Statistical Machine Learning

7주차

담당: 15기 박지우

1. XgBoost

2. LightGBM

3. CatBoost

1. XgBoost

XgBoost

- XgBoost는 기존 Gradient Tree Boosting 알고리즘에 과적합 방지를 위한 기법이 추가된 지도 학습 알고리즘

정의

1) XgBoost는 Gradient Tree Boosting
2) XgBoost는 과적합 방지를 위한 기법이 추가된 알고리즘

XgBoost

$$F_0(x) = \arg\min_c \sum_{i=1}^{n} L(y_i, c)$$

$$g_i = \left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right]_{F(x)=F_{m-1}(x)}$$

$$h_i = \left[\frac{\partial^2 L(y_i, F(x_i))}{\partial F(x_i)^2}\right]_{F(x)=F_{m-1}(x)}$$

$$\phi_m = \arg\min_{\phi} \sum_{i=1}^{n} \frac{1}{2} h_i \left[-\frac{g_i}{h_i} - \phi(x_i)\right]^2 + \gamma T + \frac{1}{2}\lambda\|\phi\|^2$$

$$F_m(x) = F_{m-1}(x) + l \cdot \phi_m(x) \qquad F_M(x) = \sum_{m=0}^{M} F_m(x)$$

XgBoost

$$l = \sum_{i=1}^{n} L(y_i, F_{m-1}(x_i) + \phi(x_i)) + \gamma T + \frac{1}{2}\lambda\|\phi\|^2$$

$$\begin{aligned}\tilde{l} &= \sum_{i=1}^{n} \left[g_i \phi(x_i) + \frac{1}{2} h_i \phi(x_i)^2\right] + \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \\ &= \sum_{j=1}^{T} \left[\left(\sum_{x_i \in R_j} g_i\right) w_j + \frac{1}{2}\left(\sum_{x_i \in R_j} h_i + \lambda\right) w_j^2 + \gamma\right]\end{aligned}$$

XgBoost

Previous Tree Models - Basic exact greedy algorithm

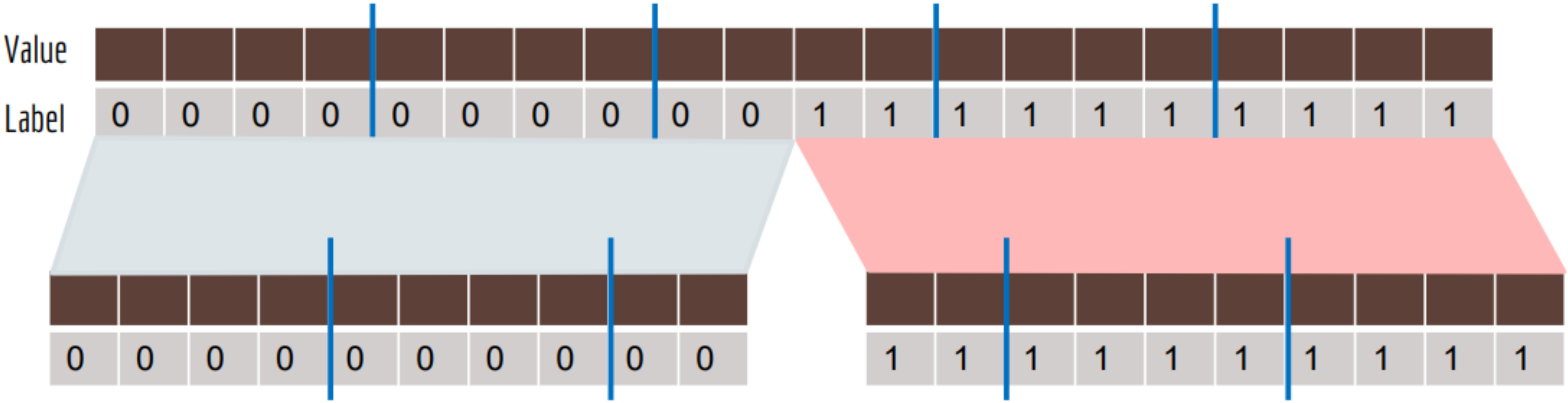
Value																				
Label	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1

Split Finding Algorithm

Value																				
Label	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1

