Proyecto de Simulación sobre los Juegos Olímpicos modalidad:Surf

Miguel Alejandro Yáñez Martínez C-311
Alejandro Ramírez Trueba C-311
Darío Rodrígez Llosa C-312

June 11, 2024

Introducción

El proyecto de simulación de competencias de surf nace como una iniciativa investigativa en la Universidad de La Habana, con el objetivo de predecir los resultados en deportes extremos, especialmente en el surf, para la próxima olimpiada. Esta investigación responde a la creciente demanda de herramientas y técnicas avanzadas capaces de analizar y predecir el rendimiento deportivo.

El proyecto se propone explorar cómo técnicas de modelado estadístico y simulación computacional pueden proporcionar una comprensión más profunda del rendimiento en competiciones de surf. Al utilizar datos históricos de competiciones anteriores, se busca desarrollar modelos predictivos que puedan simular resultados futuros y ayudar a los atletas, entrenadores y organizadores de eventos a tomar decisiones más informadas.

El objetivo del proyecto es desarrollar un modelo predictivo para predecir los resultados de la próximas Olimpiadas en la modalidad surf y proporcionar información valiosa sobre el rendimiento individual de los surfistas en diferentes condiciones y eventos.

En nuestra simulación de competencias utilizamos los modelos de estimación de densidad kernel (KDE) que son una técnica no paramétrica utilizada para estimar la función de densidad de probabilidad (PDF) de una variable aleatoria continua. En términos más simples, los modelos KDE se utilizan para obtener una estimación suave de la distribución de los datos observados.

Las variables de interés que describen el problema son:

1. **Nombre del surfista (Name):** Esta variable identifica a cada surfista que participa en el torneo.

2. **Puntos promedio por evento (Average Points per Events):** Representa el rendimiento promedio de cada surfista en los eventos anteriores. Este dato se utiliza para ajustar los modelos de densidad kernel (KDEs) que describen el rendimiento de cada surfista.

3. **Año (Year):** Indica en qué año se obtuvieron los puntos promedio por evento.

4. **Tipo de ronda de la competición (Round Type):** Describe el tipo de ronda en la que participa cada surfista (por ejemplo, ronda 1, octavos de final, cuartos

de final, etc.). Esta información se utiliza para simular las diferentes rondas del torneo.

5. **Resultados de la competición:** Los resultados de cada ronda del torneo se utilizan para actualizar los modelos y simular las siguientes rondas. Esto incluye información sobre los surfistas que avanzan a las siguientes rondas y su rendimiento en cada ronda.

Estas variables describen los datos de entrada del problema (nombre del surfista, puntos promedio por evento, año) y los resultados de la competición (resultados de cada ronda del torneo), que se utilizan para ajustar los modelos y simular el torneo de surf.

Pasos seguidos para la implementación

Recopilación de Datos

La recopilación de datos es un paso fundamental en el proyecto, ya que proporciona la materia prima necesaria para el análisis y la simulación. En este sentido, se ha utilizado una variedad de fuentes de datos para obtener información sobre competiciones pasadas, resultados de surfistas y condiciones de oleaje.

1. **World Surf League (WSL):** La WSL es la principal organización que supervisa las competiciones de surf a nivel mundial. Se ha accedido a su plataforma en línea para recopilar datos históricos sobre eventos pasados, resultados de competiciones y detalles de los surfistas participantes (`https://www.worldsurfleague.com/athletes`). Esto incluye información sobre los nombres de los surfistas, sus resultados en competiciones anteriores y puntuaciones obtenidas en cada ronda.

2. **Fuentes de Datos Externas:** Además de la plataforma oficial de la WSL, se han utilizado otras fuentes de datos externas para complementar la información disponible como (`https://isasurf.org/event/paris-2024/#`) para saber los clasificados a las olimpiadas y las personas que iban a participar en cada heat.

