MINERÍA DE DATOS

Año de cursada: 2° Año

Clase Nº 8: **Técnicas de Minería de Datos: Introducción al Aprendizaje Supervisado, Regresión Logística y Regresión Múltiple**

Contenido:

Conceptos Básicos del Aprendizaje Supervisado

- Definición y ejemplos
- Diferencia entre datos de entrenamiento y prueba
- Características y etiquetas

Modelos de Regresión vs. Clasificación

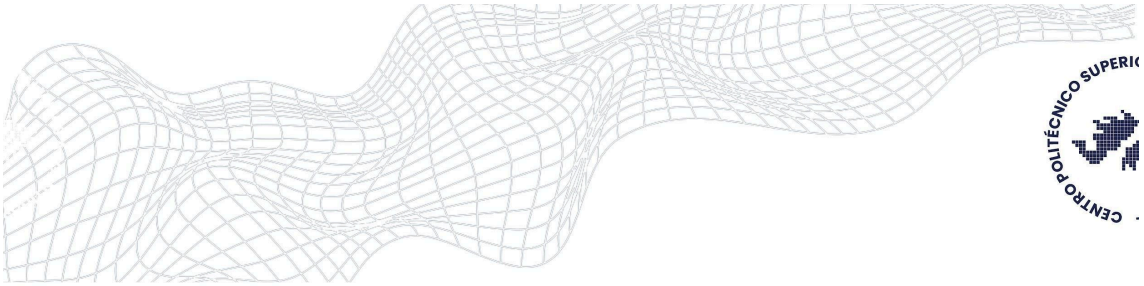
- ¿Qué es la regresión y cuándo usarla?
- ¿Qué es la clasificación y cuándo usarla?
- Diferencias clave entre regresión y clasificación

Introducción a la Regresión logística

- Concepto y aplicaciones
- Interpretación de coeficientes
- Evaluación del modelo

Actividad Práctica

- **Preparación de datos para modelos supervisados**



- **Implementación de regresión logística en un conjunto de datos real**

1. Introducción:

El aprendizaje supervisado es uno de los pilares fundamentales del aprendizaje automático y la ciencia de datos. En este enfoque, contamos con datos etiquetados, es decir, tenemos ejemplos de entradas y las salidas correspondientes. Nuestro objetivo es "aprender" de estos datos para poder hacer predicciones o clasificaciones precisas en datos nuevos y no vistos anteriormente.

Por ejemplo, imagina que tienes una caja llena de frutas mezcladas: manzanas y plátanos. Si alguien te muestra varias veces una fruta y te dice su nombre, eventualmente serás capaz de identificar por ti mismo si una fruta es una manzana o un plátano. Eso es esencialmente lo que hace el aprendizaje supervisado: "aprende" de ejemplos etiquetados.

En esta clase, no solo exploraremos los conceptos básicos del aprendizaje supervisado, sino que también nos sumergiremos en un tipo especial de modelado llamado "regresión múltiple". La regresión múltiple nos permite examinar cómo múltiples características o variables influyen en una salida.

2. Conceptos Básicos del Aprendizaje Supervisado

a. Definición y ejemplos:

El aprendizaje supervisado es una técnica de aprendizaje automático donde un modelo se entrena utilizando un conjunto de datos etiquetado. Esto significa que, para cada entrada en el conjunto de datos, hay una salida esperada, conocida como etiqueta. El objetivo principal es que, después de entrenar con este conjunto de datos, el modelo pueda predecir la etiqueta correcta para nuevas entradas no vistas.

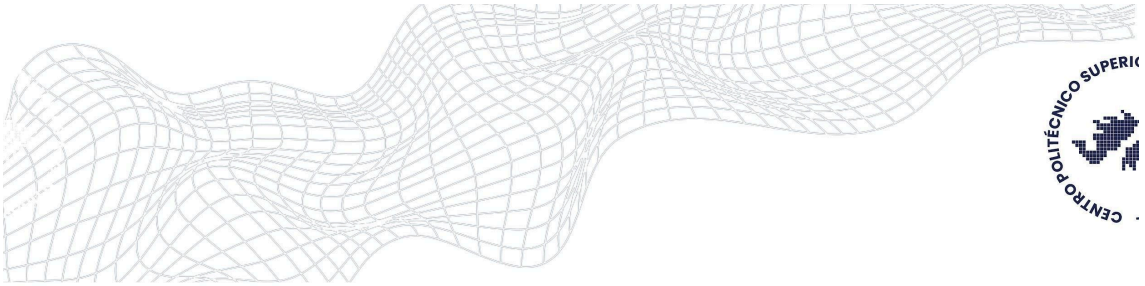
Ejemplo: Imagina que estás enseñando a un niño a diferenciar entre gatos y perros mostrándole imágenes de ambos. Cada vez que le muestras una imagen, le dices si es un gato o un perro. Con el tiempo, el niño aprenderá a identificarlos por sí mismo. En este escenario, el niño es el "modelo", las imágenes son las "entradas" y decir "gato" o "perro" es etiquetar esas entradas.

b. Diferencia entre datos de entrenamiento y prueba:

Cuando trabajamos con aprendizaje supervisado, generalmente dividimos nuestro conjunto de datos en dos partes:

Datos de entrenamiento: Se utilizan para entrenar el modelo. Es como el período de estudio antes de un examen.

Datos de prueba: Se utilizan para evaluar qué tan bien se desempeña el modelo con datos que no ha visto antes. Es como el examen después del período de estudio.



