PREPROCESAMIENTO DE DATOS
CLASE 1

AGENDA

OBJETIVO GENERAL: DESARROLLAR HABILIDADES PARA LA LIMPIEZA, TRANSFORMACIÓN, VISUALIZACIÓN Y ANÁLISIS EXPLORATORIO DE DATOS. Y PARA LA VALIDACIÓN Y EVALUACIÓN DE MODELOS PREDICTIVOS

- IMPORTANCIA DE LOS DATOS (MG. ENCINAS)
- 1. DEFINICION DEL PROBLEMA
- 2. RECOLECCION DE DATOS PARA PROYECTOS DE IA.
- 3. LIMPIEZA DE DATOS Y TRATAMIENTO DE VARIABLES.
- 4. ANÁLISIS EXPLORATORIO DE DATOS (EDA).
- 5. PREPARACIÓN PARA MODELADO Y EVALUACIÓN DE DESEMPEÑO.
- 6. INTERPRETABILIDAD Y EXPLICABILIDAD DE MODELOS.

PREPARACIÓN DE DATOS PARA PROYECTOS DE IA

OBJETIVOS:

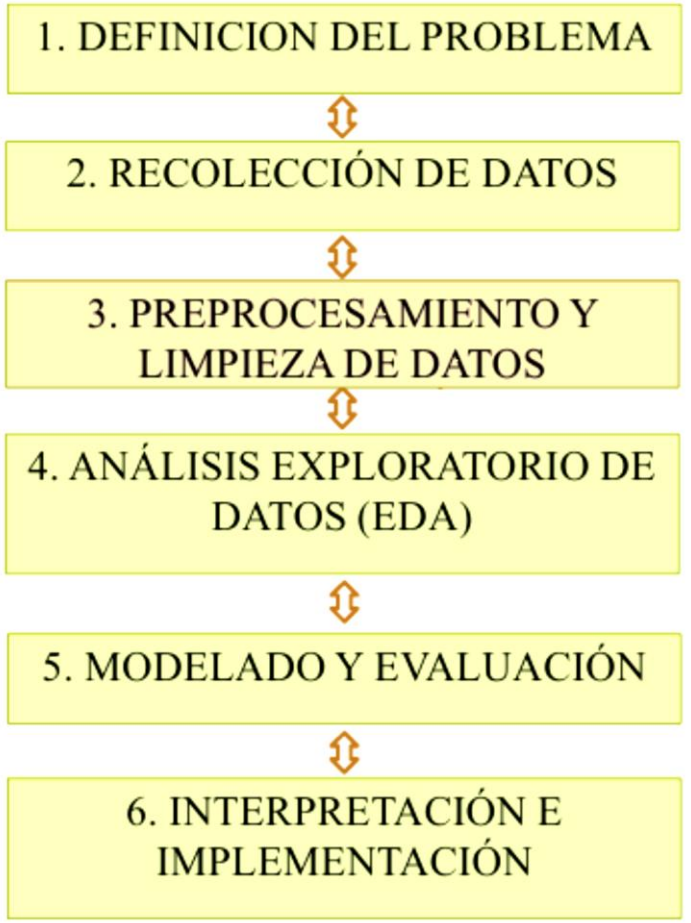
- COMPRENDER EL ROL DEL PREPROCESAMIENTO EN EL CICLO DE IA.
- RECONOCER LOS DISTINTOS TIPOS DE DATOS Y SUS FUENTES.
- APRENDER A CARGAR, INSPECCIONAR Y DESCRIBIR DATASETS EN PYTHON.
- IDENTIFICAR VALORES FALTANTES, DUPLICADOS Y ERRORES COMUNES.

FLUJO DE TRABAJO

El éxito de un modelo no depende solo del algoritmo, sino de cómo se entienden, preparan y utilizan los datos.

Se trata de un ciclo continuo de análisis, aprendizaje y mejora.

Cada etapa influye en la siguiente y posiblemente en la anterior:



FLUJO DE TRABAJO

Etapas del Análisis de Datos:

ETAPA 1:

ENMARCAR EL PROBLEMA

HACER LAS PREGUNTAS ADECUADAS

¿Cuál es el objetivo de la empresa?

¿Qué queremos estimar o predecir?

FLUJO DE TRABAJO

¿Qué queremos resolver?

- Formular claramente el objetivo del proyecto: clasificación, regresión, predicción, detección, etc.
  - Identificar variable objetivo (Y) y variables explicativas (X).
- Comprender el contexto y la aplicación real (energía, agua, salud, agricultura, etc.).
- Evaluar criterios de éxito: precisión, costo, interpretabilidad, tiempo, recursos.

FLUJO DE TRABAJO

RECOLECCIÓN DE DATOS

ETAPA 2:

RECOLECCION DE DATOS

¿Qué recursos tenemos para obtener datos?

¿Qué información es relevante?