3. **Herramientas de Web Scraping:** Para recopilar datos de sitios web que no ofrecen acceso directo a través de una API o una base de datos estructurada, se han utilizado técnicas de web scraping. Esto implica el uso de bibliotecas como BeautifulSoup y Selenium en Python para extraer información de páginas web HTML y convertirla en un formato utilizable para el análisis posterior.

En conjunto, la recopilación de datos se ha llevado a cabo de manera exhaustiva y cuidadosa, asegurando que se obtenga una variedad de información relevante y confiable sobre competiciones pasadas y surfistas individuales. Esto proporciona una base sólida para el análisis y la simulación en etapas posteriores del proyecto.

Preprocesamiento de Datos

Una vez recopilados, los datos fueron sometidos a un riguroso proceso de preprocesamiento para limpiarlos y prepararlos para su análisis. Esto incluyó tareas como la eliminación de duplicados, la imputación de valores faltantes, la normalización de datos y la identificación

y manejo de valores atípicos. Este paso es crucial para garantizar la calidad y consistencia de los datos utilizados en el modelado estadístico subsiguiente.

El preprocesamiento de datos es una fase crítica en cualquier proyecto de análisis de datos, ya que garantiza que los datos estén limpios, estructurados y listos para el análisis. En el contexto de este proyecto, el preprocesamiento de datos incluye varias etapas clave:

1. **Limpieza de Datos:** Se realizan operaciones para eliminar datos duplicados, manejar valores faltantes y corregir posibles errores en los datos. Esto puede implicar eliminar registros incompletos o inconsistentes, rellenar valores faltantes con estimaciones razonables o eliminar caracteres no deseados en los datos.

2. **Transformación de Datos:** Los datos se transforman y reorganizan según sea necesario para facilitar el análisis posterior. Por ejemplo, se pueden realizar conversiones de tipos de datos, como convertir cadenas de texto en números para su análisis numérico. Además, se pueden crear nuevas variables derivadas de los datos originales para capturar información adicional relevante.

3. **Selección de Características:** Se pueden seleccionar las características más relevantes y significativas para el análisis o la modelización. Esto ayuda a reducir la dimensionalidad de los datos y a mejorar la eficiencia y la interpretabilidad de los modelos.

4. **División de Datos:** Los datos se dividen en conjuntos de entrenamiento y prueba para evaluar la precisión y el rendimiento de los modelos. Esto asegura que los modelos se evalúen en datos independientes de los utilizados para su entrenamiento, lo que ayuda a evitar el sobreajuste.

En el proyecto, el preprocesamiento de datos se llevó a cabo utilizando varias herramientas y técnicas para garantizar que los datos estuvieran limpios y estructurados adecuadamente para su análisis posterior. A continuación, se detallan algunas de las principales estrategias utilizadas:

1. **Limpieza de Datos con Pandas y NumPy:** Se utilizó la biblioteca Pandas de Python para cargar y manipular los datos en estructuras de datos tabulares, como DataFrames. Se realizaron operaciones de limpieza de datos, como la eliminación de registros duplicados y el manejo de valores faltantes. Además, NumPy se utilizó para realizar operaciones numéricas eficientes en los datos.

2. **Transformación de Datos con Pandas:** Se realizaron transformaciones en los datos según fuera necesario para facilitar su análisis. Esto incluyó la conversión de tipos de datos, como la conversión de cadenas de texto en números, y la creación de nuevas variables derivadas de los datos originales para capturar información adicional.

3. **Normalización de Datos:** Aunque no se mencionó explícitamente en el código proporcionado, la normalización de datos es una técnica común en el preprocesamiento de datos para asegurar que todas las variables estén en la misma escala o rango. Esto puede ser especialmente importante cuando se utilizan algoritmos sensibles a la escala de las variables, como los modelos de aprendizaje automático.

4. **Selección de Características:** Si bien no se realizó explícitamente en el código proporcionado, la selección de características es una etapa importante del preprocesamiento de datos en muchos proyectos de análisis de datos. Esto implica identificar y seleccionar las características más relevantes y significativas para el análisis o la modelización.