Esta división es crucial porque queremos asegurarnos de que nuestro modelo no solo memorice los datos (lo que llamamos "sobreajuste"), sino que generalice bien a situaciones nuevas.

c. Características y etiquetas:

Características: Son las entradas del modelo. Si pensamos en el ejemplo anterior de gatos y perros, las características podrían ser el color del pelaje, el tamaño del animal, la forma de sus orejas, etc.

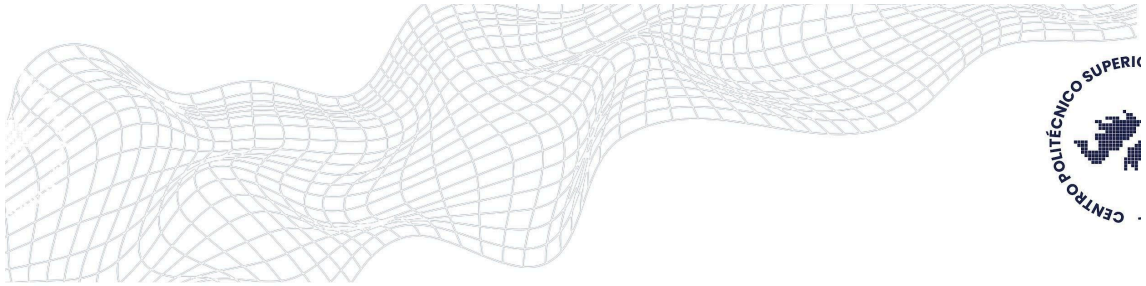
Etiquetas: Son las salidas que esperamos del modelo. En el ejemplo, las etiquetas serían "gato" o "perro".

En el contexto de un problema más técnico, como predecir el precio de una casa, las características podrían incluir el número de habitaciones, la ubicación, el tamaño del terreno, etc., y la etiqueta sería el precio de venta de la casa.

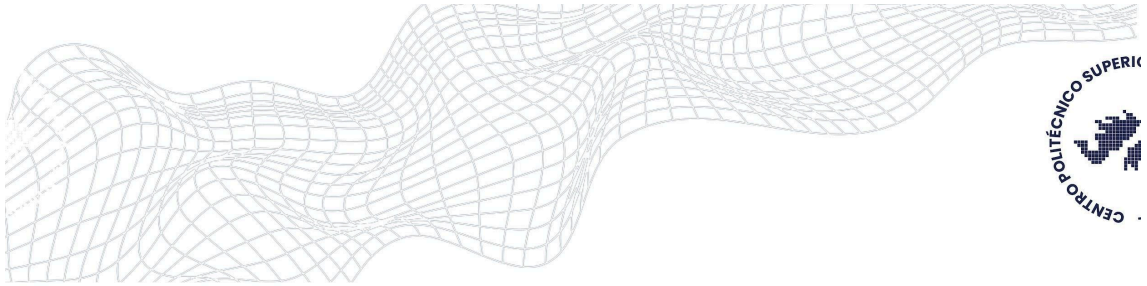
Con estos conceptos básicos, los estudiantes tendrán una base sólida para comprender los aspectos más avanzados del aprendizaje supervisado que se tratarán en las siguientes secciones.

d. Ejemplo de Aprendizaje Supervisado con el Conjunto de Datos Iris:

```
# Importando las bibliotecas necesarias

import pandas as pd
```



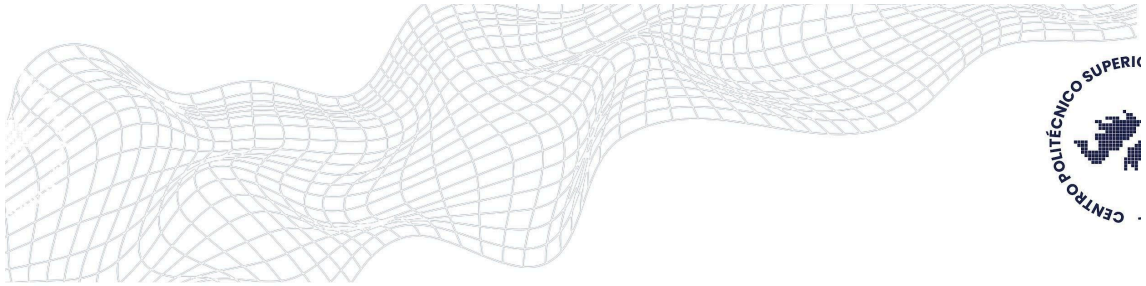
```
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score

# Cargando el conjunto de datos Iris
data = load_iris()
df = pd.DataFrame(data.data, columns=data.feature_names)
df['species'] = data.target

# Dividir el conjunto de datos en entrenamiento y prueba
X = df.drop('species', axis=1)
y = df['species']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Crear y entrenar el modelo de regresión logística
model = LogisticRegression(max_iter=200)
```



```
model.fit(X_train, y_train)


# Hacer predicciones con el conjunto de prueba
y_pred = model.predict(X_test)


# Calcular la precisión del modelo
accuracy = accuracy_score(y_test, y_pred)
print(f"Precisión del modelo: {accuracy:.2f}")
```

El aprendizaje supervisado es uno de los enfoques más comunes en el campo del aprendizaje automático. Se llama "supervisado" porque el proceso de entrenamiento del modelo se lleva a cabo bajo la "supervisión" de los datos previamente etiquetados. Es como si tuviéramos un "supervisor" o "maestro" que nos dice si nuestras predicciones son correctas o no mientras el modelo está aprendiendo. Veamos esto con más detalle:

Datos Etiquetados: En el aprendizaje supervisado, trabajamos con conjuntos de datos que tienen entradas y salidas conocidas. Estas salidas conocidas, también llamadas etiquetas o respuestas, actúan como una guía



