ACÁMICA

¡Bienvenidos/as a Data Science!

Agenda

¿Cómo anduvieron?

Dudas Comunitarias Entrega 04

Break

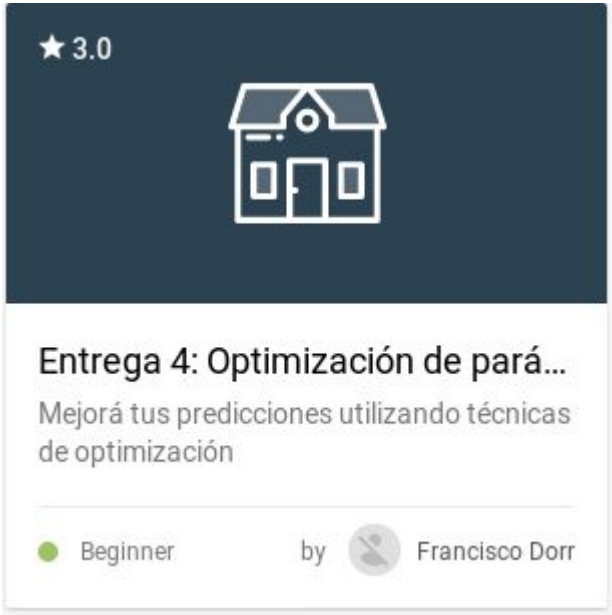
Flujo DS + ML

Cierre

¿Cómo anduvieron?

Proyecto 2:
Entrega 04

Proyecto 2: Modelado



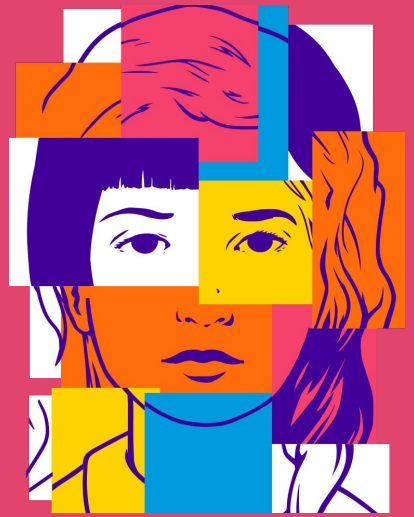
Actividad:
Dudas comunitarias

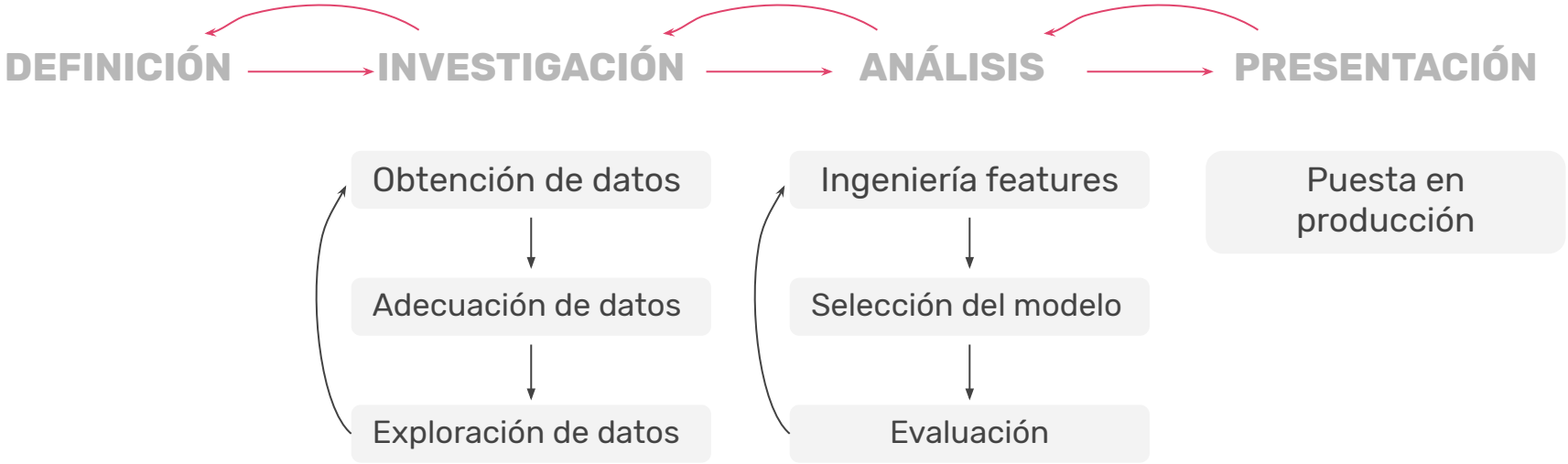
¡BREAK!

