

ACÀMICA

TEMA DEL DÍA

Optimización de Hiperparámetros

¿Cómo obtenemos el mejor modelo?

¿Cómo decidimos cuál es el mejor modelo?

Hoy trataremos de responder estas preguntas.



Agenda

Daily

Break

Hands-on training

Cierre



Daily



Daily



Sincronizando...

Bitácora



¿Cómo te ha ido?
¿Obstáculos?
¿Cómo seguimos?

Challenge



¿Cómo te ha ido?
¿Obstáculos?
¿Cómo seguimos?

Repaso de la bitácora



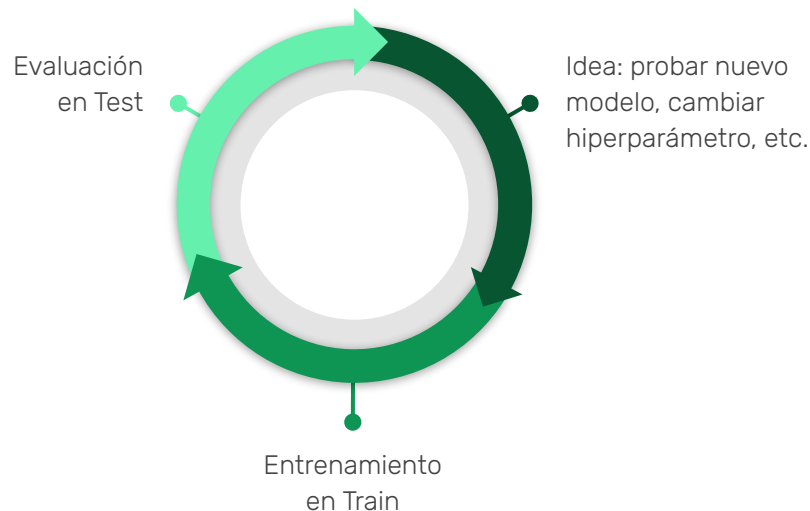
Machine Learning involucra un proceso altamente iterativo.

En general, entrenamos muchos modelos (ya sea de distinto tipo o variando hiperparámetros).

A medida que entrenamos nuevos modelos, puede ocurrir que un modelo tenga una buena performance en el conjunto de Test **por azar**.

Podemos creer que estamos seleccionando el mejor modelo disponible cuando en realidad estamos seleccionando un modelo mediocre.

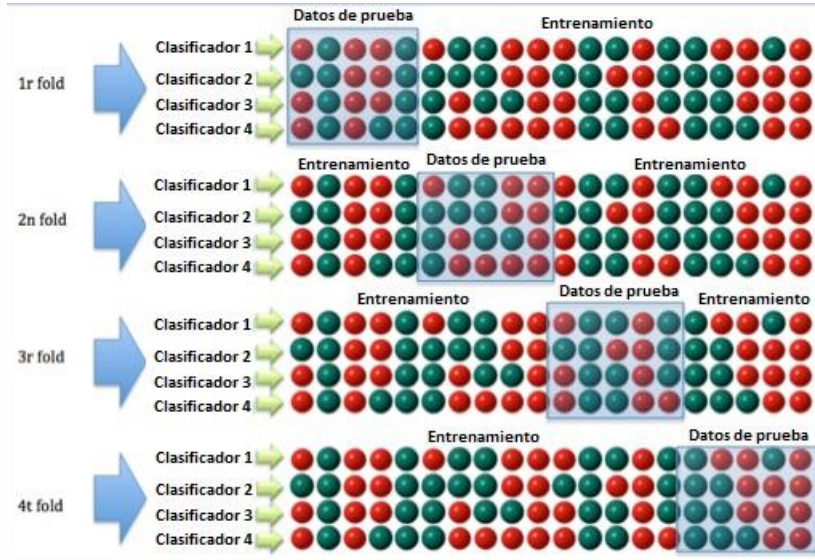
¿Se puede hacer mejor?



