Modelo de Regresión Lineal para la Predicción de Calorías

Abstract

Este documento explora la implementación de un modelo de regresión lineal múltiple para la predicción de calorías. Se menciona el dataset “Calories Burnt Prediction” y cómo se limpió, así como los criterios utilizados para seleccionar los datos pertinentes para la predicción. También se explica cómo funciona el modelo de regresión lineal, el cual presenta un error promedio aproximado de 10 calorías por predicción.

Introducción

El ser humano mantiene de manera constante un gasto energético basal, expresado en la quema de calorías, independientemente del tipo de actividad física realizada. Este gasto mínimo diario puede incrementarse en función de la intensidad, duración y tipo de ejercicio efectuado por la persona.

La estimación precisa del gasto calórico resulta compleja sin el uso de dispositivos especializados de medición. A pesar de ello, persisten creencias erróneas acerca de qué actividades favorecen en mayor medida la quema de calorías, como la comparación entre correr y realizar ejercicios de fuerza, o la influencia de factores como la edad.

En este trabajo se aborda la aplicación de técnicas de Machine Learning y de un modelo de regresión lineal múltiple para estimar el gasto calórico durante la actividad física. El enfoque considera parámetros corporales y el tiempo de ejercicio, con el propósito de aproximar de manera objetiva la cantidad de calorías consumidas.

Datos

Descripción del Conjunto de Datos

El *dataset* contiene 8 columnas: *User_Id, Gender, Age, Height, Weight, Duration, Heart_rate* y *Body_temp*. A continuación se presenta una breve descripción del significado de cada columna:

- **User_Id:** Identificador único para cada instancia, de tipo número entero.
- **Gender:** Texto con la palabra “female” para mujeres y “male” para hombres.

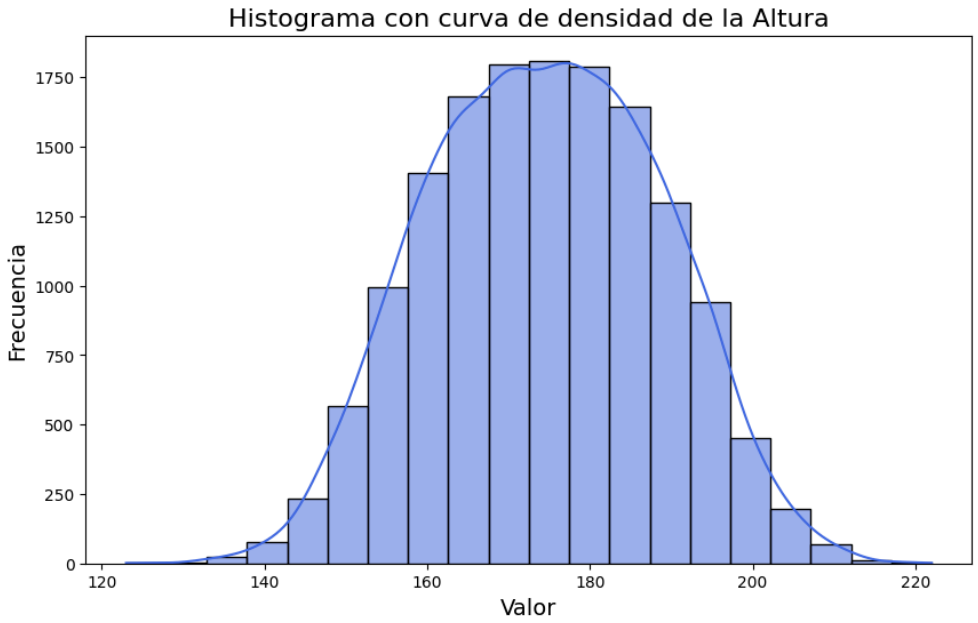
- **Age:** Número entero que representa la edad de la persona en años.
- **Height:** Número decimal que representa la estatura de la persona en centímetros.
- **Weight:** Número decimal que representa el peso de la persona en kilogramos.
- **Duration:** Minutos que duró el ejercicio.
- **Heart_rate:** Número de pulsaciones por minuto del corazón de la persona (bpm).
- **Body_temp:** Temperatura corporal de la persona en grados Celsius.

