Proyecto 1

1st Ian Augusto Cortez Gorbalan
ian.cortez@utec.edu.pe
970895554

2nd Plinio Matías Avendaño Vargas
plinio.avendano@utec.edu.pe
927144823

I. Introducción

Este proyecto busca emplear un algoritmo de regresión multivariada para realizar predicciones en base a datos reales. En el presente trabajo se va a emplear un *dataset* de incendios forestales en la región noreste de Portugal, en el parque Montesinho. En este *dataset*, se tienen 12 variables basadas en datos meteorológicos de la zona y una variable que representa el área del incendio, medida en hectáreas. El objetivo principal es lograr predecir el área del terreno que se puede incinerar en base a las características del mismo. Asimismo, como objetivo secundario, se busca determinar una cantidad de épocas y determinar el parámetro alfa que permita reducir el error de aproximación de nuestro modelo para el *dataset* utilizado.

II. Explicación del modelo empleado

A. Regresión lineal multivariada

La regresión lineal multivariada permite generar un modelo el cual aproxima o predice los valores dependientes de un dataset con K variables independientes. Formando una ecuación de regresión lineal donde se incluyen las K-variables, generando el siguiente hiper-plano

$$\sum_{i=1}^{k}(w_i * x_i) + b$$

B. Entrenamiento del modelo

Se entrenó el modelo con una cantidad de épocas determinadas y un alfa que luego se fue modificando. Cabe destacar que, el b elegido es un valor aleatorio al igual que los 12 valores del vector w. Por cada época, b y w eran actualizados empleando la derivada parcial de la función de costo respectivamente.

$$L = \frac{1}{2n}\sum_{i=1}^{n}(y_i - h(x_i))^2$$

$$\frac{\partial L}{\partial b} = \frac{1}{n}\sum_{i=1}^{n}(y_i - h(x_i))(-1)$$

$$\frac{\partial L}{\partial w_j} = \frac{1}{n}\sum_{i=1}^{n}(y_i - h(x_i))(-x_j^i)$$

III. Experimentación

A. Proceso

1) Conversión de variables categóricas: Lo primero a realizar fue la conversión de las dos variables categóricas nominales (día y mes) a su respectivo valor numérico.

2) Creación de train, testing y validation: Se separó aleatoriamente el *dataset* modificado, en 70%, 20% y 10%, para train, validation y test respectivamente.

3) Normalización de la data: Debido a la naturaleza de nuestra data, la cual presenta distintas unidades de medida, tales como viento (km/h), lluvia (mm/m^2) o el área en hectáreas, es necesario estandarizar todas las unidades de medida para así lograr que estén en un mismo rango (0-1) y las derivadas se comporten de manera adecuada (cambios muy bruscos en valores para un mismo *sample*, generan comportamientos anormales) tal como se comprobó, cuando intentamos una regresión sin la normalización obteniendo el siguiente resultado.

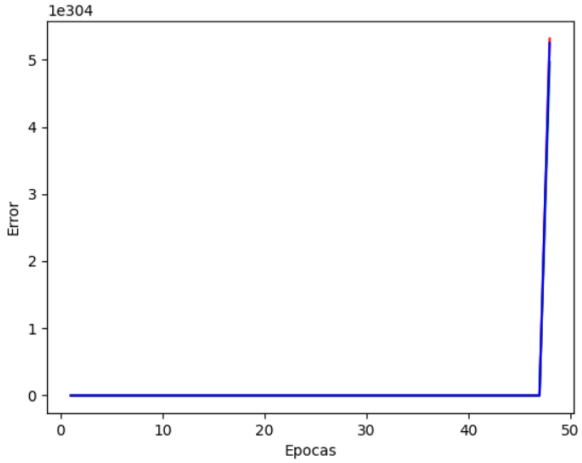

Fig. 1: Regresión sin normalización

Después de ese resultado empleamos la siguiente formula para cada dato. $z_i = (x_i - min(x))/(max(x) - min(x))$ Esto permite que cada dato se encuentre en el rango de 0-1, debido a que para que la división sea mayor a 1 se tendría que cumplir que:

$$x_i - min(x) > max(x) - min(x)$$

$$x_i > max(x)$$

Lo cual es una contradicción.

4) Determinación de hiper-parámetros: El umbral lo establecimos en 0.01, las épocas en 2000 (ya que observamos que alrededor de este numero los cambios en el error son insignificantes) y el alfa mediante prueba y error. A continuación, se presentan las gráficas mostrando el error obtenido

para los datos de entrenamiento y de validación respecto a los diferentes valores de alfa probados. Cabe destacar que, la línea verde representa el error de validación, la roja el de entrenamiento, y la azul el de *testing*.

