Universidad
Rey Juan Carlos

Práctica 1. Regresión y Clasificación con Modelos Lineales y Logísticos

Daniel Parra Segovia

02591500K

Curso 2023/2024

ÍNDICE

Ejercicio 1. Regresión Lineal (Matlab).

La función fitlm() de Matlab se usa para ajustar el modelo de regresión lineal a los datos de entrenamiento.

Resultados del ajuste lineal:

w1 = 0.2305
w2 = 0.0000
w3 = 0.2433

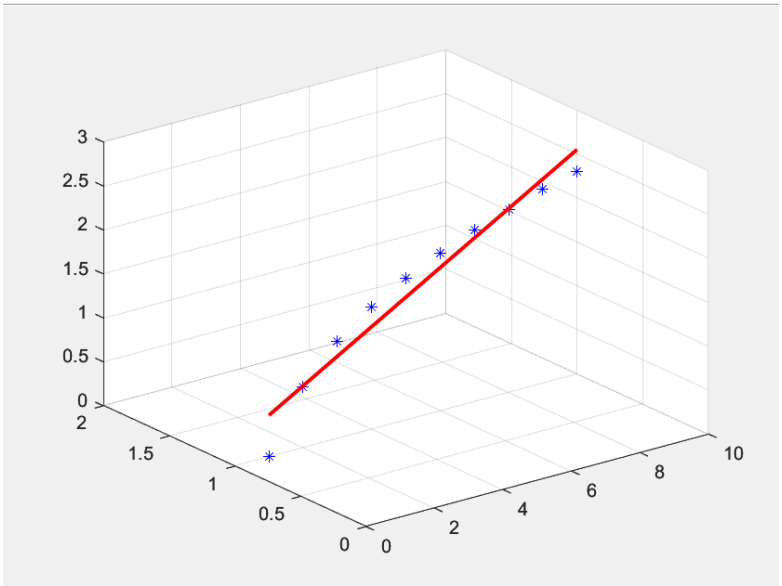

Fig. 1. Datos de entrenamiento y modelo de regresión lineal

Una recta no es la función que mejor se ajusta a los datos de entrenamiento. Una función curva como una polinómica o una función raíz cuadrada se ajustarían mejor.

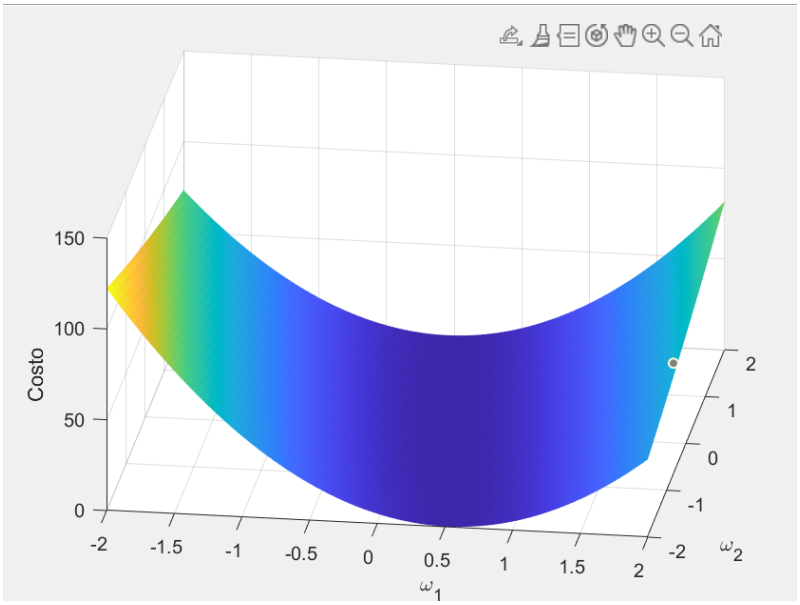

Fig. 2. Función de coste en función de los parámetros w1 y w2

Ejercicio 2. Regresión Logística binaria (Matlab).

