LIRIS

UMR 5205 CNRS

Détection d'anomalies

Haytham Elghazel

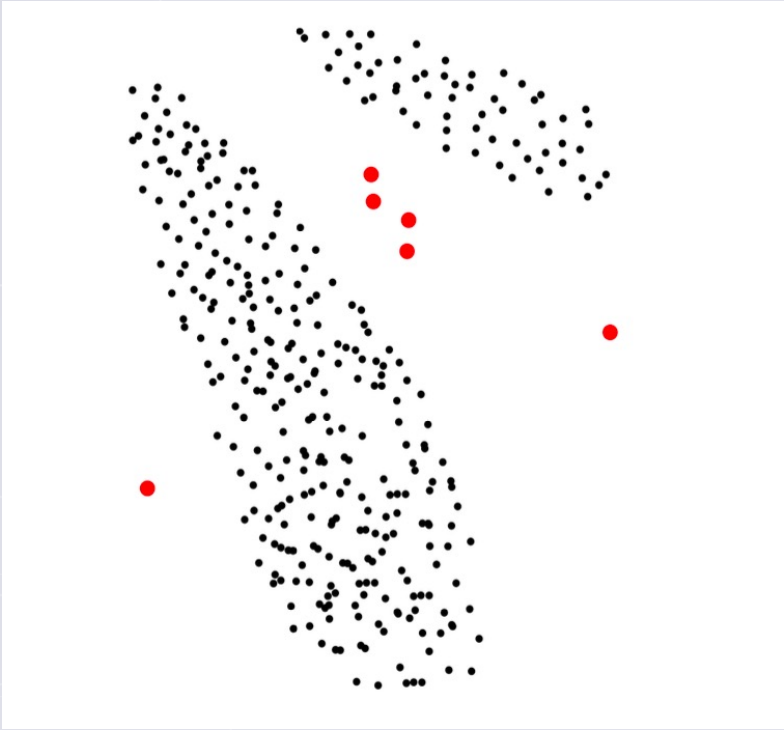
Laboratoire d'InfoRmatique en Image et Systèmes d'information

Pôle Data Science, Equipe DM2L

cnrs INSA Université Claude Bernard Lyon 1 Université Lumière Lyon 2 École Centrale Lyon

Détection d'anomalies

Identifier les instances ayant un comportement non conforme



Applications : Network intrusions, fraude de carte de crédit, surveillance, assurance

Détection d'anomalies

- Supprimer ou modifier les observations atypiques à un modèle sans justification serait totalement contraire à l'éthique.

- L'objectif est avant tout de les identifier car ce sont celles, les plus susceptibles d'être la conséquence d'une erreur (*à confirmer*) de mesure, de libellé, ou encore une anomalie, défaillance ou tentative de fraude, d'intrusion, selon le contexte.

Détection d'anomalies

- La **détection d'anomalies** contient deux familles d'approches :
  - **Détection d'outliers (Outlier detection) :** Les données d'apprentissage contiennent des outliers et qui sont des observations qui se trouvent loin des autres. Il s'agit d'apprendre à détecter les anomalies dans le jeu de données initial en cherchant des régions denses (où les données sont le plus concentrées) tout en ignorant les anomalies.
  - **Détection de nouveautés (Novelty detection) :** Ici le jeu de données d'apprentissage n'est pas pollué par des anomalies. Il s'agit de détecter des anomalies dans les données futures non observées (nouvelles données). Les outliers sont ainsi appelés nouveautés.

Plusieurs approches

- **Approches supervisées**
  - Des labels à la fois pour les instances normales et anomalies
  - Les anomalies appartiennent à la classe rare
  - Données déséquilibrées

- **Approches non supervisées**
  - Pas de labels fournis
  - Base d'apprentissage = données normales + anomalies
  - Les anomalies sont très rares

