



**IT STEP  
UNIVERSITY**

# BOOSTING METHODS, BAGGING

АНАСТАСІЯ ДЕЙНЕКО

К.Т.Н., ДОЦЕНТ

ANASTASIYA.DEINEKO@GMAIL.COM

Ансамблі алгоритмів:

Ідея «**один алгоритм добре, а сто краще**»

**Найпопулярніші ансамблеві методи для знаходження аномалій:**

- *Feature Bagging* – для кожного алгоритму беруть випадковий ознаковий підпростір;
- *Rotated Bagging* – у вибраному випадковому ознаковому підпросторі роблять випадковий поворот.

**Бегінг** – проста техніка, в якій будуються незалежні моделі, а потім комбінуються, використовуючи деяку модель усереднення (наприклад, зважене середнє, голосування більшості або нормальне середнє).

**Ансамбль** – це набір моделей, які дають відповідь (наприклад, середнє по всіх). Причина чому ми використовуємо ансамблі – кілька моделей, які намагаються отримати ту саму змінну, дадуть точніший результат, ніж одна модель.

**Бустинг** - це техніка побудови ансамблів, в якій моделі побудовані не незалежно, а послідовно!

**Ця техніка використовує ідею про те, що наступна модель навчатиметься на помилках попередньої.**

## Алгоритм градієнтного бустингу

**Градієнтний бустинг** - це техніка машинного навчання для вирішення завдань класифікації та регресії, яка будує модель передбачення у формі ансамблю слабких моделей, зазвичай дерев рішень.

Використовується середньоквадратична помилка (MSE):

$$Loss = MSE = \sum (y_i - y_i^p)^2$$

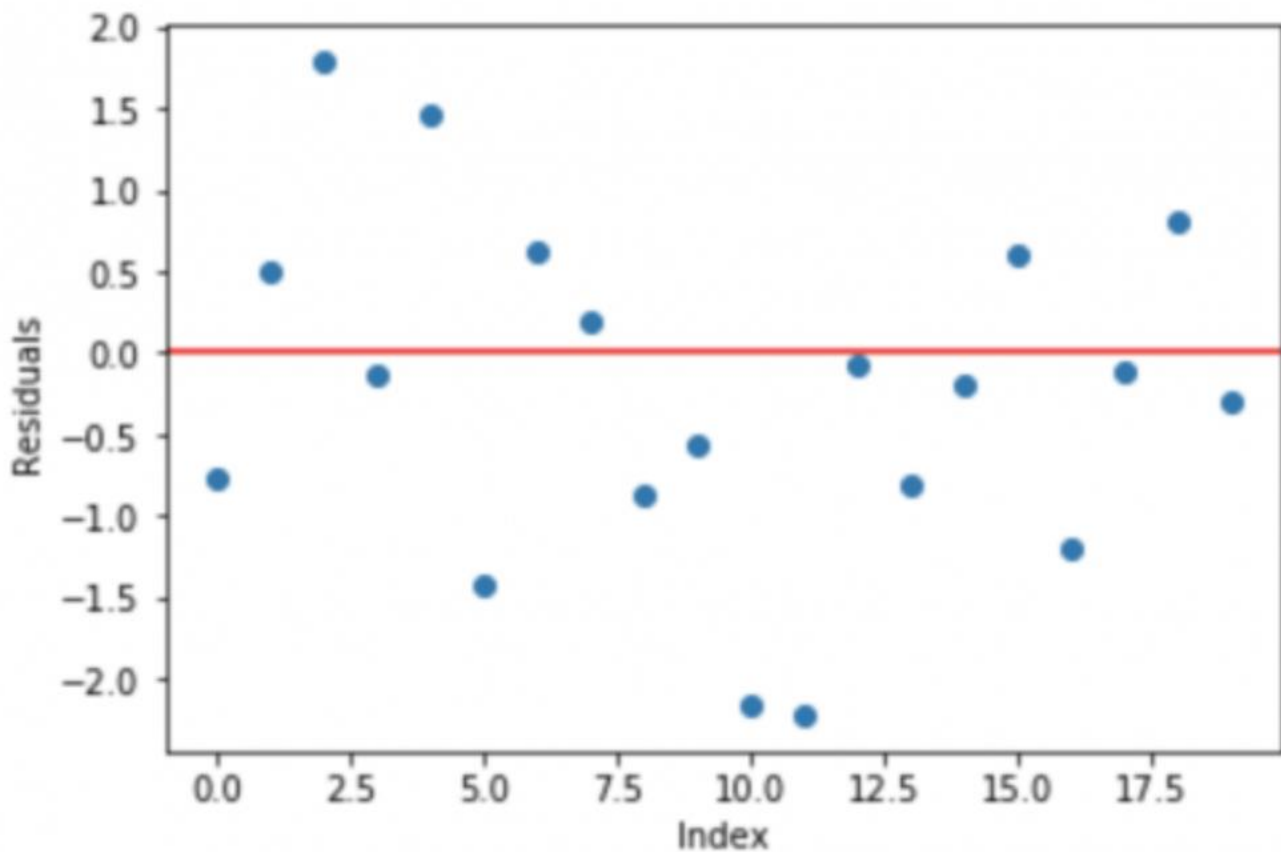
where,  $y_i$  = ith target value,  $y_i^p$  = ith prediction,  $L(y_i, y_i^p)$  is Loss function

