

Données publiques

- On entraîne l'algorithme **Local Outlier Factor (LOF)** sur données publiques

```
from sklearn.neighbors import LocalOutlierFactor
from sklearn.metrics import f1_score, recall_score, classification_report, confusion_matrix

# =====
# LocalOutlierFactor - Détection anomalies
# =====
lof = LocalOutlierFactor(n_neighbors=20, contamination=0.15, novelty=True)
lof.fit(X_train)

# Score de décision (négatif -> anomalie)
scores_lof = -lof.decision_function(X_test_with_anom)

# Seuil basé sur proportion anomalies
cont = max(0.01, min(0.5, y_test.mean()))
th = np.percentile(scores_lof, 100*(1-cont))

# Prédictions
y_pred_lof = (scores_lof >= th).astype(int)

# Évaluation
f1_lof_macro = f1_score(y_test, y_pred_lof, average='macro', zero_division=0)
recall_lof_macro = recall_score(y_test, y_pred_lof, average='macro', zero_division=0)

print("\n--- LocalOutlierFactor (macro) ---")
print("F1 macro:", f1_lof_macro)
print("Recall macro:", recall_lof_macro)
print(classification_report(y_test, y_pred_lof, zero_division=0))
print("ConfusionMatrix (LOF):\n", confusion_matrix(y_test, y_pred_lof))
```

Figure 5-Erreur ! Il n'y a pas de texte répondant à ce style dans ce document.-1 : entraînement de l'algorithme Local Outlier Factor sur données publiques

- Voici le résultat du modèle Local Outlier Factor sur données publiques

```

--- LocalOutlierFactor (macro) ---
F1 macro: 1.0
Recall macro: 1.0
      precision    recall   f1-score   support
      0.0        1.00     1.00     1.00      290
      1.0        1.00     1.00     1.00       51

      accuracy          1.00      341
macro avg        1.00     1.00     1.00      341
weighted avg     1.00     1.00     1.00      341

ConfusionMatrix (LOF):
[[290  0]
 [ 0  51]]

```

Figure 5-Erreur ! Il n'y a pas de texte répondant à ce style dans ce document.-2 : résultat model Local Outlier Factor sur données publique

Données privées

- On entraîne l'algorithme **Local Outlier Factor (LOF)** sur données privées

```

# Local Outlier Factor
lof = LocalOutlierFactor(n_neighbors=20, contamination=0.15, novelty=True)
lof.fit(X_train)
y_pred = lof.predict(X_test_with_anom)
y_pred = np.where(y_pred == 1, 0, 1)

# Scores
f1 = f1_score(y_test, y_pred, average='macro')
recall = recall_score(y_test, y_pred, average='macro')
acc = accuracy_score(y_test, y_pred)

print("== Résultats Local Outlier Factor ==")
print(f"Accuracy : {acc:.4f}")
print(f"F1-macro : {f1:.4f}")
print(f"Recall-macro: {recall:.4f}")

```

Figure 5-Erreur ! Il n'y a pas de texte répondant à ce style dans ce document.-3 : entraînement de l'algorithme Local Outlier Factor sur données privées

- Voici le résultat du modèle Local Outlier Factor sur données privées

```
==== Résultats Local Outlier Factor ====
Accuracy      : 0.8651
F1-macro      : 0.8015
Recall-macro: 0.9207
```

Figure 5-Erreur ! Il n'y a pas de texte répondant à ce style dans ce document.-4 : résultat model Local Outlier Factor sur données privées