Ph. Credit: Drew Coffmann

Flujo Data Science + Machine Learning

Las siguientes diapositivas fueron hechas por Mariela Bisso





Veamos cada etapa más en detalle

1 • Definir del problema

¿Se les ocurre cómo podemos empezar a definir un problema?

1 • Definir del problema

1. ¿Cuál es el principal objetivo?
2. ¿Qué vamos a intentar predecir?
3. ¿A qué clase de problema nos enfrentamos? ¿Clasificación binaria? ¿Multiclase?¿Regresión?
4. ¿Cuáles son los datos de entrada? ¿Están disponibles?
5. ¿Cuáles son las características de la variable objetivo?
6. ¿Cómo se va a medir la variable objetivo?

Al definir un problema...

Es fundamental tener presente que Machine Learning sólo puede ser usado para memorizar pautas que están presentes en los datos de entrenamiento, por lo tanto sólo podemos reconocer lo que hemos visto antes.

Cuando usamos Machine Learning estamos asumiendo que el futuro se comportará como el pasado, lo que no siempre es cierto.

2 • Obtener los datos

Este es el primer paso en el desarrollo de un modelo de Machine Learning. Es un paso crítico con una influencia absoluta en cómo será de adecuado el modelo. Cuanto más y mejores datos obtengamos, mejor será el rendimiento de nuestro modelo.

3 • Elegir una medida o indicador de éxito

“Si no lo puedes medir, no lo puedes mejorar.”

3 • Elegir una medida o indicador de éxito

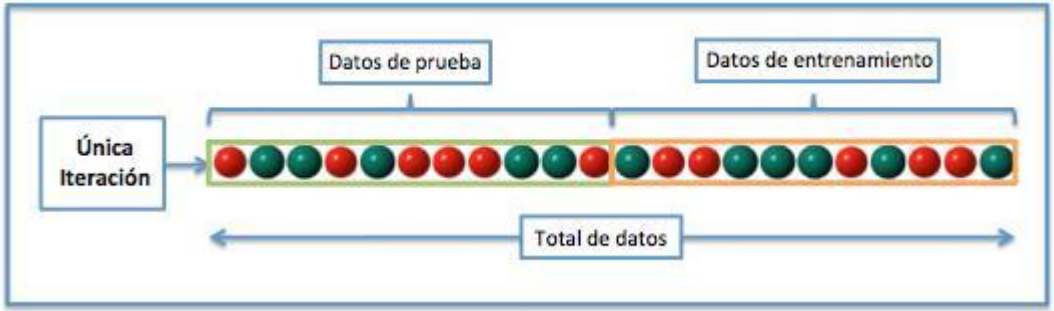
Esta medida debe estar directamente relacionada con el negocio y también con el tipo de problema al que nos enfrentamos…

4 • Establecer un protocolo de evaluación

1. Set de validación
2. Validación cruzada “K-Fold”
3. Validación cruzada aleatoria

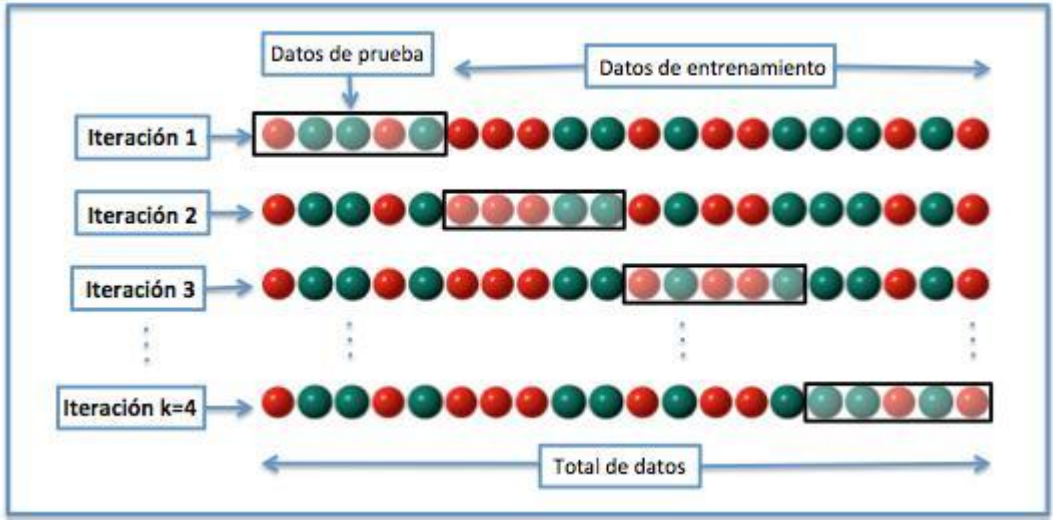
4 • Establecer un protocolo de evaluación

