
Clase 090 — Stacking (stacked generalization)

Parte: 1 — Machine Learning Clásico · Fuente: Géron, cap. 7. Duración estimada: 60 min.

Clase 090 — Stacking (stacked generalization)

Parte: 1 — Machine Learning Clásico · Fuente: Géron, cap. 7. Duración estimada: 60 min.

Objetivo

Que el alumno combine modelos heterogéneos vía stacking: entrenar varios modelos base, generar predicciones out-of-fold, y entrenar un meta-modelo (blender) sobre esas predicciones — todo con StackingClassifier / StackingRegressor de sklearn, sin leakage.

Resultados de aprendizaje

Al finalizar la clase, el alumno podrá:

1. Explicar la idea de stacking como ensamble donde un meta-modelo aprende a combinar las predicciones de los base learners.
2. Construir un StackingClassifier con varios estimadores base y un blender (típicamente regresión logística).
3. Justificar las predicciones out-of-fold (CV interna) como mecanismo para evitar que el blender vea predicciones in-sample.
4. Comparar stacking contra voting y contra un solo modelo bien tuneado, en accuracy y costo computacional.
5. Usar passthrough=True para que el blender vea también las features originales, no solo las predicciones de los base.

Temas

#	Tema	Por qué importa
1	Idea: ensamble de dos niveles	El meta-modelo aprende los errores correla
2	Base learners heterogéneos	Diversidad reduce varianza del ensamble.
3	Meta-modelo (blender)	Suele ser simple (logística, lineal) para
4	Out-of-fold predictions	Sin esto hay leakage: el blender ve scores
5	StackingClassifier / StackingRegressor	API sklearn que hace el CV interno por vos
6	passthrough	Concatenar features originales al input de
7	Costo y cuándo conviene	N entrenamientos × K folds — caro, no siem

Definiciones y características

Stacking (stacked generalization)

: Técnica de ensamble donde se entrenan M modelos base sobre el train set; sus predicciones (out-of-fold) se usan como features para entrenar un meta-modelo que aprende a combinarlas. Introducido por Wolpert (1992).

Base learners (nivel 0)

: Los modelos individuales del ensamble. Conviene que sean heterogéneos (p. ej. RandomForest + SVM +

KNN) — si todos cometen los mismos errores, stackearlos no agrega información.

Meta-modelo / blender (nivel 1)

: Modelo que toma como input las predicciones de los base learners y produce la predicción final. Suele ser simple (LogisticRegression, Ridge) porque su input ya es altamente predictivo y un modelo complejo overfitearía.

StackingClassifier / StackingRegressor

: Implementación de sklearn (sklearn.ensemble). API: StackingClassifier(estimators=[(nombre, modelo), ...], final_estimator=Logistic(), cv=5). Internamente entrena los base con CV para generar predicciones out-of-fold, después reentrena cada base sobre todo el train, y entrena el blender sobre las predicciones OOF.

Out-of-fold predictions (OOF)

: Para cada fila del train, su predicción del base learner se genera cuando esa fila estuvo en el fold de validación (no en el de entrenamiento). Esto garantiza que el blender ve predicciones generadas por un modelo que no vio esa fila — evita el leakage.

cv en stacking

: Número de folds para generar las predicciones OOF. Default 5. Más folds → menos varianza en las features del blender, pero más costo (cada base se entrena cv veces).

passthrough=True

: Concatena las features originales X al input del blender, junto con las predicciones de los base. El blender puede así aprovechar tanto los meta-features como las features crudas. Útil cuando los base learners no capturan toda la señal.

Stacking vs voting

: Voting (hard/soft) promedia o vota las predicciones de los base con pesos fijos. Stacking aprende los pesos (y combinaciones no lineales) vía el blender. Más expresivo pero más caro y más propenso a overfit si el blender es complejo.

Dataset / recursos

load_breast_cancer o make_classification(n_samples=2000) de sklearn — suficiente para mostrar el patrón sin tiempos largos de CV.