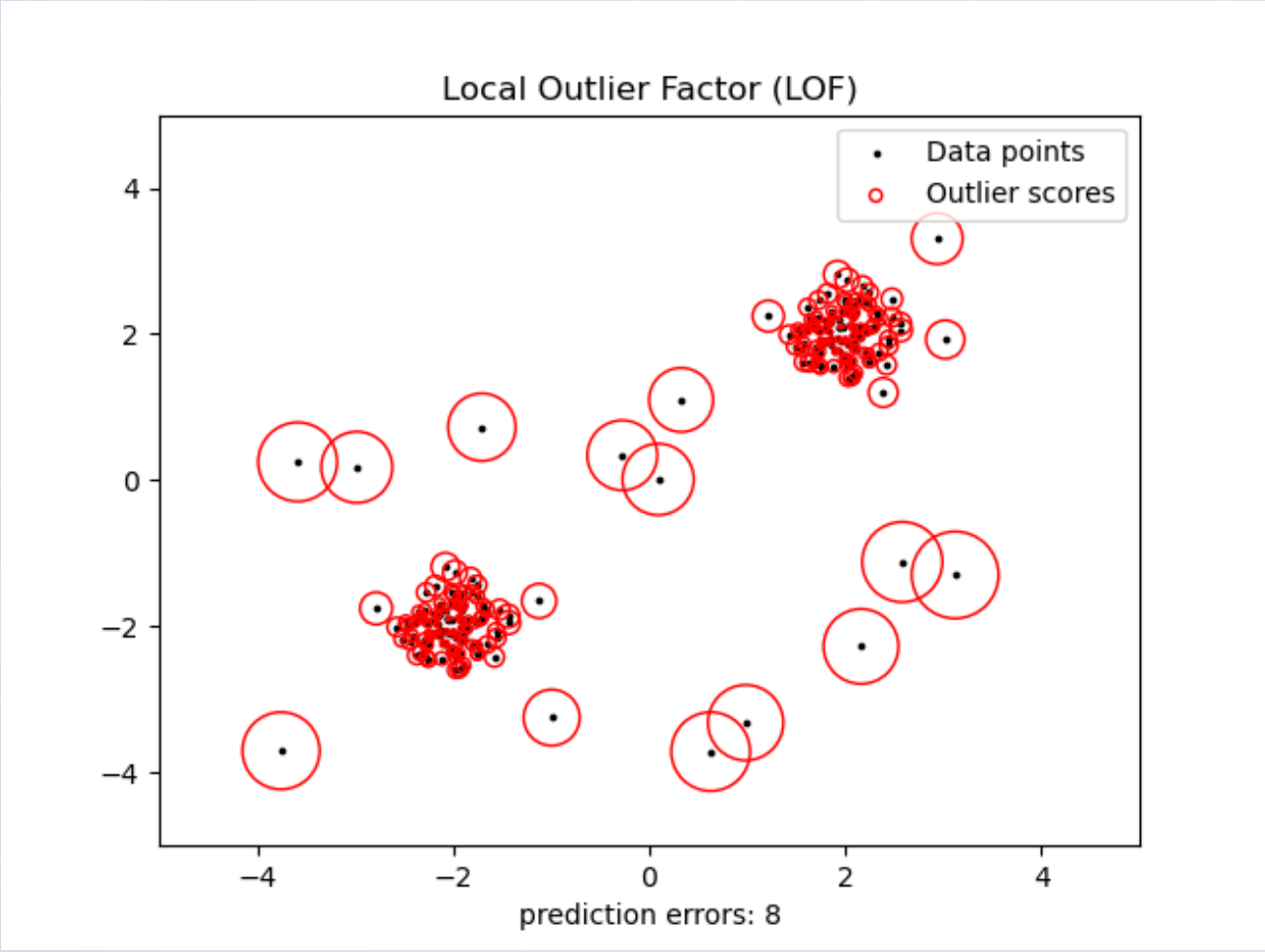
Approches non supervisées

- **Approches basées sur le voisinage**
  - L'anomalie ou l'isolement d'une observation est apprécié par la proximité des points de son voisinage
    - **Exemple :** Local Outlier Factor (LOF)
    - LOF compare la densité locale des observations. S'il existe une différence entre le point observé et ses voisins, le point est considéré comme une anomalie. Cette méthode est basée sur les k plus proches voisins : la densité locale d'une observation est évaluée en considérant les k plus proche observations de son voisinage.

Local Outlier Factor

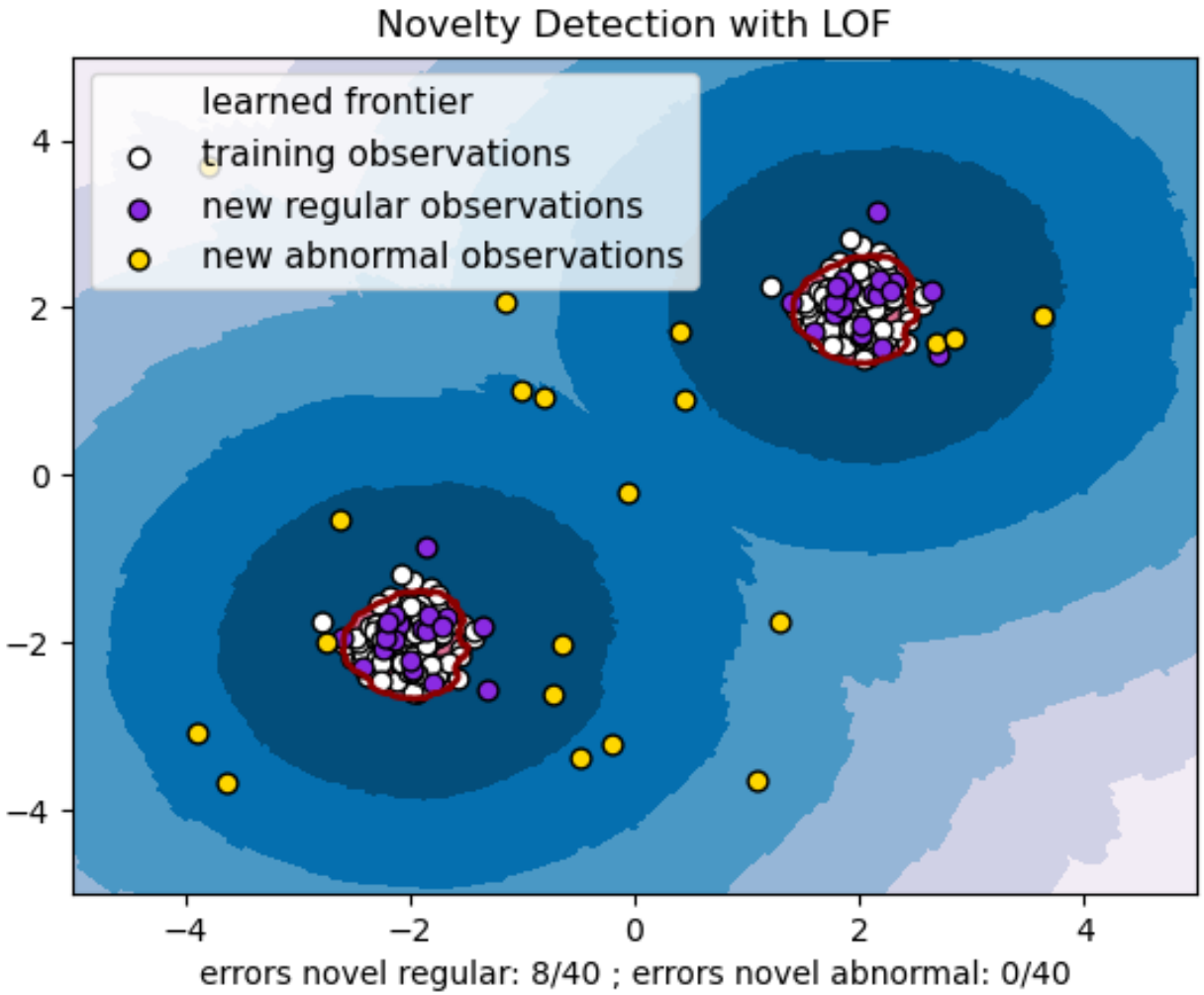
- On définit pour chaque point **x**, $\mathbf{D_k(x)}$ sa distance par rapport à son $\mathbf{k^{ème}}$ plus proche voisin et $\mathbf{N_k(x)}$ l'ensemble de ses **k** plus proches voisins.
- On définit la distance d'accessibilité $\mathbf{R_k(x,y)}$ de **x** par rapport à **y** comme étant le **max(d(x,y) et $\mathbf{D_k(y)}$)**.
- On définit la distance d'accessibilité moyenne $\mathbf{AR_k(x)}$ de **x** comme étant égale à la moyenne des distances d'accessibilité de **x** avec tous les points de son voisinage $\mathbf{N_k(x)}$
- On définit la densité d'accessibilité locale $\mathbf{f_k(x)}$ comme étant l'inverse de $\mathbf{AR_k(x)}$.
- **Une instance normale est sensée avoir une densité locale similaire à ses voisins, alors qu'une instance anormale est sensée avoir une beaucoup plus petite densité locale.**
- On définit **LOF(x)** par la moyenne du rapport $\mathbf{f_k(y)/f_k(x)}$ pour tous les **y** dans $\mathbf{N_k(x)}$.
- Le LOF mesure l'écart local d'un point par rapport à ses **k** voisins les plus proches
- **Si ce score est proche de 1, nous pouvons en conclure que l'observation est comparable à ses voisins. Si le score est inférieur à 1, nous pouvons dire que l'observation se trouve dans une région dense. Dans les deux cas, l'observation n'est pas considérée comme un outlier.**
- **Un score est largement supérieur à 1 indique qu'on à faire à un outlier.**