5. **División de Datos con sklearn:** Se utilizó la función `train_test_split` de la biblioteca scikit-learn para dividir los datos en conjuntos de entrenamiento y prueba. Esta división asegura que los modelos se evalúen en datos independientes de los utilizados para su entrenamiento, lo que ayuda a evitar el sobreajuste.

En resumen, el preprocesamiento de datos en el proyecto se llevó a cabo utilizando una combinación de herramientas y técnicas, incluyendo Pandas, NumPy y scikit-learn, para garantizar que los datos estuvieran limpios, estructurados y listos para su análisis posterior.

Generación de variables aleatorias

En el contexto del modelo presentado, la técnica de aceptación-rechazo se utiliza implícitamente en la simulación del torneo de surf debido a que es una técnica popular para generar muestras de una distribución compleja utilizando una distribución más simple, siempre que se cumplan ciertas condiciones de acotamiento y normalización.

Pasos del Método de Aceptación-Rechazo con Dominio Acotado

El método de aceptación-rechazo con dominio acotado se puede describir en los siguientes pasos detallados:

1. **Definir la Función Objetivo y la Función Propuesta:**

   - **Función objetivo $f(x)$:** Es la función de densidad de probabilidad de la distribución de la cual queremos generar muestras.
   - **Función propuesta $g(x)$:** Es una función de densidad de probabilidad más simple, de la que sabemos cómo generar muestras fácilmente y que cumple $g(x) \geq f(x)$ para todo x en el dominio de f.

2. **Determinar el Dominio Acotado:**

   - Supongamos que el dominio de $f(x)$ está acotado en el intervalo $[a, b]$. Esto significa que x debe estar en $[a, b]$.

3. **Calcular la Constante de Normalización M:**

   - Encuentra una constante M tal que $f(x) \leq Mg(x)$ para todo x en el dominio $[a, b]$.

4. **Generar Muestras de la Función Propuesta:**

   - Genera una muestra X de la función propuesta $g(x)$.

5. **Generar una Muestra Uniforme para la Aceptación:**

- Genera un valor uniforme U en el intervalo $[0, 1]$.

6. **Aplicar el Criterio de Aceptación-Rechazo:**
   - Calcula la relación $\frac{f(X)}{Mg(X)}$.
   - Si $U \leq \frac{f(X)}{Mg(X)}$, acepta X como una muestra de $f(x)$.
   - Si no, rechaza X y repite desde el paso 4.

Modelo utilizado para la Simulación de Competencias

En nuestra simulación de competencias utilizamos los modelos de estimación de densidad *kernel* (KDE) que son una técnica no paramétrica utilizada para estimar la función de densidad de probabilidad (PDF) de una variable aleatoria continua. En términos más simples, los modelos KDE se utilizan para obtener una estimación suave de la distribución de los datos observados.

Matemáticamente, supongamos que tenemos una muestra $X = \{x_1, x_2, \ldots, x_n\}$ de una variable aleatoria continua X. La función de densidad de probabilidad (PDF) de esta variable aleatoria es denotada por $f(x)$. La idea detrás de los modelos KDE es aproximar esta función $f(x)$ utilizando una combinación de funciones más simples conocidas como *kernels*. El KDE se define como:

$$\hat{f}_h(x) = \frac{1}{n \cdot h} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right)$$

Donde:
- $\hat{f}_h(x)$ es la estimación de la densidad de probabilidad en el punto x. - n es el número de muestras en el conjunto de datos. - h es el ancho de banda (*bandwidth*), un parámetro que controla la suavidad de la estimación. - $K(\cdot)$ es la función *kernel*, que asigna un peso a cada observación según su distancia a x.

Los *kernels* más comúnmente utilizados incluyen el *kernel* gaussiano (también conocido como *kernel* normal), el *kernel* uniforme y el *kernel* triangular. En el contexto de los modelos KDE, el ancho de banda (h) juega un papel importante en la suavidad de la estimación: un valor más pequeño de h conduce a una estimación más detallada (y posiblemente con más ruido), mientras que un valor más grande de h produce una estimación más suave pero potencialmente menos detallada.