Limpiar y filtrar los datos para su posterior análisis

FLUJO DE TRABAJO

ETAPA 2: Recolección de datos

- **Fuentes:** sensores, encuestas, APIs, repositorios (Kaggle, GitHub), bases institucionales.
- **Tipos de datos:** estructurados (tablas), no estructurados (texto, imágenes, audio), categóricos, numéricos, texto, fechas.
- **Aspectos clave:**
  - o Calidad (precisión, consistencia).
  - o Cantidad (suficiente para entrenar).
  - o Representatividad (cubre todos los casos posibles).

FLUJO DE TRABAJO

ETAPA 3: Preprocesamiento

Usualmente los datos se **presentan en formatos no óptimos (o incluso inadecuados) p**ara ser procesados por el modelo.

Fases del preprocesamiento:

1. **Limpieza: no siempre obtenemos un conjunto de datos sin errores de medición,** los cuales se pueden **identificar por outliers o datos atípicos** o nulos.
2. **Normalización: l*os modelos de Aprendizaje Automático son sensibles a la escala en que vienen los datos***.
   Por ejemplo, si estamos con variables climatológicas **no tenemos las mismas escalas en la medición de temperatura (25 °C) y la presión (1000 hPa)** por lo que requerimos normalizar los datos.
1. **Pureza: L*as características (feature) seleccionadas están correlacionadas***, y por lo tanto, **son redundantes** para extraer información significativa a partir de ellas.
2. **División de datos:** se dividen los datos totales en subconjuntos de **datos para entrenamiento y datos para prueba (testeo).**

FLUJO DE TRABAJO

ETAPA 3: Preprocesamiento

¿Por qué es importante el preprocesamiento de los datos?

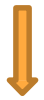
- La calidad del modelo nunca será mejor que la calidad de los datos.
- Reduce errores y sesgos.
- Aumenta la capacidad predictiva.
- Evita data leakage y sobreajuste.
- El 70–80 % del tiempo de un proyecto se dedica a esta fase .

FLUJO DE TRABAJO

Etapas del Análisis de Datos:

ETAPA 4:

ANALISIS EXPLORATORIO DE DATOS



Visualizar los datos

Localizar en los gráficos posibles tendencias, correlaciones o patrones

FLUJO DE TRABAJO

4. ANÁLISIS EXPLORATORIO DE DATOS (EDA)

Comprender el comportamiento de los datos:

- Explorar distribuciones y relaciones entre variables.
- Identificar outliers, patrones y correlaciones.
- Visualizar para generar hipótesis.
- Usar herramientas como pandas, matplotlib, seaborn

FLUJO DE TRABAJO

4. ANÁLISIS EXPLORATORIO DE DATOS (EDA)

¿Por qué es importante el EDA?
•Permite comprender el dominio antes del modelado.
•Informa decisiones sobre qué variables incluir o transformar.
•Detecta problemas ocultos (distribuciones sesgadas, correlaciones fuertes).

FLUJO DE TRABAJO

MODELADO Y EVALUACIÓN:

ETAPA 5:

MODELIZAR Y
EVALUAR LOS DATOS

Utilizar algún algoritmo innovador (de acuerdo al problema) para crear el modelo

Evaluar el modelo

FLUJO DE TRABAJO

ETAPA 5: MODELADO Y EVALUACIÓN
Entrenando y seleccionando un modelo

En este paso, es esencial **buscar cuál es el mejor modelo para nuestro conjunto de datos, esto significa encontrar el mejor algoritmo y la mejor configuración de ese algoritmo** para obtener un agente con alta performance.

Para eso, en primer lugar, debemos evaluar, **durante el proceso de entrenamiento, a nuestro modelo.**

Para **evaluar el desempeño de un modelo** de Aprendizaje Automático **se utilizan generalmente diferentes métricas estadísticas**. Esto nos permite **comparar entre dos o más modelos cuál es el óptimo.**

FLUJO DE TRABAJO

ETAPA 5: MODELADO Y EVALUACIÓN:

Entrenamiento del modelo

- Selección del algoritmo (Regresión, Árboles, Gradient Boosting, Redes Neuronales, etc.).
- División de datos: entrenamiento / validación / test.
- Ajuste de hiperparámetros y comparación de modelos.
- Uso de métricas de desempeño:
    - Clasificación: Accuracy, Precision, Recall, F1.
    - Regresión: RMSE, MAE, R^2

FLUJO DE TRABAJO

ETAPA 5: MODELADO Y EVALUACIÓN

Optimización de hiperparámetros

- Se debe tener en cuenta, **a la hora del entrenamiento para la selección del mejor modelo**, **el ajuste de los hiperparámetros**,
- **Los modelos de Aprendizaje Automático de las librerías no se encuentran con la configuración ideal** por defecto.

