Aplicación de Algoritmos de Aprendizaje Automático para el Análisis de Rendimiento Deportivo

Sergio Cortés Cepeda
matrícula:1731225
Universidad Autónoma de Nuevo León,
Facultad de Ciencias Físico Matemáticas

Noviembre 2025

Resumen

El presente artículo analiza un conjunto de datos deportivos obtenidos mediante un reloj Garmin, con el objetivo de predecir y comprender el comportamiento de variables fisiológicas y de rendimiento. Para ello se aplicaron algoritmos de aprendizaje supervisado **(Regresión Lineal y Random Forest)**orientados a la predicción de calorías y ritmo promedio, así como algoritmos no supervisados **(DBSCAN y Mean Shift)** destinados a identificar patrones y posibles grupos dentro de las sesiones de entrenamiento.

Este trabajo demuestra que la combinación de métodos supervisados y no supervisados proporciona una visión integral del rendimiento deportivo, facilitando la interpretación del comportamiento fisiológico y biomecánico durante sesiones de carrera.

1. Introducción

El análisis de datos deportivos ha cobrado relevancia en los últimos años debido al creciente uso de dispositivos electrónicos capaces de registrar información detallada del rendimiento humano.

El presente estudio utiliza registros reales obtenidos mediante un dispositivo Garmin, incluyendo variables como distancia recorrida, frecuencia cardiaca, pasos, cadencia y zancada. El objetivo principal es analizar la relación entre estas variables e identificar patrones mediante algoritmos de aprendizaje automático.

Para lograrlo, se aplicaron dos enfoques complementarios:

1. **Modelos supervisados** para predecir variables clave como calorías quemadas y ritmo promedio.
2. **Modelos no supervisados** para explorar la estructura interna del conjunto de datos y detectar posibles agrupamientos de sesiones de entrenamiento.

Además, se emplearon métodos de selección de características para identificar las variables con mayor influencia en cada objetivo. Este proceso permitió filtrar información relevante y optimizar el desempeño de los modelos.

El análisis realizado proporciona información valiosa sobre el comportamiento fisiológico durante las sesiones de carrera, permitiendo no solo estimar variables objetivo con precisión, sino también identificar patrones que pueden ser utilizados para mejorar la planificación del entrenamiento y comprender mejor el esfuerzo realizado.

2. Metodología

2.1. Conjunto de datos y variables

El conjunto de datos utilizado corresponde a sesiones de entrenamiento registradas con un reloj Garmin. Cada fila representa una actividad de carrera e incluye información de distancia, respuesta fisiológica y características biomecánicas.

A partir del análisis realizado en tareas previas (correlación, análisis de componentes principales y selección de variables), se consideraron como variables explicativas las siguientes:

- `distancia_km`: distancia recorrida en kilómetros.
- `pasos`: número total de pasos durante la sesión.
- `fc_media`: frecuencia cardiaca media.
- `fc_max`: frecuencia cardiaca máxima.

- `cadencia_media`: cadencia media de carrera.
- `zancada_m`: longitud media de zancada.
- `cadencia_max`: cadencia máxima (utilizada en algunas pruebas exploratorias).

Se trabajó con dos variables objetivo:

- `calorias`: calorías totales quemadas en la sesión.
- `ritmo_medio`: ritmo promedio de carrera.

Previo al modelado se realizó estandarización de las variables numéricas mediante

$$z = \frac{x - \mu}{\sigma},$$

a fin de evitar que las diferencias de escala afectaran el desempeño de los algoritmos, en especial en los modelos no supervisados.

2.2. Selección de características

Para identificar las variables más relevantes en la predicción de cada objetivo se aplicaron tres técnicas de selección de características:

1. **SelectKBest con prueba F** (*f_regression*), que evalúa la relación lineal entre cada predictor y la variable objetivo.
2. **SelectKBest con Información Mutua** (*mutual_info_regression*), que captura relaciones potencialmente no lineales.
3. **RFE (Recursive Feature Elimination)** con regresión lineal como estimador base, que elimina recursivamente las variables menos relevantes.