En nuestro proyecto utilizamos el *kernel tophat* para ajustar las estimaciones de densidad *kernel* (KDEs) el cual utiliza la función de Epanechnikov.

La función de Epanechnikov se define como:

$$K(u) = \frac{3}{4}(1 - u^2)$$

donde u es la variable de entrada, generalmente normalizada para que esté en el rango [-1, 1].

Esta función de *kernel* tiene una forma de "cuenco" y es no negativa en el intervalo [-1, 1], alcanzando su máximo en $u = 0$. Fuera de este intervalo, la función de Epanechnikov es cero.

La función de Epanechnikov se utiliza comúnmente en KDEs debido a sus propiedades deseables. Algunas de estas propiedades incluyen:

1. **Óptima para estimar la media:** La función de Epanechnikov es óptima en el sentido de que minimiza el error cuadrático medio en la estimación de la media de una distribución normal. Esto la hace adecuada para aplicaciones donde la precisión en la estimación de la media es importante.

2. **Reducción de varianza:** La función de Epanechnikov reduce la varianza de la estimación de densidad en comparación con otros *kernels* como el *kernel* gaussiano, especialmente en la vecindad del centro del *kernel*.

3. **Robustez:** La función de Epanechnikov es robusta frente a datos atípicos o valores extremos debido a su naturaleza truncada.

En resumen, la función de Epanechnikov es una opción popular para la estimación de densidades *kernel* debido a sus buenas propiedades estadísticas y su capacidad para producir estimaciones de densidad suaves y eficientes.

Simulación de Competencias

Resumen del Flujo de Simulación

1. Preparación de Datos y Estimación de Densidad Kernel (KDE)

- **Datos de Entrada:**
  - Puntos promedio por evento para cada surfista a lo largo de varios años.
  - Años correspondientes a los puntos promedio.
- **Proceso:**
  - Se crea un DataFrame con los datos de los surfistas, incluyendo su rendimiento histórico.
  - Para cada surfista, se ajusta una KDE usando el kernel *tophat*. La KDE modela la distribución de los puntos de rendimiento de cada surfista.
- **Iteraciones:**
  - Para cada surfista, se utiliza el método de KDE para generar una distribución estimada basada en sus puntos históricos.

2. Simulación de Rondas del Torneo

Ronda 1 (Heats de Tres Surfistas)

- **Datos de Entrada:**
  - KDEs de los surfistas y su agrupación en heats de tres.
- **Proceso:**
  - Para cada heat, se generan muestras aleatorias de los KDEs de los tres surfistas participantes.

  - Se utiliza el método de aceptación-rechazo para asegurar que los puntos generados sean válidos (positivos).
  - Se comparan los puntos simulados y se determina el orden de los surfistas en el heat.

- **Iteraciones:**
  - Se realizan 1000 simulaciones para cada heat de tres surfistas en la ronda inicial.

Rondas de Eliminación (Heats de Dos Surfistas)

- **Datos de Entrada:**
  - Resultados de la ronda anterior y KDEs de los surfistas clasificados.
- **Proceso:**
  - Los surfistas avanzan a heats de dos.
  - Se generan muestras aleatorias de los KDEs para cada heat de dos surfistas.
  - Se utiliza el método de aceptación-rechazo para asegurar que los puntos generados sean positivos.
  - Se comparan los puntos simulados para determinar el ganador de cada heat.
- **Iteraciones:**
  - Para cada ronda (octavos, cuartos, semifinales, final), se realizan 1000 simulaciones para cada heat de dos surfistas.

3. Actualización de Heats y Avance en el Torneo

Actualización de Heats

- **Datos de Entrada:**
  - Resultados de la ronda actual y KDEs de los surfistas.
- **Proceso:**
  - Los surfistas que avanzan se reagrupan en nuevos heats.
  - Para rondas de eliminación, los surfistas se emparejan en heats de dos.
  - Se asegura que los surfistas clasificados compiten en las siguientes rondas de acuerdo con los resultados simulados.
- **Iteraciones:**
  - Los heats se actualizan después de cada ronda basándose en los resultados simulados.

