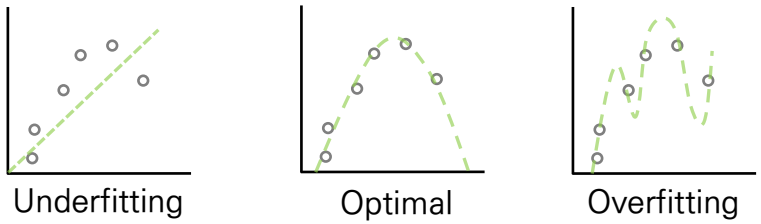
❖ Ensemble : 여러 모델을 결합하여 단일 모델보다 높은 성능 추구

Unity is Strength

"But"

Bias Variance Trade-off



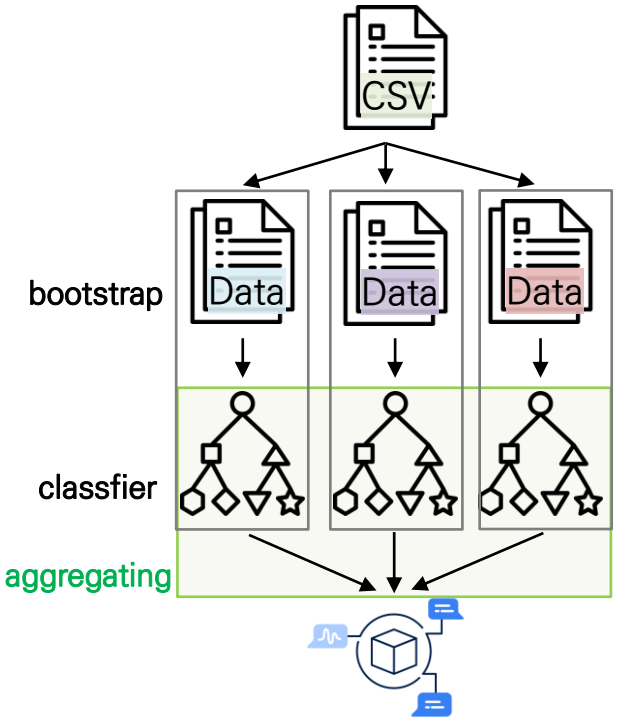


Ensemble

❖ Ensemble 기법

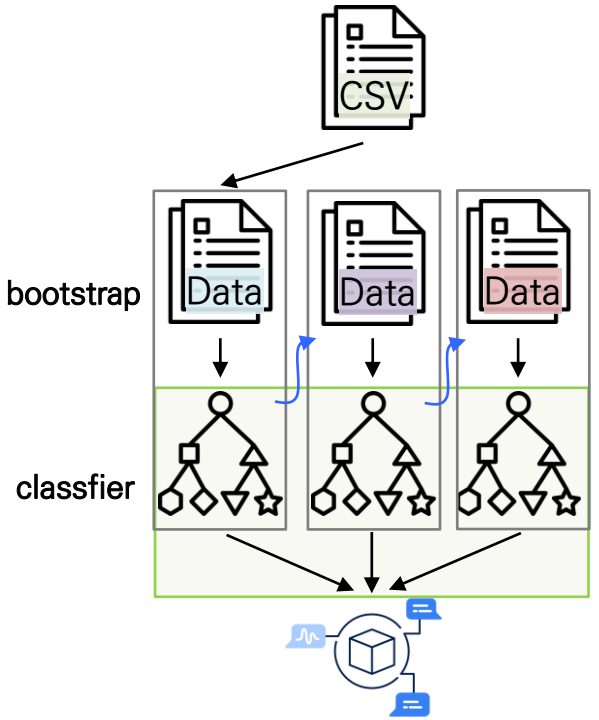
Bagging

Boostrap Aggreagting



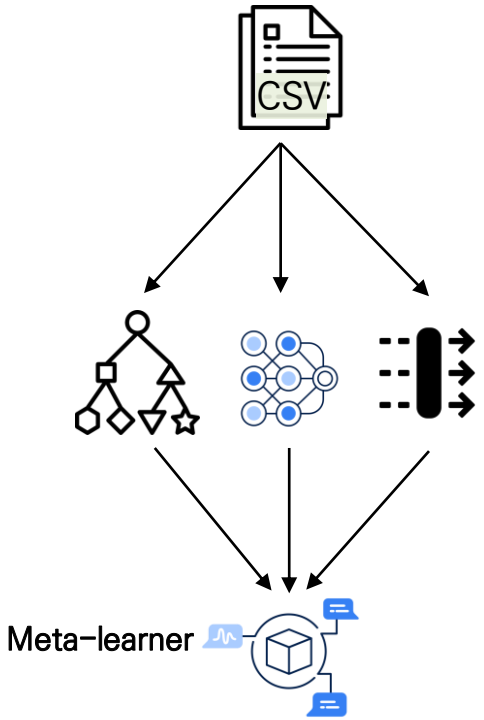
Random Forest

Boosting



Adaboost, Catboost

Stacking

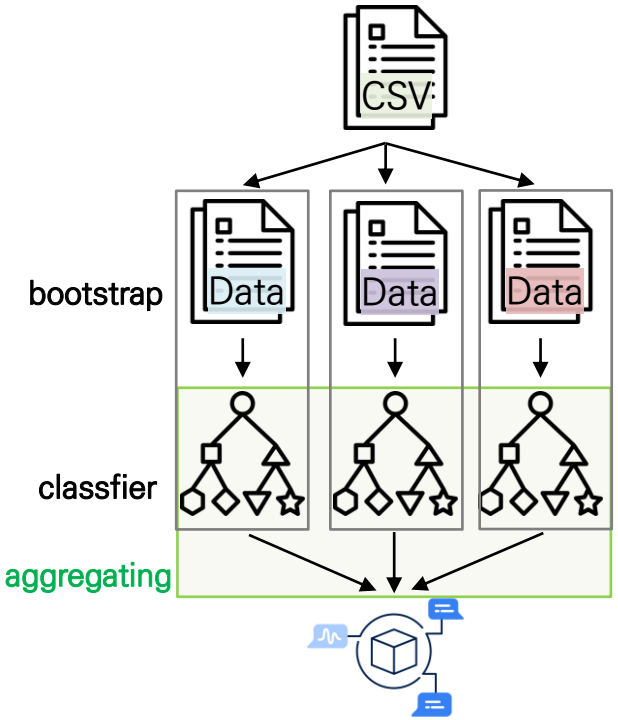


Ensemble

❖ Ensemble 기법

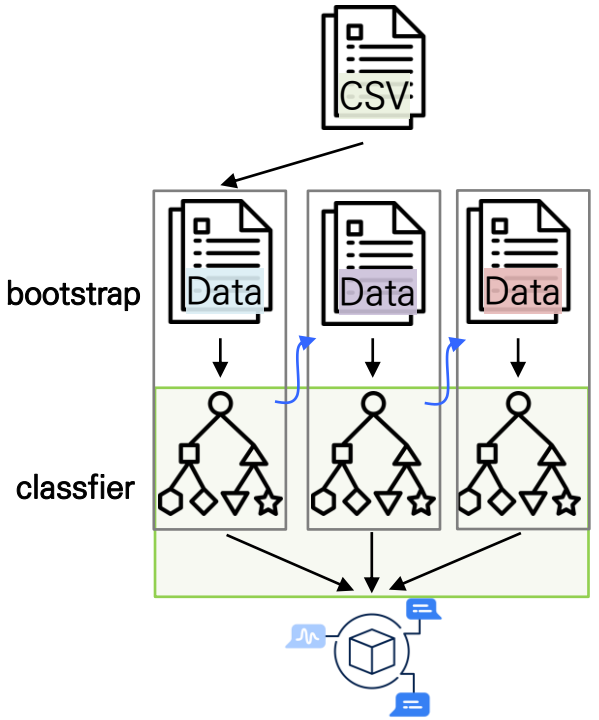
Bagging

Boostrap Aggreagting



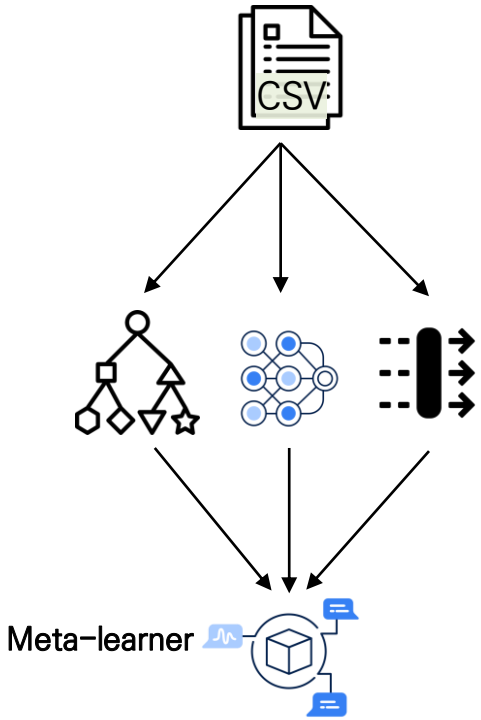
Random Forest

Boosting



Adaboost, Catboost

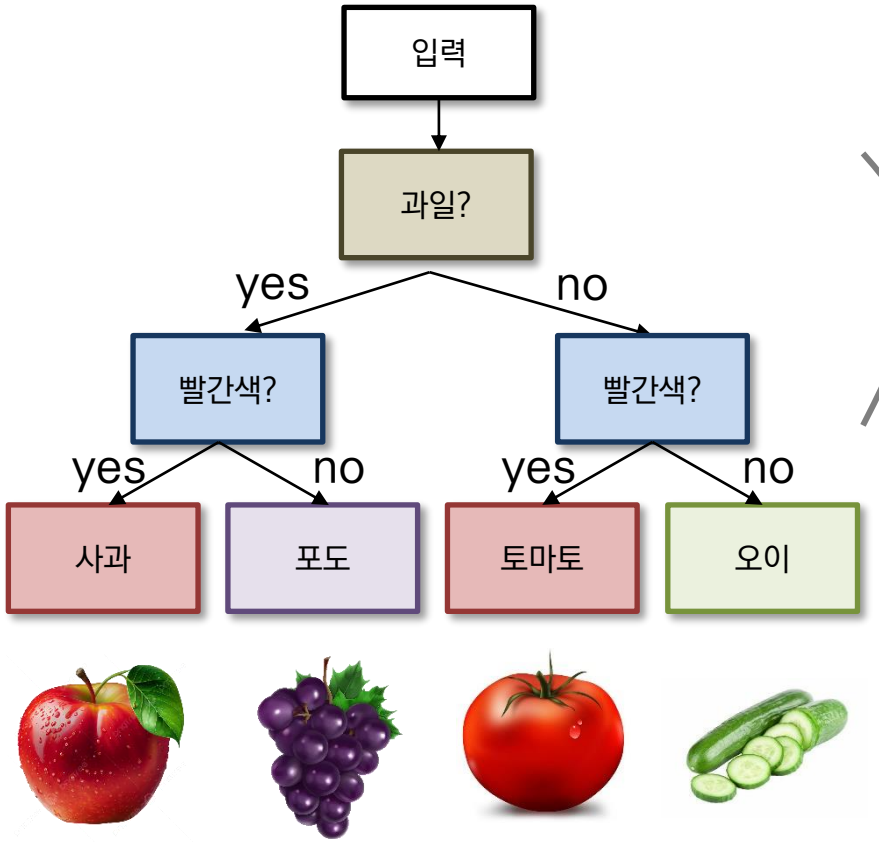
Stacking



Oblivious Decision Tree

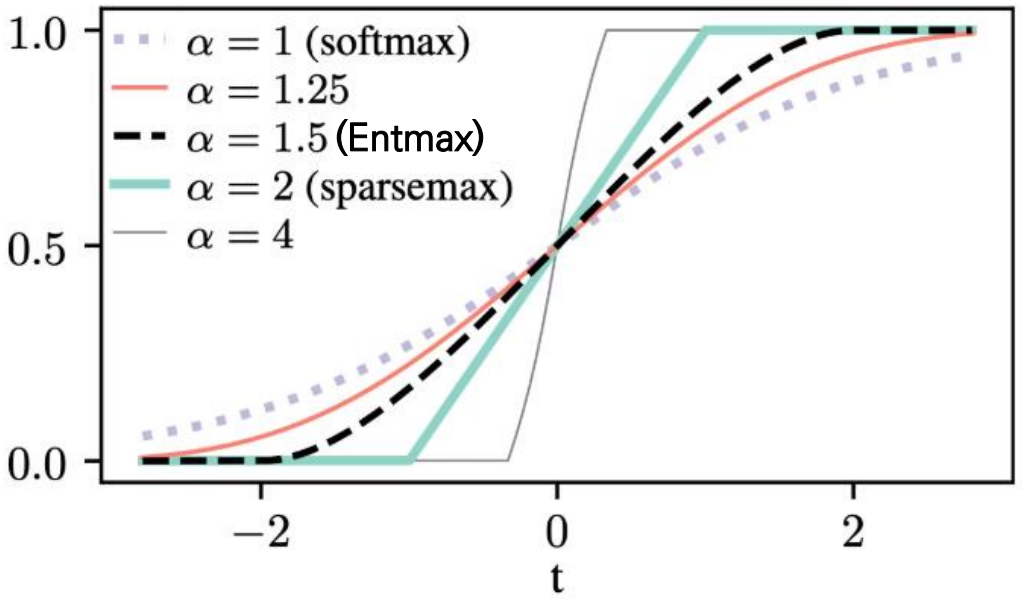
망각결정트리

❖ ODT의 구조 및 특징



- Depth(d) = 2, $2^d = 2^2 = 4$ 개 항목의 Table로 표현됨
- 동일한 깊에 있는 노드에서 동일한 분할 임계값(Splitting Threshold)
- 모델로서의 성능은 약함(weak), 과적합 측면에 강건
  ➔ Gradient Boosting과 잘 활용될 수 있음
  ➔ 병렬계산 가능 (GPU 활용)

Entmax



$$\text{Softmax}(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{n} e^{z_j}}$$ (모든 값이 0 아니며 합은 1)

