|  |  |
| --- | --- |
| **Nombre:** | Luis de la Garza González |
| **Matrícula:** | al03101869 |
| **Nombre del curso:** | Machine Learning. |
| **Trabajo:** | Actividad 004.- Introducción al aprendizaje automático supervisado. |
| **Nombre del profesor:** | Zuriel Dathan Mora Felix |
| **Fecha:** | 12 de octubre de 2025 |

**Contenido**

[**Objetivo** 3](#_Toc208334146)

[**Instrucciones:** 3](#_Toc208334147)

[**Desarrollo** 4](#_Toc208334148)

[1. Generación de conjuntos de datos experimentales y herramientas de Python. 4](#_Toc208334149)

[**a. Conjunto de datos con tres características linealmente relacionadas** 5](#_Toc208334150)

[**b. Conjunto de datos con cuatro características, relacionadas de forma no lineal** 7](#_Toc208334151)

[**3. Conjunto de datos que puedan expresar una relación lineal prevista** 9](#_Toc208334152)

[3. Métricas del error cuadrático medio y del coeficiente de determinación 13](#_Toc208334153)

[**Conclusiones sobre el aprendizaje automático** 15](#_Toc208334154)

[**Liga al código en Github** 16](#_Toc208334155)

# **Objetivo**

Resolver problemas de aprendizaje automático supervisado utilizando el lenguaje de programación Python.

# **Instrucciones:**

1. Una de las habilidades de un experto en inteligencia artificial y aprendizaje automático es tener la capacidad de generar conjuntos de datos experimentales para poner a prueba los diferentes modelos que va construyendo durante el desarrollo de su carrera. Investiga cuáles de las diversas herramientas de Python incluyen la funcionalidad de realizar esta acción y construye artificialmente los siguientes conjuntos de datos:
2. Conjunto de datos de tres características linealmente relacionados.
3. Conjunto de datos de cuatro características, relacionados de forma completa, no lineal.
4. Conjunto de datos que, de manera prevista, puedan expresar una relación lineal.
5. Investiga la manera en que se realiza el análisis de discriminarte lineal Python y utiliza este conocimiento para llevar a cabo la reducción de la dimensión y la representación gráfica del primer conjunto de datos construido en la pregunta anterior.
6. Utilizando la librería Scikit-learn, investiga cómo se implementan las métricas del error cuadrático medio y del coeficiente de determinación para evaluar la calidad de un modelo de aprendizaje supervisado.

# **Desarrollo**

# Generación de conjuntos de datos experimentales y herramientas de Python.

Una de las habilidades de un experto en inteligencia artificial y aprendizaje automático es tener la capacidad de generar conjuntos de datos experimentales para poner a prueba los diferentes modelos que va construyendo durante el desarrollo de su carrera. Investiga cuáles de las diversas herramientas de Python incluyen la funcionalidad de realizar esta acción y construye artificialmente los siguientes conjuntos de datos:

* + Conjunto de datos de tres características linealmente relacionados.
  + Conjunto de datos de cuatro características, relacionados de forma completa, no lineal.
  + Conjunto de datos que, de manera prevista, puedan expresar una relación lineal.

¡Perfecto, Luis! Justo en tu línea de dominio técnico y estructuración científica, aquí tienes una guía clara y funcional para generar conjuntos de datos sintéticos en Python, usando herramientas especializadas que permiten experimentar con modelos de aprendizaje automático.

**Herramientas de Python para generar datos sintéticos**

Las siguientes bibliotecas son ampliamente utilizadas por expertos en IA para crear conjuntos de datos artificiales:

|  |  |
| --- | --- |
| **Herramienta** | **Descripción** |
| scikit-learn | Incluye funciones como `make\_regression`, `make\_classification`, `make\_blobs`. |
| Faker | Genera datos realistas (nombres, direcciones, etc.) para simulaciones. |
| numpy + matplotlib | Útiles para crear relaciones matemáticas personalizadas y visualizarlas. |
| CTGAN / SDV | Modelos generativos para datos tabulares complejos. |

En nuestro caso, utilizaremos principalmente scikit-learn y numpy.

