



# UNIVERSIDAD TORCUATO DI TELLA

## Cómo encontrar el mejor jugador para tu Equipo de Fútbol

Tomás Glauberman, Ignacio Pardo, Juan Ignacio Silvestri

### Abstract

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 modelos de juegos de suma cero en fútbol (Hirotsu y Wright, 2006) 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 y Zimbalist, 2014; Wolf, 2015). La introducción de herramientas analíticas avanzadas ha producido resultados positivos para muchos equipos, lo que subraya 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 (Huang et al., n.d.), planteamos un proceso para comparar el impacto que tienen los jugadores sobre la performance del equipo. En un enfoque bayesiano, logramos formular una metodología para estudiar la distribución de la performance de un equipo. Además, desarrollamos un modelo de machine learning llamado Player2Vec sobre el modelo teórico de redes de jugadores para representar a cada jugador. De esta manera podemos hallar jugadores similares para luego comparar su rendimiento resultante en un nuevo equipo.

## Contenidos

0.1	Agradecimientos	4
0.2	Abstract	4
0.3	Introducción	4
0.4	Motivación Justificación del tema	4
0.5	Objetivos de Proyecto	4
0.6	Definición del problema	5
0.6.1	PSL como métrica de Performance	5
0.6.2	Modelo de Red de Jugadores	5
0.6.3	Modelo Predictivo de probabilidades de transición	6
0.6.4	Test de Sensibilidad sobre PSL	8
0.6.5	Modelo Predictivo sobre $r(J, S)$	8
0.7	Player2Vec: Embeddings de Jugadores	8
0.7.1	Node2Vec	9
0.7.2	Player2Vec	9
0.7.3	Exploración de los Embeddings	11
0.8	Hipótesis	11
0.9	Marco teórico	11
0.10	Marco metodológico	11
0.11	Resultados	13
0.12	Discusión	13
0.13	Conclusiones & Recomendaciones {#conclusiones-&-recomendaciones}	13
0.14	Referencias bibliográficas	13

0.15 Apéndices: Tablas, figuras, anexos {#apéndices:-tablas,-figuras,-anexos} . . . . . 13

**Profesor: Pablo Nussenbaum y Gustavo Vulcano**  
**CABA, Argentina. Diciembre 2024**

Escuela de Negocios Licenciatura en Tecnología Digital

## **Contenidos**

## 0.1 Agradecimientos

Este trabajo no hubiera sido posible sin la ayuda de los profesores Gustavo Vulcano (Escuela de Negocios, Universidad Torcuato Di Tella) y Santiago Gallino (The Wharton School, University of Pennsylvania). Además queremos agradecer a Ignacio Vigilante (TIC - Escuela ORT) y Tomás Spognardi (Exactas - UBA) por sus aportes al modelo de Player2Vec y al PSL Bayesiano respectivamente.

## 0.2 Abstract

Un resumen breve (entre 250 y 300 palabras) que debe incluir los puntos clave del trabajo: el problema, los objetivos, la metodología, los resultados más relevantes y las conclusiones principales. Está orientado a brindar una visión general del trabajo.

## 0.3 Introducción

Presenta el tema del trabajo, su contexto y la importancia. Aquí se debe captar el interés del lector, explicando brevemente los aspectos más importantes que se desarrollarán.

## 0.4 Motivación Justificación del tema

Explica **por qué el tema elegido es relevante**, tanto a nivel académico como práctico. Debe argumentar la importancia del trabajo para el campo de estudio o la sociedad. Similar a lo que se completó en el formulario de licitación de proyectos iniciales.

## 0.5 Objetivos de Proyecto

Define claramente **qué es lo que se busca lograr con el proyecto/investigación**. Se suelen dividir en objetivos generales (el propósito principal) y específicos (pasos concretos para alcanzarlo).

Nuestro objetivo general es el de lograr predecir cómo performaría un jugador en un nuevo equipo,

## 0.6 Definición del problema

Desarrolla el problema o la pregunta de investigación que motiva el proyecto/investigación. Aquí se describe en detalle cuál es la situación o fenómeno que se desarrollará/investigará.

