

Player2Vec y otros Modelos Probabilísticos

Framework para Evaluar Performance en Jugadores de Fútbol

Tomás Glauberma * Ignacio Pardo †
Juan Ignacio Silvestri ‡

Resumen

En la última década, el análisis deportivo ha evolucionado hacia una perspectiva cada vez más matemática y sofisticada. Aplicaciones como el uso de análisis espacial en Basketball (Goldsberry, 2012) y la investigación estadística del Brentford con Smartodds son ejemplos claros de la tendencia creciente en este campo. El béisbol, por mucho tiempo el deporte preferido para la analítica, ha experimentado una profunda transformación con la implementación de Sabermetrics (Baumer, 2015). La introducción de herramientas analíticas avanzadas ha producido resultados positivos para muchos equipos, lo que resalta el valor de estudiar métricas específicas dentro de cada deporte.

Este desarrollo se centra en el fútbol, un deporte en el cual los análisis previos se han concentrado, en su mayoría, en predecir resultados de partidos y mejorar el rendimiento de los equipos. Sin embargo, este trabajo propone un enfoque diferente al analizar el impacto de los jugadores sobre la posesión de balón y los disparos del equipo desde una perspectiva probabilística.

A partir de la métrica PSL propuesta en el paper en desarrollo *Soccer Networks* (Huang et al., n.d.) planteamos un proceso para comparar el impacto que tienen los jugadores sobre la performance del equipo. Logramos formular una metodología para estudiar la distribución de la performance de un equipo. Luego, proponemos una serie de métodos y métricas para comparar el rendimiento de dos formaciones de jugadores. Desarrollamos una forma de representación vectorial (Embeddings) de los jugadores, llamada Player2Vec, un modelo de Machine Learning también basado sobre el modelo de redes de jugadores planteado en el mismo paper del PSL. Esto último permite desarrollar modelos predictivos sobre el rendimiento de los jugadores en un equipo. Nuestro modelo final logra predecir la performance de los jugadores un 58.99% mejor que asumir las distribuciones previas como *priors*.

Palabras Clave: Fútbol, Machine Learning, Redes de Jugadores, Embeddings, Cadenas de Markov

Player2Vec and other Probabilistic Models
Framework for Evaluating Player Performance in Soccer

Abstract

*Universidad Torcuato Di Tella, tglauberma@mail.utdt.edu

†Universidad Torcuato Di Tella, ipardo@mail.utdt.edu

‡Universidad Torcuato Di Tella, jsilvestri@mail.utdt.edu

Over the last decade, sports analysis has evolved into an increasingly mathematical and sophisticated perspective. Applications such as the use of spatial analysis in Basketball and Brentford's statistical research with Smartodds are clear examples of the growing trend in this field. Baseball, long the sport of choice for analytics, has undergone a profound transformation with the implementation of Sabermetrics. The introduction of advanced analytics tools has produced positive results for many teams, highlighting the value of studying specific metrics within each sport.

This development focuses on soccer, a sport in which previous analyses have mostly concentrated on predicting match results and improving team performance as a whole. However, this work proposes a different approach by analyzing the relationship between the player as an individual and the respective player training.

Specifically, we explore the impact of players on ball possession and team shots from a probabilistic perspective. Starting from the probability of kicking before losing the ball (PSL) proposed in the developing paper "Soccer Networks", and its demonstrated positive influence on team results, we propose a process to compare the impact of players on PSL and consequently on team performance.

We succeeded in formulating a methodology to study the PSL distribution of a team for which we propose a series of methods and metrics to compare the performance of two formations of players. We developed vector representation of the players (Embeddings), called Player2Vec, based on the player graph proposed in the same paper "Soccer Networks". The latter allows us to develop predictive models about the performance of players in a team. Our final model is 58.99% better at predicting player performance than assuming previous distributions as priors, thus outperforming simpler Bayesian models.

Keywords: Soccer, Machine Learning, Player Networks, Embeddings, Markov Chains

1 Introducción

A diferencia de otros deportes como el béisbol o el basketball, el fútbol ha sido tradicionalmente menos propenso a la aplicación de técnicas avanzadas de análisis de datos y aprendizaje automático. Sin embargo, en los últimos años ha habido un crecimiento significativo en el uso de herramientas analíticas para evaluar el rendimiento de los jugadores y los equipos.

En la última década el análisis del fútbol ha evolucionado hacia una perspectiva cada vez más matemática y sofisticada. El desarrollo que más impacto tuvo sin dudas es el de la métrica de Expected Goals (**xG**) (Green, 2012), que permiten evaluar la calidad de las oportunidades de gol de un equipo. El uso de **xG** en el análisis de partidos y jugadores ha permitido una mayor capacidad predictiva y una mejor comprensión del rendimiento de los equipos. La industria que más potenció este cambio fue la de las apuestas deportivas, que comenzó a utilizar modelos predictivos para estimar las probabilidades de los partidos. La aparición de empresas como StatsBomb y Opta Sports son claros ejemplos de como la analítica de datos ha crecido en importancia en la industria del fútbol. Tanto es así que el Arsenal y el Brentford de la Premier League poseen sus propias empresas de analítica de datos; StatDNA y Smartodds (Tippett, 2019, p. 37).

