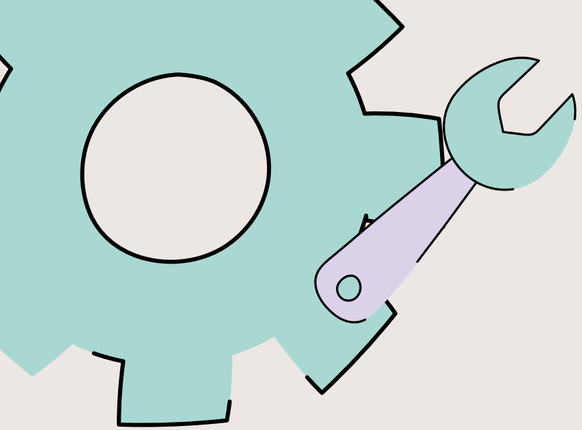
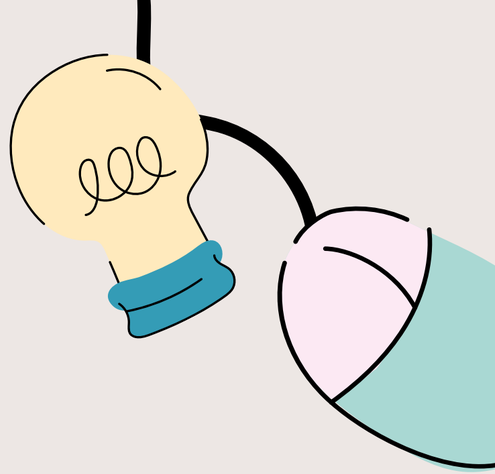
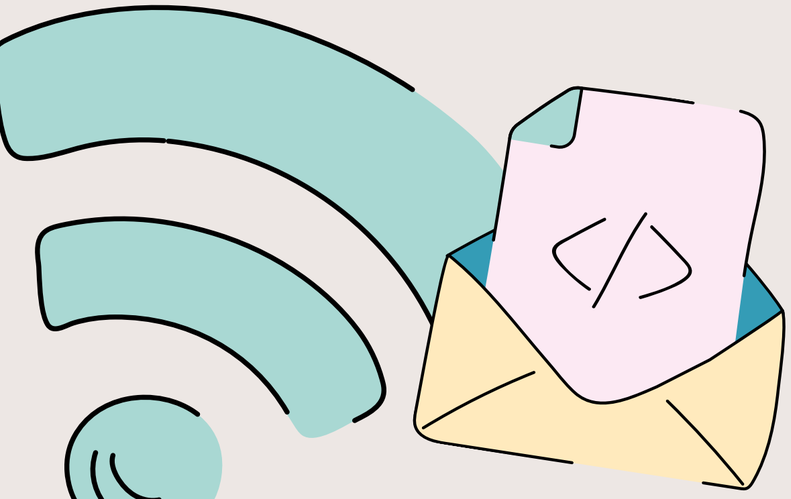


