What is XGBoost

XGBoost (eXtreme Gradient Boosting)

Out-of-core 계산 방식

메모리 초과를 방지하기 위해 데이터를 블록단위로 로드

손실함수 최소화 방식

2차 근사식을 바탕으로, 깊이를 늘려나감

규제와 결측값 처리

메모리 초과를 방지하기 위해 데이터를 블록단위로 로드

그리디 방식의 트리 확장

매 분기마다 최적의 확장 방향으로만 확장

손실함수 : 테일러 급수를 통한 2차 근사 방식

$$e^x = \frac{1}{0!} + \frac{1}{1!}x + \frac{1}{2!}x^2 + \frac{1}{3!}x^3 + \frac{1}{4!}x^4 \ldots\ldots$$

$$e^x = 1 + x + \frac{1}{2}x^2 + \frac{1}{6}x^3 + \frac{1}{24}x^4 \ldots\ldots$$

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!}$$

$$\tilde{\mathcal{L}}^{(t)} = \sum_{i=1}^{n} [g_i f_t(\_i) + \frac{1}{2} h_i f_t^2(\_i)] + \Omega(f_t),$$

그리디 방식 : 매 분기의 확장 마다, 최적의 방향으로만 진행

손실 함수 감소폭이 큰 곳으로 트리를 확장

$$\mathcal{L}_{split} = \frac{1}{2}\left[\frac{(\sum_{i\in I_L} g_i)^2}{\sum_{i\in I_L} h_i + \lambda} + \frac{(\sum_{i\in I_R} g_i)^2}{\sum_{i\in I_R} h_i + \lambda} - \frac{(\sum_{i\in I} g_i)^2}{\sum_{i\in I} h_i + \lambda}\right] - \gamma$$

자체적인 규제로 과적합 방지

결측값 처리 또한 자동으로 이루어짐