Modelo predictivo para un partido de temporada regular 2020-2021 de la NBA

Universidad Nacional de Colombia, Facultad de Ciencias,
Departamento de Física

Carlos Salas, Jaime Cocunubo, Santiago Rodriguez y Jonatan Gonzalez

20 de julio de 2021

Resumen

Se plantea como problema para el proyecto implementar la información suministrada por las fluctuaciones en las casas de apuesta al modelo mecánico-estadístico basado en el movimiento browniano de un partido de NBA que plantea Stern (1994), adecuándolo para obtener un pronóstico más informado del resultado de un partido de la temporada 2020-2021. Para ello se estudia la diferencia de puntos en un intervalo de tiempo de 48 minutos de juego (duración de un partido sin considerar un posible overtime o tiempo extra) permitiendo estimar la probabilidad de que un equipo gane dada la ventaja de puntos que tuvo en un instante del partido. Se decidió implementar la información de las fluctuaciones en las casas de apuesta en el parámetro de arrastre o drift del modelo por la forma en la que se plantea analíticamente la probabilidad, el parámetro se definió dependiente del tiempo y se relacionó con los handicap de las casas de apuesta que se encuentran en la página Nowgoal3. La página Kaggle será utilizada para recolectar los datos del partido necesarios, mientras que para su respectivo tratamiento (limpieza y categorización) se hará uso del entorno de programación Python, que además permite normalizar el tiempo de juego $t \in [0, 1]$. Adicionalmente, se precisó determinar el parámetro correspondiente a la desviación estándar del modelo con los primeros 11 partidos de la temporada del equipo Los Ángeles Lakers, donde se utiliza el método de Newton-Rhapson. Se escogieron los siguientes 4 partidos del equipo para aplicar el modelo y dar pronósticos del equipo ganador, de forma que se puedan comparar y analizar con los resultados reales, y así determinar la fiabilidad del modelo.

1. Introducción

El baloncesto es uno de los deportes con mayor cantidad de seguidores en el mundo, lo cual hace que las ganancias monetarias en este sean de gran impacto para el mercado, así como lo menciona Álvares en la columna [32]. Con esto en mente, se han desarrollado varios modelos con el objetivo de predecir el comportamiento de los equipos durante la temporada y los resultados de cada enfrentamiento con el fin de así lograr un acercamiento al resultado final. Es una de las herramientas más apropiadas, debido al comportamiento aleatorio de las variables, el empleo de la mecánica estadística y los distintos modelos que pueden ser aplicados a partir de la misma, como Guerra permite ver en su artículo [3]. Durante la investigación de los distintos métodos ya desarrollados se consiguen tres principales los cuales se adecuan según el objetivo y datos tratados para el abordamiento del problema.

Es usual encontrar en la literatura, ej.:[1], trabajos que buscan modelar el desarrollo de partidos profesionales mediante distribuciones estadísticas. Sin embargo, estos se suelen ver limitados a la hora de incluir parámetros, que en el caso del baloncesto suelen ser determinantes, entre estos se incluyen lesiones, factores externos como motivación, desgasto físico, agresividad de juego, entre otros. Todas estas variables resultan difíciles de cuantificar e incluir en los modelos, sin embargo, mediante el estudio de

las fluctuaciones en las cuotas de las casas de apuestas, es posible cuantificarlos. Adicionalmente, los modelos usuales tratan de predecir el número total de puntos al final del partido, dado que se utilizan distribuciones que sirven para conteo pero que no diferencian entre eventos para los dos equipos. En ese orden de ideas, se propone utilizar procesos que permitan predecir el resultado final el juego (Gana Local/Visitante) a partir de variables dinámicas que se dan durante los primeros cuartos del juego.

En el primero de ellos se muestra el empleo de diferentes distribuciones de probabilidad con el fin de comprender de una manera global el partido, es decir, se considera exclusivamente el número total de puntos y las variables que determina la dinámica del juego; al hacerlo de esta forma se modela cada cuarto del juego adecuando las distribuciones que mejor se ajustan a los datos recolectados. Los trabajos previos muestran un buen resultado empleando las distribuciones de Poisson, Gamma y Ley de Potencias, sin embargo, no resultan útiles a la hora de predecir el vencedor del partido, problema que se logra observar en el trabajo de Martín-González [2].

El segundo método está basado en los procesos de Markov, con los cuales se puede analizar el juego en lapsos discretos de tiempo por medio de una matriz estocástica que modela los distintos estados en los que se puede encontrar el sistema y cómo eso afecta la evolución temporal de las variables empleadas para la descripción. Con este método se podría solucionar el problema abierto que ha dejado Vracar en su artículo [10], dado que el artículo analiza el juego de una manera global en cuanto eventos (puntos marcados durante) y por medio de este método es posible predecir cual de los equipos será el ganador, tal como lo expone Shirley lo deja claro en [6].

En el tercer método se aprovecha la naturaleza aleatoria de las variables, concepto que podemos ampliar a la luz del libro de la profesora Blanco [5], para el modelo de movimiento browniano o procesos de Weiser con "drift". En ese orden de ideas se relaciona la probabilidad de que el equipo local gane el partido como función de la diferencia entre los puntajes para ambos equipos y el tiempo transcurrido, mediante el uso de la función de distribución acumulada para una distribución normal. Cabe aclarar que para el modelo planteado en [4] el parámetro "drift" se fija como una ventaja para el equipo local, de manera que se propone convertirlo en un factor dinámico relacionado con las cuotas de las casas de apuestas. Dado el objeto del curso, mecánica estadística, se optará por este método.

Así entonces se plantea como objetivo modelar una serie de partidos a partir del sistema sobre el movimiento Browniano con el fin de lograr una buena predicción para los enfrentamientos y su desarrollo en el tiempo, es decir, se busca conocer con mayor grado de certeza los resultados finales del partido de antemano, este tipo de abordamientos son ampliamente desarrollados y expuestos en contexto en los artículos de Guerra y Martín [4, 3].

Es entonces que se plantean pasos a seguir con mira de lograr el ya mencionado objetivo: se mostrará como se procede a realizar la recolecta de datos de los partidos jugados durante la temporada 2020-2021 de la NBA empleando la página Kaggle, dado que se desarrollará este modelo para un equipo en específico (los Lakers de Los Ángeles) entonces los datos relacionados a los mismos serán los elegidos y depurados a la información que se ha considerado como relevante. Como parte del modelo se ha normalizado el tiempo (es decir, la duración del partido se contiene en un dominio determinado entre 0 y 1), se tomarán 15 partidos en total,11 como entrenamiento del modelo y 4 para probarlo en ejecución. Empleando la interpretación de la ecuación de difusión de Einstein sobre el modelo Browniano expuesta en [24], contextualizado en el problema, se buscará la predicción del partido definiendo los parámetros de arrastre μ dependiente del tiempo a partir de la información probabilística obtenida de los datos de las casas de apuestas. Al tener este desarrollo se ejecutarán modelos computacionales que se determinaron analíticamente y finalmente se contrastará el modelo computacional vs el analítico, empleando como unidad de comparación lo acertado de los resultados para predecir el partido.

2. Marco teórico

Para el planteamiento de este proyecto se considerarán *modelos estáticos*, es decir que las variables del sistema no dependerán del intervalo de tiempo de manera local, en pequeños dt; el uso de esta clase de

modelos reducen las dimensiones y complejidad del sistema a analizar, con la metodología adecuada, se logra emplearlos para sistemas dinámicos como en este caso. Se tiene también varias formas de acercarse a modelos predictivos en un ámbito deportivo como por ejemplo: el empleo del movimiento browniano [4], un ajuste por medio de distribuciones de probabilidad tales como la Ley de Potencias [2] y modelos mas heurísticos dados por la colectividad.

Es importante aclarar la distinción entre un partido de baloncesto cerrado y abierto, se considera un partido cerrado donde la diferencia entre puntaje de cada equipo es menor a 11 y abierto cuando dicha diferencia es mayor o igual a 11. En el presente proyecto se seguirá la metodología de considerar los puntos como una *función de densidad de probabilidad* buscando la *distribución* que de mejor manera se ajuste al sistema en determinados lapsos discretos de tiempo y así predecir la evolución del partido.

Es por este mismo motivo que será de vital importancia mencionar que al tratar las variables del sistema de manera aleatoria y continua se está conviniendo que la misma es una variable real definida sobre un espacio de probabilidad en el cual se satisface que existe una función real no negativa e integrable p_x tal que para para todo x en los reales se satisface

$$P_x(x) = \int_{-\infty}^{x} p_x(t)dt \tag{1}$$

donde p_x será la función de densidad de la variable aleatoria x [5], la cual por hipótesis estará definida a trozos. Algunas distribuciones que suelen ser utilizadas para modelar partidos de diferentes deportes competitivos son leyes de Potencias y Normal.

- **Distribución Ley de potencias:** Esta es una relación matemática muy simple que permite relacionar variables de la forma
$$p(x) = Cx^p \tag{2}$$
en donde C y p son constantes reales propias del sistema; esta distribución es importante dado que es invariante de escala, es decir, al multiplicar por una constante la variable del sistema, $p(x)$ simplemente se ve afectada por una constante [5].

- **Distribución Normal:** Para finalizar esta sección de distribuciones se mencionará la distribución normal dada su importancia en el modelo browniano que se trabajará más adelante. Así como con las anteriores se mencionarán sus aspectos más importantes. Se dice que una variable aleatoria X tiene distribución normal de parámetros μ y δ, donde μ es un número real y δ es también un real, pero exclusivamente positivo; su función de densidad (fdd) viene dada por:
$$\Phi(x) = \frac{1}{\delta\sqrt{2\pi}} exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\delta}\right)^2\right], \text{ para x en los reales} \tag{3}$$
en donde se puede identificar a los factores μ como un factor de localización y a δ como uno de escala [5].

Para el análisis de este problema se han empleado distintas herramientas, como ya se mencionaba; una de ellas es el uso de las distribuciones de probabilidad mostradas durante intervalos de tiempo dados de forma periódica, este análisis resulta muy útil cuando el objetivo es analizar de manera global el comportamiento de los eventos, que para este caso sería la cantidad absoluta de puntos, es decir: hechos tanto por el equipo A como el B. Cuando se busca determinar cual de los dos equipos será el ganador, los modelos más útiles serán los de *Markov*[6] y el relacionado al *Movimiento browniano*[4], es por eso que ahora se procederá a dar un marco de referencia al respecto.

El primero de ellos son *Los Procesos de Markov*, al buscar generar un modelo de este estilo se consideran todas las variables necesarias en el desarrollo del partido las cuales puedan determinar el éxito o fracaso de un equipo en la cancha. En particular cuando se habla de un modelo de Markov se hace referencia a un problema en el cual es verificado cuando la probabilidad de un evento se encuentra únicamente relacionada con el evento inmediatamente anterior [7]. En [6] se hace referencia a tres factores que determina el proceso:

1. Posesión del balón, local o invitado.

2. Cómo el equipo ha ganado la posesión del balón.

3. La cantidad de puntos que han sido conseguidos por medio de la anterior posesión.

Y al tener la composición de un estado caracterizado por cada uno de estos 3 factores se conocerá la información básica del partido en ese determinado lapso de tiempo.

Se puede también expresar un proceso de Markov como una serie de experimentos en los que cada uno tiene m posibles resultados y la probabilidad obtenida se encuentra en dependencia exclusiva de los resultados de los experimentos previos. Este proceso se encuentra expresado por una matriz de transición o estocástica, la cual satisface que:

1. $a_{ij} \geq 0$

2. $\sum_j a_{ij} = 1$, para cada i fijo

estos procesos son así conocidos ya que satisfacen la propiedad de Markov, en la cual se establece que en un sistema dinámico dependiente del tiempo las probabilidades de un evento son independientes de sus pasados valores, en otras palabras "las variables aleatorias no tienen memoria", matemáticamente la podemos expresar como [8]:

$$P[X_{n+1} = x_{n+1}|X_0 = x_0, X_1 = x_1, ..., X_n = x_n] = P[X_{n+1} = x_{n+1}|X_n = x_n] \tag{4}$$

Anteriormente se habló de los procesos de Markov y su importancia en la modelación de sistemas dinámicos, y lo cierto es que el modelo del movimiento browniano es un caso particular para procesos de Markov de tiempo y espacio de estados continuos, es decir, procesos de Markov homogéneos en el tiempo.

El movimiento browniano se evidenció por primera vez en el laboratorio del botanista británico Robert Brown, el cual observó que los granos de polen suspendidos en una gota de agua seguían trayectorias aparentemente caóticas. En 1905 Albert Einstein logró explicar analíticamente este fenómeno empírico, sin embargo, la gran mayoría de literatura que se encuentra al respecto habla sobre los procesos del movimiento browniano, conocidos también como de Weiner, el cual en 1923 fundó las bases matemáticas de este proceso estocástico [21].

La contribución de Einstein al movimiento Browniano fue determinar que la densidad de probabilidad de que una partícula browniana, que se mueve de manera aleatoria en un medio, se encuentre en la posición x en un tiempo t obedece una ecuación de difusión [24]. La ecuación de difusión, una ecuación diferencial parcial, está dada por [23]:

$$\frac{\partial \rho}{\partial t} = a^2 \nabla^2 \rho \tag{5}$$

Es una ecuación con solución analítica, luego, partiendo de una solución del tipo exponencial como $\psi(x,t) = e^{(\alpha x)} \cdot e^{(\alpha t)}$ se separan las variables y se obtiene [23]:

$$\rho(x,t) = e^{i\omega x} \cdot e^{-\omega^2 a^2 t} \tag{6}$$

Aplicando la identidad de Euler:

$$\rho_{(x,t)} = [Cos(\omega x) + iSen(\omega x)] \cdot e^{-\omega^2 a^2 t} \tag{7}$$

Considerando las diferentes soluciones en función de las condiciones de frontera:

$$\rho_{(x,t)} = [A_{(\omega)} Cos(\omega x) + B_{(\omega)} Sen(\omega x)] \cdot e^{-\omega^2 a^2 t} \tag{8}$$

Teniendo en cuenta que se cuenta con n frecuencias (ω) factibles, entonces se deben considerar de forma que finalmente se halla [23]:

$$\rho_{(x,t)} = \int [A_{(\omega)} Cos(\omega x) + B_{(\omega)} Sen(\omega x)] \cdot e^{-\omega^2 a^2 t} d\omega \tag{9}$$

Hasta ahora se ha mostrado la solución general para la ecuación de difusión bidimencional. El coeficiente a se denomina el coeficiente de difusión el cual evidencia la habilidad de un soluto en disolverse en un solvente, en otras palabras, es la capacidad de extenderse sobre un medio. A partir de esta solución, considerando que en el tiempo $t = 0$ la partícula se encuentra en $x = 0$, Einstein derivó la siguiente solución al problema [24]:

$$\rho_{(x,t)} = \frac{1}{(4\pi Dt)^{3/2}} e^{-\frac{|x|^2}{4Dt}} \tag{10}$$

con D el coeficiente de difusión.

Solo como anotación, es interesante observar que una partícula libre la cual es descrita por la ecuación de Schrödinger se reduce a una ecuación de difución con constante $a = \left(\frac{i\hbar}{2\mu}\right)$ en donde μ representa la masa reducida de la partícula y de resto son constantes, es decir, a meramente depende de la masa de la partícula, interesante.

Para poder aplicarlo al problema de interés será necesario satisfacer ciertas hipótesis [21]. Por un lado, un proceso estocástico estándar de movimiento browniano debe satisfacer que, dado el proceso aleatorio $\mathbf{X} = \mathbf{X_t} : \mathbf{t} \in [\mathbf{0}, \infty)$ [21]:

1. $P(X_0 = 0) = 1$.
2. $\mathbf{X}$ tiene incrementos estacionarios. Es decir, la distribución del movimiento en un intervalo temporal, depende de la longitud del intervalo.
3. $\mathbf{X}$ tiene incrementos independientes. Es decir, las variables aleatorias son independientes.
4. X_t tiene una distribución normal con media 0 y varianza t para cada $t \in [0, \infty)$.
5. X_t debe ser continua con probabilidad 1.

Es claro que este modelo es útil en muchas situaciones, sin embargo para el caso a estudiar es necesario añadir un factor de arrastre o "drift" μ. Recordando que el movimiento browniano estándar se describe por una distribución normal, este drift se puede interpretar como un desplazamiento en el valor promedio de un proceso aleatorio. Ahora, un cambio evidente para el modelo de movimiento browniano con un drift se puede ver en la condición d para el estándar, de modo que se transforma en [21]:

- X_t tiene una distribución normal con media μt y varianza $\sigma^2 t$ para cada $t \in [0, \infty)$.

Donde el parámetro drift puede tomar valores reales $\mu \in \mathbf{R}$ y la varianza $\sigma \in [0, \infty)$. Además, estos parámetros se escogen como funciones lineales de t, dado que ahora $\mathbf{X}$ tiene incrementos independientes y estacionarios del promedio y la varianza.

