



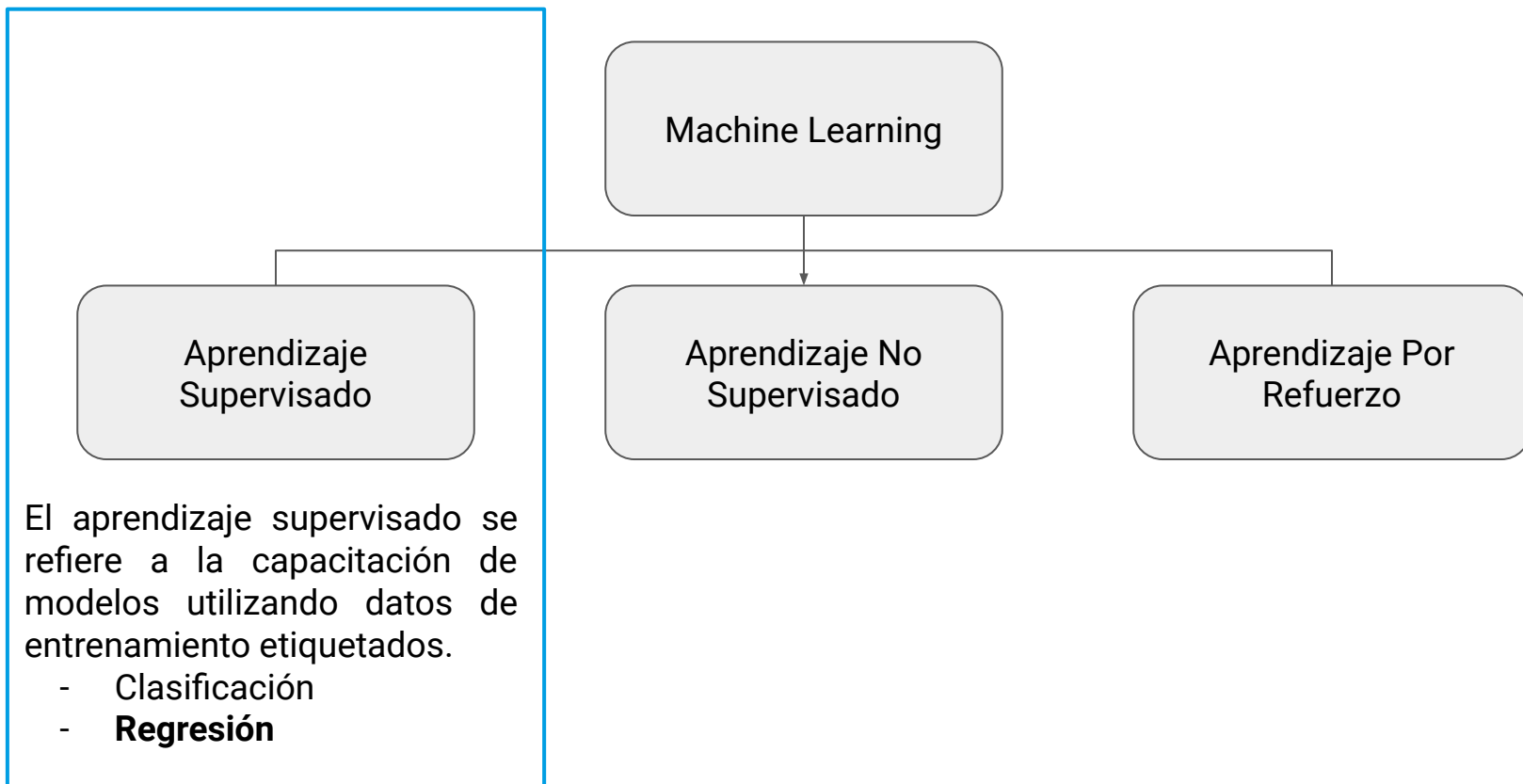
Aprendizaje Supervisado - Regresión

Tutoría

¿Donde se ubica la
regresión en las tareas
de machine learning?



Aprendizaje Supervisado



¿A qué se refieren los
conceptos de overfitting
y underfitting?



¿Qué función tiene el
kernel en el support
vector machine?



Autoaprendizaje

Recursos asincrónicos

- ¿Revisaste los recursos de la semana 5 (Guía y desafío)?
- ¿Tienes dudas sobre alguno de ellos?