Limpieza del Conjunto de Datos

El primer paso consistió en la limpieza de los datos, verificando que no hubiera valores faltantes. Posteriormente, se revisó la existencia de valores atípicos, dado que el modelo de regresión lineal es sensible a este tipo de observaciones, por lo que se puso especial atención en esta etapa.

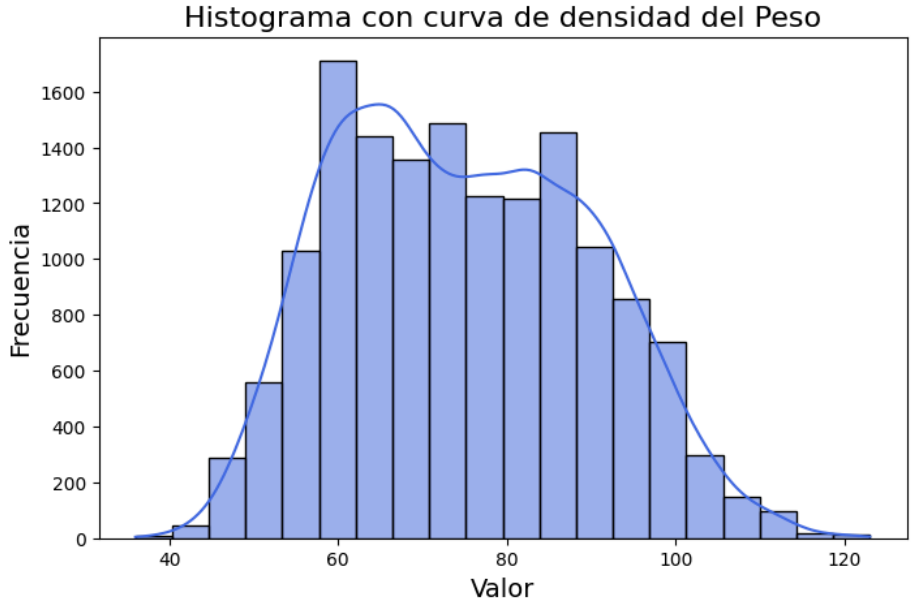
Altura

La variable de altura presentó una distribución cercana a la normal, como se observa en el histograma. Para identificar valores atípicos se aplicó un *z-test*, utilizando como criterio un valor de $Z = 3$. Toda instancia que se encontraba por encima de tres desviaciones estándar fue eliminada. De esta manera, se descartaron instancias como la de una persona con altura de 2.22 m, la cual, si bien corresponde a un valor real, no refleja una estatura promedio en la población adulta.



