Clase 099 — Detección de anomalías: Isolation Forest, LOF, One-Class SVM

Parte: 1 — Machine Learning Clásico · Fuente: Géron, cap. 9 + sklearn outlier detection.

Duración estimada: 70 min.

Clase 099 — Detección de anomalías: Isolation Forest, LOF, One-Class SVM

> *Parte: 1 — Machine Learning Clásico · Fuente: Géron, cap. 9 + sklearn outlier detection. Duración estimada: 70 min.*

Objetivo

Que el alumno detecte puntos anómalos (fraude, fallas, outliers) en datos sin etiquetas, eligiendo entre Isolation Forest, LOF, One-Class SVM y Elliptic Envelope según la geometría del problema, y entendiendo la diferencia entre outlier detection (entrenar con datos sucios) y novelty detection (entrenar limpio, predecir sobre nuevos).

Resultados de aprendizaje

Al finalizar la clase, el alumno podrá:

1. Distinguir outlier detection vs novelty detection y elegir el algoritmo acorde.
2. Entrenar IsolationForest y ajustar el hiperparámetro contamination.
3. Usar LocalOutlierFactor en modo novelty=False (fit_predict) y novelty=True (predict).
4. Aplicar OneClassSVM y EllipticEnvelope, reconociendo sus supuestos (kernel, gaussianidad).
5. Evaluar detectores de anomalías con score_samples, ROC-AUC y reglas de negocio (top-k).

Temas

#	Tema	Por qué importa
1	Outlier vs novelty detection	Define cómo se entrena y qué se predice.
2	Isolation Forest	Default sólido en alta dimensión, escala b
3	Local Outlier Factor (LOF)	Detecta anomalías locales por densidad.
4	One-Class SVM	Frontera no lineal con kernel RBF; sensibl
5	Elliptic Envelope	Asume distribución gaussiana; útil en dato
6	score_samples y umbrales	Salida continua > etiqueta binaria.
7	Evaluación sin labels	Top-k, inspección manual, ROC si hay groun

Definiciones y características

Anomaly / outlier detection

: Tarea no supervisada de identificar instancias que difieren significativamente de la mayoría. Entrena sobre un dataset contaminado (con algunas anomalías) y las marca dentro del mismo dataset. Salida sklearn: +1 (inlier) / -1 (outlier).

Novelty detection

: Variante donde el entrenamiento se hace sobre datos limpios (solo inliers), y luego en inferencia se predice si nuevos puntos son normales o novedosos. LocalOutlierFactor(novelty=True) y OneClassSVM operan en

este modo.

Isolation Forest

: Ensemble de árboles que aíslan puntos eligiendo features y splits al azar. Las anomalías quedan aisladas con menos splits (path corto). Escala bien a alta dimensión y N grande. Default recomendado.

contamination

: Hiperparámetro que indica la fracción esperada de anomalías en los datos (entre 0 y 0.5, o 'auto'). Define el umbral del score. Si te equivocás mucho, calibrá con score_samples y elegí el umbral a mano.

Local Outlier Factor (LOF)

: Compara la densidad local de un punto con la de sus k vecinos. Si la densidad es mucho menor que la de sus vecinos, es outlier. Bueno para anomalías locales (un punto raro dentro de un cluster). No escala bien a N muy grande.

One-Class SVM

: Aprende una frontera (típicamente con kernel RBF) que envuelve la región "normal" del espacio. Sensible a escala (estandarizar siempre) y al hiperparámetro nu (cota superior de la fracción de outliers). Lento en N grande; preferir SGDOneClassSVM para datasets grandes.

Elliptic Envelope

: Ajusta una gaussiana robusta a los datos y marca como outliers los puntos fuera de un elipsoide de confianza. Asume distribución unimodal aproximadamente gaussiana — si los datos son multimodales o tienen estructura compleja, falla.

score_samples(X)

: Devuelve un score continuo de "normalidad" por instancia (más alto = más normal). Útil para rankear top-k anomalías o elegir el umbral a mano en lugar de confiar en contamination.

