Algoritmos supervisados de ML aplicados a la SABRmetría del béisbol

García Luján Eric Brandon*

Resumen

Este es un artículo para la materia *Aprendizaje Automatizado*, de la Facultad de Ciencias Físico Matemáticas por parte de la Universidad Autónoma de Nuevo León. El objetivo es investigar sobre algoritmos supervisados que puedan aplicarse a un conjunto de datos de béisbol extraído de la librería *pybaseball* de python y aplicar al menos un algoritmo. Además, se dará un panorama sobre el modelo matemático que emplea y explicare por qué conviene utilizarlo en este tipo de conjunto de datos.

Para este trabajo, el artículo Use of Machine Learning and Deep Learning to Predict the Outcomes of Major League Baseball Matches servirá como una guía para la redacción de este.

Palabras Clave: Machine Learning: supervised learning — Algoritmos Supervisados: regresión, clasificación — Deportes: béisbol, Sabermetrics — Herramientas: PyBaseball,Statcast, Baseball Reference,Baseball Savant, FanGraphs

1 Introducción

El béisbol es conocido como "El rey de los deportes" y a la vez el deporte de las estadísticas y lleva años siendo caso de estudio para la estadística deportiva. En el año 1971 Bill James dio origen a la *"Sabermetría"* que es un anglicismo para SABRmetrics, el cuál viene de las siglas *Society for American Baseball Research (Sociedad Americana de Investigación del Béisbol)*.

Hace algunos años en la sabermetría solo se usaban estadísticas básicas sobre pitcheo y bateo, hoy en día recabar información de un partido de béisbol se ha vuelto más fácil, rápido y sobre todo más preciso gracias al uso de maquinas de reconocimiento de pitcheo como Trackman, FlightScope y Rapsodo.

El artículo que se usa como guía para el presente escrito menciona los algoritmos supervisados, técnicas y metodologías que han sido utilizados en estudios previos similares al nuestro y que pueden ser utilizados en nuestros datos dependiendo el tipo de problema a resolver, algunos de esos algoritmos son:

- **Predicción de jugadas o resultados específicos:** SVM, Random Forest, redes neuronales. Se puede usar para predecir si el siguiente lanzamiento será fastball, slider o si un batazo podrá ser hit.
- **Evaluación del rendimiento de jugadores:** XGBoost y deep learning. Se busca estimar métricas de desempeño, fatiga y/o efectividad ya que busca identificar las variables que más influyen.
- **Estimación del valor de los jugadores:** Random Forest, redes neuronales o regresiones múltiples. Se puede modelar (proyectar)un valor económico del jugador en función del rendimiento deportivo.
- **Predicción de lesiones:** SVM, LSTM y regresión logística. Se emplean datos biométricos, cargas de trabajo y trayectorias de lanzamiento para detectar patrones previos de lesiones.
- **Predicción de resultados de partidos o temporadas:** CGBoost, árboles de decisión y redes neuronales. Se combinan variables de jugadores, equipos, clima y rendimiento histórico para estimar probabilidades de victoria.

2 ¿El tipo de pitcheo influye para que un jugador decida hacer swing?

Este trabajo tiene como objetivo definir si el tipo de pitcheo influye para que un bateador haga swing o no. Se aplicará un algoritmo de regresión logística y uno de árbol de decisión con la finalidad de comparar ambos resultados.

*Facultad de Ciencias Físico Matemáticas, UANL, México. Correo: brandonglujan@gmail.com

2.1 Definición de la variable objetivo: Swing

Definimos como **swing** todo registro que en el campo de *description* contenga alguna de las siguientes categorías de eventos:

2.2 Definición de la variable objetivo: Swing

Definimos como **swing** todo registro que en el campo de *description* contenga alguna de las siguientes categorías de eventos:

`swinging_strike`,	*strike con swing*
`foul`,	*foul siempre implica swing*
`hit_into_play`,	*pelota puesta en juego*
`foul_tip`,	*foul siempre implica swing*
`foul_bunt`,	*foul siempre implica swing*
`swinging_strike_blocked`	*strike con swing bloqueado*

Por lo tanto, se ha creado una columna nueva en el conjunto de datos donde:

$$Swing = \begin{cases} 1, & \text{si } hay\ swing, \\ 0, & \text{si } no\ hay\ swing. \end{cases} \quad (1)$$

2.3 Explicación de los modelos a aplicar

Para comprender mejor los algoritmos aplicados se presenta a continuación una breve descripción matemática de cada uno, al aplicar ambos algoritmos lo que se desea es comparar el resultado final de cada uno.

