Análisis y Reporte sobre el desempeño del modelo de regresión lineal

José Ángel García López
A01275108

Inteligencia artificial avanzada para la ciencia de datos

Modelo
El modelo implementado para esta entrega es una regresión lineal la cual es una técnica fundamental en estadísticas y aprendizaje automático que se utiliza para modelar la relación entre una variable independiente y una variable dependiente. En este informe, presentamos una implementación práctica de una regresión lineal simple utilizando la plataforma de aprendizaje profundo TensorFlow.

Set de datos
El set de datos utilizados en este avance es el del iris el cual comúnmente se utiliza para problemas de clasificación, sin embargo quise tomar un enfoque diferente analizando la relación entre dos de sus componentes como lo es el ancho del pétalo y el largo del sépalo.

Separación de datos
Para este modelo se dividió en 60% para train, 20% para test y 20% para validación, realmente no hay una razon en especifico para estos porcentajes y de hecho es algo que se modificará en el futuro para demostrar la forma en la que cambian los resultados.

Resultados

En una etapa inicial, se realizaron diferentes pruebas con los datos crudos, esto quiere decir que simplemente importamos los datos y tal y como venias se introdujeron al modelo dando un resultado relativamente bueno dada la naturaleza del modelo implementado, cabe destacar que se implementaron 100 epochs y se utilizo el metodo de optimizacion de Gradiente descendiente usando un “learning rate” de 0.03, los resultados se presentan a continuación:

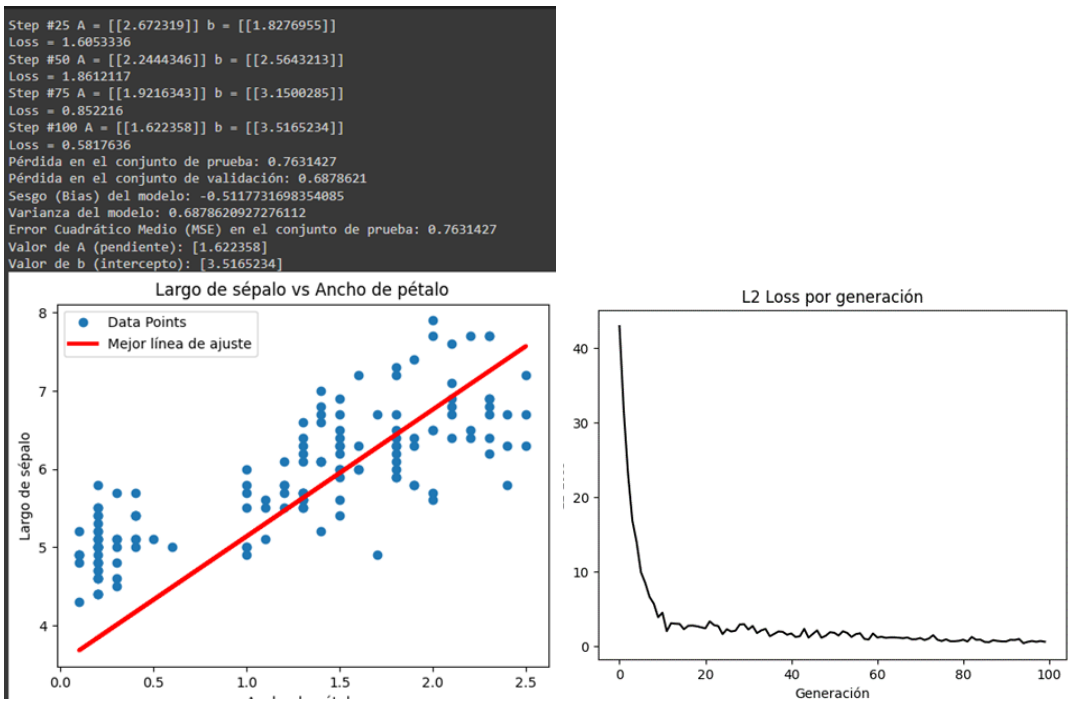

Imagen 1(LR = 0.03, epochs= 100)

Como se puede observar, el tiene fallar en los datos más próximos al o en el eje x, como mediciones estre tenemos MSE el cual en este caso particular tiene un valor de 0.7631 y un sesgo de -0.511… lo que indica que el modelo tiende a subestimar valores objetivos, con respecto al aprendizaje, la segunda gráfica muestra el desempeño del modelo al pasar los epochs que aunque sea un tanto irregular tiende a 0.