Los resultados fueron consistentes y permitieron definir, para cada objetivo, el subconjunto de variables más importante:

- **Para calorías**: `distancia_km`, `pasos`, `fc_media` y `cadencia_media`.
- **Para ritmo_medio**: `fc_max`, `cadencia_media`, `fc_media` y `zancada_m`.

2.3. Selección de variables

Antes de ajustar los modelos supervisados y no supervisados, se realizó un proceso de selección de características con el objetivo de identificar los predictores más relevantes para cada variable objetivo. Este paso constituye un preprocesamiento fundamental, ya que permite reducir la dimensionalidad del conjunto de datos, eliminar variables poco informativas y mejorar el desempeño de los modelos posteriores.

Para ello se aplicaron tres métodos complementarios: **F-score**, **Información Mutua (MI-score)** y **Eliminación Recursiva de Características (RFE)**. Cada uno de estos enfoques evalúa la relevancia de los predictores desde una perspectiva distinta, lo que permite obtener una selección más robusta y consistente.

2.4. F-score

El F-score mide la intensidad de la relación lineal entre cada variable predictora y la variable objetivo. Formalmente, evalúa cuánto varía la media del objetivo al cambiar la variable de interés. Valores altos de F-score indican que el predictor tiene una influencia significativa dentro de un modelo lineal simple. Esta métrica es especialmente útil para identificar relaciones estrictamente lineales.

2.5. Información Mutua (MI-score)

La Información Mutua cuantifica la dependencia entre dos variables, considerando tanto relaciones lineales como no lineales. A diferencia del F-score, el MI-score detecta patrones más complejos, como relaciones curvas o saturaciones. Un valor alto de MI-score implica que el predictor comparte una gran cantidad de información con la variable objetivo y, por tanto, es útil para modelos que capturan interacciones no lineales.

2.6. Eliminación Recursiva de Características (RFE)

El método RFE selecciona variables mediante un proceso iterativo que entrena un modelo repetidamente y elimina los predictores menos relevantes en cada iteración. El resultado final es un conjunto óptimo de características clasificadas por orden de importancia. Esta técnica es especialmente útil en modelos supervisados, ya que evalúa la contribución real de cada variable al desempeño predictivo del modelo.

2.7. Modelos supervisados

Para cada variable objetivo se entrenaron dos modelos de regresión:

- **Regresión Lineal**: Busca modelar la relación entre una variable dependiente y (por ejemplo, calorías o ritmo promedio) y un conjunto de variables predictoras $x_1, x_2, \ldots, x_p$. El modelo general se expresa como:

  $$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p + \varepsilon$$

  En forma matricial:

  $$y = X\beta + \varepsilon$$

  donde:

  - $X \in \mathbb{R}^{n \times p}$: matriz de variables predictoras,
  - $\beta \in \mathbb{R}^p$: vector de coeficientes,
  - $y \in \mathbb{R}^n$: vector de valores observados,
  - $\varepsilon \in \mathbb{R}^n$: términos de error.

- **Regresión de Bosques Aleatorios (Forest Regressor)**:Consiste en un ensamble de árboles de decisión, donde cada árbol $T_b(x)$ es entrenado utilizando una muestra bootstrap del conjunto de datos original y un subconjunto aleatorio de características.

  La predicción de un único árbol para una observación x se define como:

  $$T_b(x) = \frac{1}{N_b} \sum_{i \in R_b(x)} y_i,$$

  donde $R_b(x)$ es la región terminal del árbol donde cae la observación x, y N_b es el número de muestras dentro de dicha región.