El trabajo en desarrollo *Soccer Networks* (Huang et al., n.d.) propone un modelo de red de jugadores para calcular la probabilidad de disparar al arco antes de perder el balón (**PSL**), una métrica poca estudiada. En el paper se demuestra que el **PSL** tiene una alta correlación con el rendimiento del equipo y una gran importancia al nivel del **xG**.

Este trabajo profundiza en el análisis de la métrica **PSL** y propone un análisis probabilístico sobre las componentes del modelo de redes de jugadores y su inferencia en el rendimiento de los jugadores y consecuentemente del equipo. Proponemos una metodología para comparar el rendimiento de jugadores y formaciones de jugadores en base a la métrica **PSL**. Finalmente, desarrollamos un modelo de representación vectorial de los jugadores, llamado **Player2Vec**, para poder utilizarlo en modelos predictivos sobre el rendimiento de los jugadores.

A partir de la pregunta de la investigación, se plantea el problema de encontrar el jugador ideal para un equipo de fútbol. Surge la necesidad de encontrar una métrica para evaluar el impacto de un jugador en el rendimiento de un equipo y cómo definir estos agentes. Además es necesario poder representar concretamente a un Jugador J de forma vectorial para poder utilizarlo en modelos predictivos.

1.1 PSL como métrica de Performance

En el paper en proceso *Soccer Networks* (Huang et al., n.d.) se plantea la descomposición del Gol Esperado (xG) como:

$$xG(A) = P(A) \cdot PSL(A) \cdot SA(A)$$

Donde A es el equipo, $P(A)$ es el número de posesiones del balón, $PSL(A)$ es la probabilidad de patear al arco antes de perder el balón y $SA(A)$ es la probabilidad de que un disparo al arco se convierta en gol. A diferencia de la posesión del balón y la probabilidad de convertir un disparo en gol, $PSL(A)$ no es una métrica comúnmente utilizada en el análisis de fútbol ni existen modelos que la calculen. El paper *Soccer Networks* plantea un modelo de red de jugadores que permite calcular $PSL(A)$ para cada equipo.

1.2 Modelo de Red de Jugadores

Utilizando Cadenas de Markov de Tiempo Continuo (CTMC) se puede calcular la probabilidad de que un equipo pierda el balón antes de patear al arco. En este modelo de red de jugadores se plantea un modelo de 14 estados: 11 jugadores ($J_1 \dots J_{11}$), Ganancia, Pérdida y Disparo.

El grafo presentado en la figura 1 representa el modelo de red de jugadores. Cada nodo representa un estado y cada arista representa una transición entre estados. El nodo verde representa el estado de disparo al arco, el rojo la pérdida del balón, el negro la ganancia del balón por parte del equipo y los azules a los jugadores. Los ejes entre los nodos se representan con una matriz de adyacencia R donde cada valor $r(U, V)$ representa el ratio de transición entre los estados U y V .

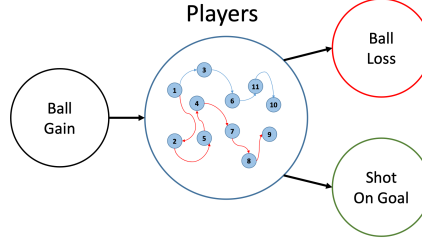


Figure 1: Modelo de Red de Jugadores

$$R = \begin{pmatrix} 0 & r(G, J_1) & \dots & r(G, J_{11}) & 0 & 0 \\ 0 & 0 & \dots & r(J_1, J_{11}) & r(J_1, L) & r(J_1, S) \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & r(J_{11}, J_1) & \dots & 0 & r(J_{11}, L) & r(J_{11}, S) \\ 0 & 0 & \dots & 0 & 1 & 0 \\ 0 & 0 & \dots & 0 & 0 & 1 \end{pmatrix}$$

Los ratios de transición posibles se calculan de la siguiente manera:

$$\begin{aligned} r(G, J_i) &= \frac{\text{Ganancias de } J_i}{\text{Tiempo Jugado por } J_i}, & r(J_i, S) &= \frac{\text{Disparos al arco de } J_i}{\text{Tiempo Jugado por } J_i} \\ r(J_i, S) &= \frac{\text{Disparos al arco de } J_i}{\text{Tiempo Jugado por } J_i}, & r(J_i, J_j) &= \frac{\text{Pases de } J_i \text{ al jugador } J_j}{\text{Tiempo jugado entre } J_i \text{ y } J_j} \end{aligned} \quad (1)$$

A partir de R , la matriz de ratio de acción sobre tiempo jugado (ganancias, pases, disparos o pérdidas), se puede obtener la matriz de transición de estados Q al normalizar sus filas.