Por lo que debemos **buscar la mejor configuración**, es decir, la **mejor combinación de los hiperparámetros** que **determinen el modelo óptimo** (con las **mejores métricas estadísticas).**

Para esto se destacan dos **técnicas: Grid Search (Búsqueda Exhaustiva), Random Search (Búsqueda Aleatoria), o Bayesian Optimization (Optimización Bayesiana).**

FLUJO DE TRABAJO

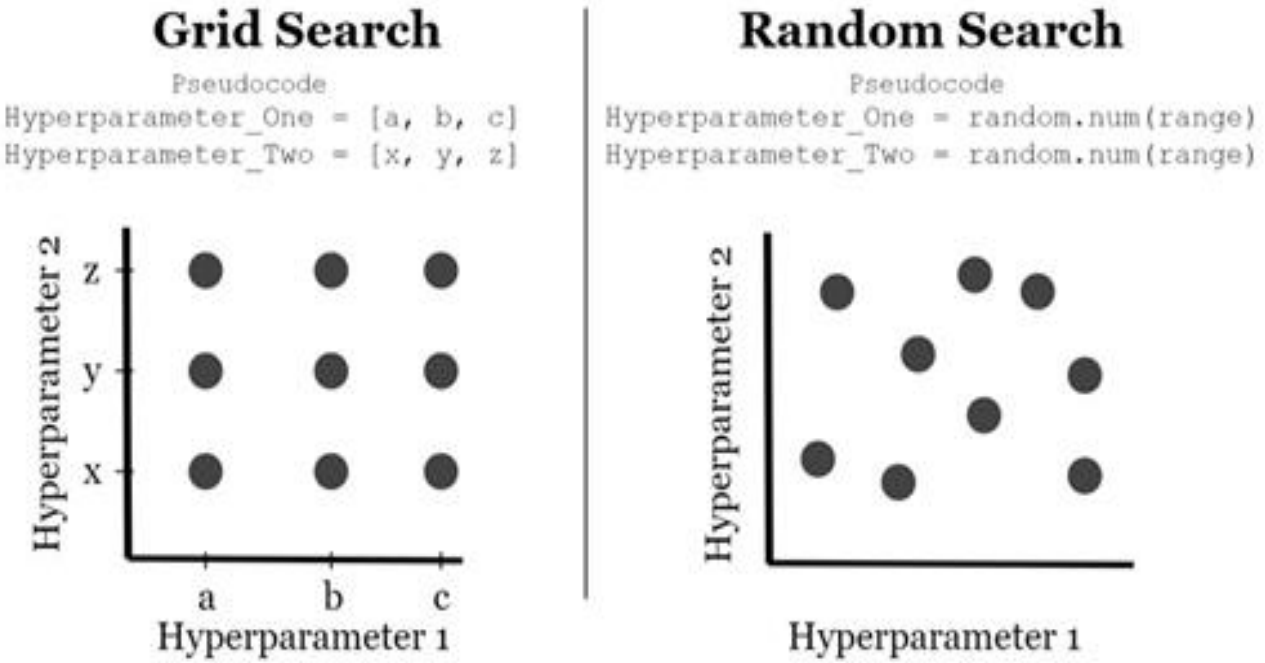
ETAPA 5: MODELADO Y EVALUACIÓN

Optimización de hiperparámetros

- **Búsqueda Exhaustiva (Grid Search): genera una grilla con los hiperparámetros y realiza todas las posibles combinaciones en esa grilla.**

- **Búsqueda Aleatoria (Random Search): Define distribuciones para cada hiperparámetro.**

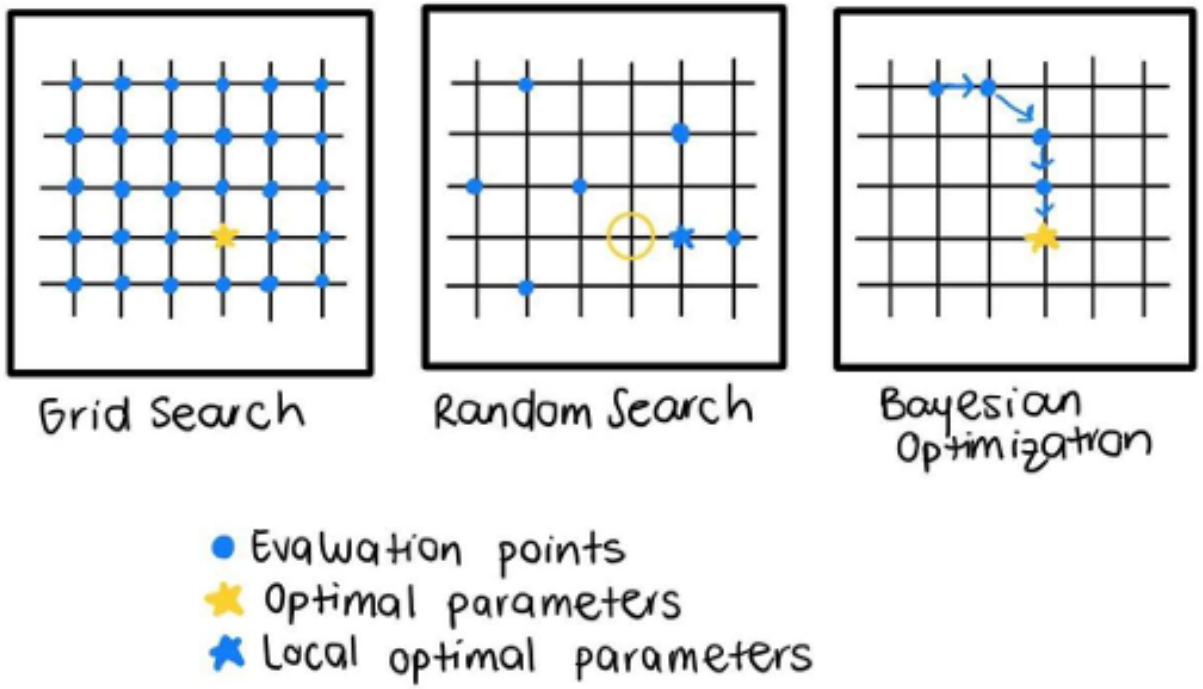
  La diferencia clave es que, en la búsqueda aleatoria, no todos los valores se prueban y los valores probados se seleccionan al azar.