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.**a. Conjunto de datos con tres características linealmente relacionadas**

El primer conjunto de datos se genera con **make\_regression** de **scikit-learn**, siguiendo los pasos

La función `make\_regression` crea datos sintéticos donde las características (variables independientes) están relacionadas linealmente con una variable objetivo (target). Es útil para validar que un modelo de regresión pueda aprender correctamente una relación lineal.

**Parámetros usados**

X, y = make\_regression(n\_samples=100, n\_features=3, noise=5.0, random\_state=42)

- n\_samples=100: Se generan 100 observaciones (filas).

- n\_features=3: Cada observación tiene 3 características (columnas).

- noise=5.0: Se añade ruido gaussiano para simular variabilidad real.

- random\_state=42: Fija la semilla para reproducibilidad.

**Esto genera:**

X: Matriz de características (100 filas × 3 columnas).

y: Vector de valores objetivo (100 valores).

Convertimos X en un DataFrame con nombres de columnas descriptivos:

df = pd.DataFrame(X, columns=["Feature 1", "Feature 2", "Feature 3"])

Y se añade la columna “Target” con los valores de y:

df["Target"] = y

Mostramos las primeras 5 filas del DataFrame para una inspección rápida:

print(df.head())

**Estructura del conjunto**

|  |  |  |  |
| --- | --- | --- | --- |
| **Feature\_1** | **Feature\_2** | **Feature\_3** | **Target** |
| -0.792521 | 0.504987 | -0.114736 | 13.149332 |
| 0.280992 | -0.208122 | -0.622700 | -20.311291 |
| 0.791032 | 1.402794 | -0.909387 | 101.284137 |

* Las columnas Feature\_1, Feature\_2, Feature\_3 son variables independientes.
* La columna Target es la variable dependiente, calculada como una combinación lineal de las tres características más el ruido.

Matemáticamente, se puede expresar como:

Donde son coeficientes generados aleatoriamente, y es el ruido.

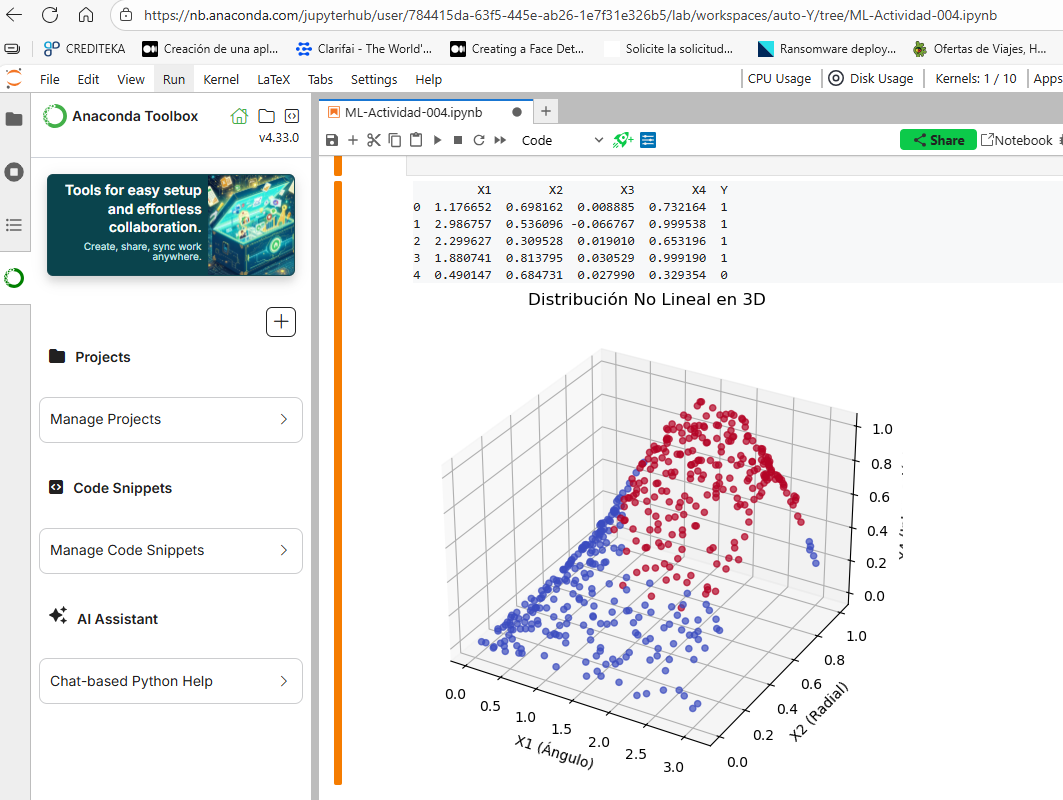
**Este tipo de conjunto es ideal para:**