$$\text{Sparsemax}(z_i) = \max\left(z_i - \tau, 0\right)$$

(일부 값이 0이며 전체 합은 1)

$$\text{Entmax}(z_i) = \max\left(z_i^{\frac{1}{\alpha-1}} - \tau, 0\right)$$

(Softmax / Sparsemax의 중간)

중요하지 않은 특징은 아예 배제, 중요한 특징에는 가중치를 고려
+ 미분 가능

Oblivious Decision Tree

Oblivious Decision Tree

X = 입력 Feature : [A,B,C,D,E,F,G]

Splitting Feature :

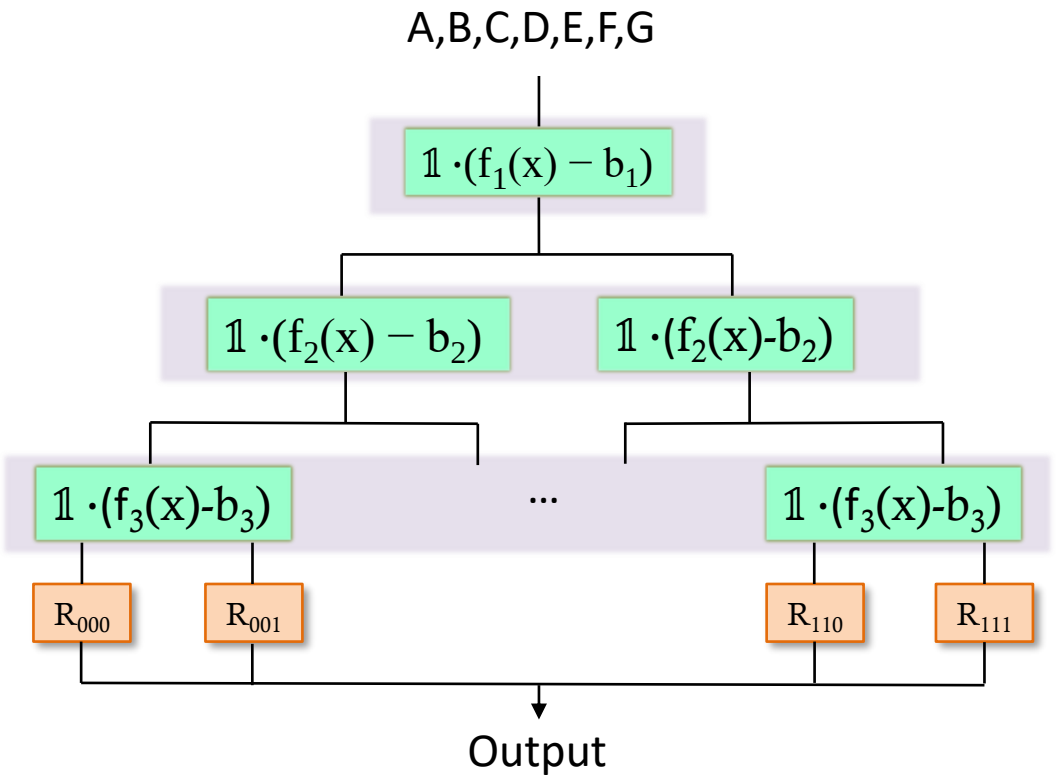
$f_1(x)$, 고정된 단일 Feature (Input)

Threshold

b_1 , 분할 평가 후 가장 잘 분할하는 값 (고정)

분기

$\mathbb{1} \cdot (f_1(x) - b_1)$ = 이진 분류

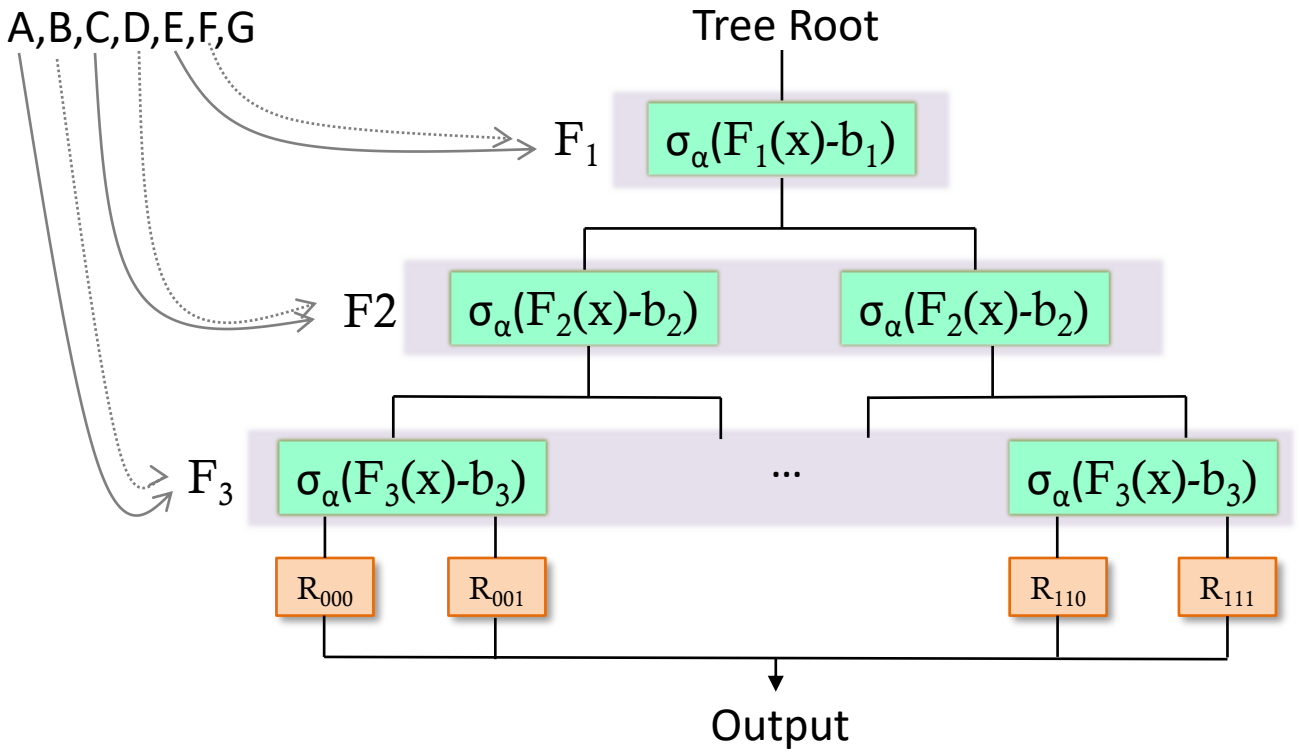


$$h(x) = R[\mathbb{1} \cdot (f_1(x) - b_1), \ldots, \mathbb{1} \cdot (f_d(x) - b_d)]$$

Differential Oblivious Decision Ensemble

Differntiable Oblivious Decision Tree

X = 입력 Feature : [A,B,C,D,E,F,G]



Splitting Feature ← Differential

F(x)−b

F = 학습가능한 "Feature 선택 Matrix"

=> $\sum_i$ wixi -b

Threshold

학습가능한 Parameter b

분기

Choice tensor = σ_α(F(x)−b)

$$\hat{h}(x) = \sum_{i_1,\ldots i_d \in \{0,1\}^d} R_{i_1,\ldots,i_d} \cdot C_{i_1,\ldots,i_d}(x)$$