Para cada par de estados U y V se define $q(U, V) = \frac{r(U, V)}{\sum_{i=1}^{14} r(U, i)}$

$$Q = \begin{pmatrix} 0 & q(G, J_1) & \dots & q(G, J_{11}) & 0 & 0 \\ 0 & 0 & \dots & q(J_1, J_{11}) & q(J_1, L) & q(J_1, S) \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & q(J_{11}, J_1) & \dots & 0 & q(J_{11}, L) & q(J_{11}, S) \\ 0 & 0 & \dots & 0 & 1 & 0 \\ 0 & 0 & \dots & 0 & 0 & 1 \end{pmatrix}$$

Finalmente a partir de la matriz de probabilidades de transición Q se puede calcular $PSL(A)$ como:

$$PSL(A) = [1, 0, \dots, 0] \cdot (I - T)^{-1} \cdot X \cdot [0, 1]^T$$

Siendo T las probabilidades de transición de los estados transitorios, X las probabilidades de transición de los estados transitorios a los estados absorbentes e I la matriz identidad (Ross, 2019).

A partir de este modelo en el paper *Soccer Networks* se evaluó para una temporada de la Premier League (EPL 2012/13) (*Opta Data from Stats Perform*, n.d.) la diferencia entre los PSL de cada equipo y luego de forma empírica se demuestra como el $PSL(A)$ tiene alta correlación positiva con el rendimiento del equipo por sobre el contrincante. Finalmente hallamos una métrica significativa de rendimiento de un equipo en la métrica PSL . Estos resultados dan a lugar a la investigación de cómo se puede aplicar esta métrica a nivel de jugador y cómo se puede comparar el rendimiento de jugadores en distintos equipos.

Para evaluar el impacto de un jugador J se debe conocer la probabilidad de transición entre J y los otros 13 estados (10 jugadores, Ganancia, Pérdida y Disparo), o bien lograr estimar la probabilidad de transición entre J y los otros 13 estados.

En este trabajo se propone un método probabilístico bayesiano para hallar la Distribución del PSL dada la distribución de probabilidades de transición entre cada uno de los 11 jugadores y los otros 13 estados.

2 Estimación de la Distribución del PSL

Considerando que cada jugador J tiene una distribución *a-priori* para cada uno de los 14 estados, podemos reformular la matriz de ratios de transición como una matriz de variables aleatorias donde cada una se distribuye según dicha distribución *a-priori* del jugador correspondiente.

2.1 Variables Aleatorias para los $r(U, V)$ y PSL por *priors*

Para actualizar la notación, sean $r_{J,V}$ la variable aleatoria que representa el ratio de transición entre el jugador J y el estado V , esto incluye $r_{J,S}$, $r_{J,L}$ y también $r_{G,J}$, así como los r_{J_i,J_j} para $i, j \in [1, 11]$. Luego $r_{J,V} \sim F_x$ la distribución *a-priori* de la variable aleatoria $r_{J,V}$. Finalmente obtenemos, para una formación dada de 11 jugadores, una matriz de variables aleatorias $\mathbf{R}$.

$$\mathbf{R} = \begin{pmatrix} 0 & r_{G,J_1} & \dots & r_{G,J_{11}} & 0 & 0 \\ 0 & 0 & \dots & r_{J_1,J_{11}} & r_{J_1,L} & r_{J_1,S} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & r_{J_{11},J_1} & \dots & 0 & r_{J_{11},L} & r_{J_{11},S} \\ 0 & 0 & \dots & 0 & 1 & 0 \\ 0 & 0 & \dots & 0 & 0 & 1 \end{pmatrix}$$

2.2 Método de Monte Carlo para estimar la distribución del PSL

Dado un equipo A con una formación de 11 jugadores L_A , se busca estimar la distribución del PSL de ese equipo a partir de las distribuciones *a-priori* de los $r(U, V)$ de cada jugador. Para ello, se propone un método de Monte Carlo para muestrear de las distribuciones *a-priori* de los $r(U, V)$ y estimar con ellas la distribución del PSL del equipo A .

De la formación L_A podemos construir la matriz de variables aleatorias $\mathbf{R}$ a partir de las distribuciones *a-priori* de los $r(U, V)$ de cada jugador.

Definimos $\hat{f}_{PSL}^N(L_A)$ como la función distribución de probabilidad empírica de los PSL_i para la formación L_A en base a N simulaciones.

El proceso de Monte Carlo para estimar la distribución del PSL de la formación L_A es el siguiente:

Input: Número de simulaciones N Input: Formación $L_A = \{J_1, J_2, \dots, J_{11}\}$ Output: Distribución del PSL del equipo A 1 $\mathbf{R} \leftarrow$ Construir la matriz de variables aleatorias a partir de las distribuciones <i>a-priori</i> de los $r(U, V)$ de cada jugador; 2 $PSL_i \leftarrow 0$ para $i = 1, 2, \dots, N$; 3 for $i = 1$ to N do 4 $R \leftarrow$ Muestrear de la matriz $\mathbf{R}$ distribuciones <i>a-priori</i> de los $r(U, V)$; 5 $Q \leftarrow$ Normalizar las filas de R; 6 $PSL_i \leftarrow PSL(Q)$; 7 end 8 Estimar la distribución del PSL del equipo A a partir de las N observaciones obtenidas de las simulaciones;
--

Algorithm 1: Simulación del PSL del equipo A

A partir de esta distribución del PSL, se puede realizar comparaciones entre diferentes formaciones de 11 jugadores.