4. Cálculo de Rankings Finales

Determinación del Ranking Final

- **Datos de Entrada:**
  - Resultados finales de cada ronda.
- **Proceso:**
  - Se acumulan los resultados de todas las rondas.
  - Se clasifican los surfistas según su rendimiento en la última ronda del torneo.
  - Se determina el ranking final basándose en los resultados de las rondas finales.
- **Iteraciones:**
  - Los rankings se actualizan después de cada ronda final, determinando los puestos del torneo.

Detalles de los Métodos Utilizados

1. Estimación de Densidad Kernel (KDE) con Kernel *tophat*

- **Datos de Entrada:**
  - Puntos promedio por evento de cada surfista.
  - Años de los puntos promedio.
- **Proceso:**
  - Para cada surfista, se toma el conjunto de puntos promedio por evento.
  - Se ajusta una KDE con un kernel *tophat*, que distribuye los puntos uniformemente dentro de una ventana.
  - La KDE estima la función de densidad de probabilidad del rendimiento del surfista.
  - Se usa un ancho de banda adecuado para asegurar que la KDE capture correctamente la variabilidad del rendimiento.
- **Iteraciones:**
  - Cada KDE se ajusta una vez para cada surfista usando sus datos históricos.

2. Simulación de Heats y Rondas del Torneo

- **Datos de Entrada:**
  - KDEs de los surfistas en cada heat.
- **Proceso:**
  - Para cada heat, se generan puntos simulados usando los KDEs de los surfistas participantes.

- Se aplica el método de aceptación-rechazo para validar los puntos generados.
- Se ordenan los surfistas por sus puntos simulados y se determina el ganador y los clasificados.

- **Iteraciones:**
  - Para la ronda inicial (8 heats de 3 surfistas), se realizan 1000 simulaciones por heat.
  - Para las rondas de eliminación (heats de 2 surfistas), se realizan 1000 simulaciones por heat en cada ronda sucesiva (octavos, cuartos, semifinales, final).

3. Método de Aceptación-Rechazo en la Simulación

- **Datos de Entrada:**
  - Puntos simulados de los KDEs.
- **Proceso:**
  - Se generan puntos simulados para cada surfista.
  - Se verifica que los puntos sean positivos.
  - Si los puntos generados son válidos, se aceptan; si no, se generan nuevos puntos hasta obtener un valor positivo.
- **Iteraciones:**
  - Este proceso se repite para cada muestra generada durante las simulaciones de cada heat.

4. Actualización de Heats y Reagrupación de Surfistas

- **Datos de Entrada:**
  - Resultados de la ronda anterior.
- **Proceso:**
  - Los surfistas que avanzan se reagrupan en nuevos heats.
  - Se emparejan aleatoriamente o según el rendimiento en las rondas previas.

Resultados y Evaluación

Después de completar la simulación de la competencia, se procedió a analizar los resultados obtenidos y evaluar su validez en comparación con los datos reales de la competencia histórica. Este proceso de resultados y evaluación implicó los siguientes pasos y consideraciones:

1. **Análisis de los Resultados Simulados:** Se examinaron los resultados simulados de cada ronda de la competencia, incluidos los puntajes de los surfistas en cada *heat* y las clasificaciones finales de cada ronda. Esto permitió comprender cómo se desarrolló la competencia simulada y quiénes fueron los surfistas destacados en cada etapa.

2. **Comparación con Datos Reales:** Se compararon los resultados simulados con los datos reales de la competencia histórica para determinar su precisión y validez. Esto implicó analizar si los surfistas que se destacaron en la simulación coincidieron con los surfistas que obtuvieron buenos resultados en la competencia real.

3. **Evaluación de la Efectividad:** Se evaluó la efectividad de la simulación considerando diversos factores, como la precisión en la predicción de los resultados, la coherencia con los datos históricos y la capacidad para identificar tendencias y patrones en el rendimiento de los surfistas.

