Feature Scaling: Normalization

১) **Normalization** আসলে কী? **(Core Idea)**

Normalization হলো এমন একটি Feature Scaling টেকনিক, যেখানে
ডেটার আপেক্ষিক সম্পর্ক **(relative difference)** ঠিক রেখে
সংখ্যাগুলোকে একটি নির্দিষ্ট সীমার **(fixed range)** মধ্যে আনা হয়।

গুরুত্বপূর্ণ কথা:

- ডেটার অর্থ বদলায় না
- কোনটা বড়, কোনটা ছোট—এই ordering নষ্ট হয় না
- শুধু **scale / unit** বদলায়

কেন এটাকে **"Normalization"** বলা হয়**?**

কারণ আমরা ডেটাকে "স্বাভাবিক" বা "কমন ফ্রেম"-এর ভেতরে আনি—
যাতে সব ফিচার একই মাপে বিচারযোগ্য হয়।

২) **Normalization** কেন দরকার**? (Real Reason)**

ধরো তুমি তিনটা জিনিস মাপছো:

- ওজন → কেজি
- দূরত্ব → মিটার
- দাম → টাকা

এগুলোর **unit** আলাদা, scale আলাদা।
মডেল কিন্তু unit বোঝে না—সে শুধু সংখ্যা বোঝে।

Normalization করলে:

- unit-এর প্রভাব উঠে যায়
- মডেল শুধু magnitude / pattern দেখে
- distance / similarity / optimization fair হয়

৩) **Min–Max Scaling (**সবচেয়ে গুরুত্বপূর্ণ**)**

এটাই সবচেয়ে বেশি ব্যবহৃত **Normalization**।

Formula

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}}$$

কী করে এই ফর্মুলা?

1. সব ডেটা থেকে **minimum** বাদ দেয় → lowest point = 0
2. তারপর পুরো range দিয়ে ভাগ করে → highest point = 1

ফলে সব মান চলে আসে **[0, 1]** রেঞ্জে

Example

ধরো বয়স: 20 – 60

- 20 → (20−20)/(60−20) = 0
- 40 → (40−20)/40 = 0.5
- 60 → (60−20)/40 = 1

Ordering ঠিক আছে, কিন্তু scale ছোট।

৪) Geometric Intuition (খুব গুরুত্বপূর্ণ)

ধরো তোমার ডেটা 2D:

- X-axis = বয়স
- Y-axis = বেতন

Min–Max Scaling করলে:

- পুরো ডেটা ঢুকে যায় **Unit Square (1×1 box)** এর ভেতর
- 3D হলে → Unit Cube
- High-dimension হলে → Unit Hypercube

Distance-based algorithm এর জন্য এটা আদর্শ পরিবেশ।

৫) Mean Normalization (কম ব্যবহৃত, কিন্তু ধারণা গুরুত্বপূর্ণ)

এটা Min–Max আর Standardization-এর মাঝামাঝি।

Formula

$$x' = \frac{x - \text{Mean}}{x_{max} - x_{min}}$$

কী করে?

- **Mean centering** করে (ডেটা 0-এর আশেপাশে আসে)
- কিন্তু scale নিয়ন্ত্রণ করে Max–Min দিয়ে

Range

সাধারণত:

- **−1** থেকে **+1** এর মধ্যে

কেন **sklearn** এ নেই?

কারণ:

- এটি খুব standard practice না
- StandardScaler + MinMaxScaler দিয়েই প্রায় সব কাজ হয়

তাই sklearn এটাকে built-in দেয়নি।

৬) Max Absolute Scaling (Sparse Data specialist)

Formula

$$x' = \frac{x}{|x|_{max}}$$

কী করে?

- সব মানকে divide করে **maximum absolute value** দিয়ে
- ফলে range হয় **[-1, +1]**

কেন **Sparse Data-**তে ভালো**?**

Sparse data মানে:

- অনেক zero (0)
- যেমন: NLP (Bag of Words, TF-IDF)

এই স্কেলিং:

- zero কে zero-ই রাখে
- sparsity নষ্ট করে না

তাই text / high-dimensional sparse data-তে এটি ideal।

৭) **Robust Scaling (Outlier** থাকলে **lifesaver)**

সমস্যা**:**

Min–Max Scaling এ যদি একটাও বড় outlier থাকে:

- Max অনেক বড় হয়ে যায়
- বাকি সব ডেটা খুব compress হয়ে যায়

Robust Scaling কী করে**?**

Mean/Min/Max বাদ দিয়ে ব্যবহার করে:

- **Median**
- **IQR = Q3 − Q1**

Formula

$$x' = \frac{x - \text{Median}}{Q3 - Q1}$$

সুবিধা

- Outlier থাকলেও scale নষ্ট হয় না
- Distribution বেশি realistic থাকে

Outlier-heavy financial / real-world data-তে এটা best choice।

৮) **Standardization** বনাম **Normalization (Conceptual Difference)**

Standardization

- Mean = 0, Std = 1
- Range fixed না
- Gradient-based model-এ বেশি stable
- Outlier sensitive (mean/std)

Normalization (Min–Max)

- Range fixed: [0,1]
- Distribution shape একই থাকে
- Distance-based model-এ খুব ভালো
- Outlier sensitive (min/max)

তাই দুটো **rival** না, বরং **context-dependent tools**।

৯) কখন কোনটা ব্যবহার করবে**? (Decision Table)**

Normalization (Min–Max) ব্যবহার করো যখন**:**

- Feature-এর natural bounds জানা
- Image data (0–255 pixel)
- KNN / K-means
- Neural network যেখানে activation bounded

Standardization ব্যবহার করো যখন**:**

- Logistic / Linear Regression
- SVM

- PCA
- Gradient Descent involved

Robust Scaling:

- Outlier অনেক
- Real-world noisy data

MaxAbs Scaling:

- Sparse / text data
- Zero preserve করা দরকার

১০) **Practical Rules (**ভুল করলে বড় ক্ষতি হয়**)**

1. **Train–Test split** আগে
2. **fit** শুধু **train data** তে
3. **transform train + test** দুটোতেই
4. Test data তে কখনও fit না

না হলে data leakage → fake performance

১১) খুব গুরুত্বপূর্ণ ভুল ধারণা **(Exam/Interview Trap)**

“Scaling করলে data change হয়ে যায়”
ভুল — distribution একই থাকে, শুধু axis scale বদলায়

“Normalization সবসময় better”
ভুল — algorithm-dependent

“Tree model এ scaling দরকার”
সাধারণত দরকার নেই

১২) **Summary**

Normalization ডেটাকে ছোট করে না**,** বুদ্ধিমান করে।
Standardization ডেটাকে মাঝখানে এনে দ্রুত শেখাতে সাহায্য করে।