

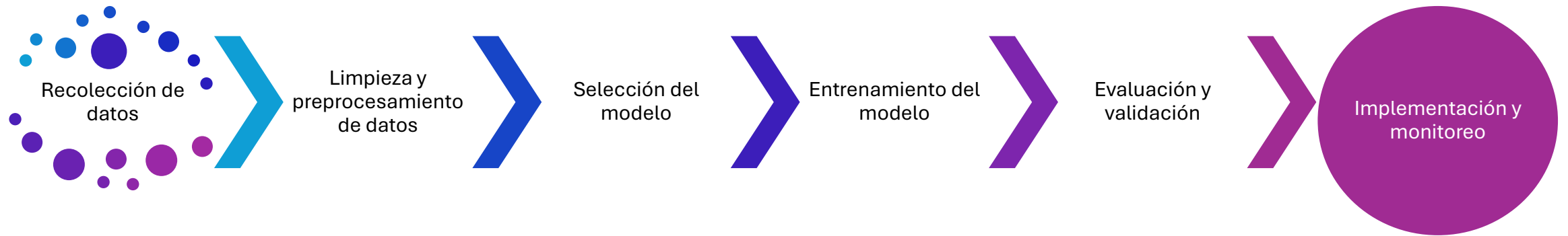
Machine Learning



Susana Medina Gordillo

susana.medina@correounivalle.edu.co

Flujo de trabajo en Machine Learning



Flujo de trabajo en Machine Learning



Evaluación y Comparación de Modelos de Clasificación

Métricas de Evaluación: El Barómetro del Rendimiento

Evaluación de modelos de Clasificación y su importancia en el aprendizaje supervisado

Matriz de confusión

En **clasificadores binarios**... la puntuación de probabilidad no es la realidad ni la verdad fundamental.

Hay cuatro resultados posibles para cada resultado de un clasificador binario.

Si se organiza **la verdad** fundamental como **columnas** y la **predicción** del modelo como **filas**, el resultado es una tabla, llamada **matriz de confusión**.

Matriz de confusión para Árboles de Decisión

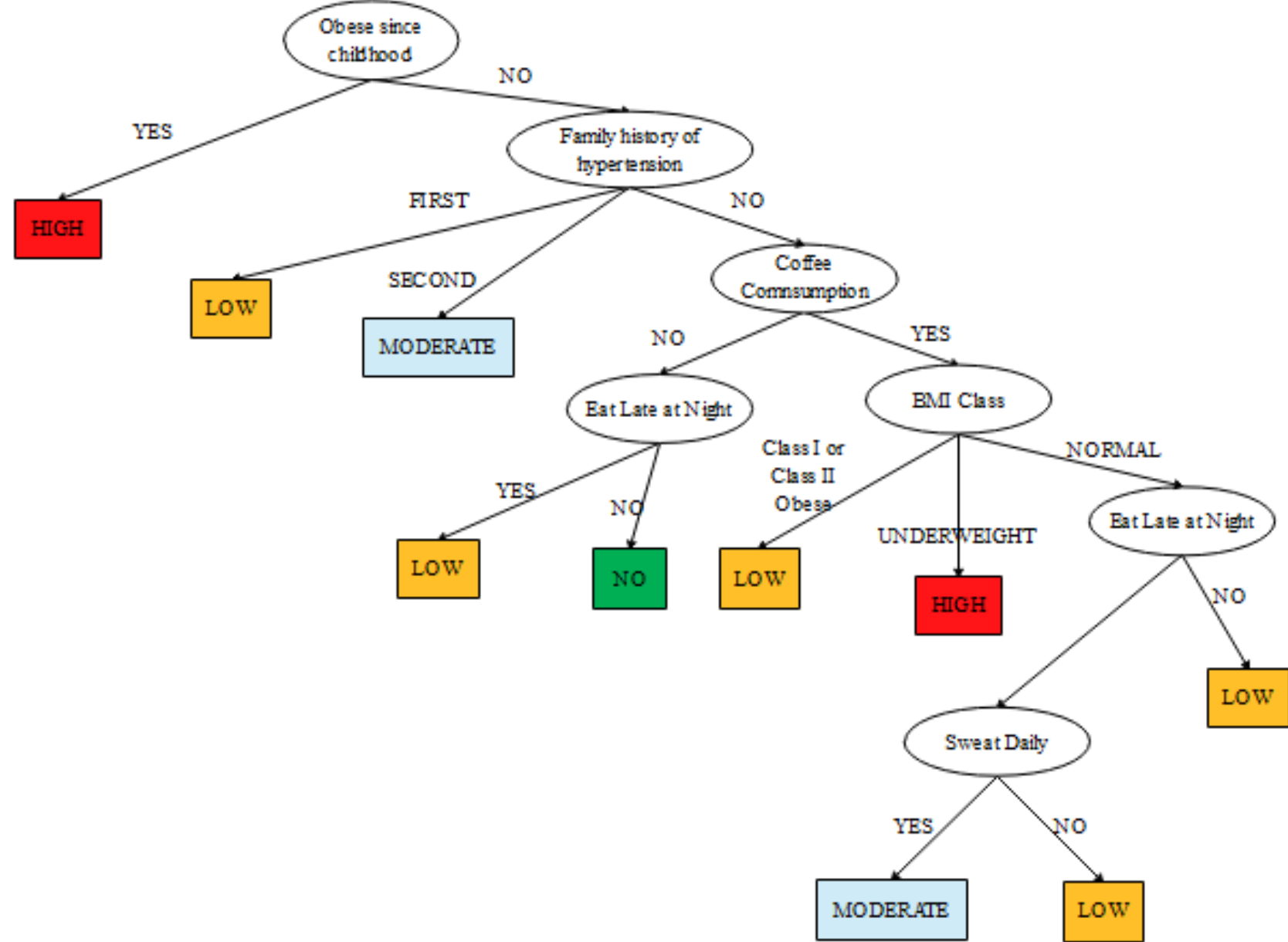
		Predicted	
		Positive	Negative
Real	Positive	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

		Predicted	
		Positive	Negative
Real	Positive	35	237
	Negative	15	5113

Matriz de confusión: spam

	Positivo real	Negativo real
Positivo previsto	Verdadero positivo (VP): Es un correo electrónico de spam clasificado correctamente como correo electrónico spam. Estos son los mensajes de spam que se envían automáticamente a la carpeta de spam.	Falso positivo (FP): Un correo electrónico que no es spam clasificado erróneamente como tal. Estos son los correos electrónicos legítimos que terminan en la carpeta de spam.
Negativo previsto	Falso negativo (FN): Un correo electrónico de spam clasificado erróneamente como no spam. Son correos electrónicos de spam que el filtro de spam no detecta y que llegan a la carpeta Recibidos.	Verdadero negativo (VN): Un correo electrónico que no es spam y que se clasificó correctamente como tal. Estos son los correos electrónicos legítimos que se envían directamente a la carpeta Recibidos.