Para mejorar el modelo, modifique el valor del “Learning rate” para demostrar el valor que toma este elemento en el código:

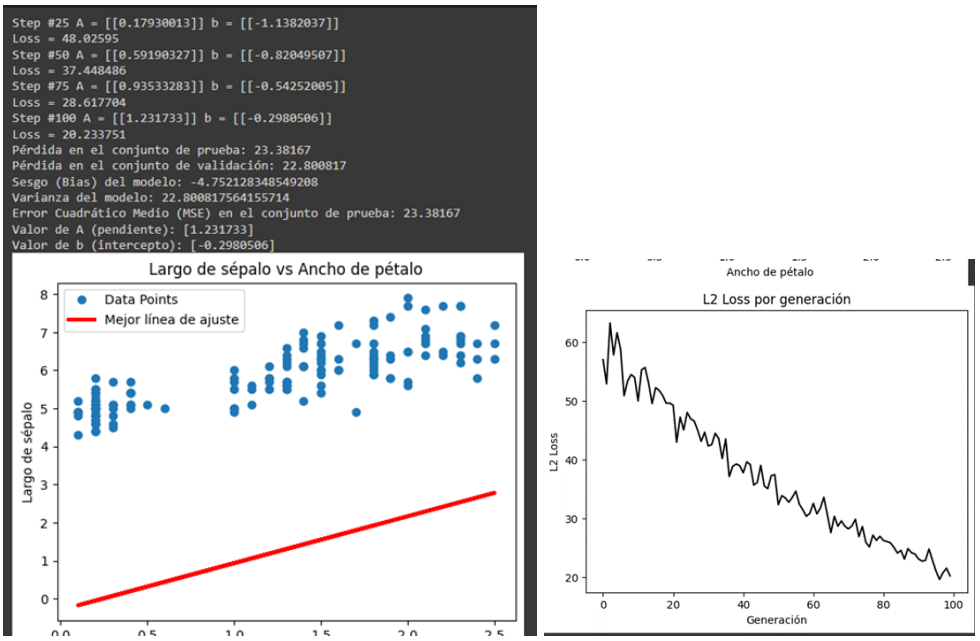

Imagen 2(LR = 0.001, epochs= 100)

En la imagen anterior se muestra los efectos de un “Learning rate” más bajo y como se puede apreciar, la cantidad de epochs no le es suficiente para lograr un resultado mínimamente viable y tal y como se ve, el modelo está muy por debajo de los valores reales, por mencionar una métrica, el valor de MSE se encuentra en el rango de las decenas siendo de 22.8 con un sesgo de -4.25 por lo que para nada es útil, lo que nos da un claro ejemplo de “Underfitting”.

A contunuacion, se presenta la tercera prueba donde también se modificó el valor del “learning rate” por uno más alto siendo este de 0.05.