Otro modo de entender este proceso con drift μ y varianza σ^2 fijos, es dando un proceso de movimiento browniano estándar $\mathbf{B} = B_t : t \in [0, \infty)$, de modo que el nuevo proceso con drift será [21]:

$$\mathbf{X}(t) = \mu t + \sigma \mathbf{B}(t) \tag{11}$$

Si ahora se normaliza el tiempo que dura el partido de baloncesto de modo que $t \in (0, 1)$, es posible definir un proceso $X(t)$ de modo que represente la diferencia de puntos entre el equipo local y el visitante, y que por ende que pueda tomar valores negativos, positivos, o el valor nulo [4]. Donde es claro que $X(1) > 0$ indica que el local ganó el partido, y $X(1) < 0$ indica lo contrario. Ahora, si se asume que este proceso se puede describir mediante un modelo de movimiento browniano con drift, se puede asociar el drift por unidad de tiempo μ a una ventaja o desventaja del equipo local sobre el visitante. Para el modelo presentado por Hal S. Stern (1994), el drift μ tenía un mismo valor para todos los encuentros estudiados, en el cual se le daba una ventaja arbitraria al local [4]. Dado que en la actualidad se tiene acceso a los datos en tiempo real o "play-by-play data", una propuesta interesante consiste en calcular el parámetro

de drift como función de las cuotas en las casas de apuestas para el mercado "Gana Local/Visitante", dado que estas varían dinámicamente como se explicará más adelante.

Luego, continuando con la idea de describir el proceso $X(t)$ como un movimiento browniano con drift, se sabe que se podrá asociar con una distribución normal $X(t) \backsim N(\mu t, \sigma^2 t)$, y de la propiedad **3** de los incrementos independientes, se tendrá que [4]:

$$X(s) - X(t) \backsim N(\mu(s-t), \sigma^2(s-t))$$

Donde $X(s) - X(t)$ para $s > t$ será independiente de $X(t)$. Ahora, recordando que la función de densidad de probabilidad de una distribución normal viene dada por (6), es posible ver que en general para una distribución normal se tendrá la función de distribución acumulada (FDA) Φ dada como la integral desde $-\infty$ hasta x para la ecuación (6), tal como se procede en la ecuación (1). Obteniendo la FDA normal [5]:

$$\Phi(X, \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{x} \exp\left[-\frac{1}{2\sigma^2}(t-\mu)^2\right] dxt \tag{12}$$

Luego, definiendo la tasa "drift rate" como el cociente μ/σ, la probabilidad de que el equipo local gane el juego vendrá dada como:

$$P(X(1) \ > \ 0) = \Phi(\mu/\sigma) \tag{13}$$

Adicionalmente, recordando que el modelo browniano es un proceso de Markov homogéneo en el tiempo, se tendrá que su transición de probabilidad vendrá dada como [21]:

$$p_t(x, y) = f_t(y - x) = \frac{1}{\sqrt{2\pi t}} \exp\left[-\frac{(y-x)^2}{2t}\right] \tag{14}$$

De modo que es extensible para el caso de un proceso de movimiento browniano con drift dado por [21]:

$$f_t(x) = \frac{1}{\sigma\sqrt{2\pi t}} \exp\left[-\frac{1}{2\sigma^2 t}(x - \mu t)^2\right] \tag{15}$$

De forma que es fácilmente demostrado a partir de la condición **4**. Ahora, de la ecuación (13) se puede hallar una equivalencia, ya que para el caso general (un t y un $l(t)$ dados) donde l es la diferencia de puntos entre el equipo local y el visitante, la probabilidad vendrá dada por una probabilidad condicional, es decir, la probabilidad de que local gane, dado que existe cierta diferencia de puntos en un tiempo t. Luego, es posible llegar a la relación [4]:

$$P_{\mu,\sigma} = P(X(1) > 0 | X(t) = l) \tag{16}$$

$$= P(X(1) - X(t) > -l) \tag{17}$$

$$= \Phi\left(\frac{l + (1-t)\mu}{\sqrt{(1-t)\sigma^2}}\right) \tag{18}$$

De manera que cuando $t \to 1$ para $l \neq 0$, entonces la probabilidad debe ser 0 o 1. Ahora, tratando $X(t)$ como una variable aleatoria continua que en cierto modo se aproxima a su valor entero más grande, se arregla el problema respecto de la necesidad de variables continuas. Respecto al modelo presentado por Stern [4], se introduce otro cambio significativo, dado que cuando hay un empate al final del partido, es decir $X(1) = 0$, el modelo considera que las probabilidades de que gane el local son de 0.5. En este modelo se plantea usar una función del drift, la varianza y tendencias anteriores del partido.

Para el problema a abordar se sabe que la variable dependiente puede tomar dos valores binarios [25], es decir:

$$Si\ X(1) > 0 \Longrightarrow Y = 1$$
$$Si\ X(1) < 0 \Longrightarrow Y = 0$$

Donde $Y = 1$ representa que el local ganó el partido, porque la diferencia de puntos en el tiempo final $t = 1$ fue mayor a cero.

Una de las técnicas de regresión más usadas para casos en los que la variable dependiente toma valores binarios, se conoce como el método probit [26]. Para que este modelo sea aplicable, es necesario también que la probabilidad condicional de uno de los dos resultados sea combinación lineal de los parámetros (variables independientes) transformadas mediante la FDA normal[26], es decir:

$$P(Y = 1|X) = \Phi(X^T\beta) \tag{19}$$
$$P(Y = 0|X) = 1 - \Phi(X^T\beta) \tag{20}$$

Donde X^T es el traspuesto del vector de variables independientes, y β un vector de coeficientes a ajustar a la regresión.

Bajo un modelo de variable latente es posible relacionar una variable observable (si gana local) a una variable latente (sea Y^*) aleatoria[25]. De forma que la variable aleatoria se puede escribir como una combinación lineal de las variables independientes, como:

$$Y_i^* = X_i\beta_i + \epsilon_i \tag{21}$$

Donde ϵ_i representa un vector que contiene los errores para una distribución normal, es decir, $\epsilon \sim N(0, 1)$. Así, la variable observable se puede relacionar con la latente como:

$$Y_i = \begin{cases} 1 & si\ Y_i^* \geq 0 \\ 0 & si\ Y_i^* < 0 \end{cases} \tag{22}$$

Ahora, existen distintos métodos para hallar los parámetros β, sin embargo Stern (1994) decidió usar el método de verosimilitud, y por ello se introducirá a continuación [4].

El método de máxima verosimilitud (o MLE por sus siglas en inglés) permite usar una muestra ξ, para hallar los parámetros de la distribución de probabilidad (en este caso la normal). Las primeras condiciones que aparecen para usar el método, se imponen sobre la muestra[27].

- Sea la muestra ξ, la cual se le atribuye a un vector de variables aleatorias Ξ cuya función de distribución es desconocida.
- Se asume la existencia de un espacio de parámetros $\Theta \subseteq R^p$ en el cual viven los vectores Ξ.
  - si Ξ es continuo, se asume que su función de densidad de probabilidad adjunta $f_\Xi(\xi; \theta_0)$ es función del parámetro θ para ξ fijos. Entonces, a esta se le llama verosimilitud:

$$L(\theta; \xi) = f_\Xi(\xi; \theta)$$

- Se necesita estimar el parámetro θ_0 asociado a la función de distribución desconocida.

Teniendo en cuenta que la idea del método de máxima verosimilitud es hallar el vector de parámetros en el espacio de parámetros más probable a haber generado la muestra, se define ahora el estimador de mayor verosimilitud $\hat{\theta}$ de θ, con el cual se tiene un parámetro que maximiza la verosimilitud de la muestra ξ [29].

$$\hat{\theta} = arg \max_{\theta \subseteq \Theta} L(\theta; \xi) \tag{23}$$

Ahora, para usar el método en el caso de una regresión probit con una distribución normal para la muestra, se tendrá la verosimilitud para una observación (y_i, x_i) [30]:

$$L(\beta; y_i, x_i) = [\Phi(x_i\beta)]^{y_i}[1 - \Phi(x_i\beta)]^{1-y_i} \tag{24}$$

Donde y_i representa el resultado de la observación (si local ganó o perdió) y x_i la diferencia de puntos. Acá es también claro que cuando:

- $y_i = 1$, entonces $L(\beta; y_i, x_i) = \Phi(x - i\beta)$
- $y_i = 0$, entonces $L(\beta; y_i, x_i) = 1 - \Phi(x - i\beta)$

Ahora, si las observaciones (eventos y resultados) se consideran independientes y con la misma distribución de probabilidad, es posible escribir la verosimilitud como una productoria, de modo que se tendrá:

$$L(\beta; y, X) = \prod_{i=1}^{N} \left(\Phi(x_i\beta))^{y_i}[1 - \Phi(x_i\beta)]^{1-y_i}\right) \tag{25}$$

Muchas veces resulta más fácil trabajar con el logaritmo de la verosimilitud, debido a que se pasa de una productoria a una sumatoria, usando la notación $log(L) \equiv l$ (revisar anexo 14.5.2) [30]:

$$l(\beta; y, X) = \sum_{i=1}^{N} [y_i \ln(\Phi(x_i\beta)) + (1 - y_i) \ln(1 - \Phi(x_i\beta))] \tag{26}$$

Recordando la intención de la ecuación (23) se quiere hallar el vector θ para el cual se maximizan la verosimilitud de la muestra, y esto se logra mediante la maximización del vector de parámetro β. Por esta razón se define el gradiente del logaritmo de la verosimilitud 14.5.3:

$$\nabla_\beta l(\beta; y, X) = \sum_{i=1}^{N} \frac{f(x_i\beta)}{\Phi(x_i\beta)[1 - \Phi(x_i\beta)]}[y_i - \Phi(x_i\beta)]x_i \tag{27}$$

Donde se usó el hecho de que para el caso de una distribución normal, la derivada de la función de distribución acumulada es la función de distribución de probabilidad:

$$\frac{dF(t)}{dt} = f(t)$$

Introduciendo la notación para un nuevo vector de variables dependientes como [30]:

$$q_i = 2y_i - 1 \tag{28}$$

De forma que:

$$q_i = \begin{cases} 1 & si\ y_i = 1 \\ -1 & si\ y_i = 0 \end{cases} \tag{29}$$

Luego, en términos de esta nueva variable se pueden reescribir la verosimilitud, su logaritmo y su gradiente respectivamente como (Idea en el anexo 14.5.4):

$$L(\beta; y, X) = \prod_{i=1}^{N} (\Phi(x_i\beta q_i)) \tag{30}$$

$$l(\beta; y, X) = \sum_{i=1}^{N} ln(\Phi(x_i\beta q_i)) \tag{31}$$

$$\nabla_\beta l(\beta; y, X) = \sum_{i=1}^{N} \frac{f(x_i\beta q_i)q_i x_i}{\Phi(x_i\beta q_i)} = \sum_{i=1}^{N} \lambda_i x_i \tag{32}$$

Una vez introducida esta nueva notación es posible escribir la matriz hessiana, es decir, la matriz de las segundas derivadas, como (véase anexo 14.5.5):

$$\nabla_{\beta\beta} l(\beta; y, X) = -\sum_{i=1}^{N} \lambda_i (x_i \beta + \lambda_i) x_i^T x_i \tag{33}$$

Recordando la ecuación (23), es posible obtener el estimador solucionando el problema de maximización:

$$\hat{\beta} = arg \max_{\beta} l(\beta; y, X)$$

Luego, siempre que exista un máximo global, el gradiente del logaritmo de la verosimilitud debe satisfacer la condición:

$$\nabla_{\beta} l(\beta; y, X) = 0 \tag{34}$$

Que usando la ecuación (27) se convierte en:

$$\sum_{i=1}^{N} \frac{f(x_i\beta)}{\Phi(x_i\beta)[1 - \Phi(x_i\beta)]} [y_i - \Phi(x_i\beta)] x_i = 0 \tag{35}$$

O equivalentemente por facilidad se puede usar la ecuación (32), obteniendo:

$$\sum_{i=1}^{N} \lambda_i x_i = 0$$

Por lo general no suele haber solución analítica a la condición de los ceros en la primera derivada, sin embargo, es posible implementar métodos numéricos que permitan hallar raíces de ecuaciones no lineales. A continuación se muestra como implementar el método de Newton-Raphson [31].

Para implementar el método de Newton-Raphson es necesario que la función sea derivable en el intervalo de interés, además se requiere conocer un valor inicial para la raíz (llámese x_0), de forma que este estimado se va mejorando recursivamente. Así, al escribir la serie de Taylor alrededor del cero tentativo, se tiene:

$$f(x) = f(x_O) + f'(x_0)(x - x_0) + 1/2 f''x_0)(x - x_0)^2 + \cdots = 0$$

Luego, al truncar la serie en el término lineal (bajo el supuesto de que el cero estimado se encuentra cerca al verdadero), se obtiene la formula iterativa:

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$$

Lo que es como la tangente en el punto x_0 que se extrapola en el eje x para dar hallar una raíz más aproximada. Si se realiza esta iteración i veces, se tiene un cero en su forma general[31]:

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)} \tag{36}$$

El método reescrito con la notación usada hasta el momento se convierte en:

$$\hat{\beta}_j = \hat{\beta}_{t-1} - \frac{\nabla_{\beta} l(\hat{\beta}_{t-1}; y, X)}{\nabla_{\beta\beta} l(\hat{\beta}_{t-1}; y, X)} \tag{37}$$

Así, teniendo en cuenta que la probabilidad dada por la ecuación (16) cuya forma se halló mediante el modelo de movimiento browniano [4], con las variables transformadas $l/(\sqrt{1-t})$ y $\sqrt{1-t}$ con coeficientes $1/\sigma$ y μ/σ. Es posible hallar la verosimilitud dada por la ecuación (25).

Entonces tomando los tres primeros cuartos de distintos partido por aparte, se llega a que como son independientes se puede escribir la verosimilitud como dos productorias, una sobre el partido de la muestra, y otra sobre el cuarto correspondiente, de forma que se escribe como:

$$L = \prod_{i=1}^{n_{partidos}} \prod_{j=1}^{3} \Phi\left(\frac{X_{ij} + (1-\frac{j}{4})\mu_{ij}}{\sqrt{(1-\frac{j}{4})\,\sigma^2}})\right)^{Y_i} \times \left(1 - \Phi\left(\frac{X_{ij} + (1-\frac{j}{4})\mu_{ij}}{\sqrt{(1-\frac{j}{4})\,\sigma^2}})\right)\right)^{(1-Y_i)} \tag{38}$$

De forma que el parámetro de la varianza pueda ser calculado por el método de máxima verosimilitud y el método de Newton-Raphson. Es claro que para el caso en el que el parámetro de arrastre no lleve índices (es decir, que no sea un parámetro dinámico del partido o el cuarto a considerar) este se puede hallar por el método anterior usando el gradiente con las derivadas respecto a la media y la varianza, sin embargo al hacerlo dinámico se tendría que introducir manualmente en la ecuación (38) y en la del cálculo de la probabilidad.

En la sección 12.5.2 de desarrollo de objetivos específicos se muestran los desarrollos analíticos para llegar a una expresión explícita como en (37).

Con la intención de modificar el parámetro de drift, es posible usar las cuotas de las casas de apuestas, y transformarlas en probabilidades aplicables a los modelos. Para ello existen dos métodos que se utilizan actualmente, el de normalización básica, y el modelo de Shin. Como mostró Erik Štrumbelj en 2014 [11], el modelo de Shin es útil para reducir el error generado por cada casa de apuesta realizando un promedio estadístico entre distintas casas. En sus resultados mostró que para partidos importantes de la NBA, la casa de apuestas online Betfair resultaba ser la más acertada en cuanto a su algoritmo de balanceo de cuotas. Por ende, se proponen usar las cuotas de esta casa, y transformarlas mediante una normalización básica, de modo que $O = (o_1, o_2, ..., o_n)$ sea un vector con las probabilidades derivadas de las cuotas de la casa de apuestas π_i. Entonces, para el caso de interés se tienen 3 posibles resultados, victoria Local, victoria Visitante, o Empate, de modo que las cuotas se transforman a probabilidades como $o_i = \frac{1}{\pi_i}$, de este modo, se pueden normalizar las probabilidades como:

$$\beta = \sum_{i=1}^{3} \pi_i \tag{39}$$

De modo que las nuevas probabilidades serán $P_i = \frac{\pi_i}{\beta}$. La razón para usar estas cuotas es que las casas las ajustan (y por ende las probabilidades) para ciertos resultados en vivo, en función de la cantidad de personas que le apuestan, debido a que estas pueden tener información oculta como posibles lesiones de algún jugador, cambios en la moral del equipo, etc.

Finalmente, la precisión de estas predicciones se pueden conocer mediante el puntaje de Brier [11], el cual se define como:

$$(\pi_i, \mathbf{a}) = \frac{1}{n}|(\pi_i - \mathbf{a})|^2 \tag{40}$$

Donde π_i representan las probabilidades estimadas, y $\mathbf{a}$ es el vector que contiene el resultado verdadero.