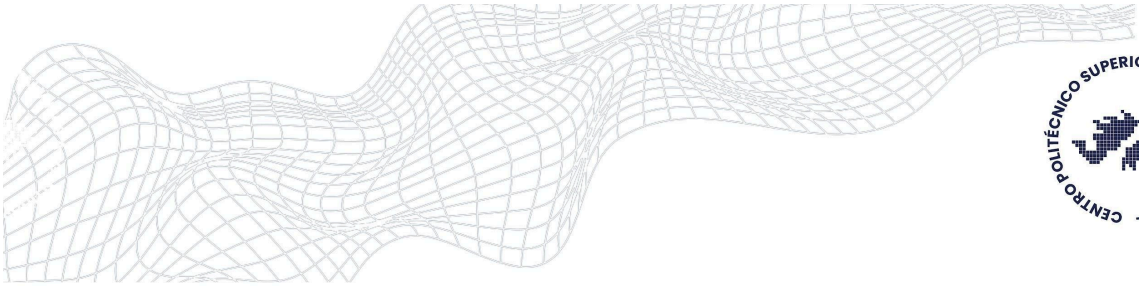
para el modelo. Durante el proceso de entrenamiento, el modelo hace predicciones basadas en las entradas y luego compara estas predicciones con las salidas reales (etiquetas) para ajustar y mejorar.

Retroalimentación: A medida que el modelo se entrena, recibe retroalimentación sobre la precisión de sus predicciones. Si un modelo predice incorrectamente, ajusta sus parámetros internos para mejorar sus predicciones futuras. Esta retroalimentación continua, basada en las etiquetas, es lo que permite que el modelo "aprenda".

Objetivo Claro: En el aprendizaje supervisado, el objetivo es claro: queremos que nuestro modelo aprenda a mapear las entradas a las salidas correctas basándose en los ejemplos proporcionados en los datos de entrenamiento. Una vez que el modelo ha sido entrenado, puede ser utilizado para predecir las salidas de nuevas entradas que no ha visto antes.

3. Modelos de regresión vs. clasificación

Regresión y clasificación son dos de los problemas más comunes en el aprendizaje supervisado. Aunque ambos involucran la predicción basada en



variables independientes, se diferencian en el tipo de resultado que se está prediciendo.

Regresión:

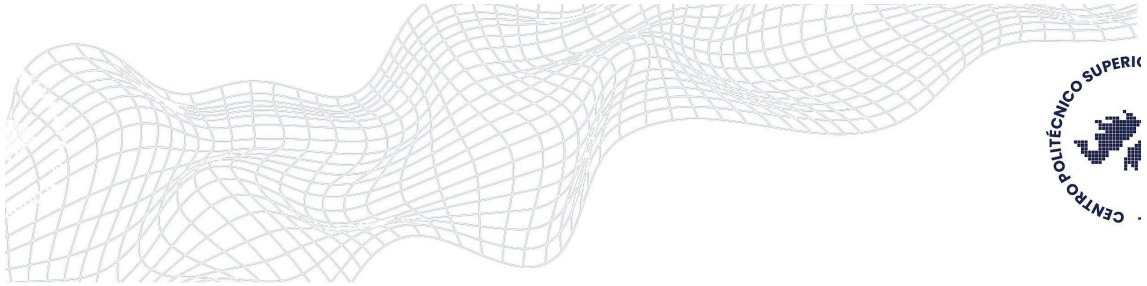
- **Objetivo:** Predecir una salida continua.
- **Ejemplo:** Predecir el precio de una casa basado en características como el número de habitaciones, ubicación, tamaño del terreno, etc.
- **Resultado:** Un valor numérico, como $250,000.
- **Modelos comunes:** Regresión lineal, regresión polinómica, regresión de árboles de decisión, entre otros.

Características clave:

1. La variable objetivo (o dependiente) es continua y tiene un orden.
2. Se busca una función que mejor se ajuste a los datos, minimizando el error entre las predicciones y los valores reales.

Clasificación:

- **Objetivo:** Predecir una salida categórica.
- **Ejemplo:** Determinar si un correo electrónico es spam o no spam basado en su contenido, remitente, etc.



- **Resultado:** Una categoría o clase, como "spam" o "no spam".
- **Modelos comunes:** Regresión logística, máquinas de soporte vectorial (SVM), árboles de decisión, redes neuronales, entre otros.

Características clave:

1. La variable objetivo es categórica y no tiene un orden inherente (aunque puede haber clasificaciones binarias o multiclase).
2. Se busca un límite o frontera que separe las clases de manera óptima.

Comparación:

- Mientras que la regresión predice un valor numérico, la clasificación asigna una etiqueta de una lista predefinida.
- En regresión, el error se mide típicamente como el error cuadrático medio (MSE), mientras que en clasificación, se utilizan métricas como precisión, exhaustividad y F1-score.
- La regresión es adecuada para problemas donde la salida es una cantidad (como el peso o el precio), mientras que la clasificación es adecuada para problemas donde la salida es una categoría (como "sí" o "no").