Response Tensor / Choice Tensor

- Response Tensor : 각 경로에서의 출력값, 최종 예측에 기여.
  R∈ R $^{2^{\wedge}d \times l}$ (2^d : Depth d Tree 가능한 분기 수, l : 출력차원수(Hyper Parameter,1~3)
- Choice Tensor : 입력 x가 각 트리 상태로 분배될 확률을 나타냄. C(x)∈$R^{2^{\wedge}d}$, $C(x) = \begin{bmatrix} c_1(x) \\ 1-c_1(x) \end{bmatrix} \otimes \begin{bmatrix} c_2(x) \\ 1-c_2(x) \end{bmatrix} \otimes \cdots \otimes \begin{bmatrix} c_d(x) \\ 1-c_d(x) \end{bmatrix}$

Deeper NODE Architecture

❖ Layer간 Concatenation

−. 모든 Layer는 이전 Layer의 모든 출력을 입력으로 사용

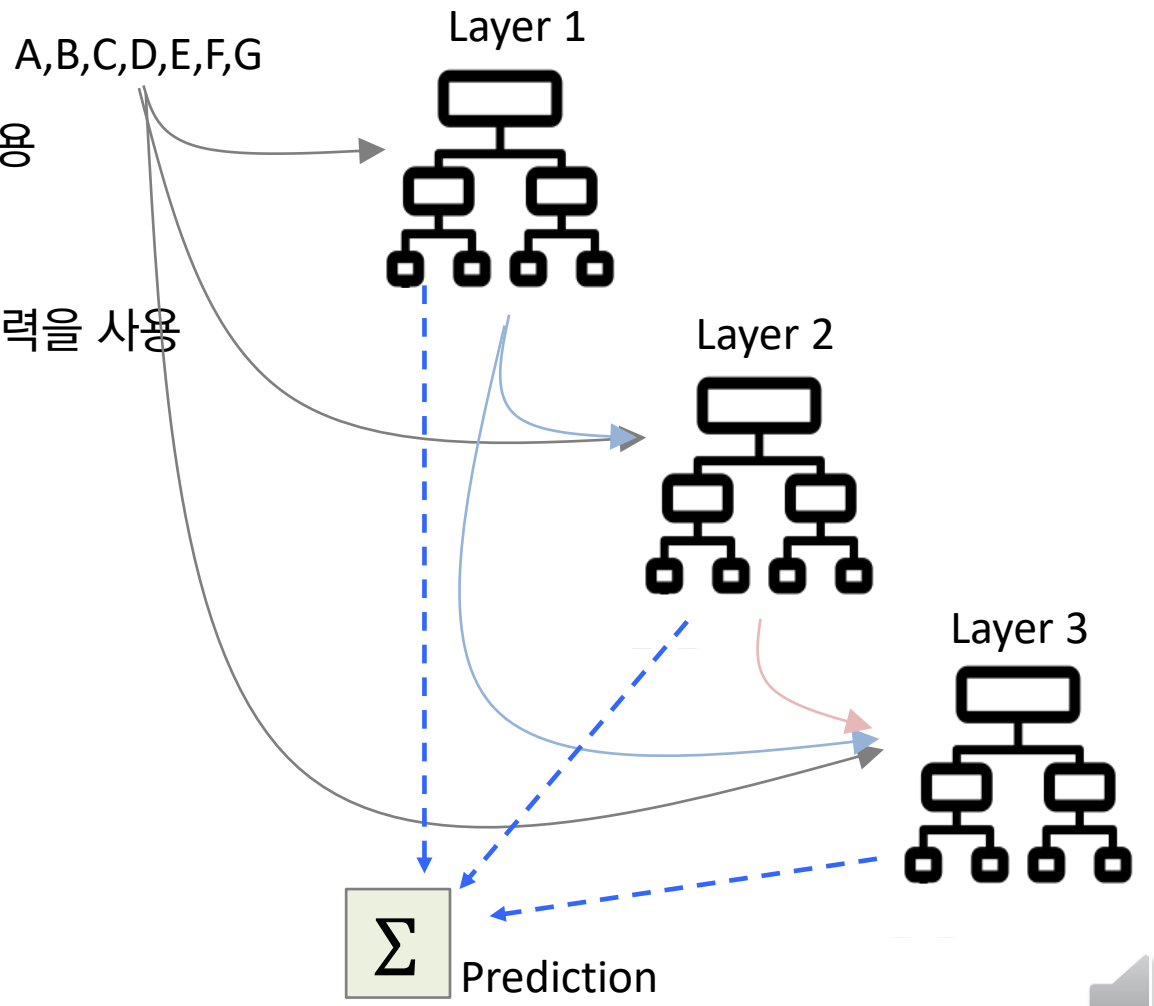
⇒ Shallow / Deep 모든 규칙 학습 가능

−. i번째 Layer의 단일 Tree는 i−1번째 Layer까지의 출력을 사용

⇒ Feature 의 복잡한 종속성을 포착 가능

+) 최종 예측값이 모든 Tree에서 예측한 값의 평균

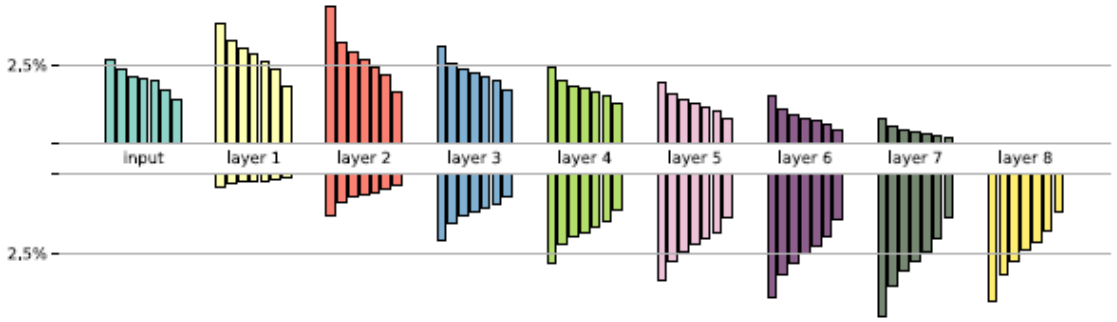
−. “Backpropagation을 통해 학습”

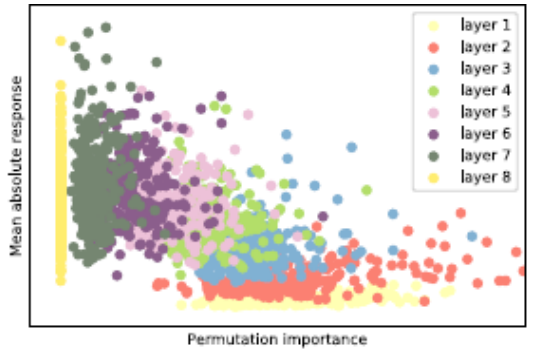


Result

	Epsilon	YearPrediction	Higgs	Microsoft	Yahoo	Click
Tuned hyperparameters						
CatBoost	$0.1113 \pm 4e{-4}$	79.67 ± 0.12	$0.2378 \pm 1e{-4}$	$0.5565 \pm 2e{-4}$	$0.5632 \pm 3e{-4}$	$0.3401 \pm 2e{-3}$
XGBoost	$0.1112 \pm 6e{-4}$	78.53 ± 0.09	$0.2328 \pm 3e{-4}$	$\mathbf{0.5544 \pm 1e{-4}}$	$\mathbf{0.5420 \pm 4e{-4}}$	$0.3334 \pm 2e{-3}$
FCNN	$0.1041 \pm 2e{-4}$	79.99 ± 0.47	$0.2140 \pm 2e{-4}$	$0.5608 \pm 4e{-4}$	$0.5773 \pm 1e{-3}$	$0.3325 \pm 2e{-3}$
NODE	$\mathbf{0.1034 \pm 3e{-4}}$	$\mathbf{76.21 \pm 0.12}$	$\mathbf{0.2101 \pm 5e{-4}}$	$0.5570 \pm 2e{-4}$	$0.5692 \pm 2e{-4}$	$\mathbf{0.3312 \pm 2e{-3}}$
mGBDT	OOM	80.67	OOM	OOM	OOM	OOM
DeepForest	0.1179	—	0.2391	—	—	0.3333

Permutation Feature Importance (PFI)





1. 초기 레이어: 정보를 추출하고 중요한 피처를 생성.
2. 깊은 레이어: 이전 레이어에서 생성된 피처를 활용하여 최종 예측을 개선.

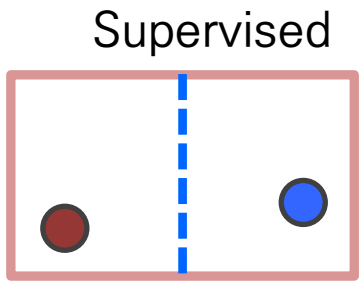
Binning as a Pretext Task: Improving Self-Supervised Learning in Tabular Domains

Kyungeun Lee [1] Ye Seul Sim [1] Hye-Seung Cho [1] Moonjung Eo [1] Suhee Yoon [1] Sanghyu Yoon [1]
Woohyung Lim [1]

사전과제로서의 비닝 : 표형식 데이터에서 자기지도학습의 발전
(ICML 2024, 4회 인용)

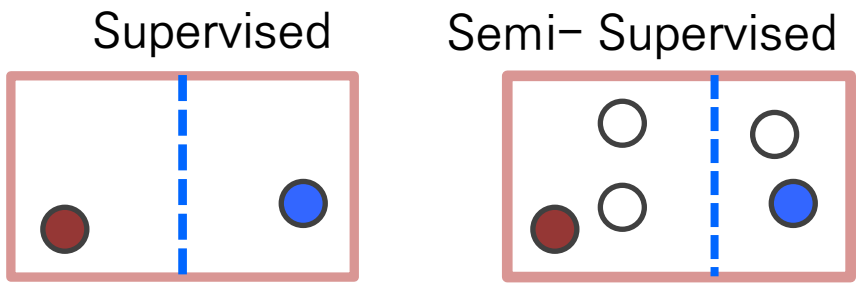
Unsupervised tabular deep learning

❖ Supervised, Semi-Supervised, Unsupervised Learning and Self-supervised Learning



