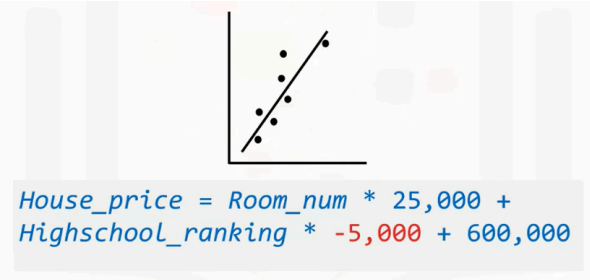
Course3_Module6

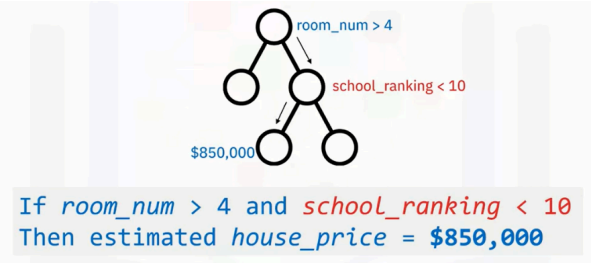
1) Model Interpretability

1.1. Tổng quan & phân loại mô hình

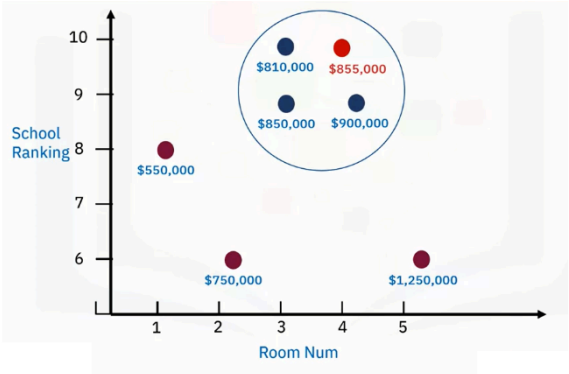
- **Tự diễn giải (self-interpretable):** tuyến tính (Linear), cây quyết định (Tree), KNN... vì cấu trúc/luật quyết định đơn giản, có thể đọc được.



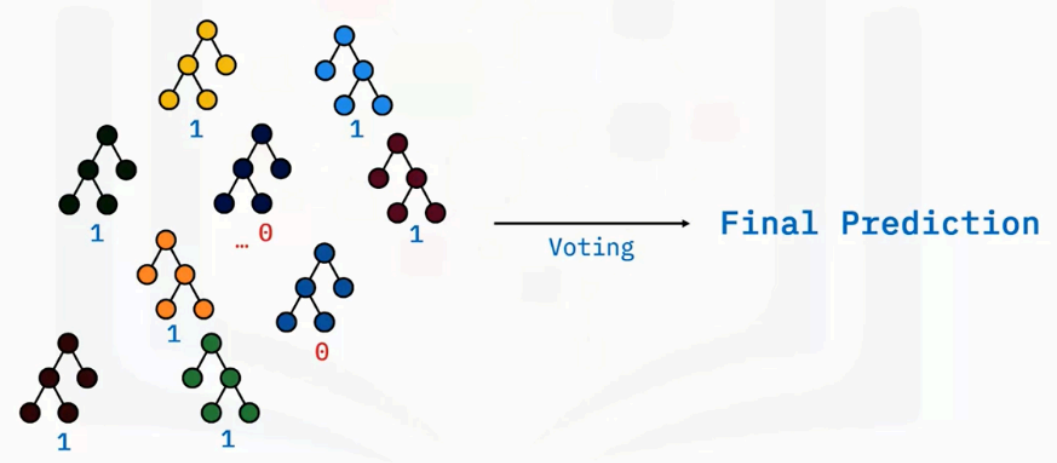
```
House_price = Room_num * 25,000 +
Highschool_ranking * -5,000 + 600,000
```




```
If room_num > 4 and school_ranking < 10
Then estimated house_price = $850,000
```



- **Khó diễn giải (black-box):** ensemble lớn (RF/GBM), SVM nhân kernel, deep nets... cần **post-hoc** để giải thích.