A partir de la pregunta de la investigación, se plantea el problema de encontrar el jugador ideal para un equipo de fútbol. En un comienzo nos encontramos planteando como definir *performance* de un jugador y cómo compararla con otros jugadores. Surgió la necesidad de encontrar una métrica evaluar el impacto de un jugador en el rendimiento de un equipo y como definir estos agentes. Además es necesario poder representar concretamente a un Jugador  $J$ .

### 0.6.1 PSL como métrica de Performance

En el paper en proceso *How to Find the Right Player for your Soccer Team? (Huang et al.)* 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 la posesión del balón,  $PSL(A)$  es la probabilidad 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 plantea un modelo de red de jugadores que permite calcular  $PSL(A)$  para cada equipo.

### 0.6.2 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.

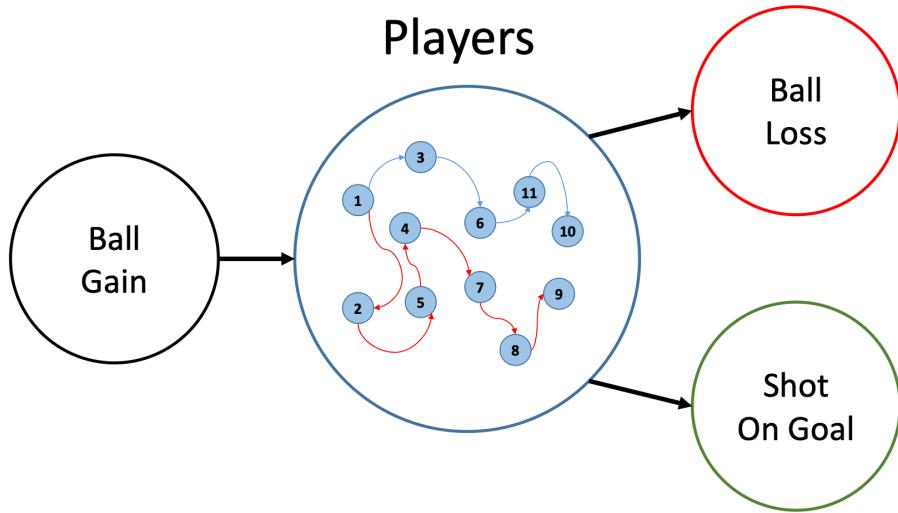


Figure 1: Modelo de Red de Jugadores

El grafo presentado en la figura 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 y el azul la ganancia del balón por parte de un jugador. Los ejes entre los nodos se representan con una matriz de adyacencia  $R$  donde cada valor  $r(U, V)$  representa la ratio de transición entre los estados  $U$  y  $V$ .

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

A partir de la matriz de ratio de acción sobre tiempo jugado  $R$  (ganancias, pases a otro jugador, disparos o pérdidas) se puede obtener la matriz de transición de estados  $Q$  para el CMTG de normalizar las filas de  $R$ :

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.

A partir de este modelo en el paper se evaluó para una temporada de la Premier League (EPL 2012/13) 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 hayamos una métrica significativa de rendimiento de un equipo en la métrica  $PSL$ . Sin embargo, da a lugar a la investigación de como se puede aplicar esta métrica a nivel de jugador y como se puede comparar el rendimiento de jugadores en distintos equipos.

Para evaluar el impacto de un jugador  $J$  se debe, o bien 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.

### 0.6.3 Modelo Predictivo de probabilidades de transición

En un comienzo se planteó desarrollar un modelo predictivo para estimar las ratios de transición entre los estados. Optamos por buscar predecir los ratios  $r$  y no las probabilidades de transición  $q$  ya que la normalización no es igual en cada instancia de  $R$ . Mas concretamente buscamos estimar la función  $f$  que mapea los estados  $U$  y  $V$  a la ratio de transición  $r(U, V)$ .

$$\hat{r}(U, V) = f(U, V, \theta)$$

Comenzamos armando un modelo para predecir únicamente los ratios de pases  $r(J_i, J_j)$  entre un jugador  $J_i$  y otro jugador  $J_j$ . Lo que correspondiera a los siguientes valores de la matriz  $R$ :