FLUJO DE TRABAJO

ETAPA 5: MODELADO Y EVALUACIÓN

Optimización de hiperparámetros



FLUJO DE TRABAJO

ETAPA 5. MODELADO Y EVALUACIÓN

Métricas para Modelos de Clasificación:

$$Accuracy\ [A] = \frac{(TP + TN)}{(TP + TN + FP + FN)}$$

$$Precisión\ [P] = \frac{TP}{(TP + FP)}$$

$$Recall\ [R] = \frac{TP}{(TP + FN)}$$

$$F1_{measure} = 2.\frac{P.R}{(P + R)}$$

$TP = verdadero\ positivo$

$TN = verdadero\ negativo$

$FP = falso\ positivo$

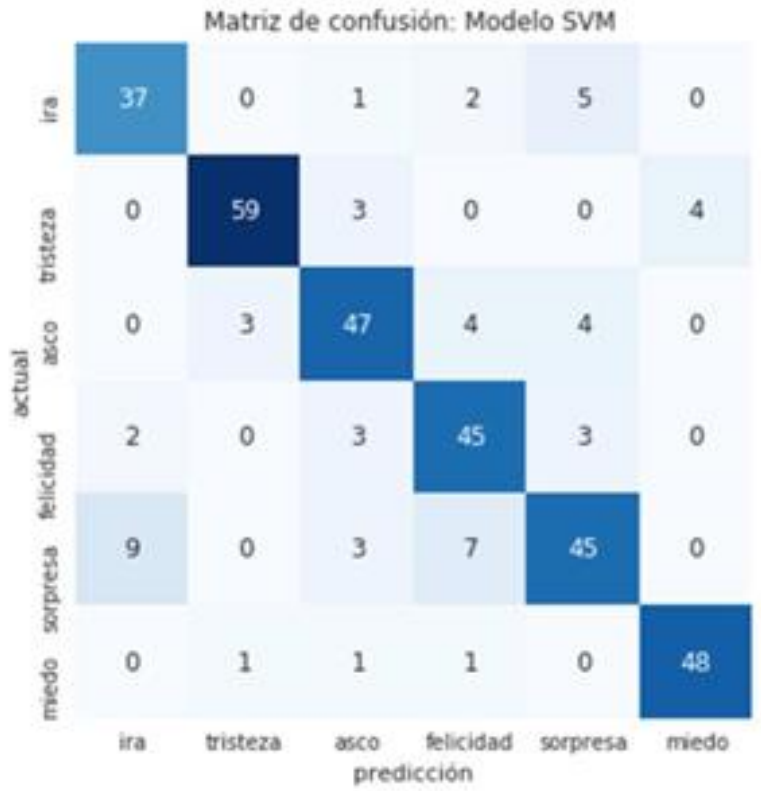
$FN = falso\ negativo$

FLUJO DE TRABAJO

ETAPA 5. MODELADO Y EVALUACIÓN

Métricas para Modelos de Clasificación:

Ejemplo de Matriz de Confusión:

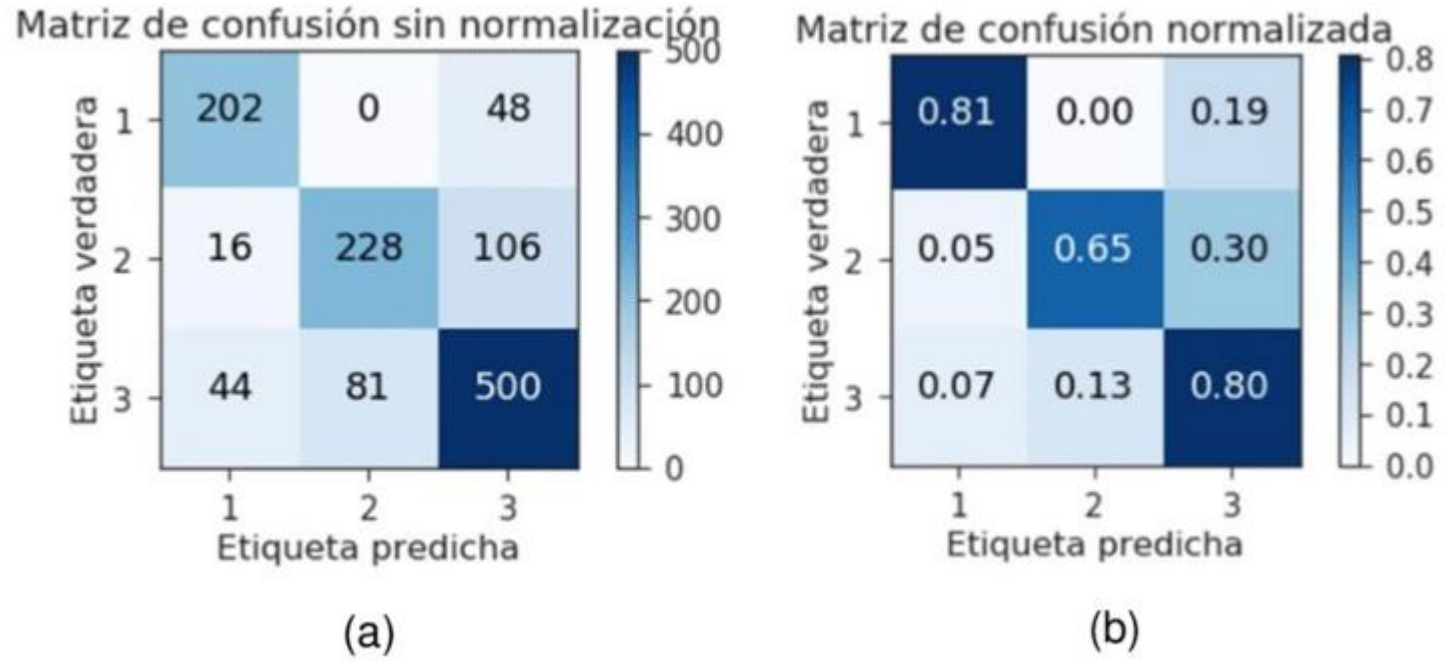


FLUJO DE TRABAJO

ETAPA 5. MODELADO Y EVALUACIÓN

Métricas para Modelos de Clasificación:

Ejemplo de Matriz de Confusión:



FLUJO DE TRABAJO

ETAPA 5. MODELADO Y EVALUACIÓN

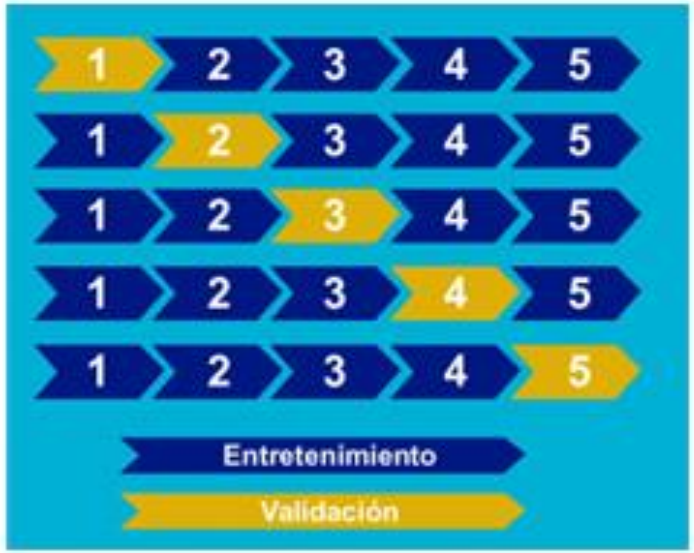
Métricas para Modelos de Regresión:

Coeficiente de Determinación: $$R^2 = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y}_i)^2}$$

Error Absoluto Medio: $$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

Raíz Cuadrática del Error Cuadrático Medio $$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}}$$

FLUJO DE TRABAJO



ETAPA 5. MODELADO Y EVALUACIÓN

Evaluación y validación

- Validación cruzada (cross-validation).
- Prevención del sobreajuste.
- Comparación entre modelos.
- Interpretación de métricas como apoyo a la toma de decisiones

FLUJO DE TRABAJO

ETAPA 5. MODELADO Y EVALUACIÓN

Validación cruzada

La validación cruzada (cross-validation), apunta a **garantizar que los resultados** obtenidos son **independientes de la partición entre datos de entrenamiento y prueba**.