Ideas fuerza



Los problemas de **Regresión** corresponden a la predicción de variables objetivo continuas..

Muchos de los modelos que se usan para clasificación tienen su adaptación a problemas de regresión como **árboles de regresión**.

Hay múltiples métricas para problemas de regresión, donde las más utilizadas son **R2, RMSE, MAPE, etc.**

/* Regresión */

Regresión

¿En qué consiste?

En este caso, tenemos una variable objetivo numérica, se intenta entender el comportamiento entre las variables predictoras y un valor continuo.

Variables Predictoras:

Se utilizan para predecir la variable objetivo

age	sex	bmi	bp	glucosa
0.038076	0.050680	0.061696	0.021872	-0.017646
-0.001882	-0.044642	-0.051474	-0.026328	-0.092204
0.085299	0.050680	0.044451	-0.005670	-0.025930
-0.089063	-0.044642	-0.011595	-0.036656	-0.009362
0.005383	-0.044642	-0.036385	0.021872	-0.046641
...
0.041708	0.050680	0.019662	0.059744	0.007207
-0.005515	0.050680	-0.015906	-0.067642	0.044485
0.041708	0.050680	-0.015906	0.017293	0.015491
-0.045472	-0.044642	0.039062	0.001215	-0.025930
-0.045472	-0.044642	-0.073030	-0.081413	0.003064

Variable Objetivo: numérico
(Valores continuos con cierta distribución)

/* Regresión Lineal */

Regresión Lineal

The diagram illustrates the components of a linear regression equation. The equation is $earn_i = \beta_0 + \beta_1 \cdot height_i + \varepsilon_i$. Labels in boxes with arrows point to specific parts: 'Variable Dependiente' points to $earn_i$; 'Pendiente' points to β_1 ; 'Variable Independiente' points to $height_i$; 'Intercepto' points to β_0 ; and 'Error' points to ε_i .

Variable Dependiente

Pendiente

Variable Independiente

$$earn_i = \beta_0 + \beta_1 \cdot height_i + \varepsilon_i$$

Intercepto

Error

Estimación de parámetros en regresión lineal

Mínimos cuadrados

Para estimar los betas de la regresión lineal (“entrenar el modelo”) se utiliza el método de Mínimos cuadrados ordinarios (MCO o OLS), con el cual se busca ajustar los betas al mínimo error.

$$\beta = \operatorname{argmin} E[(y_i - X^T \beta)^2]$$

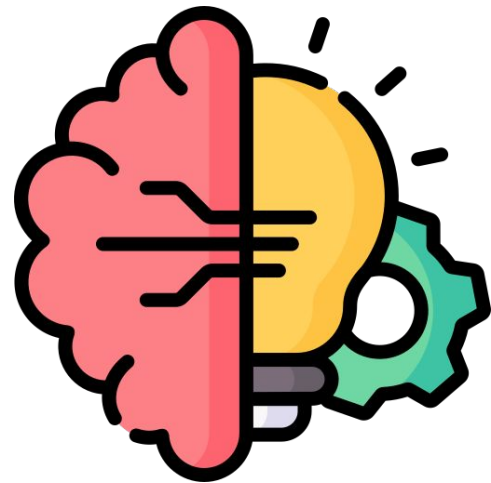
$$\beta = \sum (y_i - (\beta_0 + \beta_1 x)^2)$$

Enfoque Machine Learning

¿Qué lo caracteriza?

Recordemos que estamos observando la regresión lineal desde el enfoque de Machine Learning.

Esto significa que nuestro objetivo es poder estimar de la mejor forma en nuevos datos, es decir, poder generalizar el comportamiento del modelo.



/*Regularización*/

Ridge

Características y cálculo

$$\beta_{\text{Ridge}} = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^n (y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^p \beta_j^2$$

- Ridge modifica la superficie de penalización de los coeficientes mediante el **hiperparámetro lambda**.
- **Lambda** gobierna la **superficie de penalización** que está determinada por la cantidad de parámetros inferidos en el modelo.
- Dado que tiene una forma cuadrática, **suaviza pero no elimina** atributos irrelevantes.

Lasso

Características y cálculo

$$\beta_{\text{Lasso}} = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^n (y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^p |\beta_j|$$

- Principal diferencia con Ridge: permite seleccionar y **eliminar atributos** irrelevantes del modelo.
- De igual manera que en Ridge, el **hiperparámetro lambda** define el área de la superficie de penalización.
- La diferencia radica en la **norma de penalización**.

