ii) Stochastic Gradient Descent (SGD) :-

SGD is a type of gd where the models parameters are updated using only 1 training example (1 row) at a time

How it works :-

i) shuffle the dataset ⟶ Pick one sample ⟶ Compute gradient using that 1 sample
↓
Repeat for the next sample ⟵ update parameters

⟶ Because every update uses a diff data point, parameters get updated very frequently.

why it is used :-

- ⟶ much faster than Batch GD
- ⟶ useful for large datasets
- ⟶ helps to escape local minima because of randomness
- ⟶ works well in online/streaming learning

Problems with SGD:-

* Learning Schedules

i) Very Noisy Updates
→ Each sample gives a different gradient
→ updates jump around instead of moving smoothly.

ii) Hard to reach the Exact Minimum
Because of randomness, SGD
→ may overshoot the minimum
→ Keeps oscillating around the minimum
→ Converges to a region around minimum, not Exactly to it.

iii) Highly Sensitive to Learning Rate
If learning rate is
→ too high → model diverges
→ too low → training becomes extremely slow.
→ constant → SGD may never settle

Solved with adam / RmsProp

iv) High Variance in Updates:
Each sample can give a completely different gradient
→ Path to minimum becomes zig-zag
→ Need more epochs to converge.

v) Not Good for Very Noisy Data
If dataset already has noise, SGD becomes even more unstable.

Time Comparison:- [If Epochs are Constant for Both GD & SGD]

assume: → Dataset has N samples
→ We run E epochs for both batch GD & SGD

Batch GD:- 1 epoch = using all N samples → 1 update

SGD :- 1 epoch = using all N samples one by 1 → N updates

	per epoch	Total updates
Batch SGD	→ 1 update	E updates
SGD	→ N update	N × E updates

"If the no. of epochs is constant, SGD performs N update per while Batch GD performs only 1. So SGD takes more time - the same number of epochs. However, SGD starts convergin faster because it updates frequently.