  La predicción final del Bosque Aleatorio se obtiene promediando las salidas de los B árboles:

  $$\hat{y} = \frac{1}{B} \sum_{b=1}^{B} T_b(x).$$

  A continuación se describen detalladamente los elementos que componen esta expresión:

- **Predicción del bosque** ($\hat{\boldsymbol{y}}$): es el valor predicho por todo el conjunto de árboles para una observación x. En este proyecto corresponde, por ejemplo, a las calorías estimadas o al ritmo promedio predicho a partir de las variables registradas durante una sesión de carrera.
- **Vector de características** ($\boldsymbol{x}$): representa el conjunto de variables predictoras utilizadas para una observación. En este estudio, este vector puede escribirse como:

$$x = (\text{distancia\_km},\ \text{fc\_media},\ \text{fc\_max},\ \text{cadencia\_media},\ \text{zancada\_m},\ \text{pasos},\ \dots)$$

- **Número de árboles** ($\boldsymbol{B}$): indica cuántos árboles de decisión conforman el bosque. Este valor corresponde al parámetro `n_estimators` del modelo Random Forest. Por ejemplo, en mi caso use `n_estimators = 200`, entonces $B = 200$. Un valor mayor de B suele producir predicciones más estables.
- **Índice del árbol** ($\boldsymbol{b}$): es un contador que recorre cada árbol dentro del bosque:

$$b = 1, 2, 3, \dots, B$$

  Matemáticamente permite expresar la idea de que cada árbol realiza su propia predicción sobre la observación x.
- **Predicción del árbol individual** ($\boldsymbol{T_b(x)}$): T_b representa el árbol número b. La expresión $T_b(x)$ es la predicción que realiza dicho árbol para la observación x. En un problema de regresión, esta predicción es un número real, por ejemplo:
  - 315,2 calorías predichas,
  - 485 segundos de ritmo promedio.

En conjunto, la fórmula indica que la predicción del Random Forest es el **promedio de las predicciones individuales de todos los árboles**, lo cual proporciona robustez, reduce la varianza del modelo y mejora su capacidad de generalización.

El conjunto de datos se dividió en entrenamiento y prueba utilizando una partición del 80 % para entrenamiento y 20 % para prueba.

La calidad de los modelos se evaluó mediante las métricas:

- **MAE (Mean Absolute Error)**: mide el error absoluto promedio entre los valores observados y los valores predichos. Indica cuánto se equivoca el modelo en promedio. Valores menores implican un mejor desempeño.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|\,,$$

- **MSE (Mean Squared Error)**: promedio de los errores al cuadrado. Penaliza fuertemente los errores grandes, por lo que detecta modelos con predicciones muy alejadas del valor real.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2,$$

- **RMSE (Root Mean Squared Error)**: raíz cuadrada del MSE. Expresa el error en las mismas unidades que la variable objetivo y representa la desviación promedio de las predicciones respecto a los valores reales.

$$\text{RMSE} = \sqrt{\text{MSE}},$$

- R^2 **(Coeficiente de determinación)**: mide la proporción de la variabilidad de los datos explicada por el modelo. Valores cercanos a 1 indican un buen ajuste, mientras que valores cercanos a 0 o negativos indican bajo desempeño.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}.$$

2.8. Reducción de Dimensionalidad mediante Componentes Principales (PCA)

El Análisis de Componentes Principales (PCA) se aplicó con el propósito de proyectar las variables fisiológicas y biomecánicas en un subespacio de menor dimensión, preservando la mayor varianza posible del sistema. Matemáticamente, el procedimiento se desarrolló de la siguiente manera.

Sea $X \in \mathbb{R}^{n\times p}$ la matriz de datos que contiene n sesiones de entrenamiento y p variables fisiológicas estandarizadas. Primero, se centraron los datos en su media:

$$\tilde{X} = X - \mathbf{1}\mu^T,$$

donde μ es el vector de medias muestrales de cada variable y $\mathbf{1}$ es un vector columna de unos.