1. Set de validación
2. Validación cruzada “K-Fold”
3. Validación cruzada aleatoria



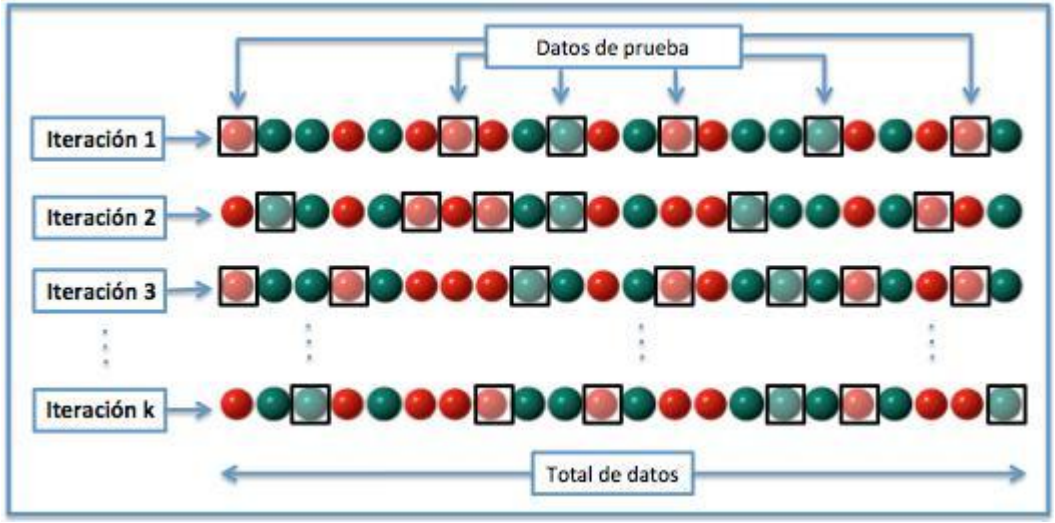
4 • Establecer un protocolo de evaluación

1. Set de validación
2. Validación cruzada “K-Fold”
3. Validación cruzada aleatoria



4 • Establecer un protocolo de evaluación

1. Set de validación
2. Validación cruzada “K-Fold”
3. Validación cruzada aleatoria



5 • Preparar los datos

1. Gestionar los datos perdidos/falantes
2. Manejar datos categóricos
3. Escalar características
4. Seleccionar características relevantes
5. División de datos en “subsets”

5.1 • Preparar los datos • Gestionar los datos perdidos

En Python, se representan típicamente con “NaN”.

El problema es que la mayoría de los algoritmos no pueden manejar esos valores faltantes, por lo que los tenemos que tener en cuenta antes de alimentar nuestro modelos con datos. Hay varias formas de tratar con ellos:

1. Eliminar las muestras o características con campos vacíos (corremos el riesgo de borrar información relevante o demasiadas muestras)
2. Estimar los campos vacíos con algún estimador preinstalado, como “Imputer Class” de la herramienta “Scikit Learn”. Primero ajustaremos nuestros datos, y después los transformaremos para estimarlos. Una aproximación común es establecer los valores faltantes como el valor medio del resto de las muestras.
3. Establecer un modelo con los datos que sí están completos en ese campo para poder predecir los datos faltantes.

5.2 • Preparar los datos • Manejar datos categóricos

Conversión de características ordinales	El caso de las tallas de ropa se puede realizar la siguiente correspondencia: L:2, M:1, S:0.
Codificación de etiquetas de clases nominales	Tres clases en color: rojo, amarillo, verde, y realizamos la codificación “one-hot”, obtendremos tres nuevas columnas, una para cada clase. Después, si tenemos una camiseta amarilla, será representado como amarillo = 1, verde = 0, rojo = 0.