2.3.1 Regresión logística

La regresión logística se utiliza para estimar la relación entre una variable dependiente y una o más variables independientes, es decir, una predicción sobre una variable categórica frente a una continua. Para su uso, la regresión logística utiliza la siguiente jerarquía de ecuaciones:

Ecuación lineal (Logit)

La combinación lineal de las características de entrada ($\mathbf{x}$) se define como el Logit (y):

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_n x_n$$

Función sigmoide

Esta función transforma el Logit (y) en una probabilidad (P) acotada entre 0 y 1:

$$P(Y = 1|\mathbf{x}) = \frac{1}{1 + e^{-y}}$$

3. Ecuación completa de la regresión logística

Sustituyendo el Logit (y) en la función sigmoide, se obtiene la expresión final:

$$P(Y = 1|\mathbf{x}) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_n x_n)}}$$

2.3.2 Árbol de decisión

Es un algoritmo supervisado no paramétrico y tiene como objetivo crear un modelo que prediga el valor de una variable objetivo (en este ejercicio, esa variable es si el bateador hace swing o no) mediante el aprendizaje de reglas de decisión simples inferidas a partir de las características de los datos.

2.4 Selección de datos

Para la selección de datos se cuenta con la librería *pybaseball* de Python. Esta librería se encarga de facilitar la extracción y agregación de datos provenientes de las principales bases de datos de análisis del béisbol: *Baseball Reference*, *Baseball Savant* y *FanGraphs*. A continuación, se presenta una descripción concisa de cada una:

- **Baseball Reference (B-Ref)** Es la base de datos histórica del béisbol más antigua y extensa. Ofrece datos históricos, estadísticas tradicionales (e.g., AVG, HR, ERA) y métricas de valor históricamente contextualizadas (e.g., bWAR).
- **Baseball Savant** Es la interfaz web oficial que visualiza y procesa los datos cinéticos y de seguimiento de alta resolución recogidos por el sistema *Statcast* de la MLB.
- **FanGraphs (FG)** Es un sitio web centrado en el análisis "sabermétrico" avanzado. Ofrece métricas que buscan medir el proceso del juego y el rendimiento subyacente del jugador, aislando factores como el azar o el fildeo (e.g., FIP, wRC+).
- **Pybaseball** Es una librería de Python diseñada para automatizar y simplificar el proceso de extracción de datos limpios y estructurados de las fuentes mencionadas, facilitando su integración directa en flujos de trabajo de análisis y modelado de datos.

Para el conjunto de datos se ha tomado el 70% del total de los datos para entrenar el modelo y el 30% para evaluar qué tan bien se ajusta nuestro modelo a datos nuevos.

3 Algoritmo de regresión logística

Se toma como variable objetivo *swing* y como única variable predictiva *pitch_type*. El modelo se entrenó con un límite de 1000 iteraciones asegurando la convergencia del algoritmo, es decir para que se acerque a una solución donde se asegura que el modelo "ha terminado de aprender" y no puede mejorar significativamente más. Los resultados obtenidos a través de esta prueba dieron una exactitud del 52.22%, es decir en el 52% de los lanzamientos nuestro modelo de regresión logística pudo predecir correctamente si el bateador haría o no swing lo que se puede definir como un modelo conservador y casi azaroso.

3.1 Métricas de desempeño para la regresión logística

Para evaluar la capacidad del modelo de regresión logística para clasificar correctamente los datos, se emplearon las métricas de *precisión* (precisión), *sensibilidad* (recall) y *F1-score*. Estas métricas permiten analizar el equilibrio entre predicciones correctas y errores para cada clase (swing y no swing) y en la tabla 1. se explica cada uno de ellos y qué es lo que miden.

Table 1: Descripción de las métricas de desempeño del modelo.

Métrica	¿Qué mide?
Precisión	Las veces que el modelo predijo *swing*, ¿cuántas predicciones fueron correctas?
Sensibilidad	De todos los swings reales, ¿cuántos logró detectar el modelo?
F1-score	Promedio armónico entre precisión y sensibilidad; mide el balance entre ambas.

Al analizar el desempeño por clase se observa que el modelo identifica con mayor facilidad los lanzamientos en los que no se realiza swing Los resultados obtenidos se presentan en la tabla 2.