Elastic Net

Características y cálculo

$$\beta_{\text{ElasticNet}} = \underset{\beta}{\operatorname{argmin}} \sum_i^n (y_i - \hat{y}_i)^2 + \lambda_1 \sum_{j=1}^p |\beta_j| + \lambda_2 \sum_{j=1}^p \beta_j^2$$

- Elastic Net combina ambas normas de penalización.
- L1 nos asegura una selección de atributos.
- L2 nos asegura una penalización parsimoniosa de los coeficientes de los atributos.
- Existe un parámetro que gobierna la dominancia entre ambas formas de penalización.

¡Manos a la obra! Probando las diferentes penalizaciones



¡Manos a la obra!

Penalizaciones con Python

Veremos a continuación la implementación de las penalizaciones con Python. Para esto, observa los pasos que mostrará tu profesor en la presentación de Jupyter Notebook (puedes abrir el archivo adjunto de tu guía para observar y replicar los pasos). En esta presentación veremos:

Regularización en regresión lineal:

- a. Ridge regression
- b. Lasso regression
- c. Elastic Net



/* Árboles de regresión */

Árboles de regresión

Construcción

Selección de la división

El árbol comienza con un nodo raíz que contiene todos los datos de entrenamiento. En cada paso, se selecciona una característica y un umbral que dividirá los datos en dos grupos.

Cálculo de la predicción

Se calcula la predicción numérica para cada región basada en los valores promedio de las muestras en esa región.

Criterio de división

La elección de la característica y el umbral se realiza de manera que la reducción en el error de predicción sea máxima después de la división. El error se mide en términos de alguna métrica, como la suma de los cuadrados de los residuos (SSE) o la desviación absoluta media (MAD).

Crecimiento del árbol

El proceso de selección y división se repite para cada región creada en pasos anteriores, hasta que se cumple algún criterio de detención, como la profundidad máxima del árbol o el número mínimo de muestras en una región.

Árboles de regresión

Hiperparámetros

Hiperparámetro	Descripción
Máximo de Profundidad	hasta qué punto puede crecer
Cantidad de atributos	cuántos atributos se deben considerar
Mínimo de muestras en un nodo particionable	con cuántas observaciones podemos seguir subdividiendo.
Mínimo de muestras en un nodo terminal	con cuántas observaciones podemos dejar de subdividir.
Criterio de split	criterio para generar la mejor división.

¡Manos a la obra! Árboles con Python



¡Manos a la obra!

Árboles con Python

Veremos a continuación la implementación de los árboles de regresión con Python. Para esto, observa los pasos que mostrará tu profesor en la presentación de Jupyter Notebook (puedes abrir el archivo adjunto de tu guía para observar y replicar los pasos).

En esta presentación aprenderemos:

1. Construcción de un árbol de regresión
2. Ajuste de hiperparámetros



/* Métricas de regresión */

Métricas de regresión

Fórmulas

Métrica	Fórmula
Error Cuadrático Medio (MSE)	$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$
Raíz del Error Cuadrático Medio (RMSE)	$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$
Desviación Absoluta Media (MAE)	$\text{MAE} = \frac{1}{n} \sum_{i=1}^n y_i - \hat{y}_i $
Error Absoluto Porcentual Medio (MAPE)	$\text{MAPE} = \frac{1}{n} \sum_{i=1}^n \left \frac{y_i - \hat{y}_i}{y_i} \right \times 100\%$
Coefficiente de determinación (R2)	$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$

Ejercicio guiado - Implementando un modelo de regresión



Ejercicio guiado

Implementando un modelo de regresión

Veremos a continuación la implementación de un modelo de regresión para predecir precios de autos, a partir de algunos datos entregados en un dataset.

Puedes abrir tu archivo de Jupyter Notebook para ir aplicando los pasos que te indicará tu profesor.



Desafío

"Prediciendo los precios de las casas"

- ¿Leíste el desafío de esta semana? ¿Comprendes bien lo que se solicita en cada caso?
- ¿Hay contenidos que necesitas repasar antes de comenzar este desafío?
- ¿Necesitas algún ejemplo o indicación para alguna pregunta o requerimiento específico?





Próxima sesión...

Veremos modelos de series temporales, los usos e implementación en python.

{desafío}
latam_

*Academia de
talentos digitales*