Local Outlier Factor



- Utilisée pour la détection de nouveautés ou d'outliers.
- LOF est une méthode est très puissante en dimension modérée.

Local Outlier Factor

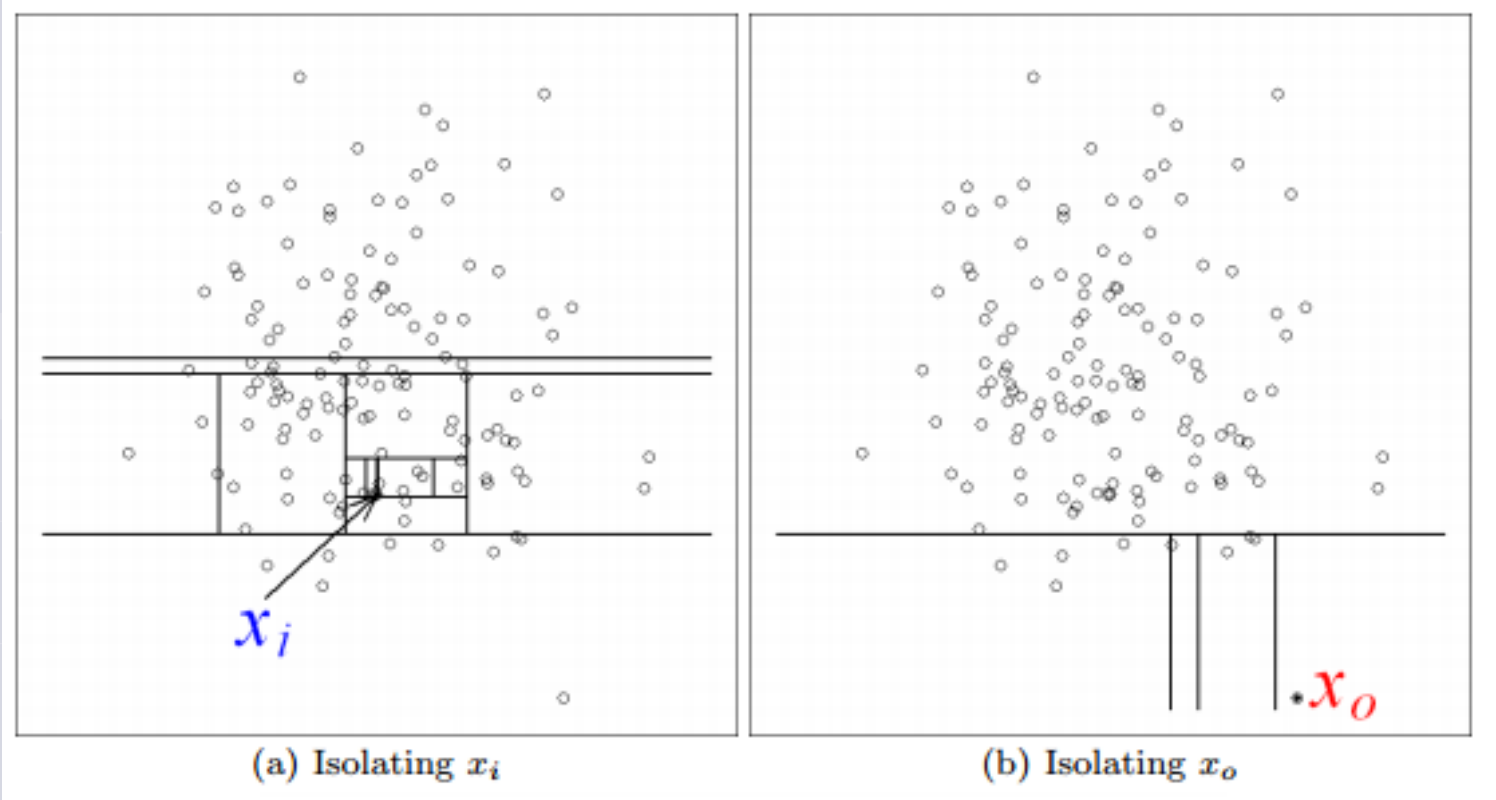


Approches non supervisées

- **PCA : Analyse en composantes principales**
- **One class SVM : Pour la détection de nouveauté.**
  - L'objectif est de séparer toutes les observations, de l'origine, dans l'espace de représentation en maximisant la marge, à savoir la distance entre l'hyperplan et l'origine.
- **Isolation Forest (Forêt d'isolation) : Pour la détection d'outliers.**

Isolation Forest

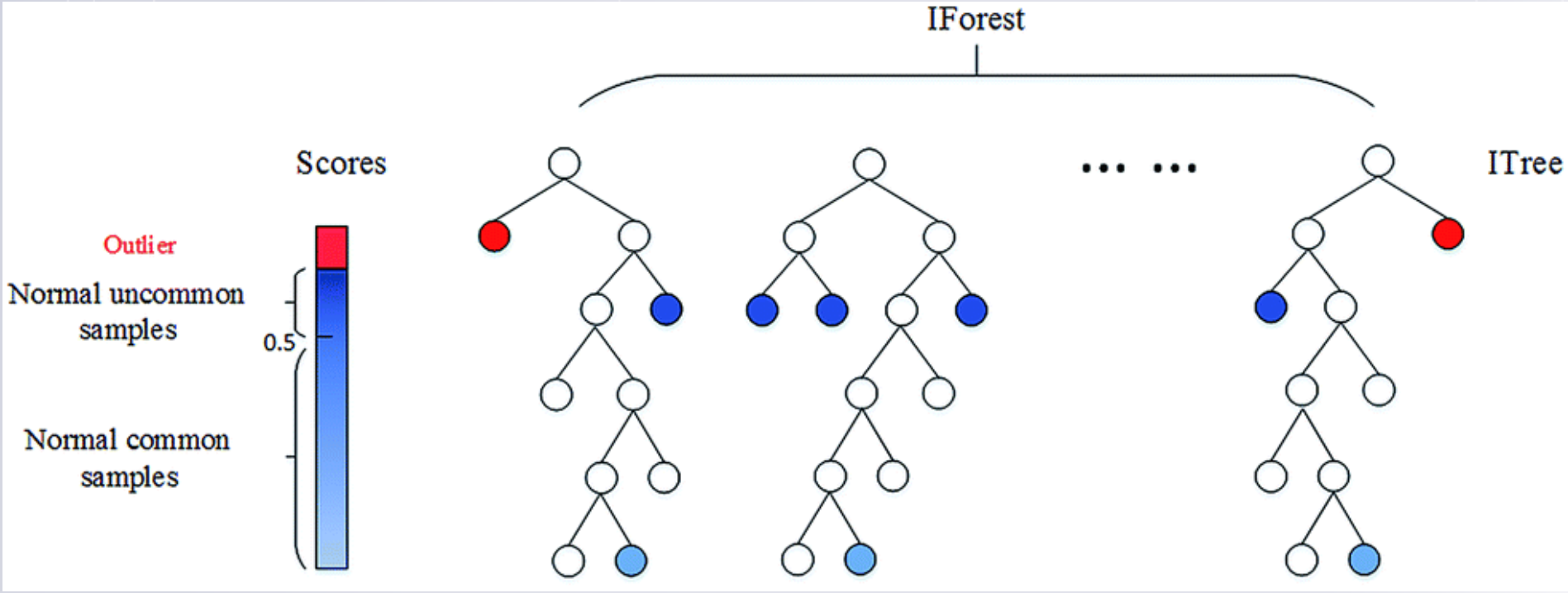
Principe : Les anomalies sont rares et différentes. Elles sont donc susceptibles au mécanisme d'isolation