Árboles de Decisión: Estructura jerárquica



Matriz de confusión: Clasificador multiclase

	NO	LOW	MODERATE	HIGH	
NO	5	0	1	0	NO
LOW	1	17	0	2	LOW
MODERATE	0	2	6	3	MODERATE
HIGH	0	0	0	8	HIGH

Ejercicio práctico #1: Clasificación

**Umbrales y
matriz de
confusión:**

Leer la sección y
realizar el quiz al
final (Verifica tu
comprensión)

12 min

developers.google.com



Curso intensivo de
aprendizaje automático

Un curso práctico para explorar los conceptos básicos del
aprendizaje automático.



Métricas de Evaluación: Clasificación

Exactitud

Recall

FPR

Precisión

F1-score

AUC-
ROC

Métricas de Evaluación: Exactitud (Accuracy)

La métrica fundamental, pero
engañosa en datasets
desbalanceados.

Mide la proporción de predicciones
correctas sobre el total

$$\text{Accuracy} = \frac{\text{correct classifications}}{\text{total classifications}} = \frac{TP + TN}{TP + TN + FP + FN}$$

Métricas de Evaluación: Recall

Llamada también **Sensibilidad**, Tasa de Verdaderos Positivos - **TPR**: Crucial cuando el costo de un falso negativo es alto.

Mide la proporción de instancias positivas reales que fueron correctamente identificadas.

$$\text{Recall (or TPR)} = \frac{\text{correctly classified actual positives}}{\text{all actual positives}} = \frac{TP}{TP + FN}$$

Métricas de Evaluación: FPR

La tasa de falsos positivos (FPR) es la proporción de todos los negativos reales que se clasificaron *incorrectamente* como positivos, también conocida como **probabilidad de falsa alarma**.

$$\text{FPR} = \frac{\text{incorrectly classified actual negatives}}{\text{all actual negatives}} = \frac{FP}{FP + TN}$$

Métricas de Evaluación: Precisión (precision)

La precisión es la proporción de todas las clasificaciones positivas del modelo que realmente son positivas

$$\text{Precision} = \frac{\text{correctly classified actual positives}}{\text{everything classified as positive}} = \frac{TP}{TP + FP}$$

Métricas de Evaluación: F1-score

La armónica entre Precisión y Recall.
Útil para buscar un **equilibrio** cuando
ambas métricas son importantes,
especialmente en datasets
desbalanceados.

$$F1 = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}} = \frac{2TP}{2TP + FP + FN}$$

Métricas de Evaluación: AUC-ROC

Área Bajo la Curva Característica Operativa del Receptor: **Robusta** ante desbalances de clase.

Grafica la TPR contra la Tasa de Falsos Positivos (FPR) a diferentes umbrales de clasificación, ofreciendo una **visión integral** de la capacidad discriminatoria del modelo.

Ejercicio práctico #2: Clasificación

**Exactitud,
recuperación,
precisión y
métricas
relacionadas:**

Leer la sección y
realizar el quiz al
final (Verifica tu
comprensión)

15 min

developers.google.com

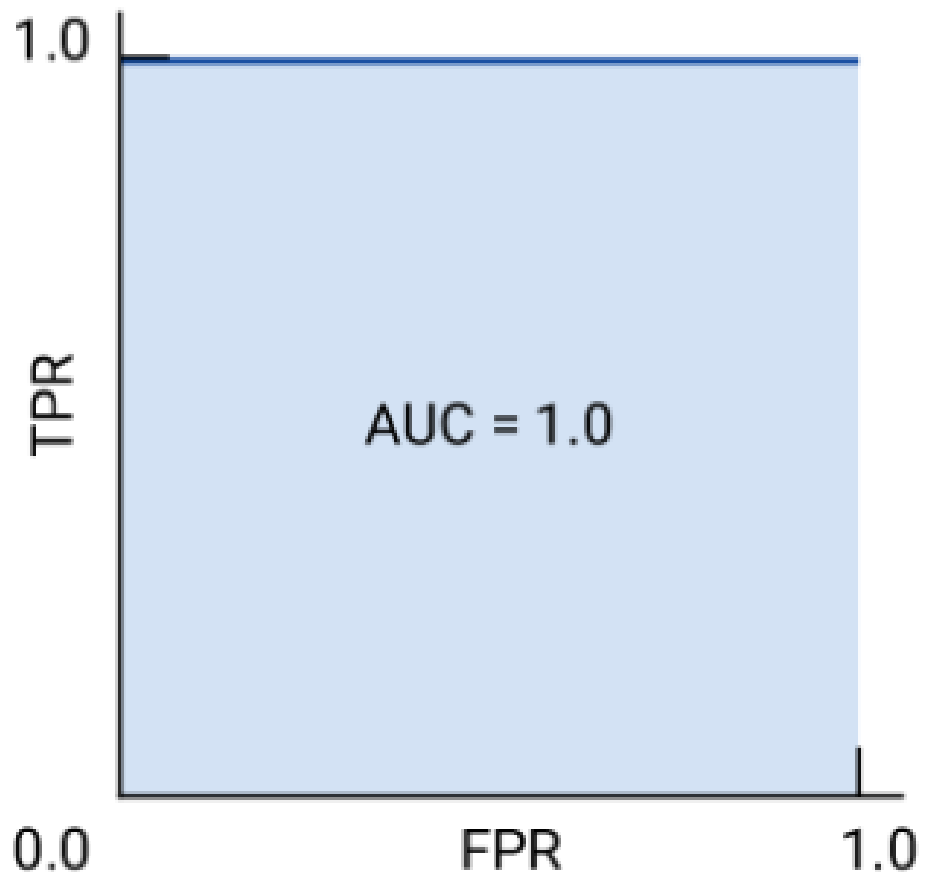


**Curso intensivo de
aprendizaje automático**

Un curso práctico para explorar los conceptos básicos del aprendizaje automático.



Métricas de Evaluación: AUC-ROC



La curva ROC es una representación visual del rendimiento del modelo en todos los umbrales. La versión larga del nombre, característica operativa del receptor, es una retención. de la detección de radares de la Segunda Guerra Mundial.

Figura 1: ROC y AUC de un modelo hipotético perfecto.