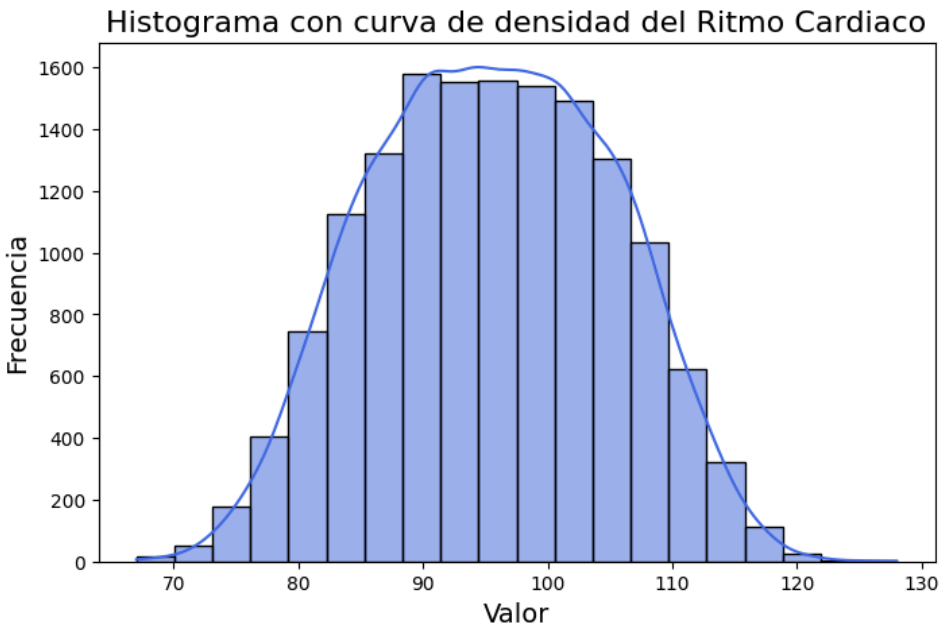
Peso

En el caso del peso, la distribución no seguía un patrón normal. Por ello, se empleó el rango intercuartílico (IQR) como método de detección de valores atípicos. Se eliminaron los valores menores a *Q1 – 1.5·IQR* y mayores a *Q3 + 1.5·IQR*.



Ritmo cardíaco

El ritmo cardíaco mostró una distribución aproximadamente normal, con valores mínimos de 67 bpm y máximos de 128 bpm. Estos valores se consideran normales durante la actividad física, por lo que no fue necesario aplicar un *z-test* ni eliminar instancias.



Temperatura corporal

Respecto a la temperatura corporal, se planteó eliminar valores extremos de 41.5 °C, ya que el rango tolerable para un adulto suele encontrarse entre 39 °C y 40 °C durante el ejercicio. Considerando la media (40 °C) y la desviación estándar (0.7 °C), se decidió eliminar los registros superiores a 40.7 °C, definidos como nuestro máximo aceptable en condiciones de esfuerzo físico.

Género

La columna correspondiente al género se transformó a valores numéricos enteros. Se asignó el valor 0 a “female” y el valor 1 a “male”. Dado que la variable sólo contenía dos categorías, no fue necesario aplicar *one-hot encoding*.

Partición del Conjunto de Datos

El *dataset* fue dividido en tres subconjuntos: *train*, *validation* y *test*. El conjunto train se utilizó para entrenar el modelo y representó aproximadamente el 70% de los datos. El conjunto *validation* se empleó para prevenir el sobreajuste (*overfitting*), Teníamos que ver que el error obtenido en train y validation fuera similar. Si el error de train era muy poco, mientras que el error de *validation* era alto, significaba que el modelo estaba aprendiendo datos y no patrones.

Finalmente, el conjunto test se reservó únicamente para la fase final, con el propósito de evaluar el desempeño del modelo en predicciones y verificar su comportamiento en un escenario real.

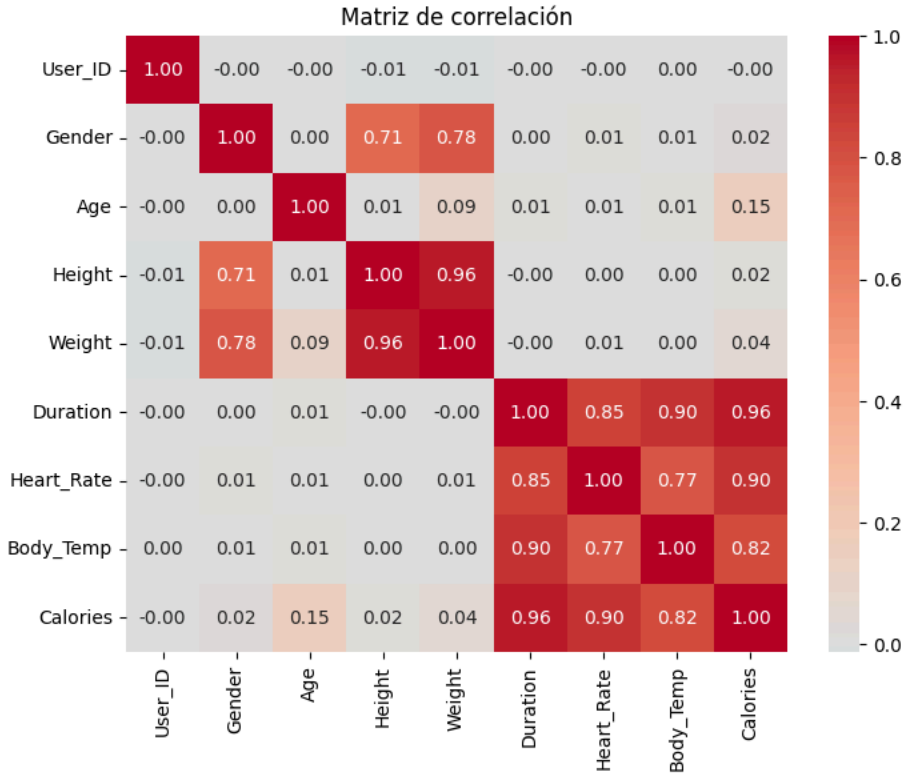
La partición de los datos se realizó con el apoyo de la librería scikit-learn.