2.3 Comparar el impacto sobre el PSL de dos jugadores en una formación

Para comparar el PSL de dos jugadores en una formación, se propone un análisis que consiste en evaluar el impacto en la distribución del PSL al reemplazar a un jugador por otro en la formación. El proceso para ello es el siguiente:

Se define la Formación $L_A = \{J_1, J_2, \dots, J_{11}\}$ como la formación original del equipo A , donde alguno de los jugadores J_i es el jugador “original”.

Se define el jugador J' a comparar con J_i y la formación $L'_A = (L_A \setminus \{J_i\}) \cup \{J'\}$ como la formación con el jugador J' en lugar de J_i .

Luego, se puede computar $\hat{f}_{PSL}^N(L_A)$ y $\hat{f}_{PSL}^N(L'_A)$ para comparar las distribuciones del PSL de las formaciones L_A y L'_A .

2.4 Comparación de Distribuciones de PSL

En la siguiente sección postulamos una serie de métodos y métricas para comparar distribuciones de PSL de dos formaciones. En orden creciente de complejidad y rigurosidad, proponemos:

1. Comparación de Momentos Estadísticos
2. Dominancia Probabilística
3. Dominancia Estocástica

Para explicar la comparación de distribuciones de PSL, se propone un ejemplo de dos formaciones de 11 jugadores distintas, en una formación L_{MC} se encuentran 10 jugadores del equipo Manchester City (MCI) + Sergio Agüero delantero del mismo equipo y en la otra L_{MC}^{Giroud} los mismos 10 jugadores del MCI + Olivier Giroud delantero del equipo Arsenal.

Se realizó el proceso de Monte Carlo para estimar la distribución del PSL de cada formación a partir de 1000 simulaciones.

2.4.1 Comparación de Momentos Estadísticos

Se puede realizar una comparación por momentos de las distribuciones de PSL de dos formaciones. Se propone comparar la media y la varianza de las distribuciones $\hat{f}_{PSL}^{1000}(L_{MC})$ y $\hat{f}_{PSL}^{1000}(L_{MC}^{\text{Giroud}})$ ya que el método de Monte Carlo nos permite obtener una muestra significativa de las distribuciones. Al no ser distribuciones normales, la skewness y la kurtosis nos proveen información adicional sobre la forma de la distribución.

Table 1: Comparación de momentos de $\hat{f}_{PSL}^{1000}(L_{MC})$ y $\hat{f}_{PSL}^{1000}(L_{MC}^{\text{Giroud}})$

Jugador	Media	Varianza	Desvío Estándar	Skewness	Kurtosis
Aguero	0.130861	0.041160	0.001694	0.554998	0.362611
Giroud	0.134403	0.043310	0.001876	0.580404	0.405658

Para este caso de ejemplo, se observa que la media y la varianza de las distribuciones de PSL de la formación L_{MC} y L_{MC}^{Giroud} son similares, aunque mayores en la formación con Giroud. Además el tercer momento (skewness) nos indica que la distribución de PSL de la formación con Agüero es más sesgada a la izquierda que la de la formación con Giroud. Por último el cuarto momento (kurtosis) nos indica que la $\hat{f}_{PSL}^{1000}(L_{MC}^{\text{Giroud}})$ tiene colas más pesadas que la $\hat{f}_{PSL}^{1000}(L_{MC})$.

2.4.2 Dominancia Probabilística

Otra forma de comparar las distribuciones de PSL de dos formaciones es a través de la dominancia probabilística.

En este caso, se puede calcular la probabilidad de que una muestra aleatoria de una distribución sea mayor que una muestra aleatoria de la otra distribución. De esta forma podemos tomar samples de las distribuciones $\hat{f}_{PSL}^{1000}(L_{MC})$ y $\hat{f}_{PSL}^{1000}(L_{MC}^{Giroud})$ y calcular la probabilidad de que un sample de la formación con Giroud sea mayor que un sample de la formación con Agüero.

Sean $X_{L_{MC}} \sim \hat{f}_{PSL}^{1000}(L_{MC})$ y $X_{L_{MC}^{Giroud}} \sim \hat{f}_{PSL}^{1000}(L_{MC}^{Giroud})$ las variables aleatorias que se distribuyen según las distribuciones de PSL de las formaciones L_{MC} y L_{MC}^{Giroud} respectivamente. Luego para evaluar si la formación con Giroud tiene dominancia probabilística sobre la formación con Agüero, se puede calcular la probabilidad $P(X_{L_{MC}^{Giroud}} > X_{L_{MC}})$.