4. **Validación de la Metodología:** Se examinó la robustez y la validez de la metodología utilizada en la simulación, incluyendo la construcción de los modelos KDE, la generación de muestras simuladas y la simulación de la competencia en sí. Se consideraron aspectos como la calidad de los datos utilizados, la adecuación de los modelos y la consistencia en los resultados obtenidos.

5. **Identificación de Mejoras Potenciales:** Se identificaron posibles áreas de mejora en la metodología de simulación y en la selección de parámetros, con el objetivo de perfeccionar el proceso y aumentar su precisión en futuras simulaciones.

El análisis de los resultados y la evaluación de la simulación de la competencia implicaron comparar los resultados simulados con los datos reales, evaluar la efectividad de la metodología utilizada y validar la capacidad de la simulación para predecir resultados precisos y coherentes con la realidad histórica. Este proceso proporcionó información valiosa sobre la utilidad y la fiabilidad de la simulación en la predicción de eventos deportivos como las competencias de surf.

Análisis de la Predicción del Top 8 de Surfistas

La predicción del top 8 de surfistas en las Olimpiadas la abordamos mediante la simulación de competencias utilizando modelos de Kernel Density Estimation (KDE). Este enfoque se basa en datos históricos de rendimiento de los surfistas, que se utilizan para construir modelos de KDE.

La simulación se inicia con la primera ronda, donde los surfistas compiten en heats de 3. Para cada heat, se generan puntuaciones simuladas utilizando los modelos KDE, y los surfistas con mejores puntuaciones avanzan a la siguiente ronda. En las rondas siguientes, los heats se reducen a 2 surfistas cada uno, y el proceso de simulación se repite hasta llegar a la final. Cada ronda refina la lista de surfistas que avanzan, acercándonos al top 8 final.

Después de cada ronda, los emparejamientos y los modelos KDE se actualizan para reflejar a los surfistas que avanzan. Esto incluye reorganizar los heats y re-evaluar las probabilidades de rendimiento. Al finalizar todas las rondas, se genera un ranking final de los surfistas basado en sus desempeños simulados. Este ranking determina el top 8, reflejando quiénes son los surfistas con más probabilidades de destacar en la competencia.

El proceso es robusto y flexible, permitiendo la integración de múltiples simulaciones para capturar la variabilidad en los resultados. Este método no solo es aplicable a competencias de surf, sino que también puede adaptarse a otras disciplinas deportivas donde el rendimiento pasado puede usarse para prever futuros resultados.

En el análisis predictivo para las Olimpiadas de surf, John John Florence emerge como el principal favorito para asegurar el primer lugar. Las simulaciones basadas en

modelos de Kernel Density Estimation (KDE) destacan a John John Florence debido a su reciente desempeño excepcional en Tahití, el mismo lugar donde se llevarán a cabo las Olimpiadas. Como subcampeón de la última competencia realizada en Tahití, John John Florence ha demostrado su capacidad para sobresalir en las olas desafiantes y únicas de Teahupo'o. Esta ventaja es crucial ya que la familiaridad y el éxito en este entorno específico son determinantes. Además, John John Florence no solo es destacado por su reciente rendimiento local, sino que también ostenta el título de número uno en el ranking global. Esta posición refleja su consistencia y superioridad a lo largo de diversas competiciones y condiciones en el circuito mundial de surf. La combinación de su habilidad demostrada en Tahití y su liderazgo global se refleja en las simulaciones, donde su probabilidad de quedar en primer lugar supera a la de sus competidores. Por estos motivos, John John Florence es el surfista con más probabilidades de ganar el oro en las Olimpiadas.

Conclusiones

En conclusión, el proyecto ha demostrado la viabilidad y utilidad de utilizar técnicas avanzadas de modelado estadístico y simulación para analizar y predecir el desempeño en competiciones de surf. Se recomienda continuar refinando los modelos y procesos de simulación utilizando datos adicionales y técnicas de modelado más avanzadas.