Métricas de Evaluación: AUC-ROC

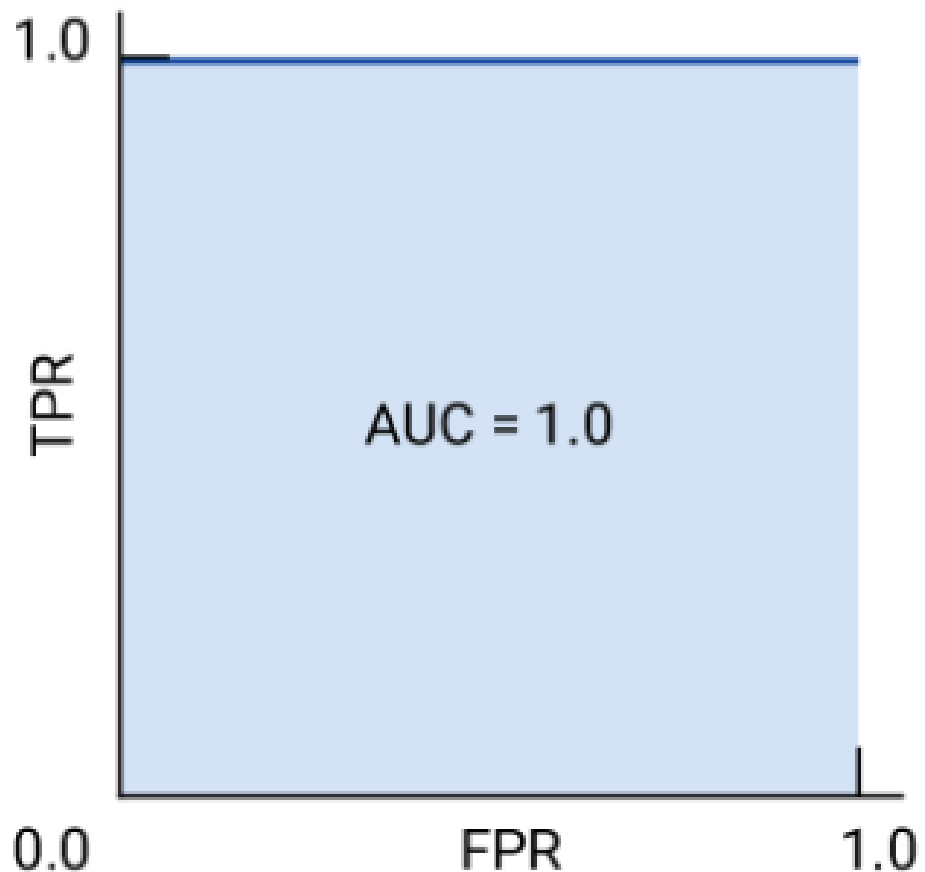


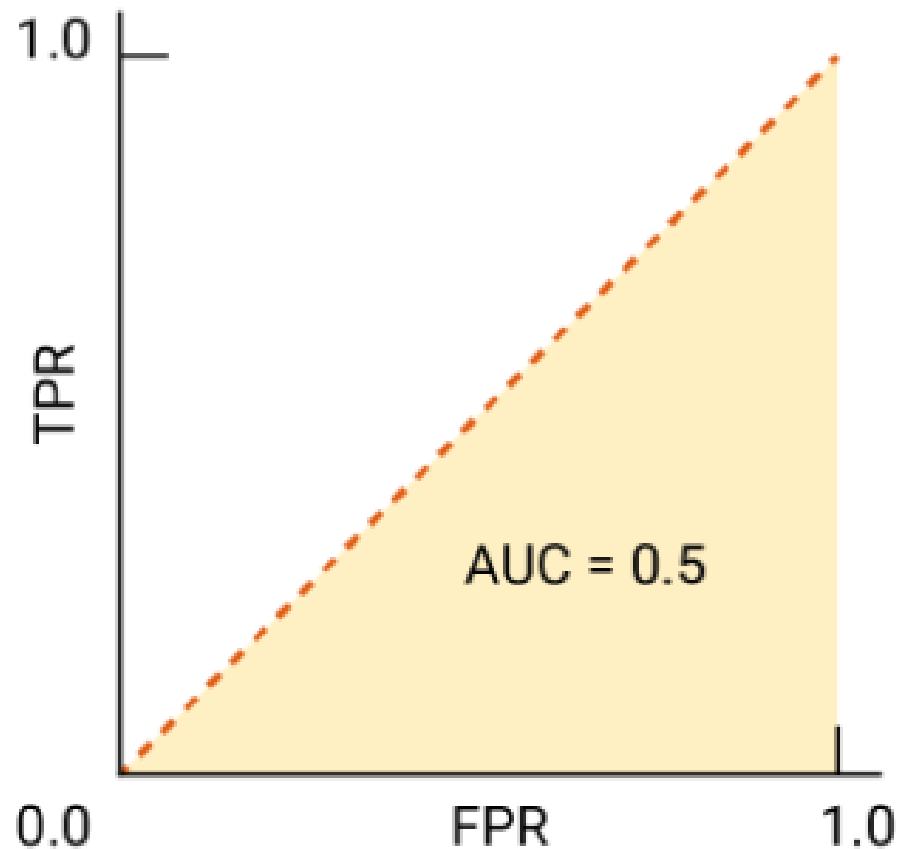
Figura 1: ROC y AUC de un modelo hipotético perfecto.

El área bajo la curva ROC (AUC) representa la probabilidad de que el modelo, si se da un ejemplo positivo y negativo elegido al azar, clasificará el positivo mayor que el negativo.

El modelo perfecto anterior, que contiene un cuadrado con lados de longitud 1, tiene un área bajo la curva (AUC) de 1.0.

Esto significa que hay un 100% de probabilidades de que el modelo clasificará de manera correcta un ejemplo positivo elegido al azar más alto que un ejemplo negativo elegido al azar.