> Hai nhóm phương pháp: intrinsic (dùng trực tiếp trên mô hình tự diễn giải) và post-hoc/model-agnostic (giải thích mô hình bất kỳ sau khi huấn luyện).

1.2. Permutation Feature Importance (PFI) — "xáo trộn để đo tầm quan trọng"

Ý tưởng: nếu ta **xáo trộn** (permute) giá trị một đặc trưng x_j → phá vỡ quan hệ với nhãn, **độ đo chất lượng** của mô hình giảm bao nhiêu chính là **độ quan trọng** của x_j.

Quy trình (step-by-step):

1. Chọn thước đo hiệu năng $m(\cdot)$ (ví dụ: accuracy/AUC/R^2) và tập kiểm tra D.
2. Tính **điểm gốc**: $s_0 = m(f, D)$.
3. Với từng đặc trưng j:
   - Lặp B lần: hoán vị ngẫu nhiên cột x_j → nhận $D_{(j)}^{(b)}$; tính $s_j^{(b)} = m(f, D_{(j)}^{(b)})$.
   - **PFI**: $\mathrm{Imp}(j) = s_0 - \frac{1}{B}\sum_{b=1}^{B} s_j^{(b)}$.
4. Xếp hạng (giá trị càng lớn càng quan trọng).

> Biến thể conditional PFI (xáo trong từng "tầng" điều kiện) giúp giảm sai lệch khi đặc trưng tương quan mạnh.

City Development	Company Size	Company Type	Education	Experience
0.92	< 10	Pvt Ltd	Phd	10 years
0.54	100 - 500	Funded Startup	Graduate	5 years
...	...	...	...	...

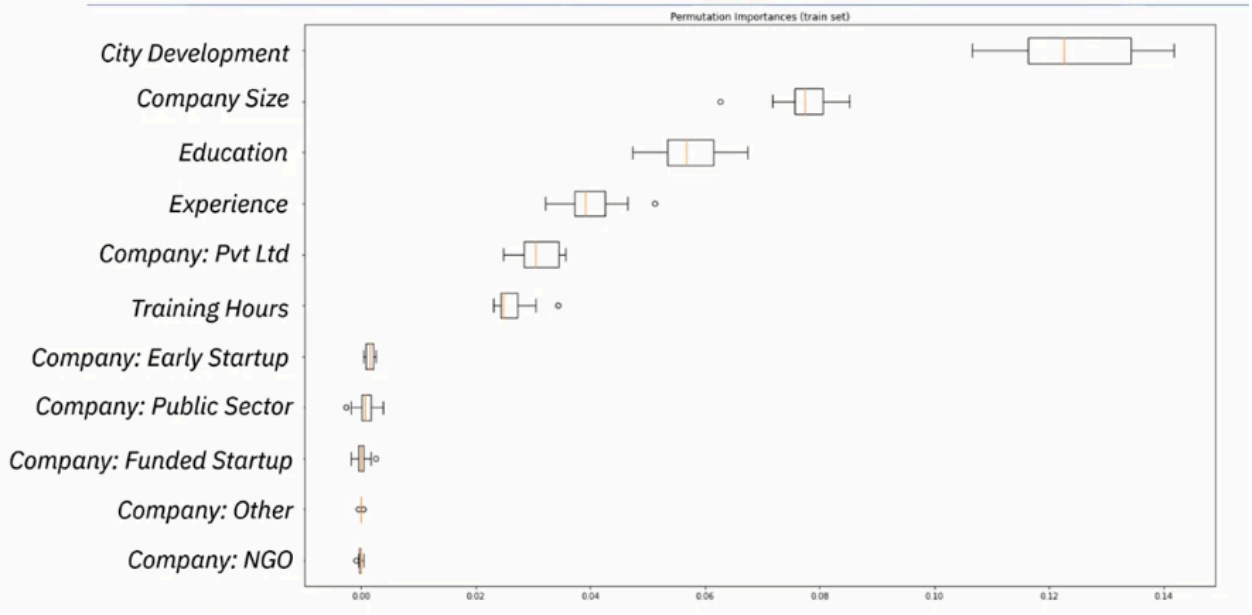
Prediction	Target
0.1	0
0.9	1
...	

