Análisis y reporte sobre el desempeño del modelo

José Benjamín Ruiz García
A01750246

Se analizará el modelo de regresión lineal con descenso de gradiente que se construyó con la librería de scikit learn para predecir el peso de un pez basado en sus medidas físicas (altura, ancho y largo).

Describir claramente el algoritmo utilizado:
La regresión lineal es un algoritmo de machine learning que busca conocer los valores de una variable "y" (variable dependiente), dados los valores de ciertas variables "X" (variables independientes). Lo que hace el modelo es buscar la ecuación de una recta que pase por la mayor cantidad de datos posibles.

$$y = m_1x_1 + m_2x_2 + \cdots + m_nx_n + b$$

Donde:

- m -> coeficientes
- x -> variables independientes
- b -> intercepción de la recta

El método del descenso del gradiente busca los coeficientes de las variables en la recta, y calcula un ajuste a estos coeficientes en cada iteración del aprendizaje:

$$\Theta_j := \Theta_j - \frac{\propto}{m} \sum_{i=1}^{n} [(h_\Theta(x_i) - y)x_i]$$

Donde:

- Θ -> coeficientes
- x -> variables independientes
- $\propto$ -> learning rate (hiperparámetro)
- n -> número de variables
- m -> número de datos
- $h_\Theta(x_i)$ -> y aproximada
- y -> valor real de y

Este descenso del gradiente minimiza el error que hay entre la "y aproximada" y la "y real", y cuando encuentra este mínimo error ya no ajusta más los coeficientes de las "X"

Describe claramente el dataset y su separación en conjunto de entrenamiento y prueba

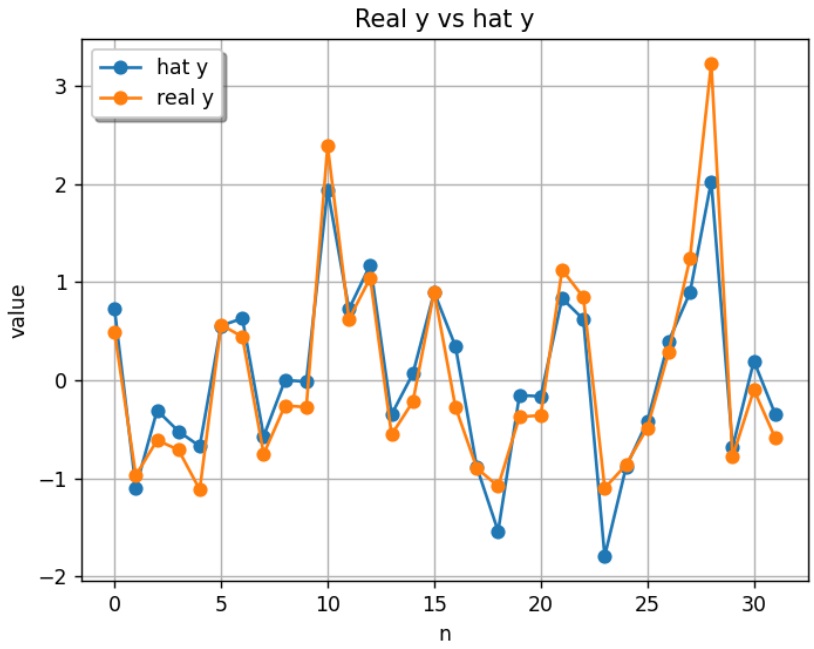
El dataset que se utilizó para el entrenamiento y prueba del modelo tiene el peso (g) y las medidas físicas del pez (longitud, largo y ancho (cm)). Se quiere saber el peso de los peces basado en esas medidas físicas. Para obtener resultados óptimos lo primero que se hizo fue estandarizar el dataset $\frac{x-\mu}{\sigma}$ para que el modelo aprenda con valores cercanos a 0 y no cree sesgos con alguna variable.
Con esto el dataset estaba listo para pasarlo al modelo. Se dividió el dataset en train (80%) y test (20%) varias veces para que el modelo realizara 5 aprendizajes y pruebas diferentes para comprobar que los resultados fueran confiables y no sólo un tema de suerte.

Sesgo, varianza, y overfitting u overfitting: bajo medio alto

```
R^2 Score with 1 split: 0.8662
Bias: 0.8243  Variance: 0.8046

R^2 Score with 2 split: 0.8542
Bias: 0.8862  Variance: 1.0328

R^2 Score with 3 split: 0.8771
Bias: 1.1864  Variance: 1.0612

R^2 Score with 4 split: 0.8519
Bias: 0.666  Variance: 0.7847

R^2 Score with 5 split: 0.875
Bias: 1.0065  Variance: 0.7035
```

Como podemos observar el sesgo y la varianza son bajos. Esto significa que el modelo hace predicciones de valor ya que, por un lado, un sesgo bajo nos dice que las predicciones fueron muy precisas, y por otro lado, una varianza baja nos dice que las predicciones tienen un grado bajo de aleatoriedad. Además, están muy balanceados, por lo que se asume que no se hizo underfitting ni overfitting en ninguna de las 5 corridas.

La siguiente gráfica muestra los resultados del modelo y da apoyo a la declaración anterior



Utilización de técnicas de regularización o ajuste de parámetros para mejorar el desempeño del modelo

El modelo dio resultados muy prometedores ya que en general la R^2 fue alta en todas las corridas, además no se hizo overfitting ni overfitting. Estos resultados se deben a que se hicieron varias pruebas variando el hiperparámetro alpha (learning rate) y el número de épocas. Estos ajustes se hicieron varias veces y se llegó a un valor de $\alpha = 0.01$ y épocas = 100.