Métricas de Evaluación: AUC-ROC

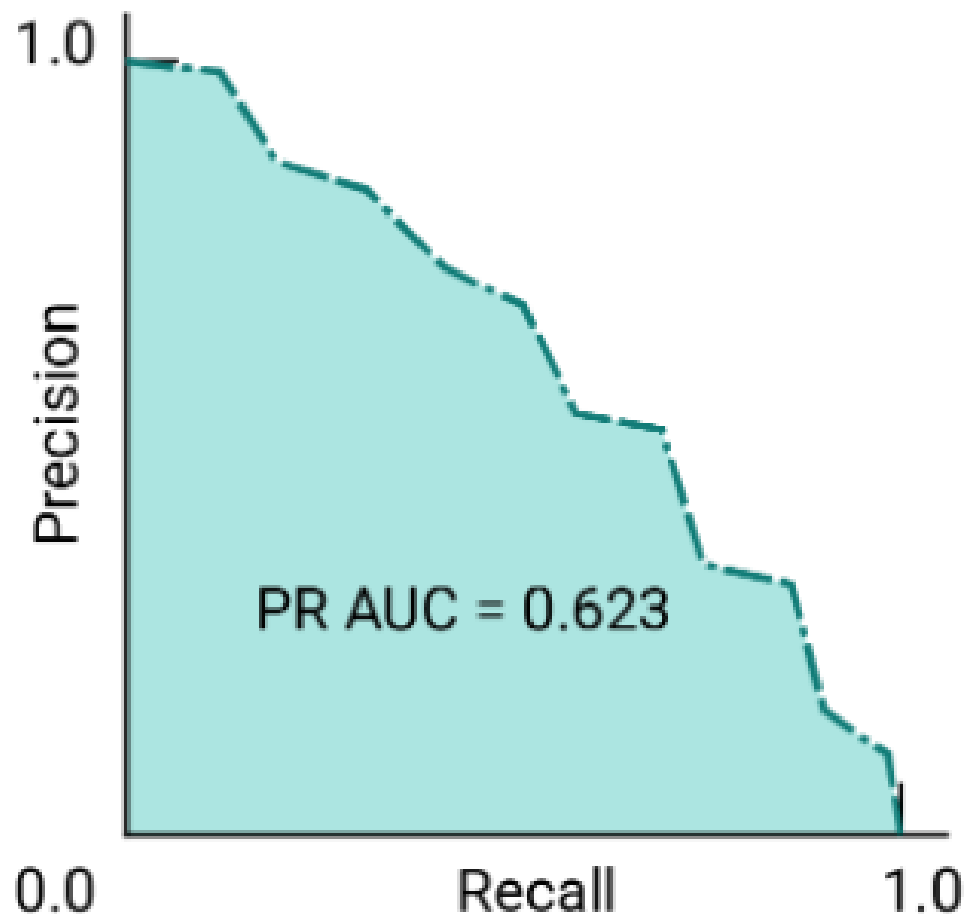


Para un **clasificador binario**, un modelo que realiza exactamente igual de bien como suposiciones o **predicciones aleatorias lanzamiento de monedas** tiene una **ROC**, que es una línea diagonal de (0,0) a (1,1).

El **AUC es 0.5**, que representa un 50% de probabilidad de clasificar correctamente un positivo aleatorio y ejemplo negativo.

Figura 2: ROC y AUC de suposiciones completamente aleatorias

Métricas de Evaluación: Recall vs Precision



Las curvas de precisión-recuperación se crean la precisión en el eje Y y la recuperación en el eje X en todos umbrales.

Cuando el **conjunto de datos está desequilibrado**, la curva de visión general precisión-recuperación (**PRC**) y el área debajo de ellas pueden ofrecer una mejor comparación visualización del **rendimiento del modelo**.

Métricas de Evaluación: Recall vs Precision

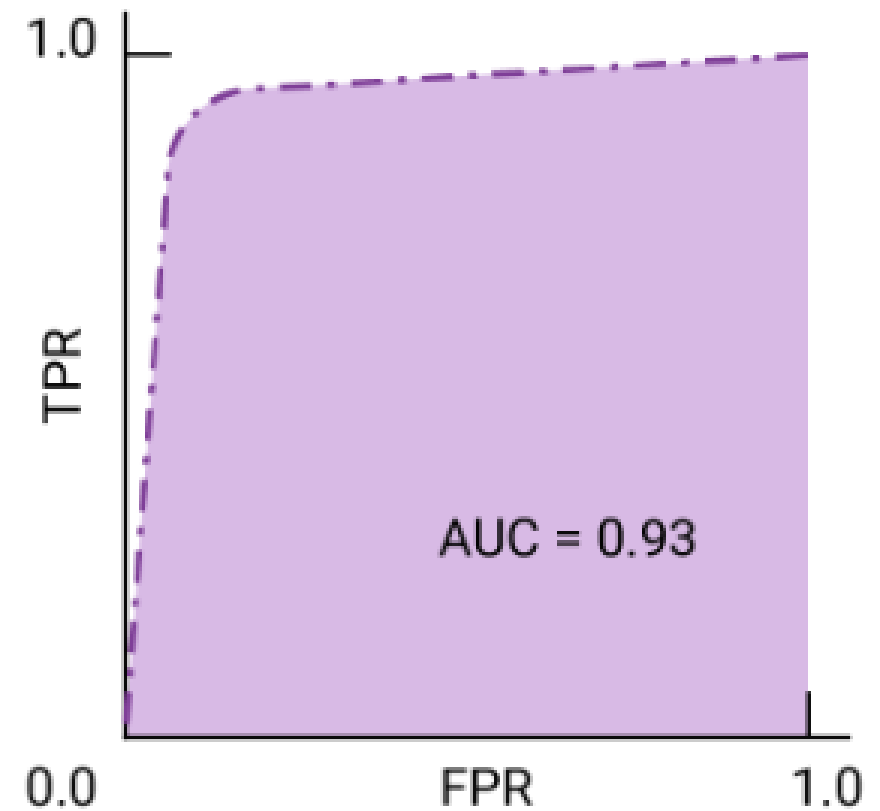
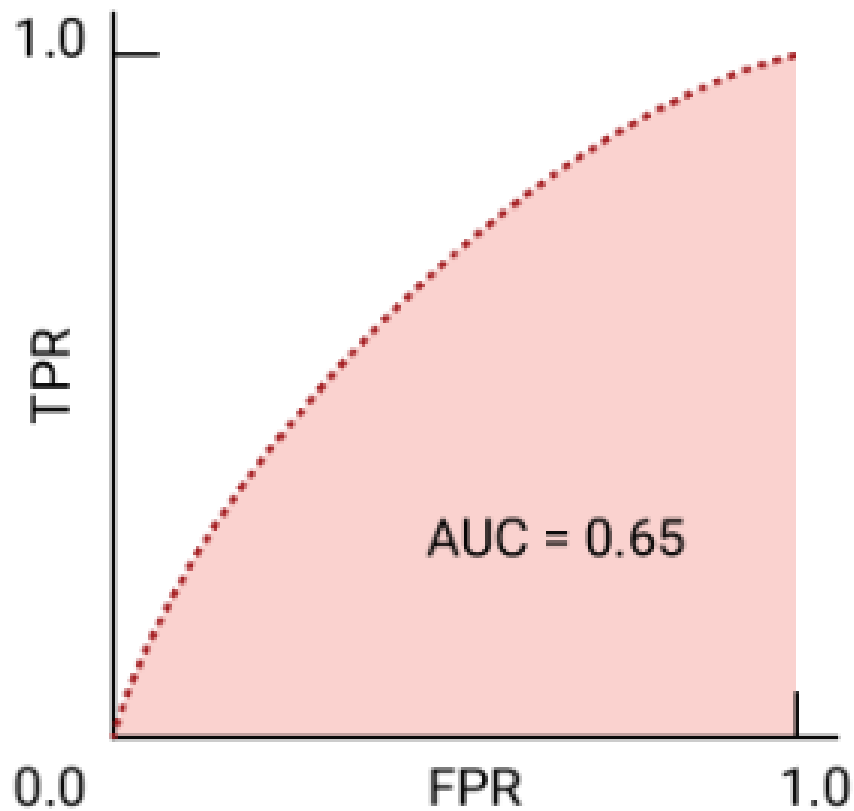


Figura 3: ROC y AUC de dos modelos hipotéticos. La curva en la a la derecha, con un AUC mayor, representa el mejor de los dos modelos.