Permutating feature values

City Development	Company Size	Company Type	Education	Experience
0.75	< 10	Pvt Ltd	Phd	10 years
0.55	100 - 500	Funded Startup	Graduate	5 years
...	...	...	...	...

Increasing Prediction Errors

Prediction	Target
0.15	0
0.85	1
...	



1.3. Partial Dependence Plot (PDP) — “ảnh hưởng biên” của đặc trưng

Ý tưởng: PDP biểu diễn **kỳ vọng dự đoán** khi quét giá trị một (hoặc hai) đặc trưng quan tâm và **lấy trung bình** qua phần còn lại.

Công thức 1D: với đặc trưng xjx_jxj và tập kiểm tra $\{x^{(i)}\}_{i=1}^{n}$,

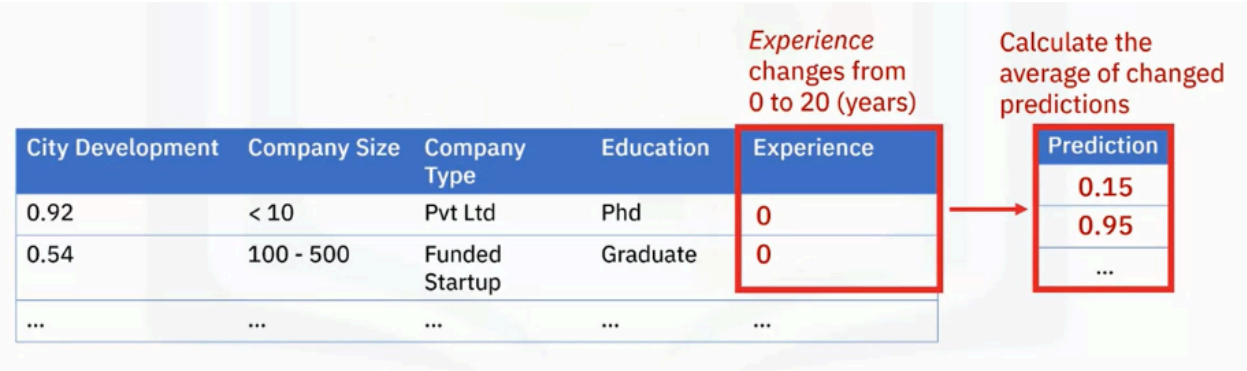
D,

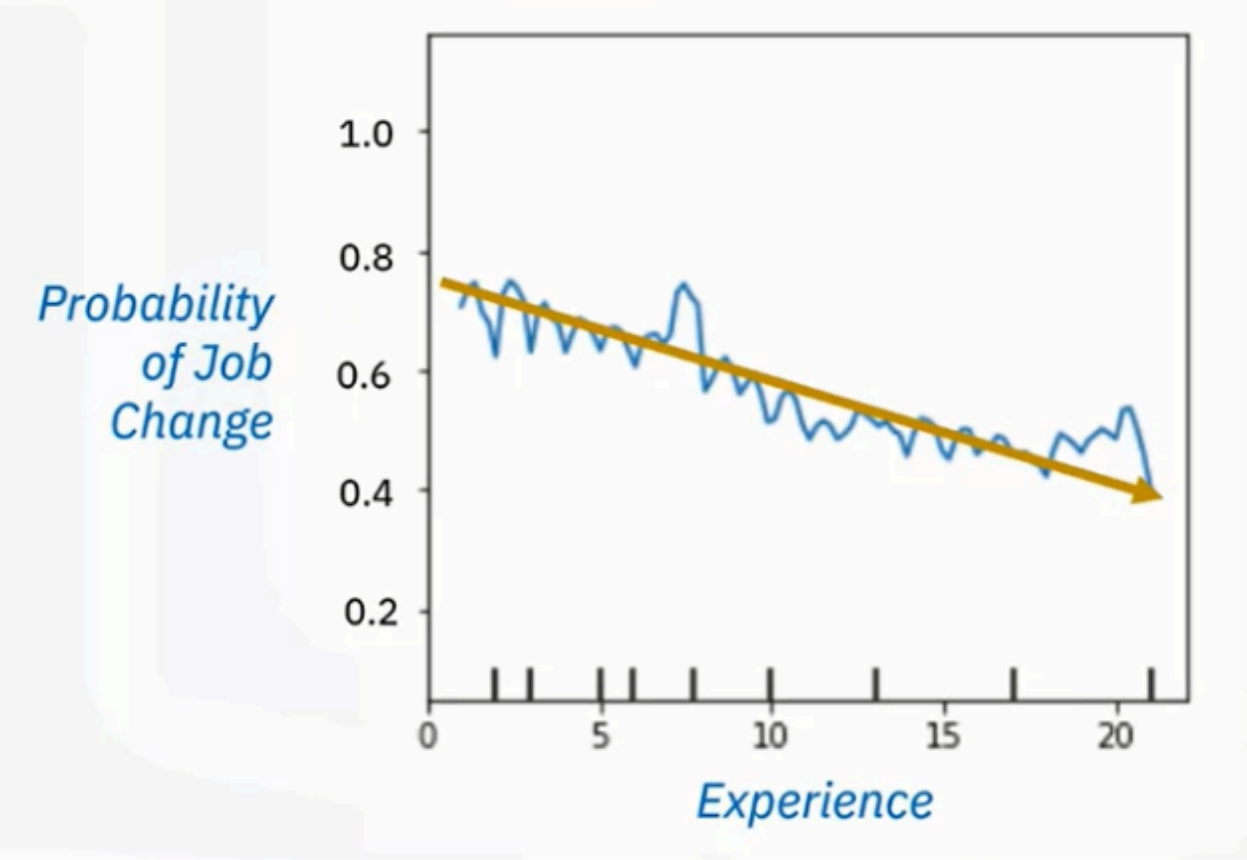
trong đó $x_{\setminus j}^{(i)}$ là vector đặc trưng của mẫu iii **trừ** thành phần j. Lặp nhiều giá trị z để vẽ đường cong PDP.

Quy trình (step-by-step):

1. Chọn đặc trưng quan tâm xjx_jxj và lưới giá trị $z_1, \ldots, z_T$.
2. Với mỗi z_t, thay $x_j = z_t$ cho **toàn bộ** nnn hàng, tính $\hat{f}_{PD}(z_t)$ theo công thức trên.
3. Vẽ $\{(z_t, \hat{f}_{PD}(z_t))\}$. (2D PDP làm tương tự với cặp đặc trưng.)