El algoritmo para calcular la dominancia probabilística es el siguiente:

Input: Distribuciones de PSL $\hat{f}_{PSL}^{1000}(L)$ y $\hat{f}_{PSL}^{1000}(L')$ Output: Probabilidad de que un sample de PSL de la formación L sea mayor que un sample de PSL de la formación con L' <pre> 1 $N \leftarrow 1000$; 2 $M \leftarrow 0$; 3 for $i = 1$ to N do 4 $PSL \leftarrow$ Muestrear de $\hat{f}_{PSL}^{1000}(L)$; 5 $PSL' \leftarrow$ Muestrear de $\hat{f}_{PSL}^{1000}(L')$; 6 if $PSL' > PSL$ then 7 $M \leftarrow M + 1$; 8 end 9 end 10 $P \leftarrow \frac{M}{N}$; </pre>

Algorithm 2: Dominancia Probabilística

Para el caso de ejemplo, se obtuvo que la probabilidad de que un sample de PSL de la formación con Giroud sea mayor que un sample de PSL de la formación con Agüero es $P(X_{L_{MC}^{Giroud}} > X_{L_{MC}}) \approx 0.5423$. De esta forma podemos concluir que la formación con Giroud tiene dominancia probabilística sobre la formación con Agüero.

2.4.3 Comparación de CDFs de las distribuciones de PSL

Otra forma de comparar las distribuciones de PSL de dos formaciones es a través de las funciones de distribución acumulada (CDF). Llamemos $\hat{F}_{PSL}^N(L)$ a la función de distribución acumulada de PSL obtenida a partir de N simulaciones del proceso de Monte Carlo para la formación L . Para una comparación formal de CDFs se propone utilizar el concepto de Dominancia Estocástica.

2.4.4 Dominancia Estocástica

Se puede evaluar la dominancia estocástica entre las CDFs $\hat{F}_{PSL}^{1000}(L_{MC})$ y $\hat{F}_{PSL}^{1000}(L_{MC}^{Giroud})$. La dominancia estocástica es una relación de orden entre dos funciones de distribución acumulada que indica si una distribución es mayor que la otra en todos los puntos.

Específicamente, podemos ver que a partir del umbral $x = 0.05346757$, $\hat{F}_{PSL}^{1000}(L_{MC}^{Giroud})$ tiene **dominancia estocástica parcial** sobre $\hat{F}_{PSL}^{1000}(L_{MC})$ (Bawa, 1982; Vulcano, n.d.).

3 Player2Vec: Embeddings de Jugadores

Para poder representar a cada jugador de forma vectorial, se desarrolló el modelo denominado Player2Vec que permite obtener un embedding de cada jugador en un espacio de n dimensiones. Este modelo hace uso de Node2Vec, que es en sí una adaptación de Word2Vec, una técnica de NLP que permite representar palabras en un espacio vectorial (Grover & Leskovec, 2016; Mikolov et al., 2013).

3.1 Modelado de la EPL 2012/13 como Grafo

A partir de una formación de 11 (Lineup), para un equipo (Team), en un partido (Match), se construye el grafo de la red de jugadores. Llamemos a estos $G_{L,T,M}$ Grafo de Lineup. Sean $l \in L = \{0, 3\}$ las formaciones posibles (en la temporada 12/13 se permitían hasta 3 cambios de jugadores), $t \in T = \{\text{Local}, \text{Visitante}\}$ los equipos que jugaron el partido y $m \in M = \{1, 2, \dots, 380\}$ los partidos de la temporada 12/13 de la EPL.

$$G_{L,T,M} = (V^{L,T,M}, E^{L,T,M})$$

L = Número de Lineup del equipo en el partido

T = Número de Equipo

M = Número de Partido

$$V^{L,T,M} = \{\text{Gain}^{L,T,M}, J_1^{L,T,M}, J_2^{L,T,M}, \dots, J_{11}^{L,T,M}, \text{Loss}^{L,T,M}, \text{Shot}^{L,T,M}\}$$

$$\begin{aligned} E^{L,T,M} = & \{(J_i^{L,T,M}, J_j^{L,T,M}, r(J_i^{L,T,M}, J_j^{L,T,M})) \mid i, j \in [1, 11]\} \\ & \cup \{(\text{Gain}^{L,T,M}, J_i^{L,T,M}, r(\text{Gain}^{L,T,M}, J_i^{L,T,M})) \mid i \in [1, 11]\} \\ & \cup \{(J_i^{L,T,M}, \text{Shot}^{L,T,M}, r(J_i^{L,T,M}, \text{Shot}^{L,T,M})) \mid i \in [1, 11]\} \\ & \cup \{(J_i^{L,T,M}, \text{Loss}^{L,T,M}, r(J_i^{L,T,M}, \text{Loss}^{L,T,M})) \mid i \in [1, 11]\} \end{aligned}$$

