**Universidad Autónoma de Querétaro**

Facultad de Ingeniería

Maestría en Inteligencia Artificial

Machine Learning

Cecilia Gabriela Rodríguez Flores

Sheila Leyva López

Gradiente Descendente

**12 de diciembre del 2022**

# Objetivo

# Implementar el algoritmo del gradiente descendente para encontrar los hiperparámetros θ, tal que, la función de coste J(θ), que en este caso corresponde al MSE, minimice su valor.

# Introducción

En aprendizaje profundo usualmente se trabaja con redes neuronales que incluyen muchas variables, entonces es necesario el uso del gradiente descendente para encontrar aquellos parámetros que permitan minimizar la función de coste, lo que a su vez permite optimizar el algoritmo diseñado. En este trabajo se utiliza el gradiente descendiente para reducir el valor el valor del MSE, sin embargo, también se puede utilizar cuando se requiere hacer clasificación, y entonces la función de coste es la entropía cruzada; incluso también se puede aplicar a funciones de coste que defina y requiera el autor, siempre y cuando la función de coste cumpla con los requerimientos para ser una función de coste.

En este trabajo se aplica dicha técnica de agrupamiento a la base de datos *Indicadores personales clave de enfermedad cardíaca.* La base de datos.

Específicamente se presenta la implementación del modelo matemático: Gradiente Descendente, con el objetivo de encontrar los hiperparámetros θ, tal que, permitan optimizar la función de coste del modelo de regresión lineal entre los datos.

# Marco Teórico

Para la comprensión de la metodología y resultados de este trabajo es necesario definir otros modelos matemáticos, tal como, regresión lineal, covarianza, correlación de Pearson y función de costo.

### Regresión lineal

La regresión lineal es un análisis, cuyo principal objetivo es la predicción de una variable dependiente, a partir del valor de otra variable, llamada independiente [1], [2], dicho enunciado se puede representar a partir de la siguiente ecuación matemática (adaptada de [2]):

Donde:

* .
* x: datos.

En otras palabras, es una hipótesis de cómo es el comportamiento de los datos en x.

### Regresión logística

Ahora bien, la regresión logística es una técnica matemática utilizada en aprendizaje automático para la clasificación de datos. Básicamente, es una neurona. Recibe este nombre dado que, primero se realiza una combinación lineal y después se aplica una función logística, aunque no es una regresión sino un modelo de clasificación [3]. El proceso se describe a través del diagrama de la Ilustración 1. Básicamente, lo que en regresión lineal se denominaba hipótesis ahora es **Z** y es la función de transferencia definida por la función sigmoidea (ver Ilustración 2).

Diagrama

Descripción generada automáticamente

*Ilustración 1. Esquema regresión logística, imagen adaptada de .*

Donde:

* X: atributos
* W: coeficientes o pesos.
* : función logística sigmoide

Diagrama

Descripción generada automáticamente con confianza media

**Dato clasificado**

*Ilustración 2. Función sigmoide. Imagen adaptada de .*

Para medir el rendimiento de un modelo, en este caso el modelo de regresión lineal, existe una función, llamada función de costo o coste , la cual calcula el error entre los valores esperados o predichos por y los valores reales .

Dicha función de coste, en el mejor de los casos, será igual al error que se desea minimizar u optimizar, existen dos tipos de errores:

* Error Cuadrático Medio (MSE, por sus siglas en inglés)(ecuación adaptada de [8]):
* Error Absoluto Medio (MAE, por sus siglas en inglés):

Donde:

Mencionado todo lo anterior, existe un método matemático que permite encontrar el valor mínimo de la función de coste, el método del gradiente descendente.

El gradiente, se considera como la versión generalizada de la derivada en cálculo diferencial de una variable. Es decir, el gradiente es el conjunto de las derivadas parciales de una función multivariable, además, como se busca el valor mínimo de la función, el gradiente es negativo o descendente [].

