

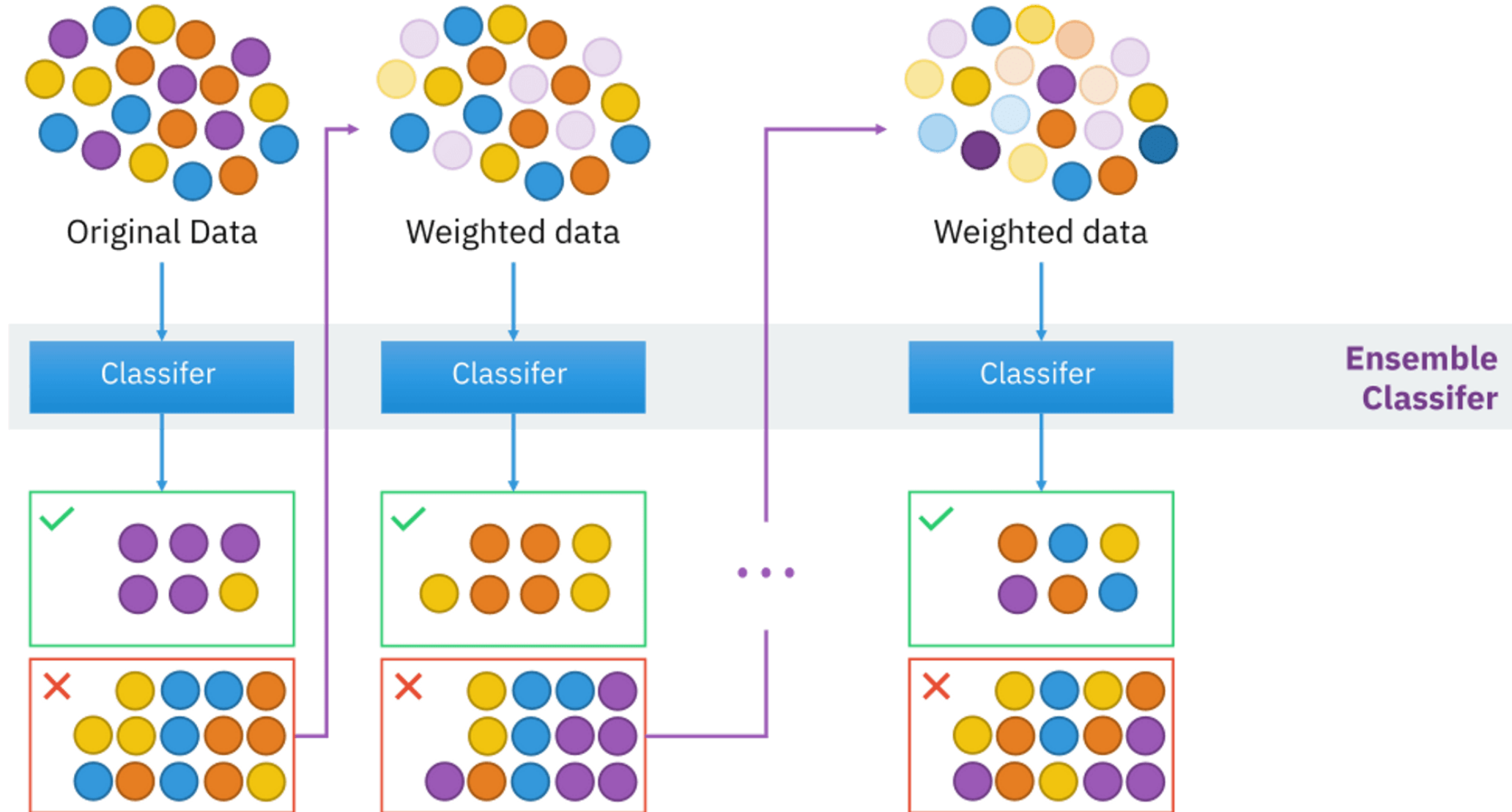
# AdaBoost Regression

## What is AdaBoost Regression?

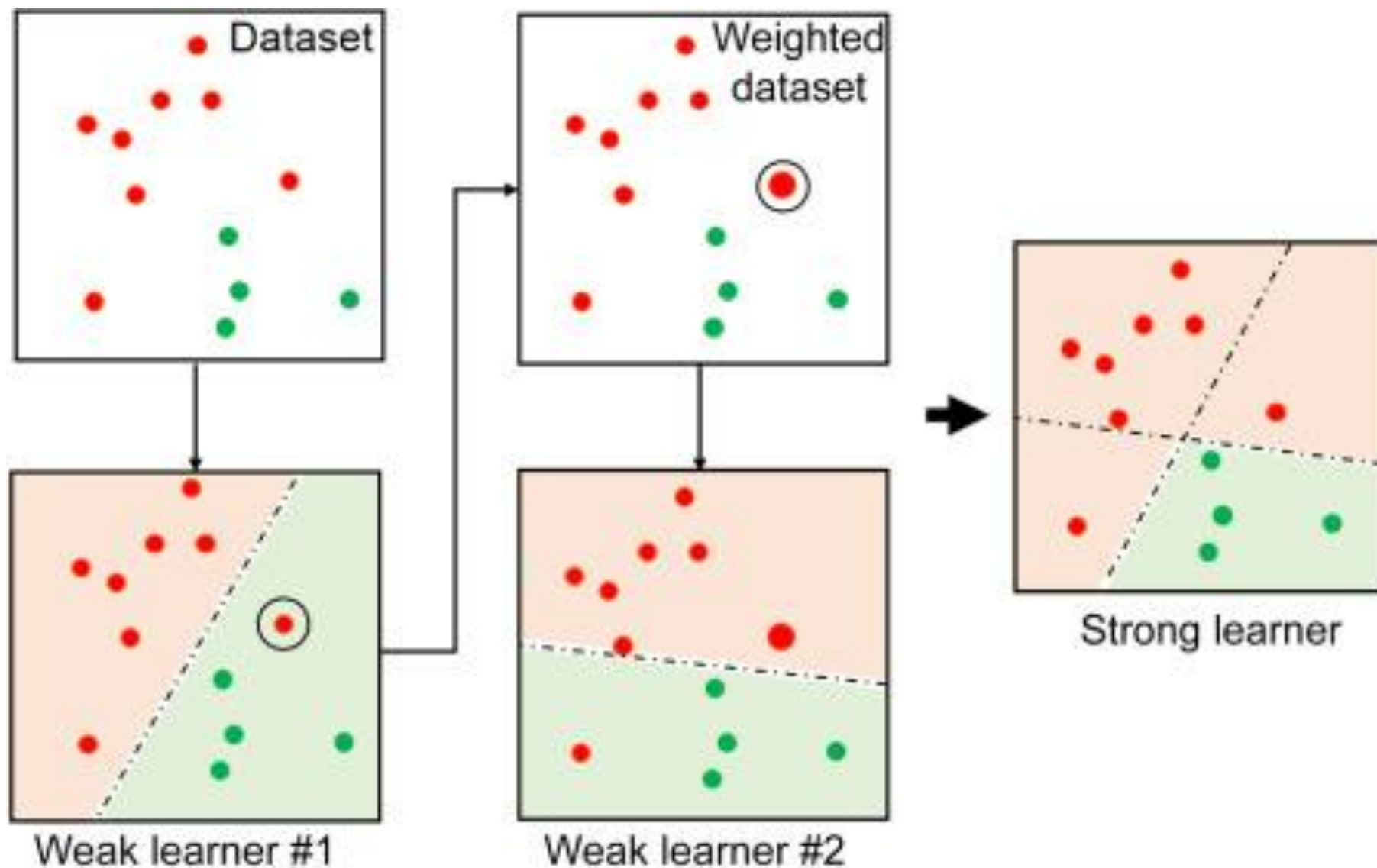
**AdaBoost (Adaptive Boosting)** is a **boosting algorithm** that combines several *weak learners* (usually simple Decision Trees) into a single *strong learner* for regression.

- A **weak learner** = a model that performs slightly better than random guessing.
- **Boosting** = build models one after another, where each new model focuses more on the **errors (residuals)** made by the previous models.
- 👉 In regression, instead of predicting categories, AdaBoost predicts **continuous values (numbers)**.

# AdaBoost Regression



# AdaBoost Regression



# How It Works (Step-by-Step)

1. **Start with data** (e.g., House Price, Insurance Charges, etc.).
2. Train a **first weak regressor** (like a small decision tree).
3. Measure the **error** (difference between predicted vs. actual).
4. Assign **higher weights** to the data points with larger errors (hard-to-predict cases).
5. Train the **next weak regressor**, focusing more on those errors.
6. Repeat the process for many weak regressors.
7. **Final Prediction = Weighted sum of all weak regressors.**

# Example: Predicting House Prices



## Example: Predicting House Prices

Imagine we want to predict **house price** using just **house size (sqft)**.

- Step 1:** First decision stump (tiny tree) predicts house price roughly.