Modelo de regresión lineal múltiple

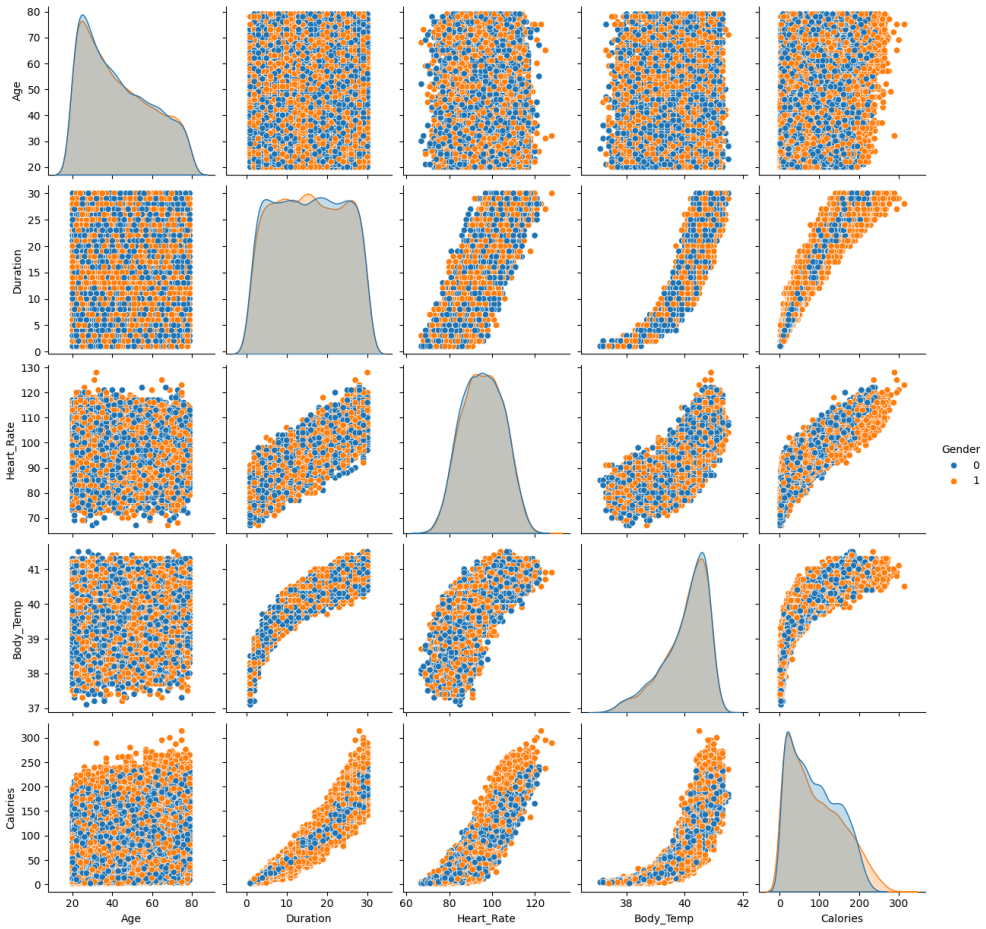
Selección de Datos

Para seleccionar las variables necesarias para el modelo de regresión, se construyó una matriz de correlación que muestra los coeficientes de correlación de Pearson entre variables. El coeficiente de correlación mide el grado en que dos variables se relacionan entre sí, con valores que van entre -1 y 1. Un valor cercano a 1 indica que cuando una variable aumenta, la otra también lo hace, mientras que un valor cercano a -1 indica que cuando una variable aumenta, la otra disminuye.

