República del Ecuador
Universidad Tecnológica Empresarial de Guayaquil - UTEG

Trabajo de Titulación
para la obtención del título de:
Ingeniero en Software

Tema:
Sistema de análisis del rendimiento basado en inteligencia artificial para corredores de media maratón en Sudamérica.

Autor/a:
Mario Andrés Bruzzone Maugé

Director de trabajo de titulación:
Ing. Grace Katiuska Viteri Guzman. MSc.

2024

Guayaquil - Ecuador

AGRADECIMIENTO

Quisiera agradecer desde lo profundo de mi corazón y alma a mi querida mamá, Ana María Maugé. Las letras, palabras o la voz en sí no bastan para expresar mi amor eterno. Si no fuera por la luz que emanas, iluminando mi camino, no estaría donde estoy, tanto en conocimiento como en lo que me he forjado a ser. Es gracias a ese corazón bondadoso que, ante cada problema o circunstancia, estuviste para darme la mano sin titubear. Te amo, mami. Gracias por todo. A mi papá, Christian Stagg, en los días en los que la vida me sacudía con incertidumbres, estuviste a mi lado con tus enseñanzas y un hombro en el que apoyarme. Compartes momentos genuinos conmigo y siempre logras sacarme la mejor de mis sonrisas. Gracias, papá. A mi tía Graciela Erazo, por su amor sin límites y un apoyo incondicional que ha estado presente toda mi vida. Gracias por darle la mano no solo a mí, sino a toda mi familia. Siempre estarás en mi corazón y alma. A mi abuelita Edy Erazo, que es mi espejo y una inspiración para seguir. Almas tan cálidas e inquebrantables no se encuentran fácilmente, y yo puedo asegurar que eres un diamante en ese aspecto. Gracias, abue, por cada abrazo y sonrisa que me das con tanto amor. A mis dos hermanos, Ian Stagg y Sebastián Bruzzone. Compartir con ustedes año a año experiencias inolvidables es algo que no cambiaría por nada. Ustedes significan mucho para mí, y por siempre serán mis hermanos. Los quiero mucho. Y, por último, pero no menos importante a mis compañeros con los que curse esta fase en mi vida y nos dimos la mano en todo momento para llegar a donde estamos, muchas gracias a todos ustedes.

DEDICATORIA

Este trabajo lo dedico a mi querida familia, mis amigos, personas con abundante sabiduría que conocí en el camino y me ayudaron a donde estoy, les dedico este trabajo en el cual puse un gran empeño y por último a cada una de las personas que de una u otra forma me han apoyado, dirigido y orientado en toda mi carrera profesional.

DECLARACIÓN DE AUTORÍA

Yo Mario Andrés Bruzzone Maugé con C.I. 0923162978, certifico que esta tesis de “Sistema de análisis del rendimiento basado en inteligencia artificial para corredores de media maratón en Sudamérica” es un trabajo original y personal. Todas las ideas plasmadas y conclusiones presentadas en este documento son producto de mi investigación y análisis.

He seguido los principios éticos y he citado adecuadamente todas las fuentes utilizadas. Las conclusiones a las que he llegado son de mi responsabilidad.

Mario Andrés Bruzzone Maugé
C.I. 0923162978

SISTEMA DE ANÁLISIS DEL RENDIMIENTO CON INTELIGENCIA ARTIFICIAL PARA CORREDORES DE MEDIA MARATÓN EN SUDAMÉRICA

Mario Andrés Bruzzone Maugé

bruzzone123@gmail.com

RESUMEN

El objetivo de este trabajo es analizar los datos de rendimiento para corredores de media maratón en Sudamérica en el lapso del 2018 al 2024, basado en inteligencia artificial. La metodología empleada incluye el uso de algoritmos de machine learning, como Random Forest y regresión lineal, para imputar datos faltantes y predecir el rendimiento futuro de los corredores. Además, se aplicó el análisis de componentes principales (PCA) para identificar factores predictivos y técnicas de clustering para segmentar a los corredores en grupos según su rendimiento. Los resultados muestran que el análisis es capaz de clasificar a los corredores en tres clusters: bajo rendimiento, potencial y élite, lo que permite establecer semáforos de alerta para monitorear su desempeño. Asimismo, se generaron predicciones de rendimiento hasta el año 2025, proporcionando información valiosa para los entrenadores y atletas. Este trabajo, resalta la necesidad de un enfoque más sistemático para agrupar y analizar la información sobre este deporte, facilitando la información ya sea para los atletas, organizadores o aficionados.

Palabras clave: Inteligencia artificial, rendimiento deportivo, machine learning, análisis predictivo, corredores de media maratón.

INTRODUCCIÓN

En la actualidad la información que existe sobre los corredores de media maratón en Sudamérica se encuentra dispersa, normalmente esta información está ubicada en los sitios web oficiales, federaciones de atletismo, plataformas deportivas entre otros y depende de muchos factores como la calidad de data por cada país, registros levantados por los organizadores de las carreras, cobertura realizada por los medios de prensa local e internacional y uso de aplicaciones para el seguimiento durante las carreras que no siempre son de dominio público. Por tal motivo, se presenta la interrogante de ¿Cómo se puede obtener información tabulada y específica sobre el rendimiento de corredores de media maratón en Sudamérica para fortalecer sus puntos débiles en competición?, esta necesidad se convierte en una oportunidad para realizar este estudio de investigación, analizando los datos de rendimiento para corredores de media maratón en Sudamérica en el lapso del 2018 al 2024, basado en inteligencia artificial. Cabe mencionar, que con dicho análisis se podrá validar la eficiencia y la precisión de los datos tabulados para predecir el rendimiento en las competiciones futuras, siendo éste capaz de procesar y analizar grandes volúmenes de datos y así manipular la data que en la actualidad se encuentra dispersa. Es relevante indicar, que a través de este análisis se podrá clasificar y realizar comparativos sobre el rendimiento entre diferentes grupos y países de corredores. Esta metodología permitirá tanto identificar patrones generales como también entender las diferencias específicas entre los corredores de distintos países que sean objeto del análisis, lo cual ayudará a guiar a entrenadores para mejorar las debilidades y desarrollar las fortalezas de los corredores para que éstos puedan competir efectivamente en el futuro.

OBJETIVOS

Objetivo general

Analizar los datos de rendimiento para corredores de media maratón en Sudamérica en el lapso del 2018 al 2024, basado en inteligencia artificial.

Objetivos específicos

- Identificar factores predictivos del rendimiento basado en los datos recopilados de los corredores de media maratón en América del Sur.
- Utilizar inteligencia artificial para el análisis de los datos generados por corredores de media maratón.
- Establecer semáforos de alerta de grupos de corredores.