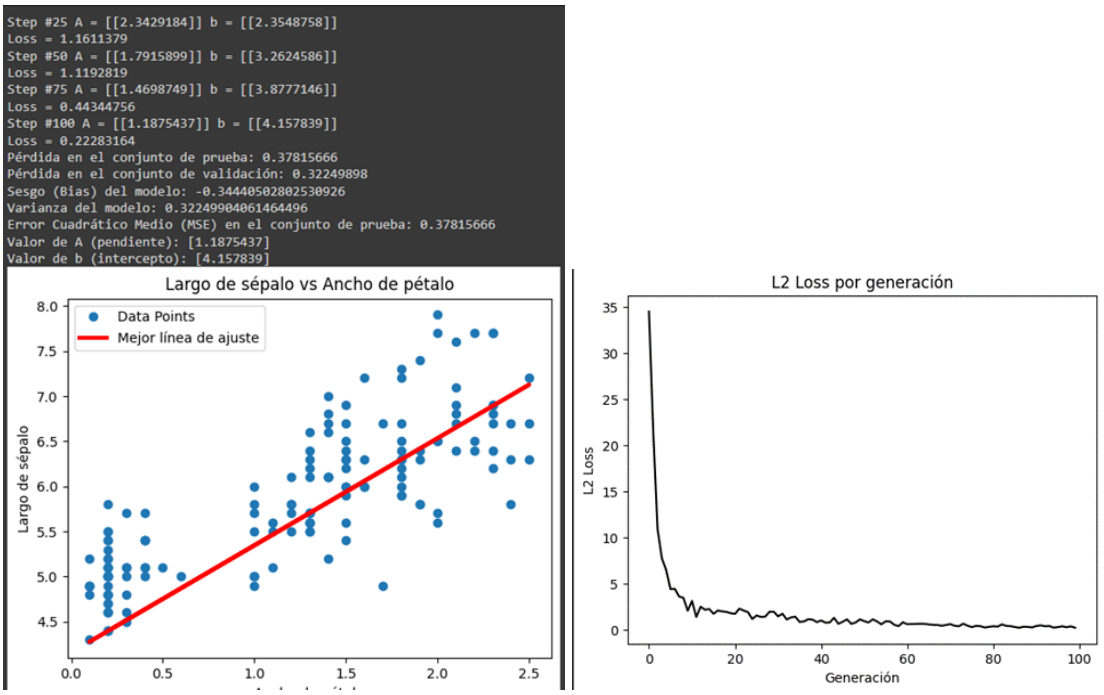

Imagen 3(LR = 0.05, epochs= 100)

Como se puede apreciar, la modificación en el valor nos mejores resultados de las dos pruebas previas, en primer lugar el valor del sesgo disminuye sin embargo aún es negativo, en el caso del MSE tenemos una baja drástica con respecto a los valores anteriores, en este caso es de 0.3781, con respecto al valor de la varianza en esta prueba resulta ser de las más altas con 0.032 lo que indica que la predicciones están dispersas alrededor de los datos reales. Para finalizar, con respecto al aprendizaje, podemos apreciar que el los valores de pérdida se precipitan de forma mas rapida ya que pasados los 40 epochs ya se tiene n valores cercanos al 0.

Diagnóstico y mejoras para el modelo

Con respecto al comportamiento del modelo anterior, es claro que es una implementación bastante rudimentaria y apresurada ya que no toma en cuenta la diversidad de los datos por lo que como primer mejora es implementar la normalización de los datos para trabajar con rangos más simples, dentro de este repositorio se pueden ver las modificaciones del código a

lo largo del tiempo por lo que solo se aprecia la etapa conde ya se implementó la mencionada normalización.

Con los datos estandarizados, se realizaron 4 pruebas más para demostrar la viabilidad de modificar el la forma en cómo aprende al modelo junto.

La primer prueba también se realizó con un valor de 0.03 manteniendo los mismos epochs(100):