5.3 • Preparar los datos • Escalar características

Este es un paso esencial en la fase de pre-procesamiento, ya que la mayoría de los algoritmos de Machine Learning tienen mucho mejor rendimiento cuando tratan con características que están en la misma escala. Las **técnicas más comunes** son:

5.3 • Preparar los datos • Escalar características

Este es un paso esencial en la fase de pre-procesamiento, ya que la mayoría de los algoritmos de Machine Learning tienen mucho mejor rendimiento cuando tratan con características que están en la misma escala. Las **técnicas más comunes** son:

NORMALIZACIÓN

Se refiere a reescalar las características en un rango [0,1], que es un caso especial de la escalación "min-max".

$$X_{changed} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

5.3 • Preparar los datos • Escalar características

Este es un paso esencial en la fase de pre-procesamiento, ya que la mayoría de los algoritmos de Machine Learning tienen mucho mejor rendimiento cuando tratan con características que están en la misma escala. Las **técnicas más comunes** son:

NORMALIZACIÓN

Se refiere a reescalar las características en un rango [0,1], que es un caso especial de la escalación "min-max".

$$X_{changed} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

ESTANDARIZACIÓN

Consiste en centrar las columnas de características con respecto a media 0 con desviación estándar 1 de forma que las columnas de características tengan los mismos parámetros que una distribución normal estándar (media cero y varianza unidad).

$$z = \frac{x - \mu}{\sigma}$$

μ = Mean
σ = Standard Deviation

5.4 • Preparar los datos • Seleccionar características relevantes

Un tipo de funcionamiento inadecuado de los modelos de Machine Learning es la existencia de redundancia en los datos. Esto hace que el modelo sea demasiado complejo con respecto a los datos de entrenamiento facilitados, y por tanto incapaz de hacer generalizaciones correctas en datos no procesados aún.

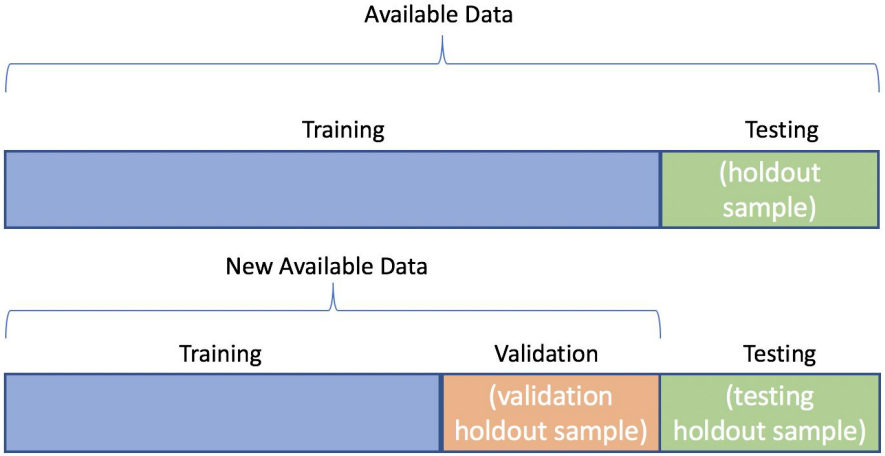
5.4 • Preparar los datos • Seleccionar características relevantes

Un tipo de funcionamiento inadecuado de los modelos de Machine Learning es la existencia de redundancia en los datos. Esto hace que el modelo sea demasiado complejo con respecto a los datos de entrenamiento facilitados, y por tanto incapaz de hacer generalizaciones correctas en datos no procesados aún.

Más adelante veremos técnicas de selección de variables (reducción de dimensionalidad).

5.5 • Preparar los datos • Dividir en “subsets”

En general, fragmentaremos nuestros datos en tres partes: conjuntos de entrenamiento, validación y pruebas. Entrenamos nuestro modelo con datos de entrenamiento, lo evaluamos con datos de validación, y finalmente, una vez está listo para usarse, lo probamos una última vez con datos de prueba.



5.5 • Preparar los datos • Dividir en “subsets”

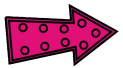
Desarrollar un modelo implica ajustar su configuración, en otras palabras, escoger ciertos valores para sus hiperparámetros.