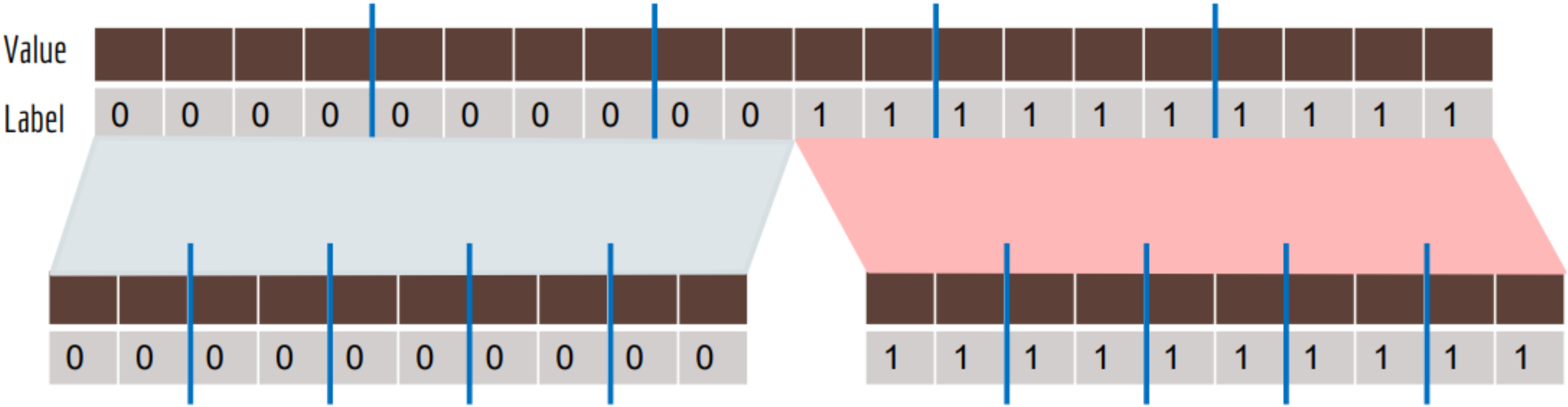
XgBoost

Global Variant

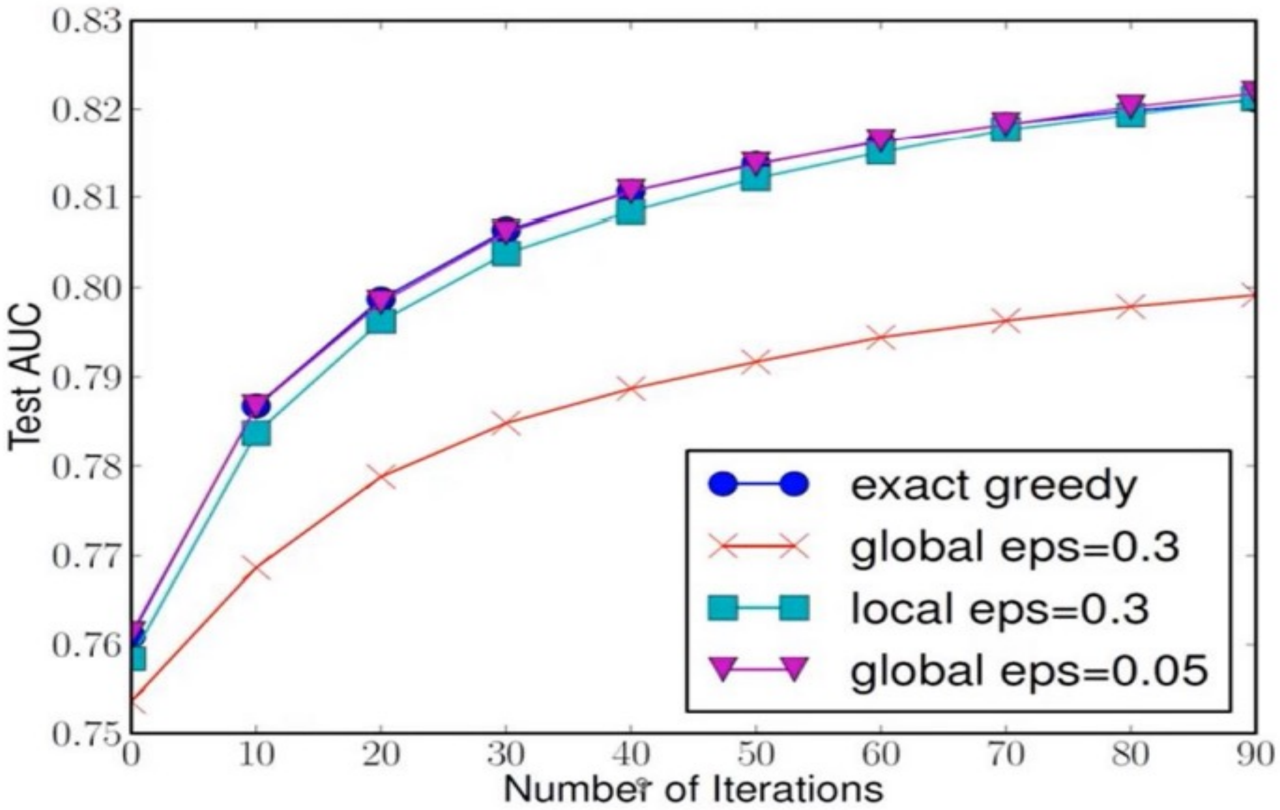


XgBoost

Local Variant

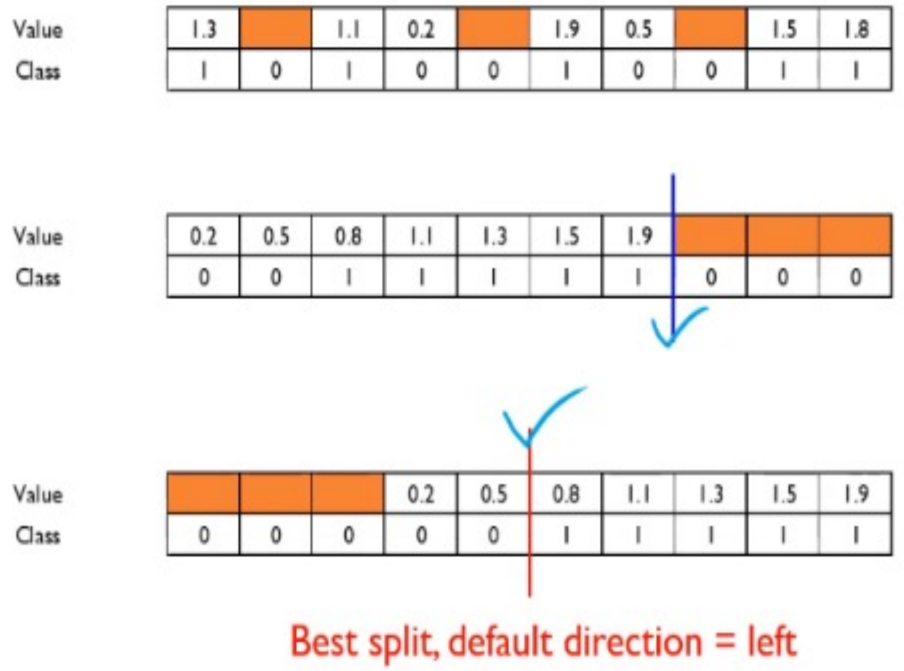


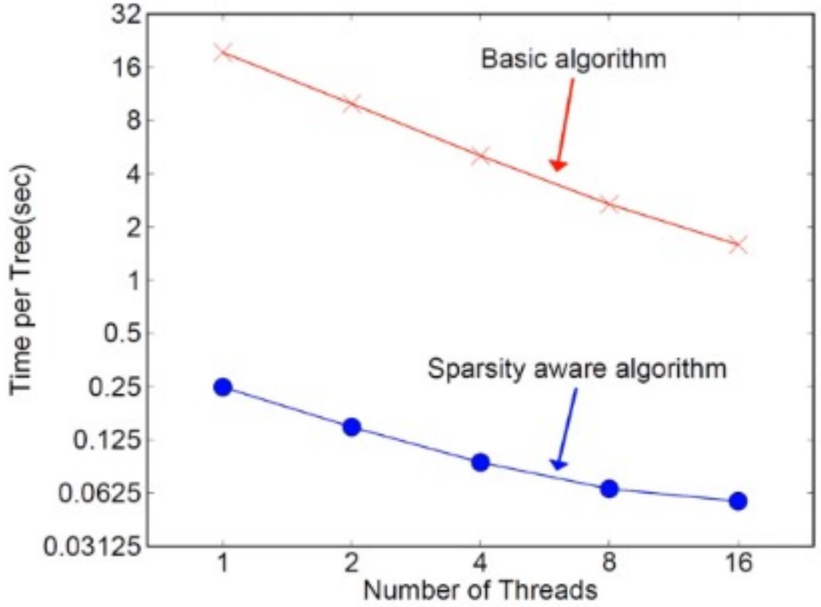
XgBoost



Number of Buckets: $1/\xi$

XgBoost


Value 1.3 1.1 0.2 1.9 0.5 1.5 1.8
Class 1 0 1 0 0 1 0 0 1 1
Value 0.2 0.5 0.8 1.1 1.3 1.5 1.9
Class 0 0 1 1 1 1 1 0 0 0
Value 0.2 0.5 0.8 1.1 1.3 1.5 1.9
Class 0 0 0 0 0 1 1 1 1 1
Best split, default direction = left