* Verificar que un modelo de regresión lineal (como `LinearRegression`) puede recuperar los coeficientes.
* Evaluar el impacto del ruido en la predicción.
* Probar visualizaciones de correlación y ajuste.

**b. Conjunto de datos con cuatro características, relacionadas de forma no lineal**

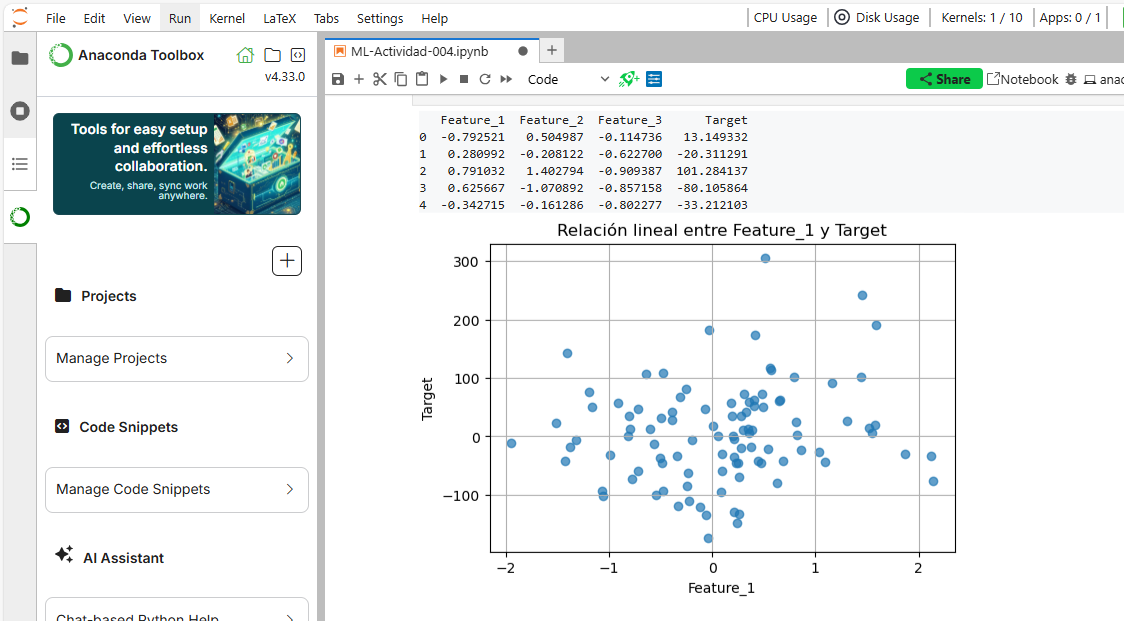
Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.Generación de un conjunto de datos sintético con cuatro características relacionadas de forma no lineal, ideal para pruebas con modelos de clasificación:

Ejecutamos el código y obtenemos la siguiente salida:

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.**3. Conjunto de datos que puedan expresar una relación lineal prevista**

Salida:

1. Análisis de discriminante lineal Python y reducción de la dimensión.

**Investiga la manera en que se realiza el análisis de discriminante lineal Python y utiliza este conocimiento para llevar a cabo la reducción de la dimensión y la representación gráfica del primer conjunto de datos construido en la pregunta anterior.**

El **Análisis Discriminante Lineal (LDA)** es una técnica supervisada que busca proyectar los datos en un espacio de menor dimensión maximizando la separación entre clases. A diferencia del Análisis de Componentes Principales (PCA), LDA utiliza las etiquetas (Y) para encontrar las direcciones más discriminantes.