> PDP phù hợp để xem đơn điệu/phi tuyến, nhưng có thể gây hiểu nhầm khi đặc trưng tương quan mạnh; ICE plot giúp soi từng cá thể.





1.4. SHAP (SHapley Additive exPlanations) — “chia công bằng” mức đóng góp

Khái niệm: SHAP gán cho mỗi đặc trưng một **giá trị đóng góp** vào dự đoán $f(x)$, thỏa các tiên đề Shapley từ lý thuyết trò chơi; là khung **thống nhất** nhiều phương pháp quan trọng hoá trước đó.

Dạng cộng tính:

$$\text{D},$$

trong đó ϕ_j là “điểm SHAP” của đặc trưng j cho mẫu xxx, ϕ_0 là kỳ vọng dự đoán. Tính ϕ_j bằng cách **trung bình biên** đóng góp của x_j qua mọi tập con đặc trưng.

Quy trình (ý tưởng):

1. Xác định phân phối nền (background) để lấy kỳ vọng.

2. Với mỗi đặc trưng j, tính đóng góp biên giữa có/không có j cho nhiều tập con;
3. Lấy **trung bình có trọng số** (theo số phần tử tập con) → ϕ_j. (Thực tế dùng xấp xỉ: TreeSHAP, KernelSHAP…).

1.5. LIME (Local Interpretable Model-Agnostic Explanations) — “mô hình thay thế cục bộ”

Ý tưởng: xung quanh một điểm x, sinh lân cận, gán **trọng số theo khoảng cách**, rồi khớp một **mô hình đơn giản** (ví dụ tuyến tính hiếm tham số) để **xấp xỉ** fff **cục bộ**.