## Метод градієнтного спуску

$$y_i^p = y_i^p + \alpha * \delta \sum (y_i - y_i^p)^2 / \delta y_i^p$$

$$\text{which becomes, } y_i^p = y_i^p - \alpha * 2 * \sum (y_i - y_i^p)$$

where,  $\alpha$  is learning rate and  $\sum (y_i - y_i^p)$  is sum of residuals



Нормальний розподіл вибірки відхилень із нульовим середнім

### Основні етапи:

- Спочатку будуюмо прості моделі та аналізуємо помилки;
- Визначаємо точки, які не вписуються у просту модель;
- Додаємо моделі, що обробляють складні випадки, які були виявлені на початковій моделі;
- Збираємо всі побудовані моделі, визначаючи вагу кожної.



## BOOSTING



## BAGGING



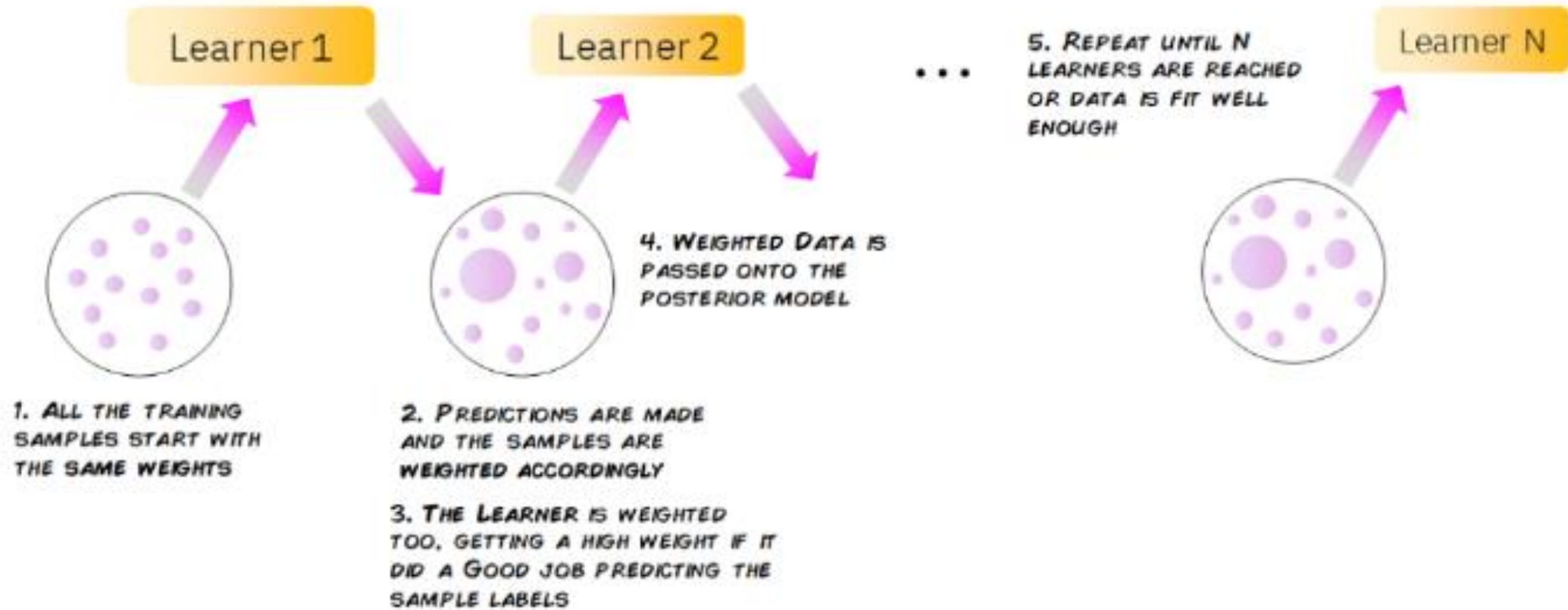
Bagging vs Boosting

# ОСНОВНІ КРОКИ РОБОТИ BOOSTING

## Крок 1: Ініціалізація ваг для кожного прикладу

- **Вхідні дані:** Ви маєте набір даних з  $n$  зразків (прикладів).
- **Вага:** Спочатку всім прикладам присвоюється рівна вага. Якщо кількість прикладів дорівнює  $n$ , то початкова вага кожного прикладу буде  $\frac{1}{n}$ .
- Мета полягає в тому, щоб навчити першу модель, де кожен приклад має однакову важливість.

