

ACÀMICA

¡Bienvenidos/as a Data Science!



Agenda

¿Cómo anduvieron?

Repaso

Explicación: Optimización de Hiperparámetros

Hands-On

Break

Buenas Prácticas de un Data Scientist

Actividad: Lanzamiento Entrega 04

Cierre



¿Dónde estamos?



¿Cómo anduvieron?



Repaso



Con un/a compañero/a, respondemos las siguientes preguntas:

1 - ¿Qué es Validación Cruzada? ¿Cómo se conoce al método de Validación Cruzada más popular?

2 - ¿Cuál es la diferencia entre un parámetro y un hiperparámetro?

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?

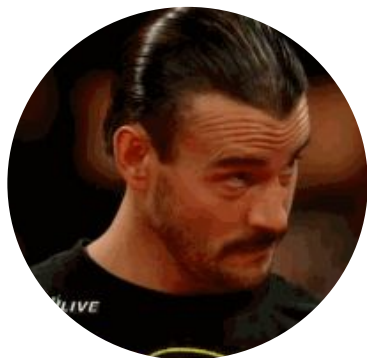
Primero, tenemos que definir una métrica a optimizar.

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.

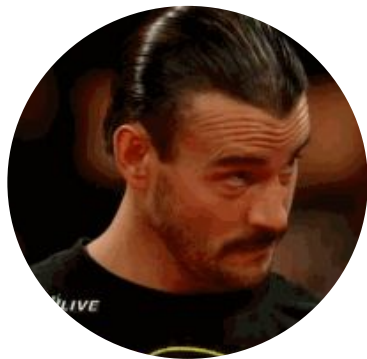
Una vez que sabemos la métrica que queremos optimizar,
¿cómo elegimos los mejores hiperparámetros para nuestro problema?

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 ~~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”).

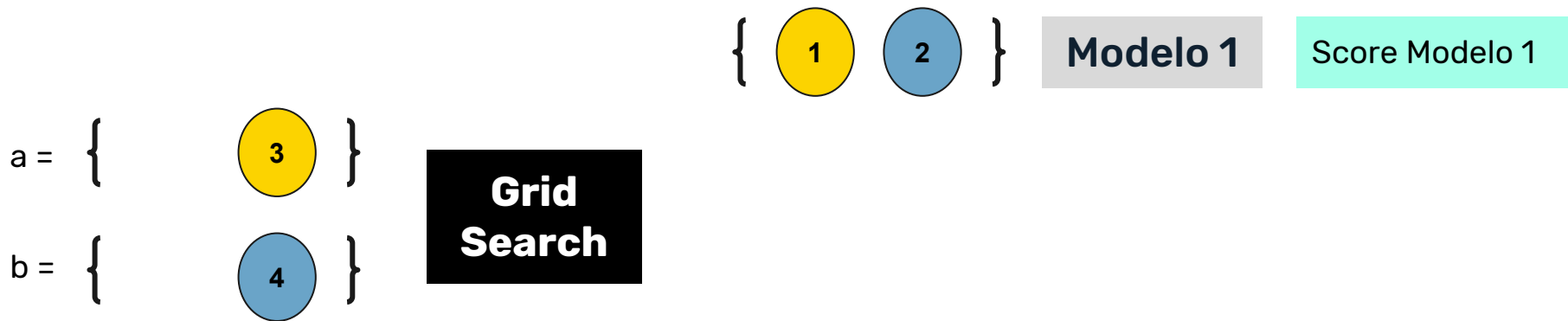
Por ejemplo, si tenemos dos hiperparámetros, a y b, que pueden tomar valores $a = \{1,2\}$ y $b = \{3,4\}$