Hàm mục tiêu (khái quát):

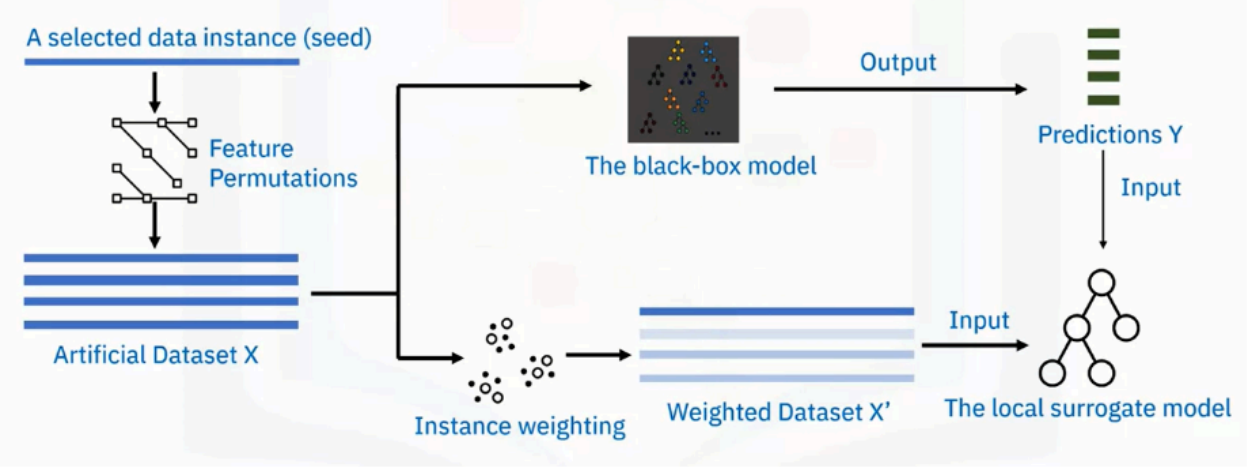
$$\text{D},$$

với G là họ mô hình interpretable (linear/rule list), π_x là hạt nhân khoảng cách (kernel) quanh x, $\mathcal{L}$ đo **độ trung thành cục bộ** (fidelity), Ω phạt độ phức tạp để lời giải “dễ hiểu”.

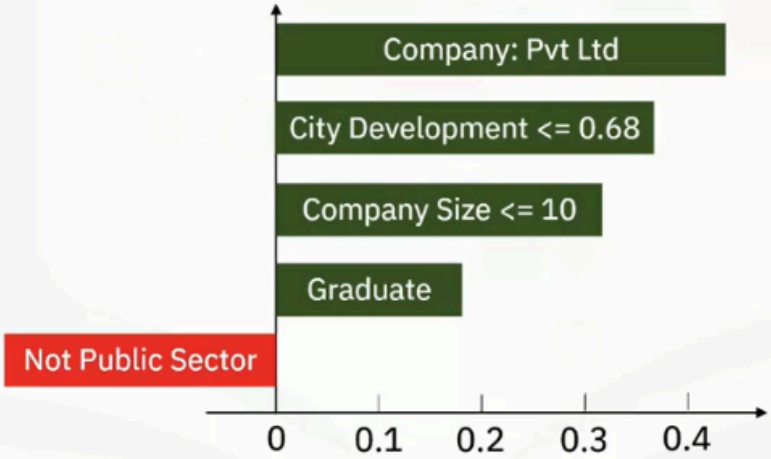
Quy trình (step-by-step):

1. Tạo NNN mẫu lân cận quanh xxx; tính $f(\cdot)$ cho từng mẫu.
2. Tính trọng số $\pi_x(\cdot)$ theo khoảng cách tới x.
3. Khớp ggg (linear/ridge/LASSO) với trọng số π_x.
4. Hệ số của g chính là lời giải thích cục bộ cho x.

> Global surrogate: khớp một mô hình đơn giản trên toàn bộ dữ liệu để bắt chước black-box; nhưng có thể không trung thành nếu dữ liệu có nhiều cụm/hành vi khác nhau → khi đó ưu tiên local surrogate (LIME).