El gradiente descendente para el caso particular en el que la función de coste es el MSE, se define como[2,3]:

Para el parámetro :

De la ecuación (6) y (7) resulta la ecuación (8):

Para el parámetro :

De la ecuación (9) y (10) resulta la ecuación (11):

Donde:

* : número total de datos.

### Learning rate

En el aprendizaje automático y las estadísticas, la tasa de aprendizaje o learning rate (lr) es un parámetro de ajuste en un algoritmo de optimización que determina el tamaño del paso en cada iteración mientras avanza hacia un mínimo de una función de pérdida. Dado que influye en qué medida la información recién adquirida anula la información anterior, representa metafóricamente la velocidad a la que "aprende" un modelo de aprendizaje automático.

### Regla 60-20-20

Este método consiste en la creación de subconjuntos, en los cuales el 60% de los datos de considera para el entrenamiento, mientras que un 20% es para la validación y otro 20% corresponde a pruebas del modelo de clasificación.

**Métricas**

### Exactitud

La exactitud es la relación entre los simples correctamente clasificados y el número total de simples en el conjunto de datos de evaluación. Esta métrica es conocida por ser engañosa en el caso de diferentes proporciones de clases, ya que asignar simplemente todos los simples a la clase predominante es una forma fácil de lograr una alta precisión [2].

|  |  |
| --- | --- |
|  | (12) |

# Materiales y métodos

### Herramientas utilizadas

* Google Collaboratory
* Conjunto de datos: *Indicadores personales clave de enfermedad cardíaca*
* AMD Ryzen 9 5900HS with Radeon Graphics 3.30 GHz
* RAM 16.0 GB
* 64-bit operating system, x64-based processor

### Conjunto de datos

En este trabajo, se utilizará el conjunto de datos: *Indicadores personales clave de enfermedad cardíaca*, datos provenientes de 400,000 adultos, obtenidos durante la encuesta anual 2020 de los Centros para el Control y prevención de Enfermedades (CDC, por sus siglas en inglés) pertenecientes al departamento de salud y servicios humanos en los Estados Unidos. Originalmente el conjunto de datos contenía alrededor de 300 atributos, sin embargo, se redujo a solo 18 variables, los cuales son los que se encuentran disponibles públicamente en la plataforma Kaggle [].

Casi la mitad de los estadounidenses (47%), incluyendo afroamericanos, indios americanos, nativos de Alaska y blancos; tienen al menos de 1 a 3 factores de riesgo de padecer alguna enfermedad cardíaca. A continuación, se agrega una breve descripción de los atributos incluidos en este conjunto de datos:

* HeartDisease: (atributo de decisión): personas encuestadas que informaron alguna vez haber padecido alguna enfermedad coronaria (CHD, por sus siglas en inglés) o infarto al miocardio(IM, por sus siglas en inglés).
* BMI: Índice de Masa Corporal.
* Smoking: personas encuestadas que han fumado al menos 100 cigarros en su vida entera.
* AlcoholDrinking: corresponde a hombres adultos que beben más de 14 tragos por semana y mujeres adultas que beben más de 7 tragos por semana.
* Stroke: responde a la pregunta: ¿alguna vez le dijeron o usted tuvo un derrame cerebral?
* PhysicalHealth: incluyendo enfermedades y lesiones físicas, responde a la pregunta: ¿durante cuántos días en los últimos 30 días su salud física no fue buena? (de 0 a 30 días)
* MentalHealth: ¿durante cuántos días en los últimos 30 días su salud mental no fue buena? (de 0 a 30 días).
* DiffWalking: responde a ¿tiene serias dificultades para caminar o subir escaleras?
* Sex: hombre o mujer.
* AgeCategory: 14 rangos de edad.
* Race: valor de raza / etnicidad imputada.
* Diabetic: responde a ¿alguna vez ha sido diagnosticada con diabetes?
* PhysiclActivity: adultos que informaron haber realizado actividad física o ejercicio en los últimos 30 días, no incluyendo su trabajo habitual.
* GenHealth: responde a ¿cómo calificarías tu salud en general?
* SleepTime: responde a un promedio de horas que duerme, en un periodo de 24 horas, la persona encuestada.
* Asthma: responde a ¿alguna vez ha sido diagnosticado con asma?