Comparando Modelos



Si queremos comparar muchos modelos:



Performance 1 de cada modelo

Performance 2 de cada modelo

Performance 3 de cada modelo

Performance k=4 de cada modelo

Promedio y
varianza
para cada
modelo

¿Cómo quedan entonces los conjuntos de datos?
¿Ya no tenemos que hacer train/test split?



¿Cómo quedan entonces los conjuntos de datos?
¿Ya no tenemos que hacer train/test split?



¡Tenemos que seguir haciéndolo!

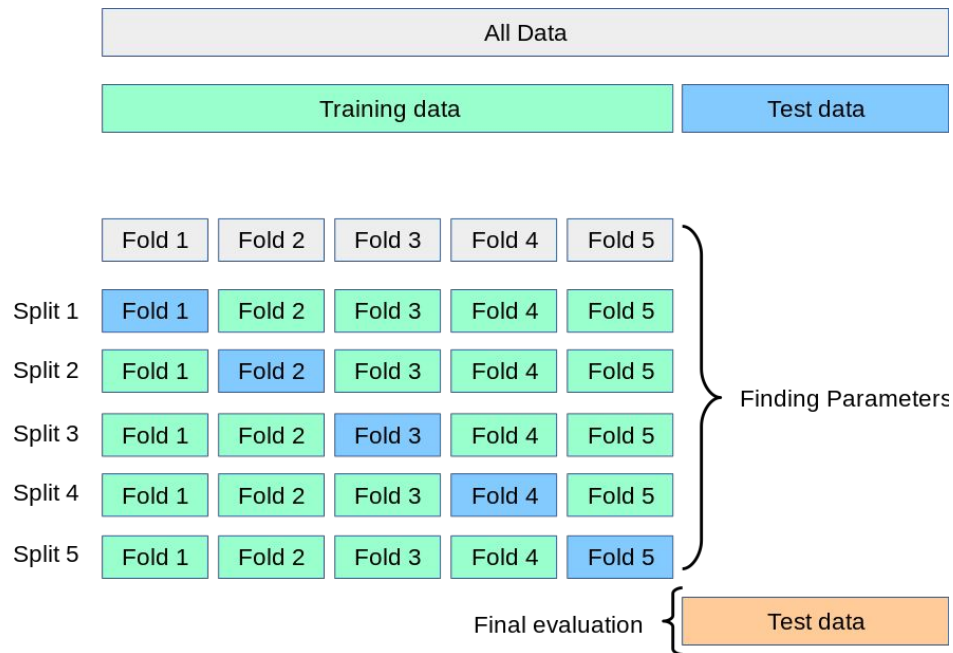
El esquema fundamental para separar nuestros datos es éste:

1. Hacemos un *train/test split* para separar dos conjuntos: uno de *desarrollo* (*dev*, también llamado *train*) y uno de *Held-Out* (también llamado a veces *test*):

Desarrollo - Experimentación con atributos, algoritmos e hiperparámetros.	Held-Out - Estimación realista de performance
--	---

2. Dentro del conjunto de desarrollo, hacemos **todas las pruebas** que consideremos necesarias y evaluamos los modelos resultantes usando **validación cruzada**. Elegimos nuestro modelo a partir del desempeño en estos datos (y las otras condiciones que consideremos para nuestro problema, como desempeño, Navaja de Ockham, etc.).
3. Evaluamos el **desempeño del modelo** elegido en el conjunto de Held-Out. Es el desempeño que vamos a reportar.

Por ejemplo, podría ser así...



Otro comentario acerca de la nomenclatura.

Lamentablemente, la comunidad no se pone de acuerdo en llamar de una única manera a cada conjunto. Algunas combinaciones típicas son train/dev/test, train/test/val, train/val/hold-out, etc.

Optimización de Hiperparámetros



¿Cómo elegimos los
mejores hiperparámetros
para nuestro problema?

¿Cómo elegimos los mejores hiperparámetros para nuestro problema?

¿Qué es mejor? ¿Con respecto a exactitud?
¿Área bajo la curva ROC?