MARCO TEÓRICO

Antecedentes

Uno de los estudios relevantes en el campo de la inteligencia artificial aplicada al deporte y la actividad física es el realizado por Martin Celestino Ortiz, titulado "El uso de la inteligencia artificial en el área del deporte y en la actividad física"(Ortiz, 2023). Este estudio evalúa cómo la inteligencia artificial puede identificar y corregir errores en los movimientos durante la actividad física. En particular, la inteligencia artificial aplicada al análisis del rendimiento deportivo de corredores de media maratón puede ayudar a identificar atletas prometedores y mejorar su desempeño. Ortiz destaca la importancia de factores como la edad, el tiempo, el país y la altura en el rendimiento de los corredores.

Otro estudio significativo es "Pacing Patterns of Half-Marathon Runners: An analysis of ten years of results Gothenburg Half Marathon"(Johansson, 2023). Este trabajo examina los patrones de ritmo de los corredores basándose en datos de diez años del Maratón de Gotemburgo, una de las mayores carreras de media maratón del mundo con más de 40,000 participantes anuales. El objetivo es reducir el riesgo de sobreesfuerzo, lesiones o colapsos entre los corredores, proporcionando asesoramiento sobre el ritmo adecuado mediante el uso de aprendizaje automático.

En concreto este trabajo de estudio se ha centrado en la utilización de una herramienta valiosa para mejorar el rendimiento y la seguridad de los corredores siendo ésta el algoritmo Random Forest para analizar estos datos y encontrar patrones de ritmo relacionados con la edad, sexo, capacidad y la temperatura del día de la carrera.

Análisis del rendimiento

El análisis del rendimiento es un estudio para establecer “determinantes para medir la calidad de los desarrollos de software, ya que permiten identificar aspectos que se deben mejorar en pro de alcanzar la satisfacción del cliente” (Gil-Vera & Seguro-Gallego, 2022).

Con respecto a las funciones que podemos mencionar, se puede destacar el poder visualizar una gran cantidad de datos para poder recabar más información que a simple vista no es posible percibir. Igualmente, el identificar y corregir problemas de rendimiento, pueden llevar a una mejora en la experiencia que recibe el usuario final, y a su vez la misma persona que trata de interpretar los datos. Dicho de otra manera, esto aumenta la satisfacción del cliente y la retención de usuarios.

Inteligencia artificial

“La inteligencia artificial aún no se ha definido formalmente, pero se ha establecido una forma aceptada de comprenderla como una rama de la informática enfocada en crear sistemas capaces de simular procesos cognitivos humanos, como el razonamiento, el aprendizaje y la resolución de problemas, la cual busca desarrollar sistemas que puedan realizar tareas que normalmente son ejecutadas por las personas, como comprender el lenguaje natural, reconocer patrones y tomar decisiones autónomas. En esencia, la inteligencia artificial busca dotar a los sistemas informáticos de la capacidad de pensar y actuar de manera inteligente.” (*Planderecuperacion*, 2023)

Sobre la definición de inteligencia artificial, también podemos encontrar el siguiente concepto oficial que es dado por la Comisión Europea de Inteligencia Artificial, que indica lo siguiente:

"La inteligencia artificial se refiere a sistemas que muestran un comportamiento inteligente al analizar su entorno y tomar acciones con algún grado de autonomía para alcanzar objetivos específicos." (European Commission, 2018).

Algoritmo Random Forest

Leo Breiman, estadista estadounidense es considerado el padre del algoritmo Random Forest, ya que introdujo el concepto y proporcionó una descripción detallada de su funcionamiento, es así como en su artículo lo define como "un algoritmo de aprendizaje automático que pertenece a la familia de los métodos de ensamble. Se trata de un modelo de clasificación y regresión que utiliza múltiples árboles de decisión para mejorar la precisión y reducir el sobreajuste. Cada árbol de decisión en el bosque se construye a partir de una muestra aleatoria de los datos de entrenamiento y un subconjunto aleatorio de características. Al combinar las predicciones de todos los árboles, Random Forest logra una alta precisión y generalización." *(Breiman, 2001).*

Cabe indicar, que se ha tomado este artículo de referencia, debido a que es una pieza fundamental en la comprensión de los algoritmos.

Machine learning

En cuanto a "Machine learning es el campo de estudio que da a las computadoras la capacidad de aprender sin ser programadas explícitamente. En términos prácticos, es un proceso en el que un sistema de software mejora su desempeño en una tarea específica mediante la experiencia." (Géron, 2019).

Para el desarrollo del presente estudio se ha tomado el aprendizaje automático o machine learning ya que es un subcampo de la Inteligencia Artificial que se enfoca en desarrollar algoritmos y modelos estadísticos, objeto de este estudio ya que nos ayuda a identificar patrones y tomar

decisiones sin ser programadas explícitamente para cada tarea.

Standard Scaler

Es un método que se utiliza para estandarizar las características de un conjunto de datos. Funciona transformando las características de tal manera que tengan una media igual a cero y una desviación estándar igual a uno. Esto se logra restando la media de cada característica y luego dividiendo el resultado por la desviación estándar de dicha característica. Dicha herramienta es usada para obtener resultados con mayor precisión e interpretación a la hora de entender los datos.

Análisis de Componentes Principales (PCA)

Es una técnica de reducción de dimensionalidad que nos permite visualizar datos multivariantes en dos dimensiones, resaltando las principales fuentes de variabilidad en el conjunto de datos.

Corredores de media maratón

Con relación a la "maratón y media maratón, éstas son disciplinas deportivas en donde los corredores que la practican corren una distancia de 42 o 21 kilómetros respectivamente a un ritmo constante según sea el caso con el objetivo de llegar a la meta en el menor tiempo posible" (Rojas, J., Heredia, K.,Azpiroz, A, 2023).

Para el presente estudio se ha tomado en consideración basar la elección de los corredores en la recolección de atletas de media maratón de toda Sudamérica que tengan tiempos por encima del promedio. Cabe indicar, que dichos corredores son categorizados como corredores profesionales.

METODOLOGÍA

La metodología para utilizar en este estudio de investigación de tesis es no experimental longitudinal, dado que se recolectaron datos en base al rendimiento de los corredores de media maratón a lo largo de los años sin manipulación de variables independientes. Se utilizará un enfoque cuantitativo para analizar los datos numéricos de rendimiento y se aplicarán técnicas estadísticas y a su vez se usará el machine learning para la imputación de datos faltantes y predicciones. El alcance de la investigación es correlacional, ya que se busca identificar y analizar las relaciones entre diferentes variables relacionadas con el rendimiento de los corredores a lo largo del tiempo. El método lógico utilizado en este estudio será el de inducción, ya que se tomará el conocimiento de las distintas etapas de los corredores de media maratón considerando su sucesión cronológica, evolución y desarrollo en el tiempo.

Considerando lo expuesto, la técnica de recolección de datos usada es la de revisión documental, para lo cual, se va a tomar la información de una población de los corredores siendo la muestra los de media maratón de Sudamérica en un rango de tiempo de 2018 – 2024 que se encuentra disponible en la página web oficial World Athletics (World Athletics, 2024). Con la finalidad de imputar los datos a la estadística disponible y lograr completar la data y establecer los factores predictivos del rendimiento, se diseñarán códigos que analizarán y procesarán la data recopilada, descubriendo los factores determinantes que influyen en el rendimiento de los corredores. Cabe mencionar, que para este análisis se utilizarán herramientas como Random Forest y Machine learning.

