# Aprendizaje Supervisado I

Guía de trabajo

September 12, 2024

## 1 Introducción al Aprendizaje Supervisado

El aprendizaje supervisado es una rama del aprendizaje automático (machine learning) que se basa en el uso de datos etiquetados para entrenar modelos. Los algoritmos de aprendizaje supervisado aprenden una función que mapea una entrada a una salida basada en ejemplos de entrada-salida pares.

### 1.1 Conceptos Clave

- **Entrenamiento y Test:** Dividir los datos en conjuntos de entrenamiento y prueba es crucial para evaluar el rendimiento de los modelos.
- **Métricas de Evaluación:** Las métricas como la precisión, el error cuadrático medio (MSE) y la exactitud son esenciales para evaluar el rendimiento del modelo.
- **Sobreajuste y Subajuste:** Equilibrar el modelo para que no se ajuste demasiado ni demasiado poco a los datos de entrenamiento.

### 1.2 Aplicaciones en el Mundo Real

- Predicción de precios de casas.
- Diagnóstico médico a partir de imágenes.
- Clasificación de correos electrónicos como spam o no spam.

## 2 Regresión Lineal

La regresión lineal es un enfoque estadístico utilizado para modelar la relación entre una variable dependiente y una o más variables independientes. Se utiliza para predecir valores continuos.

### 2.1 Fundamentos Matemáticos

El modelo de regresión lineal puede representarse matemáticamente como:

$$y = \beta_0 + \beta_1 x + \epsilon$$

donde $y$ es la variable dependiente, $x$ es la variable independiente, $\beta_0$ es el intercepto, $\beta_1$ es la pendiente, y $\epsilon$ es el término de error.

## 2.2 Implementación en Python

```python
# Importar las bibliotecas necesarias
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.datasets import load_diabetes
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score

# Cargar el dataset de Diabetes de scikit-learn
diabetes = load_diabetes()

# Crear un DataFrame de pandas con los datos y nombres de columnas
data = pd.DataFrame(data=diabetes.data, columns=diabetes.feature_names)
data['target'] = diabetes.target  # Agregar la columna de la variable objetivo

# Visualización de las primeras filas del dataset
print(data.head())

# Seleccionar una característica (BMI) para simplificar la demostración
X = data[['bmi']]  # Variable independiente
y = data['target']  # Variable dependiente

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

# Crear el modelo de regresión lineal
model = LinearRegression()

# Entrenar el modelo con el conjunto de entrenamiento
model.fit(X_train, y_train)

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

# Evaluar el rendimiento del modelo
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f"Error Cuadrático Medio (MSE): {mse:.2f}")
print(f"Coeficiente de Determinación (R²): {r2:.2f}")

# Visualizar los resultados
plt.scatter(X_test, y_test, color='blue', label='Datos Reales')
plt.plot(X_test, y_pred, color='red', linewidth=2, label='Línea de Regresión')
plt.xlabel('Índice de Masa Corporal (BMI)')
plt.ylabel('Progresión de la Diabetes')
plt.title('Regresión Lineal - Predicción de la Progresión de la Diabetes')
plt.legend()
plt.show()
```

## 2.3 Ejercicio Práctico

Implemente un modelo de regresión lineal utilizando el conjunto de datos *Boston Housing* y evalúe su rendimiento usando el MSE.

# 3 Regresión Logística

La regresión logística es un método de clasificación que se utiliza para predecir la probabilidad de una variable dependiente categórica. A diferencia de la regresión lineal, la regresión logística se utiliza para problemas de clasificación binaria.

## 3.1 Fundamentos Matemáticos

El modelo de regresión logística puede representarse con la siguiente función sigmoide:

$$P(Y = 1|X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}}$$

## 3.2 Implementación en Python

```python
# Importar las bibliotecas necesarias
import numpy as np
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 classification_report, confusion_matrix, accuracy_score
import matplotlib.pyplot as plt
import seaborn as sns

# Cargar el dataset Iris de scikit-learn
iris = load_iris()

# Crear un DataFrame de pandas con los datos
data = pd.DataFrame(data=iris.data, columns=iris.feature_names)
data['species'] = iris.target  # Agregar la columna de la variable objetivo

# Visualización de las primeras filas del dataset
print(data.head())

# Seleccionar características (todas las características) y la variable objetivo
X = data[iris.feature_names]  # Variables independientes (características)
y = data['species']  # Variable dependiente (especies)

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

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

# Entrenar el modelo con el conjunto de entrenamiento
model.fit(X_train, y_train)

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

# Evaluar el rendimiento del modelo
print("Reporte de Clasificación:\n", classification_report(y_test, y_pred))
print("Precisión del modelo (Accuracy):", accuracy_score(y_test, y_pred))

# Mostrar la matriz de confusión
conf_matrix = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(8, 6))
```

```
sns.heatmap(conf_matrix, annot=True, cmap='Blues', fmt='d',
            xticklabels=iris.target_names, yticklabels=iris.target_names)
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Matriz de Confusión - Regresión Logística')
plt.show()
```

### 3.3 Ejercicio Práctico

Utilice otro dataset para entrenar un modelo de regresión logística.

## 4 Trabajo Fuera de Clase

Los estudiantes deben implementar modelos de regresión lineal y logística utilizando datasets simples como *Boston Housing* o *Iris*. Asegúrese de analizar los resultados y preparar un informe breve que incluya la evaluación del rendimiento del modelo, las métricas utilizadas y posibles mejoras.