> En **el aprendizaje automático**, un hiperparámetro es un **parámetro** cuyo valor se establece antes de que comience el proceso de aprendizaje. Por el contrario, los valores de otros parámetros se derivan a través del entrenamiento.

5.5 • Preparar los datos • Dividir en “subsets”

Desarrollar un modelo implica ajustar su configuración, en otras palabras, escoger ciertos valores para sus hiperparámetros.

> En **el aprendizaje automático**, un hiperparámetro es un **parámetro** cuyo valor se establece antes de que comience el proceso de aprendizaje. Por el contrario, los valores de otros parámetros se derivan a través del entrenamiento.



Este ajuste se realiza con la información recibida de los datos de validación, y es en esencia una forma de aprendizaje.

5.5 • Preparar los datos • Dividir en “subsets”

El objetivo es que el modelo pueda generalizar bien con datos nuevos, basados en sus parámetros internos ajustados mientras el modelo fue entrenado y validado.

a. Proceso de aprendizaje

b. Overfitting y Underfitting

5.5 • Preparar los datos • Dividir en “subsets”

a. Proceso de aprendizaje

b. Overfitting y Underfitting

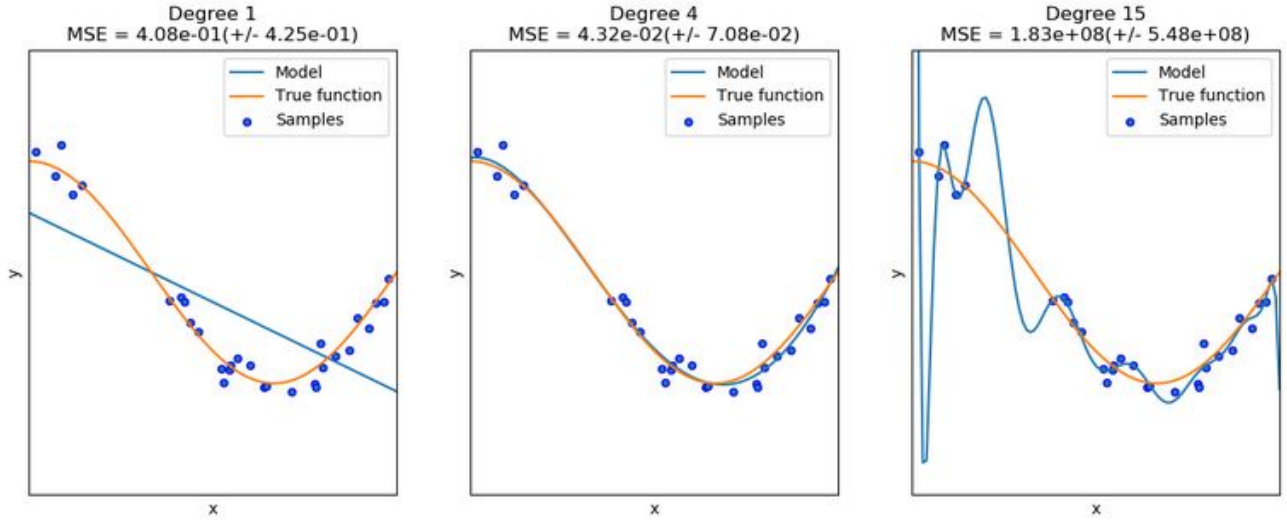
Uno de los problemas más importantes cuando trabajamos con el entrenamiento de modelos es el conflicto entre optimización y generalización.

> **Optimización** es el proceso de ajuste de un modelo para conseguir el mejor rendimiento posible de los datos de entrenamiento (proceso de aprendizaje).
>
> **Generalización** es cómo de bien se comporta el modelos ante datos no procesados aún. El objetivo es conseguir la mejor capacidad de generalización.

5.5 • Preparar los datos • Dividir en “subsets”

a. Proceso de aprendizaje

b. Overfitting y Underfitting



¿Cómo EVITAR el Overfitting?