Entre las variables con mayor correlación con *Calories,* nuestra variable a predecir, se encuentran: *Duration*, *Body_temp* y *Heart_rate*, todas con coeficientes superiores a 0.8. Además, decidí incluir la variable Age, con el fin de evaluar su posible influencia sobre el gasto calórico.



Por otra parte, el análisis de dispersión confirma que *Duration*, *Body_temp* y *Heart_rate* presentan relaciones casi lineales con Calories, mientras que la relación con *Body_temp* tiende a un comportamiento más cercano a lo exponencial.



Tratamiento de Datos

Escalar los datos es una práctica común en *Machine Learning*. En este modelo no fue la excepción, los datos fueron normalizados utilizando la siguiente fórmula:

$$z = \frac{x - \bar{x}}{\sigma}$$

donde x representa el valor original, $\bar{x}$ la media de la variable y σ su desviación estándar.

El objetivo de esta transformación es que los datos se encuentren expresados en la distancia a su media, medida en desviaciones estándar. De esta forma, se evita que variables con escalas grandes dominen el modelo. Además, este procedimiento ayuda con el algoritmo de descenso de gradiente, ya que los valores dejan de tener magnitudes muy grandes y se encuentran en un rango de desviaciones estándar, facilitando las operaciones durante el entrenamiento.

Función de hipótesis

La función de hipótesis es la que utilizamos para predecir y entrenar nuestro modelo. Se define de la siguiente manera:

$$h_{\Theta}(x) = \Theta_0 x_0 + \Theta_1 x_1 + \Theta_2 x_2 + \ \ldots \ + \Theta_n x_n \ + \ b$$

El resultado de la función de hipótesis estará expresado en las mismas unidades de la variable que deseamos predecir. En nuestro modelo, la función de hipótesis cuenta con 6 parámetros (5 θ y 1 b), lo que significa que consiste en la suma de cada parámetro multiplicado por su respectivo valor de entrada del dataset.

Costo

La forma en que calculamos el error es la diferencia entre el valor real y la predicción de nuestro modelo al cuadrado dividido entre el número de instancias, esto es conocido como Error Cuadrático Medio (MSE, *Mean Squared Error*), definido por la siguiente fórmula:

$$J(\theta, b) = \frac{1}{2m}\sum(h\theta(x_i) - y_i)^2$$

De esta manera, podemos cuantificar el error en las predicciones en términos de "calorías al cuadrado", es decir, qué tanto se desvía nuestro modelo del valor correcto.

Si se preguntan porque la división es 2m y no solo m, m siendo el número de instancias. La respuesta es que cuando saquen la derivada, el exponencial se cancelará con el que está dividiendo. Como luego se multiplica por el learning rate pues no importa dividir entre dos.

Descenso de Gradiente

La fórmula utilizada para actualizar los parámetros es:

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

Durante cada época, los parámetros se actualizan hasta que se cumple alguna condición de parada, como alcanzar un número máximo de iteraciones o en mi caso que la diferencia entre el costo anterior y el costo actual sea menor a 0.0001, menor a este número se puede decir que ya no está aprendido mucho.