$$a = \{ \text{1} \quad \text{3} \}$$

$$b = \{ \text{2} \quad \text{4} \}$$

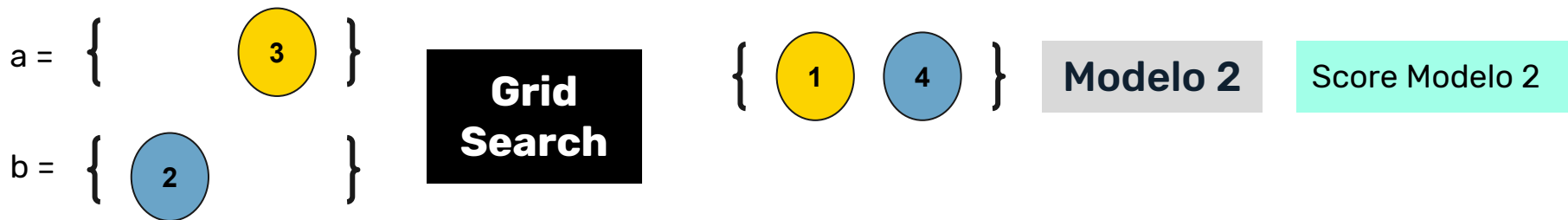
Solución más natural: hacer una búsqueda exhaustiva. Es decir probando con todos los valores de los hiperparámetros que podamos y eligiendo la mejor combinación. Éste método se llama Grid Search (“búsqueda de cuadrícula”).

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



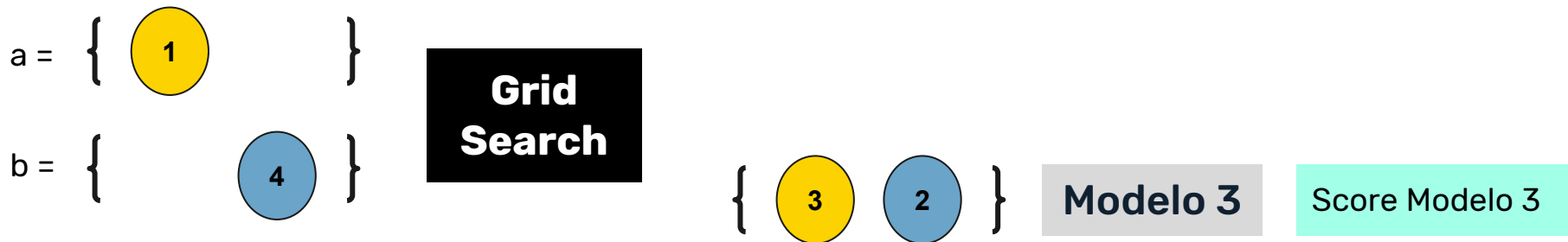
Solución más natural: hacer una búsqueda exhaustiva. Es decir probando con todos los valores de los hiperparámetros que podamos y eligiendo la mejor combinación. Éste método se llama Grid Search (“búsqueda de cuadrícula”).

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 $b = \{3,4\}$

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

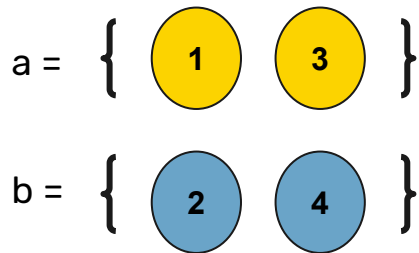
**Grid
Search**

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

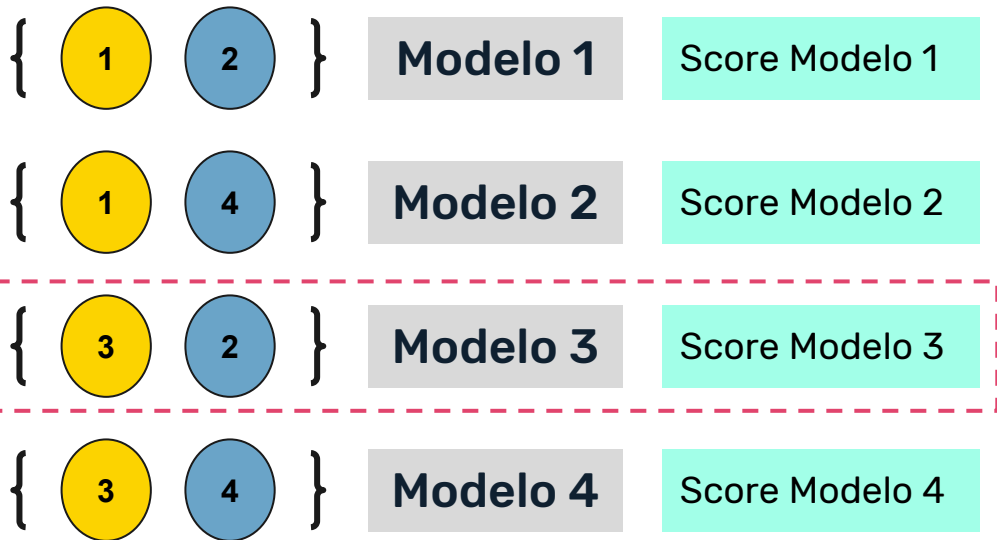
Modelo 4

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:

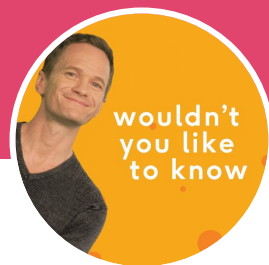
1. Elegimos los valores que puede tomar cada hiperparámetro
2. Armamos las combinaciones “todos con todos” → Tenemos nuestra grilla
3. Recorremos la grilla, entrenamos un modelo para cada combinación y lo evaluamos.
4. Elegimos los hiperparámetros que definen el mejor modelo.

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?

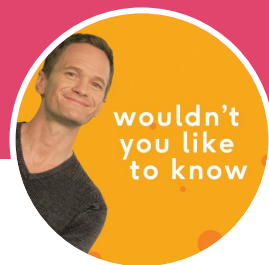


¿Con Train/Test split? ¿Con Validación Cruzada?

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 X_test.

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



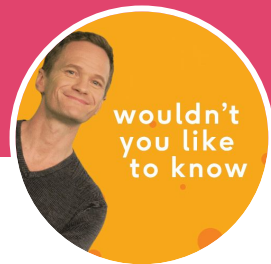
1. Elegimos Entrenamos un modelo con esos hiperparámetros con todo el conjunto de Dev (Train)
2. Evaluamos su performance en el conjunto de **X_test**.

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.

Pausa

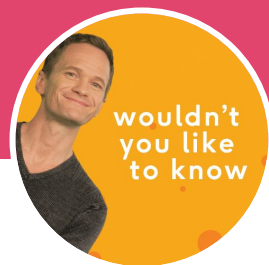


¿Qué desventajas tiene Grid Search?



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

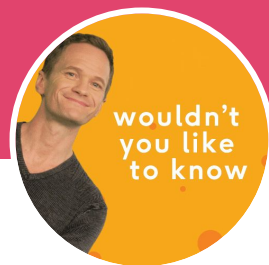
¿Qué desventajas tiene Grid Search?



¿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.

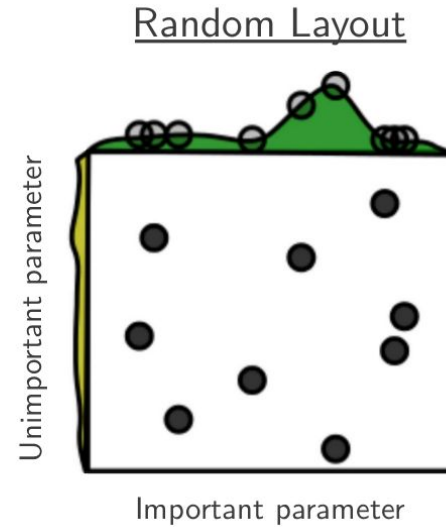
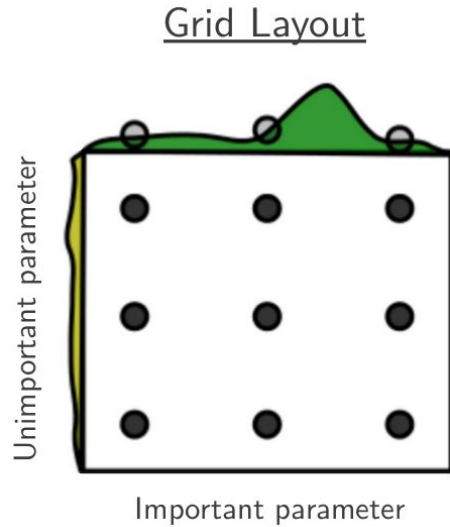
Y tal vez al “probar” con pocos valores de los hiperparámetros nos estamos perdiendo algunas cosas importantes.

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.