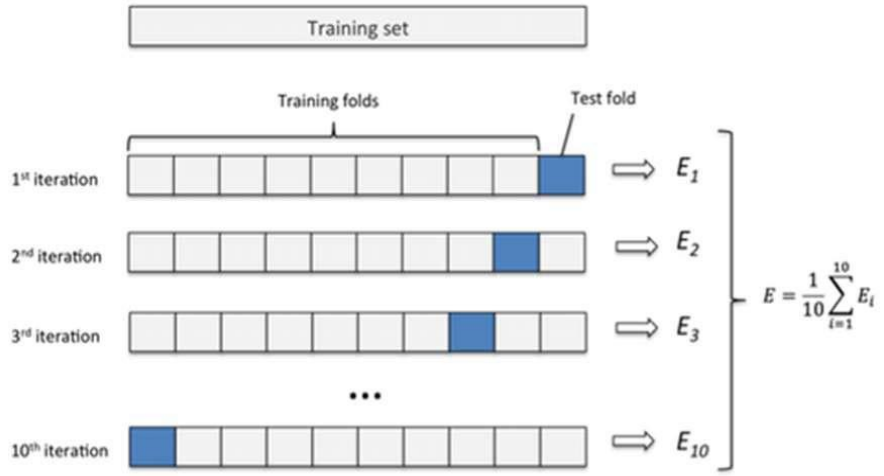
FLUJO DE TRABAJO

ETAPA 5. MODELADO Y EVALUACIÓN

Validación cruzada

Esta técnica d**ivide el conjunto de datos de entrenamiento en subconjuntos menores de entrenamiento y validación**, con lo que se estima la **capacidad de generalización del modelo**. En otras palabras, estima **qué tan correctas serán las predicciones de salida obtenidas cuando el modelo se alimenta con datos nuevos.**

A continuación, **se repite el proceso K veces y se calcula el rendimiento medio del modelo dividiendo la suma de las métricas obtenidas entre el número K de interacciones**.



FLUJO DE TRABAJO

Análisis de los resultados:

ETAPA 6:

INTERPRETACION E IMPLEMENTACION

¿Qué resultados hemos obtenido?

¿Qué hemos aprendido?

¿Los resultados tienen sentido?

FLUJO DE TRABAJO

ETAPA 6: Análisis de los resultados

- Interpretar el modelo: ¿Qué variables influyen más?
- Herramientas de interpretabilidad: SHAP, LIME, Feature Importance.
- Integración del modelo en sistemas reales (APIs, dashboards, IoT).
- Monitoreo del desempeño y actualización periódica.

