

2-qism. 10 – dars

Model, Algorithm va Technique

Asosiy tushuncha	Model	Technique	Algorithm
Ta'rif	O'qitilgan tizim, bashorat qiluvchi mahsulot	Muammoni hal qilish usuli	Ma'lumotni o'qitish uchun matematik jarayon
E'tibor markazi	Muayyan ma'lumot va natija	Muammoni hal qilish uchun keng yondashuv	Bosqichma-bosqich jarayon
Misol	Linear Regression Model	Supervised Learning Boosting (texnika)	Gradient Descent Gradient Boosting (algoritm)
Qamrov doirasi	Maxsus ma'lumot bilan chegaralangan	Modelni yoki algoritmnini tanlash uchun kengroq	Ma'lumotlarni qayta ishlash uchun aniq qadamlar
Amalga oshirish	Fit va predict	Ma'lumot tayyorlash va modelni tanlash	Modelni o'qitish uchun matematik hisoblashlar

Xulosa

1. **Method (usul)** - Bu algoritm yoki modelni **amaliyotga tatbiq etish uchun aniq qadamlar** (funksiyalar).
2. **Model** — Method`ning bir qismi sifatida ishlaydigan **yakuniy matematik natija yoki tizim**, o'qitilgandan keyin ishlatiladi.
3. **Algorithm** - Modelni qurish va optimallashtirish uchun foydalaniladigan **hisoblash**
 - **Technique** — Ma'lumotni qanday qayta ishlash yoki qanday yondashuvni tanlashni belgilovchi **umumiy usul yoki yondashuv**.

Ular bir-biriga bog'liq bo'lib, **Technique** — yondashuvni, **Algorithm** — hisoblash jarayonini, **Model** — yakuniy mahsulotni, va **Method** — modelni amaliyotda ishlatish uchun zarur bo'lgan funksiyalarni anglatadi.

BOOTSTRAPING

Bootstrapping — ma'lumotlar to'plamidan **takroriy tanlash orqali** yangi namunalar yaratish usuli. Bu usul original ma'lumotdan statistik xulosalar olish yoki model ishonchliligini baholash uchun ishlatiladi. Foydalanish :

- **Original uzunlik = Bootstrap uzunlik:** Yangi namunaning o'lchami doimo original ma'lumotga teng.
- **Qayta olish mumkin:** Har bir qiymat bir nechta marta tanlanishi mumkin.
- **Ma'lumotlar bilan ishlash:** Bootstrapping orqali ma'lumot hajmini oshirmaymiz, lekin o'sha ma'lumotlarni statistik jihatdan qayta ishlash uchun turli namunalar hosil qilamiz.

Misol:

Original ma'lumotlar:

data = [5, 7, 9, 4, 6]

Bootstrapping jarayoni:

1. **Original uzunlik:** $\text{len}(\text{data}) = 5$.
2. **Takroriy tanlash orqali yangi namuna (BS):**
 - Namuna 1: [5, 9, 5, 7, 6] (qiymat "5" ikki marta tanlandi).
3. **Namuna uzunligi:** Har bir yangi namuna uchun $\text{len}(\text{BS}) = \text{len}(\text{data}) = 5$.