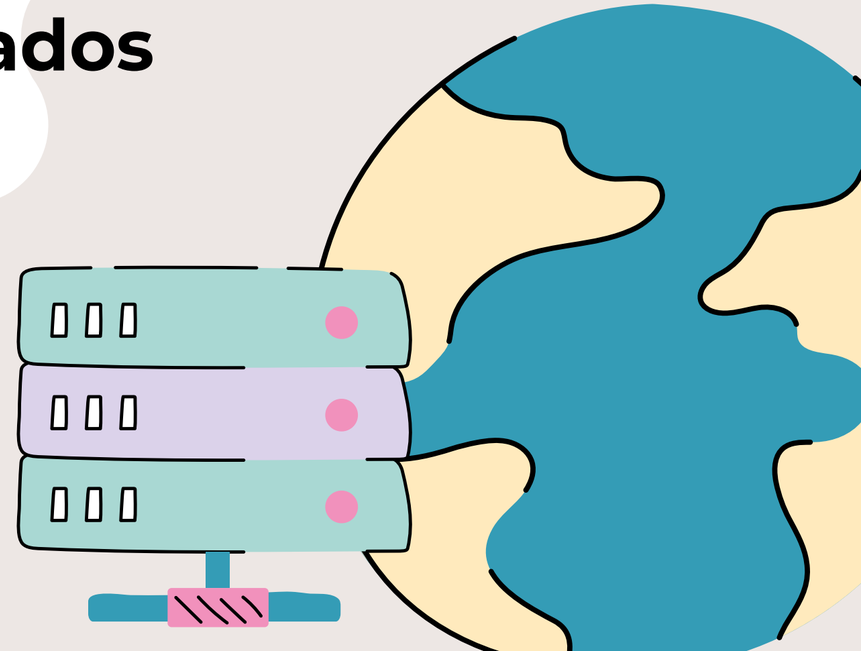


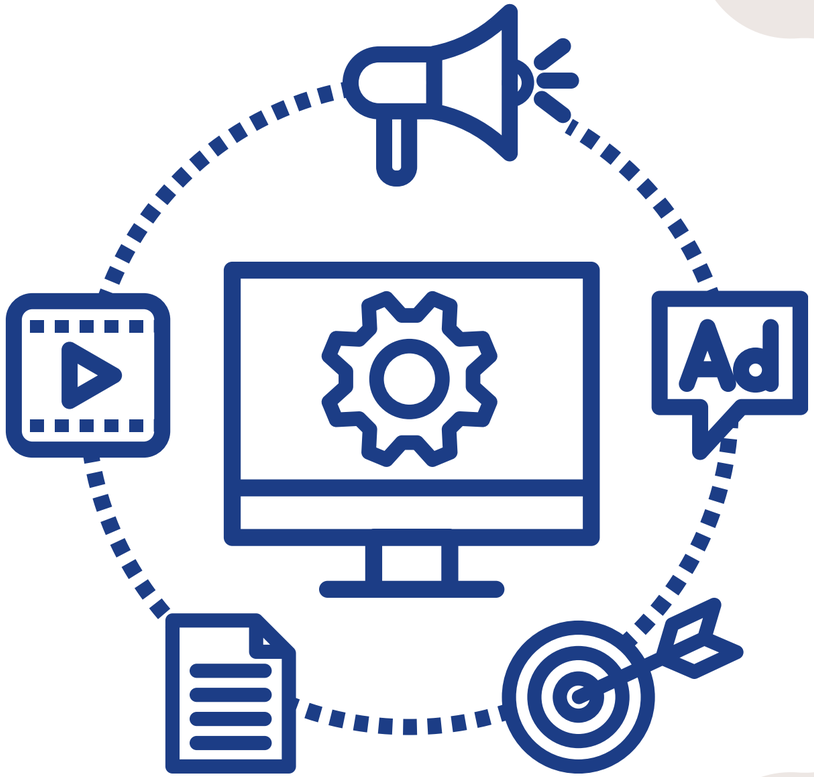
Proyectos de Machine Learning

Prácticas con datos integrados

Por Juan Duran





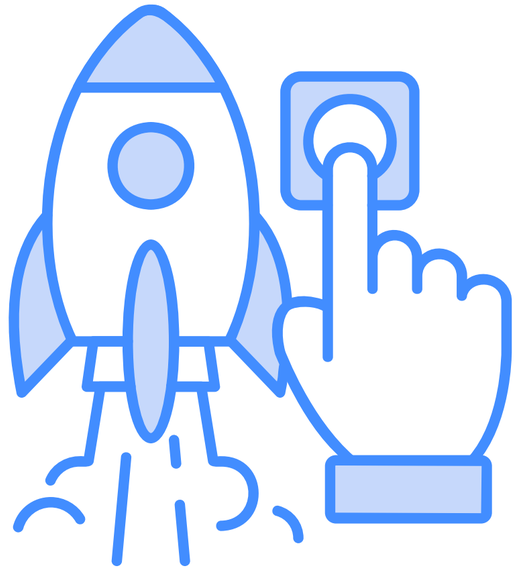


Contenidos

- Introducción 📘
- Puntos clave 📌
- Clasificación de flores iris 🌸
- Predicción de precios de viviendas 🏡
- Análisis de supervivencia en el Titanic 🚢
- Clasificación de pingüinos 🐧
- Predicción de duración de viajes en taxi 🚖
- Conclusiones 🏁