¿Cómo elegimos los mejores hiperparámetros para nuestro problema?

¿Qué es mejor? ¿Con respecto a exactitud?
¿Área bajo la curva ROC?

Primero, tenemos que definir una métrica a optimizar.

Una vez que sabemos la métrica que queremos optimizar,
¿cómo elegimos los mejores hiperparámetros para nuestro problema?

Solución más natural: probar a mano distintos valores de los hiperparámetros.

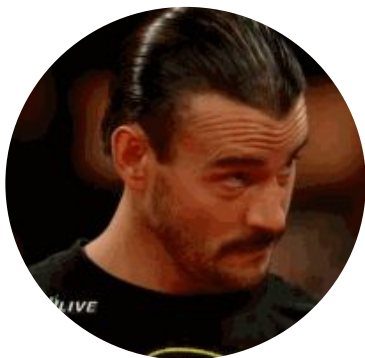
Una vez que sabemos la métrica que queremos optimizar,
¿cómo elegimos los mejores hiperparámetros para nuestro problema?

Solución más natural: probar a mano distintos valores de los hiperparámetros.



Una vez que sabemos la métrica que queremos optimizar,
¿cómo elegimos los mejores hiperparámetros para nuestro problema?

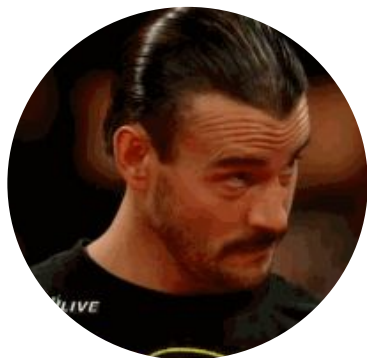
Solución más natural: ~~probar a mano distintos valores de los hiperparámetros.~~



Cansador, tedioso y poco eficiente.

Una vez que sabemos la métrica que queremos optimizar,
¿cómo elegimos los mejores hiperparámetros para nuestro problema?

Solución más natural: ~~probar a mano distintos valores de los hiperparámetros.~~



Cansador, tedioso y poco eficiente.

Pero lo usamos un poco...

Una vez que sabemos la métrica que queremos optimizar,
¿cómo elegimos los mejores hiperparámetros para nuestro problema?

Solución ~~m~~s natural: hacer una búsqueda exhaustiva. Es decir probando con todos los valores de los hiperparámetros que podamos y eligiendo la mejor combinación. Este método se llama **Grid Search** (“búsqueda de cuadrícula”).

Solución más natural: hacer una búsqueda exhaustiva. Es decir probando con todos los valores de los hiperparámetros que podamos y eligiendo la mejor combinación. Éste método se llama Grid Search (“búsqueda de cuadrícula”).

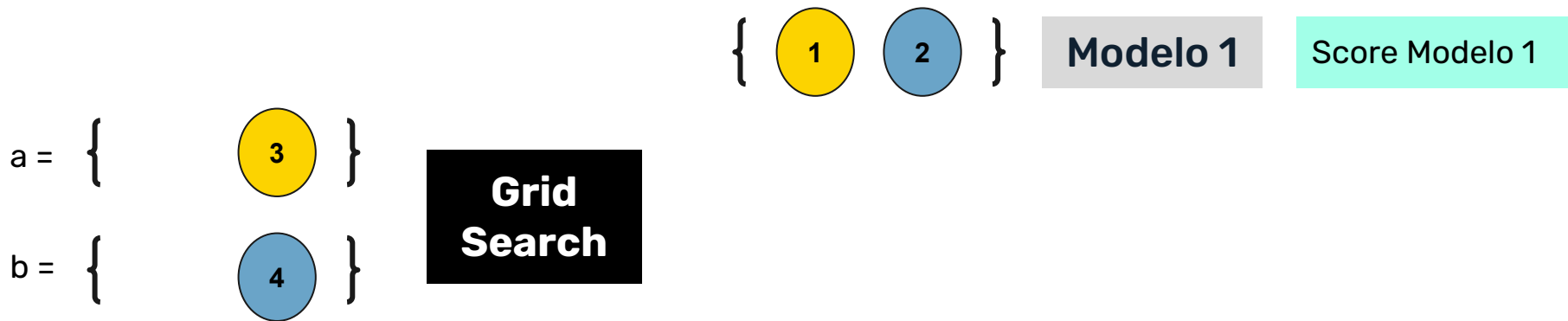
Por ejemplo, si tenemos dos hiperparámetros, a y b , que pueden tomar valores $a = \{1,2\}$ y $b = \{3,4\}$