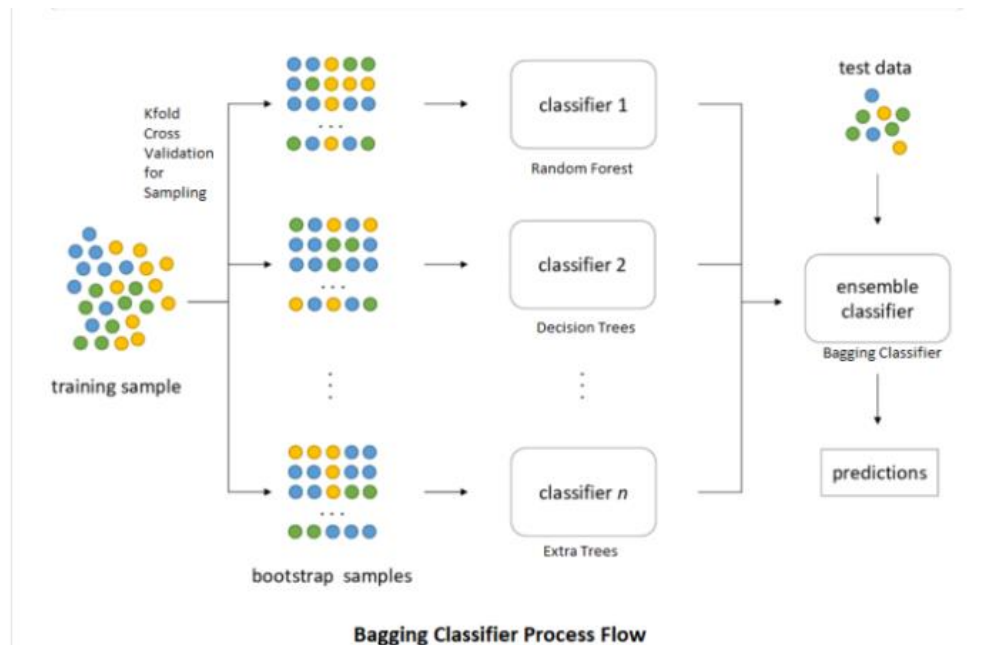
Bagging (Bootstrap Aggregating)

Bagging (Bootstrap Aggregating) — bu mashinani o'qitishda ishlatiladigan **ansambl usuli** bo'lib, asosiy maqsadi modelning aniqligini oshirish va overfittingni kamaytirishdir. Bagging asosiy **Bootstrap** usuliga asoslangan bo'lib, bir nechta modelni parallel ravishda o'qitadi va ularning natijalarini birlashtiradi (agregatsiya qiladi).

Bagging qanday ishlaydi?

1. **Bootstrap namunalar yaratish:**
 - Original ma'lumotlar to'plamidan bir nechta takroriy tanlangan namunalar (bootstrap samples) yaratiladi.
 - Har bir namunada qiymatlar qayta tanlanishi mumkin.
2. **Bir nechta modelni o'qitish:**
 - Har bir bootstrap namuna uchun alohida model (sub-model) o'qitiladi.
 - Bu modellarning turlari bir xil yoki turli bo'lishi mumkin (masalan, qaror daraxtlari).
3. **Agregatsiya (birlashtirish):**
 - Klassifikatsiyada: Har bir modelning bashoratlari ovoz berish orqali birlashtiriladi (majority voting).

- Regressionda: Modellarning o'rtacha qiymati olinadi.