Ejemplo práctico: Imagina que estás trabajando con un conjunto de datos sobre coches. Si quieres predecir el precio de un coche basado en sus

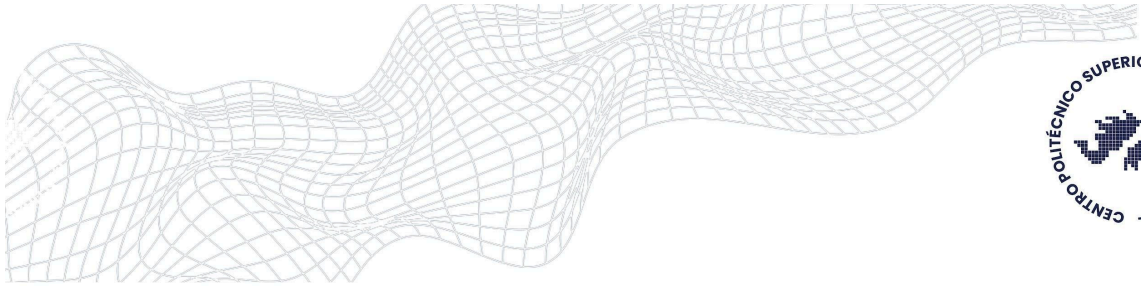
características, usarías regresión. Pero si quieres predecir si un coche es automático o manual basado en sus características, usarías clasificación.

Por ejemplo, en el caso del conjunto de datos Iris que mencionamos anteriormente:

Las entradas son las características de las flores, como la longitud y el ancho de los pétalos y sépalos.

- Las salidas o etiquetas son las especies de las flores (setosa, versicolor, virginica).
- Durante el entrenamiento, el modelo aprende a asociar ciertas características con ciertas especies basándose en los ejemplos proporcionados.
- Una vez entrenado, si le damos al modelo las características de una nueva flor que no ha visto antes, puede predecir su especie basándose en lo que ha aprendido.

En resumen, el aprendizaje supervisado es como enseñar a un estudiante con un libro de texto que tiene las respuestas al final. El estudiante resuelve los problemas y verifica sus respuestas con las del libro, aprendiendo de sus



errores, hasta que puede resolver problemas similares por sí mismo con precisión.

Regresión Logística

Introducción:

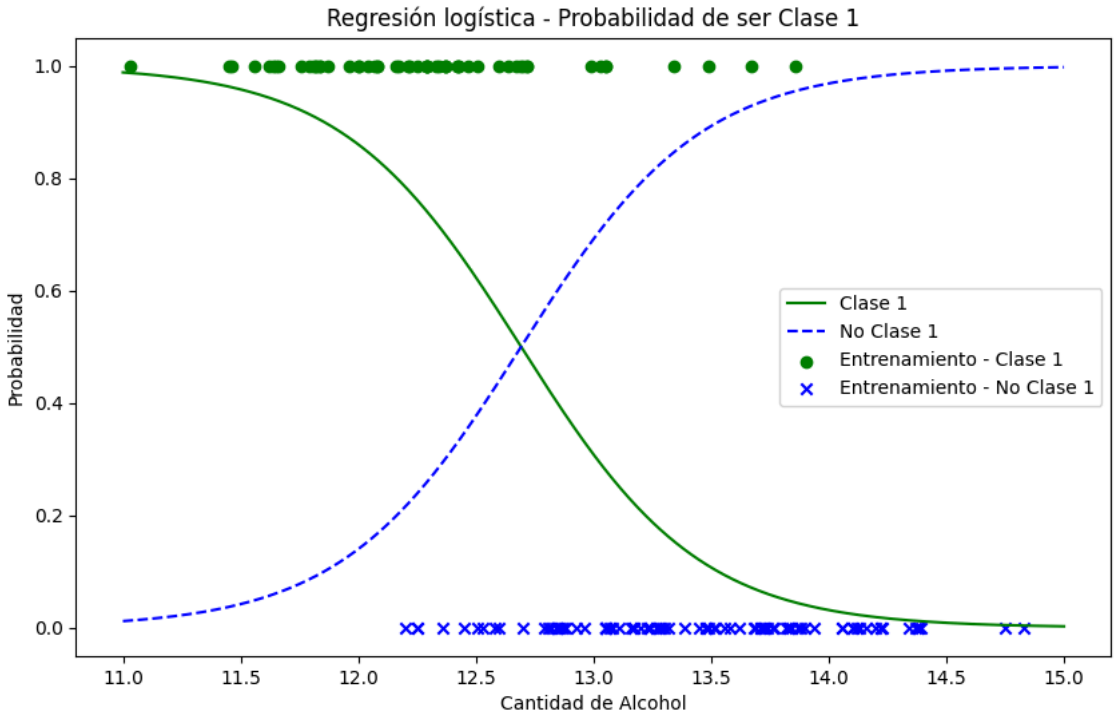
La regresión logística es un algoritmo de aprendizaje supervisado utilizado para clasificación. Aunque lleva el nombre de "regresión", se utiliza principalmente para problemas de clasificación binaria, es decir, cuando el resultado puede ser una de dos clases posibles (por ejemplo, Sí/No, Verdadero/Falso, 1/0).

Concepto Básico:

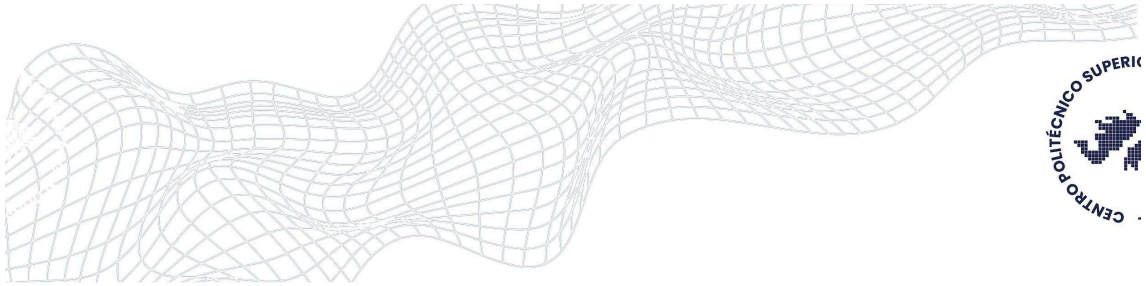
La regresión logística estima la probabilidad de que una instancia pertenezca a una clase particular. Si la probabilidad estimada es superior al 50%, el modelo predice que la instancia pertenece a esa clase (denominada clase positiva, etiquetada como "1"), y de lo contrario predice que no (es decir, pertenece a la clase negativa, etiquetada como "0").