$$a = \{ \textcircled{1} \textcircled{3} \}$$

$$b = \{ \textcircled{2} \textcircled{4} \}$$

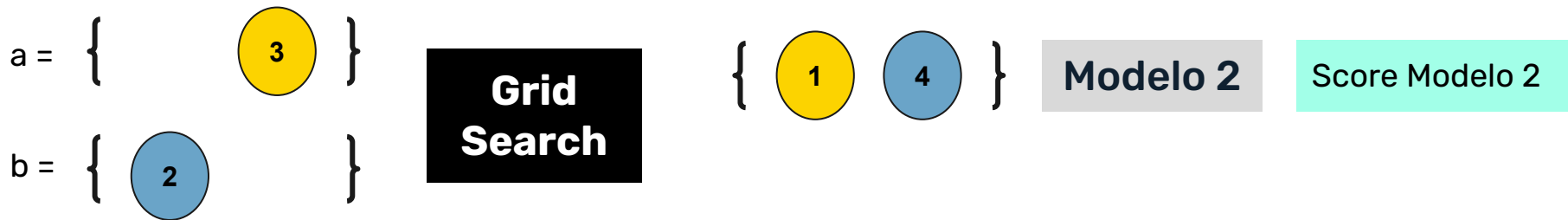
Solución más natural: hacer una búsqueda exhaustiva. Es decir probando con todos los valores de los hiperparámetros que podamos y eligiendo la mejor combinación. Éste método se llama Grid Search (“búsqueda de cuadrícula”).

Por ejemplo, si tenemos dos hiperparámetros, a y b , que pueden tomar valores $a = \{1,2\}$ y $b = \{3,4\}$



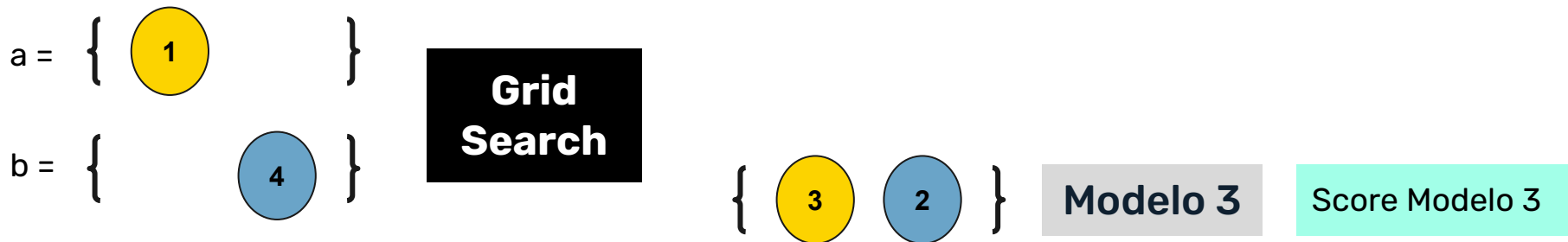
Solución más natural: hacer una búsqueda exhaustiva. Es decir probando con todos los valores de los hiperparámetros que podamos y eligiendo la mejor combinación. Éste método se llama Grid Search (“búsqueda de cuadrícula”).