3. Estado del Arte

Es bien conocido que desde la antigüedad los deportes han hecho parte de la sociedad en su modalidad de competición. De los primeros registros que se encuentran son unos artefactos para éstos deportes en la

antigua China alrededor del 1066 y 771 a.C [33]. Registros también los rastrean hasta Egípto y Persia, es en la antigua Grecia dónde se empieza a consolidar una competición multidisciplinaria donde su ganador sería premiado; esto provoca un gran crecimiento y avance en el juego, dado que la ambición de lograr escalonar en la sociedad y el honor de la victoria se volvieron cardinales para los participantes y todos aquellos que alrededor de la actividad disfrutaban del evento [15]. Siguiendo lo habitual de la sociedad, luego de conseguir lo básico de la supervivencia ésta requiere ocio, el cual o bien era la participación activa de los eventos como competidores, o, como lo era para la gran mayoría, apostar y dejar que el azar aumentase la adrenalina del juego [16]. Con el desarrollo de la probabilidad como una rama aplicada de la matemática respecto al azar, se empiezan a desarrollar modelos para poder conocer o predecir los resultados y poder enmarcar los eventos que se pudieran llegar a dar en el juego, cuantos de ellos eran favorables para poder determinar la certeza de ganar o no [5].

De modelos probabilísticos y su influencia en el análisis de sistemas físicos como los fluidos o termodinámica, nace el primer acercamiento a su aplicación: la teórica cinética. Dicha teoría, bajo grandes supuestos, empieza a obtener resultados acordes a los resultados experimentales basados en las leyes de la termodinámica. Con el tiempo, y acreditado a Ludwig Boltzmann, se logra formalizar la matemática de las leyes aplicando la probabilidad y su rigurosidad hasta entonces forjada [18].

Por toda esta evolución que ha surgido desde la competitividad, el deporte, la matemática y la física se han desarrollado modelos para poder comprender cada aspecto involucrado. Ya que para concretar los parámetros y variables adecuadas que rijan el modelo es necesario tener en cuenta las reglas y formas del juego, se han elaborado distintas investigaciones acerca de los deportes más famosos como el baseball o tenis de mesa [4]. Este trabajo en particular se enfoca en el baloncesto dada la gran cantidad de información que se maneja durante un partido y las estadísticas que diferentes casas de apuestas mantienen en libre acceso para consulta.

Es importante resaltar que modelos de un partido en esta disciplina han sido ampliamente estudiados desde diferentes puntos de vista. En el artículo [3] de 2014 se estudia la evolución de la anotación y los intervalos de anotación en los partidos de NBA correspondientes a las temporadas entre 2005 y 2010. Por otro lado, en el artículo [2] de 2016 se presenta un modelo a partir de una distribución de Poisson para el número de canastas en un intervalo de tiempo, donde se trata la anotación de puntos como un proceso aleatorio. La tesis doctoral [12] de 2016 expone una red bayesiana para modelar el progreso del total de puntos en un partido de NBA, de forma que determinan la probabilidad de que el total de puntos del partido exceda el establecido por un apostador. Con el mismo enfoque del artículo anterior, en el 2020 se publica el artículo [1], donde se realiza un modelo para la cantidad de puntos en un partido basado en procesos gamma. Un modelo basado en un proceso de Markov se presenta en el artículo [10], donde se busca modelar la progresión de un partido usando la información jugada a jugada de cada equipo. Adicionalmente, en el artículo [4] se estudia la diferencia de puntos en un partido a través de un modelo basado en el movimiento browniano.

Aunque como se expuso anteriormente, un partido de baloncesto como objeto de estudio ha sido trabajado en diferentes investigaciones, también existen modelos de partidos deportivos competitivos diferentes que contribuyen al desarrollo del mismo problema. En el 2017 se publica el artículo [13], donde se presenta un modelo de conteo para las anotaciones de un partido de football usando la distribución de Weibull. En el artículo [10], mencionado anteriormente, se expone la posibilidad de aplicar el modelo basado en el movimiento browniano a otros deportes, en particular se aplica a un partido de baseball. Por otra parte, la tesis [14] de 2014 realiza un modelo de un partido de baseball a partir de cadenas de Markov para partidos de la MLB (Major League of Baseball) de 2013.

Esto ha permitido el estudio y análisis de posibles deficiencias en los modelos o nuevas propuestas para procurar un resultado más acertado. Con esta información como punto de partida se puede concluir que un modelo mecánico-estadístico más acertado según los resultados experimentales sería el modelo basado en el movimiento browniano, tal como se propone en el artículo [4] donde se aborda el problema de predecir el resultado de un partido de NBA en temporada regular.

4. Planteamiento del problema

El proyecto propuesto plantea el estudio de un partido de baloncesto de NBA (National Basketball Association) como sistema físico compuesto de dos sistemas interactuantes, los cuales corresponden a cada equipo. Un análisis de un partido de NBA a partir del proceso gamma se presenta en la referencia [1], donde se trata el problema de modelar el proceso de la cantidad total de puntos al final del partido. Por otro lado, la referencia [2] expone un estudio del número de canastas en un intervalo de tiempo, donde la mayoría de canastas siguen una distribución de Poisson pero en el último minuto de partidos cerrados son distribuidas siguiendo una Ley de Potencias. Considerando el primer artículo se resalta que el modelo permite dar un pronóstico de la cantidad de puntos durante un partido sin diferenciar que equipo anota cada punto. También se destaca que el modelo del segundo artículo considera cualquier tipo de anotación (1 punto, 2 puntos o 3 puntos) como una canasta en un intervalo de tiempo.

Se presenta como problema la aplicación y el ajuste de un modelo mecánico-estadístico que permita pronosticar el equipo ganador durante un partido de NBA implementando la información de las fluctuaciones en las cuotas de las casas de apuestas, en contraste con el artículo [1] que modela la cantidad total de puntos del partido y el artículo [2] que estudia el número de canastas en un intervalo del partido. Como primera tentativa se considera el modelo de Poisson y de Ley de Potencias, sin embargo, este no tiene en cuenta el tipo de canasta ni el equipo que la anota, por lo que no se propone como un modelo efectivo para la realización del problema planteado. La referencia [1] plantea la posibilidad de aplicar el proceso gamma a los procesos de anotación local como visitante de forma separada, no obstante, en busca de abarcar otros modelos para el desarrollo del problema se utilizará un modelo basado en un proceso de movimiento browniano.

Tomando como partida en el artículo [4], se modelará la diferencia entre el puntaje de cada equipo en un partido de baloncesto como una variable aleatoria $X(t)$ en un intervalo de tiempo transformado a unidad $t \in (0, 1)$, donde se tratará $X(t)$ como una variable continua. Asumiendo que se puede modelar la $X(t)$ como un proceso de movimiento de Browniano, tomando $X(t)$ como una distribución normal de forma que la probabilidad de que el equipo local gane $P(X(1) > 0)$ estará dada por la función de distribución correspondiente, es decir, $P(X(1) > 0) = \Phi(\mu/\sigma)$. Específicamente, usando el modelo dado por la caminata aleatoria del movimiento browniano, la probabilidad de que el equipo local gane dado que hay una diferencia de puntos m al haberse jugado una fracción del partido está dada por [4]:

$$P_{\mu,\sigma}(m, t) = \Phi\left(\frac{m + (1-t)\mu}{\sqrt{(1-t)\sigma^2}}\right) \tag{41}$$

La probabilidad de que el equipo contrario gane se obtendrá calculando el complemento de la probabilidad ya encontrada.

La ecuación (41) se puede interpretar como un modelo probit relacionando el resultado del juego con las variables transformadas $m/\sqrt{1-t}$ y $\sqrt{1-t}$ tal como es propuesto en el artículo [4], donde se asume que las observaciones generadas para cada cuarto del partido son independientes. Con este análisis se espera poder calcular el parámetro σ de la distribución haciendo una estimación por máxima verosimilitud, es decir, calculando los máximos de la función de verosimilitud L dada por (38). A partir de esta expresión se calculará la derivada analíticamente con el fin de igualar la derivada a cero y así encontrar los puntos críticos de la función. Para calcular los ceros de la derivada de la función de verosimilitud se utilizará el método de Newton-Rhapson con un programa en Mathematica. El método de Newton-Rhapson halla las raíces de una función por medio de iteraciones que calculan la tangente de la función en un punto x_0 y la extrapola para encontrar su intersección con el eje x para obtener un nuevo punto x_1 [31].

Una parte fundamental del proyecto respecto al aspecto original que se integra es la implementación de la información proveniente de las casas de apuestas a partir del desarrollo de la ecuación de difusión para la teoría del movimiento browniano. Como se vio en el marco teórico, la densidad de probabilidad

de que una partícula browniana se encuentre en una posición X en un instante t, está dada por una función gaussiana que se interpreta como una distribución normal, donde el coeficiente de difusión está relacionado con la desviación estándar. El parámetro drift o de arrastre (μ) de la distribución permite correr la función gaussiana a lo largo del eje x para un tiempo t fijo, es decir, con este parámetro se puede ajustar en que valores de x es más probable que se encuentre la partícula browniana en el instante t. De esta manera, este parámetro μ en el modelo permite tener en cuenta muchos factores que se presentan de manera especial en un partido, así se propone escoger un parámetro dependiente del tiempo teniendo en cuenta las fluctuaciones de las cuotas en las casas de apuesta, pues factores que afectan la probabilidad de que un equipo gane como la lesión de un jugador o los problemas de faltas se verán reflejados en las apuestas en tiempo real y así como otras novedades.

La forma en la que se hará la conversión de la información de las casas de apuestas a el parámetro μ del modelo será inicialmente por tanteo. Para ello se debe tomar en cuenta la forma en la que viene la información que se tiene disponible, se tomarán los "handicap" disponibles en la página de Nowgoal3.com. Los handicap dan información en tiempo real de la probabilidad de que un equipo gane el partido en forma de una diferencia de puntos, por ejemplo, si se lesiona uno de los jugadores principales de un equipo y en ese instante están perdiendo el partido tendrán un handicap negativo distinto de la diferencia de puntos real. Teniendo esto en cuenta, se puede utilizar el valor constante de μ en el artículo [4] para plantear tentativamente la siguiente ecuación:

$$\mu(t) = 0.885 * h(t) \tag{42}$$

donde h(t) corresponde al handicap para el equipo local en un instante de tiempo t.

El ajuste que se plantea para el parámetro σ se implementará a 11 partidos de la NBA de Los Ángeles Lakers para la temporada de 2020-2021 (considerados partidos de entrenamiento para el modelo), más específicamente los primeros 11 partidos que jugó esa temporada. Se escogen a Los Ángeles Lakers como el equipo de estudio pues fue el equipo campeón de la temporada anterior, lo cual indica que fueron uno de los equipos que tuvo un desempeño más consistente, además de que para el momento en que se jugaron los partidos de estudio no se hizo ningún cambio de jugador ni se dio alguna lesión. El modelo como tal será aplicado a los 4 partidos que se jugaron enseguida de los 11 anteriores para procurar que los datos con los que se ajusta el parámetro σ sean los más reciente posibles. Por otro lado, la recolección de datos se hará usando la página Kaggle que permite descargar toda la información requerida de los partidos jugada a jugada de la temporada, en particular se utilizará una base de datos que contiene los partidos que se jugaron en el primer mes de la temporada de estudio. La información de las casas de apuesta se recolectará a través de la página nowgoal3.com, donde se tomarán los datos del "handicap" del equipo local que se encuentra disponible para los partidos de estudio jugada a jugada pues ésta presenta la información en términos de una diferencia de puntos.

Para el manejo, la limpieza y la categorización de los mismos se utilizará el lenguaje de programación Python y la librería Pandas, pues estas permiten tomar la base de datos bajada de Kaggle y descartar toda la información que no sea de interés, es decir, mantener solo la información correspondiente a los partidos de interés del tiempo de juego, de cuál equipo fue local y de la diferencia de puntos. Adicionalmente, se aprovecha para tomar la información del tiempo de juego y se normaliza, esto es, que el comienzo del partido corresponda al tiempo $t = 0$ y que el final del partido corresponda al tiempo $t = 1$. Por otro lado, el lenguaje de programación Python permitirá implementar directamente el programa correspondiente al método de Newton-Rhapson realizado en Mathematica que se plantea para determinar σ, también permitirá el uso de la función con la que se planteo la probabilidad de que un equipo gane dada la diferencia de puntos en un instante del partido resolviendo numéricamente la integral en la función de distribución acumulativa que se muestra en la ecuación (41).

Es importante aclarar que en un principio se planeaba hacer el ajuste del parámetro σ y la aplicación del modelo a los partidos correspondientes a la temporada 2019-2020 que los Lakers jugaron contra los 3 equipos que terminaron en las posiciones más cercanas a los mismos. Se tuvieron que cambiar los partidos de estudio por los presentados anteriormente debido a que la información de las casas de apuestas en vivo solo estaban disponibles para la temporada 2020-2021. Por otro lado, no se tomarán equipos en particular

como contrincantes de los Lakers pues la base de datos disponible en la página Kaggle solo contaba con información del primer mes de juego de la temporada 2020-2021, por lo tanto para poder tomar varios partidos del mismo equipo no se podían restringir los partidos a algunos encuentros particulares.

Por último, se realizarán pronósticos con el modelo de los 4 partidos especificados para diferentes instantes de tiempo correspondientes al comienzo del partido, el final del primer cuarto, el final del segundo cuarto y el final del tercer cuarto. Lo anterior se hace con el fin de poder analizar de mejor manera como cambian los resultados del modelo al aplicarlo en diferentes instantes del partido. Ahora bien, los pronósticos como tal se compararán con los resultados reales de cada partido calculando los respectivos puntajes de Brier dados por la ecuación (40), de forma que se pueda hacer un análisis de que tan acertado fue el modelo y tener una mejor idea de la fidelidad del mismo.

5. Motivación y justificación

El creciente acceso a bases de datos confiables y fáciles de manejar, se ha visto reflejado en el interés de la comunidad científica por crear modelos que describan el comportamiento de diversos sistemas mediante ajustes basados en estos datos. De acuerdo a Mordor Intelligence (2021) [22], el mercado del análisis deportivo fue valorado en 2015 por 83.6 millones USD, y creció en el 2020 a un valor de 1.05 billones USD. Siendo un mercado liderado por empresas como IBM, SAP SE, y Oracle Corporation, la implementación de modelos predictivos en los deportes se ha visto escalada mayormente a sistemas complejos adaptativos, impactando sistemas naturales (el sistema inmune, ecosistemas, sociedades) y sistemas artificiales (inteligencia artificial, redes neuronales, sistemas de computo distribuidos y paralelos)[19, 20].

La motivación para incluir un parámetro de arrastre dinámico al trabajo realizado por Hal S. Stern (1994)[4], surgió del trabajo de F.S Abril y C. J. Quimbay [9], donde se introdujo un parámetro de arrastre estocástico para una serie de tiempo no estacionaria, de modo que en este caso se variase temporalmente la media de la distribución normal. En ese orden de ideas, al implementar el parámetro de arrastre mediante la fluctuación de las cuotas en las casas de apuestas, y al entrenar el modelo con datos de partidos anteriores, se espera obtener resultados distintos para cada equipo. Con esta solución se espera poder extender el modelo a otros sistemas y áreas, donde también es posible cuantificar de algún modo variables complejas mediante ciertos indicadores o tendencias.

Un ejemplo para cuantificar y clasificar variables que pueden ser complejas recaen en el método One Hot Encoding, el cual crea una columna para cada valor distinto que exista en la característica que se está codificando, si se toma una fila específica tendrá 0 en las columnas que no cumplan la condición y 1 en el caso contrario. Un ejemplo de lo anterior podría ser codificar el sexo de las personas, donde para un hombre se le asigna un 1 a la columna “hombre” y un 0 a la columna “mujer” y lo contrario cuando se trate de una mujer [17].

6. Objetivo general

Ajustar el modelo mecánico-estadístico de un partido de NBA basado en el movimiento Browniano planteado en el artículo [4], implementando la información suministrada por la fluctuación de las cuotas en las casas de apuesta en un dado momento del partido, para obtener un pronóstico más informado del resultado de un partido al aplicarlo en la temporada 2020-2021 de la NBA.

7. Objetivos específicos

1. Recolectar los datos de los partidos de la temporada 2020-2021 de la NBA usando la página Kaggle.
2. Limpiar y categorizar los datos obtenidos de acuerdo al equipo elegido (Lakers) empleando el entorno Python y la librería Pandas.

3. Normalizar el tiempo del juego de cada partido.

4. Escoger 11 partidos de entrenamiento y 4 de prueba para el modelo desarrollado.

5. Hacer uso de la ecuación de difusión desde la interpretación dada por Einstein del movimiento Browniano, para definir los parámetros del modelo asociado al problema específico de la predicción del resultado de un partido.

6. Determinar el parámetro de arrastre μ dependiente del tiempo a partir de la probabilidad brindada por las casas de apuestas.

7. Implementar computacionalmente las distribuciones y funciones desarrolladas de manera analítica para calcular la diferencia de puntos de todos los partidos elegidos.

8. Determinar el parámetro correspondiente a la desviación estándar mediante el método computacional de Newton-Raphson con los partidos de entrenamiento.