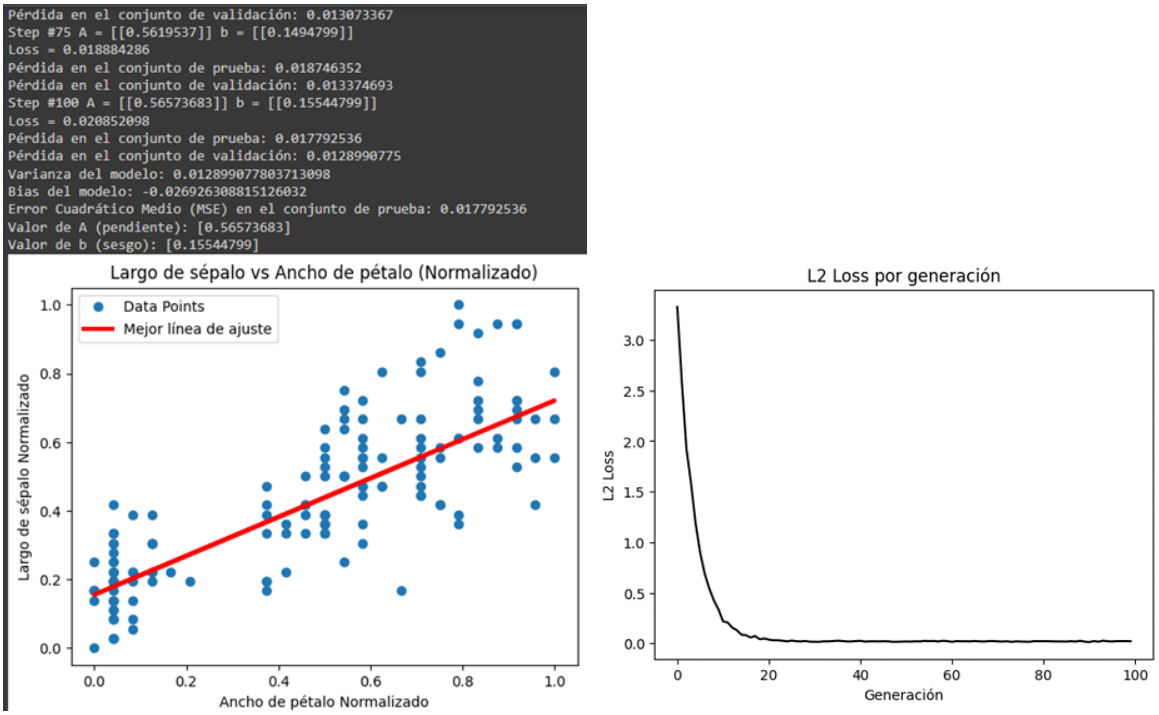

Imagen 4(LR = 0.03, epochs= 100)

Como se puede apreciar, los valores cambia debido a la normalización de los datos, pero considerando a estas como magnitudes podemos apreciar que el MSE, es el más bajo de los que se han visto así como la varianza y el sesgo del mismo, en el caso de este último se mantiene como negativo, con respecto al ritmo de aprendizaje es evidente que la gráfica tiene una líneas que podríamos considerar más regular ya que carece de picos extremos, esto como resultado de la normalización de los datos.

Como adición a la prueba anterior, también se realizaron dos más donde se mantiene el "learning rate" pero se modifican los epochs para evidenciar las consecuencias:

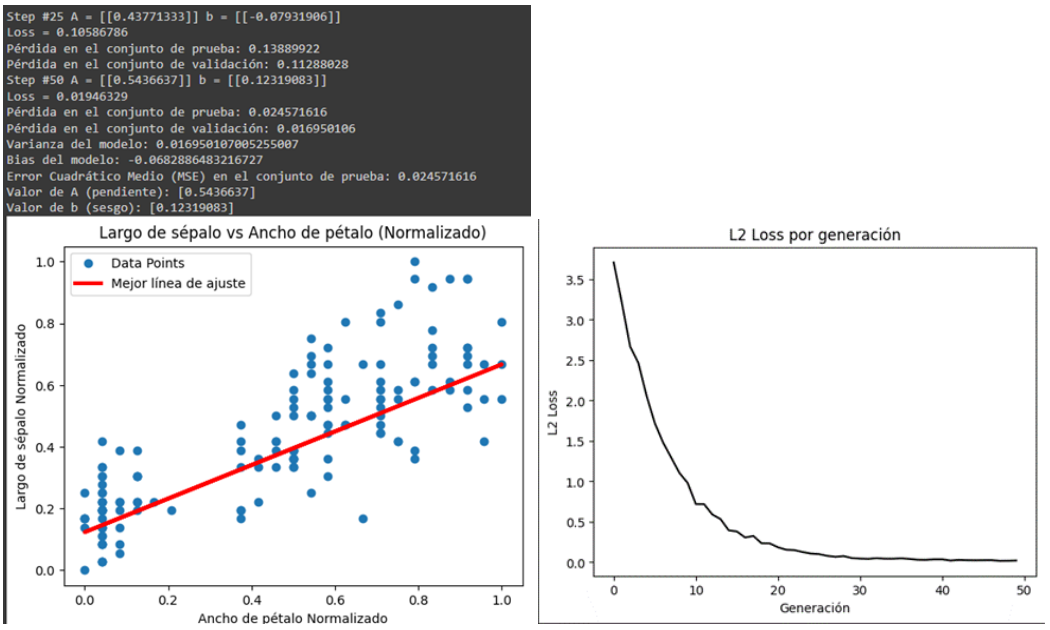

Imagen 5(LR = 0.03, epochs= 50)

Como se observa, la disminución del número de epochs resulta en un aumento considerable en el MSE, sesgo y varianza por lo que como primer punto a mencionar es que la cantidad de epochs resulta perjudicial para el modelo, dado que un MSE que tiende a 0 siempre es mejor.