Función Sigmoide:

La regresión logística utiliza la función sigmoide para comprimir los valores en un rango entre 0 y 1.



La regresión logística, en particular, se utiliza para problemas de clasificación binaria, donde el resultado puede ser una de dos clases posibles. Aunque el modelo estima probabilidades, estas probabilidades se traducen en una predicción de clase basada en un umbral, comúnmente 0.5. Si la probabilidad estimada es superior al umbral, se predice una clase, y si es inferior, se predice la otra.

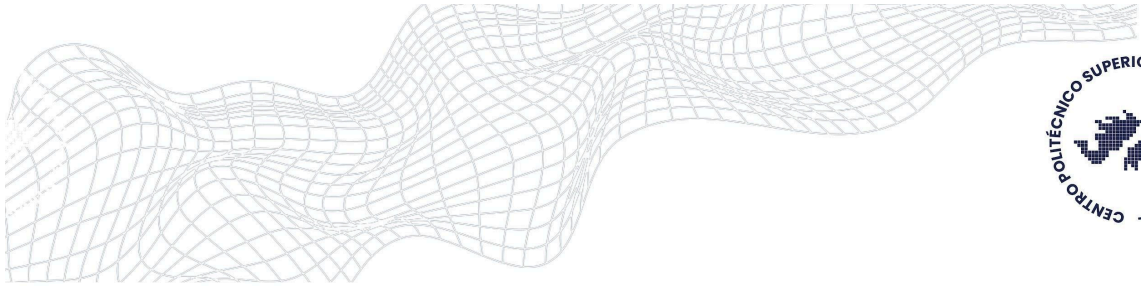


Por ejemplo, en un problema médico, podríamos usar la regresión logística para predecir si un paciente tiene una enfermedad (1) o no (0) basándonos en ciertas características, como la edad, el peso, los resultados de pruebas médicas, etc. Durante el entrenamiento, el modelo tiene acceso tanto a las características de los pacientes como a si realmente tienen la enfermedad, lo que lo convierte en un enfoque supervisado. Una vez entrenado, el modelo puede predecir la probabilidad de que un nuevo paciente (cuyos datos no se usaron en el entrenamiento) tenga la enfermedad basándose en sus características.

Vamos a considerar un ejemplo médico clásico: predecir si un paciente tiene diabetes o no, basándonos en ciertas mediciones de salud. Utilizaremos el conjunto de datos "Pima Indians Diabetes" para este propósito. Este dataset es ampliamente utilizado en la comunidad de aprendizaje automático y contiene información sobre mujeres de al menos 21 años de edad de herencia Pima Indian.

Características del conjunto de datos:

1. Número de embarazos.
2. Concentración plasmática de glucosa a 2 horas en una prueba oral de tolerancia a la glucosa.
3. Presión arterial diastólica (mm Hg).
4. Grosor del pliegue cutáneo del tríceps (mm).

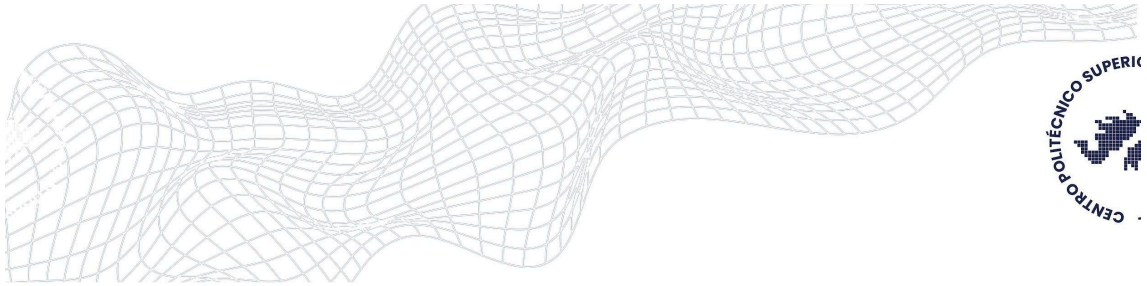


5. Insulina sérica de 2 horas (mu U/ml).
6. Índice de masa corporal (peso en kg/(altura en m)^2).
7. Función de pedigrí de diabetes.
8. Edad (años).

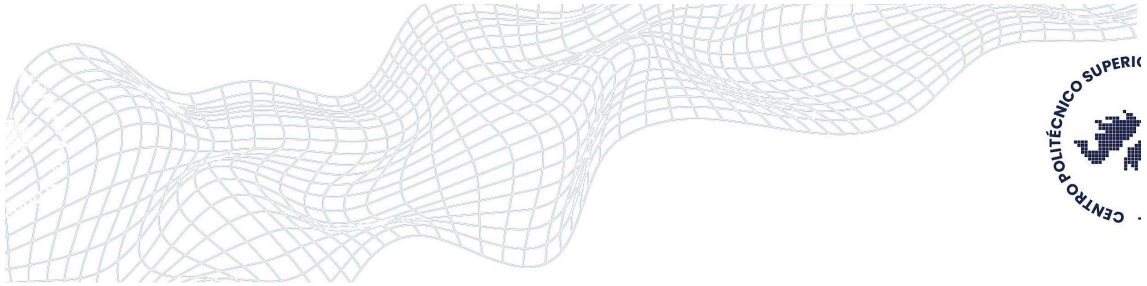
Variable objetivo:

- 0: No tiene diabetes.
- 1: Tiene diabetes.

```
# Importar las bibliotecas necesarias
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, confusion_matrix

# Cargar el conjunto de datos
url = "https://raw.githubusercontent.com/jbrownlee/Datasets/master/pima-indians-diabetes.data.csv"
```



```
column_names = ["num_pregnancies", "glucose", "blood_pressure", "skin_thickness", "insulin", "bmi", "diabetes_pedigree", "age", "label"]

data = pd.read_csv(url, names=column_names)


# Separar las características y la variable objetivo

X = data.drop("label", axis=1)

y = data["label"]


# Dividir el conjunto de datos en entrenamiento y prueba

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)


# Crear y entrenar el modelo de regresión logística

model = LogisticRegression(max_iter=1000)

model.fit(X_train, y_train)


# Hacer predicciones

y_pred = model.predict(X_test)
```



```
# Evaluar el modelo
accuracy = accuracy_score(y_test, y_pred)
confusion = confusion_matrix(y_test, y_pred)

print(f"Accuracy: {accuracy*100:.2f}%")
print("Confusion Matrix:")
print(confusion)
```

4. Introducción a la Regresión Múltiple

a. Concepto y aplicaciones

Regresión Múltiple: Es una extensión de la regresión lineal simple que permite predecir una variable dependiente a partir de dos o más variables independientes. La idea principal es encontrar un plano (o hiperplano en más dimensiones) que mejor se ajuste a los datos.

b. Aplicaciones:

Negocios: Predicción de ventas basada en factores como gastos publicitarios, precio del producto, y competencia.

Medicina: Estimar el progreso de enfermedades considerando múltiples síntomas o factores de riesgo.

Ciencias Sociales: Evaluar el impacto de políticas públicas considerando diferentes variables socioeconómicas.

Ingeniería: Optimizar procesos basándose en múltiples variables de entrada.

c. Interpretación de coeficientes

Cada coeficiente en un modelo de regresión múltiple representa el cambio esperado en la variable dependiente por un cambio de una unidad en la variable independiente correspondiente, manteniendo todas las demás variables independientes constantes.

Por ejemplo, en un modelo que predice el precio de una casa basado en su tamaño y antigüedad, el coeficiente del tamaño nos dice cuánto cambia el precio de la casa por cada metro cuadrado adicional, manteniendo la antigüedad constante.

d. Evaluación del modelo

La calidad de un modelo de regresión múltiple se evalúa a través de varios indicadores:

R-cuadrado (R^2): Representa la proporción de la variabilidad en la variable dependiente que es explicada por las variables independientes. Un R^2 cercano a 1 indica que el modelo explica una gran proporción de la variabilidad.

Error estándar: Mide la dispersión de los datos observados alrededor de la línea de regresión. Cuanto menor sea el error estándar, mejor será el ajuste del modelo a los datos.

P-valor: Se utiliza para testear la hipótesis de que una variable independiente no tiene relación con la variable dependiente. Un p-valor pequeño (generalmente menor a 0.05) indica que podemos rechazar esta hipótesis.

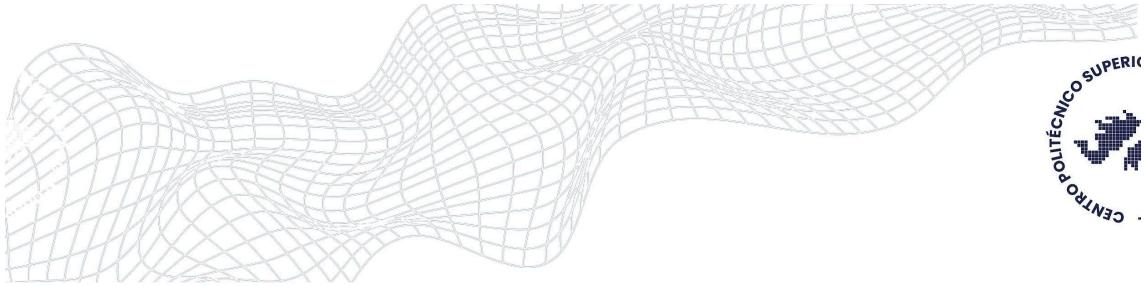
Análisis de residuos: Los residuos (diferencia entre valores observados y predichos) deben distribuirse aleatoriamente y no mostrar patrones discernibles.

Ejemplo de Regresión Múltiple

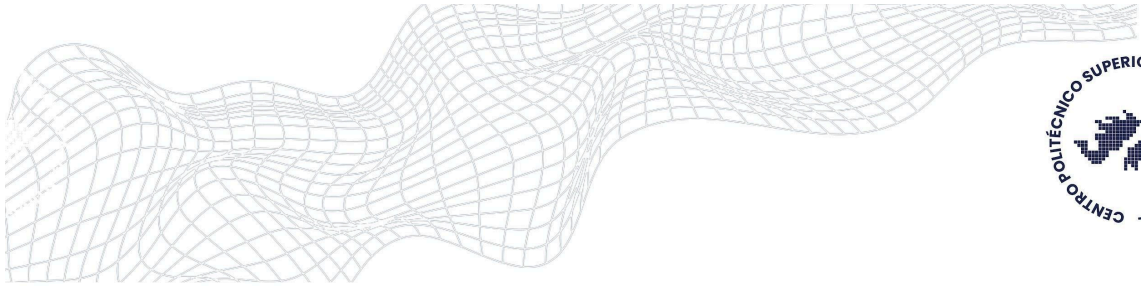
Vamos a crear un ejemplo ficticio de regresión múltiple utilizando un dataset de precios de automóviles. Imaginemos que tenemos un conjunto de datos que contiene los siguientes atributos:

- Precio: Precio del automóvil (variable dependiente).
- Año: Año de fabricación del automóvil.
- Kilometraje: Kilómetros recorridos por el automóvil.
- HP: Caballos de fuerza del automóvil.