RESULTADOS

Identificación de Factores Predictivos

Para este artículo se recogieron datos de corredores de media maratón de Sudamérica que han participado en competiciones entre los años 2018 y 2024, utilizando como fuente la página World Athletics (worldathletics, 2024). Se recopilaron datos de 294 corredores de los cuales 214 son sudamericanos y los 80 restantes son corredores profesionales que participaron en la maratón de París del año 2024, como dicha competición es de 42 km, estos se dividieron por 2 para aproximar la distancia de 21 km de la media maratón. No obstante, se consideró la ligera diferencia en los tiempos, ya que al usar los valores originales como entrenamiento podría causar sobreajuste (overfitting). Además, se empleó la biblioteca Pandas para cargar los datos desde un archivo CSV y gestionar los valores faltantes mediante métodos de imputación, quedando la información distribuida en el siguiente formato:

País | Corredor | Edad | 2018 | 2019 | 2020 | 2021 | 2022 | 2023 | 2024.

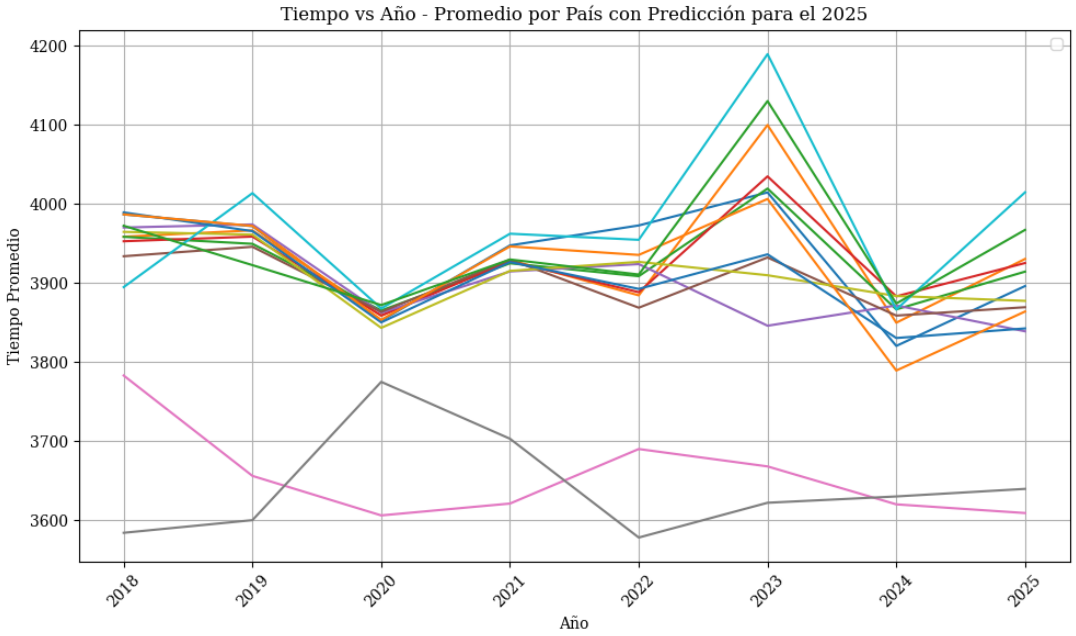
Dado que la mayoría de los corredores no han participado en todas las ediciones anuales de la media maratón, fue necesario imputar los valores faltantes. Para ello, se utilizó el algoritmo de Random Forest (Breiman, 2001), el cual crea diferentes rutas basadas en entrenamiento y testing para determinar cuáles datos son más adecuados para cada corredor de Sudamérica. Es importante indicar, que el conjunto de datos se dividió en TrainData (corredores élite) y TestData (corredores sudamericanos). El algoritmo de Random Forest fue configurado con 50 estimadores, lo que significa que 50 veces el sistema analizó las variables de cada corredor, llenando efectivamente los datos de tiempos faltantes que tenían ciertos corredores en cada año.

Análisis aplicando inteligencia artificial

En el presente estudio, se han analizado datos de corredores de media marathon aplicando inteligencia artificial que utiliza sistemas de aprendizaje automático, por lo que se ha realizado un análisis basado en Regresión Lineal para predecir los tiempos promedio de los corredores para el año 2025. Utilizando los datos históricos del 2018 al 2024, dicha regresión lineal fue entrenada para cada país, permitiendo generar predicciones específicas. Estas predicciones fueron integradas en las visualizaciones para comparar con los datos históricos y evaluar posibles tendencias futuras en el rendimiento de los corredores. Dicho esto, los datos no son del todo exactos, pero se usan para tener una referencia del lugar donde se encuentran los corredores latinoamericanos. (Liverakos, 2018)

Figura 1

Predicción de corredores de media-maratón para el año 2025



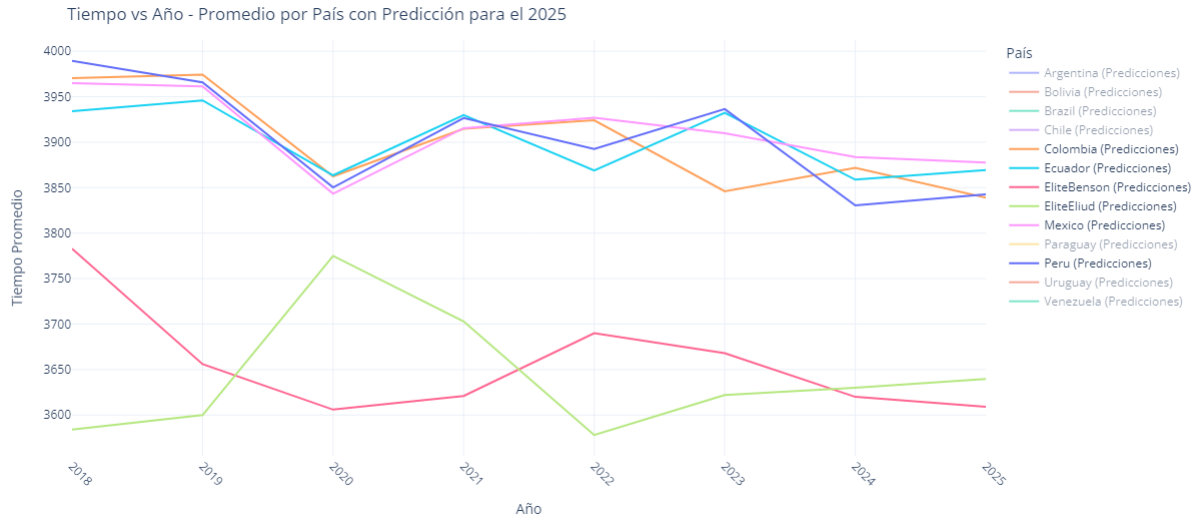
Nota. Los tiempos están expresados en segundos.