### Métodos

Para la realización de esta tarea, fue necesario importar las librerías de Pandas, numpy, etc., así como el montaje de Google Drive en el entorno de ejecución a fin de poder hacer uso de la base de datos propuesta.

Posteriormente, se optó por crear diversas funciones a fin de distribuir las tareas de una manera más eficiente como se muestra a continuación:

* Función method\_602020: realiza la separación del conjunto de datos de acuerdo con el porcentaje de 60% para el conjunto de entrenamiento, 20% para el conjunto de validación y 20% para el conjunto de prueba. Esta función requiere como entradas los datos y las etiquetas.

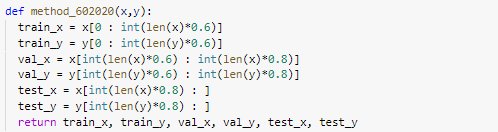


Ilustración 4. Función propuesta para la división del conjunto de datos.

* Función regre\_logistic: realiza la actualización del cálculo de los pesos de acuerdo con el número de épocas. Esta función requiere como entradas los valores de x, y, theta, alpha y épocas.

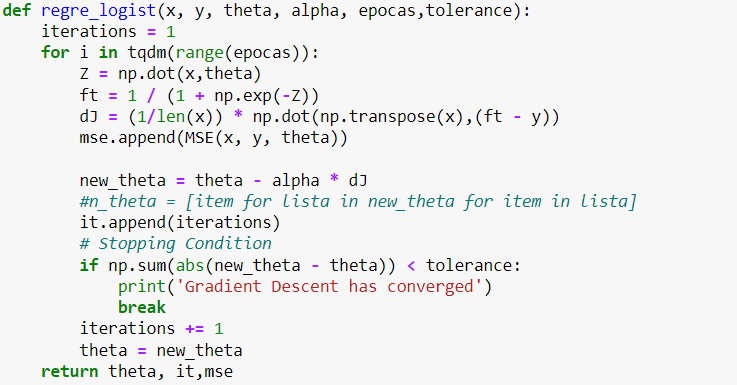


Ilustración 5. Función propuesta para el modelo de regresión logística.

* Función clasificacion: se enfoca en la obtención de la predicción de los valores de y. Esta función requiere de como entradas los valores de x y los pesos obtenidos de la etapa de entrenamiento.



Ilustración 6. Función propuesta para la obtención de la clasificación.

* Función MSE: se enfoca calcular la función de costo. Esta función requiere de como entradas los valores de y, x y thetas.

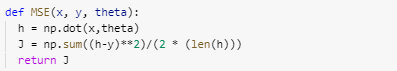


Ilustración 7. Función propuesta para la función de costo.

* Función accuracy: se enfoca en obtener la métrica de exactitud a fin de poder analizar el comportamiento del modelo. Esta función requiere como entradas los datos originales de y y los valores predichos.

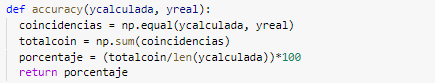


Ilustración 8. Función propuesta para la obtención de la métrica de exactitud.

### Diagrama de metodología

Para el análisis de cualquier base de datos se deben seguir los siguientes pasos:

1. Recolección: Definir y cargar la base de datos a utilizar.
2. Preparar: Comprobar la no existencia de datos faltantes, identificación de outliers, así como la normalización de los datos.
3. Analizar: Conocer la distribución de la información por cada atributo, reducción de dimensionalidad, división de los conjuntos de entrenamiento y prueba del modelo, asignando el 60% de ellos para el entrenamiento, 20% para la validación y el 20% restante a la etapa de prueba.
4. Etapa de entrenamiento y prueba: Implementación del modelo de regresión logística. Así bien, se calcula la métrica de exactitud para conocer el desempeño obtenido.