Donde cada $J_i^{L,T,M} \mid i \in [1, 11]$ es un nodo que representa a un jugador en el lineup L del equipo T en el partido M . $\text{Gain}^{L,T,M}$ es el nodo que representa la ganancia del balón, $\text{Loss}^{L,T,M}$ la pérdida del balón y $\text{Shot}^{L,T,M}$ el disparo al arco en el lineup L del equipo T en el partido M . Luego sean $J_i \mid i \in [0, 521]$ los jugadores reales de la

temporada 2012/13 de la EPL. Se construye el grafo de la red de jugadores $G_{\text{EPL-12/13}}$ como la unión de todos los grafos de lineup $G^{L,T,M}$.

$$\begin{aligned}
 G_{\text{Full}} = (V, E) &= \bigcup_{L,T,M} G^{L,T,M} \\
 V &= \{J_1, J_2, \dots, J_{521}, \text{Gain}, \text{Loss}, \text{Shot}\} \\
 &\cup \bigcup_{L,T,M} \{J_1^{L,T,M}, J_2^{L,T,M}, \dots, J_{11}^{L,T,M}, G^{L,T,M}, L^{L,T,M}, S^{L,T,M}\} \\
 E &= \bigcup_{L,T,M} E^{L,T,M} \\
 &\cup \{(J_i, J_j^{L,T,M}, r(J_i, J_j^{L,T,M})) \mid i \in [0, 521], j \in [1, 11], L, T, M\} \\
 &\cup \{(\text{Gain}, \text{Gain}^{L,T,M}, 1) \mid L, T, M\} \\
 &\cup \{(\text{Loss}^{L,T,M}, \text{Loss}, 1) \mid L, T, M\} \\
 &\cup \{(\text{Shot}^{L,T,M}, \text{Shot}, 1) \mid L, T, M\}
 \end{aligned}$$

El ratio de transición $r(J_i, J_j^{L,T,M})$ es el tiempo jugado por el Jugador J_i en el lineup L del equipo T en el partido M sobre el tiempo total jugado por el Jugador J_i

$$r(J_i, J_j^{L,T,M}) = \frac{\text{Time Played}_{J_i^{L,T,M}}}{\text{Time Played}_{J_i}}$$

3.2 Implementación

A partir de calcular las matrices de ratios $R^{L,T,M}$ para cada lineup L del equipo T en el partido M generamos el grafo dirigido $G^{L,T,M}$ haciendo uso de la librería `NetworkX` en Python para luego componerlos en G_{Full} , el grafo resultante contiene 37521 nodos y 47338 aristas para la EPL 2012/13.

Para obtener los embeddings de los jugadores, se utilizó la librería `node2vec` en Python, que implementa el algoritmo homónimo. Se configuró el modelo con una longitud de caminata de 16 nodos, 200 caminatas y un tamaño de ventana de 12 nodos. Se entrenaron 2 modelos de embeddings, uno con 64 dimensiones para utilizar en modelos de Deep Learning y otro con 3 dimensiones para exploración gráfica.

Para cada uno de los 37521 nodos se obtuvo un embedding, de los cuales nos quedamos solo con los 521 embeddings de los jugadores reales, estos finalmente son la representación vectorial de cada jugador en el espacio de embeddings.

En el caso de `Player2Vec`, los k random walks resultantes son una secuencia de jugadores y/o estados de juego en un partido de fútbol (Ganancia, Pérdida, Disparo). A modo ilustrativo los siguientes son posibles random walks obtenidos del grafo de la EPL 2012/13:

Random Walk 1: $\text{Gain} \rightarrow \text{Gain}^{L,T,M} \rightarrow J_7^{L,T,M} \rightarrow \dots \rightarrow \text{Shot}^{L,T,M} \rightarrow \text{Shot}$
 Random Walk 2: $J_{93} \rightarrow J_{93}^{L,T,M} \rightarrow J_{15}^{L,T,M} \rightarrow J_{21}^{L,T,M} \rightarrow \text{Loss}^{L,T,M} \rightarrow \text{Loss}$
 $\vdots$
 Random Walk k : $J_{12} \rightarrow J_{12}^{L,T,M} \rightarrow J_{13}^{L,T,M} \rightarrow J_{33}^{L,T,M} \rightarrow \text{Shot}^{L,T,M} \rightarrow \text{Shot}$

La cantidad de random walks k así como los otros hiperparametros del modelo de Node2Vec fueron seleccionados de forma empírica observando el resultado de los embeddings obtenidos.

4 Modelo predictivo de Distribuciones de Ratios de Transición ($r(U, V)$)

El trabajo de Player2Vec nos permite obtener embeddings de jugadores que capturan las relaciones entre ellos en un espacio vectorial. A partir de estos embeddings, se propone un modelo predictivo de las distribuciones de $r(U, V)$.