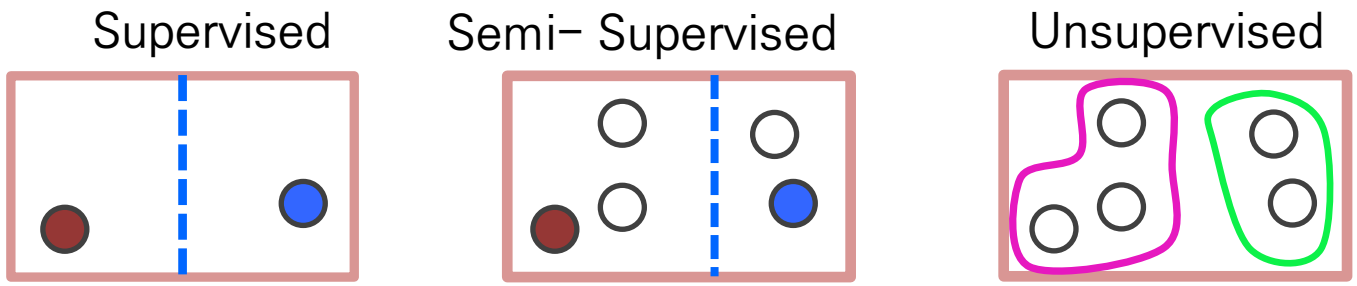
Unsupervised tabular deep learning

❖ Supervised, Semi-Supervised, Unsupervised Learning and Self-supervised Learning



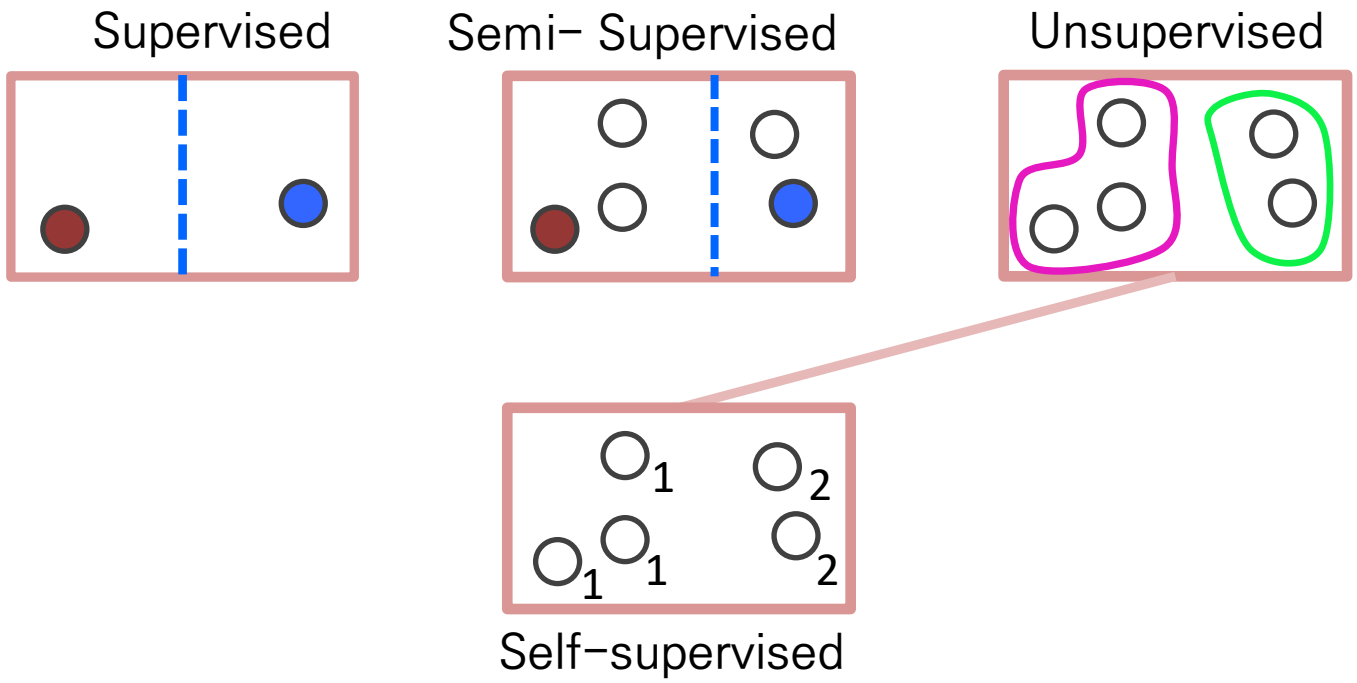
Unsupervised tabular deep learning

❖ Supervised, Semi-Supervised, Unsupervised Learning and Self-supervised Learning



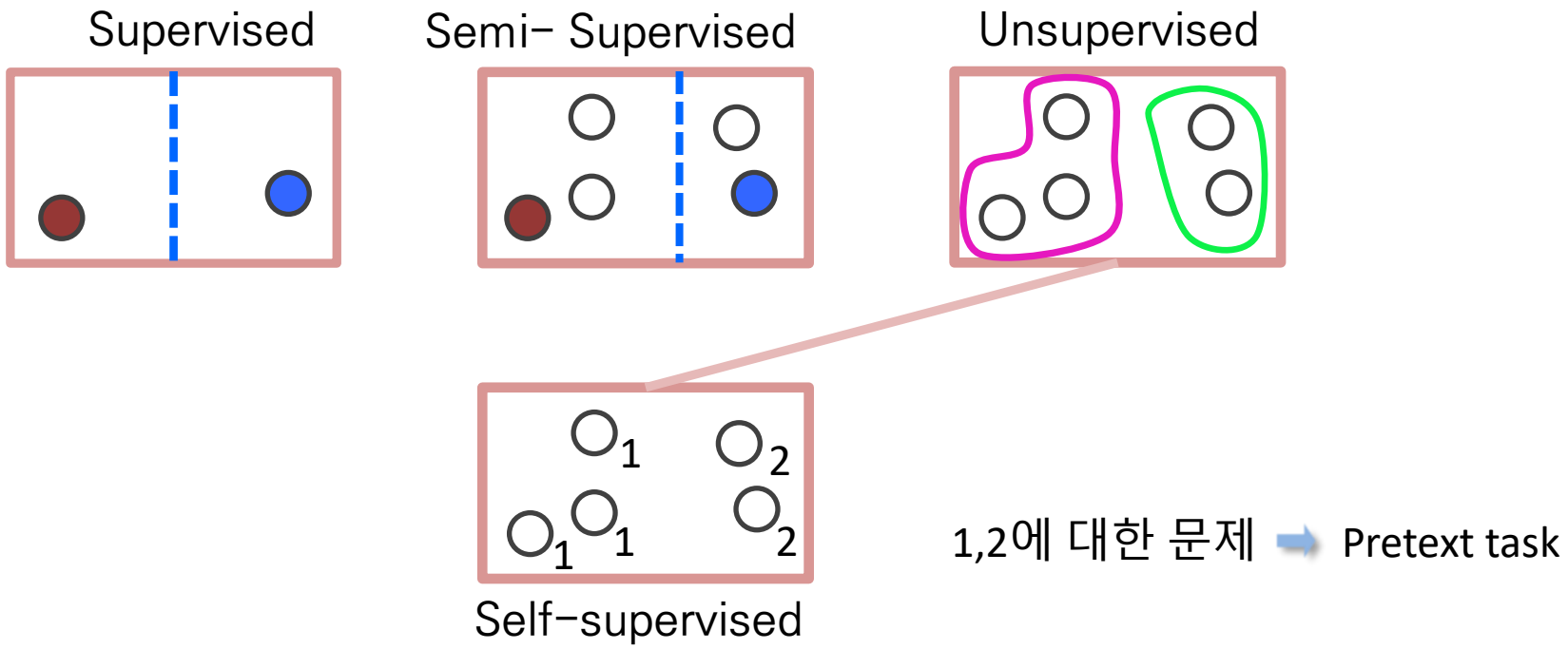
Unsupervised tabular deep learning

❖ Supervised, Semi-Supervised, Unsupervised Learning and Self-supervised Learning

