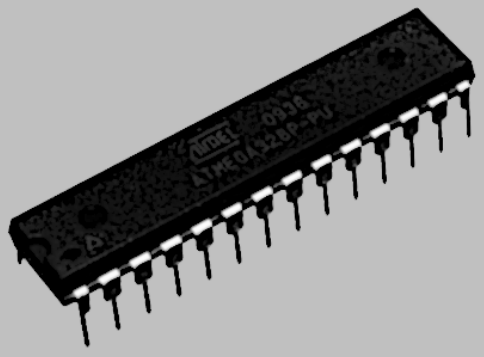

Inteligencia Artificial Embebida

Nombre: _____ Grupo: _____

Dr. Enrique García Trinidad
Tecnológico de Estudios Superiores de Huixquilucan
https://enriquegarcia.xyz
enrique.g.t@huixquilucan.tecnm.mx

Práctica 11: Clasificación con Regresión Logística

1. Introducción Teórica

A pesar de tener la palabra regresión en su nombre, la **Regresión Logística** es un algoritmo fundamental utilizado para problemas de **clasificación**. Mientras que la regresión lineal predice valores continuos, la regresión logística predice probabilidades discretas, típicamente para clasificación binaria (dos clases, representadas como 0 y 1).

El algoritmo toma una combinación lineal de las características de entrada y la pasa por una función de activación llamada **Sigmoide** (o función logística), que comprime cualquier valor de entrada en un rango entre 0 y 1. Si la probabilidad calculada es mayor a un umbral (usualmente 0.5), el modelo predice la clase 1; de lo contrario, predice la clase 0.

Además, en esta práctica introducimos el concepto de **Preprocesamiento de Datos**. Debido a que las características pueden tener diferentes escalas (por ejemplo, el precio del alquiler en miles y el área en decenas), es necesario estandarizar los datos. La estandarización asegura que la varianza de cada dimensión de las características sea 1 y la media sea 0, evitando que el resultado de la predicción sea dominado por características con valores numéricos muy grandes.

2. Desarrollo de la Práctica e Implementación

2.1. Paso 1: Importar dependencias

Importamos la clase `StandardScaler` para el preprocesamiento matemático de nuestras variables, y el modelo `LogisticRegression` para realizar la clasificación.

```
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
```

2.2. Paso 2: Definir el conjunto de datos

Definimos una matriz de características `X` donde cada elemento representa el precio de renta y el área de un inmueble. El vector `y` contiene las etiquetas que indican si la habitación se alquilará (0: no; 1: sí).

```
# Cada elemento en X denota el alquiler y el area.
# y indica si se debe alquilar la habitacion (0: no; 1: si).
X =
   [[2200,15],[2750,20],[5000,40],[4000,20],[3300,20],[2000,10],[2500,12],[1200

   [2880,10],[2300,15],[1500,10],[3000,8],[2000,14],[2000,10],[2150,8],[3400,20

   [5000,20],[4000,10],[3300,15],[2000,12],[2500,14],[10000,100],[3150,10],

   [2950,15],[1500,5],[3000,18],[8000,12],[2220,14],[6000,100],[3050,10]]

y = [1,1,0,0,1,1,1,1,0,1,1,0,1,1,0,1,0,0,0,1,1,1,0,1,0,1,0,1,1,0]
```

2.3. Paso 3: Preprocesamiento de los datos (Estandarización)

Instanciamos el objeto `StandardScaler` y aplicamos `fit_transform()` sobre los datos `X`. Esto calculará la media y la desviación estándar del conjunto de datos y transformará los valores para que estén centrados en 0.

```
ss = StandardScaler()
X_train = ss.fit_transform(X)
print(X_train)
```

2.4. Paso 4: Ajustar el modelo (Entrenamiento)

Utilizamos el método `fit` del objeto `LogisticRegression` para entrenar los parámetros del modelo usando las características estandarizadas (`X_train`) y las etiquetas (`y`).

```
# Usa el metodo fit de LogisticRegression para entrenar los
   parametros del modelo
lr = LogisticRegression()
lr.fit(X_train, y)
```