**Reducción de la dimensión del primer conjunto de datos construido en la pregunta anterior:**

Para poder llevar a cabo el LDA, vamos a discrecional el objetivo como clases numeradas del 1 al 5, ya que el objetivo o “target” no deben ser valores continuos.

**Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.**

**Representación gráfica:**

Gráfico, Gráfico de dispersión

El contenido generado por IA puede ser incorrecto.Interpretación de la gráfica:

Eje X (LDA1): primera dirección discriminante. Es la combinación lineal de las variables originales que mejor separa las clases.

Eje Y (LDA2): segunda dirección discriminante. Captura la siguiente mejor separación entre clases, ortogonal a la primera.

Estas dos dimensiones fueron calculadas por LDA para maximizar la distancia entre clases y minimizar la dispersión dentro de cada clase.

Cada punto representa una muestra del conjunto de datos original.

El color indica la clase asignada (Clase del 1 al 5).

Los puntos de una misma clase están agrupados y separados de otras clases, significa que LDA logró una buena discriminación.

# Métricas del error cuadrático medio y del coeficiente de determinación

**Utilizando la librería Scikit-learn, investiga cómo se implementan las métricas del error cuadrático medio y del coeficiente de determinación para evaluar la calidad de un modelo de aprendizaje supervisado.**

En Scikit-learn, las métricas error cuadrático medio (MSE) y coeficiente de determinación (R²) son fundamentales para evaluar modelos de regresión.

**Implementación:**

Parámetros clave:

y\_true: valores reales

y\_pred: valores predichos

sample\_weight: pesos opcionales por muestra

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.multioutput: controla cómo se agregan errores si hay múltiples salidas ('raw\_values', 'uniform\_average', o lista de pesos)

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.Ejemplo con múltiples salidas:

**Coeficiente de Determinación (r2\_score)**

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.Indica qué proporción de la varianza en los datos es explicada por el modelo. Su valor va de **−∞ a 1**, donde **1** es predicción perfecta.

**Interpretación:**

R² = 1: predicción perfecta

R² = 0: el modelo no mejora respecto a la media

R² < 0: el modelo es peor que usar la media como predicción

# **Conclusiones sobre el aprendizaje automático**

El aprendizaje automático abarca múltiples estrategias y enfoques: supervisado, no supervisado, de refuerzo y profundo; que se seleccionan según el tipo de datos (estructurados o no estructurados) y los objetivos del análisis.

En problemas donde los datos cambian con el tiempo, es crucial considerar esta evolución al seleccionar modelos, ya que puede afectar significativamente la precisión de las predicciones.

La calidad del entrenamiento depende en gran medida de una adecuada preparación de los datos, que incluye limpieza, transformación y selección de características relevantes adaptadas al contexto local.

Comprender si los datos son lineales o no lineales, y su complejidad dimensional, permite elegir modelos adecuados y evitar errores de interpretación.

La elección del modelo debe basarse en el análisis técnico del problema, considerando la naturaleza de los datos, el resultado esperado y las restricciones operativas.

La regresión lineal es uno de los modelos más simples y utilizados. Es útil para establecer relaciones entre variables y ajustar funciones mediante métodos como mínimos cuadrados o descenso de gradiente.

Python y librerías como Scikit-learn nos permiten implementar modelos de forma sencilla y eficiente, facilitando el análisis y resolución de problemas complejos con poco código.

# **Liga al código en Github**

<https://github.com/luisgg121/ML-Actividad-004.git>

Archivo:

ML-Actividad-004.ipynb