Fuente: Elaboración propia a partir de la plataforma (World Athletics, 2024)

A su vez, se realizó una gráfica interactiva en la cual se puede seleccionar que países se desean contrastar con otros y poder compararlos con corredores profesionales, cabe resaltar que el comparar el promedio de un país con un solo corredor profesional da resultados a favor de dicho corredor de elite.

Figura 2

Gráfica interactiva para comparación de corredores



Nota. En este ejemplo se tomaron los países como Colombia, Ecuador, México, Perú para compararlos con corredores élite.

Fuente: Elaboración propia a partir de la plataforma (World Athletics, 2024)

Para asegurar que todas las variables se encuentren en la misma escala, se procedió a estandarizar los datos utilizando la técnica de Standard Scaler de la biblioteca sklearn. Esta técnica transforma cada columna de los datos para que tenga una media de 0 y una desviación estándar de 1, lo que facilita el análisis subsecuente. Sin embargo, es importante notar que los valores en las gráficas resultantes, especialmente después de aplicar el Análisis de Componentes Principales (PCA), pueden exceder el rango de -1 a 1 debido a que los componentes principales son combinaciones lineales de las variables estandarizadas y están diseñados para maximizar la

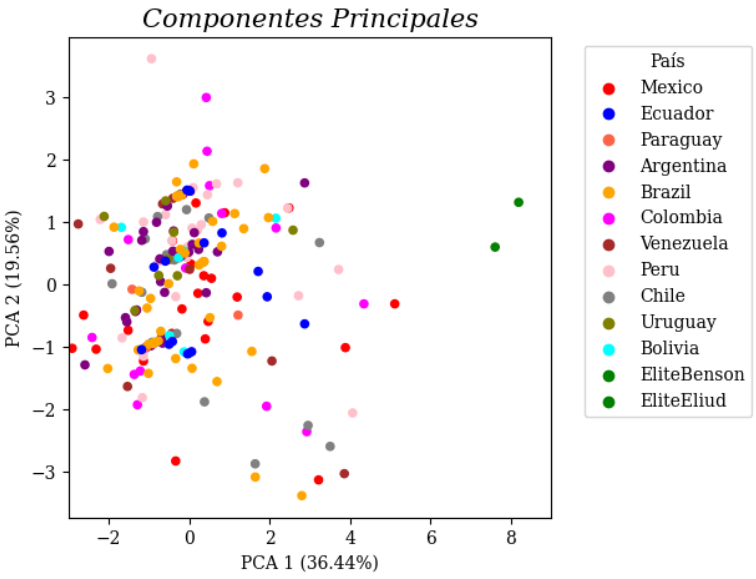
variabilidad capturada en cada eje. Por esta razón, es común que los valores de los componentes principales se extiendan más allá de este rango, como se observa en las figuras que muestran valores entre -3 y 8.

Con el fin de reducir la dimensionalidad de los datos y resaltar los factores más influyentes en el rendimiento de los corredores, se realizó un Análisis de Componentes Principales (PCA). Se extrajeron dos componentes principales que explican una parte significativa de la variabilidad de los datos. La visualización de estos componentes permitió identificar patrones y posibles factores predictivos relacionados con el rendimiento de los corredores.

En el análisis visual del gráfico de dispersión, ver figura 3, se representan a los corredores en el espacio de las dos componentes principales obtenidas. Cada país fue asignado a un color específico lo que permite identificar ciertas tendencias y agrupaciones entre los corredores de diferentes países. Por ejemplo, en el gráfico, observamos a los corredores que tienden a agruparse en un el punto “0”, lo que sugiere similitudes en sus tiempos y rendimientos en las medias maratones de Latinoamérica, así mismo, observamos como la presencia de corredores élite como Benson y Eliud se destacan (puntos verdes), lo que permite contrastar su desempeño con el resto de los corredores.

Figura 3

Desempeño de corredores de media-maratón



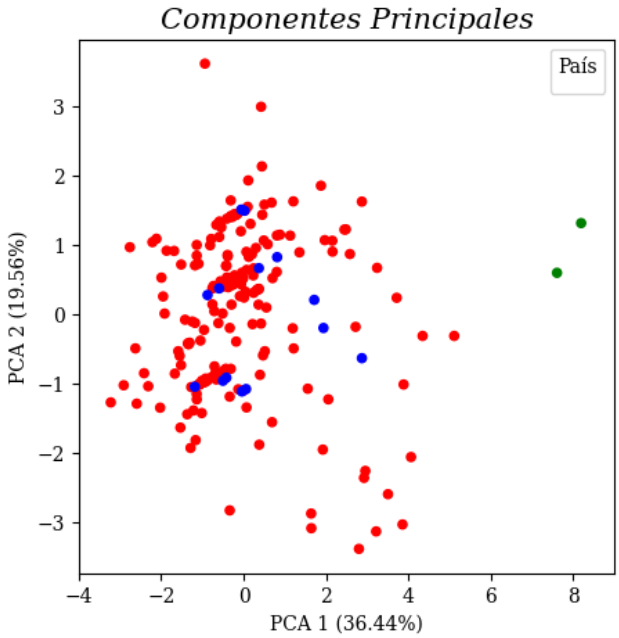
Nota. Se ha tomado como referencia a dos corredores profesionales de élite como son Benson Kipruto y Eliud Kipchoge.

Fuente: Elaboración propia a partir de la plataforma (World Athletics, 2024)

En la figura 4 nos enfocamos en Ecuador, donde podemos identificar que éstos se agrupan con el resto de los corredores de Latinoamérica y también se refleja considerablemente una distancia con los corredores profesionales de élite que compitieron en París 2024, los cuales están pintados de verde, lo que nos brinda información relevante para conocer los comportamientos y características de los distintos corredores a nivel mundial.

Figura 4

Corredores de Ecuador vs corredores de Élite



Nota. Se resaltan de color azul los corredores de Ecuador y de color verde los corredores de élite.
Fuente: Elaboración propia a partir de la plataforma (World Athletics, 2024)

Establecimiento de Semáforos de Alerta

Mediante el análisis de clustering realizado con K-Means, se identificaron 3 clusters de corredores con características similares en términos de rendimiento. Estos grupos sirven como base para establecer semáforos de alerta que permiten monitorear el desempeño de los corredores. Este enfoque permitió segmentar a los corredores en grupos homogéneos basados en sus componentes principales, facilitando la identificación de patrones de rendimiento similares. Los clusters se definieron de la siguiente manera:

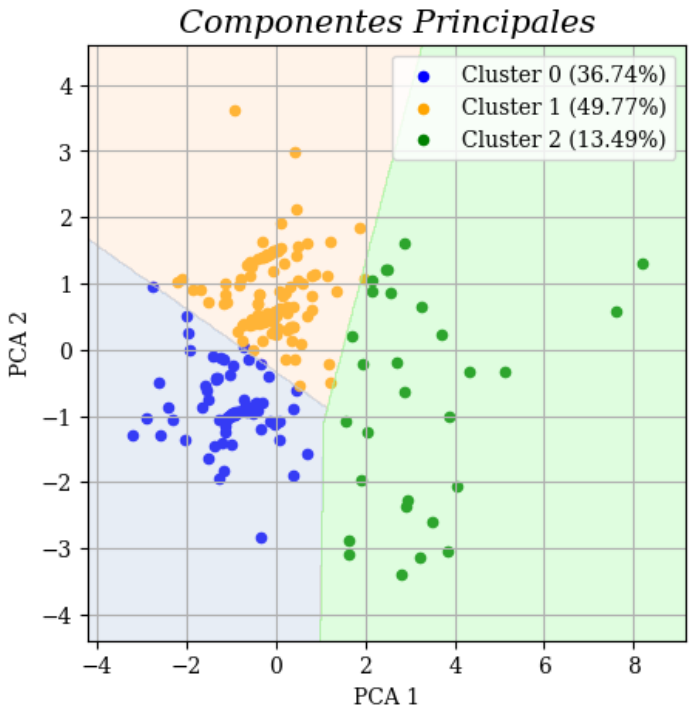
1. Corredores que no están a la altura de competir contra élites mundiales.
2. Corredores con potencial.
3. Corredores con el mejor desempeño.

En la figura 5, cada cluster está representado por un color diferente, lo que permite visualizar claramente la distribución de los corredores según su desempeño. Esta clasificación facilita la identificación de corredores con potencial y aquellos que ya están a nivel de competir contra élites mundiales. El análisis de los clusters permitió identificar tres grupos principales de corredores:

1. Cluster 0 (color azul): Corredores que no están a la altura de competir contra élites mundiales. Este grupo representa aproximadamente el 36.74% de la muestra. Indica una gran posibilidad de que ducho grupo debe de enfocarse en mejorar su rendimiento a través de entrenamientos específicos o cambios en su estrategia.
2. Cluster 2 (color verde): Corredores con el mejor desempeño. Este grupo representa aproximadamente el 13.49% de la muestra. Sugiriendo que estos corredores podrían avanzar al siguiente nivel si se implementan adecuadas intervenciones.
3. Cluster 1 (color naranja): Corredores con potencial. Este grupo representa aproximadamente el 49.77% de la muestra. Dichos corredores son aptos para estar a la altura de los corredores de élite a nivel mundial.

Figura 5

Semáforos de alerta de grupos de corredores



Nota. Se ha dividido en tres clusters para un mejor análisis de rendimiento.

Fuente: Elaboración propia a partir de la plataforma (World Athletics, 2024)

El sistema de semáforos de alerta se puede implementar en un entorno práctico a través de una plataforma digital interactiva, donde entrenadores y atletas puedan monitorear el rendimiento de los corredores en tiempo real. A medida que los corredores registren sus tiempos en futuras competiciones o sesiones de entrenamiento, el sistema actualizará automáticamente los semáforos de alerta. Las intervenciones sugeridas por el sistema pueden incluir ajustes en el entrenamiento, modificación de la dieta o descansos estratégicos para optimizar el rendimiento.

CONCLUSIONES

La presente investigación logró desarrollar un análisis tabulado y específico de rendimiento para corredores de media maratón en Sudamérica, cumpliendo así con el objetivo general planteado, es así, que a través del uso de inteligencia artificial, se identificaron factores predictivos clave para el rendimiento de los corredores, como la edad, el país de origen y los tiempos históricos de competición, estos factores fueron analizados utilizando técnicas de machine learning, como el algoritmo Random Forest y el análisis de componentes principales, lo que permitió tener un análisis de predicción preciso y fiable.

Adicionalmente se realizó un análisis basado en inteligencia artificial que predice el rendimiento de los corredores de media-maratón hasta el año 2025 en Sudamérica. Este análisis, basado en regresión lineal, facilitó la comparación de los tiempos proyectados con los datos históricos, proporcionando valiosas herramientas para los entrenadores y atletas interesados en mejorar su desempeño. Por último, se establecieron semáforos de alerta que clasifican a los corredores en tres grupos: bajo rendimiento, potencial y élite. Este sistema de clasificación permite una monitorización continua y en tiempo real, aportando una solución práctica para mejorar las estrategias de entrenamiento y el desarrollo de los corredores.

Finalmente, en el presente estudio podemos observar, que también se pueden incluir otras variables, dependiendo de los requerimientos deportivos, como por ejemplo el clima o la altitud ya que también son factores que pueden influir en el rendimiento de los corredores.

REFERENCIAS BIBLIOGRÁFICAS

Abrego, R., Guevara, P., & Giraldo, M. (n.d.). 16 ene-dic., 2024 | Revista Científica de la Universidad Especializada de las Américas. https://doi.org/10.57819/dgf7-th24

Alvero-Cruz, J. R., Carnero, E. A., Giráldez García, M. A., Alacid, F., Rosemann, T., Nikolaidis, P. T., & Knechtle, B. (2019). Cooper Test Provides Better Half-Marathon Performance Prediction in Recreational Runners Than Laboratory Tests. Frontiers in Physiology, 10. https://doi.org/10.3389/fphys.2019.01349

Artificial intelligence : a European perspective. (2018). Publications Office. https://doi.org/10.2760/936974

Bastidas García, J. M., Vargas Moreno, L. F., & Osuna Cerecer, E. R. (2023). Análisis de Rendimiento Entre Linux y Windows en la Ejecución de Videojuegos Utilizando Diferentes Niveles de Hardware. Revista Digital de Tecnologías Informáticas y Sistemas, 7(1), 9–14. https://doi.org/10.61530/redtis.vol7.n1.2023.168.9-14

Breiman, L. (2001). Random Forests. Machine Learning, 45, 5–32. https://doi.org/10.1023/A:1010950718922

Celestino Ortiz, M., Espinosa Mendez, C. M., & Flores, A. F. (2023). Artículo de revisión. In Körperkultur Science (Vol. 1, Issue 2).

Cuartero, J. O., Pueo, B., Villalón-Gasch, L., & Jiménez-Olmedo, J. M. (2023). Prediction of Half-Marathon Power Target using the 9/3-Minute Running Critical Power Test. Journal of Sports Science and Medicine. https://doi.org/10.52082/jssm.2023.525

Dimmick, H. L., van Rassel, C. R., MacInnis, M. J., & Ferber, R. (2023). Use of subject-specific models to detect fatigue-related changes in running biomechanics: a random forest approach. Frontiers in Sports and Active Living, 5.

https://doi.org/10.3389/fspor.2023.1283316

Farnham, B., Tokyo, S., Boston, B., Sebastopol, F., & Beijing, T. (n.d.). Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow Concepts, Tools, and Techniques to Build Intelligent Systems Second Edition.

Feely, C. (2022). Using Machine Learning Techniques to Support Marathon Runners. ICCBR Doctoral Consortium. https://doi.org/null

Gil-Vera, V. D., & Seguro-Gallego, C. (2022). Machine learning aplicado al análisis del rendimiento de desarrollos de software. Revista Politécnica, 18(35), 128–139. https://doi.org/10.33571/rpolitec.v18n35a9

González, C. N., Rodríguez Herlein, D. R., Talay, C. A., Trinidad, F. A., Almada, M. L., & Marrone, L. A. (n.d.). Análisis De Rendimiento Y Equidad En Tcp.