2.5. Paso 5: Predecir nuevos datos (Inferencia)

Para predecir un nuevo dato (por ejemplo, renta de 2000 y área de 8), debemos aplicar exactamente la misma transformación estadística que usamos en el entrenamiento utilizando `ss.transform()`. Luego utilizamos `predict()` para obtener la etiqueta de clase y `predict_proba()` para ver las probabilidades calculadas por la función sigmoide.

```
testX = [[2000, 8]]
X_test = ss.transform(testX)
print("Valor a predecir: ", X_test)
```

```
label = lr.predict(X_test)
print("Etiqueta predicha= ", label)

# Mostrar la probabilidad predicha
prob = lr.predict_proba(X_test)
print("Probabilidad= ", prob)
```

3. Conclusión

El modelo de regresión logística clasifica exitosamente la nueva muestra. Tras normalizar el dato (renta: 2000, área: 8) a valores de $[-0{,}689, -0{,}576]$, el modelo arroja una etiqueta predicha de **1** (sí se rentará). Las probabilidades internas muestran un 41,88 % para la clase 0 y un 58,11 % para la clase 1, confirmando la decisión del clasificador.

4. Evaluación de Comprensión

Instrucciones: Selecciona la respuesta correcta para cada una de las siguientes preguntas.

1. A pesar de llamarse Regresión Logística, ¿para qué tipo de tareas se utiliza principalmente este algoritmo en Machine Learning?
    A. Para predecir valores continuos como el precio futuro de una acción.
    B. Para clasificar datos en categorías, especialmente en clasificación binaria (0 o 1).
    C. Para agrupar datos sin etiquetas en clústeres.
    D. Para reducir la dimensionalidad de matrices complejas.

2. Según la práctica, ¿cuál es el propósito de utilizar `StandardScaler` antes de entrenar el modelo?
    A. Cambiar las etiquetas de y para que sean valores negativos y positivos.
    B. Estandarizar las características para que la media sea 0 y la varianza sea 1, evitando que características con números grandes dominen el modelo.
    C. Convertir el modelo de clasificación en un modelo de regresión lineal.
    D. Multiplicar automáticamente los datos de entrenamiento para aumentar el tamaño de la muestra.

3. En la fase de predicción (Paso 5), ¿por qué se utiliza `ss.transform(testX)` en lugar de `ss.fit_transform(testX)`?
    A. Porque `transform` ajusta una nueva media y varianza específica para el dato de prueba.

B. Porque los datos de prueba no requieren estandarización, pero el comando es obligatorio.

C. Para aplicar la misma media y desviación estándar que el modelo ya aprendió de los datos de entrenamiento, manteniendo la consistencia de la escala.

D. Porque `fit_transform` solo funciona con regresión lineal y no con regresión logística.

4. ¿Qué información devuelve la función `predict_proba(X_test)` de scikit-learn?

   A. Devuelve exclusivamente un 0 o un 1 absoluto para la clasificación.

   B. Devuelve los pesos θ (coeficientes) aprendidos por el algoritmo.

   C. Devuelve las estimaciones de probabilidad, indicando la probabilidad de que la muestra pertenezca a cada una de las clases disponibles.

   D. Devuelve el error cuadrático medio del modelo de clasificación.

5. En el contexto del conjunto de datos definido, si la etiqueta objetivo y fuera reemplazada por el valor exacto de la renta en pesos (ej. $y = [2200, 2750, 5000...]$), ¿sería adecuado seguir usando `LogisticRegression`?

   A. Sí, la regresión logística puede mapear valores en pesos sin ningún problema.

   B. No, porque la regresión logística requiere etiquetas discretas/categóricas para clasificar, por lo que una regresión lineal sería la herramienta adecuada.

   C. Sí, siempre y cuando se estandarice también el vector y con `StandardScaler`.

   D. No, porque `LogisticRegression` solo admite áreas de cuartos, no precios de rentas.