Time per Tree(sec)
32
16
8
4
2
1
0.5
0.25
0.125
0.0625
0.03125
1
2
4
8
16
Number of Threads
Basic algorithm
Sparsity aware algorithm

XgBoost

- XGBoost의 파라미터는 크게 일반, 부스터, 학습과정으로 나뉩니다

- 일반 파라미터
  booster : 어떤 부스터 구조를 쓸지 결정 - gbtree, gblinear, dart
  nthread : 몇 개의 쓰레드를 동시에 처리할지 - 디폴트 : 가능한 많이
  num_feature : feature 차원의 숫자를 정하는 옵션 - 디폴트 : 가능한 많이

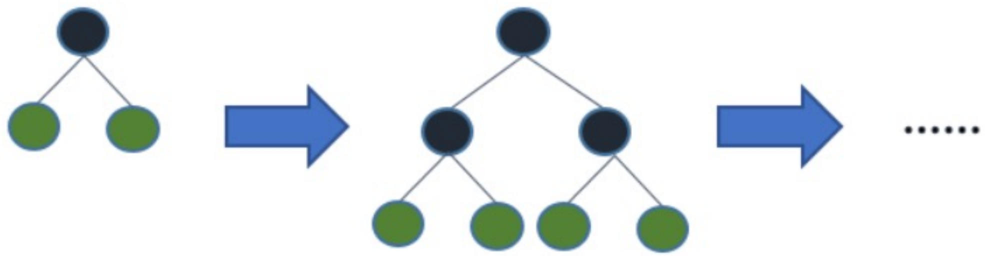
- 부스팅 파라미터
  eta : learning rate
  gamma : 트리 복잡도 파라미터. 커지면 트리 깊이가 줄어들어서 보수적인 모델이 된다. - 디폴트 : 0
  max_depth : 한 트리당 깊이 - 디폴트 : 6, 키울수록 과적합 위험 ↑
  lambda : L2 Regularization Form에 달리는 weights이다. 숫자가 클 수록 보수적인 모델이 된다.
  alpha : L1 Regularization Form weights. 숫자가 클수록 보수적인 모델이 된다.

- 학습 과정 파라미터
  object : 목적함수. reg-linear(linear-regression), binary-logistic(binary-logistic classification),
  count-poission(count data poison regression) 등 다양하다.
  eval_metric : 모델의 평가 함수를 조정하는 함수다. Rmse(root mean square error), log loss(log-likelihood),
  MAP(mean average precision) 등, 해당 데이터의 특성에 맞게 평가 함수를 조정한다.

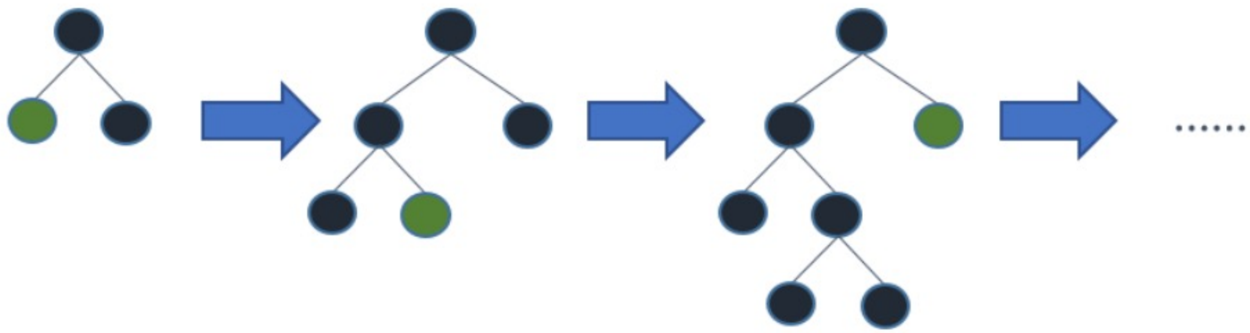
- 커맨드 라인 파라미터
  num_rounds : boosting 라운드를 결정. 적당히 큰 것이 좋고 epoch 옵션과 동일하다.