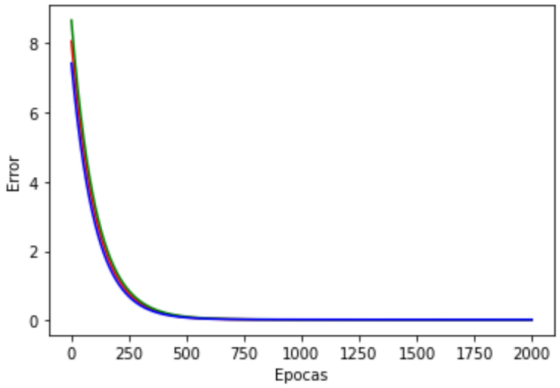

Fig. 2: Gráfico para $alfa = 0.00325$

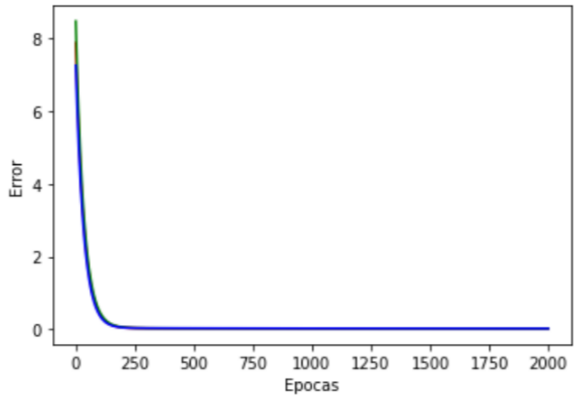

Fig. 3: Gráfico para $alfa = 0.01$

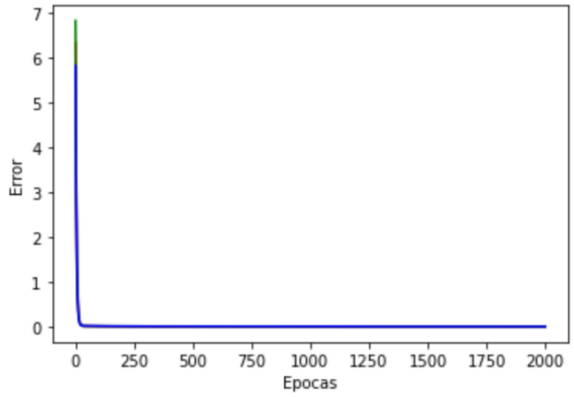

Fig. 4: Gráfico para $alfa = 0.08$

Se muestra la tabla para los errores de *testing* con los modelos generados para los valores de alfa probados.

IV. Resultados

A. Análisis del learning rate

Debido a que cuando el parámetro alfa vale 0.08, el error disminuye de manera extremadamente veloz (para los tres datasets) a comparación con las anteriores pruebas, determinamos empíricamente que ese es el valor adecuado.

TABLE I: Tablas de errores para los datos de *testing*

Valor de alfa	Error
0.00325	0.024744976372636583
0.01	0.017477620975919286
0.08	0.01255557275663338

B. Análisis de overfitting y underfitting

En base a las gráficas de los resultados se puede observar que los errores disminuyen de manera paralela para los tres datasets por lo cual no se presenta un *overfitting* ya que el modelo logra adaptarse a distintas datas. Del mismo modo, tampoco presenta un *underfitting* puesto que el hiper-plano generado logra disminuir el error de manera paulatina e iterativa. Haciendo el modelo bastante efectivo para predecir información.

V. Conclusiones

En conclusión, se logro generar un modelo efectivo que permite la determinación de la variable dependiente del dataset asignado, llegamos a esta afirmación debido a a la reducción del error en las diferentes segmentaciones de data.

VI. Enlace del código

https://colab.research.google.com/drive/1d64OqGRhDJF8AP3aCi9V9sT_BmbBdmS8?usp=sharing

References

[1] Aruchamy, V. (2021, 14 septiembre). How To Normalize Data Between 0 And 1 Range? Stack Vidhya. https://www.stackvidhya.com/how-to-normalize-data-between-0-and-1-range/

[2] Deisenroth, M. P., Faisal, A. A., & Ong, C. S. (2021). Mathematics for Machine Learning (English Edition). Cambridge University Press.