Guia para ordenar ideas.

- 1) Separar los datos de X_test 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 X_test .
- 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)
```

[\[source\]](#)

Hands-on training



Hands-on training

DS_Encuentro_23_Optimizacion.ipynb



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 is placed on a matching white saucer. In the background, a white napkin and a silver spoon are visible, though they are out of focus. The overall lighting is soft and even, highlighting the textures of the coffee and the smooth surface of the cup.

¡BREAK!



Buenas prácticas de un data scientist





Hyperparameter
tuning

Tuning process



Recomendaciones

Elección de la cantidad de hiperparámetros

Naturalmente, existe un tradeoff a la hora de elegir la cantidad de hiperparámetros. Es bueno tener en cuenta el uso de random search, a pesar de que podría ser *menos acertado* en general.

También será necesario tener en cuenta dónde corre (¿sobre qué hardware?)

Recomendaciones

Eligiendo rangos

Esto es especialmente importante si el hiperparámetro tiene un rango muy grande. ¿Queremos probar un rango entre 1 y 10000 con saltos de 1 en 1?

Quizá podamos aplicar saltos mayores y probar nuevamente en el rango que mejor se comporta.

Recomendaciones

Escala logarítmica

Cuando el orden de magnitud es más importante que el valor absoluto.

Por ejemplo: se espera que de 1 a 2 haya un mayor impacto que de 10^1 a 10^2 . Debido a que en el primer caso, un hiperparámetro duplica al otro, y no en el segundo caso.

¿Siempre hay que hacerlo a mano?

Optimización algorítmica

Para investigar:

- Optimización bayesiana
- Algoritmos genéticos
- Y más...



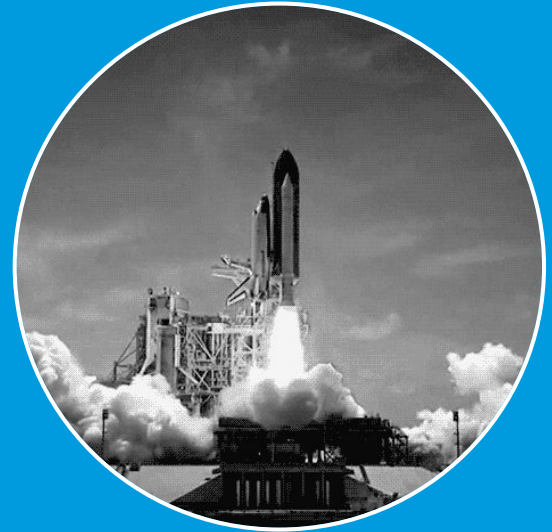
Optimización de Hiperparámetros

- 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!

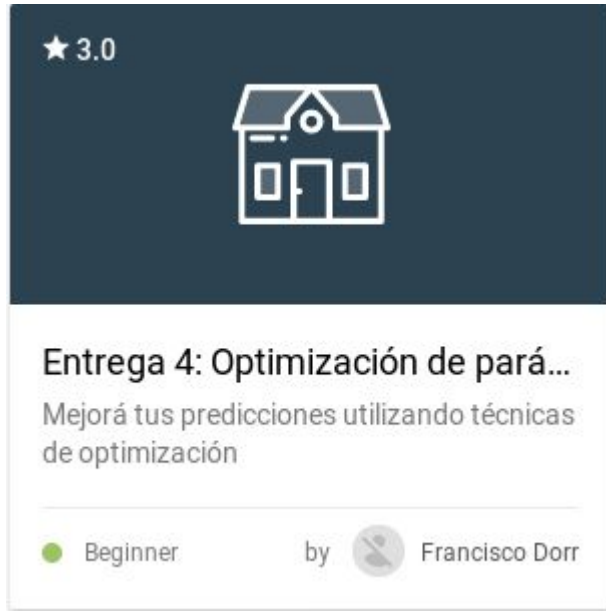
Proyecto 2:

Lanzamiento

Entrega 04



Proyecto 2: Modelado (Entrega 04)



1. Bajar los materiales.
2. Leer la Checklist
3. ¡Empezar a trabajar en la entrega!

Para la próxima

1. Trabajar en la Entrega 04.
2. Completar Notebook de hoy y atrasados.

ACÀMICA