A continuación, se muestra el diagrama de flujo que representa el proceso de desarrollo, el cual fue implementado en un programa basado en lenguaje Python.

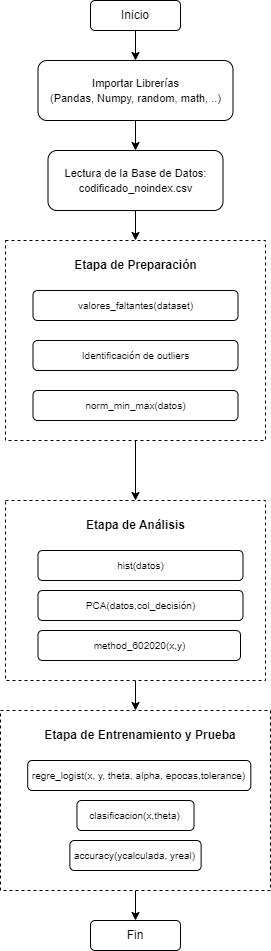


Ilustración 9. Diagrama de flujo para el análisis del conjunto de datos.

# Resultados y discusión

A continuación, se presentan las distribuciones de los atributos de la base de datos en cuestión, al cual se le efectuó una preparación, la cual comprende los siguientes puntos: la comprobación de la no existencia de datos faltantes, la identificación de outliers y la normalización de los datos.

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Ilustración 10. Distribución de los atributos de la base de datos.

A partir de haber implementado la técnica de análisis de componentes principales (PCA) en la base de datos en cuestión, se logró reducir la cantidad de atributos de 18 a 12, entre los atributos que contenían la mayor información se encuentran: BMI, PhysicalHealth, MentalHealth, AgeCategory, Race, Diabetic, Smoking, AlcoholDrinking, Stroke, DiffWalking y PhysicalActivity. Así bien, para los siguientes pasos se incluye a estos atributos seleccionados el atributo de decisión HeartDisease.

Antes de proseguir con la implementación de algoritmos de agrupamiento será necesario dividir el conjunto de datos en un conjunto de entrenamiento y un conjunto de prueba. En la Tabla 1 se presentan el total de los datos para cada uno de los conjuntos, en donde se utilizó la división 60/20/20.

Tabla 1. Conjunto de Entrenamiento, Validación y Prueba.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Conjunto de Entrenamiento | Conjunto de Validación | Conjunto de  Prueba |
| Total de datos | 191877 | 63959 | 63959 |

### Regresión Logística

A continuación, se pueden observar los parámetros propuestos para la implementación del modelo de Regresión Logística. Cabe mencionar que los valores iniciales de theta se obtuvieron de manera aleatoria. Así bien, se utilizaron únicamente los atributos con mayor relevancia en la base de datos: BMI y PhysicalHealth.

# Épocas = 10, alpha= 0.6

En la Ilustración 11, se puede observar el tiempo que tardo el modelo en la etapa de entrenamiento. Así bien, en esta ocasión el modelo no convergió.



Ilustración 11. Tiempo de entrenamiento del modelo regresión logística.

En la Ilustración 12, se puede visualizar como va modificando el valor de la función de costo con respecto al número de iteración en cuestión, en donde se puede observar que al incrementar el número de iteraciones el valor de la función de costo va disminuye hasta la iteración número 8.