Hernández, M. S. (2023). Beliefs and attitudes of canarians towards the chilean linguistic variety. Lenguas Modernas, 62, 183–209. https://doi.org/10.13039/501100011033

Inteligencia artificial: La Comisión presenta un enfoque europeo para impulsar la inversión y establecer directrices éticas. (n.d.). Retrieved August 7, 2024, https://ec.europa.eu/commission/presscorner/detail/es/IP_18_3362

Johansson, M., Atterfors, J., & Lamm, J. (2023). Pacing Patterns of Half-Marathon Runners: An analysis of ten years of results Gothenburg Half Marathon. International Journal of Computer Science in Sport, 22(1), 124–138. https://doi.org/10.2478/ijcss-2023-0014

Katyal, S. K. (2020). Private Accountability in an Age of Artificial Intelligence. UCLA Law Review, 66(1), 47–106. https://doi.org/10.1017/9781108680844.004

Knechtle, B., Barandun, U., Knechtle, P., Zingg, M. A., Rosemann, T., & Rüst, C. A. (2014). Prediction of half-marathon race time in recreational female and male runners.

SpringerPlus, 3(1), 1–8. https://doi.org/10.1186/2193-1801-3-248

Lerebourg, L., Saboul, D., Clémençon, M., & Coquart, J. B. (2022). Prediction of Marathon Performance using Artificial Intelligence. International Journal of Sports Medicine, 44(5), 352–360. https://doi.org/10.1055/a-1993-2371

Liverakos, K., McIntosh, K., Moulin, C. J. A., & O'Connor, A. R. (2018). How accurate are runners' prospective predictions of their race times? In PLoS ONE (Vol. 13, Issue 8). Public Library of Science. https://doi.org/10.1371/journal.pone.0200744

Martínez-Gramage, J., Albiach, J. P., Moltó, I. N., Amer-Cuenca, J. J., Moreno, V. H., & Segura-Ortí, E. (2020). A random forest machine learning framework to reduce running injuries in young triathletes. Sensors (Switzerland), 20(21), 1–12. https://doi.org/10.3390/s20216388

Morillo-Baro, J. P., Reigal, R. E., & Hernández-Mendo, A. (2015). Análisis del ataque posicional de balonmano playa masculino y femenino mediante coordenadas polares. RICYDE: Revista Internacional de Ciencias Del Deporte, 11(41), 226–244. https://doi.org/10.5232/ricyde

Muijlwijk, H., Willemsen, M. C., Smyth, B., & Ijsselsteijn, W. A. (2024). Benefits of Human-AI Interaction for Expert Users Interacting with Prediction Models: a Study on Marathon Running. ACM International Conference Proceeding Series, 245–258. https://doi.org/10.1145/3640543.3645205

Qué es la Inteligencia Artificial | Plan de Recuperación, Transformación y Resiliencia Gobierno de España. (Abril 7, 2024). https://planderecuperacion.gob.es/noticias/que-es-inteligencia-artificial-ia-prtr

¿Qué es un maratón? Historia y curiosidades. (30 de Marzo del 2024). Retrieved August 7, 2024, https://soymaratonista.com/que-es-un-maraton-historia-y-curiosidades-2/

Ristanović, L., Cuk, I., Villiger, E., Stojiljković, S., Nikolaidis, P. T., Weiss, K., & Knechtle, B. (2024). The pacing differences in performance levels of marathon and half-marathon runners. Frontiers in Psychology, 14. https://doi.org/10.3389/fpsyg.2023.1273451

Smyth, B., Lawlor, A., Berndsen, J., & Feely, C. (2024). Recommendations for marathon runners: on the application of recommender systems and machine learning to support recreational marathon runners. User Modeling and User-Adapted Interaction, 32(5), 787–838. https://doi.org/10.1007/s11257-021-09299-3

World Athletics home page | World Athletics. (2024). worldathletics.org. https://worldathletics.org/

ANEXOS

Anexo 1. Enlace al código y capturas de pantalla del desarrollo

https://colab.research.google.com/drive/1Cjdn19NWs9Xob4gnAGetYrQeGQAFF00-?usp=sharing

```
[1]  import numpy as np
     import pandas as pd
     import seaborn as sns
     from sklearn.impute import SimpleImputer
```

```
[2]  import sys
     '''!{sys.executable} -m pip install mglearn'''
```

```
[3]  import sklearn
     '''import mglearn'''
     from sklearn.datasets import load_breast_cancer
     import matplotlib.pyplot as plt
     from sklearn.decomposition import PCA
     %matplotlib inline
     from sklearn.metrics import jaccard_score
```

```
[4]  from sklearn.svm import SVC
     from sklearn.discriminant_analysis import LinearDiscriminantAnalysis as LDA
```

```
[5]  from google.colab import drive
     drive.mount('/content/drive')
```

---------- **Se importa la tabla del csv con los datos de los corredores** ----------

```
[6]  from google.colab import drive
     import pandas as pd
     drive.mount('/content/drive')
     datos = pd.read_csv("/content/drive/My Drive/CorredoresTab.csv")
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

```
[7]  datos.replace(0, np.nan, inplace=True)
```

---------- **Imputación de datos con el uso de Random Forest** ----------

```
[8] from sklearn.experimental import enable_iterative_imputer
    from sklearn.impute import IterativeImputer
    from sklearn.ensemble import RandomForestRegressor
```

```
[9] #Lo que realiza el siguiente código es obtener solo los valores de Road Paris para entrenamiento
    TrainData = datos[datos['Pais'].str.contains('Road Paris')]
    TrainDataR = TrainData.iloc[:, 2:] #Lo que se realiza en la siguiente linea es sacar los string debido a que el random forest no permite strings
```

```
[10] #Lo que hace esto es obtener todos los valores a excepcion de Road Paris para entrenamiento
     TestData = datos.drop(datos[datos['Pais'].str.contains('Road Paris')].index)
     first_two_cols = TestData.iloc[:, :2] # Se guardan los datos para despues usarlos
     TestDataR = TestData.iloc[:, 2:] #Se realiza de nuevo el prodedimiento de sacar strings
```

```
[11] imp_mean = IterativeImputer(estimator=RandomForestRegressor(n_estimators= 50, random_state=0)) # puedo usar tambien random_state=0 para que -
     #-siempre sea el mismo valor aleatorio
     #El nestimators sirve para crear arboles de decision, entre mas numeros mas arboles, lo que hace que tarde mas la respuesta pero da un mejor resutlado
     imp_mean.fit(TrainDataR)
     imp_mean.transform(TrainDataR)
```

```
[12] DatosImp = imp_mean.transform(TestDataR)
```

```
[13] #Transformo en un panda Dataframe para que pueda unir las columnas
     DatosImpPanda = pd.DataFrame(DatosImp)
```