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

Para poder utilizar un modelo de machine learning tradicional necesitamos de poder representar a cada jugador  $J$  de forma vectorial. Armamos un vector de métricas agregadas para un jugador al momento del partido a predecir. Estas métricas incluyen la cantidad de pases, disparos, goles, pérdidas, etc. sobre el total de tiempo jugado, ademas de el equipo en el que juega.

$$J = [\text{Passes}/90, \text{Shots}/90, \text{Goals}/90, \text{Losses}/90, \text{Time Played}, \text{Team ID}]$$

Para el modelo predictivo comenzamos utilizando un modelo de XGBoost para la regresión pero rápidamente observamos que por la naturelza de arbol al predecir con la media de las observaciones por hoja las predicciones resultaban casi discretas, por lo que viramos a explorar un modelo de regresión lineal para predecir los ratios de pases entre jugadores.

Para validar elegimos separar de forma temporal los 380 partidos de la temporada 2012/13 de la EPL: los primeros 269 partidos de entrenamiento; los últimos 111 de test ( $\mu + 2/3\sigma$ ). Ademas para construir el dataset, elegimos agarrar parejas de jugadores de los partidos de Train y removerlos de los mismos para poder en Test predecir ratios de transición entre jugadores que no se vieron en Train.

Luego de entrenar el modelo, para cada instancia de test obtuvimos la matriz de ratios de transición  $R$  y calculamos el PSL real, para luego predecir la matriz de transición  $\hat{R}$  y calcular el PSL predicho. Finalmente calculamos el coeficiente de correlación de Pearson entre el PSL real y el PSL predicho.

En el siguiente gráfico podemos observar como a pesar de predecir muy pobre los ratios de transición al resultar en un coeficiente de correlación de Pearson entre los  $r(J_i, J_j)$  y los  $\hat{r}(J_i, J_j)$  de 0.12, sin embargo al comparar el PSL real del PSL calculado a partir de  $\hat{R}$  se obtiene un coeficiente de correlación de Pearson de 0.85.