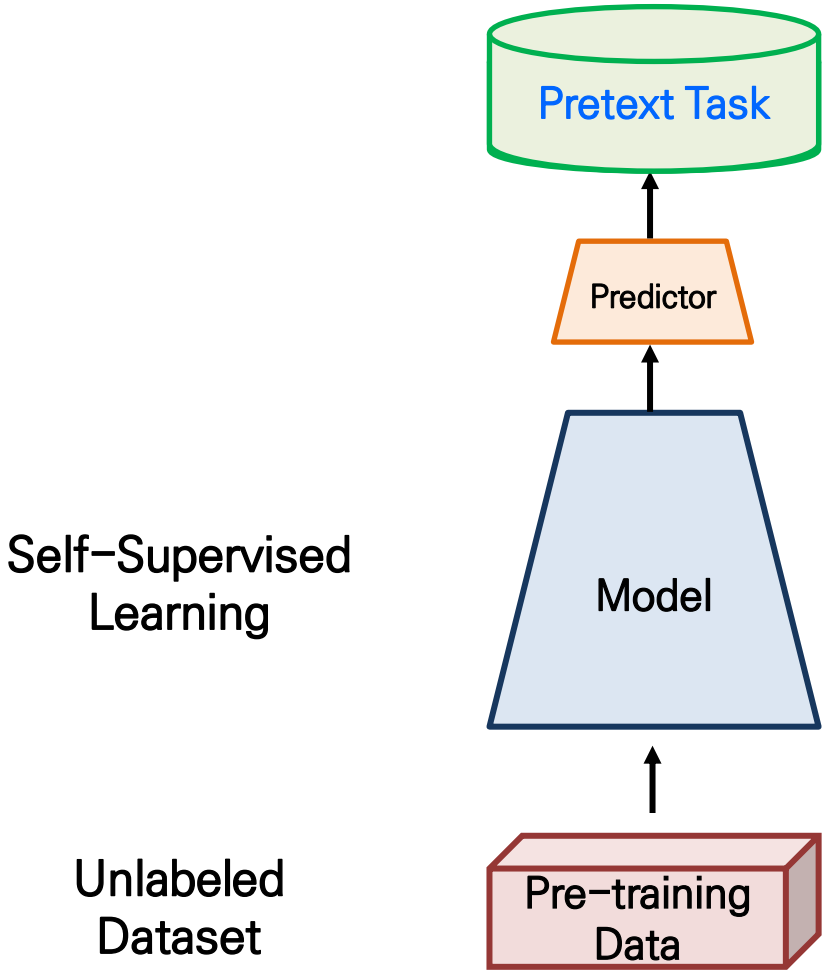


Unsupervised tabular deep learning

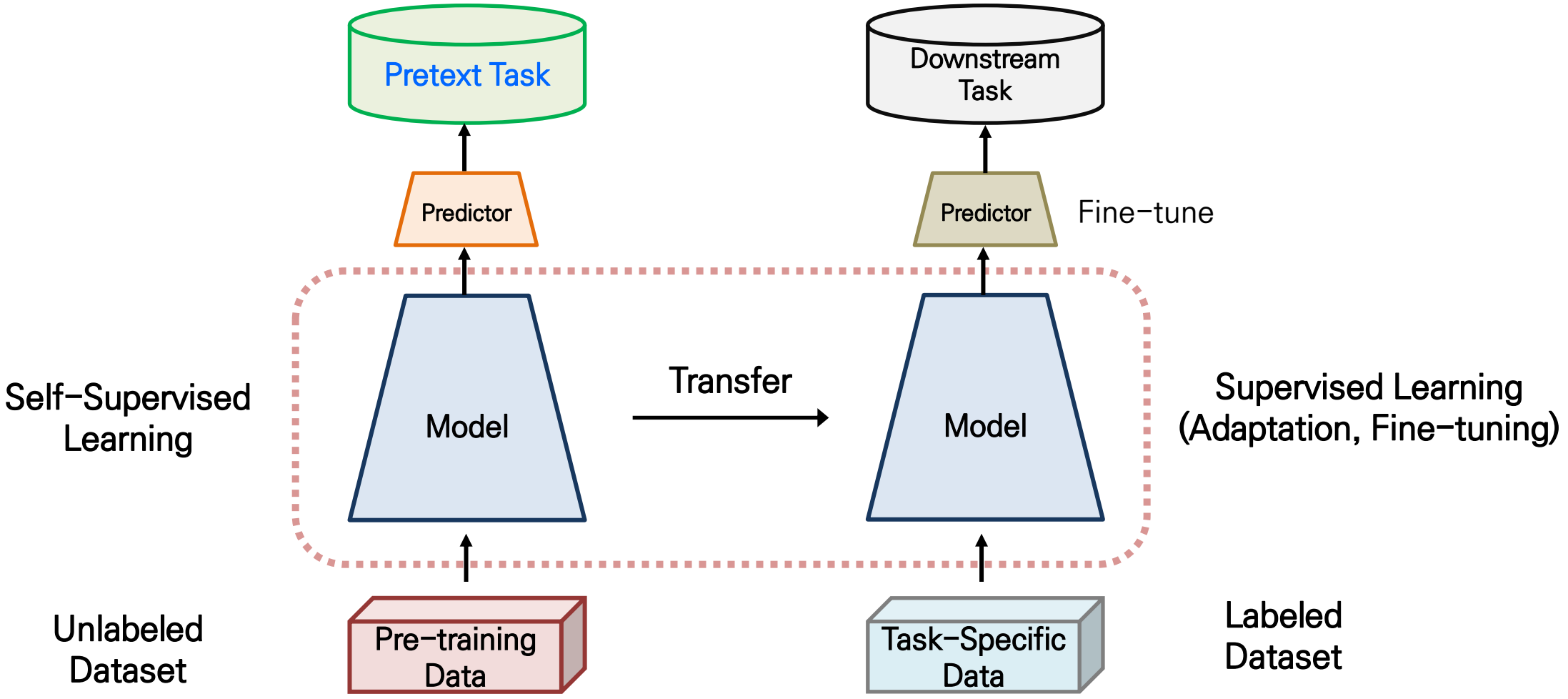
❖ Supervised, Semi-Supervised, Unsupervised Learning and Self-supervised Learning



Self-Supervised Learning

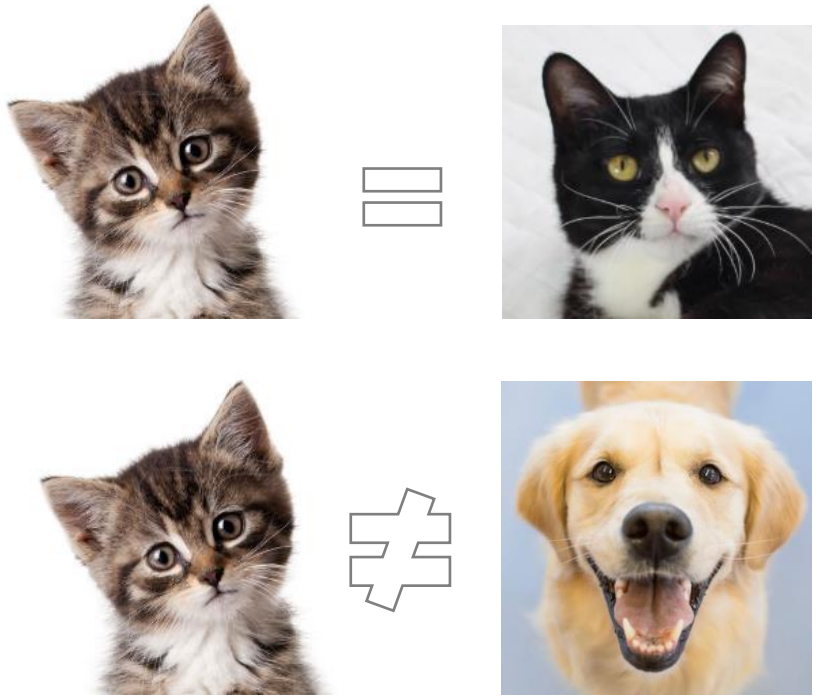


Self-Supervised Learning



In the Tabular Domain, Contrastive Learning for SSL

❖ Contrastive Learning



유사한 데이터 쌍은 더 가깝게,
다른 데이터 쌍은 더 멀게하는 표현 공간 학습

SCARF: SELF-SUPERVISED CONTRASTIVE LEARNING USING RANDOM FEATURE CORRUPTION

Dara Bahri, Heinrich Jiang, Yi Tay, Donald Metzler
Google Research
{dbahri,heinrichj,yitay,metzler}@google.com

Contrastive Mixup: Self- and Semi-Supervised learning for Tabular Domain

Sajad Darabi
UCLA
sajad.darabi@cs.ucla.edu

Shayan Fazeli
UCLA
shayan.fazeli@cs.ucla.edu

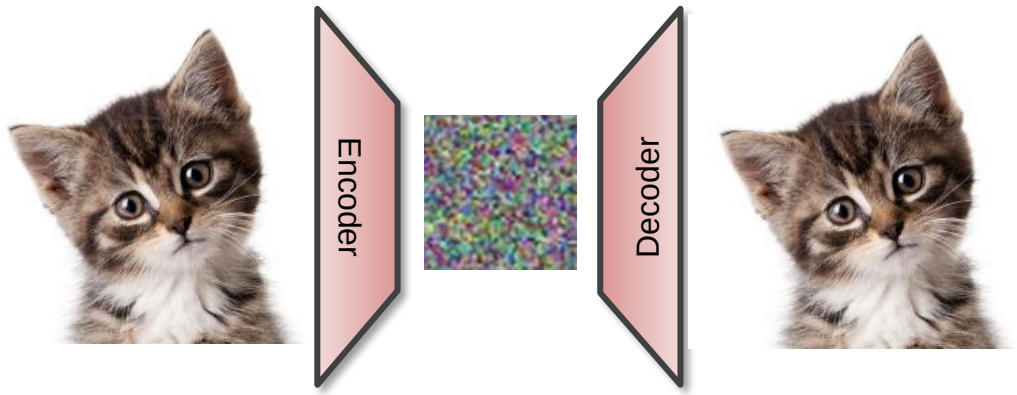
Ali Pazokitoroudi
UCLA
alipazoki@cs.ucla.edu

Sriram Sankararaman
UCLA
sriram@cs.ucla.edu

Majid Sarrafzadeh
UCLA
majid@cs.ucla.edu

In the Tabular Domain, Auto Encoder for SSL

❖ Auto Encoder



데이터를 압축(encode), 복원(decode)하여
효율적으로 표현하는 잠재 표현 학습

VIME: Extending the Success of Self- and Semi-supervised Learning to Tabular Domain

Jinsung Yoon
Google Cloud AI, UCLA
jinsungyoon@google.com

Yao Zhang
University of Cambridge
yz555@cam.ac.uk

James Jordon
University of Oxford
james.jordon@wolfson.ox.ac.uk

Mihaela van der Schaar
University of Cambridge
UCLA, Alan Turing Institute
mv472@cam.ac.uk

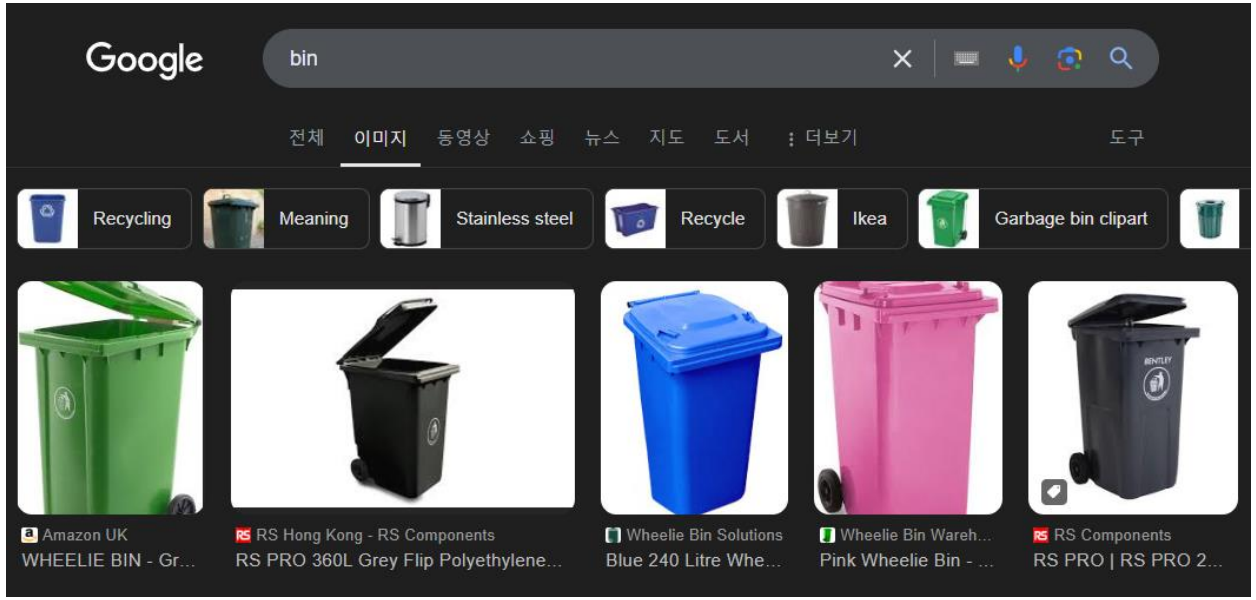
SubTab: Subsetting Features of Tabular Data for Self-Supervised Representation Learning

Talip Uçar, Ehsan Hajiramezanali, Lindsay Edwards

Respiratory and Immunology, R&D, AstraZeneca
{talip.ucar, ehsan.hajiramezanali, lindsay.edwards}@astrazeneca.com