(a) Isolating x_i (b) Isolating x_o

Isolation Forest

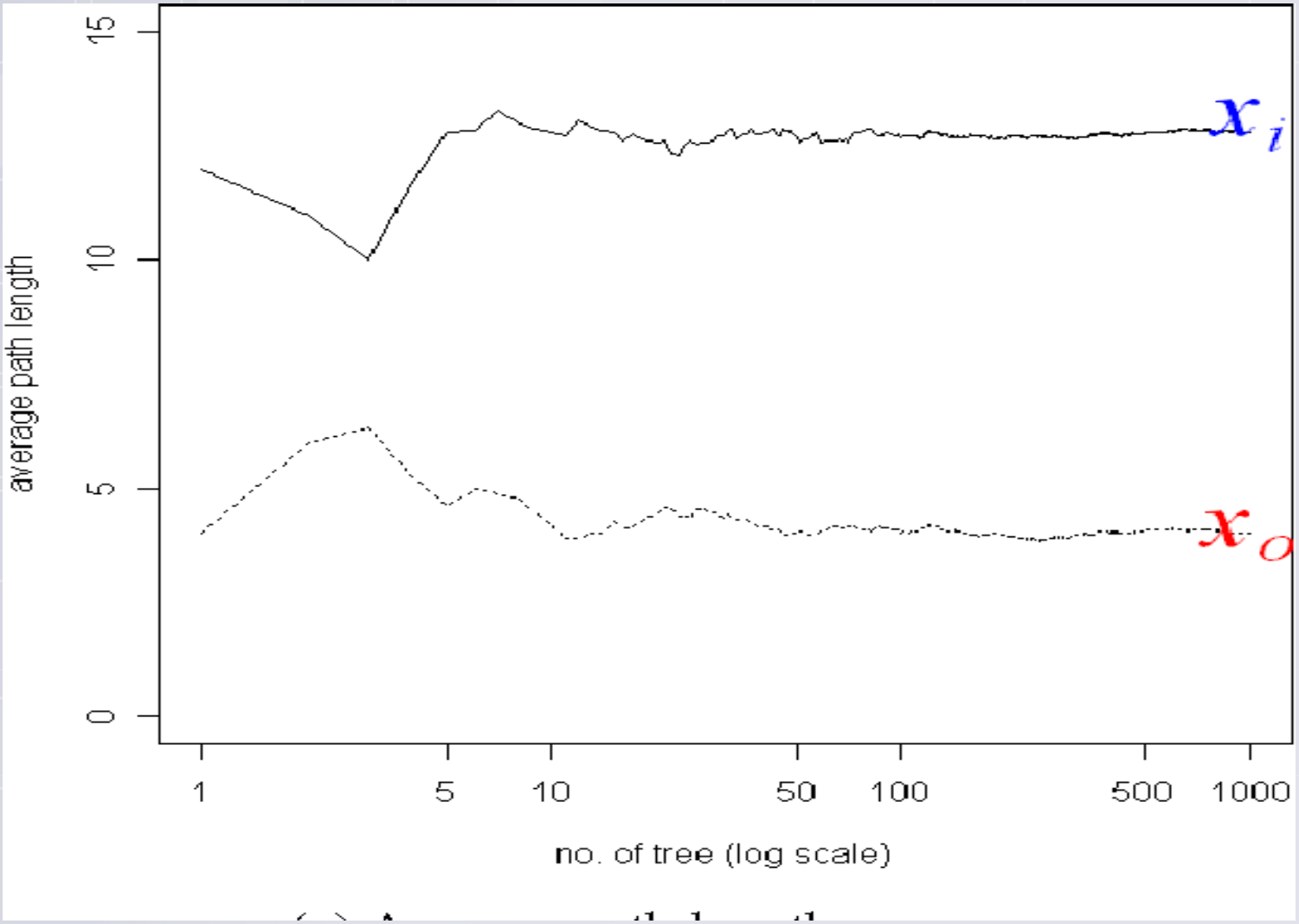
Principe : Les anomalies sont rares et différentes. Elles sont donc susceptibles au mécanisme d'isolation



Isolation Forest

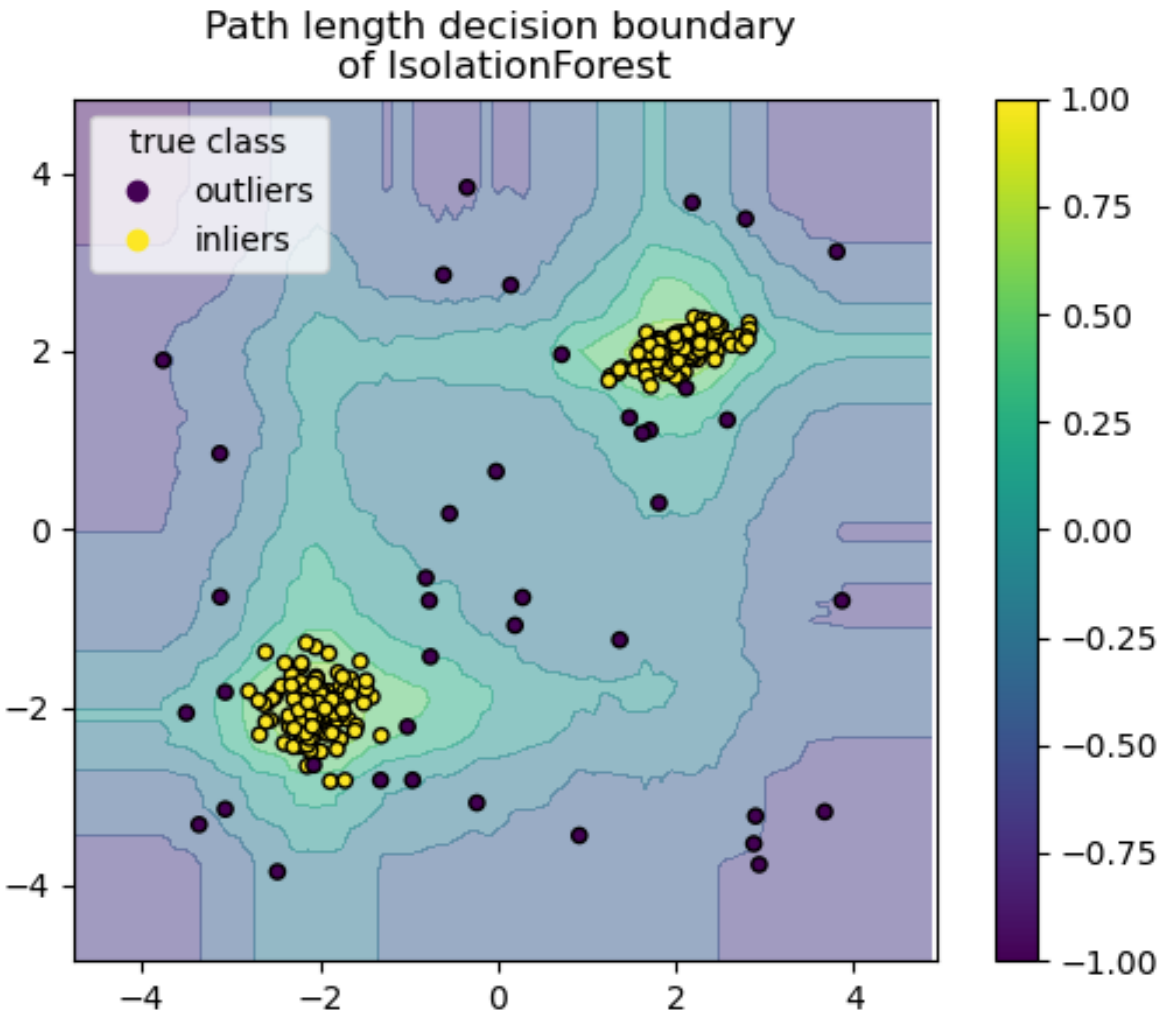
- **Le principe repose sur la construction d'un ensemble d'arbres complètement aléatoires : *isolation tree*.**
- **Chaque arbre est construit sur un échantillon aléatoire des instances**
- **Division opérée dans chaque nœud via un tirage aléatoire d'une variable et**
  - un seuil aléatoire pour une **variable quantitative**
  - une répartition aléatoire des modalités en deux groupes pour une **variable qualitative**
  - La construction de l'arbre jusqu'à l'obtention d'une observation par feuille.
- **Le score de l'isolement ou de l'anomalie d'une observation est obtenue par la *longueur du chemin atteignant cette observation*. Plus celui-ci est court, plus l'observation est considérée isolée ou atypique.**