Table 2: Métricas de desempeño de la regresión logística

Clase	Precisión	Sensibilidad	F1-score	Soporte
No Swing (0)	0.52	0.87	0.66	115,023
Swing (1)	0.50	0.14	0.22	105,771

La siguiente imagen muestra una matriz de confusión y que puede interpretarse de la siguiente forma:

- **Clase Real: No Swing**
  - **100,470 aciertos (TN):** fueron *No Swing* y el modelo predijo *No Swing*.
  - **14,553 falsos positivos (FP):** fueron *No Swing* pero el modelo predijo *Swing*.
  - Esto indica que el modelo es razonablemente bueno detectando la clase *No Swing*, aunque aún comete un número considerable de predicciones incorrectas hacia la clase *Swing*.
- **Clase Real: Swing**
  - **14,830 verdaderos positivos (TP):** fueron *Swing* y el modelo acertó.
  - **90,941 falsos negativos (FN):** fueron *Swing* pero el modelo predijo *No Swing*.
  - Esto muestra que el modelo tiene serias dificultades para identificar swings reales, presentando un número muy elevado de falsos negativos.

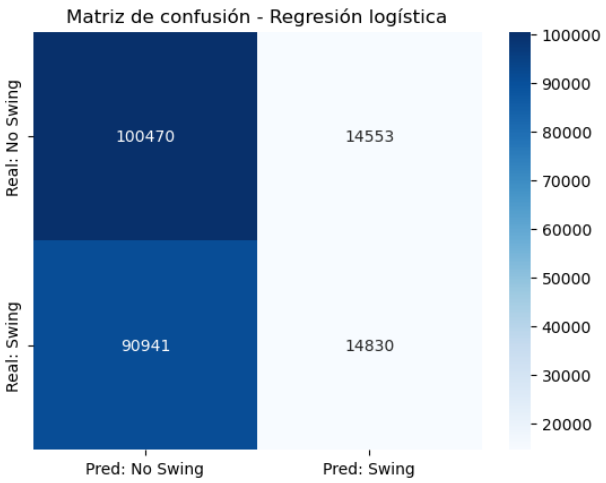

Figure 1: Matriz de confusión obtenida con la regresión logística, donde se observa el desempeño del modelo al distinguir entre swings y no swings.El modelo clasifica correctamente la mayoría de los casos de “No Swing”, pero presenta un alto número de falsos negativos al predecir swings reales, evidenciando un fuerte desbalance en la detección de la clase positiva.

4 Algoritmo de árbol de decisión

Para la segunda etapa del análisis se implementó el algoritmo de árbol de decisión con el mismo objetivo del algoritmo anterior, es decir poder predecir qué tipo de pitcheo está más asociado a que un bateador realice swing o no. Para probar este algoritmo se usó el criterio de impureza de Gini el cual evalúa cuán mezcladas están las clases dentro de cada nodo. Este modelo también se entrenó utilizando el 70% de los datos y evalúa su desempeño con el 30% restante,

4.1 Métricas de desempeño para el árbol de decisión

Al igual que en regresión logística, se realizó la evaluación de desempeño del algoritmo de árbol de decisión utilizando las métricas de *precisión* (precisión), *sensibilidad* (recall) y *F1-score* obteniendo los resultados en la tabla 3. Como se puede observar, son los mismos resultados que en el algoritmo de regresión logística.

Table 3: Métricas de desempeño del árbol de decisión

Clase	Precisión	Sensibilidad	F1-score	Soporte
No Swing (0)	0.52	0.87	0.66	115,023
Swing (1)	0.50	0.14	0.22	105,771

Como refuerzo a la tabla 3, se muestra la imagen (2) de la matriz de confusión sobre las predicciones y sus valores verdaderos. Se puede observar que es muy similar en cifras y proporciones a la regresión logística, como lo muestran las métricas anteriores.

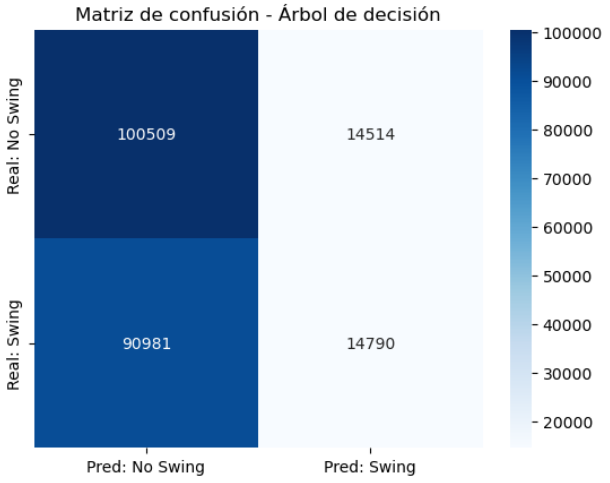

Figure 2: Matriz de confusión obtenida con el árbol de decisión, donde se observa el desempeño del modelo al distinguir entre swings y no swings.El modelo clasifica correctamente la mayoría de los casos de "No Swing", pero presenta un alto número de falsos negativos al predecir swings reales, evidenciando un fuerte desbalance en la detección de la clase positiva.