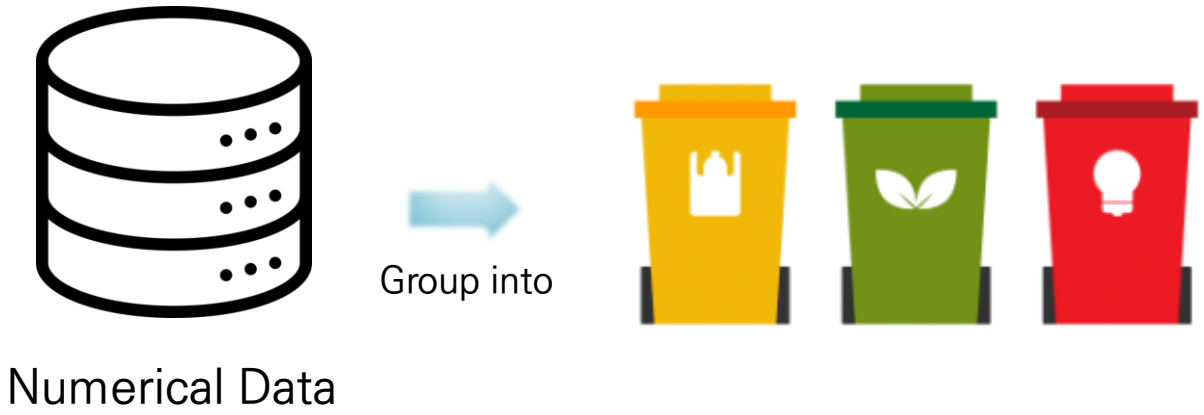
Binning as a Pretext Task

❖ BIN?



Binning as a Pretext Task

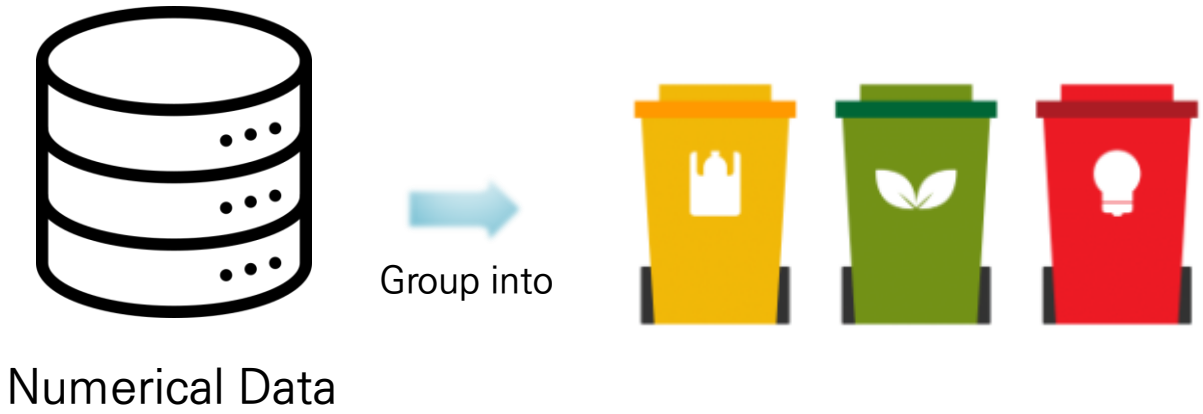
❖ BIN? Binning?



연속형 변수를 특정구간으로 나누어 범주형(Nominal), 순위형(Ordinal) 변수로 Discretization

Binning as a Pretext Task

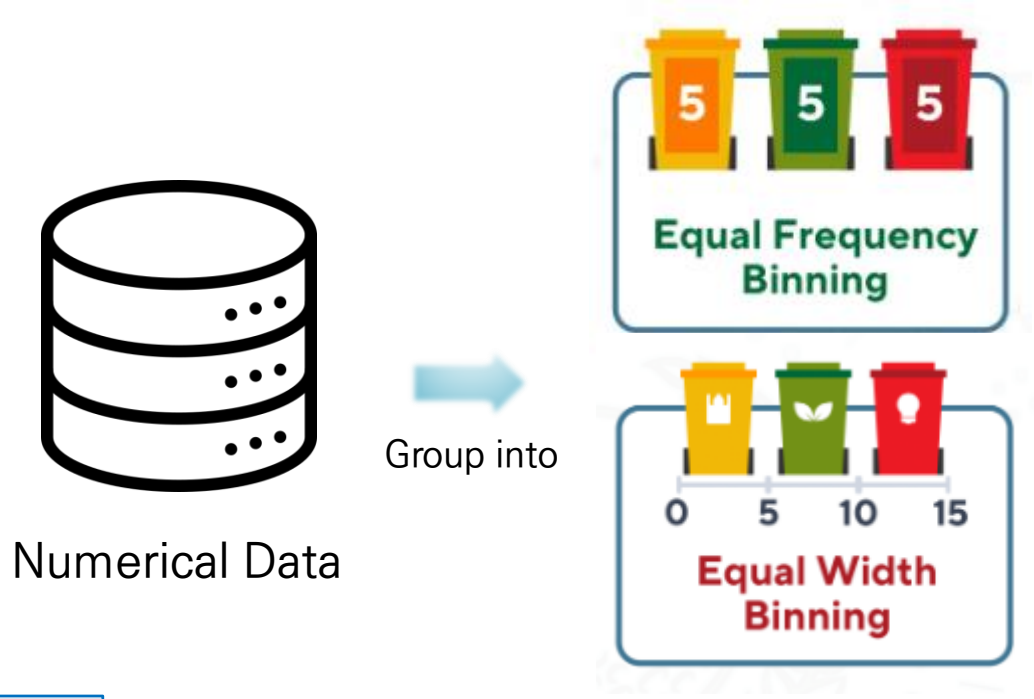
❖ BIN? Binning?



연속형 변수를 특정구간으로 나누어 범주형(Nominal), 순위형(Ordinal) 변수로 Discretization

Binning as a Pretext Task

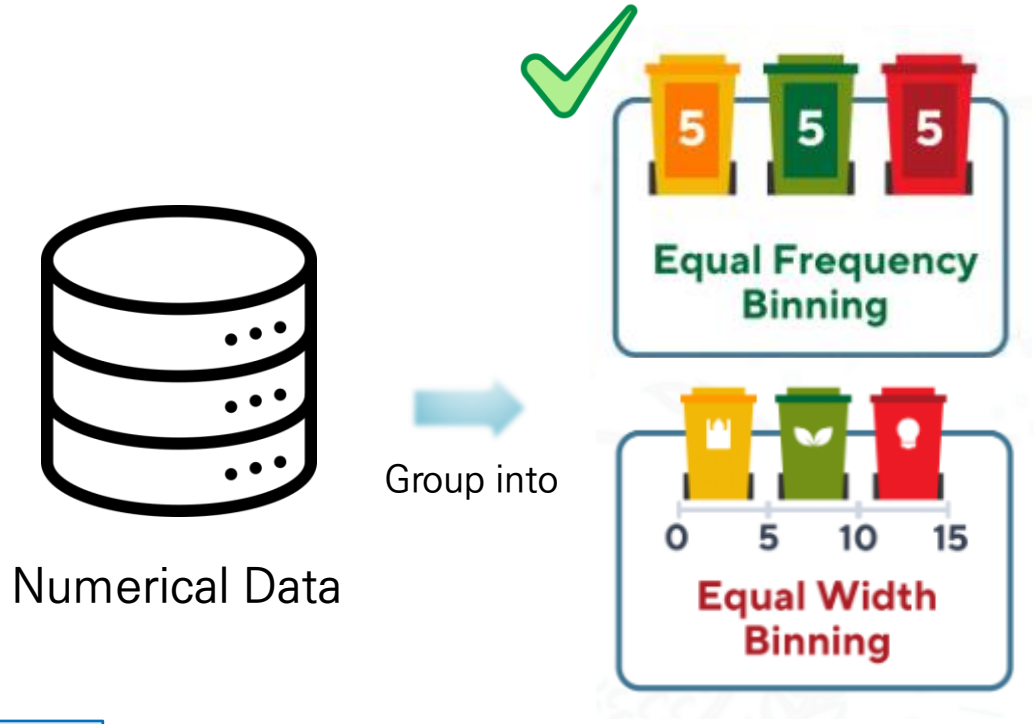
❖ BIN? Binning?



연속형 변수를 특정구간으로 나누어 범주형(Nominal), 순위형(Ordinal) 변수로 Discretization

Binning as a Pretext Task

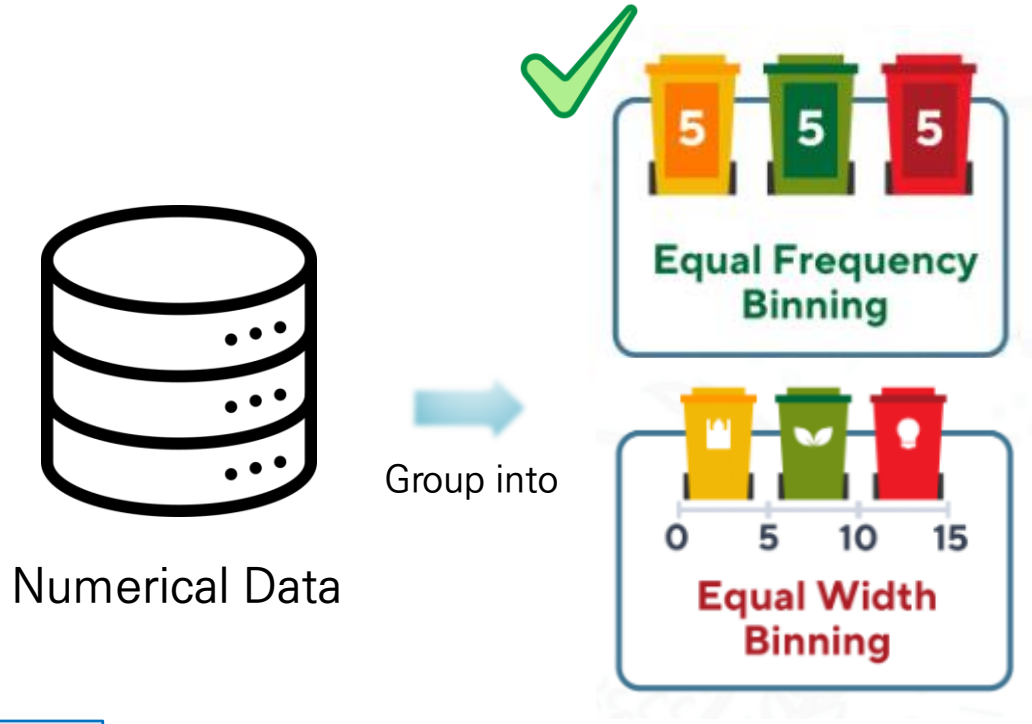
❖ BIN? Binning?