La función fitglm() de Matlab se usa para ajustar el modelo de regresión logístico binario a los datos de entrenamiento.

```
Generalized linear regression model:
    logit(y) ~ 1 + x1 + x2 + x3
    Distribution = Binomial

Estimated Coefficients:
                   Estimate      SE       tStat      pValue
                   ________    ______    ________    _______

    (Intercept)      1.9146    1.2641     1.5146     0.12987
    x1              -0.27488   1.3662     -0.2012    0.84054
    x2              -0.73401   1.5511    -0.47322    0.63605
    x3              -0.90411   2.2513     -0.4016    0.68798


20 observations, 16 error degrees of freedom
Dispersion: 1
Chi^2-statistic vs. constant model: 4.52, p-value = 0.211
```

Fig. 3. Parámetros del modelo de regresión logístico binario

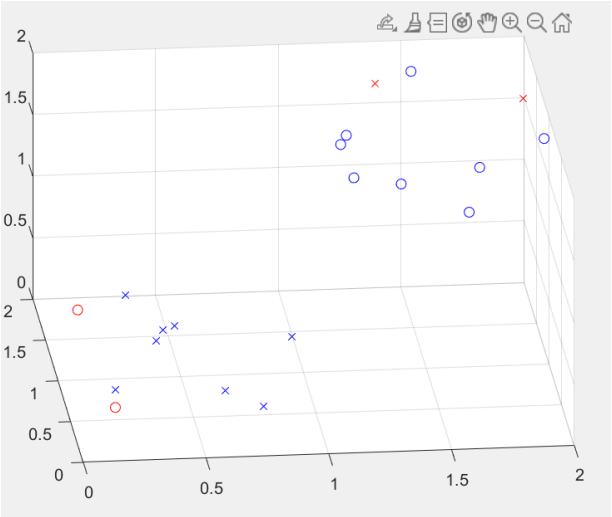

Fig. 4. Representación del modelo
En rojo los fallos, y en azul los aciertos.

El modelo falla en 4/20 de los datos de entrenamiento, un 20% de tasa de error. El modelo de regresión logístico binario no tiene suficientes parámetros para ajustarse a los datos de entrenamiento.

Esta tasa de error es aceptable con estos datos de entrenamiento. Un modelo más complejo seguramente tendría menos tasa de error, pero generalizaría peor con datos nuevos, estaría sobreajustado.

Ejercicio 1. Regresión Lineal (Python).

La función LinearRegression().fit se usa para ajustar el modelo de regresión lineal a los datos de entrenamiento.

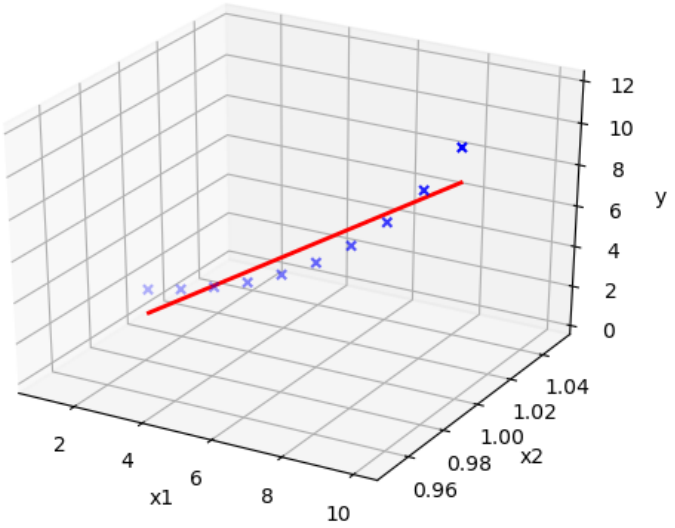

Fig. 5. Datos de entrenamiento y modelo de regresión lineal

Como en el caso anterior, una recta no es la función que mejor se ajusta a los datos de entrenamiento. Una función curva como una polinómica o una función raíz cuadrada se ajustarían mejor.

Ejercicio 2. Función de coste (Python).