Esta matriz, al igual que la anterior se interpreta de la siguiente forma:

- **Clase Real: No Swing**
  - **100,509 aciertos (TN):** fueron *No Swing* y el modelo predijo *No Swing*.
  - **14,514 falsos positivos (FP):** fueron *No Swing* pero el modelo predijo *Swing*.
  - Esto indica que el modelo es razonablemente bueno detectando la clase *No Swing*, aunque aún comete un número considerable de predicciones incorrectas hacia la clase *Swing*.
- **Clase Real: Swing**
  - **14,790 verdaderos positivos (TP):** fueron *Swing* y el modelo acertó.
  - **90,981 falsos negativos (FN):** fueron *Swing* pero el modelo predijo *No Swing*.
  - Esto muestra que el modelo tiene serias dificultades para identificar swings reales, presentando un número muy elevado de falsos negativos.

5 Conclusiones

Este estudio solo tomó como variable predictiva única al tipo de pitcheo, sin embargo si se quisiera realizar un estudio más amplio se recomienda agregar más variables de modo que el algoritmo se robustezca, además, como la finalidad de este artículo es comparar ambos modelos para la predicción de un modelo que usa solamente una variable predictiva podemos concluir que este tipo de modelos no son útiles por sus datos similares y se concluye que para que un jugador haga swing o no pueden influir más variables que no se tomaron en cuenta para este estudio como la velocidad, la distancia del pitcher al plato o tal vez la que para muchos pudiera ser obvia la ubicación de la bola al llegar a la zona de strike. A partir del análisis del código y los datos utilizados, se identifican las siguientes razones por las cuales la regresión logística y el árbol de decisión arrojaron prácticamente los mismos resultados:

- **Uso de una sola variable predictora (`pitch_type`):** El modelo únicamente emplea el tipo de pitcheo como variable independiente, lo cual limita severamente la información disponible para distinguir entre las clases *swing* y *no swing*.
- **Codificación *one-hot* que genera variables binarias simples:** La transformación con `OneHotEncoder` produce columnas binarias independientes por cada tipo de lanzamiento, lo que permite que tanto la Regresión Logística como el Árbol de Decisión aprendan reglas prácticamente idénticas.
- **La probabilidad de hacer swing no depende mayormente del tipo de pitcheo:** En Statcast, variables como la zona del lanzamiento, la velocidad, el conteo y la ubicación del lanzamiento explican mucho mejor el comportamiento del bateador. Al usar únicamente `pitch_type`, la señal predictiva es débil y ambos modelos convergen a la misma solución.
- **El Árbol de Decisión no puede crear divisiones profundas:** Aunque el árbol no tiene límite de profundidad, la falta de variables adicionales impide que genere ramas complejas.

Esto provoca que la frontera de decisión que aprende sea casi idéntica a la de la Regresión Logística.

En conjunto, estas condiciones explican por qué ambos modelos produjeron resultados equivalentes utilizando la misma información limitada.

Referencias

Amazon Web Services. (2024). *¿Qué es la regresión logística?* [Accedido: 2025-11-14]. https://aws.amazon.com/es/what-is/logistic-regression/

Anastassiou, G. A. (2021). Artificial Intelligence and Data Analytics in Baseball: A Review [Accedido: 2025-11-14]. *Applied Sciences*, *11*(10), 4499. https://doi.org/10.3390/app11104499

Fundación Alfredo Harp Helú Oaxaca. (2023). *Sabermetría en el béisbol y la revolución de la data* [Accedido: 2025-11-14]. https://fahho.mx/sabermetria-en-el-beisbol-y-la-revolucion-de-la-data/

IBM. (2024). *Regresión logística: fundamentos, aplicaciones y ejemplos* [Accedido: 2025-11-14]. https://www.ibm.com/mx-es/think/topics/logistic-regression

PyBaseball Developers. (2024). *PyBaseball: Python package for baseball data* [Accedido: 2025-11-14]. https://pypi.org/project/pybaseball/