연속형 변수를 특정구간으로 나누어 범주형(Nominal), 순위형(Ordinal) 변수로 Discretization

Binning as a Pretext Task

❖ BIN? Binning?

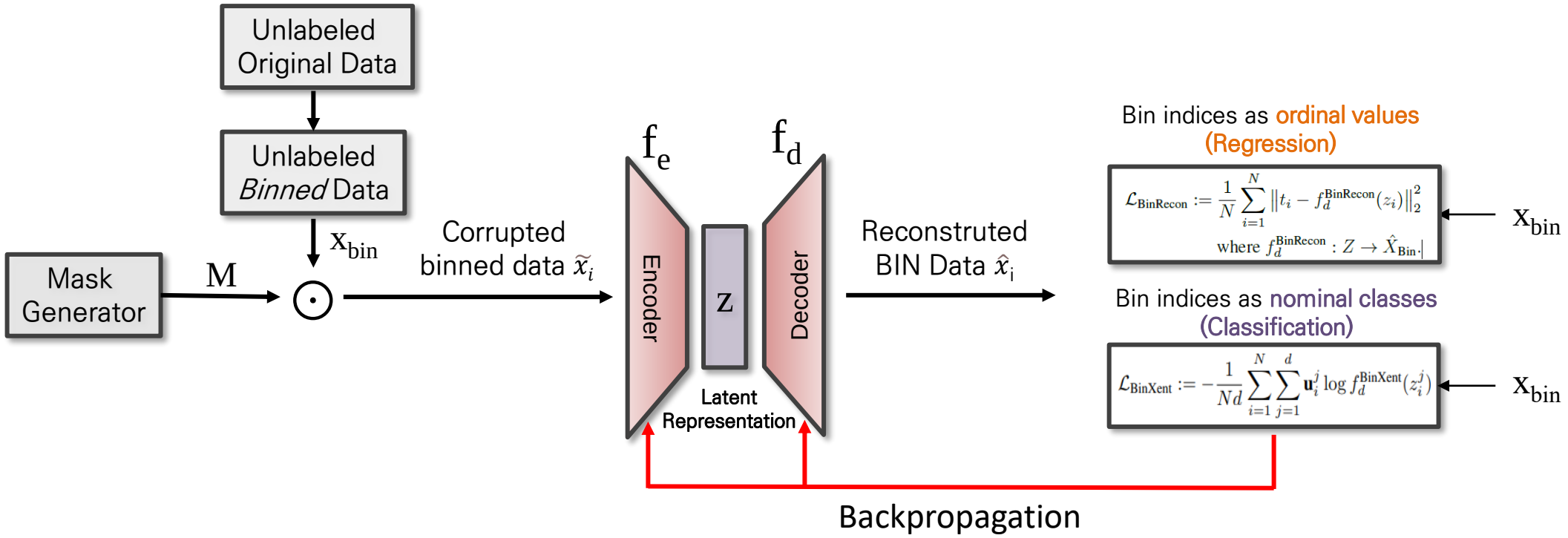


Simplify Data
Capturing Irregular function
Robust to the minor error

연속형 변수를 특정구간으로 나누어 범주형(Nominal), 순위형(Ordinal) 변수로 Discretization

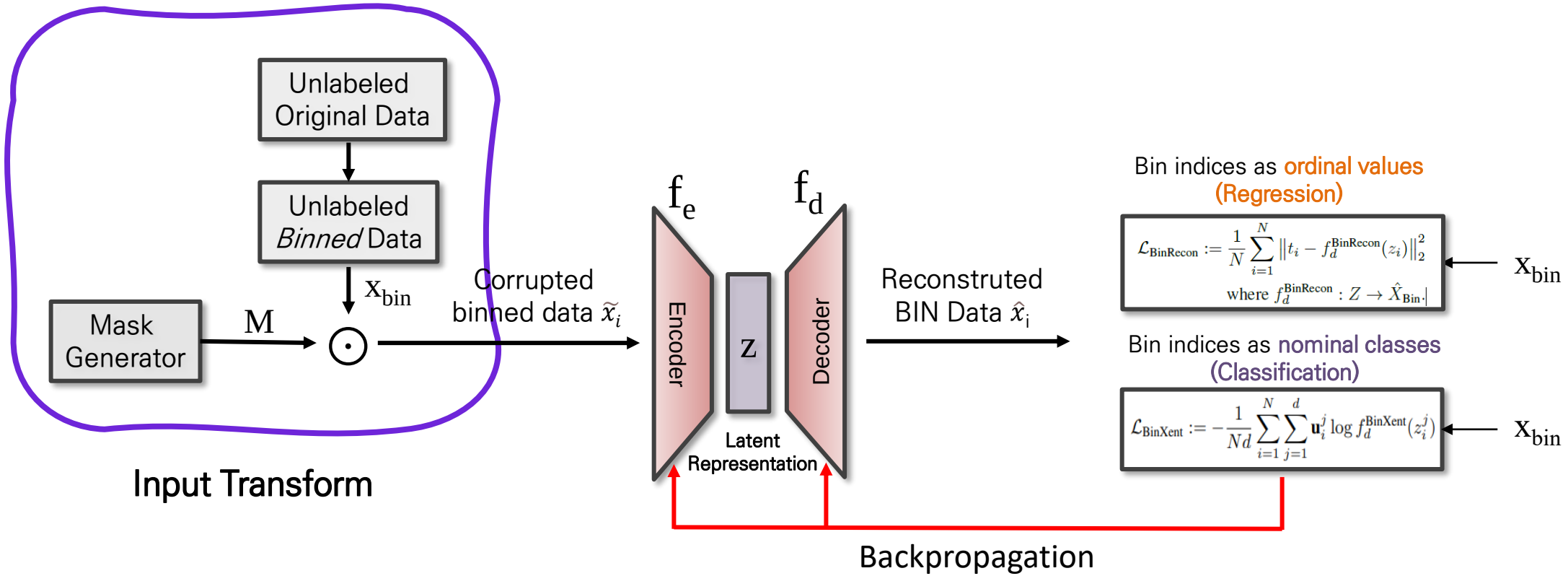
Binning as a Pretext Task

Binning as a Pretext Task for Tabular SSL



Binning as a Pretext Task

Binning as a Pretext Task for Tabular SSL



Binning as a Pretext Task

❖ Input Transform

1) Binning

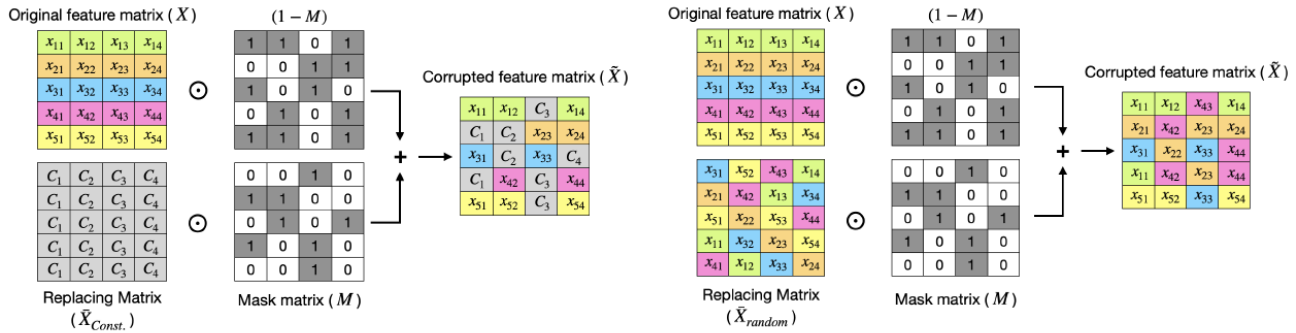
Equal Frequency Binning (Percentile) 기준 BIN 개수 (T, Parameter)로 입력값 대체