Ejercicios

1. Stacking básico. Construí un StackingClassifier con tres base learners (RandomForest, SVC con probability=True, KNN) y LogisticRegression como blender. Reportá accuracy con cross_val_score(cv=5).
2. Comparación contra base learners solos. Evaluá los tres base learners individualmente con el mismo CV. ¿El stacking supera al mejor solo? ¿Por cuánto?
3. passthrough=True. Repetí el ejercicio 1 con passthrough=True. ¿Mejora la accuracy? Pensá por qué (el blender ve features originales además de las predicciones).
4. Variando cv. Probá cv=3, cv=5, cv=10 en el stacking. Mirá accuracy y tiempo de entrenamiento. Trade-off típico.
5. Stacking vs Voting. Entrená un VotingClassifier(voting='soft') con los mismos tres base. Compará accuracy

contra stacking. Discutí costo computacional.

Homework verificable

Notebook con `make_classification(n_samples=3000, n_features=20, n_informative=10, random_state=42)`: (a) entrenar 3 base learners individuales y reportar CV-accuracy; (b) entrenar `StackingClassifier` con esos 3 + `LogisticRegression` blender; (c) repetir con `passthrough=True`; (d) entrenar `VotingClassifier(voting='soft')` con los mismos base; (e) tabla comparativa final con accuracy, std y tiempo de entrenamiento.

Criterio de aceptación: El stacking iguala o supera al mejor base learner individual. La tabla comparativa muestra los 5 modelos (3 base + stacking + voting) con accuracy media \pm std y tiempo.

Errores comunes

Síntoma / mensaje	Causa y cómo arreglar
El stacking da peor que el mejor base learner	Base learners poco diversos (todos árboles)
Blender entrenado sobre las mismas predicciones	Pasaste predicciones in-sample al blender
SVC no se puede usar en <code>StackingClassifier</code>	SVC por default no expone probas. Fix: SVC
Stacking tarda 10× más que un solo modelo	Es esperable: $M \text{ base} \times K \text{ folds} + 1 \text{ blender}$
<code>passthrough=True</code> empeora el resultado	El blender se confunde con muchas features

Preguntas frecuentes

¿Stacking vs voting — cuándo cuál?

Voting si querés algo rápido, simple y los base learners son razonablemente parejos: promedia probas (soft) o vota clases (hard) con pesos fijos. Stacking si tenés tiempo de CV y los base learners cometen errores distintos — el blender aprende a explotar esa diversidad. Stacking suele ganar ~1-3 pp de accuracy a costa de 5-10× más cómputo.

¿Qué modelo conviene como blender?

Algo simple y regularizado: `LogisticRegression` (clasificación) o `Ridge` (regresión) son la elección estándar. Modelos complejos (`RandomForest`, `XGBoost`) como blender suelen overfitear las M predicciones OOF.

¿Cuántos base learners?

3-5 está bien. Más allá, la ganancia marginal cae rápido y el costo escala lineal. Lo importante es que sean heterogéneos (familias distintas), no que sean muchos.

¿Puedo apilar stackings (multinivel)?

Técnicamente sí (Kaggle lo hace), pero la ganancia marginal vs costo es brutal y el riesgo de overfit explota. Para producción: un solo nivel de stacking es el sweet spot.

¿Sirve stacking si tengo poca data?

Mal. Con N chico, las predicciones OOF tienen mucha varianza y el blender no encuentra señal estable. Bajo ese régimen, un modelo solo bien tuneado o un `VotingClassifier` suelen ganarle.

Referencias

- Géron, cap. 7 § Stacking.
- Wolpert, D. (1992). Stacked Generalization. Neural Networks 5(2).
- sklearn StackingClassifier
- sklearn StackingRegressor
- sklearn user guide — Stacked generalization

Siguiente clase

Clase 091 — La maldición de la dimensionalidad

Apéndice: notebook (primer bloque)

Primera celda ejecutable del notebook de la clase.

```
# Imports y configuración inicial
```

Archivos complementarios

- notebook.ipynb