2. LightGBM

LightGBM



Level-wise tree growth



Leaf-wise tree growth

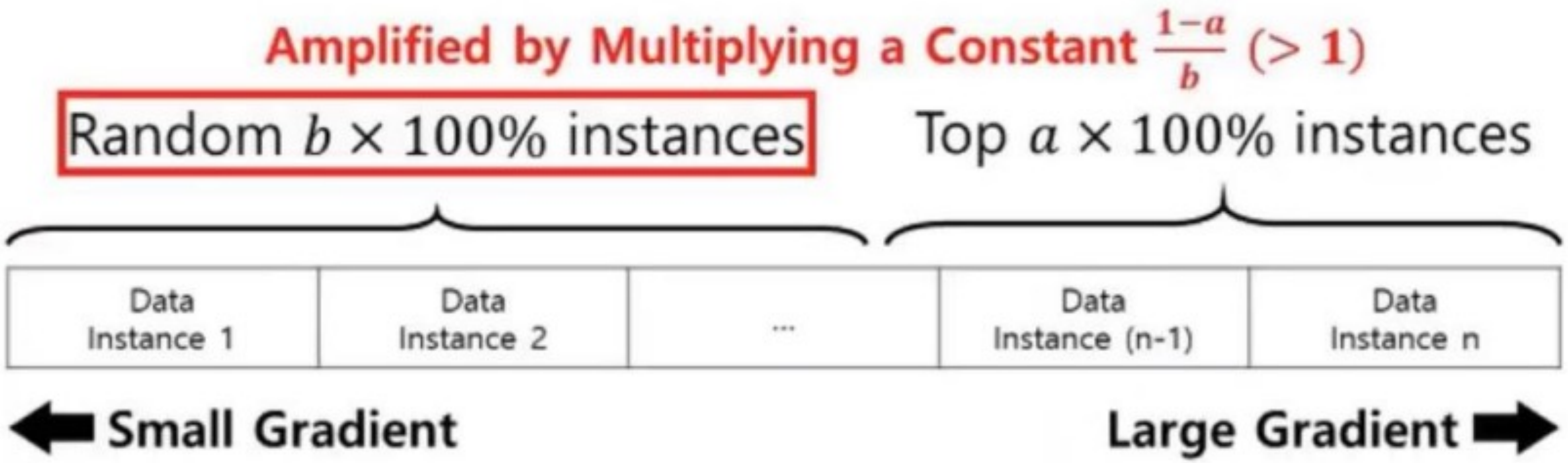
LightGBM

Gradient based One-Side Sampling (GOSS)

Exclusive Feature Bundling (EFB)

LightGBM

Gradient based One-Side Sampling (GOSS)



LightGBM

Exclusive Feature Bundling (EFB)