F1	F2	..	F4
0.07	0.87	..	0.00
0.77	0.17	..	0.30
0.58	0.11	..	0.33
0.10	0.25	..	0.95

T=2 →

F1	F2	..	F4
1	2	..	1
2	1	..	1
2	1	..	2
1	2	..	2

2) Mask

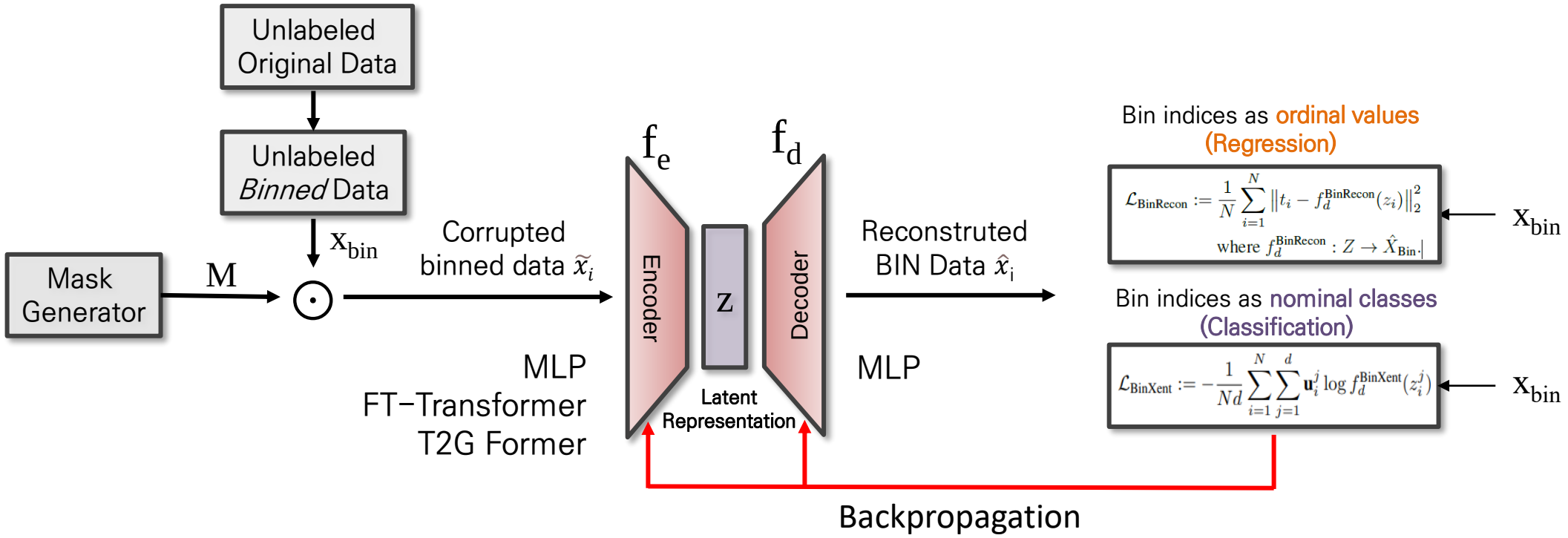


(a) Replacing value = Constant

(b) Replacing value = Random

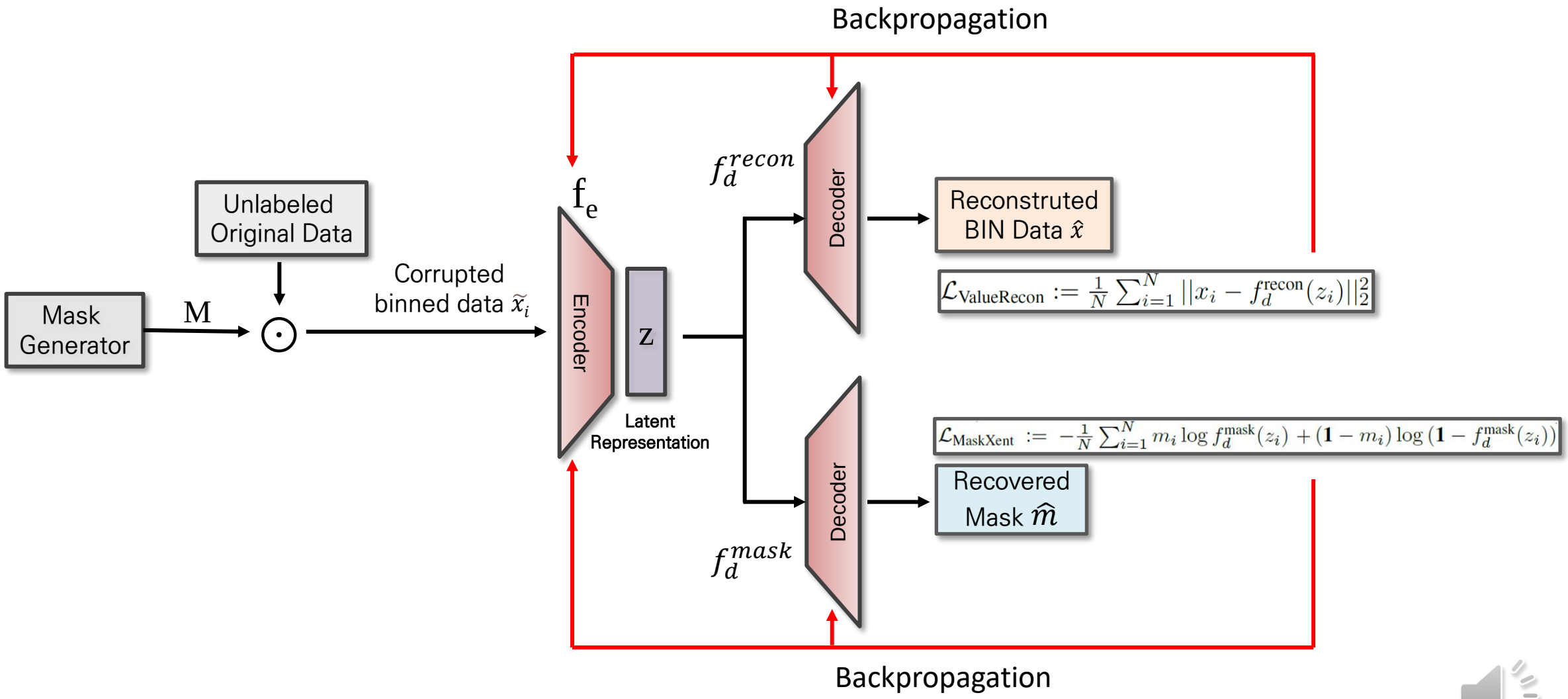
Binning as a Pretext Task

Binning as a Pretext Task for Tabular SSL



Binning as a Pretext Task

비교 방법론 (VIME, NIPS 2020)



Result

비교 방법론과 SSL Object 비교 실험

(b) Multiclass classification (Metric: Accuracy)

Masking	Replacing value	SSL Objective(s)	CO	OT	GE	VO	WQ	AL	HE	MNIST	p-MNIST	Average Rank
FALSE	-	ValueRecon	0.769	0.776	0.527	0.619	0.568	0.931	0.353	0.965	0.928	6.333
TRUE	Const.	MaskXent	0.784	0.777	0.518	0.545	0.547	0.909	0.341	0.793	0.554	9.333
TRUE	Const.	ValueRecon	0.783	0.791	0.557	0.622	0.586	0.931	0.354	0.966	0.925	4.111
TRUE	Const.	MaskXent+ValueRecon	0.750	0.774	0.519	0.610	0.571	0.931	0.360	0.941	0.907	7.444
TRUE	Random	MaskXent	0.763	0.791	0.555	0.549	0.544	0.925	0.336	0.945	0.817	8.000
TRUE	Random	ValueRecon	0.761	0.782	0.538	0.625	0.573	0.930	0.357	0.956	0.934	5.556
TRUE	Random	MaskXent+ValueRecon	0.769	0.779	0.521	0.564	0.519	0.925	0.353	0.945	0.906	8.333
FALSE	-	BinXent	0.742	0.781	0.517	0.600	0.565	0.903	0.354	0.956	0.908	8.333
FALSE	-	BinRecon	0.784	0.783	0.544	0.625	**0.592**	0.935	0.357	0.964	0.950	3.556
TRUE	Const.	BinRecon	0.812	0.792	0.559	0.647	0.581	0.943	0.359	0.974	0.964	2.222
TRUE	Random	BinRecon	**0.814**	**0.794**	**0.580**	**0.655**	0.574	**0.949**	**0.365**	**0.981**	**0.971**	**1.333**

(c) Regression (Metric: RMSE)

Masking	Replacing value	SSL Objective(s)	CA	HO	FI	MI	KI	CPU	DIA	EL	Average Rank
FALSE	-	ValueRecon	0.749	4.241	13900.720	0.784	0.163	3.876	1016.641	0.399	8.625
TRUE	Const.	MaskXent	0.709	4.548	13473.750	0.788	0.185	4.475	1259.744	0.396	8.875
TRUE	Const.	ValueRecon	0.693	4.086	13518.683	0.778	0.160	3.728	952.444	0.394	5.000
TRUE	Const.	MaskXent+ValueRecon	0.700	4.157	13915.875	0.775	0.174	5.644	2797.034	0.398	8.750
TRUE	Random	MaskXent	0.677	4.297	13826.641	0.782	0.176	3.951	1358.135	0.388	7.875
TRUE	Random	ValueRecon	0.713	4.127	13668.988	0.777	0.162	3.760	986.306	0.396	6.500
TRUE	Random	MaskXent+ValueRecon	0.701	4.136	14107.645	0.780	0.166	4.506	1917.875	0.397	8.750
FALSE	-	BinXent	0.690	4.116	**13038.762**	0.776	0.170	3.717	1207.923	0.383	4.875
FALSE	-	BinRecon	0.622	3.766	13453.309	**0.767**	**0.158**	3.208	897.645	0.370	2.250
TRUE	Const.	BinRecon	0.634	3.765	13208.133	0.773	**0.158**	**3.156**	957.801	0.371	2.375
TRUE	Random	BinRecon	**0.619**	**3.703**	13075.474	0.773	0.160	3.183	**870.283**	**0.368**	**1.625**

Result

Tree 모델 및 Deep Learning Method 와의 비교

Training network and method	Binary classification				Multiclass classification						Regression		
	HI ↑	PH ↑	OS ↑	PO ↑	CO ↑	GE ↑	VO ↑	AL ↑	HE ↑	MNIST ↑	CA ↓	HO ↓	FI ↓
Tree-based machine learning algorithms													
XGBoost	0.726	0.721	0.840	0.711	0.969	0.683	0.699	0.924	0.348	0.977	0.434	3.152	10372.778
CatBoost	0.727	0.728	0.833	0.897	0.967	0.692	0.711	0.948	0.386	0.979	0.430	3.093	10636.322
Deep learning methods													
MLP	0.714	0.724	0.896	0.901	0.968	0.659	0.692	0.960	0.378	0.983	0.513	3.146	10086.080
ResNet	0.688	0.728	0.885	0.795	0.729	0.484	0.550	0.220	0.229	0.826	0.706	4.004	10226.508
TabNet (Arik & Pfister, 2021; Gorishniy et al., 2021)	0.719	-	-	-	0.957	0.587	0.568	0.954	0.378	0.968	0.510	-	-
NODE (Popov et al., 2019; Gorishniy et al., 2021)	0.726	-	-	-	0.958	-	-	0.918	0.359	-	0.464	-	-
DCN V2 (Wang et al., 2021; Gorishniy et al., 2021)	0.723	-	-	-	0.965	-	-	0.955	0.385	-	0.484	-	-
SCARF (Bahri et al., 2021)	0.585	0.710	0.878	0.838	0.654	0.325	0.289	0.731	0.050	0.801	1.084	5.595	13632.255
SAINT (Somepalli et al., 2021)	0.713	0.728	0.886	0.877	0.943	0.691	0.713	0.932	0.378	0.981	0.581	6.186	19366.582
FT-Transformer (Gorishniy et al., 2021)	0.729	0.724	0.882	0.890	0.970	0.664	0.705	0.960	**0.391**	0.966	0.487	3.319	10206.127
PLR (MLP-Ensemble) (Gorishniy et al., 2022)	0.734	-	-	-	0.970	0.674	-	-	-	-	0.467	3.050	-
PLR (FT-T-Ensemble) (Gorishniy et al., 2022)	0.734	-	-	-	**0.972**	0.646	-	-	-	-	0.464	3.162	-
T2G-Former (Yan et al., 2023)	0.734	0.746	0.884	0.881	0.968	0.656	0.717	0.964	**0.391**	0.985	**0.455**	3.138	10750.850
SSL(MaskXent)+Fine-tuning	0.725	0.751	0.892	0.897	0.970	0.698	0.717	0.963	0.383	0.985	0.479	3.086	10204.559
SSL(ValueRecon)+Fine-tuning	0.719	0.731	0.894	0.899	0.969	0.690	0.712	0.963	0.381	0.984	0.478	3.119	10333.400
SSL(MaskXent+ValueRecon)+Fine-tuning	0.727	0.737	0.894	0.896	0.968	0.658	0.709	0.959	0.382	0.984	0.475	3.257	10708.780
Ours – SSL(BinRecon)+Fine-tuning	**0.737**	**0.764**	**0.897**	**0.904**	0.971	**0.720**	**0.728**	**0.966**	0.388	**0.986**	0.464	**2.989**	**9757.950**

Thank you