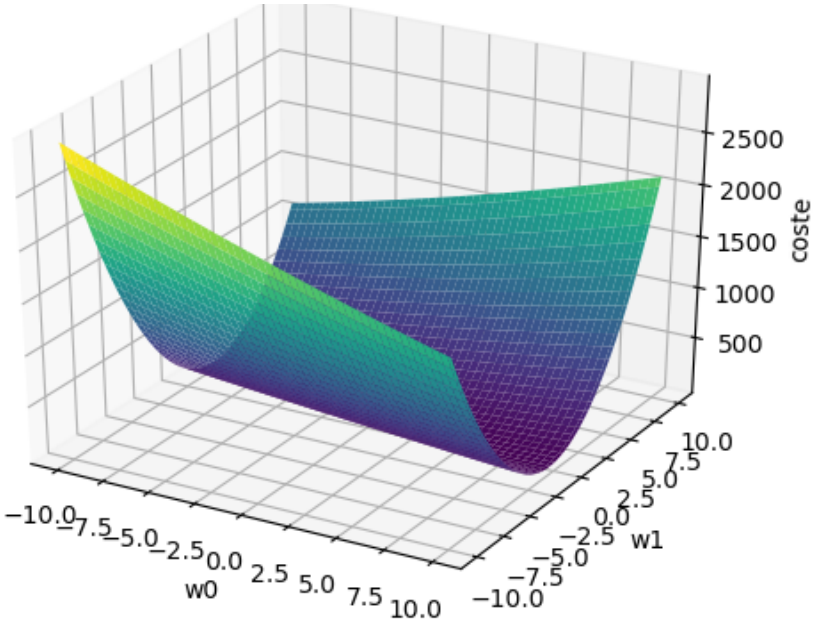

Fig. 6. Función de coste a partir del logaritmo de la verosimilitud

Ejercicio 3. Regresión Logística (Python).

Se usa la función LogisticRegression().fit() de python para construir el modelo clasificador.

Como en el ejercicio anterior, el modelo falla en 4/20 de los datos de entrenamiento, un 20% de tasa de error. El modelo de regresión logístico binario no tiene suficientes parámetros para ajustarse a los datos de entrenamiento.

Esta tasa de error es aceptable con estos datos de entrenamiento. Un modelo más complejo seguramente tendría menos tasa de error, pero generalizaría peor con datos nuevos, estaría sobreajustado.

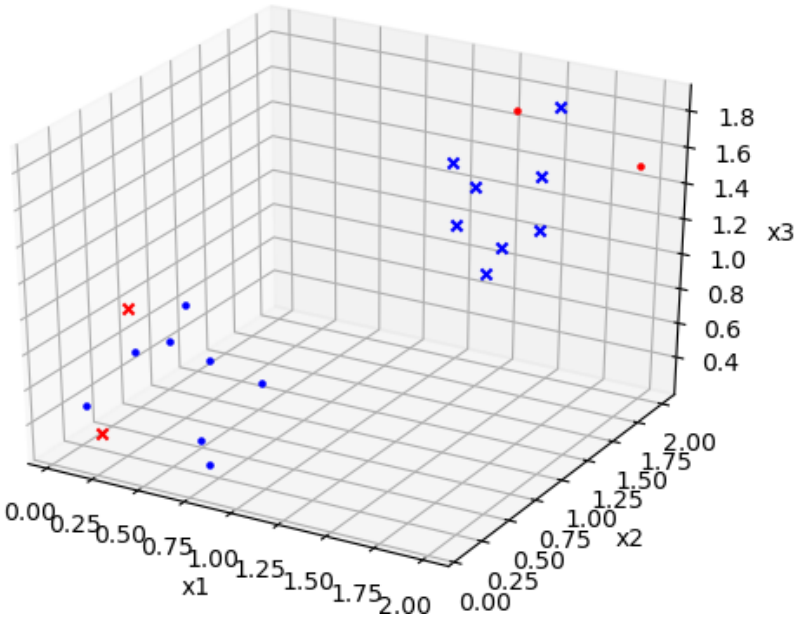

Fig. 7. Representación del modelo
En rojo los fallos, y en azul los aciertos.