9. Ejecutar el modelo con los partidos de prueba y dar un pronóstico de equipo ganador.

10. Comparar y analizar las estimaciones realizadas con los resultados reales para determinar la fiabilidad del modelo.

8. Metodología

1. Para la obtención de los datos se hace uso de la página Kaggle, la cual es una comunidad de ciencia de datos con una amplia variedad de áreas de conocimiento, en dicha página, específicamente en el link: `https://www.kaggle.com/schmadam97/nba-playbyplay-data-20182019`, se realiza la descarga de los datos correspondientes a la temporada 2020-21 en formato de extensión csv.

2. Una vez cargados los datos se procede a usar la librería Pandas, la cual permite agrupar los datos en estructuras de dos dimensiones llamadas DataFrames donde se proceden a eliminar los valores repetidos mediante funciones predeterminadas con el fin de limpiar la información a usar. Posteriormente, para categorizar, se seleccionan los partidos donde jugaron los Lakers (LAL), donde se obtuvo un total de 15 partidos en donde los Lakers fueron locales 8 veces y visitantes 7 veces.

3. Dado a que en los datos descargados cada cuarto de juego empieza en 720 segundos y termina en 0, se procede a realizar la conversión temporal donde se tiene que todo el partido comprende 2880 segundos y se busca ordenar el tiempo de manera continua, tal que el primer cuarto empiece en el segundo 0 y el partido finalice en el segundo 2880 para posteriormente dividirlo entre 2880 con el fin de que el intervalo de tiempo esté entre 0 y 1.

4. Para la elección de los partidos de prueba, se visualizan las fechas de los 15 partidos y se toman los últimos cuatro partidos de acuerdo a la fecha cuando se desarrollaron, teniendo que estos partidos corresponden al 12, 13, 15 y 18 de enero del año 2021.

5. En primer lugar se considera la diferencia de puntos entre el equipo local y el equipo visitante en función del tiempo como un proceso estocástico estándar del movimiento browniano. Luego, introduciendo un "drift" o factor de arrastre en el modelo, se llega a que la probabilidad de que un equipo gane un partido como local (dada por la distribución normal), es dependiente del tiempo, la diferencia de puntos $l(t)$, y el arrastre $\mu(t)$.

   Teniendo en cuenta que la variable dependiente es de tipo binario (local gana o local pierde), y que la probabilidad de que local gane dados ciertos parámetros corresponde a la función de distribución acumulada normal. Es posible usar la regresión probit para ajustar la desviación estándar al modelo a partir de los resultados conocidos para los 11 partidos de entrenamiento. Así, se propone hallar la desviación estándar mediante la estimación de máxima verosimilitud.

6. Inicialmente para hallar la probabilidad de las casas de apuestas, se toman de la página `http://www.nowgoal3.com/basketball/1x2-362685` las tasas de apuesta (Handicap) para cada partido en específico en distintos instantes de tiempo. Con dicho valor se halla el parámetro μ mediante la ecuación (42), haciendo que el parámetro μ sea guardado en un vector que dependa del tiempo y partido a usar debido a que la probabilidad de las casas de apuestas varía con el tiempo y el partido.

7. Antes de crear la función que permita calcular la probabilidad de que gane el equipo local se mapean los puntos de cada equipo en cada partido y se calcula la diferencia de puntos dependiente del tiempo y del partido, donde se propone almacenar dicha información en una matriz para después llamar elemento a elemento de esta.
   Con lo anterior realizado y el parámetro de arrastre definido, se define una función en Python que dependa de μ, σ, la diferencia de puntos de juego, el tiempo y tenga la forma de la ecuación (41), lo cual corresponde a la función de distribución acumulativa de una distribución normal estándar que es bien conocida y encontrada en algunas de las referencias de este proyecto. Dicha función será de vital importancia ya que permitirá calcular la probabilidad de que gane el equipo local.

8. Al tener las diferencias de puntos X_{ij}, el equipo ganador Y_i y el parámetro de arrastre μ para los partidos de entrenamiento, se procede a aplicar el método computacional de Newton-Raphson, en donde a partir de un error, máximo de iteraciones e intervalo a trabajar definidos se halla la raíz de la ecuación de interés, determinando entonces la desviación estándar o parámetro σ.

9. Una vez con todas las variables y parámetros determinados se aplica la ecuación (41) con los datos de los equipos de prueba para obtener una gráfica de probabilidad en función del tiempo y observar cual es la probabilidad de que gane el equipo local en cada partido cuando este ha finalizado, es decir cuando t=1.

10. Una vez obtenidos los pronósticos del equipo ganador para cada partido de prueba, se procede a ver que tan alta o no es dicha probabilidad en cada caso y ver que tanto varía con respecto al resultado real para así decidir si las predicciones realizadas fueron aceptables o no.

Nota: Además del notebook adjunto, se puede encontrar el código de Python en: `https://github.com/sarodriguezme/Proyecto-M.Estadistica` específicamente el archivo *Proyecto_NBA.ipynb*, es posible que en algunas ocasiones no cargue a la primera vez pero volviendo a cargar la página debería aparecer. También se puede utilizar el siguiente código QR:

9. Resultados Esperados

Con el desarrollo del proyecto se esperan lograr los siguientes resultados:

1. Con la obtención de datos de la página Kaggle se obtienen conocimientos sobre una comunidad de ciencia de datos, donde a demás de poder descargar archivos también es posible crear una cuenta, interactuar con usuarios y los proyectos que estos realicen tanto para complementar información y conceptos como para ayudar en caso de que ese usuario lo pida o presente errores en sus desarrollos. Adicionalmente hay convocatorias para participar en torneos, los cuales consisten en dejar tratar un problema abierto con unos datos en específico.

2. Para el desarrollo del proyecto es imprescindible usar un entorno que permita el manejo de datos, donde se escogió el entorno Python y la librería Pandas. Manejando dicho entorno se busca comprender el concepto de DataFrame y como a partir de funciones y comandos propios de Pandas es posible visualizar, limpiar y manejar a gusto los datos necesarios para el proyecto.

3. Cuando se normaliza el tiempo se busca entender que es un proceso necesario para la correcta ejecución de los métodos del proyecto, principalmente dado a la naturaleza de la función de distribución normal y los parámetros que caracterizan esta.

4. Entender el concepto de sobre ajuste, si se entrena un modelo con unos datos determinados y se prueba solamente en dichos datos, el modelo va a acostumbrarse a estos datos, hará buenas predicciones pero será ineficaz a la hora de usarlo con datos nuevos. Por esto se busca escoger partidos de entrenamiento y prueba en lugar de usar todos los partidos.

5. Con el planteamiento analítico se espera generalizar el modelo añadiendo un parámetro dinámico de arrastre, lo que en últimas genera un cambio en la forma funcional del gradiente y la matriz Hessiana de la verosimilitud. Además, al implementar la regresión probit y el método de mayor verosimilitud se aprende una técnica que puede ser usada en campos como Machine Learning y Redes Neuronales.

6. Al implementar el parámetro de arrastre μ como función del tiempo y de cada juego mediante las cuotas en las casas de apuestas, se espera poder realizar un mapeo efectivo, de forma que las predicciones del modelo se ajusten mejor a los resultados reales. Parte de ese trabajo se debe hacer por tanteo o ensayo y error (casi como introducir una función de proporcionalidad experimentalmente), cosa que históricamente se ha hecho muchas veces en la física.

7. El hecho de recorrer un archivo para calcular la diferencia de puntos en distintos tiempos para varios partidos como también implementar un función computacional dependiente de ciertos parámetros que haga el mismo papel de una distribución normal representa un reto que requiere destrezas en el uso de librerías numéricas como Numpy y en aplicar la correcta sintaxis para que el programa reproduzca lo que se busca.

8. Se espera obtener una desviación estándar estática que se ajuste a los partidos de entrenamiento. Un posible valor esperado es $sigma = 15.82$, obtenido por [4] para 493 partidos de la NBA en el año 1992. Este objetivo permite también adquirir conocimientos en la aplicación de métodos numéricos para hallar raíces de ecuaciones no lineales.

9. De la descripción para la dinámica de la probabilidad de tener cierta diferencia de puntos entre local y visitante $P_l(t)$, se espera poder ver una correlación entre la probabilidad y el valor para el parámetro de arrastre, de forma que haya una corrección al modelo más simple. En el desarrollo de esta sección se fortalecerán habilidades de visualización de resultados.

10. De ser posible, se espera comparar la discrepancia en los resultados del método sin arrastre y los del método con arrastre, respecto a los resultados reales de algunos partidos. Se espera que la discrepancia para los resultados del método sin arrastre sea menor, demostrando una mejora con el nuevo método. Acá se espera aprender herramientas para medir la efectividad de un modelo a un problema con resultados binarios (También muy útil en problemas de clasificación en Machine Learning y regresión logística).

10. Cronograma

Objetivo a cumplir	Tiempo estimado
Recolectar los datos de los partidos de la temporada 2020-21 de la NBA usando la página Kaggle.	Realizado
Limpiar y categorizar los datos obtenidos de acuerdo al equipo elegido (Lakers) empleando el entorno Python y la librería Pandas.	Realizado
Normalizar el tiempo del juego de cada partido.	Realizado
Escoger 11 partidos de entrenamiento y 4 de prueba para el modelo desarrollado.	Realizado
Hacer uso de la ecuación de difusión desde la interpretación dada por Einstein del movimiento Browniano, para definir los parámetros del modelo asociado al problema específico de la predicción del resultado de un partido.	Realizado
Definir el parámetro de arrastre μ dependiente del tiempo a partir de la probabilidad de las casas de apuestas.	Realizado
Implementar computacionalmente las distribuciones y funciones desarrolladas analíticamente para calcular la diferencia de puntos de todos los partidos elegidos.	22/07/2021
Determinar el parámetro correspondiente a la desviación estándar mediante el método computacional de Newton Raphson con los partidos de entrenamiento.	23/07/2021
Ejecutar el modelo con los partidos de prueba y dar un pronóstico de equipo ganador.	25/07/2021
Comparar y analizar las estimaciones realizadas con los resultados reales para determinar la fiabilidad de cada modelo.	30/07/2021

11. Recursos disponibles

- La página Kaggle, `https://www.kaggle.com/schmadam97/nba-playbyplay-data-20182019` donde se obtienen los datos de los partidos correspondientes a la temporada 2020-201, pero si el lector lo requiere puede descargar datos desde la temporada del año 2015 hasta el año 2021.
- Computadores y el lenguaje de programación Python para procesar datos en entornos de trabajo como Google Colab y notebooks de Jupyter, para este último se usó la aplicación Anaconda Navigator (anaconda 3).
- Librerias de Python como numpy (`https://numpy.org/doc/stable/reference/index.html`) y pandas (`https://pandas.pydata.org/docs/reference/index.html#api`) fueron utilizadas así como los links adjuntos con el fin de facilitar, procesar y tratar la información de una manera práctica y precisa.
- La página nowgoal3, `http://www.nowgoal3.com/basketball/1x2-362685` donde se obtienen los datos de las tasas de las casas de apuestas dependientes del tiempo para los partidos a estudiar.
- GitHub, el cual es una página web donde es posible crear repositorios propios y hacer más sencillo controlar versiones, trabajar en grupo para un proyecto y poder compartirlo con más personas para facilitar su visualización sin necesidad de tener que descargar ningún código o archivo, se puede visualizar incluso desde el celular, puede probar con el código QR que está al final de la sección de metodología.
- Adicionalmente se cuenta como recurso disponible las referencias citadas.

12. Desarrollo de los objetivos específicos

De los objetivos específicos realizados hasta la fecha, se obtuvieron los siguientes resultados:

12.1.

La recolección de los datos para la temporada a estudiar se hizo descargando el archivo de extensión csv resaltado a continuación:

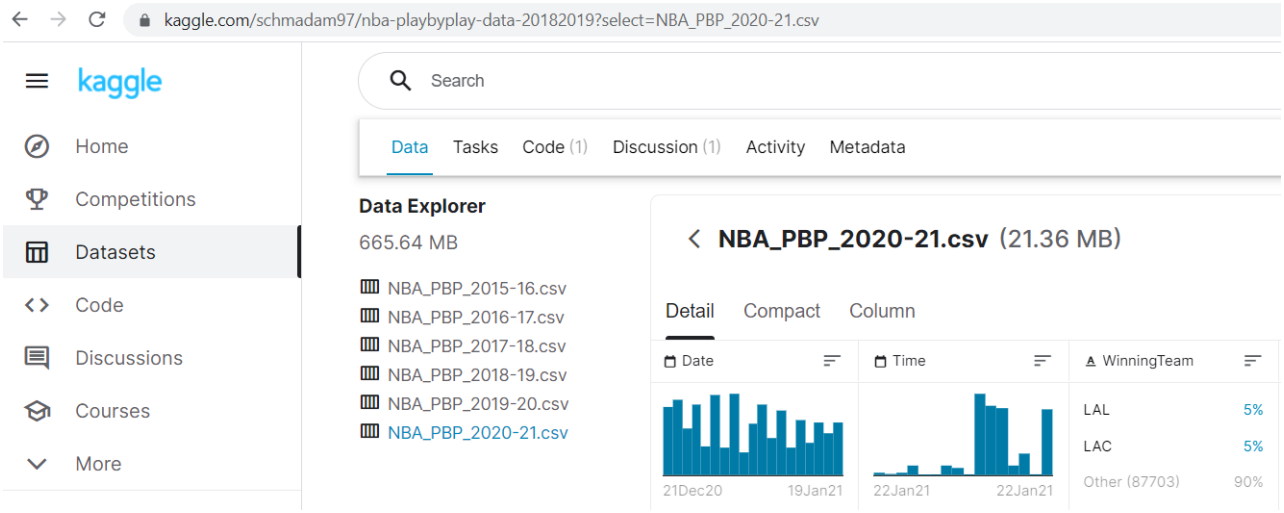

Figura 1: Entorno Kaggle donde se obtuvieron los datos de estudio.

Donde además es posible darle una mirada a los datos antes de descargarlos y ver características como el rango entre las fechas, horas, puntos anotados, ganadores, etc.

12.2.

Para la limpieza de datos se realizó el proceso explicado en la metodología e implementando los códigos del anexo 14.2 se obtuvieron los siguientes DataFrames:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 97564 entries, 0 to 97563
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   Date         97564 non-null  object
 1   WinningTeam  97564 non-null  object
 2   Quarter      97564 non-null  int64
 3   SecLeft      97564 non-null  int64
 4   AwayTeam     97564 non-null  object
 5   AwayPlay     49326 non-null  object
 6   AwayScore    97564 non-null  int64
 7   HomeTeam     97564 non-null  object
 8   HomePlay     48237 non-null  object
 9   HomeScore    97564 non-null  int64
 10  ShotType     37070 non-null  object
 11  ShotOutcome  37070 non-null  object
dtypes: int64(4), object(8)
memory usage: 8.9+ MB
```

Figura 2: DataFrame llamado df, el cual es producto de limpiar filas repetidas y filtrar las columnas de interés del DataFrame original descargado de Kaggle.

Del DataFrame obtenido se pasaron de tener 97673 filas y 41 columnas a 97564 filas y 12 columnas con respecto al DataFrame original descargado y ahora para seleccionar los partidos de los Lakers se realizan dos búsquedas, la primera que LAL sea equipo local (HomeTeam) y la otra que LAL sea equipo visitante (AwayTeam) y juntando los resultados de ambas búsquedas se obtiene:

```
df_LAL.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6927 entries, 0 to 6926
Data columns (total 14 columns):
 #   Column       Non-Null Count  Dtype 
---  ------       --------------  ----- 
 0   index        6927 non-null   int64 
 1   Date         6927 non-null   object
 2   WinningTeam  6927 non-null   object
 3   Quarter      6927 non-null   int64 
 4   SecLeft      6927 non-null   int64 
 5   AwayTeam     6927 non-null   object
 6   AwayPlay     3465 non-null   object
 7   AwayScore    6927 non-null   int64 
 8   HomeTeam     6927 non-null   object
 9   HomePlay     3462 non-null   object
 10  HomeScore    6927 non-null   int64 
 11  ShotType     2675 non-null   object
 12  ShotOutcome  2675 non-null   object
 13  W_P          6927 non-null   object
dtypes: int64(5), object(9)
memory usage: 757.8+ KB
```

Figura 3: DataFrame llamado df_LAL, el cual es producto de escoger los partidos donde jugó LAL.

Además de la reducción de filas debido a la limpieza, al filtrar los partidos de los Lakers se pasó de 97564 a 6927 filas obteniendo entonces una selección de datos adecuada y específica al problema de interés. Se menciona también que en las dos imágenes anteriores se especifica el nombre de cada columna y el tipo de objeto que maneja cada una de ellas donde 'int64' hace referencia a un numero entero y 'object' también conocido como 'string' hace referencia a una cadena de texto conformadas por letras y/o números.

12.3.

Al organizar de manera continua y normalizar el tiempo de juego de cada partido, la columna de segundos faltantes pasa a referirse a segundos transcurridos y los valores que toma son (anexo 14.3):

Renombrando la columna 'SecLeft' a 'Sec' (haciendo referencia a segundos transcurridos) y viendo los valores de segundos que hay:

```
df_game1Q=df_game1Q.rename(columns={'SecLeft':'Sec'})
```

```
df_game1Q['Sec'].unique()
```

```
array([0.        , 0.00625   , 0.01215278, 0.01701389, 0.02430556,
       0.02534722, 0.03229167, 0.03541667, 0.04236111, 0.04375   ,
       0.04652778, 0.05381944, 0.05555556, 0.05694444, 0.06423611,
       0.06493056, 0.06631944, 0.07256944, 0.07465278, 0.07673611,
       0.07986111, 0.08506944, 0.09131944, 0.09236111, 0.10034722,
       0.10104167, 0.10347222, 0.10555556, 0.11076389, 0.11145833,
       0.11215278, 0.11388889, 0.11527778, 0.12222222, 0.12326389,
       0.12569444, 0.12673611, 0.13159722, 0.13298611, 0.13333333,
       0.14131944, 0.14479167, 0.14618056, 0.14722222, 0.15208333,
       0.153125  , 0.15381944, 0.15659722, 0.16041667, 0.16354167,
       0.16423611, 0.16770833, 0.175     , 0.17673611, 0.178125  ,
       0.17986111, 0.18854167, 0.18923611, 0.19479167, 0.19861111,
       0.20625   , 0.20694444, 0.20763889, 0.21736111, 0.21805556,
       0.22013889, 0.22118056, 0.22534722, 0.22916667, 0.23541667,
       0.24444444, 0.24513889, 0.24583333, 0.25      , 0.25451389,
       0.25590278, 0.25833333, 0.25972222, 0.265625  , 0.26805556,
       0.26944444, 0.27465278, 0.27534722, 0.27673611, 0.28263889,
       0.28819444, 0.29270833, 0.29756944, 0.29826389, 0.29895833,
       0.29965278, 0.30104167, 0.30173611, 0.30694444, 0.3125    ,
       0.31493056, 0.31736111, 0.32118056, 0.32361111, 0.32951389,
       0.33506944, 0.33645833, 0.33784722, 0.33819444, 0.34201389,
       0.34305556, 0.34375   , 0.34861111, 0.35173611, 0.35347222,
       0.35520833, 0.35902778, 0.36215278, 0.36284722, 0.36493056,
       0.36666667, 0.36979167, 0.37430556, 0.37673611, 0.38194444,
       0.38229167, 0.38715278, 0.38958333, 0.39236111, 0.39340278,
       0.39756944, 0.39826389, 0.40347222, 0.40555556, 0.40902778,
       0.40972222, 0.41354167, 0.41527778, 0.41701389, 0.41979167,
       0.42118056, 0.42361111, 0.42430556, 0.425     , 0.42604167,
       0.42777778, 0.43229167, 0.440625  , 0.44166667, 0.44444444,
       0.45      , 0.45694444, 0.46180556, 0.4625    , 0.46354167,
       0.46736111, 0.47083333, 0.47222222, 0.47256944, 0.47986111,
       0.48576389, 0.48715278, 0.48819444, 0.48958333, 0.49826389,
       0.5       , 0.50590278, 0.51180556, 0.5125    , 0.51458333,
       0.52395833, 0.53125   , 0.53541667, 0.53888889, 0.54548611,
       0.54930556, 0.55451389, 0.55972222, 0.56423611, 0.56701389,
       0.56840278, 0.57048611, 0.57083333, 0.57847222, 0.57951389,
       0.58298611, 0.58402778, 0.58576389, 0.58645833, 0.5875    ,
       0.58819444, 0.59236111, 0.59340278, 0.596875  , 0.59756944,
       0.59930556, 0.60069444, 0.60486111, 0.60902778, 0.61076389,
       0.61423611, 0.615625  , 0.61631944, 0.61805556, 0.62395833,
       0.62638889, 0.62777778, 0.63125   , 0.63194444, 0.63263889,
       0.63333333, 0.634375  , 0.63888889, 0.64270833, 0.64618056,
       0.65625   , 0.65729167, 0.65833333, 0.66180556, 0.66631944,
       0.67013889, 0.67465278, 0.67604167, 0.67638889, 0.68020833,
       0.68159722, 0.68263889, 0.68958333, 0.69097222, 0.69270833,
       0.69444444, 0.69791667, 0.70104167, 0.70381944, 0.70798611,
       0.71458333, 0.71493056, 0.715625  , 0.71631944, 0.72326389,
       0.72465278, 0.72638889, 0.72708333, 0.72916667, 0.73680556,
       0.74027778, 0.74131944, 0.74895833, 0.74965278, 0.75      ,
       0.75625   , 0.76111111, 0.7625    , 0.765625  , 0.76770833,
       0.76944444, 0.76979167, 0.77673611, 0.78194444, 0.78298611,
       0.78472222, 0.78576389, 0.7875    , 0.79166667, 0.79791667,
       0.79861111, 0.80069444, 0.80868056, 0.81006944, 0.81354167,
       0.81875   , 0.82013889, 0.82638889, 0.83368056, 0.83993056,
       0.84131944, 0.84340278, 0.84756944, 0.85277778, 0.85659722,
       0.86006944, 0.86423611, 0.86493056, 0.871875  , 0.87326389,
       0.87743056, 0.878125  , 0.88194444, 0.88854167, 0.89652778,
       0.90243056, 0.90729167, 0.9125    , 0.91805556, 0.91944444,
       0.92083333, 0.92222222, 0.92708333, 0.93125   , 0.93576389,
       0.940625  , 0.94513889, 0.94583333, 0.94756944, 0.94826389,
       0.94895833, 0.95034722, 0.95555556, 0.95625   , 0.95833333,
       0.96215278, 0.96284722, 0.96458333, 0.96909722, 0.96979167,
       0.97465278, 0.975     , 0.98194444, 0.98298611, 0.98576389,
       0.98680556, 0.9875    , 0.990625  , 0.99895833, 1.        ])
```

Figura 4: Columna de segundos transcurridos del DataFrame llamado df_game1Q, el cual es una copia de df_LAL.

12.4.

Antes de escoger los partidos de entrenamiento y prueba del modelo es necesario saber cuantos partidos hay, qué equipos jugaron, la fecha y el ganador de cada partido, visualizando lo anterior (anexo 14.4):

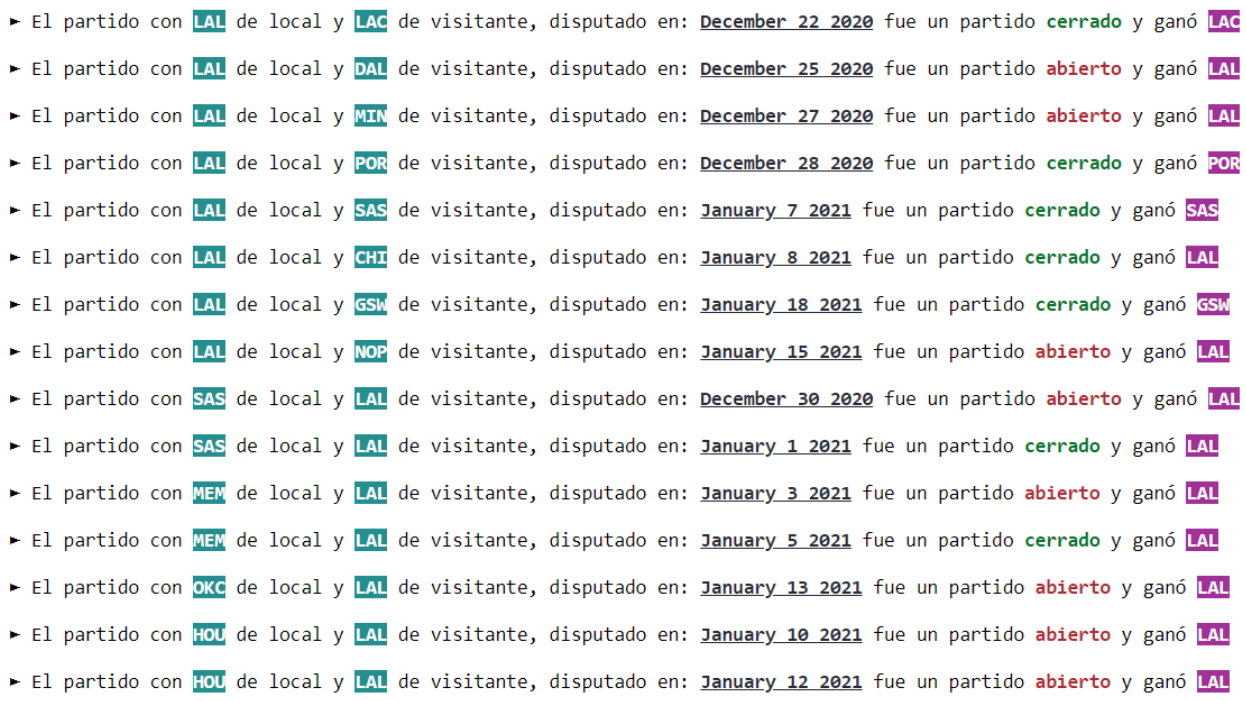

Figura 5: Fecha, tipo de partido, ganador y equipos que jugaron en cada partido de los Lakers en lo transcurrido de la temporada a estudiar.

Entonces de los 15 partidos disponibles, se usarán 11 para entrenar el modelo y hallar parámetros, por lo cual los 4 restantes serán para probar el modelo. El criterio de elección será por orden cronológico es decir que temporalmente los últimos 4 partidos serán los de prueba y corresponden a las fechas:

Local	Visitante	Fecha
Houston Rockets (HOU)	Los Angeles Lakers (LAL)	12/01/2021
Oklahoma City Thunder (OKC)	Los Angeles Lakers (LAL)	13/01/2021
Los Angeles Lakers (LAL)	New Orleans Pelicans (NOP)	15/01/2021
Los Angeles Lakers (LAL)	Golden State Warriors (GSW)	18/01/2021

Tabla 1: Partidos elegidos para probar el modelo.

12.5.

12.5.1.

En el marco teórico se dedujo la densidad de probabilidad de que una partícula browniana se encuentre en una posición x en un instante t desde la perspectiva de Einstein resolviendo la ecuación de difusión.La ecuación que describe esta densidad está dada por la ecuación (10):

$$\rho(x,t) = \frac{1}{(4\pi Dt)^{3/2}} e^{-\frac{|x|^2}{4Dt}}$$

Comparando con la expresión dada por (15) se evidencia que el parámetro correspondiente a la desviación estándar σ está relacionado con el coeficiente de difusión D, es decir que en un modelo basado en el movimiento browniano la habilidad que tienen las partículas de disolverse en un solvente determina que tan cerca están los valores a la media. Como se trata de una función gaussiana al introducir un parámetro μ (como en una distribución normal) de la siguiente forma

$$\rho(x,t) = \frac{1}{(4\pi Dt)^{3/2}} e^{-\frac{|x-\mu|^2}{4Dt}}$$

se corre la función gaussiana a lo largo del eje x para un tiempo t fijo tal como se muestra en la figura 6, es decir, con este parámetro se puede ajustar en que valores de x es más probable que se encuentre la partícula browniana en el instante t.

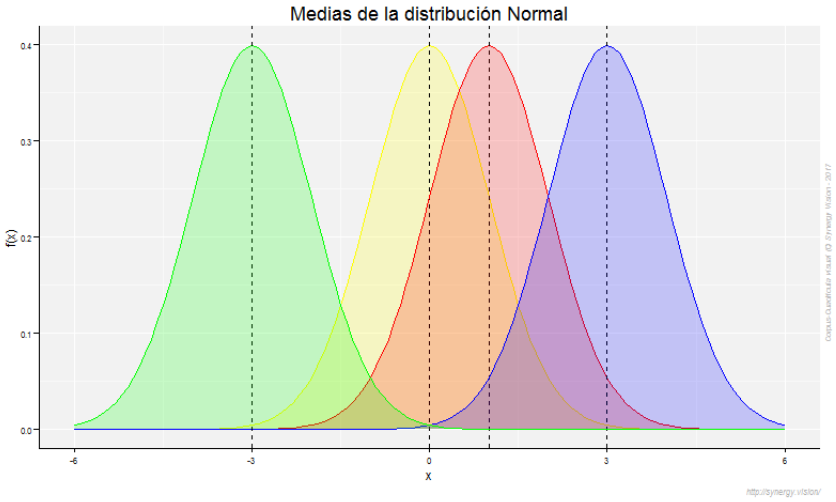

Figura 6: Función Gaussiana para diferentes valores del parámetro μ [34].

De esta manera, para el objeto de estudio al usar el modelo que se basa en el movimiento browniano el parámetro μ de la probabilidad de que el equipo local gane permitirá modificar o dar una "ventaja" a uno u otro equipo según se den las circunstancias en el partido. Es acá donde se introduce la información de las fluctuaciones de las casas de apuesta, donde situaciones como un jugador en problemas de faltas se reflejan en el corrimiento de la densidad de probabilidad.

Como se mencionó en el planteamiento del problema se hará un paso de la información de las casas de apuesta al parámetro μ haciendo uso del μ que se plantea en el artículo [4] donde se toma constante. Desde la página de Nowgoal3 se pueden encontrar los valores de handicap minuto a minuto para el equipo local jugada a jugada de los partidos de NBA (ver anexo 14.6). Los handicap son valores que permiten ver cuando las circunstancias del partido favorecen a un equipo a partir de una cantidad de puntos ficticia, es decir, al equipo local se le suman los puntos dados por el handicap a los puntos reales y con esto el equipo que tenga mayor cantidad de puntos sería el más probable que gane [35]. Teniendo esto en cuenta, se puede tantear una posible definición de $\mu(t)$ a partir del handicap $h(t)$ que está dada tentativamente por la ecuación (42):

$$\mu(t) = 0.885 * h(t)$$

12.5.2.

En primer lugar, partiendo de la ecuación (38) es necesario encontrar el logaritmo natural de la verosimilitud (procedimiento en el anexo Anexo 14.5), permitiendo encontrar:

$$l(\sigma; Y, X) = \sum_{i=1}^{n_{partidos}} \sum_{j=1}^{3} Y_i ln\left(\Phi\left(\frac{X_{ij} + (1-\frac{j}{4})\mu_{ij}}{\sqrt{(1-\frac{j}{4})\,\sigma^2}})\right)\right) \tag{43}$$
$$+(1-Y_i) ln\left(1-\Phi\left(\frac{X_{ij} + (1-\frac{j}{4})\mu_{ij}}{\sqrt{(1-\frac{j}{4})\,\sigma^2}})\right)\right)$$

Posteriormente, dado que la expresión para el logaritmo de la verosimilitud es prácticamente idéntica a la obtenida en la ecuación (27), es inmediato hallar el gradiente, que usando la siguiente notación (ecuación (44)) permite escribirlo más fácilmente:

$$a_{ij} \equiv \frac{X_{ij} + (1 - \frac{j}{4})\mu_{ij}}{\sqrt{(1 - \frac{j}{4})\,\sigma^2}} \tag{44}$$

$$\nabla_\sigma l(\sigma; Y, X) = \sum_{i=1}^{n_{partidos}} \sum_{j=1}^{3} \frac{1}{\sigma^2} \frac{\frac{X_{ij}+(1-\frac{j}{4})\mu_{ij}}{\sqrt{(1-\frac{j}{4})\sigma^2}} f\left(\frac{X_{ij}+(1-\frac{j}{4})\mu_{ij}}{\sqrt{(1-\frac{j}{4})\sigma^2}}\right)}{\Phi\left(\frac{X_{ij}+(1-\frac{j}{4})\mu_{ij}}{\sqrt{(1-\frac{j}{4})\sigma^2}}\right)\left[1 - \Phi\left(\frac{X_{ij}+(1-\frac{j}{4})\mu_{ij}}{\sqrt{(1-\frac{j}{4})\sigma^2}}\right)\right]} \left[\Phi\left(\frac{X_{ij} + (1 - \frac{j}{4})\mu_{ij}}{\sqrt{(1 - \frac{j}{4})\,\sigma^2}}\right) - y_i\right] \tag{45}$$

12.6.

Con las tasas de apuestas recolectadas de la página Nowgoal3 y la ecuación (42) se obtiene el parámetro de arrastre (μ), reportando dicho valor junto al partido, tipo de evento y demás características relacionadas como se ve a continuación (anexo 14.6):

	Local	Visitante	DifScore	Evento	Tiempo	TasaApuesta	Arrastre (u)
December 22 2020	LAL	LAC	0.0	Comienzo del partido	0.00	2.0	1.770
			-20.0	Fin del 1er cuarto	0.25	-11.5	-10.178
			-2.0	Fin del 2do cuarto	0.50	1.5	1.328
			-11.0	Fin del 3er cuarto	0.75	-7.5	-6.638
December 25 2020	LAL	DAL	0.0	Comienzo del partido	0.00	6.0	5.310
			3.0	Fin del 1er cuarto	0.25	6.5	5.752
			12.0	Fin del 2do cuarto	0.50	12.5	11.062
			14.0	Fin del 3er cuarto	0.75	11.5	10.178
December 27 2020	LAL	MIN	0.0	Comienzo del partido	0.00	10.5	9.292
			17.0	Fin del 1er cuarto	0.25	17.5	15.488
			22.0	Fin del 2do cuarto	0.50	21.5	19.028
			36.0	Fin del 3er cuarto	0.75	35.5	31.418
December 28 2020	LAL	POR	0.0	Comienzo del partido	0.00	5.5	4.868
			8.0	Fin del 1er cuarto	0.25	9.5	8.408
			-4.0	Fin del 2do cuarto	0.50	1.5	1.328
			1.0	Fin del 3er cuarto	0.75	3.5	3.098
January 7 2021	LAL	SAS	0.0	Comienzo del partido	0.00	8.5	7.522
			-8.0	Fin del 1er cuarto	0.25	3.5	3.098
			-9.0	Fin del 2do cuarto	0.50	-1.5	-1.328
			-2.0	Fin del 3er cuarto	0.75	2.5	2.212
January 8 2021	LAL	CHI	0.0	Comienzo del partido	0.00	8.5	7.522
			-5.0	Fin del 1er cuarto	0.25	3.5	3.098
			1.0	Fin del 2do cuarto	0.50	5.5	4.868
			4.0	Fin del 3er cuarto	0.75	6.5	5.752
January 18 2021	LAL	GSW	0.0	Comienzo del partido	0.00	8.5	7.522
			12.0	Fin del 1er cuarto	0.25	15.5	13.718
			16.0	Fin del 2do cuarto	0.50	15.5	13.718
			11.0	Fin del 3er cuarto	0.75	11.5	10.178
January 15 2021	LAL	NOP	0.0	Comienzo del partido	0.00	8.5	7.522
			-8.0	Fin del 1er cuarto	0.25	2.5	2.212
			-1.0	Fin del 2do cuarto	0.50	5.5	4.868
			10.0	Fin del 3er cuarto	0.75	12.5	11.062

December 30 2020	SAS	LAL	0.0	Comienzo del partido	0.00	-7.5	-6.638
			-8.0	Fin del 1er cuarto	0.25	-10.5	-9.292
			-10.0	Fin del 2do cuarto	0.50	-11.5	-10.178
			-10.0	Fin del 3er cuarto	0.75	-10.5	-9.292
January 1 2021	SAS	LAL	0.0	Comienzo del partido	0.00	-7.5	-6.638
			6.0	Fin del 1er cuarto	0.25	-3.5	-3.098
			-1.0	Fin del 2do cuarto	0.50	-6.5	-5.752
			4.0	Fin del 3er cuarto	0.75	-2.5	-2.212
January 3 2021	MEM	LAL	0.0	Comienzo del partido	0.00	-10.0	-8.850
			11.0	Fin del 1er cuarto	0.25	-2.5	-2.212
			-2.0	Fin del 2do cuarto	0.50	-8.5	-7.522
			-2.0	Fin del 3er cuarto	0.75	-6.5	-5.752
January 5 2021	MEM	LAL	0.0	Comienzo del partido	0.00	-9.5	-8.408
			5.0	Fin del 1er cuarto	0.25	-4.5	-3.982
			4.0	Fin del 2do cuarto	0.50	-4.5	-3.982
			1.0	Fin del 3er cuarto	0.75	-3.5	-3.098
January 13 2021	OKC	LAL	0.0	Comienzo del partido	0.00	-10.5	-9.292
			-9.0	Fin del 1er cuarto	0.25	-14.5	-12.832
			-12.0	Fin del 2do cuarto	0.50	-15.5	-13.718
			-22.0	Fin del 3er cuarto	0.75	-22.5	-19.912
January 10 2021	HOU	LAL	0.0	Comienzo del partido	0.00	-3.5	-3.098
			-4.0	Fin del 1er cuarto	0.25	-5.5	-4.868
			-19.0	Fin del 2do cuarto	0.50	-15.5	-13.718
			-13.0	Fin del 3er cuarto	0.75	-11.5	-10.178
January 12 2021	HOU	LAL	0.0	Comienzo del partido	0.00	-5.5	-4.868
			-21.0	Fin del 1er cuarto	0.25	-19.5	-17.258
			-23.0	Fin del 2do cuarto	0.50	-50.0	-44.250
			-26.0	Fin del 3er cuarto	0.75	-50.0	-44.250

Figura 7: Tabla de los equipos, diferencia de puntos, tipo de evento, tasa de apuesta (Handicap) y parámetro de arrastre (μ) para cada fecha de estudio.

13. Análisis de resultados obtenidos

13.1.

La elección de la página Kaggle y de un archivo de datos que contiene la información jugada tras jugada (play-by-play data) para cada partido resulta una oportunidad de uso y aplicación muy grande donde contiene la información que se requiere saber para el desarrollo del proyecto e incluso hay información como el tipo de jugada, distancia de disparo, si este fue cesta o no, entre otras que aunque no sean pertinentes para el proyecto a desarrollar sí que pueden presentar bastantes aplicaciones para un análisis de juego y hallar probabilidades de ganar, acertar, cantidad de cestas hechas y erradas en varios partidos, entre otras estadísticas. Con lo anterior como motivación se hallaron ciertas estadísticas de los partidos de los Lakers que se pueden observar en el anexo 14.7.

13.2.

Así como es posible evidenciar en el desarrollo de los objetivos específicos, se han tomado los datos brindados por las estadísticas siendo ordenados en DataFrames en donde se pueden observar distintas categorías (columnas) y su tipo de valor, bien sea un objeto (cadena de texto) o un número entero. Reduciendo así a la información importante que será tenida en cuenta para el modelo, variables importantes para la determinación del drift: la clasificación por equipos, su estado de local o visitante y la diferencia de puntos al ganar, entre otras. La elección del entorno Pandas presenta bastantes funcionalidades a la hora de categorizar, donde además de filtrar por filas (como se hizo en este ítem) también es posible filtrar mediante columnas usando una gran variedad de condiciones y funciones.

13.3.

El proceso de normalización realizado trae consigo ventajas como ordenar de manera creciente y continua los segundos transcurridos (homogeneizar el tiempo) así como poder relacionar e implementar el valor del tiempo con las demás variables a usar en el desarrollo del proyecto, en esto último es donde recae la importancia de la normalización ya que las funciones de distribución de probabilidad a usar requieren que así sea establecido el tiempo. Adicionalmente también es posible realizar gráficos de los puntos anotados en función del tiempo, los cuales pueden ser usados y extendidos a otros métodos de predicción como lo son las ventanas o series de tiempo.

13.4.

Dado que fueron seleccionados los Lakers como el equipo principal para analizar el modelo, su desarrollo y comportamiento, se puede observar en la Figura 5, producto de la filtración, que se cuenta con 15 partidos en total jugados por los Lakers de Los Ángeles durante esta temporada. De esta manera se observan sus fechas de juegos, la localidad, el tipo de juego (abierto o cerrado) así como también el ganador, información básica para la elección de los partidos de entrenamiento y prueba.

13.5.

Como fue mencionado en el marco teórico, se tenía una expresión para la densidad de probabilidad ρ la cual es función de x y t, haciendo la evidente analogía a una distribución de probabilidad normal se observa que el parámetro de desviación estándar se encuentra relacionado con D, coeficiente de difusión. Es decir, la probabilidad de encontrar la concentración de puntos alrededor de determinado punto será función de este parámetro. También el parámetro de drift μ entra como una forma de determinar el corrimiento o zona a analizar.

Teniendo en cuenta que la expresión para la matriz Hessiana se vuelve bastante engorrosa a partir de la ecuación (45), se plantea probar a calcularla numéricamente a partir del gradiente. Si esto aumenta mucho el costo computacional, se procederá a hallar la Hessiana de forma analítica. Una de las complicaciones de esta expresión analítica es que no se puede reescribir mediante el cambio de variable usual planteado en (28), dado que el sentido de este reemplazo está en aprovechar el hecho de que la distribución normal es par cuando la media se encuentre centrada en 0.

13.6.

De la relación entre el Handicap y el parámetro de arrastre (μ) prevista en el ítem anterior, se da un acercamiento a tener un conocimiento adicional al que se puede obtener en los datos descargados de Kaggle, ya que el Handicap da información de variables o tendencias previas y durante cada partido, tal como se había mencionado en secciones anteriores, sin embargo en la tabla 7 también se observa una relación directa entre la diferencia de puntos, la cual es positiva cuando gana el local y negativa en el caso contrario, y el parámetro de arrastre dando a entender que este último se refleja o entiende como un corrimiento al marcador debido a la ventaja o desventaja que tiene el equipo local.

14.2. Limpieza y categorización de los datos

Para la carga y limpieza inicial de los datos descargados se ejecuta el siguiente código:

```
df_original=pd.read_csv('NBA_PBP_2020-21.csv')   #cargar el archivo descargado previamente
df_sin_duplicados= df_original.drop_duplicates() #remover filas completamente iguales

#Ahora tomando solamente las columnas de interés:
Columnas = ['Date','WinningTeam','Quarter','SecLeft','AwayTeam','AwayPlay','AwayScore',
            'HomeTeam','HomePlay','HomeScore','ShotType','ShotOutcome']

df=df_sin_duplicados[Columnas] #creando un DataFrame con las columnas requeridas
df=df.reset_index()  #reiniciando el índice
```

Seleccionando los partidos en donde jugó LAL se usó el código:

```
dfl_LAL=df[df['HomeTeam']=='LAL']     #DataFrame para LAL de local
dfv_LAL=df[df['AwayTeam']=='LAL']     #DataFrame para LAL de visitante
df_LAL=pd.concat([dfl_LAL,dfv_LAL])   #Uniendo los 2 DataFrames anteriores
df_LAL=df_LAL.reset_index()           #Redefiniendo la cuenta del índice de cada fila
```

Los códigos anteriores permiten obtener los DataFrames mostrados en el ítem 2 de los desarrollos de los objetivos específicos y además se obtuvieron DataFrames específicos si se llega a requerir información solamente cuando LAL es local o visitante correspondientes a dfl_LAL y dfv_LAL respectivamente.

14.3. Normalización y organización de manera continua del tiempo

Creando una copia del DataFrame dfl_LAL y considerando que la conversión temporal debe tomar por separado cada cuarto, se implementa el siguiente código que trata los segundos de cada cuarto por separado:

```
df_game1=df_LAL #creando una copia

df_game11Q=df_game1[df_game1['Quarter']==1] #Conversión temporal para el primer cuarto
df_game11Q['SecLeft']=(720-df_game11Q['SecLeft'])/2880

df_game12Q=df_game1[df_game1['Quarter']==2] #Conversión temporal para el segundo cuarto
df_game12Q['SecLeft']=(1440-df_game12Q['SecLeft'])/2880

df_game13Q=df_game1[df_game1['Quarter']==3] #Conversión temporal para el tercer cuarto
df_game13Q['SecLeft']=(2160-df_game13Q['SecLeft'])/2880

df_game14Q=df_game1[df_game1['Quarter']==4] #Conversión temporal para el cuarto cuarto
df_game14Q['SecLeft']=(2880-df_game14Q['SecLeft'])/2880
```

Ya con los segundos de cada cuarto convertidos, se juntan las particiones con el comando concat (concatenación):

```
df_game1Q=pd.concat([df_game11Q,df_game12Q,df_game13Q,df_game14Q])
df_game1Q=df_game1Q.reset_index()  #redefiniendo la cuenta del índice de cada fila

#Renombrando la columna 'SecLeft' a 'Sec' (haciendo referencia a segundos transcurridos)
df_game1Q=df_game1Q.rename(columns={'SecLeft':'Sec'})
```

Con lo anterior realizado, se obtiene en la columa 'Sec' los segundos normalizados y ordenados de manera continua en el transcurso de cada partido.

14.4. Partidos de entrenamiento y prueba

Para visualizar cada fecha sin repetir se crea un DataFrame que contiene la fila correspondiente a la jugada que marca final de cada partido, de dicha fila se obtiene el puntaje final de cada equipo, el ganador, la fecha y a partir de la diferencia de puntaje se puede saber el tipo de partido (abierto o cerrado):