City Development	Company Size	Company Type	Education	Experience
0.68	<= 10	Pvt Ltd	Graduate	4 years



2) Modeling Unbalanced Classes

2.1. Vấn đề & chỉ số nên dùng

- Mất cân bằng làm **accuracy đánh lừa** (đoán luôn lớp đa số vẫn cao). Nên dùng **AUC/PR-AUC, F1, Cohen's Kappa**, cùng **Precision/Recall** và **điểm vận hành** trên ROC/PR.

Trade-off: tăng Recall thường giảm Precision → chọn **ngưỡng** theo mục tiêu nghiệp vụ.

2.2. Chiến lược dữ liệu (resampling & trọng số)

(A) **Stratified split** trước tiên

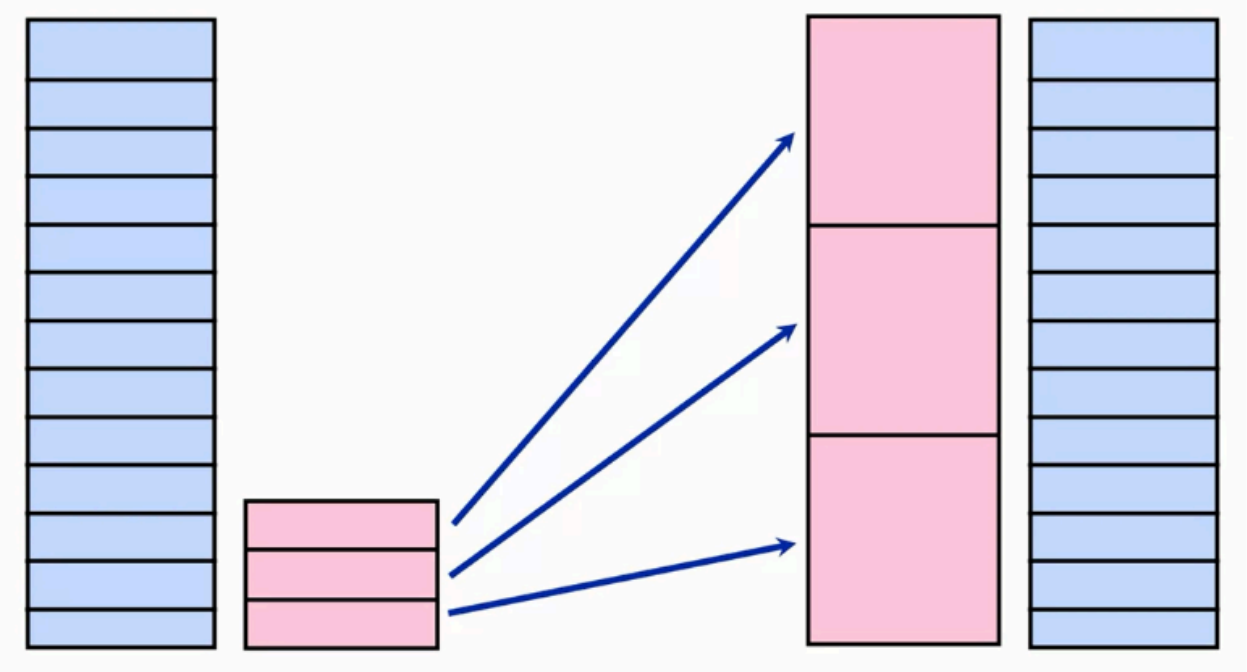
Giữ tỷ lệ lớp ở train/test/val để đánh giá công bằng.

(B) **Oversampling** lớp thiểu số

- **Random OverSampling:** lấy mẫu lặp có hoàn lại từ lớp thiểu số. Dễ dùng, hợp dữ liệu rời rạc.
- **SMOTE/ADASYN:** sinh mẫu tổng hợp giữa điểm thiểu số và láng giềng (SMOTE) hoặc **tự thích nghi** theo độ khó khu vực biên (ADASYN).

