Clase 068 — Análisis de errores

Parte: 1 — Machine Learning Clásico · Fuente: Géron, cap. 3 § Error Analysis. Duración estimada: 60 min.

Clase 068 — Análisis de errores

> *Parte: 1 — Machine Learning Clásico · Fuente: Géron, cap. 3 § Error Analysis. Duración estimada: 60 min.*

Objetivo

Que el alumno deje de mirar el accuracy global y empiece a auditar dónde se equivoca un clasificador: confusion matrix normalizada por fila, pares de clases confundidas, inspección visual de ejemplos mal clasificados, y el loop de error analysis como puerta de entrada al data-centric AI (mejorar datos, no solo modelos).

Resultados de aprendizaje

Al finalizar la clase, el alumno podrá:

1. Construir y normalizar una confusion matrix por fila (recall por clase) y leerla sin confundir filas con columnas.
2. Identificar pares de clases confundidas ordenando los off-diagonals normalizados de mayor a menor.
3. Inspeccionar visualmente ejemplos mal clasificados (hard examples) para formular hipótesis de causa raíz.
4. Decidir si la próxima iteración mejora el modelo (features, regularización, capacidad) o mejora los datos (relabel, augmentación, balancear).
5. Ejecutar el error analysis loop: entrenar → matriz → slices → hipótesis → fix → re-entrenar.

Temas

#	Tema	Por qué importa
1	Confusion matrix cruda vs normalizada por	Sin normalizar, las clases mayoritarias ta
2	Off-diagonals: qué clase se confunde con c	El error no es uniforme; suele haber 2-3 p
3	Inspección visual de hard examples	Te dice si es ruido de label, ambigüedad r
4	Slice analysis (error por subgrupo)	Accuracy global oculta sesgos por subpobla
5	Data-centric AI: cuándo arreglar datos	Más barato y efectivo que tunear hiperpará
6	Error analysis loop	Workflow iterativo y reproducible.

Definiciones y características

Confusion matrix normalizada por fila

: Matriz donde C[i,j] = P(predicho=j | real=i). La diagonal es el recall por clase; los off-diagonals son la distribución de errores condicional al label real. Siempre dividí por la suma de fila, no por el total: lo que querés ver es "del total de los i verdaderos, qué fracción cayó en j".

Error rate por clase

: 1 - recall_i. Útil para rankear clases por dificultad. Una clase con 60% recall en un modelo de 95% accuracy global es un problema invisible al accuracy.

Hard examples

: Instancias mal clasificadas con alta confianza, o cerca del borde de decisión. Inspeccionarlas a mano (50-100 alcanza) revela el 80% de las causas raíz: labels equivocados, imágenes borrosas, ambigüedad ontológica, dominio fuera de distribución.

Slice analysis

: Computar métricas no en el set completo sino en subconjuntos definidos por una variable (edad, región, tipo de cámara, longitud del texto). Detecta sesgos que el promedio entierra.

Data-centric AI

: Paradigma (Andrew Ng) que pone el foco en mejorar la calidad y consistencia de los datos en vez de iterar modelos. Mucho del error analysis es la herramienta operacional del data-centric.

Error analysis loop

: Ciclo train → confusion matrix → top-k confusiones → inspeccionar ejemplos → hipótesis (modelo/dato/feature) → intervenir → repetir. Convierte el debugging de ML de arte a proceso.

Top-k confusiones

: Los k pares (i, j) con i != j ordenados por C_norm[i,j] descendente. Son la lista priorizada de qué atacar primero.

Dataset / recursos

MNIST (sklearn.datasets.fetch_openml('mnist_784') o keras.datasets.mnist). Es el ejemplo canónico de Géron cap. 3: 10 clases, errores no uniformes (los 4↔9, 3↔5, 7↔9 son los pares clásicos que aparecen en cualquier modelo decente). Alternativa más densa: Fashion-MNIST (shirt vs t-shirt vs pullover es un caos pedagógicamente útil).

Ejercicios

1. Matriz cruda y normalizada. Entrená un SGDClassifier sobre MNIST. Calculá confusion_matrix(y_true, y_pred) y normalizá por fila (cm / cm.sum(axis=1, keepdims=True)). Plotealá con plt.matshow y poné ceros en la diagonal para que los errores se vean.