Hay dos formas de evitar el “overfitting”:

1. **Obtener más datos** es normalmente la mejor solución, un modelo entrenado con más datos generalizará mejor de forma natural.
2. **Penalizar modelos muy complejos.** Para eso,
   a. Elegimos valores de los hiperparámetros a partir de curvas de complejidad, Grid Search, etc.
   b. **La regularización** impone restricciones sobre modelos muy complejos. Si el modelo solo memoriza un pequeño número de pautas, la regularización hará que el enfoque se haga en las más relevantes, mejorando la posibilidad de generalizar bien.

¿Cómo EVITAR el Overfitting?

La regularización se realiza principalmente con las siguientes técnicas:

1. **Reduciendo el tamaño del modelo**: disminuyendo el número parámetros que el modelo tiene que aprender, y con ello, su capacidad de aprendizaje.
2. **Añadiendo regularización de peso**: en general, cuanto más simple es el modelo, es mejor. Mientras pueda aprender bien, un modelo más simple es menos proclive a sufrir "overfitting". Una forma común de conseguir esto es restringir la complejidad forzando sus pesos para solamente tomar pequeños valores, regularizando la distribución de valores de peso. Esto se realiza añadiendo a la función de pérdida un costo asociado a tener grandes pesos.

¿Cómo EVITAR el Overfitting?

Por ejemplo, para una regresión lineal:

L1 Regularization

$$\text{Cost} = \sum_{i=0}^{N} (y_i - \sum_{j=0}^{M} x_{ij} W_j)^2 + \lambda \sum_{j=0}^{M} |W_j|$$

Regularización L1:
El coste es proporcional al cuadrado del valor de los coeficientes de peso (norma L1 de los pesos).

L2 Regularization

$$\text{Cost} = \underbrace{\sum_{i=0}^{N} (y_i - \sum_{j=0}^{M} x_{ij} W_j)^2}_{\text{Loss function}} + \underbrace{\lambda \sum_{j=0}^{M} W_j^2}_{\text{Regularization Term}}$$

Regularización L2
El coste es proporcional al cuadrado del valor de los coeficientes de peso (norma L2 de los pesos)

¿Cómo EVITAR el Overfitting?

Por ejemplo, para una regresión lineal:

L1 Regularization

$$\text{Cost} = \sum_{i=0}^{N} (y_i - \sum_{j=0}^{M} x_{ij} W_j)^2 + \lambda \sum_{j=0}^{M} |W_j|$$

L2 Regularization

$$\text{Cost} = \underbrace{\sum_{i=0}^{N} (y_i - \sum_{j=0}^{M} x_{ij} W_j)^2}_{\text{Loss function}} + \underbrace{\lambda \sum_{j=0}^{M} W_j^2}_{\text{Regularization Term}}$$

Regularización L1:
El coste es proporcional al cuadrado del valor de los coeficientes de peso (norma L1 de los pesos).

Regularización L2
El coste es proporcional al cuadrado del valor de los coeficientes de peso (norma L2 de los pesos)

Más adelante veremos más en detalle regularización.

6 • Desarrollar un modelo base (benchmark)

El objetivo en este paso del proceso es desarrollar un modelo de comparación que sirva de referencia, sobre el que mediremos el rendimiento de un algoritmo mejor y más ajustado.

7 • Desarrollar un modelo mejor y ajustar sus parámetros

1. Encontrar un buen modelo
2. Ajustar los hiperparámetros del modelo

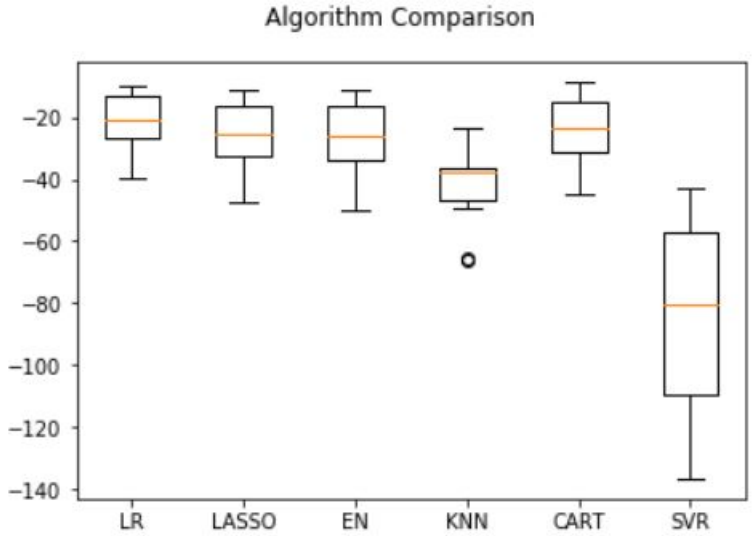
Uno de los métodos más comunes para encontrar un buen modelo es la **validación cruzada.** En la validación cruzada, estableceremos:

- Un número de subdivisiones en los que fragmentaremos nuestros datos.
- Un método de puntuación (que variará dependiendo de la naturaleza del problema — regresión, clasificación,…..)
- Algunos algoritmos apropiados que queremos comprobar

Pasaremos nuestro conjunto de datos a nuestra función de puntuación de validación cruzada y obtener el modelo que da la mejor puntuación. Éste será el que optimizaremos, ajustando sus parámetros en consecuencia.

7 • Desarrollar un modelo mejor y ajustar sus parámetros

1. Encontrar un buen modelo
2. Ajustar los hiperparámetros del modelo



7 • Desarrollar un modelo mejor y ajustar sus parámetros

1. Encontrar un buen modelo
2. Ajustar los hiperparámetros del modelo

Un algoritmo de Machine Learning tiene dos tipos de parámetros, el primer tipo son los parámetros que se aprenden a través de la fase de aprendizaje, y el segundo tipo son los hiperparámetros que transferimos al modelos de machine Learning.

7 • Desarrollar un modelo mejor y ajustar sus parámetros

1. Encontrar un buen modelo
2. Ajustar los hiperparámetros del modelo

Un algoritmo de Machine Learning tiene dos tipos de parámetros, el primer tipo son los parámetros que se aprenden a través de la fase de aprendizaje, y el segundo tipo son los hiperparámetros que transferimos al modelos de machine Learning.

Una vez identificado el modelo que usaremos, el siguiente paso es ajustar sus hiperparámetros para obtener el mayor poder predictivo posible. La forma más común de encontrar la mejor combinación de hiperparámetros se llama “Grid Search Cross Validation”.

7 • Desarrollar un modelo mejor y ajustar sus parámetros

1. Encontrar un buen modelo
2. Ajustar los hiperparámetros del modelo

El proceso sería el siguiente:

A. Establecer la matriz de parámetros que evaluaremos. Haremos esto creando un diccionario de todos los parámetros y su conjunto correspondientes de valores que deseamos para las pruebas de mejor rendimiento.

B. Establecer el número de subdivisiones de datos, el estado aleatorio y un método de puntuación.

C. Construir un objeto “K-Fold”con el número de subdivisiones seleccionado.

D. Construir un objeto de búsqueda de cuadrícula con el modelo seleccionado y ajustarlo.

¿Recuerdan la clasificación de Aprendizaje supervisado?

Aprendizaje supervisado: Clasificación

Modelos

- Árboles de decisión (Hiperparámetros: profundidad, criterio de entrenamiento, etc.)

- KNN (Hiperparámetros: cantidad de vecinos, distancia, etc.)

Métricas de evaluación

- Exactitud
- Matriz de Confusión[1]
- Precisión/Exhaustividad
- F-Score
- Curva ROC[1] / AUC

[1] Bueno, técnicamente no son métricas

Aprendizaje supervisado: Regresión

Modelos

- **Regresión lineal** (¿Hiperparámetros?)
- **Árboles de decisión** (Hiperparámetros: profundidad, etc.)
- **KNN** (Hiperparámetros: cantidad de vecinos, distancia, etc.)

Métricas de evaluación

- MAE
- MSE/RMSE

Aprendizaje supervisado

Durante las siguientes clases vamos a ver modelos avanzados:

- Support Vector Machine (SVM)
- Modelos de Ensamble: Random Forest, Adaboost, XGBoost
- Redes Neuronales

HAGAMOS UN RESUMEN DEL PROCESO...

1. Definir el problema
2. Obtener los datos
3. Elegir una medida o indicador de éxito
4. Establecer un protocolo de evaluación
5. Preparar los datos (Gestionar datos perdidos, Manejar datos categóricos, Escalar características, Seleccionar características relevantes, Dividir en “subsets”)
6. Desarrollar un modelo base
7. Desarrollar un modelo mejor y ajustar sus parámetros

Para la próxima

1. Terminar la Entrega 04
2. Ver los videos de la plataforma “Clasificación Avanzada”
3. Completar Notebooks atrasados

ACÁMICA