```
#DataFrame que contiene el puntaje al final de cada partido:
df_dif=df_LAL[df_LAL['AwayPlay']=='End of Game']
df_dif=df_dif.reset_index()
dif=[] #Vector vacío donde se guardará la diferencia de puntaje
for i in range(num_games): #Calculando diferencia de puntaje
    dif.append(np.abs(df_dif.loc[i,'AwayScore']-df_dif.loc[i,'HomeScore']))
for i in range(num_games):
    if dif[i]<11: #Imprimiendo para partidos cerrados
        print(' El partido con',cyan(df_dif.loc[i,'HomeTeam'],['bold','reverse']), 'de'
              'local y',cyan(df_dif.loc[i,'AwayTeam'],['bold','reverse']),'de visitante,'
              'disputado en:',black(df_dif.loc[i,'Date'],['bold','underlined']),'fue un '
              'partido', green('cerrado','bold'),'y ganó',magenta(df_dif.loc[i,'WinningTeam'],
                                                                  ['bold','reverse']) ,'\n')
    else: #Imprimiendo para partidos abiertos
        print(' El partido con',cyan(df_dif.loc[i,'HomeTeam'],['bold','reverse']), 'de'
              'local y',cyan(df_dif.loc[i,'AwayTeam'],['bold','reverse']),'de visitante,'
              'disputado en:',black(df_dif.loc[i,'Date'],['bold','underlined']),'fue un'
              'partido',   red('abierto','bold'),'y ganó',magenta(df_dif.loc[i,'WinningTeam'],
                                                                  ['bold','reverse']) ,'\n')
```

La sálida del código anterior corresponde a la figura 5 correspondiente al objetivo específico número 4, gracias a la salida del código anterior se seleccionan los partidos de entrenamiento y prueba de acuerdo al orden cronológico como se muestra en la tabla 1.

14.5. Desarrollo Analítico para Emplear Máxima Verosimilitud

14.5.1. Desarrollo Método de Máxima Verosimilitud

14.5.2. Logaritmo de la Verosimilitud

$$
\begin{aligned}
l(\beta; y, X) &= ln(L(\beta; y, X)) \\
&= ln\left(\prod_{i=1}^{N} \left(\Phi(x_i\beta)y_i\right)[1-\Phi(x_i\beta)]^{1-y_i}\right) \\
&= \sum_{i=1}^{N} [y_i \ln\left(\Phi(x_i\beta)\right) + (1-y_i)\ln\left(1-\Phi(x_i\beta)\right)]
\end{aligned}
$$

14.5.3. Gradiente Logaritmo de la Verosimilitud

$$
\begin{aligned}
\nabla_\beta l(\beta; y, X) &= \nabla\left(\sum_{i=1}^{N} [y_i \ln\left(\Phi(x_i\beta)\right) + (1-y_i)\ln\left(1-\Phi(x_i\beta)\right)]\right) \\
&= \sum_{i=1}^{N} \left[y_i \frac{f(x_i\beta)}{\Phi(x_i\beta)} x_i + (1-y_i)\frac{-f(x_i\beta)}{1-\Phi(x_i\beta)} x_i\right] \\
&= \sum_{i=1}^{N} \frac{f(x_i\beta)}{\Phi(x_i\beta)[1-\Phi(x_i\beta)]}[y_i - \Phi(x_i\beta)]x_i
\end{aligned}
$$

14.5.4. Nueva variable dependiente en la Verosimilitud

$$[\Phi(x_i\beta)]^{y_i} = [\Phi(x_i\beta q_i)]^{y_i}$$

Pero, teniendo en cuenta que por la simetría de la distribución normal:

$$[1 - \Phi(x_i\beta)] = \Phi(-x_i\beta)$$

De donde se pueden evaluar los casos con y_i siendo 0 o 1, y se obtiene la relación:

$$[1 - \Phi(x_i\beta)]^{1-y_i} = [\Phi(x_i\beta q_i)]^{1-y_i}$$

De donde:

$$\begin{aligned} L(\beta; y, X) &= \prod_{i=1}^{N} \left(\Phi(x_i\beta q_i)^{y_i+1-y_i}\right) \\ &= \prod_{i=1}^{N} \Phi(x_i\beta q_i) \end{aligned}$$

14.5.5. Matriz Hessiana

$$\begin{aligned} \nabla_{\beta\beta} l(\beta; y, X) &= \nabla_\beta(\nabla_\beta l(\beta; y, X)) \\ &= \nabla_\beta \left(\sum_{i=1}^{N} \frac{f(x_i\beta q_i) q_i x_i}{\Phi(x_i\beta q_i)} \right) \\ &= \sum_{i=1}^{N} \left(\frac{1}{\Phi(x_i\beta q_i)} \nabla_\beta f(x_i\beta q_i) + f(x_i\beta q_i) \nabla_\beta \left(\frac{1}{\Phi(x_i\beta q_i)} \right) \right) q_i x_i \\ &= \sum_{i=1}^{N} \left(-\frac{1}{\Phi(x_i\beta q_i)} f(x_i\beta q_i)(q_i x_i \beta) q_i x_i^T + f(x_i\beta q_i) \left(\frac{1}{\Phi(x_i\beta q_i)^2} f(x_i\beta q_i) q_i x_i^T \right) \right) q_i x_i \\ &= -\sum_{i=1}^{N} \left(\frac{f(x_i\beta q_i) q_i}{\Phi(x_i\beta q_i)} x_i\beta + \frac{f(x_i\beta q_i) q_i}{\Phi(x_i\beta q_i)} \frac{f(x_i\beta q_i) q_i}{\Phi(x_i\beta q_i)} \right) x_i^T x_i \\ &= -\sum_{i=1}^{N} \lambda_i(x_i\beta + \lambda_i) x_i^T x_i \end{aligned}$$

14.5.6. Logaritmo de la Verosimilitud

Para obtener la ecuación (43) se procedió a usar la siguiente propiedad para el logaritmo de una productoria:

$$
\begin{aligned}
ln\left(\prod_i \prod_j F(x_{ij})\right) &= ln\left(\prod_i F(x_{i1})F(x_{i2})F(x_{i3})\cdots\right) \\
&= ln\left(\prod_i G(x_i)\right) \\
&= \sum_i (ln(G(x_i))) \\
&= \sum_i \left(ln\left(\prod_j F(x_{ij})\right)\right) \\
&= \sum_i \sum_j (ln\left(F(x_{ij})\right))
\end{aligned}
$$

De forma que por reemplazo directo de esta formula en la verosimilitud hallada para el problema en la ecuación (38) se llega rápidamente a la ecuación (43).

14.6. Definición del parámetro de arrastre μ

Para determinar el parámetro de arrastre μ es necesario conocer las tasas de apuestas (handicap) para el equipo local, los cuales se pueden visualizar en la página Nowgoal3 como:

data.nowgoal3.com/NBA/2in1odds.htm?id=405625&cId=8

Full Time | First Half | Quarters

National Basketball Association - Memphis Grizzlies vs Los Angeles Lakers 03-01-2021 18:00 Sunday

Bet365 Asian Handicap Odds

Time	Score	Home	Asian Handicap	Away	Update	Status
4th Qtr 02:33	90 - 101				03-01-2021 20:10:51	Run
4th Qtr 02:55	90 - 98	0.86	-8.5	0.86	03-01-2021 20:10:15	Run
4th Qtr 03:04	90 - 98	0.86	-7.5	0.86	03-01-2021 20:09:59	Run
4th Qtr 03:18	90 - 96				03-01-2021 20:09:44	Run
4th Qtr 03:33	90 - 96	0.80	-6.5	0.95	03-01-2021 20:09:29	Run
4th Qtr 03:42	90 - 96	0.95	-6.5	0.80	03-01-2021 20:09:12	Run
4th Qtr 03:42	90 - 95	0.90	-6.5	0.83	03-01-2021 20:08:38	Run
4th Qtr 03:54	90 - 95	0.95	-6.5	0.80	03-01-2021 20:08:21	Run
4th Qtr 04:12	87 - 95	0.90	-7.5	0.83	03-01-2021 20:08:02	Run

Bet365 Over/Under Odds

Time	Score	Over	Total	Under	Update	Status
4th Qtr 03:42	90 - 95		Closed		03-01-2021 20:08:38	Run
4th Qtr 03:54	90 - 95	0.80	205.5	0.95	03-01-2021 20:08:21	Run
4th Qtr 04:12	87 - 95	0.83	201.5	0.90	03-01-2021 20:08:02	Run
4th Qtr 04:30	87 - 95	0.95	203.5	0.80	03-01-2021 20:07:46	Run
4th Qtr 04:42	85 - 95	0.76	200.5	1.00	03-01-2021 20:07:30	Run
4th Qtr 05:00	85 - 93	0.86	201.5	0.86	03-01-2021 20:07:13	Run
4th Qtr 05:10	85 - 93		Closed		03-01-2021 20:06:38	Run
4th Qtr 05:17	85 - 93	0.86	203.5	0.86	03-01-2021 20:06:22	Run
4th Qtr 05:17	85 - 92	0.86	204.5	0.86	03-01-2021 20:06:04	Run
4th Qtr 05:21	85 - 92		Closed		03-01-2021 20:05:45	Run
4th Qtr 05:34	85 - 92	0.80	203.5	0.95	03-01-2021 20:05:29	Run
4th Qtr 05:37	84 - 92	0.86	202.5	0.86	03-01-2021 20:05:13	Run
4th Qtr 05:37	83 - 92	0.86	201.5	0.86	03-01-2021 20:04:55	Run
4th Qtr 05:53	83 - 92	1.00	202.5	0.76	03-01-2021 20:04:40	Run
4th Qtr 05:55	83 - 92		Closed		03-01-2021 20:04:22	Run
4th Qtr 05:57	83 - 92	0.83	202.5	0.90	03-01-2021 20:04:03	Run
4th Qtr 06:17	83 - 90	0.90	203.5	0.83	03-01-2021 20:03:46	Run
4th Qtr 06:44	83 - 90		Closed		03-01-2021 20:02:42	Run

Figura 8: Página Nowgoal3 usada para la obtención de las tasas de apuestas

Recolectando los Handicap para los partidos de interés y organizando en una tabla:

LOCAL	VISITANTE	FECHA	INICIO	1RE CUARTO	2DO CUARTO	3ER CUARTO	CASA
LAL	LAC	22/12/2020	2	-11,5	1,5	-7,5	BET365
LAL	DAL	25/12/2020	6	6,5	12,5	11,5	BET365
LAL	MIN	27/12/2020	10,5	17,5	21,5	35,5	BET365
LAL	POR	28/12/2020	5,5	9,5	1,5	3,5	BET365
LAL	SAS	07/01/2021	8,5	3,5	-1,5	2,5	BET365
LAL	CHI	08/01/2021	8,5	3,5	5,5	6,5	BET365
LAL	GSW	18/01/2021	8,5	15,5	15,5	11,5	BET365
LAL	NOP	15/01/2021	8,5	2,5	5,5	12,5	BET365
SAS	LAL	30/12/2020	-7,5	-10,5	-11,5	-10,5	BET365
SAS	LAL	01/01/2021	-7,5	-3,5	-6,5	-2,5	BET365
MEM	LAL	03/01/2021	-10	-2,5	-8,5	-6,5	BET365
MEM	LAL	05/01/2021	-9,5	-4,5	-4,5	-3,5	BET365
OKC	LAL	13/01/2021	-10,5	-14,5	-15,5	-22,5	BET365
HOU	LAL	10/01/2021	-3,5	-5,5	-15,5	-11,5	BET365
HOU	LAL	12/01/2021	-5,5	-19,5	-50	-50	BET365

Figura 9: Handicaps para los 15 partidos de interés.

Cargando los datos anteriores, ordenados por cuartos, en forma de vectores de python (T_0 hace referencia al inicio del partido, T_1 al final del primer cuarto y así sucesivamente):

```
T_0=np.array([2.0,6.0,10.5,5.5,8.5,8.5,8.5,8.5,-7.5,-7.5,-10.0,-9.5,-10.5,-3.5,-5.5])
T_1=np.array([-11.5,6.5,17.5,9.5,3.5,3.5,15.5,2.5,-10.5,-3.5,-2.5,-4.5,-14.5,-5.5,-19.5])
T_2=np.array([1.5,12.5,21.5,1.5,-1.5,5.5,15.5,5.5,-11.5,-6.5,-8.5,-4.5,-15.5,-15.5,-50.0])
T_3=np.array([-7.5,11.5,35.5,3.5,2.5,6.5,11.5,12.5,-10.5,-2.5,-6.5,-3.5,-22.5,-11.5,-50.0])
```

Adicionalmente al parámetro de arrastre también se halla la diferencia de puntos en el final de cada cuarto, partiendo el DataFrame en tres partes mediante el código a continuación:

```
df_ejemplo=df_game1Q #Creando una copia
df_dif1=df_ejemplo[df_ejemplo['AwayPlay']=='End of 1st quarter']
df_dif1=df_dif1.reset_index()  #redefiniendo la cuenta del índice de cada fila

df_dif2=df_ejemplo[df_ejemplo['AwayPlay']=='End of 2nd quarter']
df_dif2=df_dif2.reset_index()  #redefiniendo la cuenta del índice de cada fila

df_dif3=df_ejemplo[df_ejemplo['AwayPlay']=='End of 3rd quarter']
df_dif3=df_dif3.reset_index()  #redefiniendo la cuenta del índice de cada fila

#Creando un DataFrame de 60 filas:
df_dif=pd.concat([df_dif1,df_dif1,df_dif1,df_dif1])
df_dif=df_dif.reset_index()
```

El motivo de crear el DataFrame df_dif de 60 filas es debido a que hay 4 eventos por cada partido: Inicio del partido, final del primer cuarto, final del segundo cuarto y final del tercer cuarto. Recordando que en total hay 15 partidos, se tienen entonces 60 filas donde además de los eventos, el tiempo también tendrá un orden cíclico el cual es 0, 0.25, 0.5 y 0.75. De acuerdo a este orden se crean las columnas de: Local, visitante, diferencia de puntos, evento, tiempo, tasa de apuesta y parámetro de arrastre como sigue:

```
T=np.zeros(num_games*4)     #Tasa de apuesta
DIF=np.zeros(num_games*4)   #Diferencia de puntos
E=[None] * num_games*4      #Evento
S=np.zeros(num_games*4)     #Segundos
D=[None] * num_games*4      #Fecha
```

```
L=[None] * num_games*4      #Local
V=[None] * num_games*4      #Visitante

#En el siguiente bucle i va de 0 a 56 en pasos de 4
# j va de 0 a 14 en pasos de 1
for i in range(0,num_games*4,4):
    j=float(i)/4

    S[i], E[i],T[i]=0,'Comienzo del partido',T_0[int(j)]
    S[i+1], E[i+1],T[i+1]=0.25,'Fin del 1er cuarto',T_1[int(j)]
    S[i+2], E[i+2],T[i+2]=0.5,'Fin del 2do cuarto',T_2[int(j)]
    S[i+3], E[i+3],T[i+3]=0.75,'Fin del 3er cuarto',T_3[int(j)]

    D[i]=df_dif1.loc[int(j),'Date']
    D[i+1],D[i+2],D[i+3]= ' ',' ',' '

    DIF[i+1]=int(df_dif1.loc[int(j),'HomeScore']-df_dif1.loc[int(j),'AwayScore'])
    DIF[i+2]=int(df_dif2.loc[int(j),'HomeScore']-df_dif2.loc[int(j),'AwayScore'])
    DIF[i+3]=int(df_dif3.loc[int(j),'HomeScore']-df_dif3.loc[int(j),'AwayScore'])

    L[i],V[i]=df_dif1.loc[int(j),'HomeTeam'],df_dif1.loc[int(j),'AwayTeam']
    L[i+1], L[i+2], L[i+3]=' ',' ',' '
    V[i+1], V[i+2], V[i+3]=' ',' ',' '

#Hallando el parámetro de arrastre con 3 cifras decimales
u=np.round(0.885*T,3) #Implementando el parámetro de arrastre
```

Integrando la información obtenida a manera de columnas al DataFrame *df_dif*:

```
df_dif['Local']=L
df_dif['Visitante']=V
df_dif['DifScore']=DIF.tolist()
df_dif['Evento']=E
df_dif['Tiempo']=S.tolist()
df_dif['TasaApuesta']=T.tolist()
df_dif['Arrastre (u)']=u.tolist()
```

Sin embargo las columnas que estaban guardadas anteriormente no son de interés, así que se seleccionan las columnas necesarias, se renombra el DataFrame y se toma la fecha como el índice de dicho DataFrame:

```
DF=df_dif[['Local','Visitante','DifScore','Evento','Tiempo','TasaApuesta','Arrastre (u)']]
DATE=pd.Series(D)
DF=DF.set_index(DATE)
```

Finalmente, viendo la información del DataFrame creado:

```
DF.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 60 entries, December 22 2020 to
Data columns (total 7 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Local          60 non-null     object
 1   Visitante      60 non-null     object
 2   DifScore       60 non-null     float64
 3   Evento         60 non-null     object
 4   Tiempo         60 non-null     float64
 5   TasaApuesta    60 non-null     float64
 6   Arrastre (u)   60 non-null     float64
dtypes: float64(4), object(3)
memory usage: 3.8+ KB
```

Figura 10: DataFrame llamado DF que contiene la diferencia de puntos, tipo de evento, tasa de apuesta y parámetro de arrastre en varios instantes de tiempo para cada partido de los 15 considerados a usar.

14.7. Probabilidad de ganar, acertar y cantidad de cestas hechas y erradas

Creando una nueva columna en el DataFrame original a la cual se le asignan valores binarios donde se asigna 1 cuando gana LAL y 0 al caso contrario:

```
df['W_P']=df['WinningTeam'] #creando una nueva columna

df.loc[df.W_P!='LAL', 'W_P'] = 0  #Cuando no gana LAL
df.loc[df.W_P=='LAL', 'W_P'] = 1  #Cuando gana LAL
```

El motivo de crear dicha columna de esa manera es que ahora resulta más sencillo calcular la probabilidad de ganar para LAL, debido a que solo es necesario hallar el promedio dado a que cuando gana algún otro equipo su valor en la nueva columna será cero obteniendo entonces:

Hallando las probabilidades de ganar para LAL

```
P_All=df_LAL[df_LAL['AwayPlay']=='End of Game']['W_P'].mean()
P_l=dfl_LAL[dfl_LAL['AwayPlay']=='End of Game']['W_P'].mean()
P_v=dfv_LAL[dfv_LAL['AwayPlay']=='End of Game']['W_P'].mean()

print('En general LAL posee un porcentaje de victorias del', black(np.round(P_All*100,1),['bold','underlined']),'%')
print('De local LAL posee un porcentaje de victorias del',black(np.round(P_l*100,1),['bold','underlined']),'%')
print('De visitante LAL posee un porcentaje de victorias del',black(np.round(P_v*100,1),['bold','underlined']),'%')
```

```
En general LAL posee un porcentaje de victorias del 73.3 %
De local LAL posee un porcentaje de victorias del 50.0 %
De visitante LAL posee un porcentaje de victorias del 100.0 %
```

Figura 11: Probabilidades de victorias para LAL en general, de local y de visitante.

Aprovechando el hecho que la columna de texto 'ShotOutcome' tiene dos valores: make y miss, los cuales hacen alusión si al lanzar se realizó una anotación o no, entonces es posible hallar la probabilidad de acierto de los 15 partidos escogidos mediante el siguiente código:

```
num_games=df_LAL['Date'].nunique()
P_acierto=df_LAL['ShotOutcome'].value_counts().min()/df_LAL['ShotOutcome'].value_counts().sum()

print('LAL ha jugado', black(num_games,['bold','underlined']),'partidos en el transcurso de la temporada, los cuales tuvieron un
      ' de acierto del',black(np.round(P_acierto*100,1),['bold','underlined']),'% \n (considerando cestas hechas y erradas por to
```

```
LAL ha jugado 15 partidos en el transcurso de la temporada, los cuales tuvieron un porcentaje de acierto del 46.6 %
 (considerando cestas hechas y erradas por todos los equipos)
```

Figura 12: Probabilidades de acierto para todos los partidos que jugó LAL.

Realizando lo anterior para los casos cuando LAL es local y visitante, empezando por caso local se tiene:

Los Angeles Lakers (LAL) de local:

```
dfL_LAL=dfl_LAL  #creando una copia

num_local=dfL_LAL['Date'].nunique()  #seleccionando número de fechas o partidos
P_acierto_L=dfL_LAL['ShotOutcome'].value_counts().min()/dfL_LAL['ShotOutcome'].value_counts().sum()


print(f'Hubieron {num_local} partidos donde LAL fue local los cuales tuvieron un porcentaje de acierto del {np.round(P_acierto_L*
      '(considerando cestas hechas y erradas por ambos equipos) ')
```