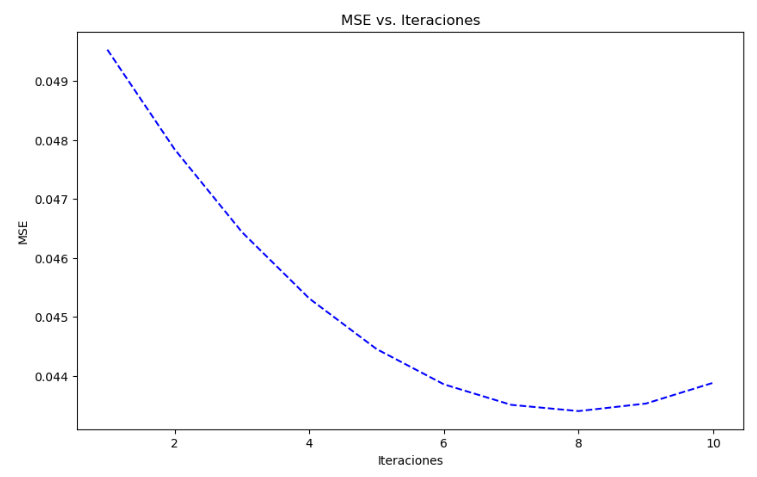


Ilustración 12.Función de costo vs Iteraciones.

Por otra parte, cabe mencionar que los valores obtenidos en la métrica de exactitud para los conjuntos de validación y prueba no supera el 80 porciento. Mientras que los valores obtenidos en la función de costo se encuentran muy cercanos al cero.

*Tabla 2. Resultados obtenidos en la métrica de exactitud y la función de costo.*

|  |  |  |
| --- | --- | --- |
|  | Exactitud | MSE |
| Conjunto de Validación | 78.64569489829421 | 0.10316054658256586 |
| Conjunto de Prueba | 78.27670851639331 | 0.06822169767905832 |

# Épocas = 100, alpha= 0.6

En la Ilustración 13, se puede observar el tiempo que tardo el modelo en la etapa de entrenamiento. Así bien, en esta ocasión el modelo no convergió.



Ilustración 13. Tiempo de entrenamiento del modelo regresión logística.

En la Ilustración 14, se puede visualizar como va modificando el valor de la función de costo con respecto al número de iteración en cuestión, en donde se puede observar que al incrementar el número de iteraciones el valor de la función de costo va incrementando de igual forma.

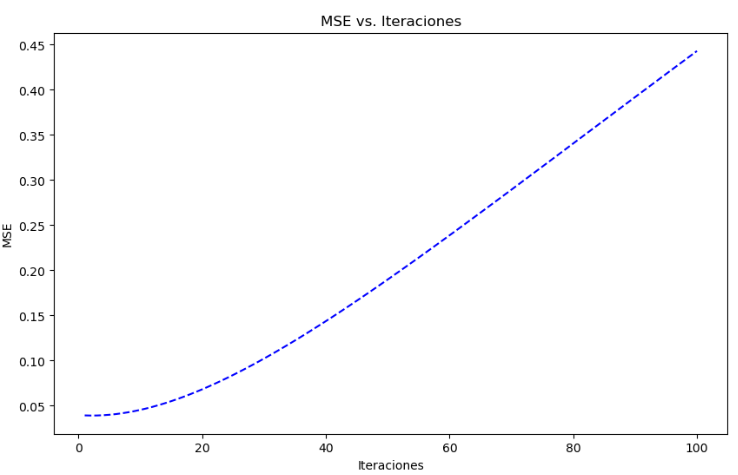


Ilustración 14. Función de costo vs Iteraciones.

Así mismo, los valores obtenidos en la métrica de exactitud para los conjuntos de validación y prueba supera el 90 por ciento. Mientras que los valores obtenidos en la función de costo se encuentran muy cercanos al cero.

*Tabla 3. Resultados obtenidos en la métrica de exactitud y la función de costo.*

|  |  |  |
| --- | --- | --- |
|  | Exactitud | MSE |
| Conjunto de Validación | 91.29285948810957 | 0.3353174145500401 |
| Conjunto de Prueba | 91.57585328100815 | 0.3353174145500401 |

# Épocas = 1000, alpha= 0.6

En la Ilustración 15, se puede observar el tiempo que tardo el modelo en la etapa de entrenamiento. Así bien, en esta ocasión el modelo no convergió.



Ilustración 15. Tiempo de entrenamiento del modelo regresión logística.