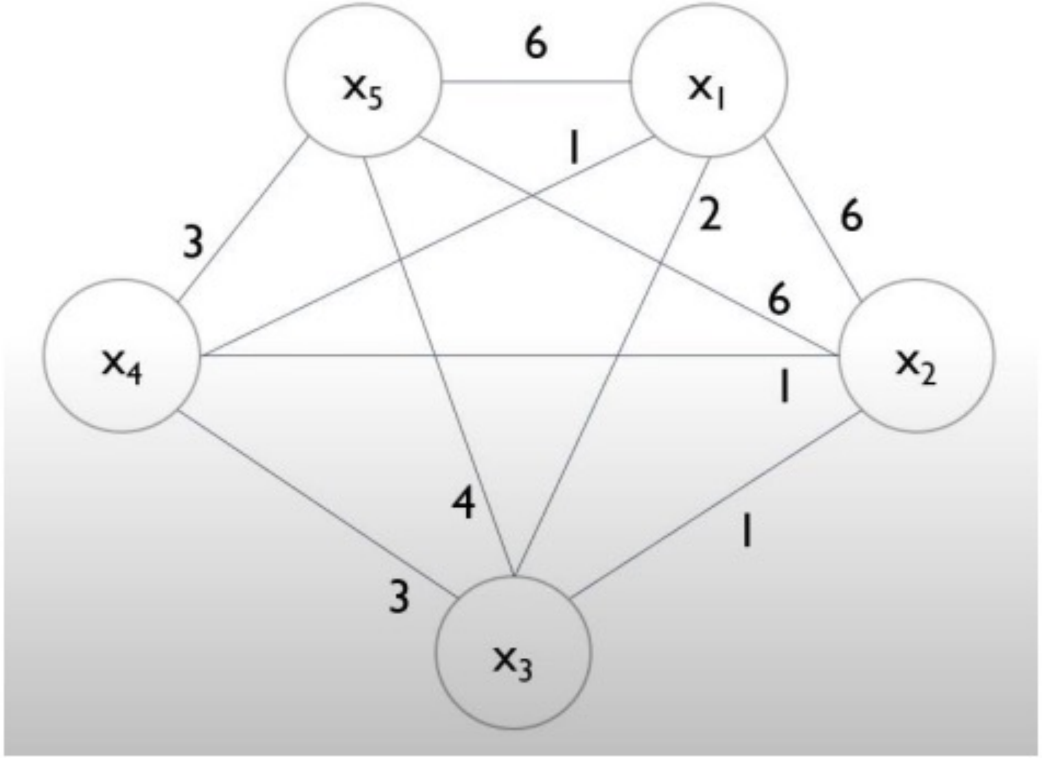
LightGBM

	x_1	x_2	x_3	x_4	x_5
I_1	1	1	0	0	1
I_2	0	0	1	1	1
I_3	1	2	0	0	2
I_4	0	0	2	3	1
I_5	2	1	0	0	3
I_6	3	3	0	0	1
I_7	0	0	3	0	2
I_8	1	2	3	4	3
I_9	1	0	1	0	0
I_{10}	2	3	0	0	2

	x_1	x_2	x_3	x_4	x_5
x_1	-	6	2	1	6
x_2	6	-	1	1	6
x_3	2	1	-	3	4
x_4	1	1	3	-	3
x_5	6	6	4	3	-

	x_5	x_1	x_2	x_3	x_4
d	19	15	14	10	8

LightGBM



LightGBM

	x_1	x_2	x_3	x_4	x_5
I_1	1	1	0	0	1
I_2	0	0	1	1	1
I_3	1	2	0	0	2
I_4	0	0	2	3	1
I_5	2	1	0	0	3
I_6	3	3	0	0	1
I_7	0	0	3	0	2
I_8	1	2	3	4	3
I_9	1	0	1	0	0
I_{10}	2	3	0	0	2

	x_5	x_1	x_4	x_2	x_3
I_1	1	1	0	1	0
I_2	1	0	1	0	1
I_3	2	1	0	2	0
I_4	1	0	3	0	2
I_5	3	2	0	1	0
I_6	1	3	0	3	0
I_7	2	0	0	0	3
I_8	3	1	4	2	3
I_9	0	1	0	0	1
I_{10}	2	2	0	3	0

LightGBM

	x_5	x_1	x_4	x_2	x_3
I_1	1	1	0	1	0
I_2	1	0	1	0	1
I_3	2	1	0	2	0
I_4	1	0	3	0	2
I_5	3	2	0	1	0
I_6	1	3	0	3	0
I_7	2	0	0	0	3
I_8	3	1	4	2	3
I_9	0	1	0	0	1
I_{10}	2	2	0	3	0

	x_5	x_{14}	x_{23}
I_1	1	1	1
I_2	1	4	4
I_3	2	1	2
I_4	1	6	5
I_5	3	2	1
I_6	1	3	3
I_7	2	0	6
I_8	3	1	2
I_9	0	1	4
I_{10}	2	2	3

3. CatBoost

CatBoost

Distinction

Target Leakage

→ Ordered TS(Target Statistics)