Introducción

La práctica de **proyectos** de machine learning es fundamental para entender y aplicar conceptos clave en el mundo **data**. En esta presentación, exploraremos varios proyectos de machine learning que se pueden realizar utilizando conjuntos de **datos integrados** en bibliotecas como **Seaborn** y **Scikit-learn**. Estos proyectos te permitirán practicar tanto la **visualización** de datos como la aplicación de técnicas de **machine learning**, sin necesidad de buscar datos adicionales. Para cada proyecto, he seleccionado un modelo específico, pero lo ideal es probar más modelos y evaluar el rendimiento de cada uno para obtener los mejores resultados.



Puntos clave

Datos Integrados

Utiliza conjuntos de datos integrados en **Seaborn** y **Scikit-learn** para practicar machine learning sin necesidad de buscar datos adicionales.

Visualización y modelado

Aprende a **visualizar** datos y a aplicar técnicas de **machine learning** para resolver problemas específicos

Evaluación de modelos

Evalúa el **rendimiento** de tus modelos utilizando **métricas** adecuadas para asegurar **resultados** precisos y confiables.

Clasificación de Flores Iris 🌸

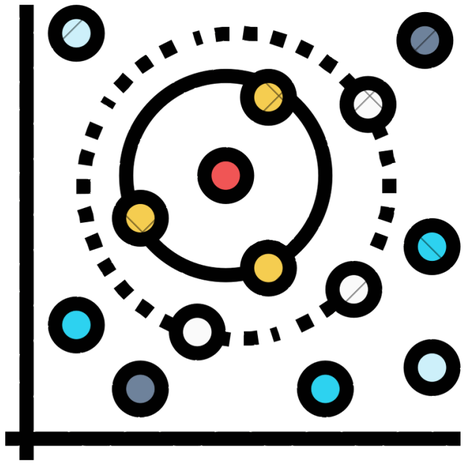
Objetivo: Clasificar las especies de flores Iris (Setosa, Versicolor, Virginica) utilizando características como el largo y ancho de los pétalos y sépalos.
Modelo: K-Nearest Neighbors
Pasos:

1. **Cargar** el conjunto de datos **Iris** desde Scikit-learn: Utiliza la función load_iris() para obtener los datos.
2. **Explorar** y **visualizar** los datos utilizando Seaborn: Crea gráficos de dispersión y de pares para entender las características.
3. **Dividir** los datos en conjuntos de entrenamiento y prueba: Utiliza train_test_split para separar los datos.
4. **Entrenar** un modelo de clasificación (**K-Nearest Neighbors**): Usa KNeighborsClassifier para entrenar el modelo.
5. **Evaluar** el rendimiento del modelo utilizando métricas como la **precisión** y la **matriz de confusión**: Calcula la precisión y crea una matriz de confusión para evaluar el modelo.

Clasificación de Iris

K-Nearest Neighbors

Visualización de Datos



Evaluación de Modelo

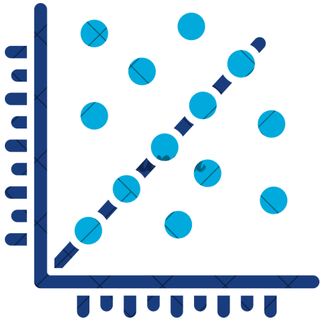
Precios de Viviendas 🏡

Objetivo: Predecir el precio de las viviendas en función de características como el tamaño, el número de habitaciones y la ubicación.
Modelo: Regresión Lineal
Pasos:

1. **Cargar** el conjunto de datos **Boston Housing** desde Scikit-learn: Utiliza la función load_boston() para obtener los datos.
2. **Limpiar** y **preprocesar** los datos: Maneja valores nulos y normaliza las características.
3. **Dividir** los datos en conjuntos de entrenamiento y prueba: Utiliza train_test_split para separar los datos.
4. **Entrenar** un modelo de regresión (**Regresión Lineal**): Usa LinearRegression para entrenar el modelo.
5. **Evaluar** el rendimiento del modelo utilizando **métricas** como el error cuadrático medio (**MSE**) y el coeficiente de determinación (**R^2**): Calcula el MSE y el R^2 para evaluar el modelo.