Isolation Forest



Convergence rapide

Isolation Forest



Approches supervisées

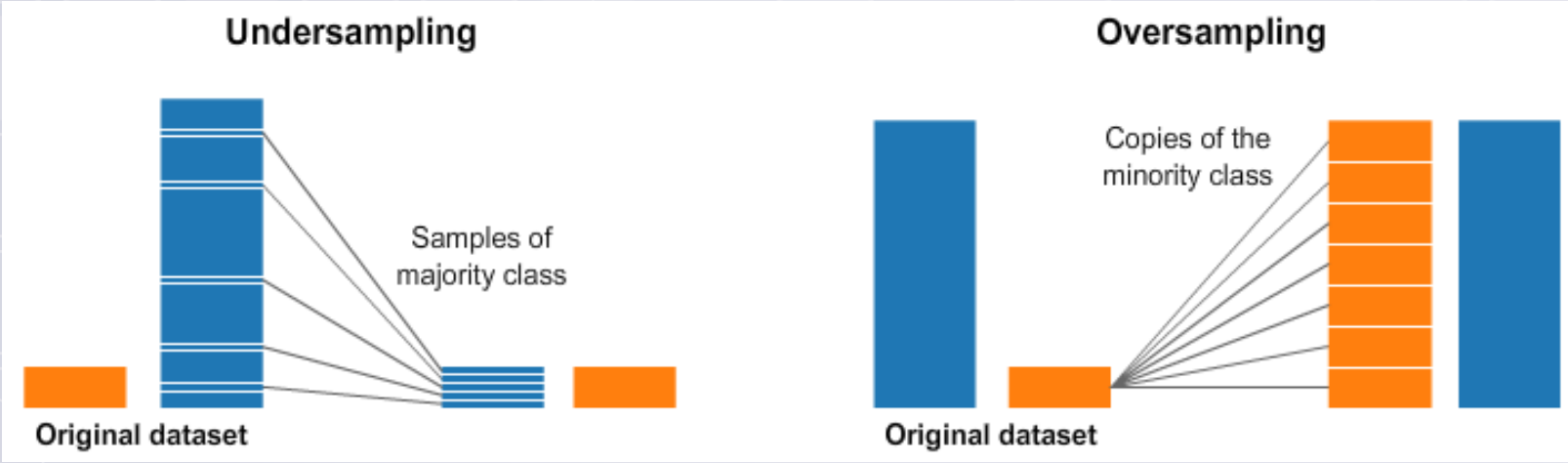
- **Approches d'apprentissage supervisées sur des données déséquilibrées**

- **Quand on entraîne un modèle de ML supervisé sur des données déséquilibrées (par exemple 95% de « normal » et 5% « anomalie »), le modèle a tendance à « ignorer » la minorité, car il apprend à minimiser une fonction de perte qui donne le même poids à chaque observation.**