Prediction Shift

→ Ordered Boosting

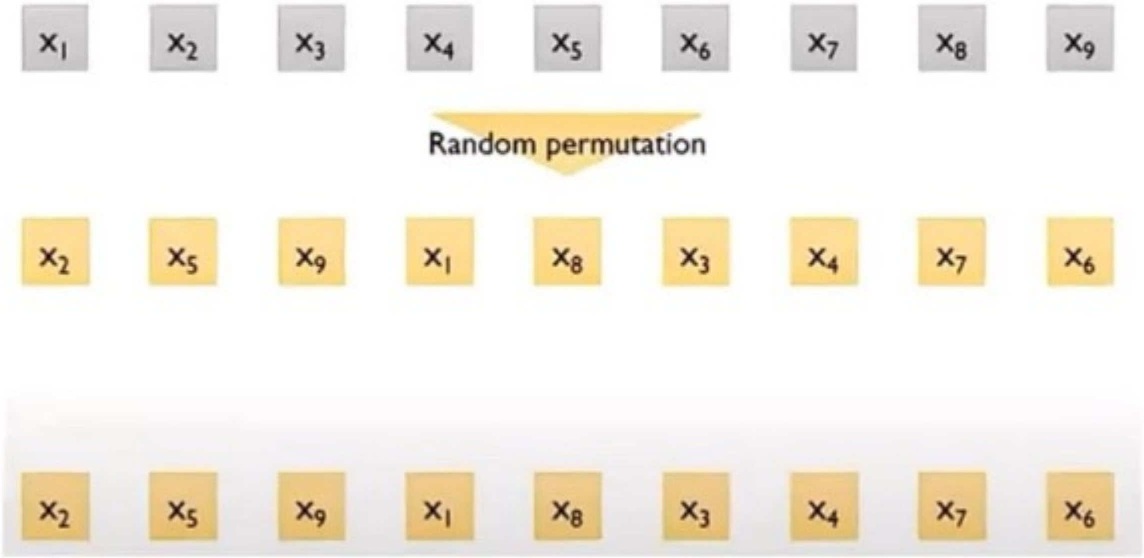
CatBoost

Ordered Target encoding

CatBoost

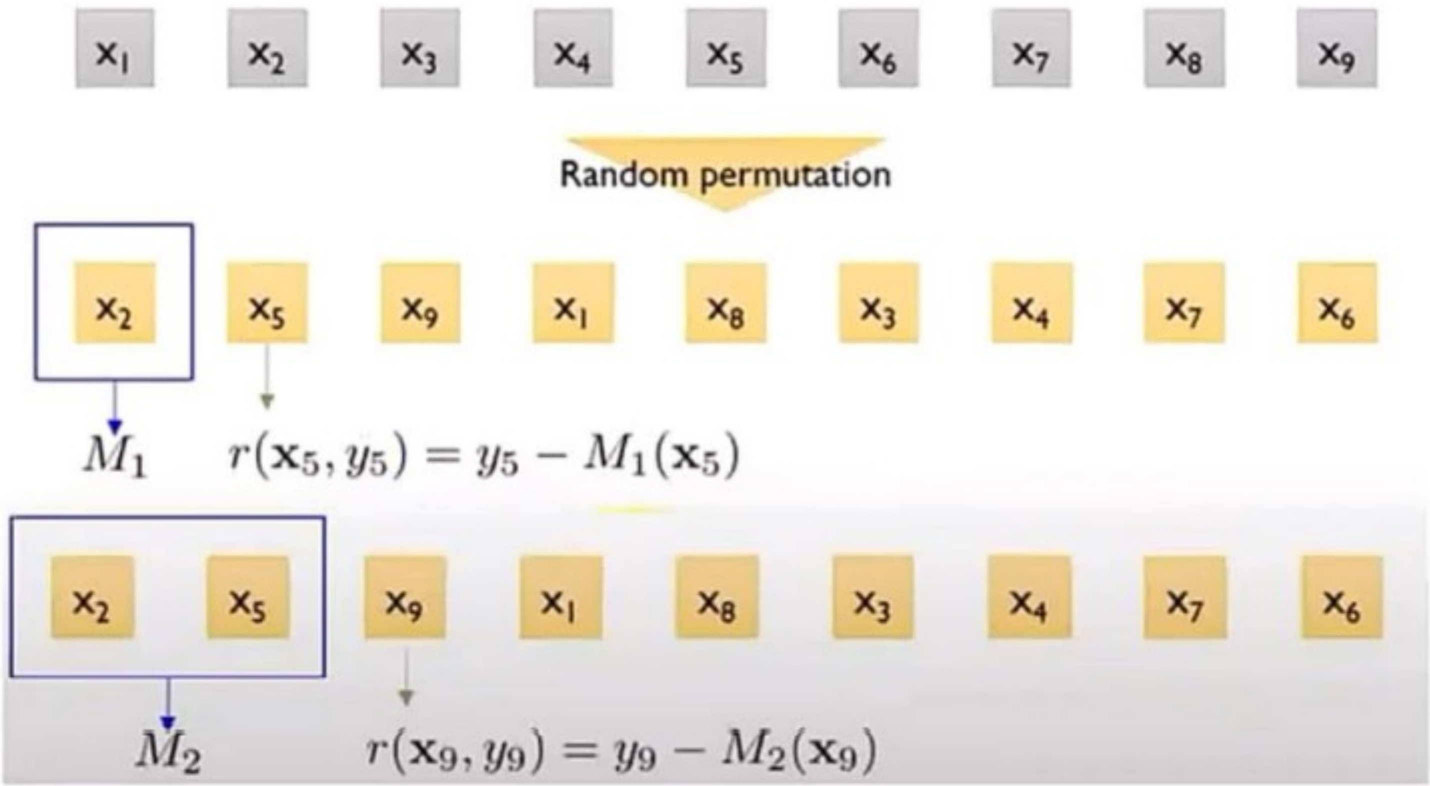
Ordered Boosting

- Ordered Boosting
  - ✓ A boosting algorithm not suffering from the prediction shift problem