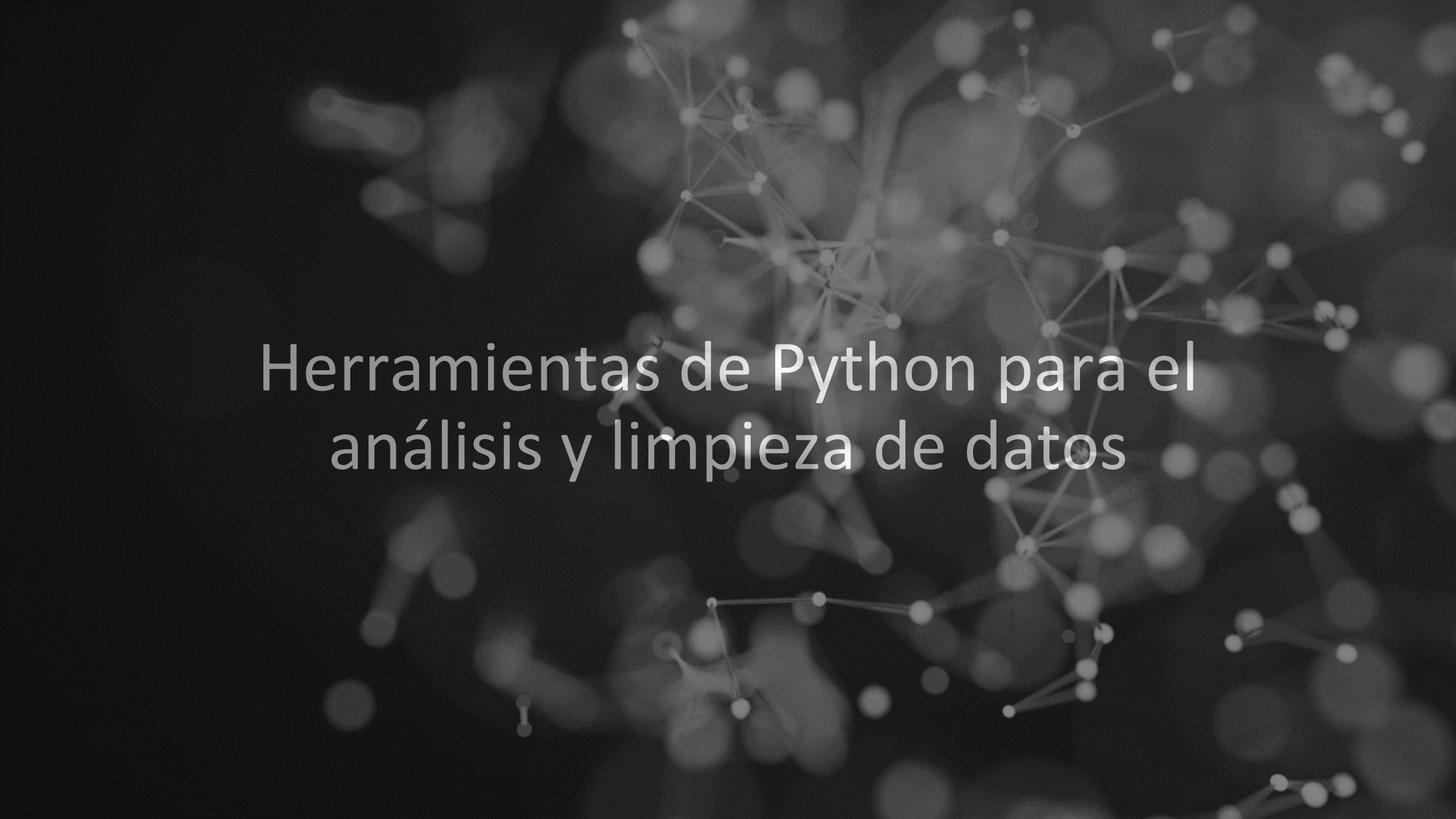
FLUJO DE TRABAJO

ETAPA 6: Evaluando modelos y prediciendo con datos nuevos

Una vez que hemos **seleccionado y ajustado un modelo** a nuestro conjunto de **datos de entrenamiento**, **podemos usar los datos de prueba para estimar el rendimiento del modelo** en los datos nuevos.

- Esto nos permite la **estimación del error de generalización del modelo**, o evaluarlo utilizando alguna otra métrica.

- **Si estamos satisfechos con el valor de la métrica** obtenida, **podremos usar el modelo para realizar predicciones** con los datos futuros.


Herramientas de Python para el
análisis y limpieza de datos

HERRAMIENTAS

- **Pandas:** Librería de código abierto que provee funciones de carga y análisis de datos, entre otras funcionalidades.
- **Colaboratory o Google Colab** **Permite a cualquier usuario escribir y ejecutar código arbitrario de Python en el navegador.** Es especialmente adecuado para tareas de aprendizaje automático, análisis de datos y educación.
- **TensorFlow** **es una biblioteca de código abierto para aprendizaje automático** desarrollada por Google.
- **Keras** **Es una biblioteca de redes neuronales de código abierto escrita en Python** Está diseñado para construir por **bloques la arquitectura de redes neuronales, incluyendo redes convolucionales y recurrentes,** que son las que permiten, junto a los bloques “más tradicionales”, **entrenar modelos Deep Learning.**

HERRAMIENTAS

- **Scikit-learn** Es una biblioteca de Python de código abierto para el **aprendizaje automático.**
- **Numpy** Es una biblioteca para el lenguaje de programación Python que da soporte para **crear vectores y matrices grandes multidimensionales, junto con una gran colección de funciones matemáticas** de alto nivel para operar con ellas.
- **Scipy** Es una biblioteca libre y de código abierto para Python. Se compone de **herramientas y algoritmos matemáticos.**
- **Matplotlib** es una biblioteca para la **generación de gráficos a partir de datos contenidos en listas o arrays** en el lenguaje de programación Python.

De datos crudos a datos utilizables

•En esta etapa el objetivo no es modelar, sino garantizar que los datos sean coherentes, completos y sin redundancias.

De datos crudos a datos utilizables
TIPOS DE DATOS

Tipo de dato	Ejemplo	Formato	Uso típico
Numérico	Temperatura, edad	int, float	Modelos predictivos
Categórico	Género, provincia	object	Clasificación
Temporal	Fecha, hora	datetime	Series de tiempo
Texto	Comentarios, reseñas	string	NLP
Geoespacial	Latitud, longitud	float	Mapas, sensores

De datos crudos a datos utilizables
FUENTES DE DATOS

- Archivos locales (.csv, .xlsx, .json, .txt)
- Bases de datos SQL
- APIs y servicios web
- Repositorios abiertos:
  - Kaggle
  - GitHub
  - Google Drive / Colab

De datos crudos a datos utilizables
CARGAR UN DATASET

- **Con Pandas se pueden cargar archivos CSV o XLS creando un dataframe de la siguiente manera:**

```
import pandas as pd

iris = pd.read_csv("iris.csv")
```

(Manejo de archivos y lectura de datos. Librerías científicas).