En la Ilustración 16, se puede visualizar como va modificando el valor de la función de costo con respecto al número de iteración en cuestión, en donde se puede observar que al incrementar el número de iteraciones el valor de la función de costo va incrementando de igual forma.

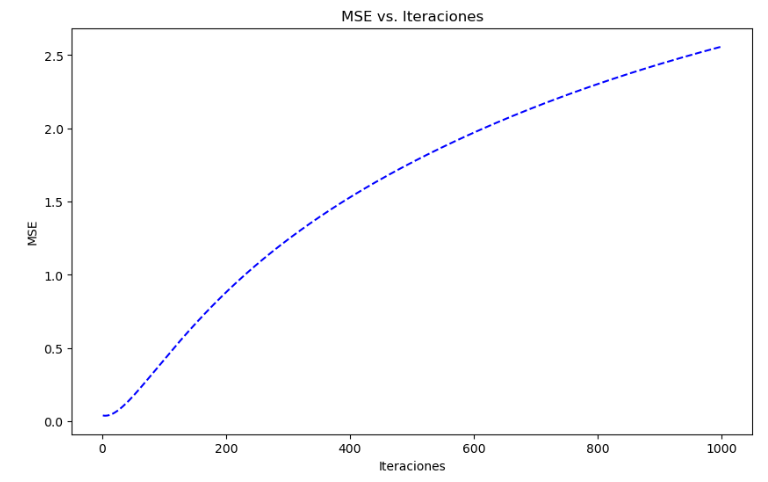


Ilustración 16. Función de costo vs Iteraciones.

Así mismo, los valores obtenidos en la métrica de exactitud para los conjuntos de validación y prueba supera el 90 por ciento. Mientras que los valores obtenidos en la función de costo para los dos conjuntos de datos varían únicamente por milésimas.

*Tabla 4. Resultados obtenidos en la métrica de exactitud y la función de costo.*

|  |  |  |
| --- | --- | --- |
|  | Exactitud | MSE |
| Conjunto de Validación | 91.20686689910724 | 2.3561674902524605 |
| Conjunto de Prueba | 91.49611469847872 | 2.353016662618978 |

# Épocas = 1000000, alpha= 0.6

En la Ilustración 17, se puede observar el tiempo que tardo el modelo en la etapa de entrenamiento, y la época en la cual el modelo convergió.

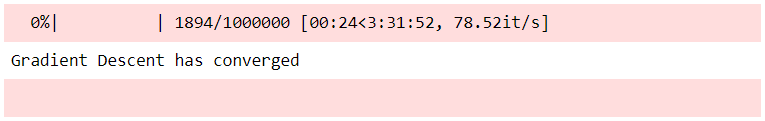


Ilustración 17. Tiempo de entrenamiento del modelo regresión logística.

En la Ilustración 18, se puede visualizar como va modificando el valor de la función de costo con respecto al número de iteración en cuestión, en donde se puede observar que al incrementar el número de iteraciones el valor de la función de costo aumenta de igual manera.

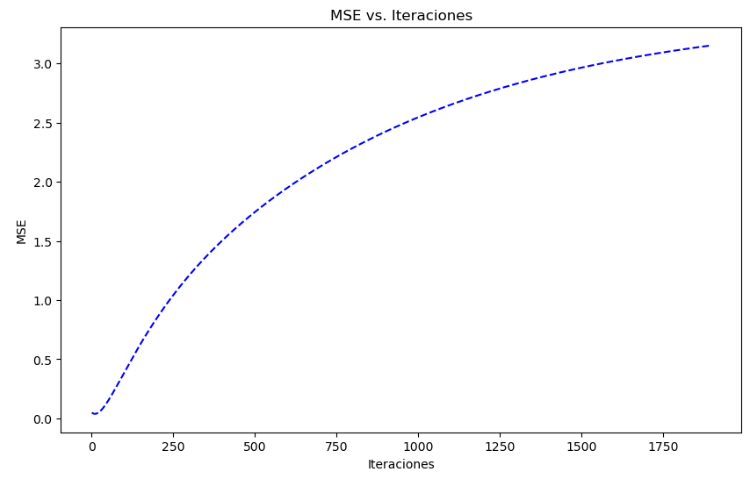


Ilustración 18. Función de costo vs Iteraciones.