Dado un jugador J_i , se obtiene su embedding $E(J_i)$ a partir del modelo de Player2Vec. Para este J_i , se busca predecir los estadísticos Media y Varianza de las distribuciones de ratios de transición de J_i ($r(\text{Gain}, J_i)$, $r(J_i, \text{Shot})$, $r(J_i, \text{Loss})$, $r(J_i, J_j)$, $r(J_j, J_i)$)

Sean μ_{Gain, J_i} y $\sigma_{\text{Gain}, J_i}$ la media y la varianza de la distribución de $r(\text{Gain}, J_i)$ respectivamente. Análogamente para las otras distribuciones.

Asumiendo normalidad para simplificación podemos decir que $r(U, V) \sim \mathcal{N}(\mu_{U,V}, \sigma_{U,V})$.

4.1 Modelo

El modelo planteado es una Red Neuronal (NN), donde la salida de $NN(E(J_i))$ es un vector de longitud 10 con las medias y varianzas de las distribuciones de ratios de transición del jugador J_i . La función de pérdida a minimizar es una ponderación de la Divergencia de Jensen-Shannon (JSD) entre la distribución real y la predicha para cada estadístico formulada en 2), donde $D_{KL}(p||q)$ es la divergencia de Kullback-Leibler entre las distribuciones p , q y $m = \frac{p+q}{2}$ formulada en 3).

$$\mathcal{L} = \frac{1}{5} \sum_{U \in \{\text{Gain}, J_i, J_j\}} \sum_{V \in \{J_i, \text{Shot}, \text{Loss}, J_j\} \setminus U} \text{JSD}(\mathcal{N}(\mu_{U,V}, \sigma_{U,V}) || \mathcal{N}(\mu_{U,V}^{\text{pred}}, \sigma_{U,V}^{\text{pred}}) \quad (2)$$

$$\text{JSD}(p||q) = \frac{1}{2} D_{KL}(p||m) + \frac{1}{2} D_{KL}(q||m) \quad (3)$$

4.2 Datos e Implementación

Para entrenar el modelo, se utilizó un dataset de eventos de partidos de la temporada 2012/13 de la EPL. Se separó la temporada en dos mitades (190 partidos cada una). Con la primera mitad, se construyó un grafo de la red de jugadores $G_{\text{First Half}}$ y se obtuvieron los embeddings de los jugadores con Player2Vec (`dimensions=64`, `window=30`, `num_walks=500`, `walk_length=40`). Con la segunda mitad, se obtuvieron las distribuciones de ratios de transición de los jugadores.

Luego, de la segunda mitad de la temporada se obtuvieron las medias y varianzas de las distribuciones de ratios de transición de los jugadores. Un 80% de los datos se utilizó para entrenar el modelo y un 20% para test. Específicamente los 521 jugadores de la temporada 2012/13 de la EPL se dividieron en 416 para entrenamiento y 105 para test.

El modelo se implementó en PyTorch. Se utilizó una red neuronal con la siguiente arquitectura:

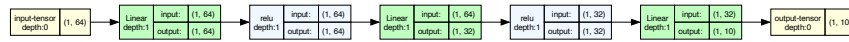


Figure 2: Arquitectura del Modelo Base

La función de pérdida utilizada fue la Divergencia de Jensen-Shannon (JSD) entre las distribuciones reales y predichas ponderando las 5 distribuciones a estimar. Se utilizó la implementación de Divergencia de Kullback-Leibler de PyTorch `nn.KLDivLoss` en la implementación del módulo `JSD` hallado en el foro de PyTorch(PyTorch Forums, 2022) y este luego se utilizó para la función de pérdida en el entrenamiento del modelo.

Con la librería `Hyperopt` (Bergstra et al., 2015), se realizó una búsqueda de hiperparámetros bayesiana. Se evaluaron distintas combinaciones de hiperparámetros y arquitecturas de NN, y se seleccionó la que mejor rendimiento presentó en el conjunto de validación.

El modelo final seleccionado obtuvo un rendimiento aceptable en el conjunto de validación, con una pérdida de 0.008916, aún mejor que la pérdida de Train del modelo sin tune de hiperparametros cuya perdida fue de $\mathcal{L}_m = 0.01505$. Se procedió a evaluar el modelo en el conjunto de testeo y se obtuvo una pérdida de $\mathcal{L}_{mt} = 0.014493$.

El entrenamiento de los modelos se realizó en una máquina con las siguientes características: Macbook Pro M1 2020, Procesador: Apple M1, Memoria RAM: 16 GB, GPU: 8-core GPU, 16-core Neural Engine, OS: macOS Sequoia 15.0.1 (24A348), Python 3.10.14, PyTorch 2.2.2 MPS (Metal Performance Shaders)

4.3 Comparación contra *priors*

Para evaluar la capacidad predictiva de nuestro modelo, comparamos los resultados obtenidos con los *priors* de las distribuciones de ratios de transición de los jugadores.

Los *priors* se obtuvieron a partir de las distribuciones de ratios de transición de los jugadores en la primera mitad de la temporada 2012/13 de la EPL. Asumiendo que las distribuciones de ratios de transición de los jugadores en la segunda mitad de la temporada son similares a las de la primera mitad, evaluamos su capacidad predictiva con la misma función de pérdida JSD ponderada, obteniendo los siguientes resultados: $\mathcal{L}_{priors} = 0.0353$, $\mathcal{L}_m = 0.01505$ y $\mathcal{L}_{mt} = 0.00795$.