Por ejemplo, si tenemos dos hiperparámetros, a y b , que pueden tomar valores $a = \{1,2\}$ y $b = \{3,4\}$



Solución más natural: hacer una búsqueda exhaustiva. Es decir probando con todos los valores de los hiperparámetros que podamos y eligiendo la mejor combinación. Éste método se llama Grid Search (“búsqueda de cuadrícula”).

Por ejemplo, si tenemos dos hiperparámetros, a y b , que pueden tomar valores $a = \{1,2\}$ y $b = \{3,4\}$



Solución más natural: hacer una búsqueda exhaustiva. Es decir probando con todos los valores de los hiperparámetros que podamos y eligiendo la mejor combinación. Éste método se llama Grid Search (“búsqueda de cuadrícula”).

Por ejemplo, si tenemos dos hiperparámetros, a y b , que pueden tomar valores $a = \{1,2\}$ y $b = \{3,4\}$

$a = \{ \text{1} \}$
 $b = \{ \text{2} \}$

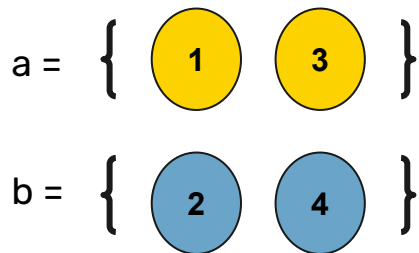
**Grid
Search**

$\{ \text{3} \text{ 4} \}$

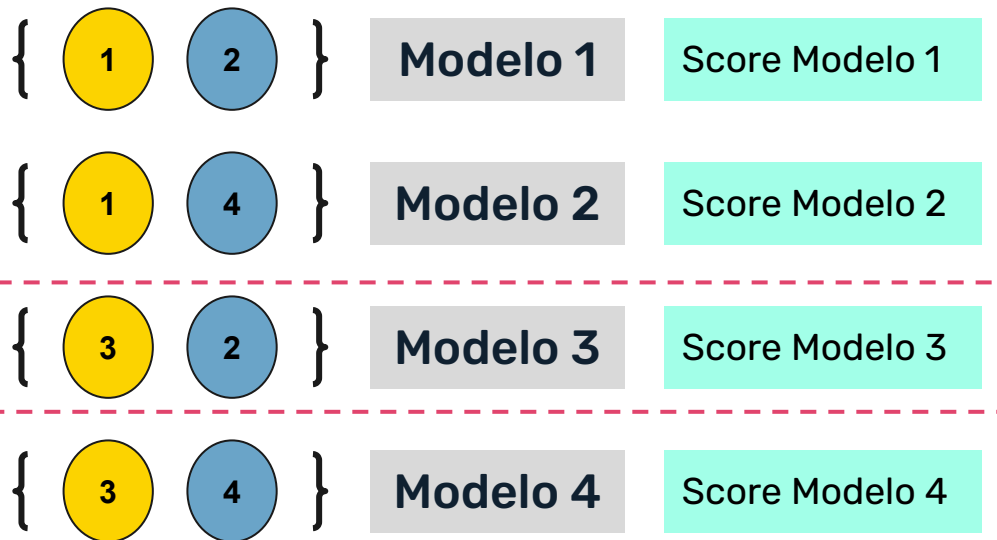
Modelo 4

Score Modelo 4

Elegimos el modelo con mejor score (o menor error si corresponde):



**Grid
Search**



Entonces, una vez que sabemos qué métrica queremos optimizar,

Grid Search consiste en:

1. Elegimos los valores que puede tomar cada hiperparámetro
2. Armamos las combinaciones “todos con todos” → Tenemos nuestra grilla
3. Recorremos la grilla, entrenamos un modelo para cada combinación y lo evaluamos.
4. Elegimos los hiperparámetros que definen el mejor modelo.

Entonces, una vez que sabemos qué métrica queremos optimizar,

Grid Search consiste en:

1. Elegimos los valores que puede tomar cada hiperparámetro
2. Armamos las combinaciones “todos con todos” → Tenemos nuestra grilla
3. Recorremos la grilla, entrenamos un modelo para cada combinación y **lo evaluamos.**
4. Elegimos los hiperparámetros que definen el mejor modelo.