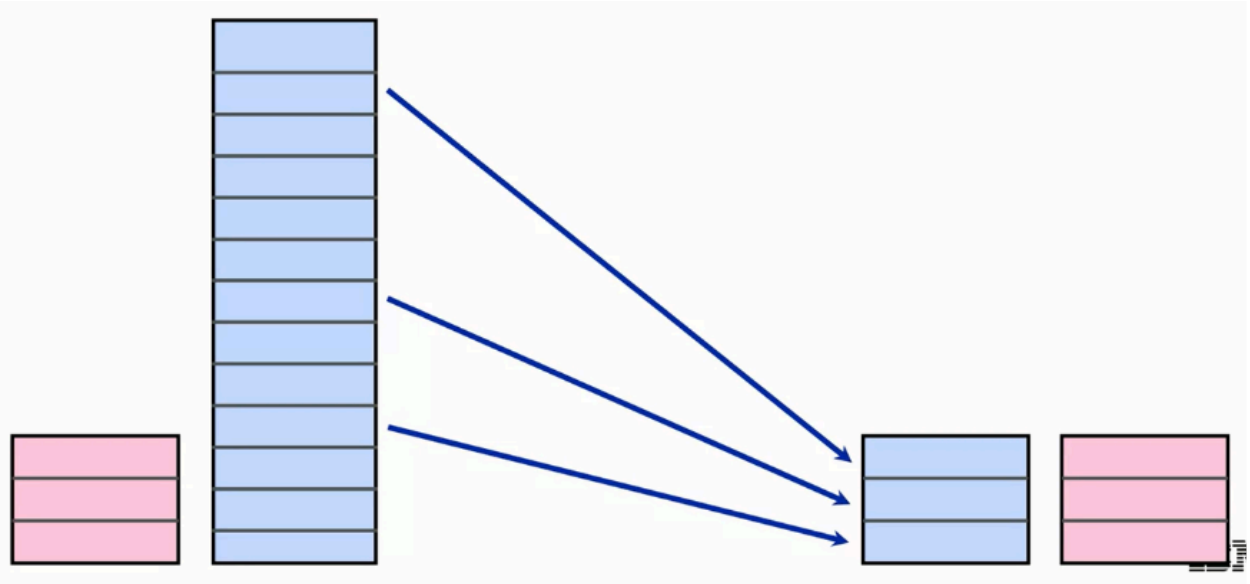
SMOTE (minh hoạ):

Chọn điểm thiểu số x, chọn láng giềng x_{NN}, sinh $x' = x + \lambda(x_{NN} - x), \lambda \sim U(0,1)$



(C) **Undersampling** lớp đa số

- **NearMiss, Tomek Links, ENN:** các kỹ thuật chọn/bỏ điểm đa số để làm ranh giới rõ hơn.



(D) **Kết hợp & Ensemble**

- **SMOTE + Tomek/ENN** (làm sạch sau oversample).
- **Balanced Bagging (Blagging)**: bagging với mẫu cân bằng từng bootstrap.

(E) **Class weights** (nếu mô hình hỗ trợ)

Đặt $w_c \propto \frac{1}{\text{freq}(c)}$ để cân bằng **hàm mất mát**, **không** phải sửa dữ liệu.

2.3. Quy trình step-by-step

1. **Khám phá & phân tích tỷ lệ lớp**; quyết định **metric** phù hợp (F1/PR-AUC).
2. **Stratified split** (train/val/test).
3. Chọn **chiến lược dữ liệu**:
   - ít dữ liệu thiểu số → ưu tiên **oversampling** (Random/SMOTE/ADASYN);
   - dữ liệu rất lớn → cân nhắc **undersampling** để tiết kiệm tính toán;
   - ranh giới nhiễu → **SMOTE+ENN/Tomek** để "làm sạch".

4. Huấn luyện nhiều mô hình/thiết lập; dùng **CV** để chọn cấu hình tốt theo **AUC/F1**.
5. **Chọn ngưỡng** trên ROC/PR theo mục tiêu (ví dụ tối đa F1 hoặc chi phí FP/FN).
6. **Diễn giải** mô hình đã chọn bằng **PFI/PDP/SHAP/LIME** để kiểm định hợp lý nghiệp vụ (feature drift, leakage…).

2.4. Ảnh hưởng điển hình khi resampling

- **Downsampling:** nâng **Recall**, hạ **Precision** khá mạnh (thiểu số được “phóng đại” tầm quan trọng).
- **Upsampling:** Recall vẫn cao hơn Precision nhưng “dịu” hơn so với downsample; thường được coi là **hợp lý** cho dữ liệu mất cân bằng.