Con lo anterior, podemos concluir que el algoritmo tiende a alterar en mayor o menor medida los resultados considerando que no hay un número mágico y universal para los modelos ya que este está fuertemente ligado al contexto.

Para terminar con las pruebas con respecto a los epochs, ahora tomamos un valor más alto (200):

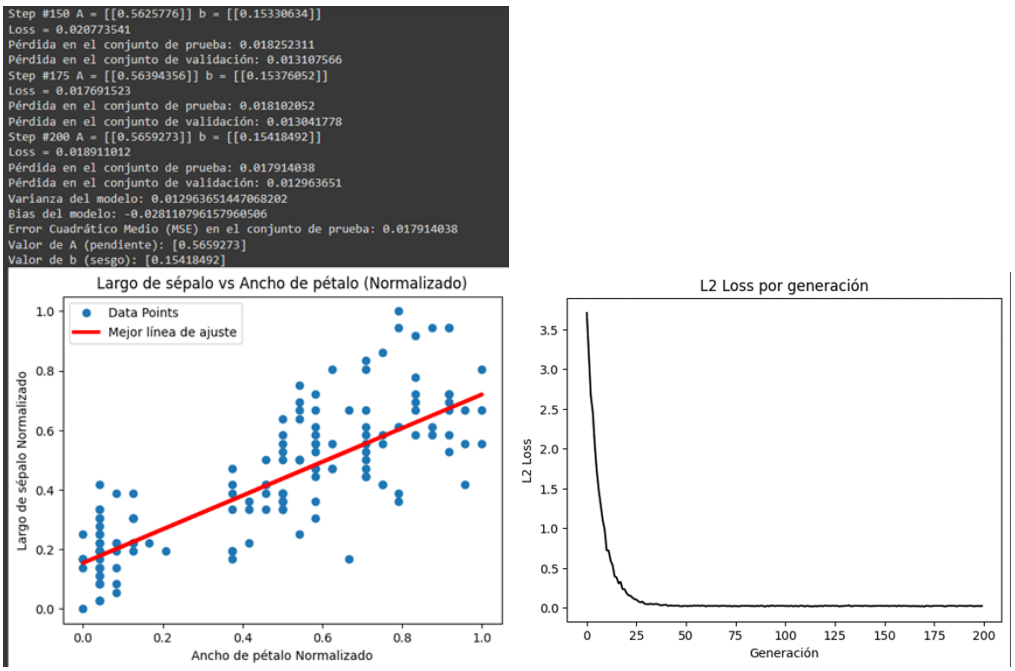

Imagen 6(LR = 0.03, epochs= 200)

En este caso podemos ver que el valor de MSE es mayor por un valor de 0.000121502, no parece significativo sin embargo con esto podemos concluir que 100 epochs es de todas las pruebas el que arroja un MSE más bajo.

Con este extenso set de pruebas podemos ver cómo afectan los valores al declarar el modelo por lo que como última estrategia para optimizar el modelo retomare una de las propuestas mencionadas al inicio de este reporte y es la de modificar los valores dentro de “Train”, Test“ y “Val” buscando el más óptimo, por lo que para las siguientes demostraciones daremos más peso al “Train”con un (80,10,10), pero dejando un “Learning rate” de 0.3 y 100 en la cantidad de epochs, con esto claro, estos son los resultados:

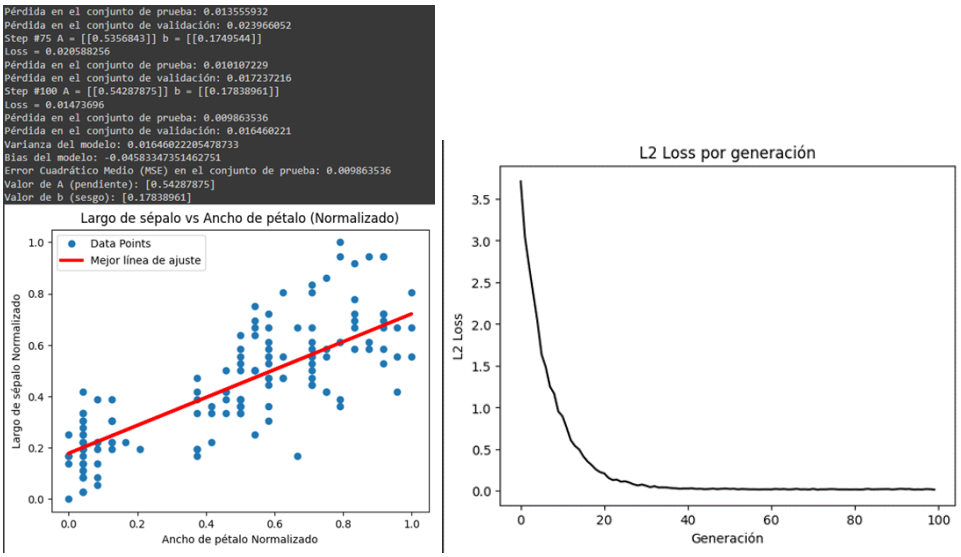

Imagen 7(LR = 0.03, epochs= 100, (80,10,10))

Con esta última prueba podemos ver que el valor de MSE es el más pequeño de los vistos sin embargo el sesgo aumento con respecto a los demás, todo esto debido a la naturaleza del modelo, en este caso no probable ver un caso de “Overfit” ya que es una simple regresión lineal, aun con eso en las pruebas se vio un caso de “Underfit”.

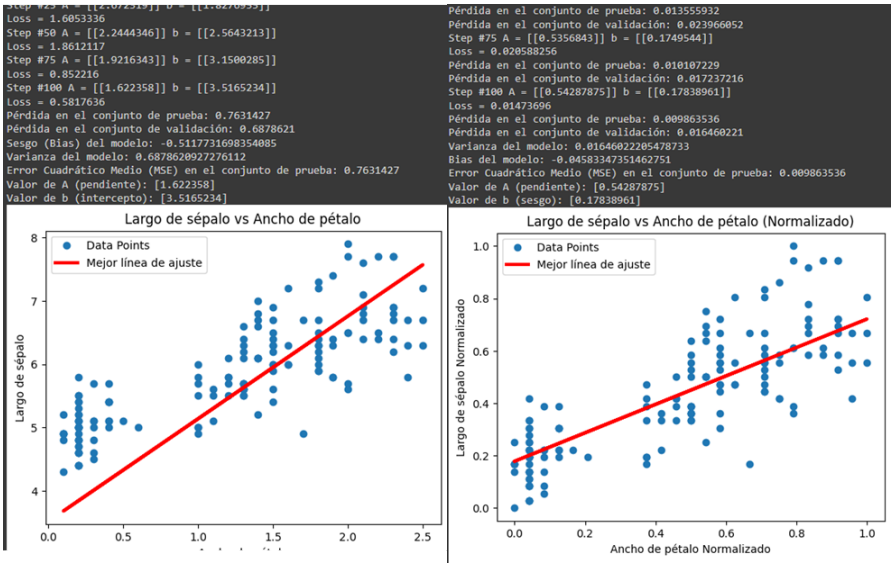

Imagen 7 Comparación del modelo inicial y el último realizado

Para concluir, considero cualquier problemas va a tener resultados diferentes con respecto al enfoque se le de y el cómo se trabaja, a lo largo de este reporte se pudo ver que un simple número puede afectar de forma significativa el resultado final del modelo que trabajamos y si consideramos que este es el más simple de los modelos ya podemos darnos una idea de los complejo que es plantear soluciones utilizando este tipo de tecnologías y metodologías.