¿Cómo evaluamos un modelo?



¿Con Train/Test split? ¿Con Validación Cruzada?

Al estar probando MUCHOS modelos, podría suceder que uno se desempeñe muy bien en el conjunto de Train simplemente por azar.

Por esto, es muy importante evaluar cada modelo creado por Grid Search con Validación Cruzada en el conjunto de Dev (Train).

¿Cómo evaluamos un modelo?



¿Con Train/Test split? ¿Con Validación Cruzada?

Al estar probando MUCHOS modelos, podría suceder que uno se desempeñe muy bien en el conjunto de Train simplemente por azar.

Por esto ,es muy importante evaluar cada modelo creado por Grid Search con Validación Cruzada en el conjunto de Dev (Train).

¡Grid Search y Validación Cruzada suelen venir juntos!

Entonces, una vez que elegimos los mejores hiperparámetros con Grid Search + Validación Cruzada:



1. Elegimos Entrenamos un modelo con esos hiperparámetros con todo el conjunto de Dev (Train)
2. Evaluamos su performance en el conjunto de Held-Out (Test).

Entonces, una vez que elegimos los mejores hiperparámetros con Grid Search + Validación Cruzada:



1. Elegimos Entrenamos un modelo con esos hiperparámetros con todo el conjunto de Dev (Train)
2. Evaluamos su performance en el conjunto de **Held-Out (Test)**.

Como la evaluación de performance de CV puede estar sesgada, es importante evaluar al final en este conjunto para obtener una estimación menos sesgada del desempeño.

Para pensar:
¿por qué puede estar sesgada la
evaluación de performance del
modelo con CV?



¿Qué desventajas tiene Grid Search?



¿Qué ocurre si tenemos, por ejemplo, cinco hiperparámetros y cinco valores para probar por hiperparámetro? ¿Qué tamaño tiene la grilla?

¿Qué desventajas tiene Grid Search?



¿Qué ocurre si tenemos, por ejemplo, cinco hiperparámetros y cinco valores para probar por hiperparámetro? ¿Qué tamaño tiene la grilla?

¿Y si, además, para cada modelo tenemos que hacer Validación Cruzada?

¿Qué desventajas tiene Grid Search?



Grid Search + CV puede ser computacionalmente muy demandante.

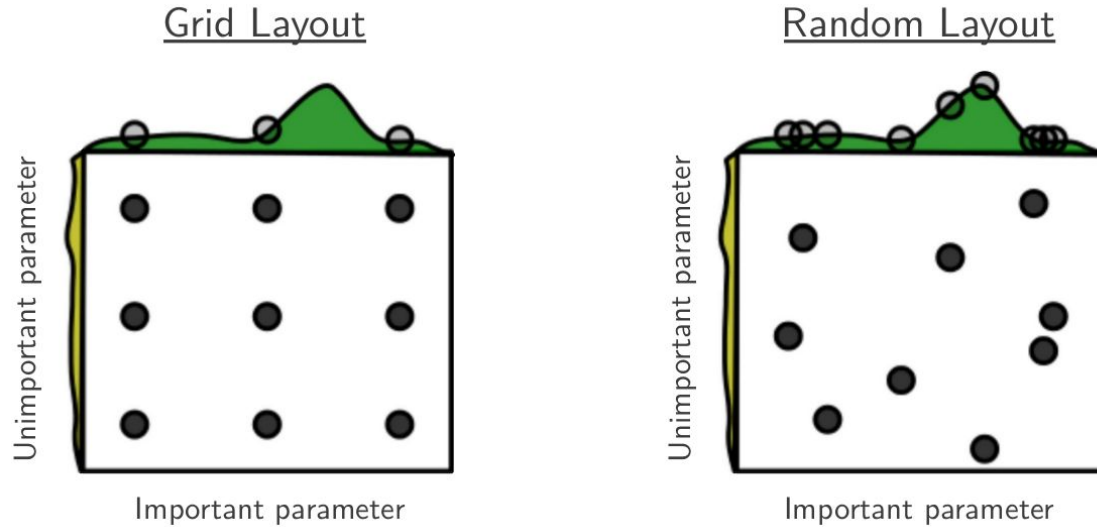
Y tal vez al “probar” con pocos valores de los hiperparámetros nos estamos perdiendo algunas cosas importantes.