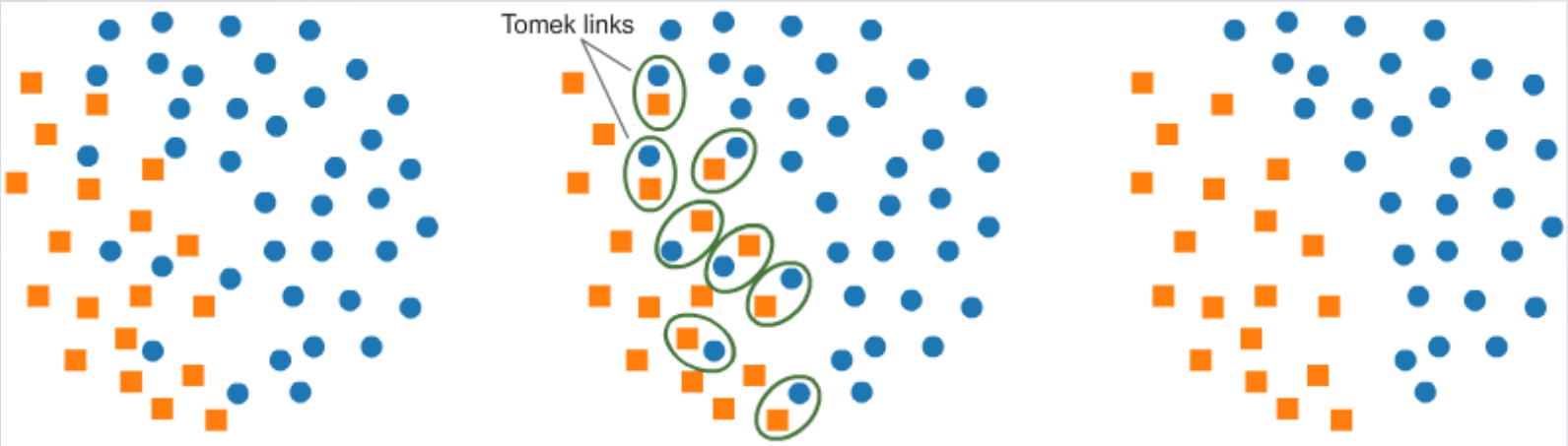
- **Adaptation des approches supervisées existantes :**
  - ***Undersampling*** : sous échantillonnage **(Bibliothèque `imblearn`)**
  - ***Oversampling*** : sur échantillonnage **(Bibliothèque `imblearn`)**
  - ***Balancing :*** Pondération des classes **(Bibliothèque `sklearn`)**

Approches supervisées

- **Random Undersampling et Random Oversampling**

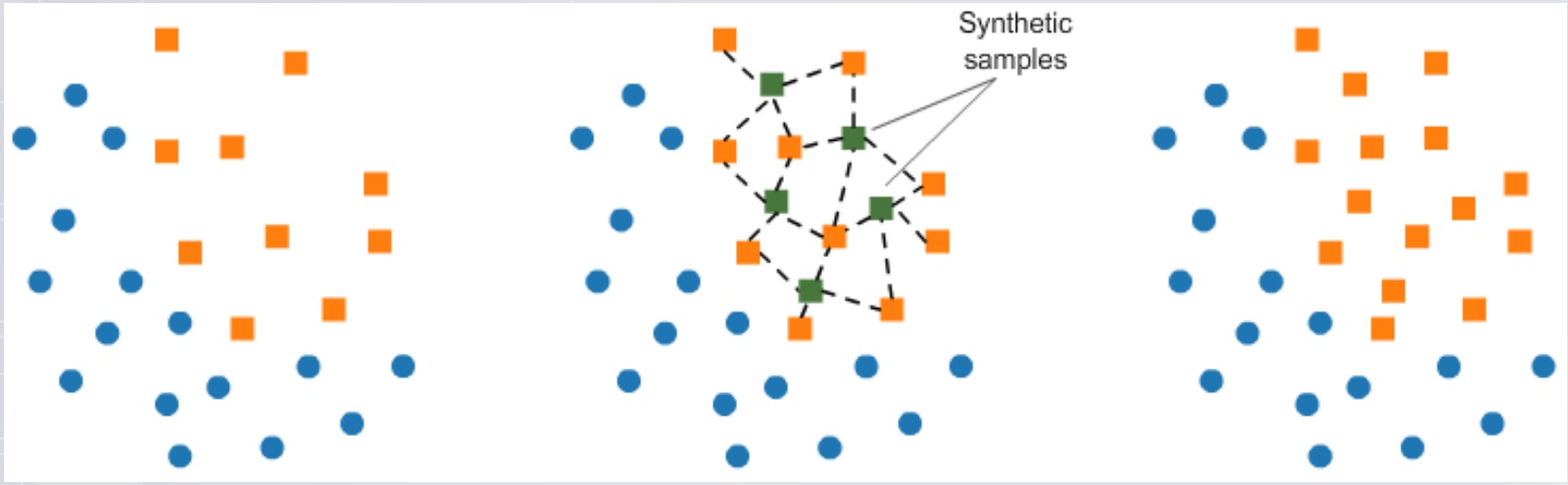


- **Tomek Links :** une approche d'undersampling



Approches supervisées

- **Synthetic Minority Oversampling Technique (SMOTE) :** une approche d'oversampling



1. Sélectionner aléatoirement une observation minoritaire "initiale".
2. Identifier ses *k* plus proches voisins parmi les observations minoritaires (où *k* est un paramètre défini par l'utilisateur).
3. Choisir aléatoirement l'un des k plus proches voisins.
4. Générer aléatoirement un coefficient α.
5. Créer un nouvel individu entre l'observation initiale et le plus proche voisin choisi, selon la valeur du coefficient α. Par exemple, si α=0.5, le nouvel individu sera positionné à mi-chemin entre l'observation initiale et le plus proche voisin choisi.

Approches supervisées

- **Balancing :** une approche de rééquilibrage des classes
  - **Au lieu de modifier les données (*undersampling* ou *oversampling*), on peut modifier la fonction de perte du modèle pour pénaliser plus fortement les erreurs sur la classe minoritaire.**
  - **Dans *scikit-learn*, beaucoup de modèles (régression logistique, SVM, forêts aléatoires, etc.) proposent l'argument *class_weight***

clf = LogisticRegression(class_weight={0:1, 1:20})

  - **Dans une régression logistique par exemple, la fonction de perte (*binary cross-entropy*) est :**

$$Loss = -y * log(p) - (1 - y) * log(1 - p)$$

$y = 1$ pour un positif (anomalie), p est la probabilité prédite par le modèle

  - **Avec class_weight={0:1, 1:20}, la fonction de perte devient :**

$$NewLoss = -20 * \ y * log(p) - 1 * (1 - y) * log(1 - p)$$

Une erreur sur un positif (*minoritaire*) compte 20 fois plus qu'une erreur sur un négatif.