El modelo logra una mejora significativa en la capacidad predictiva de las distribuciones de ratios de transición de los jugadores en comparación con asumir igualdad de distribuciones entre las dos mitades de la temporada. En concreto, nuestro modelo logra reducir la pérdida en un $\sim 58.99\%$ en comparación con asumir *priors* bayesianos.

5 Discusión

El presente trabajo presenta varias limitaciones y posibles áreas de mejora que deben ser consideradas para futuros desarrollos. En primer lugar, la cantidad de datos de entrenamiento disponibles es limitada, ya que se utilizó solo una temporada de la English Premier League (EPL). Extender el análisis a múltiples temporadas y ligas adicionales permitiría contar con un conjunto de datos más amplio y representativo. Esto no solo mejoraría la capacidad del modelo para generalizar, sino que también fortalecería su validez externa al ser probado en diferentes contextos competitivos y estilos de juego. Explorar mejor el espacio de Embeddings podría proporcionar información valiosa para la interpretación de los resultados.

Es importante destacar que el modelo carece de ciertas variables contextuales clave. No se han incluido factores como el impacto del equipo rival, las estrategias del entrenador, las condiciones específicas de la fecha (por ejemplo, clima o nivel de fatiga acumulada), ni el efecto de la localía. La incorporación de estas variables podría enriquecer el análisis y permitir una comprensión más completa del entorno competitivo en el que se desarrolla el rendimiento del jugador.

6 Conclusiones

En este trabajo se presentó un modelo predictivo de distribuciones de ratios de transición de jugadores de fútbol basado en un modelo novedoso de embeddings de jugadores. El modelo fue entrenado y validado utilizando datos de eventos de partidos de la temporada 2012/13 de la EPL y se evaluó su capacidad para predecir las distribuciones de ratios de transición de jugadores. Los resultados obtenidos muestran que el modelo es capaz de predecir mejor que los *priors* las distribuciones de ratios de transición de los jugadores, lo que sugiere que puede ser una herramienta valiosa para la evaluación de transferencias en el fútbol profesional.

7 Disponibilidad de datos y disponibilidad del código

Los datos utilizados en este proyecto se obtuvieron a través de OPTA por lo que no son de acceso público. En concreto se utilizaron datos de eventos de partidos de la temporada 2012/13 de la EPL.

El código fuente del proyecto se encuentra en <https://github.com/juanisil/TDVIII-proyecto-final> y está disponible bajo la licencia MIT. El código incluye la implementación del modelo de Player2Vec, el modelo predictivo de distribuciones de ratios de transición, la implementación de las redes de jugadores y los scripts para la obtención de los datos de eventos de partidos.

8 Referencias bibliográficas

- Baumer, B. (2015). *Sabermetric revolution : Assessing the growth of analytics in baseball*. Univ Of Pennsylvania Pr.
- Bawa, V. S. (1982). Stochastic dominance: A research bibliography. *Management Science*, 28, 698–712. <https://doi.org/10.1287/mnsc.28.6.698>
- Bergstra, J., Komer, B., Eliasmith, C., Yamins, D., & Cox, D. D. (2015). Hyperopt: A python library for model selection and hyperparameter optimization. *Computational Science & Discovery*, 8, 014008. <https://doi.org/10.1088/1749-4699/8/1/014008>
- Goldsberry, K. (2012). CourtVision : New visual and spatial analytics for the NBA. *Undefined*. <https://www.semanticscholar.org/paper/CourtVision-%3A-New-Visual-and-Spatial-Analytics-for-Goldsberry/46e4a7271de62e9118625dec935c4aef1bc0ea74>
- Green, S. (2012). *Assessing the performance of premier league goalscorers*. Stats Perform. <https://www.statsperform.com/resource/assessing-the-performance-of-premier-league-goalscorers/>
- Grover, A., & Leskovec, J. (2016). *node2vec: Scalable feature learning for networks*. arXiv.org. <https://arxiv.org/abs/1607.00653>
- Huang, E., Segarra, S., Gallino, S., & Ribeiro, A. (n.d.). *How to find the right player for your soccer team?*
- Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). *Efficient estimation of word representations in vector space*. arXiv.org. <https://arxiv.org/abs/1301.3781>
- Opta data from stats perform*. (n.d.). Stats Perform. <https://www.statsperform.com/opta/>
- PyTorch Forums, A. N. K. -. (2022). *Jensen shannon divergence*. PyTorch Forums. <https://discuss.pytorch.org/t/jensen-shannon-divergence/2626/13>
- Ross, S. M. (2019). *Introduction to probability models*. Academic Press.
- Tippett, J. (2019). *The expected goals philosophy: A game-changing way of analysing football*. Independently Published.
- Vulcano, G. (n.d.). *Decision under risk - module IV - NYU stern - master of science in business analytics*.