Posteriormente, se calculó la matriz de covarianza muestral:

$$S = \frac{1}{n-1}\tilde{X}^T\tilde{X}.$$

El PCA consiste en encontrar los vectores propios (autovectores) v_k y valores propios (autovalores) λ_k de la matriz S, tales que:

$$Sv_k = \lambda_k v_k, \quad k = 1, 2, \ldots, p.$$

Los valores propios se ordenan de mayor a menor:

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p.$$

Cada autovalor λ_k representa la varianza explicada por el k-ésimo componente principal. La proporción de varianza explicada (PVE) se define como:

$$\text{PVE}_k = \frac{\lambda_k}{\sum_{j=1}^{p} \lambda_j}.$$

Con base en este criterio, se seleccionaron los primeros tres componentes principales, ya que en conjunto explicaron el 85.7 % de la varianza total del sistema.

El k-ésimo componente principal se obtiene mediante la proyección:

$$Z_k = \tilde{X}v_k,$$

donde Z_k es la nueva variable latente que representa el componente principal. De esta forma, la matriz reducida de componentes principales está dada por:

$$Z = \tilde{X}V_r,$$

donde $V_r = [v_1\ v_2\ v_3]$ es la matriz de los tres autovectores seleccionados.

Estos componentes fueron interpretados de la siguiente forma:

- PC1: componente asociado al volumen de entrenamiento (distancia, calorías y pasos).
- PC2: componente relacionado con la intensidad fisiológica (ritmo, frecuencia cardiaca).
- PC3: componente vinculado con la mecánica de carrera (zancada y cadencia).

Finalmente, los componentes principales Z fueron utilizados como entrada para los algoritmos de agrupamiento DBSCAN y Mean Shift, lo cual permitió operar sobre un espacio de menor colinealidad y mayor interpretabilidad.

2.9. Modelos no supervisados

Con el objetivo de identificar patrones y agrupar sesiones de entrenamiento con características similares, se aplicaron dos algoritmos de *clustering* sobre las variables estandarizadas `distancia_km`, `pasos`, `fc_media`, `fc_max`, `cadencia_media` y `zancada_m`:

- **DBSCAN** (*Density-Based Spatial Clustering of Applications with Noise*), con parámetros `eps = 0.7` y `min_samples = 3`.

  El algoritmo *DBSCAN* identifica grupos en los datos a partir de la densidad local de puntos en el espacio de características.

  El criterio fundamental consiste en evaluar cuántos puntos se encuentran dentro de un vecindario de radio ε alrededor de cada observación. Dos parámetros controlan el proceso: el radio máximo ε y el número mínimo de puntos requeridos MinPts para que un punto sea considerado suficientemente denso.

  Formalmente, un punto p es clasificado como **núcleo** si se cumple:

$$|\{q \in D \mid \text{dist}(p, q) \leq \varepsilon\}| \geq \text{MinPts}$$

- **Mean Shift**, con ancho de banda estimado automáticamente mediante `estimate_bandwidth`.

  El método utiliza un kernel de ventana, cuyo ancho de banda h determina el tamaño del vecindario para calcular el desplazamiento del punto hacia regiones de mayor densidad. En este trabajo, el ancho de banda se estimó automáticamente mediante la función `estimate_bandwidth`.

  El vector de desplazamiento (*mean shift*) para un punto x se define como:

$$m(x) = \frac{\sum\limits_{x_i \in N(x)} K\left(\frac{\|x - x_i\|^2}{h^2}\right) x_i}{\sum\limits_{x_i \in N(x)} K\left(\frac{\|x - x_i\|^2}{h^2}\right)} - x$$

  donde:

  - x es el punto actual.
  - $N(x)$ es el vecindario dentro del radio determinado por h.
  - $K(\cdot)$ es una función kernel (comúnmente el kernel gaussiano).
  - $m(x)$ es el vector que señala la dirección del incremento de densidad.

  La actualización iterativa del algoritmo es:

$$x_{t+1} = x_t + m(x_t),$$

DBSCAN permite identificar regiones densas y clasificar puntos aislados como ruido, mientras que Mean Shift localiza modos de densidad sin requerir especificar el número de clusters a priori.