Así mismo, los valores obtenidos en la métrica de exactitud para los conjuntos de validación y prueba supera el 90 porciento. Mientras que los valores obtenidos en la función de costo para los conjuntos de validación y prueba se diferencian por 0.60.

*Tabla 5. Resultados obtenidos en la métrica de exactitud y la función de costo.*

|  |  |  |
| --- | --- | --- |
|  | Exactitud | MSE |
| Conjunto de Validación | 91.04113572757548 | 2.940897765695596 |
| Conjunto de Prueba | 91.35227254960209 | 2.3448863417060495 |

# Épocas = 1000, alpha= 0.01

En la Ilustración 19, se puede observar el tiempo que tardo el modelo en la etapa de entrenamiento. Así bien, en esta ocasión el modelo no convergió.



Ilustración 19. Tiempo de entrenamiento del modelo regresión logística.

En la Ilustración 20, se puede visualizar como va modificando el valor de la función de costo con respecto al número de iteración en cuestión, en donde se puede observar que para la iteración número 550 aproximadamente se consigue un menor valor en la función de costo.

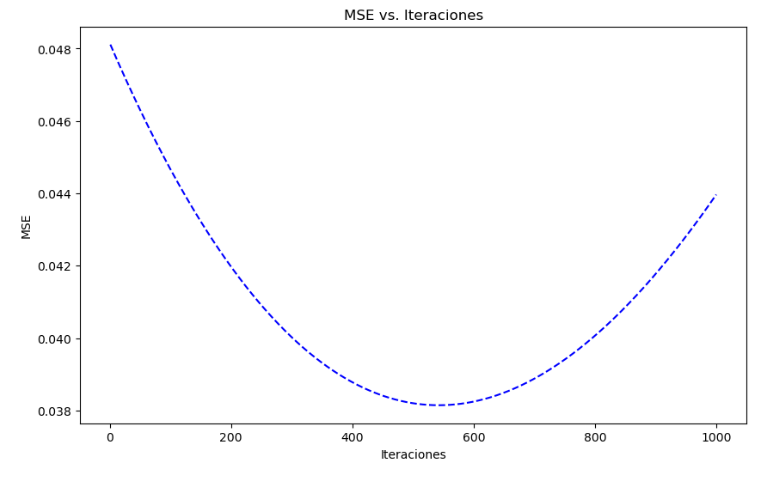


Ilustración 20. Función de costo vs Iteraciones.

Así mismo, los valores obtenidos en la métrica de exactitud para los conjuntos de validación y prueba no supera el 90 por ciento. Mientras que los valores obtenidos en la función de costo se encuentran muy cercanos al cero.

*Tabla 6. Resultados obtenidos en la métrica de exactitud y la función de costo.*

|  |  |  |
| --- | --- | --- |
|  | Exactitud | MSE |
| Conjunto de Validación | 85.20614768836285 | 0.053056347265617955 |
| Conjunto de Prueba | 85.5125940055348 | 0.05043447115978054 |

En la siguiente tabla, se puede realizar una comparación entre las diferentes pruebas realizadas de acuerdo con la variación en el número de épocas, en donde se puede observar que el mejor porcentaje de exactitud se obtuvo en la época 100 tanto para el conjunto de validación como para el conjunto de prueba. Por otra parte, el peor porcentaje obtenido en la métrica de exactitud fue en la época 10, tanto para el conjunto de validación como para el conjunto de prueba.