Dataset / recursos

Sintético con make_blobs + outliers uniformes inyectados, o sklearn.datasets.fetch_kddcup99 (detección de intrusiones, real y con labels para evaluar).

Ejercicios

1. Isolation Forest baseline. Generá 2 blobs + 5% de outliers uniformes. Ajustá IsolationForest(contamination=0.05). Plot 2D con inliers vs outliers detectados.

2. LOF local. Usá el mismo dataset pero metiendo un outlier dentro de uno de los blobs (anomalía local). Comparar Isolation Forest vs LocalOutlierFactor(n_neighbors=20): ¿cuál lo agarra?

3. One-Class SVM con escalado. Entrená OneClassSVM(kernel='rbf', nu=0.05) con y sin StandardScaler. Comparar fronteras de decisión.

4. Top-k con score_samples. Usá score_samples de Isolation Forest, ordená ascendente, y devolvé los 10 puntos más anómalos. Inspeccionálos visualmente.

5. Evaluación con labels. Sobre fetch_kddcup99 (subset), entrená Isolation Forest sin usar labels. Después calculá ROC-AUC contra las labels reales (normal. vs ataque). Reportá AUC.

Homework verificable

Notebook con dataset de transacciones sintéticas (montos, hora, comercio): (a) inyectar 2% de transacciones anómalas (montos extremos, horarios raros); (b) entrenar Isolation Forest, LOF y One-Class SVM; (c) generar tabla comparativa con precision@k=20, recall y tiempo de entrenamiento; (d) elegir el ganador y justificar.

Criterio de aceptación: los tres modelos están entrenados sobre los mismos datos escalados, la tabla compara las 3 métricas, y la justificación menciona supuestos (densidad local vs global, escalabilidad).

Errores comunes

Síntoma / mensaje	Causa y cómo arreglar
LocalOutlierFactor no tiene predict	Por default novelty=False y solo expone fi
One-Class SVM marca todo como outlier (o t	Datos sin escalar — el kernel RBF es ultra
IsolationForest con contamination muy off	Si la fracción real difiere mucho del valo
Elliptic Envelope falla con datos multimod	Asume una sola gaussiana. Fix: cambiá a Is
Evaluar con accuracy sobre clases desbalan	El modelo "todo normal" da 99% accuracy y

Preguntas frecuentes

¿Isolation Forest, LOF o One-Class SVM?

Isolation Forest primero, casi siempre: escala bien, pocos hiperparámetros, robusto en alta dimensión. LOF si sospechás anomalías locales (raros dentro de un cluster denso). One-Class SVM si el dataset es chico/mediano y la frontera "normal" es claramente no lineal — pero estandarizá sí o sí. Elliptic Envelope solo si los datos son unimodales y aproximadamente gaussianos.

¿Cómo elijo contamination?

Si conocés la prevalencia esperada (ej. 1% de fraude), poné ese valor. Si no, dejá 'auto' y después calibrá mirando la distribución de score_samples — buscá un quiebre natural en el histograma.

¿Necesito escalar las features?

Para One-Class SVM y LOF, sí (ambos usan distancias). Para Isolation Forest, no es estrictamente necesario porque parte features con splits — pero no hace daño.

¿Sirve para series de tiempo?

No directamente — estos modelos asumen i.i.d. Para series, mirá descomposición + residuos, ARIMA, Prophet, o modelos específicos como PyOD con ventanas deslizantes.

¿Cómo evalúo si no tengo labels?

Inspección manual del top-k con score_samples. Si hay labels parciales o un sample anotado, ROC-AUC y precision@k. Sin nada de eso, sanity check: ¿los outliers detectados tienen sentido para el negocio?

Referencias

- Géron, cap. 9 § Anomaly Detection.
- sklearn — Novelty and Outlier Detection

- sklearn — IsolationForest
- sklearn — LocalOutlierFactor
- Liu, Ting & Zhou (2008), Isolation Forest, ICDM.

Siguiente clase

Clase 100 — Perceptrón, MLP y backpropagation

Apéndice: notebook (primer bloque)

Primera celda ejecutable del notebook de la clase.

```
# Imports y configuración inicial
```

Archivos complementarios

- notebook.ipynb