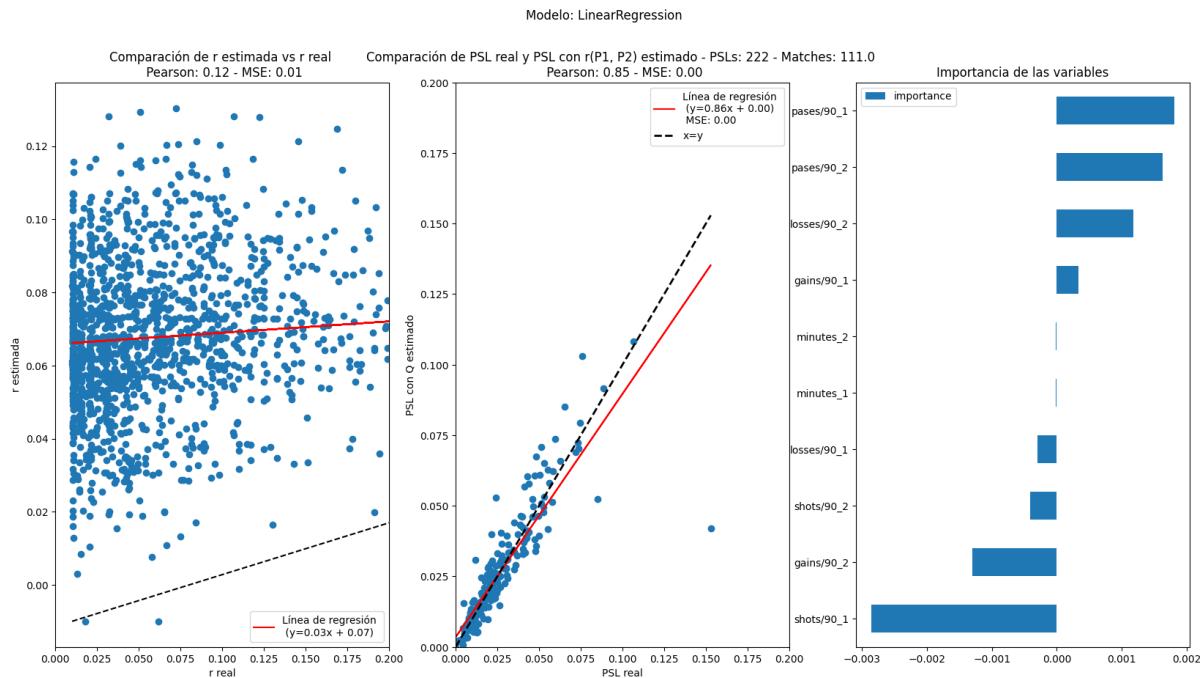


Figure 2: Resultados Modelo de Regresión Lineal

El modelo planteado no es capaz de predecir los ratios de transición, y a pesar de que desarrollamos otros modelos como XGBoost para regresión, Redes Neuronales y Redes Neuronales Probabilísticas (PNNs) no es posible predecir los ratios de transición entre los estados a partir de las métricas de los jugadores. Esto se debe principalmente a la cantidad de datos y la poca relación entre ellos. Al evaluar como resolver la predicción de los  $r(J_i, J_j)$  decidimos observar como cada ratio de transición afecta al PSL.

#### 0.6.4 Test de Sensibilidad sobre PSL

#### 0.6.5 Modelo Predictivo sobre $r(J, S)$

Luego de lo observado con el Test de Sensibilidad sobre PSL, decidimos cambiar el enfoque de la predicción de los ratios de transición entre jugadores a la predicción de los ratios de transición entre jugadores y el estado de disparo al arco. Esto se debe a que al observar la matriz de ratios de transición  $R$  se observa que los ratios de transición entre jugadores y el estado de disparo al arco son los que más afectan al PSL.

El nuevo modelo se enfoca en la siguiente sección de la matriz  $R$ :

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

Para el vector de los jugadores  $J$  se agregó tambien la posición en la que juega (Arquero, Defensor, Mediocampista, Delantero) one-hot-encoded.

Luego se entrenó un modelo de XGBoost para Regresión con el mismo split de Train y Test. Se logró obtener un mejor resultado sobre la predicciones de Train en comparación al modelo anterior, sin embargo al evaluar en Test. Se obtuvo un coeficiente de correlación de Pearson de 0.95 entre los  $r(J_i, S)$  y los  $\hat{r}(J_i, S)$  en Train, pero de 0.08 en Test.