```
Hubieron 8 partidos donde LAL fue local los cuales tuvieron un porcentaje de acierto del 48.1%
(considerando cestas hechas y erradas por ambos equipos)
```

Figura 13: Probabilidades de acierto para los partidos que jugó LAL de local.

Representando lo anterior por medio de un histograma para el equipo local:

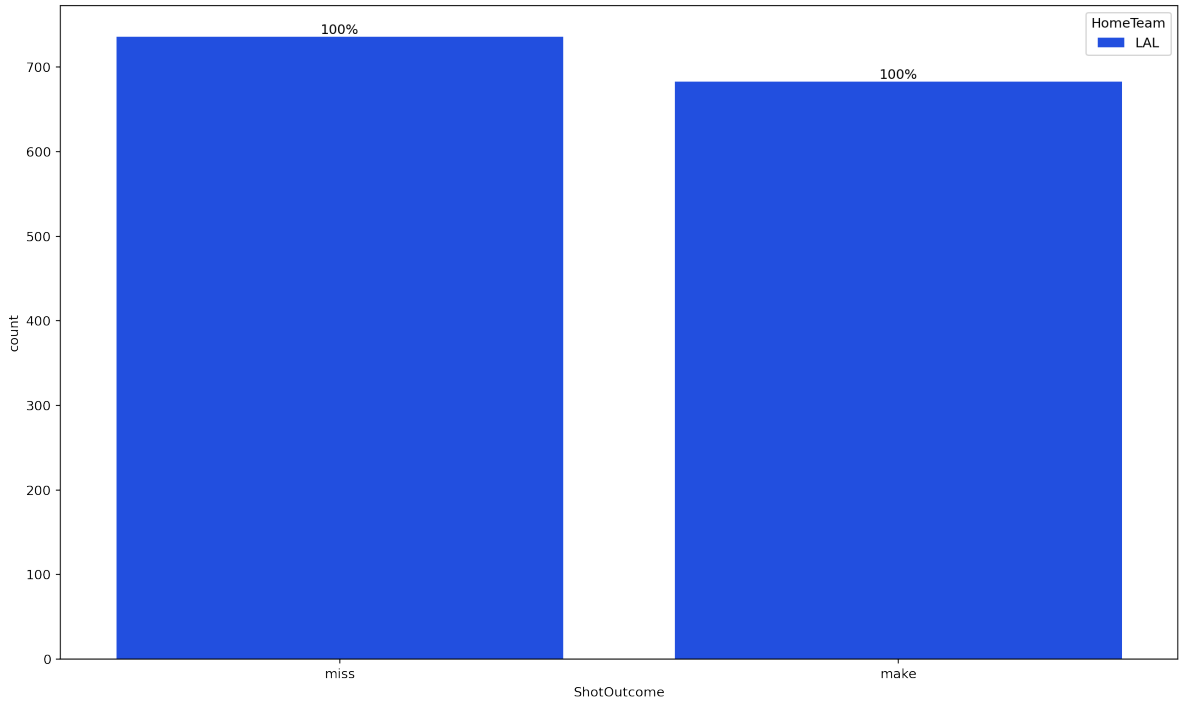

Figura 14: Número de cestas hechas y erradas por LAL de local.

Y para los equipos visitantes se obtuvo el siguiente histograma:

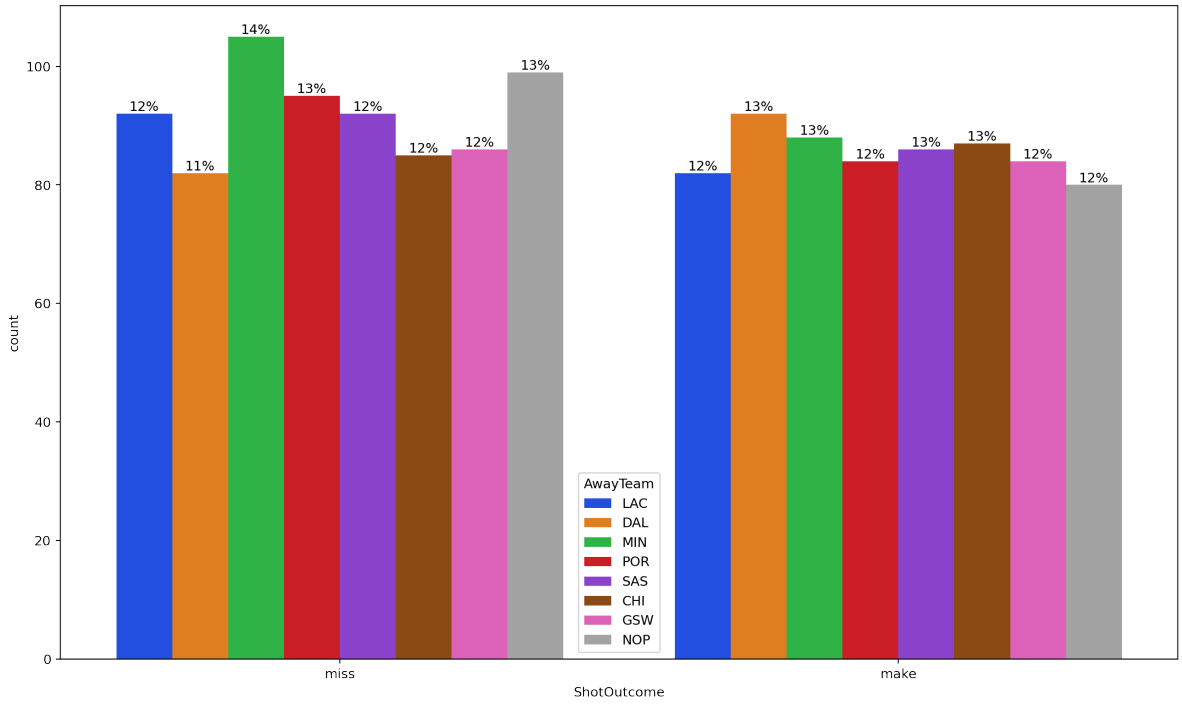

Figura 15: Número de cestas hechas y erradas por los equipos visitantes.

Ahora para el caso donde LAL es visitante, se tiene la siguiente probabilidad de acierto:

Los Angeles Lakers (LAL) de visitante:

```
dfV_LAL=dfv_LAL

num_visitante=dfV_LAL['Date'].nunique()
P_acierto_V=dfV_LAL['ShotOutcome'].value_counts().min()/dfV_LAL['ShotOutcome'].value_counts().sum()

print(f'Hubieron {num_visitante} partidos donde LAL fue local los cuales tuvieron un porcentaje de acierto del {np.round(P_aciert
      '(considerando cestas hechas y erradas por ambos equipos) ')
```

```
Hubieron 7 partidos donde LAL fue local los cuales tuvieron un porcentaje de acierto del 44.9%
(considerando cestas hechas y erradas por ambos equipos)
```

Figura 16: Probabilidades de acierto para los partidos que jugó LAL de visitante.

Representando lo anterior por medio de un histograma para los equipos locales:

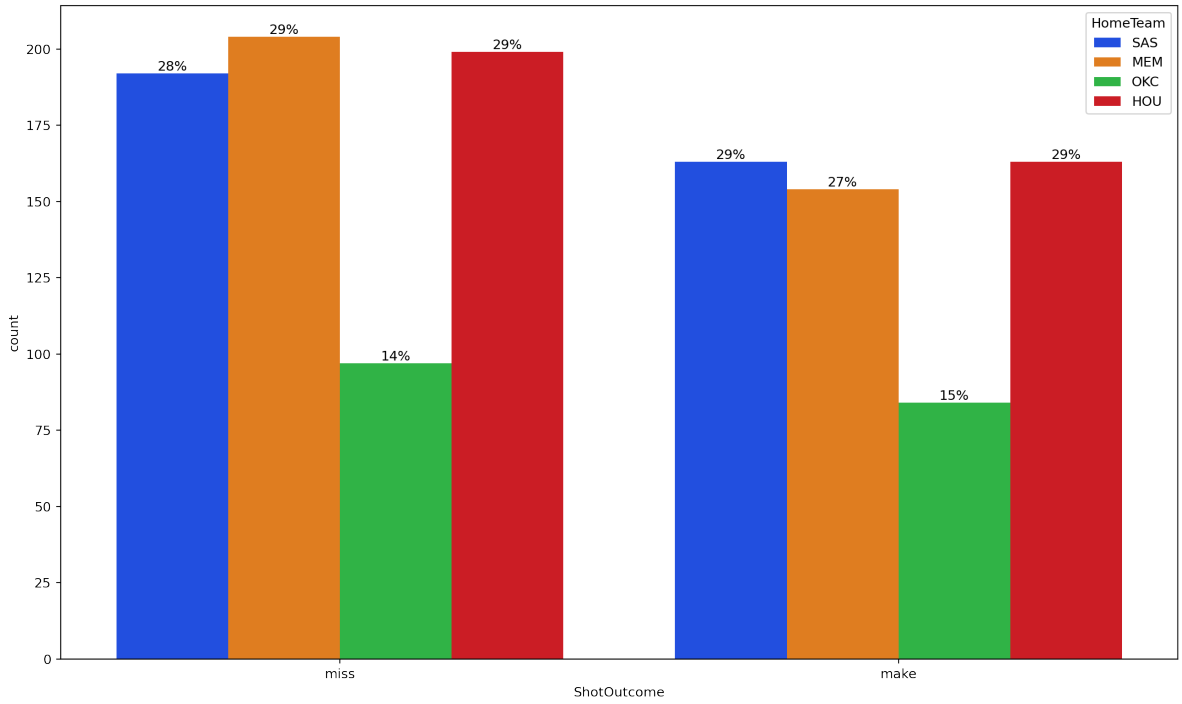

Figura 17: Número de cestas hechas y erradas por los equipos locales.

Y para el equipo visitante:

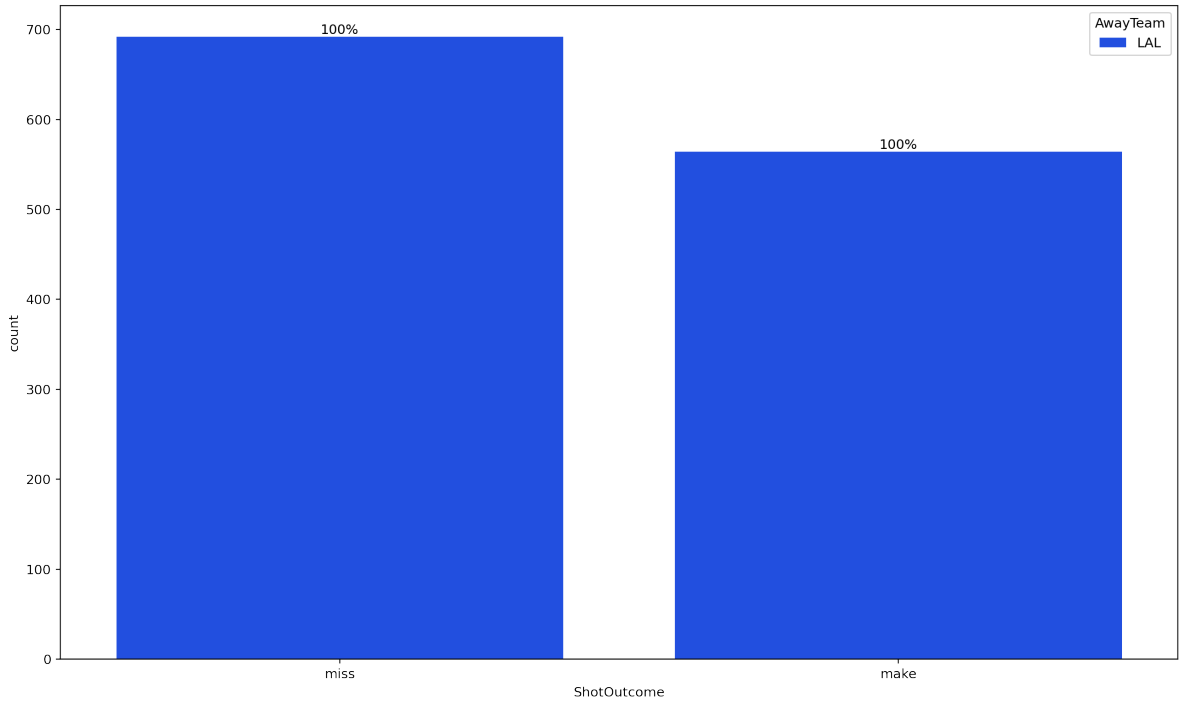

Figura 18: Número de cestas hechas y erradas por LAL de visitante.

Referencias

[1] K. Song, Y. Gao, J. Shi,
"Making real-time preditions for NBA basketball games by combinig the historical data and bookmarker's betting line",
Physica A (2020).
1, 11, 12

[2] J. Martín-González, Y. deSaá Guerra, J. García-Manso, E. Arriaza, T. Valverde-Estévez,
"The Poisson model limits in NBA basketball: Complexity in team sports",
Physica A (2016).
2, 3, 11, 12

[3] Y. de S Guerra, J. M. Martín González, S. Sarmiento Montesdeoca, D. Rodriguez Ruiz, N. Arjonilla-López, J. M. García-Manso,
"Basketball scoring in NBA games: an example of complexity",
Journal of Systems Science and Complexity (2013)
1, 2, 11

[4] Hal S. Stern,
"A Brownian motion for the progress of sports scores",
Journal of the American Statistical Association (1994).
2, 3, 5, 6, 7, 10, 11, 12, 13, 14, 17, 23

[5] L. Blanco,
"Probabilidad",
Universidad Nacional de Colombia, 2004.
2, 3, 6, 11

[6] K. Shirley,
"A Markov Model for Basketball",
Columbia University, Applied Statistics center.
2, 3

[7] J. Agbinya,
"Applied Data Analytics Principles and Applications",
River Publishers Denmark (2020).
3

[8] R. Serfozo,
"Basics of Applied Stochastic Processes",
Springer (2009).
4

[9] Abril, F. S. and Quimbay, C. J., *"Temporal fluctuation scaling in nonstationary time series using the path integral formalism"*,
American Physical Society (2021)
14

[10] P. Vracar, E. Strumbelj, I. Kononenko,
"Modeling basketball play-by-play data",
Expert Systems With Applications (2016).
2, 11

[11] E. Strumbel,
"On determinig probability forecast from betting ods",
Internal Journal of Forecasting (2014).
10

[12] E. M. Alameda,
"A dynamic Bayesian network to predict the total points scored in national basketball association games",
Iowa State University (2019).
11

[13] G. Boshnakov, T. Kharrat, I. McHale,
"A bivariate Weibull count model for forecastting association football scores",
International Journal of Forecasting (2017)
11

[14] D. Ursin,
"A Markov model with applications",
University of Wisconsin-Milwaukee (2014).
11

[15] Wikipedia,
"Deportes"
2021, `https://es.wikipedia.org/wiki/Deporte#cite_note-8` 11

[16] Red História,
"¿Cuál es el origen de las casas de apuestas?",
2020, `https://redhistoria.com/cual-es-el\-origen-de-las-casas-de-apuestas/`
11

[17] Interactive Chaos, n.d, *"One Hot Encoding"*,
`https://interactivechaos.com/es/manual/tutorial-dne-learning/one-hot-encoding`
14

[18] Kubo. Ryogo,
"Statistical Mechanics",
Springer (1965).
11

[19] M. Oldham A. T. Crooks,
"Drafting agent-based modeling into basketball analytics",
George Mason University. Department of Computational and Data Sciences.
14

[20] M. Wright,
"OR analysis of Sporting Rules - A Survey.",
European Journal of Operational Research 232(1): 1-8.
14

[21] Random, K. Siegrist,
"Brownian Motion with Drift",
`https://randomservices.org/random/brown/Drift.html`
4, 5, 6

[22] Mordor Intelligence,
"SPORTS ANALYTICS MARKET - GROWTH, TRENDS, COVID-19 IMPACT, AND FORECASTS (2021 - 2026)",
https://mordorintelligence.com/industry-reports/sports-analytics-market
14

[23] Arfken. George, Weber. Hans,
"Mathematical methods for physicists",
Elsevier academic Press (2005).
4

[24] Gaviria. Carolina,
"Movimiento Browniano de partículas con volatilidad dependiente de la posición",
Universidad de los Andes (2005). 2, 4, 5

[25] J. H. Aldrich and N. D. Forrest,
"Linear Probability, Logit, and Probit Models",
Sage University, No. 07-045 (1984). pg. 48-65. 7

[26] M. Taboga,
"Probit classification model (or probit regression)",
https://www.statlect.com/fundamentals-of-statistics/probit-classification-model
7

[27] M. Taboga,
"Maximum likelihood",
https://www.statlect.com/fundamentals-of-statistics/maximum-likelihood
7

[28] N. R. Draper and H. Smith,
"Applied regression analysis",
Wiley series in probabilityand statistics,(1998). pg. 276-286.

23

[29] I. J. Myung,
"Tutorial on maximum likelihood estimation",
Journal of Mathematical Psychology, vol. 47 No. 1 (2003). pg. 90-100. 7

[30] M. Taboga,
"Probit classification model - Maximum likelihood",
https://www.statlect.com/fundamentals-of -statistics/probit-model-maximum-likelihood
8

[31] M. D. Smith, Course Notes MIT.
"Newton-Raphson Technique",
`https://web.mit.edu/10.001/Web/Course_Notes/NLAE/node6.html`

[32] Álvares G.,
"Las casas de apuestas echan raíces en los pabellones de la NBA",
Mundo deportivo (2021).
9, 12

[33] Esparza D, Romanistiky K,
"Historia y deporte: la necesidad de estudiar génesis de deportes",
Palacky University in Olomouc (2019).
Revista Internacional de Ciencias del Deporte Volumen XV - Año XV Paginas: 119-122 - ISSN: 1885-3137 Número 56
1

[34] "*Distribución Normal*",
Synergy Vision, (2014-2019),
`https://synergy.vision/corpus/probabilidades/2017-08-14-normal.html`.
11

[35] "*¿Cómo hacer handicap apuestas NBA?*",
Apuestas Perú,
`https://apuestivas.pe/handicap-apuestas-nba/`
23