La forma en la que funciona el descenso de gradiente es simple. Primero, se calcula el error de la predicción, el cual es la diferencia entre la función de hipótesis y el valor real. Este error se multiplica por la entrada de cada instancia y posteriormente se suman los resultados para todas

las instancias, lo que constituye la pendiente del error. En un modelo de regresión lineal, el error siempre será una función cuadrática, por lo que el gradiente nos indica en qué dirección debemos movernos para reducirlo. Sin embargo, para evitar avanzar demasiado en esa dirección y caer en el mismo error pero en el lado opuesto, se introduce un valor de corrección llamado learning rate, representada como α. Este valor disminuye la magnitud del ajuste. Finalmente, se resta el producto de la pendiente y el learning rate al valor actual del parámetro para actualizarlo. Este procedimiento se aplica a cada uno de los parámetros del modelo. En nuestro caso fueron 6, todos inicializados en 1. Después de varias pruebas, se determinó que un valor de 0.1 para el learning rate fue el más adecuado, ya que permitió un entrenamiento más rápido.

Resultados

El entrenamiento del mi modelo se detuvo en la época 497, cuando se alcanzó la condición de la diferencia de costo. Los resultados obtenidos en los conjuntos de entrenamiento, validación y prueba fueron consistentes, lo que indica que el modelo generaliza de manera adecuada y no "memoriza" datos.

Conjunto	RMSE (cal)	MAE (cal)	R^2
Entrenamiento	**9.59**	**6.95**	0.968
Validación	**9.83**	**7.14**	0.966
Prueba	**10.51**	**7.64**	0.963

Interpretación

RMSE (Root Mean Squared Error) Es la desviación promedio de las predicciones respecto al valor real. El modelo mostró un error de aproximadamente 9.6 calorías en entrenamiento, 9.8 en validación y 10.5 en prueba. Esto significa que, si una persona realmente quema 300 calorías, el modelo en promedio podría predecir entre 290 y 310 calorías.

MAE (Mean Absolute Error) Representa el error absoluto medio, es decir, cuántas calorías se equivoca el modelo en promedio sin importar la dirección del error. En nuestro caso, el MAE fue

de alrededor de 7 calorías. Esto quiere decir que, si una persona quema 300 calorías, el modelo va a dar valores como 293 o 307 calorías, lo que refleja un error más "típico" que el RMSE. Pero no significa que sea mejor. Depende de lo que quieras, ya que el RMSE es más sensible con los errores grandes así que lo hace mejor si te importa que no se te pasen los errores grandes.

En mi interpretación, el MAE es lo que le dices al cliente y el RMSE es el que te va a preguntar el experto.

R^2 (Coeficiente de Determinación) Indica qué proporción de la variabilidad en las calorías quemadas es explicada por mi modelo. Mi modelo explica aproximadamente el 96% de la variación total en los datos. Si tuviéramos 100 personas con distintos tiempos de ejercicio, edades, temperaturas y ritmos cardíacos, el modelo sería capaz de explicar las calorías en 96 de ellas, mientras que el de 4 personas sin explicar.

En cuanto a los parámetros obtenidos, el modelo asignó un peso de 48.39 a la variable *Duration*, lo que confirma que el tiempo de ejercicio es el factor más determinante en el gasto calórico. La variable *Heart_rate* también tuvo una influencia considerable, con un coeficiente de 15.65, seguida por la edad con 7.58. Por otro lado, la temperatura corporal presentó un coeficiente negativo -11.57, lo que sugiere una relación inversa con la quema de calorías. El bias del modelo fue de 74.67, que puede decirse que es el gasto calórico base a partir del cual se realizan las predicciones.

Regresando al coeficiente de temperatura, no hace sentido que mientras menos temperatura se quemen más calorías. Si vimos claramente que tenía un coeficiente de Pearson de 0.82. El problema aquí es que también tiene mucha correlación con las variables *Duration* y *Heart_rate*, así que el modelo ya entiende el patrón gracias a *Duration* y *Heart_rate* y la temperatura la pone en negativo para balancear.

Creí que se debía a la mala suerte de la temperatura, y si al iniciar el modelo ponía de inicio en los parámetro, 48 para la temperatura y -11 para la *Duration*, entonces el modelo terminará ajustándose para tener la temperatura en positivo y *Duration* en negativo. Pero al final no, la mejor forma que encontró el modelo de minimizar el error fue poniendo en negativo la temperatura.

En futuros experimentos habría que ver qué pasa con mi modelo si decido quitar la temperatura como variable.