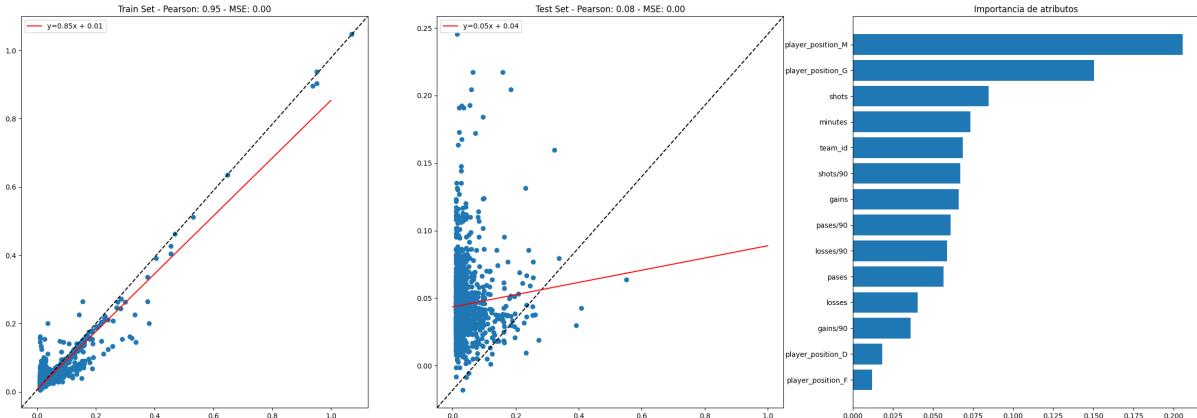


Figure 3: Resultados Modelo de XGBoost

Este resultado junto al del modelo de predicción de ratios de pases nos llevó a buscar una mejor representación vectorial de los jugadores. En la sección de Player2Vec se explica el modelo utilizado para obtener un vector de representación (embedding  $E$ ) de cada jugador. Con este embedding por construimos una red neuronal, el modelo resultante  $f(E_J, \text{partido})$  dado el embedding de los jugadores y el partido predice los ratios de transición entre jugadores y el estado de disparo al arco.

## 0.7 Player2Vec: Embeddings de Jugadores

Para poder representar a cada jugador de forma vectorial, se utilizó un modelo de Player2Vec. 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).

Un embedding es una representación numérica de objetos en un espacio de  $n$  dimensiones, donde propiedades o relaciones similares se preservan. En el contexto de jugadores, un embedding transforma las características de cada jugador en un vector de números, de tal manera que jugadores con comportamientos o atributos similares estén más cerca en este espacio vectorial. Esto facilita que modelos como redes neuronales aprendan patrones complejos a partir de estas representaciones compactas.

### 0.7.1 Node2Vec

Node2Vec es un algoritmo que aprende representaciones vectoriales (embeddings) para nodos en un grafo, preservando tanto las relaciones locales como las globales entre ellos. Utiliza técnicas de random walks para capturar el contexto de cada nodo, balanceando entre explorar nodos cercanos y lejanos. Estos embeddings son útiles para tareas de machine learning sobre grafos, ya que capturan de forma eficiente las interacciones entre nodos en el grafo.

### 0.7.2 Player2Vec

Player2Vec es una adaptación de Node2Vec para representar jugadores de fútbol en un espacio vectorial. En este caso, los nodos del grafo representan jugadores, y las aristas entre ellos reflejan la interacción entre los jugadores en partidos de fútbol. A partir de los datos de eventos de partidos (pases, disparos, goles, etc.), se construye un grafo donde los nodos son jugadores y las aristas representan la frecuencia de interacción entre ellos.

**0.7.2.1 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)
- $t \in T = \{\text{Local, Visitante}\}$  los equipos que jugaron el partido.
- $m \in M = \{1, 2, \dots, 380\}$  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

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

$$E^{L,T,M} = \{(J_i^{L,T,M}, J_j^{L,T,M}, r(J_i^{L,T,M}, J_j^{L,T,M})) \mid i, j \in [1, 11]\}$$

$$\cup \{(\text{Gain}^{L,T,M}, J_i^{L,T,M}, r(\text{Gain}^{L,T,M}, J_i^{L,T,M})) \mid i \in [1, 11]\}$$

$$\cup \{(J_i^{L,T,M}, \text{Shot}^{L,T,M}, r(J_i^{L,T,M}, \text{Shot}^{L,T,M})) \mid i \in [1, 11]\}$$

$$\cup \{(J_i^{L,T,M}, \text{Loss}^{L,T,M}, r(J_i^{L,T,M}, \text{Loss}^{L,T,M})) \mid i \in [1, 11]\}$$

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  $M$ .

En la figura se visualiza un ejemplo de un grafo de lineup  $G^{L,T,M}$  genérico con los ejes  $r(J_1^{L,T,M}, U)$  resaltados.

Luego sean: -  $J_i \mid i \in [0, 522]$  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  $G^{L,T,M}$ .