- **Si le modèle prédit $p = 0.3$ pour une observation positive $(y = 1)$, $NewLoss = -20 * log(0.3) = 10.45$**
- **Si le modèle prédit $p = 0.7$ pour une observation négative $(y = 0)$, $NewLoss = -1 * log(O.3) = 0.52$**

Approches supervisées

- **Comment calculer les poids :**

$$w_i = \frac{N}{n_{classes} \cdot count_i}$$

  - N **= nombre total d'exemples**
  - $n_{classes}$ **= nombre de classes**
  - $count_i$ **= nombre d'exemples dans la classe** i

- **Exemple : Classe 0 (« normal ») avec 90 exemples, Classe 1 (« anomalie ») avec 10 exemples**

$$w_0 = \frac{100}{2 \times 90} = \frac{100}{180} \approx 0.56$$

$$w_1 = \frac{100}{2 \times 10} = \frac{100}{20} = 5.0$$

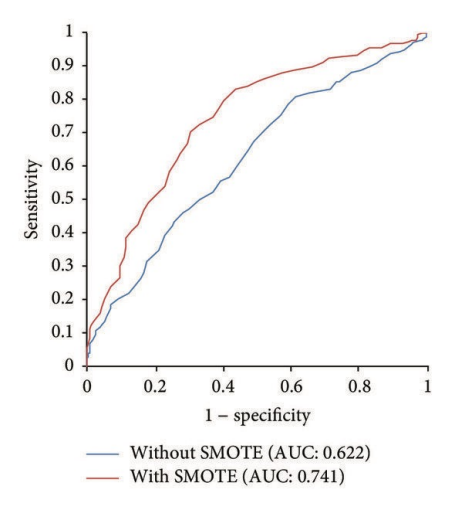
Evaluation

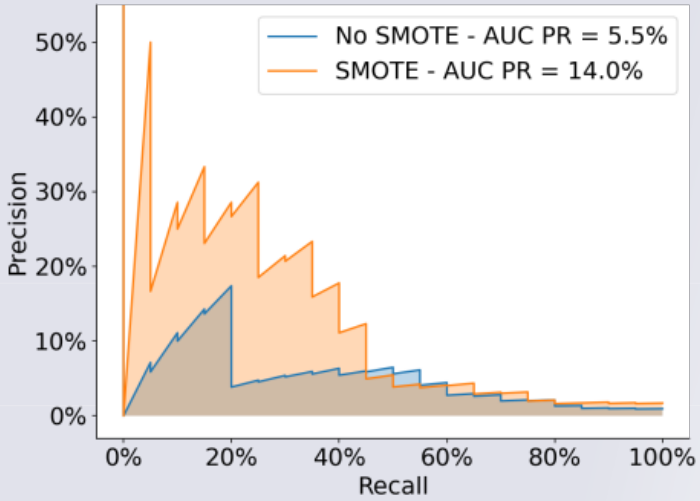
- **Attention : Bien choisir la métrique d'évaluation**
  - ***Accuracy déconseillée (trompeuse)***
  - ***Si vous vous intéressez aux classes prédites (0/1) :***
    - ***Balanced accuracy*** *est une très bonne métrique d'évaluation pour des données déséquilibrées si vous portez autant d'attention pour la classe positive que pour la classse négative.*
    - ***F1-score :*** *est une très bonne métrique d'évaluation pour des données déséquilibrées si vous portez plus d'attention à la classe positive.*

	Classifieur naïf		Mauvais Classifieur		Bon classifieur	
	999900	0	997990	1910	999890	10
	100	0	10	90	10	90
Accuracy	0,9999		0,99808		0,99998	
Précision +	0		0,045		0,9	
Rappel +	0		0,9		0,9	
Rappel -	1		0,998089809		0,999989999	
Balanced Accuracy	0,5		0,949044904		0,949994999	
F1 Score	0		0,085714286		0,9	

Evaluation

- ***Si vous vous intéressez aux probabilités prédites :***
  - ***AUC-ROC (roc_auc_score) :*** *c'est l'aire sous la courbe ROC* ***(roc_curve)*** *qui mesure la capacité globale du modèle à séparer les classes* ***sans choisir de seuil.*** *Utile si vous portez autant d'attention pour la classe positive que pour la classe négative.*
  - ***l'AUC-PR (average_precision_score) : l'aire sous la courbe (precision_recall_curve) formée par les points de coordonnées (Rappel+, Précision+) en fonction du seuil.*** *Utile si vous portez plus d'attention à la classe positive.*





Evaluation

- Plus l'AUC est élevée, plus le modèle est capable de donner des probabilités discriminantes.
- Un modèle avec AUC = 0.9 est globalement meilleur pour séparer les classes qu'un modèle avec AUC = 0.7.
- Un modèle avec une très bonne AUC peut être mal calibré et donc donnera des mauvaises précision, rappel ou F1.
- Par défaut, on prend un seuil de 0.5, Mais ce seuil est arbitraire.
- On peut déplacer ce seuil pour maximiser la précision, le rappel, ou le F1 selon les besoins métiers.
- Si le modèle a une bonne AUC ça veut dire qu'il existe quelque part un seuil optimal qui permet d'obtenir une meilleure classification 0/1 (en maximisant le F1 ou la balanced accuracy par exemple).

Evaluation

- Attention : Ne pas évaluer votre modèle sur un échantillon équilibré
- Les anomalies se produisant souvent de manière complètement nouvelle, le modèle ne pourra pas détecter les nouvelles anomalies sur lesquelles il n'a pas été entraîné.