Además, suele haber hiperparámetros mucho más importantes que otros.

¿Y entonces?



Random Search explora opciones y combinaciones al azar, de manera menos “ordenada”. En muchas circunstancias, ¡esto es más eficiente, tanto desde el punto de vista de performance del modelo como de desempeño computacional!



Resumen • Para optimizar hiperparámetros necesitamos:

- Una **métrica** (exactitud, precisión, RMSE, ROC AUC, etc.)
- Un **modelo** (regresor o clasificador)
- Un **espacio** de (hiper)parámetros. Depende del tipo de modelo que estemos usando.
- Un **método** para buscar o muestrear los candidatos
 - a. **Grid Search**: Plantea opciones y explora todas las combinaciones
 - b. **Random Search**: explora opciones y combinaciones al azar.

¡Una guía para ordenar el proceso!

1. Separar los datos de Held-Out de los de procesamiento.
2. Armar el Grid-Search, definiendo modelos y rangos de sus hiperparámetros para:
 - a. Preprocesamiento de features
 - b. Algoritmo de ML
3. Ejecutar el Grid-Search, evaluando con K-fold cross-validation la performance de cada modelo construido con cada una de todas las combinaciones posibles de las opciones planteadas en (2)
4. Elegir el modelo de mejor performance, y entrenarlo con todos los datos de entrenamiento.
5. Estimar la performance del modelo evaluando sobre el Held-out.
6. El modelo sale a producción.



`sklearn.model_selection.GridSearchCV`

```
class sklearn.model_selection.GridSearchCV(estimator, param_grid, scoring=None, n_jobs=None, iid='deprecated', refit=True,
cv=None, verbose=0, pre_dispatch='2*n_jobs', error_score=nan, return_train_score=False) \[source\]
```

`sklearn.model_selection.RandomizedSearchCV`

```
class sklearn.model_selection.RandomizedSearchCV(estimator, param_distributions, n_iter=10, scoring=None, n_jobs=None,
iid='deprecated', refit=True, cv=None, verbose=0, pre_dispatch='2*n_jobs', random_state=None, error_score=nan,
return_train_score=False) \[source\]
```

A close-up photograph of a white ceramic cup filled with a latte. The surface of the milk is decorated with intricate latte art, featuring a central heart shape surrounded by concentric, wavy lines. The cup sits on a matching white saucer. In the background, a white napkin and a silver fork are visible, though they are out of focus. The overall lighting is soft and even.

¡BREAK!



Hands-on training



Hands-on training

DS_Bitácora_22_Optimización.ipynb



Buenas prácticas de un data scientist





Hyperparameter
tuning

Tuning process



Recursos





Optimización de Hiperparámetros

- Como siempre, [la guía de Scikit-Learn](#) es una muy buena referencia.
- **Capítulo 5, “Machine Learning: Hyperparameters and Model Validation”, de [Python Data Science Handbook](#).** La última sección tiene un ejemplo con Grid Search.
- **Muy recomendable:** [Artículo](#) con código y ejemplos de optimización de Hiperparámetros, mostrando algunas técnicas más de las que vimos en la clase. Además, hace un breve repaso de algunos conceptos que vimos en las clases. ¡Notar qué dataset usan de ejemplo!

Para la próxima

- Termina el notebook de hoy.
- Lee la bitácora 23 y carga las dudas que tengas al Trello.

En el encuentro que viene uno/a de ustedes será seleccionado/a para mostrar cómo resolvió el challenge de la bitácora. De esta manera, ¡aprendemos todos/as de (y con) todas/as, así que vengan preparados/as.

ACÀMICA