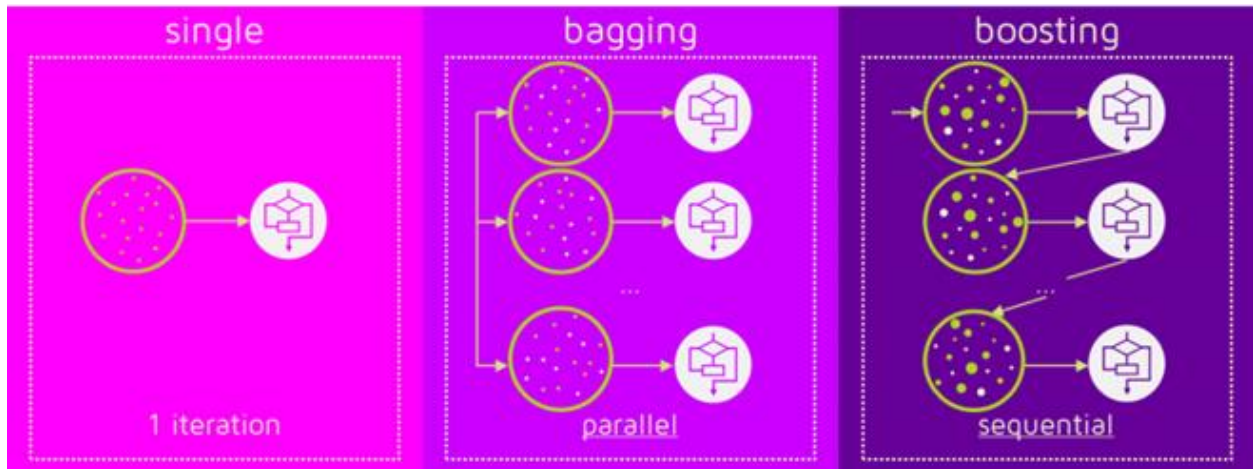
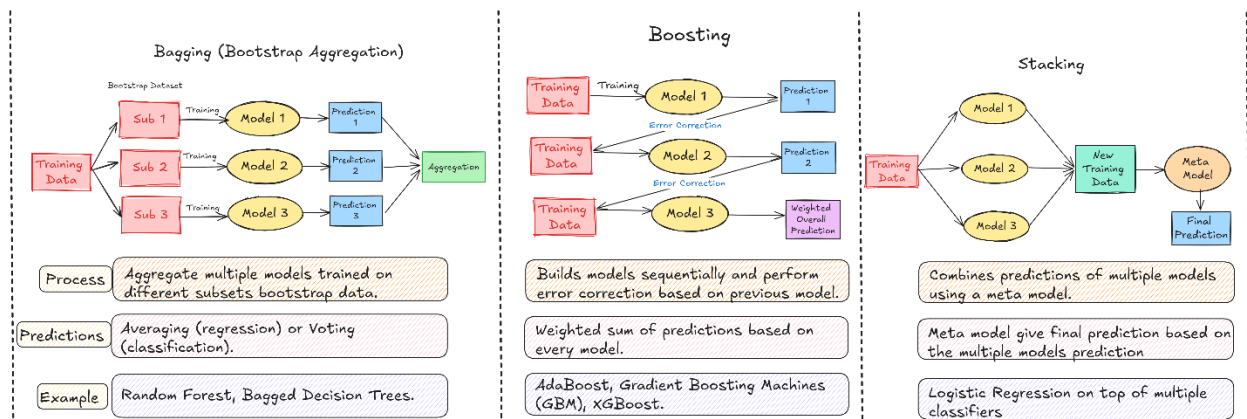
Bagging bilan Boosting o'rtasidagi farq

Xususiyat	Bagging	Boosting
Namuna olish	Bir vaqtning o'zida, takroriy tanlanadi	Ketma-ket tanlanadi, xatolarga qarab
Modellar o'qitilishi	Mustaqil (parallelda)	Bog'langan (ketma-ket)
Xatolarni kamaytirish	Variance'ni kamaytiradi	Bias va variance'ni kamaytiradi
Misol algoritmlar	Random Forest, BaggingClassifier	AdaBoost, Gradient Boosting, XGBoost

- Boosting—Biasni kamaytiradi
- Bagging--- xilma xillikni kamaytiradi

Boostingda namunalar qanday olinadi?

Boosting jarayonida **namunalar bir vaqtning o'zida emas, ketma-ket tanlanadi**. Har bir bosqichda yangi namuna **oldingi modelning natijalariga** qarab tanlanadi. Bu jarayonning asosiy maqsadi — oldingi model qayerda xato qilganini o'rganish va keyingi modelni bu xatolarni tuzatishga qaratishdir.



Ansambl usullari birliklarini tushunish bo'yicha xulosa

1. Bagging:

- Modellar mustaqil va natijalar birlashtiriladi.
- Har bir model teng vazn bilan hisobga olinadi.

2. Boosting:

- Modellar ketma-ket o'qitiladi.
- Og'irliklar xatolarni kamaytirishga qaratiladi.

3. Stacking:

- Asosiy modellarning natijalari yangi ma'lumot to'plamini yaratadi.
- Meta-model bu ma'lumotlarni birlashtiradi.

4. Blending:

- Stackingning soddalashtirilgan versiyasi, validatsiya ma'lumotlariga asoslangan.

Bagging :

$$\hat{f}(x) = \frac{1}{M} \sum_{i=1}^M f_i(x)$$

Natijalar o'rtacha yoki ovoz berish orqali birlashtiriladi.

Boosting :

$$\hat{f}(x) = \sum_{i=1}^M \alpha_i f_i(x)$$

$$w_{j+1} = w_j \cdot e^{\alpha \cdot \text{error}}$$

Ketma-ket xatolarni tuzatishga qaratilgan.

Stacking :

$$z_i = f_i(x), \quad i = 1, 2, \dots, M$$

$$\hat{f}(x) = g(z_1, z_2, \dots, z_M)$$

Meta-model asosiy natijalarni birlashtiradi.

