Proyecto de Simulación

Predicción de resultados en la MLB

Alejandro Álvarez Lamazares - C311
Marian S. Álvarez Suri - C312
Carlos A. Bresó Sotto - C312

1 Introducción

1.1 Breve descripción del proyecto

Este proyecto tiene como objetivo utilizar una simulación de eventos discretos y el análisis de una tabla de estadísticas de la Major League Baseball (MLB) para predecir los resultados de la segunda mitad del campeonato. Se busca anticipar el clasificador final de la tabla de posiciones basándose exclusivamente en los datos disponibles hasta ese momento.

1.2 Objetivos y metas

El propósito de este proyecto es aproximarse lo más posible al resultado factual de la temporada de la MLB. Se evaluará la precisión de las simulaciones utilizando los datos reales del problema como base.

1.3 Variables que describen el problema

Se realizará una simulación de la segunda mitad de la temporada de la MLB del año 2021, utilizando los resultados obtenidos en la primera mitad. Se incorporarán variables adicionales, como las posibles lesiones de jugadores, que podrían influir en el desenlace de cada partido.

2 Detalles de implementación

El código comienza importando las bibliotecas necesarias: pandas para el manejo de datos, random para la generación de números aleatorios y numpy para las operaciones matemáticas.

A continuación, se explicará el código por funciones:

- ✓ **get_history(team1, team2, results)**: devuelve el número de partidos jugados y ganados entre dos equipos.
- ✓ **simulate_injured_players(p=0.5)**: simula si hay jugadores lesionados en un equipo, utilizando una distribución binomial con $p = 0.5$, donde p es la probabilidad de éxito.
- ✓ **simulate_game(team1, team2, results, game_simulations)**: simula un partido entre dos equipos basándose en los resultados históricos y en la posibilidad de que haya jugadores lesionados. Para esto, se calcula una tasa llamada "win_rate", que se utiliza en una simulación de Monte Carlo para estimar el ganador del partido.

- ✓ **simulate_season(statistics, game_simulations)**: simula una temporada completa, haciendo que cada equipo juegue contra todos los demás. Por cada par de equipos, se simula el juego y se guardan los resultados.
- ✓ **create_results_table(total_wins)**: convierte los resultados de la simulación en un DataFrame de pandas y lo ordena por el número total de victorias.
- ✓ **get_sim_results(num_simulations=30, game_simulations=100)**: ejecuta la simulación de la temporada varias veces y devuelve los resultados en forma de tabla.
- ✓ **get_real_results()**: obtiene los resultados reales de la temporada.
- ✓ **calculate_position_distances(df_real, df_simulated)**: calcula la diferencia entre las posiciones reales y simuladas de cada equipo.
- ✓ **print_results(num_simulations, game_simulations)**: ejecuta todo el proceso varias veces y devuelve la distancia media entre las posiciones reales y simuladas de los equipos.

El código termina llamando a **print_results(150, 400)**, lo que significa que se ejecutan 150 simulaciones de la temporada, cada una con 400 simulaciones de partidos, y se imprime la distancia media entre las posiciones reales y simuladas.

3 Resultados y experimentos

La simulación realizada ha mostrado una notable coincidencia con los resultados finales de la temporada observada. A continuación, se presenta una representación gráfica que ilustra la comparación entre los datos reales y los obtenidos a través de la simulación.

3.1 Hallazgos de la simulación

3.2 Interpretación de los resultados

Para profundizar en la comparación entre los datos reales y los simulados, se aplica la función calculate_position_distances(df_real, df_simulated). Esta herramienta permite identificar y cuantificar las diferencias entre ambos conjuntos de datos, ofreciendo una visión clara sobre la precisión de la simulación.

3.3 Hipótesis extraídas de los resultados

Basándose en los hallazgos obtenidos, se puede formular la hipótesis de que, mediante el uso de simulaciones, es factible predecir con un alto grado de precisión el resultado final de la tabla de posiciones de una temporada de la Major League Baseball, incluso antes de su conclusión. Esto subraya la utilidad de las simulaciones como herramientas predictivas en el ámbito deportivo, especialmente cuando se dispone de datos relevantes de la primera mitad de la temporada.

3.4 Experimentos realizados para validar las hipótesis

Se realizaron múltiples ejecuciones de la simulación con el objetivo de confirmar que los resultados obtenidos no sean el producto de circunstancias fortuitas o aleatorias, sino que reflejen una tendencia consistente y válida. Este enfoque metodológico permite establecer una mayor confianza en la precisión y fiabilidad de los hallazgos, al minimizar el riesgo de atribuir el éxito de la predicción a factores externos o a la casualidad.

3.5 Necesidad de realizar el análisis estadístico de la simulación (Variables de interés)

3.6 Análisis de parada de la simulación

Para determinar cuándo detener la simulación, es necesario establecer un criterio de parada que garantice la obtención de resultados precisos y confiables. En este caso, se optó por realizar múltiples ejecuciones de la simulación y calcular la distancia media entre las posiciones reales y simuladas de los equipos. Este enfoque permite evaluar la consistencia y estabilidad de los resultados, así como identificar posibles tendencias o patrones que puedan surgir a lo largo de las simulaciones.