Predicción de Precios

Regresión Lineal



Datos de Viviendas

Evaluación de MSE

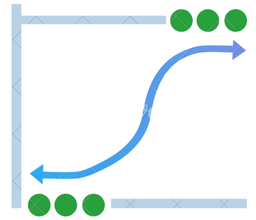
Supervivencia en el Titanic

Objetivo: Predecir la supervivencia de los pasajeros del Titanic en función de características como la edad, el género y la clase del boleto.
Modelo: Regresión Logística
Pasos:

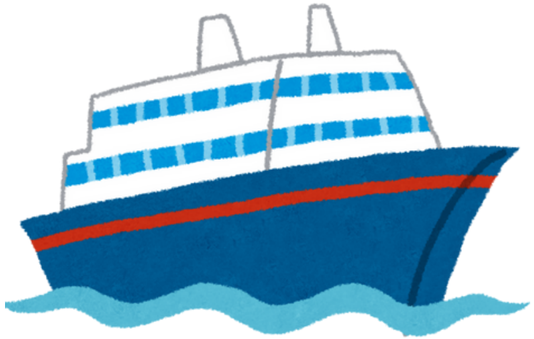
1. **Cargar** el conjunto de datos **Titanic** desde Seaborn: Utiliza la función load_dataset('titanic') para obtener los datos.
2. **Limpiar** y **preprocesar** los datos: Maneja valores nulos y convierte las variables categóricas en numéricas.
3. **Dividir** los datos en conjuntos de entrenamiento y prueba: Utiliza train_test_split para separar los datos.
4. **Entrenar** un modelo de clasificación (**Regresión Logística**): Usa LogisticRegression para entrenar el modelo.
5. **Evaluar** el rendimiento del modelo utilizando métricas como la **precisión** y la **matriz de confusión**: Calcula la precisión y crea una matriz de confusión para evaluar el modelo.

Supervivencia Titanic

Regresión Logística



Datos de Pasajeros



Evaluación de Precisión

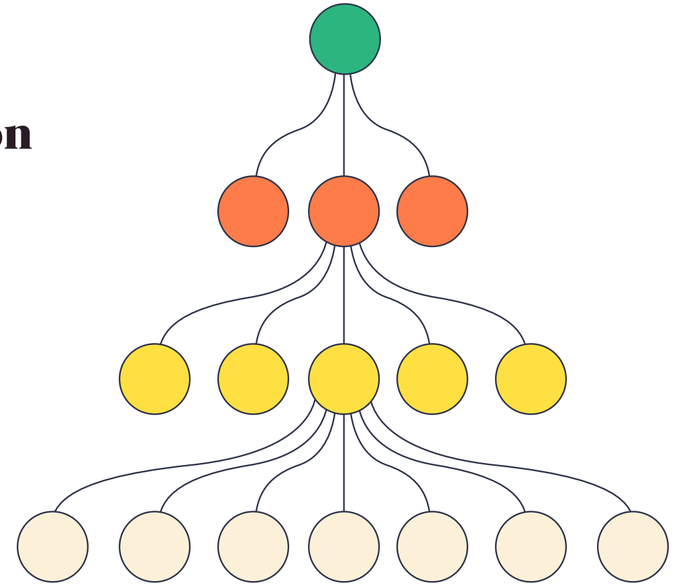
Clasificación de Pingüinos

Objetivo: Clasificar las especies de pingüinos (Adelie, Chinstrap, Gentoo) utilizando características como el largo del pico, el ancho del pico y la longitud de la aleta.
Modelo: Árbol de Decisión
Pasos:

1. **Cargar** el conjunto de datos **Penguins** desde Seaborn: Utiliza la función load_dataset('penguins') para obtener los datos.
2. **Explorar** y **visualizar** los datos utilizando Seaborn: Crea gráficos de dispersión y de pares para entender las características.
3. **Dividir** los datos en conjuntos de entrenamiento y prueba: Utiliza train_test_split para separar los datos.
4. **Entrenar** un modelo de clasificación (**Árbol de Decisión**): Usa DecisionTreeClassifier para entrenar el modelo.
5. **Evaluar** el rendimiento del modelo utilizando métricas como la precisión y la matriz de confusión: Calcula la **precisión** y crea una **matriz de confusión** para evaluar el modelo.

Clasificación Pingüinos

Árbol de Decisión



Datos de Pingüinos

Evaluación de Modelo

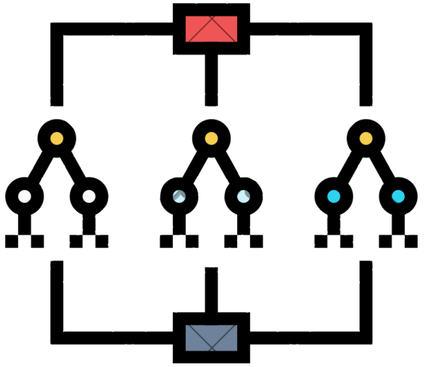
Duración de Viajes en Taxi 🚖

Objetivo: Predecir la duración de los viajes en taxi en función de características como la distancia, la hora del día y el día de la semana.
Modelo: Random Forest
Pasos:

1. **Cargar** el conjunto de datos Taxi desde Seaborn: Utiliza la función load_dataset('taxi') para obtener los datos.
2. **Limpiar** y **preprocesar** los datos: Maneja valores nulos y normaliza las características.
3. **Dividir** los datos en conjuntos de entrenamiento y prueba: Utiliza train_test_split para separar los datos.
4. **Entrenar** un modelo de regresión (**Random Forest**): Usa RandomForestRegressor para entrenar el modelo.
5. **Evaluar** el rendimiento del modelo utilizando métricas como el error cuadrático medio (**MSE**) y el coeficiente de determinación (**R^2**): Calcula el MSE y el R^2 para evaluar el modelo.

Predicción de Taxis

Random Forest



Datos de Viajes

Evaluación de MSE

Conclusiones

Importancia de la práctica

La **práctica** constante con proyectos sencillos permite familiarizarse con los conceptos y técnicas de machine learning, facilitando el **aprendizaje** y la aplicación en proyectos más complejos.

Visualización de datos

La **visualización** de datos es una herramienta poderosa para entender las características y **patrones** en los datos. Utilizar bibliotecas como Seaborn ayuda a identificar relaciones y **tendencias** que pueden influir en el modelado.

Clasificación y regresión

Trabajar con modelos de **clasificación** y **regresión** en proyectos prácticos permite entender cómo funcionan estos **algoritmos** y cómo se pueden aplicar a diferentes tipos de **problemas**.

Evaluación de modelos

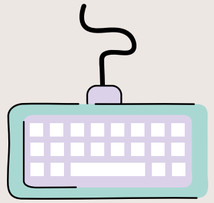
Evaluar el **rendimiento** de los modelos utilizando **métricas** adecuadas es crucial para asegurar **resultados** precisos y confiables. La práctica con diferentes métricas ayuda a entender las fortalezas y debilidades de cada modelo.

Conjuntos datos integrados

Utilizar conjuntos de **datos integrados** en bibliotecas como Seaborn y Scikit-learn facilita el inicio de proyectos de machine learning, permitiendo centrarse en el **aprendizaje** y la **implementación** sin preocuparse por la obtención de datos.

Desarrollo de habilidades

Realizar proyectos prácticos de machine learning ayuda a desarrollar **habilidades** esenciales como la **limpieza** y **preprocesamiento** de datos, la **selección** de características, el **entrenamiento** de modelos y la **evaluación** de resultados.



Gracias

Por Juan Duran

“Coding, Gaming and Leveling Up”