Clase 089 — XGBoost, LightGBM y CatBoost

Parte: 1 — Machine Learning Clásico · Fuente: docs XGBoost/LightGBM/CatBoost + Géron, cap. 7. Duración estimada: 80 min.

Clase 089 — XGBoost, LightGBM y CatBoost

> *Parte: 1 — Machine Learning Clásico · Fuente: docs XGBoost/LightGBM/CatBoost + Géron, cap. 7. Duración estimada: 80 min.*

Objetivo

Conocer las tres librerías de gradient boosting moderno que dominan tabular ML —XGBoost, LightGBM y CatBoost—, entender sus diferencias algorítmicas (level-wise vs leaf-wise, manejo de categóricas, regularización) y saber elegir la adecuada según el problema, usando sus APIs sklearn-compatibles con early_stopping_rounds.

Resultados de aprendizaje

Al terminar la clase, vas a poder:

1. Instalar y usar xgboost, lightgbm y catboost con la API sklearn (fit/predict/predict_proba).
2. Explicar la diferencia entre crecimiento level-wise (XGBoost por defecto) y leaf-wise (LightGBM) y sus implicancias en velocidad y overfitting.
3. Configurar early stopping con eval_set y early_stopping_rounds para evitar overfitting y ahorrar tiempo.
4. Manejar features categóricas: encoding manual para XGBoost, categorical_feature en LightGBM, cat_features nativo en CatBoost con ordered boosting.
5. Elegir la librería apropiada según tamaño de dataset, presencia de categóricas y necesidad de velocidad de entrenamiento o inferencia.

Temas

1. Repaso: gradient boosting como ensemble secuencial (de clase 078).
2. XGBoost: histogram-based, level-wise, regularización L1+L2, sparse-aware.
3. LightGBM: leaf-wise growth, GOSS (Gradient-based One-Side Sampling), EFB (Exclusive Feature Bundling).
4. CatBoost: ordered boosting, manejo nativo de categóricas, symmetric trees.
5. Tabla comparativa de los 3.
6. Instalación: pip install xgboost lightgbm catboost.
7. API sklearn unificada: XGBClassifier, LGBMClassifier, CatBoostClassifier.
8. early_stopping_rounds con eval_set.
9. Categorical features en cada uno.
10. Cuándo cada uno: heurísticas prácticas.

Tabla comparativa

Aspecto	XGBoost	LightGBM	CatBoost
Crecimiento del árbol	Level-wise (default)	Leaf-wise	Symmetric (oblivious)
Velocidad entrenamiento	Media	Muy rápida	Media
Velocidad inferencia	Rápida	Rápida	Muy rápida

Categóricas nativas	(requiere encoding)	(índices, ordinal)	(ordered boosting)
Manejo de NaN			
Overfitting en datasets chicos	Bajo	Más riesgo (leaf-wise)	Bajo
Hiperparámetro clave	max_depth	num_leaves	depth
GPU			

Definiciones y características

- XGBoost (eXtreme Gradient Boosting): implementación optimizada de gradient boosting con regularización L1/L2 explícita, manejo de sparsity y construcción de árboles level-wise (todos los nodos de un nivel antes de bajar). Paper de Chen & Guestrin (2016).
- LightGBM: librería de Microsoft (2017) que crece árboles leaf-wise (expande la hoja con mayor pérdida), histogram-based, mucho más rápida en datasets grandes. Más propensa a overfit en datos chicos.
- CatBoost: librería de Yandex (2017) optimizada para features categóricas; usa ordered boosting para evitar target leakage y symmetric trees (todos los splits del mismo nivel usan la misma condición) que aceleran inferencia.
- Level-wise growth: estrategia de XGBoost; expande todos los nodos de un nivel antes de pasar al siguiente. Árbol balanceado, menos overfitting, más lento.
- Leaf-wise growth: estrategia de LightGBM; expande siempre la hoja con mayor reducción de pérdida. Árbol desbalanceado, más rápido, mayor riesgo de overfit si no se controla num_leaves.
- Histogram-based splitting: bucketea features continuas en max_bin histogramas (típicamente 255) para evaluar splits en O(bins) en lugar de O(n). Usado por los tres.
- GOSS (Gradient-based One-Side Sampling): técnica de LightGBM que retiene todas las instancias con gradiente grande y submuestrea aleatoriamente las de gradiente chico, acelerando sin perder precisión.
- EFB (Exclusive Feature Bundling): en LightGBM, agrupa features sparse mutuamente excluyentes en una sola, reduciendo dimensionalidad efectiva.
- Ordered boosting: técnica de CatBoost; para cada muestra usa un modelo entrenado sin esa muestra para calcular su residual, evitando el target leakage que ocurre al encodear categóricas con target mean usando la misma muestra.
- Target leakage en categóricas: cuando se codifica una categoría usando el target de las mismas filas que se entrenan, inflando artificialmente la performance en train; CatBoost lo evita con ordered TS (target statistics).