- Step 2:** It fails on some houses (too high or too low).
- Step 3:** AdaBoost gives **more weight** to those houses.
- Step 4:** Next tree focuses on those errors.
- Step 5:** After many rounds, AdaBoost combines them → final smooth curve that predicts prices better.

## AdaBoost Regression



Example: Predicting House Prices



Imagine we want to predict house price using just house size (sqft).

Step 1

First decision stump (tiny tree) predicts house price roughly

Step 2

It fails on some houses (too high or too low)

Step 3

AdaBoost gives more weight to

Step 5

After many rounds

After many rounds, AdaBoost combines them →

Step 4

Next tree

# Python

```
from sklearn.ensemble import AdaBoostRegressor
regressor= AdaBoostRegressor(random_state=0, n_estimators=100)
regressor.fit(X, y)
```

Parameters:

***n\_estimators*** *int, default=50*

The maximum number of estimators at which boosting is terminated. In case of perfect fit, the learning procedure is stopped early. Values must be in the range *[1, inf)*.

***random\_state*** *int, RandomState instance or None, default=None*

Controls the random seed given at each *estimator* at each boosting iteration. Thus, it is only used when *estimator* exposes a *random\_state*. In addition, it controls the bootstrap of the weights used to train the *estimator* at each boosting iteration. Pass an int for reproducible output across multiple function calls. See [Glossary](#).