Blending :

$$z_i = f_i(x_{\text{val}}), \quad i = 1, 2, \dots, M$$

$$\hat{f}(x) = g(z_1, z_2, \dots, z_M)$$

Validatsiya natijalaridan foydalaniladi.

1. Bagging

1. $\hat{f}(x)$: Yakuniy bashorat (regressiyada son, klassifikatsiyada sinf).
2. M : Asosiy modellarning soni.
3. $f_i(x)$: i -modelning x kirish qiymatiga nisbatan bashorati (regressiyada son, klassifikatsiyada sinf).

Tushuntirish:

- Baggingda barcha modellar mustaqil ishlaydi va ularning natijalari **o'rtacha qiymat** (regressiyada) yoki **ko'pchilik ovoz berishi** orqali birlashtiriladi.
- **O'rtacha qiymat**: Natija barcha bashoratlarning matematik o'rtachasidir.

2. Boosting

1. $\hat{f}(x)$: Yakuniy bashorat.
2. M : Modellar soni.
3. $f_i(x)$: i -modelning x kirish qiymatiga nisbatan bashorati.
4. α_i : Har bir modelning vazni. Modellar muvaffaqiyatiga qarab dinamik ravishda belgilanadi.
5. w_i : i -namuna uchun og'irlik (boshida barcha qiymatlar bir xil boshlanadi).
6. error : Modelning xato darajasi.

Tushuntirish:

- Boostingda modellar ketma-ket o'qitiladi. Oldingi model xato qilgan namunalar **katta og'irlik bilan** qayta ishlanadi.
- Yakuniy natija har bir modelning **vaznlangan yig'indisi** sifatida olinadi.

3. Stacking

1. z_i : i-modelning bashorati. Bu yangi ma'lumotlar to'plamini tashkil etadi.
2. $f_i(x)$: i-modelning x kirish qiymatiga nisbatan bashorati.
3. g : Meta-model, natijalarni birlashtirish uchun ishlatiladigan yuqori darajadagi model (masalan, Logistic Regression).
4. $\hat{f}(x)$: Yakuniy meta-modelning bashorati.

Tushuntirish:

- Stackingda har bir modelning bashorati yangi ma'lumotlar to'plamini yaratadi.
- **Meta-model** bu yangi ma'lumotlar to'plamida o'qitiladi va yakuniy natijani beradi.

4. Blending

1. z_i : Asosiy modelning validatsiya ma'lumotlarida natijasi.
2. $f_i(x_{\text{val}})$: i-modelning validatsiya uchun bashorati.
3. g : Meta-model (masalan, Linear Regression, Logistic Regression).
4. $\hat{f}(x)$: Yakuniy natija meta-modelning validatsiya natijalarida ishlashi asosida olinadi.

Tushuntirish:

- Blending stackingga o'xshash, lekin faqat validatsiya ma'lumotlaridan foydalaniladi.
- Bu usulda meta-model validatsiya natijalariga asoslanadi.

Gradient Boosting Algorithms

GBA nima : Classification / Regression

Asosiy vazifa : Model xatoliklarini kamaytirish (ketma- ketlik ravishda) va final model yaratish

Natija : Final modelda kuchli bo'ladi (kuchsiz bo'lgan modellarni birlashtirish orqali xosil qilinadi)

Ishlashi : Baseline Prediction

Difference Between Baseline and Advanced Models

Aspect	Baseline Models	Advanced Models
Definition	Simple methods providing a basic benchmark.	Sophisticated algorithms designed for accuracy and complexity.
Purpose	Serve as a starting point or benchmark.	Solve complex problems with higher precision.
Complexity	Simple (e.g., mean, mode, last value).	Complex (e.g., neural networks, gradient boosting).
Training	Often require no training (statistical rules).	Require extensive training and hyperparameter tuning.
Performance	Low accuracy, focuses on simplicity.	High accuracy with complex data patterns.
Use Case	Quick insights, comparison for improvement.	Final prediction or decision-making.
Speed	Very fast.	Slower due to computational demands.
Resources Needed	Minimal (low computational cost).	High (requires more memory, processing power).
Example	Predicting average for regression or mode for classification.	Random Forest, XGBoost, Deep Learning models.