Ejercicio práctico #3: Clasificación

ROC y AUC:

Leer la sección y realizar el quiz al final (Verifica tu comprensión)

10 min

developers.google.com



Curso intensivo de
aprendizaje automático

Un curso práctico para explorar los conceptos básicos del aprendizaje automático.



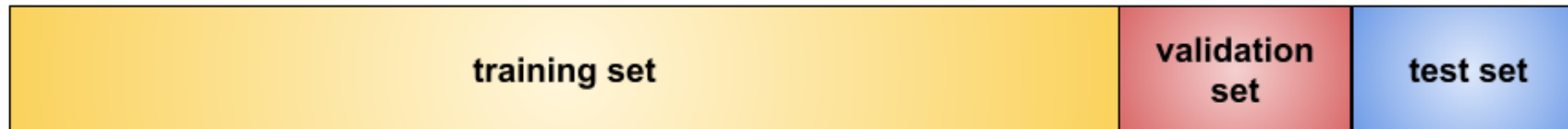
Técnicas de Validación Cruzada: Robustez en la Evaluación

División del dataset

Un **conjunto de entrenamiento (training set)** con el que se entrena el modelo.

Un **conjunto de pruebas (test set)** para evaluar el modelo entrenado.

Un **conjunto de validación (validation set)** realiza las pruebas iniciales en el modelo a medida que se entrena.



Técnicas de Validación Cruzada

Validación Cruzada (K-Fold CV)

- Divide el dataset en 'k' particiones.
- Entrena el modelo en '**k-1**' **particiones** y evalúa en la restante, repitiendo el proceso 'k' veces.
- Proporciona una **estimación más fiable** del rendimiento del modelo al reducir la variabilidad de una única división *train-test*.

Stratified K-Fold CV

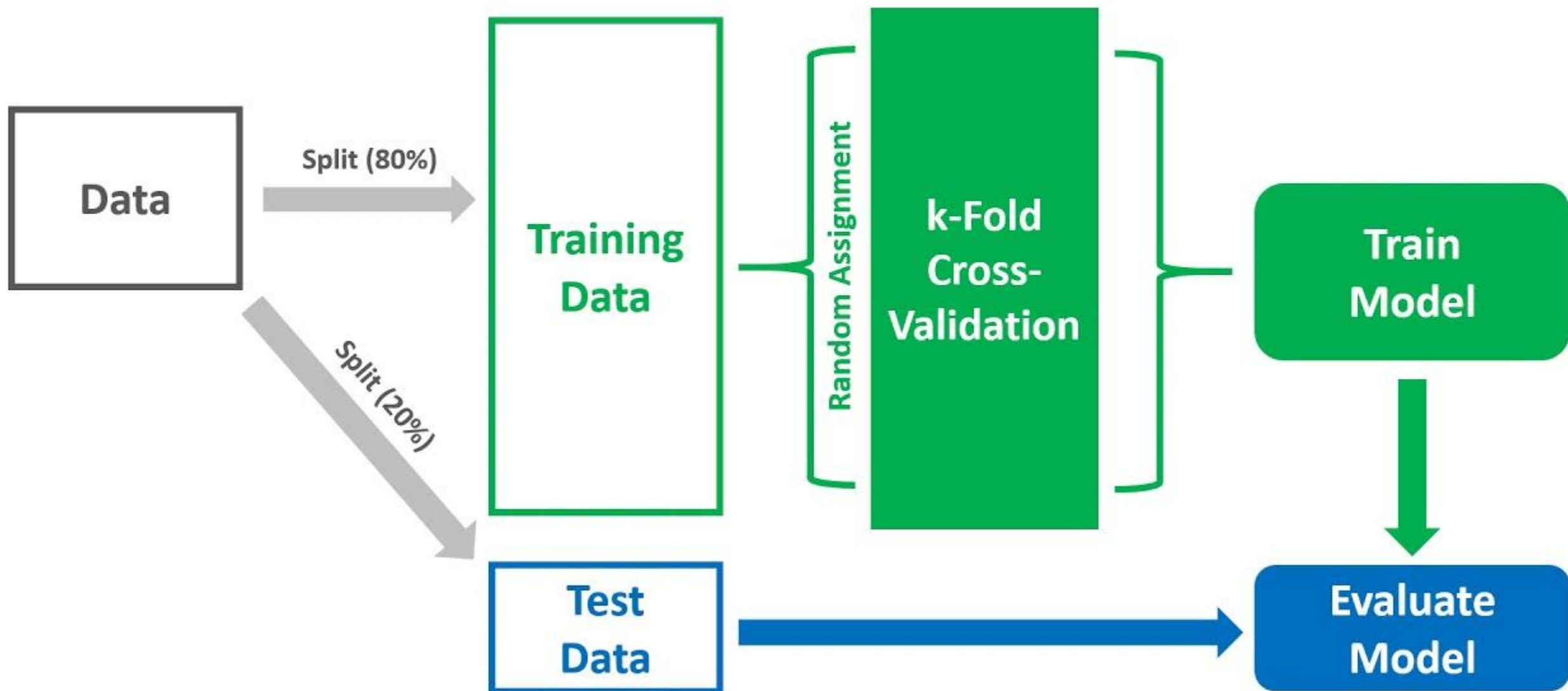
- Esencial para datasets **desbalanceados**.
- Asegura que cada partición mantenga la proporción de clases del dataset original durante la división.