*Tabla 7. Comparación de los resultados obtenidos con un learning rate = 0.6.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Épocas | Conjunto de Validación | | Conjunto de Prueba | |
| Exactitud | MSE | Exactitud | MSE |
| 10 | 78.64569 | 0.10316 | 78.27670 | 0.06822 |
| 100 | 91.29285 | 0.33531 | 91.57585 | 0.33531 |
| 1000 | 91.20686 | 2.35616 | 91.49611 | 2.35301 |
| 1000000 | 91.04113 | 2.94089 | 91.35227 | 2.34488 |

En la tabla 8, se puede realizar una comparación entre la variación del learning rate (0.6 y 0.01) para un número de épocas de 1000, en donde se pudo observar un resultado en la métrica de exactitud mayor al 90 por ciento al proponer un learning rate de 0.6, en comparación al 85 por ciento obtenido con un learning rate de 0.01.

*Tabla 8. Comparación de los resultados obtenidos en la métrica de exactitud con 1000 épocas.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Learning rate | Conjunto de Validación | | Conjunto de Prueba | |
|  | Exactitud | MSE | Exactitud | MSE |
| 0.6 | 91.20686 | 2.35616 | 91.49611 | 2.35301 |
| 0.01 | 85.20614 | 0.05305 | 85.51259 | 0.05043 |

# Conclusiones

En este programa generalizado inicialmente se generaron funciones que permitieran conocer el comportamiento de los datos mediante la regresión logística, la cual permite describir y estimar la relación entre una variable binaria dependiente y las variables independientes. Así bien, en comparación con otros algoritmos de machine learning, este no requiere de demasiados recursos computaciones, como se pudo comprobar anteriormente ya que no fue necesario la utilización de un subsampling del conjunto de datos, además el tiempo de ejecución fue considerablemente rápido si se compara con otros algoritmos.

Asimismo, la implementación del gradiente descendente permitió encontrar de forma automática el mínimo de la función de error MSE, mediante la determinación de parámetros theta para la función de regresión logística. A partir de las pruebas de variación de los parámetros: learning rate y el número de épocas de entrenamiento, se puede decir que un número de learning rate cuando es demasiado grande provoca que los cambios en los hiperparámetros theta sean también muy grandes y resulte más difícil encontrar los coeficientes que minimicen la función de costo, esto mismo sucede cuando el valor del learning rate es demasiado pequeño. Para este conjunto de datos en específico, los mejores resultados se obtuvieron cuando el valor de learning rate era alto (0.6) y el número de épocas pequeño (100), así como cuando el learning rate era bajo (0.01) y el número de épocas alto (1000).

# Referencias

[1] “Acerca de la regresión lineal - México | IBM.” [Online]. Available: https://www.ibm.com/mx-es/analytics/learn/linear-regression].

[2] “Regresión Lineal: teoría y ejemplos en Python - IArtificial.net.” [Online]. Available: https://www.iartificial.net/regresion-lineal-con-ejemplos-en-python/.

[3] “Gradiente Descendiente para aprendizaje automático - IArtificial.net.” [Online]. Available: https://www.iartificial.net/gradiente-descendiente-para-aprendizaje-automatico/.

[4] “Error Cuadrático Medio para Regresión - IArtificial.net.” [Online]. Available: https://www.iartificial.net/error-cuadratico-medio-para-regresion/.

[5] Martinez, J. (2020, September 21). Regresión Logística para la Clasificación. IArtificial.net. 2022, from https://www.iartificial.net/regresion-logistica-para-clasificacion/

[6] M. Antonio and A. Fernández, *Inteligencia artificial para programadores con prisa by Marco Antonio Aceves Fernández - Books on Google Play*. Universo de Letras. [Online]. <https://play.google.com/store/books/details/Inteligencia_artificial_para_programadores_con_pri?id=ieFYEAAAQBAJ&hl=en_US&gl=US>

[7] “Personal Key Indicators of Heart Disease | Kaggle.” https://www.kaggle.com/datasets/kamilpytlak/personal-key-indicators-of-heart-disease/code https://learn.microsoft.com/es-es/azure/machine-learning/component-reference/smote