## TRAINING BOOSTING MODELS



Training boosting models

## Крок 2: Навчання першої моделі (слабкого класифікатора)

- **Слабкий класифікатор:** На кожній ітерації Boosting навчається "слабкий" класифікатор, наприклад, просте дерево рішень (decision stump).
- Класифікатор навчається з урахуванням початкових ваг для всіх зразків.

## Крок 3: Оцінка точності та помилок

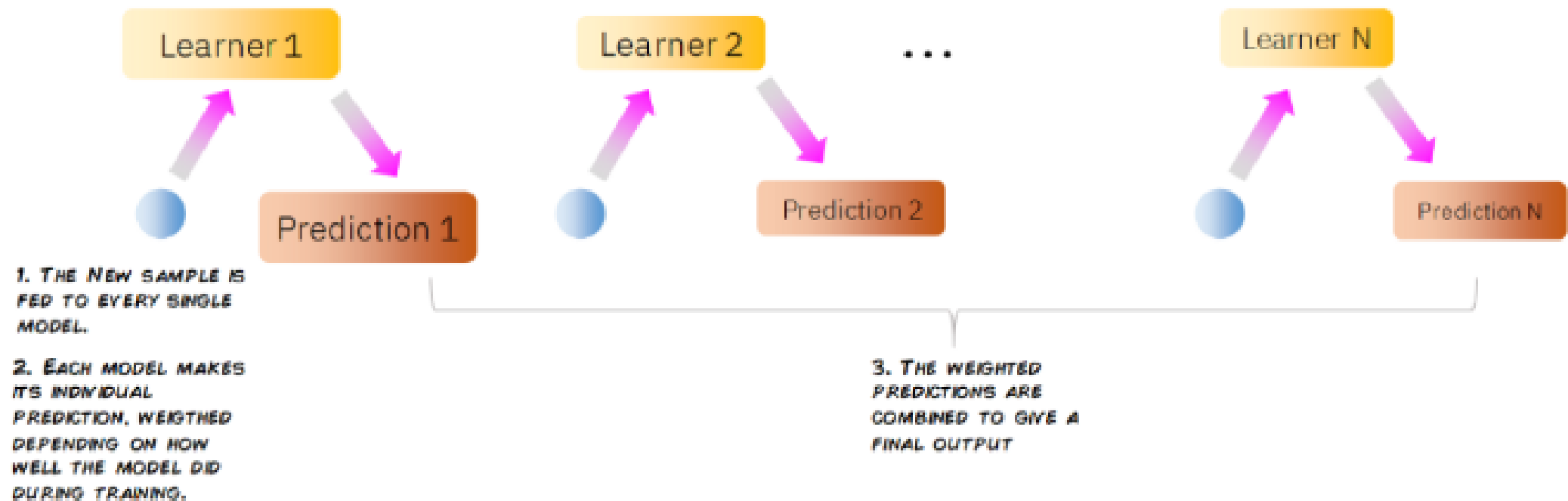
- **Оцінка результатів:** Модель класифікує всі приклади, і на основі її результатів обчислюється помилка. Помилка визначається як сума ваг всіх прикладів, які були неправильно класифіковані.

$$error = \sum_{i: y_i \neq \hat{y}_i} w_i$$

де  $y_i$  — це правильні мітки, а  $\hat{y}_i$  — передбачені моделі мітки.



## PREDICTING WITH BOOSTING MODELS



Predicting with boosting models

#### Крок 4: Обчислення ваги моделі

- **Вага слабкого класифікатора:** Кожен слабкий класифікатор отримує свою вагу в залежності від його точності (що краще він працює, то більшу вагу він отримає). Якщо модель має низький рівень помилок, їй присвоюється більша вага:

$$\alpha = \frac{1}{2} \ln \left( \frac{1 - error}{error} \right)$$

- Ця вага визначає внесок цієї моделі у фінальний ансамбль.

#### Крок 5: Оновлення ваг прикладів

- **Оновлення ваги прикладів:** Прикладам, які були неправильно класифіковані, присвоюється більша вага. Це дозволяє наступній моделі більше уваги приділити складним випадкам.

Формула для оновлення ваг виглядає так:

$$w_i \leftarrow w_i \cdot \exp(\alpha \cdot I(y_i \neq \hat{y}_i))$$

Де  $I(y_i \neq \hat{y}_i)$  — індикатор помилки (дорівнює 1, якщо модель зробила помилку, і 0 — якщо ні).

- Після цього ваги нормалізуються, щоб сума всіх ваг знову становила 1.

## Крок 6: Навчання наступної моделі

- **Повторення циклу:** Навчається наступний слабкий класифікатор, але тепер він зосереджений на прикладах, які були важкими для попереднього класифікатора, оскільки їхня вага зросла.

## Крок 7: Підсумковий прогноз

- **Ансамбль моделей:** Після кількох ітерацій створюється набір слабких моделей. Фінальний прогноз для кожного прикладу — це зважене голосування всіх слабких класифікаторів. Тобто кожен класифікатор вносить свій прогноз з урахуванням своєї ваги  $\alpha$ .

Для класифікації це зважене голосування, для регресії — зважене усереднення прогнозів.