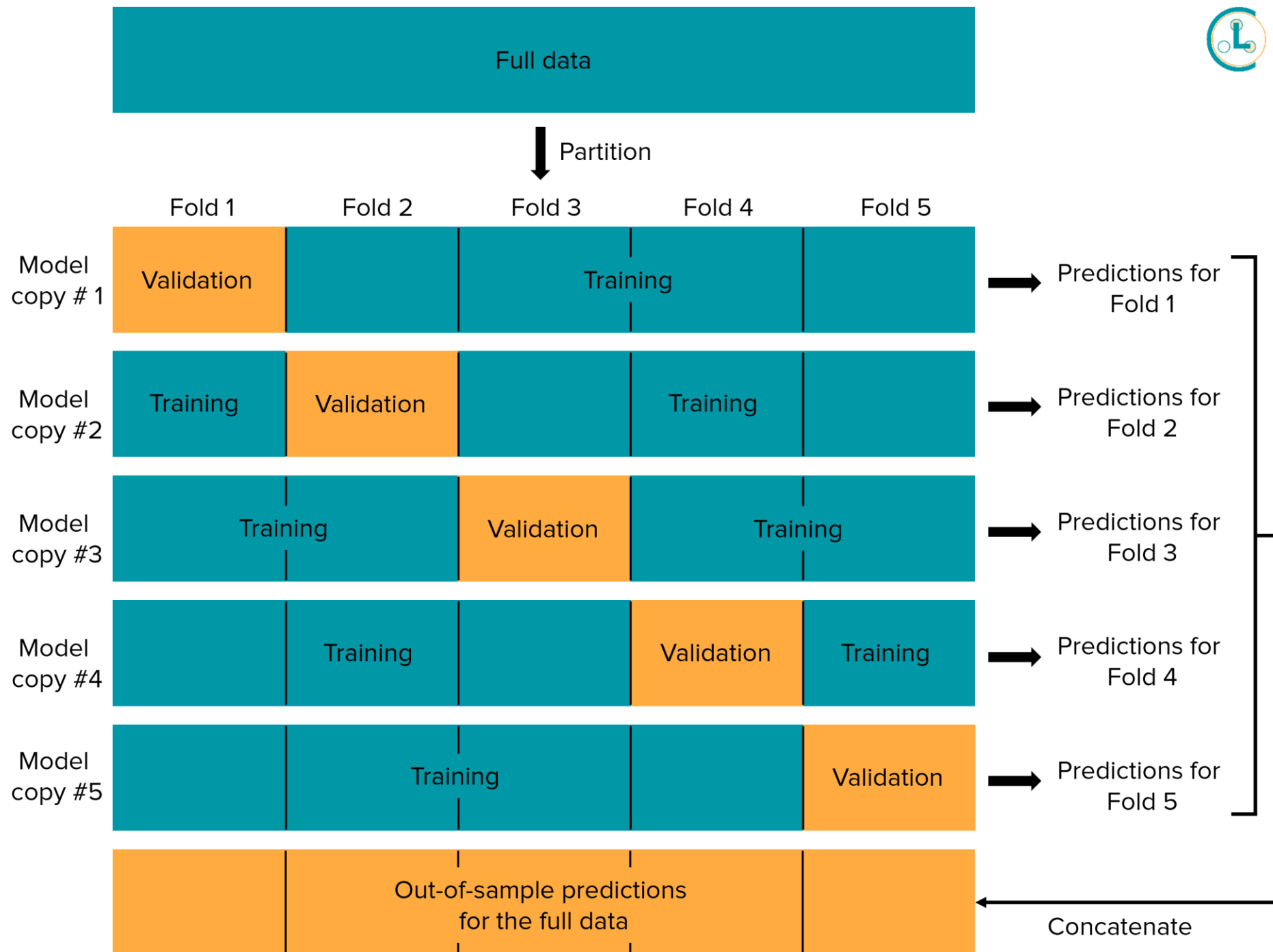
Leave-One-Out Cross-Validation (LOOCV)

- Caso extremo de K-Fold donde '**k**' es igual al **número de instancias**.
- Útil para datasets pequeños, pero **computacionalmente costoso**.

Validación Cruzada (K-Fold Cross-Validation)

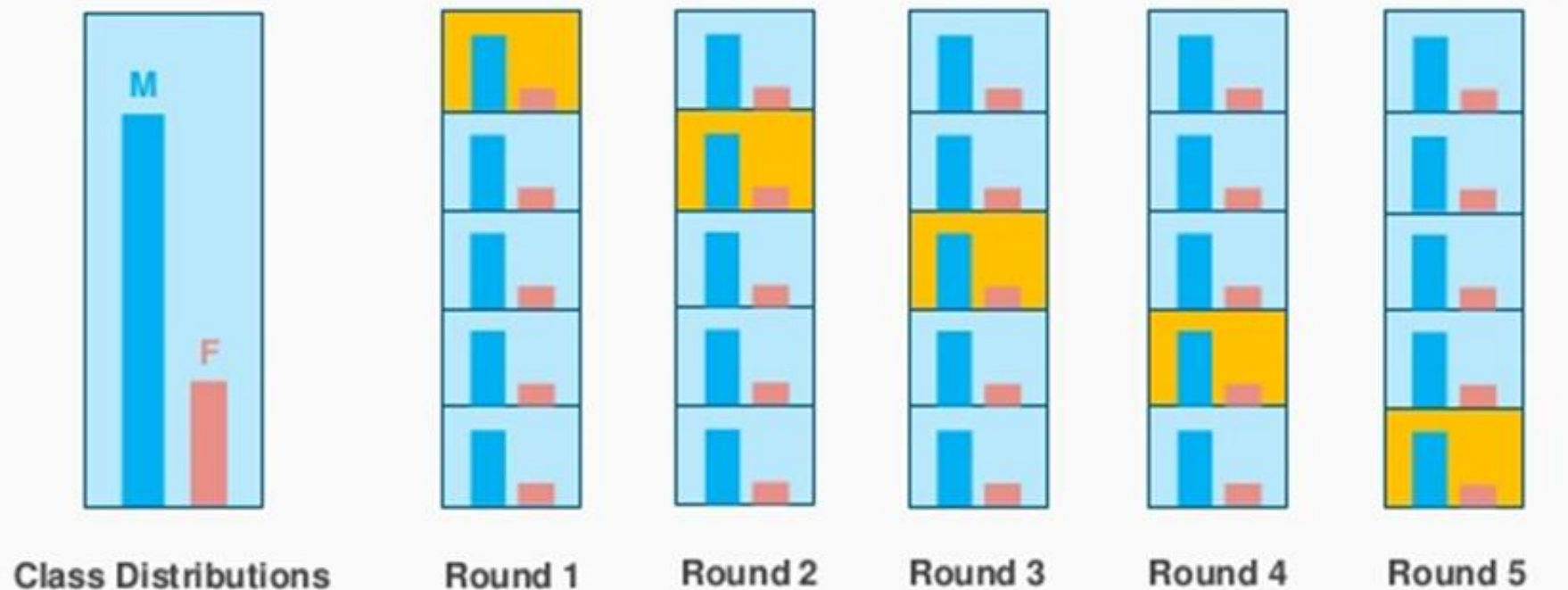
Example: k-Fold Cross-Validation





Validación Cruzada Estratificada (Stratified k-Fold Cross-Validation)

Stratified K-fold Cross Validation ($K = 5$)



Validación Cruzada dejando uno fuera:

Leave-One-Out Cross-Validation (LOOCV)

 Training Set  Validation Set



...



Comparación de Modelos: La Búsqueda del Campeón

Comparación de Modelos: La Búsqueda del Campeón

La selección del **modelo óptimo** no es universal, depende intrínsecamente del **problema**, los **datos** y los **objetivos** del negocio.

Compara modelos utilizando las **métricas de evaluación** apropiadas para el **contexto** del problema.

Considera la **interpretabilidad del modelo** frente a su rendimiento. Modelos más complejos (e.g., ensambles, SVM con kernels no lineales) pueden ofrecer mayor precisión a costa de menor transparencia.

Realiza **pruebas estadísticas** (e.g., t-tests pareados) para determinar si las diferencias en el rendimiento entre modelos son significativas.