De datos crudos a datos utilizables
Inspección inicial de datos con pandas

	Función	Propósito principal	Salida / Tipo de resultado	Qué información brinda
(Mirar los datos)	***df.head()***	Visualizar las **primeras filas** del DataFrame	Muestra por defecto las **primeras 5 filas**	Permite una **vista rápida** del contenido, nombres de columnas y tipos de datos visuales
(Dimensionar los datos)	***df.shape***	Conocer el **tamaño del dataset**	Tupla: *(n_filas, n_columnas)*	Indica cuántas filas y columnas tiene el DataFrame
(Comprender la estructura)	***df.info()***	Obtener un **resumen estructural** del DataFrame	Descripción textual (no devuelve un objeto)	Muestra el tipo de dato de cada columna, cantidad de valores no nulos y uso de memoria
(Resumir estadísticamente los datos)	***df.describe()***	Generar **estadísticas descriptivas básicas**	DataFrame con estadísticas	Calcula conteo, media, desviación, mínimos, cuartiles y máximos de las variables numéricas (y también categóricas con include='all')

De datos crudos a datos utilizables
Inspección inicial de datos con pandas

- El método `value_counts()` cuenta la cantidad de ítems en la columna seleccionada.
- Por ejemplo, para la columna del dataset iris:

```
iris['name'].value_counts()
```

```
versicolor    50
setosa        50
virginica     50
```

De datos crudos a datos utilizables
DETECCION DE VALORES FALTANTES

Tipo de problema	Descripción	Comandos principales de detección	Comandos comunes para el tratamiento
Valores faltantes	Celdas sin dato o con valores nulos (NaN) que interrumpen cálculos o entrenamientos	***df.isnull()*** → muestra True/False ***df.isnull().sum()*** → cuenta los nulos por columna	***df.dropna()*** → elimina filas/columnas con nulos ***df.fillna(valor)*** → reemplaza con un valor ***df['col'].fillna(df['col'].mean())*** → imputa con la media

De datos crudos a datos utilizables
DETECCION DE DUPLICADOS E INCONSISTENCIAS

Tipo de problema	Descripción	Comandos principales de detección	Comandos comunes para el tratamiento
Duplicados	Filas idénticas repetidas (mismo valor en todas o algunas columnas)	***df.duplicated()*** → indica True/False ***df.duplicated().sum()*** → cantidad de duplicados	***df.drop_duplicates()*** → elimina duplicados (conserva la primera ocurrencia) ***df.drop_duplicates(subset=['columna'])*** → analiza sólo una columna
Inconsistencias	Errores de formato, tipográficos o de unidad (por ejemplo, “bsas”, “Bs.As.”, “Buenos Aires”)	***df['columna'].unique()*** → lista valores únicos ***df['columna'].value_counts()*** → frecuencia de cada valor ***df.dtypes*** → tipo de dato por columna	Reemplazo manual o con diccionarios: ***df['columna'].replace({'bsas':'Buenos Aires'})***

De datos crudos a datos utilizables
INDEXADO

Es el proceso mediante el cual accedemos, identificamos o seleccionamos elementos específicos dentro de un DataFrame o Serie utilizando etiquetas (nombres) o posiciones numéricas.
Es la forma en que pandas entiende dónde están los datos dentro de una tabla. Permite ubicar filas y columnas para leer, modificar o extraer información.

Acción	Descripción	Comando / Sintaxis	Qué devuelve
Indexado (por etiquetas)	Acceso a filas o columnas usando **nombres** (índices o etiquetas)	***df.loc[fila, columna]***	Subconjunto de filas y columnas por **nombre**
Indexado (por posición)	Acceso por **posición numérica** de filas o columnas	***df.iloc[fila, columna]***	Subconjunto de filas/columnas por **posición**

De datos crudos a datos utilizables
SELECCIÓN DE COLUMNAS

Acción	Descripción	Comando / Sintaxis	Qué devuelve
Selección de columnas simples	Acceso directo a una columna como Serie	***df['columna']***	Serie (columna individual)
Selección múltiple de columnas	Acceso a varias columnas específicas	***df[['col1', 'col2']]***	DataFrame con columnas seleccionadas

De datos crudos a datos utilizables
FILTRADO

Acción	Descripción	Comando / Sintaxis	Qué devuelve
Filtrado condicional	Selección de filas que cumplen una condición lógica	***df[df['columna'] > valor]***	Filas que cumplen la condición
Filtrado por valor único	Filtrar filas donde una variable tiene un valor específico	***df[df['species'] == 'setosa']***	Subconjunto del DataFrame
Filtrado combinado	Aplicar condiciones múltiples con operadores lógicos	***df[(cond1) & (cond2)]***	Filas que cumplen ambas condiciones

De datos crudos a datos utilizables
FILTRADO

Acción	Descripción	Comando / Sintaxis	Qué devuelve
Filtrado condicional	Selección de filas que cumplen una condición lógica	***df[df['columna'] > valor]***	Filas que cumplen la condición
Filtrado por valor único	Filtrar filas donde una variable tiene un valor específico	***df[df['species'] == 'setosa']***	Subconjunto del DataFrame
Filtrado combinado	Aplicar condiciones múltiples con operadores lógicos	***df[(cond1) & (cond2)]***	Filas que cumplen ambas condiciones