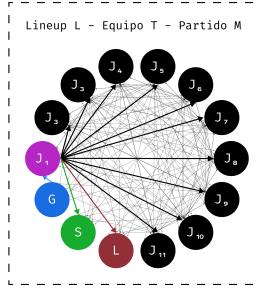


Figure 4: Grafo de Lineup

$$\begin{aligned}
 G_{\text{Full}} &= (V, E) = \bigcup_{L, T, M} G^{L, T, M} \\
 V &= \{J_1, J_2, \dots, J_{522}, \text{Gain}, \text{Loss}, \text{Shot}\} \\
 &\cup \bigcup_{L, T, M} \{J_1^{L, T, M}, J_2^{L, T, M}, \dots, J_{11}^{L, T, M}, G^{L, T, M}, L^{L, T, M}, S^{L, T, M}\} \\
 E &= \bigcup_{L, T, M} E^{L, T, M} \\
 &\cup \{(J_i, J_j^{L, T, M}, r(J_i, J_j^{L, T, M})) \mid i \in [0, 522], j \in [1, 11], L, T, M\} \\
 &\cup \{(\text{Gain}, \text{Gain}^{L, T, M}, 1) \mid L, T, M\} \\
 &\cup \{(\text{Loss}^{L, T, M}, \text{Loss}, 1) \mid L, T, M\} \\
 &\cup \{(\text{Shot}^{L, T, M}, \text{Shot}, 1) \mid L, T, M\}
 \end{aligned}$$

El ratio de transición  $r(J_i, J_i^{L, T, M})$  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  $J_i$

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

La siguiente figura es una visualización de una instancia de un Equipo en un Partido con sus lineups. En este caso el equipo hizo dos cambios en el partido ( $J_4$  por  $J_{12}$  y  $J_2$  por  $J_{13}$ ). Se puede observar como los jugadores reales  $J_4$  y  $J_{12}$  se encuentran representados por el mismo nodo  $J_4^{L, T, M}$  y lo mismo para  $J_2$  y  $J_{13}$  con  $J_2^{L, T, M}$  para sus respectivos lineups. El resto de los nodos de jugadores reales mantienen su identidad en los grafos de lineups.

El grafo resultante de la composición de todos los grafos de lineup  $G_{\text{Full}}$  se puede comprender mejor en la siguiente visualización:

Donde al igual que en la figura anterior, los nodos de jugadores reales se encuentran representados por los nodos de los lineups en los que participaron.

**0.7.2.2 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  $G_{\text{Full}}$ , el grafo resultante contiene 37521 nodos y 47338 aristas.

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.

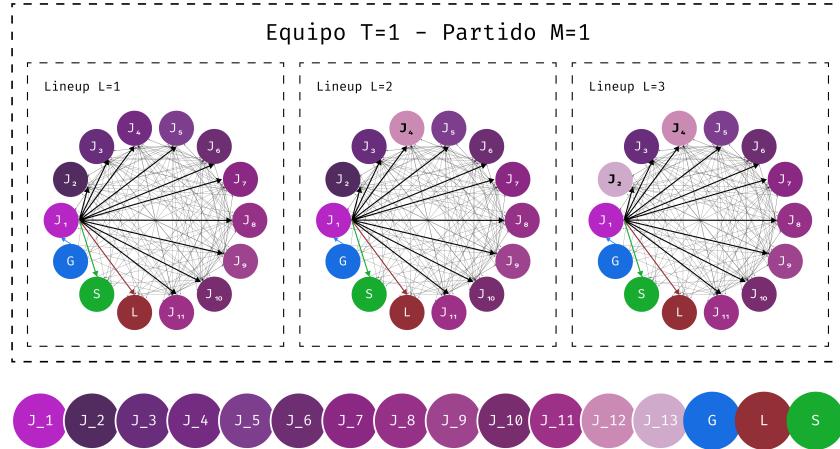


Figure 5: Grafo de Jugadores

Finalmente para cada uno de los 37521 nodos se obtuvo un embedding, de los cuales nos quedamos solo con los 522 embeddings de los jugadores reales. Estos finalmente son la representación vectorial de cada jugador en el espacio de embeddings.

### 0.7.3 Exploración de los Embeddings

## 0.8 Hipótesis

En los casos en que corresponda, la hipótesis es una propuesta que se someterá a prueba a lo largo de la investigación, basada en el planteamiento del problema.