Optimización y Validación: Afinamiento para el Éxito

Validación Cruzada y Búsqueda de Hiperparámetros: La Danza de la Optimización

- Los **hiperparámetros** controlan la arquitectura y el **comportamiento** de un modelo de Machine Learning.
- Encontrar la **combinación óptima es crucial** para maximizar el rendimiento.
- La **validación cruzada** se integra inherentemente en la **búsqueda de hiperparámetros** para evaluar el rendimiento de diferentes configuraciones sin sobreajustar (evitar *overfitting*) al conjunto de prueba.

Grid Search: Exploración Exhaustiva

Define un **conjunto discreto de valores** para cada hiperparámetro que deseas optimizar.

Grid Search evalúa exhaustivamente **todas las combinaciones** posibles de estos valores utilizando validación cruzada.

Garantiza la exploración completa del espacio de búsqueda definido, pero puede ser **computacionalmente prohibitivo** para **espacios de búsqueda grandes**.

Random Search: Exploración Eficiente

Define **rangos o distribuciones de probabilidad** para cada hiperparámetro.

Random Search muestrea **aleatoriamente** un número predefinido de combinaciones de hiperparámetros y las evalúa utilizando validación cruzada.

Más eficiente que Grid Search en espacios de búsqueda de **alta dimensión**, ya que es más probable que encuentre combinaciones valiosas incluso sin explorar exhaustivamente cada punto.

Grid Search vs Random Search

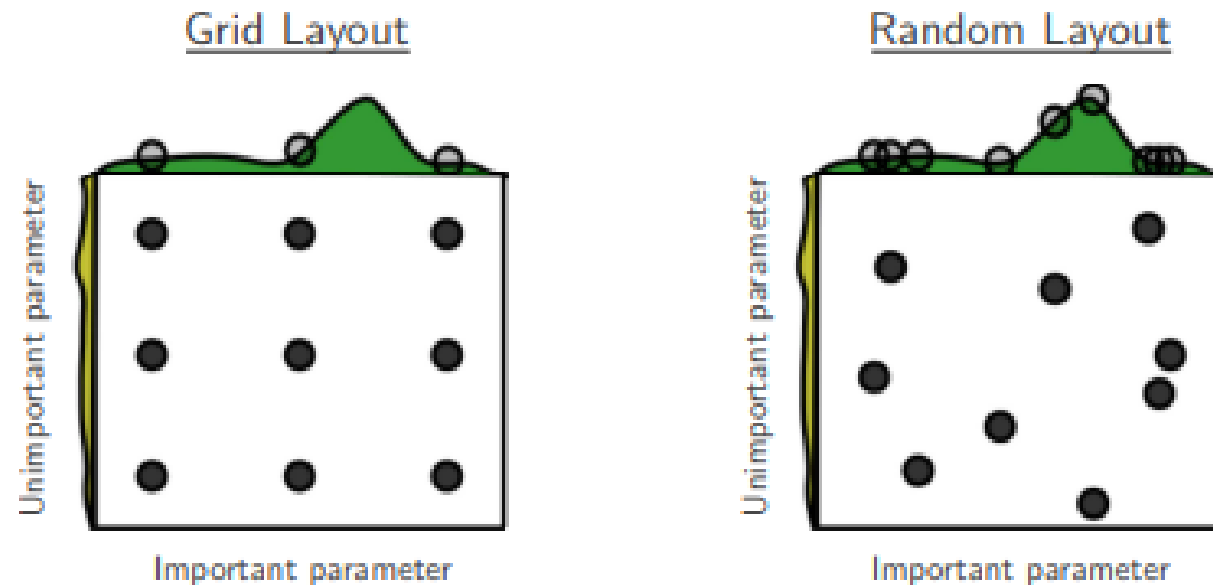


Figure 1: Grid and random search of nine trials for optimizing a function $f(x, y) = g(x) + h(y) \approx g(x)$ with low effective dimensionality. Above each square $g(x)$ is shown in green, and left of each square $h(y)$ is shown in yellow. With grid search, nine trials only test $g(x)$ in three distinct places. With random search, all nine trials explore distinct values of g . This failure of grid search is the rule rather than the exception in high dimensional hyper-parameter optimization.

scikit-learn

Machine Learning in Python

Getting Started

Release Highlights for 1.6

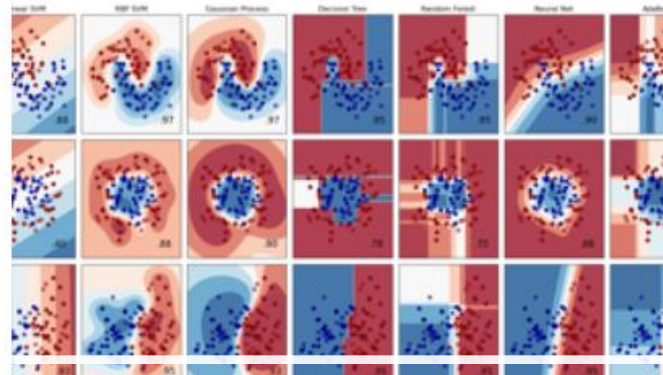
- Simple and efficient tools for predictive data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

Classification

Identifying which category an object belongs to.

Applications: Spam detection, image recognition.

Algorithms: [Gradient boosting](#), [nearest neighbors](#), [random forest](#), [logistic regression](#), and [more...](#)



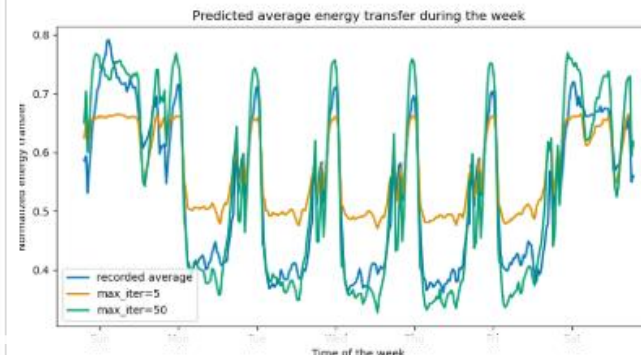
Examples

Regression

Predicting a continuous-valued attribute associated with an object.

Applications: Drug response, stock prices.

Algorithms: [Gradient boosting](#), [nearest neighbors](#), [random forest](#), [ridge](#), and [more...](#)



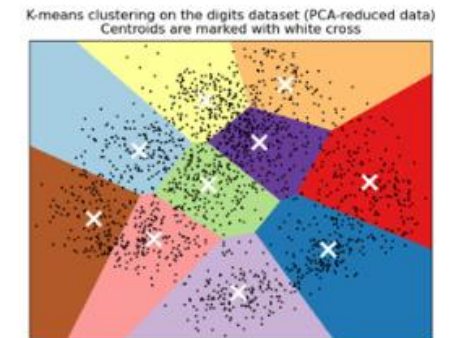
Examples

Clustering

Automatic grouping of similar objects into sets.