Dataset / recursos

- sklearn.datasets.fetch_openml('adult') o 'credit-g' — datasets con mezcla de numéricas y categóricas, ideales para comparar las 3 librerías.
- Alternativa: cualquier dataset tabular con >10k filas y columnas categóricas.
- Notebook: notebook.ipynb (no editar — se entrega).

Ejercicios

1. Instalación y smoke test: instalar las 3 librerías, importarlas e imprimir versiones. Confirmar que cargan sin error.
2. XGBoost básico: entrenar XGBClassifier sobre adult (con OneHotEncoder para categóricas), usar eval_set y early_stopping_rounds=20, reportar accuracy en test y la mejor iteración.

3. LightGBM con leaf-wise: entrenar LGBMClassifier pasando categorical_feature con los índices de columnas categóricas (encoding ordinal previo). Comparar tiempo de entrenamiento vs XGBoost.
4. CatBoost nativo: entrenar CatBoostClassifier pasando cat_features con los nombres de columnas — sin encoding manual. Verificar que la accuracy se mantiene o mejora respecto a 2 y 3.
5. Comparativa final: entrenar los 3 modelos sobre el mismo dataset con los mismos splits, reportar en una tabla: accuracy test, tiempo de fit, tiempo de predict y mejor iteración. Concluir cuál elegirías.

Homework verificable

Construí un script boosting_showdown.py que reciba --dataset <nombre openml> y entrene los tres modelos (XGBoost, LightGBM, CatBoost) con early_stopping_rounds=50, los mismos train/test_split(random_state=42), y emita un CSV resultados.csv con columnas: modelo, accuracy_test, tiempo_fit_seg, tiempo_predict_seg, mejor_iter.

Criterio de aprobado:

- Los 3 modelos entrenan sin error sobre al menos un dataset con categóricas.
- CatBoost se entrena sin OneHotEncoder previo (usa cat_features).
- El CSV contiene las 3 filas con valores no nulos.
- En el README del homework, una conclusión de 3 líneas justificando cuál elegirías.

Errores comunes

1. Olvidar early_stopping_rounds con eval_set: si pasás eval_set pero no early_stopping_rounds, entrenás las n_estimators completas sin parar — desperdiciás tiempo y podés overfittear.
2. Usar max_depth alto en LightGBM: como crece leaf-wise, num_leaves es el parámetro de control real; max_depth=-1 (sin límite) con num_leaves alto explota en overfit.
3. Pasar strings a XGBoost: XGBoost no maneja categóricas como strings por default (sí desde 1.5 con enable_categorical=True, pero limitado); olvidarse y obtener ValueError: could not convert string to float.
4. OneHotEncodear para CatBoost: anula su ventaja principal —el ordered boosting—; pasale las categóricas crudas con cat_features.
5. Comparar con n_estimators distintos: si XGBoost para en 200 y LightGBM en 800 por early stopping, comparar tiempos absolutos sin normalizar es injusto; reportá también iteraciones.

Preguntas frecuentes

1. ¿XGBoost, LightGBM o CatBoost?

- LightGBM si tenés dataset grande (>100k filas) y querés velocidad.
- CatBoost si tenés muchas categóricas de alta cardinalidad.
- XGBoost si necesitás máxima madurez del ecosistema, integración con Spark/Dask, o ya tenés pipeline montado.

1. ¿Cuál gana competencias de Kaggle? Históricamente XGBoost, hoy mezclado: LightGBM y CatBoost ganan terreno. Stacking de los tres suele rendir mejor que cualquiera solo.

1. ¿Son sklearn-compatibles? Sí — todos exponen XGBClassifier/LGBMClassifier/CatBoostClassifier con fit/predict/predict_proba/score, usables en Pipeline, GridSearchCV, cross_val_score.

1. ¿Soportan GPU? Sí los tres. XGBoost con tree_method='gpu_hist', LightGBM con device='gpu', CatBoost con task_type='GPU'. Útil con datasets grandes.

1. ¿Cómo elijo num_leaves en LightGBM? Regla práctica: num_leaves < 2^max_depth. Empezá con 31 (default) y subí gradualmente monitoreando val loss.

Referencias

- XGBoost docs — <https://xgboost.readthedocs.io/>
- LightGBM docs — <https://lightgbm.readthedocs.io/>
- CatBoost docs — <https://catboost.ai/docs/>
- Chen, T. & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. KDD. <https://arxiv.org/abs/1603.02754>
- Ke, G. et al. (2017). LightGBM: A Highly Efficient Gradient Boosting Decision Tree. NeurIPS.
- Prokhorenkova, L. et al. (2018). CatBoost: unbiased boosting with categorical features. NeurIPS. <https://arxiv.org/abs/1706.09516>
- Géron, A. Hands-On Machine Learning, cap. 7 (Ensemble Learning — mención a XGBoost).

Siguiente clase

Clase 090 — Stacking (stacked generalization)

Apéndice: notebook (primer bloque)

Primera celda ejecutable del notebook de la clase.

```
# Imports y configuración inicial
```

Archivos complementarios

- notebook.ipynb