2. Top-5 confusiones. Del array normalizado, extraé los 5 pares (i, j) con mayor valor off-diagonal. Imprimí "real=i → pred=j: XX%".

3. Galería de errores. Para el par (real, pred) peor del ejercicio 2, mostrá una grilla 5×5 de imágenes mal clasificadas. Anotá si te parecen ambiguas, mal labeladas o claramente del label real.

4. Slice por grosor de trazo. Calculá la suma de píxeles por imagen como proxy de "grosor". Dividí el test set en terciles y reportá accuracy por tercil. ¿El modelo es peor con dígitos finos o gruesos?

5. Intervención data-centric. Tomá el par confundido del ejercicio 2. Augmentá el set de entrenamiento solo con esa clase (shifts de 1px) y re-entrená. Reportá el cambio en el recall de esa clase y el accuracy global.

Homework verificable

Notebook con MNIST que entregue: (a) confusion matrix normalizada por fila como heatmap; (b) tabla con los 3 pares de clases más confundidos y su porcentaje; (c) grilla de 16 ejemplos mal clasificados del par peor,

con título real=X pred=Y; (d) tabla de accuracy por slice según una variable derivada (intensidad media, posición del centro de masa, lo que elijas); (e) un párrafo de hipótesis: ¿el error es de modelo o de datos? ¿qué probarías en la siguiente iteración?

Criterio de aceptación: Las filas de la matriz normalizada suman 1. El top-3 de confusiones está justificado numéricamente. La galería muestra ejemplos reales del par identificado. La hipótesis distingue explícitamente "mejoro modelo" vs "mejoro datos".

Errores comunes

Síntoma / mensaje	Causa y cómo arreglar
La matriz "se ve toda igual" / no se disti	No normalizaste, o no pusiste ceros en la
Normalicé por columna y los números no cie	Normalización por columna te da precision
Hago error analysis sobre el train set	Te miente: el modelo memorizó. Fix: siempr
"Tuneo hiperparámetros 3 días y no mejora"	Probablemente el techo es la calidad del l
Reporto solo accuracy global	Esconde clases minoritarias y subgrupos. F

Preguntas frecuentes

¿Cuándo mejoro el modelo y cuándo mejoro los datos?

Heurística operativa: si los ejemplos mal clasificados te confunden a vos también (ambiguos, mal labelados, fuera de dominio) → datos (relabel, limpiar, augmentar, recolectar más de esa clase). Si los errores son obvios para un humano pero el modelo los falla sistemáticamente → modelo (más capacidad, mejores features, menos regularización). En la práctica, el 70% de las veces son datos; por eso Ng popularizó el data-centric.

¿Por fila o por columna se normaliza?

Por fila (axis=1) para análisis de errores estándar: te da P(pred | real) = distribución de errores condicional a la verdad. Por columna sirve para diagnosticar precision (cuando el modelo dice j, ¿qué tan seguido es realmente j?). Géron usa por fila en cap. 3.

¿Cuántos ejemplos mal clasificados hay que mirar?

50 a 100 suele alcanzar para ver patrones. Más allá tenés rendimientos decrecientes. Lo importante es que estén estratificados por el par de confusión que te interesa, no muestreados al azar.

¿cross_val_predict o un split fijo?

cross_val_predict te da predicciones out-of-fold para todo el train set sin leakage, ideal para análisis de errores con N moderado. Para producción usá un test set holdout fijo.

¿Esto reemplaza el ROC/PR curve?

No. Las curvas ROC/PR son para threshold tuning y comparación de modelos a nivel agregado. El error analysis es para debugging cualitativo y priorización. Son complementarias.

Referencias

- Géron, Hands-On ML, cap. 3 § "Error Analysis".

- Andrew Ng — A Chat with Andrew on MLOps: From Model-centric to Data-centric AI.
- scikit-learn — confusion_matrix y ConfusionMatrixDisplay.
- scikit-learn — cross_val_predict.

Siguiente clase

Clase 069 — Regresión lineal: ecuación normal vs gradient descent

Apéndice: notebook (primer bloque)

Primera celda ejecutable del notebook de la clase.

```
# Imports y configuración inicial
```

Archivos complementarios

- notebook.ipynb