## 0.9 Marco teórico

- Proyectos de Investigación:** Aquí se exponen las teorías, antecedentes y conceptos clave relacionados con el tema de estudio. Sirve para fundamentar el trabajo con bases teóricas y estudios previos. Se deben incluir metodologías de desarrollo de software (ágil, Scrum, etc.), paradigmas de programación, tecnologías y frameworks utilizados (por ejemplo, React, Node.js, bases de datos SQL o NoSQL)
- Proyectos de Desarrollo:** En esta sección, se debe exponer la base conceptual y técnica que sustenta el proyecto. Esto incluye una revisión de metodologías de desarrollo de software (ágil, Scrum, etc.), paradigmas de programación, tecnologías y frameworks utilizados (por ejemplo, React, Node.js, bases de datos SQL o NoSQL). También se pueden mencionar investigaciones previas o estudios de casos relevantes, siempre enfocándose en cómo estas teorías y tecnologías se relacionan con el proyecto.

## 0.10 Marco metodológico

- Proyectos de Investigación:** Describe el enfoque metodológico utilizado en la investigación, es decir, cómo se va a llevar a cabo el estudio. Incluye el diseño de la investigación, las técnicas de limpieza de datos (si hubo) y el análisis que se hará. Se debe incluir un flujo o diagrama de la arquitectura de la solución planteada.
- Proyectos de Desarrollo:** Describe el proceso de desarrollo seguido durante el proyecto. Especifica cómo se organizaron los sprints, las iteraciones, o las fases del proyecto. Incluye también las herramientas empleadas para la gestión del proyecto (Jira, Trello, GitHub) y el enfoque en las pruebas de software para asegurar la calidad del producto final. Se debe incluir un flujo o diagrama de la arquitectura de la solución planteada.