```
[14] ImputedData = pd.concat([first_two_cols, DatosImpPanda], axis=1)
```

```
[15] #Poniendole nombre a las columnas que se les borro su nombre
     ImputedData.columns.values[2:] = ['Edad','2018','2019','2020','2021','2022','2023','2024']
```

```
[16] #Mostrar cantidad de valores nulos
     ImputedData.isnull().sum()
```

```
[17] set_var = set(ImputedData.select_dtypes(exclude='O').columns) - set(['Pais', 'Corredor', 'Edad'])
```

```
[18] #Mostrar cantidad de valores 0
     num_zeros = (ImputedData == 0).sum().sum()
     num_zeros_por_columna = (ImputedData == 0).sum()
     print("Número total de ceros en el DataFrame:", num_zeros)
     print("Número de ceros por columna:", num_zeros_por_columna)
```

---------- **Graficación** ----------

```
[19] import pandas as pd
     import matplotlib.pyplot as plt
     %matplotlib inline
```

```
[20] DatosN = ImputedData.iloc[:, 3:]
```

```
[21] #Se estandariza
     from sklearn.preprocessing import StandardScaler
     sc = StandardScaler(with_mean= True, with_std= True)
     completos_std = sc.fit_transform(DatosN)
     completos_std
```

```
[22] import numpy as np
     np.mean(completos_std[:,0])
```

```
[23] import numpy as np
     np.mean(completos_std[:,0])
```

```
[24] from sklearn.decomposition import PCA
     #crear modela PCA indicando que queremos 2 componentes principales
     pca= PCA(n_components = 2)
     #obtener componentes principales
     completos_pca=pca.fit_transform(completos_std)
     print(completos_pca)
```

```
[25] pca.explained_variance_ratio_
```

```
[26] np.sum(pca.explained_variance_ratio_)
```

```python
import matplotlib.pyplot as plt
import pandas as pd
from sklearn.decomposition import PCA

colors = {
    'Mexico': 'red',
    'Ecuador': 'blue',
    'Paraguay': 'tomato',
    'Argentina': 'purple',
    'Brazil': 'orange',
    'Colombia': 'magenta',
    'Venezuela': 'brown',
    'Peru': 'Pink',
    'Chile': 'Gray',
    'Uruguay': 'Olive',
    'Bolivia': 'Cyan',
    'EliteBenson': 'green',
    'EliteEliud': 'green'
}
```

```python
# Se asignaron colores basados en el país
pais_colors = ImputedData['Pais'].map(colors)

fig = plt.figure(figsize=(5, 5))
plt.rcParams["font.family"] = "serif"
x_label = "PCA 1 (" + str(round(pca.explained_variance_ratio_[0] * 100, 2)) + "%)"
y_label = "PCA 2 (" + str(round(pca.explained_variance_ratio_[1] * 100, 2)) + "%)"
ax = fig.add_subplot(1, 1, 1)
ax.set_xlabel(x_label, fontsize=10)
ax.set_ylabel(y_label, fontsize=10)
ax.set_title("Componentes Principales", fontsize=15, fontstyle="italic")
ax.set_xlim(-3, 9)

# Esto sirve para dibujar cada punto con el color correspondiente al país
scatter = ax.scatter(x=completos_pca[:, 0], y=completos_pca[:, 1], s=20, c=pais_colors)

# Aquí creo la leyenda
handles, labels = scatter.legend_elements()
legend_labels = [colors[k] for k in colors.keys()]
legend_colors = [colors[k] for k in colors.keys()]
for color, label in zip(legend_colors, colors.keys()):
    ax.scatter([], [], c=color, label=label)

# Colocar la leyenda fuera de la gráfica
ax.legend(frameon=True, title="País", bbox_to_anchor=(1.05, 1), loc='upper left')

plt.show()
```

```python
#Aqui se realiza lo mismo que en el codigo del grafico anterior pero con la diferencia de que solo se muestra Ecuador en color diferente
colors = {
    'Mexico': 'red',
    'Ecuador': 'blue',
    'Paraguay': 'red',
    'Argentina': 'red',
    'Brazil': 'red',
    'Colombia': 'red',
    'Venezuela': 'red',
    'Peru': 'red',
    'Chile': 'red',
    'Uruguay': 'red',
    'Bolivia': 'red',
    'EliteBenson': 'green',
    'EliteEliud': 'green'
}
pais_colors = ImputedData['Pais'].map(colors)

fig = plt.figure(figsize=(5, 5))
plt.rcParams["font.family"] = "serif"
x_label = "PCA 1 (" + str(round(pca.explained_variance_ratio_[0] * 100, 2)) + "%)"
y_label = "PCA 2 (" + str(round(pca.explained_variance_ratio_[1] * 100, 2)) + "%)"
ax = fig.add_subplot(1, 1, 1)
ax.set_xlabel(x_label, fontsize=10)
ax.set_ylabel(y_label, fontsize=10)
ax.set_title("Componentes Principales", fontsize=15, fontstyle="italic")
ax.set_xlim(-4, 9)

scatter = ax.scatter(x=completos_pca[:, 0], y=completos_pca[:, 1], s=20, c=pais_colors)
handles, labels = scatter.legend_elements()
legend_labels = [colors[k] for k in colors.keys()]
legend_colors = [colors[k] for k in colors.keys()]
for color, label in zip(legend_colors, colors.keys()):
    ax.scatter([], [], c=color)
ax.legend(frameon=True, title="")
plt.show()
```

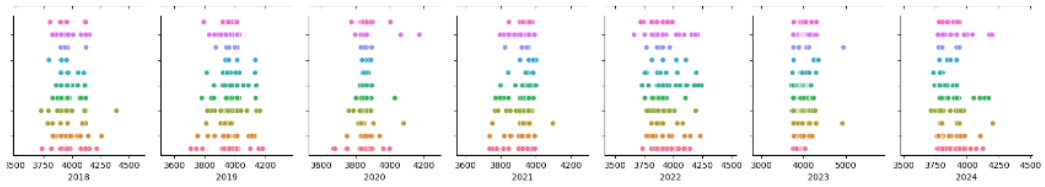
```
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.datasets import make_blobs
import numpy as np

#Aqui hago esto pero en realidad creo que si lo puedo sacar
y = completos_pca[0:,1]
X = completos_pca[0:,0]

num_clusters = 3

#Aqui lo que hago es tipo digo a Kmeans que quiero 3 clusters y esos clusters deben ser
#separdos en base a mi info de completosPCA
kmeans = KMeans(n_clusters=num_clusters, random_state=0)
clusters = kmeans.fit_predict(completos_pca)

fig = plt.figure(figsize=(5, 5))
plt.rcParams["font.family"] = "serif"
ax = fig.add_subplot(1, 1, 1)

for cluster_id in range(num_clusters):
    ax.scatter(completos_pca[clusters == cluster_id, 0], completos_pca[clusters == cluster_id, 1], s=20)

ax.legend()

ax.set_title("Componentes Principales", fontsize=15, fontstyle="italic")
ax.set_xlabel("PCA 1", fontsize=10)
ax.set_ylabel("PCA 2", fontsize=10)
ax.grid(True)
```