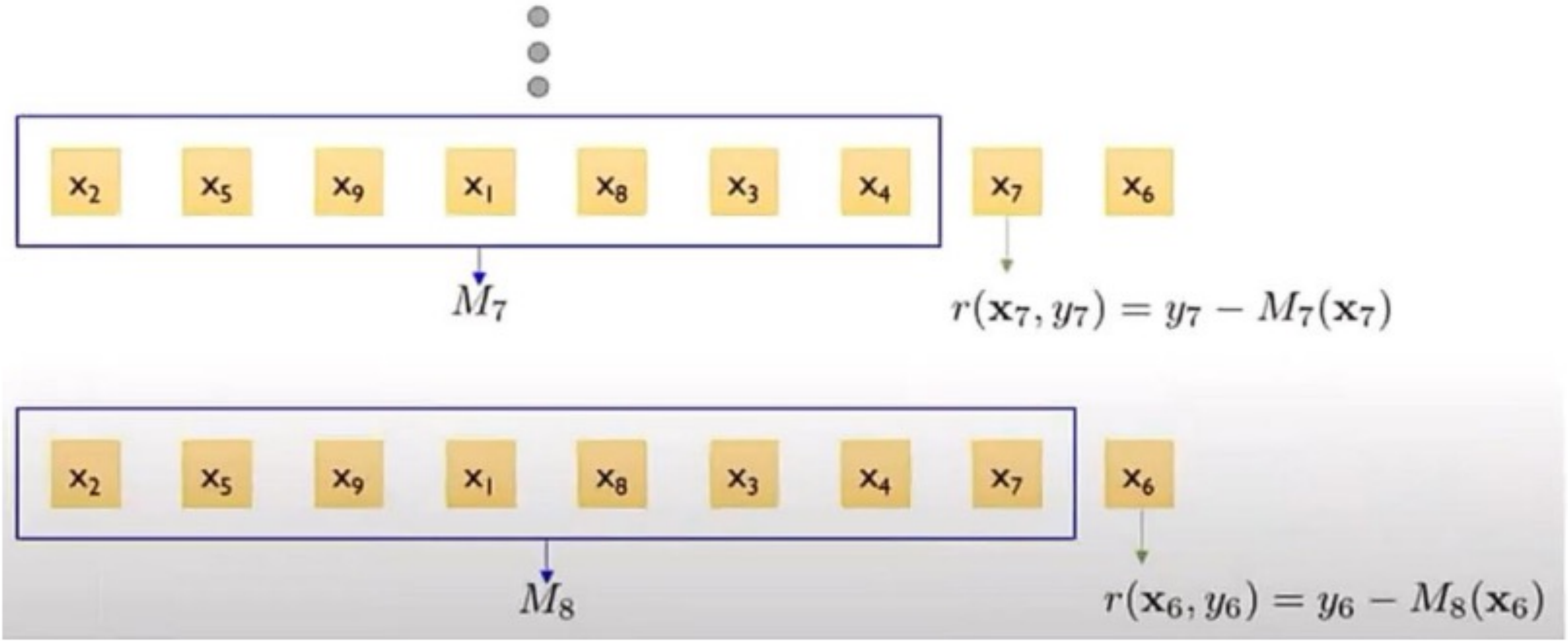
CatBoost

Ordered Boosting



CatBoost

Ordered Boosting



Coding Session

1. CatBoost

https://catboost.ai/en/docs/concepts/python-reference_catboostclassifier

2. LightGBM

https://lightgbm.readthedocs.io/en/latest/pythonapi/lightgbm.LGBMClassifier.html

3. XgBoost

https://xgboost.readthedocs.io/en/stable/python/python_api.html#xgboost.XGBRFClassifier
https://xgboost.readthedocs.io/en/stable/python/python_api.html#xgboost.XGBRFRegressor