Supongamos que queremos predecir el precio del automóvil en función de su año de fabricación, kilometraje y caballos de fuerza.



```
# Importando las bibliotecas necesarias
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import statsmodels.api as sm

# Generando datos ficticios
np.random.seed(42)
n = 100
año = np.random.choice(np.arange(2000, 2023), n)
kilometraje = np.random.choice(np.arange(5000, 200000), n)
hp = np.random.choice(np.arange(100, 400), n)
precio = 5000 + (2023 - año) * 200 - kilometraje * 0.05 + hp * 30 + np.random.normal(0, 2000, n)

# Creando un DataFrame
df = pd.DataFrame({'Año': año, 'Kilometraje': kilometraje, 'HP': hp, 'Precio': precio})
```

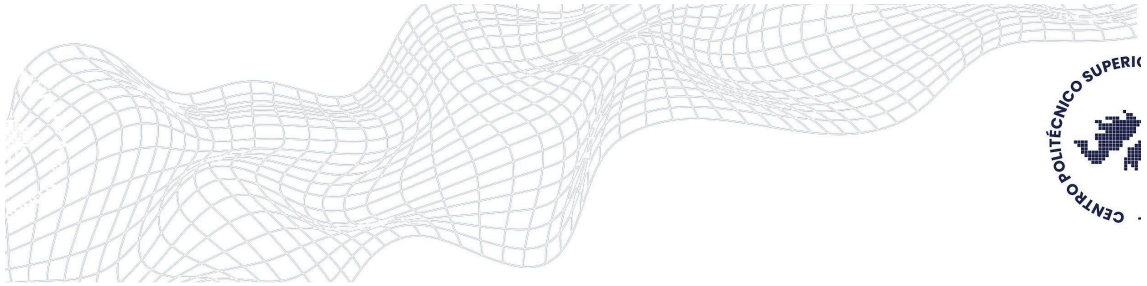


```
# Aplicando regresión múltiple
X = df[['Año', 'Kilometraje', 'HP']]
X = sm.add_constant(X)  # Añadiendo una constante (intercepto)
y = df['Precio']

modelo = sm.OLS(y, X).fit()

# Mostrando el resumen del modelo
print(modelo.summary())
```

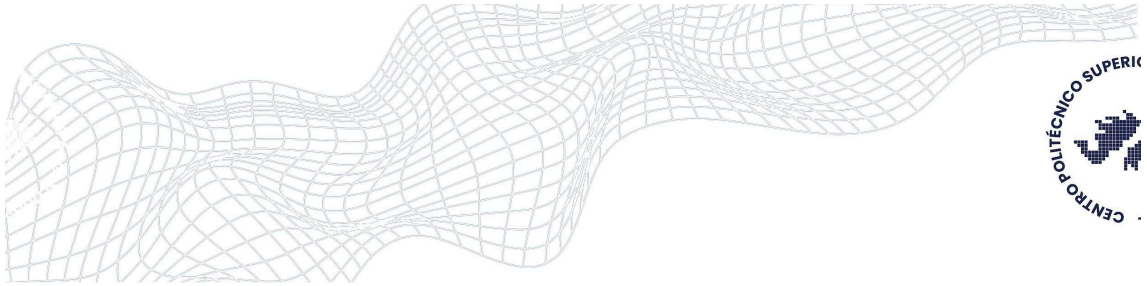
Otro Ejemplo para datos categóricos. Regresión Logística.

Vamos a usar el conjunto de datos wine de sklearn. Este conjunto de datos es el resultado de un análisis químico de vinos producidos en una misma región de Italia por tres cultivadores diferentes. Se tomaron trece diferentes medidas para diferentes constituyentes encontrados en los tres tipos de vino.

En este caso, vamos a simplificar el problema para predecir si un vino es del cultivador "1" o no, basándonos en la cantidad de "alcohol" y "flavonoides" en el vino.



```
# Importando las bibliotecas necesarias
import numpy as np
import matplotlib.pyplot as plt
from sklearn import datasets
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression

# Cargando el conjunto de datos
wine = datasets.load_wine()
X = wine.data[:, 0:1]  # Tomamos solo la cantidad de alcohol
y = (wine.target == 1).astype(int)  # 1 si es clase "1", 0 si no lo es

# Dividiendo el conjunto de datos en entrenamiento y prueba
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Creando el modelo de regresión logística
log_reg = LogisticRegression()
```