```
x_min, x_max = completos_pca[:, 0].min() - 1, completos_pca[:, 0].max() + 1
y_min, y_max = completos_pca[:, 1].min() - 1, completos_pca[:, 1].max() + 1
xx, yy = np.meshgrid(np.arange(x_min, x_max, 0.02), np.arange(y_min, y_max, 0.02))

from sklearn.svm import SVC
Clf = SVC(kernel='linear', C=1.0)
Clf.fit(completos_pca, clusters)

Z = Clf.predict(np.c_[xx.ravel(), yy.ravel()])
Z = Z.reshape(xx.shape)

plt.contourf(xx, yy, Z, alpha=0.3, cmap=plt.cm.coolwarm)

plt.show()
```

```
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.datasets import make_blobs
import numpy as np
from matplotlib.colors import ListedColormap

# Aquí defines tus datos (completos_pca) y el número de clusters
num_clusters = 3

# Aquí creas y ajustas el modelo KMeans
kmeans = KMeans(n_clusters=num_clusters, random_state=0)
clusters = kmeans.fit_predict(completos_pca)

# Calcula el porcentaje de puntos en cada cluster
cluster_counts = np.bincount(clusters)
cluster_percentages = cluster_counts / len(clusters) * 100

# Crea el gráfico
fig = plt.figure(figsize=(5, 5))
plt.rcParams["font.family"] = "serif"
ax = fig.add_subplot(1, 1, 1)

# Itera sobre los clusters para graficarlos y agregar la leyenda con los porcentajes
colors = ['blue', 'orange', 'green']
for cluster_id in range(num_clusters):
    ax.scatter(completos_pca[clusters == cluster_id, 0], completos_pca[clusters == cluster_id, 1],
               s=20, color=colors[cluster_id], label=f'Cluster {cluster_id} ({cluster_percentages[cluster_id]:.2f}%)')

ax.legend()
```

```
ax.set_title("Componentes Principales", fontsize=15, fontstyle="italic")
ax.set_xlabel("PCA 1", fontsize=10)
ax.set_ylabel("PCA 2", fontsize=10)
ax.grid(True)

# Resto del código para el gráfico de contorno y mostrar el gráfico
x_min, x_max = completos_pca[:, 0].min() - 1, completos_pca[:, 0].max() + 1
y_min, y_max = completos_pca[:, 1].min() - 1, completos_pca[:, 1].max() + 1
xx, yy = np.meshgrid(np.arange(x_min, x_max, 0.02), np.arange(y_min, y_max, 0.02))

from sklearn.svm import SVC
Clf = SVC(kernel='linear', C=1.0)
Clf.fit(completos_pca, clusters)

Z = Clf.predict(np.c_[xx.ravel(), yy.ravel()])
Z = Z.reshape(xx.shape)

# Define el colormap con los colores específicos
cmap = ListedColormap(['lightsteelblue', 'peachpuff', 'palegreen'])
plt.contourf(xx, yy, Z, alpha=0.3, cmap=cmap)

plt.show()
```

El sns.pairplot(ImputedData, hue = 'Pais') nos puede dar una pista de que es lo que podemos estar buscando, ya que se puede el rendimiento de cada pais en cada año



---------- **Grafica a base de promedios imputados** ----------

```
[ ]  ImputedData
     ImputedData.rename(
       columns={"2018": "2018"},inplace=True)
```

```
[ ]  #se saca el promedio de cada pais
     promedio_tiempos = ImputedData.groupby('Pais')[['2018', '2019', '2020', '2021', '2022', '2023', '2024']].mean()
```

```
[ ]  #Se transponen los datos
     promedio_tiempos_transp = promedio_tiempos.transpose()

     # Graficar los datos
     plt.figure(figsize=(10, 6))
     for pais in promedio_tiempos_transp.columns:
         plt.plot(promedio_tiempos_transp.index, promedio_tiempos_transp[pais], label=pais)

     plt.xlabel('Año')
     plt.ylabel('Tiempo Promedio')
     plt.title('Tiempo vs Año - Promedio por País')
     plt.legend()
     plt.grid(True)
     plt.xticks(rotation=45)
     plt.tight_layout()
     plt.show()
```

```
[ ]  #Aqui se crea la grafica con plotly para que sea interactiva
     import plotly.graph_objects as go

     fig = go.Figure()
     for pais in promedio_tiempos_transp.columns:
         fig.add_trace(go.Scatter(x=promedio_tiempos_transp.index, y=promedio_tiempos_transp[pais], mode='lines', name=pais))

     fig.update_layout(
         title='Tiempo vs Año - Promedio por País',
         xaxis=dict(title='Año', tickangle=-45),
         yaxis=dict(title='Tiempo Promedio'),
         legend=dict(x=0, y=1),
         margin=dict(l=50, r=50, t=50, b=50),
         plot_bgcolor='white',
         hovermode='x'
     )

     fig.show()
```

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
import numpy as np

# Calcular el promedio de tiempos por país
promedio_tiempos = ImputedData.groupby('Pais')[['2018', '2019', '2020', '2021', '2022', '2023', '2024']].mean()

# Crear una copia del DataFrame para trabajar con las predicciones
predicciones_tiempos = promedio_tiempos.copy()

# Entrenar el modelo de regresión lineal y predecir para cada país
for pais in promedio_tiempos.index:
    # Datos de entrada (años 2018-2024)
    X = np.array([2018, 2019, 2020, 2021, 2022, 2023, 2024]).reshape(-1, 1)
    # Datos de salida (tiempos promedio)
    y = predicciones_tiempos.loc[pais, ['2018', '2019', '2020', '2021', '2022', '2023', '2024']].values

    # Verificar si hay datos faltantes
    if np.isnan(y).any():
        print(f'Advertencia: Datos faltantes para {pais}')
        continue  # Saltar a la siguiente iteración

    # Crear y entrenar el modelo de regresión lineal
    modelo = LinearRegression()
    modelo.fit(X, y)
```

```
        # Predecir el tiempo promedio para 2025
        prediccion_2025 = modelo.predict(np.array([[2025]]))

        # Añadir la predicción al DataFrame
        predicciones_tiempos.loc[pais, '2025'] = prediccion_2025[0]

    # Transponer los datos para la gráfica
    promedio_tiempos_transp = promedio_tiempos.transpose()
    predicciones_tiempos_transp = predicciones_tiempos.transpose()

    # Graficar los datos originales y las predicciones
    plt.figure(figsize=(10, 6))

    # Graficar las predicciones
    for pais in predicciones_tiempos_transp.columns:
        plt.plot(predicciones_tiempos_transp.index, predicciones_tiempos_transp[pais], linestyle='-')

    plt.xlabel('Año')
    plt.ylabel('Tiempo Promedio')
    plt.title('Tiempo vs Año - Promedio por País con Predicción para el 2025')
    plt.legend()
    plt.grid(True)
    plt.xticks(rotation=45)
    plt.tight_layout()
    plt.show()
```