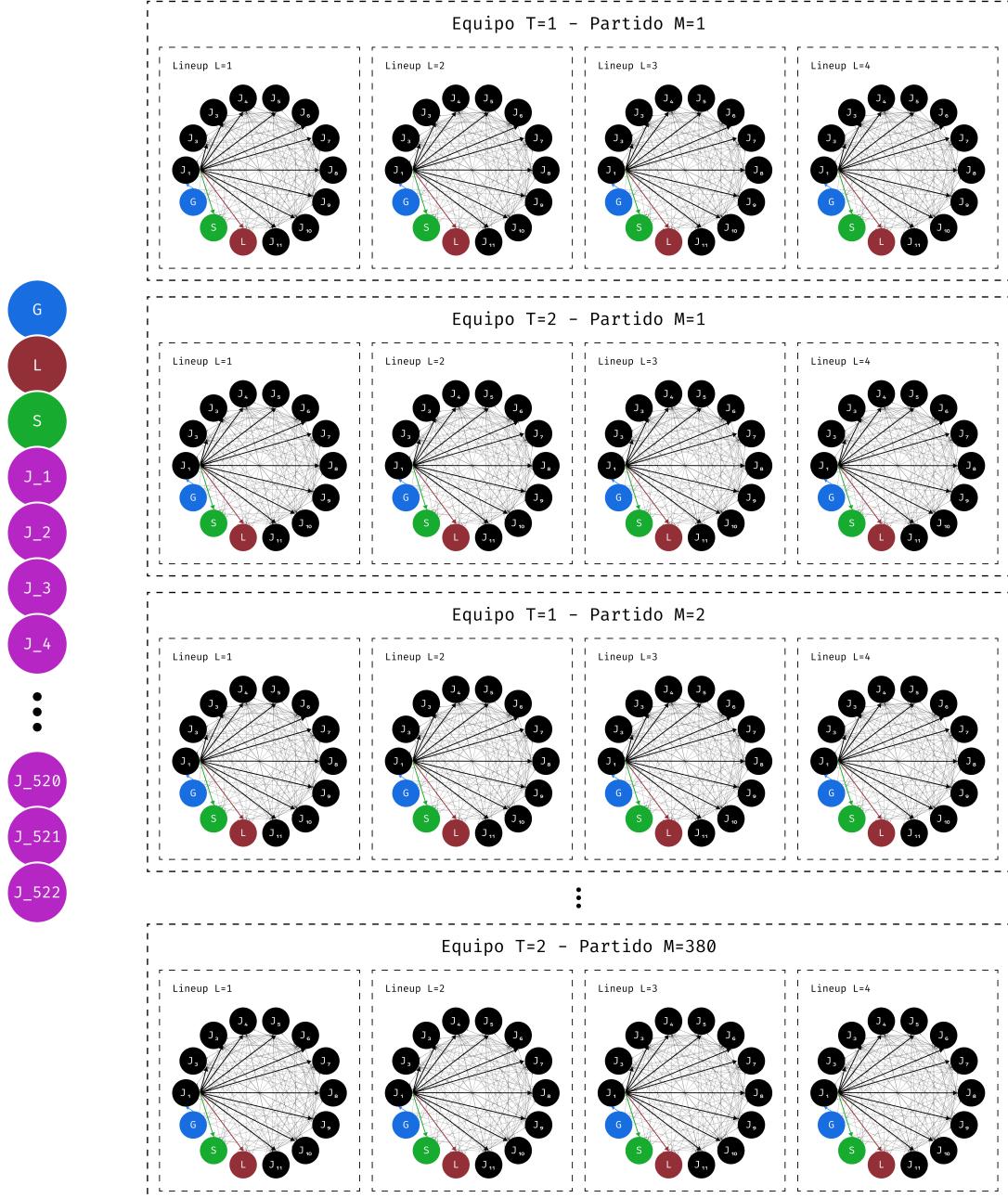


Figure 6: Grafo de Jugadores Completo

## 0.11 Resultados

- **Proyectos de Investigación:** Se presentan los datos o hallazgos obtenidos en la investigación de forma clara y organizada. En este apartado no se deben interpretar los resultados, sólo exponerlos.
- **Proyectos de Desarrollo:** Aquí se detallan los entregables del proyecto de software. Esto puede incluir versiones funcionales del software, demostraciones de características clave, documentación técnica, y análisis de los tiempos de desarrollo o eficiencia lograda. También se pueden incluir métricas como la satisfacción del cliente, el rendimiento del sistema, la escalabilidad o la compatibilidad multiplataforma, según la naturaleza del software desarrollado.

## 0.12 Discusión

- **Proyectos de Investigación:** Aquí se interpretan los resultados obtenidos, comparándolos con la literatura revisada en el marco teórico. Se analizan los hallazgos y se discute su relevancia.
- **Proyectos de Desarrollo:** En esta sección, se realiza un análisis crítico de los resultados obtenidos. Compara las metas iniciales del proyecto con los entregables finales y evalúa si las expectativas fueron cumplidas. Reflexiona sobre los desafíos técnicos encontrados, como problemas de compatibilidad, rendimiento, o integración de APIs, y cómo fueron solucionados. Además, discute el impacto potencial del software en la empresa privada para la cual fue desarrollado, evaluando su viabilidad, utilidad, y posibles mejoras para futuras versiones.

## 0.13 Conclusiones & Recomendaciones {#conclusiones-&-recomendaciones}

Se sintetizan los puntos más importantes del estudio, haciendo énfasis en si se cumplieron los objetivos y qué se aprendió a partir de los resultados obtenidos.

También, se sugieren posibles líneas de investigación futuras o iteraciones del proyecto, aplicaciones prácticas de los resultados o recomendaciones para la implementación de los hallazgos en la vida real.

## 0.14 Referencias bibliográficas

- 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>

## 0.15 Apéndices: Tablas, figuras, anexos {#apéndices:-tablas,-figuras,-anexos}

Se incluyen materiales adicionales como gráficos, tablas, cuestionarios, o documentos que sean relevantes pero no forman parte del cuerpo principal de la tesis.

## Índice de Figuras

1	Modelo de Red de Jugadores . . . . .	5
2	Resultados Modelo de Regresión Lineal . . . . .	7
3	Resultados Modelo de XGBoost . . . . .	8
4	Grafo de Lineup . . . . .	10
5	Grafo de Jugadores . . . . .	11
6	Grafo de Jugadores Completo . . . . .	12

## Índice de Tablas