Applications: Customer segmentation, grouping experiment outcomes.

Algorithms: [k-Means](#), [HDBSCAN](#), [hierarchical clustering](#), and [more...](#)



Examples

Dimensionality reduction

Reducing the number of random variables to consider.

Applications: Visualization, increased efficiency.

Model selection

Comparing, validating and choosing parameters and models.

Preprocessing

Feature extraction and normalization.

Applications: Transforming input data such as text for

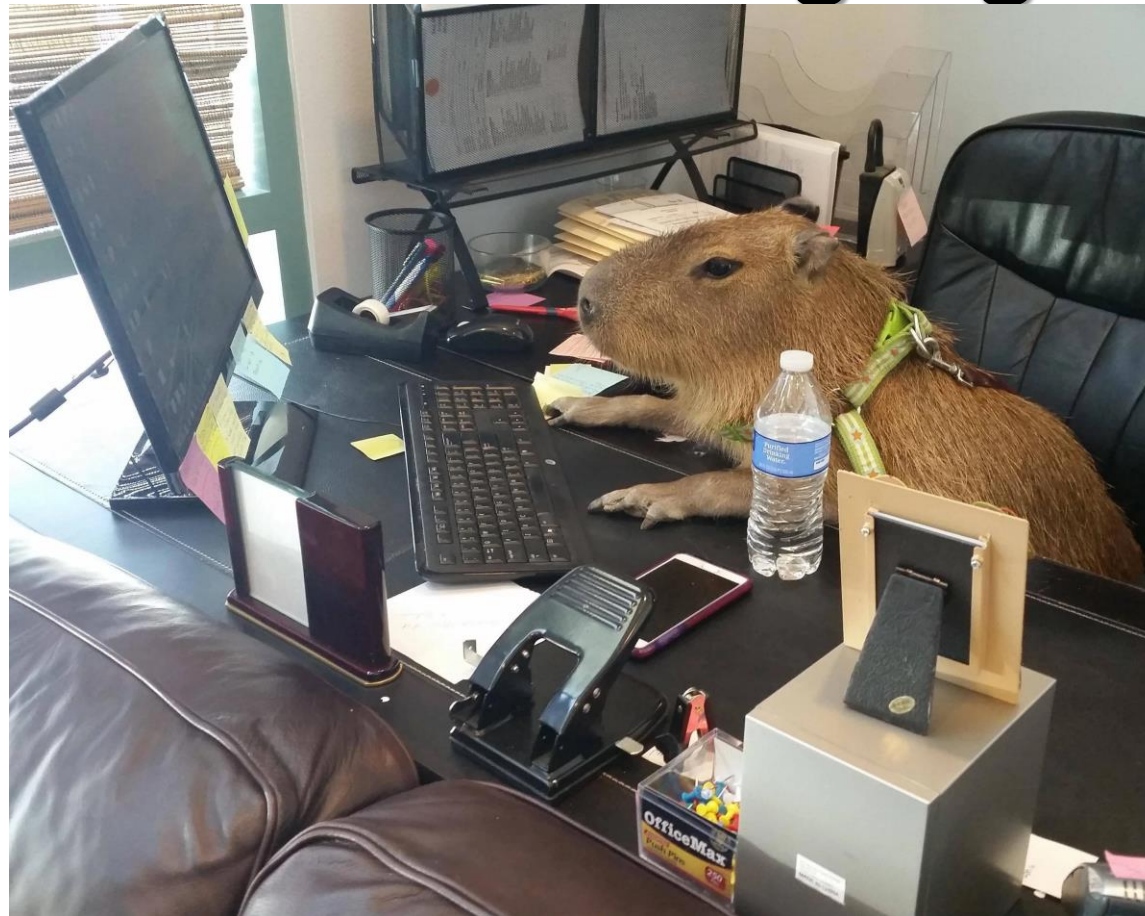
Conclusiones...

La **evaluación rigurosa**, la **validación robusta** y la optimización estratégica de **hiperparámetros** son pilares fundamentales para construir modelos de clasificación de alto rendimiento y confiables en el mundo de **la Inteligencia Artificial**.

La elección de las **métricas**, las técnicas de validación y los métodos de búsqueda debe estar guiada por una **comprensión profunda del problema** y las **características de los datos**.

Ejercicio práctico

colab.research.google.com



Ejercicio práctico: Google Colaboratory (*Colabs*)

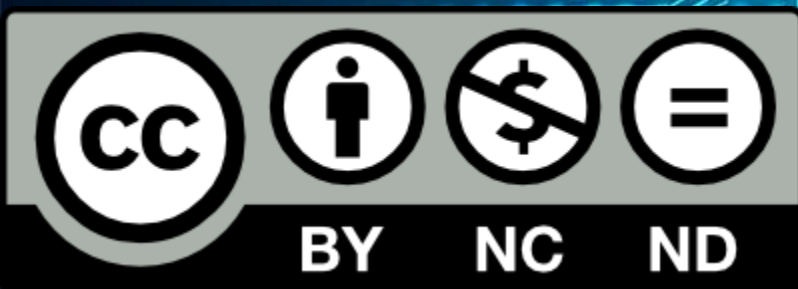
- Página oficial: <https://colab.google/>
- Abrir Colab (incluye tutorial): <https://colab.research.google.com/>
- Guía para EDA: https://colab.research.google.com/github/Tanu-N-Prabhu/Python/blob/master/Exploratory_data_Analysis.ipynb
- Guía / tutorial para Selección de características con **scikit-learn**: <https://www.datacamp.com/tutorial/feature-selection-python>



Referencias

- J. Bergstra y Y. Bengio, “Random Search for Hyper-Parameter Optimization”, Journal of Machine Learning Research, vol. 13, núm. 10, pp. 281–305, 2012, Consultado: el 8 de abril de 2025. [En línea]. Disponible en: <http://jmlr.org/papers/v13/bergstra12a.html>
- “Leave-one-out-cross-validation (LOOCV) | Python”. Consultado: el 8 de abril de 2025. [En línea]. Disponible en: <https://campus.datacamp.com/courses/model-validation-in-python/cross-validation?ex=10>
- “Clasificación | Machine Learning | Google for Developers”. Consultado: el 26 de marzo de 2025. [En línea]. Disponible en: <https://developers.google.com/machine-learning/crash-course/classification?hl=es-419>
- “scikit-learn: Machine Learning in Python”. Consultado: el 20 de febrero de 2025. [En línea]. Disponible en: <https://scikit-learn.org/stable/>
- Información e ideas presentadas basadas en el conocimiento general de modelos de lenguaje de IA. Gemini 2.9 Flash. Consultado: el 8 de abril de 2025. [En línea].

Machine Learning



Susana Medina Gordillo

susana.medina@correounivalle.edu.co