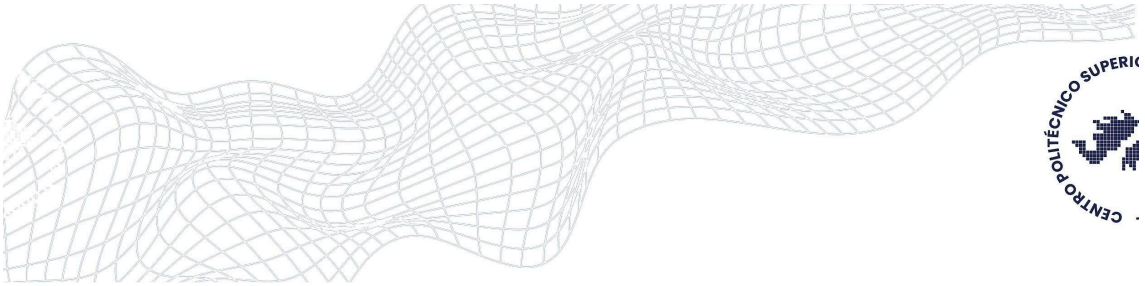
```
log_reg.fit(X_train, y_train)


# Predicciones y probabilidades
X_new = np.linspace(11, 15, 1000).reshape(-1, 1)
y_prob = log_reg.predict_proba(X_new)


# Graficando
plt.figure(figsize=(10, 6))
plt.plot(X_new, y_prob[:, 1], "g-", label="Clase 1")
plt.plot(X_new, y_prob[:, 0], "b--", label="No Clase 1")
plt.scatter(X_train[y_train==1], y_train[y_train==1], c="g", marker="o", label="Entrenamiento - Clase 1")
plt.scatter(X_train[y_train==0], y_train[y_train==0], c="b", marker="x", label="Entrenamiento - No Clase 1")
plt.xlabel("Cantidad de Alcohol")
plt.ylabel("Probabilidad")
plt.title("Regresión logística - Probabilidad de ser Clase 1")
plt.legend()
plt.show()
```

Actividad: Aplicando Aprendizaje Supervisado para Resolver Problemas Reales

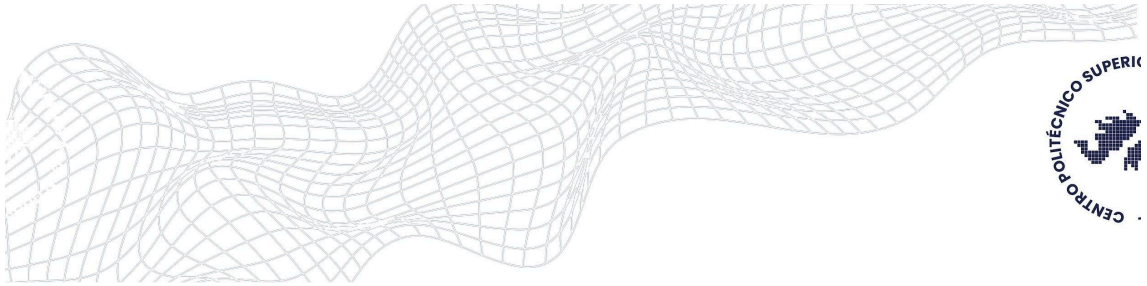
Objetivo de la Actividad: Esta actividad se enfoca en aplicar conceptos de aprendizaje supervisado, específicamente regresión logística y regresión múltiple, para resolver problemas del mundo real. A través de esta práctica, entenderás cómo los modelos predictivos pueden ayudarnos a tomar decisiones informadas basadas en el análisis de datos.

Instrucciones:

1. **Investigación Teórica:**
    - **Prompt para ChatGPT:** "Explica la diferencia entre aprendizaje supervisado y no supervisado con ejemplos."
    - **Prompt para Google Gemini:** "Busca estudios de caso recientes sobre la aplicación de regresión múltiple en el sector inmobiliario."
2. **Selección de Conjunto de Datos:**
    - Elige un conjunto de datos para el análisis. Puede ser el conjunto de datos Iris para regresión logística o un conjunto de datos sobre precios de vivienda para regresión múltiple. Asegúrate de que el conjunto de datos esté limpio y preparado para el análisis.



3. **Preparación del Conjunto de Datos:**
    - Realiza cualquier limpieza de datos necesaria, como tratar con valores faltantes o normalizar las características.
4. **Implementación de Modelos:**
    - **Regresión Logística:**
        - **Prompt para ChatGPT:** "Proporciona un ejemplo de cómo implementar un modelo de regresión logística en Python usando scikit-learn para predecir si una planta Iris es de la especie Setosa."
    - **Regresión Múltiple:**
        - **Prompt para ChatGPT:** "¿Cómo puedo usar la regresión múltiple para predecir el precio de una casa basándome en su tamaño, ubicación y número de habitaciones en Python?"
5. **Evaluación de Modelos:**
    - Evalúa tus modelos usando métricas apropiadas. Para regresión logística, considera la precisión y la matriz de confusión. Para regresión múltiple, considera R-cuadrado y el error estándar.
6. **Compartir y Discutir en el Foro:**



- Comparte tus hallazgos, incluyendo cómo preparaste los datos, el código utilizado para los modelos, y tu interpretación de los resultados.
- **Prompt para ChatGPT:** "¿Cuáles son algunas buenas prácticas al compartir análisis de datos y resultados de modelos predictivos con una audiencia no técnica?"

7. **Revisión por Pares:**
    - Ofrece feedback constructivo a tus compañeros sobre sus análisis y sugiere mejoras o alternativas basadas en tu experiencia.

Entregables:

- Un informe que incluya tu análisis, el proceso de implementación de los modelos, resultados, y cómo estos pueden ser aplicados en decisiones del mundo real.
- Participación activa en el foro, compartiendo tu trabajo y ofreciendo feedback a tus compañeros.

5. Bibliografía Obligatoria:

Han, J., Kamber, M., & Pei, J. (2012). Data mining: Concepts and techniques (3rd ed.). Elsevier. ISBN: 978-0-12-381479-1.

Python Software Foundation. (n.d.). Welcome to Python.org. Retrieved Julio 11, 2023, de https://www.python.org/

The pandas development team. (n.d.). pandas: powerful Python data analysis toolkit. Retrieved Julio 11, 2023, de https://pandas.pydata.org/

Project Jupyter. (n.d.). Project Jupyter. Retrieved Julio 11, 2023, de https://jupyter.org/